



StatView[®]

StatView
Reference

SAS[®]

Copyright

Copyright © 1998 by SAS Institute Inc. Second edition. First printing, March 1998.

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher, SAS Institute Inc.

Information in this document is subject to change without notice. The software described in this document is furnished under the license agreement packaged with the software media. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

StatView® and SAS® are registered trademarks of SAS Institute Inc. in the USA and other countries. All trademarks above are registered trademarks or trademarks of SAS Institute Inc. The symbol “®” indicates USA registration. Other brand and product names are trademarks or registered trademarks of their respective companies.

Technology License Notices

Mac2Win © software 1990–94 Altura Software Inc. All rights reserved. Mac2Win® is a registered trademark of Altura Software, Inc.

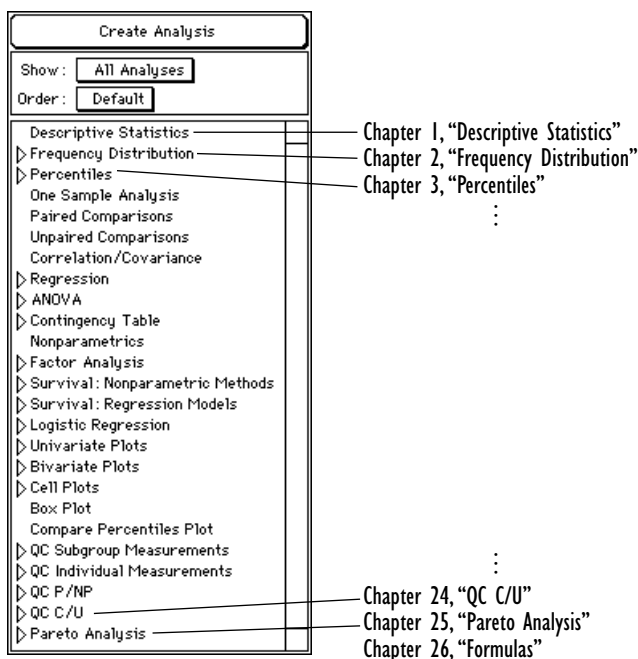
Portions of this software are copyrighted by Apple Computer Inc. Apple® and Macintosh® are registered trademarks of Apple Computer Inc.

Portions of this software are copyright by Microsoft Corporation. Microsoft®, Windows, Windows® 95, and Windows NT® are either trademarks or registered trademarks of Microsoft Corporation.

Infinity Windoid WDEF is © 1991–95 Infinity Systems.

Overview

This volume presents a reference chapter for each item in the StatView analysis browser, organized according to the analysis browser's default order.



Analysis chapters include the following sections:

1. Discussion of the analysis: the theory behind it, how to use it, and guidelines for interpreting your results
2. Dialog box settings: how to set analysis parameters and how your choices affect the results you get
3. Data requirements: how to organize your data, what types of variables to assign, and how to use buttons in the variable browser
4. Results: the tables and/or graphs you can produce and what they show
5. Templates: related templates in the Analyze menu and what they produce
6. Exercises: step-by-step examples showing you how to use the analysis

The final chapter, “Formulas,” details the functions and expression language that can be used with Formula, Recode, Series, Random Numbers, and Criteria commands. The chapter first discusses general rules for working with StatView’s expression language, then gives details and examples for each function.

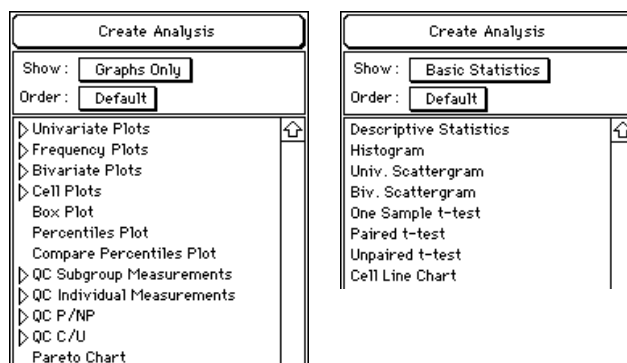
How to find what you want

Types of analyses

Both the manual and the program itself can help you find the main types of analyses:

Table of contents The “[Contents](#),” p. ix, lists the main sections of each chapter. The page number for each function is listed for the “Formulas” chapter.

Analysis browser If you want to know which analyses produce graphs, choose Graphs Only for Show in the analysis browser. Similarly, to learn which analyses produce the most basic statistics, choose Basic Statistics. Two other choices let you browse only the Quality Control or Survival Analysis items. (To see the analysis browser, open a view window by selecting New View from the Analyze menu.)



Certain tests

Suppose you want a Cramer’s V statistic and you can’t remember what type of analysis provides it. You have two ways to get an answer:

On-line help Both Windows help (Windows only) and StatView Guide (an Apple Guide, Macintosh only) index StatView’s tests by name. You can use these systems while you’re working to find tests quickly—and even get step-by-step instructions for completing the tests. If you need more discussion, look in the manual’s index...

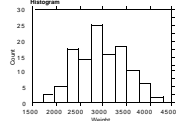
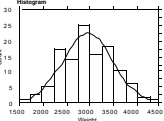
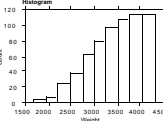
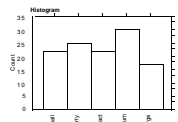
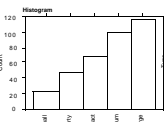
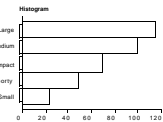
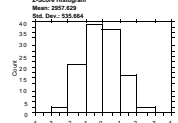
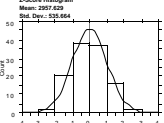
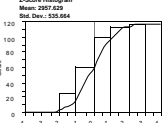



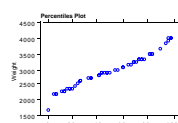
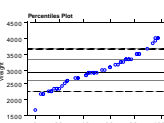
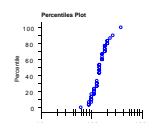
Index The index to this volume points you to the page where each test and graph type is discussed.

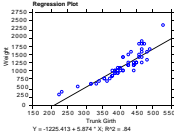
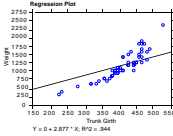
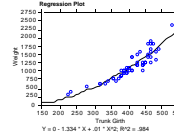
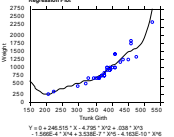
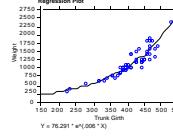
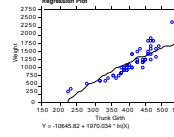
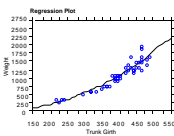
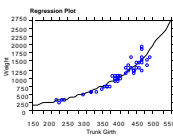
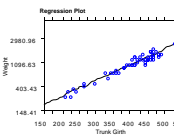
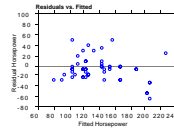
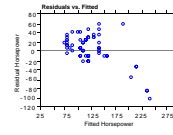
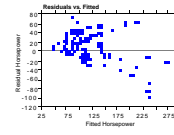
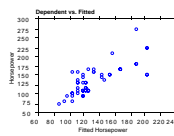
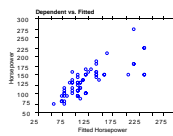
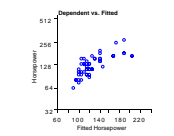
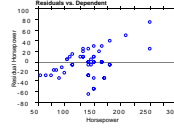
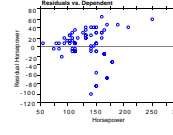
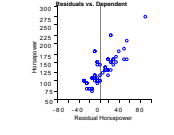
Types of graphs

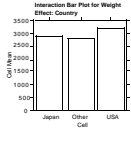
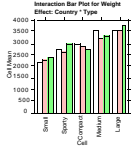
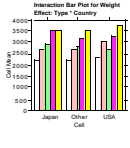
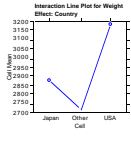
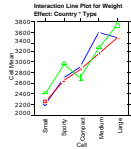
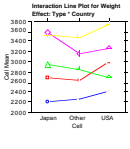
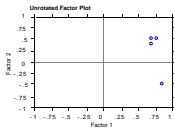
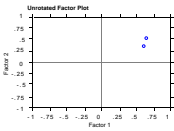
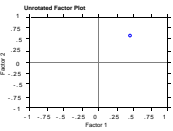
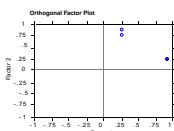
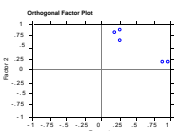
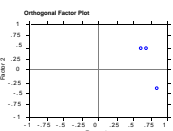
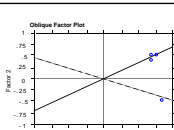
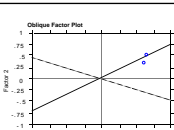
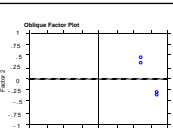
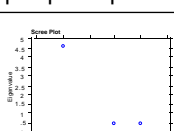
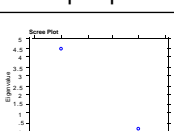
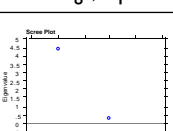
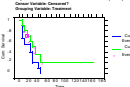
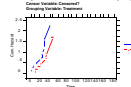
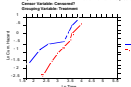
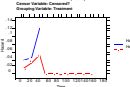
StatView produces many types of graphs for data analysis. Many StatView users are familiar with more than a few products for graphing, and each product classifies graph types differently. To aid your use of StatView, here is an overview of StatView's main graph types, with thumbnail sketches of each. Use this chart to find your way to the graph you want.

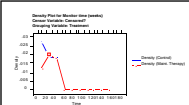
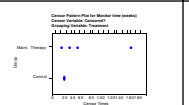
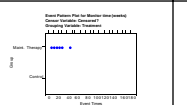
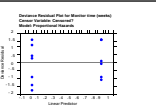
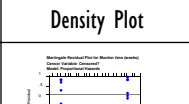
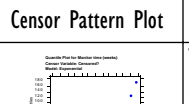

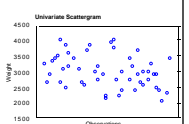
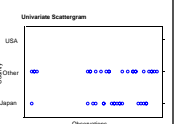
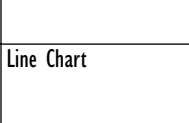
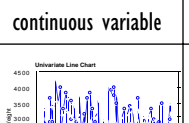
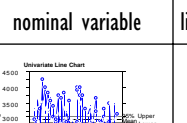

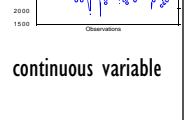
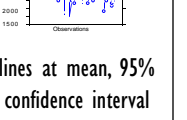

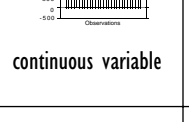
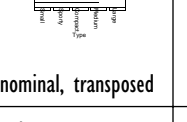
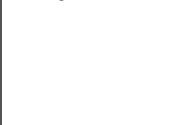
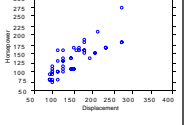
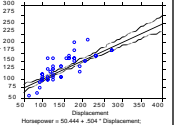
This chart is by no means exhaustive. Countless variations are possible through assigning variables in different orders or to different roles, assigning Split By variables, using Edit Analysis to adjust parameters of the graph, using Edit Display on various graph components and the graph as a whole, adding colors and fills, and so on. In this chart we simply show a handful of tiny examples and variations, so that you know where to look in the program for the type of graph you need.

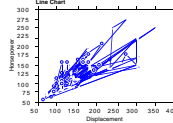
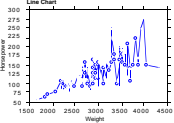
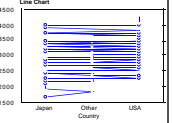
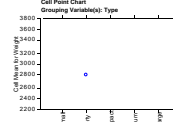
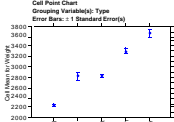
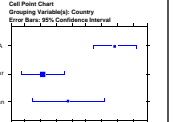
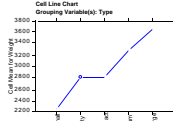
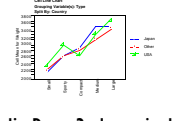
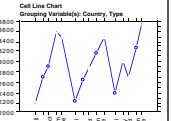
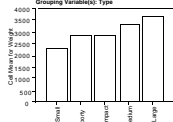
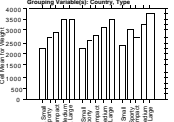
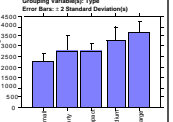
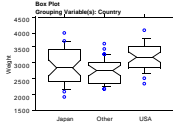
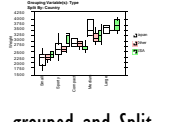
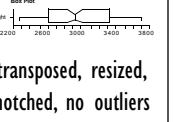
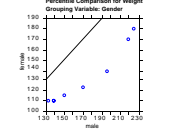
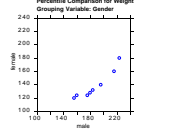
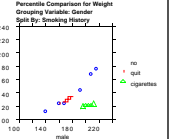
Finally, we hope that this chart will spark your imagination, giving you ideas that will take your presentation to new worlds of graphic possibility.

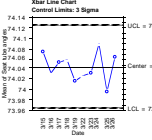
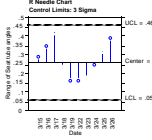
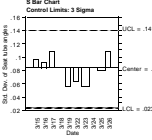
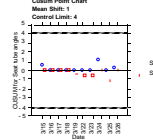
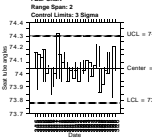
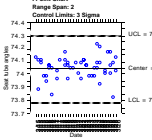
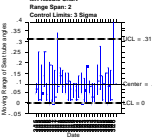
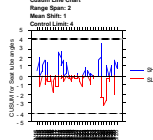
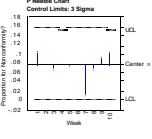
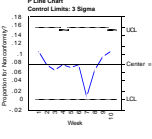
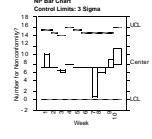
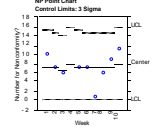
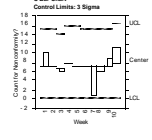
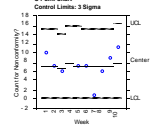
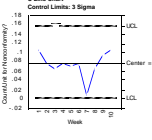
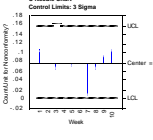
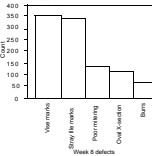
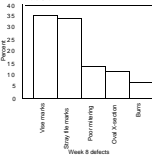
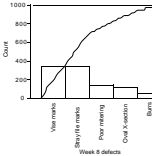
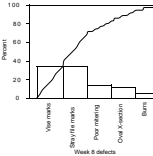
Frequency Distribution	Histogram	 continuous variable	 normal curve	 cumulative
		 nominal variable	 cumulative	 transposed, open frame
	Z-Score Histogram	 continuous variable	 normal curve	 cumulative
		 continuous variable	 nominal variable	 Split By
	Percentiles	 continuous variable	 with lines	 transposed, log scale

Regression	Regression Plot		
	 <p>simple regression</p>	 <p>simple regression, no intercept</p>	 <p>polynomial regression, order 2</p>
	 <p>polynomial regression, order 7</p>	 <p>exponential regression</p>	 <p>logarithmic regression</p>
Residuals vs. Fitted	 <p>power regression</p>	 <p>growth regression</p>	 <p>exponential, log scale for Y axis</p>
	 <p>simple regression</p>	 <p>no intercept</p>	 <p>square symbols</p>
Dependent vs. Fitted	 <p>simple regression</p>	 <p>no intercept</p>	 <p>Y axis log base 2</p>
	 <p>simple regression</p>	 <p>no intercept</p>	 <p>square, transposed</p>

ANOVA	Interaction Bar Plot	 <p>one factor</p>	 <p>two factors</p>	 <p>assign factors in different order</p>
	Interaction Line Plot	 <p>one factors</p>	 <p>two factors</p>	 <p>assign factors in different order</p>
Factor	Unrotated Factor Plot(s)	 <p>principal components</p>	 <p>iterated principal axis</p>	 <p>Harris image, Equamax</p>
	Orthogonal Factor Plot(s)	 <p>principal components</p>	 <p>iterated principal axis</p>	 <p>Harris image, Equamax</p>
	Oblique Factor Plot(s)	 <p>principal components</p>	 <p>iterated principal axis</p>	 <p>Harris image, Equamax</p>
	Scree Plot	 <p>principal components</p>	 <p>iterated principal axis</p>	 <p>Harris image, Equamax</p>
Survival	Cum. Survival Plot	 <p>Cum. Survival Plot</p>	 <p>Cumulative Hazard Plot</p>	 <p>Ln Cum. Hazard Plot</p>
				 <p>Hazard Plot</p>

	 Density Plot	 Censor Pattern Plot	 Event Pattern Plot	 Deviance Resid. Plot
	 Martingale Resid. Plot	 Quantile Plot	The first three types of plots are available with both nonparametric and regression methods, the next four with nonparametric, and the last two with regression.	
Univariate plots	Scattergram  continuous variable	Univariate Scattergram  nominal variable	Univariate Scattergram  lines at mean, std dev	
	Line Chart  continuous variable	Univariate Line Chart  lines at mean, 95% confidence interval	Univariate Line Chart  sorted	
	Bar Chart  continuous variable	Univariate Bar Chart  nominal, transposed	Univariate Bar Chart  nominal, sorted	
Bivariate plots	Scattergram  two continuous	Scattergram  linear regression with 95% confidence bands		Bivariate Scattergram with Regression  separate regression line for each group
	Bivariate Scattergram with Lowess  lowess fit, 66% tension	Bivariate Scattergram with Cubic Spline  cubic spline		Bivariate Scattergram with Supersmoother  supersmoother

	Line Chart	 two continuous	 sorted on X	 continuous Y, nominal X, sorted on Y
Cell plots	Point Chart	 continuous & nominal	 standard error bars	 transposed, mean and 95% conf. interval
	Line Chart	 continuous & nominal	 Split By a 2nd nominal	 continuous, 2 nominals
	Bar Chart	 continuous & nominal	 continuous, 2 nominals	 filled bars, 2-standard deviation error bars
Box plots	Box Plot	 standard	 grouped and notched	 transposed, resized, notched, no outliers
	Compare Percentiles	 X and Y axes equal, with X=Y line	 X and Y axes <i>not</i> equal, with X=Y line	 S and Y axes equal, with Split By

QC Subgroup	 <p>Xbar line</p>	 <p>R needle</p>	 <p>S bar</p>	 <p>CUSUM point</p>
QC Individual	 <p>I bar</p>	 <p>I point</p>	 <p>MR needle</p>	 <p>CUSUM line, no points</p>
QC P/NP	 <p>P needle</p>	 <p>P line</p>	 <p>NP bar</p>	 <p>NP point</p>
QC C/U	 <p>C bar</p>	 <p>C point</p>	 <p>U line</p>	 <p>U needle</p>
Pareto	 <p>counts</p>	 <p>percents</p>	 <p>counts, cum. curve</p>	 <p>percents, cum. curve</p>

Contents

I Descriptive Statistics 1

- Discussion 1
- Measures of central tendency 1
- Measures of variability 3
- Measures of distribution characteristics 6

- Dialog box settings 7
- Data requirements 9
- Results 9
- Templates 9
- Exercise 9

2 Frequency Distribution 13

- Discussion 13
- Histograms and pie charts 13
- z-score histograms 13
- Dialog box settings 14

- Data requirements 16
- Results 16
- Templates 16
- Exercise 17

3 Percentiles 19

- Dialog box settings 19
- Data requirements 19
- Results 20

- Templates 20
- Exercise 20

4 One Sample Analysis 23

- Discussion 23
- One sample t-test 23
- Chi-square test 24
- Tail 24
- Dialog box settings 24

- Data requirements 25
- Results 26
- Templates 26
- Exercise 26

5 Paired Comparisons 29

- Discussion 29
- Paired t -test 29
- Z-test for correlation coefficients 31
- Dialog box settings 32
- Data requirements 33

- Results 33
- Templates 34
- Exercises 34
- Paired t -test 34
- Z-test 35

6 Unpaired Comparisons 37

- Discussion 37
- Unpaired t -test 37
- F-test 38
- Dialog box settings 39
- Data requirements 39

- Standard layout 40
- Compact variable 40
- Results 41
- Templates 41
- Exercise 41

7 Correlation and Covariance 43

- Discussion 43
- Correlation coefficient 43
- Fisher's r to z 44
- Bartlett's test of sphericity 44
- Confidence intervals 44
- Listwise/pairwise deletion 45
- Covariance 45

- Partial correlation 45
- Dialog box settings 46
- Data requirements 47
- Results 47
- Templates 47
- Exercise 48

8 Regression 51

- Discussion 51
- Simple and multiple regression 52
- Polynomial regression 52
- Stepwise regression 52
- Nonlinear models 54
- Model coefficients and intercept 55
- Criteria for model quality 56
- Residuals 57
- Dialog box settings 59
- Data requirements 61

- Results 62
- Templates 63
- Exercises 64
- Simple linear regression 64
- Polynomial regression 65
- Growth regression 67
- Exponential regression 68
- Multiple regression 69
- Stepwise regression 70

9 ANOVA 73

- Discussion 73

- Hypothesis testing 74

<u>Model building</u>	76
<u>ANOVA</u>	79
<u>Regression</u>	80
<u>ANCOVA</u>	80
<u>MANOVA and MANCOVA</u>	81
<u>Repeated measures ANOVA</u>	82
<u>Post hoc tests (Multiple comparisons)</u>	84
<u>Dialog box settings</u>	89
<u>Data requirements</u>	90
<u>Factorial</u>	91

<u>Repeated measures</u>	91
<u>Results</u>	95
<u>Templates</u>	96
<u>Exercises</u>	96
<u>Fully factorial ANOVA</u>	96
<u>Repeated measures ANOVA</u>	97
<u>ANCOVA</u>	99
<u>Randomized complete block ANOVA</u>	101
<u>Latin square ANOVA</u>	105
<u>Factorial MANOVA design</u>	108

10 Contingency Tables III

<u>Discussion</u>	III
<u>Chi-square test</u>	II2
<u>Tables produced</u>	II2
<u>Additional statistics: G-statistic and Cramer's V</u>	II3
<u>2x2 contingency tables: Fisher's exact test, Phi coefficient</u>	II3
<u>Dialog box settings</u>	II4

<u>Data requirements</u>	II4
<u>Coded raw data</u>	II4
<u>Coded summary data</u>	II5
<u>Two-way table</u>	II5
<u>Results</u>	II6
<u>Templates</u>	II6
<u>Exercise</u>	II7

11 Nonparametrics II9

<u>Discussion</u>	II9
<u>One sample sign test</u>	II9
<u>Mann-Whitney U test</u>	II20
<u>Kolmogorov-Smirnov test</u>	II20
<u>Wald-Wolfowitz runs test</u>	II20
<u>Wilcoxon signed rank test</u>	II20
<u>Paired sign test</u>	II21
<u>Spearman rank correlation coefficient</u>	II21
<u>Kendall's rank correlation coefficient</u>	II21
<u>Kruskal-Wallis test</u>	II21

<u>Friedman test</u>	II22
<u>Dialog box settings</u>	II22
<u>Data requirements</u>	II23
<u>Results</u>	II24
<u>Templates</u>	II24
<u>Exercises</u>	II25
<u>One sample sign test</u>	II25
<u>Mann-Whitney U test</u>	II25
<u>Wilcoxon signed rank test</u>	II26
<u>Kendall rank correlation</u>	II27
<u>Kruskal-Wallis test</u>	II28
<u>Friedman test</u>	II28

12 Factor Analysis I31

<u>Discussion</u>	I31
<u>Data input</u>	I31
<u>Factor extraction methods</u>	I32

<u>Factor loadings</u>	I33
<u>Rotations</u>	I33
<u>Number of factors to extract</u>	I33

Transformation method	135
Factor scores	135
Dialog box settings	136
Data requirements	137

Results	138
Templates	138
Exercise	138
Plots	141

13 [Survival: Nonparametric](#) 143

Introduction to survival analysis	143
What is survival analysis?	143
Survival and hazard functions	144
Regression models	144
Parametric and nonparametric analyses	145
Censored observations	145
An example	146
Nonparametric methods	147
Discussion	148
Understanding the event time variable	149
Nonparametric survival function estimates	149

Hazard plots	150
Comparisons of survival functions	150
Dialog Box Settings	152
Survival: Nonparametric Methods dialog box	152
Survival Columns dialog box	155
Rank Tests dialog box	156
Data requirements	157
Results	159
Default Results	159
Other Results	160
Templates	163
Exercise	163

14 [Survival: Regression](#) 167

Regression methods	167
Discussion	168
Proportional hazards model	168
Stratified proportional hazard models	169
Significance tests and confidence intervals	169
Residual plots	170
Parametric models	171
Dialog Box Settings	175
Survival: Regression Models dialog box	175
Survival Columns dialog box	178

Joint Significance Tests dialog box	179
Estimation Parameters dialog box (proportional hazards)	180
Estimation Parameters dialog box (parametric models)	181
Coefficient Initial Values dialog box	182
Data requirements	183
Results	185
Default Results	185
Other Results	188
Templates	191
Exercise	191

15 [Logistic regression](#) 199

Discussion	199
Simple logistic regression model	200
Multiple logistic regression models	201

Assumptions	202
Estimating coefficients	203
Polytomous logistic regression	

models	204
Dialog box settings	204
Data requirements	205
Nominal data coding	206
Results	207

Templates	207
Exercises	207
Simple logistic regression	207
Multiple logistic regression	212
Polytomous logistic regression	214

16 [Univariate Plots](#) 217

Dialog box settings	217
Data requirements	218
Results	218

Templates	219
Exercise	219

17 [Bivariate Plots](#) 221

Fitted lines	221
Linear regression	224
Smoothing bivariate plots	225
Dialog box settings	228
Data requirements	229
Results	229
Templates	230

Exercises	230
Bivariate scattergram	230
Linear regression	231
Bivariate plot with nominal data	232
Cubic spline	233
Lowess fit	235
Supersmoother	236

18 [Cell Plots](#) 237

Dialog box settings	237
Data requirements	238
Results	239

Templates	239
Exercises	239

19 [Box Plots](#) 243

Dialog box settings	243
Data requirements	244
Results	244

Templates	244
Exercises	244

20 [Compare Percentile Plots](#) 247

Dialog box settings	247
Data requirements	247
Results	248

Templates	248
Exercise	248

21 QC Subgroup Measurements 251

- Introduction to SPC 251
 - What is statistical process control? 251
 - When is a process in control? 251
 - Process control vs. process capability 252
 - An example 254
- Subgroup measurements 257
- Discussion 257
 - Xbar (subgroup mean) charts 257
 - R (subgroup range) charts 259
 - S (subgroup standard deviation) charts 259
 - Tests for special causes 259
 - CUSUM (cumulative sum) charts 261
 - Capability indices 261
- Dialog box settings 262
 - QC Subgroup Measurements dialog box 262

- Tests for Special Causes dialog box 263
- Custom Tests dialog box 264
- CUSUM Parameters dialog box 264
- QC Line Parameters dialog box 266
- Variables dialog box 267
- CAPA Parameters dialog box 267
- Data requirements 268
- Results 269
 - Xbar Statistics results 269
 - Special Causes Definitions table 270
 - R Statistics results 270
 - S Statistics results 271
 - CUSUM Statistics results 271
 - CAPA table 272
 - Summary Table 272
- Templates 273
- Exercise 273

22 QC Individual Measurements 277

- Discussion 277
 - I (individual measurement) charts 277
 - MR (moving range) charts 278
 - Tests for special causes 278
 - CUSUM charts 279
 - Capability indices 279
- Dialog box settings 279
 - QC Individual Measurements dialog box 279
- Data requirements 280

- Results 281
 - I Statistics results 281
 - Special Causes Definitions tables 281
 - MR Statistics results 281
 - CUSUM Statistics results 282
 - CAPA results 282
 - Summary table 282
- Templates 283
- Exercise 283

23 QC P/NP 287

- Discussion 287
 - p (proportion defective) charts 288
 - np (number defective) charts 288
 - Tests for special causes and custom tests 289
- Dialog box settings 290
 - QC P/NP dialog box 290
- Data requirements 290
 - Format 1 291

- Format 2 291
- Results 292
 - p results 292
 - np results 293
 - Special Causes Definitions table 293
 - Summary table 294
- Templates 294
- Exercise 294

24 QC C/U

299

Discussion 299

c (count of defects) charts 300

u (average number of defects) charts 300

Tests for special causes and custom tests 300

Dialog box settings 301

QC C/U dialog box 301

Data requirements 302

Results 303

c results 303

u results 304

Special Causes Definitions table 305

Summary Table 305

Templates 305

Exercise 305

25 Pareto Analysis

309

Discussion 309

Dialog box settings 309

Data requirements 310

Results 311

Templates 311

Exercise 311

26 Formulas

315

Overview 315

Examples in this chapter 316

Introduction 317

Variable types and formats 318

Casewise and columnwise operations 321

Arguments 323

Order of operations 326

Remarks 329

Static and dynamic formulas 329

Date and time functions 330

Text functions 331

Operators 332

?+? 333

?-? 333

?*? or ?? 333

?/? 334

?^? or ?**? 334

+? 335

-? 335

(?) 336

Sets, intervals, and ranges 336

{...} 336

(?:?), [?:?], (?:?), [?:?] 337

<?, >? 337

Relations and logical operators 338

?<? 340

?<=? 340

?=? 340

?>=? 341

?>? 341

?<>? 341

if ? then ? else ? 341

IsMissing(?) 343

IsRowExcluded 343

IsRowIncluded 344

NOT(?) 345

false 345

true 345

? AND ? 345

? ElementOf ? 345

? IS ? 346

? ISNOT ? 347

? OR ? 347

? XOR ? 347

Functions 347

Abs(?) 348

ArcCos(?) 349

ArcCosh(?) 349

ArcCot(?) 350

-
- [ArcCsc\(?\)](#) 351
 - [ArcSec\(?\)](#) 352
 - [ArcSin\(?\)](#) 353
 - [ArcSinh\(?\)](#) 354
 - [ArcTan\(?\)](#) 355
 - [ArcTanh\(?\)](#) 356
 - [Average\(?, ...\)](#) 356
 - [AverageIgnoreMissing\(?, ...\)](#) 357
 - [BinomialCoeffs](#) 358
 - [BoxCox\(?, ?\)](#) 358
 - [Ceil\(?\)](#) 359
 - [ChooseArg\(?\)](#) 359
 - [CoeffOfVariation\(?, AllRows\)](#) 360
 - [Combinations\(?, ?\)](#) 361
 - [Concat\(?\)](#) 361
 - [Correlation\(?, ?, AllRows\)](#) 362
 - [Cos\(?\)](#) 363
 - [Cosh\(?\)](#) 364
 - [Cot\(?\)](#) 364
 - [Count\(?, AllRows\)](#) 365
 - [Covariance\(?, ?, AllRows\)](#) 365
 - [Csc\(?\)](#) 366
 - [CubicSeries\(1, 0, 0, 1\)](#) 367
 - [CumProduct\(?\)](#) 367
 - [CumSum\(?\)](#) 368
 - [CumSumSquares\(?\)](#) 368
 - [Date\(?, ?, ?\)](#) 369
 - [DateDifference\(?, ?, ?\)](#) 370
 - [Day\(?\)](#) 371
 - [DayOfWeek\(?\)](#) 372
 - [DayOfYear\(?\)](#) 372
 - [DegToRad\(?\)](#) 372
 - [Difference\(?, 1, 1\)](#) 373
 - [Div\(?, ?\)](#) 374
 - [DotProduct\(?, ?\)](#) 374
 - [e](#) 375
 - [Erf\(?\)](#) 376
 - [ExponentialSeries\(1\)](#) 376
 - [Factorial\(?\)](#) 377
 - [FibonacciSeries](#) 378
 - [Find\(?, ?, ?, false\)](#) 379
 - [Floor\(?\)](#) 380
 - [GeometricMean\(?, AllRows\)](#) 380
 - [GeometricSeries\(1, 2\)](#) 381
 - [Groups\(?, ...\)](#) 381
 - [HarmonicMean\(?, AllRows\)](#) 382
 - [Hour\(?\)](#) 382
 - [Lag\(?, 1\)](#) 382
 - [Len\(?\)](#) 383
 - [LinearSeries\(1, 1\)](#) 384
 - [Ln\(?\)](#) 385
 - [Log\(?\)](#) 385
 - [LogB\(?, ?\)](#) 386
 - [LogOdds\(?\)](#) 386
 - [MAD\(?, AllRows\)](#) 387
 - [Maximum\(?, AllRows\)](#) 387
 - [Mean\(?, AllRows\)](#) 388
 - [Median\(?, AllRows\)](#) 388
 - [Minimum\(?, AllRows\)](#) 389
 - [Minute\(?\)](#) 389
 - [Mod\(?, ?\)](#) 390
 - [Mode\(?, AllRows\)](#) 390
 - [Month\(?\)](#) 391
 - [MovingAverage\(?, ?\)](#) 391
 - [Norm\(?, AllRows\)](#) 392
 - [Now](#) 393
 - [NumberMissing\(?, AllRows\)](#) 393
 - [NumberOfRows](#) 394
 - [OneGroupChiSquare\(?, ?, ?\)](#) 394
 - [Percentages\(?, AllRows\)](#) 397
 - [Percentile\(?, ?, ?\)](#) 398
 - [Permutations\(?, ?\)](#) 399
 - [Pi](#) 400
 - [ProbBinomial\(?, ?, ?\)](#) 401
 - [ProbChiSquare\(?, 1\)](#) 402
 - [ProbF\(?, 1, 1\)](#) 402
 - [ProbNormal\(?, 0, 1\)](#) 403
 - [Probt\(?, 1\)](#) 404
 - [QuadraticSeries\(1, 0, 1\)](#) 404
 - [QuarticSeries\(1, 0, 0, 0, 1\)](#) 405
 - [RadToDeg\(?\)](#) 405
 - [RandomBeta\(1, 1\)](#) 406
 - [RandomBinomial\(?, ?\)](#) 406
 - [RandomChiSquare\(1\)](#) 407
 - [RandomExponential\(1\)](#) 407
 - [RandomF\(1, 1\)](#) 407
 - [RandomGamma\(1\)](#) 408
 - [RandomGaussian\(0, 1\)](#) 408
 - [RandomInclusion\(?\)](#) 409
 - [RandomNormal\(0, 1\)](#) 410
 - [RandomPoisson\(1\)](#) 410
 - [RandomT\(1\)](#) 411
 - [RandomUniform\(0, 1\)](#) 411
 - [RandomUniformInteger\(?, ?\)](#) 412

<u>Range(?, AllRows)</u>	412
<u>Rank(?, AllRows)</u>	413
<u>Remainder(?, ?)</u>	413
<u>ReturnChiSquare(?, ?)</u>	414
<u>ReturnF(?, 1, 1)</u>	415
<u>ReturnNormal(?, 0, 1)</u>	415
<u>ReturnT(?, 1)</u>	416
<u>Round(?)</u>	416
<u>RowNumber</u>	417
<u>Sec(?)</u>	418
<u>Second(?)</u>	419
<u>Sin(?)</u>	419
<u>Sinh(?)</u>	419
<u>Sqrt(?)</u>	420
<u>StandardDeviation(?, AllRows)</u>	420
<u>StandardError(?, AllRows)</u>	421
<u>StandardScores(?, AllRows)</u>	422
<u>Substring(?, ?, ?)</u>	422
<u>Sum(?, ...)</u>	423
<u>SumIgnoreMissing(?, ...)</u>	424
<u>SumOfColumn(?, AllRows)</u>	424
<u>SumOfSquares(?, AllRows)</u>	425
<u>Tan(?)</u>	426
<u>Tanh(?)</u>	426
<u>Time(?, ?, ?)</u>	427
<u>TrimmedMean(?, ?, AllRows)</u>	428
<u>Trunc(?)</u>	429
<u>VariableElement(?, ?)</u>	429
<u>Variance(?, AllRows)</u>	430
<u>Weekday(?)</u>	430
<u>WeekOfYear(?)</u>	431
<u>Year(?)</u>	431

A Algorithms 433

<u>General</u>	433
<u>Sum of squares calculations</u>	433
<u>Matrix inversions</u>	433
<u>Descriptive Statistics</u>	434
<u>Continuous variables</u>	434
<u>Nominal variables</u>	435
<u>Percentiles</u>	435
<u>One Sample Analysis</u>	435
<u>One sample <i>t</i>-test</u>	436
<u>Confidence interval for the mean</u>	436
<u>Chi-Square test for variance</u>	436
<u>Confidence interval for variance</u>	436
<u>Paired Comparisons</u>	437
<u>Paired <i>t</i>-test</u>	437
<u>Confidence interval for the paired mean difference</u>	437
<u>Z test and confidence interval for the correlation coefficient</u>	437
<u>Unpaired Comparisons</u>	437
<u>Unpaired <i>t</i>-test</u>	438
<u>Confidence interval for the unpaired mean difference</u>	438
<u>F test for variance ratio</u>	438
<u>Confidence interval for the variance ratio</u>	438
<u>Correlation and Covariance</u>	439
<u>Partial correlations</u>	439
<u>Bartlett's test of sphericity</u>	439
<u><i>p</i> values and confidence intervals</u>	439
<u>Regression</u>	439
<u>ANOVA</u>	440
<u>Multivariate analysis of variance (MANOVA)</u>	441
<u>Multiple comparisons</u>	443
<u>Contingency Tables</u>	445
<u>Two way tables</u>	445
<u>Nonparametrics</u>	447
<u>One sample sign test</u>	447
<u>Mann-Whitney U</u>	448
<u>Kolmogorov-Smirnov</u>	448
<u>Wald-Wolfowitz runs test</u>	448
<u>Wilcoxon signed-rank</u>	449
<u>Paired sign test</u>	449
<u>Spearman rank correlation coefficient</u>	450
<u>Kendall correlation coefficient</u>	450
<u>Kruskal-Wallis test</u>	451
<u>Friedman test</u>	452
<u>Survival analysis</u>	452
<u>Kaplan-Meier</u>	453
<u>Actuarial</u>	454
<u>Linear rank tests</u>	456
<u>Proportional hazards model</u>	458
<u>Parametric models</u>	458

<u>Estimation (proportional hazards)</u>	458	<u>Classification</u>	470
<u>Estimation (parametric models)</u>	459	<u>Global tests</u>	470
<u>Newton-Raphson iteration</u>	462	<u>Bivariate Plots</u>	471
<u>Coefficient covariances</u>	462	<u>Cubic spline</u>	473
<u>Model coefficient p values (Wald)</u>	462	<u>QC Subgroup Measurements</u>	473
<u>Confidence intervals</u>	463	<u>Sigma</u>	473
<u>Survival function and related</u>		<u>Xbar analyses</u>	474
<u>quantities</u>	463	<u>R analyses</u>	474
<u>Testing the global null hypothesis</u>	465	<u>S analyses</u>	475
<u>Joint significance tests</u>	466	<u>CUSUM analyses</u>	475
<u>Stratification (proportional hazards</u>		<u>Capability analyses</u>	476
<u>only)</u>	466	<u>QC Individual Measurements</u>	477
<u>Stepwise</u>	467	<u>Sigma</u>	477
<u>Logistic Regression</u>	467	<u>I analyses</u>	477
<u>Logistic model</u>	468	<u>MR analyses</u>	477
<u>Estimation</u>	468	<u>CUSUM and capability analyses</u>	478
<u>Parameter fitting</u>	469	<u>QC P/NP</u>	478
<u>Coefficient covariances</u>	469	<u>p analyses</u>	478
<u>Model coefficient p values (Wald</u>		<u>np analyses</u>	478
<u>test)</u>	469	<u>QC C/U</u>	479
<u>Partial correlation (R statistic)</u>	469	<u>c analyses</u>	479
<u>Confidence intervals</u>	469	<u>u analyses</u>	479
<u>Likelihood ratio tests</u>	470		

B References 481

<u>Suggested Reading</u>	481	<u>Logistic regression</u>	484
<u>General</u>	481	<u>Survival analysis</u>	485
<u>Factor analysis</u>	483	<u>QC analysis</u>	485

C Glossary 487

Index 497

Descriptive Statistics

Descriptive statistics compute numbers that summarize data rather than making comparisons between the data or its sources. Descriptive statistics fall into three categories:

1. measures of **central tendency**, which give an idea of the average value of a number or other quantity (where average can take on a variety of meanings)
2. measures of **variability**, which convey whether most measurements are clustered within a narrow range of values or spread over a large range
3. measures of an **overall distribution property** indicated by a single number

You can use descriptive statistics on measurements representing a sample of some underlying population, anecdotal evidence, available data, or the entire population. This population may be real (people who live in a particular city) or theoretical (all the plants of a particular type). The size of a population or the destructive nature of the measurement method usually makes it undesirable to undertake measuring the entire population. Descriptive statistics can be merely descriptive, but are more often estimates of usually immeasurable quantities known as **population statistics**. Since descriptive statistics are calculated from a sample of the population, they are often called **sample statistics**. Later references are to sample statistics unless otherwise stated. Sample statistics characterize the population on which they are based.

Discussion

Measures of central tendency

A descriptive statistic summarizes data with a single number. One approach uses the mean, or arithmetic average, to summarize the central tendency of a set of numbers.

Mean

The **mean** is the sum of the observations divided by the number of observations. The sum of the differences between each observation and the mean is zero. Each observation plays a part in the calculation of the mean, so difficulties can arise if your data contains **outliers**, observations that are distant from the bulk of your data. Outliers can be discarded or corrected *if* they arise from an obvious error in data collection; but often they are important to the data and

should not be ignored. A simple example concerns the salaries of employees of a small company. There are five employees: two clerks each making \$12,000 per year; two sales reps making \$15,000 and \$18,000; and the owner of the company, whose salary is \$100,000. (In practice, measurements based on a sample of only five observations should be regarded with caution.) The mean of the salaries is \$31,400. This is not an accurate reflection of the “average” employee salary of the company. The owner’s salary distorts the value of the mean since it is so much larger than the other salaries, yet all are given equal weight. The same problem occurs with the “average” price of homes in a neighborhood; several expensive homes may inflate the mean price of homes, giving the consumer a distorted image of the cost of neighborhood housing.

Median

An alternative measure of central tendency that can solve this problem is the **median**. The median is the middle value in a set of observations that is ordered from lowest to highest value. When there is an even number of observations, the median is the average of the two numbers on either side of the middle. By definition, half of the observations are less than or equal to the median, while the other half is greater than the median. For the salary example given above, the median salary is \$15,000, a much better estimate of an “average” employee salary. The effect of outliers is eliminated because only the central one or two observations determine the median. The importance of most other observations is eliminated along with the outliers since only the order of the observations is ever used in calculating the median.

Trimmed mean

A measure of central tendency that provides an alternative to discarding all the observations except the central one or two is the **trimmed mean**. This statistic is a compromise between the mean, which uses all the data, and the median, which focuses on only one or two central values. The observations that are most distant from the center of the data are eliminated (trimmed) before the mean is calculated. You decide the amount of data to be trimmed before the remaining observations are averaged; the default is 10%. The amount of data that is ignored at both extremes of the dataset is expressed as a percentage. In an example of 100 observations, 20% of which are trimmed, the 20 largest and the 20 smallest (40 observations in all) are eliminated from consideration, and the mean is calculated from the remaining 60 observations. For the salary example, the 20% trimmed mean is \$15,000.

Mode

Another measure of central tendency is the **mode**, the value that occurs most often in a dataset. Your chances of guessing the value of an observation correctly are best if you choose the mode. The mode has a number of shortcomings when used as a measure of central tendency. Data collected using a continuous measurement scale (such as height or weight) may not contain observations with the same value. In such a case, the data has to be grouped before a meaningful value of the mode is determined. Alternatively, a dataset can have several modes, making it difficult to decide the appropriate value to use. Nevertheless, the mode may be a useful measure of central tendency for a variable that takes on a limited number of values,

or where the values are mostly in one clump. The salary example has mode \$12,000, since that value appears twice and the others each appear once.

Geometric and harmonic mean

Two less common measures of central tendency are the **geometric mean** and **harmonic mean**. These measure the central tendency of a mathematical transformation of the original observations. The transformed data may have more desirable statistical properties than the raw data. With variables like economic indices and bacterial counts, for example, which exhibit more variability as their values increase, the logarithm of the data often behaves better than the untransformed data. The geometric mean is calculated from the logarithm of the variables and re-transformed to the original scale of measurement. The harmonic mean is calculated similarly, but uses a reciprocal transformation (transforms a value by dividing one by that value). The harmonic mean is sometimes used to report the central tendency of rates or ratios. The salary example has geometric mean \$20,794 and harmonic mean \$16,728.

Any variable containing zeros or negative values will return missing values for the harmonic and geometric means.

Measures of variability

A measure of central tendency alone generally does not provide enough information to summarize a set of numbers. For example, if every value in one dataset has the same value, but the values in a second dataset are spread over a wide range, the mean, median or trimmed mean for the two datasets can still be the same. The mean of the dataset containing identical values is more representative of the sample's central tendency than the mean of the more diverse sample. One effective way to display the spread or variability of a set of numbers is a **histogram** (a bar chart representing a frequency distribution). There are also several descriptive statistics that summarize variability. See [“Histograms and pie charts,” p. 13](#), for more information about histograms.

Minimum, maximum and range

A simple expression of the variability of a set of numbers is a report of the **minimum** (the smallest value in the set of numbers), the **maximum** (the largest value) and the **range** (the difference between the minimum and maximum). These values may not be representative of the rest of the dataset, so providing only the minimum, maximum and range can be misleading, but their easy interpretation might make it useful to report them in addition to other measures of variability. The minimum salary is \$12,000 and the maximum is \$100,000 for a range of \$88,000.

Variance

It is usually better to report some average measure of the difference between each value in a variable and a measure of central tendency (usually the mean). You cannot calculate a simple average because, by the definition of the mean, the average difference between each observa-

tion in a dataset and the mean must be zero. One of the most common measures which gets around this problem is the **variance**.

The variance does this by squaring the differences between the observations and the mean before averaging. The **sample variance**, which is the type of variance most commonly used, is usually calculated by dividing these squared differences by one less than the number of observations. The **population variance** is calculated by dividing the squared differences by the number of observations. StatView defaults to calculating the sample variance. If the data you are analyzing is an entire population as opposed to a sample of a population, you can choose to divide by the number of observations (n , as opposed to $n-1$) in the Descriptive Statistics dialog box. Use of the square of the differences increases the influence of observations far from the mean in calculating the variance. This may or may not be desirable, depending on the nature of your dataset. For example, if your data contains many outliers, the variance might be considerably larger than if you did not have outliers. The salary example has a large variance (1,476,800,000), due in part to the extreme upper value (\$100,000). Variance is often used as a measure of variability when the mean is used as a measure of central tendency because the sum of squares of differences from a set of data and any single value is minimized when that value is the sample mean.

Standard deviation

A consequence of using the square of the differences is that the variance is reported in the square of the original unit of measurement and can be difficult to interpret. For example, if the height of a group of plants is measured in centimeters, the variance is expressed as square centimeters. To overcome this problem, variability is usually reported as the **standard deviation** (the square root of the variance). This represents an “average” deviation from the mean in the same unit of measurement as the original observations. The salary example has standard deviation \$38,429.

Data from a **normal** (Gaussian or bell-shaped) distribution follow the empirical rule of statistics: 68% of the data is contained in the range of the mean plus or minus the standard deviation; 95% in the range of the mean plus or minus twice the standard deviation; 99.7% in the range of the mean plus or minus three times the standard deviation. Thus, a quick rule of thumb for normally distributed data is: the vast majority of observations (95%) fall within two standard deviations of the mean.

Coefficient of variation

The **coefficient of variation** (CV) is a unitless expression of variability calculated by dividing the sample standard deviation by the sample mean. It is especially useful when comparing the variability of several measurements, or when measurements are in different units. When the mean is numerically small (near zero), the coefficient of variation may be very large, even though the variation in the data is not excessive. The salary example has CV 1.224.

Standard error of the mean

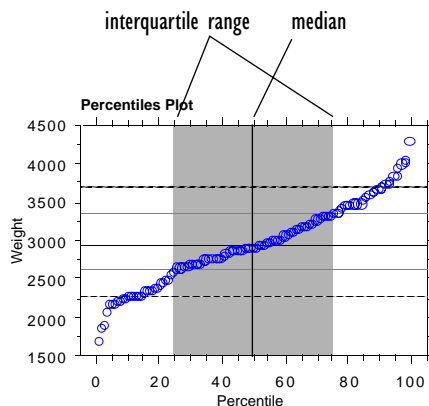
The standard deviation of a set of observations estimates the variability of the underlying population. For example, the empirical rule described above relates to the proportion of individual values that will fall within a particular range. However, it is often more meaningful to consider the variability of the sample mean, since it is the statistic that is actually used to gain insight into the central tendency of the data. The **standard error of the mean** is a statistic that estimates the variability in the sample mean you expect if you take repeated samples of the same size from the population. It is calculated by dividing the standard deviation of the observations by the square root of the number of observations. Since it is unlikely that a sample of observations would all be unusually high or low, we would expect the variability of the mean to be less than that of an individual value.

For example, a dataset contains the weights of 100 ten year old boys. You could calculate the standard deviation of the data to get an idea of the variation in weights for these individual boys. But if you repeatedly sample 100 boys from a theoretical population of ten-year-olds, it is unlikely that you would ever get a sample where most or all of the boys are unusually light or heavy; thus the variability of the mean will be less than the variabilities of the individual values.

To apply the empirical rule to the mean of a group of measurements, use the standard error of the mean instead of the standard deviation. In such a case, you estimate the standard deviation of a hypothetical population of means, and interpret the standard error of the mean relative to the mean just as you would the standard deviation relative to the observations.

Interquartile range (IQR)

In the presence of outliers, the median or trimmed mean provides a measure of central tendency. Similarly, a variety of measures of variability are appropriate when outliers are present. One measure closely related to the median is the **interquartile range** or IQR. Recall that the median is the value greater than or equal to one half of the data and less than the other half. The median is an example of a group of measures called percentiles. The n th percentile is the value such that $n\%$ of the data is equal to or less than the percentile. Thus, the median is the 50th percentile, and 90% of all values are found at or below the 90th percentile. The interquartile range is calculated by subtracting the 25th percentile from the 75th. Thus, it is the spread of values containing the central 50% of the data and, like the median, ignores the outmost points in a dataset.



Median absolute deviation (MAD)

The **median absolute deviation** (MAD) is a measure of variability that incorporates all the data, but does not give as much influence to outliers as the standard deviation. The MAD is the median of the set of absolute differences between each data point and the median of the data. The MAD is often a useful measure of variability when the median is used to describe the central tendency of the data.

Measures of distribution characteristics

While measures of central tendency and variability are useful for succinctly describing the characteristics of data, sometimes more information is needed. It may be of interest to know if the outlying values are mostly very large or very small, or if most of the values are close to the central values. Two useful statistics that describe these properties of a set of data are skewness and kurtosis.

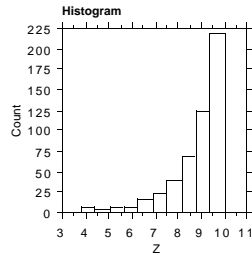
Skewness

Skewness is a reflection of the symmetry of the distribution, that is, the parts of the distribution above and below the mean. For a symmetric distribution of values, the mean and the median coincide. A histogram of the data will show one side of the data as a mirror image of the other side, with the value of the mean as the “mirror.”

A symmetric distribution has a skewness value of zero. When the number of values smaller than the mean is less than the number of values larger than the mean, the distribution is skewed to the left, or negatively skewed. In this case the tails will “stretch out” more on the left (lower) side of the distribution. The skewness value is less than zero and the mean is less than the median. In the opposite case, when the number of smaller values is greater, the distribution is skewed to the right, or positively skewed. The skewness value is greater than zero and the mean is greater than the median.

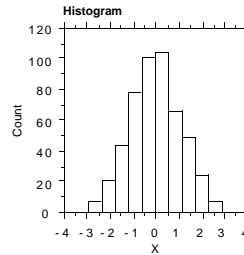
Descriptive Statistics

	Z
Mean	8.913
Skewness	-1.823
Median	9.273



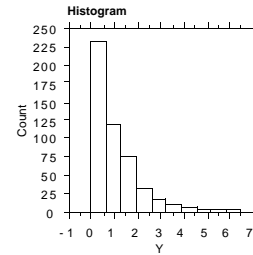
Descriptive Statistics

	X
Mean	0.000
Skewness	-.006
Median	0.000



Descriptive Statistics

	Y
Mean	1.038
Skewness	1.963
Median	.720

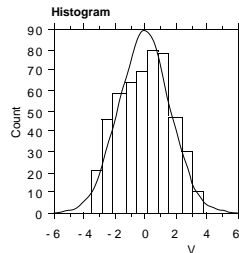


Kurtosis

Kurtosis is a measure of the amount of data in the tails (as opposed to the central part of the distribution). Kurtosis is scaled such that normally distributed data has a kurtosis value of zero. Positive kurtosis values indicate that the data is squeezed into the middle of the distribution (the tails of the distribution are slim and there are few extreme values). Negative values indicate the data has many extreme values spread out over a wide range (the tails are fat). There are terms to describe these three situations: **platykurtic**, for negative kurtosis values; **mesokurtic**, for kurtosis values near zero; and **leptokurtic**, for positive kurtosis values.

Descriptive Statistics

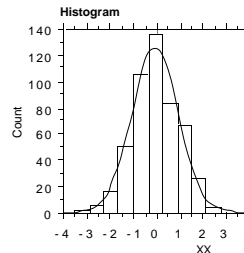
	V
Kurtosis	-.779



platykurtic

Descriptive Statistics

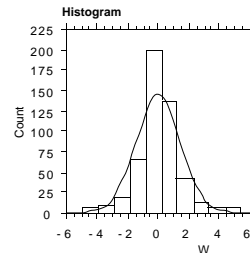
	XX
Kurtosis	.018



mesokurtic

Descriptive Statistics

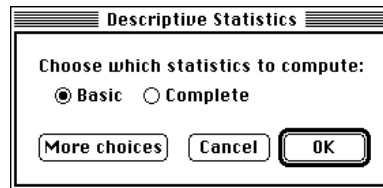
	W
Kurtosis	2.489



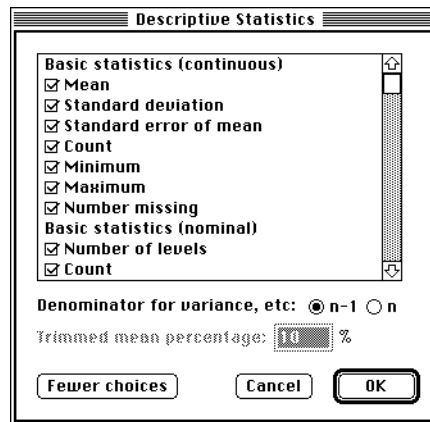
leptokurtic

Dialog box settings

When you create or edit descriptive statistics, you set the analysis parameters in two dialog boxes, a small one with few choices and an expanded one with many choices. In the first of the two, you can select either a subset of the descriptive statistics (Basic) or all the descriptive statistics (Complete) and click OK.



If you click the More choices button, you see an expanded dialog box listing all the descriptive statistics.



Using this dialog box, you can pick and choose from all the available descriptive statistics by clicking in the checkbox. The statistics are displayed in three separate groups: Basic statistics (continuous), Basic statistics (nominal) and Additional statistics. If Basic is selected in the fewer choices dialog box, then only the basic statistics are checked in the expanded dialog box. If Complete is selected, then all statistics will be checked in the expanded dialog box. Those statistics with a check mark next to them are included in the summary table. Using the expanded dialog box, you can customize which statistics to display by clicking to remove the check mark.

Denominator for variance Specify which value to use in calculating the variance. The default calculates a sample variance. See the previous discussion on the variance for more information.

Trimmed mean percentage Specify the percentage of observations to exclude at the high and low ends of the distribution when calculating the trimmed mean. The default is 10%, which trims the highest and lowest 10% of the observations before calculating the mean.

Data requirements

Descriptive statistics can be generated for one or more nominal or continuous variables.

Variable browser buttons	
Add	To generate descriptive statistics, select the variable(s) that you wish to analyze and click Add. When you select a descriptive statistics table and assign additional variables, they are added to the summary table which expands to include the new variables.
Split By	When you assign one or more split-by variables to a descriptive statistics table, results for each cell in the split-by variable(s) as well as totals for all groups are displayed in a single summary table.

Results

For explanations of the results, please see the preceding [“Discussion,” p. 1.](#)

Basic continuous	Table containing the mean, standard deviation, standard error of the mean, count, minimum, maximum, and the number missing for continuous variables.
Basic nominal	Table containing the number of levels, count, number missing and mode for nominal variables.
Additional statistics available	Table containing the above statistics and the variance, coefficient of variation, range, sum, sum of squares, geometric mean, harmonic mean, skewness, kurtosis, median, interquartile range, mode, trimmed mean, and median absolute deviation for continuous variables.

Templates

The following templates provide descriptive statistics.

Descriptive Statistics	Descriptive Statistics	Basic continuous statistics table.
	Descriptive Stats-- Complete	Complete continuous statistics table.
	Nominal Descriptive Stats	Nominal statistics table and histogram.

Exercise

In this exercise you create a set of descriptive statistics using the sample Car Data. It contains information about weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View

- In the analysis browser, double-click Descriptive Statistics
- Click OK to accept the default analysis parameters
- In the variable browser, click and drag to select from Country to Gas Tank Size, and click Add

Descriptive Statistics

	Mean	Std. Dev.	Std. Error	Count	Minimum	Maximum	# Missing
Weight	2957.629	535.664	49.735	116	1695.000	4285.000	0
Turning Circle	38.586	3.132	.291	116	32.000	47.000	0
Displacement	158.310	60.409	5.609	116	61.000	350.000	0
Horsepower	130.198	39.822	3.697	116	55.000	278.000	0
Gas Tank Size	16.238	3.076	.286	116	9.200	27.000	0

Nominal Descriptive Statistics

	# Levels	Count	# Missing	Mode
Country	3	116	0	3
Type	5	116	0	4

StatView calculates two tables, one for the continuous variables and one for nominal. For a discussion of nominal and continuous data class, see [“Data class,” p. 50 of Using StatView.](#)

It is useful to compare the subgroups of one variable that are defined by the levels of another variable. For example, comparing Turning Circle for cars from various countries will suggest which country makes the largest cars. To do this, you must first deselect the tables you just created. This avoids using the variables from existing tables in the new analysis. (New analyses are always created using the variables in any selected results.)

- Click in a blank space in the view

The tables are deselected. When deselected, the black handles around the tables disappear and the variables in the variable browser lose their usage markers.

- In the analysis browser, again double-click Descriptive Statistics
- Click More choices

The expanded Descriptive Statistics dialog box contains a scrolling list of statistics under three headings: Basic statistics (continuous), Basic statistics (nominal) and Additional statistics. Since Basic was selected in the first dialog box, all the basic statistics are selected in the expanded box. You will remove check marks from the statistics you do not wish to calculate. For this analysis, you will use only four descriptive statistics: mean, standard deviation, maximum and minimum.

- Uncheck Standard error of mean, Count, and Number missing from the continuous list
- Click OK
- In the variable browser, select Turning Circle and click Add
- In the variable browser, select Country and click Split By

Descriptive Statistics
Split By: Country

	Mean	Std. Dev.	Minimum	Maximum
Turning Circle, Total	38.586	3.132	32.000	47.000
Turning Circle, Japan	37.233	2.956	32.000	42.000
Turning Circle, Other	36.676	2.199	33.000	42.000
Turning Circle, USA	40.857	2.318	36.000	47.000

The table shows statistics broken down by the groups of the nominal variable. These results indicate that cars from the USA have the largest turning circle, and cars from Japan and other European countries have turning circles smaller than average.

Frequency Distribution

A **frequency distribution** table or graph can be useful for getting a sense of the distribution of your data. Histograms and pie charts divide your data into a number of ranges and display a bar or pie slice for each range. The height of each bar or size of pie slice is proportional to the fraction of your data which falls in that range. Frequency distribution tables and graphs can help identify some data characteristics that may influence which descriptive statistics and other analyses you will use.

Discussion

Histograms and pie charts

The graph of a frequency distribution is one of the quickest and easiest ways to get a picture of your data and perform a visual test for normality. A **histogram** divides your data into bars whose height is proportionate to the amount of data which falls in the range of the bar. A **pie chart** accomplishes the same thing with pie wedges. The advantage of a histogram is that the X axis has meaning, so you have two visual cues rather than one.

Pie charts can be useful for comparing portions of a whole, but they do not illustrate fine differences. It is easier to compare bar heights than to compare pie wedges, particularly when the differences between bars or wedges is small. When one range dominates your data, as in the percentage of the U.S. budget spent on defense, a pie chart offers a much more dramatic demonstration. When there are small differences between ranges, a histogram allows you to rank the ranges with greater ease.

z-score histograms

A **z-score histogram** converts the values so the mean is zero and the standard deviation is one. The scale is the same for all z-score histograms, regardless of the original units of measurement. This graph is particularly useful when you compare two measurements which were made on different scales. If the data are normally distributed, fewer than 1 out of 100 points will be higher than 3 or lower than -3, and only 5% of the points will be larger than 2 or smaller than -2.

Dialog box settings

When you create or edit frequency distribution results, you see the Frequency Distribution dialog box, in which you set or change the analysis parameters.

Frequency Distribution

Number of intervals: ☐ Show normal comparison

Do you wish to enter your own interval information?

☒ no ☐ yes width: initial value:

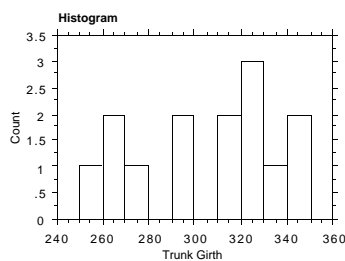
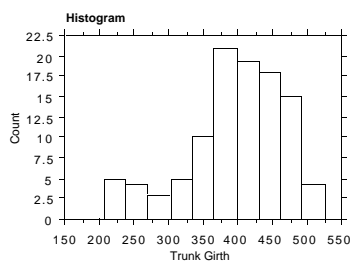
Intervals indicate: include:

Tables show: ☒ Counts ☐ Percents ☐ Relative frequencies

Histograms show:

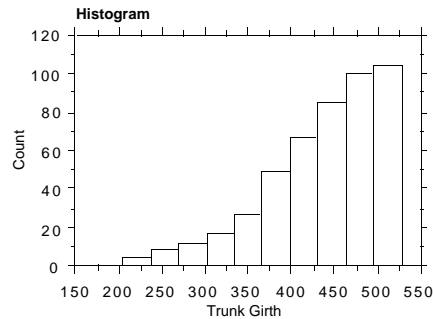
Intervals The top half of the dialog box controls the intervals in the analysis. The number of intervals is equal to the number of bars or pie slices in the resulting graph. The number of intervals defaults to 10 for continuous variables; you can enter a different number. The number of intervals for nominal variables is determined by the number of unique values of the variable. For continuous variables you can also set the interval width and the starting point. The width defaults to the range of the data divided by the number of intervals, and the initial value defaults to the lowest value in the variable, so the entire range is displayed. When you set a different width and initial value, the graph might not display the full range of the data. If this is the case, a note appears under the graph.

Changing interval width and initial value is useful when you want to examine one part of the distribution of your data more closely. The histogram on the left below was created using the defaults. The one on the right gives a closer look at the lower end of the distribution, since the interval width is set at 10 and the initial value at 250.



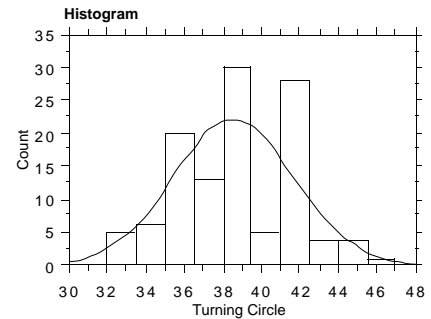
The intervals specified do not contain the entire range of the data.

When values in your data fall on an interval boundary, you can set the intervals to include the lowest value (which is the default) or the highest value. Suppose two adjacent intervals extend from 10 to 20 and 20 to 30 respectively. If one of your data values is 20, you need to know which interval to include it in. If intervals include their lowest value, 20 will go in the second interval; otherwise it will go in the first. Intervals can also be cumulative, rather than count, which is the default. Cumulative intervals include the totals of previous intervals, as shown below.



Normal comparisons The checkbox for showing normal comparisons applies only to continuous variables. If you check this option StatView draws in the histogram the expected frequency curve for a normal distribution with the same mean and standard deviation as the variable. Normal counts, percents and relative frequencies will also appear in the summary table.

Frequency Distribution for Turning Circle			
From (\geq)	To ($<$)	Count	Normal Count
32.000	33.500	5	3.997
33.500	35.000	6	8.573
35.000	36.500	20	14.683
36.500	38.000	13	20.078
38.000	39.500	30	21.924
39.500	41.000	5	19.115
41.000	42.500	28	13.308
42.500	44.000	4	7.398
44.000	45.500	4	3.284
45.500	47.000	1	1.164
Total		116	113.523



Counts, percents and relative frequencies You can display interval values as counts, percents or relative frequencies, in both the table and the histogram. The histogram can show only one scale; tables can include all three. Counts show how many observations fall inside each interval. Percents show what percentage of observations fall inside each interval, and relative frequencies show which fraction of values fall inside each interval; relative frequencies the same as percents divided by 100.

Frequency Distribution for Turning Circle				
From (\geq)	To ($<$)	Count	Rel. Freq.	Percent
32.000	33.500	5	.043	4.310
33.500	35.000	6	.052	5.172
35.000	36.500	20	.172	17.241
36.500	38.000	13	.112	11.207
38.000	39.500	30	.259	25.862
39.500	41.000	5	.043	4.310
41.000	42.500	28	.241	24.138
42.500	44.000	4	.034	3.448
44.000	45.500	4	.034	3.448
45.500	47.000	1	.009	.862
Total		116	1.000	100.000

Data requirements

Frequency distributions can be generated for nominal or continuous variables.

Variable browser buttons	
Add	To generate frequency distributions, select one or more nominal or continuous variables and click Add. Each additional variable assigned creates a new table or histogram.
Split By	When you assign one or more split-by variables to a frequency distribution table, results for each cell in the split-by variable(s) as well as totals for all groups are displayed in a single summary table. When you assign split-by variable(s) to a histogram or pie chart, a separate graph is generated for each cell.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 13](#). The histogram is the default result for a frequency distribution.

Summary Table	Table containing the upper and lower values and the count, relative frequency or percentage of total observations for each interval. A comparison to a normal distribution may also be displayed using the dialog box.
Histogram	Graph showing the percent, relative frequency, or number of observations in each interval as a bar chart. Comparison to a normal distribution may also be displayed using the dialog box.
Z Score Histogram	Graph showing the frequency distribution normalized so that the mean is zero and the standard deviation is one.
Pie Chart	Graph showing the number of observations in each interval as slices in a pie.

Templates

The following templates provide frequency distribution results.

Descriptive Statistics	Frequency Dist--Continuous	Frequency distribution table and histogram.
	Frequency Dist--Nominal	Frequency distribution table and histogram.
Graphs	Histogram	Histogram for continuous variable with normal curve.
	Pie Chart--Continuous	Pie chart for continuous variable.
	Pie Chart--Nominal	Pie chart for nominal variable.
	Scatter Matrix 4x4 w Histograms	4x4 matrix of scattergrams, with one scattergram for each X-Y pairing of continuous variables; diagonal cells have histograms with fitted normal curves.

	Scatter w Histograms	Scattergram for continuous variables; has histograms with fitted normal curves along top and right sides.
	Z-score Histogram	Z-score histogram for continuous variable.

Exercise

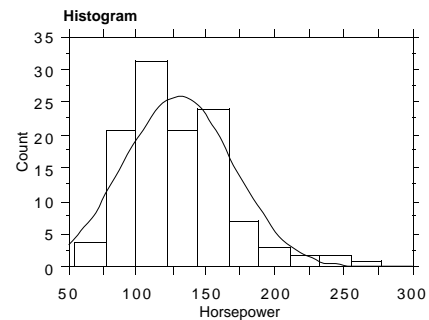
In this exercise you will create a frequency distribution using the sample Car Data. It has information on weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries. You will generate a frequency distribution of horsepower to determine whether horsepower follows a normal distribution.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Frequency Distribution, select Histogram and Summary Table and click Create Analysis
- Check Show normal comparison (turn the option on) and click OK

The Show normal comparison option overlays a normal distribution curve (sometimes called a bell-shaped curve) in the histogram and adds normal counts to the summary table.

- In the variable browser, select Horsepower and click Add

Frequency Distribution for Horsepower			
From (\geq)	To ($<$)	Count	Normal Count
55.000	77.300	4	7.255
77.300	99.600	21	14.976
99.600	121.900	31	22.774
121.900	144.200	21	25.516
144.200	166.500	24	21.063
166.500	188.800	7	12.809
188.800	211.100	3	5.738
211.100	233.400	2	1.893
233.400	255.700	2	.460
255.700	278.000	1	.082
	Total	116	112.567



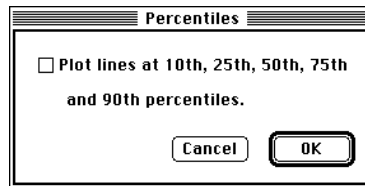
The data is positively skewed relative to a normal distribution. The histogram tells us there are no cars with horsepower in the lowest 10% or so of the hypothetical normal distribution curve.

Percentiles

A **percentiles plot** graphs observed values of a variable against its percentiles. It allows you to see the percentage of the data that is less than or equal to an observation. Percentiles plots are useful in comparing the distribution of different groups or variables. You can plot multiple variables in a single percentiles plot and use split-by variables to distinguish different groups. In addition, you can add reference lines to show the 10th, 25th, 50th, 75th, and 90th percentiles as well as display a table listing these values.

Dialog box settings

When you create or edit a percentiles plot, you see this dialog box. You can place lines the percentiles you choose:



Data requirements

Percentile tables and plots can be generated for one or more continuous variables.

Variable browser buttons	
Add	To generate a percentiles table or plot, select one or more continuous variables and click Add. Each additional variable assigned is added to the analysis.
Split By	The groups defined by any nominal variable(s) assigned using the Split By button appear in the same table or plot.

Results

The default output for this analysis is both the Summary Table and the Percentiles Plot.

Percentiles Summary Table	Table of values of the 10th, 25th, 50th (median), 75th and 90th percentiles.
Percentiles Plot	Values in each variable plotted against their percentiles. Lines indicating the 10th, 25th, 50th (median), 75th and 90th percentiles can be added to the plot using the dialog box.

Templates

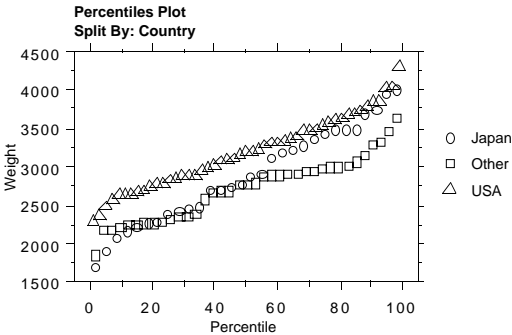
The following templates provide percentile results.

Descriptive Statistics	Percentiles	Percentiles summary table and plot for continuous variable.
Graphs	Compare Percentiles	Compare Percentiles plot for continuous variable and two-level nominal variable.

Exercise

This example uses data containing measurements of weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries. You will see whether there is a difference between the weights of cars from different countries.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Percentiles, select Percentiles Plot and click Create Analysis
- Click OK to accept the default analysis parameters
- In the variable browser, select Weight and click Add
- Select Country and click Split By



This graph shows how weights differ by country of manufacture. You can see that the 50th percentile, or median, of Japan and other countries is significantly lower than that of the U.S.

One Sample Analysis

StatView offers two one sample hypothesis tests: the *t*-test and the chi-square test. The ***t*-test** can be used to test the hypothesis that the mean of a normally distributed variable is equal to a value which you specify. The **chi-square test** can be used to test the hypothesis that the variance of a normally distributed variable is equal to a value which you specify. In each case, you can set a significance level and choose between one-tailed and two-tailed tests, as explained below.

Discussion

One sample *t*-test

The **one sample *t*-test** compares a sample mean to a hypothesized mean and determines the likelihood that the observed difference between the sample and hypothesized mean occurred by chance. The chance is reported as the *p* value. A ***p* value** close to 1 means it is very likely that the hypothesized and sample means are the same, since it is very likely that such a result would happen by chance if the null hypothesis of no difference is true. A small *p* value (for example, 0.01) means it is unlikely (only a one in 100 chance) that such a difference would occur by chance if the two means were the same. In such a case we would say that the sample mean is significantly different from the hypothesized value. The ***t* value** reported in the table expresses the difference between the mean and the hypothesized value in terms of the standard error.

Confidence interval

An alternative is to form a confidence interval around the sample mean. A **confidence interval** reports a range of values within which a particular parameter would likely occur if samples were taken repeatedly from the same distribution. If the sample mean is not significantly different from the hypothesized value, the hypothesized value is likely to be included in the confidence interval. Alternatively, when the hypothesized value is not contained in the confidence interval, the sample mean is probably not equal to that value, and the two means can be declared significantly different. Thus, the *t*-test and the confidence interval procedures provide similar information in different ways. Confidence intervals can be created using the One Sample Analysis dialog box.

Chi-square test

The **chi-square test** tests the hypothesis that the variance of a sample from a normal distribution is equal to some hypothesized value. The test compares the sample variance with the hypothesized variance and determines the likelihood that the observed discrepancy between the two occurred by chance. This likelihood is reported as the p value. A p value close to 1 means it is very likely that the hypothesized and sample variances are the same, since it is probable that such a result would happen by chance if the null hypothesis of no difference is true. A small p value (for example, 0.01) means it is unlikely (only a one in 100 chance) that the observed discrepancy would occur by chance if the two variances were the same. In such a case we would say that the sample variance is significantly different from the hypothesized variance.

Confidence interval

An alternative is to form a confidence interval around the variance of the sample. A **confidence interval** reports a range of values within which a particular parameter would most likely occur if samples were taken from the same distribution over and over again. If the sample variance is not significantly different from the hypothesized value, the hypothesized value is likely to be included in the confidence interval. Alternatively, when the hypothesized variance is not contained in the confidence interval, the sample variance is probably not equal to that value, and the two can be declared statistically different. Thus, the chi-square test and the confidence interval procedures provide similar information in different ways. Confidence intervals can be created using the One Sample Analysis dialog box.

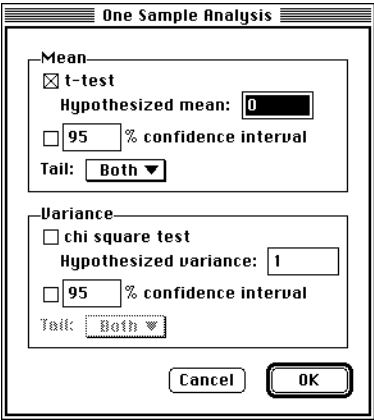
Tail

You can perform the t -test or chi-square test as a one-tailed or two-tailed test. The One Sample Analysis dialog box offers the choice of upper, lower or both tails. By default, the tests consider both possibilities: that the sample's mean/variance is larger than the hypothesized mean/variance, and that the hypothesized mean/variance is larger than the sample's. Such a test is called a two-sided or **two-tailed test**. A **one-tailed test** considers a difference in only one direction; that the difference is either greater than (upper), or less than (lower) the hypothesized mean or hypothesized variance.

There are rare instances in which only one direction of difference is possible. In such cases, a one-sided test is more sensitive to differences than a two-sided test since it considers differences in only one direction. A great deal of knowledge about the nature of the problem at hand is necessary for the one-sided test to be valid. It is essential to be sure that a difference in the other direction is physically impossible.

Dialog box settings

When you create or edit one sample analysis results, you set the analysis parameters in this dialog box:



You can elect to perform an analysis of means (*t*-test) or variances (chi-square test) and set confidence intervals for both. If you choose a *t*-test, you compare the sample mean to a hypothesized mean, which you enter yourself in the text box. If you choose a chi-square test, you compare the sample variance to a hypothesized population variance, which you enter yourself in the text box. The hypothesized mean or variance embodies the question that you want the analysis to answer; you have reason to suspect that the mean or variance has a certain value.

For both tests, you can specify whether the test/confidence interval is two-tailed or one-tailed, and if one-tailed, which tail is to be used in the analysis. If you intend to use a one-tailed test, please read the caution in the earlier section, [“Tail,” p. 24](#).

Data requirements

A one sample analysis (*t*-test or chi-square) requires one or more continuous variables.

Variable browser buttons	
Add	To generate a one sample analysis, select one or more continuous variables and click Add. Each additional continuous variable assigned is added to the existing table.
Split By	When you assign one or more split-by variable to an one sample analysis table, results for each cell in the split-by variable(s) as well as totals for all groups are displayed in a single summary table

Results

For explanation of the results, please see the preceding [“Discussion,” p. 23.](#)

Mean	One Sample <i>t</i> -test	Table generated if only <i>t</i> -test is selected. This table shows the sample mean, the degrees of freedom, and the <i>t</i> value and the <i>p</i> value for the difference between the actual and hypothesized value.
	Confidence Interval	Table generated if only confidence interval is selected. This table shows the sample mean and the upper and/or lower confidence intervals as set in the dialog box.
	One Sample Analysis	Table generated if both <i>t</i> -test and confidence interval are selected. This table combines the above tables.
Variance	Chi-square test	Table generated if only chi-square test is selected. This table shows the sample variance, the degrees of freedom, the chi-square, and the <i>p</i> value for the test.
	Confidence Interval	Table generated if only confidence interval is selected. This table shows the variance and the upper and/or lower confidence intervals as set in the dialog box.
	One Sample Analysis	Table generated if both chi-square test and confidence interval are selected. This table combines the above tables.

Templates

The following templates provide one sample analysis results.

ANOVA and <i>t</i> -tests	One-Group Variance Test	One sample analysis table with 95% confidence intervals.
	<i>t</i> -Test (One Group)	One sample <i>t</i> -test table.

Exercise

In this exercise you perform a one sample *t*-test on data from blood lipid screenings of medical students. You want to know whether the mean cholesterol level is significantly greater than 190, a point above which cholesterol levels may be unhealthy. You test the null hypothesis that the mean value for cholesterol is 190. If you reject the null hypothesis, you can conclude that the mean differs significantly from 190. Because a one-tailed test would be inappropriate, you will do a two-tailed *t*-test.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select One Sample Analysis and click Create Analysis
- For hypothesized mean, type 190 and click OK

- In the variable browser, select Cholesterol and click Add

One Sample t-test**Hypothesized Mean = 190**

	Mean	DF	t-Value	P-Value
Cholesterol	191.232	94	.336	.7373

The mean is slightly higher than the hypothesized value of 190. However, although the mean is in fact higher, you cannot reject the null hypothesis that the mean is 190 because 191.232 is well within the range of sampling variance. The p value indicates you would see a difference of this magnitude by chance more than 73% of the time.

Paired Comparisons

There are several ways you can compare two samples of experimental units. One approach compares the means of the two samples by performing a t -test. If the samples are naturally paired in some way, a paired t -test is appropriate. The most common case is a **paired comparison** of two measurements taken from the same experimental unit at different times or under different conditions.

If instead you want to compare average measurements for the two groups, rather than paired variables, an unpaired t -test is appropriate. Unpaired comparisons are described in the next chapter, [p. 37](#). Note that the paired t -test is the equivalent of a repeated measures ANOVA (see [“ANOVA,” p. 73](#)) for two repeated measurements.

Another approach examines the relationship or closeness of association between properties of paired experimental units. For example, a researcher may question how closely a bird’s body length follows its wing span. This can be done using a correlation analysis. The paired t -test and correlation analysis are described below. The tests assume that both samples are normally distributed and have the same variance. Extensions of these techniques for dealing with more than two groups or data that is not normally distributed are discussed in the chapters [“ANOVA,” p. 73](#) and [“Nonparametrics,” p. 119](#).

Discussion

Paired t -test

The most common use of a paired t -test is the comparison of two measurements from the same individual or experimental unit. The two measurements can be made at different times or under different conditions. The **paired t -test** tests the hypothesis that the mean of the differences between pairs of experimental units is equal to some hypothesized value, usually set at zero. An hypothesized value of zero is equivalent to the hypothesis that there is no difference between the two samples. The paired t -test compares the two samples and determines the likelihood of the observed difference occurring by chance. The chance is reported as the p value. A small p value (for example, 0.01) means it is unlikely (only a one in 100 chance) that such a mean difference would occur by chance under the assumption that the mean difference were zero. In such a case we would say that there is a statistically significant difference between the

two groups. The t value reported in the table expresses the difference between the mean difference and the hypothesized value in terms of the standard error.

A paired t -test is more powerful than the unpaired t -test, because it takes into account the fact that measurements from the same unit tend to be more similar than measurements from different units. For example, in a test administered before and after a training program, the usual (unpaired) t -test may not detect consistent but small increases in each individual's scores. The paired t -test is more sensitive to the fact that one measurement of each pair essentially serves as a control for the other.

The paired t -test is also appropriate when some other natural pairing exists. For example, a survey of husbands and wives is designed to test for differences of opinion on particular issues. Each couple's responses are viewed as a pair and tested for differences with a paired t -test. In some designed experiments, subjects are selected for similarities of age, race or sex. A paired t -test is appropriate to use on such measurements. The critical issue is whether a pair's responses are more likely to be similar than responses from random experimental units. When the pair's responses are likely to be consistently more similar, a paired t -test is more powerful than an unpaired t -test.

You may also wish to examine your data graphically using a cell plot. See [“Cell Plots,” p. 237](#), for a discussion of cell plots.

Mean difference confidence interval

An alternative is to form a confidence interval for the mean of the difference between the two measurements for each experimental unit. When the two measures are not significantly different, the value of zero is likely to be included in the confidence interval. Alternatively, when zero is not contained in the confidence interval, the difference is probably not zero, and the measures can be declared significantly different.

Tail

You can perform the paired t -test as a one-tailed or two-tailed test. The Paired Comparisons dialog box offers the choice of upper, lower or both tails. By default, the tests consider both possibilities: that the first group's mean is larger than the second group's mean, and that the second group's mean will be larger than the first's. Such a test is called a two-sided or **two-tailed test**.

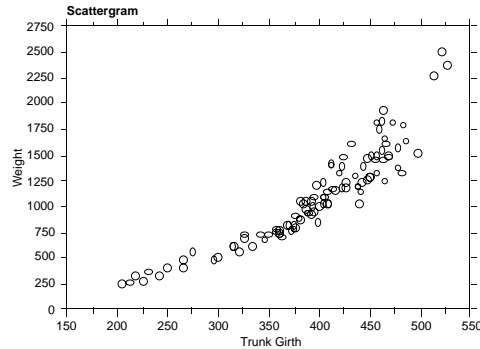
There are rare instances in which only one direction of difference is possible. In such cases, a one-sided test is more sensitive to differences than a two-sided test since it considers differences in only one direction. A great deal of knowledge about the nature of the problem at hand is necessary for the one-sided test to be valid. It is essential to be sure that a difference in the other direction is physically impossible.

A **one-tailed test** considers a difference in means in only one direction; that the difference is either greater than (upper tail), or less than (lower tail) the hypothesized difference or hypothesized correlation.

Z-test for correlation coefficients

The paired t -test is used to compare the means of measurements of the same variable taken at different times. Comparison of two variables which measure different things requires a different approach. The **z-test** tests the hypothesis that the correlation coefficient is equal to an hypothesized value, usually set at zero. An hypothesized correlation coefficient of zero is equivalent to the hypothesis that there is no correlation between variables. The z-test compares the two groups and determines the likelihood of the observed correlation occurring by chance. The chance is reported as the p value. A small **p value** (for example, 0.01) means it is unlikely (only a one in 100 chance) that such a correlation would occur by chance. In such a case we would say that there is a statistically significant difference between the two groups.

The most powerful tool for examining relationships of this sort is the **bivariate scattergram** (see [“Bivariate Plots,” p. 221](#)). A bivariate scattergram plots the values of one variable on the X axis and the values of the other on the Y axis. It is easy to see whether a relationship exists. For example, this scattergram shows a near linear relationship between two variables:



Correlation coefficient

The **correlation coefficient** is a more quantitative measure of the relationship between two variables than the bivariate scattergram. A correlation coefficient of -1 indicates that large values of one variable are exactly associated with small values of the other variable. A correlation coefficient of $+1$ indicates large values of one variable are exactly associated with large values of the other variable. The scattergram above has a correlation coefficient of 0.916.

The distinction between statistical significance and practical significance is important when using the correlation coefficient. The level of correlation that is practically significant varies from situation to situation. Generally, unless the absolute value of the correlation is greater than 0.5, the relationship between variables is not important. However, a correlation of 0.1 may be statistically significant with a large enough sample. This seems contradictory, but it means that a large enough sample size lends significance to a weak correlation. The statistical significance indicates that the value of the correlation coefficient is not zero; the decision remains whether the correlation is large enough to be important.

Correlation is useful for testing the relationship between more than two variables. The correlation of many variables can be displayed as a correlation matrix (table). The Correlation/

Covariance analysis, discussed in [“Correlation and Covariance,” p. 43](#), produces such a table. A correlation coefficient for all pairs of variables appears in the cell at the intersection of the variables’ respective row and column. A partial correlation matrix removes the linear effect of one or more variables before examining the relationships of the other variables. For more information about correlation, see [“Correlation and Covariance,” p. 43](#), and [“Nonparametrics,” p. 119](#).

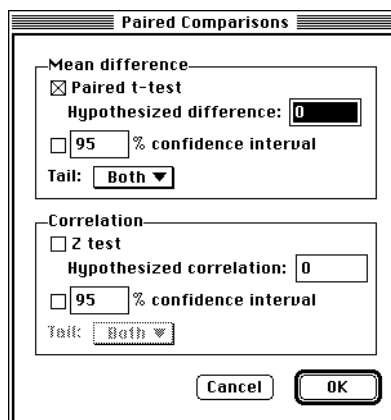
The correlation coefficient measures only the *linear* relationship between variables. It cannot reveal anything about non-linear relationships and can be misleading if used with them. Polynomial relationships can be examined using polynomial regression (see [“Regression,” p. 51](#)). In some cases, you may be able to transform the data (using the formula capability) so that the relationship becomes linear. If it is possible to divide the independent variable into groups, you can test for the presence of a more general relationship than simply linear between these groups and a dependent variable, by using ANOVA (see [“ANOVA,” p. 73](#)).

Tail

You can also perform a z -test as a one- or two-tailed test. The Paired Comparisons dialog box offers the choice of upper, lower or both tails. By default the test considers both possibilities: that the correlation coefficient is either smaller or larger than an hypothesized value. A great deal of knowledge about the nature of the problem at hand is necessary for the one-sided test to be valid. You must be certain before you start the experiment that a difference in only one direction is possible.

Dialog box settings

When you create or edit paired comparison results, you set the analysis parameters in this dialog box:



You can choose to analyze the mean difference, correlation, or both, by clicking in the appropriate checkboxes. The paired t -test computes a paired t value between two variables when the row entry for each variable is a measure on the same subject. The z -test uses Fisher’s R to z

transformation to test the hypothesis that the correlation between two variables is equal to the specified value. You can set confidence intervals for both tests, and designate either as two-tailed or one-tailed (upper or lower). Please read the caution in the [“Discussion,” p. 29](#), if you are using a one-tailed test.

Data requirements

Paired comparisons require two or more continuous variables. If more than two continuous variables are assigned, paired comparisons are calculated for all possible variable pairs.

The data for each sample of the paired comparison must be located in a single continuous variable (column). Each row entry for the two columns being analyzed must be a measure for the same subject or for observations that are naturally paired. For an introduction to dataset organization, see [“Dataset structure,” p. 49 of *Using StatView*](#). In addition, the [“Exercises,” p. 34](#), will help you see how to organize your data for this analysis.

Variable browser buttons	
Add	To generate paired comparisons, select a two or more continuous variables and click Add. Each additional variable is added to the summary table which expands to include the new variable(s).
Split By	When you assign one or more split-by variables to a paired comparisons table, results for each cell in the split-by variable(s) as well as totals for all groups are displayed in a single summary table.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 29](#). The hypothesis being tested is shown in the title of the table.

Mean difference	Paired <i>t</i> -test	Generated if only paired <i>t</i> -test is selected. This table shows the mean of the differences between pairs, the degrees of freedom, the <i>t</i> value and the <i>p</i> value for the mean difference.
	Confidence Interval	Generated if only confidence interval is selected. This table shows the difference between the group means and the upper and lower confidence intervals for that difference as set in dialog box.
	Paired Means Comparison	Generated if both paired <i>t</i> -test and confidence intervals are selected. This table combines the above tables.

Correlation	Fisher's R to Z	Generated if only z -test is selected. This table shows the correlation between variables, the number of paired observations, and the z value and the p value for the correlation.
	Confidence Interval	Generated if only confidence interval is selected. This table shows the correlation coefficient, and the upper and lower confidence intervals as set in the dialog box.
	Correlation Coefficient	Generated if both z -test and confidence intervals are selected. This table combines the above tables.

Templates

The following templates provide paired comparisons results.

ANOVA and t -tests	t -Test (Paired)	Paired t -test table with 95% confidence interval.
Correlations	Correlation Z -Test	Fisher's R to Z with 95% confidence interval.

Exercises

Paired t -test

In this exercise you will perform a paired t -test. The data used in this exercise comes from blood lipid screenings of medical students. You will determine whether initial triglyceride levels are different from those measured in the same subjects after three years.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Paired Comparisons and click Create Analysis
- Click OK to accept the default analysis parameters
- In the variable browser, select Triglycerides and Trig-3 yrs and click Add
Control-click (Windows) or Command-click (Macintosh) to select several nonadjacent variables at a time

Paired t -test				
Hypothesized Difference = 0				
	Mean Diff.	DF	t -Value	P-Value
Triglycerides, Trig-3yrs	3.419	42	.386	.7015

From this paired t -test, you can accept the hypothesis of no difference between means of the two groups. The mean difference is so small the p value indicates you are likely to see a difference of this magnitude by chance 70% of the time. You are now finished with this example. You may save the view to any folder and open it with the same dataset to perform any further analyses you wish.

Z-test

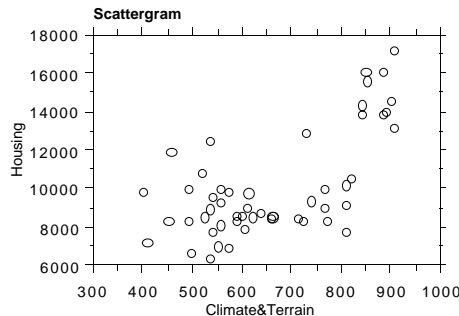
The previous exercise compares the means of groups with the same variable: triglyceride levels. Comparison of two variables which measure different things on the same or paired experimental units requires a different approach.

In this exercise you create a scattergram and calculate a correlation coefficient to determine the degree of linear relationship between two variables. The data you use rates a number of different western cities by nine criteria. You will discover whether better climate is accompanied by an increase in housing costs.

- Open Western States Rated Data from the Sample Data folder

For Climate & Terrain, a higher score is better; for Housing, the lower the score the better. The first step is to create a bivariate plot to see how linear the relationship is between the two variables in question.

- From the Analyze menu, select New View
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Click OK to accept the default parameters
- In the variable browser, select Climate&Terrain and click X Variable
- In the variable browser, select Housing and click Y Variable



You can see that there is some degree of linear relationship between higher housing costs and more desirable climate (as defined by the criteria of the study). To confirm this judgement, examine the correlation coefficient for these two variables with a paired comparisons test.

You can avoid the step of assigning the Climate&Terrain and Housing variables again, by keeping the scattergram selected and then requesting Paired Comparisons.

- Make sure the scattergram is still selected (has black handles)
- In the analysis browser, double-click Paired Comparisons
- Uncheck Paired *t*-test, and check Z test under Correlation
(We leave the hypothesized correlation set to 0 to test the hypothesis of no relationship between the variables. This test will produce a correlation coefficient and a *p* value indicating the likelihood of this correlation occurring by chance.)
- Click OK

Fisher's R to Z**Hypothesized Correlation = 0**

	Correlation	Count	Z-Value	P-Value
Climate&Terrain, Housing	.659	52	5.533	<.0001

From this test, you can conclude that a positive correlation exists between Climate&Terrain and Housing because of a significant correlation coefficient and a p value that indicates a very low likelihood that this degree of correlation could occur by chance.

Unpaired Comparisons

Unpaired comparisons are comparisons made between the average measurements of two groups rather than between paired variables within those groups. StatView performs an unpaired t -test for comparing two means and an unpaired F -test for comparing two variances, both under the assumption that your data is normally distributed. If you want to compare paired measurements of the variables rather than averages for the two groups, read about paired comparisons in the preceding chapter, [“Paired Comparisons,” p. 29](#).

Discussion

Unpaired t -test

A measurement taken from two different groups raises the question: on the average, are the measurements for one group different from the measurements for the other group? This can be answered by performing an unpaired t -test on the measurements.

The **unpaired t -test** compares the means of two groups and determines the likelihood of the observed difference occurring by chance. The chance is reported as the p value. A p value close to 1 means it is very likely that the two groups have the same mean, since it is very likely that such a result would happen by chance if the null hypothesis of no difference between groups is true. A small p value (for example, 0.01) means it is unlikely (only a one in 100 chance) that such a difference would occur by chance if the two groups had the same mean. In such a case we would say that there is a significant difference between the two means. The t value expresses the difference between the mean difference and the hypothesized value in terms of the standard error.

You may also wish to examine your data graphically using a cell plot. See [“Cell Plots,” p. 237](#), for a discussion of cell plots.

Confidence interval

An alternative is to form a confidence interval for the difference between the means of the two groups. When the two means are not significantly different, the value of zero is likely to be included in the confidence interval. Alternatively, when zero is not contained in the confi-

dence interval, the difference is probably not zero, and the measures can be declared significantly different. Confidence intervals can be created using the dialog box.

Tail

The unpaired t -test assumes that two groups are normally distributed and have the same variance. It is usually difficult to predict the direction in which the differences will lie. By default, the t -test considers both possibilities: that the first group's mean will be larger than the second group's mean, and that the second group's mean will be larger than the first's. Such a test is called a two-sided or two-tailed test.

A one-sided test is more sensitive to differences than a two-sided test since it considers differences in only one direction. A great deal of knowledge about the nature of the problem at hand is necessary for the one-sided test to be valid. You must be certain before you start the experiment that a difference in only one direction is possible.

F-test

A comparison of the variance of groups of measurements can be useful to validate the assumptions of the t -test, and for other purposes. For example, a mechanical part is manufactured by two different methods. You want to know if the size of the part differs between the two methods, and also whether one method or the other produces more consistent results. The F -test for variances shows whether the variance of one group is smaller, larger or equal to the variance of the other group.

The F -test depends on two parameters: the degrees of freedom for each of the two groups. This will be equal to the number of observations in the group minus one. Since the F -test is formed as a ratio of the two variances, the parameters are referred to as numerator degrees of freedom and denominator degrees of freedom.

Confidence interval

An alternative is to form a confidence interval for the ratio of the variances of the two groups. When the two variances are not significantly different, the value of 1 is likely to be included in the confidence interval. Alternatively, when 1 is not contained in the confidence interval, the variances are probably not equal and can be declared significantly different. Confidence intervals can be created in the dialog box.

Tail

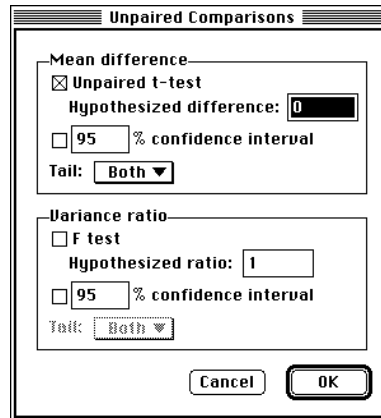
It is usually difficult to predict the direction in which the variance differences will lie. By default, the F -test considers both possibilities: that the first group's variance will be larger than the second group's, and that the second group's variance will be larger than the first's. Such a test is called a two-sided or two-tailed test.

A one-sided test is more sensitive to differences than a two-sided test since it considers differences in only one direction. A great deal of knowledge about the nature of the problem at

hand is necessary for the one-sided test to be valid. You must be certain before you start the experiment that a difference in only one direction is possible.

Dialog box settings

When you create or edit unpaired comparisons results, you set the analysis parameters in this dialog box:



You can choose to analyze the mean difference, variance ratio, or both by clicking in the appropriate checkboxes. The unpaired t -test defaults to a hypothesized value of zero. The F -test tests the hypothesis that the ratio of the two variances is equal to the hypothesized ratio, which defaults to one. You can set confidence intervals for both tests, and designate either as two-tailed or one-tailed (upper or lower). Please read the caution in the [“Discussion,” p. 37](#), if you are using a one tailed test.

Data requirements

Unpaired comparisons require a single nominal grouping variable with two or more groups and one continuous variable. If the nominal variable contains more than two groups, unpaired comparisons will be calculated for all possible pairs of groups.

To compare groups, your data must be organized in a way that allows the unpaired comparison analysis to identify which group an observation belongs to. This can be done using a column containing a separate nominal grouping variable or by using a compact variable. For an introduction to dataset organization, see [“Dataset structure,” p. 49 of *Using StatView*](#). In addition, the [“Exercise,” p. 41](#), will help you see how to organize your data for this particular analysis.

Standard layout

The dataset below shows one way to organize your data if you wished to perform an analysis comparing cholesterol levels for males and females.

	Name	Gender	Weight	Cholesterol
1	J. Suds	Male	145	168
2	H. Fitz	Female	123	167
3	R. Blunt	Male	245	265
4	T. Stout	Male	223	187
5	S. Small	Female	142	202

The cholesterol values for *both* males and females appear in a single column. The variable Gender is a separate nominal column and acts as a grouping variable that identifies the group (Male or Female) for each Cholesterol measurement. There will be one row in the dataset for each subject in the analysis.

Compact variable

If you prefer to place different groups in separate columns, StatView offers an alternative to the data organization shown above. In this dataset organization, the observations for each group appear in a single column. Your dataset will contain as many columns as there are groups being compared. If you enter your data this way, you must create a simple compact variable in order for the analysis to know which group each observation belongs to. The cholesterol measurements for male and female from the above dataset look like this in a compact variable format:

	Cholesterol	
	Male	Female
1	168	167
2	265	202
3	187	●

The male cholesterol measurements are all placed in one column and the female cholesterol measurements in another. The column identifies the group, not the row. If there are unequal numbers of observations in the two groups, missing values (.) are automatically inserted in the column with fewer observations. These missing values are ignored in the analysis.

If you plan to use a compact variable, please read the discussion [“Compact variables,” p. 84 of Using StatView.](#)

Variable browser buttons	
Add	To generate unpaired comparisons, select a single nominal grouping variable and a single continuous variable and click Add. Each additional nominal variable assigned creates a new analysis using the new nominal variable and the old continuous variable. Each additional continuous variable assigned creates a new analysis using the new continuous variable and the old nominal variable.
Split By	When you assign one or more split-by variables to an unpaired comparisons table, results for each cell in the split-by variable(s) as well as totals for all groups are displayed in a single table.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 37](#). The hypothesis being tested is shown in the title of the table.

Mean difference	Unpaired <i>t</i> -test	Generated if only unpaired <i>t</i> -test is selected. This table shows the difference between the group means, the degrees of freedom, and the <i>t</i> value and the <i>p</i> value for the mean difference.
	Confidence Interval	Generated if only confidence interval is selected. This table shows the difference between the group means and the upper and lower confidence intervals as set in dialog box.
	Unpaired Means Comparison	Generated if both unpaired <i>t</i> -test and confidence intervals are selected. This table combines the above tables.
	Group Info	Always generated and shows the count, mean, variance, standard deviation, and standard error for each group.
Variance ratio	<i>F</i> -test	Generated if only <i>F</i> -test is selected. This table shows the ratio of the group variances, the degrees of freedom in the numerator and denominator, and the <i>F</i> value and <i>p</i> value for the variance ratio.
	Confidence Interval	Generated if only confidence interval is selected. This table shows the ratio of the group variances, and the upper and lower confidence intervals as set in the dialog box.
	Variance Comparison	Generated if both <i>F</i> -test and confidence intervals are selected. This table combines the above tables.
	Group Info	Always generated and shows the count, mean, variance, standard deviation, and standard error for each group.

Templates

The following templates provide unpaired comparison results.

ANOVA and <i>t</i> -tests	Equality of Variances <i>F</i> Test	Variance comparison <i>F</i> test and group info tables.
	<i>t</i> -Test (Unpaired)	Unpaired means comparison, variance comparison, and group info tables.

Exercise

In this exercise you perform an unpaired *t*-test on census information for 506 housing tracts in the Boston area. You will examine two groups of housing tracts, those near the Charles River and those farther away from it. You will find out whether the median value of owner-occupied homes varies depending on how far houses are located from the river. To do this, you will test the null hypothesis that no difference in median housing prices exists.

- Open Boston Housing Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Unpaired Comparisons and click Create Analysis
- Click OK to accept the default analysis parameters
(We leave the hypothesized difference 0 to test the hypothesis of no difference between means of the two groups.)
- In the variable browser, select Median Value and click Add
- In the variable browser, select Charles and click Add

Charles has a G usage marker indicating it acts as a grouping variable in the analysis.

Unpaired t-test for Median Value

Grouping Variable: Charles

Hypothesized Difference = 0

	Mean Diff.	DF	t-Value	P-Value
Near, Far	6.346	504	3.996	<.0001

Group Info for Median Value

Grouping Variable: Charles

	Count	Mean	Variance	Std. Dev.	Std. Err
Near	35	28.440	139.633	11.817	1.997
Far	471	22.094	77.993	8.831	.407

You can reject the null hypothesis of no difference between the price of houses near to and far from the Charles River. The mean value is significantly higher for housing near the river than for housing far from it. The low p value indicates a probability of less than one in 10,000 that such a difference would occur by chance.

Correlation and Covariance

Correlation and covariance values indicate the degree of **linear relationship** between two variables. Computing these values generally requires a single sample with two sets of observed values on each subject or sampling unit. Correlation and covariance measure only the linear relationship between two variables. If the relationship is other than linear, these coefficients can be very misleading. Before relying on the correlation or covariance of two variables as a measure of their association, you should examine a scattergram of the two variables. In this way you can make sure there is not some nonlinear relationship which the correlation or covariance would not detect.

Discussion

Correlation coefficient

The Pearson **correlation coefficient** has an absolute value between 0 and 1, with 1 indicating a perfect linear relationship and 0 meaning no linear relationship exists. When two variables increase or decrease proportionately (as one variable increases, the other variable increases; when one decreases, so does the other), a positive correlation between them exists. When one variable increases when the other decreases proportionately, there is a negative correlation (inverse relationship). A correlation of exactly 0 almost never occurs in practice. If an exact linear relationship exists among some of the variables, the matrix is said to be singular. A **singular matrix** is not invertible, so it is not possible to compute partial correlations or Bartlett's test of sphericity. If this occurs, an error message tells you the correlation matrix is singular.

Correlation matrix

When many variables are measured, it is useful to display the correlation coefficients in a **correlation matrix**, a table in which each row or column represents a different variable in the dataset. The cell at the intersection of a row and column contains the correlation coefficient for the two variables the row and column represent. In an exercise later, you will create and interpret a correlation matrix. Other values, such as the probability that a particular correlation is different from 0, may also be displayed in similar tables.

Correlation Matrix

	Age	Weight	Height	Skinfold	Systolic BP	Diastolic BP
Age	1.000	.089	-.021	.106	.024	-.064
Weight	.089	1.000	.698	.074	.157	.136
Height	-.021	.698	1.000	-.138	.084	.063
Skinfold	.106	.074	-.138	1.000	-.099	-.038
Systolic BP	.024	.157	.084	-.099	1.000	.335
Diastolic BP	-.064	.136	.063	-.038	.335	1.000

95 observations were used in this computation.

You have the option of saving the correlation matrix as a new dataset.

Fisher's r to z

To determine if a correlation coefficient is significantly different from zero, a **Fisher's r to z transformation** is carried out on the correlation. This transforms the correlation coefficient to a variable with a standard normal distribution, allowing a probability level (p value) to be calculated for the null hypothesis that the correlation is equal to zero. One caution about judging correlation coefficients based on their significance levels: for a large enough sample, any correlation coefficient that is not exactly equal to zero will have a significant probability level.

The distinction between statistical significance and practical significance is important when using the correlation coefficient. The level of correlation that has practical significance will vary from situation to situation. Generally, unless the absolute value of the correlation is greater than 0.5, the relationship between two variables is probably not of much importance. On the other hand, with a large enough sample, a correlation of 0.1 may be significant. This may seem contradictory. It simply means that when the sample size is large enough, even a weak correlation can safely be considered different from no correlation at all. The statistical significance simply indicates that the value of the correlation coefficient is not 0; it is up to you to decide whether the magnitude of the correlation is large enough to be of importance.

Bartlett's test of sphericity

One special correlation pattern which may exist among a set of variables is **sphericity**. It means that all the variables in question are uncorrelated with each other, resulting in a correlation matrix with zeroes everywhere except the diagonal. You can test to see if a correlation matrix conforms to this pattern by requesting **Bartlett's test of sphericity**. A high chi-square and associated low p value imply that the null hypothesis of no correlation between variables can be rejected. If the matrix is **singular** (an exact linear relationship exists among some of the variables) it is not possible to compute Bartlett's test and you see an error message noting that the matrix is singular.

Confidence intervals

You may also form a **confidence interval** for the correlation between pairs of samples of experimental units. When two variables are not correlated, the value of zero is likely to be included in the confidence interval. Alternatively, when zero is not contained in the confidence inter-

val, the correlation is probably not zero, and the measures may be declared significantly correlated. You create confidence intervals using the dialog box.

Listwise/pairwise deletion

Sometimes a correlation coefficient in a correlation matrix may not agree with a value reported as a single correlation when the correlation coefficient is calculated for just two of the variables included in the matrix. This discrepancy may arise because StatView, by default, eliminates all rows that have a missing value for any of the variables for which correlations are calculated. This procedure is called **listwise deletion**. Such a correlation matrix has certain desirable statistical properties when used in further calculations, even though the deleting of cases may obscure some relationships in the data. You can override this by choosing the **pairwise deletion** option in the Correlation/Covariance dialog box; if you do so, partial correlations and Bartlett's test of sphericity are not calculated.

Covariance

When several variables are studied simultaneously, it is often of interest to determine if any or all of the variables are related to each other. One way of doing this is to calculate a measure of how much changes in one variable affect the values of the other variables. When we consider changes in the linear sense, the measure is known as **covariance**. By a linear sense, we mean that a straight line on a graph would be a good representation of the relationship between the two variables. As one variable increases, the other consistently either increases or decreases. The covariance between two variables is measured on a scale which is heavily influenced by the magnitudes of the variables involved, and may be hard to interpret if the variables being studied are measured on vastly differing scales. For this reason, the correlation coefficient is usually preferred as a measure of linear relationships, because it is standardized to be in the range of -1 to 1 , and is not affected by the scale of measurement.

Partial correlation

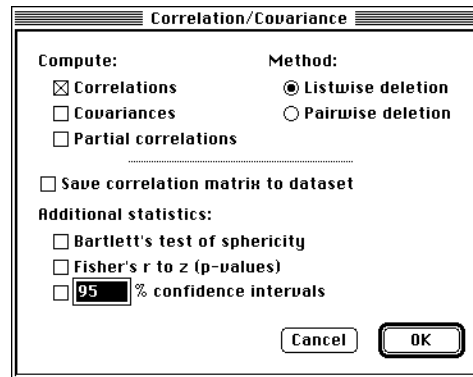
A correlation matrix may involve many variables. Since the entries in the matrix only address the relation between two variables at a time, there are many situations where the correlation coefficient may not accurately measure the strength of the relationship of interest. For example, suppose we have a dataset consisting of age, weight and a score on a fitness test.

The correlation between weight and the fitness score may mislead us into believing that there is a strong relationship between these two variables, when in fact it may be just the effect of age, since that is related to both weight and fitness score. What we would like is a measure of correlation between weight and fitness score with the effects of age removed. This is the basic idea behind partial correlation. The **partial correlation** of two variables with respect to a third is the correlation of the two variables after the linear effect of the third variable has been removed. Notice that, like the regular correlation coefficient, if non-linear relationships exist, the partial correlation coefficient may not be valid. Nevertheless, the partial correlation coefficient can be a useful tool when you are studying a set of closely related variables.

If the correlation matrix is **singular** (an exact linear relationship exists among some of the variables) it is not possible to compute partial correlations and you will see an error message noting that the matrix is singular.

Dialog box settings

When you create or edit a correlation or covariance analysis, you set the analysis parameters in this dialog box:



Select from correlation, covariance and partial correlation by clicking in the checkboxes at the top. Rows are eliminated from the analysis if they contain a missing value (listwise deletion) unless you select pairwise deletion instead. (A matrix formed with the pairwise method should not be used as input for a factor analysis.) For a further discussion of listwise and pairwise deletion, see the preceding section, [“Listwise/pairwise deletion,” p. 45](#).

At the bottom of the dialog box you can choose to generate the following additional statistics: Bartlett’s test of sphericity, Fisher’s r to z (p values), and a user specified confidence interval around the correlation coefficients.

Save correlation matrix to dataset If you check save correlation matrix, the computed correlation matrix is saved to a new dataset titled Correlation Matrix. The dataset will have as many columns and rows as variables assigned to the correlation. The names of each column are *Cor* “Variable name” where “Variable name” is the name of one of the assigned variables for the correlation.

Note that the correlation matrix dataset is a very special dataset with many features. The dataset is linked to the correlation analysis. If you change the parameters of the analysis or any of the input data, the dataset will *automatically* update to reflect the new correlation matrix. If you close the view that contains the correlation analysis, this correlation dataset will close as well. When the view is reopened, the correlation matrix dataset is automatically recreated. Please note that because this dataset is linked to your analysis, it is a “read only” dataset; you can not change any value in the dataset (except the formatting) until you break the link between the dataset and the analysis. If you plan to use this correlation matrix as an input to another analysis, such as factor analysis, the analysis must appear in the same view as the correlation analysis that dataset is associated with.

You cannot close the matrix dataset, but can hide it by clicking the close box. It is merely hidden and is accessible through the Window menu. To sever the link between the dataset and the correlation analysis, you need to choose Save As from the File menu and save the dataset under a different name. This will save on the disk a copy of the correlation matrix as a dataset. You can then open this dataset as you would any other dataset. When you save a copy of the correlation matrix dataset to your disk, StatView automatically appends the letters “UE” to the beginning of the column names to indicate that these columns are now user entered.

Data requirements

Correlation and covariance require two or more continuous variables.

Variable browser buttons	
Add	To generate a correlation, select the continuous variable(s) that you wish to analyze and click Add. Additional variables are added to the summary table which expands to include the new variables.
Split By	When you assign one or more split-by variable to a correlation or covariance analysis, results for each cell in the split-by variable(s) are displayed in a separate tables.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 43.](#)

Correlation Matrix	Matrix of correlation coefficients for all pairs of variables in the analysis.
Covariance Matrix	Matrix of covariances for all pairs of variables in the analysis.
Partial Correlation	Matrix of partial correlation coefficients for all pairs of variables in the analysis.
Correlation Analysis	Generated if confidence interval and/or Fisher’s <i>r</i> to <i>z</i> is selected in the dialog box. This table shows the correlation coefficients and the associated confidence intervals and/or <i>p</i> values for all pairs of variables.
Bartlett’s Test of Sphericity	Table containing the degrees of freedom, determinant of the correlation matrix, the chi square statistic, and <i>p</i> value.

Templates

The following templates provide correlation and covariance results.

Correlations	Bartlett’s Test of Sphericity	Bartlett’s test of sphericity table.
	Correlation Matrix	Correlation matrix table.
	Correlation Z-Test	Fisher’s <i>R</i> to <i>Z</i> with 95% confidence interval.

	Covariance Matrix	Covariance matrix table.
	Partial Correlation Matrix	Partial correlation matrix table.

Exercise

In this exercise you perform a correlation analysis on data in which different western cities are rated by nine criteria. For all but two of the variables, the higher the score, the better. For Housing and Crime, the lower the score the better. You will discover whether there is a linear correlation between any two of the criteria by creating a correlation matrix. Then you will graph correlated and uncorrelated variables in order to see a graphic representation of a high and low correlation.

- Open Western States Rated Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Correlation/Covariance and click Create Analysis
- Click OK to accept the default analysis parameters
- In the variable browser, select all the continuous variables and click Add

Correlation Matrix

	Climate&T...	Housing	Health C...	Crime	Transportation	Education	The Arts	Recreation	Economics
Climate&Terrain	1.000	.659	.445	.042	.086	.151	.442	.260	-.122
Housing	.659	1.000	.575	.147	.313	.177	.533	.397	.366
Health Care & Environment	.445	.575	1.000	.520	.399	.477	.949	.470	.262
Crime	.042	.147	.520	1.000	.289	.233	.553	.303	.239
Transportation	.086	.313	.399	.289	1.000	.302	.398	.454	.161
Education	.151	.177	.477	.233	.302	1.000	.455	.169	-.069
The Arts	.442	.533	.949	.553	.398	.455	1.000	.525	.189
Recreation	.260	.397	.470	.303	.454	.169	.525	1.000	.222
Economics	-.122	.366	.262	.239	.161	-.069	.189	.222	1.000

52 observations were used in this computation.

Each cell at the intersection of a row and column contains a correlation coefficient for the two variables represented by the row and column. Scroll the window from side to side to see the complete matrix. (We have made several columns narrower to fit the page.) Scan the matrix to see where a correlation coefficient may be high enough to indicate a linear relationship between variables. Remember, 0 means no correlation and 1 means a perfect one to one relationship. A negative value means an inverse relationship.

Health Care & Environment and The Arts have a correlation of 0.949, a very high score. Most other correlations are fairly low, between 0.3 and 0.5. Climate&Terrain and Crime have a very low correlation, 0.042. To get a better idea of what these correlations mean, look at scattergrams of the variables with high and low correlations.

- Click an empty area in the view to deselect all results
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Click OK to accept the default analysis parameters

Notice that the buttons in the variables browser have changed. The Remove and Split By buttons are the same, but the Add button has become two buttons: X Variable and Y Variable. You must assign at least one X and one Y variable to complete the analysis.



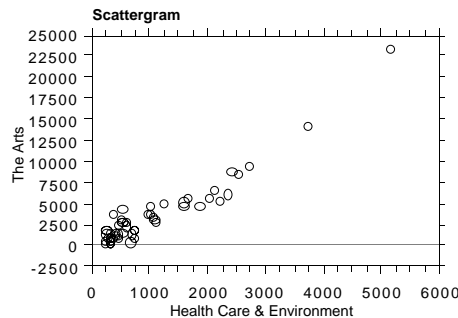
Health Care & Environment and The Arts are the two variables with the highest correlation coefficient in the matrix. Begin by creating a scattergram with these two variables.

- In the variable browser, select Health Care & Environment and click X Variable

The variable has an X usage marker indicating you have assigned it to the X axis.

- In the variable browser, select The Arts and click Y Variable

The variable has a Y usage marker indicating you have assigned it to the Y axis.



The plotted values of these variables occur along a fairly straight line, indicating that a high correlation exists between them. If there were a perfect linear relationship between Health Care & Environment and The Arts, a coefficient of one, the values would form a perfectly straight line.

If you look at a scattergram of two variables with a very low correlation, such as Climate&Terrain and Crime, you will notice that this scattergram differs from the preceding one showing a high correlation. In this one, points are scattered all over the graph rather than clustered along a fairly straight line. This graph provides visual evidence of a very low correlation between Climate&Terrain and Crime as determined in the correlation matrix. The correlation coefficient for these two is only 0.042.

Regression

Regression analysis explains or predicts the value of a dependent variable from one or more independent variables. All variables must be continuous. StatView can estimate these regression models:

1. Simple (one independent variable)
2. Polynomial (linear, quadratic, cubic, etc. terms for a single independent variable)
3. Multiple (two or more independent variables)
4. Forward and backward stepwise (for selecting from a set of possible independent variables)
5. Nonlinear (exponential, logarithmic, power, and growth models for one independent variable)

Discussion

Regression analysis is a tool for discerning relationships among variables. Given one or more variables, regression can predict a related variable and illuminate the nature of the relationship among variables. For example, you can predict a stock index based on unemployment rates or other economic indicators. You can estimate the yield of a chemical reaction using temperature, pressure and quantities of input materials.

Regression modeling is useful when all of the following conditions apply:

1. There is a linear relationship between the variable of interest (the **dependent variable**) and the variables used as predictors (the **independent variables**). As the value of any independent variable increases, the value of the dependent variable must increase or decrease consistently. In the case of nonlinear regression, the corresponding nonlinear relationship is present between the dependent and independent variables.
2. All observations (values for the dependent and independent variables) are independent of each other. If this is not the case (observations measured on the same object over time, for example), regression analysis can be used to examine relationships within your data, but the probability values for hypothesis tests will not be valid.
3. The portion of the dependent variable not explained by the independent variables is due to random error. For linear and logarithmic regression, the error is assumed to be additive and normally distributed with a constant variance. For exponential, power, and growth regression, the error is assumed to be multiplicative, and the natural logarithm of the error is

assumed to be normally distributed with a constant variance. There are diagnostics to help identify cases that do not follow this distribution and transformations that can help correct the problem. These are discussed in the later section, [“Residuals,” p. 57](#).

4. Error in the independent variables is nonexistent, or at least negligible relative to error in the dependent variable.

Simple and multiple regression

Simple regression is appropriate when you wish to model a dependent variable with exactly one independent variable. You can verify the linearity of the relationship between variables by looking at a scattergram of the two variables. For more than one independent variable, the appropriate technique to use is **multiple regression**. This takes into account the linear effect of several independent variables in predicting the dependent variable. As the name implies, multiple regression is more complex than simple regression, since relationships among the independent variables can make it difficult to interpret the results (see [“Colinearity,” p. 53](#)). If you have many independent variables, you might want to consider a model selection procedure (described later under stepwise regression).

Polynomial regression

When the relationship between a dependent variable and an independent variable is not linear, **polynomial regression** can be a useful tool. As stated earlier, a linear relation implies that the dependent variable's values must consistently increase or decrease as the value of the independent variable increases. By including terms for the square, cube, fourth power, etc. of the original variables, this strict linear relation is no longer required. For example, if you include the square of a variable as an independent variable, then the dependent variable can rise and fall (or fall and rise) once as the original variable's value increases. Similarly, the cube of a variable will allow for two changes in direction of the dependent variable as the independent variable increases. In addition, a polynomial regression can be useful when the relationship between a dependent variable and an independent variable follows a curve, for example if the dependent variable's rate of increase is less as the value of the independent variable increases.

Remember, however, that polynomial regression is just a mathematical tool for fitting a curve, and while it can be useful for prediction, care should be taken before assuming that the underlying relationship between the two variables being studied is actually polynomial.

Stepwise regression

In regression analysis, a model selection procedure helps choose the independent variables that are most useful in explaining or predicting your dependent variable. StatView offers forward and backward stepwise selection.

Forward selection starts with an empty model and adds independent variables in order of their ability to predict the dependent variable. **Backward selection** starts with all the independent variables in the model and at each step removes the one that is least useful in predicting

the dependent variable. The criteria for adding and deleting variables is the **partial F -ratio**, which is the square of the t -test value for the null hypothesis that the coefficient of the variable in question is equal to zero.

The forward procedure starts with no variables in the model except the intercept (if present). The backwards procedure starts with all the variables in the model. Both procedures then use the same algorithm to enter or remove variables. First, the partial F -ratio for each variable in the model is examined. If the least of these is less than the **F -to-remove** you specify, the corresponding variable is removed. Otherwise, the partial F -ratio for each variable not in the model is examined. If the greatest of these is greater than the **F -to-enter** you specify, the corresponding variable is entered. This completes one step. Stepping stops when no variable is entered or removed.

The default criteria are appropriate for most models, but you can adjust them to suit your needs. For example, if you wish to build a model containing only variables that seem very useful for prediction (i.e., a model with few variables), then raise the criteria for entering variables by increasing the value of the F -to-enter and F -to-remove.

Force

Variables can be **forced** into the model using the Force button on the variable browser. In the forward procedure, all forced variables are entered at step 0; in the backward procedure, *all* variables are entered at step 0. In either procedure, forced variables are never removed from the model regardless their partial F -ratios.

Stepwise regression summary

StatView displays regression summary tables to help you assess the quality of the regression model at each step. Also, a stepwise regression summary table, displayed only for stepwise models, reports the number of steps, number of variables entered, F -to-enter and F -to-remove.

Colinearity

Forward and backward stepwise selection techniques do not always choose the same model due to the close relationship between independent variables in regression studies. When a variable is considered for entry or removal, its importance can be highly influenced by the presence of other variables in the model. You can identify sets of related variables by using both forward and backward selection and comparing the chosen models. If variables appear in one model but not the other, they can be too closely related to provide useful information; one of them should be removed. This phenomenon is known as **colinearity**.

When you perform a regression with many variables, some of the independent variables will inevitably be related. If the relationships are not too strong (if the maximum correlations between any two independent variables is less than 0.8), this is not likely to cause problems. However, if there are strong relationships among some of the independent variables, your results can be difficult to interpret or even useless. In a stepwise regression, one indication of

colinearity is the sign of the estimated coefficient for a particular variable changing depending on which other independent variables are included in the model.

Nonlinear models

StatView can estimate the four most commonly used nonlinear transformations of the simple linear regression model:

1. Exponential
2. Logarithmic
3. Power
4. Growth

StatView computes estimates for the coefficients of these models by first linearizing the transformations and proceeding with the usual linear regression calculations, and then back-transforming the estimates into the terms of the nonlinear equation. (For example, to compute the exponential model discussed below, StatView first logs the values of the dependent variable you specify, represented here by Y , then performs its usual calculations. The resulting intercept is then exponentiated to correspond to the original nonlinear form of the equation.) Note that this method differs from the generalized nonlinear fitting performed by other statistical programs, which fit arbitrary models by iteratively minimizing a loss function or by iteratively maximizing likelihood.

Exponential

Exponential transformations are useful for fitting data that increase or decrease at high rates. One common use is to model allometric data—measures of the change in proportion of various anatomical parts of an organism throughout the organism's growth cycle. The basic form of the exponential transformation is this:

$$Y = b_0 e^{b_1 X}$$

StatView estimates the linearized form of the model:

$$\ln Y = \ln b_0 + b_1 X$$

StatView's linearization constrains Y to positive values, since logarithms of negative or zero values are undefined. Negative or zero data cause error messages.

Logarithmic

Logarithmic transformations are useful for modeling slow-growth data. For example, metal powder subjected to high temperatures will tend to form crystals whose size are a logarithm of the time of treatment. The basic form of the logarithmic transformation is this:

$$Y = b_0 + b_1 \ln X$$

By definition the logarithmic model cannot be used with negative or zero values in the independent variable. Negative or zero data cause error messages. This model is already linear, so StatView estimates it directly after transforming X values.

Power

Power transformations are often used in industrial situations; for example, tool life can be modeled as a power of cutting speed. Power transformations are also useful with allometric data, e.g., relating the mass of a fish to its length throughout its growth cycle. The basic form of the power transformation is this:

$$Y = b_0 X^{b_1}$$

StatView estimates the linearized form of the model:

$$\ln Y = \ln b_0 + b_1 \ln X$$

StatView's linearization constrains Y and X to positive values, since logarithms of negative or zero values are undefined. Negative or zero data cause error messages.

Growth

Growth transformations are often used to model population growth over time. The basic form of the growth transformation is this:

$$Y = e^{b_0 + b_1 X}$$

StatView estimates the linearized form of the model:

$$\ln Y = b_0 + b_1 X$$

StatView's linearization constrains Y to positive values, since logarithms of negative or zero values are undefined. Negative or zero data cause error messages.

Model coefficients and intercept

An **intercept** is the expected value of the dependent variable if all the independent variables had values of zero. In many cases its purpose is to correct for differences in units of measurement between the dependent and independent variables.

StatView automatically includes an intercept as part of a regression model unless you specify otherwise (for nonlinear regression, the b_0 or $\ln b_0$ term cannot be removed). The Regression dialog box contains a checkbox labeled "No intercept in model," which removes the intercept and forces the model through the origin. It might be appropriate to remove the intercept from the model, but do so with caution. Sometimes there is a physical reason to remove the intercept: it is known ahead of time that if the independent variable(s) are 0, the dependent variable must be 0 (the weight of a tree must be 0 if its height is 0). Some of the statistics produced by StatView have a different interpretation when the intercept is removed from the model. You can test for significance of the intercept; the coefficients table provides a

p value for the intercept along with the coefficients for the variable(s). The standard error of the intercept is also provided in the coefficients table.

A linear regression model is an equation $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + \text{error}$, where y is the dependent variable, x_1, x_2, \dots are the independent variables, and b_0 is the intercept. The model intercept and coefficients b_1, b_2, \dots for each variable are listed with their standard errors in the model coefficient table. Note that in a simple regression, the intercept and coefficient of the independent variable in the model coefficient table are the intercept and slope of the regression line.

Standardized regression coefficients

Since the magnitudes of independent variables might vary widely, it is difficult to compare the relative importance of a regression coefficient for one variable with that of another variable. For this reason, **standardized regression coefficients** are often useful in determining which independent variables in a regression are most important in helping to predict the dependent variable. Standardized coefficients are calculated as if all of the independent variables had variance 1; thus two standardized coefficients can be directly compared, regardless of differences in the scale of the variables involved.

Criteria for model quality

R squared

The simplest statistic to assess the quality of a regression model is the R^2 value, also called the **coefficient of determination**. It is the proportion of the dependent variable's variability that is explained by the independent variables (with a maximum value of 1). Thus, an R^2 of 0.80 means that 80% of the dependent variable's variation is explained by the independent variable(s). An R^2 close to one can be achieved by including many independent variables in the model. If the number of independent variables in a model is close to the number of observations, interpret the R^2 with extreme caution.

One problem with the use of R^2 is that the number of variables is not explicitly included in the formula used to calculate it. Thus, when you assign an additional independent variable to an existing regression, the value of R^2 is guaranteed to increase. A modification of R^2 known as the **adjusted R^2** attempts to remedy this situation by applying a "penalty" to the R^2 value based on the number of variables assigned. The adjusted R^2 is especially useful for comparing a variety of models with different numbers of independent variables.

Upper-case R^2 vs. lower-case r^2 In the case of simple linear regression (one independent variable), R^2 is the coefficient of simple determination and is equal to r^2 , the square of the correlation coefficient. Both represent the proportion of variability in the dependent variable that can be explained by a straight-line relationship with the independent variable. However, for multiple linear models (more than one independent variable), R^2 is the coefficient of *multiple* determination (representing the proportion of variability in the dependent variable that can be explained by a straight-line relationship with a *set of* independent variables) and is not the same as the squared correlation coefficient, r^2 . In any case, R^2 is the correct notation and

is preferred by most statistical packages, although for simple linear regression the r^2 notation would not be incorrect.

***t*-test**

You can assess the adequacy of each independent variable in the model with a *t*-test. This tests the hypothesis that there is no linear relationship between the dependent variable and the independent variable. This differs from the hypothesis of no correlation between the two variables (read about *z*-tests in the chapter [“Paired Comparisons,” p. 29](#)). The *t* value displayed through the regression takes into account the other variables in the regression model, whereas correlation is performed for only two variables at a time. The *t* values and associated *p* values for the intercept and each model coefficient can be found in the model coefficients table.

ANOVA statistics

Another measure of model quality is the **regression ANOVA table**. This table uses the sum of squares and mean squares to calculate an *F* statistic, as a standard ANOVA (see [“ANOVA,” p. 73](#)) does. The probability of the *F*-statistic for a regression is a guide to how important the independent variable(s) are in explaining the behavior of the dependent variable; a low *p* value associated with an *F*-statistic means it is unlikely that an *F*-statistic as large as the one calculated would have happened by chance. Thus we assume that the variable(s) in question are useful for explaining variation in the dependent variable.

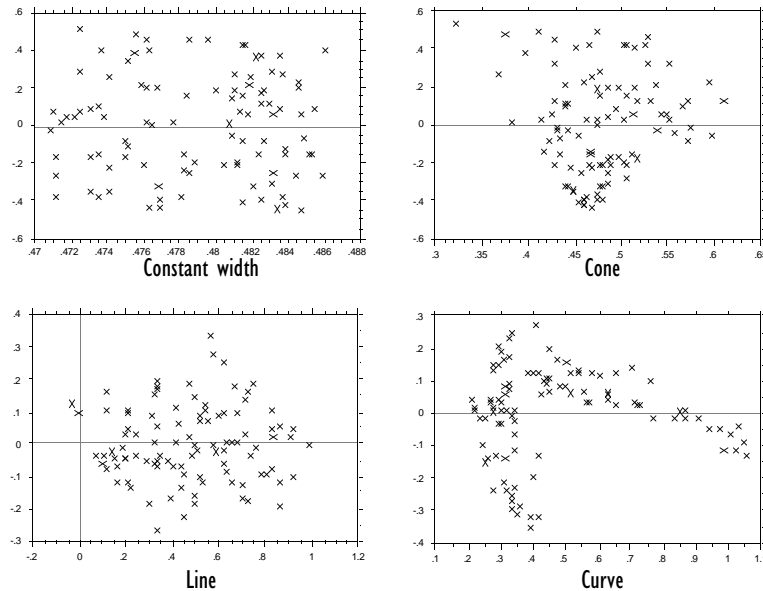
Residuals

Because a regression model rarely estimates the value of the dependent variable exactly, there is a difference between the predicted or fitted value of the dependent variable and its actual value. This difference is known as the **residual**.

Residual plots

Residuals are useful in helping you identify **outliers**, observations that behave very differently than the bulk of the observations. The residuals from a regression represent the portion of the data that is not explained by the model. In the residual plots described below, any point that is distant from most of the points on the plot is considered an outlier and its origin investigated. If it is clear that the observation is an error (for example, a mistake in data transcription or entry), then you correct it or delete it from the analysis. The fact that an observation does not fit in with the other observations in the analysis does not justify its removal. Before removing outliers, always investigate the source of the outlier to provide justification based on the context of the data collection process. If an unusual residual is the only reason for deleting an observation, it is best to leave it in the model and continue to investigate the cause of the unusual residual. Sometimes these observations contain important information about your data.

One useful residual plot is the plot of residuals versus fitted values. The following are some different shapes for this plot.



If the assumptions of the regression are met, the plot of residuals versus fitted values will show a band of constant width independent of the fitted value. The cone shape is a common deviation from this pattern, as in the upper right plot where the spread of residuals is wider for larger fitted values. This tells you that the variance of the observations increases as the mean increases. That generally indicates a need to transform the dependent variable by a logarithmic or square root transformation before regression is carried out. If the data are counts, for example, a square root transformation is often helpful.

Another useful residual plot uses the residuals plotted against each of the independent variables in the model. Once again, the expected pattern, if the assumptions are met, is a band of residuals of constant width throughout the range of the regressor. If the assumption of a purely linear relationship between the dependent and independent variable is not appropriate, the residual plots will display a systematic deviation from the constant width pattern. For example, if the residuals tend to lie in a band that curves either upward or downward, as in the lower right plot, the addition of a new term representing the square of the regressor might improve the fit. Similarly, the cone shape pattern suggests that a transformation of the regressor in question might be in order. The plot of residuals versus independent variables might be useful when colinearity is suspected among the independent variables.

The assumption of independence of observations might be violated when observations are measured across time. As with the other violations of assumptions, a residual plot can help make this clear, though the observed pattern of the residuals might be more subtle. A plot of residuals versus a variable representing time should, as always, show no discernible pattern. Any regularity, such as noticeable cyclical patterns, indicates that a more complex analysis is necessary to accommodate the time series nature of the data.

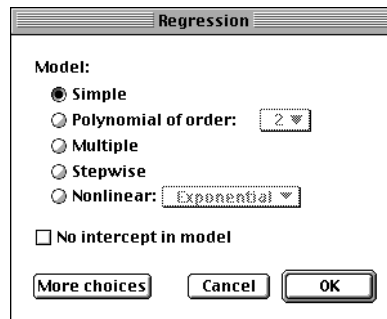
Residual statistics

Along with residual plots, StatView produces statistics which help summarize the behavior of the residuals. These include the number of residuals greater than zero and less than zero. Since the mean value of the residuals is guaranteed to equal zero, these two numbers can give you a feel for the symmetry of the residuals. If they are symmetric, the two numbers should be approximately equal. If not, the residuals might be skewed, and a transformation or a different model might be appropriate.

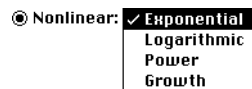
The remaining residual statistics help assess the level of first degree autocorrelation within the residuals, i.e., the level of correlation between each residual and the residual immediately before it in the dataset. Thus, they are only of value if the observations in your data are ordered in a meaningful way. These statistics are labelled $SS[e(i) - e(i-1)]$, Durbin-Watson, and Serial Autocorrelation. An autocorrelation close to -1 or 1 implies a high degree of correlation between the residuals.

Dialog box settings

When you create or edit a regression analysis, you set the analysis parameters in two dialog boxes, a small one with few choices and an expanded one with many choices. In the first of the two, you can select simple, polynomial, multiple or stepwise regression and click OK to accept the default parameters.



Model If you choose polynomial, you must specify an order or degree of the polynomial between 2 and 9. If you choose stepwise, your model will be created using the default stepwise parameters (forward stepwise with an F -to-Enter of 4.000 and an F -to-Remove of 3.996). You can change these parameters in the expanded dialog box by clicking More Choices. If you choose Nonlinear, you must choose which nonlinear transformation to use.



If you select a result and click Edit Analysis, you will not be able to change the model type from multiple or stepwise to simple, polynomial, or nonlinear regression. You must instead create a new analysis with the desired model type.

No intercept This option lets you remove the intercept from the model. Please read the cautions discussed under [“Model coefficients and intercept,” p. 55](#) before doing so. Suppressing the intercept is not allowed for nonlinear models.

More choices To see an expanded dialog box with additional choices for regression analysis, click More choices. You can return to the smaller dialog box by clicking the Fewer choices button.

The image shows a 'Regression' dialog box with the following settings:

- Model:** Simple (dropdown menu)
- Order:** 2 (dropdown menu)
- ☐ **No intercept in model**
- Stepwise parameters:** Forward (selected radio button), Backward (radio button)
- F-to-enter:** 4 (text box)
- F-to-remove:** 3.996 (text box)
- Save to dataset:** Residuals (checkbox), Fitted (checkbox), Predicted (checkbox)
- Compute values for:** included rows (selected radio button), all rows (radio button)
- Confidence level:** 95 % (text box)
- Plot confidence bands for:** Mean (checkbox), Slope (checkbox)
- Buttons at the bottom: Fewer choices, Cancel, OK

Options in the top section of this dialog box are the same as in the Fewer Choices dialog box.

Stepwise parameters The section below that is available only if you choose stepwise regression. You can specify forward or backward, and set the partial F -ratio criteria for entering and removing variables. The F -to-remove defaults to 3.996, and must be lower than the F -to-enter, which defaults to 4.

Residual, fitted, and predicted values There are checkboxes allowing you to generate and save residual, fitted and predicted values. These values are saved to the dataset containing the dependent variable and are dynamically linked to the analysis. They are assigned the name Fitted Y, Residual Y, or Predicted Y, where Y is the name of the dependent variable for the regression. StatView identifies the source of these columns that are generated as part of an analysis as Analysis Generated variables (see [“Save to dataset,” p. 61](#)).

StatView distinguishes fitted and predicted values as follows:

1. Fitted values are values of the dependent variable predicted by the analysis using the data with which the regression model were fit.
2. Predicted values are values of the dependent variable predicted by the regression model using new data. You enter these data into the columns which contain the independent variable(s), leaving missing values in the dependent column. These values can be entered into any row in the independent variable(s). The predicted values will appear in the same row in the Predicted Y column. Note that predicted values will also be generated for any row that contains a missing value for the dependent variable if predicted values is checked in the dialog box. However, predicted, fitted, and residual values have a missing value if any row is missing for the independent variable.

Included rows or All rows You can choose whether to compute residual, fitted and predicted values using all the rows in the dataset, or for only the included rows. If you select Included rows, the values are calculated for just the included rows of the dataset; excluded rows contain

missing values. If you select All rows, the values are calculated for all rows in the dataset regardless of their included or excluded state.

Confidence bands and intervals The bottom choices in the dialog box allow you to plot confidence bands for the mean, slope, or both in the simple regression plot. The confidence level text box specifies the level for the mean and slope for the regression plot and is also used with the confidence interval table. Confidence intervals are only available for simple linear regression.

Save to dataset An **analysis generated variable** is dynamically tied to the regression analysis that created it. If you change the parameters of the model or any of the data in the independent or dependent variables, the analysis generated variable in the dataset will automatically update. In addition, the variable is associated with the view that contains the analysis, not the dataset in which they appear. This means that it will automatically be added to the dataset which contains the dependent variable when the view which contains the regression is reopened and the regression analysis recalculated. If you close the view, the variable will be removed from the dataset. One consequence of this is that if you plan to use an analysis generated variable in a formula, you need to open the view containing the regression analysis for the formula to compute.

Because these variables are dynamic, you can generate a graph or statistic using the residual, fitted, or predicted values, that will also automatically update when the model or underlying data change. You can create a histogram or box plot showing the distribution of your residuals and the plot will stay current with any changes you make to your model. Note that any result created using analysis generated variables must be located in the same view as the regression analysis.

To break the link between an analysis generated variable and the analysis, change its source to User Entered. This causes all ties to the analysis to be broken and the letters “UE” appended to the front of the variable name to indicate that it is now user entered. Any change to the regression that created it will have no effect on the variable, and they act just as any user-entered variable would. If you delete any of these analysis generated columns it is equivalent to turning off the Save options in the Regression dialog box.

Data requirements

Simple, polynomial, and nonlinear regression models require one continuous independent and one continuous dependent variable. Multiple and stepwise regression require one or more continuous independent variables and one continuous dependent variable.

Variable browser buttons		
Simple, polynomial, and nonlinear regression	Independent	Select the continuous variable which is the independent variable for the model and click the Independent button. Each additional independent variable assigned creates a new analysis with the new independent and the previous dependent variable.

	Dependent	Select the continuous variable which is the dependent variable for the model and click the Dependent button. Each additional dependent variable assigned creates a new analysis using the new dependent variable and the old independent variable.
	Force	The Force button is the same as the Independent button for all regression analyses except Stepwise regression (see below).
	Split By	When you assign one or more split-by variables to any regression results, results for each cell in the split-by variable(s) are displayed in separate tables and plots.
Multiple and stepwise regression	Independent	Select the continuous variables which are the independent variables for the model and click the Independent button. Additional independent variables are added to the model.
	Dependent	Select the continuous variable which is the dependent variable for the model and click the Dependent button. Each additional dependent variable assigned creates a new analysis using the new dependent variable and the old independent variable(s).
	Force	The Force button allows you to force continuous variables into a stepwise regression. Each forced variable will automatically be an independent variable of the model even if these variables do not meet the model criteria. For a multiple regression, the Force button is the same as the Independent button, except that variables entered with the Force button appear first in tables.
	Split By	When you assign one or more split-by variables to any regression results, results for each cell in the split-by variable(s) are displayed in separate tables and plots.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 51](#). The Regression Summary, ANOVA table, and Regression Coefficients table are the default output for this analysis. Most of the results below are computed for both linear and nonlinear regression; exceptions are noted within the table.

Regression Summary	Table containing count, number missing, correlation coefficient (R), R^2 , adjusted R^2 , and RMS residual. For exponential, growth, and power models, R , R^2 , adjusted R^2 , and RMS residual are not computed.
ANOVA Table	Table containing the degrees of freedom, sum of squares, mean squares, F value, and p value for the regression ANOVA. The table is not computed for exponential, growth, and power models.
Regression Coefficients	Table containing the coefficients of the regression equation. Standardized coefficients, standard error, t value and p value are also displayed. For exponential and power models, standard error, t value and p values are not computed for the intercept term. An additional row for the log-intercept term is shown for exponential and power models.
Confidence Intervals	Table containing both regular regression coefficients and their upper and lower confidence intervals as set in the dialog box. This table is not available for stepwise regression.

Residual Statistics	Table containing the number of positive or zero residuals, the number of negative residuals, and autocorrelation statistics. For exponential, power, and growth models, the Durbin-Watson statistic is not computed.
Regression Plot	A scattergram of the dependent vs. the independent variable with regression line and equation. For simple regression, confidence intervals can be added for the mean and slope using the dialog box. Not available for multiple regression.
Residual Plots	Graphs of residuals vs. fitted dependent and of dependent vs. fitted dependent are available. For a stepwise regression, these plots will include information for the last step.

For further options on plotting scattergrams with fitted regression lines or smoothed curves, see [“Bivariate Plots,” p. 221](#).

Additional stepwise regression results

The following tables appear only if stepwise regression is selected. The Stepwise summary always appears. The Variables in Model and Variables Not in Model tables appear if regression coefficients are requested.

Stepwise Regression Summary	Table containing <i>F</i> -to-enter, <i>F</i> -to-remove, number of steps, variables entered, variables forced and the stepwise procedure used.
Variables in Model	Table containing the names and coefficients of the variables entered into the model at each step. Standardized coefficients, standard error, and the <i>F</i> -to-Remove are also displayed.
Variables Not in Model	Table containing the partial correlation and the <i>F</i> -to-Enter of the variables not entered into the model at each step.

Templates

The following templates provide regression results.

Graphs	Bivariate Regression Plot	Bivariate scattergram with regression line and equation.
Regression	Exponential Regression	Simple regression summary and coefficients tables and regression plot using the exponential transformation.
	Growth Regression	Simple regression summary and coefficients tables and regression plot using the growth transformation.
	Logarithmic Regression	Simple regression summary, ANOVA, and coefficients tables and regression plot the logarithmic transformation.
	Power Regression	Simple regression summary and coefficients tables and regression plot using the power transformation.
	Regression--Multiple	Multiple regression summary, ANOVA, and coefficients tables.
	Regression--Polynomial	Polynomial regression summary, ANOVA, and coefficients tables; polynomial regression plot.
	Regression--Simple	Simple regression summary, ANOVA, and coefficients tables; regression plot.

	Regression--Stepwise	Stepwise summary table and regression plot; for each step, ANOVA, coefficients, summary, Variables In, and Variables Not In tables.
	Residual Stats--Simple Regr	Simple regression residual statistics table, residuals vs. fitted and residuals vs. dependent plots.

Exercises

Several of these exercises analyze the Tree Data sample dataset. In the 1930s, the weights and trunk girths were measured for eight specimens from each of thirteen root-stocks, for a total of 104 tree specimens.

Simple linear regression

We will perform a simple regression to predict the weight of trees from their girth. This makes it possible to get accurate estimates of weight without having to cut trees down and weigh them, a destructive and difficult process. Your first step is to perform a simple regression to see whether there is a linear relationship between weight and girth. A high R squared (R^2) would indicate a strong linear relationship.

- Open Tree Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Regression, select Regression Summary and Regression Coefficients and click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent results
- Click OK to accept the default parameter settings
- In the variable browser, select Trunk Girth and click Independent
- In the variable browser, select Weight and click Dependent

Regression Summary
Weight vs. Trunk Girth

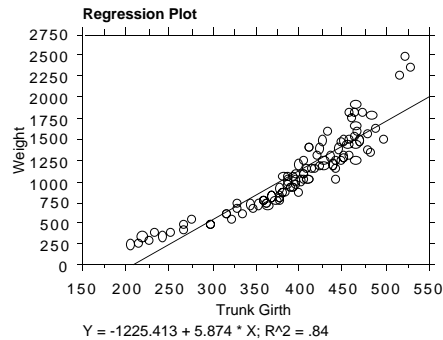
Count	104
Num. Missing	0
R	.916
R Squared	.840
Adjusted R Squared	.838
RMS Residual	183.606

Regression Coefficients
Weight vs. Trunk Girth

	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
Intercept	-1225.413	102.361	-1225.413	-11.971	<.0001
Trunk Girth	5.874	.254	.916	23.101	<.0001

You can see from the high R^2 value in this summary table that there seems to be a clear relationship between Weight and Trunk Girth. Now, to examine the relationship and to confirm the notion that it is linear, create a regression plot. This is a bivariate scattergram of Weight vs. Trunk Girth with a regression line added.

- Make sure at least one table is still selected
- In the analysis browser, select Regression Plot and click Create Analysis



Notice that you did not need to assign variables to this plot. The preceding table was selected when you created this plot, so StatView treats the plot as additional output from the existing regression analysis, rather than a newly requested analysis you are creating from scratch.

Polynomial regression

The plot shows that the weight of trees increases faster than it would if there were a strictly linear relationship. The spread of points is curved with values at the ends above the regression line and those in the middle below it. The relationship between Weight and Trunk Girth might be better explained by adding a quadratic term in Trunk Girth. You can test this hypothesis by changing the current analysis to a polynomial regression.

- Make sure at least one result is still selected
- Click Edit Analysis (a button at the top of the view window)

The Regression dialog box reappears so that you can change parameter settings.

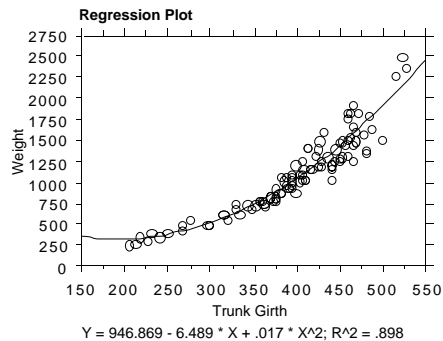
- Choose Polynomial of order and click OK (keep the order setting of 2)

Regression Summary Weight vs. Trunk Girth

Count	104
Num. Missing	0
R	.948
R Squared	.898
Adjusted R Squared	.896
RMS Residual	147.192

Regression Coefficients Weight vs. Trunk Girth

	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
Intercept	946.869	297.493	946.869	3.183	.0019
Trunk Girth	-6.489	1.640	-1.012	-3.956	.0001
Trunk Girth^2	.017	.002	1.944	7.597	<.0001



In these results, $p < 0.0001$ for the squared term shows that the quadratic term is useful for explaining the relationship between the two variables. The graph shows how well the second order polynomial regression fits the data.

Predicted values

Now you can use this model to predict a tree's weight based on its trunk girth. StatView will predict a value for any row in the dataset that has a value for the independent variable and a missing value for the dependent variable.

- Uncheck Recalculate (in the upper left corner of the view window)

This prevents predicted values from being calculated one at a time while you add each independent value to the dataset. We will enable calculation after adding all the new independent values to the dataset.

- Make sure at least one of the regression results is selected
- Click Edit Analysis
- Click More choices
- Select Predicted (after Save to dataset) and click OK
- Select Tree Data from the Window menu to bring it forward

A new Predicted Weight variable at the end of the dataset contains missing values (.). Your predicted values will appear in this column. Next, we will add four new rows to the dataset by adding values at the bottom of Trunk Girth.

- At the bottom of Trunk Girth column (after row 104), enter the values 500, 600, 700, 800
- Select the view from the Window menu to bring it forward
- Check Recalculate
- Select Tree Data from the Window menu to bring it forward

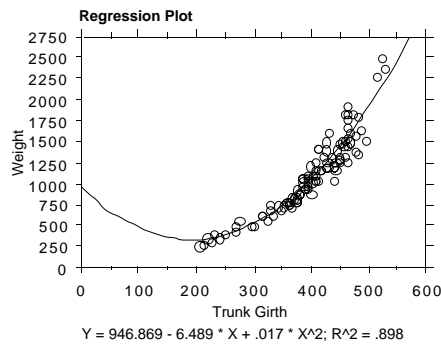
In the Predicted Weight column of the dataset, the following values appear:

500	•	1909.081
600	•	3111.125
700	•	4649.705
800	•	6524.818

The second order polynomial regression model predicts these values for weight based on the trunk girth values you entered.

While the polynomial regression plot appears to be a reasonable fit, one aspect is troubling: it would not be an effective model for predicting weight from *smaller* girth measurements. The parabolic behavior of a quadratic fit doesn't make biological sense, which becomes apparent if we extend the horizontal axis to zero:

- Click in the blank area of the view to deselect all results, then click the horizontal scale of the Regression Plot result to select it
- Click Edit Display
- Change the From bound to 0 and click OK



Adding smaller Trunk Girth values to the dataset would reveal similar results in the Predicted Weight column.

Growth regression

A nonlinear model might be more suitable. Let's try fitting a growth regression model:

- Make sure at least one result is still selected
- Click Edit Analysis (a button at the top of the view window)

The Regression dialog box reappears so that you can change parameter settings.

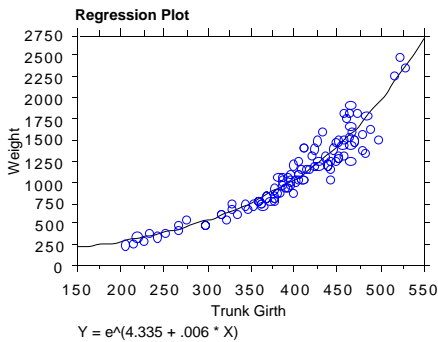
- Click the Fewer Choices button
- Choose Nonlinear, select Growth, and click OK

Regression Summary
Weight vs. Trunk Girth
Y = e^a(b0 + b1*X)

Count	104
Num. Missing	0
R	.
R Squared	.
Adjusted R Squared	.
RMS Residual	.

Regression Coefficients
Weight vs. Trunk Girth
Y = e^a(b0 + b1*X)

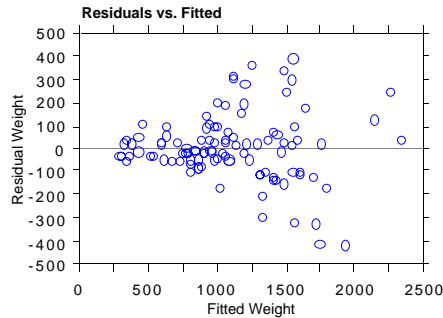
	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
b0	4.335	.063	4.335	68.920	<.0001
b1	.006	1.562E-4	.972	41.545	<.0001



We can tell intuitively from the regression plot that the growth regression fit is fairly good, and unlike the polynomial curve, the growth curve shows reasonable behavior for narrower-trunked trees. The p values indicate that both terms are useful for explaining the relationship between girth and weight.

Let's examine a plot of residuals vs. fitted values to assess this model further:

- Make sure at least one result is still selected
- In the analysis browser under Regression, double-click Residuals vs. Fitted Values



The plot has a slight cone shape, suggesting that a logarithmic transformation of the dependent variable might help (see [“Residual plots,” p. 57](#)). So, let's try an exponential regression.

Exponential regression

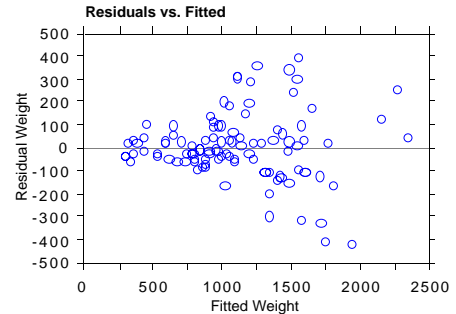
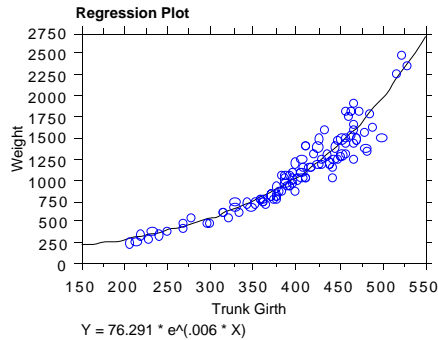
- Make sure at least one result is still selected
- Click Edit Analysis (a button at the top of the view window)
- From Nonlinear, select Exponential, and click OK

Regression Summary
Weight vs. Trunk Girth
 $Y = b_0 * e^{b_1 * X}$

Count	104
Num. Missing	0
R	.
R Squared	.
Adjusted R Squared	.
RMS Residual	.

Regression Coefficients
Weight vs. Trunk Girth
 $Y = b_0 * e^{b_1 * X}$

	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
b0 (from ln(b0))	76.291
ln(b0)	4.335	.063	4.335	68.920	<.0001
b1	.006	1.562E-4	.972	41.545	<.0001



These results appear to be nearly identical, right down to the residuals. How could that be? Look at the equations under the regression plots. A little algebra reveals that for these data, the two fits are nearly equal:

$$\begin{aligned}
 Y &= e^{(4.335 + 0.006x)} & Y &= 76.291 e^{0.006x} \\
 &= e^{4.335} \times e^{0.006x} \\
 &= 76.325 e^{0.006x}
 \end{aligned}$$

Since we still see some cone-like spreading to the right in the residuals plot, we need to exercise caution predicting values too far beyond the range of the data.

Multiple regression

We turn now to a multiple regression model. The Car Data sample dataset has information on 116 cars compiled by *Consumer Reports*. This information includes data about weight, gas tank size, turning circle, horsepower and engine displacement for cars from different countries. We want to find out whether there is a relationship between gas tank size and other variables.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Regression, select Regression Summary and Regression Coefficients, and click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent results
- Choose Multiple and click OK
- In the variable browser, select Gas Tank Size, and click Dependent

- Select Weight, Turning Circle, Displacement, and Horsepower and click Independent

Regression Summary
Gas Tank Size vs. 4 Independents

Count	116
Num. Missing	0
R	.852
R Squared	.727
Adjusted R Squared	.717
RMS Residual	1.637

Regression Coefficients
Gas Tank Size vs. 4 Independents

	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
Intercept	2.551	2.553	2.551	.999	.3200
Weight	.004	.001	.781	7.820	<.0001
Turning Circle	-.021	.082	-.021	-.256	.7985
Displacement	-.001	.006	-.019	-.176	.8610
Horsepower	.011	.006	.139	1.689	.0940

The p values in the Regression Coefficients table tell you that Weight is the only variable useful in predicting gas tank size. In addition, an adjusted R^2 value of 0.717 indicates a fairly strong overall relationship. To confirm the relationship between Gas Tank Size and Weight graphically, you might want to plot these two variables using a bivariate plot.

Stepwise regression

In this exercise you perform a stepwise regression using census data for 506 housing tracts in the Boston area from Belsley, Kuh, and Welch (1980). You will determine what factors are most useful in predicting the median value (in thousands of dollars) of homes. Variables include crime rate, percentage of land zoned for large lots, percentage of non-retail business acres, nearness to the Charles river, nitrogen oxygen concentration (ppb), average number of rooms, percentage of units built before 1940, weighted distance to five employment centers, accessibility to radial highways, property tax rate (\$ per \$10,000), district pupil/teacher ratio, and percentage of lower status population.

- Open Boston Housing Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Regression, select Regression Coefficients and click Create Analysis
- Choose Stepwise

The stepwise setting produces a forward stepwise regression with an F -to-Enter of 4.000 and an F -to-Remove of 3.996. If you would like to change these parameters, click the More Choices button.

- Click OK
- In the variable browser, select all the continuous variables except Median Value and click Independent
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent variables
- In the variable browser, select Median Value and click Dependent

The analysis calculates and results appear in the view. The Stepwise Regression Summary table indicates that nine variables were entered into the model in nine steps.

Stepwise Regression Summary Median Value vs. 11 Independents	
F-to-Enter	4.000
F-to-Remove	3.996
Number of Steps	9
Variables Entered	9
Variables Forced	0
Stepwise Procedure	Forward

To see which variables were entered and which were not, scroll to the bottom of the view and examine the information for step 9. All variables were entered in the model except Industry and Before 1940.

Variables In Model
Median Value vs. 11 Independents
Step: 9

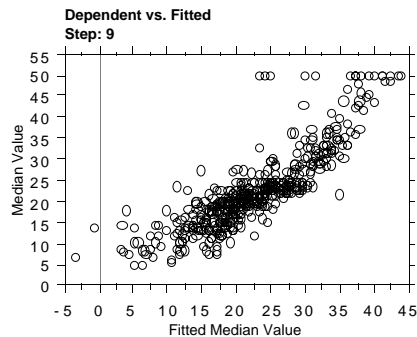
	Coefficient	Std. Error	Std. Coeff.	F-to-Remove
Intercept	42.003	4.950	42.003	72.002
Crime	-.128	.033	-.120	14.943
Zone	.046	.014	.117	11.142
NOX	-.173	.036	-.219	23.339
Rooms	3.712	.413	.284	80.751
Dist. Empl.	-1.552	.189	-.355	67.295
Highways	.300	.064	.284	21.698
Tax Rate	-.013	.003	-.243	14.975
Pupil/Teach...	-.964	.131	-.227	53.854
Low status	-.554	.048	-.430	133.711

Variables Not In Model
Median Value vs. 11 Independents
Step: 9

	Partial ...	F-to-Enter
Industry	.025	.300
Before 1940	.020	.200

This result suggests that all nine variables entered in the model are somehow significant in explaining the dependent variable, Median Value. It gives no details about the individual variables themselves. You can examine these data further with the Dependent vs. Fitted plot.

- Make sure at least one result is still selected
- In the analysis browser under Regression, select Dependent vs. Fitted and click Create Analysis



The houses with the highest median values cluster at the top of the graph in a straight line of points suggesting that their predicted values have no relation to the actual values. This suggests that we should reanalyze the data using two separate models: one for high value houses and one for all other values. Using the Recode command, you could create a nominal variable

from Median Value that divides the dataset into two such groups. You would then assign this variable as a split-by variable to perform such an analysis.

ANOVA

An **analysis of variance (ANOVA)** studies the effect of nominal independent variables on a continuous dependent variable. (A **nominal** variable can take on only a limited number of values, whereas a **continuous** variable can take on any value over a wide range. See [“Data class,” p. 50 of Using StatView](#) for a discussion of nominal and continuous data classes.)

A **repeated measures analysis of variance (repeated measures ANOVA)** studies the effect of nominal independent variables (“between factors” or “between-subject effects”) on a continuous response variable within successive measurements (“within factors” or “within-subject effects”). StatView expects within factors (the repeated measures) to be stored as compact variables in the dataset; see [“Compact variables,” p. 84 of Using StatView](#).

An **analysis of covariance (ANCOVA)** studies the effect of both nominal and continuous independent variables on a continuous dependent variable.

A **multivariate analysis of variance (MANOVA)** studies the simultaneous effect of nominal independent variables on several continuous dependent variables.

A **multivariate analysis of covariance (MANCOVA)** studies the simultaneous effect of nominal and continuous independent variables on several continuous dependent variables.

StatView does not compute repeated measures ANCOVA, MANOVA, or MANCOVA designs.

Discussion

Analysis of variance determines the significance of the **effects** in a model by calculating how much of the variability in the dependent variable can be explained by the effect in question. It does this by calculating a quantity called the **mean square**, which is mathematically similar to the variance. This quantity is calculated by dividing the sum of squares of deviations from the means by the **degrees of freedom** for the effect (the number of parameters that the model is estimating to test for the significance of the effect). For main effects, the number of degrees of freedom is one less than the number of discrete values for the factor in question. The degrees of freedom for an interaction is the product of the degrees of freedom of each of the factors contained in the interaction. Finally, this mean square is divided by an estimate of error variance known as the **residual mean square**. This ratio (mean square of the effect divided by residual mean square) results in an **F-statistic** that can be used to test the importance of the effect in question.

The probability (***p* value**) of the *F*-statistic for an effect is a guide to how important that effect is in explaining the behavior of the dependent variable; a low *p* value associated with an *F*-statistic for an effect means it is unlikely that an *F*-statistic as large as the one calculated would have happened by chance. Thus we assume that the effect in question is important in helping to explain the dependent variable. The **power** value gives the probability of correctly rejecting a false null hypothesis, and **lambda** is a quantity called the “noncentrality parameter” used in the calculation of power.

Post hoc tests evaluate pairwise differences among levels of main effects, with protection against simultaneous test error.

In the following sections, we explore each of these concepts in more detail. First, we review the basics of hypothesis testing; then we discuss the components of [M]AN[C]OVA models and each type of model. Finally, we discuss post hoc tests.

Hypothesis testing

Hypothesis testing is the formal statistical technique of collecting data to answer questions about something through the use of a statistical model. Each question asked about a study should be stated in the form of a null hypothesis. A **null hypothesis** states that there are no differences between the values of the dependent variable that can be explained by the differences in the independent variables of your model. For example, if you were comparing several quality control procedures for manufacturing computer chips, an appropriate null hypothesis would be that there are no quality differences between chips manufactured under the various quality control procedures.

The hypothesis tests in the analysis of variance are known as **omnibus tests**, because they test a null hypothesis against the collection of all alternative hypotheses. Taking the quality control example, assume that there are four different procedures being compared. The null hypothesis is that there are no differences among the four techniques as measured by the mean purity of the chips produced. What circumstances would cause this null hypothesis to be rejected? One possibility is that three of the four techniques are equivalent, but the fourth is better than the others; another is that three the four are equivalent, but the fourth is worse. Still another possibility is that two techniques are equivalent but result in lower purity than the other two.

A single null hypothesis is always the basis for a statistical test, and the results of a test simply lead you either to reject the null hypothesis (if you observe significant differences) or to accept it (if you do not observe significant differences). Failure to detect significant differences that would enable you to reject a null hypothesis means that you must continue to assume that the independent variable(s) has no effect on the dependent variable, unless and until more evidence arises to demonstrate measurable, significant differences.

Each term entered into a linear model generates a hypothesis test, where the *F* statistic is a measure of whether or not the null hypothesis should be rejected. The *F*-test compares the observed *F* value with the value that would be expected theoretically if the null hypothesis were true, and it reports the probability that an *F* statistic as large as that observed could have been observed simply by chance. (Even when the null hypothesis is true, it's possible that the particular data observed could result in a higher *F*-ratio than expected.) A small probability

means that an F statistic is unlikely to occur by chance, so the null hypothesis should be rejected.

A typical **significance level** (or “cutoff level”) used for declaring differences significant is 0.05. This means that if the null hypothesis were true, you would incorrectly reject it only 5% of the time, if you reject the null hypothesis when the probability of the corresponding F statistic is 0.05. The significance level to be used in interpreting hypothesis tests should be stated before carrying out the analysis, which is why StatView asks you to specify an “alpha value” before computing the analysis.

The incorrect rejection of the null hypothesis when it is actually true is known as a **type I** or **alpha error**. You should control for this type of error by setting an appropriate significance level and interpreting the hypothesis tests as described.

You can adjust the significance level or **alpha value** as needed. For example, if you conducted a study to determine whether an expensive treatment should be applied to a population, you would need to avoid accidentally rejecting the null hypothesis of no need for the treatment when in fact there was no need. In such a case, you would set your significance level lower than 0.05, perhaps to 0.01 or even 0.001. However, you would want to follow a different approach if you were screening for further study a large number of potentially useful techniques. In this case, you could tentatively reject a null hypothesis of no difference among the techniques because it would be more vital that a useful technique not be incorrectly rejected. A significance level of 0.10 or perhaps even 0.25 might be appropriate. Remember, although a significance level of 0.05 is often used, it is not the best level for every situation.

Setting the significance level to a specific value controls the type I error of a statistical test, but there is another type of possible error in hypothesis test performance: the **type II** or **beta error**. Beta error occurs when the null hypothesis is not rejected, even though it is not true. For example, suppose you were conducting a survey of customers in two stores about the amount of money they spent on clothing in the last month. The null hypothesis for the survey would be that there is no difference between the two stores in the amount of money spent. Suppose your budget limited you to questioning only five customers in each store. It would not be surprising if, due to the small number of subjects in the study, you were unable to assert that there were any differences. It might even be that the observed averages were very different, but the statistical test was unable to declare the difference statistically significant.

When you set an alpha level, you are choosing a level of probability for making a type I error (where you fail to reject a null hypothesis that is in fact false), so choosing a smaller alpha value and minimizing the probability of type I error means increasing the probability of type II error (where you accept a null hypothesis that is in fact false). These relationships can be summarized like this:

	Reject null hypothesis	Accept null hypothesis
Null hypothesis true	Type I error α	Correct decision $1 - \alpha$
Null hypothesis false	Correct decision $1 - \beta$ (power)	Type II error β

Power and Lambda

The ability of a statistical test to declare a true difference “statistically significant” is known as the **power** of a statistical test. Obviously the power will vary depending upon which of the many alternative hypotheses are in fact true, how many subjects are studied, and other details of the experimental design. For these and other reasons, it is much harder to guard against type II error than type I error. To ensure that a hypothesis test is carried out with reasonable power, make sure you base your analyses on a sufficient number of experimental units, and that an appropriate design has been chosen for carrying out and analyzing the experiment or study.

Two statistics that help you assess the power of a test are power and lambda. **Power** describes the probability of concluding that each effect is significant when in fact it is significant. Beta (not shown by StatView) is simply one minus power, or power is one minus beta. **Lambda** is a quantity that is used to compute power. It is sometimes called “partial eta squared” (partial η^2) or “noncentrality value,” because the power value comes from a computation of the non-central F distribution, based in part on lambda and indirectly on alpha. The method for calculating power and lambda appears under [“Power and lambda,” p. 441](#).

Model building

This section is a conceptual overview of model building to help you make the right decisions about dependent and independent variables, main effects, and interactions in your models. Subsequent sections examine each type of model (ANOVA, ANCOVA, MAN[C]OVA, and repeated measures ANOVA) in more detail.

Dependent variables

The first decision to be made in building a model is the choice of **dependent variable**. The dependent variable is the variable whose value you are trying to estimate or predict. For example, if you were looking at the effect of different fertilizers on the yield of corn, the yield would be your dependent variable. To predict college GPA score from aptitude tests, the college GPA scores would be the dependent variable. For a comparison of different advertising strategies to determine which one resulted in the most sales, a measure of sales would be the dependent variable.

In some cases there is more than one dependent variable. In a study of the effects of different diets on mice, for example, the growth of the mice might be measured in various ways, such as length, girth, body weight, head circumference, and so on. You could study each of these dependent variables individually in several individual ANOVAs, but if the dependent variables

are correlated, you should study the effect of diet on all the measurements at once in a MANOVA.

In any case, the dependent variable must always be a *continuous* variable. For nominal dependent variables you should consider other methods, such as logistic regression; see the chapter [“Logistic regression,”](#) p. 199.

Independent variables

The rest of the variables in any model are known as **independent variables**. These are the variables to be used in predicting or estimating the dependent variable (or variables). Looking at it another way, the independent variables are the variables which you suspect will explain differences seen in the dependent variable.

A model with one dependent variable and one independent variable as a simple regressor is known as a **simple regression model**. If there is more than one independent variable, but all the independent variables are entered as simple regressors, the model is known as a **multiple regression model**. Fitting a variable as a simple regressor is a good idea if it is appropriate for your data for the importance of that variable. In statistical terms, only one degree of freedom is given up from the estimate of residual variability by including a simple regressor in the model. This, in turn, makes the tests that StatView performs more sensitive to any true relationships that might exist between the independent variables and the dependent variable. However, there is a price to be paid. When you add a simple regressor to a model, you are assuming that its behavior is the same over the entire range of the independent variable. For instance, in a salary study, adding years of education as a simple regressor involves the assumption that no matter how much education a person has, more education still suggests higher salary potential than less education, and that salary increases at a fixed rate.

If the assumption of a consistent linear relationship throughout the range of the independent variable is not supported, it is more appropriate to analyze a nominal version of the independent variable as a **factor**. Models in which all of the independent variables are treated as factors are known as **analysis of variance** models.

For example, suppose you are trying to determine the effect of advertising on sales. If you used the number of advertisements as a simple regressor, you would be assuming that the effect of more advertisements doesn't diminish as they increase in number. If you suspect that sales will level off after a certain number of advertisements, then you would want to treat number of advertisements as a factor, not as a simple regressor. To accommodate the added flexibility in describing the relationship between number of advertisements and sales, ANOVA estimates several parameters, one fewer than the number of values of advertisements studied.

Another example where adding a variable to a model as a factor would be appropriate is in a study of the effect of different exercise plans on blood cholesterol level. Suppose subjects were randomly divided into groups: one group that ran each day, another group that ran three times a week, and a third group that attended daily aerobics classes. The three different types of exercise would represent three **levels** of a factor to be added to a model, with the dependent variable being blood cholesterol level at the end of the experiment. It would be meaningless to fit type of exercise as a simple regressor, because there is no scale on which the three types of

exercise can be assigned that would meet the assumption of a linear relationship throughout the range of the variable.

Models with both regressors and factors are known as **analysis of covariance** models. These models typically arise in one of two ways. First, a continuous variable measured before an experiment or study begins might be expected to affect the value of the dependent variable. For example, in a study to test the effects of different exercise programs on weight loss, one might guess that the initial weights of the participants would affect the amount of weight they lose over the course of the study. The model would include starting weight as a regressor, or **covariate**. Notice that it is not really of interest whether the initial weight is helpful in predicting the weight loss; it is included the model to remove its effect so that the influence of the different programs can be more accurately measured. Second, an experiment might happen to have both continuous and nominal independent variables, i.e., both factors and regressors. For example, to study the sales of soft drinks at several different stores, you might test whether the average temperature in the area influenced the overall sale of soft drinks. You would be interested in *both* the effect of the different stores as levels of a factor *and* the effect of the temperature as a covariate.

Main effects and interactions

After deciding which independent variables should be used to help explain or predict the values of a dependent variable, and whether they should be entered into the model as regressors or factors, a third consideration for your model is that of main effects and interactions.

As an example, consider a study of training programs to teach people how to use a new program on a computer. The dependent variable to be measured is the time it takes students to complete a particular task on the computer using the program. Some students have had previous computer training and some have not. It is felt that this difference might influence the results of the study, so previous training, with two levels, is entered into the model as a factor. The students are randomly divided into three groups: one group which receives an instruction manual, one which receives classroom training, and a third which views a videotape on the use of the program. Type of training is entered into the model as a factor (with three levels).

Effects in a model that consist of a single variable are known as **main effects**. The word “main” in “main effect” doesn’t mean that a main effect is the main point of interest but rather that it is “not an interaction effect.” In fact, a main effect could be less interesting than an **interaction effect**, where the effect of one factor differs according to the level of another factor (or factors). For example, in the computer training study it might be most interesting to know whether the relative merits of the three programs were the same for both novice and experienced users. A significant result (a low p value, e.g. $p < 0.05$) for an interaction leads you to reject the null hypothesis that the effect of one is the same regardless of the other. You should then examine the means table or interaction plots, which show the means of the dependent variable for each combination of factors, to determine the source of the differences. The exercise “[Factorial,](#)” p. 91, illustrates the use of interaction plots. (You can also create interaction plots for bars, lines, and point with the Cell Plots analysis described in “[Cell Plots,](#)” p. 237.)

ANOVA

Analysis of variance (ANOVA) is useful in the same kinds of situations as regression analysis—when you are trying to relate the effect of one or more independent variables on a dependent variable. However, ANOVA models are used when an independent variable has nominal (non-numerical) values, such as hair color (with possible values black, brown, blonde and red) or the state in which subjects live (such as California, New York, Texas, and so on). In cases like these, it is impossible to calculate a linear relationship between the value of the independent variable (such as black, brown or blonde) and the dependent variable. Although such a variable could be coded numerically (1=black, 2=brown, and so on), it would not be appropriate to include it as a continuous regressor since it has no inherent numerical or even ordinal value.

Since the sum of squares calculated for main effects (the effects composed of only one variable) in an ANOVA model are used to test the null hypothesis that the mean of the dependent variable is the same regardless of the level the main effect, ANOVA models can often detect differences even when the relationship is more complex than the simple linear one required by regression analysis. (The price paid for the extra sensitivity is that the linear model contains more parameters to account for these differences, which reduces the power of the analysis.)

Thus, ANOVA models can be useful even if the independent variable is continuous, if it is known or assumed that the relationship between the independent variable and the dependent variable will not easily be explained by a linear or polynomial relationship, or by some other simple relationship that is easily linearized. For example, increasing the concentration of a fertilizer increases yield of a plant up to a certain point, but the yield remains constant after that point. Such a relationship, called a **plateau**, is not linear. In cases like this, you can create a new nominal variable from the original independent variable by dividing the original independent variable's values into a few non-overlapping categories and using this new variable as one of the factors in your ANOVA model. For information on how to do this, see [“Recode data,” p. 117 of *Using StatView*](#).

One other benefit of ANOVA models is their ability to detect interaction between factors in a model. An interaction between factors means that the effect of one of the factors differs depending on the level(s) of the other factor(s) involved in the interaction. For example, if you were interested in the effect of different types of fertilizer on the yield of different varieties of corn, it might be the case that some types of fertilizer were more effective on some varieties of corn than on other varieties. A main effect test of type of fertilizer, for example, would average out the effects of variety and would not address this question. Similarly, the main effect test for variety would average out the effects of fertilizer. However, the interaction test of variety by fertilizer (labeled “variety*fertilizer” in the ANOVA table) would test the null hypothesis that the effect of fertilizer is the same regardless of the corn variety. An equivalent null hypothesis is that the effect of variety of corn is the same regardless of fertilizer type.

Models whose effects are all factors can detect the widest variety of interactions. Although you can enter interaction terms composed of regressors only, be aware that introducing such terms implies a linear relation between the dependent variable and the arithmetic product of the independent variables involved in the interaction. This might not always be the case for the regressors you study. The increased ability to detect interactions in the ANOVA model as opposed to regression comes at the price of additional parameter estimates and potentially decreased sensitivity.

Regression

When you add an independent variable to a linear model as a regressor, you assume that an increase in the independent variable will cause a proportionate increase or decrease in the dependent variable. You can include a continuous variable as an independent variable in an ANOVA model. Usually the purpose of that functionality is for including covariates for ANCOVA or MANCOVA models, but it also makes it possible for you to compute regression models using the ANOVA procedure and assigning only continuous variable(s) as independent variable(s). One reason you might want to consider doing so is that the ANOVA table includes lambda and power results (discussed under [“Power and Lambda,” p. 76](#)), which are not available from the Regression procedure. However, regression models usually should include a constant (intercept) term, which is not possible from the ANOVA procedure. For more information, see the chapter [“Regression,” p. 51](#).

ANCOVA

A model containing both factors and regressors is known as **analysis of covariance**. The name derives from the fact that in an analysis of variance model, a regressor as well as the factors may affect the dependent variable. Models such as these can be looked upon as analysis of variance models with the addition of a “nuisance” regressor that affects the dependent variable and whose effect should be removed, as much as possible, before the actual analysis of variance takes place. However, there is no reason to think of the covariate as being more or less important than the factors in these kinds of models. StatView lets you test not just the regressors but also their interactions with the factors.

To understand when these tests may be helpful, it is useful to explain some terminology. When a single regressor is fit to a dependent variable, the linear model can be summed up by two parameters: the intercept and the slope. The **intercept** is that part of the predicted value which the regressor doesn’t explain. The **slope** is the multiplier of the regressor’s value that scales it to the values of the dependent variable. If the slope is a large number (either positive or negative), then changes in the regressor result in large changes in the dependent variable. If the slope is small (close to zero), then changes in the regressor do not affect the value of the dependent variable very much. To test the significance of the relationship of the regressor to the dependent variable, you simply include the regressor in the model.

When factors are also present in the model, the situation becomes more complex. It might be that the slope for the regressor is the same for each level of the factor. On the other hand, it might be that the regressor has a different effect on the dependent variable depending on the value of the factors in the model. Suppose you are studying weight gains of three groups of volunteers under a special diet: a sedentary group of office workers, a group of active college students, and a group of marathon runners in training. Suppose that you have also measured the calorie intake of the subjects during the course of the study and want to include that information in the analysis. It would be reasonable to suspect that the calorie intake might affect the three groups differently.

To test the null hypothesis that the slopes are the same for the different levels of a factor, you include the interaction of the factor and covariate. In the soft drink sales example, suppose

some stores were located in shopping malls and others were located outside. It might be reasonable to suspect that temperature would not have the same influence in all stores, which you could test by including the interaction between temperature and store. This type of experimental design is known as **grouped regression**, because StatView actually performs a separate regression for each group. Since it compares the results of several regressions, a test for factor-covariate interaction is sometimes called a test for **homogeneity of slopes** or a test for **common slopes**. A significant result tells you that the slope of the regressor differs depending on the level of the factor (or combination of levels of the factors, if there is more than one).

We can go further and test homogeneity of slopes with respect to an interaction of several factors by adding an interaction term with the factors and the covariate. If the hypothesis of homogeneity of slopes cannot be rejected, then the effect of the covariate can be adequately estimated by a single, common slope, and we can eliminate the interaction involving the covariate from the model.

You can examine the interaction graphically by creating a scattergram of the dependent variable vs. the covariate, with separate fitted regression lines for each level of the factor(s) in question and then comparing the slopes of the lines. This technique is demonstrated in the exercise [“ANCOVA,” p. 99](#).

In simple regression, the intercept is usually not of much interest. However, the test for the significance of a factor in an ANCOVA is often called a test for **common intercepts**. If the hypothesis of common intercepts cannot be rejected (i.e., the probability level corresponding to the factor is greater than the significance level you have chosen), then it might be appropriate to remove the factor from the analysis and examine a simple regression model. However, if the hypothesis of homogeneity of slopes is rejected, it is customary to keep the factor in the model.

MANOVA and MANCOVA

Many experimental situations have more than one dependent variable. For example, in studying the effects of a special diet on volunteers, you might measure their weights, blood cholesterol levels, waistlines, oxygen consumption on a treadmill, and so on. Or, a study of pollution levels in different settings might take measurements of oxides of nitrogen, carbon monoxide and particulate matter.

In either example, the measurements are *qualitatively different* from each other—that is, the things measured are inherently different from each other. When this is *not* the case—where the same thing is measured several times on the same subject—repeated measures ANOVA methods are more appropriate. Examples of this would be weight gains after one, two, and three months on a diet, or nitrogen oxide levels measured repeatedly on the same factories using various filtering systems. The subjects (the people or the factories) are the same, and the variables being measured (weights or pollution levels) are the same. What changes is time or circumstance. For more information, see [“Repeated measures ANOVA,” p. 82](#).

In situations where a variety of different measurements are recorded, one obvious alternative is to analyze each dependent variable separately. There are two drawbacks to this approach. First, it might be difficult, if not impossible, to make sense of the reams of output that would be generated. It could be that some variables are influenced by one effect in the model, while oth-

ers are influenced by different effects, and some are not significantly influenced by any of the effects. In such cases, it would be hard to come up with a simple, easy-to-explain summary of the results. It is also possible that a subtle pattern of changes might not be apparent, even after careful study of the output. However, there is an even more serious problem. When several measurements are taken on the same experimental units, they tend to be correlated: that is, the values of some of the variables can be readily predicted by the values of the other variables. The correlations of multiple variables often do not contain much information about the underlying process because each variable is a different way of looking at the same thing. This is not always the case, but it is wise to take the possibility into account.

The technique that performs analysis of variance on more than one dependent variable and explicitly takes into account the correlation among the dependent variables is known as **multivariate analysis of variance**, or MANOVA. StatView's MANOVA results are multivariate counterparts of the tests you would see if you were analyzing only one dependent variable. For example, if you requested multivariate tests for a one-factor factorial design, with teaching method as the grouping variable and dependent variables math score, history score, and reading comprehension score, you would see tables with multivariate tests for the null hypothesis that teaching method had no simultaneous effect on all three scores. The null hypothesis for a univariate model with only math score would simply test that teaching method had no effect on the math score.

This subtle difference can be important when the observations are highly correlated, because you might be misled into overestimating the significance of your results when the individual univariate tests are all significant. Similarly, if there is a subtle difference between the groups that can only be discerned when considering all three scores simultaneously, the MANOVA tests might be able to detect it, where the univariate analyses would not. The multivariate tests help you decide whether the significance is due to different relationships among the dependent variables or just to one underlying mechanism being measured several ways.

In the univariate analysis of variance, most statisticians agree that the statistic of choice to test the null hypotheses generally associated with linear models is the following: an F -test that uses a statistic formed as a ratio of the mean square attributable to an effect and the mean square attributable to error. Because such a consensus does not exist for multivariate hypothesis tests, StatView provides the four most popular multivariate statistics. Each of these tests is formed from the eigenvalues of matrices that are analogous to the mean squares used in univariate hypothesis tests, but they represent different statistical approaches to the multivariate problem. In many cases, the tests lend the same conclusions. However, in some cases they will not, because each test is more sensitive against some alternative hypotheses than others, although none has been shown to be universally superior to the others. The choice of statistic, therefore, is often rather arbitrary. **Wilks' Lambda** is favored by some statisticians because it is derived by the maximum likelihood technique, which has been shown to be effective and useful for deriving similar hypothesis tests in other experimental situations. StatView also computes **Roy's Greatest Root**, the **Hotelling-Lawley trace**, and **Pillai's trace**.

Repeated measures ANOVA

Many times an experiment or study will result in several measurements being taken on each experimental unit. (An experimental unit is the smallest object involved in a study, for exam-

ple, a student in a study on teaching methods, a store in a sales survey, or a manufactured item in a quality control study.) These several measurements may represent different things, such as height and weight, or the levels of different substances in a sample of blood, or the amount of money spent on various budget items. In such cases, you can analyze your data with either a set of univariate analyses, or with a multivariate analysis. (See [“ANOVA,” p. 79](#), and [“MANOVA and MANCOVA,” p. 81](#).)

There are also many situations where the measurements taken on each experimental unit are essentially the same but measured under different times or experimental conditions. Examples would be the level of a given substance in the blood at 1, 2, 5, and 10 days after treatment, or the performance of students on a particular test at ages 5, 6, and 7, or the productivity of workers under a variety of environmental conditions. Sets of measurements like these are known as **repeated measurements**. The statistical technique often used to analyze them is known as repeated measures analysis of variance.

The main distinction between a repeated measures analysis and a standard multivariate analysis is that in a repeated measures analysis of variance, the different measurements each represent essentially the same quantity measured on the same experimental unit but under different conditions. Often the measurements are simply repeated over time, but repeated measures analysis of variance can be appropriate in other settings as well.

Some repeated measures designs, especially those where the effect of interest is time, have no alternative. In other cases, an alternative may exist. For example, in the productivity study, an alternative to measuring each of the workers under each of the conditions would be selecting a large group of workers, randomly assigning them to different environmental conditions, and measuring their productivity. However, this alternative has two potential drawbacks. First, it might be expensive, difficult, or even impossible to find enough subjects. Second, there is the danger that despite randomization, a larger proportion of the most productive workers could end up in one group, causing a false association between that group's environment and its increased productivity. This effect is eliminated in the repeated measures design, as each subject is its own control, so individual effects can be removed. This property subjects repeated measures designs to a natural restriction in randomization. This is one of the reasons why repeated measures designs require special analysis.

The appropriate measure of variability for assessing the effects involving the repeated measure (known as “within-subject effects”) differs from the measure used to assess effects averaged over subjects (known as “between-subjects effects”). A term labelled “Subject(Group)” is automatically added to your ANOVA table. It is a **within subjects error** term. StatView must calculate more than one estimate of variability to assess the importance of the different effects in a repeated measures design, since the variability of measurements taken on the same individual is generally smaller than that of measurements taken on different individuals. For those effects that compare differences among the grouping variables (**between subjects** tests), the usual estimate of residual error is appropriate. But for tests involving the repeated measure itself (**within subject** tests), a separate estimate of error must be calculated.

Post hoc tests (Multiple comparisons)

When your ANOVA determines that some of the effects in your model are significant, you will usually want to examine the mean values of the dependent variable for each level of the factor(s) to determine which means are different from each other. In the corn variety/fertilizer example, it would be helpful to examine the mean value of yield for each level of variety and fertilizer, and each of their combinations, to determine which fertilizer types and/or corn varieties result in the highest yield.

When you are testing main effects, there are several tests available to help you find out where the differences in the dependent variable's values are coming from. These tests, known as **post hoc tests**, or **multiple comparisons**, are specifically designed to make many comparisons among a group of means and still present results that are accurate at the significance levels that they report.

StatView offers a variety of post hoc tests. Each test addresses a potentially important consideration of a researcher that no other procedure addresses. However, if you do not have a preference for a particular procedure, the Games-Howell is one of the more useful, recently developed post hoc procedures. The Dunnett is a good alternative if you want to compare a control mean to a collection of treatment means. All tests are based on two-tailed, null hypothesis comparisons, so they make no distinction between the case where a given mean is larger than another mean and the case where a given mean is smaller.

Each test defines a particular critical difference for a given pair of means. These critical differences vary as a function of cell sample sizes and variances (a **cell** is one level of a factor), the number of means involved in a set of comparisons, concern about either type I (alpha) or type II (power) errors, and whether or not you want the type I and type II error rates to be associated with a single comparison between two means or with a set of comparisons among a collection of means.

The following table summarizes the assumptions of each test and shows the maximum number of means allowed for a set of comparisons.

Test Usefulness of test	Significant <i>F</i> -ratio	Homogeneity of variance	Equal cell <i>n</i>	Cell normality	Maximum number of means
Fisher PLSD all pairwise comparisons with multiple <i>t</i> statistic	yes	yes	yes	yes	no limit
Tukey-Kramer control overall Type I error	no	yes	either equal cell <i>ns</i> or ratios $\geq 3:1$	yes	20
SNK all pairwise comparisons, ordered from smallest to largest	yes	yes	yes	yes	20
Scheffé's robust to violations of assumptions	yes	no	no	no	no limit

Games/Howell robust to unequal n s, heterogeneous variances, non- normality	yes	no	no; cell n s ≥ 6	no	20
Bonferroni/Dunn all pairwise comparisons	no	yes	yes	yes	no limit
Dunnett comparison of set of treatment means to a control mean	no	no	no	yes	20

Some of StatView's tests control the probability of type I error per *comparison*, while other procedures control the error probability per *set of comparisons*. (Recall that **alpha error** or **type I error** is the probability of incorrectly rejecting a true null hypothesis—that is, the probability of concluding that a pair of means are significantly different when they are really not different.) The following table summarizes how each post hoc addresses type I errors. If you specify a low alpha value, error rates associated with violations of the assumption of normality are almost negligible, as is the difference between error rate per comparison versus error rate per set of comparisons.

Procedure	Error Summary	
Fisher PLSD	$p = \alpha$ per comparison	and $p > \alpha$ per set of comparisons
Tukey-Kramer	$p \leq \alpha$ per comparison	and $p = \alpha$ per set of comparisons
SNK	$p = \alpha$ by layer of comparison	and $p > \alpha$ per set of comparisons
Scheffé's	$p < \alpha$ per comparison	and $p < \alpha$ per set of comparisons
Games-Howell	$p = \alpha$ per comparison	and $p = \alpha$ per set of comparisons
Bonferroni /Dunn	$p < \alpha$ per comparison	and $p = \alpha$ per set of comparisons
Dunnett	$p \leq \alpha$ per set of comparisons	

Post hoc tests produce tables like the following. The first column reports the mean difference between groups. The second column reports the mean difference that would be required for it to be significant at the level you set in the dialog box. The third column reports the probability that there is no difference between groups. The "S" to the right of a row appears only when the difference is significant at the alpha level you chose.

Fisher's PLSD for Weight
Effect: Country
Significance Level: 5 %

	Mean Diff.	Crit. Diff	P-Value	
Japan, Other	165.865	132.607	.0147	S
Japan, USA	-306.653	125.126	<.0001	S
Other, USA	-472.518	117.555	<.0001	S

If you determine that an interaction among some of the factors in your model is significant, you should then examine the means of the dependent variable for each combination of the factors in question to get more insight into what the interaction means. However, there are no statistical tests for interactions like the multiple comparisons tests for main effects. You could split your data by one of the factors and perform a multiple comparisons test on the other fac-

tor to help determine where the significant interaction is arising from, but keep in mind that such a test does not use all of your data, so it might not be powerful enough to establish where the differences lie. In many cases, examining an interaction plot or means table can be more worthwhile.

Fisher's Protected Least Significant Difference

Assuming that a significant F -ratio has been defined (an F -ratio is significant if the reported p value is less than a pre-specified significance level), **Fisher's PLSD** evaluates all possible pairwise comparisons with a multiple t -statistic. This multiple t -test assumes that the means have been ordered from smallest to largest. It determines the critical value to be exceeded, for any pair of comparisons, on the basis of the maximum number of steps between the smallest and largest mean. StatView implements the test in a general way for use with unequal as well as equal sample n s. The original PLSD assumed equal sample size.

The PLSD is the most liberal post hoc procedure of the three available in StatView. By insisting that the associated main effect be significant, $p < \alpha$, Fisher argued that the associated probability of a Type I error across all pairwise comparisons would be approximately α .

It is possible for an effect to have a significant F -ratio associated with it but not have any significant pairwise comparisons. This occurs when the contrasts of some linear combinations of the means, not necessarily pairwise, are significantly different. The probability of a type I error is also inflated when the sample sizes are unequal.

Scheffé's F

Scheffé's F (1953) procedure for post hoc comparisons is very robust to violations of the assumptions typically associated with multiple comparison procedures. It may be used when you have unequal cell n s as well as when you have **heterogeneous** variances, that is, in the case where the variances of the cells are not equal. (In the case of heterogeneous variances, the basic assumptions of the analysis of variance are violated, and the significance levels associated with all the hypothesis tests must be interpreted with caution.) This procedure was developed with the assumption that all possible comparisons would be made; in StatView, the procedure has only been implemented to make pairwise comparisons of means.

The Scheffé is the most conservative of the paired comparison procedures. However, because it was the first paired comparison procedure with demonstrated robustness to assumption violations, it has enjoyed a long popularity and is still used by many researchers.

Bonferroni/Dunn

The **Bonferroni/Dunn** procedure is a multiple comparison procedure for making all possible pairwise contrasts amongst a collection of means. There are $(p(p-1)/2)$ comparisons when you implement the Dunn as a procedure for comparing all pairwise differences for p means. It has no limit on the number of comparison means that may be contrasted. This procedure tends to be less conservative than Scheffé's F ; it is more likely to determine that differences are significant.

The procedure is attributed to Dunn (1961) and based on the Bonferroni inequality and is sometimes referred to as the Bonferroni t -procedure or the Bonferroni/Dunn test.

Dunnett's Test

In an experiment it is often desirable to compare the collection of treatment means to a control mean. The Dunnett (1955, 1964) is such a specialized multiple-comparison procedure. If there is a total of p means, then there will be $p - 1$ paired comparisons for this comparison procedure, whereas for the other post hoc procedures there will be $p(p - 1)/2$ comparisons. When computing the probable "error" associated with the contrasts, the Dunnett considers only the $p - 1$ comparisons to the control. It is therefore more efficient than the general post hoc procedures when its use is appropriate. Generally, it may be assumed that when using the Dunnett, as opposed to the Tukey or some other multiple comparison procedure, a smaller difference will be required for significance.

As implemented, the Dunnett can be used when the control group n and the comparison group n are unequal. It can also be used when the control group variance is not equal to the comparison group variance. As with the Games-Howell procedure, the critical value to exceed is determined in part by the variances and cell n s associated with each pairwise comparison.

Tukey-Kramer Test

The **Tukey-Kramer Test**, or Tukey's HSD (Honestly Significant Difference), originally developed by John Tukey in 1953, is an extension of Fisher's PLSD. It is intended to keep the experiment-wise probability of a type I error at alpha. Since it controls for overall error, the Tukey-Kramer test detects fewer significant differences than other tests. (See Keselman and Rogan, 1978, for a thorough discussion of Tukey's procedure.)

While it makes the same assumptions as the LSD, the Tukey HSD uses the studentized range statistic instead of the Student t -distribution. The Tukey HSD, when all cell n s are equal, determines a single critical value that all comparisons must exceed to achieve significance. This critical value is a function of the total number of means involved in the collection of comparisons.

The original HSD assumed all cell n s to be equal. However, Kramer (1956) modified it to be used with post hoc tests having unequal cell n s. This modification is applied to Tukey's procedure to allow the HSD to be used when the cell n s are not equal. By the early 1980s researchers had discovered that this modification made the Tukey procedure very robust to violations of equal cell n s (Jaccard, Becker and Wood, 1984; Games, Keselman and Rogan, 1981; Dunnett, 1980).

The original Tukey test is calculated if all cell n s are equal. The Kramer (1956) modification, properly referred to as the Tukey-Kramer test is calculated if at least one pair of cells has unequal n s. Although this procedure is similar to an extension of the Tukey HSD, there is an important distinction: the value that a pairwise comparison must exceed for significance changes every time that a cell n changes.

With regard to error, Dunnett (1980) and Keselman & Rogan (1978) both suggest that with extreme discrepancies amongst cell n s—ratios of 3:1 or greater—the Kramer modification of

the HSD is also conservative and behaves very much like the HSD for equal cell n s. However, Dunnett also suggests that when cell n s are approximately equal the Kramer modification is no longer conservative. For approximately equal cell n s, the probability of a type I error is greater than alpha. Thus, it is best to assume that the HSD is appropriate for either equal cell n s or very discrepant cell n s.

Games-Howell Test

Perhaps the most robust of a new genre of multiple comparison procedures is the one developed by **Games and Howell** (1976). This procedure seems to be very robust with cells having unequal n s and heterogeneous variances, as well as those violating the assumption of normality (Jaccard, Becker and Wood, 1984; Games, Keselman and Rogan, 1981; Keselman and Rogan, 1978; Dunnett, 1980a and 1980b).

While in the Tukey tradition, this procedure utilizes a Behrens-Fisher approach to estimating the error of a comparison, and an approximation procedure that follows from Smith (1936), Welch (1949), and Satterwaite (1946) for estimating degrees of freedom. It also requires each cell n to be at least 6.

This procedure defines a different value for each pairwise comparison to exceed for significance. The critical value to exceed is determined in part by the variances and cell n s associated with each pairwise comparison.

Student-Newman-Keuls Test

The **Student-Newman-Keuls** test is a post hoc that makes all pairwise comparisons. It orders all means from smallest to largest.

If you assume that there are r means, the largest difference will involve means that are r steps apart. This difference is tested for $p = \alpha$. If it is significant, the differences associated with means $r - 1$ steps apart are tested for $p = \alpha$. If they are all significant, the differences associated with means $r - 2$ steps apart are tested for $p = \alpha$, etc. Thus the procedure is sometimes called a stepwise or layered multiple comparison procedure.

For this multiple comparison procedure, the error rate deals with the set of comparisons associated with a particular step, e.g., all comparisons that are $r - 2$ steps apart. Therefore it has neither an experiment-wise nor comparison-wise error rate.

Limitations of post hoc tests

Repeated measures designs Multiple comparison procedures are designed to allow comparisons between several groups of uncorrelated means, under the assumption that the means are normally distributed with a common variance. Many of these methods rely on results based on the order statistics of uncorrelated variables derived from normal distributions with the appropriate variances. However, repeated observations on a given subject are correlated, and so the means based on these groupings (i.e., within subjects factors) are correlated. Therefore, comparing the means of within factors with post hoc tests is not recommended.

Since the between factors summarize observations over the within factors, multiple comparison tests are appropriate for the between subjects effects in a design. However, these are usually not the factors of major interest in repeated measures designs, so the use of multiple comparisons procedures for these factors is usually not a high priority among users.

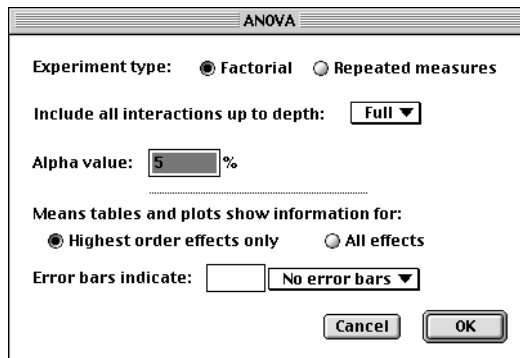
Interaction effects These multiple comparison tests are designed to allow comparisons of uncorrelated means. Strictly speaking, they should really only be used in models with a single factor. As a compromise to utility, however, the notion of multiple comparisons is generally extended to allow comparing the means corresponding to the levels of any single factor in the design. You should remember that in most cases the only information from the rest of the design that is used in the multiple comparisons test is the error mean square; the means being compared are simply arithmetic means, ignoring any other factors in the model.

When a factor of interest is an interaction, it is much more difficult to ignore the other terms in the model when comparing means. Due to imbalances in designs, arithmetic means are often not consistent with the linear model being considered. Furthermore, there is some question whether the multiple comparisons procedures are still valid when a certain structure is being assumed through the modeling process. In other words, it is somewhat awkward to claim that you are modeling the mean of a particular cell as the sum of several terms in the linear model but then to use the simple arithmetic mean to compare these cells. The same problems exist in the case of a single factor; however, it is much easier to rethink the problem as being one of comparing several means in the single factor case, because it is such a natural extension of the spirit under which the procedures are derived.

An alternative would be to arrange your dataset so that the design were essentially a one way ANOVA where each level of the single factor represented a unique combination of the factors in the desired interaction. Then you could run the usual multiple comparisons tests. Since these tests would ignore the underlying concept of interaction, however, their use would be questionable. Furthermore, since a given interaction usually has many factor combinations, the post hoc test tends to be less than optimally useful, since the procedure must protect itself against errors from the many comparisons being performed.

Dialog box settings

You set analysis parameters for ANOVA results in this dialog box:



The image shows a dialog box titled "ANOVA". It contains the following settings:

- Experiment type:** ☒ Factorial ☐ Repeated measures
- Include all interactions up to depth:** Full ▼
- Alpha value:** 5 %
- Means tables and plots show information for:** ☒ Highest order effects only ☐ All effects
- Error bars indicate:** No error bars ▼
- Buttons:** Cancel, OK

Experiment type You must first choose the type of your ANOVA, either factorial or repeated measures. If you specify a repeated measures design, StatView automatically builds the correct ANOVA table for this type of model. Remember, if you select repeated measures, your dataset must contain a compact variable to identify the within factor(s). For more information on compact variables, see the following section, [“Data requirements,” p. 90.](#)

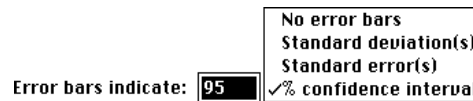
Include all interactions up to depth Choose the depth of interactions to be included in the model. The default is Full, which includes all main effects and all interactions at every depth. You can choose 1 for just the main effects, 2 for all main effects and second-order (two-factor) interactions, and so on, up to 7 for all main effects through seven-factor interactions.

The commentaries on each type of model in the preceding [“Discussion,” p. 73,](#) offer some advice for determining which interactions should be included and how to interpret the significance of interaction effects. Generally, you should begin by including full interactions; depending on the results, you might then want to click Edit Analysis and adjust the depth of the model.

Alpha value Specify as a percentage the alpha value (significance level) to be used for post hoc tests and power calculations. The default is 5%, or $\alpha = 0.05$.

Means tables and interaction plots The choices at the bottom of the dialog box control how many means tables and interaction plots are displayed (if you have selected these results from the analysis browser). If you choose “Highest order effects only,” StatView produces the means tables and interaction plots for only the effects of the highest order, according to your Depth choice. If you choose “All effects,” means tables and interaction plots appear for each effect included the model.

Error bars StatView can add error bars to your graphs. You can choose among no error bars, or the number of standard deviations or standard errors you specify, or confidence intervals for the percentage you specify.



Note: post hoc tests are no longer requested in the ANOVA dialog box. Instead you must select the post hoc test(s) you desire in the analysis browser. See [“Results,” p. 95.](#)

Data requirements

A factorial ANOVA requires one or more nominal independent variables with one continuous dependent variable. MANOVA requires one or more nominal independent variables and two or more continuous dependent variables. ANCOVA and MANCOVA models include one or more continuous independent variables. A repeated measures ANOVA requires a single compact variable and optionally one or more nominal variables.

Your data must be organized in a way that allows StatView to identify which group(s) the observations belong to. For a repeated measures design, you must create a compact variable to identify the groups of the within factor(s). For an introduction to dataset organization including compact variables, see [“Datasets,” p. 49 of Using StatView.](#) In addition, the [“Exercises,”](#)

[p. 96](#), will help you see how to organize your data for both factorial and repeated measures experiments.

Factorial

In a factorial experiment, you assign one or more nominal variables (factors) and one or more continuous variable (the dependent variables). The nominal variables are the independent variables for the analysis. Your dataset needs to be organized so that all the values of the dependent variable appear in a single column. Each nominal variable appears in a separate column. The nominal variables divide your dependent data into groups. There will be one row in the dataset for each subject or other experimental unit in the analysis.

	Height	Gender	Weight
1	Tall	Male	145
2	Tall	Female	123
3	Short	Male	245
4	Tall	Male	223
5	Short	Female	142

The dataset above shows the organization for a factorial ANOVA. All observations for the dependent variable, Weight, are in a single column. The grouping variable Height is a separate nominal column identifying the group (tall or short) for each Weight measurement. The variable Gender is another separate nominal column that identifies the group (male or female) for each Weight measurement. Each row in the dataset represents a separate, unique, subject in the experiment.

Some users may wish to use compact variables to identify the groups of the between factors for their factorial ANOVA. In a compact variable, the values of the columns (variables) in the usual dataset organization become the rows in the dataset with the compact variable. If you plan to use a compact variable, please read [“Compact variables,” p. 84 of *Using StatView*](#).

Repeated measures

In a repeated measures experiment, you can have one or more between factors and one or more within factors. Between factors must be set up as individual nominal columns. A single within factor must be set up as **compact variables**, and multiple within factors must be set up as **complex compact variables**.

One between factor and one within factor

Consider an experiment testing the mobility of six athletes, male and female, at four temperatures (60°, 70°, 80° and 90° Fahrenheit). The dataset for this experiment would have six rows, one for each subject in the experiment, and five columns. One column would indicate the gender of the subject. This nominal column would be a between factor in the repeated measures experiment. The other four columns would record the mobility measurements taken at the four different temperatures.

For StatView to understand that these four columns are related and represent different groups (or levels) of the within factor, they must be combined into a single compact variable. To cre-

ate a compact variable, you select the columns that represent the groups of the within factor and click the Compact button at the top of the dataset. You then need to enter a name for the variable. You might enter the data like this:

	Gender	60	70	80	90
1	male	2.40	3.60	4.80	5.00
2	male	4.30	4.50	5.00	4.25
3	female	5.30	6.00	2.60	5.10
4	female	4.50	2.70	6.00	4.00
5	male	4.70	1.50	4.70	3.65
6	female	2.30	4.00	2.15	4.00

To compact the columns into a single compact variable, select the four columns, click Compact, and enter a name for the repeated measure, such as “Mobility.” Now your dataset might look like this:

	Gender	Mobility			
		60	70	80	90
1	male	2.40	3.60	4.80	5.00
2	male	4.30	4.50	5.00	4.25
3	female	5.30	6.00	2.60	5.10
4	female	4.50	2.70	6.00	4.00
5	male	4.70	1.50	4.70	3.65
6	female	2.30	4.00	2.15	4.00

A compact variable is a unique data structure. All the cells of a compact variable taken together are the measurement variable (the continuous dependent variable), and the way those cells are arranged into four columns indicates their group memberships (the nominal within factor). Therefore, in the variable browser, the nominal part of a compact variable is listed in a drop-down list under the name of the continuous variable:

Gender	<input type="checkbox"/>	<input type="checkbox"/>
▼ Mobility	<input checked="" type="radio"/>	<input type="checkbox"/>
Temperature	<input type="checkbox"/>	<input type="checkbox"/>

For more detail on creating and understanding compact variables, see [“Compact variables,” p. 84 of Using StatView.](#)

Two between factors and two within factors

Suppose the mobility experiment had another between factor for type of athlete (swimmer or runner) and the four mobility measurements were repeated a week later for an additional within factor. You would simply add a nominal variable Sport for the additional between factor. For the additional within factor, you would need to create a complex compact variable. Your dataset might look like this:

	Gender	Sport	Mobility							
			Week 1				Week 2			
			60	70	80	90	60	70	80	90
1	male	swimmer	2.40	3.60	4.80	5.00	2.50	3.70	4.70	4.80
2	male	runner	4.30	4.50	5.00	4.25	4.20	4.65	5.10	4.20
3	female	swimmer	5.30	6.00	2.60	5.10	5.25	6.05	2.40	4.95
4	female	runner	4.50	2.70	6.00	4.00	4.50	2.50	6.15	4.00
5	male	swimmer	4.70	1.50	4.70	3.65	4.80	1.70	4.70	3.45
6	female	runner	2.30	4.00	2.15	4.00	2.15	3.80	2.10	4.15

In the variable browser, you would see a continuous variable (Mobility) with two nominal parts, Trial and Temperature:

Gender	N	↑
Sport	N	
▼Mobility	⊙	
Trial	N	
Temperature	N	

Step-by-step instructions for creating complex compact variables appear under [“Complex compact variable,”](#) p. 89 of *Using StatView*.

Model design	Variable browser buttons (and their usage markers)	
ANOVA	Independent (X)	Select a nominal variable (or the nominal part of a compact variable) for each factor and click Independent. Additional variables are added to the analysis.
	Dependent (Y)	Select one continuous variable (or the continuous part of a compact variable) and click Dependent. If additional Dependent variables are assigned, the analysis becomes a MANOVA.
MANOVA	Independent (X)	Select a nominal variable (or the nominal part of a compact variable) for each factor and click Independent. Additional variables are added to the analysis.
	Dependent (Y)	Select two or more continuous variables (or the continuous parts of compact variables) and click Dependent. Additional Dependent variables are added to the analysis.
ANCOVA	Independent (X)	Select a nominal grouping variable (or the nominal part of a compact variable) for each factor and click Independent. Then select a continuous variable (or the continuous part of a compact variable) for each covariate and click Independent. Or, you can select the name of a compact variable and click Independent to assign both parts at once, the continuous part as a covariate and the nominal part as a factor. Additional nominal or continuous Independent variables are added to the analysis.
	Dependent (Y)	Select one continuous variable (or the continuous part of a compact variable) and click Dependent. If additional Dependent variables are assigned, the analysis becomes a MANCOVA.
MANCOVA	Independent (X)	Select a nominal grouping variable (or the nominal part of a compact variable) for each factor and click Independent. Then select a continuous variable (or the continuous part of a compact variable) for each covariate and click Independent. Or, you can select the name of a compact variable and click Independent to assign both parts at once, the continuous part as a covariate and the nominal part as a factor. Additional nominal or continuous Independent variables are added to the analysis.
	Dependent (Y)	Select two or more continuous variables (or the continuous parts of compact variables) and click Dependent. Additional Dependent variables are added to the analysis.

Repeated measures ANOVA	Independent (X)	Select a nominal variable (or the nominal part of a compact variable) for each between factor (if any) and click Independent. Any additional nominal variables added as Independents are added to the analysis as additional between factors.
	Dependent (Y)	Select one compact variable containing the within factor(s) and click Dependent. (Multiple within factors intended for a single repeated measures ANOVA must be entered as a single complex compact variable; see “Data requirements,” p. 90.) Any additional compact variable added as a Dependent causes the analysis to clone with the new variable as the within factor.
All models	Split By (S)	When you assign one or more split-by variables to an [M]AN[C]OVA, results for each cell defined by the split-by variable(s) are displayed separately.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 73.](#) The default output for this statistic is the ANOVA table.

ANOVA Table	Table containing the degrees of freedom, sum of squares, mean square, F value, p value, lambda, and power for each effect in the ANOVA model.
Means Table	Table containing the count, mean, standard deviation, and standard error for each group or combination of groups in the nominal variable(s).
ANOVA Coefficients Table	Table containing the coefficient, standard error, t -test, and p value for the intercept term and each level of each effect. (For interaction effects, each combination of levels is listed. For interactions that include covariates, each combination of levels for the nominal variable/s in that covariate is listed.) Not available for repeated measures models.
Interaction Bar Chart	Graph displaying the means of each group or combination of groups in the nominal variable(s) as bars. Error bars may be added using the dialog box. For factorial designs with more than one factor, the factor assigned last is used as a legend variable. For repeated measures designs, between factors appear in the legend.
Interaction Line Chart	Graph displaying the means of each group or combination of groups in the nominal variable(s) as points connected by lines. Error bars may be added using the dialog box. For factorial designs with more than one factor, the factor assigned last is used as a legend variable. For repeated measures designs, between factors appear in the legend.
MANOVA Tables	Table containing the statistics, F values, numerator and denominator degrees of freedom, and p values for Wilks' Lambda, Roy's Greatest Root, Hotelling-Lawley Trace, and Pillai Trace.

Post Hoc Tests	Tables containing Fisher's PLSD, Scheffé's <i>F</i> , Bonferroni/Dunn, Dunnett's, Tukey-Kramer, Games-Howell, and Student-Newman-Keuls statistics for each main effect. The tables show the mean difference and critical difference for all test. Fisher, Scheffé's, and Bonferroni/Dunn are the default tests; for a different combination of tests, select the specific tests you want from the Post Hoc Tests list in the analysis browser. For Fisher, Scheffé, and Bonferroni/Dunn, the table includes the <i>p</i> value for the difference between all pairs of groups in the nominal variable(s). For Dunnett's, Tukey-Kramer, Games-Howell, and Student-Newman-Keuls, mean differences are compared to critical differences from stored tables for the specified value of alpha. The symbol "S" appears to the right of each row if the mean difference exceeds the critical difference.
----------------	---

Note that for interaction charts, StatView places groups of the first variable in the interaction (the first variable assigned to the model) in the legend. Cell plots (see ["Cell Plots," p. 237](#)) offer additional control over creating interaction plots.

Templates

The following templates provide ANOVA results.

ANOVA and t-tests	ANOVA or ANCOVA	ANOVA, means, and Fisher's PLSD tables; interaction bar plot.
	ANOVA Post Hoc Tests	Interaction line plots and Fisher's PLSD, Scheffé's, Bonferroni/Dunn, Dunnett's, Tukey-Kramer, Games-Howell, and Student-Newman-Keuls tables.
	Interaction Bar Chart	Interaction bar plot with 95% confidence level error bars.
	Interaction Line Chart	Interaction line plot with 95% confidence level error bars.
	MANOVA or MANCOVA	ANOVA, means, MANOVA, and Fisher's PLSD tables; interaction bar plot.
	Repeated Measures ANOVA	For each effect, ANOVA, means, and MANOVA tables, interaction bar chart.

Exercises

Fully factorial ANOVA

In this exercise you perform a factorial ANOVA using data on weight and type for 116 cars from different countries. You will determine whether car weight is related to the type and country of origin of cars.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under ANOVA, select ANOVA Table and click Create Analysis
- Click OK to accept the default analysis parameters

- In the variable browser, select Country and Type and click Independent
- Select Weight and click Dependent

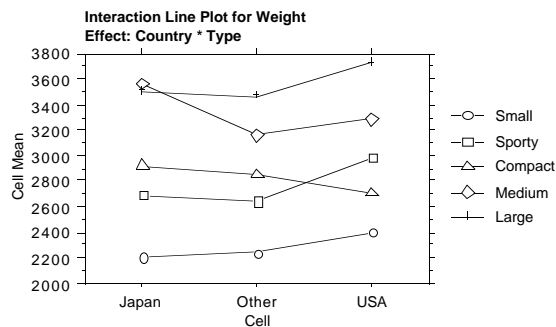
Since Weight is the dependent variable in the analysis, a Y usage marker appears next to it in the variable browser. Type and Country are independent variables, marked with X.

ANOVA Table for Weight

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Country	2	246287.238	123143.619	1.663	.1946	3.327	.331
Type	4	14307811.192	3576952.798	48.317	<.0001	193.266	1.000
Country * Type	8	1404272.808	175534.101	2.371	.0221	18.969	.874
Residual	101	7477200.453	74031.688

From this ANOVA table, you can see that Type has a strong influence on the variable Weight, as indicated by the low p value, < 0.0001 . The interaction of Type and Country also seems to have a strong influence. The main effect of Country does not, however, as its much higher p value shows. We will now examine the interaction of Type and Country more closely with an interaction plot.

- Make sure the ANOVA table is still selected
- In the analysis browser under ANOVA, select Interaction Line Plot and click Create Analysis



Notice that the lines for the different types of cars are spread out over the range of weights. This confirms that the type of car has a significant main effect. To understand the interaction between type and country, concentrate on the places in the graph where the lines are not parallel. For example, sporty cars made in the USA are heavier than other sporty cars, but USA compact cars are lighter than other compact cars. You might also like to produce an interaction chart which uses side-by-side bars to show this information.

Repeated measures ANOVA

In this exercise, we perform a repeated measures ANOVA using data from a study of industrial health, testing the effectiveness of several techniques for teaching the use of a respirator mask. Subjects are divided randomly into three groups: a control group that received no training in the use of the mask; a group that received a detailed instruction sheet; and a third group that attended a thirty minute class. The effectiveness of the mask (measured as the amount of particulate matter that passed through the mask while performing a fixed task—lower amounts

mean better effectiveness) was measured for each of the subjects before training and also one and two weeks after training.

We want to find out whether, averaged over time, there is any difference in effectiveness among the three teaching techniques. We have two questions about the within-subjects factor, Time: whether the test scores change over time if averaged over treatments, and whether the pattern of change over time is the same for each teaching technique.

- Open Teaching Effectiveness Data from the Sample Data folder

The first column of the dataset contains the group labels for teaching technique—Control, Instructions and Lecture—in a category variable. The three remaining columns are a compact variable recording the effectiveness of the mask before training (week 0) and 1 and 2 week after training. Repeated measures designs require that data are arranged in a compact variable (see “[Compact variables](#),” p. 84 of *Using StatView*). Assigning a compact variable to a repeated measures analysis as a Dependent variable also assigns the nominal portion of the compact variable as a between-subjects factor.

- From the Analyze menu, select New View
- In the analysis browser under ANOVA, select ANOVA Table and click Create Analysis
- Choose Repeated Measures and click OK
- In the variable browser, select Teaching and click Independent
- Select Effectiveness and click Dependent.

ANOVA Table for Effectiveness

	DF	Sum of Squares	Mean Square	F-Value	P-Value
Teaching	2	26.751	13.376	2.154	.1370
Subject(Group)	25	155.225	6.209		
Time	2	18.926	9.463	8.783	.0005
Time * Teaching	4	18.171	4.543	4.216	.0051
Time * Subject(Group)	50	53.869	1.077		

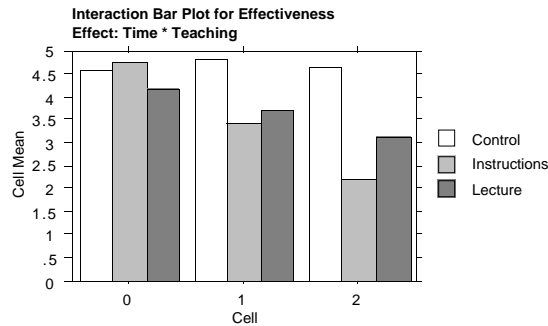
The between-group main effect for teaching technique is not significant. This means that averaged over the three times, there was no difference in the effectiveness scores of the three teaching methods. This test could be misleading, however, since it includes the pretraining (week 0) scores, which you would expect to be the same for all groups.

In many repeated measures experiments, the between-group main effect and interaction tests have this limitation and are therefore not the main focus of the analysis. Keep in mind, however, that including these effects reduces the estimate of residual error, making the tests more powerful, and providing an opportunity to study the between-subjects by within-subjects interactions, which are usually of great interest.

You can see that time after training had a very significant effect. This makes sense: as the subjects became more familiar with the respirator masks, they learned to use them more effectively. Of special interest is the significant teaching technique-by-time interaction, indicating that the patterns of changes in effectiveness over time differed by teaching technique.

We can look at an interaction plot to see how effectiveness differs among groups and times:

- Make sure the ANOVA table is still selected
- In the analysis browser under ANOVA, select Interaction Bar Plot and click Create Analysis



Now we can see how the significance of the Time*Teaching interaction arose. The control group had very little change in effectiveness over time, but the two experimental groups saw considerable improvements. The group that attended the lecture showed progress over time, but the group with the instruction sheets showed even more progress. The instruction sheets seem to have been the most effective teaching method.

ANCOVA

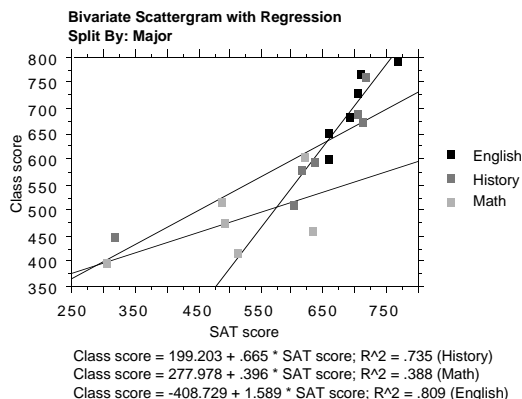
Suppose a university English department wants to know whether its first-year composition course is as effective for history and math majors as it is for English majors. We could do a simple analysis of variance with final class scores as the dependent variable and major as the factor. However, students could have differing verbal abilities, and we must control for that by including their Verbal SAT (Scholastic Achievement Test) scores as a covariate. We might have data such as these (which are simulated):

- Open Writing Scores from the Sample Data folder

The main question, of course, is whether the course is equally effective for students of different majors. Secondly, we want to estimate the average class score for students in each major. Finally, we want to know whether SAT scores are effective for controlling for variability among individual students.

The first concern is to test for homogeneity of slopes—that is, whether the interaction term is significant. A visual way to do this is to create a bivariate plot of the dependent by the covariate, with separate lines for each group in the factor.

- From the Analyze menu, select New View
- In the analysis browser under Bivariate Plots, select Bivariate scattergram and click Create Analysis
- Choose Regression lines, and for “When split, show lines for,” choose “each group separately”
- Click OK
- In the variable browser, select SAT Score and click X
- Select Class score and click Y
- Select Major and click Split By



None of the three regression lines are flat, suggesting that the covariate is a meaningful term to include. The lines have slightly different slopes, but it's unclear whether the slopes are *significantly* different. The lines are roughly parallel, suggesting that the covariate-factor interaction probably is *not* significant, but we should test that to be sure. We will need to examine the statistical results. (We also notice how sparse the dataset is, which should give us pause in interpreting results.)

- Make sure the plot is still selected
- In the analysis browser under ANOVA, double-click ANOVA Table
- Click OK to accept the default analysis parameters
- In the variable browser, select Major and click Remove, then click Independent
(The other variable assignments are fine the way they were “adopted” from the plot.)

StatView automatically treats Major as a factor since it is nominal, and it treats SAT Score as a covariate or regressor since it is continuous. In the variable browser, both variables have X usage markers to indicate that they are independent variables, and Class score has a Y usage marker to indicate that it is a dependent variable.

ANOVA Table for Class score

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
SAT score	1	49255.924	49255.924	15.652	.0016	15.652	.966
Major	2	7732.295	3866.148	1.229	.3246	2.457	.215
SAT score * Major	2	11580.897	5790.449	1.840	.1979	3.680	.306
Residual	13	40911.036	3147.003	-	-	-	-

We can tell from the F and p values that the SAT Score*Major interaction term is not significant, just as we expected from the roughly parallel regression lines for each factor level. Therefore, we can remove the interaction from the model. (Notice that we focus our attention on the interaction term before paying much attention to the results for the main effects. That the interaction term is not significant is good news; parallel slopes are one of the requirements for analysis of covariance.) We remove the interaction term by choosing interactions up to depth 1—that is, to main effects only:

- Make sure the result is still selected
- Click Edit Analysis

- For the option Include all interactions up to depth, choose 1 (instead of Full) and click OK

ANOVA Table for Class score

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
SAT score	1	71902.115	71902.115	20.547	.0004	20.547	.994
Major	2	22414.844	11207.422	3.203	.0695	6.405	.516
Residual	15	52491.933	3499.462

The F and p values for Major are still not significant, which is good news for the department: the class appears to be as effective for history and math majors as for English majors. (Strictly speaking, the result only tells us we cannot reject the null hypothesis that scores are the same for students of different majors.) Meanwhile, the F and p values for SAT Score show that SAT Score clearly *is* useful for predicting class score, so including the term in the model is useful: it controls for differences already in existence before the experiment.

Randomized complete block ANOVA

The sample dataset Flax Oil Content, from Steel and Torrie (1980), shows percentage measurements of oil content in flaxseed grown in each of four different locations for six different treatments. At each location one plant was inoculated with bacteria as a seedling, one plant in early bloom, one in full bloom, one at a lower dose in full bloom, and one when the plant was ripening. A sixth plant in each location was a control case, not inoculated at all. There was no replication of treatment by location combinations.

- Open Flax Oil Content from the Sample Data folder

	Treatment	Oil content			
		Location 1	Location 2	Location 3	Location 4
1	Seedling	4.4	5.9	6.0	4.1
2	Early bloom	3.3	1.9	4.9	7.1
3	Full bloom	4.4	4.0	4.5	3.1
4	Full bloom (1/100)	6.8	6.6	7.0	6.4
5	Ripening	6.3	4.9	5.9	7.1
6	Uninoculated (Control)	6.4	7.3	7.7	6.7

The measurements, recorded as oil percentage minus 30, are organized in a compact variable. Note that adding or subtracting a constant to each value in a dataset doesn't change the results of the analysis, because all of the sums of squares to be calculated are corrected for the overall mean.

The purpose of the experiment was to determine whether the treatments had any effect on the oil content of the flaxseed. The experiment was replicated in four different locations so that the results of the experiment could be generalized over a wider area. Without such replication, it could have been argued that conclusions might apply only to a certain planting location.

A randomized complete block experiment differs from the usual factorial experiment in that one factor (in this example, location) is included in the analysis simply to control variability and make the experiment more meaningful—not because the effect of that factor is thought to be interesting. This factor is known as the **blocking factor**, or simply the **block**.

Usually only one observation is taken for each treatment and block combination. Therefore, the effect of any interaction between the treatment and blocks cannot be assessed in the usual way. Because of this, the randomized complete block is only appropriate when you know that

there is no interaction between the blocking factor and the treatment. For the analysis to be valid, the experimenter must be certain that the behavior of the treatments is the same in each of the locations studied.

The oil content may be uniformly higher or lower for one location than another, as long as this is true for each of the treatments. In fact, one reason to include treatment in the model is to control for differences of that sort. The randomized complete block analysis is not appropriate if the behavior of the treatments differs among blocks. For example, if one of the blocks were very wet and another dry, and you knew that the soil's moisture content changed the behavior of the bacteria, then a randomized complete block analysis would be inappropriate.

Remember, there must be *no interaction* between the treatments and the blocks. That point is usually confirmed by the researcher's knowledge of the subject matter or by previous experiments. Remember in turn that you must restrict the model to interactions of depth 1—that is, to main effects only.

- From the analysis browser under ANOVA, select ANOVA Table and click Create Analysis
- Select 1 for the option Include all interactions up to depth and click OK
(Only main effects are appropriate for randomized complete block designs.)
- Select Oil content (just the continuous part) and click Dependent
- From the variable browser, select Treatment and Location (the nominal part of the compact variable Oil content) and click Independent

The variable browser's X and Y usage markers indicate that the variables (or compact variable parts) are assigned to independent and dependent roles for the analysis, respectively.

ANOVA Table for Oil content

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Treatment	5	31.652	6.330	4.816	.0080	24.081	.910
Location	3	3.141	1.047	.797	.5147	2.390	.178
Residual	15	19.716	1.314

We can ignore the F and p values for the blocking factor (Location), since that term is only included to control variability, but it is reassuring that we cannot reject the null hypothesis (that oil content is the same among different locations). Being unable to reject that null hypothesis is a requirement for a randomized complete block design to be valid. The F value for Treatment is 4.816, with a p value of .0080, indicating significant differences among treatments. Thus, the time of inoculation by bacteria does have an effect on the oil content of the flax seed.

To find the source of the treatment differences, it is useful to examine a table of means.

- Make sure the ANOVA table is still selected
- In the analysis browser under ANOVA, double-click Means Table

Means Table for Oil content
Effect: Treatment

	Count	Mean	Std. Dev.	Std. Err.
Early bloom	4	4.300	2.233	1.117
Full bloom	4	4.000	.638	.319
Full bloom (1/100)	4	6.700	.258	.129
Ripening	4	6.050	.915	.457
Seedling	4	5.100	.990	.495
Uninoculated(Control)	4	7.025	.585	.293

Means Table for Oil content
Effect: Location

	Count	Mean	Std. Dev.	Std. Err.
Location 1	6	5.267	1.419	.579
Location 2	6	5.100	1.961	.800
Location 3	6	6.000	1.213	.495
Location 4	6	5.750	1.716	.700

The lowest oil percentages are evident in those plants where inoculation took place in early or full bloom. We can disregard the means table for Location, since it serves only as a blocking factor. We can use post hoc tests to compare these means.

- Make sure one of the results is still selected
 - In the analysis browser under ANOVA/Post-hoc tests, select all seven tests and click Create Analysis
- Click and drag or Shift-click to select several adjacent results

Fisher's PLSD for Oil content
Effect: Treatment
Significance Level: 5 %

	Mean Diff.	Crit. Diff	P-Value	
Early bloom, Full bloom	.300	1.728	.7165	
Early bloom, Full bloom (1/100)	-2.400	1.728	.0097	S
Early bloom, Ripening	-1.750	1.728	.0475	S
Early bloom, Seedling	-.800	1.728	.3394	
Early bloom, Uninoculated (Control)	-2.725	1.728	.0043	S
Full bloom, Full bloom (1/100)	-2.700	1.728	.0046	S
Full bloom, Ripening	-2.050	1.728	.0232	S
Full bloom, Seedling	-1.100	1.728	.1949	
Full bloom, Uninoculated (Control)	-3.025	1.728	.0020	S
Full bloom (1/100), Ripening	.650	1.728	.4352	
Full bloom (1/100), Seedling	1.600	1.728	.0671	
Full bloom (1/100), Uninoculated (Control)	-.325	1.728	.6941	
Ripening, Seedling	.950	1.728	.2595	
Ripening, Uninoculated (Control)	-.975	1.728	.2477	
Seedling, Uninoculated (Control)	-1.925	1.728	.0313	S

Again we ignore the results for the blocking factor, Location. Since Fisher's PLSD is the most liberal of the post hoc tests, it is not surprising that it shows significant results (indicated by an "S" to the right of the p value) for the most pairs of treatment levels. Recall that the critical difference (1.728) is the difference between a given pair of means that would be required for the test to be significant at the alpha level set in the ANOVA dialog box (here, .05).

Scheffe for Oil content
Effect: Treatment
Significance Level: 5 %

	Mean Diff.	Crit. Diff	P-Value
Early bloom, Full bloom	.300	3.088	.9996
Early bloom, Full bloom (1/100)	-2.400	3.088	.1834
Early bloom, Ripening	-1.750	3.088	.4879
Early bloom, Seedling	-.800	3.088	.9598
Early bloom, Uninoculated (Control)	-2.725	3.088	.1015
Full bloom, Full bloom (1/100)	-2.700	3.088	.1064
Full bloom, Ripening	-2.050	3.088	.3237
Full bloom, Seedling	-1.100	3.088	.8625
Full bloom, Uninoculated (Control)	-3.025	3.088	.0567
Full bloom (1/100), Ripening	.650	3.088	.9835
Full bloom (1/100), Seedling	1.600	3.088	.5800
Full bloom (1/100), Uninoculated (Control)	-.325	3.088	.9994
Ripening, Seedling	.950	3.088	.9199
Ripening, Uninoculated (Control)	-.975	3.088	.9116
Seedling, Uninoculated (Control)	-1.925	3.088	.3878

By contrast, Scheffé is the most conservative of the post hoc tests, and by its standards none of the pairs of means are significantly different.

Bonferroni/Dunn for Oil content
Effect: Treatment
Significance Level: 5 %

	Mean Diff.	Crit. Diff	P-Value
Early bloom, Full bloom	.300	2.824	.7165
Early bloom, Full bloom (1/100)	-2.400	2.824	.0097
Early bloom, Ripening	-1.750	2.824	.0475
Early bloom, Seedling	-.800	2.824	.3394
Early bloom, Uninoculated (Control)	-2.725	2.824	.0043
Full bloom, Full bloom (1/100)	-2.700	2.824	.0046
Full bloom, Ripening	-2.050	2.824	.0232
Full bloom, Seedling	-1.100	2.824	.1949
Full bloom, Uninoculated (Control)	-3.025	2.824	.0020
Full bloom (1/100), Ripening	.650	2.824	.4352
Full bloom (1/100), Seedling	1.600	2.824	.0671
Full bloom (1/100), Uninoculated (Control)	-.325	2.824	.6941
Ripening, Seedling	.950	2.824	.2595
Ripening, Uninoculated (Control)	-.975	2.824	.2477
Seedling, Uninoculated (Control)	-1.925	2.824	.0313

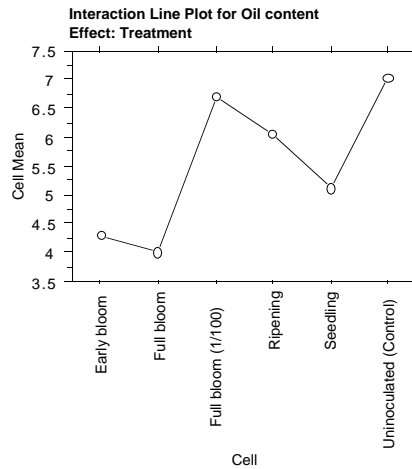
Comparisons in this table are not significant unless the corresponding p-value is less than .0033.

Bonferroni/Dunn tends to fall between the two, as evidenced in this case by the single significant pair. Notice that StatView warns that only comparisons with p values less than 0.0033 are significant for the alpha value (.05) specified. The Tukey-Kramer, Dunnnett, Games-Howell, and Student-Newman-Keuls test are similar.

We can examine these results graphically with an interaction line plot or bar chart. The relative oil content levels are of greater interest to us than their actual values (recall that the researchers subtracted 30 from each measurement when recording the data, so the measurements are already somewhat abstract). StatView chooses a vertical (Y axis) scale for line plots to suit the range of the data for line charts, whereas it prefers a vertical scale from 0 to the data maximum (when practical) for bar charts. Therefore, we will choose a line plot.

- Make sure at least one of the results is still selected

- In the analysis browser under ANOVA, select Interaction Line Plot and click OK



Again, we disregard the result for Location. The wide difference in oil content between full bloom inoculation and no inoculation illustrates the single significant comparison in the Bonferroni/Dunn results. The researchers can safely conclude that inoculation, particularly at the time of full bloom, decreases oil content.

Latin square ANOVA

To determine whether the moisture content of turnip green leaves is affected by time in storage, researchers classified the leaves of five turnip plants into five size groups, subjected these leaves to one of five lengths of storage time according to a specific pattern, and finally measured the moisture content of each leaf.

Since it is reasonable to suspect that the moisture content might vary from plant to plant, several different plants were sampled in a Latin square design. Like the randomized complete block design, Latin square experimental designs include factors that are intended solely to reduce variability and give analysis results validity over a wider range of samples. Another approach to this experiment would be to think of the plants as replicates within a leaf size/time of storage classification and to analyze the model as a two-way factorial design. The disadvantage of such an approach is that it would not account for any differences between plants—they would be interpreted as part of the residual error, possibly making the analysis insensitive to true differences.

Latin square designs must be applied with caution, because not every possible combination of factor levels is observed; our example has a single observation for each plant/leaf size combination, which means that numerous possible plant/leaf size/storage time combinations are *not* observed. Therefore, the analysis is invalid in the presence of any interaction, even if the interaction has no practical consequences. For example, if storage time dramatically affects moisture content for a particular leaf size, then the Latin square analysis would not be valid.

The key feature of the Latin square design is that each treatment appears exactly once for each combination of two blocking factors. This balance, combined with the lack of interactions, is required for the analysis to be valid. Thus, a Latin square is appropriate only when it is possible to create two blocking factors for your data, each of which has the same number of levels as the number of treatments in the experiment. In our examples, the researchers studied five storage times, so they used five leaf types from five individual plants.

Latin squares are useful for agricultural experiments that must control for variability in fertility of a field, when it is convenient to divide the field into a number of rows and columns matching the number of treatments. Although one could work out by hand the treatment combinations needed, researchers usually refer to published tables of these designs, such as those in Cochran and Cox (1957). Not all numbers of treatments can be accommodated by Latin squares designs, so it is wise to consult a reference early in the planning stage of an experiment.

- Open Turnip Moisture from the Sample Data folder

	Storage Time	Plant	Leaf Size	Moisture Content
1	V	1	A	6.67
2	IV	1	B	7.15
3	I	1	C	8.29
4	III	1	D	8.95
5	II	1	E	9.62
6	II	2	A	5.40
7	V	2	B	4.77

Plant and Leaf Size are the blocking factors, Time of Storage is the treatment factor, and Moisture Content is the dependent measurement variable. Observe how each storage time is represented once in each plant and once in each leaf size. The design is more apparent when arranged in a compact variable. (Unfortunately “compact” arrangement is considerably less convenient to enter in the dataset in this case; the way it reveals the data design is its value.)

- Open Turnip Moisture Compact from the Sample Data folder

	Storage	Moisture Content																			
		P1				P2				P3				P4				P5			
		A	B	C	D	E	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
1	I	.	.	8.29	7.54	.	.	.	9.68	4.92	.	.	.	6.16	.	.	.
2	II	9.62	5.40	.	.	.	8.53	.	.	.	7.29	.	.	.	5.83	.	.
3	III	.	.	.	8.95	.	.	.	6.93	7.32	.	.	.	5.00	.	.	.	7.83	.	.	.
4	IV	.	7.15	5.40	.	.	.	9.99	.	.	.	7.08	4.88
5	V	6.67	4.77	.	.	.	8.50	.	.	.	7.85	8.51	.

According to the assumptions of the Latin square design, we must include only main effects in the model. If we attempted to include interaction effects, error messages about matrix singularity would quickly alert us to our mistake, since so many cells have few if any data points.

- From the Analyze menu, select New View
- In the analysis browser under ANOVA, select ANOVA Table and click Create Analysis
- Choose 1 for Include all interactions up to depth
- Click OK

We will demonstrate variable assignments with both versions of the dataset; you may choose either one or repeat the steps above to try both methods.

- In the variable browser, choose Turnip Moisture for Dataset
- Select Moisture Content and click Dependent
- Select Plant, Leaf Size, and Storage Time and click Independent

Using the compact version of the dataset works similarly. The only tricky thing is that here it is important to assign Moisture as a Dependent before attempting to assign its nominal parts as Independents: StatView does not allow you to assign the nominal part of a compact variable to an ANOVA unless you have already assigned its continuous part to the analysis.

- In the variable browser, choose Turnip Moisture Compact for Dataset
- Click the triangle to the left of Moisture Content to expose the nominal parts of the compact variable
- Select Moisture Content and click Dependent
- Select Storage Time and the nominal parts of the compact variable, Plant and Leaf Size, and click Independent

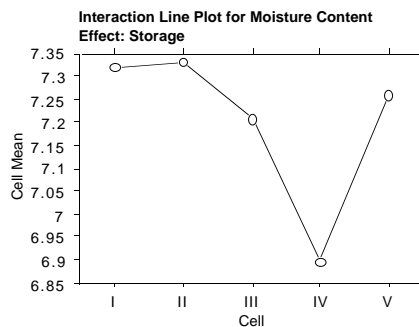
The results from either dataset are the same:

ANOVA Table for Moisture Content

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Storage	4	.627	.157	.233	.9147	.931	.085
Plant	4	28.885	7.221	10.714	.0006	42.857	.996
Leaf Size	4	23.708	5.927	8.794	.0015	35.176	.986
Residual	12	8.088	.674

The significant F ratios for Plant and Leaf Size (10.714, with p value 0.0006 and 8.794 with p value 0.0015, respectively) indicate that these factors served their purpose, removing variability from the analysis. Otherwise they are of no great interest. However, we cannot reject the null hypothesis that storage time does not affect moisture content, because the F ratio 0.233 is so low, with p value 0.9147. In other words, the researchers have not found evidence to support their theory that storage time affects moisture content. A glance at interaction line plots will explain this:

- Make sure the ANOVA Table is still selected in the view
- From the analysis browser under ANOVA, double-click Interaction Line Plot



Observe that, while moisture content does seem to drop off as storage times increase to treatment IV, it jumps back up again for the *longest* storage time, treatment V. This makes no sense. Also note that the least and greatest cell means are less than half a percent apart. What little difference we see is probably random.

Factorial MANOVA design

Suppose you are an exercise physiologist who wants to determine whether stretching and wearing ankle weights has any effect on the value of treadmill exercise. You could test this hypothesis by measuring calories burned, average speed in meters per minute, and oxygen consumed in liters for a number of subjects who you have previously determined have roughly the same level of physical fitness, divided randomly into four groups: with or without ankle weights, and with or without a period of stretching before the exercise. This would be a 2×2 factorial design.

Suppose you tested twenty subjects and recorded measurements such as the following:

- Open Exercise from the Sample Data folder

	Pre-stretch	Ankle weights	Energy (cal)	Speed (m/min)	Oxygen (l)
1	No	No	106.9	87.8	34.3
2	No	No	84.0	92.9	25.4
3	No	No	97.5	85.3	29.2
4	No	No	97.1	82.4	31.7
5	No	No	99.5	82.4	29.5
6	No	Yes	100.2	83.9	36.3
7	No	Yes	101.0	85.3	44.0
8	No	Yes	118.5	85.4	47.3
9	No	Yes	104.5	79.6	44.3
10	No	Yes	111.2	85.2	44.7
11	Yes	No	82.8	82.7	26.8
12	Yes	No	80.4	89.0	20.2
13	Yes	No	95.6	87.5	33.8
14	Yes	No	82.0	78.3	18.0
15	Yes	No	83.2	89.0	28.6
16	Yes	Yes	89.1	86.7	28.3
17	Yes	Yes	106.4	80.5	38.2
18	Yes	Yes	98.3	79.6	36.7
19	Yes	Yes	89.2	82.3	29.9
20	Yes	Yes	104.6	87.6	43.8

The goal of this experiment is to determine whether pre-stretching and wearing ankle weights change the outcome measurements for the exercise. One interesting point is that we know that the null hypothesis that wearing ankle weights has no effect is almost certainly false. However, we don't know whether the effects of pre-stretching, if any, are the same whether or not the ankle weights are worn. Thus, the ankle weights serve to some extent as a blocking factor in the experiment, even though it is a complete factorial design.

If we were only interested in one of the measurements of exercise value, such as calories burned, we could simply analyze the data with a two-way factorial ANOVA design. However, we want to know whether or not the factors affect three measurements (energy, velocity and oxygen consumption) simultaneously. This is why we should take correlations among dependent variables into account by examining StatView's MANOVA tables of results for multivariate hypothesis tests.

- From the Analyze menu, select New View
- In the analysis browser under ANOVA, select ANOVA Table and MANOVA Tables and click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select multiple nonadjacent analyses.
- Click OK to accept the default analysis parameters

(We leave the default interaction depth, Full, because we must test the interaction of weights and pre-stretching—we don't yet know whether the effect of stretching will be the same regardless of whether weights are worn.)

- In the variable browser, select Pre-stretch and Ankle Weights and click Independent Shift-click or click and drag to select multiple adjacent variables
- Select Energy (cal), Speed (m/min), and Oxygen (l) and click Dependent Shift-click or click and drag to select multiple adjacent variables

The interaction Pre-stretch * Ankle Weights does not appear to be significant: the p value for the term is nowhere close to the alpha level of 5% in any of the ANOVA or MANOVA tables. This means that the effects of ankle weights during are the same with or without pre-stretching. Another way to view it is that pre-stretching has the same effect with or without weights. Therefore, we can remove the term from the model and examine only main effects.

- Make sure at least one of the tables is still selected
- Click Edit Analysis
- Choose 1 for Include all interactions up to depth and click OK

MANOVA Table for Pre-stretch

	Value	F-Value	Num DF	Den DF	P-Value
S	1.000
M	.500
N	6.500
Wilks' Lambda	.602	3.309	3	15	.0491
Roy's Greatest Root	.662	3.309	3	15	.0491
Hotelling-Lawley Trace	.662	3.309	3	15	.0491
Pillai Trace	.398	3.309	3	15	.0491

The MANOVA results for Pre-stretch show a p value of 0.0491 for the three measurements, so at the 5% significance level, we reject the null hypothesis that stretching has no effect.

MANOVA Table for Ankle Weights

	Value	F-Value	Num DF	Den DF	P-Value
S	1.000
M	.500
N	6.500
Wilks' Lambda	.358	8.963	3	15	.0012
Roy's Greatest Root	1.793	8.963	3	15	.0012
Hotelling-Lawley Trace	1.793	8.963	3	15	.0012
Pillai Trace	.642	8.963	3	15	.0012

The MANOVA results for Ankle Weights are even more clear. The p value of 0.0012 means that we should definitely reject the null hypothesis that ankle weights have no effect. This is no surprise—we expected that ankle weights would have a significant effect.

ANOVA Table for Energy (cal)

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Pre-stretch	1	591.872	591.872	10.696	.0045	10.696	.884
Ankle Weights	1	649.800	649.800	11.743	.0032	11.743	.913
Residual	17	940.688	55.335

ANOVA Table for Speed (m/min)

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Pre-stretch	1	2.450	2.450	.172	.6836	.172	.067
Ankle Weights	1	22.472	22.472	1.577	.2261	1.577	.209
Residual	17	242.200	14.247

ANOVA Table for Oxygen (l)

	DF	Sum of Squares	Mean Square	F-Value	P-Value	Lambda	Power
Pre-stretch	1	194.688	194.688	7.329	.0149	7.329	.729
Ankle Weights	1	672.800	672.800	25.327	.0001	25.327	.999
Residual	17	451.602	26.565

From the univariate tests, we can see that pre-stretching has a significant effect on energy and oxygen consumption. Likewise, ankle weights have a significant effect on energy and oxygen consumption. However, neither seem to have much effect on speed, which is a bit surprising. Now that we know both factors are significant, we want to know *how* they effected the outcome measurements.

- Make sure at least one of the results is still selected
- In the analysis browser under ANOVA, double-click Means Tables or one of the interaction plots

Means Table for Energy (cal)

Effect: Ankle Weights

	Count	Mean	Std. Dev.	Std. Err.
No weights	10	90.900	9.405	2.974
Weights	10	102.300	9.046	2.861

Means Table for Speed (m/min)

Effect: Ankle Weights

	Count	Mean	Std. Dev.	Std. Err.
No weights	10	85.730	4.307	1.362
Weights	10	83.610	2.938	.929

Means Table for Oxygen (l)

Effect: Ankle Weights

	Count	Mean	Std. Dev.	Std. Err.
No weights	10	27.750	5.370	1.698
Weights	10	39.350	6.556	2.073

Clearly, wearing weights increases oxygen consumption (27.75 vs. 39.35 liters) and energy consumption (90.9 vs. 102.3 calories), and also decreases speed (85.7 vs. 83.6). The tables for pre-stretching show that whether subjects stretch or not also has a slight effect on the outcome measurements.

Contingency Tables

Contingency table analyses determine whether a relationship exists between two nominal variables. Other statistics (t -tests, regressions, means, correlation tests) apply to dependent variables that are continuous, that is, they are capable of taking on many different values with an obvious ordering to them like height, weight, income, chemical concentration, sales, etc. Tests applied to continuous variables lose their validity with nominal variables that do not have an ordered, continuous property. (See [“Dataset structure,” p. 49 of *Using StatView*](#) for a discussion of nominal and continuous variable classes.)

Height is a continuous variable because an underlying meaning to the ordering of values applies to it—sixty inches is clearly bigger than fifty inches—and this relationship holds through the range of the scale. But hair color and eye color, for example, cannot constitute continuous variables, for there is no natural ordering to brunette, blonde, red and black; nor to blue, gray, brown and green.

Thus, even if we recode a variable representing hair color as brunette=1, black=2, red=3 and so forth, any tests performed on the transformed variable would be pointless. (It is possible for a nominal variable to be ordered, but StatView provides no special tests for this case.) For example, it is meaningless to say that brunette is only one third of red. In addition, if we study the relationship of hair color and eye color, we cannot calculate a mean for hair color because there is no numerical quantity we can assign to a particular hair color that helps describe it.

Discussion

When you collect data, it may be wise to think in terms of a two-way tabular arrangement in which you categorize each observation into one group for each of two nominal (grouping) variables. Such an arrangement is called a **contingency table**. The intersection of a row and column in the table is called a cell. If you study the cross-classification of eye color and hair color, for example, each cell would contain a count of observations for each possible combination of hair and eye color groups: blue eyes/brown hair, brown eyes/brown hair, blue eyes/blonde hair, brown eyes/blonde hair and so forth. It could look something like this:

	Brown hair	Blonde hair	Black hair	Red hair
Brown eyes	21	10	7	2
Blue eyes	9	17	2	3
Green eyes	1	3	1	3

Chi-square test

It may be of interest to study this contingency table to see which combinations of groups have more or less observations than would be expected if the two variables were independent. For this you can apply the **chi-square test** for independence. The hypothesis of independence states that the likelihood of an observation falling into one group for one variable is independent of the other group the observation falls into. To calculate this test, StatView finds the expected value for the number of observations for every combination of groups based on the hypothesis of independence and compares the expected with the observed values in each cell.

(The chi-square test is not valid when the minimum expected value is less than five. You may have cells in your contingency tables with observed values less than five without causing any problems. The key issue is whether or not the expected values are greater than five. You can print a table of expected values for your contingency table.)

A low chi-square value and high probability (p value) would suggest accepting the null hypothesis. If the hypothesis of independence were not rejected for the example given, the chi-square test would indicate that people with blonde hair are no more likely to have blue eyes than any other color eyes, and that people with brown eyes are no more likely to have brown hair than any other color hair. If rejected—a large chi-square value and correspondingly low probability—the test would show that a relationship between certain variable groups exists. You would then study the contingency table to see which combinations of groups have more or fewer observations than would be expected if the two variables were independent. You can do this by comparing the contingency table (observed frequencies) to the expected values table, or by examining a table of post hoc cell contributions to the overall chi-square statistic.

Tables produced

In addition to the contingency table itself, StatView offers a variety of displays with the groups of one variable in the cross classification displayed in the rows of a table and the groups of the other displayed in the columns of the table. One set of tables displays the percents of row or column totals. In a table displaying the Percents of Row Totals, for example, column percentages represent the proportion of data in the first variable that falls into each group of the second variable. Under the hypothesis of independence, the column percentages within each group of the first variable (each row of the table) should be the same. You can compare the values in a given row with the totals displayed at the bottom of the table and determine which cells are out of line. The cells that stand out indicate a larger or smaller proportion falling in a particular combination of groups than would be expected under the hypothesis of independence. A similar analysis holds for the Percents of Column Totals table, except that you compare the values in the rows with the totals on the right hand side of the table.

Post hoc cell contributions

An alternative to studying percents is to study the table of **post hoc cell contributions**. These numbers are a form of standardized residual that indicate what each cell in the table contributes to the chi-square statistic. Since they are calculated to follow a standard normal distribu-

tion, absolute values greater than, for example, 1.96 for a 0.05 probability level indicate that the cell in question provides significant information about the combinations of groups of the variables whose occurrence is different than would be expected under the hypothesis of independence. An example of the use of post hoc cell contributions is given in the [“Exercise,” p. 117.](#)

Cell chi-squares

The chi-square statistic reported in the summary table is the sum of the values in the cell chi-squares table. By examining this table, you can tell which cells have observed frequencies that differ most from what is expected under the hypothesis of independence. This is the same information obtained from the post hoc cell contributions, except that the cell chi-squares are compared to the total chi-square whereas post hoc cell contributions are compared to the normal distribution.

Additional statistics: *G*-statistic and Cramer’s *V*

An alternative statistic for testing the hypothesis of independence between two categorical variables is the *G*-statistic. The *G*-statistic is derived using a statistical principle known as the likelihood ratio principle.

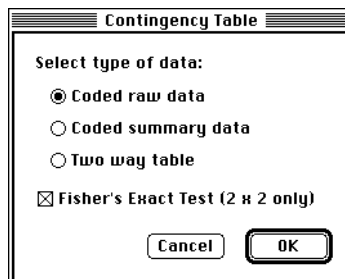
Another statistic, the contingency coefficient, is offered by analogy to the correlation coefficient, which is used to test the association between two continuous variables. An attractive feature of the correlation coefficient is that it is always in the range of -1 to 1 , so that several different relationships can be compared on an equivalent scale. The **contingency coefficient** is a transformation of the chi-square statistic so that the contingency coefficient is in the range of 0 and 1 . Thus it can be useful for comparing associations between different pairs of variables. Closely related to the contingency coefficient, and testing the same hypothesis of no association between variables, is **Cramer’s *V*** (pronounced kruh-merz’). High values of these statistics indicate that there is dependence between the variables. The range of *V* is from 0 to 1 , so its interpretation is more in line with that of a correlation coefficient.

2x2 contingency tables: Fisher’s exact test, Phi coefficient

Other statistics are available in the summary table for the special case of 2×2 tables (in which both variables studied have exactly two groups). **Fisher’s exact test** is calculated by enumerating all possible rearrangements of the observations and comparing the number of unusual rearrangements to the observed counts under the assumption of no association between the two variables. The probability levels reported for this test are exact, not large sample approximations like the *G*-statistic and chi-square described earlier. The continuity correction for a 2×2 table, and its associated *p* value is an alternative technique which is used to make the probability level of the 2×2 test for independence closer to the exact probability level. The **phi coefficient** is similar to the contingency coefficient in that it is bounded in the range from 0 to 1 ; it is the same as Cramer’s *V* except in the special case of 2×2 tables. Its interpretation is similar to that of the correlation coefficient and may be especially useful if the categories for each of the variables have a natural ordering.

Dialog box settings

When you create or edit a contingency table, you set the analysis parameters in this dialog box:



You use this dialog box to specify how your data is organized for the contingency table analysis. Please see the preceding section, [“Data requirements,” p. 114](#), for more information and for examples of these types of data. This dialog box also allows you to disable computation of Fisher’s Exact Test (available only for 2×2 data). This option is provided so that you can avoid the lengthy computation required for large datasets.

Data requirements

Variable requirements differ depending on the type of data being analyzed.

1. Coded raw data requires two nominal variables.
2. Coded summary data requires two nominal variables and one continuous variable.
3. A two-way table requires two or more continuous variables. In the cases where continuous variables are required, those variables represent counts based on the levels of the nominal variables in your analysis.

The discussions below describe how to enter data each way for a study to determine whether a relationship exists between eye color and gender for eight athletes.

Coded raw data

Coded raw data for this example would contain two nominal columns: one indicating the eye color and the other the gender for each athlete. The dataset would contain eight rows, one for each athlete. A dataset organized in this manner would appear as:

	Eye Color	Gender
1	Brown	Male
2	Blue	Male
3	Blue	Female
4	Green	Male
5	Brown	Female
6	Blue	Male
7	Green	Female
8	Brown	Male

The nominal variables appear as separate columns in the dataset. Each row identifies the eye color group and the gender group for an athlete.

Coded summary data

Coded summary data for this example would contain two nominal grouping variables in columns and an additional column with the count in each combination of groups (cell). A dataset organized in this manner would appear as:

	Eye Color	Gender	Count
1	Blue	Female	1
2	Blue	Male	2
3	Brown	Female	1
4	Brown	Male	2
5	Green	Female	1
6	Green	Male	1

The dataset contains six rows, one for each possible combination of eye color and gender: blue eyes/female, blue eyes/male, brown eyes/female, and so on. Each combination is made up of entries in the nominal Eye Color and Gender columns. The count for each combination appears in the count column.

You are not required to have as many rows as there are combinations. If duplicate combinations appear in your data, StatView will sum the counts for that combination. Also, if a fractional value appears in a count column, the value will be rounded to the nearest integer.

Two-way table

To use a two-way table, you enter a contingency table of observed values directly into a dataset as input for the analysis. Each column is a column of the contingency table and each row a row of the table. The observed frequencies are entered as individual observations. There will be as many columns as groups in one nominal variable and as many rows as groups in the second nominal variable. A dataset organized in this manner would appear as:

	Male	Female
1	2	1
2	2	1
3	1	1

The two columns represent the two gender groups: male and female. The three rows the three eye color groups: blue, brown and green. The values in each cell are the counts for the particular combination.

Variable browser buttons	
Add	For coded raw data, select two nominal variables and click Add. For coded summary data, select two nominal variables and one continuous variable and click Add. For a two-way table, select two or more continuous variables, and click Add. For raw data, each additional nominal variable assigned creates a new analysis. For coded summary data, each additional nominal or continuous variable assigned creates a new analysis. For a two-way table, each additional variable you assign is added to the existing analysis.
Split By	When you assign one or more split-by variables to a contingency table result, results for each cell in the split-by variable(s) are displayed in separate tables.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 111](#). The Summary and Observed Frequencies tables are the default output for this analysis.

Summary Table	Table containing the degrees of freedom, the chi-square statistic and associated <i>p</i> value, the <i>G</i> -squared statistic and its associated <i>p</i> value, the contingency coefficient, and Cramer's <i>V</i> for the analysis. If 2x2 data are used, the Fisher's exact test, the continuity correction with its associated <i>p</i> value are displayed, and the phi coefficient is displayed instead of Cramer's <i>V</i> .
Observed Frequencies	Table containing the number of observations in each cell (combination of groups) of the dataset with totals for each group in the grouping variables.
Percents of Row/Column Totals	Table containing the percentage of the observations in each group of one grouping variable that fall into each group of the second grouping variable.
Percents of Overall Total	Table containing the percent of total observations in the dataset that falls in each cell (combination of groups).
Expected Values	Table containing the expected values for the number of observations in each cell (combination of groups) if the variables were independent.
Post Hoc Cell Contributions	Table containing the post hoc cell contributions for each cell (combination of groups).
Cell Chi Squares	Table containing the chi-squares statistic for each cell (combination of groups).

Templates

The following templates provide contingency table results.

Correlations	Contingency Table--2 Way Data	Summary and observed frequencies tables for two-way data.
--------------	-------------------------------	---

	Contingency Table--Raw Data	Summary and observed frequencies tables for raw data.
	Contingency Table--Summary Data	Summary and observed frequencies tables for summary data.

Exercise

In this exercise you will perform a contingency table analysis of coded raw data. The dataset contains information on weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries. You will determine whether some countries tend to produce larger or smaller cars than other countries.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Contingency Table, select Summary Table and Observed Frequencies and click Create Analysis
- Click OK to accept the default parameter, Coded raw data
- In the variable browser, select Type and click Add, then select Country and click Add (It is important to add the variables in this order.)

Note that the groups of the first variable appear as rows of the contingency table; the groups of the second variable appear as columns. The variables are highlighted with G usage markers indicating grouping variables assigned to the analysis. The analysis calculates and tables appear in the view.

Summary Table for Type, Country

Num. Missing	0
DF	8
Chi Square	25.814
Chi Square P-Value	.0011
G-Squared	27.861
G-Squared P-Value	.0005
Contingency Coef.	.427
Cramer's V	.334

Observed Frequencies for Type, Country

	Japan	Other	USA	Totals
Small	7	12	3	22
Sporty	10	4	11	25
Compact	3	12	7	22
Medium	6	8	16	30
Large	4	1	12	17
Totals	30	37	49	116

The high chi-square and low p values in the summary table suggest a relationship between country and car size. You will now determine which cells are contributing to the large chi-square values by examining post-hoc cell contributions.

- Make sure at least one table is still selected
- In the analysis browser, select Post Hoc Cell Contributions and click Create Analysis

Post Hoc Cell Contributions for Type, Country

	Japan	Other	USA
Small	.709	2.532	-3.017
Sporty	1.823	-1.925	.201
Compact	-1.455	2.532	-1.100
Medium	-.852	-.714	1.428
Large	-.238	-2.491	2.561

You did not have to assign variables to the Post Hoc Cell Contributions table. The variables analyzed in the tables preceding it were used because those tables were selected when you created Post Hoc Cell Contributions.

Relative to what is expected if the distribution of car sizes were the same for each country, the Other group has more small cars than Japan, and more still than the USA. The USA, however, has many more cars categorized in the Large group. You may want to examine the table of expected values to verify that the discrepancies arise from the cells with large post hoc cell contributions.

Nonparametrics

Nonparametric statistics test hypotheses about data for which the underlying distribution of the data is not assumed. Rather than estimate the parameters of a hypothesized distribution, then perform a computation on these estimates (parametric statistics), nonparametrics employ alternatives such as sequentially ranking observations from all groups or variables of interest or comparing two groups observation by observation to test hypotheses.

Discussion

Most of the hypothesis tests presented in other chapters require the data being studied to fulfill certain assumptions, usually regarding the nature of the underlying distribution from which the data arises. In order for the probability levels presented by a t -test to be valid, for example, the data being studied must come from a normal distribution. These assumptions are so important that many statisticians feel that a significant probability value associated with a test statistic needs to be interpreted as either evidence that the null hypothesis is false or evidence that the assumptions of the test have been violated.

Occasionally the assumptions required for a parametric test are not met because of the nature of the data. If you are measuring the amount of time it takes people to do a simple task, you might know that most responses will be around zero, with fewer and fewer responses corresponding to increasing time. This would not result in a normal distribution of data since normal distribution must be symmetric, with equal amounts of data on either side of the mean. In other cases, your examination of the data (or residuals from regression or analysis of variance) might indicate that the assumptions of the analysis are not being met. Under such circumstances, performing one of the nonparametric tests described in this chapter can be appropriate.

One sample sign test

The **one sample sign test** is the nonparametric equivalent of the one sample t -test. It tests whether the values of a variable are centered around a specified value. That is, it tests the hypothesis that the median of a distribution is equal to some hypothesized value by comparing the number of observations above and below that value.

Mann-Whitney U test

The **Mann-Whitney U test** is useful in the same cases as an unpaired t -test. It is the nonparametric version of the two group unpaired t -test. Recall that a t -test tests the hypothesis that the means of the two groups are equal, assuming normality of the observations. The Mann-Whitney U tests the hypothesis that the distributions underlying the two groups are the same. The requirements for validity of the Mann-Whitney test are that the two groups of observations come from continuous distributions and are independent of each other, both within and between groups. Since the Mann-Whitney test does not look at the observations but instead considers their ranks, it is resistant to outliers in either of the groups being compared.

Kolmogorov-Smirnov test

The **Kolmogorov-Smirnov test** tests whether the distribution of a continuous variable is the same for two groups. That is, it tests the null hypothesis that two distributions are the same under the assumption that the observations from the two distributions are independent of each other. It is calculated by comparing the two distributions at a number of points and then considering the maximum difference between the two distributions. (The actual data points are not compared, but a function of the points is calculated and compared.) Since this test relies on the maximum value in a set of numbers, it may be heavily influenced by outliers and should be used with caution if outliers are suspected.

Wald-Wolfowitz runs test

The **Wald-Wolfowitz runs test** tests whether the two groups of observations have been randomly sampled from the same population. This test compares two groups assumed to be independent of each other by combining the data for both groups, ranking the data and counting the number of runs present in the ranked data. A run is a sequence of consecutive observations from one or the other of the groups. (Only the number of runs is important, not their lengths.) If the two samples come from different distributions, we would expect many groups of small runs, while if observations from one group tend to be larger than those from the other group, we would see only a few runs in the data. Since the test is based on ranks, it is resistant to outliers.

The Wald-Wolfowitz test looks at the data across the entire range, whereas the Kolmogorov-Smirnov test looks at the maximum difference between the distributions. If there are only one or two outliers, the Kolmogorov-Smirnov may mistakenly state that the two distributions are different.

Wilcoxon signed rank test

The **Wilcoxon signed rank test** is appropriate in the same cases that a paired t -test would be used; it is the nonparametric version of the paired t -test (see [“Paired Comparisons,” p. 29](#)). It is based on the rank of the differences between each pair of observations in the dataset, and

tests the hypothesis that sum of the ranks is equal to zero under the assumption that the distribution of ranks is symmetric about 0.

Paired sign test

The **paired sign test**, or two sample sign test, is useful in the same situations that a paired t -test is used. It is another nonparametric version of the paired t -test. It tests the hypothesis that one of the paired variables is just as likely to be greater than the other variable as it is to be less than the other variable, without regard for the magnitude of the difference. Thus, it makes very few assumptions about the underlying distributions from which the data arise. If you feel that the differences between the two paired variables you are studying will be symmetric around some value, the Wilcoxon signed rank test is more powerful.

Spearman rank correlation coefficient

The **Spearman rank correlation** coefficient, sometimes referred to as **Spearman's rho**, is an alternative to the usual correlation coefficient. Since it is based on the ranks of the data and not the data itself, it is resistant to outliers. It calculates a correlation coefficient based on the ranks of the values of two variables. The null hypothesis tested by Spearman's rho is that the two variables are independent of each other, against an alternative hypothesis that the rank of a variable is correlated with the rank of another variable. Spearman's rho ranges in value from -1 (indicating high ranks of one variable occur with low ranks of the other variable) through 0 (indicating no correlation between the variables) to $+1$ (indicating high ranks of one variable occur with high ranks of the other variable).

Kendall's rank correlation coefficient

Kendall's tau is an alternative to Spearman's rho and is useful in the same situations as Spearman's rho. In general, the interpretation of these two statistics results in similar conclusions about the data. Kendall's tau also ranges from -1 through 0 to $+1$.

Kruskal-Wallis test

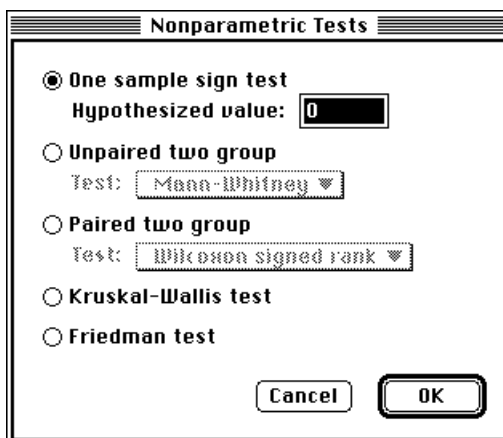
The **Kruskal-Wallis test** is a nonparametric equivalent of a one-way analysis of variance by ranks, i.e., it tests the null hypothesis that three or more groups all come from the same distribution. It is basically calculated as a regular ANOVA, but uses the ranks of the data and is therefore resistant to outliers. Along with the test statistic, StatView displays a table including the mean rank for each of the groups to aid you in determining which group tends to have larger values than the others.

Friedman test

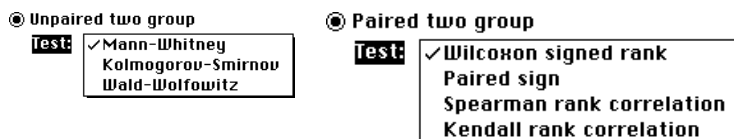
The **Friedman test** is a two-way analysis of variance by ranks for matched samples. It is a special case of a nonparametric two-way ANOVA in which, for each of several groups (usually called blocks), there are a number of observations, each representing the response for that group to a particular treatment. It tests the hypothesis that the effects of the treatments are the same against the hypothesis that at least one of the treatments has an effect different from the others. Like most of the other nonparametric tests, it is based on ranks and is therefore resistant to outliers.

Dialog box settings

When you create or edit nonparametric results, you set the analysis parameters in this dialog box:



There are ten nonparametric tests to choose from in this dialog box. There are no further parameters for any of these tests except the One Sample Sign Test, for which you specify the hypothesized value around which you believe the values are centered. For paired and unpaired two groups, you must choose a test:



If you are editing nonparametric results by selecting a result and clicking Edit Analysis, you will not always be able to switch from one particular test to another. For example, you will not be able to switch to an unpaired two group test from a paired two group test if you have specified variables which the unpaired test cannot use (i.e., a second continuous variable).

Data requirements

The nonparametric statistics are divided into five groups. Each group requires a different data organization as described below. For an introduction to dataset organization, see [“Dataset structure,” p. 49 of Using StatView.](#)

Data for Mann-Whitney, Kolmogorov-Smirnov, and Wald-Wolfowitz tests must be organized in the same manner as for unpaired comparisons analysis. Please see [“Data requirements,” p. 39,](#) for a complete discussion of the required data organization. In addition, there are exercises for both the Mann-Whitney *U* and Kolmogorov-Smirnov tests (see [“Exercises,” p. 125\).](#) Data for the Kruskal-Wallis test must be organized in the same manner as for factorial analysis of variance experiments. Please see [“Data requirements,” p. 90,](#) for a complete discussion of the required data organization. In addition, see the exercise [“Kruskal-Wallis test,” p. 128.](#)

Data for the Friedman test must be entered so that each column contains information on a single sample (or treatment). Each row contains the response of a particular group for the treatment. The dataset will contains as many columns as there are different samples (or treatments) and as many rows as there are responses for the treatment. See the exercise, [“Friedman test,” p. 128.](#)

Nonparametric test	Requirements	Additional variables
One Sample Sign	one continuous variable	Each additional variable creates a new analysis.
Mann-Whitney <i>U</i> , Kolmogorov-Smirnov, Wald-Wolfowitz Runs	one continuous variable and one nominal variable with two levels	Each additional nominal and/or continuous variable creates a new analysis for each nominal/continuous pair.
Wilcoxon Signed Rank, Paired Sign, Spearman Rank Correlation, Kendall Rank Correlation	two continuous variables	Each additional continuous variable creates a new analysis for each pair.
Kruskal-Wallis	one nominal variable with more than two levels and one continuous variable	Each additional nominal and/or continuous variable creates a new analysis for each nominal/continuous pair.
Friedman	three or more continuous variables	Each additional variable is added to the existing analysis.

Variable browser buttons

Add	To generate nonparametric statistics, select the variable(s) that you wish to analyze and click Add.
Split By	When you assign one or more split-by variable to a nonparametric analysis, results for each cell in the split-by variable(s) are displayed in separate tables or plots.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 119.](#)

One Sample sign test	Table containing the number of observations above, below and equal to the hypothesized value, the p value for the analysis.
Mann-Whitney U test	Table containing the U and U prime statistics, tied and untied z values and p values, and the number of ties. Table containing the count, sum and mean of the rankings for each group in the analysis.
Kolmogorov-Smirnov test	Table containing the degrees of freedom, the number of observations in each group, the maximum difference between groups, and the chi-square statistic and p value for the analysis.
Wald-Wolfowitz Runs test	Table containing the number of runs in the combined groups, the number of observations in each group, the mean and standard deviation used in the z value, and the z value and the p value for the difference between groups.
Wilcoxon Signed Rank test	Table containing the number of differences between pairs, and tied and untied z values and p values, and the number of ties. Table containing the count, sum and mean of the rankings for each group in the analysis.
Paired Sign test	Table containing the number of differences above, below and equal to 0 and the p value for the analysis.
Spearman Rank Correlation	Table containing the sum of squared differences and the Rho (with and without correction for ties) for the groups, the tied and untied z values and p values, and the number of ties in each group.
Kendall Rank Correlation	Table containing the sum of squared differences and the tau (with and without correction for ties) for the groups, the tied and untied z values and p values, and the number of ties in each group.
Kruskal-Wallis test	Table containing the degrees of freedom, number of groups and ties, and the H and p values, with and without correction for ties. Table containing the count, sum and mean of the rankings for each group in the analysis.
Friedman test	Table containing the degrees of freedom, number of groups and ties, and the chi-square and p value, with and without correction for ties. Table containing the count, sum and mean of the rankings for each group in the analysis.

Note that some of the tests above show a correction for ties. Ties occur when two observations have the same value. Nonparametric tests assume that no two values are the same. In some tests, StatView is able to make a correction for the presence of ties; where it cannot, a warning message is produced if ties are present.

Templates

The following templates provide nonparametric results.

Nonparametrics	Friedman	Friedman test and rank info tables.
	Kendall Correlation	Kendall correlation test table.
	Kolmogorov Smirnov	Kolmogorov Smirnov test table.

Kruskal Wallis	Kruskal Wallis test and rank info tables.
Mann Whitney	Mann Whitney U test and rank info tables.
One Sample Sign Test	One sample sign test table.
Paired Sign	Paired sign test table.
Spearman Correlation	Spearman correlation table.
Wald Wolfowitz	Wald Wolfowitz runs test table.
Wilcoxon Signed Rank	Wilcoxon signed rank test and rank info tables.

Exercises

One sample sign test

In this exercise you will perform a one sample sign test using data from blood lipid screenings of medical students. You are concerned with one variable here: Cholesterol. You will find out if the cholesterol level of the students differs significantly from 190, a point above which cholesterol levels may be unhealthy. You will test the null hypothesis that the value for cholesterol is 190. If you reject the null hypothesis, you can conclude that student cholesterol levels differ significantly from 190.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Enter “190” as hypothesized value and click OK to accept the other settings
- In the variable browser, select Cholesterol and click Add

The variable name appears highlighted with an X usage marker next to it indicating you have assigned a continuous variable to the analysis. The analysis calculates and this table appears in the view.

One-Sample Sign Test for Cholesterol	
Hypothesized Value: 190	
# Obs. > Hyp. Value	48
# Obs. < Hyp. Value	43
# Obs. = Hyp. Value	4
P-Value	.6752

You cannot reject the null hypothesis. The p value is large, and there are roughly the same number of observations above and below the hypothesized value of 190.

Mann-Whitney U test

In this exercise you perform a Mann-Whitney U test using census data for 506 housing tracts in the Boston area. You will examine two groups of housing tracts, those near the Charles River and those farther away from it. You will find out whether median housing prices vary

depending on how far houses are located from the river. This is the nonparametric equivalent of the unpaired t -test exercise ([“Exercise,” p. 41](#)). You may wish to compare results between the two tests.

- Open Boston Housing Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Choose Unpaired two group and click OK
(Leave Mann-Whitney selected for the test)
- In the variable browser, select Median Value and click Add
- In the variable browser, select Charles and click Add

Mann-Whitney U for Median Value
Grouping Variable: Charles

U	5605.500
U Prime	10879.500
Z-Value	-3.160
P-Value	.0016
Tied Z-Value	-3.160
Tied P-Value	.0016
# Ties	129

Mann-Whitney Rank Info for Median Value
Grouping Variable: Charles

	Count	Sum Ranks	Mean Rank
Near	35	11509.500	328.843
Far	471	116761.500	247.901

These results indicate a difference in price between houses near and far from the Charles River. The mean rank for housing near the river is much higher than that for housing far from it. Though the unpaired t -test produced the same conclusion, it could have been fooled had there been significant outliers. The unpaired t -test, since it compares means, can be dramatically influenced by a few outliers. A nonparametric test, however, deals only with the rankings of the observations and cannot be affected by outliers.

Wilcoxon signed rank test

In this exercise you perform a Wilcoxon Signed Rank test using data from blood lipid screenings of medical students. You will determine whether initial triglyceride levels are different from those measured in the same subjects after three years. (This is the nonparametric equivalent of the exercise, [“Paired \$t\$ -test,” p. 34](#). You may wish to compare results between the two tests.)

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Choose Paired two group and click OK
(Leave Wilcoxon Signed Rank selected)
- In the variable browser, select Triglycerides and Trig-3 yrs and click Add
Control-click (Windows) or Command-click (Macintosh) to select several nonadjacent variables

Wilcoxon Signed Rank Test for Triglycerides, Trig-3yrs

# 0 Differences	1
# Ties	7
Z-Value	-.013
P-Value	.9900
Tied Z-Value	-.013
Tied P-Value	.9900

52 cases were omitted due to missing values.

Wilcoxon Rank Info for Triglycerides, Trig-3yrs

	Count	Sum Ranks	Mean Rank
# Ranks < 0	22	450.500	20.477
# Ranks > 0	20	452.500	22.625

52 cases were omitted due to missing values.

There is no significant difference in triglyceride levels between the initial measurements and those made three years later because the p values are very large and the mean ranks are quite close in value.

Kendall rank correlation

In this exercise you perform a Kendall rank correlation. The dataset consists of different western cities rated by nine criteria. You will discover whether there is a relationship between two of the variables, Climate&Terrain and Housing. For Climate&Terrain, the higher the score, the better. For Housing, the lower the score the better.

- Open Western States Rated Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Choose Paired two group, select Kendall rank correlation for the test, and click OK
- In the variable browser, select Climate&Terrain and Housing and click Add

Kendall Rank Correlation for Climate&Terrain, Housing

Score	494.000
Tau	.373
Z-Value	3.898
P-Value	<.0001
Tau corrected for ties	.374
Tied Z-Value	3.913
Tied P-Value	<.0001
# Ties, Climate&Terrain	8
# Ties, Housing	0

The low Tau in these results shows a low correlation between Climate&Terrain and Housing. Compare these results to those in the Correlation chapter; see [“Exercise,” p. 48.](#)

Kruskal-Wallis test

In this exercise you perform a Kruskal-Wallis test using data on weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries. You will determine whether some countries tend to produce larger or smaller cars than other countries.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Select Kruskal-Wallis test and click OK. Two table placeholders appear in the view.
- In the variable browser, select Weight and Country and click Add
Control-click (Windows) or Command-click (Macintosh) to select several nonadjacent variables at once

An X usage marker indicates that Weight is assigned as a continuous variable; a G marker indicates that Country is assigned as a grouping variable.

Kruskal-Wallis Test for Weight	
Grouping Variable: Country	
DF	2
# Groups	3
# Ties	15
H	16.054
P-Value	.0003
H corrected for ties	16.056
Tied P-Value	.0003

Kruskal-Wallis Rank Info for Weight			
Grouping Variable: Country			
	Count	Sum Ranks	Mean Rank
Japan	30	1633.500	54.450
Other	37	1609.500	43.500
USA	49	3543.000	72.306

The small p values indicate that there is a difference in weight depending on the country of origin. The mean rank for the group Other is the lowest, and the rank for cars made in the USA is the highest.

Friedman test

In this exercise you perform a Friedman test using data from a wine tasting in which fifteen people rated six red wines. Each wine was rated using criteria commonly used to judge wine quality. The totals for each judge and wine were calculated. You will determine whether there is a difference in the quality of the wines as determined by the judges. The judges are the blocks; the brand of wine is the treatment.

- Open Wine Tasting Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Nonparametrics and click Create Analysis
- Select Friedman test and click OK
- In the variable browser, select all the continuous variables and click Add

Friedman Test for 6 Variables

DF	5
# Groups	6
# Ties	8
Chi Square	27.552
P-Value	<.0001
Chi Square corrected for ties	28.142
Tied P-Value	<.0001

Friedman Rank Info for 6 Variables

	Count	Sum Ranks	Mean Rank
Wine A	15	56.500	3.767
Wine B	15	57.500	3.833
Wine C	15	66.000	4.400
Wine D	15	27.500	1.833
Wine E	15	36.500	2.433
Wine F	15	71.000	4.733

The large chi-square value indicates that the judges rated the wines differently. Examining the Rank Info table shows the order in which the wines were ranked.

Factor Analysis

Factor analysis reduces a large number of correlated variables to a smaller, more manageable number of factors. A **factor** is a linear combination of related variables that can take the place of the original variables in further analysis. The structure of the factors (the variables represented by each factor) is the most important information resulting from a factor analysis. The number of factors and sufficient dimensionality is also important from a theoretical standpoint, but StatView handles those for you.

Factor analysis is useful when you have many correlated measurements among the experimental units (subjects, plants, etc.) and want to concentrate on a smaller number of values than the number of measurements at hand; or you want to learn about the interrelationships among variables. This technique is known as **dimensionality reduction**. Consider a study of the anatomy of a species of bird, for which you record 100 measurements (beak length, beak width, weight, length of body, length of tail, etc.). It is reasonable to assume that the measurements will be correlated with each other. A factor analysis can help you understand which variables are related to each other, as well as provide a means for you to analyze fewer variables than the original 100.

Discussion

Data input

You can apply factor analysis to two types of data: raw data and a correlation matrix. **Raw data** occurs in standard row and column format (variables in columns, observations in rows). More observations than variables are required in the dataset. **Correlation matrix** data requires a Pearson correlation matrix, which has to be determined from a single pool of subjects rather than from different samples of subjects. You need to know the number of cases used to determine the matrix; StatView uses it for multivariate significance tests performed on the data. StatView uses only the values in the lower left of the correlation matrix. (The part of the correlation matrix below the diagonal is a mirror image of the part above the diagonal.) Thus, you may use either a square correlation matrix (such as one created using the Correlation analysis) or a lower left correlation matrix as input. Note that if your input is a correlation matrix, make sure that all rows in the dataset are included. If you have excluded any rows, make sure you do not add the corresponding column to the analysis.

To calculate the **factor scores**, that is, the values of each of the factors for each of the observations in the data, you must perform the analysis using raw data.

Factor extraction methods

Four factor extraction methods are available in StatView: principal components analysis, Harris image analysis, Kaiser image analysis and iterated principal axis.

Principal components analysis

The **principal components analysis** performs a simple eigenvalue-eigenvector analysis of the correlation matrix in its original form. (**Eigenvalues**, sometimes called **characteristic roots**, **latent roots** or just **roots**, are a mathematical function of a matrix, and are used in many mathematical and statistical techniques.) Principal components analysis is a “classical” technique, often appropriate if your dataset represents a random sample of observations, and the variables you choose are a fairly complete collection of those that are of interest to you. If you are not sure which technique is most appropriate for your data, rely on principal components analysis.

Image analysis

Image analysis is focused more on the sampling of variables than the sampling of subjects. If you can think of the variables in your data as a sample of variables from a potentially large (possibly immeasurable) universe of variables, an image analysis may be more suitable than principal components. Image analysis techniques tend to extract more factors than non-image analysis methods. They factor a modification of the original correlation matrix, the image variance covariance matrix. Due to the large number of factors that generally define an image factor solution, the final rotated solution usually has a large number of zero loadings. However, the non-zero loadings are not always as large as those of the more traditional factor analytic model. Two types of image analysis are available in StatView: Harris and Kaiser. The **Harris** technique appeared in the original literature of factor analysis. **Kaiser’s** technique is a modification that produces a factor pattern whose interpretation can be carried out similar to the more traditional principal components technique.

Iterated principal axis

Iterated principal axis factor extraction is a modification of the principal components technique. It uses the information from the initial principal components extraction to improve the quality of the factor solution. It assumes that the initial number of factors determined by the principal components technique is the correct one, and finds a set of factors that most fully explain the original correlation matrix. To do this, it replaces the diagonal entries of the matrix (by definition equal to 1) with an estimate of the **communality** of each variable (a measure of how closely it relates to the estimated factor solution). It then recalculates the communalities, and continues to factor the adjusted matrix until the communalities no longer change. With this technique, you must choose between three methods for estimating the initial communalities.

Extraction method: Iterated principal axis▼ ✓ SMC
Off-diagonal
1

SMC uses the **squared multiple correlation** of the variable with all the other variables. Off-Diagonal uses the largest correlation between the variable and any other single variable. 1 simply starts the process with the original correlation matrix. The iterated principal axis method is appropriate if you are certain that your data can be very well explained with a small number of factors. Due to its iterative nature, it requires more computing time than the other methods.

Factor loadings

The factor extraction method you choose depends on the nature of your data and the questions you want to answer. The results of a factor analysis are summarized by a primary pattern matrix. For each factor, the entries in this matrix represent the coefficients (often called **loadings**) of the linear combination of the original variables that define that factor. A rescaled version of this matrix, the **oblique solution reference structure matrix**, is displayed in StatView.

Rotations

The coefficients initially produced by a factor extraction method are difficult to interpret because their magnitude varies widely. To get around this, you transform the factor pattern matrix by one or more transformations or **rotations**. The rotation helps you see the structure of the matrix more clearly by transforming it so that, for a given factor, as many variables as possible have either large coefficients or coefficients near zero. You can identify which variables make up a large part of the factor (the large coefficients) and which variables are not very important in that factor. You can then use your knowledge of the dataset to assign meanings to the factors that were extracted. You can experiment with different rotations before deciding which one helps you see the underlying structure of your data best.

For many datasets, determining the number of factors and identifying the important variables in them will satisfy your needs. You may want to go further and incorporate into other analyses the insights into the structure of your data obtained through factor analysis. One easy way to do this is to save the factor scores and later plot or analyze them. For each factor extracted, every observation in your dataset has a **factor score**, provided that the raw data is available. This score is a measure of the magnitude of the variables underlying the factor in question for that observation. You can use the factor scores as you would use other variables to produce plots, compare groups, etc. Factor scores are artificially constructed from a number of different variables so assumptions underlying many statistical procedures may not be met for these scores. Therefore, probability levels reported for hypothesis tests using factor scores should be judged with caution.

Number of factors to extract

An important decision in the extraction stage of your analysis is the number of factors to retain for further study. This number is usually a function of the eigenvalues. Your options are:

state the number of factors you wish to retain; choose the method default, which varies with each factor extraction technique; or specify the technique you want to use. The defaults for each extraction technique are described after the discussion of the available criteria. [delete “In all cases the number of factors extracted is at least two.”]

Factors to extract:	<input checked="" type="checkbox"/> Method default <input type="checkbox"/> Roots greater than one <input type="checkbox"/> 75% variance rule <input type="checkbox"/> Root curve <input type="checkbox"/> User specified	<input type="text"/>
----------------------------	---	----------------------

If you select a technique that depends on the data, there are three criteria used to determine the number of factors: roots greater than 1, root curve analysis and extraction of 75% of the variance.

Roots greater than 1

The roots greater than 1 criterion retains as many factors as there are eigenvalues greater than or equal to 1. Since the sum of the eigenvalues of the correlation matrix is equal to the number of variables, the average value of an eigenvalue is 1. This criteria essentially retains all factors whose eigenvalues are “above average,” and tends to extract a larger number of factors than necessary.

Root curve

The root curve criterion is based on a plot of eigenvalues from largest to smallest. It looks for a point in this graph where there is a dramatic shift, i.e., one eigenvalue that is markedly smaller than the next largest one. The number of factors retained corresponds to the number of eigenvalues before this dramatic change. When you use this criterion, you also get a plot of the eigenvalues versus their ranks, called a scree plot, to help you assess the adequacy of the solution.

75% variance rule

The 75% variance criterion is determined by retaining factors until 75% of the original variance is explained by the factors retained. Since the eigenvalues are determined in order of decreasing magnitude, each eigenvalue accounts for less variance than the preceding one. When the sum of the proportionate contributions of the eigenvalues exceeds 0.75, factors are no longer retained in the final solution.

User specified

If you specify the number of factors to extract, it cannot exceed the number of variables. In practice, most useful factor solutions have a maximum number of factors less than half the number of variables.

Method default

The default number of factors extracted for principal components is two or the number determined by the 75% variance rule, whichever is greater. The default method for the two image analysis models is Harris eigenvalues greater than 1. Harris eigenvalues are the eigenvalues of the image variance-covariance matrix. If you apply one of the three criteria discussed above in place of the default method, the criterion is applied to a modification of the Harris eigenvalues. If you enter a specified number of factors greater than that which might be determined by the image analysis default method, the number determined by the default will override. The default method for determining the number of factors with the iterated principal axis method is to use the number of eigenvalues greater than 1.

Transformation method

You can consider the initial factor solution as your final solution matrix, but it is often difficult to interpret the results of a factor analysis without further transformation. You can choose one of three orthogonal transformations to define a final solution: varimax, equamax and quartimax. An **orthogonal** transformation is one that retains a basic property of the initial factor solution, namely that the factors extracted are uncorrelated with each other. While this property is attractive from a mathematical point of view, it can make it difficult to see the underlying structure of your data.



StatView automatically applies an additional transformation, the orthotran transformation, to the orthogonal transformation you choose in order to make the underlying structure clearer. It does this by relaxing the requirement that the factors remain uncorrelated. If this does not improve the solution, it retains the original orthogonally transformed structure. When the orthotran procedure does perform an additional transformation, the resulting factor pattern is said to be **oblique**, i.e., the factors are not uncorrelated with each other.

Factor scores

If you have a non-singular correlation matrix, you can compute regression estimate factor score weights. This option is available only if you input raw data, since the factor scores are a function of the variable values for each observation in the dataset. Your factor scores are unrotated if you did not choose a transformation method. You have a choice of saving a transformed solution as orthogonal or oblique factor scores. Orthogonal factor scores show zero intercorrelations; oblique scores are correlated. For more information on saving factors scores, see [“Save factor scores,” p. 137](#).

Dialog box settings

When you create or edit a factor analysis, you see this dialog box:

The screenshot shows the 'Factor Analysis' dialog box with the following settings:

- Input:** ☒ Raw data, ☐ Correlation matrix - # cases: []
- Extraction method:** Principal components (dropdown), SME (dropdown)
- Factors to extract:** Method default (dropdown), []
- Transformation method:** Orthotran (dropdown), Varimax (dropdown)
- Save to dataset:**
 - ☒ Factor scores: Oblique (dropdown)
 - ☐ Correlation matrix
- Buttons:** Cancel, OK

All the choices in the dialog box are discussed in greater detail in preceding pages. First you must specify the type of input data, raw data in row and column format, or a correlation matrix. If your input is a correlation matrix, the number of cases used to determine the correlation matrix must be entered.

You choose the factor extraction method from the pop-up menu. If you choose iterated principal axis extraction method, you must also specify the initial communality estimate as SMC (squared multiple correlations), off-diagonal, or 1 (see earlier [“Discussion,” p. 131](#)). You also choose the method for determining how many factors to extract, from the pop-up menu. More detail on these choices can be found in the earlier section [“Number of factors to extract,” p. 133](#).

There are three transformation methods to choose from in addition to the automatic orthotran transformation. They are varimax, equamax, and quartimax. You may also choose no transformation. If your input data is raw data, the checkboxes at the bottom of the dialog box let you save either factor scores or a correlation matrix.

Save a correlation matrix If you check save correlation matrix, the computed correlation matrix is saved to a new dataset titled Factor Analysis Correlation Matrix. The dataset will have as many columns and rows as variables assigned to the factor analysis. The names of each column are *Cor* “Variable name” where “Variable name” is the name of one of the assigned variables for the factor analysis.

Note that the correlation matrix dataset is a very special dataset with many features. The dataset is linked to the factor analysis. If you change the parameters of the analysis or any of the input data, the dataset will *automatically* update to reflect the new correlation matrix. If you close the view that contains the factor analysis, this correlation dataset will close as well. When the view is reopened, the correlation matrix dataset will automatically be recreated. Please note that because this dataset is linked to your analysis, it is a “read only” dataset; you can not change any value in the dataset (except the formatting) until you break the link between the dataset and the analysis. In addition, the variables in this dataset can only be used in the view which contains the factor analysis that it is linked to.

To sever the link between the dataset and the factor analysis, you need to choose Save As from the File menu and save the dataset under a different name. This will save on the disk a copy of

the correlation matrix as a normal dataset. You can then open this dataset as you would any other dataset. When you save a copy of the correlation matrix dataset to your disk, StatView automatically appends the letters “UE” to the beginning of the column names to indicate that these columns are now user entered.

Save factor scores This option is available only if you input raw data, as opposed to a correlation matrix, since the factor scores are a function of the variable values for each observation in the dataset. Your factor scores are unrotated if you did not choose a transformation method. You can save a transformed solution as orthogonal or oblique factor scores. Orthogonal factor scores show zero intercorrelations; oblique scores are correlated.

The factor scores are appended to the end of the dataset to which the first specified variable belongs. They are assigned the names Obl 1, Obl 2, etc., or Orth 1, Orth 2, etc., depending on the type of scores saved. StatView identifies the source of these variables as **analysis generated**. They are dynamically linked to the factor analysis that created them. If you change the parameters of the analysis or any of the input data, the variables in the dataset automatically update. In addition, the variables are tied to the view that contains the analysis, not the dataset in which they appear. They will automatically be added to the dataset again when the view is reopened and the factor analysis recalculated. If you close the view that contains the factor analysis, the variables will be removed from the dataset. Note that one consequence of this is that if you plan to use an analysis generated factor scores in a formula, you need to open the view which contains the factor analysis in order for the formula to compute.

Since these variables are dynamic, if you generate a graph or statistic of these factor scores, these graphs or statistics will update when the analysis changes. If you plan to create new analyses or graphs from the factor scores, such as a histogram or descriptive statistics, these results must be contained in the same view as the factor analysis.

To break the link between an analysis generated variable and the analysis, change its source to User Entered. This causes all ties to the analysis to be broken and the letters “UE” appended to the front of the variable name to indicate that it is now user entered. Any change to the factor analysis that created it will have no effect on the variable, and they will act just as any user-entered variable would.

Data requirements

Factor analysis requires three or more continuous variables.

Variable browser buttons	
Add	To generate a factor analysis, select three or more continuous variable(s) and click Add. When you select a factor analysis result and assign additional variables, they are added to the existing analysis.
Split By	When you assign one or more split-by variable to a factor analysis, results for each cell in the split-by variable(s) are displayed in separate tables and plots.

Results

The following results are available for factor analysis. The Basic output is the default.

Basic output	Summary table, eigenvalues, unrotated factors, communality summary, oblique solution primary pattern matrix, and oblique solution reference structure.
Supplemental output	Correlation matrix, partial correlation matrix, eigenvectors, orthogonal transformation, primary intercorrelations, oblique factor score weights, orthogonal factor score weights.
Advanced output	Variable sampling, variable complexity, proportionate variance contributions.
Plots	Unrotated factor plot, orthogonal factor plot, oblique factor plot, scree plot.

Templates

The following templates provide factor analysis results.

Factor Analysis	Factor Analysis Plots	Unrotated factor, oblique factor, orthogonal factor, and scree plots.
	Factor Analysis--Basic	Factor analysis summary, eigenvalues, unrotated factors, communality summary, oblique solution primary pattern matrix, and oblique solution reference structure tables.
	Factor Analysis--Complete	Factor analysis summary, eigenvalues, eigenvectors, unrotated factors, communality summary, oblique solution primary pattern matrix, and oblique solution reference structure, oblique score weights, orthogonal score weights, orthogonal solution, partial correlation matrix, primary intercorrelations, proportional variance contributions, sampling adequacy, unrotated factors, and variable complexity tables. Unrotated factor, oblique factor, orthogonal factor, and scree plots.

Exercise

In this exercise you perform a factor analysis to find the factors that best explain variability in a correlation matrix of eight physical measurements.

- Open Eight Physical Variables Data from the Sample Data folder
- From the Analyze menu, select New View
- From the analysis browser under Factor Analysis, select Basic Output and click Create Analysis
- Enter 305 for # cases and click OK
- In the variable browser, select all the variables and click Add

Factor analysis summary The summary table notes the number of variables used in the analysis, the factor procedure used to determine the number of factors, the transformation procedure and the number of factor scores defined. It also includes Bartlett's chi-square test; a

significant chi-square value suggests that the collection of coefficients in the correlation matrix differ from 0 and most likely do not occur by chance.

Factor Analysis Summary

Number of Variables	8
Est. Number of Factors	4
Number of Factors	2
Number of Cases	305
Number Missing	0
Degrees of Freedom	35
Bartlett's Chi Square	2116.975
P-Value	<.0001

Factor Extraction Method: Principal Components

Extraction Rule: Method Default

Transformation Method: Orthotran/Varimax

Eigenvalues The eigenvalues are presented in an order that corresponds to their size. Typically, there are as many eigenvalues as there are variables, and the sum of the eigenvalues equals the sum of the diagonal elements of the matrix from which they are determined. The variance proportion is an estimate of the proportion of variance that the eigenvalue and its eigenvector account for when they are used to define a factor.

Usually, StatView divides the number of variables by two to determine an initial estimate of the number of eigenvalues (also an initial estimate of the number of factors). The many rules for determining the number of final factors are then applied to the eigenvalues. You may override the number of eigenvalues determined initially by entering a number of factors in the dialog box. The eigenvalues displayed are of no great value in the interpretation of the factor solution. They are displayed for completeness and for those who wish to address subjectively the number-of-factors question.

Eigenvalues

	Magnitude	Variance Prop.
Value 1	4.673	.584
Value 2	1.771	.221
Value 3	.481	.060
Value 4	.421	.053

Unrotated factors Once the number of factors is determined, it is necessary to determine the correlation of each variable with each factor, a value typically referred to as a factor loading. Most modern-day factor analysts view this unrotated factor matrix as the initial step in determining a desirable factor solution matrix. The square of a loading represents the proportion of variance of the variable that can be predicted by the factor.

Unrotated Factors

	Factor 1	Factor 2
height	.859	-.372
arm span	.842	-.441
forearm length	.813	-.459
lower leg length	.840	-.395
weight	.758	.525
bitrochanteric diameter	.674	.533
chest girth	.617	.580
chest width	.671	.418

Communality summary Computing the sum of the squared loadings by row results in a proportion, the final communality estimate, that represents the total proportion of variance of the variable that can be predicted by the factors.

Communality Summary

	SMC	Final Estimate
height	.816	.877
arm span	.849	.903
forearm length	.801	.872
lower leg length	.788	.861
weight	.749	.850
bitrochanteric diameter	.604	.739
chest girth	.562	.717
chest width	.478	.625

Prior to a factor analysis, the total proportion of variance of a variable is estimated by the squared multiple correlation (SMC) of the variable with all the other variables. The communality estimates and the SMC are reported in the communality summary table. Some analysts think of the SMC as the initial communality estimate, while others think of the largest off-diagonal entry associated with the variable as the initial communality estimate. When a singular (determinant equal to 0) correlation matrix is analyzed, the initial communality estimate is assumed to be 0.

You can see from this communality summary table that approximately 82 percent of the variation in height is predictable in a linear regression equation using the other seven variables. This conclusion is derived from the SMC of height with all the other variables. When two factors are used to predict height, approximately 88% of the variation is predictable, an improvement of approximately 6%.

Oblique solution primary pattern matrix When determining an oblique solution, StatView uses an algorithm that simply takes a given orthogonal solution and releases the restriction of orthogonality. The algorithm, the orthotran solution, always defines a simple structure solution that is good as or better than the associated orthogonal simple structure solution.

Oblique Solution Primary Pattern Matrix

	Factor 1	Factor 2
height	.919	.034
arm span	.973	-.046
forearm length	.971	-.079
lower leg length	.928	3.423E-4
weight	-4.694E-5	.922
bitrochanteric diameter	-.063	.890
chest girth	-.145	.911
chest width	.043	.768

There are two types of oblique solutions: a primary pattern solution and a reference structure solution. These two are quite similar; indeed, one is a column rescaling of the other. The pattern solution defines loadings that are regression coefficients for predicting the standard score of a variable in terms of the defined factors. The reference structure solution defines loadings that are correlations. Both solutions have good simple structure in that the high loadings are high, and the low loadings are near zero.

Oblique solution reference structure When comparing a primary pattern solution to a reference structure solution, it is immediately apparent that the large loadings are larger in the pri-

mary pattern solution. Sometimes these primary pattern values become larger than 1, simply because they are regression weights. Regardless of whether you use a primary pattern or reference structure solution, the conclusions should be the same. For this data, it is clear that the first four variables are associated with the first factor and not associated with the second factor. Using similar logic, it is apparent that the second four variables are associated with the second factor. To name the factors, you choose a name that represents the essence of the variables loading on it. The first factor could be named bone structure, the second factor could be named flesh factor.

Oblique Solution Reference Structure

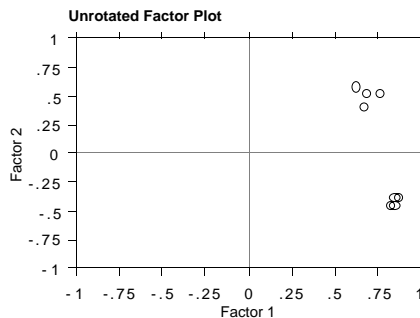
	Factor 1	Factor 2
height	.795	.030
arm span	.842	-.040
forearm length	.840	-.068
lower leg length	.803	2.962E-4
weight	-4.062E-5	.798
bitrochanteric diameter	-.055	.770
chest girth	-.126	.788
chest width	.038	.664

For these data you would arrive at the same factor name if you used an orthogonal solution. Is it reasonable to assume that body weight or flesh is independent of bone structure? If you believe so, then you may be satisfied with an orthogonal solution. If, however, you assume that taller people are generally heavier and fleshier than shorter people, you will be satisfied with an oblique solution.

Plots

StatView provides several plots associated with factor analysis. In this part of the exercise, you create two: one associated with the unrotated factor solution, and one associated with the oblique solution. Within any particular set of plots, all pairwise factor plots are presented.

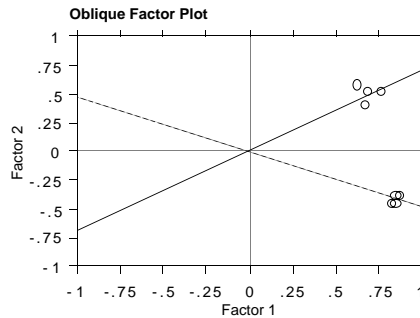
- Make sure one of the previous results is selected
- In the analysis browser, select Unrotated Factor Plot and click Create Analysis



The plot of the unrotated solution allows you to make a quick judgment regarding the potential simple structure of the factor solution. For this data, two distinct clusters of points are apparent in the unrotated plot. An ideal factor solution for the variables would have one axis

passing through the cluster of variables 1 through 4 in the upper right quadrant, and the other axis passing through the other cluster. If the data were under-factored (which is not possible with the eight physical variables), you might see points scattered through all four quadrants with no definitive clusters of points. If the data were over-factored, you would see many points near the point of intersection of the two axes, and perhaps one or two points defining a cluster.

- In the analysis browser, select Oblique Factor Plot and click Create Analysis



The plot of the oblique solution shows the oblique axes, primary axes, passing through the clusters of points as they do for the eight physical variables. The plotted primary axes are not at right angles because they are correlated. In this example, the simple structure of the oblique solution is quite good; the primary axes pass directly through the clusters. When the orthogonal solution passes axes through the clusters, the oblique solution and the orthogonal solution are identical and the factor intercorrelations are zero.

Survival: Nonparametric

This is the first of two chapters regarding StatView's survival analysis tools. This chapter introduces survival analysis in general and goes on to discuss StatView's Survival: Nonparametric analyses in particular. The second, [“Survival: Regression,” p. 167](#), discusses the Survival: Regression analyses.

Introduction to survival analysis

What is survival analysis?

Survival analysis is a suite of statistical techniques used to evaluate data consisting of the elapsed time between two events of interest. A typical example, on which the name is based, concerns the length of time that seriously ill patients survive. In such a case, the survival time is often measured from the initiation of treatment (i.e., the beginning of monitoring) and ends, typically, with death (the **event**). Some questions that may interest an investigator collecting such data are:

1. What is the mean time from initiation of treatment to death?
2. What is the probability that a patient will survive five years after treatment begins?
3. Does the patient's condition before commencement of treatment affect the length of survival time?
4. How do survival times after an experimental treatment compare with those for patients exposed to a standard treatment?
5. What factors, by lengthening or shortening the time from treatment to death, influence the success of the treatment?

Survival analysis methods are the statistical tools designed to answer these kinds of questions.

Survival analysis methods can be applied to a broad class of problems in engineering, economics, demography, and the social and natural sciences. In engineering, reliability studies are carried out to evaluate how long certain components or systems function before they fail. In job-mobility investigations, the length of time that an individual remains in a certain job is a primary focus. A study of fertility may wish to estimate factors which influence the time from menarche to first birth for a population of women. These different applications all use survival analysis techniques, although they may be given a name more appropriate to the topic under

study. Other names for survival analysis include failure-time analysis, reliability theory, and lifetime data analysis. It is important to note that the ideas of survival analysis can be applied to the study of any variable with non-negative values, not just those that arise from measurements of elapsed time. For example, health care analysts are often concerned with the properties of the total costs associated with treating a specific disease in an individual. In this case the variable “cost” plays the role of the non-negative variable whose behavior you wish to explain in the analysis; observations which are *minimum values* of the cost of treating patients (perhaps because their treatment is not complete) would be considered censored observations.

Survival and hazard functions

In addition to familiar statistical summaries for describing the properties of variables within a single population or comparing characteristics across populations, there are three closely related functions that play a special role in most survival analyses. The first is the **survival function**, usually denoted by $S(t)$, which, for any specified time t , gives the probability of an individual’s survival at least to time t . In a population, $S(t)$ then yields the proportion of the population that will survive beyond time t . Thus, if the survival variable T measures the time in years from diagnosis of a certain cancer until death, then $S(5)$ is the probability of surviving five years or more. The survival function is closely related to the associated distribution function of T .

The **hazard function** provides an alternative way to convey the same information as a survival function, but it is particularly appealing because of its natural interpretation on a chronological time scale. The hazard function, often denoted by $\lambda(t)$, gives for any specified time t the instantaneous risk of failure at time t among individuals who have survived *at least* to time t . Note that, for a continuously monitored population, $\lambda(t)$ is not the proportion of individuals who fail at time t ; instead it measures the proportion among those individuals *at risk at time t* who fail at time t . Thus, the hazard function provides a way to look dynamically at how the risk of failure changes as time progresses. An increasing hazard function reflects increasing risk as time progresses and vice versa. A constant hazard function indicates that the risk of failure is unaffected by the length of time an individual has already survived. The **cumulative hazard function**, $\Lambda(t)$, measures the cumulative risk to which an individual is exposed up to time t ; it equals the negative of the logarithm of the survival function.

As suggested above, these three functions (the cumulative survival, hazard, and cumulative hazard functions) are closely related, as indicated by the following equality:

$$S(t) = e^{-\int_0^t \lambda(u) du} = e^{-\Lambda(t)}$$

As a consequence of this relationship, if you know just one of these functions, you can infer the values of the other two.

Regression models

As in many quantitative analyses, we are often most interested in relationships between variables; thus, in survival analysis, we may wish to determine which factors influence survival time. Typically, as in other statistical investigations, regression analysis is used to investigate

and quantify the effects of explanatory variables on an outcome of interest—here, survival time. In survival analysis two forms of regression analysis have proved useful. The first employs standard linear regression models in which survival time plays the role of the dependent variable. Often, a transformed version of survival time (usually the logarithm of survival time) is used, leading to a linear model on the log time scale—the so-called **accelerated failure time model**. In this model, the effects of changes in explanatory variables are quantified in terms of the multiplicative effects of such changes on survival time. That is, if the dose of a treatment given to one individual is one unit greater than that given to another, the model measures whether the individual's survival time is twice as long, three times as long, or half as long, and so on.

The second kind of model describes the way the hazard function is affected by changes in the explanatory variable. The popular **proportional hazards model** describes the effect of changes in explanatory variables in terms of the multiplicative effect on the hazard function. That is, in the example of the last paragraph, this model indicates whether the patient receiving the higher dose is subject to twice the hazard, three times the hazard, or half the hazard throughout the subject's monitoring period.

Parametric and nonparametric analyses

In statistical analyses, investigators often have considerable flexibility in how much structure they are willing to assume regarding the variables under study. For example, in describing the survival properties of a population of items under test in a reliability study, an investigator may wish to assume that the underlying hazard function is constant; that is, that the risk of failure for items on test is not influenced by the amount of time on test. This is equivalent to a parametric assumption that survival times are drawn from an exponential distribution. Subsequently, the constant hazard can be estimated from survival times of a sample of items. Alternatively, the investigator may be unwilling to make such a strong assumption and, at least initially, leave the form of the hazard function unspecified. Based on survival data, a nonparametric estimate of the cumulative hazard function and the associated survival function can then be calculated.

Censored observations

Collecting survival information on a sample of individuals often involves longitudinal follow-up to monitor a subject's failure status. Sometimes it is impossible to determine the exact time of failure, particularly in cases where a study ends before an individual has experienced the event. Such cases require specialized statistical techniques that allow one to use both complete observations and incomplete information simultaneously.

A **censored** observation is one for which only partial information is available on the survival time of an individual under study; for example, **right censoring** refers to the case where it is known only that a survival time exceeds a known value, t_C . This kind of information will be available when we have monitored an individual for t_C time units and the individual still has not failed when observation ceases. For censored observations it is important that the dataset reflects whether an individual's recorded information has been censored. Understanding the

patterns and properties of censoring and how these can influence your observations are crucial to the correct interpretation of survival data.

An example

An investigative team of clinicians, data managers and statisticians have organized a clinical trial of a new chemotherapy in the treatment of a certain form of lung cancer. Upon referral from oncologists and with informed consent, 300 patients recently diagnosed with lung cancer are randomly assigned either to the new treatment (at one of two doses) or to the current standard therapy, each group comprising 100 patients. The patients are carefully monitored for side effects and their health status is followed for two years after treatment or until the time of death. Data collection ends when the follow-up of all patients is completed. For each patient, the dates of diagnosis, initiation of treatment, and end of follow-up or death are recorded in a data file. For individuals whose length of follow-up was less than two years, the reason for cessation of monitoring—death, removal from study, etc.—is also recorded. Other relevant information is also collected in each case, including the stage of cancer at diagnosis, age of patient at diagnosis, and other clinical and demographic measurements.

After the data are collected, the team is eager to study the results of the trial. Initially they consider the data obtained from the 100 patients assigned to receive the standard treatment. It is decided to measure the relevant survival variable as the time from diagnosis until death. In the dataset, these data are recorded in the **event time** variable. Using a Kaplan-Meier analysis, the investigators obtain an estimate of the survival function. For comparison, using a parametric regression model, they also fit a Weibull survival function to the data. By examining the results, the team is convinced that the Weibull model is inadequate, and notes that, for the standard therapy group, the Kaplan-Meier estimate of the survival function is similar to analogous curves based on historical data on the effects of the standard treatment. The latter comparison is helpful in ascertaining whether there might be any survival differences for individuals enrolled in the present trial as compared with past patients.

The Kaplan-Meier estimate (a nonparametric method) of the survival curves for the two experimental groups are plotted on the same graph as the standard group. The Kaplan-Meier estimate also allows rank tests to be used to compare the three survival curves. Among these, the logrank test is chosen. This test evaluates whether the observed differences among the survival curves can be attributed to chance variation or to actual differences among the three groups.

If the hazard functions in the three groups are assumed to be proportional, the proportional hazards regression model can be used to quantify the **relative hazard** obtained by comparing the two new treatment groups to the standard group.

Before interpreting the results, the investigators use StatView to check the proportional hazards assumption, by examining plots of estimates of the three hazard functions. For example, a graph of the log cumulative hazard functions for each group shows three approximately parallel curves. These and other evaluations suggest that the proportional hazards assumption is reasonable. The regression analysis shows that the hazard function is reduced by about 20 percent under treatment in the low-dose group, and 25 percent in the larger-dose group. The dif-

ference between the two different doses of the new treatment could be due merely to chance variation.

With this important part of the analysis completed, the investigators now consider the role of other factors in survival and whether certain patient characteristics might be associated with treatment efficacy. First, basic patient characteristics are entered as covariates into the proportional hazards model to investigate their influence on survival time and to examine whether compensation for these effects might improve the precision associated with the treatment group comparisons. It is discovered not only that younger patients survive longer, but also that the higher dosage is considerably more effective than the lower dosages. Specifically, the hazard is reduced in the younger patients with the high dose of the new treatment by 45 percent compared with the standard treatment in patients of average age. Although the results are not definitive, this finding suggests further investigation of the appropriate dose level for the new treatment in younger patients.

Finally, information related to the causes of censoring is examined and the data are evaluated to determine whether specific patient characteristics are associated with the chance of being censored. This analysis helps the investigators assess their assumption that censoring is not associated with the risk of mortality.

Thus, a full analysis of survival data uses many of the options available in StatView. Effective use of the right combination of these tools is the key to appropriate analysis, interpretation, and reporting of survival data.

Nonparametric methods

In survival analysis, the time that elapses until the occurrence of an event of interest—hereafter referred to as the **event time**—is recorded for a sample of individuals from a defined population. As indicated in [“Introduction to survival analysis,” p. 143](#), some of these observations may be censored, because, for instance, the study may end before the event occurs for particular individuals. For such individuals, only a lower bound for the event time is known; that is, it is known only that they did not experience the event within a certain time interval. Having both **uncensored** (or **complete**) and **censored** (or **incomplete**) observations as data, an investigator typically wants to study the characteristics of the survival and hazard functions (see [“Introduction to survival analysis,” p. 143](#), for an explanation of these terms).

Specifically, comparison of the survival and hazard functions across *natural groups* of individuals is often a key issue. What constitutes a natural group depends on the context in which the data are collected. In randomized clinical trials, the primary groups are usually the various treatment groups to which individuals are assigned. In observational studies, the groups might be determined by natural characteristics of the individuals, such as occupation or age, or may reflect some condition to which individuals have been exposed, such as a history of smoking. Since in these cases, group membership is not assigned at random (a person cannot be *assigned* a gender, for instance), comparisons of survival functions among such groups must be interpreted with caution.

In analyzing survival data, it may be appropriate to assume that the hazard function belongs to a family of equations of a simple mathematical form, the parameters of which are all

defined. The choice of an appropriate parametric family will depend on external information about the population's survival properties. The data also help determine if a particular parametric model is appropriate. We return to this issue in the next chapter, "[Survival: Regression](#)," p. 167, where we discuss methods for estimating and comparing survival functions that are based on parametric models.

As one alternative to strictly parametric models, in this chapter we consider ways to estimate and compare survival functions that are *not* based on any specific parametric model. Such procedures are referred to as **nonparametric**. These methods are valuable both when there is little, if any, *a priori* information on which to base the choice of a specific parametric model and for providing a benchmark estimate of the survival function (i.e., one requiring minimal assumptions about the data) that can be compared to estimates that emerge from specific parametric models.

Event times can be recorded on either **continuous** or **discrete** scales. ("Continuous" in this context should not be confused with the continuous data class used in StatView. As explained in "[Data requirements](#)," p. 157, the event time variable must always be continuous, regardless of whether it is measured on a continuous or discrete scale.) For example, consider a case in which event times are recorded in days, even though it typically takes several months or years for the event to occur. For such cases, few, if any, individuals are likely to share exactly the same event time, and it would be appropriate to use a continuous scale. On the other hand, if event times are recorded only to the nearest month or year, it is probably more appropriate to treat the event times as discrete, because a relatively large proportion of individuals will share identical event times. For data recorded on a continuous scale, it is possible, in principle, to estimate the survival function parametrically over a continuous interval of time. With discrete data, however, the survival function can be estimated only at a few discrete time points. If it is determined that event times should be treated as discrete data, **actuarial** (also called **life table**) estimates of the survival function should be used.

A key assumption that underlies both the estimation and comparison of survival functions is that the causes of censoring of observations are *not related* to event times. For example, if follow-up is terminated for some individuals—who thus become censored observations—because their survival prognosis is poor (that is, the occurrence of their event is imminent), the methods described in this chapter, and throughout this manual, are inappropriate.

Discussion

The first concern when analyzing survival data should be to estimate the underlying survival function. Later, it may be valuable to compute separate survival function estimates for groups and/or strata of the population. Examination of the latter estimates can provide insight into causes of survival patterns and their variation across groups of interest. In particular, an estimate of the survival function yields estimates both of the probability of surviving a set period of time—for example, one year, five years, etc.—and of the uncertainty associated with these estimates. Beyond characterizing the survival patterns for the population under study, these estimates are useful for establishing the prognosis of future individuals, and for comparison with other groups or populations.

Understanding the event time variable

Before describing various estimates of the survival and related functions, it is important to say a little about the definition of the key variable in a survival analysis. This is the event time variable, which measures the time that elapses until the occurrence of the event of interest, or, in the case of censored observations, until the individual is no longer monitored. As indicated in [“Data requirements,” p. 157](#), it is necessary to enter information that distinguishes among those individuals who are followed until the event occurred and those who were subject to censoring. Of more fundamental importance to the investigator is the need to define carefully both the *origin* of the event time (which may begin, for example, with the date of diagnosis, the date of randomization, the date of treatment initiation, etc.) and the endpoint of interest (for example, death from any cause, death from a specific cause, relapse, etc.). In addition, the investigator should consider the choice of the numerical scale for the event time variable. (Note that this issue is distinct from considerations surrounding the use of continuous and discrete scales, as discussed on [p. 148](#).) In many cases, this may merely be the selection of a particular chronological scale such as days, weeks, or years. In other cases—for example, in the monitoring of machine failure patterns—an alternative to chronological time may be suitable. For cars, accumulated mileage until failure might be preferable to time since manufacture as the **event time** variable; for electronic components, the number of switches on or off until failure may be more relevant than time until failure.

Nonparametric survival function estimates

In a preliminary analysis of survival data, the investigator might begin by plotting a nonparametric estimate of the survival function, $S(t)$, against time. (As discussed in [“Introduction to survival analysis,” p. 143](#), $S(t)$ indicates the proportion of the population for whom, at time t , the event of interest has not yet occurred.) If the event times are treated as continuous, then it is conventional to use the **Kaplan-Meier** estimator of the function $S(t)$. (Note that the Kaplan-Meier estimator is sometimes referred to as the **product limit** estimator.) The survival function generated by this estimator is a step function. As a step function, $\hat{S}(t) = 0$ at the origin ($t = 0$) and it remains at that value until the first **jump point**, (i.e., event time) where it takes on a value less than 1, remains “flat” until the next jump point, and so on. The set of times at which $\hat{S}(t)$ changes is simply the set of uncensored event times in the dataset.

If the event time measurements are discrete, or if the dataset is very large, an **actuarial** estimate of the survival function may be used. In this method, the uncensored and censored event times are grouped into predefined intervals on the time axis, and the survival function is estimated at the beginning of each interval.

Graphs of the estimated survival function can be embellished in several useful ways. For instance, symbols indicating observed event times or observed censoring times (or both) can be included on the graph. It is particularly useful to indicate where censoring occurred during follow-up or monitoring, which in turn could indicate nonrandom causes of censoring, particularly when estimated survival curves are compared across groups.

Hazard plots

While the cumulative survival function plot is useful in its own right, it is difficult to infer the form of the underlying hazard function directly from this graph. This task is made simpler by examining a variety of hazard plots. The first of these is the **cumulative hazard plot**, which graphs an estimate of the cumulative hazard function, $\Lambda(t)$, against time t . This plot provides insight into the development of hazard over time; for example, changes in the rate at which the cumulative hazard function grows reflect whether the hazard function is increasing or decreasing. This kind of qualitative sense of the shape of the hazard function is enhanced by the **ln cumulative hazard plot**, which graphs an estimate of $\ln(\Lambda(t))$ against the logarithm of time, $\ln(t)$. This plot is particularly useful for judging whether some of the parametric models of the chapter [“Survival: Regression,” p. 167](#), adequately describe the survival properties reflected in the data. Specifically, if the event times are sampled from a Weibull distribution, the log cumulative hazard plot should produce points that lie approximately on a straight line; if the slope of the approximating line is close to 1, an exponential model may be appropriate.

With interval-grouped data, the hazard plot graphs the estimate of the hazard function, $\lambda(t)$, against time, based on the actuarial estimate of the survival function. This plot allows a direct interpretation of how the risk of failure evolves over time.

Comparisons of survival functions

The detection of differences in survival patterns among groups of subjects is often a primary motivation for survival analysis. A first step to this end is the construction of *separate* estimates of the survival function for each distinct group. These might be plotted on different graphs, or, more usefully, on the same graph. These plots allow immediate comparison of estimated survival probabilities and observed censoring times. A next step is to assess whether observed differences in estimated survival functions might be due to chance variation alone; this can be achieved through a variety of tests to evaluate the equivalence of survival functions across groups.

Comparing survival functions across groups—rank tests

Statisticians have suggested various procedures to test the hypothesis that survival functions among groups are equal. One way to evaluate the equality of survival functions is as follows: First, consider each of the observed event times in turn; for each such time and each group, one can calculate how many individuals of the original dataset were at risk of failure (at each event time, some may have already failed or been censored, so these cases are no longer at risk). For each observed event time, the proportion of individuals at risk in each group is evaluated in comparison to the group membership of the individual who actually failed. For example, if at the early event times an active treatment group and a placebo group include roughly the same number of individuals at risk, but the observed events all belong to the placebo group, this would provide evidence that, initially, the risk of failure is higher in the placebo group. So, one way to compare survival functions among groups is to calculate these comparisons at each event time and then combine this information across all event times. Different test statistics are obtained according to how one **weights** the evidence obtained at dis-

tinct event times. The different weighting schemes correspond to tests with different names, but all are generically referred to as **rank tests**, because they depend only on the ordering of the event times and not on the numerical values.

The most common of these tests is the **logrank** test (also known as the **Mantel-Cox** or **Mantel-Haenszel** test); it gives equal weight to all observations and is best suited to detecting differences among survival curves for which the underlying hazard functions are proportional. (Such proportionality is usually indicated by parallel lines in cumulative hazard plots for these groups. Another method to test this assumption is to plot the logarithm of the survival estimate for one group against the logged estimate for the other group: the resulting plot should be close to a straight line through the origin.) An alternative weighting leads to the **Breslow-Gehan-Wilcoxon** test, which gives greater weight to times with more observations in the risk set; it is, therefore, less sensitive than the logrank test to late events when few subjects remain in the study. If there are no censored observations, this test simplifies to the **Wilcoxon** test. Another generalization of the Wilcoxon test is the **Tarone-Ware** test, which gives a weighting between the logrank and Breslow-Gehan-Wilcoxon tests. A further variant is the **Peto-Peto-Wilcoxon** procedure, which uses an estimate of the survival function for its weightings. Finally, there is the **Harrington-Fleming** family of tests, in which the weighting is controlled by a parameter ρ .

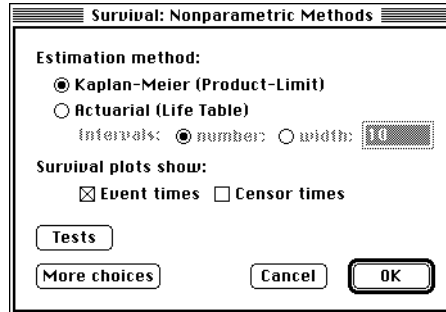
Usually, these test statistics provide very similar summaries of the evidence for or against the hypothesis that the survival functions of the various groups are equivalent, at least in datasets that are moderately large. The Harrington-Fleming test with $\rho=0$ is identical to the logrank test; with $\rho=1$, Harrington-Fleming is similar to the Peto-Peto-Wilcoxon test. In the Breslow-Gehan-Wilcoxon test, the weighting depends on the censoring patterns in the dataset and so can lead to anomalous results if censoring is common and differs substantially across the groups.

Sometimes one group may be at lower risk early in the monitoring period, but at higher risk later. It is important to note that none of the tests described are effective at detecting this kind of difference. Use of these tests, therefore, should always be supplemented by visual comparison of the estimated survival curves for the various groups.

All of the tests for comparing groups can be replaced by analogous **tests for trend** among groups. This may be appropriate when there is a natural ordering associated with the groups (for example, in a case where groups are defined by varying dosage levels of a drug). Trend tests are intended to detect departures from the null hypothesis—i.e., that the survival functions among groups are equivalent—in the direction of increasing or decreasing survival proportions as one moves through the groups in the specified order. If the ordering of the groups can be quantified—for example, by a measure of dose—then the group variable that comprises these values can be used as a covariate with the regression methods in the chapter [“Survival: Regression,” p. 167](#), to examine more closely the relationship between the covariate and survival.

Dialog Box Settings

Survival: Nonparametric Methods dialog box



The settings in this dialog box control the computation and display of all results within the Survival: Nonparametric Methods header in the analysis browser. By default, this dialog box is accessed by clicking the Create Analysis button after choosing any result within the Survival: Nonparametric Methods header. If you prefer, the more choices version of this dialog box can be made the default by changing the setting of the Survival Analysis Preferences dialog box (see [“Survival Analysis preferences,” p. 230 of Using StatView](#)). The fewer choices dialog box also can be accessed by clicking the Fewer choices button in the more choices version of the Survival: Nonparametric Methods dialog box (see below).

Estimation method These radio buttons allow you to choose between two methods for computation of the survival function. The Kaplan-Meier (Product-Limit) option calculates the survival function by the Kaplan-Meier method, and is the default. The Actuarial (Life Table) option calculates the survival function by the actuarial method. If the actuarial method is enabled, the Intervals options are enabled, and the Survival table: Sort by options (see below) are disabled.

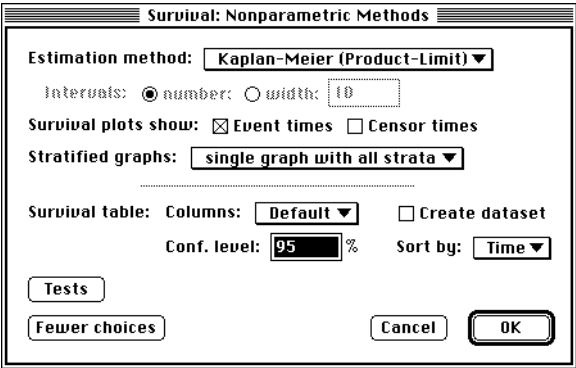
Intervals These radio buttons and text field allow you to set the intervals used in computing the actuarial survival function. If the number option is chosen, the actuarial estimate is based on a specified number of evenly divided intervals, the number of which is set in the text field following the width option. If the width option is chosen, the estimate is based on evenly divided intervals of specified width. This width (in units of the event time variable) is set in the text field following this option. These radio buttons and text field are active only if the Actuarial (Life Table) estimation method is selected.

Survival plots show These checkboxes allow you to specify the data that are displayed on any cumulative survival plots that are created. If the Event times checkbox is enabled (the default), symbols denoting the occurrence of uncensored events are plotted on cumulative survival plots. If the Censor times checkbox is enabled, symbols denoting the occurrence of censored events are plotted on cumulative survival plots.

Tests Clicking this button opens the Rank Tests dialog box, described under [“Rank Tests dialog box,” p. 156](#).

More choices Clicking this button opens the more choices version of the Survival: Nonparametric Methods dialog box. This dialog box is described immediately below.

More choices



Additional options are available in the More choices dialog box. This dialog box is accessed by clicking the More choices button in the fewer choices version of the Survival: Nonparametric Methods dialog box. If you prefer, this more choices version of the Survival: Nonparametric Methods dialog box can be made the default by changing the setting of the Survival Analysis Preferences dialog box (see [“Survival Analysis preferences,” p. 230 of Using StatView](#)).

Stratified graphs This pop-up menu allows you to specify how data from stratified analyses are displayed in graphs. If the Single graph with all strata option is chosen (the default), results for all strata are displayed in a single graph. If the Separate graph for each stratum option is chosen, results for each stratum are displayed in separate graphs.

Survival table: Columns This pop-up menu allows you to specify which columns are displayed in the computed survival table and saved to a dataset, if specified. There are three options available from this pop-up menu: Default, Complete and Specify....

The choice of contents of the survival table depends on whether the estimation method is Kaplan-Meier or actuarial. The following tables show which values will be included in the survival table if this pop-up menu is set to either Default or Complete.

Estimation method	Default columns	Additional columns for Complete
Kaplan-Meier	Time Status Cumulative Survival Cumulative Failure Survival Standard Error Cumulative Events Cumulative Censored Remain at risk	Case Cumulative Survival Confidence Limits

Estimation method	Default columns	Additional columns for Complete
Actuarial	Interval Start Interval End Number Entered Number Censored Number Events Effective Number at Risk Conditional Probability of Event Conditional Probability of Survival Cumulative Survival Cumulative Failure Survival Standard Error	Interval Midpoint Conditional Prob. Event Standard Error Hazard Hazard Standard Error Density Density Standard Error Median Residual Lifetime MRL Standard Error Cumulative Survival Confidence Limits Hazard Confidence Limits Density Confidence Limits

If the Specify... option is chosen from the Survival table: Columns menu, the Survival Columns dialog box appears. This dialog box allows you to specify any combination of the columns listed above that correspond to the chosen estimation method. See [“Survival Columns dialog box,” p. 155.](#)

Survival table: Create dataset Enable this checkbox to create a survival table dataset. The contents of this dataset are the columns specified by the Survival table: Columns pop-up menu. By default, this option is disabled.

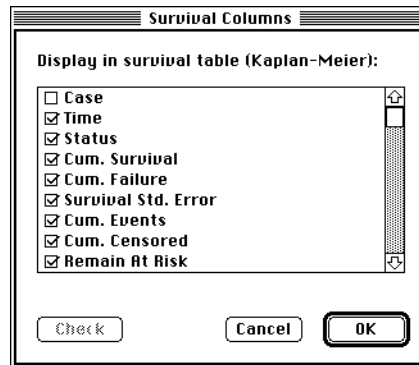
Survival table: Conf. level This text field allows you to set the confidence level used to compute the survival table confidence limits columns. These columns are: Cumulative Survival Confidence Limits for Kaplan-Meier estimates and Cumulative Survival Confidence Limits, Hazard Confidence Limits and Density Confidence Limits for actuarial estimates. The value entered here must be greater than 0 and less than 100. The default is 95 percent confidence limits.

Survival table: Sort by This pop-up menu gives you a choice of methods to sort the contents of the survival table. If the Time option (the default) is chosen, the rows in the survival table will be sorted by event time, from smaller to larger values. If the Case option is chosen, the rows of the survival table will be sorted by the ordering of cases in the dataset that holds the event time variable. The Sort by pop-up menu is available only if the estimation method is Kaplan-Meier.

Tests This button opens the Rank Tests dialog box, described under [“Rank Tests dialog box,” p. 156.](#)

Fewer choices This button opens the fewer choices version of the Survival: Nonparametric Methods dialog box, described above.

Survival Columns dialog box



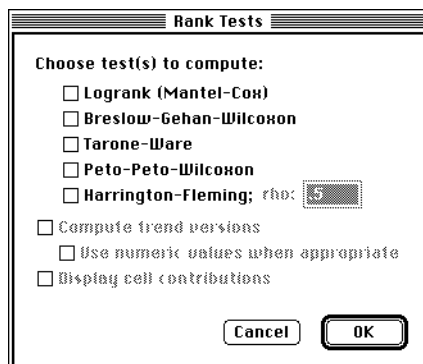
This dialog box is accessed by choosing the Specify option from the Survival table: Columns pop-up menu in the more choices version of the Survival: Nonparametric Methods dialog box.

Display in survival table (Kaplan-Meier/Actuarial) Items that are checked in this scrolling list will appear in the associated survival tables that appear in the view or that are saved to a dataset. An item is checked or unchecked by clicking in the box to the left of the item, or by selecting any combination of items, then clicking the Check/Uncheck button. Shift-click and Control-click (Windows) or Command-click (Macintosh) to select multiple items.

The choices available in this scrolling list depend on whether the estimation method is Kaplan-Meier or actuarial. These choices are summarized above in the description of the Survival table: Columns pop-up menu in the more choices version of the Survival: Nonparametric Methods dialog box.

Check/Uncheck This button allows you to check or uncheck items selected in the Columns to display scrolling list. If any of the selected items are unchecked, clicking this button will check them. If all of the selected items are checked, the button name changes to Unchecked; clicking it unchecks the selected items. This button is disabled if no items in the scrolling list are selected.

Rank Tests dialog box



This dialog box is accessed by clicking the Tests button in the Survival: Nonparametric Methods dialog box. The Compute trend versions and Display cell contributions checkboxes are enabled only when at least one of the rank tests is enabled.

Logrank (Mantel-Cox) Checking this box enables the logrank test. This test is sometimes called the Mantel-Cox or Mantel-Haenszel test.

Breslow-Gehan-Wilcoxon Checking this box enables the Breslow-Gehan-Wilcoxon test.

Tarone-Ware Checking this box enables the Tarone-Ware test.

Peto-Peto-Wilcoxon Checking this box enables the Peto-Peto-Wilcoxon test.

Harrington-Fleming Checking this box enables the Harrington-Fleming test. This automatically enables the rho: text field.

rho This text field allows you to enter a value for rho, the weight parameter used to calculate the Harrington-Fleming test. You may enter any non-negative value. With rho=0, the Harrington-Fleming test is equivalent to the logrank test. With rho=1, it is almost identical to the Peto-Peto-Wilcoxon test. This field is active only when the Harrington-Fleming checkbox is enabled.

Compute trend versions Checking this box enables trend versions of the chosen rank tests. Trend versions of these tests check for linear trends in the ordering of means for the specified group levels. The ordering of group levels for the trend tests is explained below under Use numeric values when appropriate. Enabling the Compute trend versions checkbox activates the Use numeric values when appropriate checkbox, and inactivates the Display cell contributions checkbox.

Use numeric values when appropriate When this checkbox is enabled, the numeric values (if present) in the group variable are used to order the group levels for the trend tests. This allows you to specify arbitrary (e.g., nonlinear) relationships among group levels. If Compute trend versions is checked and this checkbox is disabled, the case ordering of the group levels in the group variable (i.e., the order of the group levels in the dataset) is used for the trend tests.

Display cell contributions When this checkbox is enabled, separate tables are displayed for each of the selected rank tests, showing the contribution of each group or stratum level to an estimate of the overall chi-square statistic. This item is not available when the Compute trend versions checkbox is enabled.

Data requirements

Nonparametric survival analyses require only one continuous variable, the event time variable. The rows of the event time variable have the times at which the event or censoring occurred for each subject. This is usually the time of death or failure for uncensored observations, or the censor time for censored observations. The event time variable can have any positive value.

In addition to the continuous variable, a nominal censor variable is necessary if any of the event times are censored. This variable indicates whether each subject is censored (incomplete) or uncensored (complete). When the censor variable is not specified, all cases are assumed to be uncensored. When specified, the censor variable can be assigned only particular values. Use Uncensored (data type: string or category) or 0 (data type: integer or real) in the censor variable to indicate that a particular event time is *not* censored. Use any non-zero numeric value, or Censored to indicate that a particular event time *is* censored.

The Survival Analysis Preferences dialog box allows you to change this behavior so that 0 indicates *censored* observations. See [“Survival Analysis preferences,” p. 230 of Using StatView](#).

If there are treatment or study groups present in your data, these can be specified with an optional nominal variable, called the group variable. In general, separate survival estimates are calculated for each level of the group variable. Data from specified group levels are displayed in all graphs and are necessary for computation of any rank tests that are enabled.

In both the nonparametric and proportional hazards analyses, strata are specified with a nominal variable, called the stratification variable. All cases with the same value of the stratification variable are assigned to the same stratum. For nonparametric models, a stratification variable affects analyses in much the same way as does the group variable: separate survival estimates are calculated for each level of the stratification variable. However, the key difference between the effect of the stratification and group variables is how they are used in rank tests. In rank tests, data are pooled across strata to compare survival functions among group levels. Thus, the group variable provides the levels that are compared in the rank tests, while the levels of the stratification variable affect the computation of the rank tests, are not the groups that are compared. Strata may thus be regarded as sources of variation that must be accounted for, but are not themselves of particular interest.

Below is an example of one dataset with all variables properly formatted and ready for use in a nonparametric analysis.

	Event Time	Censor Variable	Group Variable	Stratification Variable
Type:	Integer	Category	Category	Integer
Source:	User Ente...	User Entered	User Entered	User Entered
Class:	Continuous	Nominal	Nominal	Nominal
Format:	•	•	•	•
Dec. Places:	•	•	•	•
1	63	Uncensored	Treatment	2
2	51	Uncensored	Control	3
3	45	Uncensored	Control	3
4	29	Uncensored	Treatment	2
5	26	Uncensored	Control	3
6	59	Uncensored	Control	2
7	63	Censored	Control	1
8	50	Uncensored	Treatment	3
9	37	Uncensored	Treatment	2
10	40	Uncensored	Treatment	1

The following table explains how to use the buttons in the variable browser to assign such variables to a nonparametric survival analysis.

Variable browser buttons	
Time	Select one event time variable (continuous), then click the Time button. Usage is indicated by a T in the variable browser. A second continuous variable assigned with the Time button is used as a new event time variable. This creates a new analysis using all previously specified censor, group, stratification and split by variables.
Censor	Select one censor variable (nominal), then click the Censor button. Acceptable values are 0 (must be Type: Integer or Real), or Uncensored (Type: String or Category) for uncensored observations, and any other numeric value or Censored to indicate censored observations. Usage is indicated by a C in the variable browser. NOTE: The Survival Analysis Preferences dialog box allows you to change the meaning of values in the censor variable so that 0 indicates censored observations. See “Survival Analysis preferences,” p. 230 of Using StatView. Each additional censor variable creates a new analysis using all other variables already specified.
Group	Select one group variable (nominal), then click the Group button. This creates separate estimates of the survival function for each group level. The group variable provides the levels that are compared in the rank tests. Usage is indicated by a G in the variable browser. Each additional group variable creates a new analysis using all other variables already specified.
Strata	Select one stratification variable (nominal), then click the Strata button. This creates separate estimates of the survival function for each stratum. Results for rank tests use data pooled across strata. Usage is indicated by the symbol # in the variable browser.
Split By	When you assign one or more split-by variables (nominal) to a nonparametric survival analysis, results are displayed separately for each cell defined in the split-by variable(s). Usage is indicated by an S in the variable browser. Each additional stratification variable creates a new analysis using all other variables already specified.

If you routinely create analyses first, then assign variables, you will find that the analyses will begin computing as soon as you have specified the event time variable. This may be unduly time consuming, especially if you assign censor, group, and stratification variables in sequence after the event time variable. To avoid this, do one of the following: (1) assign variables *first*, then create your analyses; (2) always assign the event time variable after all other variables have been assigned; or (3) disable the Recalculate box in the view before adding variables, then enable it once variable assignment is complete. If you choose to assign variables before creating the analysis, you can configure the variable browser by deselecting all results in the view, then clicking on any item within the Survival: Nonparametric Methods header in the analysis browser.

Results

Default Results

Default results are those created by selecting the Survival: Nonparametric Methods header in the analysis browser. They can also be selected individually by opening the Survival: Nonparametric Methods header.

Summary Table

This table is created by selecting Summary Table within the Survival: Nonparametric Methods header in the analysis browser.

# Obs	Gives the total number of observations for which all variable specifications are complete.
# Events	Gives the number of positive, uncensored event times.
# Censored	Gives the number of censored event times.
% Censored	Gives the percentage of valid observations in the event time variable that are censored.
# Missing	Gives the number of observations with missing variable specifications.
# Invalid	Gives the number of observations with invalid variable specifications, due, for instance, to negative values for the event time variable, or to uninterpretable values in the censor variable.
Other contents	Labels to the left of each row are group and stratum levels as specified by the group and stratification variables.

Survival Statistics Table

This table is created by selecting Survival Statistics Table within the Survival: Nonparametric Methods header in the analysis browser.

Estimate	Gives the estimated value of the cumulative survival function at the indicated percentile of the CDF for each group and stratum level, if specified. If estimation method is Kaplan-Meier, table also gives the estimate of the mean value of the cumulative survival function.
Standard Error	Gives the standard error about the estimate of the cumulative survival function at the indicated percentile for each group and stratum level, if specified. If estimation method is Kaplan-Meier, table also gives the standard error about the estimated mean of the cumulative survival function.
Other contents	Labels to the left of each row are the values of the estimated percentiles, corresponding to the first, second and third quartiles. Also gives the group and stratum names if specified.

Cumulative Survival Plot

This graph is created by selecting Cumulative Survival Plot within the Survival: Nonparametric Methods header in the analysis browser.

Plotted lines	These give the estimated value of the cumulative survival function. Each line represents the estimate for every level defined by the interaction of any specified group and stratification variables. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, all strata and group levels appear in a single graph; otherwise, functions for each stratum appear on separate graphs.
Plotted points	These optionally give the time and corresponding value of the cumulative survival function for censored and uncensored events. Display of uncensored and censored events is controlled by Survival plots show: checkboxes in the Survival: Nonparametric Methods dialog box.

Other Results

Survival Table

This table is created by selecting Survival Table within the Survival: Nonparametric Methods header in the analysis browser. Separate tables are created for each level of the group and stratification variables. The columns included in this table vary with the estimation method used and the setting of other parameters in the Survival: Nonparametric Methods dialog box. See [“Survival: Nonparametric Methods dialog box,” p. 152](#) and [“More choices,” p. 153](#).

The contents of this table can be saved to a dataset by enabling the Create dataset checkbox in the more choices version of the Survival: Nonparametric Methods dialog box. If this option is enabled, results from all strata and groups are saved to the same dataset.

Cumulative Hazard Plot

This graph is created by selecting Cumulative Hazard Plot within the Survival: Nonparametric Methods header in the analysis browser.

Plotted lines	These give the estimated value of the cumulative hazard function. Different lines/symbols represent the estimates for every level defined by the interaction of any specified group and stratification variables. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, then all strata and group levels appear in a single graph; otherwise, functions for each stratum appear on separate graphs.
---------------	---

Ln Cumulative Hazard Plot

This graph is created by selecting Ln Cumulative Hazard Plot within the Survival: Nonparametric Methods header in the analysis browser.

Plotted lines	These give the estimated values of the natural log of the cumulative hazard as a function of the natural log of the event time variable. Different lines/symbols represent the estimates for every level defined by the interaction of any specified group and stratification variables. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, all strata and group levels appear in a single graph; otherwise, functions for each stratum appear on separate graphs.
---------------	---

Hazard Plot

This graph is created by selecting Hazard Plot within the Additional Results subheader within the Survival: Nonparametric Methods header. It computes only if the estimation method is actuarial (the estimation method is specified in the Survival: Nonparametric Methods dialog box).

Plotted lines	These give the estimated value of the hazard function. Different lines/symbols represent the hazard estimates for every level defined by the interaction of any specified group and stratification variables. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, all strata and group levels appear in a single graph; otherwise, functions for each stratum appear on separate graphs.
---------------	--

Density Plot

This graph is created by selecting Density Plot within the Additional Results subheader within the Survival: Nonparametric Methods header. It computes only if the estimation method is actuarial (specified in the Survival: Nonparametric Methods dialog box).

Plotted lines	These give the estimated value of the density function. Different lines/symbols represent the density estimates for every level defined by the interaction of any specified group and stratification variables. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, all strata and group levels appear in a single graph; otherwise, functions for each stratum appear on separate graphs.
---------------	--

Censor Pattern Plot

This graph is created by selecting Censor Pattern Plot within the Additional Results subheader within the Survival: Nonparametric Methods header. It is also created if the Additional Results subheader is selected.

Plotted points	These give the incidence of all censored events by event time. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, censor patterns for all strata and group levels appear in a single graph; otherwise, censor patterns for each stratum appear in separate graphs.
----------------	---

Event Pattern Plot

This graph is created by selecting Event Pattern Plot within the Additional Results subheader within the Survival: Nonparametric Methods header. It is also created if the Additional Results subheader is selected.

Plotted points	These give the incidence of all uncensored events by event time. If Single graph with all strata is enabled in the more choices version of the Survival: Nonparametric Methods dialog box, event patterns for all strata and group levels appear in a single graph; otherwise, event patterns for each stratum appear on separate graphs.
----------------	---

Rank Test Table

This table is created whenever any rank tests are enabled in the Rank Tests dialog box, which is accessed by clicking the Tests button in the Survival: Nonparametric Methods dialog box. Results from all enabled tests are displayed in a single table.

Chi-Square	Gives the value of the chi-square statistic computed for each of the indicated tests.
DF	Gives the degrees of freedom associated with the chi-square statistic computed for each of the indicated tests.
P-Value	Gives the p value, or probability of Type I error, based on the chi-square value and the degrees of freedom for each of the indicated tests.
Other contents	If more than one rank test is enabled, row labels give the names of the corresponding tests.

Rank Test Cell Contributions Table

This table is created whenever the Display cell contributions checkbox is enabled in the Rank Tests dialog box. (This dialog box is accessed by clicking the Tests button in the Survival: Nonparametric Methods dialog box.) Separate cell contribution tables are created for each rank test enabled in the Rank Tests dialog box. Each table displays results for all strata and group levels. This table cannot be computed with trend versions of the rank tests.

Sum Weighted Obs.	Gives the sum of the weighted observed values for each cell defined by the interaction of the specified group and stratification variables.
Sum Weighted Exp.	Gives the sum of the weighted expected values for each cell defined by the interaction of the specified group and stratification variables.
Contribution	Gives the contribution of each cell to a conservative estimate of the overall chi-square statistic. Note that this estimate is not the same as the computed value of the overall chi-square statistic given in the Rank Test table.

Templates

The following templates provide nonparametric survival analysis results.

Survival Analyses	Actuarial Analysis	Survival summary, actuarial survival, and rank test tables; actuarial cumulative survival, density, hazard, and ln cumulative hazard plots.
	Kaplan-Meier Analysis	Survival, survival summary, and rank test tables; cumulative hazard plot, cumulative survival plot, and ln cumulative hazard plots.

Exercise

In this exercise, you will use the Kaplan-Meier method and rank tests to evaluate differences in survival patterns among groups of subjects. Suppose that you must analyze data from a randomized clinical trial that studied whether a certain treatment regimen administered to individuals suffering from a specific disease delayed the time until relapse. The dataset *AML Survival Data* in the Sample Data folder contains information on such a trial conducted by Embury et al. (1977) at Stanford University (cited in Miller, 1981). The investigators were concerned with the efficacy of maintenance therapy for acute myelogenous leukemia (AML). Initially, patients were treated by chemotherapy until remission. Then, these patients were randomized into two groups—a treatment group that received maintenance therapy and a control group that did not. Individuals in both groups were followed until they suffered a relapse, the event of interest in this example. The event time variable is defined as the length of time in remission, i.e., the time from entry into the study until relapse.

In this exercise, you will use the nonparametric procedures of this chapter to estimate the survival functions for both the therapy and control groups and compare survival properties across the groups.

- Open *AML Survival Data* from the Sample Data folder

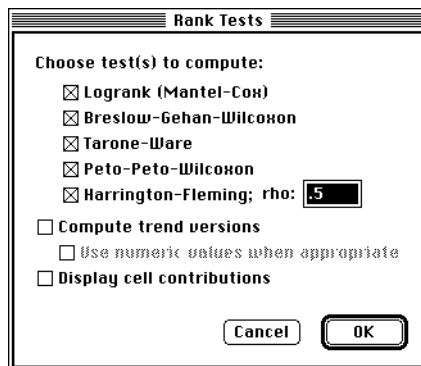
Scroll through the dataset to examine its contents. You will notice three variables: “Monitor time (weeks)” gives, for each patient, the elapsed time in weeks from entry into study until relapse or cessation of monitoring; “Censored?” is a binary variable with value 1 if the observation is censored or 0 if relapse was observed; and “Treatment” indicates whether each patient was in the control group or received maintenance therapy.

- Choose **New View** from the **Analyze** menu
- From the analysis browser, select **Survival: Nonparametrics** (This is equivalent to selecting the default results: **Summary Table**, **Survival Statistics**, and **Cumulative Survival Plot**.)
- Click **Create Analysis**

The **Survival: Nonparametric Methods** dialog box now appears on the screen. Notice that **Kaplan-Meier**—the estimation method you want—is selected by default. However, because you want to test whether there is a significant difference between the control and maintenance therapy groups, you also want to create some rank tests.

- Click the **Tests** button

- Check each of the five test checkboxes
(When the Harrington-Fleming checkbox is checked, you can select a value for rho; for this example, leave it at the default value of 0.5.)



- Click OK
- In the main dialog box, click OK

Empty result placeholders now appear on screen. Each result has below it a note instructing you to add variables to the analysis using the variable browser. You need to enter the event time, censor, and group variables to this analysis before it will compute.

Because the analysis will not compute until the event time variable is assigned, it is advisable to assign “Monitor time (weeks)” last to avoid computation after each variable assignment.

- In the variable browser, select Treatment and click Group
- Select Censored? and click Censor
- Select Monitor time (weeks) and click Time

A G usage marker indicates that Treatment is assigned as the grouping variable. Similarly, a C marker shows that Censored? is the censoring variable, and a T shows that Monitor time (weeks) is the time variable.



The summary table provides information on the number of patients (observations), observed deaths (events), and censored observations in each treatment group. Here we see that there are a total of 23 observations, of which 12 are in the control group and 11 are in the treatment group that received maintenance therapy. There is one censored observation in the control group, and four censored values in the maintenance therapy group.

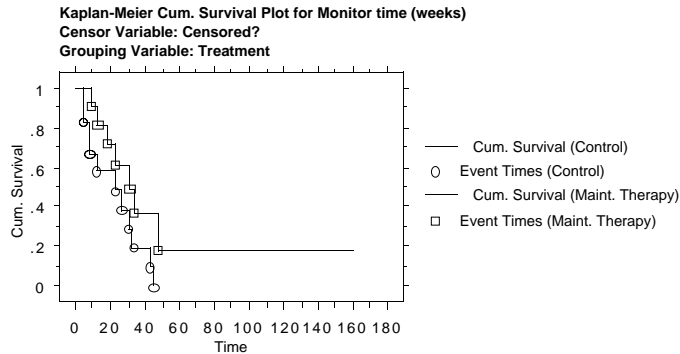
Survival Summary Table for Monitor time (weeks)

Sensor Variable: Censored?

Grouping Variable: Treatment

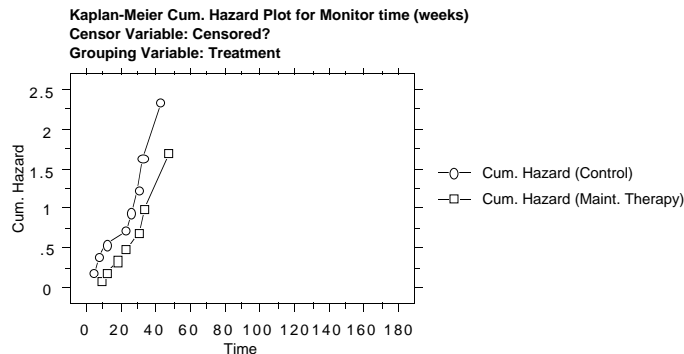
	# Obs.	# Events	# Censored	% Censored	# Missing	# Invalid
Control	12	11	1	8.333	0	0
Maint. Therapy	11	7	4	36.364	0	0
Total	23	18	5	21.739	0	0

Now, scroll down to the Kaplan-Meier survival plot. This graph shows separate Kaplan-Meier estimates for each treatment group. It is immediately apparent that the estimated survival curve for the maintenance therapy group lies above the estimated survival function for the controls, suggesting that individuals receiving therapy take longer to relapse.



Now let's take a look at a cumulative hazard plot.

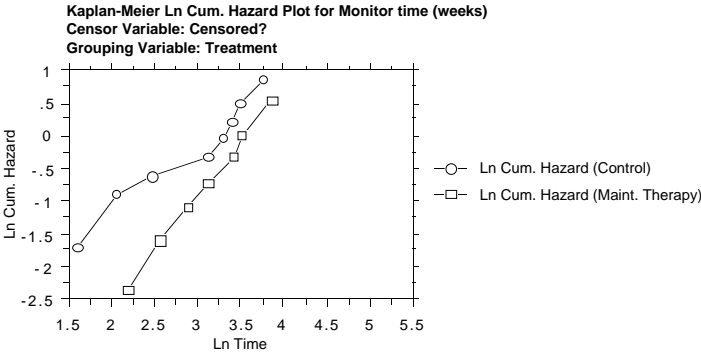
- Make sure at least one of the results is still selected
- From the analysis browser under Survival: Nonparametric Methods, select Cumulative Hazard Plot and click Create Analysis



The cumulative hazard plots for the two treatment groups show a pattern consistent with that in the cumulative survival plot. The cumulative hazard function for the control group is higher than that for the therapy group. Note that, for each group, the slope of the cumulative hazard plot becomes slightly steeper as time progresses, suggesting that the risk of relapses increases with time from entry into the study.

Now create the log cumulative hazard (or log minus log survival) plot.

- Make sure at least one of the results is still selected
- From the analysis browser under Survival: Nonparametric Methods, select Ln Cumulative Hazard Plot and click Create Analysis



This graph provides some clues about the sort of parametric model (discussed in the next chapter, [“Survival: Regression,” p. 167](#)) you might wish to use to model these data. Given the small sample size, a straight line approximates the data for both groups reasonably well. This suggests that a Weibull model may be appropriate to describe the variation in survival times of these groups. Furthermore, the slopes of the approximating lines are somewhat greater than 1, which suggests that the exponential model (which is a special case of the Weibull) may fit less well than the general Weibull model. We return to these considerations in the discussion of parametric models in the next chapter.

Although you must be careful not to draw too many conclusions from these graphs, some suggestive patterns do emerge. For example, the log cumulative hazard plots for the two groups are roughly parallel, indicating the underlying hazard functions are approximately proportional. However, there is a hint that the log relative hazard for the two groups—as measured by the vertical distance between the two curves—decreases over time. This might indicate that the beneficial effects of maintenance therapy decline after about 23 to 25 weeks, although a substantial therapeutic effect persists through the end of the common monitoring period of 40 weeks or so. Although there are not enough data to allow thorough examination of such conjectures, the analysis suggests valuable questions to be examined in a more definitive study.

Since the plots indicate that the hazard functions are approximately proportional, it is reasonable to test the equality of the survival estimates from the two groups using the rank tests.

Rank Tests for Monitor time (weeks)
Censor Variable: Censored?
Grouping Variable: Treatment

	Chi-Square	DF	P-Value
Logrank (Mantel-Cox)	3.396	1	.0653
Breslow-Gehan-Wilcoxon	2.723	1	.0989
Tarone-Ware	2.982	1	.0842
Peto-Peto-Wilcoxon	2.708	1	.0998
Harrington-Fleming (rho = .5)	3.019	1	.0823

The rank tests table gives the χ^2 statistics for the requested tests, with their associated (two-sided) p values. The results of these tests are qualitatively similar; each test suggests that the observed survival difference between the two groups may be real, although the comparisons are not statistically significant (at the 0.05 level) probably due to the small sample sizes.

Survival: Regression

This is the second of two chapters regarding StatView's survival analysis tools. The previous chapter, [“Survival: Nonparametric,” p. 143](#), introduces survival analysis in general and goes on to discuss StatView's Survival: Nonparametric analyses in particular. This chapter discusses the Survival: Regression analyses.

Regression methods

In the previous chapter, [“Survival: Nonparametric,” p. 143](#), we considered the use of nonparametric methods for estimation of the survival function and for comparison of these estimates among specified groups of interest. For cases in which differences among groups can be quantified—for example, by the dosage of a drug—or in which the relation between a variable and survival is of interest, it is natural to extend these techniques to regression models. An example of a variable—or **covariate**—that may be associated with survival is age at diagnosis of a certain disease.

Regression models are widely used with both continuous outcome variables (**linear models**) and outcome variables that are dichotomous or are counts (**generalized linear models**). For survival data that are subject to censoring as described in [“Introduction to survival analysis,” p. 143](#), a useful regression model that uses **time to event** as the dependent variable, is the **proportional hazards model**, sometimes called the **Cox model** because it was introduced by David Cox in 1972.

The proportional hazards regression model can be described as follows: Consider a covariate, denoted by Z . In the baseline group, defined by $Z = 0$, the hazard function is denoted by $\lambda_0(t)$, but its shape is unspecified in the model. For general levels of the covariate Z , the usual regression assumption is that the hazard for such levels is the baseline hazard *multiplied* by an exponential function of Z ; that is,

$$\lambda(t; Z) = \lambda_0(t)e^{\beta Z}.$$

With this assumption, note that the hazard for individuals with $Z = 1$ is e^{β} times the hazard in the baseline group *for all values of t* . In fact, the name proportional hazards model is derived from the fact that $e^{\beta Z}$ is constant over time; this ensures that hazard functions at different levels of the covariate are proportional, with the constant of proportionality dependent on the regression coefficient β and the difference in covariate values. The regression coefficient β is interpreted as the logarithm of the relative hazard between groups that differ in levels of Z by

one unit; alternatively, the relative hazard induced by increasing Z by one unit is e^β . Note that a positive regression coefficient β means that increases in the covariate Z are associated with increased hazard and thus with *shortened* expected event times. Conversely, a negative regression coefficient β indicates that increases in Z lead to a lower hazard and *longer* lifetimes.

In the chapter [“Survival: Nonparametric,” p. 143](#), we discussed nonparametric methods for producing survival function estimates, based on data (possibly censored) sampled from a certain population, without using *a priori* assumptions regarding the shape of the “true” survival function for that population. **Parametric models**, by contrast, rely on the additional assumption that we know an appropriate family of survival distributions for the population of interest. Since each of the families that we wish to use to describe the survival function has one or two unknown parameters that must be estimated, this approach is referred to as **parametric survival modeling**.

The disadvantage of parametric models is that distortion can be introduced into estimates of the survival function if the choice of a parametric family is not appropriate to the population under study. However, if our assumed parametric family provides an adequate description of the survival function, these estimates can be considerably more precise than those obtained from the nonparametric techniques of the chapter [“Survival: Nonparametric,” p. 143](#).

In a fashion similar to that used for proportional hazards models, you can apply regression methods with a parametric model. In the case of a parametric model, variation in survival distributions across covariate groups are specified by a regression equation as in a proportional hazards model. However, unlike a proportional hazards model, a parametric model assumes prior knowledge of the survival distribution at all levels of the covariate, up to a finite number of unknown parameters; these functions are left unspecified in a proportional hazards model. For example, with an exponential parametric model, the hazard function is assumed to be constant for any value of the covariate, Z , with levels of the constant hazard function determined by the specific value of Z . Again, the benefit of this kind of parametric regression model is increased precision for estimates of regression coefficients; the disadvantage is that answers may be biased if your choice of a parametric family (in this case, the exponential model) is incorrect.

One consequence of the trade-off between parametric and nonparametric approaches is the need to carefully examine whether a parametric model adequately fits the observed data. We will elaborate on methods for achieving this in the following discussion.

Discussion

Proportional hazards model

In examining the results from fitting a proportional hazards regression model to survival data, we follow procedures similar to those used for more familiar regression models. It is important to understand how to interpret the reported estimates of model parameters, how to test hypotheses regarding these parameters, and how to assess the adequacy of the model.

Parameter estimates in the proportional hazards model

Estimates of covariate coefficients, given by the vector $\hat{\beta}$, are interpreted as estimates of the log relative hazard associated with a unit increase in the associated covariate, holding all other covariates fixed. It follows that $e^{\hat{\beta}}$ gives the relative hazard for two groups that differ only in the relevant covariate, and then only by one unit. Thus, it is helpful to use a suitable choice of scale for the covariate so that a unit change provides a meaningful comparison. The relative hazard is constant over time, explicitly reflecting the proportional hazards assumption.

There is no explicit intercept term in a proportional hazards model. The role of the intercept is played by the baseline hazard function, $\lambda_0(t)$, which describes the hazard for the group whose covariate values are all set to zero. Information on baseline survival properties is provided by the estimate of the baseline cumulative survival function, S_0 .

It is useful to examine various plots associated with the baseline estimate of the survival function, specifically a cumulative survival plot, a cumulative hazard plot, and a plot of the natural log of the cumulative hazard versus log time. The interpretation of these plots is analogous to the interpretation of the single group plots discussed in the chapter [“Survival: Nonparametric,” p. 143](#). These plots can also be used to assess the plausibility of certain parametric models for the baseline hazard, and thus they may suggest the use of parametric regression models discussed under [“Parametric models,” p. 171](#).

Note that the survival and hazard functions for groups at all levels of the covariates are directly related by the proportional hazards assumption to the baseline versions of those functions. Specifically, the proportional hazards assumption entails that, whatever the value of the covariates, the shape of the hazard function is the same, with changes only in absolute level. Therefore, it is often easier to interpret results if the covariates are coded so that the baseline group represents a meaningful level of the covariates. For example, if patient age upon entry into a study is used as a covariate, it would be helpful to record age as the difference in age from a baseline value, such as 50 years old, rather than to record age on its original scale. If you were to use values of age in actual years, the baseline group would refer to individuals with age zero at the time of entry into the study, which would not provide a meaningful reference group.

Stratified proportional hazard models

In many cases, the proportional hazards assumption is reasonable *within* certain groups of the population, referred to as **strata**, but not for purposes of comparing individuals from different strata. The model can be extended to accommodate such cross-strata comparisons by allowing the baseline hazard function to vary across strata. Then, the hazard function for the i th stratum, specified by the model, is given by $\lambda_{0i}(t)e^{\beta Z}$. Estimates of regression coefficients are interpreted just as in the unstratified case. Now, however, estimates of the baseline cumulative survival function (associated with λ_{0i}) are provided for each stratum.

Significance tests and confidence intervals

Standard hypotheses of interest in the context of a proportional hazards regression model concern the association of a specific covariate or group of covariates with survival. For example, in

a randomized clinical trial, a primary question is whether treatment is associated with longer survival. Such qualitative hypotheses are accommodated in the model by setting the relevant regression coefficients β to zero. As in linear regression models, the null hypothesis $\beta = 0$ can be examined by any one of three procedures—the *Wald test*, the *score test*, and the *likelihood ratio test*. These tests should yield similar results in large samples; in small samples, the likelihood ratio procedure is usually the method of choice.

Since relative hazard comparisons, as measured by e^{β} , are easier to interpret, *confidence intervals* are given for e^{β} for each covariate in the regression model.

Residual plots

An important part of regression modeling is the assessment of how well the regression model fits the data. Typically, the goodness of fit of a proportional hazards model is examined with plots of residuals.

A graph of so-called **martingale residuals** plots those residuals against the fitted value of the linear predictor (i.e., $\beta'Z$, which is the sum of the products of each covariate multiplied by the respective regression coefficient), for each case in the dataset. Similarly, the graph of **deviance residuals** plots those residuals against the fitted value of the linear predictor and provides an alternative view of the goodness of fit of the entire model to the data. Residuals of this kind are analogous to residuals in linear regression, quantifying, for each data point, the difference between an observation and its predicted value based on the fitted model. StatView also allows you to save these residuals, as well as the **score residuals**, to a dataset. Once saved, residuals can also be graphed against covariates singly or against the event time variable.

Martingale, deviance, and score residuals

The values of martingale residuals lie between $-\infty$ and 1. If the fitted model is adequate, the martingale residuals are uncorrelated with each other and have an average value of zero. Unlike residuals derived in linear models, however, martingale residuals are not symmetrically distributed about zero. Therefore, some care and experience is necessary in examining these plots. Another type of residuals, available only for proportional hazards models, are deviance residuals, which can span the entire range of real values and are much more symmetrically distributed about zero if the fitted model is adequate. For these reasons, plots of deviance residuals may be easier to interpret than plots of martingale residuals.

Unlike martingale and deviance residuals, score residuals are computed *for each covariate*—thus, the score residual associated with a specific covariate directly reflects the adequacy of the model to describe the association of that particular covariate with the risk of failure. Score residuals tend to be closer to zero for censored observations; for uncensored cases, they represent the deviation of the observed covariate value from a weighted average of covariate values in the risk set at the observed failure time for that case.

Interpreting residuals

In general, these residual plots are interpreted very similarly to residual plots in simple linear regression analysis. First, the residual plot should be examined to see if there are unusually large values of the residuals for certain data points. These cases might be examined to determine whether they possess any unusual characteristics. Furthermore, it is often instructive to fit the regression model with such data points excluded, to assess the effect of these cases on estimated regression coefficients and relevant hypothesis tests.

Second, unless you detect a systematic pattern of the residuals in these plots, you can assume that the model fits the data reasonably well. The appearance of certain patterns in residual plots does not always mean that the model is completely inappropriate, however; it may suggest that slight modifications to the model may improve the fit. For example, in the case of a single covariate Z , curvature in the residual plot may indicate that it is preferable to use a transformed scale for Z or that one should include polynomial terms such as Z^2 in the model. Similarly, clusters of large residuals for small values of $\beta'Z$ suggest that the model might be inadequate when t is large or small. Patterns in residual plots can also indicate that the hazard functions are not proportional among all levels of a particular covariate, i.e., that the assumption of proportional hazards is inappropriate. In such cases it can be worthwhile to stratify the model by the problematic covariate if this covariate is not of primary interest (i.e., if parameter estimates for this covariate are unnecessary). Viewed in this way, stratification allows you to “get around” the proportional hazards assumption by stratifying the model on those variables for which the proportional hazards assumption does not hold. For a stratified model, the residuals from different strata are plotted with different symbols in the same graph. This allows you to evaluate the adequacy of the model within each stratum.

When several covariates have been included in the model, it is useful to save the residuals to a dataset so that they can be plotted against each covariate in turn, using bivariate plots. In addition, a plot of residuals against the case (row) number or identification number can be useful (for instance, for determining if there is a lack of fit for individuals entering the study at certain times). Finally, plotting the residuals against a covariate not included in the model is valuable for helping you determine whether you should add the covariate to the model. In particular, if a new covariate should be added to the model to improve the fit, residual plots against this covariate should display some pattern or correlation; absence of a pattern suggests that the new covariate will add little to the model's ability to explain the observed failure patterns in the data.

If the model accurately captures the covariate's effect on failure, then the score residuals for that covariate should appear as a random pattern about zero. Note that if the covariate is an indicator variable associated with a discrete variable, such as gender, then the score residuals for that covariate against failure time will appear as two horizontal bands on either side of zero, with all score residuals for censored observations tending to be closer to zero.

Parametric models

We now turn to the use of parametric models to describe both survival properties of a single population and the variation of these properties across levels of covariates. Since the way to fit

a single parametric population model in StatView is to create the appropriate parametric regression model with no covariates, we begin with such models.

StatView allows you to use various parametric families to describe survival functions either for an entire population or as part of a regression model. The choice of an appropriate model depends on prior knowledge of the survival process under investigation, useful interpretation of model parameters, and the model's ability to adequately fit the observed data. Understanding the basic properties of different parametric families is helpful for suggesting a reasonable initial choice of a parametric model.

Four parametric models are available in StatView; these are the exponential family, the Weibull family, the lognormal family, and the loglogistic family. An **exponential** distribution requires specification of a single parameter and has a constant hazard function. The other three families require two parameters to describe their properties and each possesses more flexible hazard functions than does the exponential family. The **Weibull** hazard function is either strictly increasing, strictly decreasing, or constant. If the hazard is constant, then the Weibull reduces to an exponential model; that is, the exponential model is a special case of the Weibull family. Further possibilities are allowed in the other two families, the lognormal and the loglogistic. In the **lognormal** family, the hazard function increases from 0 at $t = 0$ to a maximum and then decreases towards 0 again as t becomes large. For the **loglogistic** model the hazard function either always decreases or resembles the lognormal in that it can increase to a maximum before declining back to zero for large t .

Parametric regression models

Each of the four parametric families mentioned above can be extended to account for the effects of covariates through use of a parametric regression model. It is standard in parametric models to use a model somewhat different from the proportional hazards model introduced in [“Proportional hazards model,” p. 168](#), although, as discussed below, in some cases the models coincide.

In particular, for each of the parametric models, it is possible to write the failure time random variable in the form $\log T = \mu + \sigma W$, where the **error** variable W has mean zero and conforms to the distribution of the specific model under consideration. When covariates are present, this suggests the regression model

$$\log T = \mu + \beta'Z + \sigma W,$$

where Z is the vector of covariates, yielding a regression model that is linear in the *logarithm* of time to an event. For each of the four families, we can fit this regression model using standard parametric techniques.

These regression models—often referred to as **accelerated failure time models**—are log-linear in T , so the regression coefficients β have the following interpretation: If β_j is the coefficient corresponding to the j th covariate Z_j , then a unit increase in Z_j induces a multiplicative change of $e^{\beta_j Z_j}$ in the time to failure, if all other covariates are fixed. That is, if T_0 is the random variable measuring time to failure when $Z_j = 0$, then

$$T(Z_j) = e^{\beta_j Z_j} T_0$$

gives the time to failure at an arbitrary level of Z_j , again all other things being equal. Note that if the regression coefficient β_j is positive, increases in the covariate Z_j reflect increases in the time to an event and therefore reflect *increasing* lifetimes. This is in contrast to proportional hazards models, for which increasing values of covariates with positive coefficients imply *decreasing* lifetimes.

For exponential and Weibull families, the accelerated failure time regression model accommodates a different parameterization and interpretation, which are consistent with the formulation of the proportional hazards model. We will refer to this alternative parameterization as the **relative hazard parameterization**, as contrasted with the **log time parameterization** that StatView uses for all parametric models. This relative hazard parameterization for Weibull and exponential families is as follows. For these families, using the model $\log T = \mu + \beta'Z + \sigma W$ is equivalent to assuming that, for individuals with covariate value Z , the hazard function is $\lambda_0(t)e^{\gamma'Z}$, for a suitable choice of γ , the hazard function $\lambda_0(t)$ takes either the exponential or Weibull form. Thus, the exponential and Weibull accelerated failure time regression models are special cases of the proportional hazards model. The difference in the parametric analyses is that they take advantage of a specified shape for the baseline hazard function, whereas this is left unspecified in the general version of the proportional hazards model. Note that in the relative hazard parameterization, the regression coefficients differ from those provided by the log time parameterization. Specifically, for a given covariate, the regression coefficients provided by the log time parameterization (β) equal the negatives of the relative hazard coefficients (γ) for exponential models, and, for Weibull models, the coefficients (β) equal the negatives of the relative hazard coefficients (γ) multiplied by the scale parameter (σ). In either case, the coefficients provided by the two alternative parameterizations will have different signs; this is because a covariate associated with *increasing* the time to failure T must consequently *reduce* the hazard or risk of failure and vice versa. To reiterate, StatView provides only the log time parameter estimates for Weibull and exponential models. The conversion given above, however, allows you to compute relative hazard estimates for comparison with fitted proportional hazards models. The Weibull family, including the exponential as a special case, is the only parametric family for which the accelerated failure time model and the proportional hazards model are consistent. For example, in the accelerated failure time model based on the loglogistic or lognormal families, the hazard functions for different levels of the covariates are not proportional.

Fitting a parametric survival family for a single population

In many preliminary data analyses, it will be valuable to fit a specific parametric family to survival data without adjusting for covariates. This is accomplished by fitting the appropriate regression model without adding any covariates. Thus, to model a sample of data assumed to be taken from a lognormal population, you would create the lognormal regression analysis without assigning any covariates with the variable browser.

Significance tests and confidence intervals

As discussed above, a parametric regression model is fit using standard parametric techniques. These methods yield estimates of the log time regression coefficients β and, if appropriate, estimates of the scale parameter necessary to fit the modeled distribution for the error term W

in the regression model. Significance tests for null hypotheses involving one or more of the regression coefficients are carried out in exactly the same fashion as for proportional hazards models. If requested, confidence intervals are given for coefficients e^{β_j} , which describe the multiplicative effect on the time to event of a unit change in the corresponding covariate Z_j , if all other covariates are held fixed. Neither confidence intervals nor coefficient significance tests are available for stepwise models.

Plots for checking your models

As for proportional hazards models, it is important to assess whether a parametric regression model adequately fits observed survival data. Specifically, you might determine that the additional assumption of a specific parametric model for the error term in the log-linear model is inappropriate after examination of the data.

In the single group situation, you can assess the selected parametric family in a variety of ways. First, it is informative to compare the fitted survival curve based on the parametric assumption to the model-free Kaplan-Meier estimate discussed in the chapter [“Survival: Nonparametric,” p. 143](#). Discrepancies in these two estimates of the underlying cumulative survival function may indicate that the parametric model is inadequate and may suggest why the model fits poorly and thus indicate a more suitable choice of parametric family. If preferred, similar comparisons can be carried out using plots of the cumulative hazard or log cumulative hazard functions. In the case of the Weibull family, the log cumulative hazard plot against $\log(t)$ will be linear; this allows it to be easily compared with the same plot based on the Kaplan-Meier estimator. In the special case of an exponential model, the log cumulative hazard plot should be both linear and have a slope of one. For example, if the Kaplan-Meier version of the log cumulative hazard plot is roughly linear, but with a slope other than one, then an exponential model is inappropriate, although a Weibull model might be suitable.

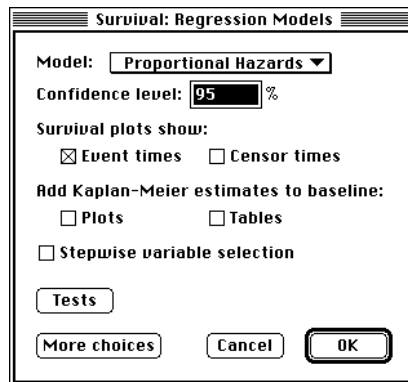
As with proportional hazards models, you can estimate the baseline cumulative hazard plot as well as the regression coefficients for parametric regression models. This is a model-based estimate of the survival function for the group in which all covariates are zero. As in the single group case, these plots, and their cumulative hazard counterparts, can be examined to investigate the suitability of the parametric assumption in the accelerated failure time regression model.

To graphically check how well a parametric model describes observed survival data, it is helpful to create **quantile plots**. The idea of these plots is to compare the **observed quantiles** of the distribution of the data points with **estimated quantiles** based on the particular parametric model. Specifically, in the case of a single population (i.e., a model without covariates), the quantile plot graphs the parametric estimate of the quantiles against the observed values, the latter obtained from the Kaplan-Meier estimate of the survival function. If the parametric model is appropriate, the plotted points should lie, approximately, on a straight line with a slope equal to one. In the regression setting (i.e., for models with covariates), the same technique can be used but now applied to standardized residuals of the observed survival times in order to remove the effects of the covariates. Again, if the particular parametric family underlying the regression model is reasonable, the plotted points should be close to a straight line with a slope of one.

StatView also allows you to create **martingale residual** plots and to save these residuals to a dataset. Again, martingale residuals can be investigated to expose unusually large residuals and can be saved and plotted against observed failure times, covariates included in the model, and covariates not included that are potential candidates to add to the model.

Dialog Box Settings

Survival: Regression Models dialog box



The settings in this dialog box control the computation and display of all results within the Survival: Regression Models header in the analysis browser. By default, this dialog box is accessed by clicking the Create Analysis button after choosing any result within the Survival: Regression Models header in the analysis browser. If you prefer, the more choices version of this dialog box can be made the default by changing the setting of the Survival Analysis Preferences dialog box (see [“Survival Analysis preferences,” p. 230 of Using StatView](#)). The fewer choices version of this dialog box also can be accessed by clicking the Fewer choices button in the more choices version of the Survival: Regression Models dialog box.

Model This pop-up menu allows you choose among various regression models. The available models are: Proportional Hazards (the default), Exponential, Weibull, Lognormal and Loglogistic.

Confidence level This text field allows you to set the confidence level used to compute the confidence intervals displayed in the Confidence Intervals Table. The default is 95 percent confidence limits. This option is inactive for stepwise models.

Survival plots show These checkboxes allow you to specify the data that are displayed on any cumulative survival plots that are created. If the Event times checkbox is enabled (the default), symbols indicating the occurrence of uncensored events are plotted on cumulative survival plots. If the Censor times checkbox is enabled, symbols indicating the occurrence of censored events are plotted on cumulative survival plots.

Add Kaplan-Meier estimates to baseline These checkboxes allow you to add the corresponding Kaplan-Meier estimates to baseline regression model plots and tables. Enabling the Plots checkbox adds the Kaplan-Meier estimates to the Baseline Cumulative Survival Plot, Baseline

Cumulative Hazard Plot, and Baseline Ln Cumulative Hazard Plot. Enabling the Tables checkbox adds the Kaplan-Meier estimates of the cumulative survival, cumulative hazard, and the natural log of the cumulative hazard to the Baseline Survival Table. It also adds these quantities to the baseline survival dataset, if the Create baseline survival dataset checkbox is enabled in the more choices version of this dialog box.

Stepwise variable selection Checking this box enables forward stepwise variable selection. Checking this option is equivalent to choosing Forward from the Stepwise pop-up menu in the more choices version of the Survival: Regression Models dialog box. Furthermore, this checkbox is also enabled if Backward is chosen from the Stepwise pop-up menu in the more choices version of the Survival: Regression Models dialog box. For more control over stepwise model parameters, click the More Choices button.

Tests If two or more covariates have already been specified, clicking this button opens the Joint Significance Tests dialog box; otherwise, a warning dialog box appears. The Joint Significance Tests dialog box is described under [“Joint Significance Tests dialog box,” p. 179](#).

More choices Clicking this button opens the more choices version of the Survival: Regression Models dialog box. This dialog box is described immediately below.

More choices

Additional options are available in the More choices dialog box. By default, this dialog box is accessed by clicking the More choices button in the fewer choices version of the Survival: Regression Models dialog box. If you prefer, this more choices version of the Survival: Regression Models dialog box can be made the default by changing the setting of the Survival Analysis Preferences dialog box (see [“Survival Analysis preferences,” p. 230 of Using StatView](#)).

Stepwise This pop-up menu allows you to choose among standard (non-stepwise), forward selection, and backward selection stepwise regression models. If the Don't use option is chosen (the default), the standard model is enabled. If the Forward option is chosen, a forward stepwise model is enabled. If the Backward option is chosen, a backward stepwise model is enabled. If either stepwise option is enabled, the Enter p and Remove p text fields are activated. Either stepwise model deactivates the Confidence level and Tests items.

Enter p This text field allows you to set the p value that determines entry of specified covariates in a stepwise model. The unentered covariate with the smallest p value below this critical value is entered into the model on the next step. This Enter p value must be between 0 and 1,

and less than or equal to the Remove p value. The default value is 0.05. This text field is active only after a stepwise model is enabled.

Remove p This text field allows you to set the p value that determines removal of specified covariates in a stepwise model. Except for forced covariates (which are never removed from a model) the covariate in the model with the largest p value greater than this critical value is removed from the model on the next step. This value may be any value greater than or equal to the Enter p value and less than 1. The default value is 0.05. This text field is active only after a stepwise model is enabled.

Add columns to dataset This pop-up menu allows you to save specific computed values to the dataset as analysis generated variables. All computed values are evaluated at the corresponding values of the event time variable and all covariates in the model.

If the None option is chosen (the default), no columns are saved to the dataset. The following table indicates those values saved to the dataset with the Default and Complete options enabled:

Default columns	Additional columns for Complete
Regression estimate of the cumulative survival function	KM estimate of the cumulative survival function
Regression estimate of the cumulative hazard function	KM estimate of the cumulative hazard function
Regression estimate of the natural log of the cumulative hazard function	KM estimate of the natural log of the cumulative hazard function
	Linear predictor of the regression estimate
	Standard error of the linear predictor
	Martingale residuals
	Deviance residuals (proportional hazards models only)
	Score residuals (proportional hazards models only)

Note that all Kaplan-Meier estimates are computed as if there were no covariates in the model.

By choosing the Specify... option, the Survival Columns dialog box appears, allowing you to choose which columns to save to the dataset; see [“Survival Columns dialog box,” p. 178.](#)

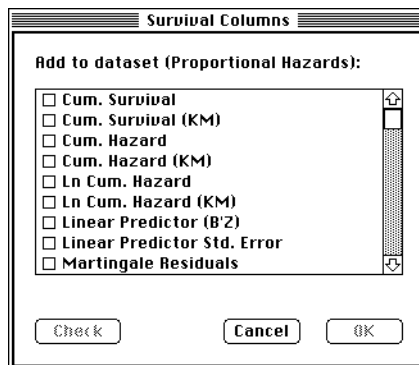
Create baseline survival dataset Checking this box creates a separate dataset with all of the computed values in the baseline survival table. The contents of this dataset are partially controlled by the Add Kaplan-Meier estimates to baseline: Tables checkbox.

Tests If two or more covariates have already been specified, clicking this button opens the Joint Significance Tests dialog box; otherwise, a warning dialog box appear; see [“Joint Significance Tests dialog box,” p. 179.](#)

Est. Pars. Clicking this button opens the Estimation Parameters dialog box; see [“Estimation Parameters dialog box \(proportional hazards\),” p. 180](#) and [“Estimation Parameters dialog box \(parametric models\),” p. 181.](#)

Fewer choices Clicking this button opens the fewer choices version of the Survival: Regression Models dialog box, discussed above.

Survival Columns dialog box



This dialog box is accessed by choosing the Specify option from the Add columns to dataset pop-up menu in the more choices version of the Survival: Regression Models dialog box.

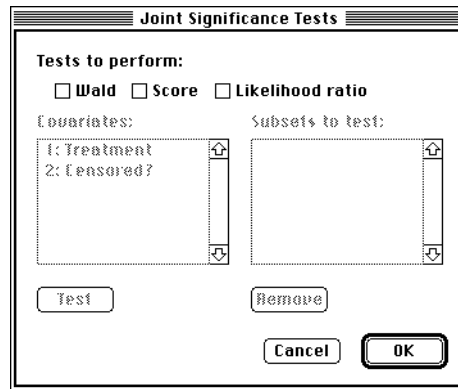
Add to dataset (Proportional Hazards/Parametric Models) Items that are checked in this scrolling list will appear in the dataset containing the event time variable. An item is checked or unchecked by clicking in the box to the left of the item, or by selecting any combination of items, then clicking the Check/Uncheck button. Shift-click and Control-click (Windows) or Command-click (Macintosh) to select multiple items.

Note that many of the items in this list are followed by “(KM),” which indicates the Kaplan-Meier estimate of the preceding quantity. Those items with no parenthetical suffixes are estimates of the type indicated in the Model pop-up menu in the Survival: Regression Models dialog box.

If the chosen regression model is proportional hazards, this scrolling list includes Deviance Residuals and Score Residuals. If the chosen regression model is one of the parametric models, deviance and score residuals are not available.

Check/Uncheck This button allows you to check or uncheck items selected in the Add to dataset (Proportional Hazards/Parametric Models) scrolling list. If any of the selected items are unchecked, clicking this button will check them. If all of the selected items are checked, the button name changes to Uncheck; clicking it unchecks the selected items. This button is disabled if no items in the scrolling list are selected.

Joint Significance Tests dialog box



This dialog box is accessed by clicking the Tests button in either the fewer or more choices versions of the Survival: Regression Models dialog box. *Important:* this dialog box does not appear unless two or more covariates have already been specified. Also note that joint significance tests are not computed for stepwise models.

Note that joint significance tests only evaluate whether combinations of covariates, or levels of nominal covariates, make a significant contribution to a particular regression model. If instead you are interested in testing for differences among covariate coefficients, or weighted combinations of these coefficients, please see [“How can I make comparisons among coefficients for linear hypotheses?”](#) p. 248 of *Using StatView*.

Tests to perform These checkboxes allow you to choose among any combination of Wald, Score and Likelihood ratio tests of the hypotheses appearing in the Subsets to test scrolling list.

Covariates This scrolling list shows all covariates (each preceded by a number) that have been previously specified using the variable browser. The covariates you select from this list are used to construct the joint significance tests. This list is inactive unless one or more of the Tests to perform checkboxes is enabled.

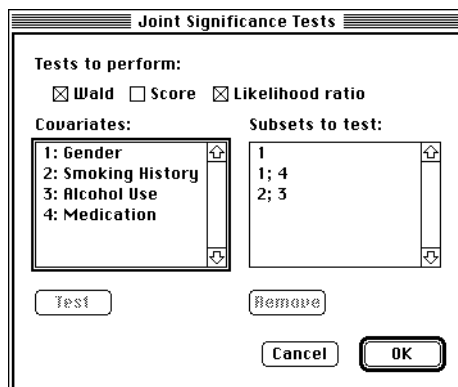
Test When this button is clicked, the covariates selected in the Covariates scrolling list are added as a defined subset to the Subsets to test scrolling list. If no covariates are selected, this button is inactive.

Subsets to test This scrolling list shows all defined combinations of covariates that are to be evaluated by the tests enabled with the Tests to perform checkboxes. If none of the tests is enabled, this list is inactive.

Remove Clicking this button removes the covariate subsets selected in the Subsets to test scrolling list. If no covariate subset is selected, this button is inactive.

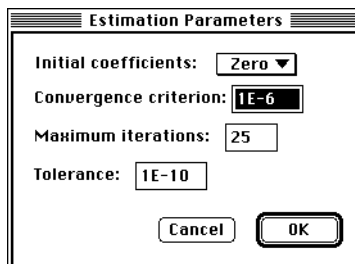
To construct a subset to be evaluated by the selected tests, Control-click (Windows) or Command-click (Macintosh) those covariate(s) in the Covariates scrolling list whose joint significance you wish to test, then click the Test button. The numbers preceding these covariates then appear on a single line in the Subsets to test scrolling list. Each line in the Subsets to test scrolling list represents a separate hypothesis that will be evaluated by the tests enabled with

the Tests to perform checkboxes. An example, showing that three different hypotheses have been defined, is illustrated below.



In this example, the three hypotheses that have been specified are in the Subsets to test scrolling list. The first hypothesis tests the significance of excluding the covariate Gender from the model. It was specified by selecting “1: Gender” from the Covariates scrolling list, then clicking the Test button. The second hypothesis tests the significance of excluding both “Gender” and “Medication” from the model. It was specified by command-clicking on “1: Gender” and “4: Medication” from the Covariates scrolling list, then clicking the Test button. The third hypothesis tests the significance of excluding both “Smoking History” and “Alcohol Use” from the model. It was specified by shift-clicking on “2: Smoking History” and “3: Alcohol Use” from the Covariates scrolling list, then clicking the Test button. All three hypotheses will be evaluated with both the Wald and likelihood ratio tests.

Estimation Parameters dialog box (proportional hazards)



This dialog box is accessed by clicking the Est. Pars. button in the more choices version of the Survival: Regression Models dialog box. It appears only if Proportional Hazards is selected from the Model pop-up menu. If one of the parametric models is chosen, the parametric models version of this dialog box appears.

Initial coefficients This pop-up menu allows you to specify initial values for the model coefficients before the iterative fitting process begins. If the Zero option is chosen (the default), all coefficients are initially set to 0. If the Specify... option is chosen and at least one covariate has already been specified, the Coefficient Initial Values dialog box appears; otherwise, a warn-

ing dialog box appears. The Coefficient Initial Values dialog box allows you to set specific initial values for each regression coefficient; see [“Coefficient Initial Values dialog box,” p. 182](#).

Convergence criterion This text field allows you to set a value for the iterative convergence criterion. If the relative change in the partial likelihood function between iteration steps is less than this value, the model-fitting process stops. You may enter any value greater than 0 and less than 1; the default value is 0.000001. Smaller values of this parameter result in the same number or more iterations as previous fits of the same data, while larger values result in the same number or fewer iterations.

Maximum iterations This text field allows you to set a value for the maximum number of iterations for the fitting process. The fitting process stops after this number of iterations, even if the convergence criterion has not been satisfied. You may enter any non-negative integer in this field. The default value is 25.

Important: if large values are entered for Maximum iterations, or very small values for Convergence criterion, the time required to fit a model may increase significantly. We suggest that you edit these values with caution.

Tolerance This text field allows you to set a value for the sweep tolerance. When a “pivot” is less than this tolerance, the model-fitting process stops and an error message appears. The tolerance value is useful for detecting multicollinearity among independent variables. You may enter any value greater than 0 and less than 1. The default value is 0.0000000001 (i.e., 10^{-10}). Higher tolerance values reduce the model’s tolerance of colinearity among independent variables and make abortion of the fitting process more likely. We suggest you edit this value cautiously.

If you wish to see how well specific coefficients fit your data, you can specify coefficient values using the Specify... option in the Initial coefficients pop-up menu, then set Maximum iterations to 0, then run the model.

Estimation Parameters dialog box (parametric models)

This dialog box is accessed by clicking the Est. Pars. button in the More choices version of the Survival: Regression Models dialog box. It appears only if one of the four parametric models is selected from the Model pop-up menu. If Proportional Hazards is chosen from the Model

pop-up menu, the proportional hazards version of this dialog box appears (see [“Estimation Parameters dialog box \(proportional hazards\),”](#) p. 180).

Initial coefficients This pop-up menu allows you to set the values for the model coefficients before the iterative fitting process begins. If the OLS option is chosen, all coefficients are initially set to their ordinary least squares estimates. If the Zero option is chosen (the default), all coefficients are initially set to 0. If the Specify... option is chosen and at least one covariate has already been specified, the Coefficient Initial Values dialog box appears; otherwise, a warning dialog box appears. The Coefficient Initial Values dialog box allows you to set specific initial values for each regression coefficient. This dialog box is described below.

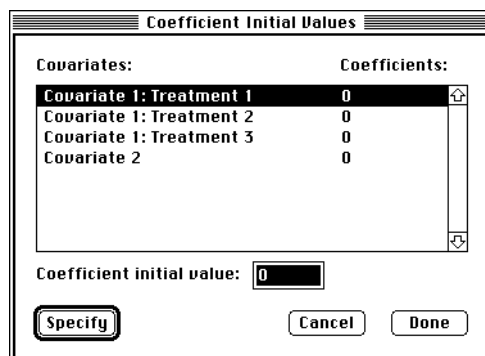
Intercept This pop-up menu allows you to set a value for the model intercept term. If the Initial OLS option is chosen, the intercept is initially set to its ordinary least squares estimate. If the Initial option is chosen (the default), you may enter an initial value for the intercept in the text field following the pop-up menu. The default initial value is 0. If the Fixed option is chosen, you may enter a fixed value for the intercept, which the fitting procedure will not change.

Scale This pop-up menu allows you to set values for the model scale term. If the Initial OLS option is chosen, the scale term is set to its ordinary least squares estimate. If the Initial option is chosen (the default), you may enter an initial value for the scale term in the text field following the pop-up menu. The default initial value is 1. If the Fixed option is chosen, you may enter a fixed quantity for the scale term, which the fitting procedure will not change. This pop-up is inactive if the Model pop-up menu is set to Exponential.

Convergence criterion, Maximum iterations, and Tolerance These text fields all function identically to those in the proportional hazards version of this dialog box. For more information about these parameters, see [“Estimation Parameters dialog box \(proportional hazards\),”](#) p. 180.

Don't transform time variable Enabling this checkbox prevents the parametric fitting procedure from log transforming the event time variable before fitting the model. This allows the use of previously log transformed event time variables in accelerated failure time models.

Coefficient Initial Values dialog box



This dialog box is accessed by choosing the Specify... option from the Initial coefficients: pop-up menu in the Estimation Parameters dialog box. *Important:* this dialog box does not

appear unless at least one covariate has already been specified; otherwise, a warning dialog box appears.

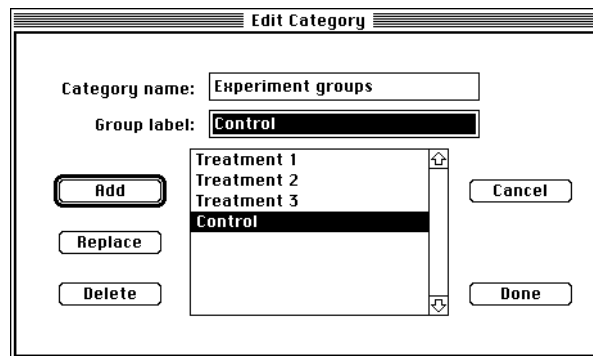
Covariates Coefficients This scrolling list shows all covariates (or in the case of nominal covariates, all except the last level of the covariate) that have been previously specified using the variable browser, as well as their initial values. To change the initial value of a covariate (or of a level of a nominal covariate), you must first select it from this scrolling list.

Coefficient initial value This text field allows you to edit the initial value of any covariate (or, in the case of nominal covariates, all except the last group level of the covariate) that has been previously specified using the variable browser. After selecting from the scrolling list the item whose initial value you wish to edit, enter the new value in this text field, then click the Specify button to change the initial value to the value you have entered.

Specify Clicking this button changes the initial value of the item selected in the scrolling list to the value that is in Coefficient initial value text field.

For nominal covariates, the coefficients associated with each group level should be thought of as values relative to the last group level specified in the covariate. This is a consequence of the convention used in StatView that the value of the coefficient for this last group level is always 0.

Because of this convention, we suggest the following: the group level with which you want the other group levels compared should be created *last* when you are formatting and entering your data. For example, if you wish to model the effect of a nominal covariate that has a control group and three treatment groups, then you probably want to compare the effect of the treatments to the control. Therefore, when creating this nominal variable, you should use a category that has Control as the last defined group level, as pictured below.



Data requirements

Except for the replacement of the group variable with the covariate variable(s), data organization and variable types for survival analysis regression models are very similar to those for non-parametric survival analyses. There are two major differences, however: The first is that among the regression models, only proportional hazards models accept a stratification variable. Stratification variables are ignored by the parametric (Weibull, exponential, lognormal, and loglogistic) regression models. The second difference, pertaining to the use of covariates in

regression models, in contrast to the use of the group variable in nonparametric analyses, is summarized in the next paragraph. For a discussion of event time and censor variables, please see [“Data requirements,” p. 157.](#)

As suggested above, survival analysis regression models require the specification of an event time variable and, in some cases, at least one covariate variable, hereafter referred to as a covariate. (Specifically, proportional hazards models require at least one covariate, while the covariate is optional in parametric models.) Like the group variable used in nonparametric survival analyses, the covariate may be used to indicate study or treatment groups in regression models. Unlike the group variable, however, covariates can be either nominal *or* continuous, and you may specify more than one covariate in a single model. Regression models also allow one or more covariates to be forced into stepwise models.

The following picture shows a dataset with all variables properly formatted and ready for entry in a survival analysis regression model.

	Event Time	Censored Variable	Covariate 1	Covariate 2	Covariate 3	Covariate 4	Strata
Type:	Integer	Category	Category	Real	Real	Real	Integer
Source:	User Entered	User Entered	User Entered	User Entered	Dynamic Fo...	Dynamic Fo...	User Ent...
Class:	Continuous	Nominal	Nominal	Continuous	Continuous	Continuous	Nominal
Format:	•	•	•	Free Forma...	Free Forma...	Free Forma...	•
Dec. Places:	•	•	•	1	3	3	•
39	61	Uncensored	Treatment	21.0	.273	-.208	2
40	35	Uncensored	Treatment	18.5	-.376	-.235	3
41	20	Censored	Treatment	20.7	-.050	-.337	1
42	39	Uncensored	Control	20.1	-.213	-.823	1
43	46	Uncensored	Control	20.3	1.933	-.306	2
44	61	Uncensored	Control	20.7	.356	6.237E-5	3
45	56	Uncensored	Control	20.5	-1.614	-.397	2

The following shows how to use the buttons in the variable browser to assign these variables to a survival regression model.

Variable browser buttons	
Time	Select one event time variable (continuous), then click the Time button. Usage is indicated by a T in the variable browser. A second continuous variable assigned with the Time button is used as a new event time variable. This creates a new analysis using all previously specified censor, covariate, stratification and split by variables.
Censor	Select one censor variable (nominal), then click the Censor button. Acceptable values are 0 (must be Type: Integer or Real), or Uncensored (Type: String or Category) for uncensored observations, and any other numeric value or Censored to indicate censored observations. Usage is indicated by a C in the variable browser. NOTE: The Survival Analysis Preferences dialog box allows you to change the meaning of values in the censor variable so that 0 indicates censored observations. See “Survival Analysis preferences,” p. 230 of Using StatView for details. Each additional censor variable creates a new analysis using all other variables already specified.
Covariate	Select one or more covariates (nominal or continuous), then click the Covariate button. Usage is indicated by an X in the variable browser. Each additional covariate is added to the analysis.

Force	Select one or more covariates (nominal or continuous), then click the Force button. Necessary only for forcing entry of a covariate into a stepwise model. If stepwise is not specified in the Survival: Regression Models dialog box, forced covariates are treated like any other covariate. Usage is indicated by an F in the variable browser. Each additional forced covariate is added to the analysis.
Strata	Select one stratification variable (nominal), then click the Strata button. The stratification variable, if specified, is used only by the proportional hazards regression model; it is ignored by the parametric regression models. Usage is indicated by the symbol # in the variable browser. For proportional hazards models, each additional stratification variable creates a new proportional hazards analysis using all other variables already specified.
Split By	When you assign one or more split-by variables (nominal) to a survival analysis regression model, results are displayed separately for each cell defined in the split-by variable(s). Usage is indicated by an S in the variable browser.

If you routinely create analyses first, then assign variables, you will find that the analyses will begin computing as soon as you have specified an event time variable and, for proportional hazards models, a covariate. This may be unduly time consuming, especially if you must then assign additional covariates, censor, and stratification variables. To avoid this, do one of the following: (1) assign variables *first*, then create your analyses; (2) always assign the event time variable after all other variables have been assigned; or (3) disable the Recalculate box in the view before adding variables, then enable it once variable assignment is complete. If you choose to assign variables before creating the analysis, you can configure the variable browser by de-selecting all results in the view, then clicking on any item within the Survival: Regression Models header in the analysis browser.

Results

Default Results

Default results are those created by selecting the Survival: Regression Models header in the analysis browser. They can also be selected individually by opening the Survival: Regression Models header.

Summary Table

This table is created by selecting the Summary Table item within the Survival: Regression Models header in the analysis browser.

# Obs	Gives the total number of valid observations for which no variable specifications are missing.
# Events	Gives the number of valid uncensored event times.
# Censored	Gives the number of censored observations in the event time variable.
% Censored	Gives the percentage of event times that are censored, relative to the total number of valid observations in the event time variable.

# Missing	Gives the number of observations with missing variable specifications.
# Invalid	Gives the number of observations with invalid variable specifications, due, for instance, to negative values of the event time variable, or to uninterpretable values in the censor variable.
Other contents	Labels to the left of each row are stratum names, if the stratification variable has been specified.

Global Null Hypothesis Tests Table

This table is created by selecting the Summary Tables item within the Survival: Regression Models header in the analysis browser.

Chi-Square	Gives the chi-square statistic computed for the indicated test. Evaluates the significance of simultaneous exclusion of all covariates from the model.
DF	Gives the degrees of freedom for the associated chi-square statistic.
P-Value	Gives the <i>p</i> value (the probability of rejecting a true null hypothesis) for the associated chi-square statistic.
Other contents	Labels to the left of each row indicate the tests that are computed.

Stepwise Summary Table (stepwise only)

This table is produced whenever one of the two stepwise methods is enabled in the Survival: Regression Models dialog box. It does not require selection of the Summary Tables item within the Survival: Regression Models header in the analysis browser.

P-to-Enter	Gives the value entered in the Enter <i>p</i> text field in the more choices version of the Survival: Regression Models dialog box.
P-to-Remove	Gives the value entered in the Remove <i>p</i> text field in the more choices version of the Survival: Regression Models dialog box.
Number of Steps	Gives the total number of variable entry and removal steps required to satisfy the specified P-to-Enter and P-to-Remove criteria.
Variables Entered	Gives the total number of covariates (forced and unforced) in the model at the conclusion of the stepwise procedure.
Variables Forced	Gives the number of covariates forced into the model using the Force button in the variable browser.

Model Coefficients Table

This table is created by selecting the Model Coefficients Table item within the Survival: Regression Models header in the analysis browser. Information for each continuous covariate

occupies one row of this table; information for each nominal covariate occupies one row of the table plus one additional row for each covariate level.

DF	Gives the degrees of freedom associated with each continuous covariate, or with each level of each nominal covariate. For parametric models, also gives the degrees of freedom associated with the intercept and scale parameters.
Coef	Gives the model estimate of the regression coefficient associated with each continuous covariate, or with each level of each nominal covariate. For parametric models, also gives the model estimate of the intercept and scale parameters.
Std Error	Gives the estimates of the asymptotic standard error about each coefficient.
Coef/SE	Gives the values of each coefficient divided by its standard error.
Chi-Square	Gives the values of the chi-square statistic associated with the hypothesized exclusion of the individual continuous covariates or with each level of each nominal covariate.
P-Value (P-to-Remove if stepwise)	Gives the p value (probability of rejecting a true null hypothesis) for the associated chi-square statistic.
Exp(Coef)	Gives the value of e^{coef} for each estimated coefficient. This quantity gives a more easily interpreted measure of the relative effect on the model of the individual coefficients than do the untransformed coefficients.
Other contents	Row labels are the names of each covariate, or level for nominal covariates.

Variables Not In Model Coefficients Table (stepwise only)

This table is produced only if one of the two stepwise methods is enabled in the Survival: Regression Models dialog box. It is created by selecting the Model Coefficients Table item within the Survival: Regression Models header in the analysis browser.

P-to-Enter	Gives the p value associated with any assigned covariates not entered in the model.
------------	---

Baseline Cumulative Survival Plot

This graph is created by selecting the Cumulative Survival Plot item within the Survival: Regression Models header in the analysis browser.

Plotted line(s)	These give the estimated value of the baseline cumulative survival function for each specified stratum.
Plotted points	When present, these give the time and corresponding value of the baseline cumulative survival function for censored and uncensored events. Display of these events is controlled by Survival plots show checkboxes in the Survival: Regression Models dialog box. If Add Kaplan-Meier estimates to baseline: Plots is enabled in the Survival: Regression Models dialog box, plotted points also show the Kaplan-Meier estimate of the survival function. Separate plots are provided for each stratum in proportional hazards models.

Other Results

Confidence Intervals Table

This table is created by selecting the Confidence Intervals Table item within the Survival: Regression Models header in the analysis browser. It is not computed if either of the stepwise methods is enabled in the Survival: Regression Models dialog box.

Exp(Coef)	Gives the value of e^{coef} for each estimated coefficient.
<%> Lower	Gives the lower confidence limit for the value e^{coef} for each estimated coefficient. The magnitude of the confidence interval is controlled by the Confidence level text field in the Survival: Regression Models dialog box. The default is 95 percent confidence limits.
<%> Upper	Gives the upper confidence limit for the value e^{coef} for each estimated coefficient. The magnitude of the confidence interval is controlled by the Confidence level text field in the Survival: Regression Models dialog box. The default is 95 percent confidence limits.

Baseline Survival Table

This table is created by selecting the Baseline Table item within the Survival: Regression Models header in the analysis browser. Separate tables are created for each specified stratum in proportional hazards models.

Time	Gives the uncensored event times.
Cumulative Survival	Gives the values of the model estimate of the baseline cumulative survival function for the indicated event times.
Cumulative Survival (KM)	Gives the values of the Kaplan-Meier cumulative survival function for the indicated event times. Appears only if Add Kaplan-Meier estimates to baseline: Tables is enabled in the Survival: Regression Models dialog box.
Cumulative Hazard	Gives the values of the model estimate of the baseline cumulative hazard function for the indicated event times.
Cumulative Hazard (KM)	Gives the values of the Kaplan-Meier cumulative hazard function for the indicated event times. Appears only if Add Kaplan-Meier estimates to baseline: Tables is enabled in the Survival: Regression Models dialog box.
Ln Cumulative Hazard	Gives the values for the model estimate of the natural log of the baseline cumulative hazard function for the indicated event times.
Ln Cumulative Hazard (KM)	Gives the values of the Kaplan-Meier estimate of the natural log of the cumulative hazard function for the indicated event times. Appears only if Add Kaplan-Meier estimates to baseline: Tables is enabled in the Survival: Regression Models dialog box.

Baseline Cumulative Hazard Plot

This graph is created by selecting the Cumulative Hazard Plot item within the Survival: Regression Models header in the analysis browser.

Plotted lines	These give the estimated value of the baseline cumulative hazard function. If Add Kaplan-Meier estimates to baseline: Plots is enabled in the Survival: Regression Models dialog box, this graph also shows the Kaplan-Meier estimate of the cumulative hazard function. Separate plots are provided for each stratum specified in proportional hazards models.
---------------	---

Baseline Ln Cumulative Hazard Plot

This graph is created by selecting the Ln Cumulative Hazard Plot item within the Survival: Regression Models header in the analysis browser.

Plotted lines	These give the natural log of the estimated value of the baseline cumulative hazard function versus the natural log of the event time variable. If Add Kaplan-Meier estimates to baseline: Plots is enabled in the Survival: Regression Models dialog box, this graph also shows the Kaplan-Meier estimate of the natural log of the cumulative hazard function. Separate plots are provided for each stratum specified in proportional hazards models.
---------------	---

Iteration History Table

This table is created by selecting Iteration History Table within the Additional Results sub-header within the Survival: Regression Models header in the analysis browser.

Contents	Shows the coefficient estimates for each continuous covariate, or each group level for nominal covariates, at each iteration of the fitting process. Also gives the estimates for the intercept and scale parameters for parametric models (scale parameter is excluded from exponential models). The log likelihood for the fit at each iteration is also given in the bottom row of the table.
----------	--

Coefficient Correlations Table

This table is created by selecting Coef Correlations Table within the Additional Results sub-header within the Survival: Regression Models header in the analysis browser.

Contents	Gives the pairwise correlations between all the coefficients in the Model Coefficients Table.
----------	---

Coefficient Covariances Table

This table is created by selecting Coef Covariances Table within the Additional Results sub-header within the Survival: Regression Models header in the analysis browser.

Contents	Gives the pairwise covariances between all the coefficients in the Model Coefficients Table.
----------	--

Martingale Residual Plot

This graph is created by selecting the Residual Plots item within the Additional Results subheader within the Survival: Regression Models header in the analysis browser. It is also created if the Additional Results subheader is selected. Separate graphs are created for each specified stratum in proportional hazards models.

Plotted points	These give the values of the martingale residuals on the vertical axis and the corresponding values of the linear predictor ($\beta'Z$) on the horizontal axis.
----------------	---

Deviance Residual Plot

Available only for proportional hazards models, this graph is created by selecting the Residual Plots item within the Additional Results subheader within the Survival: Regression Models header in the analysis browser. It is also created if the Additional Results subheader is selected. Separate graphs are created for each specified stratum.

Plotted points	These give the values of the deviance residuals on the vertical axis and the corresponding value of the linear predictor ($\beta'Z$) on the horizontal axis.
----------------	--

Quantile Plot

This graph is created by selecting Quantile Plot within the Additional Results subheader within the Survival: Parametric Models header in the analysis browser. It is also created if the Additional Results subheader is selected. Results are not computed for this plot if the model is proportional hazards.

Plotted points	On the vertical axis, points give the value of the estimated baseline inverse cumulative distribution function (CDF) evaluated at the distinct values of a Kaplan-Meier (KM) estimate of the CDF. These are plotted against the corresponding KM values on the horizontal axis. If there are no covariates in the model, the KM estimate is based on the observed times. If there are covariates in the model, the observed times are adjusted for the covariates using the estimated model coefficients and the KM estimate is based on these adjusted times. If the event times are drawn from the modeled distribution, the points should form a straight line.
----------------	--

Joint Significance Tests Table

This table is created by specifying one or more joint significance tests using the Joint Significance Tests dialog box. One table is created for every subset of covariates specified in this dialog box. See [“Joint Significance Tests dialog box,” p. 179](#). NOTE: Joint significance tests are not computed for stepwise models.

Chi-Square	Gives the value of the chi-square statistic for the indicated test of the specified hypothesis.
DF	Gives the degrees of freedom for the indicated test of the specified hypothesis.

P-Value	Gives the p value (probability of rejecting a true null hypothesis) for the associated chi-square statistic and degrees of freedom.
Other contents	If more than one test is enabled with the Tests to perform checkboxes in the Joint Significance Tests dialog box, the names of the enabled tests are given in the column to the left of the table.

Templates

The following templates provide survival regression analysis results.

Survival Analyses	Cox (Prop. Hazards) Model	Confidence intervals, global null hypothesis tests, and model coefficients and survival summary tables; baseline cumulative hazard, baseline cumulative survival, baseline ln cumulative hazard, deviance residuals, and Martingale residuals plots.
	Exponential Model	Confidence intervals, global null hypothesis tests, model coefficients, and survival summary tables; baseline cumulative hazard, baseline cumulative survival, baseline ln cumulative hazard, quantile, and Martingale residuals plots.
	Loglogistic Model	Confidence intervals, global null hypothesis tests, model coefficients, and survival summary tables; baseline cumulative hazard, baseline cumulative survival, baseline ln cumulative hazard, quantile, and Martingale residuals plots.
	Lognormal Model	Confidence intervals, global null hypothesis tests, model coefficients, and survival summary tables; baseline cumulative hazard, baseline cumulative survival, baseline ln cumulative hazard, quantile, and Martingale residuals plots.
	Weibull model	Confidence intervals, global null hypothesis tests, model coefficients, and survival summary tables; baseline cumulative hazard, baseline cumulative survival, baseline ln cumulative hazard, quantile, and Martingale residuals plots.

Exercise

This exercise illustrates how to fit survival regression models. The data are from a prospective study of the occurrence of coronary events—usually heart attacks. Covariates that may influence the risk of a coronary event include smoking behavior, blood pressure history, and cholesterol level. These data are from the Western Collaborative Group Study (described in Rosenman *et al.* (1975), among other places) of 3,154 male employees from ten California companies during 1960–1961. (The data we will analyze in this example are a randomly selected subsample of the complete dataset.) The original purpose of the study was to investigate the effects of behavior type and smoking habits on heart disease. The researchers also collected information on other possible risk factors that are not included in this dataset.

After recruitment, the study followed participants for nine years, although a few were lost to follow-up (i.e., censored) before the end of the study. The time variable of interest was the interval from entry into the study until the appearance, as determined by a medical expert, of coronary heart disease. The dataset `wcgs Data` in the Sample Data folder contains event time and censor variables for 614 participants, as well as measurements of two covariates of interest—smoking behavior at study entry (measured in the number of cigarettes smoked per day) and behavior type (a nominal variable with two levels, referred to as Type A and Type B). Individuals were classified into behavior types on the basis of an interview; in general terms, Type A behavior is characterized by aggressiveness and competitiveness, whereas Type B behavior is considered more relaxed and noncompetitive. In this subsample of the `wcgs` data, events were observed in 60 individuals.

In this exercise, you will use both a proportional hazards regression model and a parametric accelerated failure time model to fit these failure time data.

- Open `wcgs Data` from the Sample Data folder

The four variables include two covariates (Cigarettes and Personality Type), a censor variable (Censor), and the event time variable (Time), which consists of the number of days from entry into the study until the occurrence of either the event or censoring.

First we fit a proportional hazards model to the observed data, using Cigarettes as the only covariate.

- From the Analyze menu, select New View
- In the analysis browser, select Survival: Regression Models
(This is equivalent to selecting the default results: Summary Tables, Model Coefficients Table, and Cum. Survival Plot.)
- In the variable browser, select Time and click Time
- Select Censor and click Censor
- Select Cigarettes and click Covariate
- In the analysis browser, click Create Analysis
- Click OK to accept the default analysis parameters

This creates the default survival regression results: Survival Summary Table, Global Null Hypothesis Tests Table, the Model Coefficients Table, and the Baseline Cumulative Survival Plot. Scroll down to the Model Coefficients Table to see how cigarette consumption affects this model.

Model Coefficients for Time							
Censor Variable: Censor							
Model: Proportional Hazards							
	DF	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
Cigarettes	1	.016	.008	2.055	4.222	.0399	1.016

These results show an estimated regression coefficient of 0.016, indicating a relative hazard for a coronary event of $e^{0.016} = 1.016$ associated with an increase in cigarette consumption of one cigarette per day at study entry. This is not a particularly meaningful comparison since the covariate difference is so small; a single cigarette each day would not be expected to have a major effect. The relative hazard associated with a more substantial increase of, say, 20 ciga-

rettes (one pack) per day is immediately available from these results and is given by $e^{20\beta} = 1.377$, indicating a 37.7 percent increase in hazard throughout the follow-up period.

- In the analysis browser under Survival: Regression Models, select Confidence Intervals Table and click Create Analysis

Confidence Intervals for Time				
Censor Variable: Censor				
Model: Proportional Hazards				
	Exp(Coef)	95% Lower	95% Upper	
Cigarettes	1.016	1.001	1.032	

This table shows the information necessary to calculate the 95-percent confidence intervals about the relative hazard associates with consumption of one pack of cigarettes per day: $(1.001^{20}, 1.032^{20}) = (1.020, 1.877)$. Note that the p value for testing the null hypothesis ($H_0: \beta = 0$) that smoking consumption does not influence the risk of a coronary event is less than 0.05, indicating that observed differences in the rates of coronary events among individuals with varying cigarette consumption are unlikely to have arisen by chance variation.

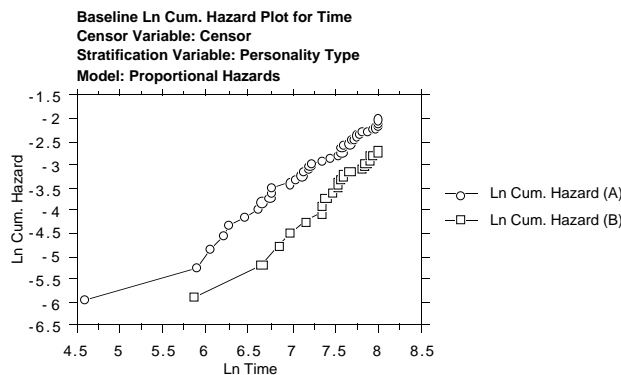
Now we expand the model to see if Personality Type adds any explanatory information. However, before you use Personality Type as a covariate in this model, it is important to determine whether the hazard functions for personality types A and B are, in fact, proportional. (Note that we have also assumed that hazards are proportional over the levels of smoking—later, we shall consider briefly the fit of our overall model.) To evaluate the proportionality assumption, you can assign Personality Type as a stratification variable to the present model.

- Make sure at least one of the results is still selected
- In the variable browser, select Personality Type and click Strata
- In the analysis browser under Survival: Regression Models, select Ln Cum. Hazard Plot and click Create Analysis

With Personality Type used to stratify the model, the baseline hazard function is allowed to vary between the two behavior types, but the effects of cigarette smoking are assumed to be the same in both groups, as specified by the proportional hazards assumption.

Model Coefficients for Time							
Censor Variable: Censor							
Stratification Variable: Personality Type							
Model: Proportional Hazards							
	DF	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
Cigarettes	1	.014	.008	1.763	3.109	.0779	1.014

The regression coefficient for cigarette consumption is now 0.014. The following graph gives the ln cumulative baseline hazard plots for both personality types.



This graph illustrates that these two curves are approximately parallel, suggesting that it is reasonable to assume that the hazard functions for the two behavior types are proportional. This indicates that it would be appropriate to use Personality Type as a covariate. Note, however, that the observation with the smallest event time in the Type A group appears unusual; it might be worthwhile to eliminate this observation and refit the model to determine whether this particular observation has unduly influenced the model estimates.

Now that you know that the assumption of proportional hazards is reasonable, you can remove Personality Type as a stratification variable and reassign it as a covariate.

- Make sure at least one of the results is still selected
- In the variable browser, select Personality Type and click Remove, then click Covariate

Confidence Intervals for Time
Censor Variable: Censor
Model: Proportional Hazards

	Exp(Coef)	95% Lower	95% Upper
Cigarettes	1.014	.999	1.029
Personality Type: A	1.866	1.094	3.184

Model Coefficients for Time
Censor Variable: Censor
Model: Proportional Hazards

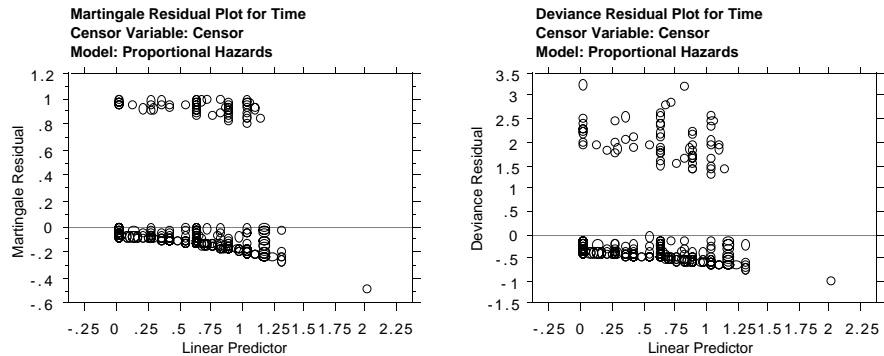
	DF	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
Cigarettes	1	.014	.008	1.783	3.178	.0746	1.014
Personality Type: A	1	.624	.273	2.290	5.244	.0220	1.866

Note from the results in the model coefficients table that the relative hazard associated with a pack per day increase in cigarette consumption is now estimated to be $e^{20 \times 0.014} = 1.323$, very similar to the estimate produced without Personality Type in the model; this indicates that adjustment for behavior type makes very little difference to the relationship between cigarette consumption and the risk of a coronary event. The relative hazard for coronary heart disease, comparing Type A and Type B individuals, is $e^{0.624} = 1.866$, with an associated 95-percent confidence interval given by (1.094, 3.184). In this comparison, therefore, Type A individuals are estimated to have nearly twice the hazard of a coronary event than do Type B individuals.

Before turning to the analysis of the same covariates using a parametric accelerated failure time regression model, you should check whether the proportional hazards model adequately

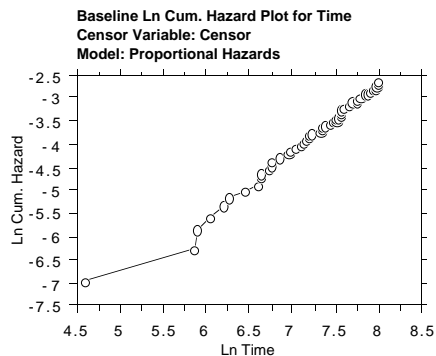
describes the effects of these two covariates (cigarette consumption and personality type) on coronary heart disease. Both the martingale and deviance residuals plotted against the linear predictor can help you evaluate the model.

- Make sure at least one of the results is still selected
- In the analysis browser under Survival: Regression Models' Additional Results subheading, select Residuals Plots and click Create Analysis



The graph of martingale residuals plots these residuals against the linear predictor—in this case, a linear combination of smoking consumption and the variable describing personality type. Martingale residuals are always negative for censored observations, which explains why most of the residuals in this plot are less than zero. Although these residuals have mean zero, you can see from the graph above that they are not symmetrically distributed about zero. In this example, where more than 90 percent of the observations are censored, both the martingale and deviance residual plots are somewhat difficult to interpret because of the large number of negative residuals close to zero and the fact that heavy censoring of this kind means that it is inappropriate to assume that deviance residuals are normally distributed. Nevertheless, there is no noticeable pattern to either residual plot, which suggests that the linear part of the model assumption is reasonable. Note that one individual—with a linear predictor value close to 2—has an extreme combination of covariate values. Examination of the saved residuals indicates that this is case number 485 in the dataset, corresponding to a cigarette consumption of 99 cigarettes per day. This suspiciously resembles a data entry code for a missing value; even if the reported value is accurate, it would be advisable to refit the models with this individual deleted to determine its influence on the results.

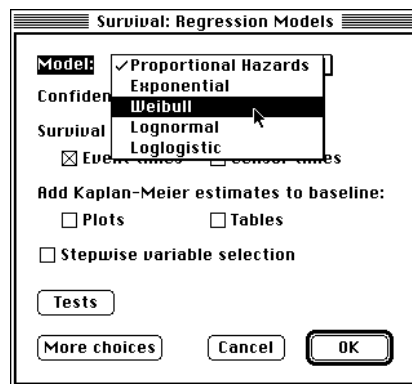
Before we try to fit a parametric model of these data, we should examine the baseline In cumulative hazard plot.



The points in this graph roughly approximate a straight line, which suggests that a Weibull model for the baseline hazard might adequately fit the behavior of these data. The slope of the approximating line is clearly different from 1, which indicates that the exponential model is not likely to fit the data as well as the Weibull model.

Finally, you should try using a Weibull regression model to describe the dependence of the event time variable on these same two covariates.

- Click in the empty space of the view to deselect all results
- In the analysis browser under Survival: Regression Models, Control-click (Windows) or Command-click (Macintosh) to select Summary Tables, Model Coefficients Table, and Quantile Plot
(Quantile Plot is found under the Additional Results subheader.)
- In the variable browser, select Time and click Time
- Select Censor and click Censor
- Select Cigarettes and Personality Type and click Covariate
- In the analysis browser, click Create Analysis
- In the dialog box: for Model, choose Weibull



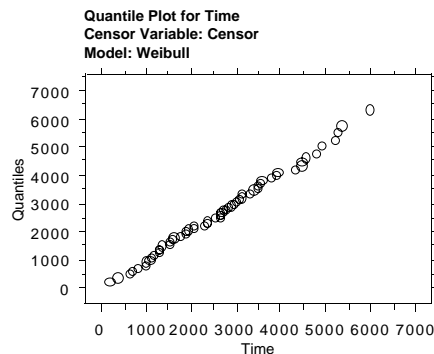
- Click OK
- Select the resulting table and click Edit Display
- Choose 6 decimal places and click OK

Model Coefficients for Time
Censor Variable: Censor
Model: Weibull

	DF	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
Scale	1	.665753	.082351	8.084352	•	•	•
Intercept	1	9.817419	.280208	35.036195	1227.534972	<.0001	18350.633294
Cigarettes	1	-.009137	.005275	-1.732186	3.000467	.0832	.990905
Personality Type: A	1	-.419467	.188196	-2.228881	4.967911	.0258	.657397

The coefficients reported here refer to the accelerated failure time model. Therefore, an increase of 20 cigarettes per day is estimated to reduce time to the event by a factor of $e^{20 \times -0.009137} = 0.83$, i.e., by approximately 17 percent. In other words, the time to an event for those individuals who smoke 20 cigarettes per day is estimated to be 17 percent shorter, on average, than for nonsmokers. This result is qualitatively consistent with the analysis based on a proportional hazards regression model, which shows that increases in cigarette consumption are associated with higher hazards and shorter times to coronary events. A closer comparison of the two approaches is also possible with the above output. Recall from the [“Discussion,” p. 168](#), that you can also give a Weibull regression model a relative hazard interpretation. In particular, this alternative interpretation yields proportional hazards across levels of the covariates. The estimated log relative hazard is given by the value of the model coefficient from the Weibull fit, multiplied by -1, and then divided by the estimated scale parameter (σ). In our example, this yields a log relative hazard for a unit increase in cigarette consumption given by $0.009137/0.666 = 0.014$ (to three decimal places). For the personality types, the Weibull model gives a log relative hazard comparison between the two groups of $0.419/0.666 = 0.629$, with Type A individuals having the higher hazard. These estimates are very close to those obtained above from the proportional hazards regression model. In passing, note that the scale parameter is estimated to be 0.666—indicating that the hazard is increasing—with an estimated standard error of 0.082. This strongly suggests that the true scale is significantly smaller than one, which indicates that the Weibull regression model provides a better fit to the data than the exponential version, as was also suggested by the ln cumulative hazard plot considered above.

Finally, the quantile plot provides a graphical means for assessing the appropriateness of the Weibull assumption for fitting this accelerated failure time model.



The fact that the quantile plot closely approximates a straight line of slope 1 passing through the origin, suggests that the Weibull assumption is appropriate for these data.

Before concluding, you should be aware that the data on this cohort are considerably more extensive than reported here. Furthermore, the follow-up for heart disease *mortality* (as opposed to the mere incidence of a coronary event) continued beyond the time frame of the original study (a report on 22 years of follow-up is given in Ragland and Brand (1988)). Curiously, this later analysis of the mortality data found no association between behavior type and heart disease mortality. Part of the explanation of the anomalous results between analysis of coronary events and mortality may be that behavior type is related to the chances of successful recovery from initial coronary events.

Logistic regression

Logistic regression is a modeling technique analogous to linear regression. It examines the relationship between an outcome (or dependent) variable with one or more independent variables. The primary difference from linear regression is that the dependent variable, rather than being continuous, is a nominal variable. In the most common case, the dependent variable is **binary** or **dichotomous**—that is, it has two possible values. However, the technique can also be employed for a **polytomous** (many-valued) nominal response variable. StatView can perform both dichotomous and polytomous logistic regression with one or more independent variables.

Logistic regression methods can be applied in a wide variety of settings. In many biomedical examples a binary dependent variable might indicate whether an individual contracts a certain disease in a specified time period. Independent variables of interest might include smoking history, age, and alcohol consumption patterns for each individual. However, the methodology extends to much broader settings where, for example, the outcome might be whether an individual voted Republican or Democratic in the last presidential election, and independent variables might include family income, marital status, gender, parental voting history, etc.

Discussion

This discussion assumes familiarity with the analogous linear regression modeling technique and its assumptions, which are discussed in the chapter [“Regression,” p. 51](#). If we use the symbol Y to denote the dependent (or **outcome** or **response**) variable, the linear regression model discussed in that chapter is based on the assumption that

$$E(Y|x_1) = b_0 + b_1x_1, \quad (\text{Eq. 15.1})$$

where $E(Y|x_1)$ is read as “the conditional mean of Y given x_1 ” in the simplest case where there is only one independent (or **explanatory** or **covariate**) variable of interest.

For logistic regression, the dependent variable assumes only two values—traditionally coded $Y = 1$ and $Y = 0$. Then,

$$E(Y|x_1) = p_{x_1}, \quad (\text{Eq. 15.2})$$

where p_{x_1} is the probability that $Y = 1$ for any individual for whom $X_1 = x_1$. The linear model ([Eq. 15.1](#) above) suffers from two problems. First, while its right side can potentially take any value, its left side—being a probability, as we see in [Eq. 15.2](#)—is constrained to lie

between 0 and 1. Second, interpreting the coefficients is somewhat less natural when the outcome is binary, and as we will discuss later, the model cannot be applied directly to what are known as case-control or retrospective studies.

Simple logistic regression model

Therefore, the logistic regression model instead predicts a *nonlinear transformation* of $E(Y|x_1)$ (the left side of [Eq. 15.1](#)) from the independent variable:

$$\log\left(\frac{p_{x_1}}{1-p_{x_1}}\right) = b_0 + b_1 x_1 \quad (\text{Eq. 15.3})$$

Solving this equation for p_{x_1} (recall that exponentiation is the inverse of logarithm), we get the simple logistic regression model, which suffers from neither of the earlier problems:

$$E(Y|x_1) = p_{x_1} = \frac{e^{(b_0 + b_1 x_1)}}{1 + e^{(b_0 + b_1 x_1)}} \quad (\text{Eq. 15.4})$$

Once the coefficients of the model b_0 and b_1 are known (or estimated), we can use this formula to calculate the probability of a given response, say $Y = 1$, for any specified value of the covariate X_1 .

The nonlinear transformation we used is the log of the odds that $Y = 1$, where **odds** refers to the probability that $Y = 1$ divided by one minus the same probability, given x_1 :

$$\log\left(\frac{p_{x_1}}{1-p_{x_1}}\right)$$

This is called the **log odds** of the dependent variable when $X_1 = x_1$, which the model assumes to change *linearly* with changes in X_1 , as seen in [Eq. 15.3](#). This is the key linearity assumption of the logistic regression model, as there is no *a priori* reason why the risk or probability that the outcome variable $Y = 1$ should vary with X_1 in this way. Goodness-of-fit tests for the model provide one check for the validity of this assumption.

Interpreting coefficients

To use and interpret a regression model effectively, it is crucial to understand the meaning of the model coefficients— b_0 and b_1 in [Eq. 15.3](#). First, consider the **intercept** term, b_0 : this coefficient is simply the log odds associated with the outcome $Y = 1$ for individuals whose independent variable $X_1 = 0$. In other words, the coefficient b_0 determines the baseline (i.e., $X_1 = 0$) probability that $Y = 1$. Specifically, from [Eq. 15.4](#):

$$P(Y = 1 | X_1 = 0) = p_0 = \frac{e^{b_0}}{1 + e^{b_0}}$$

Now, consider the **slope coefficient**, b_1 . [Eq. 15.3](#) shows that this coefficient measures the change in the log odds that $Y = 1$ (the log odds ratio) associated with a *unit* change in the

independent variable, X_1 . In other words, b_1 provides a measure of the **impact** (or association) of the variable X_1 with the dependent variable Y . Specifically:

$$\text{Odds Ratio}(X_1 = x_1 + 1 \text{ compared to } X_1 = x_1) = e^{b_1},$$

The fact that the odds ratio associated with a unit change in X_1 does not depend on the level of X_1 is due to the linearity assumption of the model. When the variable X_1 is binary, e.g. gender, then the term e^{b_1} is simply the odds ratio relating the factor (such as gender) to the outcome and is usually estimated from a 2×2 contingency table.

The attraction of the odds ratio as a measure of association is twofold. First, in rare outcome settings (such as when p_{x_1} is small for all values of the independent variable, X_1) it approximates the **relative risk** (e.g., a relative risk of ten in a smoking/lung cancer study would indicate that subjects who smoke are ten times more likely to develop lung cancer than subjects who don't), which is easier to interpret. Second, the odds ratio can be calculated from case-control study data, as we discuss under "[Case-control studies](#)," p. 203, whereas other measures of association such as **excess risk** (for example, the absolute difference in risk of lung cancer, comparing smokers to nonsmokers) cannot.

Multiple logistic regression models

Extending the simple logistic regression model ([Eq. 15.3](#)) to accommodate several independent variables simultaneously is straightforward. With independent variables X_1, X_2, X_3, \dots , we work with the following model:

$$\log \left(\frac{p_{x_1, x_2, x_3, \dots}}{(1 - p_{x_1, x_2, x_3, \dots})} \right) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots \quad (\text{Eq. 15.5})$$

As in [Eq. 15.4](#), we can solve this equation to express $p_{x_1, x_2, x_3, \dots}$ in terms of the coefficients:

$$p_{x_1, x_2, x_3, \dots} = E(Y|x_1, x_2, x_3, \dots) = \frac{e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots}}{1 + e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots}} \quad (\text{Eq. 15.6})$$

The coefficients in the multiple logistic regression model have interpretations similar to the single independent variable case, except for one subtle but important difference. As for the case with one independent variable, the intercept coefficient b_0 is the log odds associated with the outcome $Y = 1$ for individuals whose covariate values are all zero; that is, $X_1 = X_2 = X_3 = \dots = 0$. Again, from [Eq. 15.6](#):

$$P(Y = 1 | X_1 = X_2 = X_3 = \dots = 0) = p_{0, 0, 0, \dots} = \frac{e^{b_0}}{1 + e^{b_0}}$$

The slope coefficient associated with the j th independent variable X_j , b_j in [Eq. 15.5](#), measures the change in the log odds that $Y = 1$ associated with a *unit* increase in the independent variable, X_j , *controlling for* or *holding fixed* all other independent variables X_k for $k \neq j$.

That is, b_j is equal to the log odds ratio associated with a unit increase in X_j with all other independent variables held at a fixed level. Specifically:

$$\text{Odds Ratio}(X_j = x_j + 1 \text{ compared to } X_j = x_j | \text{all other covariates held fixed}) = e^{b_j}$$

This odds ratio measures the strength of the relationship between the independent variable X_j and the outcome Y , controlling for the potential confounding effects of all other independent variables in the model, at least to the extent that their impact is adequately represented by the linear assumption of the model (Eq. 15.5).

Assumptions

Fitting a logistic regression model to a set of data is appropriate only when all of the following conditions apply:

1. The independent variables are assumed to have a linear relationship with the log odds based on the probability $p_{x_1, x_2, x_3, \dots} = P(Y = 1 | x_1, x_2, x_3, \dots)$ associated with the binary dependent variable as described by Eq. 15.5. In particular, as the value of any independent variable increases, the probability that the dependent variable is coded as 1 must increase or decrease consistently. This is the “**dose response**” assumption of the model. Of course, the model also specifies the linearity of this response as measured on the log odds scale for the probability p . Note that more complex relationships can be modeled in logistic regression by including functions of the independent variables (for example, by including a formula variable X_1^2) as additional independent variables. Goodness-of-fit tests are useful for comparing the ability of different models to fit the observed data.
2. All cases (the values of the dependent and independent variables) are assumed to be independent of each other. When this is not true (for example, when observations are measured on the same subject over time), a logistic regression model could still be applied, but more sophisticated techniques than those available in StatView would be needed to estimate standard errors and p values associated with hypothesis tests.
3. It is assumed that, for a given set of values for the independent variables, the variation of the dependent variable Y (the pattern of observed 0s and 1s) is consistent with a random response with fixed probability p . In some cases, although the probability might vary with the independent variables according to the model given in Eq. 15.5, the variation in responses at any fixed value might be more than expected (this is known as **extra-Binomial variation**), or less than expected, such as when values of the outcome are identical at each set of specific covariate values.

As with any modeling procedure, the goal of logistic regression is to find the best-fitting, most parsimonious model that has a reasonable interpretation in the context of the example.

Estimating coefficients

Random samples

We assume that the observations arise from a simple random sample from some population, or at least a random sample at each of a specified set of fixed covariate values. The latter reflects the fact that the logistic regression model is a conditional model for the dependent variable given the independent variables (as is the linear regression model). In such cases, the model coefficients are estimated by the maximum likelihood technique. In the simple logistic model, [Eq. 15.3](#), the likelihood function based on a set of n independent observations, $(x_{1i}, y_i): i = 1, \dots, n$, is given by:

$$l(b_0, b_1) = \prod_{i=1}^n p_{x_{1i}}^{y_i} (1 - p_{x_{1i}})^{1-y_i}$$

Estimates of b_0 and b_1 are obtained by maximizing this function over all possible values of b_0 and b_1 . The log of the likelihood function provides a relative measure of how adequately the independent variables explain the pattern of observed responses. A simple multiple of this function, $-2\log l(b_0, b_1)$, is given a special name—it is known as the **deviance**. Estimates of the standard deviation and consequently the standard error of the maximum likelihood estimates of b_0 and b_1 can be computed from the likelihood function.

Case-control studies

In many settings, it can be expensive or impossible to obtain random samples of the dependent variable, even at pre-specified values of the independent variables. For example, the dependent variable might describe whether an individual contracts a disease after exposure to an environmental agent in a situation where the disease could take decades to produce clinically identifiable symptoms. Further, in many examples the overall frequency of $Y = 1$ responses may be so low that enormous random samples would be needed to obtain enough such outcomes to permit an effective analysis. For example, suppose you wished to investigate the propensity to use mental health services in a certain population where the overall frequency of use in a given time period was less than 1%. To overcome these obstacles, an alternative sampling strategy is to sample *separately* a set of observations where $Y = 1$ (**cases**) and a set of observations where $Y = 0$ (**controls**).

It can be shown that the maximum likelihood estimates of logistic regression model coefficients are still appropriate for such samples *except for the intercept coefficient, b_0* . The estimate of the intercept should be ignored for case-control data since it reflects the extent to which cases are over- or under-sampled in data collection as compared to their natural frequency. In our mental health example, you could generate a case-control sample of an equal number of users (cases) and nonusers (controls); this 50% frequency of users and nonusers in the data, determined by the sampling, would be picked up in the estimate of the intercept coefficient but would not affect the estimates of slope coefficients b_1, b_2, \dots .

Polytomous logistic regression models

In many situations the outcome variable Y may assume more than two distinct values. For example, if the outcome is an individual's vote in an election, there may be three candidates (Democratic, Republican, and Independent). Polytomous logistic regression models are designed to extend the logistic regression model to such a setting.

When dealing with polytomous outcome data it is important to distinguish whether the scale of Y is ordered or not. The voting example in the last paragraph has no natural ordering; the example where Y represents levels of agreement (strongly agree, agree, disagree, strongly disagree) clearly possesses an apparent order. StatView assumes **qualitative** or **unordered** dependent variables. Although this model can be applied to the ordered case, it does not take advantage of ordering.

For simplicity, we describe the polytomous logistic regression model for the case where Y can fall into any of three levels and is coded 0, 1, or 2. Note that the coding does *not* indicate ordering in the relevant levels of Y . With independent variables X_1, X_2, X_3, \dots , the polytomous logistic regression model is described by two equations, analogous to [Eq. 15.5](#):

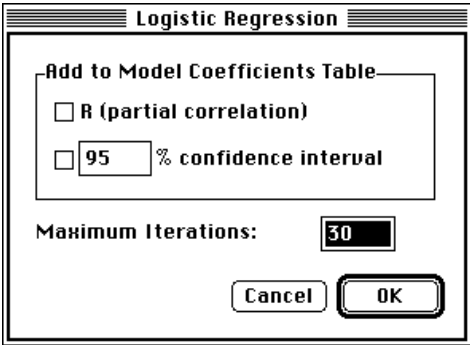
$$\begin{aligned}\log\left(\frac{\Pr(Y = 1 | x_1, x_2, \dots)}{\Pr(Y = 0 | x_1, x_2, \dots)}\right) &= b_{10} + b_{11}x_1 + b_{12}x_2 + \dots \\ \log\left(\frac{\Pr(Y = 2 | x_1, x_2, \dots)}{\Pr(Y = 0 | x_1, x_2, \dots)}\right) &= b_{20} + b_{21}x_1 + b_{22}x_2 + \dots\end{aligned}\tag{Eq. 15.7}$$

The coefficients of both these equations have similar interpretations to those in the dichotomous case. For example, b_{1j} is the log odds ratio associated with a unit increase in X_j , holding all other covariates fixed, *when comparing individuals whose outcome variable is either $Y = 0$ or $Y = 1$* (that is, we ignore individuals for whom $Y = 2$). Thus the coefficient b_{1j} measures the impact of changes in X_j on the probability that $Y = 1$, given that Y is either 0 or 1.

Similarly, b_{2j} is the analogous log odds ratio when comparing individuals whose outcome variable is either $Y = 0$ or $Y = 2$. Comparisons of individuals with Y restricted to $Y = 1$ or $Y = 2$ can also be derived by taking differences of the coefficients in [Eq. 15.7](#). For example, the log odds ratio for comparing $Y = 2$ to $Y = 1$, associated with a unit increase in X_j , holding all other covariates fixed, is simply $b_{2j} - b_{1j}$.

Dialog box settings

When you create or edit a logistic regression analysis, a dialog box asks whether to include partial correlations and confidence intervals to the model correlation table, and it offers a chance to control the number of fitting iterations. To accept the default choices (no partial correlations or confidence intervals, and at most 30 iterations) for an analysis, simply click OK.



R (partial correlation coefficient) You can choose whether to compute *R*, an approximate measure of the partial correlation coefficient for each design variable and logit level.

Confidence intervals You can choose whether to compute the upper and lower confidence intervals for the percentage level you specify. No confidence intervals are computed by default, but if you check the box on, the default level is 95%. Type a different number in the text box to specify a different level.

Maximum iterations By default, StatView iterates until the maximum likelihood tolerance (convergence criterion) is reached, in a maximum of 30 iterations. However, you may specify a different limit for the number of fitting iterations for the model.

Data requirements

StatView can perform logistic regression with unlimited independent variables. The dependent variable must be nominal with two or more levels (up to 32,000). The dependent variable can have any type, as long as its class is nominal. The independent variable(s) can be continuous or nominal.

Variable browser buttons	
Independent	Select one or more continuous or nominal variables that are the independent variable for the model and click the Independent button. Additional independent variables are added to the existing model.
Dependent	Select the nominal variable that is the dependent variable for the model and click the Dependent button. The variable can have two levels or more, up to a limit of 32,000 levels. Additional Dependent variables create additional analyses.
Split By	When you assign one or more split-by variables to any logistic regression analysis, results for each cell in the split-by variable(s) are displayed in separate tables.

If you clone logistic regression results by Control-Shift-clicking (Windows) or Command-Shift-clicking (Macintosh) the Independent button of the variable browser, the existing independent variables are *replaced* with the new independent variables. Cloning with a new Dependent variable produces a new analysis with the same independent variables.

Nominal data coding

StatView uses the first level of a nominal variable as the **reference level**—the level against which other levels are compared. For a nominal with a numeric type (real, integer, long integer, currency, or date/time), levels are sorted from smallest to greatest. For a nominal with type string, levels are sorted alphanumerically (such as 1, 11, 2, 22, A, B, C). For a nominal with type category, levels are sorted according to their order in the category definition. Generally, the easiest (and most computationally efficient) choice is a category variable whose levels are defined in order so that the desired reference level is the first level. (If you need to change the order of levels in an existing category variable, see [“How can I reorder category variables?,” p. 238 of Using StatView.](#))

For example, a model with a category dependent variable with levels “No disease” and “Disease,” in that order, and a category independent variable with groups A, B, C, D, and E, in that order, would have a coefficients table like this:

Logistic Model Coefficients Table for Outcome						
	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
Disease: constant	-2.251	.743	-3.028	9.171	.0025	.105
Group: B	.577	.866	.666	.444	.5052	1.781
Group: C	.605	.792	.764	.584	.4449	1.831
Group: D	1.414	.766	1.846	3.409	.0649	4.113
Group: E	1.638	.782	2.094	4.386	.0362	5.146

Suppose we instead used a string variable with the values “No disease” and “Disease.” When alphabetized, “Disease” comes before “No disease” and consequently will be the outcome against which other outcomes are compared. Also suppose we used a numeric grouping variable with the order reversed, e.g., A=5, B=4, C=3, D=2, E=1, so that the E or 1 group is now the reference level. Our results would be completely different, because levels of the variables would be compared in different combinations.

Logistic Model Coefficients Table for Outcome String						
	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)
No disease: constant	.613	.244	2.518	6.339	.0118	1.846
Group Integer: 2	-.224	.306	-.733	.537	.4637	1.251
Group Integer: 3	1.033	.366	2.824	7.975	.0047	2.810
Group Integer: 4	1.061	.507	2.092	4.376	.0365	2.889
Group Integer: 5	1.638	.782	2.094	4.386	.0362	5.146

Note that some programs, such as SAS, JMP, and SYSTAT, pick the *last* level rather than the first as the reference level. Therefore, when comparing results from several programs, you must be careful to code your nominal variables as needed to get the results you intend from each program.

Results

For explanation of the results, please see the preceding [“Discussion,” p. 199](#). The summary and coefficients tables are the default output. See [“Nominal data coding,” p. 206](#), for details on how to control which quantities are compared.

Logistic Summary	Table containing count, number missing, number of response levels in the dependent variable, number of fit parameters in the model, log likelihood, intercept log likelihood, and R^2 .
Model Coefficients	Table containing the coefficient for each designed variable and logit in the model, along with its standard error, the ratio of the coefficient to its standard error, the Wald chi-square statistic, the type I error probability, R , relative likelihood, and optional upper and lower bounds for the confidence limit (if chosen).
Whole Model Fit	Table containing the degrees of freedom, chi-squared statistic, and p value for the Pearson, Deviance, and Likelihood Ratio tests.
Logistic Likelihood Ratio Tests	Table containing the degrees of freedom, G likelihood ratio statistic, and p value associated with excluding each independent variable from the model.
Classification Table	Table containing the predicted and observed outcome categorizations based on their probabilities.
Iteration History	Table displaying the coefficients used in successive iterations (until the convergence criterion is reached) for each design variable for each level of the logit. The final row shows the log of the likelihood estimate for the model at each iteration.

For further options on plotting scattergrams with fitted simple regression lines see [“Bivariate Plots,” p. 221](#).

Templates

The following templates provide regression results.

Regression	Logistic Regression	Logistic Summary table and Model Coefficients table with 95% confidence intervals.
------------	---------------------	--

Exercises

Simple logistic regression

The first example is based on a very simple dataset relating coffee consumption to incidence of pancreatic cancer, as described in MacMahon *et al.* (1981). These data arose from a case-control study, and for this illustration we will use the data for male subjects. Case Outcome is a binary category variable recording whether each individual represents a case (pancreatic cancer) or a control (no cancer). Daily Coffee is a continuous variable recording how much coffee each individual drinks: 0 for none, 1.5 for 1–2 cups per day, 3.5 for 3–4 cups per day, or 5.5

for 5 or more cups per day. Any Coffee converts Daily Coffee to a binary category variable (either no coffee or some coffee) with a dynamic formula:

```
if "Daily Coffee"=0
  then "No coffee"
  else "Some coffee"
```

We can create a frequency table to see an overview of the data:

- Open the file Coffee and Pancreatic Cancer from the Sample Data folder
- Copy the Daily Coffee variable and Paste it into the Input column
- From the Class pop-up menu for Daily Coffee.2 (the new copy of the variable), select Nominal
(We need a nominal version of the variable for the contingency table, but we want to keep the original continuous variable for a logistic regression model.)
- From the Analyze menu, select New View
- From the analysis browser under Contingency Table, select Observed Frequencies and click Create Analysis (or double-click Observed Frequencies)
- Click OK to accept the default parameters
- From the variable browser, select Case Outcome and Daily Coffee.2 and click Add

Observed Frequencies for Case Outcome, Daily Coffee.2

	0.0	1.5	3.5	5.5	Totals
No pancreatic cancer	32	119	74	82	307
Pancreatic cancer	9	94	53	60	216
Totals	41	213	127	142	523

First we will perform a very simple regression analysis to examine the association between coffee drinking and the incidence of pancreatic cancer. We want to see whether coffee consumers tend to be more likely than expected to get the disease. To do this we'll use the dichotomous independent variable Any Coffee, which ignores the level of coffee consumption amongst drinkers.

- Click in the blank area of the view to deselect the frequency table
- In the analysis browser under Logistic regression, select Summary Table, Model Coefficients, and Likelihood Ratio, and then click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent items
- In the Logistic Regression dialog box, check the 95% confidence interval option (turn it on) and click OK
- From the variable browser, select Case Outcome and click Dependent
- From the variable browser, select Any Coffee and click Independent

Logistic Summary Table for Case Outcome

Count	523
# Missing	0
# Response Levels	2
# Fit Parameters	2
Log Likelihood	-350.862
Intercept Log Likelihood	-354.559
R Squared	.010

The summary table reports the number of observations, the number of possible outcomes (two), and details regarding the maximized log likelihood function.

Logistic Model Coefficients Table for Case Outcome

	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	R	Exp(Coef)	95% Lower	95% Upper
Pancreatic cancer: constant	-1.269	.377	-3.362	11.303	.0008	-.115	.281	.134	.589
Any Coffee: Some coffee	.984	.388	2.535	6.426	.0112	.079	2.676	1.250	5.730

Logistic Likelihood Ratio Tests Table for Case Outcome

	DF	Chi-Square	P-Value
Any Coffee	1	7.393	.0065

The model being fit here is given in [Eq. 15.3](#), where the independent variable X_1 is dichotomous and measures whether an individual consumes any coffee ($X_1 = \text{Some coffee}$) or not ($X_1 = \text{No coffee}$). A unit increase in X_1 represents the difference between coffee consumers and abstainers, and the estimated odds ratio comparing these two groups is given by

$$e^{\hat{b}_1} = e^{0.984} = 2.676.$$

The coefficients table also provides the 95% confidence interval for this odds ratio, namely,

$$(e^{0.984 - (1.96 \times 0.388)}, e^{0.984 + (1.96 \times 0.388)}) = (1.250, 5.730).$$

The p value for testing the null hypothesis ($H_0: b_1 = 0$) that coffee consumption is unrelated to incidence of pancreatic cancer is 0.0112. This is known as the **Wald test**, and it is a test of the relationship between an independent and dependent variables based on the size of \hat{b}_1 in relation to the standard error of this estimate. The p value is 0.0065 for the **likelihood ratio test**, which compares the likelihood or deviance of the fitted model including Any Coffee as an independent variable with that of a model that does *not* include it. Both tests suggest some influence of coffee consumption in pancreatic cancer occurrence.

Note that, since these data arise from a case-control study, the estimated intercept coefficient should be ignored. In fact, $e^{\hat{b}_0} / (1 + e^{\hat{b}_0}) = 0.219$, reflecting the frequency of pancreatic cancer cases amongst coffee abstainers in the *dataset* ($9/41 = 0.220$), as designated by the sampling design, and not the frequency of cases in the population.

Since the variable Any Coffee is a simple dichotomous explanatory variable, the logistic model, in this case, does not take advantage of any linear assumption. The results above can thus be directly obtained from the standard 2×2 frequency table relating Any Coffee to Case Outcome, which we can create by adopting variable assignments from the logistic regression analysis:

- Make sure at least one of the Logistic Regression results is selected (has black handles)
- From the analysis browser under Contingency Table, select Observed Frequencies and click Create Analysis (or double-click Observed Frequencies)
- Click OK

Observed Frequencies for Any Coffee, Case Outcome

	No pancreatic cancer	Pancreatic cancer	Totals
No coffee	32	9	41
Some coffee	275	207	482
Totals	307	216	523

Note that the estimated odds ratio comparing pancreatic cancer incidence among coffee consumers and abstainers can be calculated from this table as

$$\frac{32 \times 207}{9 \times 275} = 2.676$$

We now turn to the analysis of a more complex, ordinal variable, Daily Coffee, which takes the level of an individual's daily coffee consumption into account. Since this variable is ordinal and assumes several levels (four), its class is set as continuous. Therefore, the logistic model now invokes a linearity assumption and the results obtained cannot be calculated simply from the observed frequencies table on [p. 208](#). To fit the simple logistic model given by [Eq. 15.3](#), where the independent variable X_1 now stands for Daily Coffee, we clone the analysis with a different independent variable:

- Make sure at least one of the Logistic Regression results is selected (has black handles)
- In the variable browser, select Daily Coffee and Control-Shift-click (Windows) or Command-Shift-click (Macintosh) the Independent button

The following results are obtained:

Logistic Summary Table for Case Outcome

Count	523
# Missing	0
# Response Levels	2
# Fit Parameters	2
Log Likelihood	-354.163
Intercept Log Likelihood	-354.559
R Squared	.001

Logistic Model Coefficients Table for Case Outcome

	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)	95% Lower	95% Upper
Pancreatic cancer : constant	-.479	.169	-2.832	8.019	.0046	.619	.445	.863
Daily Coffee	.043	.048	.889	.791	.3738	1.044	.950	1.148

Logistic Likelihood Ratio Tests Table for Case Outcome

	DF	Chi-Square	P-Value
Daily Coffee	1	.791	.3737

As before, the odds ratio associated with a unit increase in Daily Coffee is given by $e^{0.043} = 1.044$. Recalling the scale of Daily Coffee, it may be more useful to consider the odds ratio associated with an increase of 1–2 or 1.5 units (comparing nondrinkers to those who drink 1–2 cups/day) or 2 units (comparing those who drink 1–2 cups/day to those who drink 3–4 cups/day, or those who drink 3–4 cups/day to those who drink 5 or more cups/day; the latter two odds ratio comparisons are assumed equivalent by this logistic regression model). (You could compare these results to those substituting Daily Coffee.2, the nominal version of the variable.)

For increases of 1.5 or 2 units in Daily Coffee, the relevant odds ratios are obtained from the output as $e^{1.5 \times 0.043} = 1.067$ and $e^{2 \times 0.043} = 1.090$, respectively.

We can easily obtain a Whole Model Fit Table and Logistic Classification Table:

- Make sure that at least one of the results is still selected (has black handles)

- In the analysis browser under Logistic Regression, select Whole Model Fit Table and Logistic Classification Table and click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent items

Logistic Whole Model Fit Table for Case Outcome

	DF	Chi-Square	P-Value
Pearson	2	6.462	.0395
Deviance	2	6.828	.0329
Likelihood Ratio	1	.791	.3737

Two measures of goodness-of-fit are provided, based on Pearson or deviance residuals. Since this dataset contains information on individuals who fall into only four possible independent variable values (0 cups/day, 1–2 cups/day, etc.), both goodness-of-fit statistics are based on comparing the fit of the current logistic model to the so-called saturated model that allows the probability that the outcome variable $Y = 1$ to vary arbitrarily over all distinct levels of the independent variables. The two p values for the goodness-of-fit tests are sufficiently small, 0.0395 and 0.0329 respectively, to suggest that the logistic model in terms of the continuous variable Daily Coffee does not fit the data adequately.

Comparing the maximized log likelihoods seen in the summary tables (–350.862 for the model with Any Coffee, and –354.163 for the model with Daily Coffee) suggests that the data is better described by the Any Coffee model, which allows the risk of pancreatic cancer to be higher for coffee drinkers but not to vary with the amount of coffee consumed per day. However, the noticeable lack of a dose response in the Daily Coffee model casts substantial doubt on the biological plausibility of the apparent relationship between coffee drinking and the incidence of pancreatic cancer.

Logistic Classification Table for Case Outcome

	Predicted No pancreatic cancer	Predicted Pancreatic cancer	Percent	Correct
Observed No pancreatic cancer	307	0	100.00%	
Observed Pancreatic cancer	216	0	0.00%	
Overall			58.70%	

Finally, we consider the Logistic Classification Table. The results here are the same as they would be for the model with Any Coffee. In both models, the predicted binary outcomes given in the table are obtained by calculating for each observation the estimated response probability \hat{p}_x from [Eq. 15.4](#), with the relevant estimated parameter values, \hat{b}_0 and \hat{b}_1 , substituted. For a given case, the predicted outcome is 1 if $\hat{p}_x > 0.5$ and 0 otherwise. The value 0.5 is often called the **prior probability** that an observation has the response $Y = 1$. In many settings, a different prior probability might be more appropriate. Here, all observations have predicted value 0, in part because about 59% (i.e., $307/523$) of the data are controls with $Y = 0$. In this case, the apparent association of coffee drinking with the incidence of pancreatic cancer is not sufficiently strong to raise any predicted probability that $Y = 1$ above 0.5. The predicted responses would change somewhat with the choice of a different prior, but it is important to note that the overall rate of correct classification (here 58.7%) is not a measure of goodness-of-fit but a reflection of residual variation—variation in the dependent variable that remains after accounting for coffee drinking behavior. In this regard, note that a substantial number of coffee drinkers are controls.

Multiple logistic regression

Our next example illustrates the use of several independent variables in a logistic regression model. The data arise from a randomly selected subset of 614 participants in the Western Collaborative Group study, described in Rosenman *et al.* (1975) among others. Individual observations correspond to male employees selected from ten California companies during the years 1960–61. At recruitment, covariates such as smoking practices and behavior type were measured for each participant. Subsequently, each study subject was followed for up to nine years to determine whether an individual had a coronary heart disease (CHD) event as determined by a medical expert.

- Open WCGS Data from the Sample Data folder

	Cigarettes	Personality Type	Censor	Time
Type:	Integer	Category	Integer	Integer
Source:	User Entered	User Entered	User En...	User E...
Class:	Continuous	Nominal	Nominal	Contin...
Format:
Dec. Places:
1	25	A	1	1664
2	20	B	1	3064
3	0	B	1	3102
4	9	B	0	2426
5	0	A	1	3070
6	0	A	1	3101
7	0	B	1	3101

Cigarettes gives the reported number of cigarettes smoked per day at recruitment. Personality Type gives a measure of personality type as assessed by interview; this is a nominal variable where Type A is considered more aggressive and competitive, and Type B is considered more relaxed and noncompetitive. Censor takes the value 1 if the participant was not subject to coronary heart disease throughout follow up and 0 otherwise. (Censor is not used here but is relevant to a survival analysis of this dataset; see [“Exercise,” p. 191](#) in the [“Survival: Regression” chapter](#).) Finally, Time measures the number of days from entry into the study until occurrence of heart disease or end of follow up, whichever occurs first.

First, we create a new variable, CHD Outcome, that recodes the CHD event cases as 1 and the no CHD event cases as 0.

- From the Manage menu, select Formula
- Specify this formula and click Compute:
1 – Censor
- Rename the variable: CHD Outcome
- Change its Type from Real to Integer
- Change its Class from Continuous to Nominal

Now we analyze the data:

- From the Analyze menu, select New View
- In the analysis browser under Logistic regression, select Summary Table, Model Coefficients, and Likelihood Ratio, and then click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent items
- Check the 95% confidence interval option and click OK

- In the variable browser, select Cigarettes and click Independent
- In the variable browser, select CHD Outcome and click Dependent

Logistic Summary Table for CHD Outcome

Count	614
# Missing	0
# Response Levels	2
# Fit Parameters	2
Log Likelihood	-194.587
Intercept Log Likelihood	-196.507
R Squared	.010

Logistic Model Coefficients Table for CHD Outcome

	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)	95% Lower	95% Upper
1 : constant	-2.459	.189	-12.994	168.845	<.0001	.086	.059	.124
Cigarettes	.017	.009	2.005	4.021	.0449	1.017	1.000	1.035

Logistic Likelihood Ratio Tests Table for CHD Outcome

	DF	Chi-Square	P-Value
Cigarettes	1	3.840	.0501

The coefficient table shows an estimated logistic regression coefficient for “Cigarettes” of 0.017, indicating an odds ratio for CHD of $e^{0.017} = 1.017$ associated with an increase in consumption of one cigarette per day at study entry. This is not a particularly useful risk group comparison since a single extra cigarette smoked per day would not be expected to increase an individual’s risk substantially. The estimated odds ratio associated with an increase of 20 cigarettes per day is easy to compute from the information provided as $e^{20 \times 0.017} = 1.405$, yielding an approximate increase in risk of 40%.

Now we add the second independent variable Personality Type to see whether this provides any additional explanatory information.

- Be sure at least one of the results is still selected (has black handles)
- In the variable browser, select Personality Type and click Independent

Logistic Summary Table for CHD Outcome

Count	614
# Missing	0
# Response Levels	2
# Fit Parameters	3
Log Likelihood	-192.070
Intercept Log Likelihood	-196.507
R Squared	.023

Logistic Model Coefficients Table for CHD Outcome

	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)	95% Lower	95% Upper
1: constant	-2.157	.223	-9.679	93.685	<.0001	.116	.075	.179
Cigarettes	.015	.009	1.746	3.048	.0808	1.015	.998	1.032
Personality Type: B	-.629	.286	-2.202	4.848	.0277	.533	.304	.933

Logistic Likelihood Ratio Tests Table for CHD Outcome

	DF	Chi-Square	P-Value
Cigarettes	1	2.935	.0867
Personality Type	1	5.035	.0248

Note that the estimated odds ratio associated with a pack per day increase in cigarette consumption is now $e^{20 \times 0.015} = 1.350$; the similarity of this estimate to the one obtained from the model that only included “Cigarettes” indicates that there is little confounding effect of Personality Type on our understanding of the relationship between cigarette consumption and CHD incidence. In other words, the apparent association of cigarette consumption and CHD in the first model is not “explained away” on the basis of individuals’ personality types. The odds ratio for CHD associated with being Type A as against Type B, controlling for cigarette consumption, is estimated to be $e^{-0.629} = 0.533$, with an associated 95% confidence interval of $(1.072, 3.289) = (0.933^{-1}, 0.304^{-1})$. It is usually preferable to describe the odds ratio in terms of how much greater the risk is in the higher risk group (here, Type A) rather than the other way around. (To learn how to code your data to get the comparisons you want, see [“Nominal data coding,” p. 206.](#))

Further analyses of this dataset in the [“Survival: Regression” chapter](#) incorporate the Time variable. See the [“Exercises,” p. 207.](#)

Polytomous logistic regression

This exercise illustrates how to fit a polytomous logistic regression model. The data are from a prospective study of the findings of a colonoscopy screening study on individuals considered to be at high risk of colon cancer, from Grossman, et al. (1989). The purpose of the study was to determine the role of past history—for example, a history of previous colon lesions, a family history of cancer, age, etc.—in predicting the findings of a current colonoscopy. The cases considered here correspond to 406 individuals who had adenoma findings in previous colon examinations and who are therefore considered to be at high risk of a subsequent significant finding.

- Open Colonoscopy from the Sample Data folder

The dependent variable Finding assumes three levels: 2 corresponds to a significant finding on the screening, namely a large tubular adenoma (>1 cm in diameter) or an advanced neoplasm; 1, to a finding of small tubular adenoma, and 0, to a negative examination. The independent variable of interest is age at the time of screening. Age ranged from 30–39 years (coded as 35) to 70 years and older (coded as 75).

- From the Analyze menu, select New View
- In the analysis browser under Logistic regression, select Summary Table, Model Coefficients, and Likelihood Ratio, and then click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent items
- Check the 95% confidence interval option and click OK
- From the variable browser, select Finding and click Dependent
- From the variable browser, select Age and click Independent

Logistic Summary Table for Finding

Count	406
# Missing	0
# Response Levels	3
# Fit Parameters	4
Log Likelihood	-314.562
Intercept Log Likelihood	-327.030
R Squared	.038

Logistic Model Coefficients Table for Finding

	Coef	Std. Error	Coef/SE	Chi-Square	P-Value	Exp(Coef)	95% Lower	95% Upper
1 : constant	-3.911	.810	-4.828	23.310	<.0001	.020	.004	.098
Age	.045	.013	3.570	12.742	.0004	1.046	1.021	1.073
2 : constant	-6.771	1.415	-4.784	22.890	<.0001	.001	7.157E-5	.018
Age	.074	.021	3.483	12.130	.0005	1.077	1.033	1.123

The Logistic Model Coefficients Table gives an estimate of age effects on colonoscopy findings in terms of a *unit* or *one-year* increase in age. We can calculate the odds ratio associated with a 20-year increase when comparing the chances of a small adenoma (1) against a negative finding (0) as $e^{20 \times 0.045} = 2.460$. We can calculate the 95% confidence interval for that odds ratio as $(1.021^{20}, 1.073^{20}) = (1.515, 4.093)$. Similarly, the odds ratio for an increase of 20 years of age in comparing the chances of a large adenoma or neoplasm (2) versus a negative finding (0) would be $e^{20 \times 0.074} = 4.393$. However, the odds ratio associated with the 20-year age increase when comparing major (2) to minor (1) findings is only $e^{20 \times (0.074 - 0.045)} = 1.786$.

We can assess the statistical significance of age in these pairwise comparisons by examining the two p values given in the Logistic Model Coefficients Table. However, the overall effect of age on the colonoscopy outcome is best assessed by the likelihood ratio test with two degrees of freedom, which yields a p value that is less than 0.001.

Logistic Likelihood Ratio Tests Table for Finding

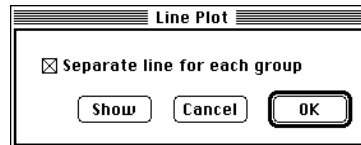
	DF	Chi-Square	P-Value
Age	2	24.936	<.0001

In summary, it appears that age is strongly related to the chances of a positive findings on a colonoscopy screening examination. Given a positive finding, however, increased age is only moderately associated with an increased chance of a significant finding as against a less major adenoma.

Univariate Plots

Univariate plots show the distribution of a variable in a plot with a single numeric axis, the Y axis. Each observation is plotted along the horizontal axis in the sequence the data appears in the dataset. You can display the observations as points in a scattergram, as points connected by lines in a line chart, or as bars in a bar chart. You can plot multiple variables in a single univariate plot and use split-by variables to distinguish different groups within the variables. You can also add reference lines to show the variable's mean plus or minus one or more standard errors or standard deviations as well as a specified confidence interval. Univariate charts with standard deviation reference lines are identical to individual measurement quality control charts.

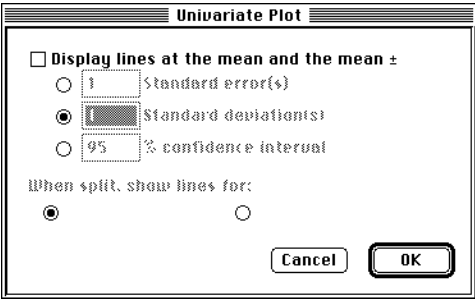
If you are using split-by variables you can specify whether to display a separate line for each group or a single line for all groups. This choice is in the Line Plot dialog box (described below), accessible through the Create Analysis button, or by clicking Edit Analysis when the entire graph is selected. If a univariate line plot displays information on several groups, the plot shows separate lines for each group or one line for all groups. To change this setting, click on one of the points to select just the plot, and click the Edit Display button. You see this dialog box:



By default there are no values displayed on the horizontal axis, but you can optionally choose to display observation numbers on this axis. The observation number ranges from one to n , where n is the number of non-missing, non-excluded values in the variable. The first such value has observation number 1, the second observation number 2, etc. For other modifications you can make to this graph, see [“Customizing results,” p. 179 of Using StatView](#).

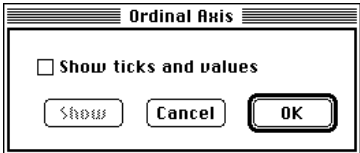
Dialog box settings

When you create a univariate plot or edit it using the Edit Analysis button, you see the dialog box below. You can add lines for the mean, standard deviations, standard error and confidence intervals.



If you choose to display lines at the mean, you must also display lines around the mean at a specified standard error, standard deviation, or confidence interval. The option at the bottom of the dialog box determines how these lines appear when you assign a split-by variable.

There is an additional setting for univariate plots, found in a separate dialog box. By default, the horizontal axis has no ticks or values displayed on the axis. You choose to add an axis whose value ranges from 0 to the count of values for the variable displayed. Select the horizontal axis by clicking on it. Click Edit Display and the Ordinal Axis dialog box appears:



Click the checkbox to show ticks and values and click OK. To preview the change first, click the Show button. You cannot modify other aspects of this axis.

Data requirements

Univariate plots can be generated for one or more continuous or nominal variables.

Variable browser buttons	
Add	To generate a univariate plot, select one or more variables and click Add. Each additional variable assigned is added to the same plot.
Split By	The cells of any nominal variable(s) assigned using the Split By button appear in the legend.

Results

The default univariate plot is a scattergram.

Scattergram	Shows observations as points. Lines indicating the mean, standard deviations, standard error and confidence intervals can be added to the plot with Edit Analysis.
-------------	--

Line Chart	Shows observations as points connected by lines. Lines indicating the mean, standard deviations, standard error and confidence intervals can be added to the plot with Edit Analysis.
Bar Chart	Shows observations as bars. Lines indicating the mean, standard deviations, standard error and confidence intervals can be added to the plot with Edit Analysis.

Templates

The following templates provide univariate plots.

Graphs	Univariate Bar Chart	Univariate bar chart for continuous variable and optional Split By variable.
	Univariate Line Chart	Univariate line chart for continuous variable and optional Split By variable.
	Univariate Scattergram	Univariate scattergram for continuous variable and optional Split By variable.

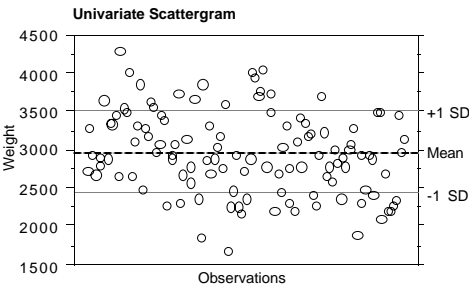
Exercise

In this exercise you create a univariate scattergram to examine the distribution of car weights. The dataset you will use has measurements of weight, gas tank size, turning circle, horsepower and engine displacement for 116 cars from different countries.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Univariate Plots, select Scattergram and click Create Analysis

If you did not wish to display additional information you would click OK without checking any other options. For this example you will add a line at the mean as well as at one standard deviation above and below the mean.

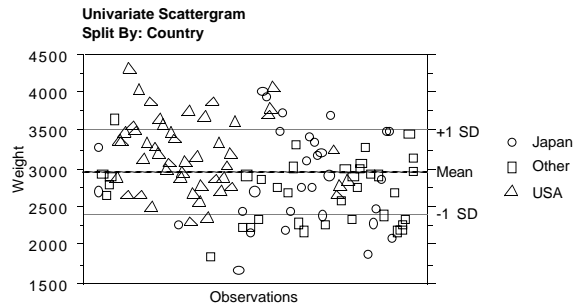
- Check Display lines and click OK
- In the variable browser, select Weight and click Add



The graph displays the individual observations of each car's weight along with lines indicating the variable mean and values at plus and minus one standard deviation. Visual inspection shows that approximately 50 cars fall outside plus or minus one standard deviation of the mean.

This dataset also includes a nominal variable identifying the manufacturing country for each car. We can use this variable to split the observations into different groups.

- Make sure the graph is still selected
- In the variable browser, select Country and click Split By



The three different countries, Japan, Other, and USA, are distinguished by different plotting symbols. You can see that most of the USA cars are heavier than average.

You can display this information as a line chart or bar chart by choosing the appropriate graph from the analysis browser. You can also draw different mean and standard deviation lines for each group rather than for the entire variable by clicking the Edit Analysis button and changing that parameter in the dialog box.

Bivariate Plots

A **bivariate plot** graphs the relationship between two variables, X and Y. It can display the observations as points with or without connecting lines.

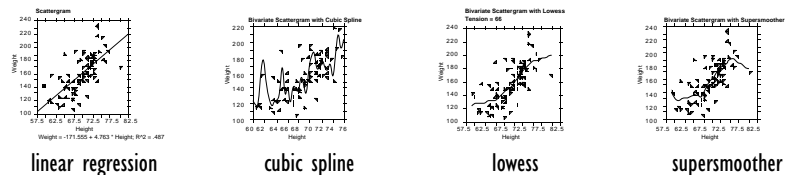
In a bivariate plot each individual observation Y_i is plotted against X_i for $i = 1$ to n , the number of observations of X and Y. You can plot multiple variable pairs in a bivariate plot and use split-by variables to distinguish different groups. You can also use nominal variables in a bivariate plot to construct a point graph that distinguishes the measurements of the groups of a nominal variable.

You can add a simple **regression line** to the bivariate plot with or without **confidence bands** around the mean and slope of the regression line. **Cubic spline**, **lowess**, and **supersmoother** fits are also available for bivariate plots. You can plot different subgroups of your data by adding a split-by variable, displaying a single fitted line for the entire graph or displaying separate lines for each subgroup. You can plot more than one pair of variables on a single graph and display fitted lines for each.

Edit Display lets you modify the structure and appearance of bivariate plots; see [“Customizing results,” p. 179 of Using StatView](#).

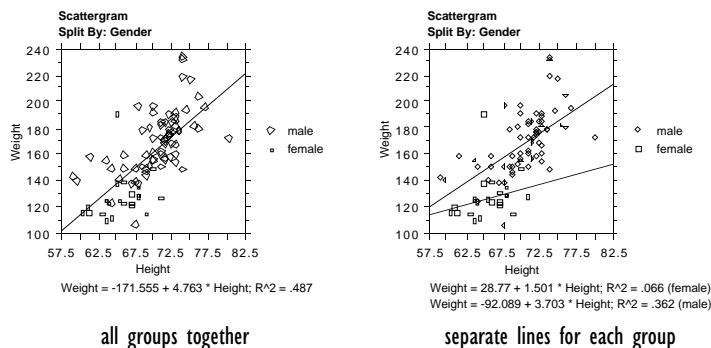
Fitted lines

StatView offers four curve fits for bivariate plots: cubic spline, lowess smoother, supersmoother, and regression (with or without confidence bands). We discuss each separately below, after demonstrating how the various types of fitted lines work in StatView and offering some general cautions. In this discussion, we’ll look at the various ways of fitting curves to a plot of weight against height from the Lipid Data.



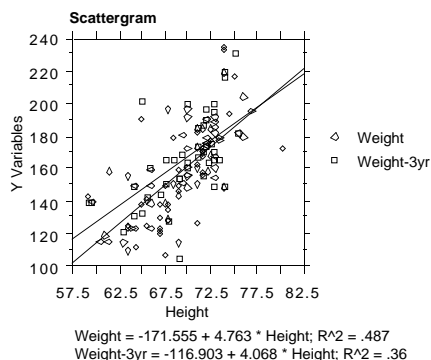
Split-by variables

You can plot different subgroups of your data by adding a split-by variable, displaying a single fitted line for the entire graph or displaying separate lines for each subgroup.



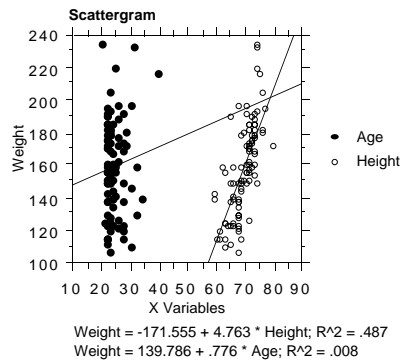
Multiple variables

You can plot more than one pair of variables on a single graph. StatView always displays separate fitted lines for each pair.



Watch out for dissimilar scales

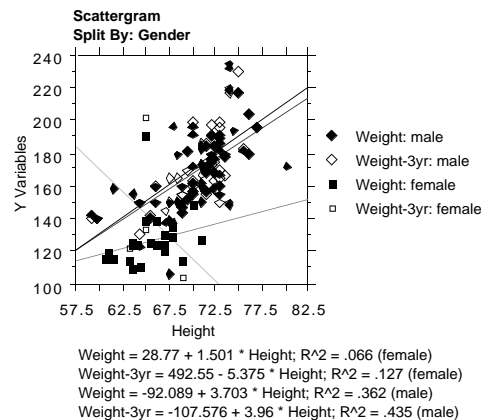
When combining several variable pairings in a single graph, you should beware of misleading distortions caused by differing scales among the variables. For example, in the plot above, two Y variables (original weight and weight after three years) can be combined in a single plot because they measure the same individuals on the same scale, and showing them together helps us compare the regression line slopes. However, if we include age on the x axis as well, the dissimilar scales cause all the points for each variable to be squeezed together in thin bands:



This plot is misleading, because it makes it look like height and weight have strong relationships and steep slopes. You can tell from the equations shown below the graph that the fit actually accounts for relatively little variance and the slopes are not especially steep, but you shouldn't count on your readers looking that closely. If variables have dissimilar scales, you should graph each pair of variables separately.

Less is more

Avoid cramming too much information into a single graph. For example, the second plot of weight against height split by gender on [p. 222](#) allows us to compare height-weight relationships between men and women, and it suggests that men are heavier than women of the same height and also that taller men are proportionally heavier than women—neither of which are surprising, since men generally have broader builds than women. The plot of Weight and Weight-3yr by height on [p. 222](#) lets us compare the height-weight relationships for subjects at the outset of a medical study and again after three years. While the subjects as a whole seem to have lost a slight amount of weight, the relationship between height and weight seems to be about the same. However, the following plot, which attempts to compare initial and third-year weights between men and women, attempts too much. Even with different plotting symbols and line types, this plot is difficult to interpret:

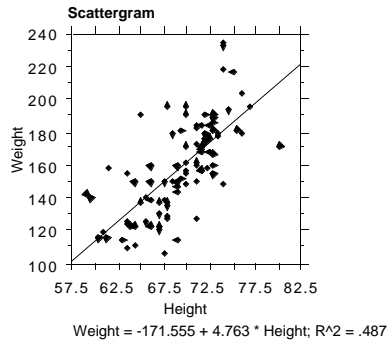


Linear regression

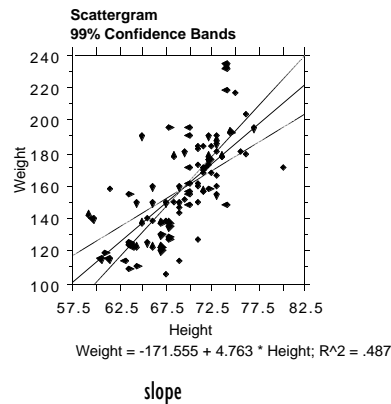
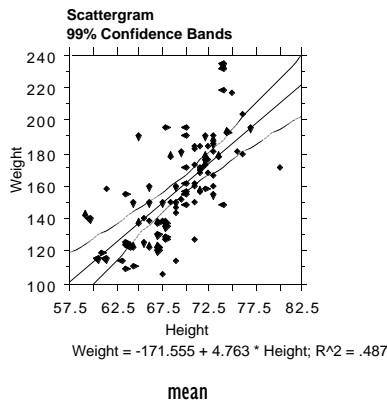
A regression line is a simple description of the relationship between Y and X having the general form

$$Y = b + mX,$$

where b is the **intercept** (where the line crosses the Y axis at $X = 0$), and m is the **slope** (change in Y divided by change in X). The exact equation for the line and the value of R^2 (which tells how much of the variance is accounted for by the model) are shown at the bottom of the graph.



You can add confidence bands for the mean and/or the slope:



A bivariate plot with a regression line is an excellent graph to use in conjunction with regression analysis to check visually how well a model fits the data. The Regression analysis also offers a regression plot, but that plot is limited to one independent and one dependent variable. The Bivariate Plot analysis offers more flexibility. You can combine more than one independent (X) and dependent (Y) variable in a single plot, with separate regression lines for each X - Y pairing. You can also plot subgroups of your data by adding a split-by variable, displaying either a single fitted line for all the points or separate lines for each subgroup of points.

Keep in mind that a scattergram with regression lines is just a scattergram with fitted lines. If you need to see summary information, an ANOVA table, residuals, or other such information

you must perform a regression analysis. For more information, see the chapter [“Regression,” p. 51.](#)

Smoothing bivariate plots

Researchers often find it useful to add other types of fitted curves to bivariate plots. StatView offers three forms of smoothing: cubic spline, lowess, and supersmoother.

Reasons for smoothing a graph are many. Sometimes a fitted curve is useful for reducing distraction from “noise” or random error in a plot and getting an idea of which nonlinear models might be effective for prediction. Sometimes the goal is to emphasize the shape in a plot, so that the graph is more effective as a presentation tool. Other times, smoothing is used to help interpolate values falling between a dataset’s cracks.

Most smoothers work by estimating an “average” value for y a few x values at a time, such as the first four points, then estimating again for second through fifth points, then the third through sixth points, etc. (Here, “average” does not necessarily refer to the arithmetic mean but rather to whichever estimation method is being used.) The points being included in each calculation are called the **window**, and the number of points included at a time is the **window width**. A wider window produces a “smoother,” stiffer curve. A narrower window produces a looser curve that clings more tightly to the data points.

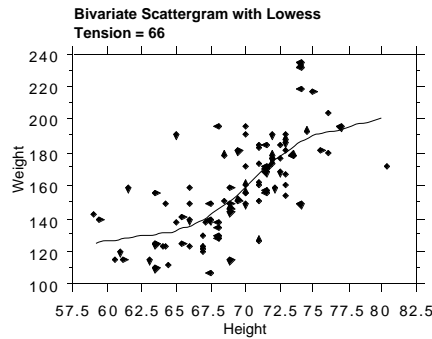
StatView’s cubic spline smoother uses a window width of four points. Lowess uses the proportion of the dataset you specify. Supersmoother calculates its own variable window width by examining local slopes and variances at different regions along the x axis.

Researchers usually find a smoother curve easier to interpret and describe—for example, one might observe from a downward-sloping linear regression, “As the dose of medication increases from 0 to 60 mg, the blood pressure drops.” However, a looser fit sacrifices less detail—for example, from a lowess curve with a low tension setting one might observe, “Doses between 0 and 5 mg have negligible effect, while doses from 5 to 50 mg cause a steady decrease in blood pressure, after which the effectiveness of the drug reaches a plateau; doses above 60mg are toxic.”

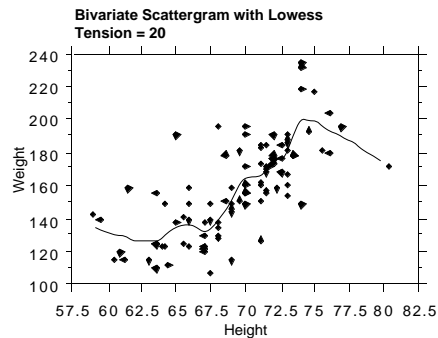
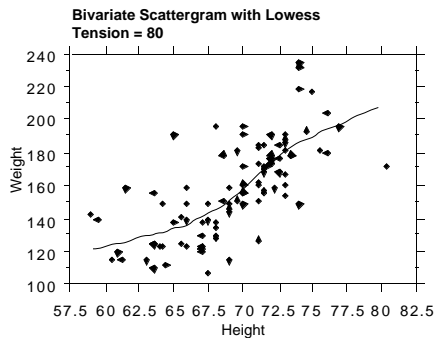
Keep in mind that a smoothing is just a simplification of your data; it may or may not make any sense in the context of the phenomena you are studying. The type of smoothing you choose, the parameters you set, and the way you interpret the graph should all take into account what your data mean, what your goal is in smoothing the graph, and what you learn from other methods of analyzing the data.

Lowess

A more robust smoothing procedure than simple linear regression is locally weighted regression known as **lowess** (LOcally WEighted Scatterplot Smoother). StatView offers a least-squares fit for lowess curves. For details on how lowess is computed, see Cleveland (1981); simply looking at a few plots will give you an intuitive understanding of the procedure.



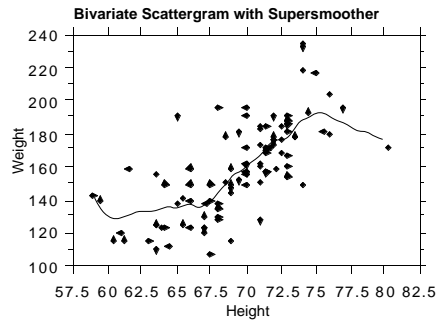
A **tension** parameter between 1 and 100% controls how tightly the curve follows the data. The value you specify determines how many of the graph's points are included in the window for each computation. A higher tension (e.g., 80) produces a tighter, straighter curve; a lower tension (e.g., 20) produces a looser curve more strongly influenced by nearby data points. You should choose a tension high enough to produce a smooth curve but low enough to convey the shape of the data accurately. The default is 66.



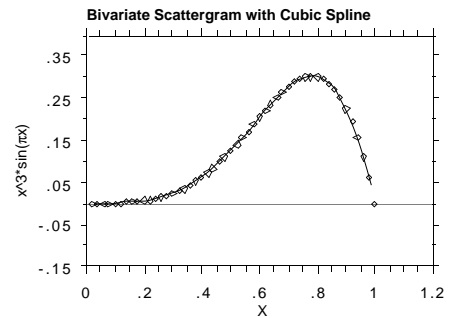
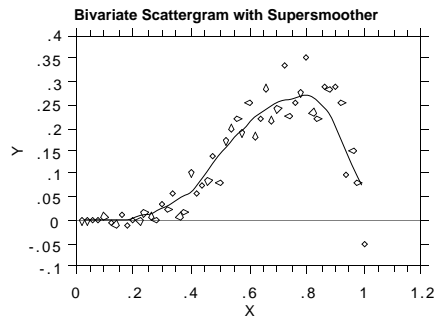
No matter what your goals in smoothing a bivariate plot, you would be well advised to start with a lowess smoothing of your data, so that you don't overlook any patterns that might be obscured by linear regression or the other smoothers. Notice how the lowess fit of weight by height calls attention to an S-shape in the data. Another advantage of lowess is its robustness to outliers.

Supersmoother

Supersmoother is a smoothing method designed to vary its own tension locally to suit the data. In areas of greater curvature or smaller variance, it uses less smoothing; in areas of lesser curvature or greater variance, it uses more smoothing. In this way, the supersmoother seeks to reveal the underlying relationship between x and y values in "signal-noise" data—data where y values are thought to be some function of x plus random error. Supersmoother uses a local cross-validation technique to determine how much smoothing (the **span** of the smoother) is needed in each region along the x axis.

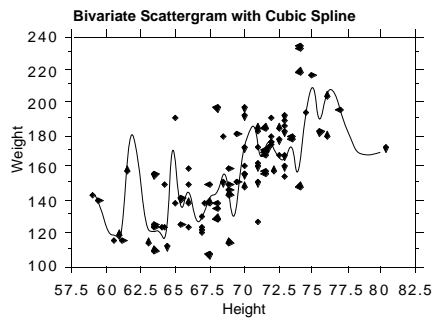


Supersmoother is especially effective for hiding the noise in y values that are thought to be some function of the x values. For example, below is a supersmoother fit for data generated by a formula as $y = x^3 \sin(\pi x)$ plus uniform random noise. Notice that the curve's shape is quite close to that of the underlying function itself (from a formula without noise).



Cubic spline

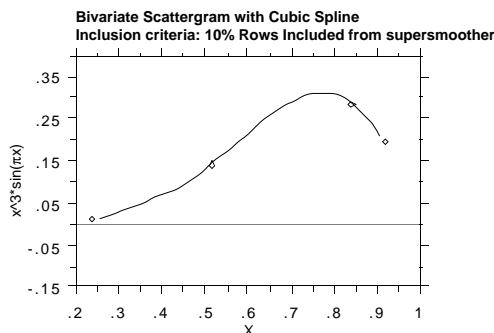
Cubic spline smoothing fits a series of cubic (third-order) polynomials to fit a moving window of data, four points at a time. These polynomials are connected to produce a smooth curve passing through each of the actual data points whenever possible.



Because the Weight by Height plot has multiple Y values for some values at X , it isn't possible for cubic spline to connect all points.

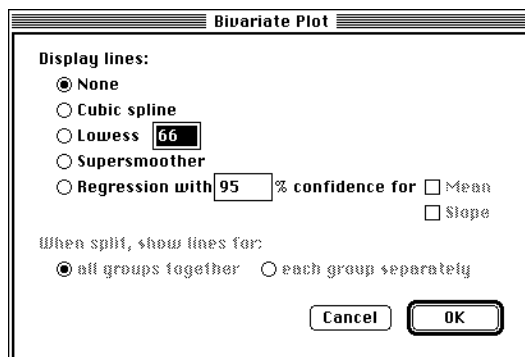
Cubic spline fits are useful for interpolating values between data points, since they usually connect the actual data points. (Other smoothing methods convey the shape of the plot without necessarily coinciding with any measured values.) For that very reason, however, cubic splines are only useful when you believe your data measurements contain little or no error, since large amounts of scatter from the “true” values would unduly influence the shape of the curve and cause it to be misleading.

Consider how well this spline fitting of just four points chosen at random (using Random Criteria, discussed under [“Criteria pop-up menu,” p. 128 of Using StatView](#)) along the function $y = x^3 \sin(\pi x)$ approximates the actual curve above:



Dialog box settings

When you create a bivariate plot or edit it using the Edit Analysis button, you see the dialog box below.



Display lines You can optionally add one of four types of fitted lines to your graph; for a discussion, see [“Fitted lines,” p. 221](#). For lowess, specify the percentage of the dataset (between 1 and 100) to include in each window. The default is 66%.

If you display regression lines, you can also specify a confidence level and show confidence bands for the mean of Y as predicted by the regression for a given value of X . You can also

show confidence bands for the slope of the regression line. The default is 95% confidence, but you can specify a different percentage.

When split, show lines for When the graph is split by one or more nominal variable(s), values for each group of subgroup of the data are shown by different plotting symbols, colors, or fill patterns. You can choose fitted lines (regression, cubic spline, lowess, or supersmoother) that are computed and drawn separately for each (sub)group or a single fitted line for all the data. For graphs with more than one *X-Y* variable pair, fitted lines are always computed and drawn separately for each pair; this option only applies to split-by variables.

Data requirements

Bivariate plots can be generated for one or more continuous or nominal X variables vs. one or more continuous or nominal Y variables. Fitted lines are only available for plots of continuous variables.

If there is a single X variable and more than one Y variable, each Y variable is plotted against the X variable. The same rule applies if there is a single Y variable and more than one X variable. If multiple X and Y variables are plotted, the first X assigned is plotted against the first Y, the second X against the second Y, and so on.

Variable browser buttons	
X Variable	Select one or more variables and click X Variable. Additional variables are added to the same plot.
Y Variable	Select one or more variables and click Y Variable. Additional variables are added to the same plot.
Split By	The groups of any nominal variable(s) assigned using the Split By button appear in the legend.

Results

The default plot is a scattergram.

Scattergram	Shows one point for each X-Y pair. Regression lines, confidence bands, and equations, or cubic spline, lowess, or supersmoother fits may be added for the entire plot or for each group of a split-by variable.
Line Chart	Shows one point for each X-Y pair. The points are connected by lines. Regression lines, confidence bands, and equations, or cubic spline, lowess, or supersmoother fits may be added.

Templates

The following templates provide bivariate plots.

Graphs	Bivariate Line Chart	Line chart for continuous X and Y variables and optional split-by variable.
	Bivariate Line Chart (Groups)	Line chart for nominal X and continuous Y variables and optional split-by variable.
	Bivariate Regression Plot	Scattergram with regression line and equation for continuous X and Y variables and optional split-by variable. If you assign a split-by variable, you get separate regression lines for each group.
	Bivariate Scattergram	Scattergram for continuous X and Y variables and optional split-by variable.
	Bivariate Scattergram (Groups)	Scattergram for nominal X and continuous Y variables and optional split-by variable.
	Cubic Spline Fit	Scattergram with cubic spline fit for continuous X and Y variables and an optional split-by variable. If you assign a split-by variable, you get separate fitted lines for each group.
	Lowess Curve Fit	Scattergram with lowess curve fit (tension=66%) for continuous X and Y variables and an optional split-by variable. If you assign a split-by variable, you get separate fitted lines for each group.
	Scatter Matrix 3x3	3x3 matrix of scattergrams, with one scattergram for each X-Y pairing of continuous variables.
	Scatter Matrix 4x4 w Histograms	4x4 matrix of scattergrams, with one scattergram for each X-Y pairing of continuous variables; diagonal cells have histograms with normal curves.
	Scatter w Histograms	Scattergram for continuous variables; has histograms with normal curves along top and right sides.
	Supersmoother Curve Fit	Scattergram with supersmoother fit for continuous X and Y variables and an optional split-by variable. If you assign a split-by variable, you get separate fitted lines for each group.

Exercises

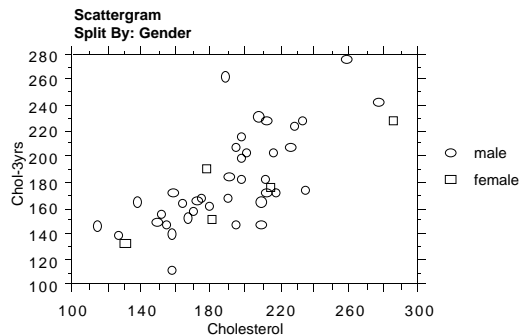
Bivariate scattergram

The dataset used in this chapter’s discussions, Lipid Data, records blood lipid levels and other cardiovascular risk factors measured for medical students when they were freshmen and again when they were seniors. In this exercise you examine the relationship between freshmen’s cholesterol counts and those taken three years later, after they had received instruction on reducing cholesterol through dieting. You will also examine whether this relationship is the same for male and female students.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Click OK to accept the default parameter settings
- In the variable browser, select Cholesterol and click X Variable
- Select Chol-3yrs and click Y Variable

The dataset includes a nominal variable which identifies the gender of the student. We can use this variable to split the observations into the different groups.

- In the variable browser, select Gender and click Split By



The male and female observations are distinguished by different plotting symbols, colors, or fill patterns, depending on your Graph Preferences.

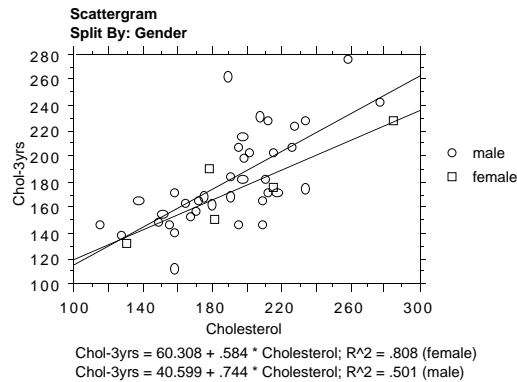
Linear regression

To determine whether a different relationship exists between the cholesterol levels for males and females we can calculate a simple regression and add the fitted line to the graph.

- Click Edit Analysis

We have the option of displaying a single line for all observations or calculating a different regression for each group. We want separate lines for each group, so we can compare males and females.

- Choose Regression lines
- Choose Each group separately and click OK

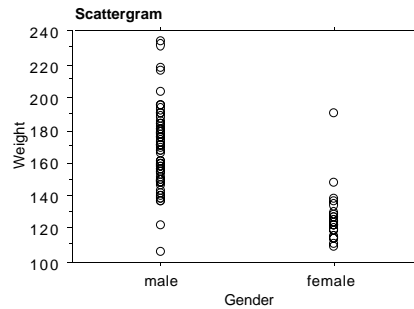


Notice that the equation for each line as well as the R^2 values are added to the bottom of the graph. You can see there is a slight difference between males and females. The difference is not significant, which you can see if you add confidence bands for the mean (Click Edit Analysis to see the dialog box). If we showed the regression line for all groups together there would be only a single regression line and the simple regression would be calculated using all the data.

Bivariate plot with nominal data

Bivariate plots can have nominal as well as continuous variables assigned to the X or Y axis. When assigning a nominal variable, you can construct a graph to compare the different distributions of the data in each of the nominal variable's groups. The previous example showed a difference between the cholesterol reduction in male and female students. We can use the bivariate plot to examine the differences between the weights of the male and female students.

- Make sure that Lipid Data is still open
- If any results are still selected, click in the empty area of the view window to turn the selection off
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Click OK to accept the default analysis parameters
- In the variable browser, select Gender and click X Variable
- Select Weight and click Y Variable

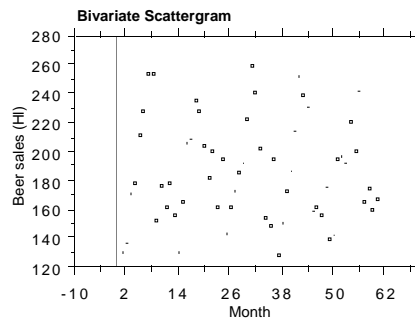


As you might expect, weights for female students are less than those of male students.

Cubic spline

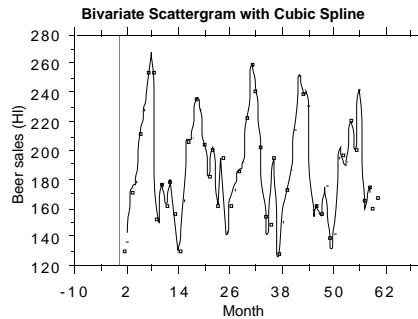
Next we will examine time series data from Neter, Wasserman, and Whitmore (1988). Beer Sales records monthly sales of beer in hectoliters, along with the average high and low temperatures in the region, over a period of five years. First, let's use a cubic spline to see how beer sales change over time.

- Open Beer Sales from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Click OK to accept the default parameters
- In the variable browser, select Month and click X Variable
- Select Beer Sales (Hl) and click Y Variable



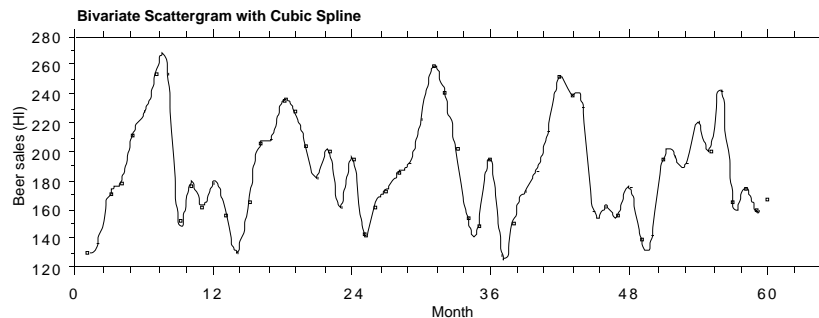
It is hard to see much of anything in this plot. Adding a cubic spline is often helpful for visualizing patterns in time series data.

- Make sure the graph is still selected
- Click Edit Analysis
- Choose Cubic spline and click OK



Now we can see that the sales follow a pretty clear seasonal pattern each year. However, we can still make this plot easier to read. Many researchers find it helpful to make plots of time series data several times wider than they are tall, especially when the data are periodic with several cycles. For monthly data, they prefer axis ticks at each year.

- Click and drag the selection handles on the right axis to the right to make the graph wider
- Select the X-axis and click Edit Display
- Specify bounds of 0 and 61 for From and To
- Specify 12 for Major interval width
- Click OK

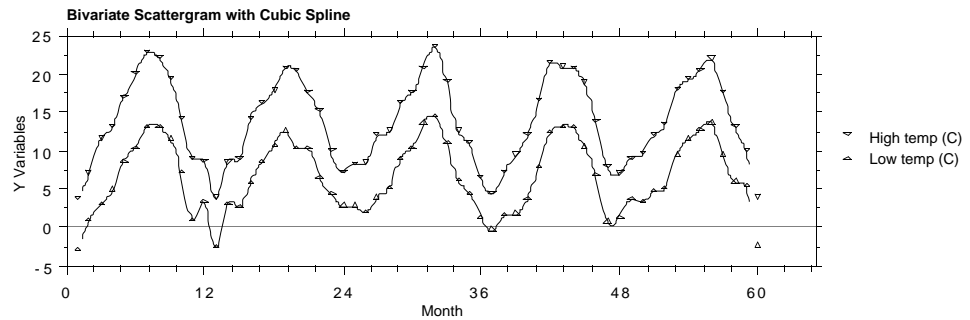


Notice how the cubic spline curve guides your eye and makes a seasonal trend apparent. We can easily see that beer sales are at their lowest each January, and they tend to rise steadily from winter to summer, dropping again in the autumn.

Having discovered a seasonal pattern to beer sales, we might want to look for a relationship between beer sales and temperatures. Do people buy more beer when the weather is hotter? Let's clone this cubic spline plot for high and low temperatures.

- Make sure the graph is still selected
- In the variable browser, select both High temp (C) and Low temp (C)
Shift-click or click and drag to select several adjacent variables.
- Control-Shift-click (Windows) or Command-Shift-click (Macintosh) the Y Variable button

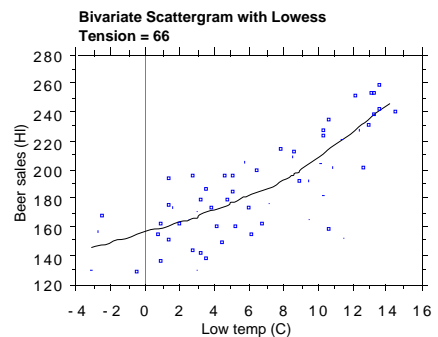
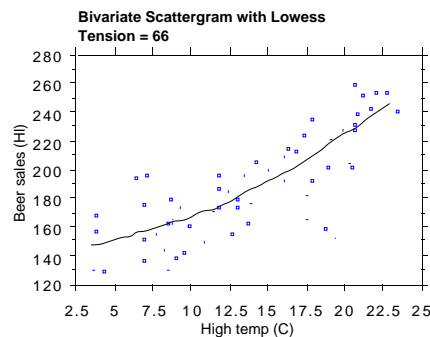
Next, adjust the size and axis settings for the graph the same way you did on the last plot. You might also want to use the Draw Palette to choose special plotting symbols, as we did.



Lowess fit

Beer sales certainly do seem to follow the same pattern as temperatures do over time. Can we use the temperatures to predict beer sales? Lowess would be a good way to start looking for a relationship.

- Click in the blank area of the view to make sure nothing is selected
- In the analysis browser under Bivariate Plots, select Scattergram and click Create Analysis
- Choose Lowess and click OK
- In the variable browser, select High temp (C) and click X Variable
- Select Beer Sales (HI) and click Y Variable
- Select Low temp (C) and Control-Shift-click (Windows) or Command-Shift-click (Macintosh) the X Variable button

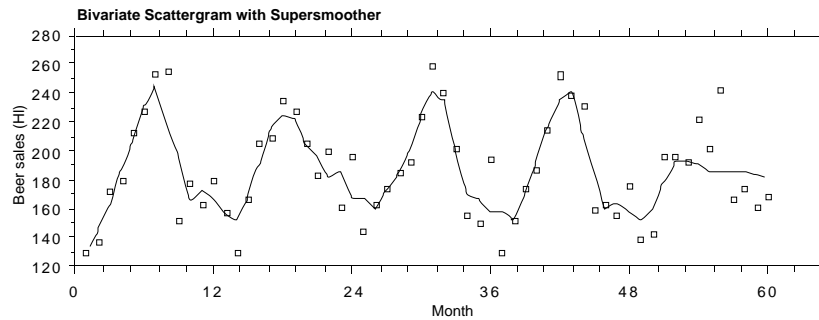


These lowess curves suggest that temperatures might indeed be a good predictor for beer sales.

Supersmoother

Time series data are often so “jaggy” that it is difficult to detect patterns in the data. Smoothing methods such as supersmoother can simplify the plot.

- In the view window, select the plot of Beer sales (HI) by Month
- Click Edit Analysis
- Choose Supersmoother and click OK



Cell Plots

Cell **plots** graph means or sums of variables and can show the variability around means. They are useful for showing the side by side comparison of continuous variables measured for each of several nominal groups.

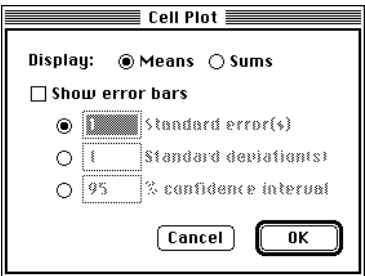
When your data fall into groups, it is common to question whether some factor affects the groups in the same way or affects each group differently. You may know that the means of the two groups are different, but you also want to know the effect of one or more additional factors on the relationship. Cell plots present a set of lines, bars or points so you can visually compare variable to variable and group to group. This is extremely useful in conjunction with any statistic that tests differences among groups, such as ANOVA and *t*-tests.

As an example, suppose you have two nominal variables A and B, and a continuous variable Y. You may know that the mean of Y is different for different levels of A or of B, but the question remains whether there is any interaction effect, i.e., whether the relationship among the means for the different levels of A is affected by the level of B and vice versa. In a cell line plot with A on the axis and B in the legend, the lines will show you whether this interaction is present or not: if not, the lines will have the same pattern for each level of B; if so, the lines will show different patterns depending on the levels of B.

Cell plots can depict data as bar charts (often referred to as side-by-side bar charts), line charts, or point charts. You can choose which graphing variable appears on the horizontal axis and which appears in the legend. If you are examining means, you have the option of adding error bars. Edit Display lets you modify the structural appearance of cell line plots; see [“Customizing results,” p. 179 of *Using StatView*](#).

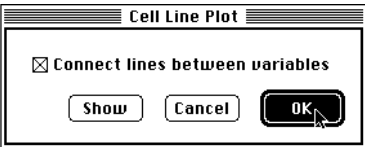
Dialog box settings

When you create a cell plot or edit it using the Edit Analysis button, you see the dialog box below. Cell plots have two simple statistics associated with them—sums and means. You choose which to graph for the variables you select. If you select means, you can also specify whether to display error bars. These choices are found in the Cell Plot dialog box:



If you want to show error bars, you must choose Means at the top of the dialog box; you cannot use error bars with Sums. If you show error bars, they can represent a specified number of standard errors, a specified number of standard deviations, or confidence intervals at a specified level. When you display error bars but they do not show in a particular cell, it is because there is only one observation in that cell.

There is an additional setting for cell line charts, found in a separate dialog box. For plots with more than one variable, you can eliminate the lines that connect points from different variables. Select only the plot (not the entire graph) by clicking on a point or line. Click the Edit Display button and the Cell Line Plot dialog box appears:



Uncheck the option to turn it off and click OK. To preview the change first, click Show.

Data requirements

Cell plots can be generated for one or more continuous variables. Nominal grouping variables are optional.

Variable browser buttons	
Add	To generate a cell plot, select one or more continuous variables and click Add. The groups of any nominal variable assigned using the Add button appear on the horizontal axis. Each additional continuous variable assigned is added to the same plot. Each additional nominal variable assigned creates new cells which are shown on the horizontal axis.
Split By	The cells of any nominal variable(s) assigned using the Split By button appear in the legend.

Results

For explanation of the plots, please see the preceding discussion. The default plot is a line chart.

Point Chart	Shows the means or sums of the cells or variables as points. Error bars can be displayed for means.
Line Chart	Shows the means or sums of the cells or variables as points connected by lines. Error bars can be displayed for means.
Bar Chart	Shows the means or sums of the cells or variables as bars. Error bars can be displayed for means.

Templates

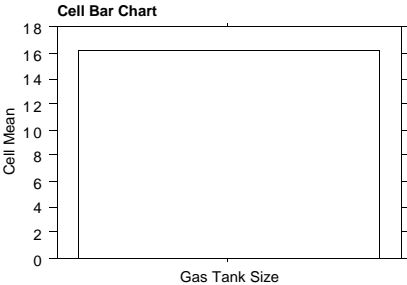
The following templates provide cell plots.

Graphs	Cell Bar Chart	Cell bar chart for continuous measurement variable, nominal variable for the horizontal axis, and optional Split By variable for the legend.
	Cell Line Chart	Cell line chart for continuous measurement variable, nominal variable for the horizontal axis, and optional Split By variable for the legend.
	Cell Point Chart	Cell point chart for continuous measurement variable, nominal variable for the horizontal axis, and optional Split By variable for the legend.

Exercises

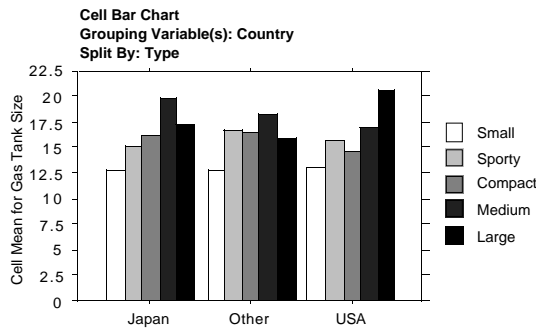
The dataset you will use in this exercise contains measurements of gas tank size for 116 cars of various types from different countries. You will compare the average size of gas tanks for each country of manufacture as well as see whether the type of car affects gas tank size.

- Open Car Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser under Cell Plots, select Bar chart and click Create Analysis
- Click OK to accept the default parameters
- In the variable browser, select Gas Tank Size and click Add



The bar represents the mean size of all 116 gas tanks. We want to compare size for both country of origin (Japan, Other and USA) and type of car (Small, Sporty, Compact, Medium and Large). The groups of any nominal variable assigned to the cell plot using the Add button appear on the horizontal axis. The groups of any nominal variable assigned to the cell plot using the Split By button appear in the legend, and appear side-by-side within the other groups in the bar chart. Whether to add a nominal variable or split by a nominal variable depends on which factor you wish to emphasize in the graph. In this exercise, we are primarily interested in how the type of car affects the size of the gas tank in a particular country.

- In the variable browser, select Country and click Add
- In the variable browser, select Type and click Split By



Now you can see for each country of manufacture how the type of car affects the gas tank size. You can also see how the pattern of the effect of type on gas tank size varies from country to country. If you were interested in examining gas tank size with the roles of country and type reversed, you would construct the cell bar chart differently.

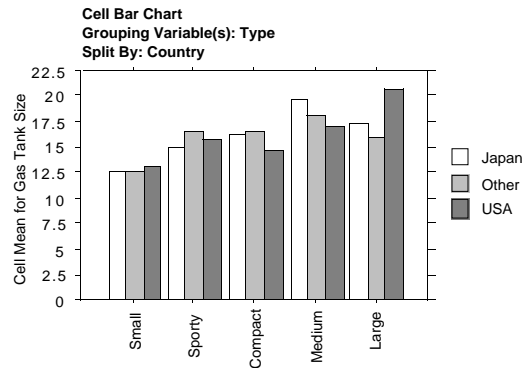
- In the variable browser, select Type and Country and click Remove

You will now assign these variables in a different order.

- Select Type and click Add

The horizontal axis has five different tick marks, one for each group, with a single bar representing the means of the gas tanks sizes for each type of car.

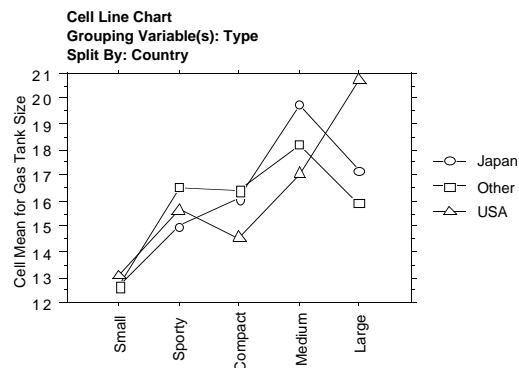
- Select Country and click Split By



Now you can see for each type of car how the country of manufacture affects gas tank size. You can also see how the pattern of the effect of country on gas tank size varies from type to type.

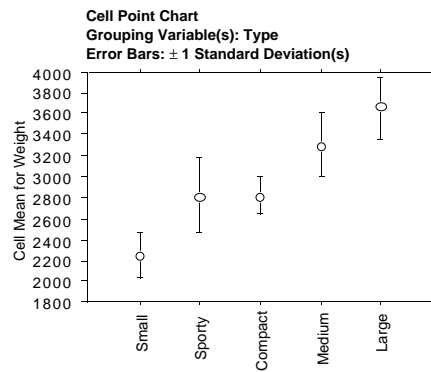
By choosing a line chart you can display the different groups as lines with different symbols as opposed to side-by-side bars.

- Make sure the bar chart is still selected
- In the analysis browser under Cell Plots, select Line Chart and click Create Analysis



Point charts are similar to line charts except they display the value of the mean as a single point as opposed to a line. Points are not connected with lines and points for a split cell are displayed side-by-side instead of stacked. They are most useful when you are displaying error bars as well.

- Click in the empty space of the view to deselect all results
- In the analysis browser, select Point Chart and click Create Analysis
- Check Show error bars, choose standard deviation, and click OK (Accept the default setting of 1 standard deviation.)
- In the variable browser, select Weight and Type and click Add



These examples have compared different groups, but you can also use cell plots to compare the means or sums of different variables. To do that you would use the Add button to assign the continuous variables to the cell plot. A bar or point appears for each assigned variable.

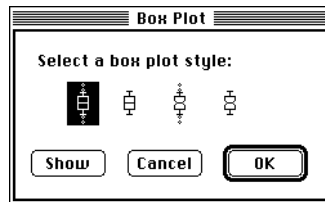
Box Plots

A **box plot** is a graph for displaying the 10th, 25th, 50th, 75th and 90th percentiles of a variable. You can use box plots to compare variable distributions, or to see the distribution of a single variable. Each box plot is composed of five horizontal lines that display the 10th, 25th, 50th, 75th and 90th percentiles of a variable. All values for the variable above the 90th percentile and below the 10th percentile are plotted separately, so box plots are especially useful for displaying outliers.

The box plot allows you a great deal of flexibility, comparing not only the distribution of an entire variable or variables but also comparing the distributions of groups defined by nominal variables. In addition, you can plot the outliers and display notched box plots that represent a 95% confidence interval around the median. Edit Display lets you modify the appearance of box plots; see [“Customizing results,” p. 179 of Using StatView](#).

Dialog box settings

Box plots have no analysis parameters, but you can choose whether to display notches representing a 95% confidence interval for the median. Select the interior of the plot and click Edit Display to display the Box Plot dialog box.



Data requirements

Box plots can be generated for one or more continuous variables. Nominal grouping variables are optional.

Variable browser buttons	
Add	To generate a box plot, select one or more continuous variables and click Add. The groups of any nominal variable assigned using the Add button appear on the horizontal axis. Each additional continuous variable assigned is added to the same plot. Each additional nominal variable assigned creates new cells which are shown on the horizontal axis.
Split By	The cells of any nominal variable(s) assigned using the Split By button appear in the legend.

Results

For explanation of the plots, please see the preceding discussion.

Box Plot	Shows the 10th, 25th, 50th (median), 75th and 90th percentiles of a variable. Values above the 90th and below the 10th percentile are plotted as points.
Notched Box Plot	Shows the same information as a Box Plot with the addition of a notch showing the 95% confidence interval around the median.

Templates

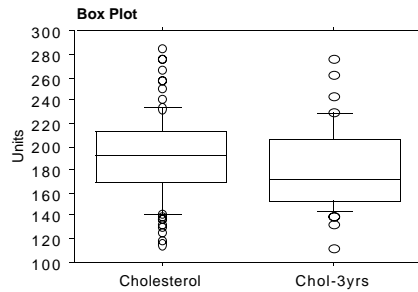
The following template provides box plots.

Graphs	Box Plot	Box plot for continuous measurement variable, nominal variable for the horizontal axis, and optional Split By variable for the legend.
--------	----------	--

Exercises

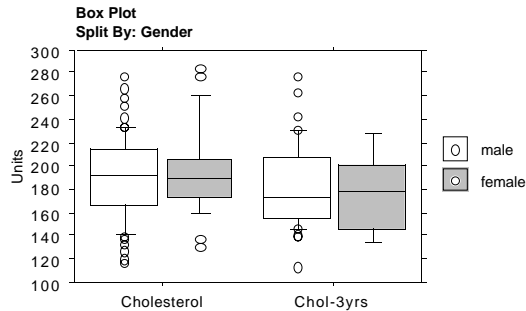
The data used in the following exercises comes from medical students. Blood lipid levels and other cardiovascular risk factors are evaluated in students as freshmen and later as seniors. In these exercises you examine the distribution of several of the lipid measurements. You will also see if there are any differences between the distributions for males and females.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Box Plot and click Create Analysis
- In the variable browser, select Cholesterol and Chol-3yrs and click Add



The box plot allows you to compare the distributions of these variables. Box plots work similarly to cell plots discussed above. You can group boxes along the horizontal axis as well as using the legend to distinguish groups. To examine whether the distributions compare for males and females:

- In the variables browser, select Gender and click Split By



The male and female groups appear next to each other so you can compare their distributions. You could just as easily add nominal variables which would break the groups out along the horizontal axis by using the Add rather than the Split By button.

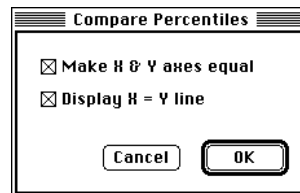
Compare Percentile Plots

A **compare percentiles plot** allows you to compare the distributions of two groups of one or more continuous variables. It graphs 19 corresponding percentiles of one group set against another group. The percentiles graphed are the 1, 2, 3, 4, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 96, 97, 98, and 99th percentiles.

If either group has less than fifty values, not all percentiles can be calculated, so the plot displays as many percentiles as can be computed. The plot is designed to compare two groups only, so the assigned nominal variable can contain only two groups.

Dialog box settings

When you create a compare percentiles plot or edit it using the Edit Analysis button, you see the dialog box below. The first setting makes the axis lengths equal. The second displays a diagonal line to makes it easier to see if the percentiles for the two groups are similar. If identical, they would lie exactly on this line. Both options are checked (turned on) by default:



Data requirements

Compare percentile plots are generated using one nominal variable with two groups only and one or more continuous variables.

Variable browser buttons	
Add	To generate a compare percentiles plot, select a nominal variable with two groups only and one or more continuous variables and click Add. Each additional variable assigned is added to the same plot.
Split By	The cells of any nominal variable(s) assigned using the Split By button appear in the legend.

Results

For explanation of the plot, please see the preceding discussion.

Compare Percentiles Plot	Shows nineteen percentiles of one group on the vertical axis against the corresponding percentiles of another group on the horizontal axis.
--------------------------	---

Templates

The following templates provide percentile results.

Descriptive Statistics	Percentiles	Percentiles summary table and plot for continuous variable.
Graphs	Compare Percentiles	Compare Percentiles plot for continuous variable and two-level nominal variable.

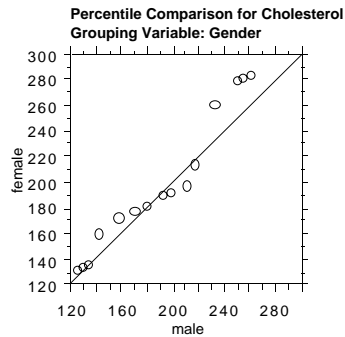
Exercise

The data used in the following exercise comes from medical students. Blood lipid levels and other cardiovascular risk factors are evaluated in students as freshmen and later as seniors. In the following exercises you will compare the distribution of cholesterol values for male and female freshmen.

- Open Lipid Data from the Sample Data folder
- From the Analyze menu, select New View
- In the analysis browser, select Compare Percentiles and click Create Analysis

You have two options which help you analyze the information displayed in the graph. You can make axes the same size in order to produce a square graph. You can also display a reference line which fits the line $X = Y$. If the distributions of both variables is equal, all points fall on this $X = Y$ line.

- Leave both options selected and click OK
- In the variable browser, select Gender and Cholesterol and click Add
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent variables



The 1st, 2nd, 98th, and 99th percentiles are missing in this chart; recall that not all percentiles can be calculated when a group has fewer than fifty values, as is the case with females. The point in the lower left hand corner of the graph is the 3rd percentile of females plotted against the third percentile of males. The point at the upper right hand corner is the 97th percentile of females plotted against the 97th percentile of males. At the 50th percentile the cholesterol values are almost exactly equal—the value lies almost directly on the $X = Y$ line. Below the 50th percentile, the female cholesterol count is higher than the male at every percentile. However, between the 50th and 80th percentiles the male cholesterol count is higher than the female count. At the upper extreme, the 90th to the 97th percentiles, the females once again exceed the males.

QC Subgroup Measurements

This chapter, the first of five regarding StatView's quality control tools, introduces quality control and statistical process control (SPC) in general and then goes on to discuss QC Subgroup Measurements methods in particular. Subsequent chapters discuss StatView's other QC methods: [“QC Individual Measurements,” p. 277](#), [“QC P/NP,” p. 287](#), [“QC C/U,” p. 299](#), and [“Pareto Analysis,” p. 309](#).

Introduction to SPC

Industries employ **processes** to manufacture products. Laboratory technicians use a **process** to sample blood. Educators employ processes to educate students. Baseball players employ a process when they swing at a pitched ball. In all of these cases, a process may be regarded as any series of actions that produces a measurable result. In each case, we can use data about the **results** of a process (be these results widgets, student test scores or whether a swatted ball is a hit or an out) to infer important characteristics about the process itself.

What is statistical process control?

Statistical process control (SPC) concerns itself with particular *statistical* characteristics of processes. Whether SPC is used to analyze measurements or attributes of items, the goal of most SPC analyses is to evaluate whether a process matches the statistical definition of being in control. Understanding the statistical concept of control is the key to understanding much of what quality control statistics are all about.

When is a process in control?

What does it mean for a process to be in control? In the context of SPC statistics, an **in control** process is one that produces items that vary within the limits proscribed by a particular statistical distribution. Though the distribution that is used depends on the particular process control statistic (see the “Discussion” sections in each chapter), it is the distributions that provide the basis for computation of **control limits**. If the data conform to the assumptions embodied in these distributions, then an in control process, according to statistical theory, will only very rarely violate (exceed) the computed control limits. It stands to reason, then, that the *most likely* cause of a violation of these limits is that the data do not match the assumptions of the

statistics. Violations usually indicate that items produced do not come from the distribution that the particular SPC statistic assumes.

There are several reasons why data may not conform to the distribution assumed by a particular SPC statistic. Among the most common reasons are: 1) the process *inherently* is not well-modeled by the distribution assumed in the chosen QC statistic, and 2) the process is well-modeled by the distribution assumed in the chosen QC statistic, but factors presently in the process cause the items produced to deviate from their expected distribution. This distinction is rather subtle but very important. A violation due to reason 1 suggests that the process is behaving properly, but the analyst is not using the correct distribution to model it. Violations due to reason 2 imply that the analyst is using the appropriate model of the process, but the process itself is not behaving properly (i.e., the process is **out of control**).

Clearly, the analyst must try to distinguish between these 2 possible causes of violations. To determine if the violation is due to reason 1, the QC analyst should re-examine the data and the process to make sure that she is using the correct SPC statistic and to be certain that the process can be modeled by *any* SPC statistic. This generally means that the analyst must evaluate whether the data deviate significantly from a particular distribution. After confirming that the violation is not due to reason 1, the analyst usually concludes that the cause of the violation is reason 2. Violations due to reason 2 are due to **assignable causes**. The task of bringing the process under control then becomes that of isolating and then eliminating any assignable causes.

Suppose, for instance, that a technician in a chemistry lab is pipetting a culture medium into petri dishes (this is a process). Measurements of the amounts of medium in each petri dish are the measurements of items produced. At some point the technician replaces the tip of the pipette. Because not all pipet tips are identical, the new pipet tip will deliver either more or less medium than did the original tip. The process has therefore changed. If the process has changed substantially (i.e., the new pipet tip is significantly larger or smaller than the original), then the measurements will very likely violate the control limits. In this case, changing to a new pipet tip is the assignable cause that causes measurements to violate the control limits. After changing to a new pipet tip, the technician is sampling from a different distribution, one with a different mean and probably a different variance than that sampled from the original pipet tip. The result is an out of control process.

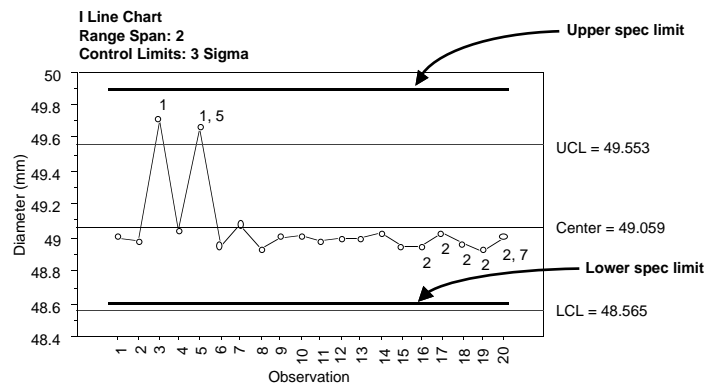
Process control vs. process capability

Up to this point, we have only discussed process control. The concept of control is always defined in statistical, rather than absolute terms. As mentioned above, if a process is in control, that means only that the process produces items that vary within the limits proscribed by a particular statistical distribution. Control is not equivalent to **repeatability** or **precision**. For instance, measurements from an in control process can vary quite substantially as long as they do not depart from the expectations based on a particular statistical distribution (i.e., the variance for an in control process can be arbitrarily large, and thus not very precise).

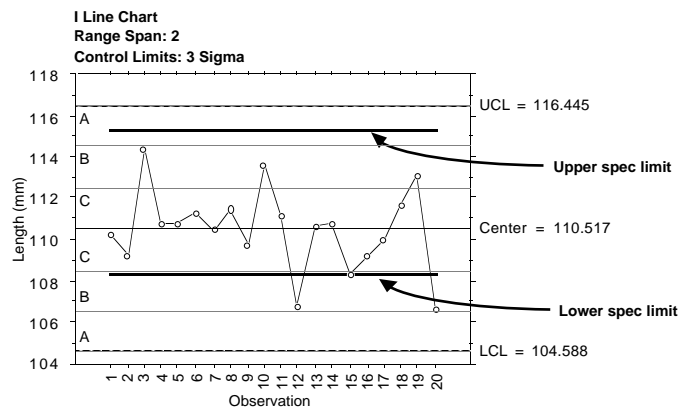
By contrast, the related concept of **process capability** does place absolute bounds on the range of acceptable variation in a process. To say that a process is **capable** means that it is both in control and that a high percentage of the items it produces are within certain specification

limits. Specification limits are the upper and lower acceptable values for measurements of the items produced by a process: items with measurements above or below these limits are rejected. What constitutes criteria of acceptability could be determined, for instance, by engineers who require that a part match certain specifications to function properly, or by the demands of the marketplace, which require that products meet certain standards of, say, durability or performance.

In fact, it is quite possible that a process that produces items well within specification limits may be out of control and therefore not capable. Below is an example of an individual measurement chart for a process that is out of control, even though the items it produces are within the specification limits (the latter indicated by the heavy black lines). Along with showing two observations beyond the upper control limit (points labeled 1), these data show other violations of control as well (indicated by all points labeled with numbers; see [“Tests for special causes and custom tests,” p. 289](#)).



Conversely, a process that is in control may be producing items that are not within specification limits, as illustrated in the individual measurement chart below. In this case, there are two measurements (observations 12 and 20) that are beyond the lower specification limit, yet these are still well within the control limits.



These examples highlight the relationship between the concepts of process control and process capability. These characteristics of processes must be examined separately and in sequence.

First, you must establish that the process is in control; then you can determine whether or not the process is capable. Obviously, statistical process control involves much more than simply producing items within specification limits. Processes also must be statistically well-behaved.

Why examine process control and process capability?

Why is it necessary to make the distinction between process control and process capability? The most important reason is also the most pragmatic: the steps taken to remedy an out of control process generally differ from those used to remedy an in control process with poor capability (i.e., the process yields unacceptably few items within specification limits).

For example, when the QC analyst recognizes that a process is out of control, she must attempt to trace the cause to a specific source of systematic variability (i.e., an assignable, or special cause) and to eliminate this cause. (The explanations given for each of the eight tests for special causes are of particular help when initiating such an investigation. See [“Tests for special causes and custom tests,” p. 289.](#)) Examples of systematic causes of variation are: trends in operator fatigue, systematic occurrence of impurities in manufacturing materials, drift in the adjustment of manufacturing devices and production of items by different operators.

When, however, the QC analyst finds that an in control process is not very capable, she uses a different strategy to correct the problem. Low capability generally has one or both of two causes: 1) the process is not centered on the target value (the optimal value set by specifications), or 2) there is too much **random variability** in the process. The first cause is generally the easiest to fix: simply adjust the process (often the machinery used) to produce items closer to the specification target. The second cause is generally more difficult to trace and often more costly to fix. There are many potential causes of random variability within manufacturing processes, but it is precisely because the variation appears random that such causes are so hard to identify (keep in mind that known causes of variation are, by definition, non-random). One common cause of random variability is worn production machinery, which, in some cases, can be remedied only by replacement or costly reconditioning.

Now, we consider the application of StatView's SPC statistics to a real life problem.

An example

The Acme Fastener Company has begun to manufacture bolts. The important properties of a bolt are its length, its diameter, whether it has any nicks or scratches on the threads, and whether it has any discolorations.

The first thing Acme wants to do is establish that their production process is in control. In the language of SPC, they want to be sure that what is governing the inevitable variations in length and diameter of their bolts is a **constant-cause system**, not assignable causes. Putting it another way, they want to be sure that the variation in length and diameter of their bolts is random and centered around a mean—that it follows a normal distribution—rather than being due to non-random causes such as variation among machines or production by different operators. At this point, they are not primarily concerned with whether the lengths and diam-

eters of the bolts are within the desired specification limits. They are mainly concerned with describing the pattern of variation in length and diameter.

They begin by developing control charts. They produce one series of control charts for length, and one for diameter. We will concentrate on the diameter charts, but what we say about them will apply equally well to the length charts.

As a result of reading textbooks on SPC, Acme decides that they should measure and record the diameters of four bolts every half hour. In other words each subgroup will consist of measurements of diameters from four bolts.

They will plot \bar{X} bar (sample mean), S (sample standard deviation) and $CUSUM$ (cumulative sum) charts. That is, each point on the \bar{X} bar chart will be the mean of the four diameter measurements for a subgroup, and each point on the S chart will be the standard deviation of the four diameter measurements for a subgroup. The $CUSUM$ chart plots cumulative sums and is used to complement the information in the \bar{X} bar chart. To generate their charts, the QC analysts at Acme simply choose among the QC Subgroup Measurements items in the StatView analysis browser.

They measure diameters of bolts from 30 subgroups, and they use these data to establish an upper control limit (UCL), a lower control limit (LCL), and center line for the \bar{X} bar and S charts. They find that they have to make a few adjustments in their lathes, and that they have to correct the techniques of several operators, but soon their process is in control. Although the adjustments took some time, they were able to easily recreate the same analyses; with StatView's template feature, all they need to do is save the view and then they can rerun the same analyses whenever they get new data.

Now the Acme management wants to know if the bolt diameters adequately meet specifications. Long before production ever began, Acme held meetings with design and production engineers, marketing and customer support personnel to establish specification limits for their bolt diameter measurements. These limits are intended to enforce production of bolts that meet market requirements, without resulting in excessive rejection of bolts and a process that is too costly to implement. Using these specification limits, Acme will now determine if the bolt production process is capable, i.e., does it produce bolts that meet specifications?

Using the capability analysis supplied within QC Subgroup Measurements, Acme finds that all of their capability indices are well above 1.33. This means that only a very tiny percentage of bolts have diameters that do not fall within the specification limits. Acme is quite satisfied with this result.

They quickly realize, however, that they will be spending a lot of time on their SPC analyses if they continue to analyze every dimension and property of their bolts. There is an alternative. A less labor-intensive measure of process control is p/np analysis. This analysis allows Acme simply to count the number of unacceptable bolts, and use these counts to evaluate whether or not the process is in control.

The big advantage of p/np analyses is that you can use any rejection data, regardless of the causes of the rejections. So, using p/np analyses, Acme can analyze the entire process by essentially pooling data from a number of observations and measurements. Suppose, for instance, Acme uses the following criteria for rejection of any bolt:

1. if more than one scratch on a thread, reject bolt;
2. if more than one nick on a thread, reject bolt;
3. if more than two discolorations, reject bolt;
4. if bolt diameter greater than D_u millimeters or less than D_l millimeters reject bolt;
5. if bolt length greater than L_u millimeters or less than L_l , reject bolt.

Using a p chart from StatView, Acme plots the fraction of nonconforming items (i.e., rejected bolts) per subgroup (each subgroup comprises a subsample of bolts produced in an hour). Since the values from the various subgroups are expected to follow a particular statistical distribution, the p chart can plot control limits based on this distribution (or some approximation of it). As it turns out, none of the subgroup proportions is beyond the control limits, so Acme concludes that the process is in control.

To make it easier to move bolts to and from the inspection station, Acme now wants to put the bolts in boxes, each containing 100 bolts. Since it is much easier to record the number of defective (nonconforming) bolts per box, rather than keep a running count of all of the defective bolts, Acme shifts to using c/u analyses. The c/u statistics analyze the numbers of nonconformities per inspection unit (in this case, the number of defective bolts per box). Acme chooses to use u charts, which plot the average number of nonconformities per inspection unit for each subgroup.

Having standardized on u charts for all of their preliminary SPC analyses, Acme creates a u chart template that uses a preset value of u . This preset value is based on Acme's production history; using this **historical value** allows Acme management to see immediately if the number of defects per inspection unit has significantly improved or declined relative to the historical average.

Acme also uses Pareto charts in association with their u charts. Though the u charts do a good job of tracking the number of nonconformities *of a particular type*, it is also essential that Acme knows the relative frequencies of the various types of nonconformities; this information is concisely summarized in Pareto charts. If Acme sees that almost all of their defective items are attributable to just one or two types of defects, then they probably shouldn't spend much time trying to improve the incidence of other types of defects. This is just one of the ways that Pareto charts can help QC analysts and process engineers decide which problems are most worthy of their attention.

As you can see, a complete quality improvement program can involve a variety of the SPC analyses available in StatView. Effective use of all of these analyses is the key to improving quality. Each is appropriate under specific circumstances. In overview, the measurement analyses (individual and subgroup analyses) provide very specific information about the results of a process. These analyses focus on one measurement at a time (e.g., bolt diameter) and are of great utility when trying to track down specific assignable causes. Attribute analyses (e.g., counts of defective bolts) provide less specific information about a process (e.g., bolts can be defective for any number of reasons), but can be used effectively as a less labor-intensive means to monitor a process once it is under control. Finally, Pareto charts can be used effectively to identify the most problematic sources of defects in a process. Far from mutually exclusive, these analyses should be regarded as complementary when trying to establish a complete approach to quality improvement.

Subgroup measurements

In statistical process control, measurements of items resulting from a process and sampled in natural **subgroups** are analyzed with subgroup measurement statistics. What constitutes a natural subgroup depends on the context in which the data are collected. In some cases, subgroups could be blocks of time (e.g., days) or they could represent partitions of items from different sources (e.g., different machine operators). For background on the application of subgroup analyses, when to use them and advice on sampling procedures, see the [“Discussion,” p. 257](#), and any standard text, e.g., Grant and Leavenworth (1988).

Another requirement of analyses in this and the next chapter ([“QC Individual Measurements,” p. 277](#)) is that the measurements analyzed must be expressed on a continuous scale. Examples of such continuous measurements are length, weight, velocity or brightness. In general terms, a continuous measurement is any quantity that can (in theory) take on any value within a particular interval. Another way to think of continuous measurements is that they are not discrete. Values such as Blue, Red, and Yellow, the colors of balloons, are discrete values; values such as 2.3, 2.4, 1.9, the lengths in centimeters of rivets, are continuous.

Both subgroup and individual measurement analyses are based on similar statistical assumption. They differ in the way in which parameters (such as σ) can be estimated for the two types of data organization.

Discussion

When performing subgroup or individual measurement analyses, usually the primary concern of the QC analyst is to evaluate process control and capability (see the preceding [“Introduction to SPC,” p. 251](#), for an explanation of these terms). Control charts are the primary tools of the QC analyst in evaluating process control; capability indices are the most commonly used metrics for the evaluation of process capability.

What are the statistical assumptions that allow useful application of control charts and capability indices? The central assumption is that the quantities plotted are from a normal (i.e., Gaussian) distribution. Some implications of this assumption are discussed in the sections that follow.

Xbar (subgroup mean) charts

In analysis of subgroup measurements, the QC analyst might begin by plotting an Xbar along with an R or an S chart (see below). Conventionally, these charts are considered together because they provide complementary information about process variation.

The Xbar chart provides information about variation *among subgroup means*. Specifically, Xbar charts plot the means of the measurements from each of a series of subgroups. Should at least one of the means be very different from the others, it may be that the process is out of control. To help the analyst evaluate whether a particular subgroup mean is especially high or low, the

Xbar chart shows lines indicating the control limits for the process mean. These control limits define the range of values within which the means should fall for a process that is in control.

How are these control limits determined? This is largely a matter of judgment and experience. The analyst sets the parameters for calculation of the control limits so that subgroup means beyond these limits are relatively unlikely if the process is in control. By convention, these limits are based on the **3-sigma rule**; in statistical terms, this rule means that the upper and lower control limits are placed 3 estimated standard deviations above and below the expected value of the mean (or center line) for each subgroup. Formulas for these computations are in [“Algorithms,” p. 433](#).

As suggested above, Xbar charts require that the means from the subgroups are normally distributed. This requirement is not as restrictive as it might seem. In fact, there is a statistical maxim known as the **Central Limit Theorem**, which predicts that the means from subgroups should be normally distributed even when the measurements in these subgroups are not. (This is true as long as all subgroups are drawn from the same population.) For this reason, subgroup Xbar charts often can be used when individual measurement charts (I charts) cannot.

Considerations when setting *k*-sigma

The statistical interpretation of 3-sigma limits is that there is a probability of 0.0027 (i.e., about 3 times out of 1,000) that any *individual* subgroup mean will exceed these limits. It is important to recognize that this is not equivalent to the probability that *any* subgroup mean in a Xbar chart will exceed the control limits. A conservative estimate of that probability when using 3-sigma control limits is 0.0027 times the number of subgroups. For a chart with 20 subgroups plotted with 3-sigma limits, the probability that at least one subgroup mean will exceed the control limits (assuming that the means are normally distributed) is $0.0027 \times 20 = 0.054$.

This highlights a potential problem with the unexamined use of 3-sigma control limits within *all* control charts, especially when plotting results from many subgroups. The problem is that the probability of a false out of control signal increases as you add more subgroups. (A false out of control signal is one that is not due to any assignable cause, but just happens by chance.) Suppose, for instance, you always chart 50 subgroups on your Xbar charts, and you want the probability of a false out of control signal to be no more than 0.05. This means that the probability of exceeding the control limits would be $0.05/50 = 0.001$ for each *individual* subgroup mean. If you do all of the necessary calculations, you will find that you should use control limits based on 3.29 times sigma to achieve this probability. Though this doesn't seem like much of a difference, using 3.0 as the sigma multiplier instead of 3.29 actually increases the probability of false out of control signals nearly 2.7 fold! This demonstrates one important reason why the user has control over the sigma multiplier for all SPC analyses in StatView.

R (subgroup range) charts

The Xbar chart tells only part of the story about process control. Many QC analysts create R (range) charts along with their Xbar charts, because the R chart provides information about the magnitude of variation among measurements *within subgroups*, information that is also essential for evaluating process control.

Why do QC analysts need such a measure of variation within subgroups? The reason is that if variation within subgroups differs substantially among subgroups, then it is unlikely that the causes of these differences are random, i.e., the process is probably out of control.

As suggested by its name, an R chart plots the range of the measurements from each subgroup for a series of subgroups. (Range is defined as the absolute value of the difference between the high and low measurements in each subgroup.) As with Xbar charts, R charts also plot expected values (center lines) and control limits for the ranges from each subgroup. “[Algorithms](#),” p. 433, gives the formulas for these computations. The cautions regarding the unexamined application of 3-sigma control limits in Xbar charts pertain to R charts as well.

Since the range is based only on two values, it is a fairly rough estimate of the variation among measurements within a subgroup. The popularity of this chart is probably due to the relative ease with which subgroup ranges can be computed by hand.

S (subgroup standard deviation) charts

Many analysts now prefer the S (standard deviation) chart over the R chart, because subgroup standard deviations usually provide a more accurate estimate of variation within subgroups. Should you wish to create an S chart whenever you create an Xbar chart, you may find it easiest to create a template that combines these two results.

S charts plot the standard deviation of the measurements within each subgroup for each of a series of subgroups. As with Xbar and R charts, S charts also plot the expected value (center line) and the control limits for the standard deviation from each subgroup. Since these expected values and control limits are based on theoretical distributions of standard deviations from normal populations, you should examine your data carefully to evaluate the assumption of normality.

Tests for special causes

Tests for special causes are intended to detect particular sorts of non-random patterns in Xbar, I, p/np and c/u results, any one of which might indicate that the process is out of control. Due to conventions in how the tests are calculated, both the standard and custom tests are only available when subgroups are of equal size.

The standard suite of tests in StatView is a refinement by Nelson (1984, 1985) of the original Western Electric rules (Western Electric, 1956). With the exception of rule 1, none of these tests should be assumed to have well-determined probabilities for **false signals**. It should be kept in mind, however, that the greater the number of tests used simultaneously, the greater the probability of a false signal for any given chart.

Accordingly, Nelson (1985) recommends that combinations of the standard tests should be applied judiciously. In particular, tests 1–4 compose a good suite for detection of many common assignable causes, while tests 5–8 generally should be left for more advanced diagnoses of specific problems.

As mentioned previously, all 8 of the standard tests are also applicable to I charts (see the next chapter, [“QC Individual Measurements,” p. 277](#)). Tests 1–4 are usually applicable to p/np and c/u charts as well; for more information, see the chapters, [“QC P/NP,” p. 287](#), and [“QC C/U,” p. 299](#).

Standard tests for special causes

Below are the descriptions and interpretations for each of the eight standard tests for special causes. Note that these tests refer to zones A, B and C in control charts. These zones are defined as bands of constant width where zone A is between 2 and 3 sigmas above and below the center line, zone B is between 1 and 2 sigmas above and below the center line, and zone C is between 0 and 1 sigma above and below the center line.

1. **1 point beyond zone A** detects a shift in the process mean, μ , an increase in the estimated standard deviation, σ , or a single aberration.
2. **9 consecutive points above or below center line** detects a shift in the process mean.
3. **6 consecutive increasing or decreasing points** detects a trend or drift in the process mean.
4. **14 consecutive alternating points** detects systematic alternating effects, such as alternating use of different machines, operators or materials.
5. **2 of 3 consecutive points in zone A or beyond** detects a shift in the process mean, or an increase in the standard deviation. The 2 points must be in the same A band (i.e., above *or* below the center line).
6. **4 of 5 consecutive points in zone B or beyond** detects a shift in the process mean. The 4 points must be in the same B band (i.e., above *or* below the center line).
7. **15 consecutive points in zone C** detects stratification of subgroups when the observations in a single subgroup come from various sources with different means. The points must be on both sides of the mean.
8. **8 consecutive points outside zones C** detects stratification of subgroups when the observations in one subgroup come from a single source, but subgroups come from different sources with different means. The points must be on both sides of the mean.

Custom tests for special causes

Some users may find that their work requires modifications of the parameters that are used in defining the standard suite of special causes tests. For instance, you may prefer rule 5 to be defined as “3 of 4 consecutive points beyond 2 sigma.” StatView allows you to customize the parameters used to define these tests.

All eight of the custom tests for special causes have the same logical structure as the standard tests. Their difference from the standard tests is that the custom tests let you define the num-

ber of points used in the calculation of a violation and define critical values with arbitrary multiples of sigma rather than with zones about the center line.

If you commonly reuse the same suite of custom tests, you may find it easiest to create an analysis with your custom suite of tests and then save this analysis as a template. You can then use this template for all subsequent analyses.

CUSUM (cumulative sum) charts

Another important, though less frequently used tool for evaluating process control is the CUSUM (CUMulative SUM) chart, sometimes referred to as a CSCC (Cumulative Sum Control Chart). When used with the FIR (Fast Initial Response) option, some experts prefer it to Xbar charts for detecting particular types of out of control processes (Ryan, 1989).

Each CUSUM chart plots two cumulative sums. These are the high (S_{Hi}) and low (S_{Li}) sums of the standardized deviates of subgroup means from the process mean. An increase in the High sum indicates an increase in the process mean; an increase in the Low sum indicates a decrease in the process mean. Should either of these sums exceed the CUSUM control limit (h), the process is out of control.

Enabling the FIR (Fast Initial Response) option generally increases the sensitivity of CUSUM to shifts in the process mean, without unduly increasing the probability of false out of control signals. In fact, for detecting shifts in the process mean, FIR CUSUM generally outperforms Xbar charts (Ryan, 1989). It should be noted, however, that CUSUM analyses generally are not sensitive to other sorts of assignable causes, such as those that increase variation among or within subgroups. Therefore, you should use CUSUM analyses in conjunction with Xbar and R or S charts whenever you cannot exclude other sources of variation.

Capability indices

Once you have established that a process is in control, you can generate a table of capability indices. These indices measure how well a stable process meets specifications. Maximization of process capability is an ongoing effort and is often regarded as the ultimate goal of any quality improvement program.

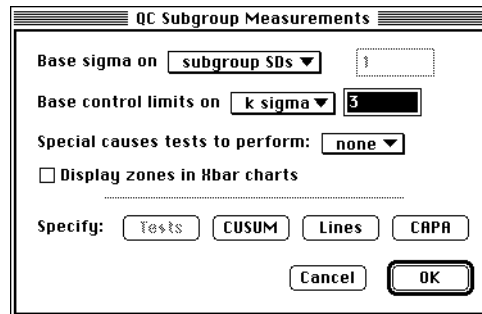
StatView offers a variety of capability indices, each appropriate and useful in particular circumstances. In general, these indices tell you different things about the distribution of your measurements relative to specifications. If you require a single capability index, C_{pm} , is favored when a target value is specified, while C_{pk} generally is preferred when it is not. Formulas for these indices are in [“Algorithms,” p. 433](#).

While capability standards vary widely among applications and processes, conventionally, a C_{pk} index of 1.33 is regarded as the minimum acceptable value in manufacturing. Assuming that the measurements are normally distributed (an assumption that is central to the proper application of capability indices), a C_{pk} index of 1.33 implies that, on average, only 6 items out of 100,000 are beyond specification limits.

Unacceptable capability values can be due to two causes. Either the measurements are not **centered** relative to the specifications, and/or there is excessive variability among measurements. If the centering index, k , is close to 0, the measurements are well centered relative to specifications. In such a case, an unacceptable capability index is probably due to excessive variation, which is reflected in a high value of σ relative to the range of the specification limits.

Dialog box settings

QC Subgroup Measurements dialog box



The settings in this dialog box apply to all subgroup measurement analyses. Some analyses have additional parameters that are set in dialog boxes accessed by clicking Specify buttons.

Base sigma on This pop-up menu allows you to set a calculation method or a value for sigma, the estimate of the process standard deviation. The default calculation method of σ is based on subgroup standard deviations, i.e., it is the square root of a weighted average of the subgroup variances. Alternately, σ may be computed from subgroup ranges, in accordance with the formula shown in [“QC Subgroup Measurements,” p. 473](#), by choosing subgroup ranges. A third alternative is that you can specify a value for σ by choosing Specify and entering a value in the text field.

Base control limits on This pop-up menu allows you to specify the values for k or alpha that are used to compute UCL and LCL. By default, control limits are computed with k -sigma (the default value of k is 3). Alternately, you can base the control limits on alpha. The default value of alpha is 0.002; it is the Type I error probability of exceeding the control limits if the process is in control. (If you use other values of alpha, please see the note on alpha-based calculation of the range, [p. 475](#).) It is important to note that k and alpha are mutually exclusive: only one can be used for any analysis. If you want to set the control limits directly as constants or as variables taken from a dataset, click on the Lines button (see [“QC Line Parameters dialog box,” p. 266](#)).

Special causes tests to perform This pop-up menu allows you to perform either the standard tests for special causes or the custom tests. When None is chosen (the default) the Tests button is dimmed and no tests will be performed. If you choose Standard or Custom, the Tests button is activated. See [“Tests,” p. 263](#), [“Tests for Special Causes dialog box,” p. 263](#), and [“Custom Tests dialog box,” p. 264](#), for more information.

Display zones in Xbar charts When enabled, this checkbox causes display of zones A, B, and C in Xbar control charts. It is important to note that zones can be displayed only when subgroup sizes are equal. By default, this option is disabled.

Tests If Standard is chosen from the Special causes tests to perform pop-up menu, this button opens the Tests for Special Causes dialog box. If Custom is chosen from Special causes tests to perform, this button opens the Custom Tests dialog box. These dialog boxes are described under [“Tests for Special Causes dialog box,” p. 263](#), and [“Custom Tests dialog box,” p. 264](#).

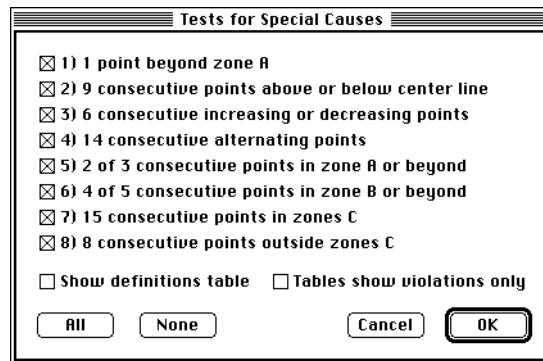
CUSUM This button opens the CUSUM Parameters dialog box; see [“CUSUM Parameters dialog box,” p. 264](#). Note that CUSUM results are displayed only if you create a CUSUM result from the analysis browser.

Lines This button opens the QC Line Parameters dialog box, which allows you to set values for the center line, UCL and LCL for all control charts. See [“QC Line Parameters dialog box,” p. 266](#).

CAPA This button opens the CAPA Parameters dialog box; see [“CAPA Parameters dialog box,” p. 267](#).

Tests for Special Causes dialog box

With Standard chosen from Special causes tests to perform, this dialog box appears when you click the Tests button in the QC Subgroup Measurements or QC Individual Measurements dialog boxes. A modified version of this dialog box showing only tests 1–4 appears when you click the Tests button in the QC P/NP or QC C/U dialog boxes.



Among subgroup measurement analyses, the settings in this dialog box apply only to Xbar charts. The same tests are available for I charts in individual measurement analyses. Tests 1–4 are available for *p*, *np*, *c* and *u* charts. See the following chapters, [“QC Individual Measurements,” p. 277](#), [“QC P/NP,” p. 287](#), and [“QC C/U,” p. 299](#).

Checking the box before each test activates that test. By default, all tests are checked.

Show definitions table When this checkbox is enabled, the active tests and their definitions are listed in a results table in the view. By default, this setting is disabled.

Tables show violations only When this checkbox is enabled, results tables show results only for those subgroups that violate the special causes tests. By default, this setting is disabled.

None Clicking this button disables all eight of the tests.

All Clicking this button enables all eight of the tests.

Custom Tests dialog box

With Custom chosen from Special causes tests to perform, this dialog box appears when you click the Tests button in the QC Subgroup Measurements or QC Individual Measurements dialog boxes. A modified version of this dialog box showing only tests 1–4 appears when you click the Tests button in the QC P/NP or QC C/U dialog boxes.

Custom Tests

☒ 1) 1 point beyond 3 sigma

☒ 2) 9 consecutive points above or below center line

☒ 3) 6 consecutive increasing or decreasing points

☒ 4) 14 consecutive alternating points

☒ 5) 2 of 3 consecutive points beyond 2 sigma

☒ 6) 4 of 5 consecutive points beyond 1 sigma

☒ 7) 15 consecutive points within ± 1 sigma

☒ 8) 8 consecutive points outside ± 1 sigma

☐ Show definitions table ☐ Tables show violations only

All None Cancel OK

As with the standard tests for special causes, the settings for the custom tests apply only to Xbar charts among the subgroup measurement analyses. The same tests are also available for I charts in individual measurement analyses. Tests 1–4 are available for \bar{p} , $n\bar{p}$, \bar{c} and \bar{u} charts. See the following chapters, [“QC Individual Measurements,” p. 277](#), [“QC P/NP,” p. 287](#), and [“QC C/U,” p. 299](#).

The text boxes in this dialog box allow you to specify both the number of points and the critical number of sigmas used to define each test. The default values are those for the standard tests for special causes. All text fields for numbers of points must be given positive integer values; those for multiples of sigma can take any positive real values.

All other settings in this dialog box work just like those in the Tests for Special Causes dialog box, above.

CUSUM Parameters dialog box

The CUSUM Parameters dialog box appears when you click the CUSUM button in the QC Subgroup Measurements or QC Individual Measurements dialog boxes.

The options you choose in this dialog box establish the parameters for a CUSUM analysis. However, no results are displayed unless you choose a CUSUM result from the analysis browser. If you want to do a CUSUM analysis on individual measurements, choose a CUSUM result under QC Individual Measurements in the analysis browser.

Mean shift This text field allows you to enter the mean shift, in standard units, that you wish to detect with the CUSUM procedure. For instance, enter 1.5 if you want to detect a mean shift of 1.5 standard deviations. The default value is 1.

Control limit This text field allows you to specify h , the CUSUM out of control threshold. The default value of h without FIR enabled is 4; with FIR, the default is 5.

Charts show violations If you enable this checkbox, charts will display an H symbol next to points that violate the upper control limit and an L symbol next to those that violate the lower control limit. By default, this option is enabled.

Tables show violations only If you enable this checkbox, the CUSUM results table will display information only for those observations/subgroups which are beyond the upper or lower control limits. If not checked, these tables will display information for all subgroups. By default, this option is disabled.

Invert lower sum If you enable this checkbox, then S_{Li} is displayed as the negative of S_{Li} as described in [“Algorithms,” p. 433](#). By default, this option is enabled.

Specify process mean This checkbox and associated text field allow you to specify a value for the process mean used in the CUSUM calculations. If not checked (the default), the calculated value of the process mean is used.

Use FIR This checkbox allows you to enable FIR. With FIR enabled, values of S_{Hi} and S_{Li} are set to $h/2$ both initially and following a violation (i.e., when either value exceeds h). When not checked (the default), values of S_{Hi} and S_{Li} are set to 0 initially, and are not reset following a violation. This option cannot be used with the On violation radio buttons.

On violation These radio buttons allow you to choose what CUSUM does after a violation occurs. The Do Nothing option (the default) leaves the values of S_{Hi} and S_{Li} unchanged. The Reset option changes these to the initial value. Note that enabling FIR is equivalent to choosing Reset and an initial value of $h/2$.

Initial value This text field allows you to specify the initial value of S_{Hi} and S_{Li} if either On violation: radio button is enabled. If On violation: Reset is enabled, the value in this text field is also the reset value.

QC Line Parameters dialog box

The QC Line Parameters dialog box appears when you click the Lines button in the QC Subgroup Measurements, QC Individual Measurements, QC P/NP or QC C/U dialog boxes.

This dialog box permits you to specify constants or variables for center lines, UCLs and LCLs on all control charts.

In the following options, if you select Variable, you cannot then edit the corresponding text box, because variables can be specified only in the Variables dialog box. For this reason, names of variables in these text boxes are dimmed.

Chart Use this pop-up menu to select the particular control chart to which you want to apply the options in this dialog box. It is important to note that the line parameters for one chart do not carry over to the other charts when you change the Chart pop-up. If this dialog box is accessed through the QC Subgroup Measurements dialog box, the pop-up shows Xbar Chart, R Chart and S Chart. If accessed through the QC Individual Measurements dialog box, the pop-up shows I Chart and MR Chart. If accessed through the QC P/NP dialog box, the pop-up shows P Chart and NP Chart. If accessed through the QC C/U dialog box, the pop-up shows C Chart and U Chart.

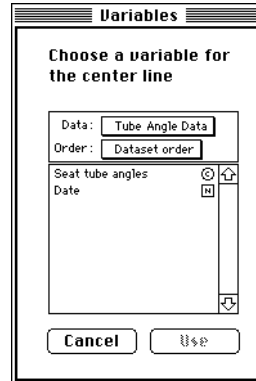
Center line This pop-up menu and associated text field allow you to calculate or to assign a constant or a variable for the center line or the currently selected chart. A constant is any numerical value you enter. If you choose to assign a variable, you get the Variables dialog box, explained under “[Variables dialog box](#),” p. 267.

UCL, LCL These pop-up menus and associated text fields allow you to calculate or to assign constants or variables for the control limits. By default, these values are calculated from the data, as indicated by the Calculated choice. You also have a choice for how to assign a constant or a variable to these lines. When (abs) follows constant or variable, the values used are the actual values of the constant or of the cases in the variable. For choices in which (rel) follows Constant or Variable, the values are measured relative to the center line, with positive values measured above the center line, and negative values measured below.

Cancel This This button returns all settings to their previous values for the current chart selected in the Chart pop-up menu. It has no effect on the settings for any other charts that have been edited since clicking the Lines button.

Cancel All This button returns all settings to their previous values for all charts that have been edited since clicking the Lines button.

Variables dialog box



If you choose variables for any of the line specifications, you will get the Variables dialog box. This dialog box has a format and function that is very similar to the variable browser.

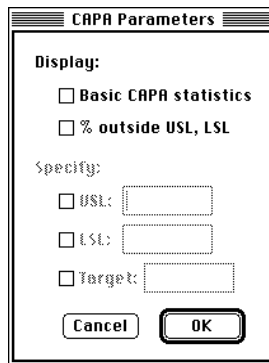
Data This pop-up menu enables you to select the dataset from which the variables are chosen. It gives you access to open datasets, or allows you to open a closed dataset. Normally, the dataset you choose here will be different from the one that is used for the analysis itself.

Order This pop-up menu allows you to select between dataset, alphabetical, class and usage orderings of the variables that appear in the scrolling list. This option only affects the order of variables in the scrolling list; it has no effect on which variables appear in this list.

Choose a variable This scrolling list allows you to choose a variable from those contained in the selected dataset. Either double-click on the variable, or select a variable and click Use to assign that variable to a line. Note that if there are n subgroups in the analyzed data, only the first n rows in any line variable are used.

CAPA Parameters dialog box

The CAPA Parameters dialog box appears when you click the CAPA button in the QC Subgroup Measurements or QC Individual Measurements dialog boxes.



The image shows a dialog box titled "CAPA Parameters". It contains two sections: "Display:" and "Specify:". Under "Display:", there are two checkboxes: "Basic CAPA statistics" and "% outside USL, LSL". Under "Specify:", there are three checkboxes: "USL:", "LSL:", and "Target:", each followed by a text input field. At the bottom of the dialog are "Cancel" and "OK" buttons.

The capability analysis table is displayed in the view only if one or both of the Display checkboxes is enabled. When either Display checkbox is enabled, the OK button is not activated unless one or more specification parameters are entered.

Basic CAPA statistics This checkbox allows you to display, in the capability analysis table, the various C_p indices and the centering index, k . These values will be computed only if the appropriate specifications are entered.

% outside USL, LSL This checkbox allows you to display in the capability analysis table the percentage of observations outside the USL and LSL. These values are computed only if the specification parameters are entered.

USL Enabling this checkbox requires you to specify a value for the USL (upper specification limit) in the text box. You must specify either USL or LSL to compute a capability index. The value of USL must be greater than the Target and the LSL.

LSL Enabling this checkbox requires you to specify a value for the LSL (lower specification limit) in the text box. You must specify either USL or LSL to compute a capability index. The value of LSL must be less than the Target and the USL.

Target Enabling this checkbox requires you to specify a value for the specification target in the text box. This value is required only for computation of C_{pm} . The value for the Target must be between those for the USL and the LSL.

When only USL or LSL is entered, all calculated minima are the quantities computed for the limit that is specified, i.e., the unspecified limit is ignored. See [“Capability analyses,” p. 476](#) for more on computation of capability indices.

Data requirements

QC subgroup measurement analyses require one continuous and one nominal variable. These are referred to as the measurement and subgroup variables, respectively. The measurement variable has the measurements that are the object of analysis, e.g., bolt lengths. The subgroup variable indicates the subgroup from which each measurement is taken, as pictured below.

	Measurement	Subgroup
► Type:	Real	String
► Source:	User Entered	User Entered
► Class:	Continuous	Nominal
► Format:	Free Format Fi...	•
► Dec. Places:	2	•
1	20.10	'Week 1
2	20.43	'Week 1
3	20.17	'Week 1
4	20.57	'Week 2
5	19.95	'Week 2
6	20.04	'Week 2
7	19.11	'Week 3
8	19.46	'Week 3
9	19.77	'Week 3

Please note the following when assigning variables to these analyses:

1. The number of measurements is determined by the total number of included cases (rows) in the measurement variable.
2. Subgroup sizes are determined by the number of cases with a particular value of the subgroup variable.
3. The ordering of subgroups in any subgroup measurement result is determined by the alpha-numeric values within the subgroup variable.

If all of your subgroups have the same number of measurements, you can probably use a formula to generate the values of the subgroup variable. This will save you repetitive and potentially inaccurate typing of subgroup names. For instructions, see [“How can I generate subgroup and labeling variables?,” p. 242 of Using StatView.](#)

Variable browser buttons	
Add	Select one measurement variable (continuous), and one subgroup variable (nominal), then click the Add button. Each additional measurement variable creates a new analysis using the original subgroup variable. Each additional subgroup variable creates a new analysis using the original measurement variable.
Split By	When you assign one or more split-by variables (nominal) to a subgroup measurement analysis, results are displayed separately for each cell defined by the split-by variable(s).

Results

Xbar Statistics results

Xbar charts can be plotted as line, point, needle, or bar plots. You make this choice in the analysis browser. The default result is a line chart.

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table shows results only from those subgroups that violate one or more of the chosen tests.

Xbar charts	Plotted points	Give the mean for each subgroup.
	Center, UCL and LCL lines	Center line gives the mean of all measurements (the process mean, μ), or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the mean for each subgroup, or the values specified in the Lines dialog box.
Xbar table	Count	Gives the number of measurements in each subgroup.
	Mean	Gives the mean from each subgroup.
	Center	Gives the mean of all measurements (the process mean, μ), or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the mean for each subgroup, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are subgroup names as specified by the subgroup variable. Numbers to the right of each row are the numbers of any violated special causes tests that are currently enabled.

Special Causes Definitions table

This is a rather unusual result table because it displays no computed results. These definitions are displayed mainly to aid the interpretation of violations that appear on control charts. This table is displayed only if Show definitions table is enabled in the Tests for Special Causes or Custom Tests dialog box.

Contents	Gives the definitions for those tests enabled in either the Tests for Special Causes or the Custom Tests dialog box, depending on which is chosen from the Special causes tests to perform pop-up menu.
----------	---

R Statistics results

R charts can be plotted as line, point, needle, or bar plots, depending upon which items are selected in the analysis browser. The default graph is a line chart. Both the center line and the control limits for R charts will vary among subgroups if subgroup sizes vary

R charts	Plotted points	Give the range for each subgroup.
	Center, UCL and LCL lines	Center line gives the predicted value of the range for each subgroup, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the range for each subgroup, or the values specified in the Lines dialog box.

R table	Count	Gives the number of measurements in each subgroup.
	Range	Gives the range for each subgroup.
	Center	Gives the predicted value of the range for each subgroup, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the range for each subgroup, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are subgroup names as specified by the subgroup variable.

S Statistics results

S charts can be plotted as line, point, needle, or bar plots, depending upon which items are selected in the analysis browser. The default graph is a line chart. Both the center line and the control limits for S charts will vary among subgroups if subgroup sizes vary.

S chart	Plotted points	Gives the standard deviation for each subgroup.
	Center, UCL and LCL lines	Center line gives the predicted value of the standard deviation for each subgroup, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the standard deviation for each subgroup, or the values specified in the Lines dialog box.
S table	Count	Gives the number of measurements in each subgroup.
	Std. Dev.	Gives the standard deviation for each subgroup.
	Center	Gives the predicted value of the standard deviation for each subgroup, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the standard deviation for each subgroup, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are subgroup names as specified by the subgroup variable.

CUSUM Statistics results

CUSUM charts can be plotted as line, point, needle, or bar plots. These are available within the CUSUM Statistics heading in the analysis browser. When the Tables show violations only option is checked in the CUSUM Parameters dialog box, this table shows results only from those subgroups with values of S_{Hi} or S_{Li} that exceed the control limits.

CUSUM chart	Plotted points	Gives the high and low cumulative sums for each subgroup. These are keyed in the legend.
	Center line	Gives the zero cumulative sum.
	Upper and lower broken lines	Gives the control limits for the SHi (high sum) and SLi (lower sum). If the Invert lower sum option is disabled in the CUSUM Parameters dialog box, the upper line gives the control limit for both sums.

CUSUM table	Count	Gives the number of measurements in each subgroup.
	z	Gives the standardized deviate of each subgroup mean from the process mean.
	SH, SL	Gives the values of SHi and SLi for each subgroup. If the Invert lower sum option is disabled in the CUSUM Parameters dialog box, SLi is positive.
	Other contents	Labels to the left of each row are subgroup names as specified by the subgroup variable. H or L appear to the right of any row for which SHi or SLi exceeds the control limit.

CAPA table

The capability analysis table is displayed in the view only if specification parameters are entered in the CAPA Parameters dialog box.

Cp, Cpm, CPU, CPL, Cpk	Give the values for the various capability indices.
k	Gives the process centering index.
% > USL, % < LSL	Give the percentage of observations above USL and below LSL.
Norm % > USL, % < LSL	Give the percentage of observations from a normal population (mean= μ , standard deviation= σ), that are above the USL and below LSL.

Summary Table

The summary table shows the following.

K sigma	Gives the sigma multiplier that is used to determine control limits. A missing value (.) indicates that alpha, rather than k -sigma, is used to compute control limits.
Alpha	Gives alpha, the Type I probability of exceeding the control limits. A missing value (.) indicates that k -sigma, rather than alpha, is used to compute control limits.
Sigma	Gives the estimate of sigma as specified in the QC Subgroup Measurements dialog box.
Xbar Center	Gives the value of μ , the process mean.
R Center, S Center	Give weighted estimates of the process range and standard deviation, respectively. A missing value (.) indicates that subgroup sizes are unequal; see R and S results tables instead.
# Groups, # Obs, # Missing	Give number of subgroups, included rows, and missing cases, if any, in the analysis.

Templates

The following templates provide QC subgroup measurement analyses.

QC Analyses	Subgroup Measurements Analysis	Box plot; Xbar and S line charts with 3-sigma control limits; FIR CUSUM line chart; summary table; histogram with normal curve; and descriptive statistics with notes on interpretation.
	Xbar & R Charts	Xbar and R line charts with 3-sigma control limits; summary table.
	Xbar & S Charts, Specify Lines	Xbar and S line charts with control limits given by continuous variables you specify; and summary table.
	Xbar, S & CUSUM Charts	Xbar and S line charts with 3-sigma control limits; FIR CUSUM line chart; and summary table.

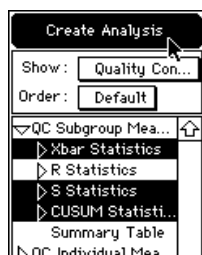
Exercise

Suppose that you are in charge of the quality control effort at a bicycle manufacturer that specializes in limited production frames. The most popular model your company produces is a day touring model called the “Arribe!”, which is a racing-style frame for weekend warriors. This is the product that we will analyze in this example.

The seat tube angle of a bicycle frame can dramatically affect the finished bicycle’s handling characteristics. This is the angle formed by the intersection of the tube that holds the seat post (the seat tube) with the top horizontal frame tube (the top tube). Typically, a small seat tube angle (less than 72°) endows the frame with forgiving (soft) handling characteristics. Weekend warriors want frames that are responsive and quick; they prefer frames with steep seat tube angles (c. 74°). The “Arribe!” is manufactured with these specifications in mind.

In this exercise, we will use the subgroup measurement statistics to see if the frame manufacturing process is in control and capable.

- Open Tube Angle Data from the Sample Data folder
- Scroll through the dataset to examine its contents. You’ll notice that for each of ten days (two work weeks), a technician measured and recorded the seat tube angles from all ten frames produced in the shop.
- Select New View from the Analyze menu
- (Optional) In the analysis browser under Show, choose Quality Control
- In the analysis browser under QC Subgroup Measurements, select Xbar Statistics, S Statistics, and CUSUM Statistics
- Control-click (Windows) or Command-click (Macintosh) to select nonadjacent results.
- Click Create Analysis

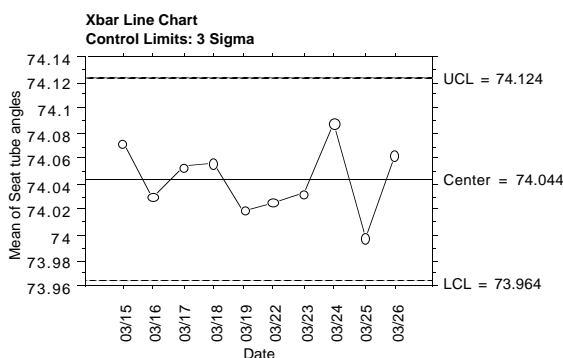


- For Special causes tests to perform, choose Standard
- Click CUSUM
- Check Use FIR and click OK
- Click OK

This creates empty Xbar, S and CUSUM charts in the view.

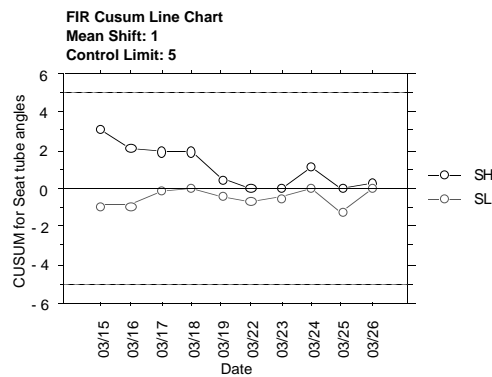
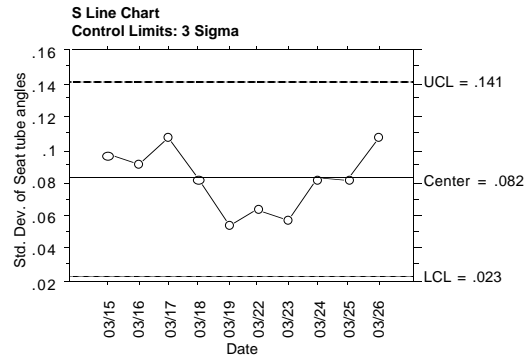
- In the variable browser, select “Seat tube angles” and “Date” and click Add

The Seat tube angles variable is the measurement variable; it appears in the variable browser with an X usage marker. The Date variable is the subgroup variable; it appears with a G usage marker. The analysis calculates and the three completed results appear in the view.



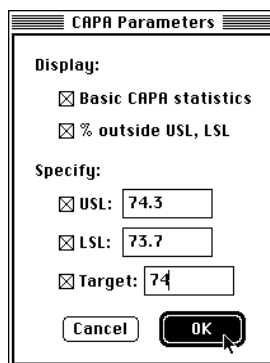
The Xbar line chart that appears at the top of the view indicates that the overall mean of the seat tube angles for the frames in this sample is 74.04 degrees. Because the sample size in each subgroup is the same (10), the control limits (74.12 and 73.96 for the UCL and LCL, respectively) are constant across subgroups. Since there are no test labels next to any of the plotted points, we know these data violate none of the tests for special causes.

Scroll down the view to the S and CUSUM charts. The S chart shows that the estimates of the subgroup standard deviations are above average from 3/15–3/17, then decline on 3/18 to below average values from 3/19–3/23 (the 2 day gap in the sequence is a weekend), then increase from 3/24–3/26. All of these estimates of variation within subgroups are well within the control limits. Corroborating what was indicated in the Xbar chart, the CUSUM chart does not show any indications of a shift in the process mean. From all available evidence, this process appears to be in control.



The next step is to see if the process is capable, i.e., is it producing frames within specification limits? You will now create a capability analysis to answer this question.

- Click one of the results to select it
- Click Edit Analysis (the button at the top of the view)
- Click CAPA
- In the CAPA Parameters dialog box, check all the boxes (turn all the options on)
- Specify 74.3, 73.7 and 74 for the USL, LSL, and Target values and click OK
(These values are derived from an independent engineering analysis of the variation in seat tube angle that yields acceptable performance characteristics.)



CAPA Parameters

Display:

☒ Basic CAPA statistics

☒ % outside USL, LSL

Specify:

☒ USL: 74.3

☒ LSL: 73.7

☒ Target: 74

Cancel OK

- Click OK

Capability Analysis for Seat tube angles

Grouping Variable: Date

USL: 74.3

LSL: 73.7

Target: 74

Cp	1.182
Cpm	1.048
CPU	1.009
CPL	1.355
Cpk	1.009
k	.146
% > USL	1.000
Norm. % > USL	.124
% < LSL	0.000
Norm. % < LSL	.002

The somewhat low values for C_p , C_{pm} and C_{pk} may not be acceptable. Assuming that the data are normally distributed, approximately 12 frames out every 10,000 will have seat tube angles greater than the specification limits (Norm % > USL = 0.124). Since the value of k is not close to 0 and since CPL is appreciably greater than CPU, the relatively low values of the capability indices are due, at least in part, to the fact that the process mean (see Xbar chart) is somewhat greater than the specification target value (i.e., the data are not centered relative to specifications). Though these results do not suggest drastic revision of the production process, the production manager may want to check the alignment of the frame jig and perhaps adjust it slightly.

QC Individual Measurements

This chapter, the second of five regarding StatView's quality control tools, discusses QC Individual Measurements methods. For a general introduction to quality control, see the previous chapter, [“QC Subgroup Measurements,” p. 251](#). Subsequent chapters discuss StatView's other QC methods: [“QC P/Ni,” p. 287](#), [“QC C/U,” p. 299](#), and [“Pareto Analysis,” p. 309](#).

Discussion

As with subgroup measurement analyses, individual measurement analyses are used to evaluate whether a process that produces items with continuous measurements is in control and capable. Unlike the analyses in the previous chapter, [“QC Subgroup Measurements,” p. 251](#), individual measurements analyses require that measurements are not grouped with other measurements. Put another way, each subgroup has only a single measurement. Criteria for deciding when to use individual measurements are discussed in standard texts such as Ryan (1989).

Most of the considerations reviewed in the discussion of subgroup measurement statistics pertain also to individual measurement statistics. Individual measurement statistics require that the measurements be **normally distributed**. With the help of StatView's formula capabilities, you can perform analyses to help you decide when your data are not normally distributed. These techniques are summarized in [“Normality Test,” p. 233 of *Using StatView*](#).

The main differences between individual and subgroup measurement statistics are due to differences in how certain key parameters are estimated. Because there is no within subgroup variation for individual measurement analyses, the methods used to estimate process variation (embodied in the parameter sigma) are different from those used in subgroup measurement analyses. Furthermore, the lack of subgroups means that there can be no range or standard deviation charts for individual measurements. In individual measurement analyses, these subgroup charts are replaced by the moving range chart.

I (individual measurement) charts

In a QC analysis of individual measurements, an I chart together with an MR chart is often the focus of inspection. As with Xbar and R or S charts, I and MR charts are often considered together because they provide complementary information about process variation.

The I chart is essentially the individual measurement equivalent of the Xbar chart: it provides information about variation among measurements. If there are large differences among measurements, the process might be out of control. As with any control chart, the I chart also shows lines indicating the control limits and the center line for the process.

Statistically, the control limits in I charts are based on the same assumptions and are calculated in the same way as control limits for Xbar charts. Accordingly, the interpretations and cautions mentioned in the discussion of Xbar charts apply to I charts as well. For more information, please read [“Xbar \(subgroup mean\) charts,” p. 257](#).

MR (moving range) charts

As noted above, QC analysts often create MR charts along with their I charts. The MR chart provides additional information about the magnitude of variation among measurements in a sample. As such, MR charts take the place in individual measurement analyses of R or S charts in subgroup measurement analyses.

As suggested by its name, an MR chart plots the moving range among measurements in a sample. The moving range is defined as the absolute value of the difference between minimum and maximum values of measurements in a sequence. Typically, this difference is between consecutive measurements (range span = 2), though other range spans can be used as well. MR charts also plot the expected value (center line) and control limits for the moving ranges among measurements.

Tests for special causes

Both the standard and the custom tests for special causes for I charts are identical in definition and interpretation to those applied to Xbar analyses; see [“Tests for special causes,” p. 259](#).

If you are involved in clinical SPC, you probably use a variation of the Westgard rules (Westgard and Barry, 1986). With the exception of the $R_{4,s}$ Westgard rule, these can be easily coded as custom tests as follows:

1. The $1_{3,s}$ rule is equivalent to test 1 with a 3 sigma setting.
2. The $2_{2,s}$ rule is equivalent to test 5 with 2 of 2 consecutive points beyond 2 sigma.
3. The $4_{1,s}$ rule is equivalent to test 6 with 4 of 4 consecutive points beyond 1 sigma.
4. The 10_{xbar} rule is equivalent to test 2 with 10 consecutive points.

If you would also like to check the $R_{4,s}$ rule, we suggest that you use the following dataset formula for a nominal string variable:

```
if Range("Measurement variable", OnlyIncludedRows) > 4 * "sigma"
then "violation"
else .
```

Construct this formula using the value of sigma from the individual measurements summary table. When computed, this formula returns “violation” if the range of the measurement variable exceeds 4 times the estimate of sigma.

CUSUM charts

The individual measurement CUSUM procedure is identical to that used for subgroup measurements with one exception: the sums plotted for individual measurements are the cumulative sums of the adjusted standardized deviate of each measurement (rather than subgroup means) from the process mean. (For more information, please see [“CUSUM \(cumulative sum\) charts,” p. 261](#).) In general, most experts recommend that CUSUM analyses be applied in conjunction with I and MR charts for the broadest detection of assignable causes.

Capability indices

The capability indices for individual measurements are computed just as they are for subgroup measurements. Please read [“Capability indices,” p. 261](#) for more information regarding the application of these indices.

Dialog box settings

QC Individual Measurements dialog box

The settings in this dialog box apply to all individual measurement analyses. Some analyses have additional parameters that are set by clicking the Specify buttons in this dialog box.

Base sigma on This pop-up menu allows you to set a calculation method or a value for sigma, the estimate of the standard deviation. If calculated, sigma can be based on the standard deviation of all the measurements (overall standard deviation, the default) or on the average moving range (average MR), as described in [“Sigma,” p. 477](#). Alternately, you can assign a value to sigma by choosing specify.

Base control limits on This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262](#).

Range span This text field allows you to specify the value of *rs* (range span), which is the number of cases used in calculating moving ranges (see [“MR analyses,” p. 477](#)). The default value is 2.

Special causes tests to perform This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262](#).

Display zones in I charts When enabled, this checkbox causes display of zones A, B, and C in I control charts. By default, this option is disabled.

All the Specify buttons and the dialog boxes they access are identical to the corresponding items in the QC Subgroup Measurements dialog box. See [“QC Subgroup Measurements dialog box,” p. 262](#).

Data requirements

Individual measurement analyses require one continuous and, optionally, one nominal variable. These are referred to as the measurement and labeling variables, respectively. The measurement variable holds the measurements that are the object of analysis, e.g., bolt diameter. The optional labeling variable is used to identify the measurement data, as pictured here:

	Measurement	Labeling
► Type:	Real	Date/Time
► Source:	User Entered	User Entered
► Class:	Continuous	Nominal
► Format:	Free Format Fi...	12:00 AM
► Dec. Places:	2	●
1	20.10	9:00 AM
2	20.43	10:00 AM
3	20.17	11:00 AM
4	20.57	12:00 PM
5	19.95	1:00 PM
6	20.04	2:00 PM
7	19.11	3:00 PM

The following conventions apply to variable use in all individual measurement analyses:

1. The number of observations is equal to the total number of included cases in the measurement variable.
2. The ordering of cases in any QC individual measurement result is determined by the order of the measurements in the dataset.

Variable browser buttons	
Add	Select one measurement variable (continuous) and, optionally, one labeling variable (nominal), then click the Add button. Each additional measurement variable creates a new analysis using the original labeling variable. Each additional labeling variable creates a new analysis using the original measurement variable.
Split By	When you assign one or more split-by variables (nominal) to an individual measurement analysis, results are displayed separately for each cell defined by the split-by variable(s).

Results

I Statistics results

I charts can be plotted as line, point, needle, or bar plots. You make this choice in the analysis browser. The default graph is a line chart.

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table shows results only from those measurements that violate one or more of the chosen tests.

I chart	Plotted points	Give the value for each measurement.
	Center, UCL and LCL lines	Center line gives the mean of all measurements (the process mean, μ), or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the process mean, or the values specified in the Lines dialog box.
I table	Obs	Gives the value of each measurement.
	Center	Gives the mean of all measurements (the process mean, μ), or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the process mean, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are case numbers or measurement names as specified by the labeling variable. Numbers to the right of each row are those of any violated special causes tests that are currently enabled.

Special Causes Definitions tables

This table is identical to the corresponding table available within subgroup measurements. Please see [“Special Causes Definitions table,” p. 270.](#)

MR Statistics results

MR charts can be plotted as line, point, needle, or bar plots, depending upon which items are selected in the analysis browser. The default graph is a line chart.

MR chart	Plotted points	Give the value for each moving range.
	Center, UCL and LCL lines	Center line gives the average moving range, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the average moving range, or the values specified in the Lines dialog box.

MR table	Mov. Range	Gives the value for MR_i as defined in “MR analyses,” p. 477 . With range span = 3, for instance, the first 2 values in this table are missing.
	Center	Gives the average moving range, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the average moving range, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are case numbers or measurement names as specified by the labeling variable.

CUSUM Statistics results

CUSUM charts and tables for individual measurements are identical to those created with subgroup measurements with the following exceptions: individual measurement CUSUM charts and tables show the value of the range span in their titles, and individual measurement CUSUM tables do not have a Count column. Please see [“CUSUM Statistics results,” p. 271](#), for more information.

CAPA results

The individual measurement Capability Analysis results table is identical to that created for subgroup measurements, except that the individual measurement table shows the value of the range span in its title. Please see [“CAPA results,” p. 282](#), for more information.

Summary table

The summary table shows the following.

K sigma	Gives the sigma multiplier that is used to determine control limits. A missing value (.) indicates that alpha, rather than k -sigma, is used to compute control limits.
Alpha	Gives alpha, the Type I probability of exceeding the control limits. A missing value (.) indicates that k -sigma, rather than alpha, is used to compute control limits.
Sigma	Gives the estimate of sigma as specified in the QC Individual Measurements dialog box.
Xbar	Gives the value of μ , the process mean.
MRbar	Gives the average of the moving ranges.
# Obs, # Missing	Give number of included rows and missing cases in the analysis.

Templates

The following templates provide QC individual measurement analyses.

QC Analyses	Ind & Moving Range Charts	I and MR line charts with 3-sigma control limits; and summary table.
	Ind & MR Charts with Westgard	I and MR line charts with 3-sigma control limits; and summary table with Westgard rules.
	Ind & MR Charts, Specify Lines	I and MR line charts with control limits given by continuous variables you specify; and summary table.
	Ind Measurements Analysis	I and MR line charts with 3-sigma control limits; FIR CUSUM line chart; summary table; histogram with normal curve; and descriptive statistics with notes on interpretation.
	Ind, MR & CUSUM Charts	I and MR line charts with 3-sigma control limits; FIR CUSUM line chart; and summary table.

Exercise

Previously, you completed an exercise to evaluate whether a bicycle frame manufacturing process was in control and capable, with respect to the seat tube angle measurement (see [“Exercise,” p. 273](#)). You might recall that although the process appeared to be in control, the capability analysis indicated that the process was not as capable as it could be. In response to your findings, the production manager made a slight adjustment to the frame jig in an attempt to bring the seat tube angles closer to their target value of 74°. In this exercise, you will use individual measurement analyses to see if the process, after adjustments, is still in control.

- Open Tube Angle Data Post Adj from the Sample Data folder

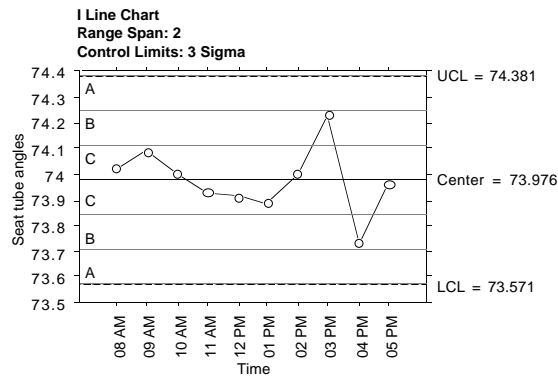
These are data only from the day following the adjustments to the frame jig. The seat tube angle from one frame was measured each hour.

- Select New View from the Analyze menu
- In the analysis browser under QC Individual Measurements, select I Statistics, MR Statistics, and CUSUM Statistics and click Create Analysis
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent items
- For Special causes tests to perform, select Standard
- Check Display zones in I charts
- Click CUSUM
- In the CUSUM Parameters dialog, check Use FIR and click OK
- Click OK

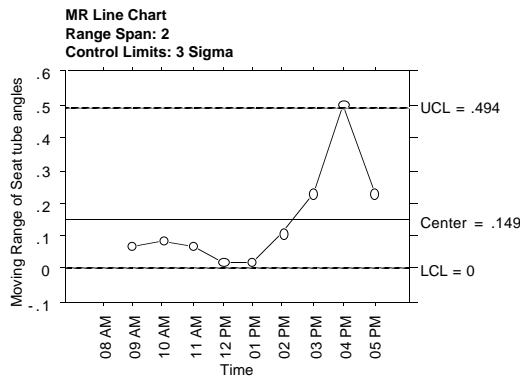
This creates empty I, MR and CUSUM charts in the view.

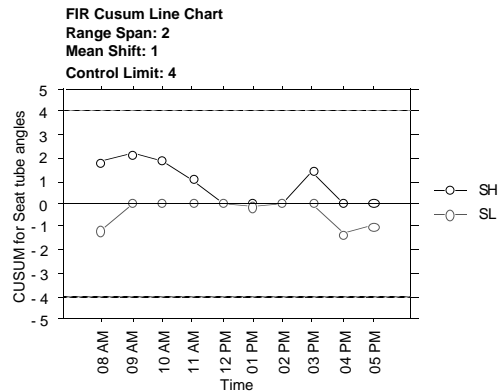
- In the variable browser, select Seat tube angles and Time and click Add

The Seat tube angles variable is the measurement variable; it appears in the variable browser with an X usage marker. The Time variable is the labeling variable; it appears with a G usage marker. The analysis calculates and the three completed results appear in the view.



The I chart at the top of the view gives no indication that the process is out of control. Though the 3 and 4 PM measurements are farther from the mean than the other measurements, all are still well within the control limits. Since no test numbers appear in the plot, we know there are no violations of the tests for special causes.





On the other hand, a quick look at the MR chart (pictured above with the CUSUM chart) suggests that the process might be out of control. In particular, the moving range for the 4 PM measurement (which is the difference between the 3 and 4 PM measurements) is slightly beyond the UCL. Since the I and CUSUM charts do not suggest a shift in the process mean, this could indicate an increase in variation.

After you relay this information to the production manager, he finds that one of the clamps on the frame jig is not as tight as it could be. Though it is difficult to tell from the data when the problem began, the I chart suggests that the clamp might have come loose between 2 and 3 PM. Luckily, even though the process might have been out of control, the frames produced in the late afternoon are still within specification limits.

QC P/NP

This chapter, the third of five regarding StatView's quality control tools, discusses QC *p/np* analyses. For a general introduction to quality control, see the preceding chapter, [“QC Subgroup Measurements,” p. 251](#). Other chapters discussing StatView's QC methods are [“QC Individual Measurements,” p. 277](#), [“QC C/U,” p. 299](#), and [“Pareto Analysis,” p. 309](#).

Discussion

Because it is not always possible, practical or desirable to evaluate measurements from items, QC analysts sometimes gather and analyze data based on item attributes. An attribute typically is some descriptive characteristic of items, rather than a measurement. Typically, these attribute data are in the form of counts of items with particular characteristics.

The most common types of count data for the purposes of quality control are tallies of observations that do not meet the criteria of acceptability, e.g., numbers of nonconforming (i.e., defective) items, or numbers of nonconformities (i.e., defects) per item from a larger sample of items. While both *p/np* and *c/u* analyses are used to analyze attribute data, they have different applications: *p/np* statistics are used to analyze numbers or proportions of nonconforming items; the next chapter, [“QC C/U,” p. 299](#) discusses *c/u* statistics used to analyze data on the numbers of nonconformities *per inspection unit* from a sample of inspection units.

In *p/np* analyses, the data for individual items can have only one of two values, typically defective/not defective. Accordingly, these data follow a binomial distribution.

Although the form of the data is different, *p/np* analyses, like analyses based on continuous measurements, rely heavily on control charts. Accordingly, the interpretation of *p/np* control limits is very similar to that for measurement charts. It should be unlikely that, due simply to random effects, points will lie beyond control limits. Therefore, points beyond control limits are attributed to assignable causes, and require corrective action.

By convention, most *p/np* charts use 3-sigma limits. Unfortunately, these limits often do not approximate the intended probabilities of the binomial distribution. This is because the normal approximation limits (i.e., those based on *k*-sigma) are symmetrical and the binomial distribution is not. A rule of thumb for minimum subgroup size when using *k*-sigma limits is: if n_i is the number of items in a subgroup, and p is the proportion of nonconforming items over all subgroups, then both $n_i p$ and $n_i(1 - p)$ should be greater than 5. See Ryan (1989)

for details. Because the data available often do not meet these requirements, many analysts prefer control limits based on alpha rather than those based on multiples of sigma.

Typically, analysis of attribute data in a quality improvement program is carried out for a number of reasons (Grant and Leavenworth, 1988), among them:

1. To quantify the average of and the variation in the proportion of nonconforming items produced by a process over time.
2. To discover an increase in the number of nonconforming items so that the process can be corrected.
3. To discover a decrease in the number of nonconforming items, which can indicate relaxed inspection standards or point to causes of quality improvement which could be integrated into the process.
4. To suggest places for the use of measurement charts to diagnose quality problems.

Arguably, reason 3 is one of the more important motivations of any quality *improvement* program. In many cases, however, p cannot fall below the 3-sigma lower control limit, because this limit is 0 (StatView sets the LCL to 0 whenever its computed value is ≤ 0). In fact, the 3-sigma LCL will be 0 whenever $p < 9/(9 + n_i)$ (Ryan, 1989).

Accordingly, if it is important to detect a significant *decrease* in the number of nonconforming items, you might have better results if you base the control limits on alpha, rather than on k -sigma. Since control limits based on alpha are derived from cumulative probabilities of the binomial distribution, they are asymmetrical, with the lower tail being shorter than the upper tail. For practical purposes, this means that the alpha-based lower control limit often will be greater than 0 when a comparable k -sigma lower control limit would be less than 0. Since no subgroup can have fewer than 0 nonconforming items, the alpha-based control limits improve the chances of detecting a significant decrease in the number of nonconforming items.

p (proportion defective) charts

In an SPC analysis of counts of nonconforming items, a p chart is a good place to start. The p chart summarizes how the proportion of nonconforming items per subgroup compares among subgroups. If there are large differences among subgroups in the proportion of nonconforming items, the process might be out of control. As with any control chart, the p chart also shows lines indicating the control limits and the center line for the process.

Statistically, the control limits and center lines in p and np charts are based on estimates of the expected patterns of variation in a sample of binomial observations. As noted above, due to the asymmetry of this distribution, the interpretations of k -sigma and alpha-based control limits differ substantially.

np (number defective) charts

The np chart summarizes how the number (rather than the proportion) of nonconforming items per subgroup varies among subgroups. The np chart is often used with the p chart whenever the number of items sampled is constant among subgroups, or the actual number of

nonconforming items per subgroup is of special interest. Along with the number of nonconforming items, np charts show the usual lines indicating the control limits and the center line for the process.

Most analysts recommend np charts only when sample sizes are constant among subgroups. As mentioned above, the statistical bases of calculations for the center line and control limits for np charts are the same as those for p charts. However, because np charts plot center lines corresponding to expected *numbers*, rather than proportions, center lines in np charts will vary among subgroups with differing sample sizes.

Tests for special causes and custom tests

Of the eight tests for special causes used with \bar{X} and I charts, only the first four are applicable to p and np charts (Nelson, 1984). Below are the descriptions and interpretations for each of the four standard tests for special causes as applied to p and np charts.

Note that these tests refer to zones A, B and C. These zones are defined as bands of constant width where Zone A is between 2 and 3 sigmas above and below the center line, Zone B is between 1 and 2 sigmas above and below the center line, and Zone C is between 0 and 1 sigma above and below the center line. Due to the requirement that zones be of constant width, tests for special causes can be performed only on data for which all subgroups are of equal size.

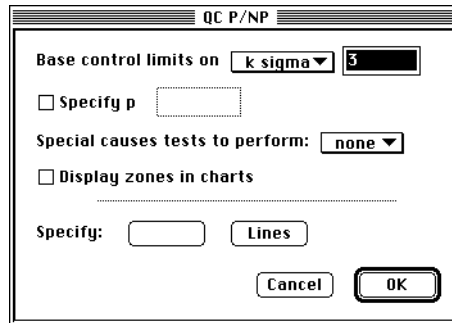
1. **1 point beyond zone A** detects a shift in the proportion of nonconforming items, p , an increase in the estimated standard deviation in the production of defects, or a single aberrant subgroup.
2. **9 consecutive points above or below center line** detects a shift in the proportion of nonconforming items.
3. **6 consecutive increasing or decreasing points** detects a trend or drift in the proportion of nonconforming items.
4. **14 consecutive alternating points** detects systematic alternating effects, such as alternating use of different machines, operators or materials.

It should be kept in mind that a positive result for any of the four tests could be caused by changes in inspection standards that have nothing to do with the process, *per se*. Therefore, standardization and uniform application of criteria for the identification of nonconforming items are critical to the effective application of these tests.

As is true for measurement analyses, the four custom tests for special causes in p/np analyses have the same logical structure as the standard tests. Their difference from the standard tests is that the custom tests give you the ability to define the number of points involved in the calculation of a violation and they allow you to define critical values with arbitrary multiples of sigma rather than with zones about the center line.

Dialog box settings

QC P/NP dialog box



Base control limits on This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262.](#)

Specify p This checkbox and associated text field allow you to specify a value for p , the proportion of nonconforming items over all subgroups. If no value is specified (the default), p is calculated from the data.

Special causes tests to perform This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262.](#)

Display zones in charts When enabled, this checkbox causes display of zones A, B, and C in p and np control charts. It is important to note that zones can be displayed only when subgroup sizes are equal. By default, this option is disabled.

All the Specify buttons and the dialog boxes they access are identical to the corresponding items in the QC Subgroup Measurements dialog box. See [“QC Subgroup Measurements dialog box,” p. 262.](#)

Data requirements

Data for p/np analyses can be in one of two formats. All p/np analyses require one continuous variable, referred to as the nonconformity variable. At least one other variable is also required. If your data are in format 1, then a nominal variable called the **subgroup variable** is required. If your data are in format 2, another continuous variable called the **item count variable** is required.

1. If your data are in format 1, every row has the data for a single item inspected. The nonconformity variable indicates whether each item is conforming (value=0) or nonconforming (value=1). The subgroup variable indicates the subgroup from which each item is taken. If, for instance, the values in one row for the nonconformity and subgroup variables are 1 and 3 pm, then this indicates a nonconforming item from the 3 pm subgroup. In

- another row, values of 0 and 11 am indicate a conforming item from the 11 am subgroup.
2. In the second format, each row has data for a number of items inspected. In this case the nonconformity variable is a count of numbers of nonconforming items. The item count variable is the total number of items inspected for each number of nonconforming items. The subgroup variable is optional; it indicates the subgroup from which each number of nonconforming items is taken. If, for instance, the values in one row of the nonconformity, item count and subgroup variables are 14, 205 and March 3, then there are 14 nonconforming items out of 205 items inspected from the March 3 subgroup.

To summarize, if the nonconformity variable indicates whether *each* item is or is not a nonconforming item (i.e., it is in binomial form, with all values either 0 or 1) then you must use a subgroup variable and you cannot use an item count variable. This is format 1. If, however, the nonconformity variable is a *count* of nonconforming items, then you must use an item count variable; you can, but are not required to use a subgroup variable. This is format 2.

If all of your subgroups are represented with the same number of rows, you can probably use a formula to generate the values of the subgroup variable. This will save you repetitive and potentially inaccurate typing of subgroup names. See [“How can I generate subgroup and labeling variables?,” p. 242 of Using StatView.](#)

Format 1

In Format 1, the number of cases is equal to the total number of items in the entire sample. Subgroup sizes are determined by the number of cases in each subgroup. The ordering of subgroups on the cell axis is determined by the alpha-numeric value of the subgroup variable.

Variable browser buttons	
Add	Select one nonconformity variable (continuous) and one subgroup variable (nominal), then click Add. Each additional nonconformity variable creates a new analysis using the original subgroup and item count variables. Each additional subgroup variable creates a new analysis using the original nonconformity and item count variables.
Item Count	No variables should be specified with the Item Count button.
Split By	When you assign one or more split-by variables (nominal) to a <i>p/np</i> analysis, results are displayed separately for each cell defined by the split-by variable(s).

Format 2

In Format 2, the number of cases must always be less than or equal to the total number of items inspected. In lieu of the optional subgroup variable, each case in the dataset is a separate subgroup, and subgroup sizes are the item counts for each row. If a subgroup name appears in more than one row of the dataset, nonconformity counts and item counts are summed for all rows having that subgroup name. The ordering of subgroups in any *p/np* result is determined

either by the ordering of cases in the dataset, or by the alpha-numeric value of the optional subgroup variable.

Variable browser buttons	
Add	Select one nonconformity variable (continuous) and, optionally, a subgroup variable (nominal), then click Add. Each additional nonconformity variable creates a new analysis using the original subgroup and item count variables. Each additional subgroup variable creates a new analysis using the original nonconformity and item count variables.
Item Count	Select an item count variable (continuous), then click the Item Count button. Each additional item count variable creates a new analysis using the original nonconformity and subgroup variables.
Split By	When you assign one or more split-by variables (nominal) to a <i>p/np</i> analysis, results are displayed separately for each cell defined by the split-by variable(s).

When using Format 2, *p/np* analyses exclude all cases for which the value of the nonconformity variable divided by the item count variable is greater than 1. If no item count variable is assigned, Format 1 is assumed, and any cases for which the value of the nonconformity variable is greater than 1 are excluded.

Results

p results

A *p* chart can be plotted as a line, point, needle, or bar plot. The choice is made in the analysis browser. The default graph is a line chart.

Since the center line is the overall proportion of nonconforming items across all subgroups, it is a constant. The UCL and LCL, however, depend upon subgroup sample sizes and so might vary from subgroup to subgroup. These limits are wider for subgroups with fewer observations (see [“p analyses,” p. 478](#)).

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table shows results only from those subgroups that violate one or more of the chosen tests.

P chart	Plotted points	Give proportion of nonconforming items for each subgroup.
	Center, UCL and LCL lines	Center line gives the overall proportion of nonconforming items, or the value specified in the Lines dialog box. UCL and LCL give the upper and lower control limits about the average proportion of nonconforming items, or the values specified in the Lines dialog box.

P table	Item Count	Gives the number of items in each subgroup.
	Proportion	Gives the fraction of sampled items that do not conform for each subgroup.
	Center	Gives the overall proportion of nonconforming items, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the overall proportion of nonconforming items, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are either row numbers or the subgroup names specified by the subgroup variable. Numbers to the right of each row are the numbers of any violated special causes tests that are currently enabled.

np results

An *np* chart can be plotted as a line, point, needle, or bar plot. The choice is made in the analysis browser. The default graph is a line chart.

In contrast to *p* charts, the center line, as well as the UCL and the LCL for *np* charts can all vary among subgroups depending on the number of items in each subgroup.

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table shows results only from those subgroups that violate one or more of the chosen tests.

NP chart	Plotted points	Give the number of nonconforming items for each subgroup.
	Center, UCL and LCL lines	Center line gives the expected number of nonconforming items for each subgroup, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the expected number of nonconforming items, or the values specified in the Lines dialog box.
NP table	Item Count	Gives the number of items in each subgroup.
	Number	Gives the number of nonconforming items for each subgroup.
	Center	Gives the expected number of nonconforming items for each subgroup, or the values specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the expected number of nonconforming items, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are either row numbers or the subgroup names specified by the subgroup variable. Numbers to the right of each row are the numbers of any violated special causes tests that are currently enabled.

Special Causes Definitions table

This table displays no computed results. These definitions are displayed mainly to aid the interpretation of violations that appear on control charts. The contents of this table are dis-

played only if Show definitions table is checked in the Tests for Special Causes or Custom Tests dialog boxes.

Contents	Gives the definitions for those tests enabled in either the Tests for Special Causes or the Custom Tests dialog box, depending on which is chosen from the Special causes tests to perform pop-up menu.
----------	---

Summary table

The summary table shows the following.

K sigma	Gives the sigma multiplier that is used to determine control limits. A missing value (.) indicates that alpha, rather than <i>k</i> -sigma, is used to compute control limits.
Alpha	Gives alpha, the Type I probability of exceeding the control limits. A missing value (.) indicates that <i>k</i> -sigma, rather than alpha, is used to compute control limits.
P	Gives P, the overall proportion of nonconforming items across all subgroups, or the value specified in the QC P/NP dialog box.
Num Groups, Total Item Count, Num Missing	Give number of subgroups, items, and missing cases, if any, in the analysis.

Templates

The following templates provide QC *p/np* analyses.

QC Analyses	P NP, 3 Sigma, Format 1	For format 1 data, P and NP line charts with 3-sigma control limits; summary table.
	P NP, 3 Sigma, Format 2	For format 2 data, P and NP line charts with 3-sigma control limits; summary table.
	P NP, Alpha, Format 1	For format 1 data, P and NP line charts with alpha=0.0027 control limits; summary table.
	P NP, Alpha, Format 2	For format 2 data, P and NP line charts with alpha=0.0027 control limits; summary table.

Exercise

Exercises in previous chapters evaluate whether a bicycle frame manufacturing process is in control and capable, with respect to the seat tube angle measurement. Sometimes, though, it just isn't practical to measure and analyze every single characteristic of an item to see if a process is in control. Instead, it is often more cost-effective to simply evaluate whether an item is defective or not and to use this information to evaluate process control.

In this exercise, you will analyze this sort of data recorded from frame tubes prior to assembly. Frame tubes need to be meticulously filed, mitered and sanded before they are joined (usually brazed) into a complete frame. The tube ends are then inspected to assure that they fit together properly. Rather than base your analyses on each of the measures that affect whether tubes fit together, you will analyze a single characteristic, specifically whether each individual tube is defective (i.e., is a nonconforming item) or not.

- Open Tube Defects Data from the Sample Data folder

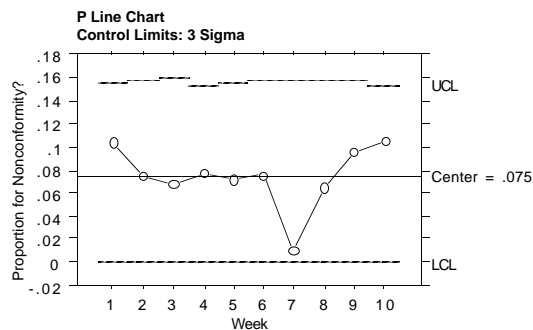
These are data for frame tubes prepared over a 10 week period. Over this period, between 88 and 105 frame tubes per week were prepared and inspected. The Nonconformity? variable codes whether each frame tube inspected is defective (scored as 1) or not (scored as 0).

- Select New View from the Analyze menu
- In the analysis browser under QC P/NP's P Statistics subheader, select Line Chart and Results Table
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent results
- Click Create Analysis
- Click OK to accept the default analysis parameters

This creates an empty p line chart and results table in the view.

- In the variable browser, select Nonconformity? and Week and click Add

Nonconformity? appears in the variable browser with an X usage marker; the subgroup variable Week appears with a G usage marker. The analysis calculates and the two completed results appear in the view.



The center line of the p chart at the top of the view indicates that 7.5% of all tubes inspected are nonconforming (defective). This is not a huge number, but it definitely leaves room for improvement.

One thing to check before proceeding is whether k -sigma gives a reasonable estimate of the control limits. According to our rule of thumb for using the normal approximation (see the ["Discussion," p. 287](#)), no subgroup should have fewer than $5/0.075 \approx 67$ items. A quick look at the p results table shows that the fewest number of items sampled is 88 in week 3, which is well above the suggested minimum.

P Results Table for Nonconformity?

Grouping Variable: Week

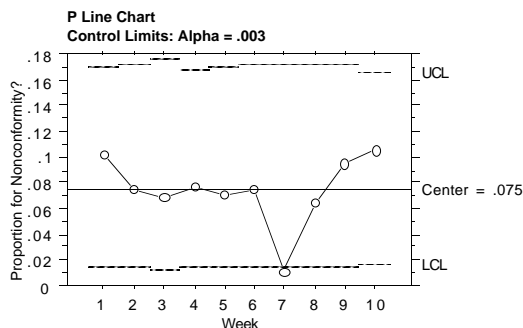
Control Limits: 3 Sigma

	Item Count	Proportion	Center	UCL	LCL
1	97	.103	.075	.155	0.000
2	94	.074	.075	.157	0.000
3	88	.068	.075	.159	0.000
4	103	.078	.075	.153	0.000
5	98	.071	.075	.155	0.000
6	93	.075	.075	.157	0.000
7	94	.011	.075	.157	0.000
8	93	.065	.075	.157	0.000
9	95	.095	.075	.156	0.000
10	105	.105	.075	.152	0.000

On the other hand, we see from this table that there were not enough tubes inspected in any week to give an LCL > 0 when using 3-sigma limits (the minimum is $9/0.075 - 9 = 111$; see the “[Discussion,](#)” p. 287). This means that there is no way to see if, for instance, the improvement seen in week 7 is significant: there can be no proportions < LCL when LCL is 0. This is important, because if the decline in defective items seen in week 7 is significant, you would like to identify the assignable cause for that improvement and incorporate it into the tube preparation process.

You can get a better estimate of the actual binomial probability for the week 7 result if you create a p chart using alpha-based, rather than sigma-based, control limits. Let’s suppose that you want the probability of an out of control signal to be about equal to that from 3-sigma limits. 3-sigma limits are equivalent to a Type I error probability of approximately 0.0027 for each subgroup. Therefore, you will recompute the analysis with control limits based on an alpha of 0.0027.

- Click either the p line chart or the p results table to select it
- Click the Edit Analysis button at the top of the view
- Select alpha from the Base control limits on pop-up menu, then specify 0.0027 for its value
- Click OK



P Results Table for Nonconformity?

Grouping Variable: Week

Control Limits: Alpha = .003

	Item Count	Proportion	Center	UCL	LCL
1	97	.103	.075	.171	.014
2	94	.074	.075	.172	.014
3	88	.068	.075	.176	.012
4	103	.078	.075	.167	.015
5	98	.071	.075	.170	.014
6	93	.075	.075	.173	.013
7	94	.011	.075	.172	.014
8	93	.065	.075	.173	.013
9	95	.095	.075	.172	.014
10	105	.105	.075	.166	.016

These results show that the decline in defective items seen in week 7 is slightly below the LCL for that week. You can therefore conclude that there were assignable causes at work in week 7. Unlike assignable causes identified in measurement analyses, those below the LCL in p/np anal-

yses are desirable because they indicate a significant decline in the production of nonconforming items. These results suggest that you should thoroughly investigate the production process in week 7, try to identify any assignable causes that could account for the improvement and take steps to integrate these causes into the production process.

QC C/U

This chapter, the fourth of five regarding StatView's quality control tools, discusses QC *c/u* analyses. For a general introduction to quality control, see the preceding chapter, [“QC Subgroup Measurements,” p. 251](#). Other chapters discussing StatView's QC methods are [“QC Individual Measurements,” p. 277](#), [“QC P/NP,” p. 287](#), and [“Pareto Analysis,” p. 309](#).

Discussion

Like *p/np* analyses, *c/u* analyses are also used to analyze item attributes. Unlike *p/np* statistics, *c/u* statistics are used to analyze counts of some attribute from items (or inspection units) in a sample, where the attribute is a particular *thing*, usually a kind of defect, e.g., numbers of bubbles in glass beakers, or numbers of scratches on polished mirrors. It is also appropriate to use *c/u* statistics in situations where the inspection unit is, say, a box of items, so long as there is very nearly the same number of items in each inspection unit.

As with the other analyses already described, *c/u* analyses use control charts for evaluating whether or not a process is in control. Control limits are computed such that points that lie beyond them are attributed to assignable causes which should either be eliminated from the process if the point is $> \text{UCL}$, or incorporated into the process if the point is $< \text{LCL}$.

Like the binomial distribution for *p/np* charts, the Poisson distribution on which *c/u* charts are based is asymmetrical. Conventionally, the normal approximation to the Poisson that is implied by using *k*-sigma control limits is considered adequate only when the mean count per inspection unit (*u*) is greater than 5 (Ryan, 1989). Furthermore, for the 3-sigma LCL to be greater than 0, the average number of nonconformities per subgroup ($n_i u$) must be greater than or equal to 9. Therefore, it may be more appropriate in many situations to use alpha-based control limits rather than *k*-sigma limits.

Occasionally, QC analysts use *c/u* charts to analyze combined counts of different types of nonconformities. In general, this is not appropriate because the resulting distribution often is not approximated by the Poisson. Even when counts of the individual nonconformities each come from a Poisson distribution, the combined counts generally will not (Ryan, 1989).

c (count of defects) charts

When analyzing the number of nonconformities from individual inspection units, it is conventional to use a c chart. The c chart summarizes how the total number of nonconformities varies among subgroups. When there is only a single inspection unit per subgroup, the plotted points in this chart are equivalent to the number of nonconformities per inspection unit. If a few subgroups have many more or far fewer nonconformities than others, this may indicate that the process is out of control.

As with any control chart, the c chart also shows lines indicating the control limits and the center line for the process. Because the center lines in c charts correspond to expected counts of nonconformities from each subgroup, center lines will vary among subgroups that comprise different numbers of inspection units.

Statistically, the control limits and center lines in c and u charts are based on estimates of the expected patterns of variation from samples taken from a Poisson distribution. As noted above, due to the asymmetry of the Poisson, the limits predicted by k -sigma and alpha-based estimates can differ substantially.

u (average number of defects) charts

If the number of inspection units is not constant among subgroups, a u chart probably should be used along with or instead of a c chart. The u chart summarizes how, for each subgroup, the average of the number of nonconformities per inspection unit (rather than the total number of nonconformities) compares for all subgroups.

Like other control charts, u charts show the usual lines indicating the control limits and the center line for the process. As mentioned above, the statistical bases of the center line and the control limits for u charts are the same as those for c charts. However, because u charts plot center lines corresponding to expected averages per inspection unit, rather than expected total counts, center lines in u charts do not vary among subgroups, even when they have different sample sizes.

Tests for special causes and custom tests

Below are the descriptions and interpretations for each of the four tests for special causes as applied to c and u charts. As with p/np charts, only the first four of the eight tests for special causes are applicable.

Note that these tests refer to zones A, B, and C. These zones are defined as bands of constant width where Zone A is between 2 and 3 sigmas above and below the center line, Zone B is between 1 and 2 sigmas above and below the center line, and Zone C is between 0 and 1 sigma above and below the center line.

1. **1 point beyond zone A** detects a shift in the average number of nonconformities per inspection unit, u , an increase in the estimated standard deviation in the production of nonconformities, or a single aberrant subgroup.
2. **9 consecutive points above or below center line** detects a shift in the average number of

nonconformities per inspection unit.

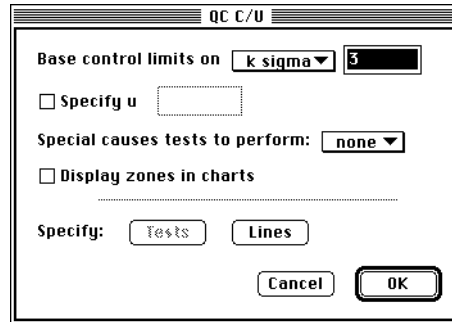
3. **6 consecutive increasing or decreasing points** detects a trend or drift in the average number of nonconformities per inspection unit.
4. **14 consecutive alternating points** detects systematic alternating effects, such as alternating use of different machines, operators, or materials.

As with p/np charts, a positive result for any of the four tests could be caused by changes in inspection standards that have nothing to do with the process, per se. Therefore, standardization and uniform application of criteria for the identification of nonconformities are critical to the effective application of these analyses.

The four custom tests for special causes have the same logical structure as the standard tests. Their difference from the standard tests is that the custom tests give you the ability to define the number of points involved in the calculation of a violation and they allow you to define critical values with arbitrary multiples of sigma rather than with zones about the center line.

Dialog box settings

QC C/U dialog box



Base control limits on This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262.](#)

Specify u This checkbox and associated text field allow you to specify a value for u , the average number of nonconformities per inspection unit for the process. If no value is specified (the default), u is calculated from the data.

Special causes tests to perform This item functions identically to the pop-up menu of the same name in the QC Subgroup Measurements dialog box. For more information, please see [“QC Subgroup Measurements dialog box,” p. 262.](#)

Display zones in charts When enabled, this checkbox causes display of zones A, B, and C in c and u control charts. It is important to note that zones can be displayed only when subgroup sizes are equal. By default, this option is disabled.

All the Specify buttons and the dialog boxes they access are identical to the corresponding items in the QC Subgroup Measurements dialog box. See [“QC Subgroup Measurements dialog box,” p. 262.](#)

Data requirements

All QC *c/u* analyses require only one continuous variable, referred to as the nonconformity variable. Optionally, these analyses may accept either or both a nominal variable and another continuous variable, the latter specified using the Unit Count button. These variables are called the subgroup and unit count variables, respectively. An example dataset with all 3 variables is pictured below.

	Nonconformity	Subgroup	Unit Count
► Type:	Integer	Date/Time	Integer
► Source:	User Entered	User Entered	User Entered
► Class:	Continuous	Nominal	Continuous
► Format:	•	1 / 1	•
► Dec. Places:	•	•	•
1	39	4 / 1	23
2	32	4 / 1	24
3	36	4 / 1	24
4	36	4 / 2	22
5	40	4 / 2	21
6	40	4 / 2	21
7	40	4 / 3	22
8	35	4 / 3	22
9	35	4 / 3	20

For each row in the dataset, the nonconformity variable gives a number of nonconformities. If no unit count variable is specified, then the value in the nonconformity variable is assumed to be for a single inspection unit. If the unit count variable is specified, then the value in the nonconformity variable is the number of nonconformities for the number of inspection units in the unit count variable. If no subgroup variable is specified, then each row is assumed to be from a different subgroup. If a subgroup variable is specified, then counts from the nonconformity and unit count variables are summed for all cases with the same value of the subgroup variable.

If all of your subgroups have the same number of measurements, you can probably use a formula to generate the values of the subgroup variable. This will save you from repetitive and potentially less accurate typing. See [“How can I generate subgroup and labeling variables?,” p. 242 of Using StatView.](#)

Variable browser buttons	
Add	Select one nonconformity variable (continuous) and, optionally, a subgroup variable (nominal). Then click the Add button. Each additional nonconformity variable creates a new analysis using the original subgroup and unit count variables. Each additional subgroup variable creates a new analysis using the original nonconformity and unit count variables.

Unit Count	Optionally, you can select a unit count variable (continuous), then click the Unit Count button. Each additional unit count variable creates a new analysis using the original nonconformity and subgroup variables.
Split By	When you assign one or more split-by variables (nominal) to a <i>c/u</i> analysis, results are displayed separately for each cell defined by the split-by variable(s).

Because two of the three data variables are optional, there are four distinct scenarios that determine how the variables are interpreted in any *c/u* analysis:

1. Only the nonconformity variable is specified. In this scenario, the number of cases equals the number of subgroups which equals the number of inspection units, i.e., there is one inspection unit per subgroup.
2. Only the nonconformity and the subgroup variables are specified. In this scenario, each case is a separate inspection unit, and the number n_i of inspection units in each subgroup is determined by the number of cases with the same value of the subgroup variable.
3. Only the nonconformity and unit count variables are specified. In this scenario, each case represents the totals from a subgroup, with the unit count variable indicating the number n_i of inspection units in each subgroup.
4. The nonconformity, subgroup and unit count variables are all specified. In this scenario, the number n_i of inspection units in each subgroup is the sum of the values for the unit count variable for each level of the subgroup variable. The number of cases in the dataset does not correspond necessarily to either the number of inspection units or the number of subgroups.

Results

c results

A *c* chart can be plotted as a line, point, needle, or bar plot. The choice is made in the analysis browser. The default graph is a line chart. For *c* charts the center line, UCL and LCL vary among subgroups of different sizes.

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table shows results only from those subgroups that violate one or more of the chosen tests.

C chart	Plotted points	Give the number of nonconformities for each subgroup.
	Center, UCL and LCL lines	Center line gives the expected number of nonconformities from each subgroup, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the expected number of nonconformities, or the values specified in the Lines dialog box.

C table	Unit Count	Gives the number of inspection units in each subgroup.
	Count	Gives the number of nonconformities in each subgroup.
	Center	Gives the expected number of nonconformities for each subgroup, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the expected number of nonconformities, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are either row numbers or the subgroup names specified by the subgroup variable. Numbers to the right of rows are those of any violated special causes tests that are currently enabled.

u results

A u chart can be plotted as a line, point, needle, or bar plot. The choice is made in the analysis browser. The default graph is a line chart. Unlike c charts, for u charts only the UCL and LCL, but not the center line, vary with the number of items in each subgroup.

When the Tables show violations only option is checked in the Tests for Special Causes or Custom Tests dialog boxes, this table will show data only from those subgroups that violate one or more of the chosen tests.

U chart	Plotted points	Give the average number of nonconformities per inspection unit for each subgroup.
	Center, UCL and LCL lines	Center line gives the expected number of nonconformities per inspection unit for each subgroup, or the value specified in the Lines dialog box. UCL and LCL lines give the upper and lower control limits about the expected number of nonconformities per inspection unit, or the values specified in the Lines dialog box.
U table	Unit Count	Gives the number of inspection units in each subgroup.
	Count/Unit	Gives the number of nonconformities per inspection unit in each subgroup.
	Center	Gives the expected number of nonconformities per inspection unit for each subgroup, or the value specified in the Lines dialog box.
	UCL, LCL	Gives the upper and lower control limits about the expected number of nonconformities per inspection unit, or the values specified in the Lines dialog box.
	Other contents	Labels to the left of each row are either row numbers or the subgroup names specified by the subgroup variable. Numbers to the right of rows are those of any violated special causes tests that are currently enabled.

Special Causes Definitions table

This table displays no computed results. These definitions are displayed mainly to aid the interpretation of violations that appear on control charts. This table is displayed only if Show definitions table is checked in the Tests for Special Causes or Custom Tests dialog boxes.

Contents	Gives the definitions for those tests enabled in either the Tests for Special Causes or the Custom Tests dialog box, depending on which is chosen from the Special causes tests to perform pop-up menu.
----------	---

Summary Table

The summary table shows the following.

K sigma	Gives the sigma multiplier that is used to determine control limits. A missing value (.) indicates that alpha, rather than <i>k</i> -sigma, is used to compute control limits.
Alpha	Gives alpha, the Type I probability of exceeding the control limits. A missing value (.) indicates that <i>k</i> -sigma, rather than alpha, is used to compute control limits.
U	Gives <i>u</i> , the average number of nonconformities per inspection unit across all subgroups, or the value specified in the QC C/U dialog box.
Num Groups, Total Unit Count, Num Missing	Give the number of subgroups, inspection units, and missing cases, if any, in the analysis.

Templates

The following templates provide QC *c/u* analyses.

QC Analyses	C/U, 3 Sigma Limits	C and U line charts with 3 sigma control limits; summary table.
	C/U, Alpha Limits	C and U line charts with alpha=.003 control limits; summary table.

Exercise

Previously, you analyzed the proportion of nonconforming (defective) frame tubes from the tube preparation process (see “[Exercise,](#)” p. 294). This is one way of using an item attribute to evaluate process control. Of course, individual items (or inspection units) always have more than a single attribute. For instance, none of the frame tubes inspected was *perfect*, i.e., each one had at least a few imperfections, such as stray file marks, a few burrs or an imperfectly mitered butting surface. Data such as these can also be useful for evaluating process control.

In this exercise, you will use *c/u* statistics to analyze the most common type of defect, the number of stray file marks per frame tube, to see if the filing process is in control.

- Open File Mark Data from the Sample Data folder

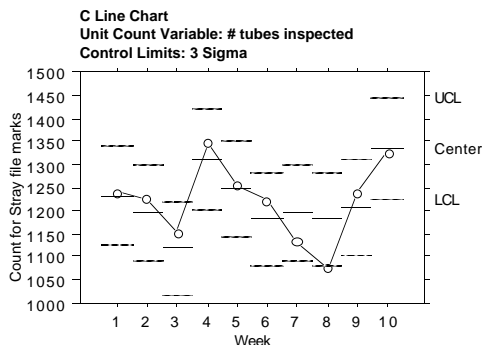
These are data for the same frame tubes that were analyzed earlier. Rather than show data from individual tubes, these data are summary counts for the total number of file marks for all of the tubes inspected in a given week.

- Select New View from the Analyze menu
- In the analysis browser under QC C/U, select C Statistics and U Statistics and click Create Analysis
- Click OK to accept the default parameters

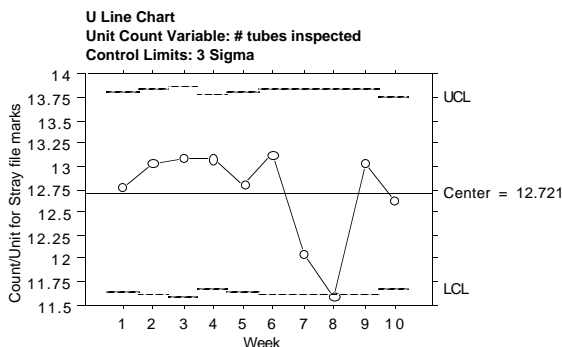
The empty c and u line charts appear in the view.

- In the variable browser, select Stray file marks and Week and click Add
Control-click (Windows) or Command-click (Macintosh) to select nonadjacent variables
- Select # tubes inspected and click Unit Count

Stray file marks appears in the variable browser with an X usage marker; the variable Week appears with a G usage marker and # tubes inspected appears with a C usage marker. The analysis calculates and the two completed results appear in the view.



The c line chart that appears at the top of the view plots the total number of nonconformities and their associated control limits for each week. Though there is no statistical reason why you should not look at the data in this way, the fact that each week has a different value of the center line (owing to the different numbers of inspection units in each week) makes this a rather difficult chart to read. Instead, consider the u chart just below it:



Since the data plotted are the average numbers of nonconformities per inspection unit for each week, the center line is constant. This means that the plotted values from week to week can be compared directly to one another.

The \bar{u} chart gives you some insight into stray file marks as a potential criterion of defectiveness. Recall (from [“Exercise,” p. 305](#)) that there were significantly fewer defective tubes prepared in week 7. The \bar{u} chart above is consistent with this, showing for week 7 a fairly low average number of stray file marks per tube. It also shows, however, that the average number of stray file marks per tube is even lower for week 8, a value that is below the LCL for the process. In tandem with the p/\bar{np} results, this chart suggests that it would be worthwhile to take a closer look at any assignable causes in both weeks 7 and 8, and to try to integrate these into the process.

Pareto Analysis

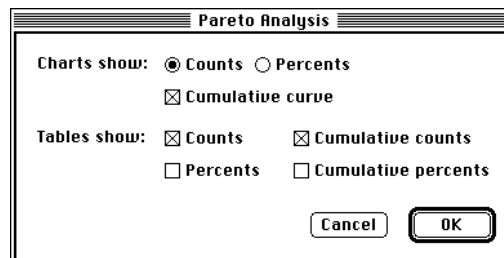
This chapter, the last of five regarding StatView's quality control tools, discusses Pareto analyses. For a general introduction to quality control, see the preceding chapter, [“QC Subgroup Measurements,” p. 251](#). Other chapters discussing StatView's QC methods are [“QC Individual Measurements,” p. 277](#), [“QC P/NP,” p. 287](#), and [“QC C/U,” p. 299](#).

Discussion

In quality control work, sometimes very simple summaries of data can be very valuable. Pareto charts are a case in point. Frequently, Pareto analyses are used to create an easily interpretable summary that can be used to make decisions about where effort should go to improve product quality. A Pareto analysis is simply a frequency distribution of types of defects, with the ordering of the defects determined by their frequency (ordered most to least frequent).

Since types of defects are ordered on a Pareto chart from most to least frequent, identifying the most prevalent types of defects is a simple matter: they are the ones on the left of the graph. Since particular types of defects often are closely related to specific procedures or treatments in the manufacturing process, the Pareto chart gives a good indication of where in the process to concentrate the quality improvement effort.

Dialog box settings



Counts/Percents These radio buttons allow the user to display in the Pareto chart either counts or percentage frequencies for each type of defect.

- Cumulative curve** If this checkbox is enabled (the default), the Pareto chart will plot a curve charting the cumulative frequency of observations across the types of defects.
- Counts** If this checkbox is enabled (the default), the Pareto table will display counts for each type of defect.
- Percents** If this checkbox is enabled, the Pareto table will display the percentage of observations attributable to each type of defect. By default, this option is not checked.
- Cumulative counts** If this checkbox is enabled (the default), the Pareto table will display the cumulative sum of observations attributable to the types of defects, from most to least frequent.
- Cumulative percents** If this checkbox is enabled, the Pareto table will display the cumulative percentage of observations attributable to the types of defects, from most to least frequent. By default, this option is not checked.

Data requirements

Pareto analyses require one nominal variable (the defect type variable) and, optionally, a continuous variable (the defect count variable).

If the defect count variable is not specified, each row in the dataset is tabulated as a single defect of the type indicated in the defect type variable. The total number of rows in the dataset is then equal to the total number of defects observed. An example of such data is pictured below.

	Defect Type
1	Misaligned threads
2	Too long
3	Misaligned threads
4	Too weak
5	Scratches
6	Scratches
7	Distortions
8	Too long
9	Too long
10	Too long
11	Scratches
12	Distortions
13	Misaligned threads
14	Distortions
15	Scratches

If specified, the values of the defect count variable are summed for all rows with a particular value for the defect type variable. An example dataset with both variables is pictured below.

	Defect Type	Defect Count
1	Too weak	322
2	Too long	317
3	Misaligned threads	425
4	Scratches	182
5	Distortions	180

Variable browser buttons	
Add	Select one defect type variable (nominal) and, optionally, one defect count variable (continuous). Then click the Add button. Each additional defect type variable creates a new analysis using the original defect count variable. Each additional defect count variable creates a new analysis using the original defect type variable.
Split By	When you assign one or more split-by variables (nominal) to a Pareto analysis, results are displayed separately for each cell defined by the split-by variable(s).

Results

Pareto charts and tables show the following.

Pareto chart	Plotted bars	Give the incidence (as counts or percentages) of each defect type.
	Cumulative curve	Gives the cumulative sum or percentage of defects attributable to the defect types, summed from most to least frequent.
Pareto table	Count	Gives the number of defects attributable to each defect type.
	Percent	Gives the proportion of all defects attributable to each defect type.
	Cum Count	Gives the cumulative sum of the number of defects attributable to each defect type, summed from most to least frequent.
	Cum Percent	Gives the cumulative proportion of all defects attributable to each defect type, summed from most to least frequent.
	Other contents	Labels to the left of each row are specified by the defect type variable.

Templates

The following template provides Pareto results.

QC Analyses	Pareto Chart & Table	Pareto chart and table.
-------------	----------------------	-------------------------

Exercise

In previous chapters (see [“Exercise,” p. 294](#), and [“Exercise,” p. 305](#)), you analyzed defect attribute data to evaluate whether the frame tube manufacturing process is in control. Along the way, however, these analyses suggested that some assignable causes may have been at work

in weeks 7 and 8. In this exercise, you will use Pareto analysis to look at the differences in the frequencies of types of tube defects between weeks 7 and 8. This information can help you diagnose the assignable causes of the p/np and c/u results.

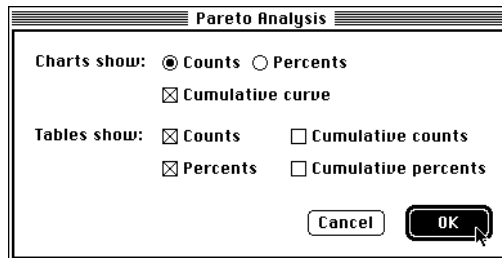
Inspection of prepared frame tubes typically reveals five common types of defects: stray file marks, vise marks, oval cross-section, metal burrs and poor mitering. Of course, these defects are not necessarily equivalent. It may be acceptable for a tube to have fifteen or more relatively superficial file marks, but only the most minor of mitering defects.

You will use the information from the Pareto analysis to see if the pattern of defects is the same for weeks 7 and 8.

- Open Types of Defects Data from the Sample Data folder

This dataset has two variables, one with a random sample of 1000 nonconformities recorded from all frames in week 7 (Week 7 defects), the other with the corresponding data from week 8 (Week 8 defects).

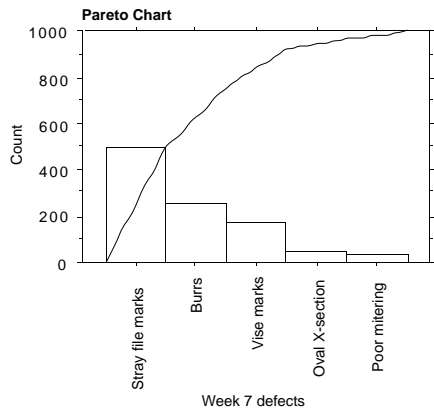
- Select New View from the Analyze menu
- In the analysis browser under Pareto Analysis, select Pareto Chart and Results Table
- Click Create Analysis
- Uncheck Cumulative counts
- Check Percents
- Click OK



The empty Pareto chart and results table now appear in the view.

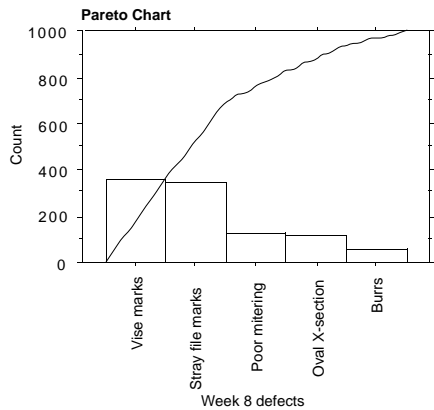
- In the variable browser, select Week 7 defects and Week 8 defects and click Add

This generates two analyses with the same parameters, one for Week 7 defects, the other for Week 8 defects.



Pareto Table for Week 7 defects

	Count	Percent
Stray file marks	493	49.300
Burrs	258	25.800
Vise marks	172	17.200
Oval X-section	40	4.000
Poor mitering	37	3.700



Pareto Table for Week 8 defects

	Count	Percent
Vise marks	352	35.200
Stray file marks	341	34.100
Poor mitering	131	13.100
Oval X-section	115	11.500
Burrs	61	6.100

These charts and tables make clear that the relatively innocuous defects, like stray file marks and metal burrs, occur with higher relative frequency in week 7 than in week 8, while poor mitering, a quite serious defect, has a higher relative frequency in week 8 than in week 7. These results suggest that whatever changes to the process in week 8 that caused the favorable decline in stray file marks and burrs could be correlated with an increase in mitering defects. If this is so, it may be wise to try to incorporate into the process whatever assignable causes appeared in week 7, and to exclude those that appeared in week 8.

Certainly, you cannot conclude from the Pareto charts alone what the differences are between the processes in the two weeks. The differences between week 7 and week 8 could, for instance, have a very simple basis: a frame technician in week 8 may spend more time on filing and other finish work at the expense of time spent on mitering the tubes. This is just one of many possibilities. The Pareto analysis can only give you a better idea of where to look for the sources of defects. Neither it, nor any other QC statistic can be a substitute for first-hand knowledge of the production process.

Formulas

StatView's Formula, Recode, Series, and Random Numbers commands let you create, manipulate, and transform data. Various Criteria commands let you control which data are used for analyses. All these features share a common mathematical expression language as well as a large set of operations, relations, date/time functions, text functions, and numerical functions. This reference chapter discusses that expression language and introduces each of the operations, relations, and functions.

This chapter does *not* discuss the Formula, Recode, Series, and Random Numbers commands themselves. If these are unfamiliar, please consult the chapter [“Managing data,” p. 107 of *Using StatView*](#).

Overview

The section [“Introduction,” p. 317](#), discusses general concepts: working with variable types and formats, rules about arguments and syntax, how formulas and criteria are evaluated, and special discussions of the date/time and text functions.

Subsequent sections detail the various types of functions. A table on the next page lists which functions are discussed in each section. The function types are those seen in function browsers (the scrolling function lists seen in many of StatView's data management windows). You may also view function lists in alphabetical order.

Order: by Function Type	Order: Alphabetical
<ul style="list-style-type: none"> ▷ Date/Time ▷ Logical ▷ Mathematical ▷ Probabilities ▷ Random Numbers ▷ Series ▷ Special Purpose ▷ Statistical 	<ul style="list-style-type: none"> ArcCos(?) ArcCosh(?) ArcCot(?) ArcCsc(?) ArcSec(?) ArcSin(?) ArcSinh(?) ArcTan(?)

A complete index appears at the back of the book.

Section	Function type	Functions discussed
“Operators,” p. 332	Mathematical	$+$, $-$, $*$, $/$, $()$, $^$, $**$, unary $+$, unary $-$
“Sets, intervals, and ranges,” p. 336	Special Purpose	$\{ \}$, $()$, $[]$, $(:]$, $[:]$, $<$, $<=$, $>=$, $>$

“Relations and logical operators,” p. 338	Logical	<, <=, =, >=, >, <>, NOT, AND, ElementOf, IS, ISNOT, OR, XOR, false, if...then...else, IsMissing, IsRowExcluded, IsRowIncluded, NOT, true
“Functions,” p. 347	Date/Time	Date, DateDifference, Day, DayOfWeek, DayOfYear, Hour, Minute, Month, Now, Second, Time, Weekday, WeekOfYear, Year
	Mathematical	Abs, Average, AverageIgnoreMissing, Ceil, Combinations, CumProduct, CumSum, CumSumSquares, Difference, Div, DotProduct, e, Erf, Factorial, Floor, Lag, Ln, Log, LogB, Mod, MovingAverage, Norm, Percentages, Permutations, Pi, Remainder, Round, Sqrt, Sum, SumIgnoreMissing, Trunc
	Probabilities	ProbBinomial, ProbChiSquare, ProbF, ProbNormal, Probt, ReturnChiSquare, ReturnF, ReturnNormal, ReturnT
	Random Numbers	RandomBeta, RandomBinomial, RandomChiSquare, RandomExponential, RandomF, RandomGamma, RandomGaussian, RandomNormal, RandomPoisson, RandomT, RandomUniform, RandomUniformInteger
	Series	BinomialCoeffs, CubicSeries, ExponentialSeries, FibonacciSeries, GeometricSeries, LinearSeries, QuadraticSeries, QuarticSeries, RowNumber
	Special Purpose	ChooseArg, VariableElement
	Statistical	BoxCox, CoeffOfVariation, Correlation, Count, Covariance, GeometricMean, Groups, HarmonicMean, LogOdds, MAD, Maximum, Mean, Median, Minimum, Mode, NumberMissing, NumberOfRows, OneGroupChiSquare, Percentile, Range, Rank, StandardDeviation, StandardError, StandardScores, SumOfColumn, SumOfSquares, TrimmedMean, Variance
	Text	Concat, Find, Len, Substring
	Trigonometric	ArcCos, ArcCosh, ArcCot, ArcCsc, ArcSec, ArcSin, ArcSinh, ArcTan, ArcTanh, Cos, Cosh, Cot, Csc, DegToRad, RadToDeg, Sec, Sin, Sinh, Tan, Tanh

Examples in this chapter

If you try examples shown in this chapter, your results may look a little different from ours, because we choose variable attributes that make the effects of each formula easier to see at a glance. We often:

- 1. set decimal places to 0 (or as few as necessary)
- 2. close the attribute pane, or scroll it down to show summary statistics
- 3. increase or decrease the width of columns

Usually we keep the default type and format: real and free format fixed.

For instance, our example for division looks like this:

A	B	A/B	A÷3
-4	5	-.800	-1.333
-3	-2	1.500	-1.000
●	4	●	●
0	●	●	0.000
1	0	●	.333
5	4	1.250	1.667

But if you changed A+3's type to integer, the results would look different. And if you didn't change the decimal places for A and B, didn't make the columns narrower, and didn't close the attribute pane, the whole window would look different:

	A	B	A/B	A+3
Type:	Real	Real	Real	Integer
Source:	User Entered	User Entered	Dynamic Fo...	Dynamic For...
Class:	Continuous	Continuous	Continuous	Continuous
Format:	Free Form...	Free Forma...	Free Format...	•
Dec. Places:	3	3	3	•
1	-4.000	5.000	-.800	-1
2	-3.000	-2.000	1.500	-1
3	•	4.000	•	•
	0.000	•	•	0
	1.000	0.000	•	0
6	5.000	4.000	1.250	2

Date/time formatting varies according to system software and international configuration. Examples in this manual use a variety of formats.

Finally, be aware that StatView does calculations in the fullest precision of the machine you are using, and results can differ slightly between platforms.

Warning!

To make dataset illustrations easy to read, we often name our variables by actual formula definitions, such as "A+B." We do this so that you can easily identify what each column demonstrates. In practice, though, *you should not give variables names that match existing function names or category level names*. If you have ambiguous expressions, StatView may not interpret your formulas quite the way you intend. *Always give your variables unique, meaningful names*. All function names appear in the table of contents.

Introduction

This section is a general introduction to StatView's expression language, which you may use in the following areas of StatView:

- The Recode command in the Manage menu (see ["Recode data," p. 117 of Using StatView](#)) lets you convert the values of an existing variable to nominal values and lets you change missing values in a variable to some new value. It relies on mathematical and statistical functions, which are discussed in the "Functions" section.
- The Series command in the Manage menu (see ["Series," p. 121 of Using StatView](#)) lets you generate new variables containing special types of series. Series functions are discussed in the "Functions" section.
- The Random Numbers command in the Manage menu (see ["Random numbers," p. 123 of Using StatView](#)) lets you generate new variables with random data from various distributions. Random Numbers functions are discussed in the section ["Functions," p. 347](#).
- The Create Criteria and Edit/Apply Criteria commands in the Manage menu (see ["Create criteria," p. 124](#) and ["Edit/Apply Criteria," p. 129 of Using StatView](#)) and items in the Criteria pop-up menu in the dataset window let you use logical expressions to determine

whether rows are included in statistical and graphical analyses. Criteria rely on functions discussed in [“Sets, intervals, and ranges,” p. 336](#), and [“Relations and logical operators,” p. 338](#).

- The Formula command in the Manage menu (see [“Formula,” p. 109 of *Using StatView*](#)) is the most flexible tool for creating and transforming data. Nearly all of StatView’s expression language is found in the Formula window.

Variable types and formats

StatView works with seven types of variables. Numeric data types are real, integer, and long integer. Most of StatView’s functions are intended for manipulating these numeric types. Numeric data also have text representations, and these representations can be manipulated with text functions; see [“Text functions,” p. 331](#).

Text data types are string and category. StatView provides several text functions for manipulating string data; see [“Text functions,” p. 331](#).

Date/time and currency data have both numeric content and text representation, and they can be manipulated with both numeric functions and text functions. Also, StatView provides a special set of functions for handling date/time data; see [“Date and time functions,” p. 330](#).

Below, we discuss how variable types are handled for the numeric functions that comprise most of the StatView expression language. Text and date/time functions—and their handling of various data types—are discussed separately in the subsequent sections, [“Text functions,” p. 331](#), and [“Date and time functions,” p. 330](#). For details on setting and changing data attributes, see [“Variable attributes,” p. 73 of *Using StatView*](#).

Real

Most of StatView’s functions are numeric. They expect numeric arguments and produce numeric results with type real.

Numeric functions automatically convert all numeric arguments to real numbers before doing any computations. Since type real accepts the greatest range of numbers and allows the greatest precision in calculations, this conversion has no harmful consequences. However, if you change variables produced by formulas to types other than real, you may be surprised by some of the consequences. See the discussions of each data type, below.

You may also use text functions to manipulate character representations of real data; see [“Text functions,” p. 331](#).

Integer and long integer

Integers are whole numbers (no digits after the decimal), so changing results to integer can make them appear “wrong.” Real numbers are rounded up or down to the nearest integer

when you change to type integer. Also, real numbers that exceed the limits of integers or long integers are converted to missing values:

$$-32767 \leq \text{integers} \leq 32767$$

$$-2147483647 \leq \text{long integers} \leq 2147483647$$

In this dataset, variables B and C have been set to be A in their formulas, but we also changed their types, so out of range values are missing and fractional values are rounded:

	A	B	C
Type:	Real	Integer	Long Integer
Source:	User Entered	Dynamic Form...	Dynamic Form...
Class:	Continuous	Continuous	Continuous
Format:	Free Format ...	•	•
Dec. Places:	1	•	•
1	-32768.0	•	-32768
2	-32767.0	-32767	-32767
3	32767.0	32767	32767
4	32768.0	•	32768
5	-2147483648	•	•
6	-2147483647	•	-2147483647
7	2147483647	•	2147483647
8	2147483648	•	•
9	1.5	2	2
10	-3.2	-3	-3

The text functions can be used to manipulate character representations of integer and long integer variables; see [“Text functions,” p. 331](#).

String

String variables can be manipulated with StatView’s text functions, ChooseArg, Concat, Find, Len, and Substring. These are discussed in [“Text functions,” p. 331](#).

If you convert numeric function results to string, the current format’s character representation is copied exactly. This is seen below in variable D, which began as a real variable (with the default three decimal places) set by a formula to be the same as A.

If you convert string results to numeric, or if you use a formula to set a variable equal to a string variable, the values are changed to missing, except those values that happen to be valid numeric values. This is seen in variable F, which is set by formula to be the same as E but has the default type, real:

	A	D	E	F
Type:	Real	String	String	Real
Source:	User Entered	Dynamic Form...	User E...	Dyna...
Class:	Continuous	Nominal	Nominal	Contin...
Format:	Free Format ...	•	•	Free F...
Dec. Places:	1	•	•	3
1	-32768.0	-32768.0	the	•
2	-32767.0	-32767.0	quick	•
3	32767.0	32767.0	brown	•
4	32768.0	32768.0	fox	•
5	-2147483648	-2147483648.0	jumped	•
6	-2147483647	-2147483647.0	over	•
7	2147483647	2147483647.0	the	•
8	2147483648	2147483648.0	4	4.000
9	1.5	1.5	lazy	•
10	-3.2	-3.2	dogs.	•

Similarly, if you convert string to currency or date/time, only those values that happen to be valid in the new type are kept, and all others are missing.

Category

Category variables may be manipulated with text functions (see [“Text functions,” p. 331](#)), relations, and logical operators (see [“Relations and logical operators,” p. 338](#)). Category variables are introduced in [“Categories,” p. 80 of Using StatView](#).

Using category variables with numeric functions works in one of two ways, depending on whether you first create the variable and then change its type to category, or you first set its type to category and then create its values:

- 1. If you first create a variable and then change it from real to category, it is given initial group names that are “Group for” and the character representation of the numbers in their original numeric format. This is seen below in B, which was first set to the formula A and then changed from real (and the default three decimal places) to category.
- 2. If you first set a variable to category and then create its values with a formula, values are mapped onto group names according to the underlying integer “indices” of the group names. This is seen below in variable C. Our category definition has groups “Small,” “Medium,” and “Large,” so values 1, 2, and 3 are changed to those groups, respectively; 1.5 is rounded to 2 and changed to “Medium,” and all other values are missing.

	A	B	C
Type:	Real	Category	Category
Source:	User Entered	Dynamic Formula	Dynamic Form...
Class:	Continuous	Nominal	Nominal
Format:	Free Format Fi...	●	●
Dec. Places:	3	●	●
1	-1.000	Group for -1.000	●
2	0.000	Group for 0.000	●
3	1.000	Group for 1.000	Small
4	2.000	Group for 2.000	Medium
5	3.000	Group for 3.000	Large
6	5.000	Group for 5.000	●
7	1.500	Group for 1.500	Medium

Currency

Currency data are numeric data with special formatting options. See the Format pop-up menu in the attribute pane for choices available.

In every other regard, currency data are the same as real data. The numeric content of currency variables can be manipulated with numeric functions according to the same rules described above for type real. Below, G is set by formula to A, and then changed to have type currency and Japanese yen format. While the numbers *look* different, they behave the same.

	A	G
Type:	Real	Currency
Source:	User Entered	Dynamic Formula
Class:	Continuous	Continuous
Format:	Free Format ...	(¥1,234,567.890) (Japan)
Dec. Places:	1	3
1	-32768.0	(¥32,768.000)
2	-32767.0	(¥32,767.000)
3	32767.0	¥32,767.000
4	32768.0	¥32,768.000
5	-2147483648	(¥2,147,483,648.000)
6	-2147483647	(¥2,147,483,647.000)
7	2147483647	¥2,147,483,647.000
8	2147483648	¥2,147,483,648.000
9	1.5	¥1.500
10	-3.2	(¥3.200)

You may use text functions to manipulate character representations of currency data.

Date/Time

Date/time data are numeric data with special formatting options. Special functions for working with date/time values are discussed separately in the section “Date and time functions.” (The exact formats available may vary according to your installation of system software; see the Format pop-up menu in the attribute pane for choices available to you.)

Date/time data keep time by counting the number of seconds elapsed since midnight of 1 January 1904. Their numeric contents are positive integers ranging from 1 to 4,294,967,295, inclusive. Any values outside this range are replaced with missing values, and any fractional parts are discarded. (If you attempt to enter an invalid date in a date/time data cell, you get an error message.) Within these limitations, you may apply numeric functions as you see fit.

Be sure you understand what you’re asking formulas to do. This bizarre formula is perfectly valid, although it may not be meaningful to divide dates by seven seconds and then add four seconds:

"Some times"/7 + 4

	Some times	Bizarre
Type:	Date/Time	Date/Time
Source:	Dynamic For...	Dynamic Formula
Class:	Continuous	Continuous
Format:	12:00:00 AM	01.01.04 00:00:00
Dec. Places:	•	•
1	01:03:59 AM	01.17.17 17:17:47
2	02:05:00 AM	01.17.17 17:26:30
3	03:06:01 AM	01.17.17 17:35:13
4	04:07:02 AM	01.17.17 17:43:56
5	05:08:03 AM	01.17.17 17:52:39

You may also use text functions to manipulate character representations of date/time data; see [“Text functions,” p. 331.](#)

Casewise and columnwise operations

For each function, we specify its direction of operation: whether the function works horizontally or vertically—**casewise** or **columnwise**. When functions work columnwise, we specify

whether they produce the same result for every row or whether results differ from row to row. (Some terms: A **case** is a horizontal row of data. A **variable** is a vertical column of data. A single case of a single numeric variable is a number, or **constant**. A number you specify—such as 7—is also a constant.)

Casewise

Many functions operate horizontally on a **casewise** basis, producing a separate result for each case of the new variable based on the values in that case of the variable(s) specified as argument(s) to the function. For example, adding two variables produces a new variable, which is a column of sums for each row. Here, the variable A+B is produced by adding across A and B once for each row:

A+B

	A	B	A+B
1	-4	5	1
2	-3	-2	-5
3	•	4	•
4	0	•	•
5	1	0	1
6	5	4	9

Casewise operations are “refreshed” between rows. That is, StatView adds -4 and 5 on row 1, records its answer of 1 in the new variable, and starts “fresh” for row 2 by adding -3 and -2. (It does *not* carry the answer from the previous row into the new operation—that is, it does *not* add -4, 5, -3, and -2 to get its answer for row 2.)

Columnwise

Other functions operate vertically on a **columnwise** basis. Columnwise functions work with all the values (and the length) of a column to compute a new column result. Columnwise functions do *not* “refresh” between rows.

That new variable may be a single answer repeated down a column. An example of this is Mean, which computes the mean of the variable you specify, basing its computations upon all cases (rows) of that variable:

Mean(A, AllRows)

A	Mean of A
-4	-.2
-3	-.2
•	-.2
0	-.2
1	-.2
5	-.2

Or, that variable may be a variable of different numbers. An example of this is CumSum, which adds each value to the next, recording its “sum in progress” down the new variable:

CumSum(A)

A	CumSum of A
-4	-4
-3	-7
•	•
0	-7
1	-6
5	-1

What this means...

A **casewise** function is always evaluated across rows, and a result in one row does not depend on a result in a previous row. If a value in one row of an argument variable changes, only that row of the formula variable is recomputed.

A **columnwise** function is one that is evaluated downwards, where results are related to results in the rows above, or where the same result fills every row of the column. A columnwise function must be calculated all at once, and it must be recalculated if any value in an argument variable changes.

An important rule: columnwise functions do *not* accept any expressions or other functions as arguments. You may, however, combine casewise and columnwise functions in an expression, and you may nest columnwise functions inside casewise functions. For example, you may multiply (casewise) two sums (columnwise):

```
SumOfColumn(A, AllRows) * SumOfColumn(B, AllRows)
```

And you may nest columnwise and casewise functions inside other casewise functions:

```
Sum(Mean(A, AllRows), StandardDeviation(A, AllRows))
```

But you may not put an expression or function inside a columnwise function:

```
SumOfColumn(A*B, AllRows)  
CumProduct(Mean(D, AllRows))
```

(We occasionally use ~~strike-through text~~ to show formulas that would produce errors.)

If you need to combine functions that do not work together, you can set an intermediate variable to the result of one function, and then apply the other function to that variable, e.g.:

```
A*B  
SumOfColumn(C)  
Mean(D, AllRows)  
CumProduct(E)
```

Arguments

StatView functions are applied to the **arguments** you specify. The argument is the object of the action. In the formula A+7, + is the function and A and 7 are the arguments of the function. StatView represents arguments—things you must specify—with question marks.

Placeholders

In this manual, we use more meaningful placeholders for arguments than the question marks you see in the program. For example, we say that *var* – *var2* does casewise subtraction of *var2* from *var*. Usually we indicate with these placeholders what sort of argument is expected:

Placeholder	What you should supply in its place
<i>var, var2, ...</i>	variables such as A or Weight—usually constants are also acceptable
<i>value, value2, ...</i>	constant values—numbers, or a date/time values, or text strings
<i>n, m, r,</i>	constants—usually integers
<i>x, y, z, a, b, ...</i>	constants—usually real numbers
<i>p</i>	a probability or percentage—a number between 0 and 1, or perhaps a variable containing such numbers
<i>text</i>	a text value (a string constant) or perhaps a variable containing text values
<i>date</i>	a date/time variable or a date/time value in quotation marks and formatted to match any format in the format pop-up menu
<i>expr</i>	any valid expression—any complete combination of functions, relations, and arguments that can be evaluated, such as “Log(A)+7” or “Log(A)>7” or “Log(A)>7 AND Sin(B)=0”

Commas

Many functions take several arguments. For example, the RandomNormal(*mean, stdev*) function generates a series of random numbers from the normal distribution with the mean given by the first argument and the standard deviation given by the second:

RandomNormal(1.5, 3.0)

A comma separates the arguments. However, many international number formats use comma for the decimal character, so StatView must use a different character to separate arguments. For example, French Canadian numeric formatting uses commas for decimals and semi-colons between arguments:

RandomNormal(1,5; 3,0)

StatView adapts automatically to the numeric formatting you specify in the Regional Settings (Windows) or Numbers (Macintosh) control panel. If you have difficulty opening a dataset created on a foreign system, make sure any formulas use separator characters appropriate to *your* configuration.

Variables and constants

Most StatView functions work with variables and constants alike as columns. StatView usually interprets a constant as being a column filled all the way down with that number. For example, you could specify A+7, which is interpreted as “the column of numbers stored in variable A plus a column of sevens.” In simpler terms, think of it as “in each row, add the value of A and the value 7.”

A	A+7
-4	3
-3	4
•	•
0	7
1	8
5	12

Quotation marks

Any variable name or string value that contains spaces or special characters should be enclosed in quotation marks. Also use quotation marks with any variable name or constant value that is the same as a function or operator name. For example, if you want the constant string value “e” instead of the constant 2.718..., use quotation marks. If you use the buttons and browsers in the formula and criteria editing windows, most of the quotation marks you need are provided automatically.

The following is valid:

```
if "Turning Circle" > 40
  then "very large"
  else "typical"
```

But this would cause problems:

```
if Turning Circle > 40
  then very large
  else typical
```

A quoted single word is always interpreted as a string value, even if the word is a function or variable name. Quote single words (such as “Weight” or “e”) with caution!

Expressions

In many cases an argument can also be a longer expression. For example, ?+? means that you can specify a variable or constant in place of each question mark:

A+7

However, you could also replace each question mark with a larger expression:

A*B + C*8

Row inclusion

Many columnwise functions have a final argument that controls which rows of the column are used for computations. For example, Mean(A, AllRows) computes the mean of the variable A, including all rows of A in its computations.

AllRows is the default setting for all such functions—you don’t even need to type it, because StatView types it for you automatically when you double-click Mean in the function browser or when you begin typing Mean and it finishes the typing for you.

If, however, you have included or excluded certain rows, you may want to restrict computations to those rows you've chosen to include, or to those you've chosen to exclude. You may do so by replacing the default `AllRows` argument with `OnlyIncludedRows` or `OnlyExcludedRows`.

Row numbers are dimmed in the dataset window for all rows that are excluded, whether by criteria or by manual exclusion. Here we see the results of three Mean formulas, one with each row inclusion argument, when rows 4 and 5 are excluded:

Mean(A, AllRows)

Mean(A, OnlyExcludedRows)

Mean(A, OnlyIncludedRows)

	A	Mean All	Mean Exc	Mean Incl
1	-4	-.200	.500	-.667
2	-3	-.200	.500	-.667
3	•	-.200	.500	-.667
4	0	-.200	.500	-.667
5	1	-.200	.500	-.667
6	5	-.200	.500	-.667

How do you include and exclude rows? By using criteria, by double-clicking row numbers in the dataset, or by selecting rows and using the Include and Exclude commands in the Manage menu. These are discussed in detail under [Include and exclude rows \[p. 108\]](#) and [“Create criteria,” p. 124 of Using StatView](#).

Missing values

Missing values in a variable usually propagate themselves into the new variables created by formulas—cases having missing values in any of the variables listed as arguments to the function usually get a missing value as a result for that row. In a few cases, a single missing value causes missing values for every result of a function.

Missing values propagate missing values for most logical evaluations (except those using `IS`, `ISNOT`, and `IsMissing`), and any evaluation of missing in a criterion results in row exclusion.

For each function, we specify how missing values affect computations.

Several functions are provided specifically to handle missing values: `IS`, `ISNOT`, `IsMissing`, `AverageIgnoreMissing`, `NumberMissing`, and `SumIgnoreMissing`.

Order of operations

StatView obeys the rules of algebra in evaluating expressions. Operations are performed in this order:

1. Functions without arguments, such as `RowNumber`
2. NOT, unary minus (negative), and unary plus (positive)
3. Functions with arguments, such as `Log(?)`
4. Exponentiation
5. Multiplication and division

- 6. Addition and subtraction
- 7. Comparisons
- 8. Logical conjunctions
- 9. Parentheses

Since parentheses are evaluated last, you can use parentheses to override the normal order of operations. This causes anything inside parentheses to be evaluated before it is “used” by any other operation.

For example, multiplication is usually performed before addition:

$$\begin{aligned} &1+3*4 \\ &=1+12 \\ &=13 \end{aligned}$$

But if we group 1+3 inside parentheses, the addition is performed first, because the multiplication step *4 must wait for the contents of the parentheses:

$$\begin{aligned} &(1+3)*4 \\ &=4*4 \\ &=16 \end{aligned}$$

If more than one set of parentheses are used, expressions are evaluated “inside” first:

$$\begin{aligned} &(1+(3*4))*4 \\ &=(1+12)*4 \\ &=13*4 \\ &=52 \end{aligned}$$

A more complicated example lets us show every possible step in the hierarchy of operations.

$$\text{NOT (RowNumber*Log(C) = B^A-4) AND B+C/D < A}$$

	A	B	C	D	Result
1	-4	5	1	5	0
2	-3	-2	2	5	0
3	5	4	3	4	1
4	0	8	4	3	0
5	1	0	5	2	0
6	5	4	6	1	0

Result is a dynamic variable with the formula shown above. On row 1, the formula is evaluated in this sequence of steps:

Step	Explanation
NOT (RowNumber*Log(C) = B^A-4) AND B+C/D < A	
=NOT (1*Log(C) = B^A-4) AND B+C/D < A	RowNumber=1
=NOT (1*0 = B^A-4) AND B+C/D < A	Log(C)=0
(NOT would ordinarily be executed in this step, but it has to wait for its argument, and parentheses are last in the order of operations.)	
=NOT (1*0 = 0.0016-4) AND B+C/D < A	B^A=0.0016
=NOT (0= 0.0016-4) AND B+0.2 < A	1*0=0, C/D=0.2

=NOT (0 = -3.9984) AND 5.2< A	0.0016-4=-3.9984, B+0.2=5.2
=NOT (0) AND 0	0=-3.9984 is false; 5.2<-4 is false (1 is true, 0 is false)
=1 AND 0	"NOT false" is true. (The parentheses are finally done, leaving NOT and AND to be evaluated. NOT takes precedence over AND.)
=0	"true AND false" is false

Similarly, on row 3:

Step	Explanation
NOT (RowNumber*Log(C) = B^A-4) AND B+C/D < A	
=NOT (3*Log(C) = B^A-4) AND B+C/D < A	RowNumber=3
=NOT (3*0.477 = B^A-4) AND B+C/D < A	Log(C)=0.477
=NOT (3*0.477 = 0.0016-4) AND B+C/D < A	B^A=1024
=NOT (1.431 = 1024-4) AND B+0.75 < A	3*0.477=1.431 C/D=0.75
=NOT (1.431 = 1020) AND 4.75< A	1024-4=1020, B+0.75=4.75
=NOT (0) AND 1	1.431=1020 is false; 4.75<5 is true
=1 AND 1	"NOT false" is true
=1	"true AND true" is true

Left to right evaluation

Exponentiation is performed right to left, meaning that X^Y^Z is interpreted as $X^{(Y^Z)}$. All other operations of equal precedence are performed from left to right. If you need to force right-to-left evaluation, use parentheses.

In many cases, the results would be the same right-to-left as left-to-right, but there are exceptions. For example, this series of logical evaluations yields opposite results when parentheses change the sequence (1 is true and 0 is false):

1 OR 1 AND 0
1 AND 0
0
1 OR (1 AND 0)
1 OR 0
1

Another example is when multiplying and dividing with zero. Here, if we use parentheses to evaluate right to left, we get division by zero, which is undefined, and the result is a missing value:

1/3*0
.333*0
0
1/(3*0)
1/0
.

When in doubt, use parentheses to be sure operations are performed in the order you want.

Remarks

You can embed remarks or comments in formulas to document what the formula does. Simply begin your remark with a forward slash and an asterisk (/*) and end it with an asterisk and a forward slash (*/). You can also begin it with left parenthesis and asterisk and end it with asterisk and right parenthesis. For example,

```
/* recode Country into foreign or domestic */
/* 3 March 1996 */
if Country ElementOf {Japan, Other}
  then "foreign" /* this groups all imports under "foreign" */
  else if Country=USA
    then "domestic" (* this groups all American cars under "domestic" *)
    else "XXX"
/* the else expression flags any rows that don't match Japan, Other, or USA
with "XXX" in case I missed something in my dataset */
```

A remark may be several lines long. StatView ignores any characters it finds in between the /* and */ strings, even if they are valid expressions. You may place a comment anywhere in a formula where a space could be, *except* inside a pair of parentheses. For example, you cannot place a comment inside the argument list for the function Mean:

~~Mean(A, OnlyIncludedRows/* in case criteria are used */)~~

Static and dynamic formulas

You can create variables with two different types of formulas: static formulas and dynamic formulas. Variables with static formulas are computed once from the current state of the dataset and only updated if you reopen the formula dialog and click Compute. Variables with dynamic formulas are computed from the current state of the dataset, and they are updated whenever any changes are made to the dataset that affect the variable.

For example, suppose you create both a dynamic variable and a static variable with the same formula:

A+B

	A	B	A+B Dynamic	A+B Static
Type:	Real	Real	Real	Real
Source:	Use...	Us...	Dynamic Formula	Static Formula
Class:	Con...	Co...	Continuous	Continuous
Format:	Fre...	Fre...	Free Format Fix...	Free Format Fi...
Dec. Places:	0	0	0	0
1	-4	5	1	1
2	-3	-2	-5	-5
3	•	4	•	•
4	0	•	•	•
5	1	0	1	1
6	5	4	9	9

Then you realize that the first two values in A are wrong—they should be positive. You correct those values, and the dynamic variable updates. The static variable does not.

	A	B	A+B Dynamic	A+B Static
Type:	Real	Real	Real	Real
Source:	Use...	Us...	Dynamic Formula	Static Formula
Class:	Con...	Co...	Continuous	Continuous
Format:	Fre...	Fre...	Free Format Fix...	Free Format Fi...
Dec. Places:	0	0	0	0
1	4	5	9	1
2	3	-2	1	-5
3	●	4	●	●
4	0	●	●	●
5	1	0	1	1
6	5	4	9	9

Formula and Recode create dynamic variables by default. Series and Random Numbers create static variables by default (for more information, see the chapter [“Managing data,” p. 107 of Using StatView](#)). To switch any variable from one to the other, use the Source pop-up menu in the variable attribute pane.

Date and time functions

StatView provides a set of functions designed for manipulating date/time values in the Gregorian calendar. The date/time functions are: [Date\(?, ?, ?\) \[p. 369\]](#), [DateDifference\(?, ?, ?\) \[p. 370\]](#), [Day\(?\) \[p. 371\]](#), [DayOfWeek\(?\) \[p. 372\]](#), [DayOfYear\(?\) \[p. 372\]](#), [Hour\(?\) \[p. 382\]](#), [Minute\(?\) \[p. 389\]](#), [Month\(?\) \[p. 391\]](#), [Now \[p. 393\]](#), [Second\(?\) \[p. 419\]](#), [Time\(?, ?, ?\) \[p. 427\]](#), [Weekday\(?\) \[p. 430\]](#), [WeekOfYear\(?\) \[p. 431\]](#), and [Year\(?\) \[p. 431\]](#).

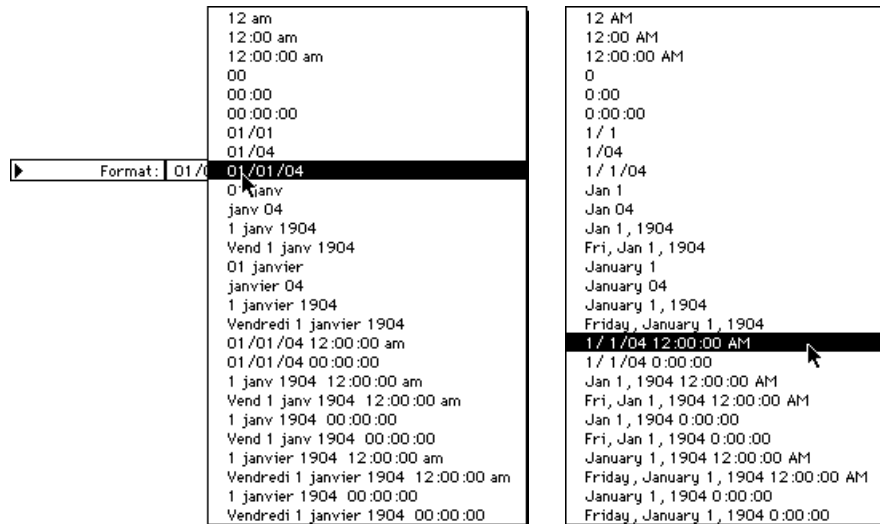
Date/time data measure time by counting the number of seconds elapsed since midnight of 1 January 1904, C.E. The numeric contents of date/time variables are positive integers ranging from 1 to 4,294,967,295, inclusive. Any values outside this range are replaced with missing values, and any fractional parts are discarded. Within these limitations, you may apply any numeric functions you see fit.

Date/time variables have special formatting options for translating numbers of seconds into recognizable dates and times. StatView provides a set of functions designed specifically for working with date and time values. These functions properly interpret 60 seconds as a minute, 60 minutes as an hour, 24 hours as a day, etc.

It is important to understand how date and time values are interpreted and displayed. See [Type \[p. 73\]](#) and [“Format,” p. 79 of Using StatView](#).

Formats

The formats available to you in the Format pop-up menu of the variable attribute pane will vary according to your system and international configuration. For example, if you choose the French Canadian date and time formats, you see the Format choices shown at the left. If you choose the default U.S. formats, you see the choices shown at the right.



Changing data types

Converting date/time variables to type integer or long integer risks loss of data. For example, 4,294,967,295 is too large a number to be stored as integer or long integer, so it is replaced with a missing value. Switching back to date/time variables does not recover the value from missing (but you can Undo the conversion).

Converting date/time values to type category risks loss of data, also. Category variables are limited to 255 levels, so if you have more than 255 different values in a date/time variable, you will lose data.

Converting to types string, currency, and real is safe.

Arguments

When specifying a date/time value as an argument to a function, you may write the value in any format you see in *your* Format pop-up menu. The formats illustrated above and the examples in this manual may not be valid. Always enclose date/time values in quotation marks.

Direction

All date and time functions are casewise—they are evaluated separately for each case.

Text functions

StatView provides a set of functions designed for manipulating text values. These functions are mostly useful with text variables—those with type string and category. Their behavior with text values is straightforward. StatView's text functions are Concat, Find, Len, and Substring. ChooseArg is a special purpose function that is also useful with text variables.

Informative variables

StatView formulas and criteria may only be based on nominal and continuous variables, so if you want to manipulate informative string variables, you must first change their class to nominal.

Numeric formats

Text functions may also be used to manipulate numeric variables—those with type real, integer, long integer, currency, and date/time. Text results are based on the current character representation for the numeric variable, as set by that variable’s current format.

For instance, C is a string variable given by the formula A. Since A has free format fixed and one decimal place, C also shows free-formatted numbers with one decimal place:

	A	C
Type:	Real	String
Source:	User Entered	Dynamic Form...
Class:	Continuous	Nominal
Format:	Free Format...	•
Dec. Places:	1	•
1	-32768.0	-32768.0
2	-32767.0	-32767.0
3	32767.0	32767.0
4	32768.0	32768.0
5	-2147483648	-2147483648.0
6	-2147483647	-2147483647.0
7	2147483647	2147483647.0
8	2147483648	2147483648.0
9	1.5	1.5
10	-3.2	-3.2

Changing data types

All formulas produce variables with type real by default. In most cases you will want to change text function results to have type string or category.

Direction

All text functions are casewise—they are evaluated separately for each row.

Operators

Most of StatView’s operators appear in the keypad area of the Formula window. You may also type them by hand. Operators are used with numeric variables.

?+?

The plus sign does casewise (horizontal) addition of variables and constants. That is, for each case, the values from each column are added to produce a value for that case in the new column. Missing values propagate further missing values when added:

A + B
A + B + 7

A	B	A+B	A+B+7
-4	5	1	8
-3	-2	-5	2
•	4	•	•
0	•	•	•
1	0	1	8
5	4	9	16

You can also use Sum, which accepts multiple variable or constant arguments; Sum(A,B,C,7) is equivalent to A+B+C+7. SumIgnoreMissing is the same, except that missing values are ignored unless a row is missing in every variable.

For columnwise (vertical) addition, see [SumOfColumn\(?, AllRows\)](#) [p. 424] or [CumSum\(?\)](#) [p. 368].

?-?

The minus sign does casewise subtraction of variables or constants. Missing values propagate further missing values when subtracted:

A – B
A – B – 3

A	B	A-B	A-B-3
-4	5	-9	-12
-3	-2	-1	-4
•	4	•	•
0	•	•	•
1	0	1	-2
5	4	1	-2

You can also use Sum with negative arguments, e.g., Sum(A, -B, -C); see [Sum\(?, ...\)](#) [p. 423]. For columnwise subtraction (subtracting a previous case from each case all the way down a column, etc.), see [Difference\(?, 1, 1\)](#) [p. 373].

?*? or ??

The asterisk does casewise multiplication of variables or constants. You can also list two adjacent arguments for multiplication—this is like the *ab* notation for *a***b*. Missing values propagate further missing values when multiplied.

A*B
A(-1)

A	B	A*B	A(-1)
-4	5	-20	4
-3	-2	6	3
•	4	•	•
0	•	•	0
1	0	0	-1
5	4	20	-5

For columnwise multiplication, use CumProduct, which multiplies all the cases of a variable and returns the product “in progress” down the rows of a new variable; see [CumProduct\(?\)](#) [p. 367].

!/?

The slash sign does casewise division of variables or constants. Missing values and division by zero propagate missing values.

A/B

A÷3

A	B	A/B	A÷3
-4	5	-.800	-1.333
-3	-2	1.500	-1.000
•	4	•	•
0	•	•	0.000
1	0	•	.333
5	4	1.250	1.667

A/B has a missing value on row 5 because division by zero is undefined.

?^? or ?**?

The caret ^ and double asterisk ** signs do casewise exponentiation. To raise one argument (a variable or constant) to the power of another, link them with either symbol. Missing values propagate missing values.

A^B

A	B	A^B
-4	5	-1024.000
-3	-2	.111
•	4	•
0	•	•
1	0	1.000
5	4	625.000

You can compute the reciprocal of a power by using a negative exponent. The second row above illustrates this:

$$(-3)^{-2} = \frac{1}{(-3)^2} = \frac{1}{9}$$

You can compute the *n*th root of an argument by raising it to the power 1/*n*. For example, you can compute the square root of A by computing A^(1/2), or the cube root with A^(1/3), etc. Square roots are also built-in with [Sqrt\(?\)](#) [p. 420].

+?

The plus sign + before an argument does casewise unary addition. This is a “positive” sign. Its argument may be a variable, a constant, or another function. Missing values are unchanged.

+A
+B

A	B	Positive A	Positive B
-4	5	-4	5
-3	-2	-3	-2
•	4	•	4
0	•	0	•
1	0	1	0
5	4	5	4

The “positive” function does *not* make negative values positive. What, then, does it do? It simply allows you to include an explicit positive sign so that formulas are easier to read. You might want to include explicit positives in a case like this:

(-Debits)*12/((+Credits)*12)

Use absolute value if you need to convert negative values to positive values; see [Abs\(?\)](#) [p. 348].

-?

The minus sign – before an argument does casewise unary subtraction. This is a “negative” sign. Its argument may be a variable, a constant, or another function. Missing values are unchanged.

-A
-B

A	B	Negative A	Negative B
-4	5	4	-5
-3	-2	3	2
•	4	•	-4
0	•	0	•
1	0	-1	0
5	4	-5	-4

The negative of a positive value is negative, but the negative of a negative value is positive. Zero is without sign.

Since StatView can ignore missing values with addition but not subtraction, you might combine SumIgnoreMissing (see [SumIgnoreMissing\(?, ...\)](#) [p. 424]) and unary subtraction:

SumIgnoreMissing(A, -B, -C, -D)

(?)

Parentheses are used to show grouping and control order of evaluation. ([Order of operations \[p. 326\]](#) discusses this in detail.) If you want to override the normal order of operation, use parentheses to group quantities to be evaluated first. For example:

$$\begin{aligned} 1 + 3 \times 4 &= 13 \\ (1 + 3) \times 4 &= 16 \end{aligned}$$

If more than one set of parentheses are used, expressions are evaluated “inside” first:

$$(1 + (3 \times 4)) \times 4 = (1 + 12) \times 4 = 13 \times 4 = 52$$

Parentheses are also used to delimit arguments for many functions, e.g., $\log(A)$ or $\cos(B)$. Don't worry about typing these parentheses; StatView does it for you when you select functions from the list or the keypad.

A formula definition using parentheses might look like this:

$\text{Log}(A) * (B - C)$

In this example, the first set of parentheses delimits the argument for the log function (StatView inserts these automatically when you click the log button). The second set forces StatView to subtract B and C before multiplying by the log; without parentheses, the multiplication would be performed before the subtraction (and after the log).

Sets, intervals, and ranges

Set, interval, and range functions may be used with both numeric and text data. They are usually used to define criteria or to define conditional transformation formulas.

{...}

Braces create a set containing the elements you list. Its arguments should be values.

For example, Car Data in the Sample Data folder has a nominal variable Type whose values are Small, Sporty, Compact, Medium, and Large. Suppose you only wanted to work with those cars that belong to the Compact, Medium, or Large categories. You could create this criterion:

Type ElementOf {Compact, Medium, Large}

Or, you might use set membership as a test inside if...then to control how a transformation is done:

```
if Type ElementOf {Compact, Medium, Large}
  then Weight*"Turning Circle"/Mean("Turning Circle", AllRows)
  else Weight
```

See [if ? then ? else ? \[p. 341\]](#) and the relation [? ElementOf ? \[p. 345\]](#).

(?:?), [?:?], (?:?), [?:?]

You can indicate numeric intervals in StatView by using colons and grouping marks. Use a colon to separate two endpoint numbers. Use parentheses to indicate an open interval and brackets to indicate a closed interval. You may describe an interval that is half open (open on one end and closed on the other) by using a parenthesis on the open end and a bracket on the closed end.

What do we mean by “open” and “closed”? An **open endpoint** is one where you want the values that are strictly greater than or less than *but not equal to* your endpoint. A **closed endpoint** is one where you want the endpoint included.

Expr	Interval
(1:3)	$1 < x < 3$
[1:3)	$1 \leq x < 3$
(1:3]	$1 < x \leq 3$
[1:3]	$1 \leq x \leq 3$

Such ranges are useful for testing whether a variable’s values belong to a range you specify. Suppose, for instance, you have body temperature readings and want to transform those that are greater than or equal to 36.5° but strictly less than 38° Celsius to their Fahrenheit equivalents. You might use a formula like this:

```
if Temperature ElementOf [36.5:38)
  then Temperature*9/5 + 32
  else .
```

Temperature	Fahrenheit
37.9	100.2
37.6	99.7
36.4	•
37.3	99.1
37.0	98.6
36.7	98.1
38.0	•
38.1	•
39.1	•
38.2	•

See [if ? then ? else ? \[p. 341\]](#) and the relation [? ElementOf? \[p. 345\]](#).

<?, >?

A relation sign followed by an argument returns a range. You may type <= or =< for “less than or equal to,” and you may type >= or => for “greater than or equal to.”

| Expr | Range |
|-------------|----------------|
| < <i>n</i> | $(-\infty, n)$ |
| > <i>n</i> | $(n, -\infty)$ |
| <= <i>n</i> | $(-\infty, n]$ |
| >= <i>n</i> | $[n, \infty)$ |

The range notation is mostly useful in combination with set notation (braces) for criteria; see [{...}](#) [p. 336] and [“Create criteria,” p. 124 of Using StatView.](#)

Suppose you have angle measurements and want to study only those cases whose angles are strictly greater than pi radians. You could create a criterion:

Angles ElementOf {>pi}

Or, if you wanted the cases whose angles are strictly outside the range between plus and minus pi radians, you could create this criterion:

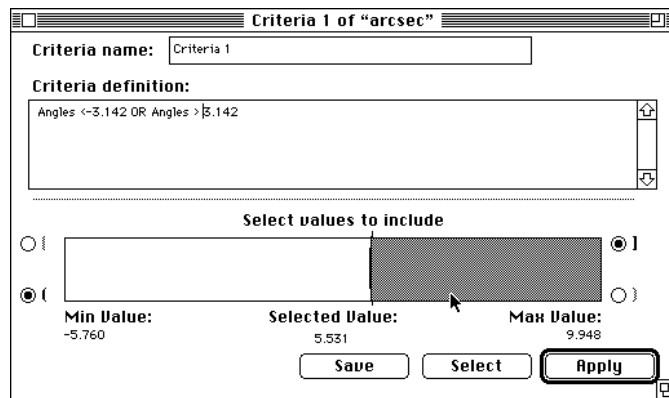
Angles ElementOf {<-pi, >+pi}

Other ways of writing this criterion include:

Abs(Angles) > pi

Angles <-pi OR Angles > pi

Or, you may prefer to use the graphic interface built into the Criteria dialog box:



Relations and logical operators

StatView's relations and logical operators can be used with Formula and Criteria. The result of any logical expression is an evaluation "true" or "false." They can be used with both text and numeric data. For text data, comparisons such as "less than" and "greater than" mean "before" and "after" in ASCII order, which is how text values are compared.

Relations and logical operators work somewhat differently with formulas than with criteria. Formulas create Boolean (true/false) variables, and Criteria create inclusion conditions. For more information about formulas and criteria, see the chapter [“Managing data,” p. 107 of Using StatView.](#)

Formulas

With Formula, you use logical expressions to create new Boolean variables; these variables contain ones for those cases where the expression evaluates to true, zeros where it evaluates to false, and missing values (.) if numeric, blank if string) where either side of the expression is

missing. (See [? IS ? \[p. 346\]](#), [? ISNOT ? \[p. 347\]](#), and [IsMissing\(?\) \[p. 343\]](#) for special handling of missing values.) You can use if...then...else formulas to recode variables; see [if ? then ? else ? \[p. 341\]](#).

With Formula, you may place variables, constants, or expressions on either side of the relation. Following are some valid formulas:

```
A < 7
8 Log(A) >= B + C
9I = A
A < -7 OR A > +8
B > -3 AND B < 5
if B>=A AND (B>I OR B<-3)
  then "This"
  else "That"
```

Criteria

If you are creating or editing criteria, cases where the comparison evaluates to true are included (or selected, if you click the Select button). Cases where the comparison evaluates to false or missing are excluded and their row numbers are dimmed in the dataset window. See [? IS ? \[p. 346\]](#), [? ISNOT ? \[p. 347\]](#), and [IsMissing\(?\) \[p. 343\]](#) for special handling of missing values.

With Criteria, the first argument (the left side of the comparison) *must* be a variable. The second argument (the right side of the comparison) may be a variable, a constant, or some larger expression. Following are some valid criteria:

```
A < 7
A = 9I
A < -7 OR A >=+7
B >= -3 AND B < 3
B>=A AND (B>I OR B<-I)
```

Truth tables

The tables below use formulas to show the results of all possible comparisons of positive, negative, zero, and missing values. The same expressions used for criteria would result in formula rows with trues (1) being included and formula rows with falses (0) or missings (.) excluded. `is` and `isnot` handle missing values differently than `=` and `≠` do; see [? IS ? \[p. 346\]](#) and [? ISNOT ? \[p. 347\]](#).

| A | B | A<B | A≤B | A=B | A≥B | A>B | A≠B | A IS B | A ISNOT B |
|----|----|-----|-----|-----|-----|-----|-----|--------|-----------|
| -1 | -1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| -1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| -1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| -1 | • | • | • | • | • | • | • | 0 | 1 |
| 0 | -1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | • | • | • | • | • | • | • | 0 | 1 |
| 1 | -1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | • | • | • | • | • | • | • | 0 | 1 |
| • | -1 | • | • | • | • | • | • | 0 | 1 |
| • | 0 | • | • | • | • | • | • | 0 | 1 |
| • | 1 | • | • | • | • | • | • | 0 | 1 |
| • | • | • | • | • | • | • | • | 1 | 0 |

Remember, negative numbers of greater magnitude are less than negative numbers of lesser magnitude, e.g., -4 is less than -3. If you want to compare magnitude without regard to sign, use Abs for absolute values. All comparisons return missing values (.) if either or both arguments are missing.

| A | B | A AND B | A OR B | NOT A | A XOR B | true | false |
|---|---|---------|--------|-------|---------|------|-------|
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | • | • | 1 | 0 | • | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | • | • | • | 1 | • | 1 | 0 |
| • | 1 | • | 1 | • | • | 1 | 0 |
| • | 0 | • | • | • | • | 1 | 0 |
| • | • | • | • | • | • | 1 | 0 |

?<?

A “less than” comparison uses the < symbol and returns true (1) if the first argument is strictly less than the second argument. “Less than” returns false (0) if the first argument is equal to or greater than the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

?<=?

A “less than or equal to” comparison uses the <= or <= symbols and returns true (1) if the first argument is less than or equal to the second argument. “Less than or equal to” returns false (0) if the first argument is strictly greater than the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

?=?

An “equal to” comparison uses the = symbol and returns true (1) if the first argument is exactly equal to the second argument. “Equal to” returns false (0) if the first argument is less or greater than the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

You may not specify missing (.) as an argument to =, e.g.:

```
if A = .
  then "unknown"
  else "recorded"
```

To check for missing values, either use IS or IsMissing:

```
if IsMissing(A)
  then "unknown"
  else "recorded"

if A IS .
  then "unknown"
  else "recorded"
```

You can “fill in” missing values in one variable with values from another variable by using a formula such as this (see [NOT\(?\)](#) [p. 345]):

```
If NOT IsMissing(A)
  then A
  else B
```

?>=?

A “greater than or equal to” comparison uses the >= or => symbols and returns true (1) if the first argument is greater than or equal to the second argument. “Greater than or equal to” returns false (0) if the first argument is strictly less than the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

?>?

A “greater than” comparison uses the > symbol and returns true (1) if the first argument is strictly greater than the second argument. “Greater than” returns false (0) if the first argument is equal to or less than the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

?<>?

A “not equal to” comparison uses the <> symbol and returns true (1) if the first argument is less or greater than the second argument. “Not equal to” returns false (0) if the first argument is exactly equal to the second argument. Missing values propagate missing values. Logical expressions are evaluated casewise.

if ? then ? else ?

If *expr* then *expr2* else *expr3* clauses do conditional formulas. The “if *expr*” phrase usually uses a logical expression such as

if Weight > 150

or

if Weight>150 AND Gender=male

For rows whose result of the If... test is true (1), the “then *expr2*” phrase
then "Typical"

or

then Age*Log(Cholesterol)

sets the value of the new variable on that row. When the “if...” test evaluates to false (0), the
“else...” phrase

else "Low"

or

else Age + Cholesterol^2

sets the value on that row.

When the “if...” test evaluates to missing, the result is missing. Therefore, you may want to
consider using functions designed for missing values, such as [IsMissing\(?\) \[p. 343\]](#), [? IS ? \[p. 346\]](#), and [? ISNOT ? \[p. 347\]](#). Logical expressions are evaluated casewise.

Following are some examples. This transformation creates pre- and post-retirement categories
based on 65 as retirement age (don’t forget to change the formula variable to type string):

if Age<65
then "Working age"
else "Retirement age"

Age	Retired?
29	Working age
30	Working age
67	Retirement age
54	Working age
72	Retirement age

This transformation assigns every third row to a different group. It does so by testing whether
the row number divides evenly by 3 (see [Mod\(?, ?\) \[p. 390\]](#)):

if Mod(LineNumber, 3)=0
then "Group3"
else if Mod(LineNumber, 3)=2
then "Group2"
else "Group1"

groups
Group1
Group2
Group3
Group1
Group2
Group3
Group1
Group2
Group3

You can use if...then...else to recode continuous variables into nominal groups. This formula recodes a continuous variable, Age, into a nominal variable with string values. The final else... statement puts any “leftover” cases that haven’t yet been assigned into the group “teenager.”

```
if Age<5
  then "toddler"
  else if Age<13
    then "child"
    else "teenager"
```

A better, safer way to recode variables is to specify exactly what should go in the last group (teenager) and then set any “leftover” cases to a value such as “ZZZ.” After computing the variable, either look at a frequency distribution summary table or sort on the variable and look for any ZZZ values. If you find any, you might need to fix your formula.

```
if Age<5
  then "toddler"
  else if Age<13
    then "child"
    else if Age<20
      then "teenager"
      else "ZZZ"
```

IsMissing(?)

IsMissing(*var*) returns true (1) for every case that is missing in the variable you specify. IsMissing returns false (0) for all other cases. Logical expressions are evaluated casewise.

IsMissing(A)

A	IsMiss(A)
-4	0
-3	0
•	1
0	0
1	0
5	0

Above, we see that the missing value in the third row for A puts a 1 (true) in the second column; all other cases are nonmissing and therefore 0 (false).

IsMissing is useful for Criteria when you don’t want to exclude rows with missing values on your test variable. Suppose you want to exclude cases with extreme weight values, but you don’t want to exclude cases where Weight is missing:

Weight<300 OR IsMissing(Weight)

IsRowExcluded

IsRowExcluded returns true (1) for every case that is currently excluded in the dataset.

IsRowExcluded returns false (0) for any row that is currently included. IsRowExcluded takes no arguments. Logical expressions are evaluated casewise.

IsRowExcluded

	A	IsRowIncluded	IsRowExcluded
1	-4	1	0
2	-3	0	1
3	•	1	0
4	0	0	1
5	1	0	1
6	5	1	0

Inclusion and exclusion are the combined effect of any criteria in effect and any manual Include and Exclude commands. Excluded rows' numbers are dimmed (grayed) in the dataset window.

IsRowExcluded is the complement of IsRowIncluded, [p. 344](#).

IsRowIncluded

IsRowIncluded returns true (1) for every case that is currently included in the dataset. IsRowIncluded returns false (0) for any row that is currently excluded. IsRowIncluded takes no arguments. Logical expressions are evaluated casewise.

IsRowIncluded

	A	IsRowIncluded	IsRowExcluded
1	-4	1	0
2	-3	0	1
3	•	1	0
4	0	0	1
5	1	0	1
6	5	1	0

Inclusion and exclusion are the combined effect of any criteria in effect and any manual Include and Exclude commands. Excluded rows' numbers are dimmed (grayed) in the dataset window.

You might want to use IsRowIncluded or IsRowExcluded to record an inclusion or exclusion you create by hand. Remember, besides using criteria, you can select rows and use the Include and Exclude commands in the Manage menu. Also, you may double-click any row number to toggle the row between included and excluded.

Suppose, for example, you first use a criterion Gender=male and then double-click certain individual cases whose Weight measurements were extremely low or extremely high. Sometimes this is more efficient than putting together complex criteria. Now, you can record this combination of activities for future use with IsRowIncluded or IsRowExcluded. For example, you might set a variable "Typical cases" to IsRowIncluded or a variable "Outliers" to IsRowExcluded. You'll probably want to set the variable to be a static formula (or change it to user entered after computing).

See also [IsRowExcluded \[p. 343\]](#) for the complement of IsRowIncluded.

NOT(?)

The logical operator NOT reverses true/false values. NOT(*expr*) returns true (1) if the expression evaluates to false, false (0) if the expression evaluates to true, and missing (.) or blank string) if the expression evaluates to missing. Logical expressions are evaluated casewise.

false

The operator “false” returns the value false (0) for every case. False takes no arguments. Logical expressions are evaluated casewise.

true

The operator “true” returns the value true (1) for every case. True takes no arguments. Any nonzero, nonmissing value is interpreted as true. Logical expressions are evaluated casewise.

? AND ?

The logical conjunction AND does intersection. Two expressions joined with AND return true (1) if the expressions on both sides are both true; they return false (0) if either or both sides are false. Missing values on either side propagate missing values. Logical expressions are evaluated casewise.

? ElementOf ?

A *var* ElementOf *set* or a *var* ElementOf *interval* comparison returns true (1) when values of the *var* are members of the *set* or *interval* you specify and false (0) when they are not. The first argument should be a variable, and the second argument should be either a set or a range. Missing values propagate missing values. Logical expressions are evaluated casewise.

A ElementOf {1,2,3}

A	A IsIn {1,2,3}
0	0
2	1
4	0
6	0
.	.

ElementOf is especially useful when recoding nominal data, such as this example with Car Data (see [if ? then ? else ? \[p. 341\]](#)):

```
if Country ElementOf {Japan, Other}
  then "foreign"
  else if Country=USA
    then "domestic"
    else "XXX"
```

Country	import?
Japan	foreign
Japan	foreign
Other	foreign
Other	foreign
Other	foreign
Other	foreign
Other	foreign
Other	foreign
USA	domestic
USA	domestic
USA	domestic

We use the final else “XXX” to double-check that Country has no other values that could incorrectly be recoded to domestic. A quick glance at Maximum in the variable attribute pane confirms that the new variable has no rows with XXX, so we know no other Country values were present.

ElementOf is also useful for criteria based on nominal data. This criterion would include only those rows with values Japan and Other, and rows with USA (or missing values, if there were any) would be excluded:

Country ElementOf {Japan, Other}

A set is either a range or a list of elements separated by commas and enclosed by “{” and “}.” An interval is a pair of endpoints separated by a colon “:” and enclosed in parentheses “()” or brackets “[]” as discussed in [“Sets, intervals, and ranges,” p. 336.](#)

? IS ?

An is comparison is an “equal to” comparison that also considers two missing values to be equal to each other. IS returns true (1) if the first expression is equal to the second argument or if both expressions are missing. IS returns false (0) if the arguments are unequal or if only one expression is missing. IS is useful with nominal data. Logical expressions are evaluated case-wise.

Ordinarily, a missing value on either side of a relation causes that case to be missing, because, for example, two unreported ages or weights or names cannot be assumed to be equal (or unequal) just because they were both unrecorded. StatView provides IS and ISNOT for those situations in which missing values can be considered to “match.”

You might want to use IS rather than = when setting a variable according to the values of more than one other variable, as below. We might expect the first formula to produce values of true for rows 3 and 4 (where X=3) and row 1 (where Y=4). Why do we get missing on row 1? Because X is missing on row 1, the first if... then ? else ? [p. 341]) has already evaluated that row to be missing. The second formula uses IS and gets the desired result: row 1 is also true.

```

if X=3
  then 1
  else if Y=4
    then 1
    else 0
if X IS 3

```

```
then I
else if Y=4
  then I
  else 0
```

	X	Y	with =	with IS
1	●	4	●	1
2	2	1	0	0
3	3	6	1	1
4	3	●	1	1
5	5	8	0	0

? ISNOT ?

An ISNOT comparison is a “not equal to” comparison that considers two missing values to be equal. ISNOT returns true (1) if the first expression is strictly less or greater than the second expression. ISNOT returns false (0) if both expressions are exactly equal or both expressions are missing. ISNOT is useful with nominal data. Logical expressions are evaluated casewise.

Ordinarily, a missing value on either side of a relation causes that case to be missing, because, for example, two unreported ages or weights or names cannot be assumed to be equal (or unequal) just because they were both unrecorded. StatView provides IS and ISNOT for those situations in which missing values can be considered to “match.”

You might want to use ISNOT rather than \neq in situations like the example shown for [? IS ?](#) [p. 346].

? OR ?

The logical conjunction OR does union. Two expressions joined with OR return true (1) if either or both expressions are true; they return false (0) if both expressions are false; and they return missing values (. or blank strings) if one is false and the other missing, or if both are missing. Logical expressions are evaluated casewise.

? XOR ?

The logical conjunction XOR does exclusive or. Two expressions joined with XOR return true (1) if one is true and the other false; they return false (0) if both are true or both are false; and they return missing values (. or blank strings) if either or both are missing. Logical expressions are evaluated casewise.

Functions

StatView provides an array of date/time functions for working with date/time data; text functions for manipulating text data and character representations of numeric data; mathematical, statistical, probabilistic, and trigonometric functions for generating and transforming data;

and random data and series functions for generating data. Functions appear in alphabetical order.

Abs(?)

Abs(*var*) returns the casewise absolute value of the variable you specify. Missing values are unchanged.

Abs(A)

Abs(A+B)

A	B	A	A+B
-4	5	4	1
-3	-2	3	5
•	4	•	•
0	•	0	•
1	0	1	1
5	4	5	9

Absolute value is often written with vertical bars. Absolute value is defined by:

$$\begin{aligned} |x| &= x \text{ for all } x \geq 0 \\ &= -x \text{ for all } x < 0 \end{aligned}$$

Casually, absolute value removes negative signs from the quantity it contains, but it does so *after* evaluating the quantity inside. In the example below, the first case for Abs(A+B) is evaluated $|-4 + 5| = |1| = 1$, *not* $|-4 + 5| = 4 + 5 = 9$.

Absolute values are often used for studying “absolute magnitude”—that is, you want to know how large some numbers are, but you don’t care whether the numbers are negative or positive. For example, you might want to work with absolute residuals from a regression. Absolute values are also useful with functions that require non-negative arguments. For instance, you may want to examine the square root of a variable. Square roots, however, are undefined for negative numbers, so you must first apply the absolute value then the square root, e.g., Sqrt(Abs(A)). Otherwise, missing values result.

Researchers often use a Likert scale, where multiple-choice answers indicate a range of response, such as 1–5 being a scale from “strongly agree” to “strongly disagree.” Other questions might reverse the scale, so that the survey doesn’t seem to encourage one opinion over another. To reverse scores from one direction to the other, take the absolute difference between the score and the maximum, and add one. For example, to flip 1–5 to 5–1, use this formula:

I + Abs(Likert–5)

Likert	Reverse
1	5
2	4
3	3
4	2
5	1
1	5
2	4
3	3
4	2
5	1

Grouping parentheses are allowed both inside and outside the absolute value argument, e.g., (A–Abs(B–C) and Abs(A–(B–C)); see (?) [p. 336]. StatView asks you to correct any mistakes you make before it will compute any formula.

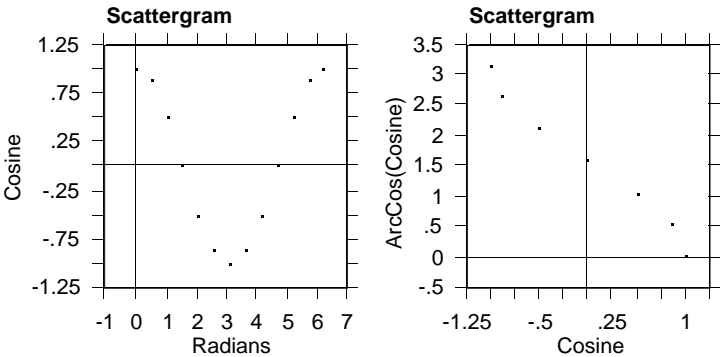
ArcCos(?)

ArcCos(*var*) returns the arccosine in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcCos(Cosine)

Radians π	Radians	Cosine	ArcCos(Cosine)
zero	0.000	1.000	0.000
$\pi/6$.524	.866	.524
$\pi/3$	1.047	.500	1.047
$\pi/2$	1.571	0.000	1.571
$2\pi/3$	2.094	-.500	2.094
$5\pi/6$	2.618	-.866	2.618
π	3.142	-1.000	3.142
$7\pi/6$	3.665	-.866	2.618
$4\pi/3$	4.189	-.500	2.094
$3\pi/2$	4.712	0.000	1.571
$5\pi/3$	5.236	.500	1.047
$11\pi/6$	5.760	.866	.524
2π	6.283	1.000	0.000

Arccosine is often denoted by $\cos^{-1}x$, because it is the inverse function of the cosine function. The arccosine of x is any angle whose cosine is x . Since cosine is a periodic function, many angles have any given cosine, so arccosine is usually understood to mean the “principal value” for a given cosine, which is by convention the angle falling between 0 and π having that cosine. A graph of arccosine against cosine shows this relationship.



ArcCos returns angles expressed in radians. You can convert radians to degrees with RadToDeg(?) [p. 405]. You may specify the value π with Pi [p. 400].

ArcCosh(?)

ArcCosh(*var*) returns the hyperbolic arccosine of a variable or constant. Missing values propagate missing values. The function works casewise.

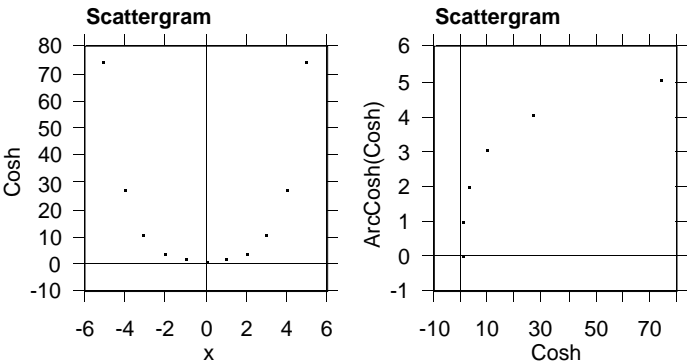
ArcCosh(Cosh)

x	Cosh	ArcCosh(Cosh)
-5	74.210	5
-4	27.308	4
-3	10.068	3
-2	3.762	2
-1	1.543	1
0	1.000	0
1	1.543	1
2	3.762	2
3	10.068	3
4	27.308	4
5	74.210	5

Hyperbolic arccosine is often denoted by $\cosh^{-1}x$, because it is the inverse function of the cosh function. If $\cosh x = y$, then $\cosh^{-1}y = x$. As hyperbolic functions are constructed from exponential functions and exponents inverse to logs, inverse hyperbolic cosine has a logarithmic expression:

$$\cosh^{-1}x = \ln(x \pm \sqrt{x^2 - 1})$$

where $x \geq 1$ and the plus in \pm is used for the principal value. (Either value for ArcCosh would be valid; just as ArcCos takes its preferred “principal value” from the interval between 0 and π , so ArcCosh takes its principal value from the result of the plus sign rather than the minus sign.) Graphs of cosh and arccosh echo their exponential and logarithmic meanings and show the effects of the principal value convention:



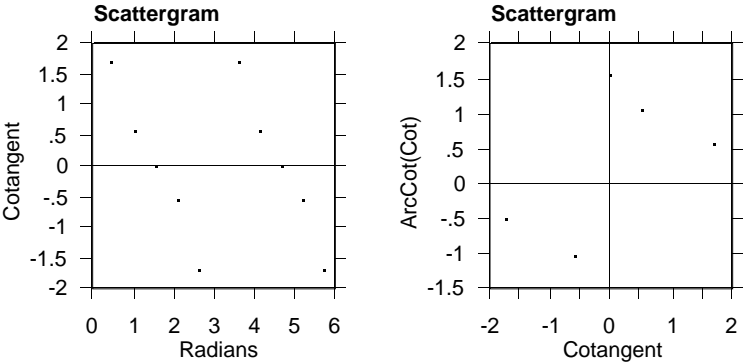
ArcCot(?)

ArcCot(*var*) returns the arccotangent in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcCot(Cotangent)

Radians π	Radians	Cotangent	ArcCot(Cot)
zero	0.000	•	•
$\pi/6$.524	1.732	.524
$\pi/3$	1.047	.577	1.047
$\pi/2$	1.571	0.000	1.571
$2\pi/3$	2.094	-.577	-1.047
$5\pi/6$	2.618	-1.732	-.524
π	3.142	•	•
$7\pi/6$	3.665	1.732	.524
$4\pi/3$	4.189	.577	1.047
$3\pi/2$	4.712	0.000	1.571
$5\pi/3$	5.236	-.577	-1.047
$11\pi/6$	5.760	-1.732	-.524
2π	6.283	•	•

Arccotangent is often denoted by $\cot^{-1} x$, because it is the inverse function of the cotangent function. The arccotangent of x is any angle whose cotangent is x . Since cotangent is a periodic function, many angles have any given cotangent, so arccotangent is usually understood to mean the “principal value” for a given cotangent, which is by convention the angle falling between $-\pi/2$ and $\pi/2$ having that cotangent. A graph of arccotangent against cotangent shows this relationship.



ArcCot returns angles expressed in radians. You can convert radians to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

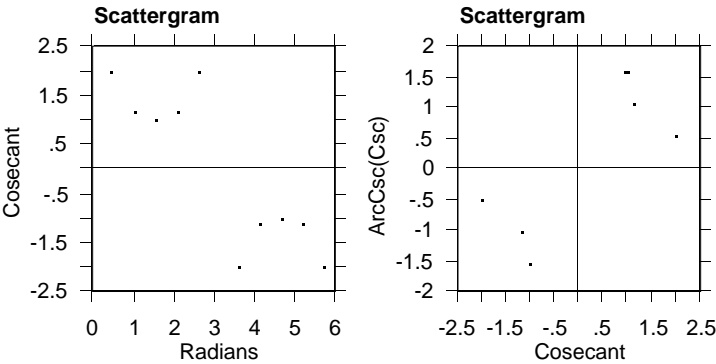
ArcCsc(?)

ArcCsc(*var*) returns the arccosecant in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcCsc(Cosecant)

Radians π	Radians	Cosecant	ArcCsc(Csc)
zero	0.000	•	•
$\pi/6$.524	2.000	.524
$\pi/3$	1.047	1.155	1.047
$\pi/2$	1.571	1.000	1.571
$2\pi/3$	2.094	1.155	1.047
$5\pi/6$	2.618	2.000	.524
π	3.142	•	•
$7\pi/6$	3.665	-2.000	-.524
$4\pi/3$	4.189	-1.155	-1.047
$3\pi/2$	4.712	-1.000	-1.571
$5\pi/3$	5.236	-1.155	-1.047
$11\pi/6$	5.760	-2.000	-.524
2π	6.283	•	•

Arccosecant is often denoted by $\csc^{-1} x$, because it is the inverse function of the cosecant function. The arccosecant of x is any angle whose cosecant is x . Since cosecant is a periodic function, many angles have any given cosecant, so arccosecant is usually understood to mean the “principal value” for a given cosecant, which is by convention the angle falling between $-\pi/2$ and $\pi/2$ having that cosecant. A graph of arccosecant against cosecant shows this relationship.



ArcCsc returns angles expressed in radians. You can convert radians to degrees with [RadTo-Deg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

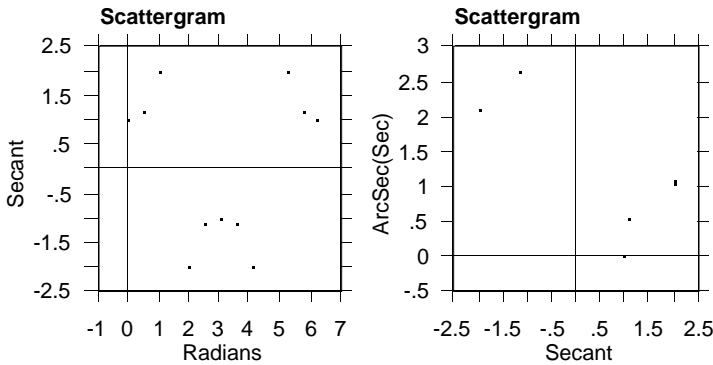
ArcSec(?)

`ArcSec(var)` returns the arcsecant in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

`ArcSec(Secant)`

Radians π	Radians	Secant	ArcSec(Sec)
zero	0.000	1.000	0.000
$\pi/6$.524	1.155	.524
$\pi/3$	1.047	2.000	1.047
$\pi/2$	1.571	•	•
$2\pi/3$	2.094	-2.000	2.094
$5\pi/6$	2.618	-1.155	2.618
π	3.142	-1.000	•
$7\pi/6$	3.665	-1.155	2.618
$4\pi/3$	4.189	-2.000	2.094
$3\pi/2$	4.712	•	•
$5\pi/3$	5.236	2.000	1.047
$11\pi/6$	5.760	1.155	.524
2π	6.283	1.000	0.000

Arcsecant is often denoted by $\sec^{-1}x$, because it is the inverse function of the secant function. The arcsecant of x is any angle whose secant is x . Since secant is a periodic function, many angles have any given secant, so arcsecant is usually understood to mean the “principal value” for a given secant, which is by convention the angle falling between 0 and π having that secant. A graph of arcsecant against secant shows this relationship.



ArcSec returns angles expressed in radians. You can convert radians to degrees with [RadTo-Deg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

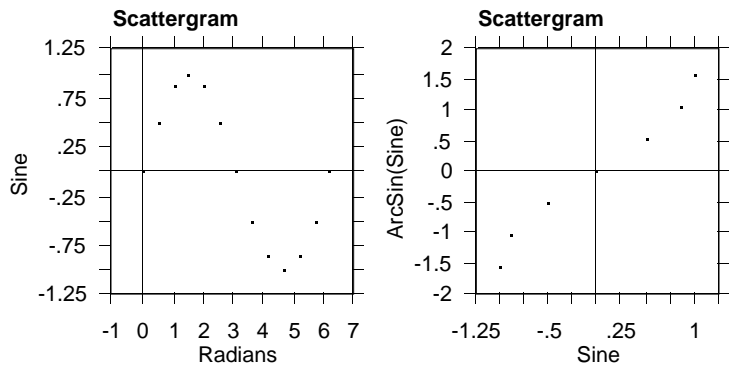
ArcSin(?)

ArcSin(*var*) returns the arcsine in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcSin(Sine)

Radians π	Radians	Sine	ArcSin(Sine)
zero	0.000	0.000	0.000
$\pi/6$.524	.500	.524
$\pi/3$	1.047	.866	1.047
$\pi/2$	1.571	1.000	1.571
$2\pi/3$	2.094	.866	1.047
$5\pi/6$	2.618	.500	.524
π	3.142	0.000	0.000
$7\pi/6$	3.665	-.500	-.524
$4\pi/3$	4.189	-.866	-1.047
$3\pi/2$	4.712	-1.000	-1.571
$5\pi/3$	5.236	-.866	-1.047
$11\pi/6$	5.760	-.500	-.524
2π	6.283	0.000	0.000

Arcsine is often denoted by $\sin^{-1} x$, because it is the inverse function of the sine function. The arcsine of x is any angle whose sine is x . Since sine is a periodic function, many angles have any given sine, so arcsine is usually understood to mean the “principal value” for a given sine, which is by convention the angle falling between $-\pi/2$ and $+\pi/2$ having that sine. A graph of arcsine against sine shows this relationship.



ArcSin returns angles expressed in radians. You can convert radians to degrees with [RadTo-Deg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

ArcSinh(?)

ArcSinh(*var*) returns the hyperbolic arcsine of a variable or constant. Missing values propagate missing values. The function works casewise.

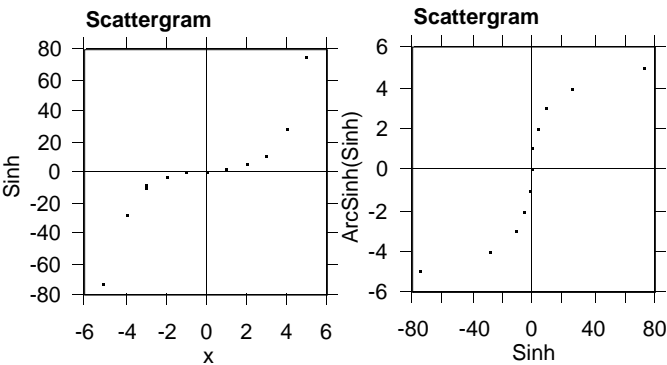
ArcSinh(Sinh(x))

x	Sinh	ArcSinh(Sinh)
-5	-74.203	-5
-4	-27.290	-4
-3	-10.018	-3
-2	-3.627	-2
-1	-1.175	-1
0	0.000	0
1	1.175	1
2	3.627	2
3	10.018	3
4	27.290	4
5	74.203	5

Hyperbolic arcsine is often denoted by $\sinh^{-1} x$, because it is the inverse function of the sinh function. More precisely, if $\sinh x = y$, then $\sinh^{-1} y = x$. As hyperbolic functions are constructed from exponential functions and exponents inverse to logs, inverse hyperbolic sine has a logarithmic expression:

$$\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1})$$

Graphs of sinh and arcsinh echo their exponential and logarithmic meanings:



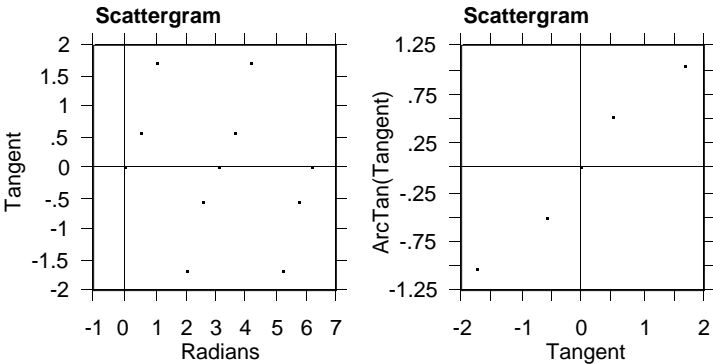
ArcTan(?)

$\text{ArcTan}(\text{var})$ returns the arctangent in radians of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcTan(Tangent)

Radians π	Radians	Tangent	ArcTan(Tangent)
zero	0.000	0.000	0.000
$\pi/6$.524	.577	.524
$\pi/3$	1.047	1.732	1.047
$\pi/2$	1.571	•	•
$2\pi/3$	2.094	-1.732	-1.047
$5\pi/6$	2.618	-.577	-.524
π	3.142	0.000	0.000
$7\pi/6$	3.665	.577	.524
$4\pi/3$	4.189	1.732	1.047
$3\pi/2$	4.712	•	•
$5\pi/3$	5.236	-1.732	-1.047
$11\pi/6$	5.760	-.577	-.524
2π	6.283	0.000	0.000

Arctangent is often denoted by $\tan^{-1}x$, because it is the inverse function of the tangent function. The arctangent of x is any angle whose tangent is x . Since tangent is a periodic function, many angles have any given tangent, so arctangent is usually understood to mean the “principal value” for a given tangent, which is by convention the angle falling between $-\pi/2$ and $\pi/2$ having that tangent. A graph of arctangent against tangent shows this relationship.



ArcTan returns angles expressed in radians. You can convert radians to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

ArcTanh(?)

ArcTanh(*var*) returns the hyperbolic arctangent of a variable or constant. Missing values propagate missing values. The function works casewise.

ArcTanh(Tanh(x))

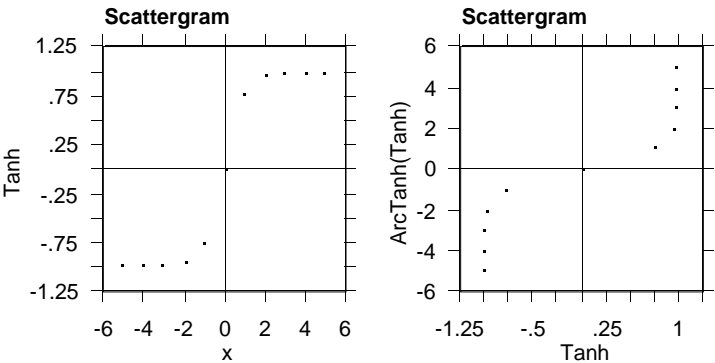
x	Tanh	ArcTanh(Tanh)
-5	-1.000	-5
-4	-.999	-4
-3	-.995	-3
-2	-.964	-2
-1	-.762	-1
0	0.000	0
1	.762	1
2	.964	2
3	.995	3
4	.999	4
5	1.000	5

Hyperbolic arctangent is often denoted by $\tanh^{-1}x$, because it is the inverse function of the tanh function. If $\tanh x = y$, then $\tanh^{-1}y = x$. As hyperbolic functions are constructed from exponential functions and exponents inverse to logs, inverse hyperbolic functions have a logarithmic expression:

$$\tanh^{-1}x = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right)$$

where $|x| < 1$.

Graphs of tanh and arctanh echo their exponential and logarithmic meanings:



Average(?, ...)

Average(*var*, *var2*, ...) computes the casewise average of the values in the variables you specify. Missing values propagate missing values. The function works casewise.

Average(Quiz, Homework, Test)

Student	Quiz	Homework	Test	AveScore
Pebbles	67	74	93	78
BamBam	•	72	93	•
Wilma	67	74	98	80
Fred	70	77	95	81
Betty	83	82	93	86
Barney	80	92	98	90

Average is the sum of all values, divided by the number of values. The average, like the mean, is a measure of central tendency of a set of observations. If you record students' grades for a series of assignments, tests, and quizzes in a series of variables (columns), you might use the Average function to compute the average score for each student for the school term.

Average also accepts constants as arguments. If you specify only constants, the result is a variable filled with a single answer, e.g., Average(3,1,8) produces a column of 4s. You might also specify several variables and one number. For instance, you might average the students' scores with a single "high" score as a way of adding grace points to their final scores:

Average(Quiz, Homework, Test, 95)

Average is a casewise (horizontal) function. If you want a columnwise average, use [Mean\(?, AllRows\)](#) [p. 388]. If one or more of the variables has missing values and you want an average computed for every case using as many values as are present, use [AverageIgnoreMissing\(?, ...\)](#) [p. 357].

AverageIgnoreMissing(?, ...)

AverageIgnoreMissing(*var*, *var2*, ...) computes the casewise average of the nonmissing values in the variables you specify. That is, for each row, AverageIgnoreMissing adds the nonmissing values of the variables (or constants) you specify, and then divides by the number of values that were nonmissing. If every variable being averaged is missing, a missing value results. By contrast, the Average function returns a missing value whenever any value on that case was missing. The function works casewise.

AverageIgnoreMissing(Quiz, Homework, Test)

Student	Quiz	Homework	Test	AveScore
Pebbles	67	74	93	78
BamBam	•	72	93	82
Wilma	67	74	98	80
Fred	70	77	95	81
Betty	83	82	93	86
Barney	80	92	98	90

Compare the AveScore results obtained here using AverageIgnoreMissing with those in the Average example, above. Since BamBam had no score for the quiz (perhaps he was out sick that day), he got no final grade. Here, he gets a final grade computed from just his homework and test scores.

[Average\(?, ...\)](#) [p. 356] and AverageIgnoreMissing are casewise (horizontal) functions. If you want columnwise averages, use [Mean\(?, AllRows\)](#) [p. 388].

BinomialCoeffs

BinomialCoeffs generates the series of binomial coefficients of order $n-1$, where n is the number of rows in the dataset. BinomialCoeffs takes no argument. The function works column-wise; results differ from row to row.

BinomialCoeffs

Binom Coeffs
1
6
15
20
15
6
1

Binomial coefficients of order n are the coefficients of terms of x in the polynomial expansion of $(1+x)$ to the n th power. Above we see the ten binomial coefficients of order 9. Specifically, the binomial coefficients are the results of the combinatorics in the following expansion:

$$(1+x)^n = \binom{n}{0}x^0 + \binom{n}{1}x^1 + \binom{n}{2}x^2 + \dots + \binom{n}{n}x^n$$

The binomial coefficients of order n are the numbers in the $(n+1)$ th row of Pascal's triangle:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
... etc. ...

You can compute specific coefficients with the Combinations function. Don't confuse binomial coefficients with the binomial *distribution*, which is featured in [ProbBinomial\(?, ?, ?\)](#) [p. 401] and [RandomBinomial\(?, ?\)](#) [p. 406].

BoxCox(?, ?)

BoxCox(*var*, *y*) computes the Box-Cox transformation of order y of the variable you specify. The first argument must be a variable, and the second argument must be a constant. Missing values propagate missing values. BoxCox is a casewise transformation.

BoxCox(Cholesterol, 2)

Cholesterol	BoxCox order 2
197	19404.0
181	16380.0
190	18049.5
131	8580.0
172	14791.5
222	27144.0

The Box Cox transformation can be used to make certain nonlinear models linear. The value of the transformed variable is defined on each case as

$$\frac{(x^y - 1)}{y}, \text{ where } y \neq 0$$
$$\ln x, \text{ where } y = 0$$

In the first row above, the order 2 Box-Cox transformation of cholesterol is calculated by

$$\frac{197^2 - 1}{2} = \frac{38809 - 1}{2} = \frac{38808}{2} = 19404$$

Ceil(?)

Ceil(*var*) rounds values of the variable you specify to the next greater integer. Missing values propagate missing values. The function works casewise.

Ceil(A)

A	RoundedA	TruncatedA	Floor of A	Ceil of A
-1.200	-1.000	-1.000	-2.000	-1.000
-3.915	-4.000	-3.000	-4.000	-3.000
.
.051	0.000	0.000	0.000	1.000
1.238	1.000	1.000	1.000	2.000
4.800	5.000	4.000	4.000	5.000

The ceiling of any number is the next greater integer, regardless of the size of its fractional part and regardless of sign. Thus, the ceiling of -1.2 is -1 , even though 0.2 is less than one-half, and even though the ceiling of $+1.2$ is 2 . Remember, for negative numbers, “greater” and “lesser” can seem backwards: -1 is greater than -2 . As do all computations, Ceil works with actual stored values rather than the way values are displayed. For example, the value -1.9 is displayed in a format with no decimal places as -2 , but its ceiling is -1 .

Related functions are [Round\(?\)](#) [p. 416], [Trunc\(?\)](#) [p. 429], and [Floor\(?\)](#) [p. 380]; a detailed comparison of Round, Trunc, Floor, and Ceil is made in the discussion of Round.

ChooseArg(?)

ChooseArg(*var*, *value1*, *value2*, *value3*, ...) uses values in the index *var* to choose from the argument *values* you specify. The *values* you specify needn't be unique. The *values* may be variable names, in which case that variable's value on a row is used as the new variable's value. Text *values* must be enclosed in quotation marks. (Variable names containing spaces must also be enclosed in quotation marks.) The function works casewise.

ChooseArg uses the values of the variable you specify as an index to the values you list. ChooseArg's behavior varies according to variable type:

If the index variable is categorical and a row has the *n*th category name, that row in the new variable has the *n*th item in your replacement list. If the index *var* has more values than the number of replacement values you list, missing values result.

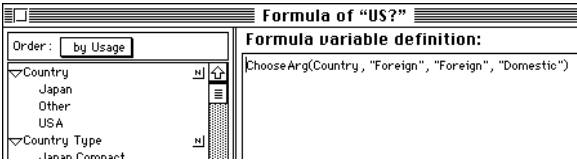
If the index variable is numeric (real, integer, long integer, currency, or date/time) each row's value is rounded to the nearest integer *n*, and the new variable has the *n*th item in your replacement list. (Date/time values are converted to real numbers of seconds; see the discussion of date/time functions for details.) If the index *var* has negative, zero, or missing values, or values that exceed the number of replacement values, missing values result.

If the index variable is string, missing values result.

ChooseArg(Country, "Foreign", "Foreign", "Domestic")

Country	US?
Japan	Foreign
Japan	Foreign
Other	Foreign
Other	Foreign
Other	Foreign
Other	Foreign
Other	Foreign
Other	Foreign
USA	Domestic
USA	Domestic

Above, we combine a three-level categorical variable to two levels. We join Japan (the first category level) and Other (the second category level) in a single level by specifying “Foreign” as the new value for both. Notice that the variable browser in the Formula window shows the ordered values of categorical variables. If you click the triangle to the left of the variable's name, you can easily determine what sequence your new, replacement values should take.



If your variable is numeric and has fractional or negative values or a wide range, you may prefer to use as indices ranks of the values rather than the values themselves:

ChooseArg(Rank(var, AllRows), value, value2, value3, ...)

Don't forget to change the formula variable to a type appropriate to the replacement values.

CoeffOfVariation(?, AllRows)

CoeffOfVariation(*var*, AllRows) computes the coefficient of variation of the variable you specify; by default, calculations are based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

CoeffOfVariation(A, AllRows)

	A	CV of A
Variance:	17.819	0.000
Coeff. of Variation:	-17.819	0.000
Minimum:	-4.000	-17.819
1	-4.000	-17.819
2	-3.000	-17.819
3	•	-17.819
4	0.000	-17.819
5	1.000	-17.819
6	5.000	-17.819

Coefficient of variation is a measure of relative variability. It is standard deviation divided by mean. If the mean is near zero, the quotient may tend to be large, even if the variation is not. Coefficient of variation is also shown in the summary pane.

Related functions are [Mean\(?, AllRows\)](#) [p. 388] and [StandardDeviation\(?, AllRows\)](#) [p. 420].

Combinations(?, ?)

Combinations(*n*, *r*) computes the casewise unordered combinations of *n* objects taken *r* at a time, where *n* and *r* can be variables or constants. Cases with *r* greater than *n*, negative values, or missing values are missing. The function works casewise.

Combinations(*n*, *r*)

Permutations(*n*, *r*)

n	r	Comb(n,r)	Perm(n,r)
-2	1	•	•
3	•	•	•
4	3	4	24
5	1	5	5
5	2	10	20
5	3	10	60
5	6	•	•

Combinations(*n*, *r*) computes the number of *r*-object *unordered* subsets that can be taken from *n* objects, or

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

For example, Combination(5,3) on the second to last row is 10, which means that you could choose 10 distinct committees of 3 people from a group of 5 people:

$$\binom{5}{3} = \frac{5!}{3!2!} = \frac{120}{6 \times 2} = 10$$

For *ordered* combinations, see [Permutations\(?, ?\)](#) [p. 399]. Both Combinations and Permutations rely on the use of factorials (such as *n*!), which can be computed individually with [Factorial\(?\)](#) [p. 377]; factorials are defined in that entry.

Concat(?)

Concat(*text*, *text2*, ...) concatenates or combines the text strings you specify. Unless the results are numeric, you should change the resulting variable to have type string. The arguments *text*,

text2, etc., may be variables or constants; constants must be enclosed in quotation marks. (Variable names containing spaces must also be enclosed in quotation marks.) If you supply a variable as argument, Concat uses its exact values in the current format's display. Changing formats can change results. The function works casewise.

Concat(Country, " ", Type)

Country	Type	Country Type
Japan	Small	Japan Small
Japan	Medium	Japan Medium
Other	Medium	Other Medium
Other	Compact	Other Compa...
Other	Compact	Other Compa...
Other	Compact	Other Compa...

Concat can be used as an alternative to [Groups\(?, ...\) \[p. 381\]](#) for merging two category variables into one. (Do not forget to change the type of the new variable to category.) Usually Concat is used with string variables, but other variable types also work.

Correlation(?, ?, AllRows)

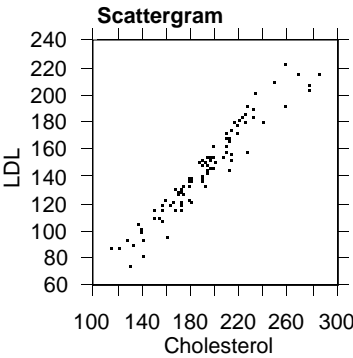
Correlation(*var*, *var2*, AllRows) computes the Pearson correlation coefficient of the two variables you specify; by default, it uses AllRows of the variables, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the third argument. Cases with missing values on either variable are excluded from calculations. The function works columnwise and produces the same result for every row.

Correlation("Chol-3yrs", "LDL-3yrs", AllRows)

Chol-3yrs	LDL-3yrs	CorrChol-LDL
182	145.8	.965
151	102.1	.965
169	130.6	.965
133	64.8	.965
166	123.8	.965
228	188.7	.965

Pearson correlation measures the degree of linear relationship between two variables. A positive correlation means that as one variable increases, so does the other. A negative correlation means that as one increases, the other decreases, and *vice versa*. Correlations range from zero (no consistent relationship) to one (absolutely consistent relationship). However, correlation coefficients are meaningless unless you verify (with a scattergram, perhaps) that the relationship is linear and the data are bivariate normal (that is, the points fall roughly in an ellipse).

The high positive correlation (0.965) of cholesterol and low density lipoproteins (LDL) in Lipid Data shows that cholesterol and LDL increase and decrease together, and a scattergram confirms that the relationship is linear. However, the point cloud is not particularly elliptical.



You may use the Lag function to lag a variable and then correlate the variable against its lag to test autocorrelation (the degree to which values depend on preceding values, as might be the case with time series data).

A related function is [Covariance\(?, ?, AllRows\)](#) [p. 365]. The correlation and covariance analyses (see [“Correlation and Covariance,”](#) p. 43) provide additional statistics useful for assessing linear relationships. Also see the Spearman and Kendall rank order correlation analyses in [“Nonparametrics,”](#) p. 119.

Cos(?)

Cos(*var*) returns the cosine of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works case-wise.

Cos(Radians)

Radians π	Radians	Cosine
zero	0.000	1.000
$\pi/6$.524	.866
$\pi/3$	1.047	.500
$\pi/2$	1.571	0.000
$2\pi/3$	2.094	-.500
$5\pi/6$	2.618	-.866
π	3.142	-1.000
$7\pi/6$	3.665	-.866
$4\pi/3$	4.189	-.500
$3\pi/2$	4.712	0.000
$5\pi/3$	5.236	.500
$11\pi/6$	5.760	.866
2π	6.283	1.000

Sines, cosines, and tangents relate angles to the coordinates of points in planes. The cosine of an angle in a right triangle is the ratio of the length of the leg adjacent to the angle to the length of the hypotenuse.

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

Cosh(?)

Cosh(*var*) returns the hyperbolic cosine of a variable or constant. Missing values propagate missing values. The function works casewise.

Cosh(x)

x	Cosh
-5	74.210
-4	27.308
-3	10.068
-2	3.762
-1	1.543
0	1.000
1	1.543
2	3.762
3	10.068
4	27.308
5	74.210

The hyperbolic trigonometric functions (sinh, cosh, and tanh, sometimes pronounced “sinch, cosh, and tanch”) are analogous to the trigonometric functions sine, cosine, and tangent. They are constructed from the functions e^x and e^{-x} and bear a relationship to the unit hyperbola that is analogous to the trigonometric functions’ relationship to the unit circle.

The hyperbolic cosine is defined by

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

and like cosine, cosh(x) has value 1 at $x = 0$. Cosh is defined for all real numbers and ranges from 1 to infinity.

Cot(?)

Cot(*var*) returns the cotangent of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works casewise.

Cot(Radians)

Radians π	Radians	Cotangent
zero	0.000	•
$\pi/6$.524	1.732
$\pi/3$	1.047	.577
$\pi/2$	1.571	0.000
$2\pi/3$	2.094	-.577
$5\pi/6$	2.618	-1.732
π	3.142	•
$7\pi/6$	3.665	1.732
$4\pi/3$	4.189	.577
$3\pi/2$	4.712	0.000
$5\pi/3$	5.236	-.577
$11\pi/6$	5.760	-1.732
2π	6.283	•

The cotangent of an angle in a right triangle is the ratio of the length of the leg adjacent to the angle to the length of the leg opposite. Recall that the tangent of an angle in a right triangle is

the ratio of the length of the leg opposite the angle to the length of the leg adjacent. Therefore, the cotangent is the reciprocal of the tangent:

$$\cot x = \frac{1}{\tan x}$$

Recall that tangents approach plus or minus infinity asymptotically as their arguments approach values $\pi/2$, $3\pi/2$, etc., so cotangents converge to zero at these points and approach minus infinity as angles approach π , 2π , etc. Cotangents at these points are undefined, so Cot produces missing values. (On some platforms, differences in the numerics environments may produce extreme values rather than missing values.)

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

Count(?, AllRows)

Count(*var*, AllRows) computes the number of nonmissing values (often represented as *n* in formulas for statistics) in a variable. By default, Count uses AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. The function works columnwise and produces the same result for every row.

Count(A, AllRows)

	A	CountOfA
range:	2	6
Count:	4	6
Missing Cells:	2	0
1	-4	4
2	-3	4
3	•	4
4	•	4
5	1	4
6	5	4

Count is the number of cases in a variable minus the number of cases that have missing values (.). Counts are also shown in the summary pane. Most statistics are computed from formulas that involve the count of cases in the variable(s) being analyzed. For instance, mean is defined as the sum of the nonmissing cases divided by the count.

The Count function is useful for computing your own statistics. See examples shown in the discussions of Percentile, StandardDeviation, and Variance for some ideas.

[NumberOfRows](#) [p. 394] gives the number of cases in a variable, whether missing or non-missing. [NumberMissing\(?, AllRows\)](#) [p. 393] gives the number of missing values. Count(*var*, AllRows) and NumberMissing(*var*, AllRows) sum to NumberOfRows.

Covariance(?, ?, AllRows)

Covariance(*var*, *var2*, AllRows) computes the covariance coefficient of the two variables you specify; by default, it uses AllRows of the variables, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the third argument. Cases with missing values on either

variable are excluded from calculations. The function works columnwise and produces the same result for every row.

Covariance("Chol-3yrs", "LDL-3yrs", AllRows)

Chol-3yrs	LDL-3yrs	CovarChol-LDL
182	145.8	1339.114
151	102.1	1339.114
169	130.6	1339.114
133	64.8	1339.114
166	123.8	1339.114

Covariance is a measure of joint variance that, like correlation, measures the degree of relationship between two variables. A positive covariance means as one variable increases, the other also increases; a negative covariance means as one increases, the other decreases. Covariance of the variables X and Y is given by the formula

$$\frac{\sum_{i=1}^n (x_i - \mu_X)(y_i - \mu_Y)}{n - 1}$$

where x_i and y_i are values on each case, n is the count, and μ_X and μ_Y are sample means.

Related functions are [Variance\(?, AllRows\)](#) [p. 430] and [Correlation\(?, ?, AllRows\)](#) [p. 362]. Also, the correlation and covariance analyses provide additional statistics useful for assessing linear relationships.

Csc(?)

Csc(*var*) returns the cosecant of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works case-wise.

Csc(Radians)

Radians π	Radians	Cosecant
zero	0.000	•
$\pi/6$.524	2.000
$\pi/3$	1.047	1.155
$\pi/2$	1.571	1.000
$2\pi/3$	2.094	1.155
$5\pi/6$	2.618	2.000
π	3.142	•
$7\pi/6$	3.665	-2.000
$4\pi/3$	4.189	-1.155
$3\pi/2$	4.712	-1.000
$5\pi/3$	5.236	-1.155
$11\pi/6$	5.760	-2.000
2π	6.283	•

The cosecant of an angle in a right triangle is the ratio of the length of the hypotenuse to the length of the leg opposite the angle. Recall that the sine of an angle in a right triangle is the ratio of the length of the leg opposite the angle to the length of the hypotenuse. Therefore, the cosecant is the reciprocal of the sine:

$$\csc x = \frac{1}{\sin x}$$

As the angle (Radians) approaches π and 2π (and so on), sine approaches zero, and thus cosecant approaches plus or minus infinity. Cosecant is undefined at these points, so Csc produces missing values. (On some platforms, differences in the numerics environments may produce extreme values rather than missing values.)

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

CubicSeries(1, 0, 0, 1)

`CubicSeries(a, b, c, d)` generates a series of values equal to $a + bx + cx^2 + dx^3$, where x is one less than the row number. By default, the arguments are 1, 0, 0, 1, but you may specify any constants. The function works columnwise; results differ from row to row.

`CubicSeries(1, 0, 0, 1)`

	Cubic
1	1
2	2
3	9
4	28
5	65
6	126
7	217
8	344
9	513
10	730

In each row i of the new variable, the quantity $a + bx + cx^2 + dx^3$ is evaluated for $x = i - 1$, and that result is the value for the row. For example, the fourth row above is computed by:

$$1 + 0 \times 3 + 0 \times 3^2 + 1 \times 3^3 = 1 + 0 + 0 + 27 = 28$$

See also [QuadraticSeries\(1, 0, 1\)](#) [p. 404] and [QuarticSeries\(1, 0, 0, 0, 1\)](#) [p. 405].

CumProduct(?)

`CumProduct(var)` computes the cumulative product of all nonmissing values in a variable. On each row in the new variable is the “product in progress,” and the final row shows the cumulative product based on all rows. Missing values are marked with missing values in the new variable, but they do not affect the cumulative product. The function works columnwise; results differ from row to row.

`CumProduct(A)`

A	CumProd of A
-1	-1
-2	2
•	•
3	6
2	12
3	36
4	144
5	720
-6	-4320
7	-30240

For the first case, the cumulative product is the first value, -1. Since $-1*(-2)=2$, the cumulative product on the second case is 2. The third case is missing, and the cumulative product is marked by missing, but for the fourth case we resume where we left off: $2*3$ is 6. On the final row, we see the final result.

For casewise multiplication, use the asterisk (*), e.g., $A*B$; see [? * ? or ? ? ? \[p. 333\]](#).

CumSum(?)

CumSum(*var*) computes cumulative sums of all nonmissing values in a variable. Each row in the new variable shows the “sum in progress,” and the final row shows the cumulative sum of all rows. Missing values are marked with missing values in the new variable, but they do not affect the cumulative sum. The function works columnwise; results differ from row to row.

CumSum(A)

A	CumSum of A
-1	-1
-2	-3
•	•
3	0
2	2
3	5
4	9
5	14
-6	8
7	15

For the first case, the cumulative sum is the first value, -1. Since $-1 + -2$ is -3, the second case is -3. The third row shows a missing value, since A is missing, but the sum-in-progress resumes in the fourth: $-3 + 3 = 0$. On the final row, we see the final cumulative sum result, so $-1 + -2 + 3 + 2 + 3 + 4 + 5 + -6 + 7$ is 15.

For casewise addition, see [? + ? \[p. 333\]](#), [Sum\(?, ...\) \[p. 423\]](#), or [SumIgnoreMissing\(?, ...\) \[p. 424\]](#). Sum adds values for each row on all the variables you specify. SumIgnoreMissing is the same except that missing values are ignored. Also see [SumOfColumn\(?, AllRows\) \[p. 424\]](#), which fills a new variable with a single sum.

CumSumSquares(?)

CumSumSquares(*var*) computes cumulative sums of squares of all nonmissing values in a variable. On each row in the new variable is the sum of squares “in progress,” and the final row shows the cumulative sum of squares on all rows. Missing values are marked with missing

values in the new variable, but they do not affect the sum in progress. The function works columnwise and produces the same result for every row.

CumSumSquares(A)

SumOfSquares(A, AllRows)

	A	CumSSq	SSq
Missing values:			
Sum:	15	433	1530
Sum of Squares:	153	41973	234090
1	-1	1	153
2	-2	5	153
3	•	•	153
4	3	14	153
5	2	18	153
6	3	27	153
7	4	43	153
8	5	68	153
9	-6	104	153
10	7	153	153

In the example above, the first row is simply $(-1)^2$ or 1. The second row is $(-1)^2 + (-2)^2 = 5$, the third is missing, the fourth is $(-1)^2 + (-2)^2 + 3^2 = 14$, etc. The last row of CumSSq, 153, is the final sum of squares. That final result is the Sum of Squares shown in the summary pane, and it fills all rows of the variable SSq, created with SumOfSquares; see [SumOfSquares\(?, AllRows\)](#) [p. 425].

Date(?, ?, ?)

Date(*year*, *month*, *day*) returns the exact second at midnight of the year, month, and day you specify. Date is casewise. Notice that *year* comes first, then *month*, then *day*. The function works casewise.

Date(Yr, Mo, Dy)

Yr	Mo	Dy	Nice dates
1994	1	24	24 Jan 1994
1992	4	30	30 Apr 1992
1984	6	2	2 Jun 1984
1970	2	7	7 Feb 1970

The example above shows how to use Date to combine year, month, and date values stored in separate columns. Data imported from other programs may have date/time values separated into several numeric-type columns, and Date puts those columns together into date/time values. (Don't forget to change the type of the new variable to date/time and choose a format you like.)

Other programs store date values as text strings. You can use the Substring (p. 422) and Date (p. 369) functions to convert these to dates:

Date(1900+Substring(Text, 5, 2), Substring(Text, 1, 2), Substring(Text, 3, 2))

Text	Dates
012494	Monday, 24 January 1994
043092	Thursday, 30 April 1992
060284	Saturday, 2 June 1984
020770	Saturday, 7 February 1970

Or, suppose you need to group date values together by month (see [Year\(?\)](#) [p. 431] and [Month\(?\)](#) [p. 391]):

Date(Year(mydates), Month(mydates), I)

You may also build dates with formulas such as this one (see [RowNumber](#) [p. 417]):

Date(1970+RowNumber, RowNumber, RowNumber)

	Some dates
21	09.21.92
22	10.22.93
23	11.23.94
24	12.24.95
25	01.25.97
26	02.26.98
27	03.27.99

This example shows how invalid dates are reinterpreted. Consider the computations for row 25, where month=25 carries a 2 into the years value:

year=1970+25, month=25, day=25
year=1970+25+2, month=1, day=25
year=1997, month=1, day=25

Carrying happens whenever the number of days is greater than the length of the current month (28, 29, 30, or 31), or whenever the number of months is greater than 12.

DateDifference(?, ?, ?)

DateDifference(*date1*, *date2*, *units*) subtracts *date1* from *date2*, in the time *units* specified (1=years, 2=months, 3=weeks, 4=days, 5=hours, 6=minutes, 7=seconds). The first two arguments may be date/time variables or date values enclosed in quotation marks and the third argument must be a number 1, 2, 3, 4, 5, 6, or 7. The resulting variable has values that are a real number of the unit you specify in the third argument. The function works casewise.

For example, suppose we want to know how long the Berlin Wall divided East and West Berlin. We could do this a number of ways. First, we could enter variables for the day construction of the wall was completed, 17 Aug 1961, and for the day the Brandenburg Gate reopened in Berlin on 22 December 1989. Then, we could use the formula to make a third variable, yrs Berlin divided:

DateDifference("Gate opens", "Berlin Wall completed", I)

	Berlin Wall completed	Gate opens	yrs Berlin divided
Type:	Date/Time	Date/Time	Real
Source:	User Entered	User Ent...	Dynamic Formula
Class:	Continuous	Continuous	Continuous
Format:	01.01.04 12:00:00 AM	01.01.04	Free Format Fixed
Dec. Places:	•	•	3
1	08.17.61 12:00:00 AM	12.22.89	28.348

Notice several things. The first two variables have type date/time and two different formats. The third variable counts a number of years and is not date/time but real.

We could instead supply both date values as arguments directly:

DateDifference("Dec 22, 1989", "08/17/61", 1)

Many ways of typing a date are valid, but always enclose dates in quotation marks. This time, we list the earlier date first (for a negative difference) and set the third argument to 2 for months:

DateDifference("08/17/61", "Dec 22, 1989", 2)

	Difference
1	-340.172

Dynamic formulas fill only as many rows as already exist in the dataset, so we must insert a row: Control-click (Windows) or Command-click (Macintosh) the border between the variable-name row and the empty data area. The answer found in the Difference variable, -340 months, is negative because our formula specifies the earlier date first.

Of course, you can also compute the differences between entire columns of dates:

DateDifference("Some dates", "Other Dates", 1)

DateDifference("Some dates", "Other Dates", 3)

DateDifference("Some dates", "Other Dates", 6)

	Some dates	Other dates	Diff yrs	Diff days	Diff mins
18	06.10.97	06.10.97	0	0	0
19	07.19.90	07.19.95	-5	-261	-2629440
20	08.20.91	08.20.95	-4	-209	-2103840
21	09.21.92	09.21.95	-3	-156	-1576800
22	10.22.93	10.22.95	-2	-104	-1051200
23	11.23.94	11.23.95	-1	-52	-525600
24	12.24.95	12.24.94	1	52	525600
25	01.25.97	01.25.95	2	104	1052640
26	02.26.98	02.26.95	3	157	1578240
27	03.27.99	03.27.95	4	209	2103840
28	04.28.00	03.21.95	5	266	2685600

Caution: StatView assumes a fixed month-length of the number of seconds in a year divided by twelve, but months actually have differing lengths. Therefore, DateDifference results in months (third argument 2) can be misleading.

Day(?)

Day(*date*) returns the day number (1–31) of the *date* specified. The *date* argument may be a variable or constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The function works casewise.

Day("Other dates")

Other dates	Day
01.01.95	1
02.02.95	2
03.03.95	3
04.04.95	4
05.05.95	5
06.06.95	6

DayOfWeek(?)

DayOfWeek(*date*) returns an index indicating the day of the week (1=Sunday, 2=Monday, etc.) of the *date* specified. The *date* argument may be a variable or constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The function works casewise.

DayOfWeek("Other dates")

Other dates	Weekday
01.01.95	1
02.02.95	5
03.03.95	6
04.04.95	3
05.05.95	6

If you want day names, change the variable to category, and edit the category to have levels Sunday, Monday, Tuesday, ..., Saturday.

Weekday
Sunday
Thursday
Friday
Tuesday
Friday
Tuesday

[Weekday\(?\) \[p. 430\]](#) is synonymous.

DayOfYear(?)

DayOfYear(*date*) returns the number of the day of the year (1–366) of the *date* specified. The *date* argument may be a variable or constant. For example, 62 means that 3 March is the sixty-second day of a non-leap year. The function works casewise.

DayOfYear("Other dates")

Weekday
Sunday
Thursday
Friday
Tuesday
Friday
Tuesday

DegToRad(?)

DegToRad(*var*) converts angle measurements in a variable (or a constant) from degrees to radians. Missing values propagate missing values. The function works casewise.

DegToRad(Degrees)

Radians π	Degrees	Radians
zero	0	0.000
$\pi/6$	30	.524
$\pi/3$	60	1.047
$\pi/2$	90	1.571
$2\pi/3$	120	2.094
$5\pi/6$	150	2.618
π	180	3.142
$7\pi/6$	210	3.665
$4\pi/3$	240	4.189
$3\pi/2$	270	4.712
$5\pi/3$	300	5.236
$11\pi/6$	330	5.760
2π	360	6.283

StatView's trigonometric functions work with measurements in radians, so DegToRad conversions are necessary if you ordinarily work with data measured in degrees. A circle has 360 degrees or 2π radians ($2 \times 3.1416... = 6.2832...$ radians). Above, Radians π is an informative variable entered by hand to make the Radians values easier to read.

To convert radians back to degrees, use [RadToDeg\(?\)](#) [p. 405]. StatView's trigonometric functions are [Sin\(?\)](#) [p. 419], [Cos\(?\)](#) [p. 363], [Tan\(?\)](#) [p. 426], [Sec\(?\)](#) [p. 418], [Csc\(?\)](#) [p. 366], [Cot\(?\)](#) [p. 364], [ArcSin\(?\)](#) [p. 353], [ArcCos\(?\)](#) [p. 349], [ArcTan\(?\)](#) [p. 355], [ArcSec\(?\)](#) [p. 352], [ArcCsc\(?\)](#) [p. 351], and [ArcCot\(?\)](#) [p. 350].

Difference(?, 1, 1)

Difference(*var*, *n*, *j*) computes the difference between a variable *var* and its lag at *n* places, repeating that operation *j* times over. The argument *n* may be any integer, negative or positive; *j* must be a positive integer. The function works columnwise; results differ from row to row.

Difference(A, 1, 1)

Difference(A, 1, 2)

Difference(A, 2, 1)

Difference(A, 2, 2)

Difference(A, -1, 1)

A	Lag(A,1)	Lag(A,2)	Diff(A,1,1)	Diff(A,1,2)	Diff(A,2,1)	Diff(A,2,2)	Diff(A,-1,1)
1	•	•	•	•	•	•	-1
2	1	•	1	•	•	•	-2
4	2	1	2	1	3	•	-3
7	4	2	3	1	5	•	-4
11	7	4	4	1	7	4	-5
16	11	7	5	1	9	4	-6
22	16	11	6	1	11	4	-7
29	22	16	7	1	13	4	-8
37	29	22	8	1	15	4	-9
46	37	29	9	1	17	4	•

Differencing is useful for removing additive trends from time series data. It is best defined by example. Consider the series A and its first and second lags. Difference(A,1,1) differences A by one cell, one time. This amounts to the casewise subtraction $A - \text{Lag}(A,1)$: 1–, is missing in the first row, 2–1 is 1 in the second, 4–2 is 2, etc.

Similarly, Difference(A,1,2) differences A by one cell, two times; in other words, it differences A by one cell, and then differences it by one cell again. This is the same as $A - 2 * \text{Lag}(A,1) +$

Lag(A,2). Or to put it another way: if B is Diff(A, 1, 1), then Diff(A, 1, 2) is Diff(B, 1, 1). We also show the results of Difference(A,2,1), Difference(A,2,2), and Difference(A, -1, 1); notice how a negative n argument differences “up.”

See [Lag\(?,1\) \[p. 382\]](#) if you need to do other transformations involving lagged variables. Also useful with time series data is [MovingAverage\(?, ?\) \[p. 391\]](#).

Div(?, ?)

Div(*var*, *var2*) does casewise division of *var* by *var2* and returns the integer part of the quotient. Both arguments can be variables or constants. Missing values or division by zero propagate missing values.

Div(A, B)

A	B	Div(A,B)
-12	5	-2.0
-3	-2	1.0
•	4	•
0	3	0.0
1	0	•
15	4	3.0

Div truncates a quotient to its integer part, discarding all digits after the decimal place. You could get the same result by using Trunc(A/B). For example in row 1, $-12/5$ is -2.4 , so the result is -2 .

[Mod\(?, ?\) \[p. 390\]](#) and [Remainder\(?, ?\) \[p. 413\]](#) return the remainder after dividing. Mod and Remainder are synonymous for positive arguments; for negative arguments, they differ. See the discussions of each for details.

DotProduct(?, ?)

DotProduct(*var*, *var2*) computes the dot product of the two vectors (variables) you specify. The result of any dot product is a constant, so the function returns a variable with the constant on every row. Any row with a missing value for either variable is ignored. The function works columnwise and produces the same result for every row.

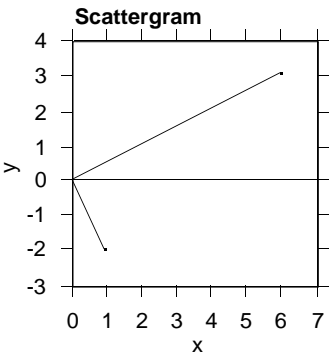
DotProduct(A, B)

A	B	A*dotB
1	6	0
-2	3	0

The dot product (also called scalar product or inner product) is a linear algebra operation that reduces a horizontal (row) vector and a vertical (column) vector to a single constant by multiplying the first numbers of each vector, the second numbers of each vector, etc., and then adding those products. One interesting property: if the dot product of two vectors is 0, then the two vectors are orthogonal, meaning that the lines connecting the points to the origin are perpendicular to each other in space.

For example, consider the vectors (1, -2) and (6,3). These points represent lines from the origin (0,0). Their dot product is $(1 \times 6) + (-2 \times 3) = 6 - 6 = 0$, so we know the vectors

are perpendicular. We could instead examine a graph: we transpose the data cells to place X coordinates in one column and Y coordinates in another, plot the points in a bivariate scattergram, and then use drawing tools to connect them to the origin.



Computing the dot product is more practical (and more precise) than scrutinizing graphs, especially with vectors in many-dimensional space.

Another use of DotProduct is shown in [“How can I estimate the survival function at other covariate values?” p. 245 of Using StatView. Norm\(? , AllRows\) \[p. 392\]](#) is also useful for linear algebra computations.

e

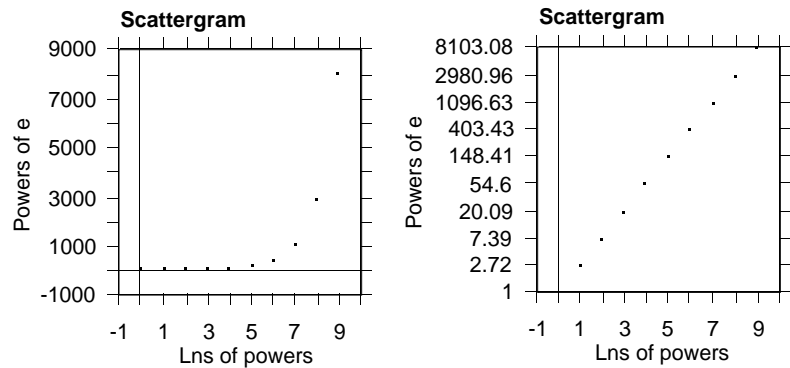
The function e produces the constant *e*, which is approximately 2.71828... Unless you use *e* in combination with other functions, it returns a variable in which every case is *e*, so we call it a constant. The function works casewise.

```
e
A + e
e^(RowNumber-1)
Ln(e^(RowNumber-1))
```

	A	e	A+e	Powers of e	Lns of powers
1	-4	2.718	-1.282	1.000	0
2	-3	2.718	-.282	2.718	1
3	•	2.718	•	7.389	2
4	0	2.718	2.718	20.086	3
5	1	2.718	3.718	54.598	4
6	5	2.718	7.718	148.413	5
7	34	2.718	36.718	403.429	6
8	-72	2.718	-69.282	1096.633	7
9	•	2.718	•	2980.958	8
10	3	2.718	5.718	8103.084	9

You can use *e* in combination with other functions as seen above.

This example illustrates the inverse relationship between *e^x* (often represented as “exp(x)”) and the natural logarithm (also known as the base *e* logarithm; see [Ln\(?\) \[p. 385\]](#)). The relationship is better seen in graphs. The second plot uses a log *e* vertical scale to “straighten” the relationship between Powers and Lns.



[ExponentialSeries\(1\)](#) [p. 376] produces a variable in which each case is *e* times the value of the previous case, beginning with the first case equal to the argument; for example, Powers of e above could be given by [ExponentialSeries\(1\)](#).

Erf(?)

[Erf\(*var*\)](#) computes the error function of *var*. The argument *var* may be either a constant or a variable. Missing values are ignored. The function works casewise.

[Erf\(RowNumber\)](#)

	Erf(RowNum.)
1	.843
2	.995
3	1.000
4	1.000
5	1.000
6	1.000
7	1.000
8	1.000
9	1.000
10	1.000

The error function is a special case of the incomplete gamma function and is related to the normal CDF. Erf(*x*) is defined as follows.

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2}$$

ExponentialSeries(1)

[ExponentialSeries\(*x*\)](#) generates a series whose values are *x* times the powers of *e*, starting with *e*⁰. By default, *x* is 1, but you may specify any constant. The function works columnwise; results differ from row to row.

[ExponentialSeries\(1\)](#)

[ExponentialSeries\(2\)](#)

	Expo-1	Expo-2
1	1.000	2.000
2	2.718	5.437
3	7.389	14.778
4	20.086	40.171
5	54.598	109.196
6	148.413	296.826
7	403.429	806.858
8	1096.633	2193.266
9	2980.958	5961.916
10	8103.084	16206.168

Row *i* of the ExponentialSeries(*x*) is equal to xe^{i-1} . Above, ExponentialSeries(1) is the powers of *e* starting in row 1 with e^0 , then in row 2 e^1 , in row 3 e^2 , etc. ExponentialSeries(2) has values $2e^0, 2e^1, 2e^2, \dots$

Also see [e \[p. 375\]](#), for the constant *e*.

Factorial(?)

Factorial(*var*) does casewise factorials of the variable you specify. If you specify a constant, Factorial returns a variable whose values are all that number factorial. Cases with negative or missing values are missing. The function works casewise.

A!
B!
7!

A	A!	B	B!	Seven!
-4	•	1	1	5040
-5	•	2	2	5040
•	•	3	6	5040
•	•	3	6	5040
1	1	6	720	5040
5	120	5	120	5040
5	120	8	40320	5040
6	720	7	5040	5040

Factorial is a basic operation used for many probability computations. It is usually represented by an exclamation point (!) and defined as follows:

$n! = n(n-1)! , \text{ where } n \geq 0, 0! = 1 , \text{ and } 1! = 1 ;$

or $n! = \prod_{i=1}^n i , \text{ where } n > 0 .$

Factorials are used to count the ordered permutations of *n* objects in which repetition is not allowed; for example, how many ways can you rearrange the letters in the word BOAT into distinct four-letter words? There are 4 possibilities for the first letter (B, O, A, or T), times 3 possibilities for the second letter (the three letters you didn't use already), times 2 possibilities for the third letter (the two letters you haven't used), times 1 possibility for the last letter (the one letter still left); hence, $4 \times 3 \times 2 \times 1 = 4! = 24 .$

Related functions are [Permutations\(?, ?\) \[p. 399\]](#) and [Combinations\(?, ?\) \[p. 361\]](#).

FibonacciSeries

FibonacciSeries generates the Fibonacci series. The number of Fibonacci values that are generated depends on the number of rows you specify if you use the Series command, or the number of rows in the dataset if you use the Formula command. This function takes no arguments. The function works columnwise; results differ from row to row.

FibonacciSeries

Fibonacci
1
1
2
3
5
8
13
21
34
55

By definition, the first two values of the Fibonacci series are 1, and each subsequent value is the sum of the previous two. Thus we have 1, 1, 1+1=2, 1+2=3, 2+3=5, 3+5=8, etc. An interesting property of the Fibonacci series is that if you difference the series (subtract from each value its previous value), then the ratio of each difference to the previous difference (which we compute by dividing the differenced series by the lag of the differenced series) approaches the golden ratio (1.6180339887...) as RowNumber ([p. 417](#)) approaches infinity:

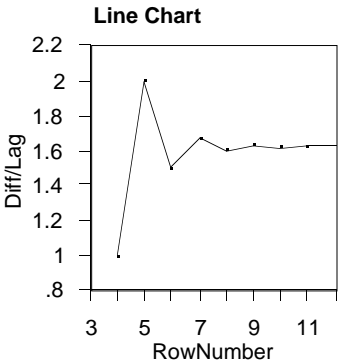
FibonacciSeries

Difference(Fib, 1, 1)

Lag(Diff, 1)

Diff/Lag

	Fib	Diff	Lag	Diff/Lag
1	1	●	●	●
2	1	0	●	●
3	2	1	0	●
4	3	1	1	1.000000000
5	5	2	1	2.000000000
6	8	3	2	1.500000000
7	13	5	3	1.666666667
8	21	8	5	1.600000000
9	34	13	8	1.625000000
10	55	21	13	1.615384615
11	89	34	21	1.619047619
12	144	55	34	1.617647059
13	233	89	55	1.618181818
14	377	144	89	1.618090909
15	610	233	144	1.618055556
16	987	377	233	1.618033989
17	1597	610	377	1.618033989
18	2584	987	610	1.618033989
19	4181	1597	987	1.618033989
20	6765	2584	1597	1.618033989
21	10946	4181	2584	1.618033989
22	17711	6765	4181	1.618033989
23	28657	10946	6765	1.618033989
24	46368	17711	10946	1.618033989
25	75025	28657	17711	1.618033989
26	121393	46368	28657	1.618033989
27	196418	75025	46368	1.618033989
28	317811	121393	75025	1.618033989
29	514229	196418	121393	1.618033989
30	832040	317811	196418	1.618033989
31	1346269	514229	317811	1.618033989
32	2178309	832040	514229	1.618033989
33	3524558	1346269	832040	1.618033989
34	5702867	2178309	1346269	1.618033989
35	9227415	3524558	2178309	1.618033989
36	14930262	5702867	3524558	1.618033989
37	24157677	9227415	5702867	1.618033989
38	39088149	14930262	9227415	1.618033989
39	63245816	24157677	14930262	1.618033989
40	102334165	39088149	24157677	1.618033989
41	165580181	63245816	39088149	1.618033989
42	267914346	102334165	63245816	1.618033989
43	433494527	165580181	102334165	1.618033989
44	701408873	267914346	165580181	1.618033989
45	1134903394	433494527	267914346	1.618033989
46	1836312267	701408873	433494527	1.618033989
47	2971215661	1134903394	701408873	1.618033989
48	4807527928	1836312267	1134903394	1.618033989
49	7778743589	2971215661	1836312267	1.618033989
50	12586269025	4807527928	2971215661	1.618033989
51	20365583040	7778743589	4807527928	1.618033989
52	32951842065	12586269025	7778743589	1.618033989
53	53317114105	20365583040	12586269025	1.618033989
54	86268956170	32951842065	20365583040	1.618033989
55	139583860340	53317114105	32951842065	1.618033989
56	225852816495	86268956170	53317114105	1.618033989
57	365436676835	139583860340	86268956170	1.618033989
58	591290493330	225852816495	139583860340	1.618033989
59	956726169165	365436676835	225852816495	1.618033989
60	1548019256495	591290493330	365436676835	1.618033989
61	2501370162240	956726169165	591290493330	1.618033989
62	4049389418735	1548019256495	956726169165	1.618033989
63	6550769581075	2501370162240	1548019256495	1.618033989
64	10600159192110	4049389418735	2501370162240	1.618033989
65	17150918773185	6550769581075	4049389418735	1.618033989
66	27755818965300	10600159192110	6550769581075	1.618033989
67	44911737738485	17150918773185	10600159192110	1.618033989
68	72667556703785	27755818965300	17150918773185	1.618033989
69	117679314442270	44911737738485	27755818965300	1.618033989
70	190358904024055	72667556703785	44911737738485	1.618033989
71	308038218466330	117679314442270	72667556703785	1.618033989
72	498397122490385	190358904024055	117679314442270	1.618033989
73	796835340956715	308038218466330	190358904024055	1.618033989
74	1275232463447100	498397122490385	308038218466330	1.618033989
75	2036558304024055	796835340956715	498397122490385	1.618033989
76	3295184206530000	1275232463447100	796835340956715	1.618033989
77	5331711410554055	2036558304024055	1275232463447100	1.618033989
78	8626895617084100	3295184206530000	2036558304024055	1.618033989
79	13958386034038155	5331711410554055	3295184206530000	1.618033989
80	22585281649342300	8626895617084100	5331711410554055	1.618033989
81	36543667683380455	13958386034038155	8626895617084100	1.618033989
82	59129049332722755	22585281649342300	13958386034038155	1.618033989
83	95672616916103210	36543667683380455	22585281649342300	1.618033989
84	154801925648825965	59129049332722755	36543667683380455	1.618033989
85	250137016222729210	95672616916103210	59129049332722755	1.618033989
86	404938941871555175	154801925648825965	95672616916103210	1.618033989
87	655076958103284385	250137016222729210	154801925648825965	1.618033989
88	1060015919214839560	404938941871555175	250137016222729210	1.618033989
89	1715091877328023945	655076958103284385	404938941871555175	1.618033989
90	2775581896542863505	1060015919214839560	655076958103284385	1.618033989
91	4491173773860707450	1715091877328023945	1060015919214839560	1.618033989
92	7266755670373571005	2775581896542863505	1715091877328023945	1.618033989
93	11767931444241332460	4491173773860707450	2775581896542863505	1.618033989
94	19035890402414903465	7266755670373571005	4491173773860707450	1.618033989
95	30803821846638474970	11767931444241332460	7266755670373571005	1.618033989
96	49839712249053378435	19035890402414903465	11767931444241332460	1.618033989
97	79683534095691853405	30803821846638474970	19035890402414903465	1.618033989
98	127523246344745231870	49839712249053378435	30803821846638474970	1.618033989
99	203655830402438509315	79683534095691853405	49839712249053378435	1.618033989
100	329518420653077041190	127523246344745231870	79683534095691853405	1.618033989



If you want a variation on the Fibonacci series, you may include the function in a larger expression:

Log(FibonacciSeries*7)

Find(?, ?, ?, false)

Find(*var*, *findstring*, *n*, false) searches *var*'s values for *findstring* and returns the position of its first occurrence after the *n*th character (or 0 if it is not found). The *n* argument must be a positive integer. The fourth argument specifies whether the search should be case-sensitive; the default is false for case-insensitive searching, but you may specify true instead. If you supply a variable as the *findstring* argument, Find uses its exact values in the current format's display; changing formats can change results. If you supply a constant, you must enclose it in quotation marks. An optional fourth argument (1 or 2) specifies whether to handle text values as single-byte or double-byte strings; see below. The function works casewise.

Find(Model, "a", 1, false)

Model	'Where is a?
Acura Integra	1
Acura Legend V6	1
Audi 100	1
Audi 80	1
Audi 90	1
BMW 325i	0
BMW 535i	0
Buick Century	0
Buick Electra V6	13
Buick Le Sabre V6	11

Above, we find the first occurrence of the letter "a" at or after the first character in each value of Model in Car Data. (Model is an informative variable. To use it in a formula such as this, you must first change its class to nominal.) We can search for the second "a" by specifying the position of the first occurrence plus 1 as the starting point:

Find(Model, "a", Find(Model, "a", 1, false)+1, false)

Model	'Where is 2nd a?
Acura Integra	5
Acura Legend V6	5
Audi 100	0
Audi 80	0

In practice, it may be faster (and easier) to save the results of one Find in a variable and use that variable as the *n* argument:

Find(Model, "a", I, false)

Find(Model, "a", "Where is a?" + I, false)

You may include an optional fourth argument for specifying whether to handle text values as single-byte or double-byte strings. Find assumes a fourth argument 1 for single-byte strings (English, German, French, Spanish, etc. all use single-byte characters); specify 2 to use Find with strings containing double-byte characters, such as Japanese, Chinese, or Arabic characters. This example shows how to use Find with both single-byte (English) and double-byte (Japanese) characters.

Find(Teaching, "c", I, false, 1)

Find("教育方法", "書", I, false, 2)

	Teaching	教育方法	Find c	Find 書
1	Control	とくになし	1	0
2	Control	とくになし	1	0
3	Instructions	説明書を付与	7	3
4	Instructions	説明書を付与	7	3
5	Lecture	講習30分	3	0
6	Lecture	講習30分	3	0

See [Substring\(?, ?, ?\)](#) [p. 422] and [Len\(?\)](#) [p. 383] for more Find examples.

Floor(?)

Floor (*var*) rounds values of the variable you specify to the next lesser integer. Missing values are propagated. The function works casewise.

Floor(A)

A	RoundedA	TruncatedA	Floor of A	Ceil of A
-1.200	-1.000	-1.000	-2.000	-1.000
-3.915	-4.000	-3.000	-4.000	-3.000
•	•	•	•	•
.051	0.000	0.000	0.000	1.000
1.238	1.000	1.000	1.000	2.000
4.800	5.000	4.000	4.000	5.000

The floor of any number is the next lesser integer, regardless of the size of its fractional part and regardless of sign. Thus, the floor of -1.2 is -2 even though 0.2 is less than one-half and even though the floor of $+1.2$ is 1 . Remember, for negative numbers, “greater” and “lesser” can seem backwards: -2 is less than -1 . As do all computations, Floor works with actual stored values rather than the way values are displayed. For example, the value 1.9 is displayed in a format with no decimal places as 2 , but its floor is 1 .

Related functions are [Round\(?\)](#) [p. 416], [Trunc\(?\)](#) [p. 429], and [Ceil\(?\)](#) [p. 359]; a careful comparison of Round, Trunc, Ceil, and Floor is made in the entry for Round.

GeometricMean(?, AllRows)

GeometricMean(*var*, AllRows) computes the geometric mean of the variable you specify; by default, GeometricMean bases its calculations on AllRows of the variable, but you may instead specify OnlyIncludedRows and OnlyExcludedRows as the second argument. The geometric mean is undefined for variables containing negative or zero values. Missing values are ignored. The function works columnwise and produces the same result for every row.

GeometricMean(A, AllRows)

	A	GeotMean of A	HarMean of A
1	1.237	1.940	1.829
2	2.687	1.940	1.829
3	2.719	1.940	1.829
4	1.342	1.940	1.829
5	2.268	1.940	1.829

The geometric mean is a measure of position typically used with ratio data. It is defined as the *n*th root of the cumulative product of values in a variable, where *n* is the number of nonmissing values in the variable (Count). In the example above, 1.940 is the fifth root of the cumulative product. You can build your own formula for cumulative geometric means:

CumProduct(A)^(1/Count(A, AllRows))

The difference is that such a formula shows cumulative geometric means, whereas Geometric-Mean shows a single, final answer in all rows of the new variable.

A similar measure of position is the [HarmonicMean\(?, AllRows\)](#) [p. 382], which is useful with difference data.

GeometricSeries(1, 2)

GeometricSeries(*a*, *b*) generates a series with initial value *a* and each subsequent value at a common ratio *b* to the previous value. Both arguments *a* and *b* must be constants; they are 1 and 2 by default. The function works columnwise; results differ from row to row.

GeometricSeries(1,2)

Geo-1,2
1
2
4
8
16
32
64
128
256
512

Row *i* of the geometric series with *a* and *b* is ab^{i-1} .

To generate a series in which each value is the sum (rather than the product) of the previous value and a given constant, use [LinearSeries\(1, 1\)](#) [p. 384].

Groups(?, ...)

Groups(*var*, *var2*, ...) computes the groups or cells formed by combining several grouping variables. Missing values propagate missing values. The function works casewise.

Groups(Gender, Employment)

	Gender	Employment	SubGroups	NamedSubs
Type:	Category	Category	Real	Category
Source:	User En...	User Ente...	Dynamic ...	Dynamic For...
Class:	Nominal	Nominal	Nominal	Nominal
1	Male	Employed	1	Male-Emp1
2	Male	Unemployed	2	Male-Unemp1
3	Male	Employed	1	Male-Emp1
4	Female	Unemployed	4	Fem-Unemp1
5	Female	Employed	3	Fem-Emp1
6	Female	Unemployed	4	Fem-Unemp1
7	Male	Employed	1	Male-Emp1
8	Male	Unemployed	2	Male-Unemp1
9	Female	Employed	3	Fem-Emp1

Groups shows how several grouping variables nest with each other. You might collapse several grouping variables into one to simplify analyses, or you may use the new variable as a visual aid.

The example above shows how Gender and Employment might be nested together to form a new grouping variable, SubGroups. The NamedSubs variable shows how you might use named categories for these subgroups. (For illustration purposes, we worked with a copy of the SubGroups variable, but you could change the SubGroups variable itself to have type category.)

HarmonicMean(?, AllRows)

HarmonicMean(*var*, AllRows) computes the harmonic mean of the variable you specify; by default, HarmonicMean bases its calculations on AllRows of the variable, but you may instead specify OnlyIncludedRows and OnlyExcludedRows as the second argument. The harmonic mean is undefined for variables containing negative or zero values. Missing values are ignored. The function works columnwise and produces the same result for every row.

HarmonicMean(A, AllRows)

	A	GeoMean of A	HarMean of A
1	1.237	1.940	1.829
2	2.687	1.940	1.829
3	2.719	1.940	1.829
4	1.342	1.940	1.829
5	2.268	1.940	1.829

The harmonic mean is a measure of position typically used with difference data. It is defined as the count divided by the sum of reciprocals of the values.

[GeometricMean\(?, AllRows\)](#) [p. 380] is a similar measure of position typically used with ratio data.

Hour(?)

Hour(*date*) returns the hour number (0–23) of the *date* specified. The *date* argument may be a variable or a constant. (Remember, all date/time values are an exact second of an exact day, and unspecified times are assumed to be exactly midnight.) The function works casewise.

Hour("Some times")

Some times	Hour	Minute	Second
01:03:59 AM	1	3	59
02:05:00 AM	2	5	0
03:06:01 AM	3	6	1
04:07:02 AM	4	7	2
05:08:03 AM	5	8	3
06:09:04 AM	6	9	4

Lag(?,1)

Lag(*var*, *n*) lags the variable you specify by the number of cells *n* you specify. Leading values are filled in with missings. Missing values within a variable are copied at the lag just as non-missing values are. The function works columnwise; results differ from row to row.

Lag(A, 1)

Lag(A, 2)

Lag(A, -2)

A	Lag(A,1)	Lag(A,2)	Lag(A,-2)
2	•	•	8
4	2	•	16
8	4	2	32
16	8	4	64
32	16	8	128
64	32	16	256
128	64	32	512
256	128	64	1024
512	256	128	•
1024	512	256	•

Lagging moves a variable “down” the column one or more places. Above, Lag(A,1) effectively copies the cells in A and then pastes them a notch lower in the new variable. Lag(A,2) moves the values down two notches. The leading values (as many values as you specify in the second argument) are filled with missings, and as many values are chopped off the bottom of the variable. That is, a lagged variable will never be “longer” than the original. Lag(A, -2) moves the variable two notches “up” the column.

Lagging is a useful step in many sorts of variable transformations, especially when working with time series data. See also [Difference\(?, 1, 1\) \[p. 373\]](#), which subtracts from a variable its lag at the number of places you specify (and repeats that operation as many times as you specify).

Len(?)

Len(*text*) returns the length in characters of the *text* you specify. The *text* argument may be either a variable or constant. If you supply a variable as the *text* argument, Len returns the number of characters (letters, numbers, spaces, etc.) in the current format’s display of each value; changing formats can change results. If you supply a constant, you must enclose it in quotation marks. An optional second argument (1 or 2) specifies whether to handle text values as single-byte or double-byte strings; see below. The function works casewise.

Len(Model)

Model	Length of Model
Acura Integra	13
Acura Legend V6	15
Audi 100	8
Audi 80	7
Audi 90	7
BMW 200i	8

Len “measures” the length of a variable’s values. Usually Len is used with string variables, but other variable types also work. Above, we compute the length of each model name in the sample dataset, Car Data. (Model is an informative variable. To use it in a formula such as this, you must first change its class to nominal.)

Len is most useful in combination with other text functions such as Find, Substring, Concat. For example, we can combine Substring, Find, and Len to separate model names from the Model variable:

Substring(Model, Find(Model, " ", 1, false)+1, Len(Model))

Model	Model name
Acura Integra	Integra
Acura Legend V6	Legend V6
Audi 100	100
Audi 80	80
Audi 90	90
DMU 275E	275E

This example finds the substring of Model starting right after the first space and including up to as many characters as the total length of Model. (You must change the formula variable to have type string.)

You may include an optional second argument for specifying whether to handle text values as single-byte or double-byte strings. Len assumes a second argument 1 for single-byte strings (English, German, French, Spanish, etc. all use single-byte characters); specify 2 to use Len with strings containing double-byte characters, such as Japanese, Chinese, or Arabic characters. This example shows how to use Len with both single-byte (English) and double-byte (Japanese) characters.

```
Len(Teaching, 1)
Len("教育方法",1)
Len("教育方法",2)
```

	Teaching	1byte TLen	教育方法	1byte 教Len	2byte 教Len
1	Control	7	とくになし	10	5
2	Control	7	とくになし	10	5
3	Instructions	12	説明書を付与	12	6
4	Instructions	12	説明書を付与	12	6
5	Lecture	7	講習30分	10	5
6	Lecture	7	講習30分	10	5

LinearSeries(1, 1)

LinearSeries(*a*, *b*) creates a series with initial value *a* and each subsequent value *b* greater than its predecessor. The function works columnwise; results differ from row to row.

```
LinearSeries(1, 2)
```

Linear-1,2
1
3
5
7
9
11
13
15
17
19

Row *i* of the linear series with *a* and *b* is $a + b(i - 1)$.

To generate a series in which each value is the product (rather than the sum) of the previous value and a given constant, use [GeometricSeries\(1, 2\)](#) [p. 381].

Ln(?)

Ln(*var*) returns the base *e* logarithm (“natural logarithm”) of the argument, where *e* is a constant whose value is approximately 2.718. Logarithms of negative numbers and zero are undefined; these and missing values produce missing values. The function works casewise.

Ln(A)

A	Ln(A)
1.000	0
2.718	1
7.389	2
20.086	3
54.598	4
148.413	5
403.429	6

Logarithms are defined as the inverse of exponents—for instance, $\text{Ln}(2.718)=1$ means that $e^1 = 2.718$. Therefore, exponentiating the log of an argument returns the argument. Logarithms are useful because they reduce multiplication, division, and exponentiation to simpler operations, addition, subtraction, and multiplication:

$$\ln(xy) = \ln x + \ln y$$

$$\ln\left(\frac{x}{y}\right) = \ln x - \ln y$$

$$\ln(x^n) = n \ln x$$

Thus, logging data can “simplify” or “straighten” the relationship of two variables in an analysis. When scattergrams show curved or spreading relationships between variables, it is often useful to try logging one or both variables. This is illustrated in the discussion of the *e* function.

See [e \[p. 375\]](#) for the constant *e*, [Log\(?\) \[p. 385\]](#) for common (base 10) logarithms and [LogB\(?, ?\) \[p. 386\]](#) for logarithms to other bases.

Log(?)

Log(*var*) returns the base 10 logarithm (“common logarithm”) of the constant or variable you specify. Logs of negative numbers and zero are undefined; these and missing values produce missing values. The function works casewise.

Log(A)

A	Log(A)
-100	•
-10	•
•	•
0	•
10	1
100	2

Properties of logarithms are discussed under [Ln\(?\) \[p. 385\]](#), which produces natural (base *e*) logarithms. If you need logarithms to other bases, use [LogB\(?, ?\) \[p. 386\]](#).

LogB(?, ?)

LogB(*var*, *b*) returns the base *b* logarithm of the variable or constant you specify. Logs of negative numbers and zero are undefined; these and missing values produce missing values. The function works casewise.

Log(A, 2)

A	LogBase2 of A
.125	-3.000
.250	-2.000
.500	-1.000
1.000	0.000
2.000	1.000
4.000	2.000
8.000	3.000
16.000	4.000
32.000	5.000
64.000	6.000

Properties of logarithms are discussed under Ln(?) [p. 385], which produce natural (base *e*) logarithms. For common (base 10) logarithms, see Log(?) [p. 385].

LogOdds(?)

LogOdds(*var*) computes the log odds transformation of the variable or constant you specify. Missing values propagate missing values. The function works casewise.

LogOdds("Prop heads")

Heads in 5	Prop heads	Log(Odds ratio)
2	.4	-.405
1	.2	-1.386
1	.2	-1.386
4	.8	1.386
4	.8	1.386
1	.2	-1.386
1	.2	-1.386
2	.4	-.405
3	.6	.405
5	1.0	•

The log odds transformation is useful for stabilizing the variance of response data that are expressed as a proportion of successes. The value of the transformed variable is defined on each case as

$$\ln\left(\frac{x}{1-x}\right)$$

where *x* is the value of the original variable on that case.

The example shows a log odds transformation for a variable recording proportions of successes in a coin-toss experiment. The first variable counts the number of times 5 coin tosses produced heads. The second column converts this to proportions, where the numbers of heads are divided by 5 tosses. The third column shows the log-odds transformation of those proportion data. The first row, for instance, shows that the first trial had an outcome of 2 heads (or

successes) from 5 tosses. The second row converts this to 0.4, meaning 40% of the trials were heads (successes), by dividing 2 by 5. The third row is obtained by:

$$\ln\left(\frac{0.4}{1-0.4}\right) = \ln\left(\frac{0.4}{0.6}\right) = \ln(0.666...) = -0.405...$$

The last case of the transformed variable is missing because division by zero is undefined.

MAD(?, AllRows)

MAD(*var*, AllRows) computes the median absolute deviation from the median of the variable you specify; by default, MAD bases calculations on AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

MAD(A, AllRows)

A	MedianOfA	MADofA
-4	-1.500	4.500
-5	-1.500	4.500
•	-1.500	4.500
•	-1.500	4.500
1	-1.500	4.500
5	-1.500	4.500

The MAD is a measure of variability (or spread) analogous to the standard deviation. As standard deviation averages the variability of actual points from the mean, MAD takes the median of differences between points and the median; and, as median is less vulnerable to extreme data points than the mean, MAD is less vulnerable to outliers than standard deviation.

Related functions are [Mean\(?, AllRows\)](#) [p. 388], [Median\(?, AllRows\)](#) [p. 388], and [Standard-Deviation\(?, AllRows\)](#) [p. 420].

Maximum(?, AllRows)

Maximum(*var*, AllRows) identifies the largest value in the variable you specify; by default, Maximum is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Maximum(A, AllRows)

	A	MaxOfA
Minimum:	-4	5
Maximum:	5	5
•	•	•
1	-4	5
2	-3	5
3	•	5
4	•	5
5	1	5
6	5	5

Maximum is also shown in the summary pane. The Maximum function can be useful in combination with other functions and operators or when you need to have the value available as a variable to some analysis.

Understand that “maximum” is the *greatest* value. Even negative values of great magnitude (for example, −1,000) are smaller than positive values of small magnitude (for example, 0.001). If you want to know the number of greatest *magnitude*, use absolute values, e.g., set B to Abs(A), then do Maximum(B, AllRows).

Mean(?, AllRows)

Mean(*var*, AllRows) computes the mean of the variable you specify; by default, Mean is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Mean(A, AllRows)

	A	MeanOfA
Mean:	-.250	-.250
Std. Deviation:	4.113	0.000
1	-4.000	-.250
2	-3.000	-.250
3	●	-.250
4	●	-.250
5	1.000	-.250
6	5.000	-.250

Mean is a measure of the central tendency of a variable and is defined as the sum of all non-missing values divided by the number of nonmissing values, so Mean(A) is the same as SumOfColumn(A)/Count(A). Mean is also shown in the summary pane for each variable; the Mean function is mostly useful for computing other statistics or when you need to have the mean available as a variable to some analysis.

Mean is a columnwise (vertical) function. If you want a casewise mean (the mean for each row of several variables), use [Average\(?, ...\) \[p. 356\]](#) or [AverageIgnoreMissing\(?, ...\) \[p. 357\]](#).

Median(?, AllRows)

Median(*var*, AllRows) computes the columnwise median of the variable you specify; by default, Median is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Median(A, AllRows)

A	MedianOfA
-4.000	-1.000
-3.000	-1.000
●	-1.000
●	-1.000
1.000	-1.000
5.000	-1.000

Median, like mean, is a measure of the central tendency of a variable. Median is defined as the middle value of a variable if all the values are listed in order of size. (So, for a variable with 7 values, the median is the 4th largest value.) If a variable has an even number of values, the median is the mean of the two middle values. (So, for a variable with 8 values, the median is the sum of the 4th and 5th value, divided by two.)

To find other percentile values, see the discussion under [Percentile\(?, ?, ?\) \[p. 398\]](#).

Minimum(?, AllRows)

Minimum(*var*, AllRows) identifies the least value in the variable you specify. By default, Minimum is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Minimum(A, AllRows)

	A	MinOfA
Minimum:	-4	-4
Maximum:	5	-4
1	-4	-4
2	-3	-4
3	•	-4
4	•	-4
5	1	-4
6	5	-4

Minimum is also shown in the summary pane. The Minimum function can be useful in combination with other functions and operators or when you need to have the value available as a variable to some analysis.

Understand that “minimum” is the *smallest* value. Even negative values of great magnitude (for example, -1,000) are smaller than positive values of small magnitude (for example, 0.001). If you want to know the number of smallest *magnitude*, use absolute values, e.g., set B to Abs(A) and then do Minimum(B, AllRows).

Minute(?)

Minute(*date*) returns the minute number (0–59) of the *date* specified. The *date* argument may be a variable or a constant. (Remember, all date/time values are an exact second of an exact day, and unspecified times are assumed to be exactly midnight.) The function works casewise.

Minute("Some times")

Some times	Hour	Minute	Second
01:03:59 AM	1	3	59
02:05:00 AM	2	5	0
03:06:01 AM	3	6	1
04:07:02 AM	4	7	2
05:08:03 AM	5	8	3
06:09:04 AM	6	9	4

Mod(?, ?)

Mod(*var1*, *var2*) computes *var1* modulo *var2*, which is the remainder after dividing *var1* by *var2* to an integer result. Both arguments may be constants or variables. Missing values in either argument or division by zero propagates missing values. The function works casewise.

Mod(A, B)

A	B	Div(A,B)	Mod(A,B)	Rem(A,B)
-5	2	-2	-1	-1.0
-7	2	-3	-1	1.0
-12	7	-1	-5	2.0
-3	-2	1	-1	1.0
•	4	•	•	•
0	3	0	0	0.0
1	0	•	•	•
17	4	4	1	1.0

Ordinarily, division is computed to as many decimal places as is necessary for an exact answer, within the limits of the variable’s precision. For instance, 5/3 is 1.6666... (an infinite series of 6s after the decimals). Mod(5,3) stops dividing when it reaches the decimal point and then records the remainder, or the leftover part—this is the way children learn long division:

$$\begin{array}{r} 1 \text{ r } 2 \\ 3 \overline{)5} \\ \underline{3} \\ 2 \end{array}$$

Children are taught to divide until the amount at the bottom is smaller than the divisor, and then write that leftover part as “remainder 2.” This casual definition is sufficient for positive numbers, but for negative numbers, a more precise definition is needed. Mod(*var1*, *var2*) is formally defined as *var1*–(Trunc(*var1*/*var2*))**var2*.

[Remainder\(?, ?\)](#) [p. 413] is the same as Mod for positive arguments, but for negative arguments, Remainder and Mod are different. The formal definition of Remainder is *n*–(Round(*n*/*m*))**m*. See [Trunc\(?\)](#) [p. 429] and [Round\(?\)](#) [p. 416] for details; briefly, rounding goes up or down to the nearest integer, whereas truncation deletes digits after the decimal. Finally, see [Div\(?, ?\)](#) [p. 374] for the integer part of a quotient.

Mode(?, AllRows)

Mode(*var*, AllRows) identifies the value that occurs most often in the variable you specify; by default, Mode is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. When no single mode exists, missing values result. The function works columnwise and produces the same result for every row.

Mode(A, AllRows)

A	Mode of A
1	2
1	2
2	2
2	2
2	2
2	2
3	2
3	2
4	2
4	2

Mode can be used as a measure of central tendency for variables that can take a limited number of values, or where the values clump together.

Month(?)

Month(*date*) returns the month number (1–12) of the *date* specified. The *date* argument may be a variable or a constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The function works casewise.

Month("Other dates")

Other dates	Month
01.01.95	1
02.02.95	2
03.03.95	3
04.04.95	4
05.05.95	5
06.06.95	6

If you want month names instead of numbers, change the variable to type category and edit the category to have values January, February, etc.

MovingAverage(?, ?)

MovingAverage(*var*, *n*) computes a moving average for the variable you specify as the first argument, averaging *n* neighboring rows at a time. A missing value on any row propagates missing values on that row and the next *n*–1 rows. The function works columnwise; results differ from row to row.

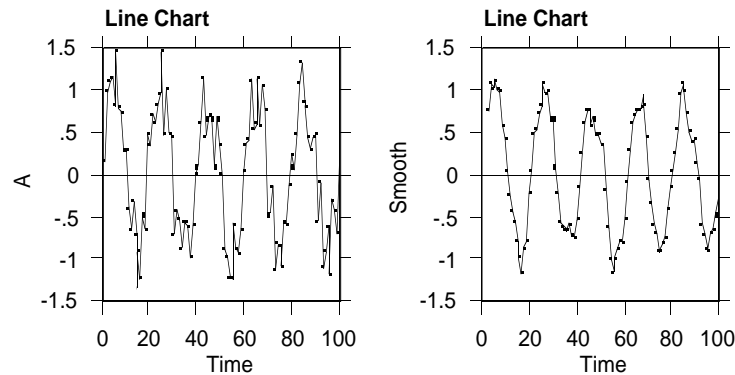
MovingAverage(A, 3)

A	Smooth	Time
.138	•	1
.971	•	2
1.103	.738	3
1.159	1.078	4
.801	1.021	5
1.439	1.133	6
.779	1.006	7

A moving average with a window of 3, for example, averages the first three values and records that answer in the third case of the new variable. Then, it averages the second, third, and fourth values and records that is the fourth case of the new variable. The third, fourth, and fifth values are averaged for the fifth case, etc. For a window of width *n*, the first *n*–1 values of the new variable are missing.

Moving averages are often used with time series data. Usually measurements taken over time will show, along with any long-term trends, some random short-term fluctuation. For instance, toy sales data may have a long-term tendency to increase near holidays, but weekly totals will tend to bob up and down a small amount. It can be helpful to smooth this “noise” by using moving averages.

The data above seem to follow a periodic function; smoothing these data with `MovingAverage(A,3)` makes the trend more apparent. (In fact, these data are values along a sine wave with random uniform noise added.)



For casewise (horizontal) averages, see [Average\(?, ...\)](#) [p. 356] and [AverageIgnoreMissing\(?, ...\)](#) [p. 357]. For a single average on an entire variable, use [Mean\(?, AllRows\)](#) [p. 388] or see `Mean` in the summary pane. Other functions useful with time series data are [Difference\(?, 1, 1\)](#) [p. 373], [Lag\(?,1\)](#) [p. 382], and the date/time functions.

Norm(?, AllRows)

`Norm(var, AllRows)` computes the Euclidean norm of the variable you specify; by default, `Norm` uses `AllRows` of the variable, but you may instead specify `OnlyIncludedRows` or `OnlyExcludedRows` as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

`Norm(A, AllRows)`

A	Norm of A
-4	7.141
-3	7.141
•	7.141
0	7.141
1	7.141
5	7.141

Euclidean norm is a linear algebra function; the norm of a vector is its magnitude, or length. It is computed by squaring all its values, adding them, and then taking the square root of that sum. `Norm` is equivalent to `Sqrt(SumOfSquares(var, AllRows))`.

[DotProduct\(?, ?\)](#) [p. 374] is another linear algebra function.

Now

Now returns the current date and time, at the time you click Compute in the Formula window. Now takes no arguments. The function works casewise.

Now

A Now from history	
Date/Time	
Dynamic Formula	
Continuous	
Friday, 1 January 1904 00:00:00	
•	
Wednesday, 3 May 1995 15:21:33	
Wednesday, 3 May 1995 15:21:33	
Wednesday, 3 May 1995 15:21:33	
Wednesday, 3 May 1995 15:21:33	

As do all the functions, Now creates a variable with type Real. Be sure to change its type to Date/Time and choose an appropriate format. Also be sure that your Date & Time control panel is set correctly.

Now is only current at the exact second you click Compute. Even in a dynamic formula involving other variables that may change, the value of Now does not update itself; however, if you reopen its formula window (by selecting Static or Dynamic Formula from the source pop-up menu in the attribute pane) and click Compute again, it is updated.

Shortcut You may enter 0 in a date/time data cell to get the current date at midnight.

NumberMissing(?, AllRows)

NumberMissing(*var*, AllRows) counts the number of cases in a variable that have missing values. By default, NumberMissing uses AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. The function works columnwise and produces the same result for every row.

NumberMissing(A, AllRows)

	A	NumMiss	NumRows	CountOfA
Count:	4	6	6	6
Missing Cells:	2	0	0	0
1	-4	2	6	4
2	-5	2	6	4
3	•	2	6	4
4	•	2	6	4
5	1	2	6	4
6	5	2	6	4

NumberMissing shows the same result as Missing Cells in the summary pane for the variable. Its complement is [Count\(?, AllRows\)](#) [p. 365], which counts the number of *nonmissing* values and is also shown in the attribute pane. NumberMissing(*var*, AllRows) and Count(*var*, AllRows) sum to NumRows, which counts all cases in the dataset, whether missing or nonmissing.

NumberOfRows

NumberOfRows counts the number of rows in the dataset. NumberOfRows takes no argument. If you provide one by mistake, e.g., NumberOfRows(A), it is interpreted as multiplication. NumberOfRows produces the same result for every row.

NumberOfRows

	A	NumMiss	NumRows	CountOfA
Count:	4	6	6	6
Missing Cells:	2	0	0	0
1	-4	2	6	4
2	-5	2	6	4
3	•	2	6	4
4	•	2	6	4
5	1	2	6	4
6	5	2	6	4

Since all variables in a dataset must by definition have the same number of rows, “shorter” variables are padded with missing values at the bottom. If analyses show missing values you didn’t expect, check whether NumberOfRows is greater than the number of observations you recorded for the variable you’re studying.

[Count\(?, AllRows\)](#) [p. 365] shows the number of nonmissing values in a specific variable, and [NumberMissing\(?, AllRows\)](#) [p. 393] shows the number of missing values. Count(*var*, AllRows) and NumberMissing(*var*, AllRows) add to NumberOfRows.

OneGroupChiSquare(?, ?, ?)

OneGroupChiSquare(*obsvar*, *expvar*, *n*) computes a one group chi-square test comparing observed counts in the *obsvar* against expected counts in the *expvar*. If you set *n* to 0, the formula computes the chi-square statistic. If you set *n* to any nonzero number, the formula computes the probability for the chi-square test. Missing values are ignored. The function works columnwise and produces the same result for every row.

OneGroupChiSquare(Observed, Expected, 0)

OneGroupChiSquare(Observed, Expected, 7)

Gender	Observed	Expected	ChiSq	P
Males	70	90	8.889	.003
Females	110	90	8.889	.003

Chi-square tests compare observed data with expected outcomes if the null hypothesis is true. Above we compare the number of female and male graduates from nursing school. Our null hypothesis is that an equal number of men and women graduate. Obviously more women than men graduated in this case, but is the difference significant? The one group chi-square test returns a chi-squared value of 8.889 and a probability of 0.003, so we can reject the null hypothesis that this nursing school produces an even mix of graduates.

Suppose we suspect that the national mint has been producing “unfair” coins—coins that are more likely to land heads than tails. To test this, we toss a coin ten times and record the number of times we get heads, and we repeat that experiment with one thousand coins. Since a “fair” coin toss follows a binomial distribution, our null hypothesis is that the probability dis-

tribution is binomial with count 10 and probability 0.5. (If you'd like to generate fake data to try this problem yourself, use the Series command with RandomBinomial(10, 0.5) and 1000 rows (see [RandomBinomial\(?, ?\) \[p. 406\]](#)). Your numbers may differ from ours, but the test outcome should be similar.)

#Heads
5
8
5
4
6
6
2
7
5
5
6

The first thing we need to do is convert these data to frequency data. We open a new view and create a Frequency Summary Table for #Heads, where we specify even intervals of width 1 and initial value 0, and we choose to include highest values (we'll see why in a moment):

Frequency Distribution

Number of intervals: 10 ☐ Show normal comparison

Do you wish to enter your own interval information?

☐ no ☒ yes width: 1 initial value: 0

Intervals indicate: Count include: Highest value

Tables show: ☒ Counts ☐ Percents ☐ Relative frequencies

Histograms show: Counts

Cancel OK

We get a table something like this, which we can Copy and Paste into a new dataset:

Frequency Distribution for #Heads		
From (>)	To (≤)	Count
0.000	1.000	8
1.000	2.000	53
2.000	3.000	117
3.000	4.000	212
4.000	5.000	218
5.000	6.000	211
6.000	7.000	112
7.000	8.000	59
8.000	9.000	10
9.000	10.000	0
Total		1000

	Column 1	Column 2	Column 3
Type:	String	String	String
Source:	User Entered	User Entered	User Entered
1	Frequency ...		
2	From (>)	To (≤)	Count
3	0.000	1.000	8
4	1.000	2.000	53
5	2.000	3.000	117
6	3.000	4.000	212
7	4.000	5.000	218
8	5.000	6.000	211
9	6.000	7.000	112
10	7.000	8.000	59
11	8.000	9.000	10
12	9.000	10.000	0
13	Total		1000

Next, we select and delete the first column, the first two rows, and the last row, and we change the data types to real (or integer):

	#Heads	Counts
Type:	Real	Real
Source:	User ...	User ...
1	1	8
2	2	53
3	3	117
4	4	212
5	5	218
6	6	211
7	7	112
8	8	59
9	9	10
10	10	0

Next, we can use the binomial cumulative distribution function `ProbBinomial` (see [ProbBinomial\(?, ?, ?\)](#) [p. 401]) to generate the probabilities we should expect for each outcome if our null hypothesis (that the data follow a binomial distribution with parameters 10 and 0.5) is true. Remember, the CDF functions compute the proportion of data falling *at or below* the value you specify. So, we generate `ProbBinom` with the formula `ProbBinomial(#Heads, 10, 0.5)`.

`ProbBinomial(#Heads, 10, 0.5)`

Since our counts are not cumulative, we want the probabilities for exactly each number of heads. To do this, we could use `Difference(ProbBinom, 1, 1)`, but that would leave a missing value in the first case. So would subtracting a lagged version of the variable from the variable. Instead, we use `SumIgnoreMissing(ProbBinom, -Lag(ProbBinom, 1))`; see [SumIgnoreMissing\(?, ...\)](#) [p. 424]. Finally, we multiply these expected probabilities by sample size (1000) to get expected counts:

`SumIgnoreMissing(ProbBinom, -Lag(ProbBinom, 1))`

`DiffProb*1000`

#Heads	Counts	ProbBinom	DiffProb	Expected	ChiSq	P
1	8	.010742	.010742	10.7	12.547	.184
2	53	.054688	.043945	43.9	12.547	.184
3	117	.171875	.117188	117.2	12.547	.184
4	212	.376953	.205078	205.1	12.547	.184
5	218	.623047	.246094	246.1	12.547	.184
6	211	.828125	.205078	205.1	12.547	.184
7	112	.945312	.117188	117.2	12.547	.184
8	59	.989258	.043945	43.9	12.547	.184
9	10	.999023	.009766	9.8	12.547	.184
10	0	1.000000	.000977	1.0	12.547	.184

Now we're ready to compare observed counts with the expected counts we've computed. We'll use `OneGroupChiSquare` twice; once with the third argument set to 0 for the chi-square statistic, and again set to 1 for the test probability:

`OneGroupChiSquare(Counts, Expected, 0)`

`OneGroupChiSquare(Counts, Expected, 1)`

#Heads	Counts	ProbBinom	DiffProb	Expected	ChiSq	P
1	8	.010742	.010742	10.7	12.547	.184
2	53	.054688	.043945	43.9	12.547	.184
3	117	.171875	.117188	117.2	12.547	.184
4	212	.376953	.205078	205.1	12.547	.184
5	218	.623047	.246094	246.1	12.547	.184
6	211	.828125	.205078	205.1	12.547	.184
7	112	.945312	.117188	117.2	12.547	.184
8	59	.989258	.043945	43.9	12.547	.184
9	10	.999023	.009766	9.8	12.547	.184
10	0	1.000000	.000977	1.0	12.547	.184

Since P is well above the most liberal criterion 0.05, we cannot reject the hypothesis that our experiments follow a binomial distribution for count 10 and probability 0.5. The coins must be fair.

Percentages(?, AllRows)

Percentages(*var*, AllRows) returns for each row that row's percentage contribution to the sum of the column specified. By default, percentages are computed on AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise; results differ from row to row.

Percentages(A, AllRows)

	A	percentage of A
Missing Cells:	0	0
Sum:	55.000	100.000
1	1.000	1.818
2	2.000	3.636
3	3.000	5.455
4	4.000	7.273
5	5.000	9.091
6	6.000	10.909
7	7.000	12.727
8	8.000	14.545
9	9.000	16.364
10	10.000	18.182

For example, the Sum in the attribute pane for A shows that the Sum of A (the total if you add all the values in the variable) is 55. The values of the “percentage of A” variable are equal to the value of A for that row divided by 55 and multiplied by 100. Simply, 1 is 1.818 percent of 55, 2 is 3.636 percent of 55, etc.

Percentages are one way to standardize values for variables with different magnitude. Suppose, for example, that you are comparing annual income, meal expenses, and clothing expenses valued in German marks. However, since income likely falls in the tens of thousands and the expenses might be closer to a few thousand, it would be misleading to use the variables together in a regression or a factor analysis. The magnitude of the income values would overwhelm the significance of other variables. Converting each variable to show values as percentages of a whole makes the variables more comparable.

Do not confuse percentages with percentiles. Remember, Percentage translates each value into a percentage of the sum. [Percentile\(?, ?, ?\) \[p. 398\]](#) gives the number below which a given percentage of the other values lie.

Percentile(?, ?, ?)

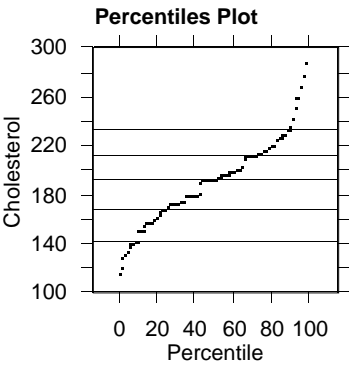
Percentile(*var*, *p*, AllRows) computes the value at the *p*th percentile for the variable you specify. By default, the percentile is computed from AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the third argument. The *var* and *p* arguments may be variables or constants. Missing values are ignored. The function works column-wise; results differ from row to row.

Percentile(Cholesterol, 50, AllRows)

Cholesterol	CholPercentile
197	191.000
181	191.000
190	191.000
131	191.000
172	191.000

It is difficult to give a strict definition of percentile that makes sense. For most purposes, a percentile answers the question, “What value is the cut-off point, where such-and-such percentage of the cases are equal to or smaller than that value?” So, the 10th percentile of a variable is a number that 10 percent of the values are as small as or smaller than. A given percentile is not necessarily a value in the variable. For instance, with an even number of cases, the 50th percentile (the median) is in between the middle two cases. Another exception is when many of the values in a variable repeat themselves.

With the Lipid data above, Percentile(Cholesterol, 50, AllRows) fills a new column with the value 191, which is the 50th percentile of the variable Cholesterol when using all rows of the variable. Consider a percentiles plot of the same variable. This graph plots each value in the variable against its percentile, and draws lines at the 10th, 25th, 50th, 75th, and 90th percentiles. The 50th percentile line intersects the curve of data points at the *y*-value 191, just as the Percentile function reports.



To find several percentiles of a variable in one step, create a variable with the percentile values you want, then supply that variable as the *p* argument. For example, we could produce every 20th percentile of Cholesterol this way:

Percentile(Cholesterol, %levels, AllRows)

Cholesterol	%levels	%iles
197	20	160.000
181	40	180.000
190	60	197.000
131	80	217.000
172	•	•
233	•	•
194	•	•

You can convert raw scores to their percentile equivalents by writing your own formula combining the Rank and Count functions. For example, with Car Data, you may convert Weight values to percentiles:

$$(\text{Rank}(\text{Weight}, \text{AllRows})-0.5)/\text{Count}(\text{Weight}, \text{AllRows})*100$$

Weight	Wt %ile
2700	32.328
3265	69.397
2935	52.155
2670	28.448
2790	40.086
2895	46.982

To convert raw scores to their nearest *n*th percentile, first convert scores to their percentile equivalents, as above, then use the Round function to find the nearest-*n*th “clump” of percentiles. For example, suppose you want to see the percentile equivalents of Weight values in 10-percentile increments:

$$(\text{Round}(\text{"Wt \%ile"/10}))*10$$

Weight	Wt %ile	Wt nrst 10%ile
2700	32.328	30
3265	69.397	70
2935	52.155	50
2670	28.448	30
2790	40.086	40

You could combine those formulas into a single step if you preferred. Be aware that clumping percentiles into intervals as in this examples does produce “100th percentile” values.

Do not confuse percentiles with percentages. Remember, the Percentile function gives the number below which a given percentage of the values lie. [Percentages\(?, AllRows\) \[p. 397\]](#) translates each value into a percentage of the variable’s sum. To find the *n*th largest raw value of a variable, see VariableElement.

Permutations(?, ?)

Permutations(*n*, *r*) computes the casewise permutations of *n* objects taken *r* at a time, where *n* and *r* can be variables or constants. Cases with *r* greater than *n*, negative values, or missing values are missing. The function works casewise.

$$\text{Permutations}(n, r)$$

n	r	Perm(n,r)
-2	1	•
3	•	•
4	3	24
5	1	5
5	2	20
5	3	60
5	6	•

Permutations(*n*, *r*) computes the number of *r*-object ordered combinations taken from *n* objects (such as the number of four letter words taken from a set of nine letters), or

$$\frac{n!}{(n-r)!}$$

For example, Permutations(5,3) on the second to last row is 60, which means that you could assign a president, vice-president, and secretary combination of three people in 60 different ways if you had a group of five people to choose from:

$$\frac{5!}{(5-3)!} = \frac{5!}{2!} = \frac{120}{2} = 60$$

For *unordered* combinations, see the [Combinations\(?, ?\) \[p. 361\]](#) function. Both Permutations and Combinations rely on the use of factorials (such as *n!*), which can also be computed individually with the [Factorial\(?\) \[p. 377\]](#) function; factorials are defined in that entry.

Pi

Pi returns the constant Pi, which is approximately 3.14159... Unless you use Pi in combination with other functions, it returns a variable in which every case is π . The function works casewise.

pi

A + pi

	A	Pi	A+Pi
1	-4.000	3.14159	-.85841
2	-3.000	3.14159	.14159
3	•	3.14159	•
4	0.000	3.14159	3.14159
5	1.000	3.14159	4.14159
6	5.000	3.14159	8.14159

You can use Pi in combination with other functions, as seen in examples for the trigonometric functions Sin, Cos, Tan, etc., which take arguments in radians (which are multiples and fractions of π).

For the trig function examples, we generated Radians values with the formula

$$(\text{RowNumber} - 1)/6*\text{pi}$$

and we entered by hand the values of the informative string variable “Radians π .” These familiar values are provided to make the examples easier to read. The other variables are created by formulas using Radians as argument:

Sin(Radians)

Cos(Radians)

	Radians π	Radians	Sine	Cosine
1	zero	0.000	0.000	1.000
2	$\pi/6$.524	.500	.866
3	$\pi/3$	1.047	.866	.500
4	$\pi/2$	1.571	1.000	0.000
5	$2\pi/3$	2.094	.866	-.500
6	$5\pi/6$	2.618	.500	-.866
7	π	3.142	0.000	-1.000
8	$7\pi/6$	3.665	-.500	-.866
9	$4\pi/3$	4.189	-.866	-.500
10	$3\pi/2$	4.712	-1.000	0.000
11	$5\pi/3$	5.236	-.866	.500
12	$11\pi/6$	5.760	-.500	.866
13	2π	6.283	0.000	1.000

These data points are classic examples in the study of the unit circle. Pi is formally defined as the ratio of a circle’s circumference to its diameter and is commonly seen in the formulas:

Circumference = πd
Area = πr^2

ProbBinomial(?, ?, ?)

ProbBinomial(x, n, p) computes the cumulative distribution function at x of a binomial random variable with count n and probability p . The arguments may be variables or constants; results are computed casewise. All three arguments should be positive; illegal or missing values in any argument propagate missing values.

ProbBinomial(A, 5, 0.5)

A	CDF Binom 5, .5
0	.031
1	.188
2	.500
3	.812
4	.969
5	1.000

A cumulative distribution function computes the probability that a random value from that distribution falls below a value x you provide. In other words, a CDF returns the proportion of the distribution that is less than x . CDF functions are useful for creating your own statistical tests, e.g., type I errors.

In the example above, we computed the CDF at A for a binomial random variable with 5 events and probability 0.5 of each event being a success. We can interpret the results for the second-to-last case, for example, as follows: 96.9% of the values of a normal binomial variable fall at or below the value 4. Another way of stating this is that we have a 0.969 probability of having four or fewer successes in five trials of an experiment having equal chances for success and failure, such as five fair coin tosses.

To generate random Binomial data, see [RandomBinomial\(?, ?\)](#) [p. 406]. Also note that the Bernoulli distribution is a special case of the binomial distribution in which $n=1$.

ProbChiSquare(?, I)

ProbChiSquare(*x*, *df*) computes the cumulative distribution function at *x* of a chi-square random variable with *df* degrees of freedom. The arguments may be variables or constants; results are computed casewise. Both arguments should be greater than zero, and *df* must be integer; negative, zero, fractional, and missing values propagate missing values.

```
ProbChiSquare(A, I)
ReturnChiSquare("CDF chi-sq", I)
```

A	CDF chi-sq	InvCDF chi-sq
-4	●	●
-3	●	●
●	●	●
0	●	●
1	.683	1
5	.975	5

A cumulative distribution function computes the probability that a random value from that distribution falls at or below a value *x* you provide. In other words, a CDF returns the proportion of the distribution that is less than or equal to *x*. CDF functions are useful for creating your own statistical tests, e.g., type I errors.

In the example above, we computed the CDF at A for a chi-square random variable with 1 degree of freedom; then we computed the inverse CDF by applying ReturnChiSquare to the CDF chi-sq values. Applying the inverse CDF to the CDF returned the original values from A. We can interpret the results for the last case, for example, as follows: 97.5% of the values of a chi-square random variable fall at or below the value 5. Another way of stating this is that we have a 0.975 probability of choosing at random a value that is 5 or less from a chi-square random variable with 1 degree of freedom.

For the inverse CDF, see [ReturnChiSquare\(?, ?\) \[p. 414\]](#). To generate random chi-square data, see [RandomChiSquare\(1\) \[p. 407\]](#).

ProbF(?, I, I)

ProbF(*x*, *df*, *df2*) computes the cumulative distribution function at *x* of an F random variable with *df* degrees of freedom in the numerator and *df2* degrees of freedom in the denominator. The arguments may be variables or constants, and degrees of freedom must be positive integers; results are computed casewise. All three arguments should be greater than zero; negative, zero, and missing values propagate missing values.

```
ProbF(A, I, I)
ReturnF("CDF F", I, I)
```

A	CDF F	InvCDF F
-4.0	●	●
39.9	.900	39.9
647.8	.975	647.8
4052.0	.990	4052.0
161.4	.950	161.4
16211.0	.995	16211.0

A cumulative distribution function computes the probability that a random value from that distribution falls at or below a value x you provide. In other words, a CDF returns the proportion of the distribution that is less than or equal to x . CDF functions are useful for creating your own statistical tests, e.g., to compute p for an F test.

In the example above, we computed the CDF at A for an F random variable with 1 and 1 degrees of freedom; then we computed the inverse CDF by applying ReturnF to the CDF F values. Applying the inverse CDF to the CDF returned the original values from A. We can interpret the results for the last case, for example, as follows: 99.5% of the values of an F random variable fall at or below the value 16211. Another way of stating this is that we have a 0.995 probability of choosing at random a value that is less than or equal to 16211 from an F random variable with 1 and 1 degrees of freedom.

For the inverse CDF, see [ReturnF\(?, 1, 1\) \[p. 415\]](#). To generate random F data, see [RandomF\(1, 1\) \[p. 407\]](#).

ProbNormal(?, 0, 1)

ProbNormal(x , *mean*, *stdv*) computes the cumulative distribution function at x of a normal random variable with mean *mean* and standard deviation *stdv*. The arguments may be variables or constants; results are computed casewise. The *mean* and *stdv* are 0 and 1 by default, but you may supply other values. Missing values in any argument propagate missing values.

ProbNormal(A, 0, 1)

ReturnNormal("CDF Normal", 0, 1)

A	CDF Normal	InvCDF Normal
-4	.00003	-3.99958
-3	.00135	-2.99997
•	•	•
0	.50000	.00007
1	.84134	.99982
5	1.00000	4.99923

A cumulative distribution function computes the probability that a random value from that distribution falls at or below a value x you provide. In other words, a CDF returns the proportion of the distribution that is less than or equal to x . CDF functions are useful for creating your own statistical tests, e.g., type I errors.

In the example above, we computed the CDF at A for a normal random variable with mean 0 and standard deviation 1; then we computed the inverse CDF by applying ReturnNormal to the CDF Normal values. Applying the inverse CDF to the CDF returned (almost) the original values from A. We can interpret the results for the second-to-last case, for example, as follows: 84% of the values of a normal random variable fall at or below the value 1; since 1 is the standard deviation, this is not surprising. Another way of stating this is that we have a 0.84 probability of choosing at random a value that is less than or equal to 1 from a Normal random variable with mean 0 and standard deviation 1.

For the inverse CDF, see [ReturnNormal\(?, 0, 1\) \[p. 415\]](#). To generate random Normal data, see [RandomNormal\(0, 1\) \[p. 410\]](#).

Probt(?, 1)

Probt(*x*, *df*) computes the cumulative distribution function at *x* of a *t* random variable with *df* degrees of freedom. The arguments may be variables or constants, and *df* must be a positive integer; results are computed casewise. Missing values in either argument propagate missing values.

```
Probt(A, 1)
ReturnT("CDF t", 1)
```

A	CDF t	InvCDF t
-4	.078	-4
-3	.102	-3
•	•	•
0	.500	0
1	.750	1
5	.937	5

A cumulative distribution function computes the probability that a random value from that distribution falls at or below a value *x* you provide. In other words, a CDF returns the proportion of the distribution that is less than or equal to *x*. CDF functions are useful for creating your own statistical tests, e.g., to compute *p* for a *t*-test.

In the example above, we computed the CDF at A for a *t* random variable with 1 degree of freedom; then we computed the inverse CDF by applying ReturnT to the CDF *t* values. Applying the inverse CDF to the CDF returned the original values from A. We can interpret the results for the last case, for example, as follows: 93.7% of the values of a *t* random variable fall at or below the value 5. Another way of stating this is that we have a 0.937 probability of choosing at random a value that is less than or equal to 5 from a *t* random variable with 1 degree of freedom.

For the inverse CDF, see [ReturnT\(?, 1\)](#) [p. 416]. To generate random *t* data, see [RandomT\(1\)](#) [p. 411].

QuadraticSeries(1, 0, 1)

QuadraticSeries(*a*, *b*, *c*) generates a series of values equal to $a + bx + cx^2$, where *x* is one less than the row number. By default, the arguments are 1, 0, 1, but you may specify any constants. The function works columnwise; results differ from row to row.

```
QuadraticSeries(1, 0, 1)
```

	Quad-1,0,1
1	1
2	2
3	5
4	10
5	17
6	26
7	37
8	50
9	65
10	82

In each row i of the new variable, the quantity $a + bx + cx^2$ is evaluated for $x = i - 1$ and that result is the value for the row. For example, the fourth row above is computed by $1 + 0 \times 3 + 1 \times 3^2 = 1 + 0 + 9 = 10$ because we used the default values $a=1$, $b=0$, $c=1$.

See also [CubicSeries\(1, 0, 0, 1\)](#) [p. 367] and [QuarticSeries\(1, 0, 0, 0, 1\)](#) [p. 405].

QuarticSeries(1, 0, 0, 0, 1)

QuarticSeries(a, b, c, d, e) generates a series of values equal to $a + bx + cx^2 + dx^3 + ex^4$, where x is one less than the row number. By default, the arguments are 1, 0, 0, 0, 1, but you may specify any constants. The function works columnwise; results differ from row to row.

QuarticSeries(1, 0, 0, 0, 1)

	Quar-1,0,0,0,1
1	1
2	2
3	17
4	82
5	257
6	626
7	1297
8	2402
9	4097
10	6562

In each row i of the new variable, the quantity $a + bx + cx^2 + dx^3 + ex^4$ is evaluated for $x = i - 1$ and that result is the value for the row. For example, the fourth row above is computed by $1 + 0 \times 3 + 0 \times 3^2 + 0 \times 3^3 + 1 \times 3^4 = 1 + 0 + 0 + 0 + 81 = 82$, because we used the default values $a=1$, $b=0$, $c=0$, $d=0$, $e=1$.

See also [CubicSeries\(1, 0, 0, 1\)](#) [p. 367] and [QuadraticSeries\(1, 0, 1\)](#) [p. 404].

RadToDeg(?)

RadToDeg(var) converts angle measurements in the variable (or constant) you specify from radians to degrees. Missing values propagate missing values. The function works casewise.

RadToDeg(Radians)

Radians π	Degrees	Radians
zero	0	0.000
$\pi/6$	30	.524
$\pi/3$	60	1.047
$\pi/2$	90	1.571
$2\pi/3$	120	2.094
$5\pi/6$	150	2.618
π	180	3.142
$7\pi/6$	210	3.665
$4\pi/3$	240	4.189
$3\pi/2$	270	4.712
$5\pi/3$	300	5.236
$11\pi/6$	330	5.760
2π	360	6.283

StatView's trigonometric functions work with measurements in radians, so RadToDeg conversions are necessary if you prefer to interpret results or do further analyses using measurements in degrees. A circle has 360 degrees or 2π radians ($2 \times 3.1416... = 6.2832... \text{ radians}$).

Above, Radians π is an informative variable entered by hand to make the Radians values easier to read.

To convert degrees to radians, use [DegToRad\(?\)](#) [p. 372]. StatView's trigonometric functions are [Sin\(?\)](#) [p. 419], [Cos\(?\)](#) [p. 363], [Tan\(?\)](#) [p. 426], [Sec\(?\)](#) [p. 418], [Csc\(?\)](#) [p. 366], [Cot\(?\)](#) [p. 364], [ArcSin\(?\)](#) [p. 353], [ArcCos\(?\)](#) [p. 349], [ArcTan\(?\)](#) [p. 355], [ArcSec\(?\)](#) [p. 352], [ArcCsc\(?\)](#) [p. 351], and [ArcCot\(?\)](#) [p. 350].

RandomBeta(1, 1)

RandomBeta(p, q) generates a series of random numbers from the beta distribution with parameters p and q . You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants; p and q are 1 by default. The function works columnwise; results differ from row to row.

RandomBeta(3, 3)

RandBeta-3,3
.136
.401
.228
.645
.686
.428
.352
.383
.423
.814

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomBinomial(?, ?)

RandomBinomial(n, p) generates a series of random numbers from the binomial distribution with count n and probability p . You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants. The function works columnwise; results differ from row to row.

RandomBinomial(5, 0.5)

RandBinom
1
3
3
3
2
3
3
2
4
2

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

The Bernoulli distribution is a special case of the binomial distribution. For random Bernoulli numbers, set n to 1.

RandomChiSquare(1)

RandomChiSquare(df) generates a series of random numbers from the chi-square distribution with df degrees of freedom. You may supply a random number generator seed, if you wish, as an optional second argument. Both arguments must be constant, and df must be a positive integer; the default is 1 degree of freedom. The function works columnwise; results differ from row to row.

RandomChiSquare(1)

RandChi-1
2.373
.508
.028
.494
.337
.951
1.593
1.354
.315
.302

RandomExponential(1)

RandomExponential(t) generates a series of random numbers from the exponential distribution with rate t . You may supply a random number generator seed, if you wish, as an optional second argument. Both arguments must be constant, and df should be a positive integer; the default is 1 degree of freedom. The function works columnwise; results differ from row to row.

RandomExponential(2)

RandExp-2
.490
.066
.671
1.035
.044
.293
.066
1.148
.340
.496

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomF(1, 1)

RandomF(df , $df2$) generates a series of random numbers from the F distribution with df degrees of freedom in the numerator and $df2$ degrees of freedom in the denominator. You may

supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants, and degrees of freedom must be positive integers. The function works columnwise.

RandomF(2, 3)

RandF-2,3
.588
1.736
102.028
1.763
1.385
.225
.677
1.438
21.405
.477

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomGamma(1)

RandomGamma(*t*) generates a series of random numbers from the gamma distribution of order *t*. You may supply a random number generator seed, if you wish, as an optional second argument. All arguments must be constants; the default is order 1. The function works columnwise; results differ from row to row.

RandomGamma(5)

RandGam-5
2.926
.662
2.858
2.861
1.239
.579
.865
1.160
2.400
2.081

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomGaussian(0, 1)

RandomGaussian(*mean*, *stdv*) generates a series of random numbers from the normal, or Gaussian, distribution with mean *mean* and standard deviation *stdv*. You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants; the defaults are mean 0 and standard deviation 1. The function works columnwise; results differ from row to row. RandomGaussian is synonymous with RandomNormal.

RandomGaussian(0, 1)

RandNorm=0,1
-.280
.270
-.151
1.256
1.467
2.291
-.148
.773
2.061
-1.092

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomInclusion(?)

RandomInclusion(p) does random sampling from your dataset by including any row at probability p . You may supply a random number generator seed, if you wish, as an optional second argument. All arguments must be constants. The function works columnwise; results differ from row to row.

This function is not used with Formula, Random Numbers, Series, or Create Criteria. Rather, it works in the background when you select Random... from the Criteria pop-up menu. The only time you ever see this function is when you Edit Criteria and select a criterion previously created with Random... Such a criterion might be called “60% Rows Included.” So, if you need to do something fancy with a random selection:

- Select Random... from the Criteria pop-up menu in the dataset window
- Specify a probability (type 50 for 50%; 0.5 means 0.5%) and click OK
- From the Manage menu, select Edit/Apply Criteria
- Select the “ p % Rows Included” criterion and click Edit
- Edit the complex criteria definition

When rows are excluded, their row numbers are dimmed. Also, the Criteria pop-menu reflects the inclusion in effect. Any analyses in the View window are then confined to those cases that remain, which is noted in its title.

Descriptive Statistics

Inclusion criteria: 50% Rows Included from Lipid Data

Mean	199.718
Std. Dev.	31.981
Std. Error	5.121
Count	39
Minimum	115.000
Maximum	267.000
# Missing	0

		Lipid				
		Criteria: 50% Rows Included				

You may also do inclusion and exclusion by manually double-clicking row numbers, or by selecting rows and using the Include and Exclude commands from the Manage menu; see [“Include and exclude rows,” p. 108 of Using StatView.](#)

If, instead, you want to include at random some percentage p of all rows, create a random variable with the distribution of your choice, sort on that variable, and include the first p percent of the rows. (If you need to avoid losing the original sort order of the dataset, see [RowNumber](#) [p. 417].)

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random inclusion in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomNormal(0, 1)

RandomNormal(*mean*, *stdv*) generates a series of random numbers from the normal, or Gaussian, distribution with mean *mean* and standard deviation *stdv*. You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants; the defaults are mean 0 and standard deviation 1. The function works columnwise; results differ from row to row. RandomNormal is synonymous with RandomGaussian.

RandomNormal(0, 1)

RandNorm-0,1
- .280
.270
-.151
1.256
1.467
2.291
-.148
.773
2.061
-1.092

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomPoisson(1)

RandomPoisson(*mean*) generates a series of random numbers from the Poisson distribution with mean *mean*. You may supply a random number generator seed, if you wish, as an optional second argument. Both arguments must be constants, and *mean* must be positive. Poisson random variables take integer values. The function works columnwise; results differ from row to row.

RandomPoisson(1)

RandPois-1
4
3
2
3
4
5
3
5
2
4

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomT(1)

RandomT(*df*) generates a series of random numbers from Student's *t* distribution with *df* degrees of freedom. You may supply a random number generator seed, if you wish, as an optional second argument. Both arguments must be constants, and *df* must be a positive integer; the default for *df* is 1. The function works columnwise; results differ from row to row.

RandomT(3)

RandT-3
-.456
1.712
1.732
-.323
-1.001
.197
-2.527
-3.305
.242
-.719

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomUniform(0, 1)

RandomUniform(*n*, *m*) generates a series of uniform random numbers from the interval between *n* and *m*, inclusive. You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants; the default interval is (0,1). The function works columnwise; results differ from row to row.

RandomUniform(0, 2)

RandUnif-0,2
.324
.963
1.463
.744
1.704
1.128
1.582
.412
.941
.698

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomUniformInteger(?, ?)

RandomUniformInteger(*n*, *m*) generates a series of uniform random integers from the interval between *n* and *m*, inclusive. You may supply a random number generator seed, if you wish, as an optional third argument. All arguments must be constants, and *n* and *m* should be integers. The function works columnwise; results differ from row to row.

RandomUniformInteger(40, 50)

RanUnifInt~40,50	
	50
	43
	43
	43
	45
	40
	40
	48
	47
	40

Specifying a random number generator seed ensures consistent results. For example, if you want to assign homework involving random numbers in which all students should get the same results, direct your students to specify a certain number as the seed.

RandomUniformInteger can produce repeated values. If you need a variable of *unique* random integers, create a variable of consecutive integers with RowNumber or LinearSeries, create a random variable with RandomNormal, then sort on the random variable. (Be sure to use a static formula so that the random normal values don't update.) You now have a column of unique integers in random order.

RowNumber

RandomNormal(0,1)

	RanInts	RanNorm
1	4	-.243
2	6	-.100
3	10	.346
4	5	.782
5	9	.788
6	1	.897
7	2	.904
8	7	1.068
9	3	1.643
10	8	1.686

If you want to avoid sorting your dataset, create these variables in a separate dataset, Copy the random integers, and Paste them into your main dataset. Remember, StatView lets you have multiple datasets open at once. Or, see [RowNumber \[p. 417\]](#) for “unsorting” tips.

Range(?, AllRows)

Range(*var*, AllRows) computes the range of the variable you specify; by default, Range is based on AllRows, but you may instead specify OnlyIncluded Rows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Range(A, AllRows)

	A	Range of A
Minimum:	-4	9.0
Maximum:	5	9.0
Range:	9	0.0
1	-4	9.0
2	-3	9.0
3	•	9.0
4	0	9.0
5	1	9.0
6	5	9.0

Range is the most basic measure of spread, equal to the difference between the maximum and minimum values. All three statistics are also shown in the summary pane for the variable.

See also [Minimum\(?, AllRows\)](#) [p. 389] and [Maximum\(?, AllRows\)](#) [p. 387] and “[Descriptive Statistics](#),” p. 1.

Rank(?, AllRows)

Rank(*var*, AllRows) computes the rank of each value in the variable you specify; by default, Rank uses AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Tied values have tied ranks (as shown below). Missing values propagate missing values; they cannot be ranked. The function works columnwise; results differ from row to row.

Rank(A, AllRows)

A	RankOfA
-4.000	2.0
-5.000	1.0
•	•
•	•
1.000	3.0
5.000	4.5
5.000	4.5
6.000	6.0

Ranks are the row numbers that would result if you sorted all the values in ascending order (least to greatest). Tied ranks are averaged, as seen above.

Many nonparametric statistics, such as sign test and Spearman rank order correlation, are based on rank values rather than raw values. It is often helpful to examine ranks alongside raw values when interpreting the results of such statistics; see “[Nonparametrics](#),” p. 119.

Remainder(?, ?)

Remainder(*var1*, *var2*) gives the remainder result of *var* divided by *var2*. Both arguments may be constants or variables. Missing values in either argument or division by zero propagates missing values. The function works casewise.

Remainder(A, B)

A	B	Div(A,B)	Mod(A,B)	Rem(A,B)
-5	2	-2	-1	-1.0
-7	2	-3	-1	1.0
-12	7	-1	-5	2.0
-3	-2	1	-1	1.0
•	4	•	•	•
0	3	0	0	0.0
1	0	•	•	•
17	4	4	1	1.0

Ordinarily, division is computed to as many decimal places as necessary for an exact answer, within the limits of the variable’s precision. For instance, 5/3 is 1.6666... (an infinite series of 6s after the decimals). Remainder(5,3) stops dividing when the quotient reaches the decimal point and then records the remainder, or the leftover part—this is the way children learn long division:

$$\begin{array}{r} 1 \text{ r } 2 \\ 3 \overline{)5} \\ \underline{3} \\ 2 \end{array}$$

Children are taught to divide until the amount at the bottom is smaller than the divisor, and then write that leftover part as “remainder 2.” This casual definition is sufficient for positive numbers, but for negative numbers, a more precise definition is needed. Formally, Remainder(*var1*, *var2*) is defined as *var1*–(Round(*var1*/*var2*))**var2*.

The [Mod\(?, ?\)](#) [p. 390] function is the same as Remainder for positive arguments, but for negative arguments, Remainder and Mod are different. The formal definition of Mod(*var1*, *var2*) is *var1*–(Trunc(*var1*/*var2*))**var2*. See [Trunc\(?\)](#) [p. 429] and [Round\(?\)](#) [p. 416] for details; briefly, rounding goes up or down to the nearest integer, whereas truncation deletes digits after the decimal. Finally, see [Div\(?, ?\)](#) [p. 374] for the integer part of a quotient.

ReturnChiSquare(?, ?)

ReturnChiSquare(*alpha*, *df*) computes the inverse cumulative distribution function at probability *alpha* of a chi-square random variable with *df* degrees of freedom. The arguments may be variables or constants; results are computed casewise. The values for *alpha* should be between 0 and 1, and *df* should be positive integers; for illegal values or missing values in either argument, missing values are propagated.

```
ProbChiSquare(A, I)
ReturnChiSquare("CDF chi-sq", I)
```

A	CDF chi-sq	InvCDF chi-sq
-4	•	•
-3	•	•
•	•	•
0	•	•
1	.683	1
5	.975	5

An inverse CDF gives the critical value at or below which the proportion *alpha* of the distribution lies. In other words, it returns the value *x* at which we have an *alpha* probability of choosing at random from the distribution a value less than or equal to *x*.

For the CDF, see [ProbChiSquare\(?, 1\) \[p. 402\]](#). To generate random chi-square data, see [RandomChiSquare\(1\) \[p. 407\]](#).

ReturnF(?, 1, 1)

ReturnF(*alpha*, *df*, *df2*) computes the inverse cumulative distribution function at probability *alpha* of an F random variable with *df* degrees of freedom in the numerator and *df2* degrees of freedom in the denominator. The arguments may be variables or constants; results are computed casewise. The values for *alpha* should be between 0 and 1, and *df* and *df2* should be positive integers; for illegal values or missing values in any argument, missing values are propagated.

ProbF(A, 1, 1)

ReturnF("CDF F", 1, 1)

A	CDF F	InvCDF F
-4.0	•	•
39.9	.900	39.9
647.8	.975	647.8
4052.0	.990	4052.0
161.4	.950	161.4
16211.0	.995	16211.0

An inverse CDF gives the critical value at or below which the proportion *alpha* of the distribution lies. In other words, it returns the value *x* at which we have an *alpha* probability of choosing at random from the distribution a value less than or equal to *x*.

For the CDF, see [ProbF\(?, 1, 1\) \[p. 402\]](#). To generate random F data, see [RandomF\(1, 1\) \[p. 407\]](#).

ReturnNormal(?, 0, 1)

ReturnNormal(*alpha*, *mean*, *stdv*) computes the inverse cumulative distribution function at probability *alpha* of a normal random variable with mean *mean* and standard deviation *stdv*. The arguments may be variables or constants; results are computed casewise. The values for *alpha* should be between 0 and 1; *mean* is 0 and *stdv* is 1 by default, but you may specify other values. Missing values in any argument propagate missing values.

ProbNormal(A, 0, 1)

ReturnNormal("CDF Normal", 0, 1)

A	CDF Normal	InvCDF Normal
-4	.00003	-3.99958
-3	.00135	-2.99997
•	•	•
0	.50000	.00007
1	.84134	.99982
5	1.00000	4.99923

An inverse CDF gives the critical value at or below which the proportion *alpha* of the distribution lies. In other words, it returns the value *x* at which we have an *alpha* probability of choosing at random from the distribution a value less than or equal to *x*.

Suppose you want to know if a variable A is normally distributed. You may build a normality test using the ReturnNormal function. First, generate an “ideal” normal variable that has the same mean and standard deviation as your variable:

```
ReturnNormal(Rank(A, AllRows)/Count(A, AllRows), Mean(A, AllRows),
StandardDeviation(A, AllRows))
```

Next, combine A and the new variable in a compact variable with two levels. You might name these “actual” and “ideal.” Finally, do a Kolmogorov-Smirnov test on the compact variable (K-S is found in the analysis browser under Nonparametrics). If p is significant, you may conclude that the variables are from different distributions—in other words, that A is not normally distributed. The QC Analyses/K-S normality test template performs this test.

For the CDF, see [ProbNormal\(?, 0, 1\) \[p. 403\]](#). To generate random normal data, see [RandomNormal\(0, 1\) \[p. 410\]](#).

ReturnT(?, 1)

ReturnT(α , df) computes the inverse cumulative distribution function at probability α of a t random variable with df degrees of freedom. The arguments may be variables or constants; results are computed casewise. The values for α should be between 0 and 1, and df should be a positive integer. Illegal or missing values in either argument propagate missing values.

```
Probt(A, 1)
```

```
ReturnT("CDF t", 1)
```

A	CDF t	InvCDF t
-4	.078	-4
-3	.102	-3
•	•	•
0	.500	0
1	.750	1
5	.937	5

An inverse CDF gives the critical value at or below which the proportion α of the distribution lies. In other words, it returns the value x at which we have an α probability of choosing at random from the distribution a value less than or equal to x .

For the CDF, see [Probt\(?, 1\) \[p. 404\]](#). To generate random t data, see [RandomT\(1\) \[p. 411\]](#).

Round(?)

Round(var) rounds each value of the variable or constant you specify to the nearest integer. Numbers with fractional parts greater than 0.5 are rounded up or down to the nearest *even* integer. Numbers with fractional parts exactly equal to 0.5 are rounded up or down to the nearest *even* integer. Missing values are propagated. The function works casewise.

```
Round(A)
```

A	RoundedA	TruncatedA	Floor of A	Ceil of A
-1.200	-1.000	-1.000	-2.000	-1.000
-3.915	-4.000	-3.000	-4.000	-3.000
•	•	•	•	•
.051	0.000	0.000	0.000	1.000
1.238	1.000	1.000	1.000	2.000
4.800	5.000	4.000	4.000	5.000

Rounding removes decimal portions of numbers by increasing to the next greater whole number whenever the fraction is greater than or equal to one-half, and by decreasing to the next smaller whole number whenever the fraction is less than one-half. As do all computations, Round computes from the actual stored values of numbers rather than the displayed values. For example, the value 3.495 displays with one decimal place as 3.5, but it rounds to 3, not 4.

This example shows how numbers exactly halfway between integers are rounded either up or down to the nearest even integer.

Round(A)

A	Round(A)
-4.5	-4
-3.5	-4
-2.5	-2
-1.5	-2
-.5	0
.5	0
1.5	2
2.5	2
3.5	4
4.5	4

Negative numbers round the same as positive numbers—for example, 3.4 rounds to 3, and -3.4 rounds to -3 (not -4). Remember, for negative numbers “greater” and “lesser” can seem backwards: -3.4 is greater than -3.6, and -3 is greater than -4.

Related functions are [Trunc\(?\)](#) [p. 429], [Floor\(?\)](#) [p. 380], and [Ceil\(?\)](#) [p. 359]. Whereas the behavior of Round differs according to the size of the fractional part, Trunc, Floor, and Ceil all ignore the size of the fractional part. Trunc truncates all decimal portions—it chops off the digits after the decimal, regardless of the size of that fractional value. (Thus, truncation varies by sign: it rounds negative numbers to the next *greater* integer and positive numbers to the next *lesser* integer.) Floor converts all values to the next lesser integer regardless of sign and the size of the fractional part. Ceil (short for ceiling) converts all values to the next greater integer regardless of sign and the size of the fractional part. (Thus, the floor of -1.2 is the next *lesser* integer, -2; the ceiling of -1.2 is the next *greater* integer, -1. The floor of +1.2 is 1; the ceiling, 2.)

RowNumber

RowNumber shows the number of each row. RowNumber takes no arguments. If you provide one by mistake, e.g., RowNumber(A), it is interpreted as multiplication. The function works columnwise; results differ from row to row.

RowNumber

A – RowNumber

	A	#	A minus #
1	-4	1	-5
2	-5	2	-7
3	•	3	•
4	•	4	•
5	1	5	-4
6	5	6	-1

RowNumber merely fills a new variable with the same information seen in the first column of the dataset window. RowNumber is usually used in combination with other functions, as in A minus #, which subtracts the value of # (the current row number) from each case of A.

One common use for RowNumber is in “unsorting” a dataset. The Sort command in the Manage menu lets you sort rows of a dataset according to one or more key variables. If you might want to return to the original dataset order, before sorting you should create an index variable containing the current RowNumber values. (Do this with a static formula or else change the variable to user entered so that when you sort, the RowNumber values are not updated.) Sort the dataset. When you are ready to return to the original order, Sort on the index variable you created.

Sec(?)

Sec(*var*) returns the secant of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works case-wise.

Sec(Radians)

Radians π	Radians	Secant
zero	0.000	1.000
$\pi/6$.524	1.155
$\pi/3$	1.047	2.000
$\pi/2$	1.571	•
$2\pi/3$	2.094	-2.000
$5\pi/6$	2.618	-1.155
π	3.142	-1.000
$7\pi/6$	3.665	-1.155
$4\pi/3$	4.189	-2.000
$3\pi/2$	4.712	•
$5\pi/3$	5.236	2.000
$11\pi/6$	5.760	1.155
2π	6.283	1.000

The secant of an angle in a right triangle is the ratio of the length of the hypotenuse to the length of the leg adjacent to the angle. Recall that the cosine of an angle in a right triangle is the ratio of the length of the leg adjacent to the angle to the length of the hypotenuse. Therefore, the secant is the reciprocal of the cosine:

$$\sec x = \frac{1}{\cos x}$$

As the angle (Radians) approaches $\pi/2$ and $3\pi/2$ (and so on), cosine approaches zero, and thus secant approaches plus or minus infinity. Secants are undefined at these points, so Sec produces missing values. (On some platforms, differences in the numerics environments may produce extreme values rather than missing values.)

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

Second(?)

Second(*date*) returns the second number (0–59) of the *date* specified. The *date* argument may be a variable or a constant. (Remember, all date/time values are an exact second of an exact day, and unspecified times are assumed to be exactly midnight.) The function works casewise.

Second("Some times")

Some times	Hour	Minute	Second
01:03:59 AM	1	3	59
02:05:00 AM	2	5	0
03:06:01 AM	3	6	1
04:07:02 AM	4	7	2
05:08:03 AM	5	8	3
06:09:04 AM	6	9	4

Sin(?)

Sin(*var*) returns the sine of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works casewise.

Sin(Radians)

Radians π	Radians	Sine
zero	0.000	0.000
$\pi/6$.524	.500
$\pi/3$	1.047	.866
$\pi/2$	1.571	1.000
$2\pi/3$	2.094	.866
$5\pi/6$	2.618	.500
π	3.142	0.000
$7\pi/6$	3.665	-.500
$4\pi/3$	4.189	-.866
$3\pi/2$	4.712	-1.000
$5\pi/3$	5.236	-.866
$11\pi/6$	5.760	-.500
2π	6.283	0.000

Sines, cosines, and tangents relate angles to the coordinates of points in planes. The sine of an angle in a right triangle is the ratio of the length of the leg opposite the angle to the length of the hypotenuse.

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

Sinh(?)

Sinh(*var*) returns the hyperbolic sine of a variable or constant. Missing values propagate missing values. The function works casewise.

Sinh(x)

x	Sinh
-5	-74.203
-4	-27.290
-3	-10.018
-2	-3.627
-1	-1.175
0	0.000
1	1.175
2	3.627
3	10.018
4	27.290
5	74.203

The hyperbolic functions (sinh, cosh, and tanh, sometimes pronounced “sinch, cosh, and tanch”) are analogous to the trigonometric functions sine, cosine, and tangent. They are special combinations of the exponential functions e^x and e^{-x} and bear a relationship to the unit hyperbola that is analogous to the trig functions’ relationship to the unit circle.

The hyperbolic sine is defined by

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

and like sine, $\sinh(x)$ has value 0 at $x=0$. Sinh is defined for all real numbers and ranges from plus to minus infinity.

Sqrt(?)

Sqrt(*var*) produces the positive square root of a variable or constant. Missing values propagate missing values. The function works casewise.

Sqrt(Abs(A))
Sqrt(B)

	A	B	Sqrt(A)	Sqrt(B)
1	-4	5	2.000	2.236
2	-3	-1	1.732	●
3	●	4	●	2.000
4	0	●	0.000	●
5	1	0	1.000	0.000
6	5	4	2.236	2.000

Square roots of negative numbers are undefined for real numbers, so StatView returns missing values for negative arguments. If you are studying magnitude of a variable without regard to its sign, it may be appropriate to use absolute values, as shown above, to prevent missing values from negative arguments.

Square root produces by definition positive square roots. That is, 64 is equal to not only 8^2 but also -8^2 , but Sqrt(64) is only 8.

StandardDeviation(? , AllRows)

StandardDeviation(*var*, AllRows) computes the standard deviation of the variable you specify; by default, StandardDeviation is based on AllRows of the variable, but you may instead spec-

ify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

StandardDeviation(A, AllRows)

	A	sigma
Mean:	-750	4.646
Std. Deviation:	4.646	0.000
1	-4	4.646
2	-5	4.646
3	•	4.646
4	•	4.646
5	1.000	4.646
6	5.000	4.646

Standard deviation is a measure of variability; unlike variance, it is expressed in the same unit of measurement as the original variable. Standard deviation is formally defined as the square root of variance. Normally distributed data have 95% of the observations falling within 1.96 standard deviations to either side of the mean. Standard deviation is also shown in the summary pane for each variable.

StatView uses $n-1$ in the denominator, which is preferred for sample standard deviation. For population standard deviation, n is often preferred. If you want n instead of $n-1$, you can compute your own standard deviation:

$$\text{Sqrt}((\text{SumOfSquares}(\text{A}, \text{AllRows}) - \text{Count}(\text{A}, \text{AllRows}) * \text{Mean}(\text{A}, \text{AllRows})^2) / \text{Count}(\text{A}, \text{AllRows}))$$

[Variance\(?, AllRows\) \[p. 430\]](#) function also uses $n-1$; however, in the Descriptive Statistics analysis (see [“Descriptive Statistics,” p. 1](#)), you may compute Variance with either n or $n-1$, so you could use a formula to take the square root of that result. Still another option would be to use a formula to multiply Variance by its usual denominator of $n-1$, divide by the one you want, n , and then take the square root of that:

$$\text{Sqrt}(\text{Variance}(\text{A}, \text{AllRows}) * (\text{Count}(\text{A}, \text{AllRows}) - 1) / \text{Count}(\text{A}, \text{AllRows}))$$

StandardError(?, AllRows)

StandardError(*var*, AllRows) computes the standard error of the mean of the variable you specify; by default, StandardError is based on AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

StandardError(A, AllRows)

	A	StdErr of A
Mean:	-750	0.000
Std. Deviation:	4.646	0.000
Std. Error:	2.323	0.000
1	-4.000	2.323
2	-5.000	2.323
3	•	2.323
4	•	2.323
5	1.000	2.323
6	5.000	2.323

The standard error of the mean is a measure of the variability of the mean. Sometimes called SEM, it is computed from the standard deviation (above, 4.646) divided by the square root of

the count (2); the result, 2.323, is also shown in the summary pane for the variable. The SEM shows how much variability you should expect among sample means if you take multiple samples from the same population.

Related functions are [Mean\(?, AllRows\)](#) [p. 388], [StandardDeviation\(?, AllRows\)](#) [p. 420], and [Count\(?, AllRows\)](#) [p. 365].

StandardScores(?, AllRows)

StandardScores(*var*, AllRows) standardizes each case of the variable you specify; by default, StandardScores is based on AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values propagate missing values. The function works columnwise; results differ from row to row.

StandardScores(A, AllRows)

A	Z scores
-4.000	-.700
-5.000	-.915
•	•
•	•
1.000	.377
5.000	1.238

Standard scores, sometimes called standardized values or *z*-scores, are obtained by dividing the difference between each value and the mean by the standard deviation. Standardizing a variable gives it mean 0 and standard deviation 1 and makes it easier to compare variables of dissimilar magnitude.

You could also standardize data by building a formula with [Mean\(?, AllRows\)](#) [p. 388] and [StandardDeviation\(?, AllRows\)](#) [p. 420] functions:

$(A - \text{Mean}(A, \text{AllRows})) / \text{StandardDeviation}(A, \text{AllRows})$

Substring(?, ?, ?)

Substring(*text*, *n*, *m*) reads the *text* you specify and returns the *m*-character substring starting at the *n*th character. If the *text* is fewer than *n+m*-1 characters long, it returns fewer than *m* characters. The source *text* may be either a variable or a constant, and *n* and *m* must be positive integers. If you supply a variable as the *text* argument, Substring uses its exact values in the current format's display; changing formats can change results. If you supply a constant, you must enclose it in quotation marks. If *n* is negative or greater than the length of *text*, missing values result. An optional fourth argument (1 or 2) specifies whether to handle text values as single-byte or double-byte strings; see below. The function works casewise.

Substring(Model, 3, 6)

Model	Sub 3,6
Acura Integra	ura In
Acura Legend V6	ura Le
Audi 100	di 100
Audi 80	di 80
Audi 90	di 90
BMW 325i	w 325i

Substring is used to extract part of a text value. Usually, the *text* argument is a string variable, although the function also works with other variable types.

Above, we use substring to extract six characters from the middle of each Model value in Car Data, starting from the third character and moving six to the right. Each letter, number, space, and symbol in a string counts as a character. The fourth and fifth rows have values fewer than six characters long, since the Model values were fewer than eight characters long. (Change Model from class informative to nominal so you can use it in a formula, and change the formula variable to have type string.)

Substring is usually used in combination with other text functions such as Len and Find. For example, we can use Find to locate the position of the first space in Model names, then read the next 99 characters after that space, thus extracting all characters after the first word. (The “1, false” arguments to Find specify that the search should start on the first character of each Model value, and that case-sensitivity should be “off.”) This gives us model names without makes. (We’ve scrolled down to find some multi-word models.)

Substring(Model, Find(Model, " ", 1, false)+1, 99)

	Model	Model only
20	Chevrolet Caprice V8	Caprice V8
21	Chevrolet Cavalier	Cavalier
22	Chevrolet Corvette V8	Corvette V8
23	Chevrolet Lumina	Lumina
24	Chevrolet Lumina APV V6	Lumina APV V6
25	Chrysler Imperial V6	Imperial V6
26	Chrysler Le Baron Coupe	Le Baron Coupe

Specifying 99 as the number of characters to read is a brute-force way to read to the end of the string. No values are actually that long—99 is just an arbitrarily large number.

You may include an optional fourth argument for specifying whether to handle text values as single-byte or double-byte strings. Substring assumes a fourth argument 1 for single-byte strings (English, German, French, Spanish, etc. all use single-byte characters); specify 2 to use Substring with strings containing double-byte characters, such as Japanese, Chinese, or Arabic characters. See [Find\(?, ?, ?, false\) \[p. 379\]](#) and [Len\(?\) \[p. 383\]](#) for examples handling double- and single-byte characters.

Sum(?, ...)

Sum(*var*, *var2*, ...) does casewise addition of the variables or constants you specify. Summing arguments is a shorthand equivalent to linking arguments with plus (+) signs. Missing values propagate missing values.

Sum(A, B)

Sum(A, B, 7)

A + B

A + B + 7

A	B	Sum(A,B)	Sum(A,B,7)	A+B	A+B+7
-4	5	1	8	1	8
-3	-2	-5	2	-5	2
•	4	•	•	•	•
0	•	•	•	•	•
1	0	1	8	1	8
5	4	9	16	9	16

For casewise addition in which missing values are ignored, use [SumIgnoreMissing\(?, ...\)](#) [p. 424]. For columnwise (vertical) addition, see [Sum\(?, ...\)](#) [p. 423]; Sum in the data attribute pane summary statistics; [CumSum\(?\)](#) [p. 368], which computes cumulative sums of the rows of a variable; and [SumOfColumn\(?, AllRows\)](#) [p. 424], which fills a new variable with a single sum.

SumIgnoreMissing(?, ...)

SumIgnoreMissing(*var*, *var2*, ...) does casewise addition of the variables or constants you specify. Missing values are ignored.

Sum(A, B)
SumIgnoreMissing(A, B)

A	B	Sum(A,B)	SumIgnoreMissing
-4	5	1	1
-3	-1	-4	-4
•	4	•	4
•	•	•	•
1	0	1	1
5	4	9	9

SumIgnoreMissing is the same as Sum except that missing values are ignored unless every variable is missing for a case, as seen in the fourth case above. For Sum, a missing value in any variable produces a missing value in the new variable, as seen in the third and fourth cases above.

For columnwise addition, see [Sum\(?, ...\)](#) [p. 423]; Sum in the data attribute pane summary statistics; [CumSum\(?\)](#) [p. 368], which computes a cumulative sum for each row; or [SumOfColumn\(?, AllRows\)](#) [p. 424], which fills a new variable with a single sum.

SumOfColumn(?, AllRows)

SumOfColumn(*var*, AllRows) adds the values of the variable you specify to produce a single sum in a new variable. (This sum is also shown in the summary pane for the variable.) By default, AllRows are used in the calculations, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise; results differ from row to row.

SumOfColumn(A, OnlyIncludedRows)

	A	SumOfCol of A	CumSum of A
1	-1	8	-1
2	-2	8	-3
3	•	8	•
4	3	8	0
5	2	8	2
6	3	8	5
7	4	8	9
8	5	8	14
9	-6	8	8
	7	8	15

The row number for row 10 is dimmed, meaning it has been excluded. Since $-1 + -2 + 3 + 2 + 3 + 4 + 5 + -6$ is 8, the new variable is 8.

Compare this result with that of the CumSum function. First, CumSum shows the “sum in progress” on each row, whereas SumOfColumn shows only a single answer. Second, CumSum always uses AllRows, so the 7 on the last row is included for a final sum of 15.

For casewise addition, use [?+?](#) [p. 333], [Sum\(?, ...\)](#) [p. 423], or [SumIgnoreMissing\(?, ...\)](#) [p. 424]. Sum adds values for each row on all the variables you specify. SumIgnoreMissing is the same except that missing values are ignored.

SumOfSquares(?, AllRows)

SumOfSquares(*var*, AllRows) adds the squares of the nonmissing values of the variable you specify in the first argument; by default, it uses AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

SumOfSquare(A, AllRows)

	A	SSq
Missing Cens:	2	0
Sum:	-3	402
Sum of Squares:	67	26934
1	-4	67
2	-5	67
3	•	67
4	•	67
5	1	67
6	5	67

Sum of squares is also shown in the summary pane. The computation for A is $(-4)^2 + (-5)^2 + 1^2 + 5^2 = 16 + 25 + 1 + 25 = 67$.

Sum of squares is used in computation of many statistics (and other functions, such as [Variance\(?, AllRows\)](#) [p. 430] and [StandardDeviation\(?, AllRows\)](#) [p. 420]), and you may find occasion to use it in formulas to compute special statistics not provided by StatView’s analyses.

For cumulative sums of squares, use the [CumSumSquares\(?\)](#) [p. 368] function. For a horizontal sum of squares, use a formula such as $A^2 + B^2 + \dots$, or $\text{Sum}(A^2, B^2, \dots)$.

Tan(?)

Tan(*var*) returns the tangent of a variable or constant. The angle measurements in *var* are assumed to be in radians. Missing values propagate missing values. The function works case-wise.

Tan(Radians)

Radians π	Radians	Tangent
zero	0.000	0.000
$\pi/6$.524	.577
$\pi/3$	1.047	1.732
$\pi/2$	1.571	•
$2\pi/3$	2.094	-1.732
$5\pi/6$	2.618	-.577
π	3.142	0.000
$7\pi/6$	3.665	.577
$4\pi/3$	4.189	1.732
$3\pi/2$	4.712	•
$5\pi/3$	5.236	-1.732
$11\pi/6$	5.760	-.577
2π	6.283	0.000

Sines, cosines, and tangents are used to relate angles to the coordinates of points in planes. The tangent of an angle in a right triangle is the ratio of the length of the leg opposite the angle to the length of the leg adjacent to the angle.

Tangents approach plus or minus infinity asymptotically as their arguments approach $\pi/2$, $3\pi/2$, etc. Tangents are undefined at these values, so Tan produces missing values. (On some platforms, differences in the numerics environments may produce extreme values rather than missing values.)

If you have angles measured in degrees, you can convert them to radians with [DegToRad\(?\)](#) [p. 372]. Radians, in turn, can be converted to degrees with [RadToDeg\(?\)](#) [p. 405]. You may specify the value π with [Pi](#) [p. 400].

Tanh(?)

Tanh(*var*) returns the hyperbolic tangent of a variable or constant. Missing values propagate missing values. The function works casewise.

Tanh(x)

x	Tanh
-5	-1.000
-4	-.999
-3	-.995
-2	-.964
-1	-.762
0	0.000
1	.762
2	.964
3	.995
4	.999
5	1.000

The hyperbolic trigonometric functions (sinh, cosh, and tanh, often pronounced “sinch, cosh, and tanch”) are analogous to the trigonometric functions sine, cosine, and tangent. They are

constructed from the functions e^x and e^{-x} and bear a relationship to the unit hyperbola that is analogous to trigonometric functions' relationship to the unit circle.

The hyperbolic tangent is defined by

$$\tanh x = \frac{\sinh x}{\cosh x}$$

and like tangent, $\tanh(x)$ has value 0 at $x=0$. Tanh is defined for all real numbers and ranges from -1 to 1 .

Time(?, ?, ?)

Time (*hour, minute, second*) returns the time specified on the current date. You must change the formula variable to type date/time and choose an appropriate format. The function works casewise.

Time(Hr, Min, Sec)

Hr	Min	Sec	Nice times
1	3	59	01:03:59 AM
2	5	0	02:05:00 AM
3	6	1	03:06:01 AM
4	7	2	04:07:02 AM
5	8	3	05:08:03 AM
6	9	4	06:09:04 AM

The example above shows how to use Time to combine hour, minute, and second values stored in separate columns. Data imported from other programs may have date/time values separated into several numeric-type columns, and Time puts those columns together into date/time values. (Don't forget to change the type of the new variable to date/time and to choose a format you like.)

Other programs store time values as text strings. You can use [Substring\(?, ?, ?\) \[p. 422\]](#) and [Time\(?, ?, ?\) \[p. 427\]](#) to convert these to times:

Time(Substring(Text, 1, 2), Substring(Text, 3, 2), Substring(Text, 5, 2))

Text	Times
010359	01:03:59
020500	02:05:00
030601	03:06:01
040702	04:07:02
050803	05:08:03
060904	06:09:04

You may also build times with formulas such as this one:

Time(LineNumber, LineNumber+3, LineNumber-2)

	Some times
1	01:03:59 AM
2	02:05:00 AM
3	03:06:01 AM
4	04:07:02 AM
5	05:08:03 AM
6	06:09:04 AM
7	07:10:05 AM
8	08:11:06 AM
9	09:12:07 AM
10	10:13:08 AM

This example shows how invalid times are reinterpreted. Consider the first case, where RowNumber-2 would have been -1. This is reinterpreted as:

hour=1, minute=4, second=-1
hour=1, minute=3, second=59

This sort of carrying also happens when hours are greater than 23 or minutes or seconds are greater than 59.

TrimmedMean(?, ?, AllRows)

TrimmedMean(*var*, *p*, AllRows) computes the trimmed mean of *var* using the percentage *p* you specify. The value for *p* must be less than 50 and greater than or equal to zero. By default, TrimmedMean is based on AllRows, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the third argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

TrimmedMean(A, 10, AllRows)

A	Trim10% Mean of A
-4	-.200
-3	-.200
•	-.200
0	-.200
1	-.200
5	-.200

Trimmed mean is a measure of central tendency similar to the mean, except that trimmed mean is based on only the “inner” portion of the data after trimming the top and bottom *p* percent of values. In a variable with 100 values, a 10% trimmed mean discards the ten smallest and the ten largest values, and then computes the average (sum divided by 80) of the values remaining. When *p*=0, the trimmed mean is equal to the mean. As *p* approaches 50, the trimmed mean approaches the median.

Trimmed means offer an advantage over the mean for variables with extreme values on either end. For example, the mean salary of all people living in a neighborhood may be drastically influenced up or down by a few extremely wealthy residents or a few homeless residents with negligible income, but the trimmed mean gives a realistic sense of the average income among the most typical residents.

See also the [Mean\(?, AllRows\)](#) [p. 388], [Median\(?, AllRows\)](#) [p. 388], and [Mode\(?, AllRows\)](#) [p. 390], and “Descriptive Statistics,” p. 1.

Trunc(?)

Trunc(*var*) truncates the fractional portion from the values of the variable you specify. Missing values are propagated. The function works casewise.

Trunc(A)

A	RoundedA	TruncatedA	Floor of A	Ceil of A
-1.200	-1.000	-1.000	-2.000	-1.000
-3.915	-4.000	-3.000	-4.000	-3.000
.
.051	0.000	0.000	0.000	1.000
1.238	1.000	1.000	1.000	2.000
4.800	5.000	4.000	4.000	5.000

Truncation removes all digits after the decimal point. Thus, the behavior of truncation varies by sign: it effectively rounds negative numbers to the next *greater* integer and positive numbers to the next *lesser* integer. As do all computations, Trunc works with actual stored values rather than the way values are displayed. For example, the value 1.9 is displayed in a format with no decimal places as 2, but it truncates to 1.

Related functions are [Round\(?\)](#) [p. 416], [Floor\(?\)](#) [p. 380], and [Ceil\(?\)](#) [p. 359]; a careful comparison of Round, Floor, Ceil, and Trunc is made in the entry for Round.

VariableElement(?, ?)

VariableElement(*var*, *n*) returns the current character representation of the *n*th row's value for the *var* you specify. The first argument must be a variable, and the second argument *n* must be a valid row number (or a variable containing such numbers). The function works casewise.

VariableElement(Weight, 5)

	Weight	Fifth weight
1	2700	2790
2	3265	2790
3	2935	2790
4	2670	2790
5	2790	2790
6	2895	2790
7	2640	2790

Above, we find the value in the fifth row of the Weight variable in Car Data.

Suppose you wanted to know the sum of the 5th largest and 5th smallest values of the variable Weight in Car Data. Sort the variable (use Sort from the Manage menu) in increasing order, then sum those elements. If you know the variable has, say, 100 nonmissing (Weight does not!), you could do:

VariableElement(Weight, 5) + VariableElement(Weight, 96)

If you don't know the number of nonmissing values, you can use Count:

VariableElement(Weight, 5) + VariableElement(Weight, Count(Weight, AllRows)-4)

Weight	Answer
1695	6100
1845	6100
1900	6100
2075	6100
2170	6100
2185	6100

VariableElement is handy for simulating spreadsheet functionality. You might use it to supply an argument to a function when you expect to change that value frequently, since it is easier to edit data values than formulas.

Variance(?, AllRows)

Variance(*var*, AllRows) computes the variance of the variable you specify; by default, Variance is based on AllRows of the variable, but you may instead specify OnlyIncludedRows or OnlyExcludedRows as the second argument. Missing values are ignored. The function works columnwise and produces the same result for every row.

Variance(A, AllRows)

	A	VarOfA
Std. Error:	2.323	0.000
Variance:	21.583	0.000
1	-4.000	21.583
2	-5.000	21.583
3	•	21.583
4	•	21.583
5	1.000	21.583
6	5.000	21.583

Variance is a measure of variability about the mean. Variance is expressed in a square of the units of the variable; for instance, if you are measuring length in meters, variance is a quantity of area in square meters. Consequently, variance can be difficult to interpret, and standard deviation (the square root of variance) is often preferred; see [StandardDeviation\(?, AllRows\)](#) [p. 420].

StatView uses *n*–1 in the denominator for the Variance function’s computations; if you prefer *n*, you can build your own formula:

Variance(A, AllRows)*(Count(A, AllRows) – 1)/Count(A, AllRows)

Or, use the Descriptive Statistics analysis (see “[Descriptive Statistics](#),” p. 1, which allows you to choose *n* or *n*–1. For sample variance, *n*–1 is generally preferred; for population variance, *n* is usually preferred.

Weekday(?)

Weekday(*date*) returns an index indicating the day of the week (1=Sunday, 2=Monday, etc.) of the *date* specified. The *date* argument may be a variable or constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The DayOfWeek function is synonymous. The function works casewise.

Weekday("Other dates")

Other dates	Weekday
01.01.95	1
02.02.95	5
03.03.95	6
04.04.95	3
05.05.95	6

If you want day names, change the variable to category, and edit the category to have levels Sunday, Monday, Tuesday, ..., Saturday.

Weekday
Sunday
Thursday
Friday
Tuesday
Friday
Tuesday

WeekOfYear(?)

WeekOfYear(*date*) returns the week number of the year (1–54) of the *date* specified. The *date* argument may be a variable or constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The function works casewise.

WeekOfYear("Other dates")

Other dates	Week
01.01.95	1
02.02.95	5
03.03.95	9
04.04.95	14
05.05.95	18
06.06.95	23

Year(?)

Year(*date*) returns the year number (1904–2040) of the *date* specified. The *date* argument may be a variable or constant. (Remember, all date/time values are an exact second of an exact day, and unspecified dates are assumed to be the current date.) The function works casewise.

Year("Other dates")

Other dates	Year
01.01.95	1995
02.02.95	1995
03.03.95	1995
04.04.95	1995
05.05.95	1995
06.06.95	1995

Algorithms

General

Sum of squares calculations

Several statistics require calculation of the sum of squared deviations (sum of squares):

$$\sum (X - \bar{x})^2$$

StatView uses an algorithm that provides more accurate results for the sum of squared deviations than the Monroe Calculator variance formula:

$$\sum X^2 - \frac{(\sum X)^2}{n}$$

StatView uses the following algorithm for the sum of squared deviations:

$$\sum (X - k)^2 - n(k - \bar{x})^2$$

where k is the first non-missing, non-excluded value for the variable, and \bar{x} is the calculated variable mean.

In addition, several statistics require that the sum of deviation cross products be calculated:

$$\sum (X - \bar{x})(Y - \bar{y})$$

StatView uses the following algorithm for the sum of deviation cross products:

$$\sum (X - a)(Y - b) - n(a - \bar{x})(b - \bar{y})$$

where (a, b) is the first non-missing, non-excluded X, Y pair, \bar{x} is the X variable mean, and \bar{y} is the Y variable mean.

Matrix inversions

Several statistics require matrix inversions. StatView uses the Sweep Operator procedure to invert matrices.

Descriptive Statistics

Continuous variables

n = number of non-missing, non-excluded values

Count = n

Mean, referred to below as \bar{x} or \bar{y}

$$\bar{x} = \frac{\sum X}{n}$$

Variance

$$s^2 = \frac{\sum (X - \bar{x})^2}{n - 1}$$

Standard Deviation

$$s = \sqrt{s^2}$$

Standard Error of the Mean

$$s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

Coefficient of Variation = s/\bar{x}

Minimum = smallest value among X

Maximum = largest values among X

Range = Maximum – Minimum

Sum = $\sum X$

Sum of squares = $\sum X^2$

number missing = count of the missing values

Geometric Mean = $\sqrt[n]{\prod X}$

Harmonic Mean

$$\left(\frac{\sum \frac{1}{X}}{n} \right)^{-1}$$

Kurtosis = $(m_4/m_2^2) - 3$

Skewness = $m_3 / (m_2 \sqrt{m_2})$, where

$$m_2 = \frac{\sum (X - \bar{x})^2}{n}$$

$$m_3 = \frac{\sum (X - \bar{x})^3}{n}$$

$$m_4 = \frac{\sum (X - \bar{x})^4}{n}$$

Mode = unique most commonly occurring value among X

Median = 50th percentile (see [“Percentiles,” p. 435](#))

Interquartile Range (IQR) = 75th percentile – 25th percentile (see “Percentiles” below)

Median Absolute Deviation from the Median (MAD) = Median(D), where

$$D = |X - \text{Median}(X)|$$

$p\%$ Trimmed Mean = $(X_{k+1} + \dots + X_{n-k}) / (n - 2k)$, where the X s are sorted from smallest to largest and k is chosen so that k observations represent $p\%$ of the data

Nominal variables

Count, number missing, and mode are as above

Number of levels = number of uniquely occurring values among X

Percentiles

The p th percentile using linear interpolation is $(1 - f)x_k + f^*x_{k+1}$, where x_k and x_{k+1} are the k th and $(k+1)$ st non-missing, non-excluded values in the variable, after sorting the X s from smallest to largest.

k is the integer part of v and f is the fractional part of v : $v = (np)/100 + 0.5$, where n is the count and p is the desired percentile.

One Sample Analysis

N = number of observations

DF = $N - 1$

$$\text{SE} = \text{standard error of } \bar{x} = \frac{s}{\sqrt{N}}$$

One sample t -test

U = hypothesized mean, entered by user

$$t = \frac{(\bar{x} - U)}{\text{SE}}$$

Confidence interval for the mean

t_a is the (two-tailed) critical value of the t distribution at level a and degrees of freedom

$$\text{lower} = \bar{x} - t_a \text{SE}$$

$$\text{upper} = \bar{x} + t_a \text{SE}$$

Chi-Square test for variance

σ^2 = hypothesized variance, entered by user

$$\chi^2 = \text{DF} \frac{s^2}{\sigma^2}$$

Confidence interval for variance

x_l = lower chi-square critical value, level a , DF degrees of freedom

x_u = upper chi-square critical value, level a , DF degrees of freedom

$$\text{upper} = \text{DF} \frac{s^2}{x_u}$$

$$\text{lower} = \text{DF} \frac{s^2}{x_l}$$

Paired Comparisons

N = number of paired observations

$$D = X_1 - X_2$$

$$\text{DF} = N - 1$$

\bar{D} = mean of D

s_d = standard deviation of D

$$\text{SE} = \text{standard error of } \bar{D} = \frac{s_d}{\sqrt{nN}}$$

Paired t -test

Δ = hypothesized mean difference, entered by user

$$t = (\bar{D} - \Delta) / \text{SE}$$

Confidence interval for the paired mean difference

t_α is the (two-tailed) critical value of the t distribution at level α and DF degrees of freedom

$$\text{lower} = \bar{D} - t_\alpha \text{SE}$$

$$\text{upper} = \bar{D} + t_\alpha \text{SE}$$

Z test and confidence interval for the correlation coefficient

These are calculated using the r to z transformation discussed under Correlation/Covariance, below.

Unpaired Comparisons

N_1 = number of observations in group 1

N_2 = number of observations in group 2

$$\text{DF} = N_1 + N_2 - 2$$

\bar{x}_1 is the mean of the group 1 observations

\bar{x}_2 is the mean of the group 2 observations

$$D = \bar{x}_1 - \bar{x}_2$$

s_1 is the standard deviation of the group 1 observations

s_2 is the standard deviation of the group 2 observations

Standard error:

$$SE = \sqrt{\frac{s_1^2(N_1 - 1) + s_2^2(N_2 - 1)}{DF}} \times \frac{N_1 + N_2}{N_1 N_2}$$

Unpaired t -test

Δ = hypothesized mean difference, entered by user

$$t = (D - \Delta) / SE$$

Confidence interval for the unpaired mean difference

t_a is the (two-tailed) critical value of the t distribution at level α and DF degrees of freedom

$$\text{lower} = \bar{D} - t_a SE$$

$$\text{upper} = \bar{D} + t_a SE$$

F test for variance ratio

VR = hypothesized variance ratio, entered by user

$$F = \frac{s_1^2 / s_2^2}{VR}$$

$$DF = N_1 - 1, N_2 - 1$$

Confidence interval for the variance ratio

$$\text{lower} = \frac{(s_1^2 / s_2^2)}{(F(N_1 - 1, N_2 - 1, \alpha))}$$

$$\text{upper} = (s_1^2/s_2^2) \cdot (F(N_2 - 1, N_1 - 1, a))$$

where $F(n, m, a)$ is the critical value of the F distribution with n and m degrees of freedom at level a .

Correlation and Covariance

Covariances and correlations are computed in StatView using provisional means.

Partial correlations

Where PC is the partial correlation matrix and IC is the inverse of the correlation matrix:

$$\text{PC}_{ij} = \frac{-\text{IC}_{ij}}{\sqrt{\text{IC}_{ii}\text{IC}_{jj}}}$$

Bartlett's test of sphericity

$$\chi^2 = -N \ln(\det(C))$$

$$df = \frac{n(n+1)}{2} - 1$$

N = number of observations

n = number of variables

$\det(C)$ = determinant of the correlation matrix

p values and confidence intervals

These are computed using the transformation $z = \frac{1}{2} \ln\left(\frac{1+r}{1-r}\right)$, which has an approximately normal distribution with mean $= \frac{1}{2} \ln\left(\frac{1+R}{1-R}\right)$ and variance $= \frac{1}{N-3}$ when the data are a random sample of N observations from a bivariate normal population with correlation R .

Regression

StatView applies the Sweep Operator to the $\mathbf{X}^T\mathbf{X}$ matrix of cross product deviations in order to calculate regression coefficients. Sweeping operations are discussed in Draper and Smith

(1981), Hocking (1985) and Goodnight (1979). The sweeping operation is used to add and delete variables from the regression equation. Beta coefficients, partial correlations, multiple correlation, partial F s and residual sum of squares are computed as each variable enters (or leaves) the regression equation. The calculation of confidence bands for the mean and confidence intervals for the slope of a simple regression is discussed in Draper and Smith (1981) and Sokal and Rohlf (1981).

ANOVA

The technique used for calculating the sums of squares for the various tests reported by StatView is the reduction technique as described by Searle (1971, pp. 246–248). The basic idea of the reduction technique is as follows. First a model is fit with all entered main effects and interactions (the full model), and the residual sum of squares, RSS_{full} is calculated. Then for each main effect or interaction to be tested, another model is fit, containing all the terms in the model except the one currently being considered. Once again, the residual sum of squares is calculated. Let the residual sum of squares for the model excluding only effect A (where A is any main effect or interaction in the model) be denoted RSS_A . Then the sum of squares for testing the hypothesis that effect A has no influence on the dependent variable is calculated as: $\text{SS}_A = \text{RSS}_A - \text{RSS}_{\text{full}}$. This calculation is carried out for each term in the model.

The reduction sums of squares are calculated using a method described in detail by Hocking (1985, pp. 146 - 148). First, the matrix $\mathbf{X}^T\mathbf{X}$ is calculated, using a full rank parameterization for the design matrix \mathbf{X} . In this parameterization, the first element of each row of the design matrix is a 1 (for the intercept), and for a nominal main effect with k levels, there are $k-1$ columns in the design matrix. For all but the last level of the factor, a 1 is placed in the column corresponding to the level of that factor for a given observation (row), while for observations with the last level of the factor, all $k-1$ columns are filled with -1 s. Covariates are simply entered as the column of values of the covariate. The columns corresponding to interaction terms for a particular row are formed as the Kronecker product of the columns corresponding to all main effects contained in the interaction. Finally, the values of the dependent variables are stored as the last columns in the design matrix.

The matrix $\mathbf{X}^T\mathbf{X}$ is swept on its columns corresponding to entered effects. (See Goodnight (1979), for a description of the Sweep Operator). The square submatrix in the lower right hand corner of the $\mathbf{X}^T\mathbf{X}$ matrix is the sum of squares and cross products, SSCP , matrix. Its number of columns equals the number dependent variables. The SSCP for the fully swept $\mathbf{X}^T\mathbf{X}$ is called the error SSCP matrix, \mathbf{E} . The residual sum of squares for each dependent variable in the full model, (RSS_{full}), is the corresponding diagonal element of \mathbf{E} . Due to the reversibility of the sweep operator, RSS_A for any effect A can be calculated by re-sweeping the columns corresponding to the effect in question in the fully swept $\mathbf{X}^T\mathbf{X}$ matrix, and extracting the appropriate diagonal element of the SSCP matrix. The hypothesis SSCP matrix, \mathbf{H}_A , formed by subtracting \mathbf{E} from this partially swept SSCP matrix, is used in multivariate tests. [Note: in models with one dependent variable, the SSCP matrices have only one element, so RSS_{full} and RSS_A are simply the lower right hand element of $\mathbf{X}^T\mathbf{X}$ after the appropriate sweeping operations have been performed.] The sums of squares for each effect are then calculated from RSS_{full} and RSS_A as described above.

Repeated Measures

Repeated measures models are computed via multivariate analysis of variance. This allows multivariate tests of hypotheses involving repeated measures to be computed in addition to the usual univariate tests. The multivariate tests are formed by applying transformations to the \mathbf{H}_A and \mathbf{E} matrices. The transformation matrices are constructed by taking the Kronecker products of matrices of orthogonal polynomial contrasts, and column vectors of 1s. For example, in a model with two within (repeated) factors U and V that have 2 and 3 levels respectively, the transformation matrix for effects involving only U would be $\mathbf{M}_u = \mathbf{O}(2) \otimes \mathbf{J}(3)$; for those involving V alone, $\mathbf{M}_v = \mathbf{J}(2) \otimes \mathbf{O}(3)$; and for those involving the interaction of U and V , $\mathbf{M}_{uv} = \mathbf{O}(2) \otimes \mathbf{O}(3)$; where $\mathbf{O}(n)$ is an $n \times (n-1)$ contrast matrix and $\mathbf{J}(n)$ is a column vector of $n-1$'s. If in this example there were also a between effect A , the test for the interaction of U with A would be formed from the transformed hypothesis matrix $\mathbf{M}_u^T \mathbf{H}_A \mathbf{M}_u$ and error matrix $\mathbf{M}_u^T \mathbf{E} \mathbf{M}_u$.

Power and lambda

Power is computed as the CDF of the non-central F distribution based on four parameters: f_{crit} , numerator degrees of freedom (p , the degrees of freedom associated with the null hypothesis), denominator degrees of freedom ($q = n - \text{residual df} - 1$), and lambda (estimated as the hypothesis sum of squares divided by the residual mean square). Here f_{crit} is the critical value for the central F distribution at level α with p and q degrees of freedom.

Multivariate analysis of variance (MANOVA)

The sums of squares due to hypothesis and error that we examine for ANOVA models are replaced by matrices of sums of squares and cross products (SSCP) for MANOVA models. Likewise we no longer examine F -ratios but instead consider multivariate tests and their F approximations. Rather than computing the ratio of hypothesis to error sums of squares, we examine eigenvalues of $\mathbf{H}\mathbf{E}^{-1}$, where \mathbf{H} is the hypothesis SSCP matrix and \mathbf{E} is the error SSCP matrix. Let $\lambda_1, \lambda_2, \dots, \lambda_s$ be the nonzero eigenvalues of $\mathbf{H}\mathbf{E}^{-1}$ listed in decreasing order (i.e., $\lambda_1 > \lambda_2 > \dots > \lambda_s$).

For all multivariate tests

v_H = degrees of freedom for \mathbf{H}

v_E = degrees of freedom for \mathbf{E}

d = number of dependent variables

$s = \min(v_H, d)$ and is also the number of nonzero eigenvalues

$$m = \frac{|v_H - d| - 1}{2}$$

$$n = \frac{v_E - d - 1}{2}$$

Wilks' Lambda

$$\Lambda = \prod_{i=1}^s \frac{1}{1 + \lambda_i}$$

$$r = v_H + v_E - \frac{d + v_H + 1}{2}$$

$$t = \begin{cases} 1 & (v_H \cdot d = 2) \\ \sqrt{\frac{d^2 v_H^2 - 4}{d^2 + v_H^2 - 5}} & (v_H \cdot d \neq 2) \end{cases}$$

$$k = \frac{d v_H - 2}{4}$$

$$F \text{ value: } \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \left(\frac{rt - 2k}{d v_H} \right)$$

Numerator DF: $d v_H$

Denominator DF: $rt - 2k$

$$p = 1 - \text{ProbF} \left(\frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \left(\frac{rt - 2k}{d v_H} \right), d v_H, rt - 2k \right)$$

(This notation is from StatView's expression language; see [“ProbF\(?, 1, 1\),” p. 402](#) for more information. It can be read as, “the CDF at the F value of an F random variable with $d v_H$ and $rt - 2k$ degrees of freedom.” You could compute this p value yourself using a formula variable with this definition, substituting the F value and degrees of freedom values shown in the MANOVA table.)

Roy's Greatest Root

$$\Theta = \lambda_1 \text{ (the largest eigenvalue of } \mathbf{H}\mathbf{E}^{-1} \text{)}$$

$$t = \max(v_H, d)$$

$$F \text{ value: } \Theta \cdot \frac{v_E - t + v_H}{t}$$

Numerator DF: t

Denominator DF: $v_E - t + v_H$

$$p = 1 - \text{ProbF}\left(\Theta \cdot \frac{v_E - t + v_H}{t}, t, v_E - t + v_H\right)$$

Hotelling-Lawley Trace

$$T_{HL} = \sum_{i=1}^s \lambda_i$$

$$F \text{ value: } \frac{T_{HL}}{s} \left(\frac{2(sn+1)}{s(2m+s+1)} \right) = \frac{2T_{HL}(sn+1)}{s^2(2m+s+1)}$$

Numerator DF: $s(2m+s+1)$

Denominator DF: $2(sn+1)$

$$p = 1 - \text{ProbF}\left(\frac{2T_{HL}(sn+1)}{s^2(2m+s+1)}, s(2m+s+1), 2(sn+1)\right)$$

Pillai Trace

$$T_P = \sum_{i=1}^s \frac{\lambda_i}{1 + \lambda_i}$$

$$F \text{ value: } \frac{T_P}{s - T_P} \left(\frac{s(2n+s+1)}{s(2m+s+1)} \right) = \frac{T_P}{s - T_P} \left(\frac{2n+s+1}{2m+s+1} \right)$$

Numerator DF: $s(2m+s+1)$

Denominator DF: $s(2n+s+1)$

$$p = 1 - \text{ProbF}\left(\frac{T_P}{s - T_P} \left(\frac{2n+s+1}{2m+s+1} \right), s(2m+s+1), s(2n+s+1)\right)$$

Multiple comparisons

Multiple comparisons are discussed in Winer (1971) and Milliken and Johnson (1984). The formulas used are listed below.

For all multiple comparison tests

k is the number of groups

α is the user-entered significance level

n_i = number of observations in group i

\bar{x}_i = the mean of the observations in group i

s_i = the standard deviation of the observations in group i

$r = 1/n_i + 1/n_j$ where groups i and j are being compared.

MSE is the error mean square

v_E is the error degrees of freedom

MD = $\bar{x}_j - \bar{x}_i$ is the difference of the group means being compared

A difference is declared significant if $|\text{MD}| > D$, where D is the test specific critical difference defined below.

Fisher's Protected Least Significant Difference (PLSD)

$D = t\sqrt{r \times \text{MSE}}$, where t is the (two-tailed) critical value of the t distribution at level α and v_E degrees of freedom.

Scheffé F test

$D = \sqrt{F \times \text{MSE} \times (k-1) \times r}$, where F is the critical value of the F distribution at level α and degrees of freedom $k-1$ and v_E .

Bonferroni/Dunn

$D = t\sqrt{r \times \text{MSE}}$, where t is the (two-tailed) critical value of the t distribution at level α/m and degrees of freedom v_E , and m is the number of comparisons, $m = k(k-1)/2$.

Dunnett's Test

$D = t_{\text{Dunnett}} \sqrt{\text{MSE} \times \left(\frac{1}{n_i} + \frac{1}{n_k} \right)}$, where n_k is the number of counts in the last (control)

group, and t_{Dunnett} is the critical value from a two-tailed Dunnett's table at significance α with $(k-1)$ comparisons and v_E degrees of freedom.

Tukey-Kramer Test

$D = q \sqrt{\frac{r \times \text{MSE}}{2}}$ where q is the critical value of the studentized range at significance α with k means and v_E degrees of freedom.

Games-Howell Test

$$D = q \sqrt{\frac{1}{2} \left(\frac{s_i^2}{n_i} + \frac{s_j^2}{n_j} \right)} \text{ where } q \text{ is the critical value of the studentized range at significance } \alpha$$

with k means and v' degrees of freedom, with

$$v' = \frac{\left(\frac{s_i^2}{n_i} + \frac{s_j^2}{n_j} \right)^2}{\frac{s_i^4}{n_i^2(n_i-1)} + \frac{s_j^4}{n_j^2(n_j-1)}}.$$

Student-Newman-Keuls Test

$D = q \sqrt{\frac{\text{MSE}}{H}}$ where H is the harmonic mean of counts in all groups, and q is the critical value of the studentized range at significance α with δ means and v_E degrees of freedom, where $\delta = |\text{Rank}(\bar{x}_i) - \text{Rank}(\bar{x}_j)| + 1$ (i.e., it is one more than the number of steps between the i th and j th means when they have been placed in ascending order).

Contingency Tables

Two way tables

n = number of observations

r = number of rows of contingency table

c = number of columns of contingency table

$$\text{DF} = (r - 1)(c - 1)$$

$$\chi^2 = \sum \frac{(O - E)^2}{E} \text{ where } E = (CR)/N, \text{ the expected values}$$

C = column total

R = row total

O = observed value

N = grand total

G statistic

$$2[\sum O \ln(O) - \sum R \ln(R) - \sum C \ln(C) + N \ln N]$$

Contingency coefficient

$$\sqrt{\frac{\chi^2}{\chi^2 + N}}$$

Phi

$$\sqrt{\frac{\chi^2}{N}}$$

Cramer's V

$$\sqrt{\frac{\chi^2}{N(q-1)}}$$

Note: when $r = c = 2$, V is the same as Phi where $q = \min(r, c)$

Chi-square with continuity correction ($r = c = 2$ only)

$$\chi^2 = \frac{N(|AD - BC| - N/2)^2}{(A+B)(C+D)(A+C)(B+D)}$$

where:

A = observed value in row 1, column1

B = observed value in row 1, column2

C = observed value in row 2, column1

D = observed value in row 2, column2

Post-hoc cell contribution

$$\frac{O - E}{\sqrt{E\left(1 - \frac{R}{N}\right)\left(1 - \frac{C}{N}\right)}}$$

Cell chi-square

$$\frac{(O - E)^2}{E}$$

Fisher's Exact Test ($r = c = 2$ only)

Define $P(n_{11}, n_{12}, n_{21}, n_{22}) = \frac{r_1!r_2!c_1!c_2!}{N!n_{11}!n_{12}!n_{21}!n_{22}!}$, where

$$r_1 = n_{11} + n_{12} = A + B$$

$$r_2 = n_{21} + n_{22} = C + D$$

$$c_1 = n_{11} + n_{21} = A + C$$

$$c_2 = n_{12} + n_{22} = B + D$$

$$N = n_{11} + n_{12} + n_{21} + n_{22} = A + B + C + D$$

Let $p_0 = P(A, B, C, D)$. If $AD \leq BC$, let $l = \min(A, D)$, $u = \min(B, C)$, and $s = 1$. Otherwise, let $l = \min(B, C)$, $u = \min(A, D)$, and $s = -1$.

Let $p_1 = p_0 + \sum_{i=1}^l P(A-si, B+si, C+si, D-si)$ and $p_2 = \sum_{i=1}^u p_{2i} I(p_{2i} \leq p_0)$,

where $p_{2i} = P(A+si, B-si, C-si, D+si)$ and $I(p_{2i} < p_0) = 1$ if $p_{2i} \leq p_0$ and $= 0$ if $p_{2i} > p_0$.

The exact p value is given by $p_1 + p_2$.

Nonparametrics

One sample sign test

U = user specified hypothesized value

N_+ = number of observations $> U$

N_- = number of observations $< U$

$$N = N_+ + N_-$$

Exact p value:

$$\left(\frac{1}{2}\right)^{N-1} \sum_{i=0}^n \binom{N}{i}, \text{ where } n = \min(N_+, N_-)$$

Approximate p value:

$$\text{mean} = N/2, \text{ standard deviation} = \sqrt{N}/2, Z = (N_+ - \text{Mean})/\text{SD}.$$

Mann-Whitney U

n_1 = number of observations in group 1

n_2 = number of observations in group 2

$$N = n_1 + n_2$$

R_1 = \sum Ranks of first group

R_2 = \sum Ranks of second group

$$U_1 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

$$U_2 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

$$U = \min(U_1, U_2)$$

$$U' = n_1 n_2 - U$$

$$\text{Mean} = (n_1 n_2)/2$$

$$\text{Standard deviation} = \sqrt{n_1 n_2 (n_1 + n_2 + 1)/12}$$

$$Z = (U - \text{Mean})/\text{Standard deviation}$$

Correction for Ties

Standard deviation becomes $\sqrt{\frac{n_1 n_2}{N(N-1)} \left(\frac{N^3 - N}{12} - \sum T \right)}$, where $T = \frac{t^3 - t}{12}$ and t is the number of observations tied for a given rank.

Kolmogorov-Smirnov

See Siegel, pp. 127–136, and Hollander.

Wald-Wolfowitz runs test

n_1 = number of observations in group 1

n_2 = number of observations in group 2

R = number of runs. A run is any sequence of scores from the same group.

$$\text{Mean} = \frac{2n_1n_2}{n_1 + n_2} + 1$$

$$\text{Standard deviation} = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1 + n_2)^2(n_1 + n_2 - 1)}}$$

$$Z = \frac{|R - \text{Mean}| - 0.5}{\text{Standard deviation}}$$

Note that there are no corrections for ties. Ties may invalidate the results.

Wilcoxon signed-rank

$D = X - Y$ for each matched pair

N = number of matched pairs excluding those with a D of zero

R = Rank of $|D|$

$R_+ = \sum R$ with $D > 0$

$R_- = \sum R$ with $D < 0$

$T = \min(R_+, R_-)$

Mean = $N(N + 1)/4$

Standard deviation = $\sqrt{\frac{N(N + 1)(2N + 1)}{24}}$

$Z = (T - \text{Mean})/\text{Standard deviation}$

Correction for Ties

Standard deviation = $\sqrt{\frac{N(N + 1)(2N + 1) - \sum T}{24}}$ where $T = t^3 - t$ and t is the number of observations tied for a given rank.

Paired sign test

N_+ = number of pairs with $X_1 > X_2$

N_- = number of pairs with $X_1 < X_2$

$N = N_+ + N_-$

Exact p value

$$\left(\frac{1}{2}\right)^{N-1} \sum_{i=0}^n \binom{N}{i}, \text{ where } n = \min(N_+, N_-)$$

Approximate p value

$$\text{Mean} = N/2, \text{ SD} = \sqrt{N}/2, Z = (N_+ - \text{Mean})/\text{SD}.$$

Spearman rank correlation coefficient

N = number of matched pairs

R_x = Rank of X_i

R_y = Rank of Y_i

$D = R_x - R_y$ for each matched pair

Rho

$$\rho = 1 - \frac{6 \sum D^2}{N(N^2 - 1)}$$

$$Z = \rho \sqrt{N-1}$$

Correction for Ties

$$\rho = \frac{\sum x^2 + \sum y^2 - \sum D^2}{2 \sqrt{\sum x^2 \sum y^2}}$$

$$\sum x^2 = \frac{N^3 - N}{12} - \sum T_x$$

$$\sum y^2 = \frac{N^3 - N}{12} - \sum T_y$$

$$T_x = \frac{t^3 - t}{12} \text{ where } t \text{ is the number of X observations tied for a given rank.}$$

$$T_y = \frac{t^3 - t}{12} \text{ where } t \text{ is the number of Y observations tied for a given rank.}$$

Kendall correlation coefficient

N = number of matched pairs

C = Kendall statistic determined as follows:

Rank the observations on the X variable from 1 to N . Rank the observations on the Y variable from 1 to N . Arrange the list of N subjects so that the X ranks of the subjects are in their natural order, i.e., 1, 2, 3, ..., N . For each Y rank, count the number of ranks below it which are larger. Then subtract the number of ranks below it which are smaller. The sum of this for each Y is C .

$$t = \frac{C}{\frac{1}{2}N(N-1)}$$

Standard deviation

$$\sqrt{\frac{2(2N+5)}{9N(N-1)}}$$

$z = t/\text{Standard deviation}$

Correction for Ties

$$t = \frac{C}{\sqrt{\left(\frac{1}{2}N(N-1) - \sum T_x\right)\left(\frac{1}{2}N(N-1) - \sum T_y\right)}}$$

$$T_x = \frac{t^2 - t}{2} \text{ where } t \text{ is the number of } X \text{ observations tied for a given rank}$$

$$T_y = \frac{t^2 - t}{2} \text{ where } t \text{ is the number of } Y \text{ observations tied for a given rank}$$

Kruskal-Wallis test

k = number of groups

n_j = number of cases in the j th group

$N = \sum n_j$, the number of cases in all groups combined

R_j = sum of ranks in the j th group

$$H = \frac{\frac{12}{N(N+1)} \sum_{j=1}^k \frac{R_j^2}{n_j} - 3(N+1)}{1 - \frac{\sum T}{N^3 - N}}$$

where $T = \sum (t^3 - t)$, t is the number of tied observations in a tied group of scores, and $\sum T$ directs one to sum over all groups of ties.

Friedman test

k = number of variables

N = number of rows

$R_i = \sum R$ for each variable where R is the score ranked by row, $i=1, \dots, k$.

$$\chi_r^2 = \frac{12}{Nk(k+1)} \sum R_i^2 - 3N(k+1)$$

Correction for ties

$$\chi_r^2 = \frac{12 \sum \left(R_i - N \left(\frac{k+1}{2} \right) \right)^2}{Nk(k+1)} - \frac{\sum T}{k-1}$$

Survival analysis

For both nonparametric methods and regression models, $\{(T_i, C_i), i = 1, \dots, N\}$; T_i is the event time or censor time for the i th individual; C_i is 0 if T_i is an event time, 1 if it is a censor time.

For nonparametric methods, data may optionally be divided into G groups and/or S strata.

For regression models, $\mathbf{z}_i = (z_{1i}, \dots, z_{pi})$, a vector of covariates, may be observed for each individual (required for proportional hazards models). For proportional hazards models, the data may be divided into S strata.

Event Times

Let $t_1 < t_2 < \dots < t_E$ be the distinct ordered event times (i.e., the distinct sorted times T_i for which $C_i = 0$). Define $t_0 = 0$ and $t_{E+1} = \infty$.

Survival (Distribution) Function (SDF)

$S(t) = \text{Prob}(T > t)$, where T = time to event

Cumulative Distribution Function (CDF)

$F(t) = \text{Prob}(T \leq t) = 1 - S(t)$

Probability Density Function (PDF)

$$f(t) = \frac{dF}{dt} = F'(t)$$

Hazard Function

$$\lambda(t) = \frac{f(t)}{S(t)}$$

Cumulative Hazard Function

$$\Lambda(t) = \int_0^t \lambda(u) du = -\ln(S(t))$$

Ln Cumulative Hazard Function

$$\ln(\Lambda(t)) = \ln(-\ln(S(t)))$$

Kaplan-Meier

Let n_i = number surviving just prior to t_i

e_i = number of events at t_i

$r_i = n_i - e_i$ (remain at risk)

c_i = number censored in $[t_{i-1}, t_i)$

Survival Function

$$\hat{S}(t_i) = \prod_{j=1}^i \frac{1 - e_j}{n_j}$$

Failure (CDF)

$$\hat{F}(t) = 1 - \hat{S}(t)$$

Survival s.e.

$$\hat{\sigma}(\hat{S}(t_i)) = \hat{S}(t_i) \sqrt{\sum_{j=1}^i \frac{e_j}{n_j r_j}}$$

Cumulative Events

$$\sum_{j=1}^i e_j$$

Cumulative Censored

$$\sum_{j=1}^i c_j$$

Confidence Interval

$$\text{UCL}(\hat{S}) = \hat{S} + z_{\alpha/2} \hat{\sigma}(\hat{S})$$

$$\text{LCL}(\hat{S}) = \hat{S} - z_{\alpha/2} \hat{\sigma}(\hat{S})$$

where $z_{\alpha/2}$ is the (two-tailed) critical value of the normal distribution at significance level α .

Quantiles

Let j be such that $\hat{S}(t_{j-1}) > (1-p) \geq \hat{S}(t_j)$; then $\hat{t}_p = t_j$

Quantile standard error

$$\hat{\sigma}(\hat{t}_p) = \frac{\hat{\sigma}(\hat{S}(\hat{t}_p))}{\hat{f}(\hat{t}_p)}, \text{ where } \hat{f}(\hat{t}_p) \text{ is an estimate of the density at } \hat{t}_p$$

Mean Survival Time

$$\hat{\mu} = \sum_{i=0}^{E-1} \hat{S}(t_i)(t_{i+1} - t_i)$$

Mean standard error

$$\hat{\sigma}(\hat{\mu}) = \sqrt{\frac{M}{M-1} \sum_{i=1}^{E-1} \frac{\mu_i^2 e_i}{n_i r_i}}, \text{ where } \mu_i = \sum_{j=i}^{E-1} \hat{S}(t_j)(t_{j+1} - t_j) \text{ and } M = \sum_{j=1}^E e_j$$

Actuarial

Interval i (denoted I_i) is $[\tau_{i-1}, \tau_i) = [(i-1)\Delta t, i\Delta t)$, $i = 1, \dots, M$

$\tau_{mi} = (i-1/2)\Delta t$ (interval i midpoint)

n_i = number entering I_i (number entered)

e_i = number events in I_i (number events)

c_i = number censored in I_i (number censored)

$$n_i' = n_i - \frac{c_i}{2} \text{ (effective number at risk)}$$

$$\hat{q}_i = \frac{e_i}{n_i'} \text{ (conditional probability of failure)}$$

$$\hat{p}_i = 1 - \hat{q}_i \text{ (conditional probability of survival)}$$

$$\hat{\sigma}(\hat{q}_i) = \sqrt{\hat{q}_i \hat{p}_i / n_i'} \text{ (conditional probability of failure standard error)}$$

Survival function

$$\hat{S}(\tau_i) = \prod_{j=1}^i \hat{p}_j$$

Failure (CDF)

$$\hat{F}(t) = 1 - \hat{S}(t)$$

Survival s.e.

$$\hat{\sigma}(\hat{S}(\tau_i)) = \hat{S}(\tau_i) \sqrt{\sum_{j=1}^i \frac{\hat{q}_j}{n_j' \hat{p}_j}}$$

Density

$$\hat{f}(\tau_{mi}) = \frac{\hat{S}(\tau_{i-1}) \hat{q}_i}{\Delta t}$$

Density standard error

$$\hat{\sigma}(\hat{f}(\tau_{mi})) = \hat{f}(\tau_{mi}) \sqrt{\sum_{j=1}^{i-1} \frac{\hat{q}_j}{n_j' \hat{p}_j} + \frac{\hat{p}_i}{n_i' \hat{q}_i}}$$

Hazard

$$\hat{\lambda}(\tau_{mi}) = \frac{2\hat{q}_i}{\Delta t(1 + \hat{p}_i)}$$

Hazard standard error

$$\hat{\sigma}(\hat{\lambda}(\tau_{mi})) = \hat{\lambda}(\tau_{mi}) \sqrt{\frac{1 - \left(\frac{\Delta t \hat{\lambda}(\tau_{mi})}{2}\right)^2}{n_i' \hat{q}_i}}$$

Median Residual Lifetime (MRL)

Let j be such that $\hat{S}(\tau_{j-1}) \geq \hat{S}(\tau_i)/2 > \hat{S}(\tau_j)$; then

$$\hat{M}_i = \tau_{j-1} - \tau_i + \Delta t \frac{\hat{S}(\tau_{j-1}) - \frac{\hat{S}(\tau_i)}{2}}{\hat{S}(\tau_{j-1}) - \hat{S}(\tau_j)}$$

MRL standard error

$$\hat{\sigma}(\hat{M}_i) = \frac{\hat{S}(\tau_i)}{2\hat{f}(\tau_{mj})\sqrt{n_i'}}$$

Confidence intervals

$$\text{UCL}(\hat{g}) = \hat{g} + z_{\alpha/2} \hat{\sigma}(\hat{g})$$

$$\text{LCL}(\hat{g}) = \hat{g} - z_{\alpha/2} \hat{\sigma}(\hat{g})$$

where \hat{g} is one of \hat{S} , \hat{f} , or $\hat{\lambda}$, and $z_{\alpha/2}$ is the (two-tailed) critical value of the normal distribution at significance level α .

Quantiles

Let i be such that $\hat{S}(\tau_i) \geq 1 - p > \hat{S}(\tau_{i+1})$. Then estimate the p th quantile by linear interpolation:

$$\hat{t}_p = \tau_i + \Delta t \frac{\hat{S}(\tau_i) - (1 - p)}{\hat{S}(\tau_i) - \hat{S}(\tau_{i+1})}$$

Quantile standard error

$$\hat{\sigma}(\hat{t}_p) = \frac{1}{\hat{f}(\tau_{m_i})} \sqrt{\frac{p(1-p)}{n_i'}}$$

Linear rank tests

Let n_{ij} = size of risk set for group j at t_i

e_{ij} = number of events for group j at t_i

$$n_i = \sum_{j=1}^G n_{ij}$$

$$e_i = \sum_{j=1}^G e_{ij}$$

$$r_i = n_i - e_i$$

Test statistics

$$\mathbf{v}' \mathbf{V}^{-1} \mathbf{v} \cong \chi_{G-1}^2, \text{ where } v_j = \sum_{i=1}^E w_i \left(e_{ij} - \frac{n_{ij} e_i}{n_i} \right) \text{ and}$$

$$V_{jk} = \sum_{i=1}^E \frac{w_i^2 (n_i n_{ik} \delta_{jk} - n_{ij} n_{ik}) e_i r_i}{n_i^2 (n_i - 1)}$$

for $j, k = 1, \dots, G-1$, $\delta_{jk} = 1$ if $j = k$ or 0 otherwise

Weights

Logrank (Mantel-Cox)

$$w_i = 1$$

Breslow-Gehan-Wilcoxon

$$w_i = n_i$$

Tarone-Ware

$$w_i = \sqrt{n_i}$$

Peto-Peto Wilcoxon

$$w_i = \tilde{S}_i = \prod_{j=1}^i \frac{1 - e_j}{n_j + 1}$$

Harrington-Fleming

$$w_i = \hat{S}_{i-1}^p = \left(\prod_{j=1}^{i-1} \frac{1 - e_j}{n_j} \right)^p$$

Here, \hat{S} is the KM estimate of the survival function, computed separately for strata but pooled for groups. \tilde{S} is a slight modification of this estimate, with $n_j + 1$ replacing n_j .

Cell Contributions ($j = 1, \dots, G$)

Sum weighted observed

$$O_j = \sum_{i=1}^E w_i e_{ij}$$

Sum weighted expected

$$E_j = \sum_{i=1}^E \frac{w_i n_{ij} e_i}{n_i}$$

Contribution

$$\frac{(O_j - E_j)^2}{E_j}$$

Note that the sum of the cell contributions, which is an approximate chi-square statistic, is not the same as the statistic $\mathbf{v}'\mathbf{V}^{-1}\mathbf{v}$ and is conservative (Peto and Pike, 1973).

Stratification

$$\mathbf{v} = \sum_{i=1}^S \mathbf{v}_i \text{ and } \mathbf{V} = \sum_{i=1}^S \mathbf{V}_i$$

where \mathbf{v}_i and \mathbf{V}_i are computed for the i th stratum.

Trend versions

Let $\mathbf{d} = (d_1, \dots, d_G)'$ be a set of weights for the groups. (If there is a numeric grouping variable and numeric values are used, these are the levels of the grouping variable; otherwise $\mathbf{d} = (-(G-1), \dots, -3, -1, 1, 3, \dots, (G-1))'$ if G is even, and $\mathbf{d} = ((-(G-1))/2, \dots, -1, 0, 1, \dots, (G-1)/2)'$ if G is odd.)

The test statistic in this case is $\frac{(\mathbf{d}'\mathbf{v})^2}{\mathbf{d}'\mathbf{V}\mathbf{d}} \equiv \chi_1^2$

Proportional hazards model

Hazard function

$\lambda(t; \mathbf{z}) = \lambda_0(t) e^{\beta' \mathbf{z}}$ where $\mathbf{z} = (z_1, \dots, z_p)$ is a vector of covariates and β is a vector of unknown coefficients. $\lambda_0(t)$ is the baseline hazard function, corresponding to $\mathbf{z} = 0$.

Survival function

$S(t; \mathbf{z}) = (S_0(t))^{e^{\beta' \mathbf{z}}}$ where $S_0(t) = e^{-\int_0^t \lambda_0(u) du}$ is the baseline survival function.

Parametric models

$$Y = \ln(T) = \mu + \beta' \mathbf{z} + \sigma W$$

where W is a random variable with a distribution specified by the model chosen (see below); models available are exponential, Weibull, lognormal and logistic. Note that the model distribution refers to the distribution of the untransformed response time. Also, for the exponential model $\sigma \equiv 1$.

If the Don't transform time variable option is checked, it is assumed that the event time variable contains the values of Y_i rather than those of T_i .

Estimation (proportional hazards)

$R_i = \{\text{individuals at risk just before } t_i\}$

$E_i = \{\text{individuals who fail at } t_i\}$

$e_i = \text{number of elements of } E_i = \text{number events at } t_i$

$$s_i = \sum_{l \in E_i} \mathbf{z}_l$$

Partial likelihood function

Assuming no ties, i.e., $e_i \equiv 1$,

$$L(\beta) = \prod_{i=1}^E \frac{e^{\beta' \mathbf{z}_i}}{\sum_{l \in R_i} e^{\beta' \mathbf{z}_l}}$$

Breslow approximate likelihood

In case of ties,

$$L(\beta) = \prod_{i=1}^E \frac{e^{\beta' s_i}}{\left(\sum_{l \in R_i} e^{\beta' z_l} \right)^{e_i}}$$

Log-likelihood

$$\ell(\beta) = \ln(L(\beta)) = \sum_{i=1}^E \left(\beta' s_i - e_i \ln \left(\sum_{l \in R_i} e^{\beta' z_l} \right) \right)$$

First derivatives (score function)

$U(\beta) = \frac{\partial}{\partial \beta} \ell(\beta)$, where

$$U_j(\beta) = \frac{\partial \ell(\beta)}{\partial \beta_j} = \sum_{i=1}^E \left(s_{ij} - e_i \frac{\sum_{l \in R_i} z_{lj} e^{\beta' z_l}}{\sum_{l \in R_i} e^{\beta' z_l}} \right)$$

Second derivatives (information matrix)

$$I(\beta) = - \left(\frac{\partial^2}{\partial \beta^2} \ell(\beta) \right)$$

$$\begin{aligned} I_{jk}(\beta) &= - \left(\frac{\partial^2}{\partial \beta_j \partial \beta_k} \ell(\beta) \right) \\ &= \sum_{i=1}^E e_i \left[\frac{\sum_{l \in R_i} z_{lj} z_{lk} e^{\beta' z_l}}{\sum_{l \in R_i} e^{\beta' z_l}} - \frac{\sum_{l \in R_i} z_{lj} e^{\beta' z_l}}{\sum_{l \in R_i} e^{\beta' z_l}} \cdot \frac{\sum_{l \in R_i} z_{lk} e^{\beta' z_l}}{\sum_{l \in R_i} e^{\beta' z_l}} \right] \end{aligned}$$

Estimation (parametric models)

$$y_i = \ln(T_i)$$

$$w_i = \frac{y_i - \mu - \beta' z_i}{\sigma}$$

$\sigma^{-1} g(w_i)$ is the probability density of y_i (see below for distribution-specific definitions)

$$S_g(w_i) = \int_{w_i}^{\infty} g(x) dx$$

$$\theta = (\sigma, \mu, \beta)$$

Likelihood function

$$L(\theta) = \prod_{i=1}^N (\sigma^{-1} g(w_i))^{1-C_i} (S_g(w_i))^{C_i}$$

Log-likelihood

$$\ell(\theta) = \ln(L(\theta)) = \sum_{i=1}^N \left\{ (1-C_i) \ln[\sigma^{-1} g(w_i)] + C_i \ln(S_g(w_i)) \right\}$$

First derivatives (score function)

$$U(\theta) = \frac{\partial}{\partial \theta} \ell(\theta), \text{ where}$$

$$U_1(\theta) = \frac{\partial}{\partial \sigma} \ell(\theta) = \sigma^{-1} \sum_{i=1}^N (w_i a_i + C_i - 1)$$

$$U_2(\theta) = \frac{\partial}{\partial \mu} \ell(\theta) = \sigma^{-1} \sum_{i=1}^N a_i$$

$$U_{j+2}(\theta) = \frac{\partial}{\partial \beta_j} \ell(\theta) = \sigma^{-1} \sum_{i=1}^N z_{ij} a_i, \quad j = 1, \dots, p$$

$$\text{where } a_i = (C_i - 1) \frac{\partial}{\partial w_i} \ln(g(w_i)) + C_i \lambda_g(w_i) \text{ and } \lambda_g(w) = \frac{g(w)}{S_g(w)}$$

Second derivatives (information matrix)

$$I(\theta) = -\left(\frac{\partial^2}{\partial \theta^2} \ell(\theta)\right) \text{ where}$$

$$I_{11}(\theta) = -\left(\frac{\partial^2}{\partial \sigma^2} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N (w_i^2 A_i + 1 - C_i)$$

$$I_{12}(\theta) = I_{21}(\theta) = -\left(\frac{\partial^2}{\partial \sigma \partial \mu} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N w_i A_i$$

$$I_{1,j+2}(\theta) = I_{j+2,1}(\theta) = -\left(\frac{\partial^2}{\partial \sigma \partial \beta_j} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N z_{ij} w_i A_i, j = 1, \dots, p$$

$$I_{22}(\theta) = -\left(\frac{\partial^2}{\partial \mu^2} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N A_i$$

$$I_{2,j+2}(\theta) = I_{j+2,2}(\theta) = -\left(\frac{\partial^2}{\partial \mu \partial \beta_j} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N z_{ij} A_i, j = 1, \dots, p$$

$$I_{j+2,k+2}(\theta) = -\left(\frac{\partial^2}{\partial \beta_j \partial \beta_k} \ell(\theta)\right) = \sigma^{-2} \sum_{i=1}^N z_{ij} z_{ik} A_i, j, k = 1, \dots, p$$

$$\text{where } A_i = \frac{d\alpha_i}{dw_i}$$

Exponential

Same as for Weibull (below), except that $w_i = y_i - \mu - \beta' z_i$ and probability density of y_i is $g(w_i)$ and $U_1 = I_{1j} = I_{j1} \equiv 0$. Calculations involving U and I are performed on the last $p+1$ elements of the former and the submatrix of the latter of dimension $p+1$ formed by excluding the first row and column.

Weibull

$$g(w_i) = e^{w_i - e^{w_i}}$$

Lognormal

$$g(w_i) = \frac{1}{\sqrt{2\pi}} e^{-w_i^2/2}$$

Loglogistic

$$g(w_i) = \frac{e^{w_i}}{(1 + e^{w_i})^2}$$

OLS initial parameter estimates

Initial estimates of μ , β from the regression model $Y = \mu + \beta'z + \varepsilon$; then estimate σ as

$$\frac{1}{s_g} \sqrt{\frac{\text{RSS}}{N-p-1}}, \text{ where RSS is the residual sum of squares from the regression and } s_g \text{ is 1.28,}$$

1.00, 1.81 for the Weibull, lognormal and loglogistic models, respectively.

Newton-Raphson iteration

Find $\hat{\beta}$ such that $U(\hat{\beta}) = 0$

$\hat{\beta}^{j+1} = \hat{\beta}^j + \Delta\beta^j$, where $\hat{\beta}_0 = 0$ and $\Delta\beta^j = I^{-1}(\hat{\beta}_j)U(\hat{\beta}_j)$, and the algorithm ter-

minates when $\left| \frac{\ell(\hat{\beta}^j) - \ell(\hat{\beta}^{j-1})}{\ell(\hat{\beta}^j)} \right| \leq c$. If $\ell(\hat{\beta}^j) < \ell(\hat{\beta}^{j-1})$, then the step is repeated

with the step size halved, i.e., $\hat{\beta}^{j+1} = \hat{\beta}^{j-1} + \frac{\Delta\beta^{j-1}}{2}$

Coefficient covariances

$$\hat{V}(\hat{\beta}) = I^{-1}(\hat{\beta})$$

Model coefficient p values (Wald)

For continuous or two-level nominal variables (including the dummy variables corresponding to levels of multi-level nominals):

$$\frac{\hat{\beta}_i^2}{\hat{V}_{ii}(\hat{\beta})} \cong \chi_1^2$$

For nominal variables with $k > 2$ levels, where $\hat{\beta}_i$ is the vector of coefficients associated with the first $k - 1$ levels of variable i and $\hat{V}_i(\hat{\beta})$ is the corresponding submatrix of $\hat{V}(\hat{\beta})$:

$$\hat{\beta}_i' \hat{V}_i^{-1}(\hat{\beta}) \hat{\beta}_i \cong \chi_{k-1}^2$$

Confidence intervals

$$\text{UCL}\left(e^{\hat{\beta}_i}\right) = e^{\hat{\beta}_i + z_{\alpha/2} \hat{V}_{ii}^{1/2}(\hat{\beta})}$$

$$\text{LCL}\left(e^{\hat{\beta}_i}\right) = e^{\hat{\beta}_i - z_{\alpha/2} \hat{V}_{ii}^{1/2}(\hat{\beta})}$$

where $z_{\alpha/2}$ is the (two-tailed) critical value of the normal distribution at significance level α .

Survival function and related quantities

Baseline survival function

Proportional hazards

$\hat{S}_0(t_i) = \prod_{j=0}^{i-1} \hat{\alpha}_j$ where $\hat{\alpha}_0 = 1$ and $\hat{\alpha}_i$ is the solution to

$$\sum_{j \in E_i} \frac{e^{\hat{\beta}'z_j}}{1 - \hat{\alpha}_i} = \sum_{l \in R_i} e^{\hat{\beta}'z_l}. \text{ If } e_i = 1, \text{ then } \hat{\alpha}_i = \left(1 - \frac{e^{\hat{\beta}'z_i}}{\sum_{l \in R_i} e^{\hat{\beta}'z_l}} \right)^{e^{\hat{\beta}'z_i}}. \text{ Other-}$$

wise, an iterative solution is required, using as an initial value $\hat{\alpha}_{0i}$, where

$$\ln(\hat{\alpha}_{0i}) = \frac{-e_i}{\sum_{l \in R_i} e^{\hat{\beta}'z_l}}. \text{ If the iterative solution fails to converge at time } t_j, \text{ the base-}$$

line survival function is missing for times $\geq t_j$.

Exponential

$$\hat{S}_0(t) = e^{-\hat{\alpha}t} \text{ where } \hat{\alpha} = e^{-\hat{\mu}}$$

Weibull

$$\hat{S}_0(t) = e^{-\hat{\alpha}t^{\hat{\gamma}}} \text{ where } \hat{\gamma} = 1/\hat{\sigma} \text{ and } \hat{\alpha} = e^{-\hat{\mu}/\hat{\sigma}}$$

Lognormal

$\hat{S}_0(t) = 1 - \Phi\left(\frac{\ln(t) - \hat{\mu}}{\hat{\sigma}}\right)$ where Φ is the standard normal cumulative distribution function.

Loglogistic

$\hat{S}_0(t) = \frac{1}{1 + \hat{\alpha}t^{\hat{\gamma}}}$ where $\hat{\gamma} = 1/\hat{\sigma}$ and $\hat{\alpha} = e^{-\hat{\mu}/\hat{\sigma}}$

Survival function evaluated at the observations

Proportional Hazards

$$\hat{S}(T_i; \mathbf{z}_i) = \hat{S}_0(T_i)^{e^{\hat{\beta}'\mathbf{z}_i}}$$

Parametric Models

$$\hat{S}(T_i; \mathbf{z}_i) = \hat{S}_0\left(T_i^{e^{(-\hat{\beta})'\mathbf{z}_i}}\right)$$

Cumulative hazard

$$\hat{\Lambda}_0(t_i) = -\ln(\hat{S}_0(t_i))$$

$$\hat{\Lambda}(T_i; \mathbf{z}_i) = -\ln(\hat{S}(T_i; \mathbf{z}_i))$$

Linear predictor and its standard error

$$LP_i = \hat{\beta}'\mathbf{z}_i$$

$$\hat{\sigma}(LP_i) = \sqrt{\mathbf{z}_i' \hat{\mathbf{V}}(\hat{\beta}) \mathbf{z}_i}$$

Residuals

Let $\delta_i = 1 - C_i$, i.e., δ_i is 1 if T_i is an event time and 0 if T_i is a censor time. For the proportional hazards model, let $E_i^* = \{\text{individuals with } T \leq T_i \text{ and } \delta = 1\}$. Also, if T_i is an

event time, let $P_i = \sum_{k \in R_i} e^{\hat{\beta}'\mathbf{z}_k}$ and $Q_{ij} = \sum_{k \in R_i} z_{kj} e^{\hat{\beta}'\mathbf{z}_k}$

Martingale Residuals

$\hat{\epsilon}_i^M = \delta_i - \hat{\Lambda}_0(T_i)e^{\hat{\beta}'z_i}$, where $\hat{\Lambda}_0(T_i)$ is an estimate of the baseline cumulative hazard function. For parametric models, $\hat{\Lambda}_0$ is defined above. For proportional hazards, it is defined as $\hat{\Lambda}_0(T_i) = \sum_{k \in E_i} \frac{1}{P_k}$

Deviance Residuals (proportional hazards only)

$$\hat{\epsilon}_i^D = \text{sgn}(\hat{\epsilon}_i^M) \sqrt{-2[\hat{\epsilon}_i^M + \delta_i \ln(\delta_i - \hat{\epsilon}_i^M)]}$$

Score Residuals (proportional hazards only)

$$\hat{\epsilon}_i^S = (\hat{\epsilon}_{i1}^S, \dots, \hat{\epsilon}_{ip}^S), \text{ where } \hat{\epsilon}_{ij}^S = \delta_i \left(z_{ij} - \frac{Q_{ij}}{P_i} \right) - e^{\hat{\beta}'z_i} \left[\sum_{k \in E_i} \left(\frac{z_{ij}}{P_k} - \frac{Q_{kj}}{P_k^2} \right) \right]$$

Quantiles (parametric models only)

No covariates in model

The quantities plotted are $\hat{F}^{-1}(1 - \hat{p}_i)$ vs. $T_{(i)}$ where \hat{F} is the model distribution CDF with estimated scale and intercept parameters, $T_{(i)}$ is the i th sorted event time, and \hat{p}_i is the Kaplan-Meier estimate of the survival function at $T_{(i)}$.

Covariates in model

Let $T_{0i} = T_i e^{-\hat{\beta}'z_i}$ be the event or censor time for the i th individual, adjusted for covariates. The quantities plotted are $\hat{F}^{-1}(1 - \hat{p}_{0i})$ vs. $T_{0(i)}$ where \hat{F} is the model distribution CDF with estimated scale and intercept parameters, $T_{0(i)}$ is the i th sorted event time, and \hat{p}_{0i} is the Kaplan-Meier estimate of the survival function at $T_{0(i)}$ computed using the T_{0i} .

Testing the global null hypothesis

For $H_0: \beta = 0$, there are three different statistics with p degrees of freedom.

Wald

$$\chi_w^2 = \hat{\beta}' \hat{V}^{-1}(\hat{\beta}) \hat{\beta}$$

Score

$$\chi_S^2 = U'(0)I^{-1}(0)U(0)$$

Likelihood ratio

$$\chi_{LR}^2 = 2[\ell(\hat{\beta}) - \ell(\mathbf{0})]$$

Joint significance tests

H_0 : some subset β^* of the coefficients are 0.

Let r be the number of elements of β^* , $\hat{\beta}_r$ be the MPLE of the coefficients when β^* is restricted to be $\mathbf{0}$, and $\hat{\beta}^*$ be the coefficients of $\hat{\beta}$ corresponding to β^* . Also let $\mathbf{U}(\beta_0)$, $\mathbf{I}(\beta_0)$ be the elements corresponding to β^* of the score function and information matrix, respectively, evaluated under the model with β^* constrained to $\mathbf{0}$.

Three different χ^2 statistics with r degrees of freedom:

Wald

$$\chi_W^2 = \hat{\beta}^* \hat{\mathbf{V}}^{-1}(\hat{\beta}^*) \hat{\beta}^*$$

Score

$$\chi_S^2 = \mathbf{U}'(\hat{\beta}_0) \mathbf{I}^{-1}(\hat{\beta}_0) \mathbf{U}(\hat{\beta}_0)$$

Likelihood ratio

$$\chi_{LR}^2 = 2[\ell(\hat{\beta}) - \ell(\hat{\beta}_r)]$$

Stratification (proportional hazards only)

$$\lambda_i(t; \mathbf{z}) = \lambda_{0i}(t) e^{\beta' \mathbf{z}}, i = 1, \dots, S$$

$$\ell(\beta) = \sum_{i=1}^S \ell_i(\beta)$$

$$\mathbf{U}(\beta) = \sum_{i=1}^S \mathbf{U}_i(\beta)$$

$$\mathbf{I}(\beta) = \sum_{i=1}^S \mathbf{I}_i(\beta)$$

where λ_i , ℓ_i , \mathbf{U}_i , and \mathbf{I}_i are the hazard, log likelihood, score function and information matrix, respectively, for the i th stratum.

Stepwise

Remove variable i for which

$$\chi^2(i) = \frac{\hat{\beta}_i^2}{\hat{V}_{ii}(\hat{\beta})} \text{ is smallest if } P(\chi_1^2 > \chi^2(i)) > \text{P-to-Remove}.$$

Enter variable i for which

$$\chi^2(i) = \frac{U_i^2(\tilde{\beta})}{I_{ii}^*(\tilde{\beta})} \text{ is largest if } P(\chi_1^2 > \chi^2(i)) < \text{P-to-Enter},$$

where $\tilde{\beta}$ is a vector containing the estimated coefficients for variables in the model and 0s for variables not in the model, and $I^*(\tilde{\beta})$ is the information matrix swept on all rows corresponding to variables in the model.

For nominal variables with $k > 2$ levels, the above χ^2 statistics become:

$$\chi^2(i) = \hat{\beta}_i' \hat{V}_i^{-1}(\hat{\beta}) \hat{\beta}_i \equiv \chi_{k-1}^2 \text{ and } \chi^2(i) = U_i'(\hat{\beta}) I_i^{-1}(\hat{\beta}) U_i \hat{\beta} \equiv \chi_{k-1}^2$$

where the subscript i refers to the $k-1$ elements of the vectors and the $k-1 \times k-1$ elements of the matrices corresponding to the dummy variables representing the first $k-1$ levels of variable i .

Logistic Regression

For a model in which \mathbf{x}' is the vector of p covariates and Y is the nominal response variable with $R+1$ levels coded as $[0, R]$, define $\mathbf{x} = (1, \mathbf{x}')$. Let $\pi_r(\mathbf{x}) \equiv P(Y = r | \mathbf{x})$ be the conditional probability of response $Y = r$ given \mathbf{x} .

The logit functions are:

$$g_r(\mathbf{x}) = \ln \left(\frac{\pi_r(\mathbf{x})}{\pi_0(\mathbf{x})} \right).$$

Nominal independent variables are represented by collections of design variables. A K -level nominal independent variable A is represented by $K-1$ design variables (dummy variables) d_1, d_2, \dots, d_{K-1} . They have values $d_k = \delta_{k \text{Level}(A)}$, where δ is the Kronecker delta (defined below) and $\text{Level}(A)$ is the ordinal level of A numbered from 0 to $K-1$. So, for example, when A is at its first level, all d_k are zero; when A is at its second level, only $d_1 = 1$; etc.

The Kronecker delta is

$$\delta_{uv} = \begin{cases} 1 & (u = v) \\ 0 & (u \neq v) \end{cases}.$$

Logistic model

$$\begin{aligned} g_r(\mathbf{x}) &= \mathbf{x} \cdot \mathbf{B}_r \\ &= b_{r0} + x_1 b_{r1} + \dots + x_p b_{rp} \end{aligned}$$

where \mathbf{B}_r are vectors of coefficients each of length $p + 1$. (Note that $\mathbf{B}_0 = \mathbf{0}$ since $g_0(\mathbf{x}) = \ln 1 = 0$.) Solving for the π s we have

$$\pi_s(\mathbf{x}) = \frac{e^{g_s(\mathbf{x})}}{\sum_{r=0}^R e^{g_r(\mathbf{x})}}.$$

Estimation

Partial likelihood function

For a sample of n independent observations the conditional likelihood function is

$$L(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_R) = \prod_{i=1}^n \prod_{r=0}^R \pi_r(\mathbf{x}_i)^{\delta_{y_i r}}.$$

Log-likelihood

$$\begin{aligned} \ell(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_R) &= \ln(L(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_R)) \\ &= \sum_{i=1}^n \left\{ \left(\sum_{r=1}^R \delta_{y_i r} g_r(\mathbf{x}_i) \right) - \ln \left(\sum_{r=0}^R e^{g_r(\mathbf{x}_i)} \right) \right\} \end{aligned}$$

First derivatives

$$\frac{\partial \ell}{\partial B_{rk}} = \sum_{i=1}^n x_{ki} (\delta_{y_i r} - \pi_r(\mathbf{x}_i)) \text{ for } r = 1, 2, \dots, R \text{ and } k = 0, 1, \dots, p.$$

Second derivatives (information matrix)

$$\mathbf{I}(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_R) = - \left(\frac{\partial^2 \ell}{\partial \mathbf{B}^2} \right) \text{ where } \frac{\partial^2 \ell}{\partial B_{rk} \partial B_{r'k'}} = \sum_{i=1}^n x_{ki} x_{k'i'} \pi_r(\mathbf{x}_i) (\pi_{r'}(\mathbf{x}_i) - \delta_{rr'}) \text{ for } r \text{ and } r' = 1, 2, \dots, R, \text{ and } k \text{ and } k' = 0, 1, \dots, p, \text{ so } \mathbf{I} \text{ is an } R(p+1) \times R(p+1) \text{ matrix.}$$

Parameter fitting

A modified Newton-Raphson iterative procedure is used to find parameters $\hat{\mathbf{B}}_r$ for $r = 0, 1, \dots, R$ such that $\ell(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \dots, \hat{\mathbf{B}}_R)$ is maximized, simultaneously solving the

$$R(p+1) \text{ equations } \left. \frac{\partial \ell}{\partial B_{rk}} \right|_{(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \dots, \hat{\mathbf{B}}_R)} = 0.$$

Coefficient covariances

$$\hat{\mathbf{V}}(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \dots, \hat{\mathbf{B}}_R) = \mathbf{I}^{-1}(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \dots, \hat{\mathbf{B}}_R)$$

Model coefficient p values (Wald test)

For continuous or two-level nominal variables (including the dummy variables corresponding to levels of multi-level nominals):

$$\chi_{rk}^2 \cong \frac{\hat{B}_{rk}^2}{\hat{V}_{rk, rk}(\hat{\mathbf{B}})}$$

Partial correlation (R statistic)

$$R_{rk} = \begin{cases} \pm \sqrt{\frac{\chi_{rk}^2 - 2}{-2\ell_0}} & \chi_{rk}^2 > 2 \\ 0 & \chi_{rk}^2 \leq 2 \end{cases}$$

where the sign is that of the coefficient \hat{B}_{rk} , χ_{rk}^2 is the Wald statistic, and ℓ_0 is the log likelihood for a model containing only the intercepts (i.e., $B_{rk} = 0$ for $r = 0, 1, \dots, R$ and $k = 1, 2, \dots, p$).

Confidence intervals

$$\text{UCL}\left(e^{\hat{B}_{rk}}\right) = e^{\hat{B}_{rk} + z_{\alpha/2} \hat{V}_{rk, rk}^{1/2}(\hat{\mathbf{B}})}$$

$$\text{LCL}\left(e^{\hat{B}_{rk}}\right) = e^{\hat{B}_{rk} - z_{\alpha/2} \hat{V}_{rk, rk}^{1/2}(\hat{\mathbf{B}})}$$

where $z_{\alpha/2}$ is the (two-tailed) critical value of the normal distribution at significance level α .

Likelihood ratio tests

$$\chi_u^2 = 2(\ell(\hat{\mathbf{B}}) - \ell_u)$$

where ℓ_u is the log likelihood for a model refit with variable u excluded. The degrees of freedom for the test is $R(p + 1)$ minus the number of parameters fitted in the model excluding variable u .

Classification

For each \mathbf{x}_i the predicted response is $\hat{y}_i = r$ where r satisfies:

$$\pi_r(\mathbf{x}_i) \Big|_{\hat{\mathbf{B}}} = \max \left(\pi_0(\mathbf{x}_i) \Big|_{\hat{\mathbf{B}}}, \pi_1(\mathbf{x}_i) \Big|_{\hat{\mathbf{B}}}, \dots, \pi_R(\mathbf{x}_i) \Big|_{\hat{\mathbf{B}}} \right)$$

(i.e., r is the most probable response).

Global tests

Arrange the data according to distinct values of \mathbf{x} (i.e., unique covariate patterns) labeled by $g = 1, 2, \dots, G$ where the number of distinct covariate patterns is $G \leq n$. Let ρ_{gr} be the number of responses $y = r$ in the covariate group g , let n_g be the number of members of g , and let π_{gr} be the probability of response $y = r$ for group g predicted by the fitted model.

Pearson

$$\chi_P^2 = \sum_{g=1}^G \sum_{r=0}^R \frac{(\rho_{gr} - n_g \pi_{gr})^2}{n_g \pi_{gr}}$$

$$\text{DF} = R(G - p - 1)$$

Deviance

$$\chi_D^2 = 2 \sum_{g=1}^G \sum_{r=0}^R \rho_{gr} \ln \left(\frac{\rho_{gr}}{n_g \pi_{gr}} \right)$$

$$\text{DF} = R(G - p - 1)$$

Likelihood Ratio

$\chi_{LH}^2 = 2(\ell(\hat{\mathbf{B}}) - \ell_0)$ where ℓ_0 is the log likelihood for a model containing only the intercepts (i.e., $B_{rk} = 0$ for $r = 0, 1, \dots, R$ and $k = 1, 2, \dots, p$).

$$DF = Rp$$

Bivariate Plots

The scatterplot smoothers are applied to a set of points (x_i, y_i) for $i = 1, \dots, n$ yielding a smoothed function s such that $y_i = s(x_i) + \varepsilon_i$, where ε_i are the residuals.

Lowess

The lowess method, described in detail by Cleveland (1979), is outlined here.

Let $0 < t \leq 1$ and r be the nearest integral value to nt . Then $100t$ is the called “tension.” Let d_k be the distance from x_k to its r th nearest neighbor.

Let

$$B(u) = \begin{cases} (1 - u^2)^2 & |u| < 1 \\ 0 & |u| \geq 1 \end{cases}$$

$$T(u) = \begin{cases} (1 - |u|^3)^3 & |u| < 1 \\ 0 & |u| \geq 1 \end{cases}$$

The smooth function s is produced by the steps:

1. For each $k = 1, \dots, n$ fit a line to the points (x_i, y_i) by weighted least squares using

$$\text{weights } w_i(x_k) = T\left(\frac{x_i - x_k}{d_k}\right). \text{ Denote the value of the fitted line evaluated at } x_k \text{ as } \hat{y}_k.$$

2. Let $e_i = y_i - \hat{y}_i$, and let σ be the median value of $|e_i|$. Define robustness weights

$$\delta_i = B(e_i / 6\sigma). \text{ Compute a new set of } \hat{y}_i \text{ s by repeating the fit procedure of step 1 using weights } \delta_i w_i(x_k).$$

3. Repeat step 2.

For any value u on the interval (x_1, x_n) , call the nearest bracketing x values x_i and x_j . Then the smoothed function $s(u)$ is the linear interpolation between the points (x_i, \hat{y}_i) and (x_j, \hat{y}_j) .

Supersmoother

The method, described in detail by Friedman (1984), is outlined here.

Order the data by ascending value of x .

Define the symmetric k -nearest neighbor smoothing function, $\hat{y}(k) = N(\mathbf{x}, \mathbf{y}; k)$ where for each $j = 1, \dots, n$ a line is fit by least squares to the points (x_j, y_j) for which $x_{j-k/2} \leq x_i \leq x_{j+k/2}$. The fit value, \hat{y}_j , is the value of the j th line evaluated at x_j .

Define the cross-validated residuals for a span of k neighbors centered about x_j

$$r_{(j)}(k) = \frac{r_j(k)}{1 - H_{jj}(k)}$$

where $r_j(k) = y_j - \hat{y}_j(k)$, and $H_{jj} = \frac{1}{k} + \frac{(x_j - \bar{x})^2}{V}$ with \bar{x} the average and V the variance of the x_i in the span.

The steps in the procedure are as follows:

1. Perform nearest neighbor smoothing for spans including $\frac{1}{20}$, $\frac{1}{5}$, and $\frac{1}{2}$ of the points.

Compute $\hat{y}(k_b) = N(\mathbf{x}, \mathbf{y}; k_b)$ with $k_1 = \frac{n}{20}$, $k_2 = \frac{n}{5}$, and $k_3 = \frac{n}{2}$.

2. Smooth the cross-validated residuals from step 1 with a k_2 span and choose, point by point, the span value that produces the smallest smoothed residual. Compute $r'_{(i)}(k_b) = N_i(\mathbf{x}, \mathbf{r}_{(i)}(k_b); k_2)$ for each of the three k 's, where $\mathbf{r}_{(i)}$ denotes the vector of $r_{(i)}$'s. Then form

$$\kappa_i = \begin{cases} k_1 & (\tilde{r}_i = r'_{(i)}(k_1)) \\ k_2 & (\tilde{r}_i = r'_{(i)}(k_2)) \\ k_3 & (\tilde{r}_i = r'_{(i)}(k_3)) \end{cases}$$

where $\tilde{r}_i = \min(r'_{(i)}(k_1), r'_{(i)}(k_2), r'_{(i)}(k_3))$.

3. Smooth the vector of best spans. $\kappa'_i = N_i(\mathbf{x}, \kappa; k_2)$.
4. Use the smoothed best spans to interpolate between the smoothed curves from step 1.

Let

$$f_i = \frac{k_2 - \kappa'_i}{k_2 - k_1}$$

$$g_i = \frac{k_3 - \kappa'_i}{k_3 - k_2}$$

$$\hat{y}_i = \begin{cases} f_i \hat{y}_i(k_1) + (1 - f_i) \hat{y}_i(k_2) & \kappa'_i \leq k_2 \\ g_i \hat{y}_i(k_2) + (1 - g_i) \hat{y}_i(k_3) & \kappa'_i > k_2 \end{cases}$$

For any value u on the interval (x_1, x_n) call the nearest bracketing x values x_i and x_j . Then the smoothed function $s(u)$ is the linear interpolation between the points (x_i, \hat{y}_i) and (x_j, \hat{y}_j) .

Cubic spline

The cubic spline is computed assuming the natural boundary conditions on the end points,

$$\text{i.e., } y''(x_1) = 0 \text{ and } y''(x_n) = 0 \text{ where } y'' = \frac{\delta^2 y}{\delta x^2}.$$

QC Subgroup Measurements

Exact probabilities or critical values for constants are calculated wherever possible and practical. Those probabilities or constants taken from tables are noted below.

Sigma

Sigma (σ), which is the estimate of the process standard deviation, is computed in one of two ways. It may be based on subgroup standard deviations:

$$\sigma = \frac{s_g}{c_4(k)}$$

or it may be based on subgroup ranges:

$$\sigma = \frac{\sum \frac{f_i r_i}{d_2(n_i)}}{\sum f_i}$$

where s_g is the square root of a weighted average of the subgroup variances

$$= \sqrt{\frac{\sum_i \sum_j (x_{ij} - \bar{x}_i)^2}{\sum_i (n_i - 1)}}$$

$c_4(k)$ is an unbiasing constant with k degrees of freedom

$$= \sqrt{\frac{2}{k-1}} \frac{\Gamma(k/2)}{\Gamma((k-1)/2)}$$

$$\Gamma(k) = (k-1)\Gamma(k-1)$$

$$\Gamma(1) = 1$$

$$\Gamma(1/2) = \sqrt{\pi}$$

and k degrees of freedom usually = total number of observations – number of subgroups + 1.

In the equations above,

$$f_i = \frac{(d_2(n_i))^2}{d_3(n_i)}$$

r_i = the range of subgroup i ; n_i is the number of observations in subgroup i ; $d_2(k)$ is the expected value of the range of k normal observations with standard deviation 1 (this value is read from a table); $d_3(k)$ is the standard deviation of the range of k normal observations with standard deviation 1 (also read from a table).

Xbar analyses

The center line is computed as $cl = \mu$, where μ is the mean of all measurements from all subgroups.

If control limits are based on k -sigma, then

$$UCL = cl + \frac{k\sigma}{\sqrt{n_i}}$$

$$LCL = cl - \frac{k\sigma}{\sqrt{n_i}}$$

If control limits are based on alpha, then

$$UCL = cl + z_{\alpha/2} \frac{\sigma}{\sqrt{n_i}}$$

$$LCL = cl - z_{\alpha/2} \frac{\sigma}{\sqrt{n_i}}$$

where $z_{\alpha/2}$ is the standardized normal score.

R analyses

The center line is computed as $cl = d_2(n_i)\sigma$.

If control limits are based on k -sigma, then

$$UCL = cl + kd_3(n_i)\sigma$$

$$LCL = cl - kd_3(n_i)\sigma$$

unless LCL is < 0 , in which case $LCL = 0$.

If control limits are based on alpha, then

$$UCL = D_{1-\alpha/2}(n_i)\sigma$$

$$LCL = D_{\alpha/2}(n_i)\sigma$$

where values of D with n_i degrees of freedom are retrieved from a table from Harter (1960). Note that StatView retrieves values of D for only nine values of α : 0.4, 0.2, 0.1, 0.05, 0.02, 0.01, 0.002, 0.001 and 0.0002. For all values of α within this range, D is interpolated by the method suggested by Harter. For α less than 0.0002, StatView uses the value for $\alpha = 0.0002$. For the most accurate results, avoid interpolated values by setting α to one of the values given above.

S analyses

The center line is computed as $cl = c_4(n_i)\sigma$.

If control limits are based on k -sigma, then

$$UCL = cl + kc_5(n_i)\sigma$$

$$LCL = cl - kc_5(n_i)\sigma$$

unless $LCL < 0$, in which case $LCL = 0$.

In the equations above, $c_5(k)$ is an unbiasing constant. If s is a sample standard deviation with k degrees of freedom, then the standard deviation

$$\frac{s}{c_5(k+1)} = \sigma$$

Therefore,

$$c_5(k) = \sqrt{1 - (c_4(k))^2}$$

If control limits are based on alpha, then

$$UCL = \sigma \sqrt{\frac{\chi_{1-\alpha/2, n_i-1}^2}{n_i-1}}$$

$$LCL = \sigma \sqrt{\frac{\chi_{\alpha/2, n_i-1}^2}{n_i-1}}$$

where χ^2 is the chi-square value of indicated probability with $n_i - 1$ degrees of freedom.

CUSUM analyses

The high and low cumulative sums are calculated as

$$S_{Hi} = z_i - k + S_{Hi-1}$$

$$S_{Li} = -z_i - k + S_{Li-1}$$

S_{Hi} and S_{Li} are independently set to 0 if their computed values are < 0 .

In the equations above,

$$z_i = \frac{(\mu_i - \mu)\sqrt{n_i}}{\sigma}$$

μ_i is the mean of the i th subgroup, μ is the mean from all measurements and k is dev/2, where dev is the magnitude (in standard units) of the mean shift to be detected, as specified by the user.

Capability analyses

$$C_p = \frac{USL - LSL}{6\sigma}$$

$$C_{pm} = \frac{\min(T - LSL, USL - T)}{3\sqrt{s^2 + \frac{n(\mu - T)^2}{n-1}}}$$

$$CPU = \frac{USL - \mu}{3\sigma}$$

$$CPL = \frac{\mu - LSL}{3\sigma}$$

$$C_{pk} = \min(CPU, CPL)$$

$$k = \frac{2|m - \mu|}{USL - LSL}$$

$$\% > USL(\text{observed}) = 100 \left(\frac{\# \text{ obs} > USL}{n} \right)$$

$$\% > USL(\text{expected}) = 100 \times P\left(Z > \frac{USL - \mu}{s}\right)$$

$$\% < LSL(\text{observed}) = 100 \left(\frac{\# \text{ obs} < LSL}{n} \right)$$

$$\% < LSL(\text{expected}) = 100 \times P\left(Z < \frac{LSL - \mu}{s}\right)$$

Where USL and LSL are the user-specified upper and lower specification limits, respectively; s is the standard deviation of all measurements:

$$s = \sqrt{\frac{\sum (x_i - \mu)^2}{n-1}}$$

T is the user-specified target value for the process; m is the average of USL and LSL; and Z is a standard normal variable.

QC Individual Measurements

Sigma

For individual measurements, σ , when computed, can be based on one of two different methods of calculation. The default method is based on the standard deviation of the measurements:

$$\sigma = \frac{s}{c_4(n)}$$

Alternately, the more traditional method is based on the average moving range of the measurements:

$$\sigma = \frac{\overline{MR}}{d_2(rs)}$$

In the equations above, \overline{MR} is the average of the moving ranges of the data over a window (range span) of size rs :

$$\overline{MR} = \frac{\sum_i MR_i}{n - rs + 1}$$

where $MR_i = \text{Range}(x_{i-rs+1}, x_{i-rs+2}, \dots, x_i)$, i.e., MR_i is the moving range for the i th measurement; x_i is the i th included measurement in the dataset; n is the total number of included measurements in the dataset; and $d_2(rs)$ is the expected value of the range of rs normal observations with the standard deviation assumed to be 1. It is read from a table.

In the equation for σ based on the standard deviation of the measurements, $c_4(n)$ is the same unbiasing constant as used in the subgroup measurement calculations.

I analyses

All center lines and control limits are computed in the same way as the equivalent values for Xbar analyses, above.

MR analyses

The center line is computed as $cl = \overline{MR}$.

If control limits are based on k -sigma, then

$$\text{UCL} = \text{cl} + k\sigma d_3(rs)$$

$$\text{LCL} = \text{cl} - k\sigma d_3(rs)$$

If control limits are based on alpha, then

$$\text{UCL} = D_{1-\alpha/2}(rs)\sigma$$

$$\text{LCL} = D_{\alpha/2}(rs)\sigma$$

Note that this is closely related to the formula that is used to calculate alpha-based control limits for R charts. As is the case for R charts, D is retrieved from a table. The same cautions regarding interpolated values apply here as well.

CUSUM and capability analyses

See CUSUM and capability analysis computations for subgroup measurements, above.

QC P/NP

p analyses

The center line is computed as $\text{cl} = p$, where p is the total number of nonconforming items divided by the total number of items for all subgroups.

If control limits are based on k -sigma, then

$$\text{UCL} = \min\left(\text{cl} + k\sqrt{\frac{p(1-p)}{n_i}}, 1\right)$$

$$\text{LCL} = \max\left(\text{cl} - k\sqrt{\frac{p(1-p)}{n_i}}, 0\right)$$

If control limits are based on alpha, then

$$\text{LCL} = \frac{\text{LCL}(np)}{n_i}$$

$$\text{UCL} = \frac{\text{UCL}(np)}{n_i}$$

where $\text{LCL}(np)$ and $\text{UCL}(np)$ are defined below.

np analyses

The center line is computed as $\text{cl} = n_i p$ unless it is specified in the Lines dialog box.

If control limits are based on k -sigma, then

$$\text{UCL} = \min(\text{cl} + k\sqrt{n_i p(1-p)}, n_i)$$

$$\text{LCL} = \max(\text{cl} - k\sqrt{n_i p(1-p)}, 0)$$

If control limits are based on alpha, LCL is calculated by setting

$1 - \alpha/2 = I_p(\text{LCL}, n_i + 1 - \text{LCL})$ and then solving for LCL, where $I_p(\alpha, \beta)$ is the incomplete

beta function, $I_p(\alpha, \beta) = \frac{\int_0^p t^{\alpha-1} (1-t)^{\beta-1} dt}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}}$. Similarly, UCL is calculated by setting

$\frac{\alpha}{2} = I_p(\text{UCL}, n_i + 1 - \text{UCL})$ and then solving for UCL. Note that for all p/n analyses with

alpha-based control limits, LCL is set to 0 for all values of $\alpha < 2(1-p)^n$.

QC C/U

c analyses

The center line is computed as $\text{cl} = n_i u$ where u is the average number of nonconformities per inspection unit over all subgroups, and n_i is the number of inspection units in the i th subgroup.

If control limits are based on k -sigma, then

$$\text{UCL} = \text{cl} + k\sqrt{n_i u}$$

$$\text{LCL} = \max(\text{cl} - k\sqrt{n_i u}, 0)$$

If control limits are based on alpha, LCL is calculated by setting $1 - \frac{\alpha}{2} = P(\text{LCL} + 1, n_i u)$ and

then solving for LCL, where $P(\alpha, \beta)$ is the incomplete gamma function:

$$P(\alpha, \beta) = \frac{\int_0^\beta e^{-t} t^{\alpha-1} dt}{\Gamma(\alpha)}$$

Similarly, UCL is calculated by setting $\frac{\alpha}{2} = P(\text{UCL} + 1, n_i u)$ and then solving for UCL.

u analyses

The center line is computed as $\text{cl} = u$.

If control limits are based on k -sigma, then

$$\text{UCL} = \text{cl} + k \sqrt{\frac{u}{n_i}}$$

$$\text{LCL} = \max\left(\text{cl} - k \sqrt{\frac{u}{n_i}}, 0\right)$$

If control limits are based on alpha, then

$$\text{LCL} = \frac{\text{LCL}(c)}{n_i}$$

$$\text{UCL} = \frac{\text{UCL}(c)}{n_i}$$

where $\text{LCL}(c)$ and $\text{UCL}(c)$ are defined above.

Note that for all c/u analyses with alpha-based control limits, LCL is set to 0 for all values of $\alpha < 2e^{-n_i u}$.

References

Suggested Reading

- Belsey, D.A., Kuh, E., Welsch, R.E. 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley, New York.
- Bock, R.D. 1975. *Multivariate Statistical Methods in Behavioral Research*. McGraw-Hill, New York.
- Freedman, D., Pisani, R., and Purves, R. 1978. *Statistics*. W.W. Norton & Company, New York.
- Johnston, J. 1984. *Econometric Methods*. 3rd ed. McGraw-Hill, New York.
- Lehmann, E.L. 1975. *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, San Francisco; McGraw-Hill, New York.
- Snedecor, G.W. and Cochran, W.G. 1989. *Statistical Methods*. Iowa State University Press, Ames. Iowa.
- Steel, R.G.D. and Torrie, J.H. 1980. *Principles and Procedures of Statistics: a Biometrical Approach*. McGraw-Hill, New York.

General

- Afifi, A. and Azen, S. 1979. *Statistical Analysis: A Computer Oriented Approach*. Academic Press, New York.
- Andrews, D.F., and Herzberg, A.M. (eds.). 1985. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. Springer-Verlag, New York.
- Bishop, Yvonne, M.M., Fienberg, S.E., and Holland, P.W. 1975. *Discrete Multivariate Analysis: Theory and Practice*. M.I.T. Press, Cambridge, Mass., London.
- Chambers, J.M., Cleveland, W.S. Kleiner, B., and Tukey, P.A. 1983. *Graphical Methods for Data Analysis*. Wadsworth Statistics/Probability Series, Belmont, California.
- Cleveland, W.S. 1979. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*. 74:368. 829–36.

- Cleveland, W.S. 1981. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *The American Statistician*. 35. 54.
- Cleveland, W.S. 1985. *The Elements of Graphing Data*. Wadsworth Advanced Book Program, Monterey, California.
- Cochran, W.G. and Cox, G.M. 1957. *Experimental Designs*. John Wiley & Sons, New York.
- Draper, N. and Smith, H. 1981. *Applied Regression Analysis*. 2nd ed. John Wiley & Sons, New York.
- Dunn, O.J. 1961. Multiple Comparisons Among Means. *Journal of the American Statistical Association*. 56. 52–64.
- Dunnett, C.W. 1955. A Multiple Comparison Procedure for Comparing Several Means with a Control. *Journal of the American Statistical Association*. 50. 1096–1121.
- Dunnett, C.W. 1964. New Tables for Multiple Comparisons with a Control. *Biometrics*. 20, 482–491.
- Dunnett, C.W. 1980a. Pairwise Multiple Comparisons in the Homogeneous Variance, Unequal Sample Size Case. *Journal of the American Statistical Association*. 75. 789–95.
- Dunnett, C. W. 1980b. Pairwise Multiple Comparisons in the Unequal Variance Case. *Journal of the American Statistical Association*. 75. 796–800).
- Everitt, B.S. 1977. *The Analysis of Contingency Tables*. Chapman and Hall Ltd., London.
- Friedman, J.H. 1984. A Variable Span Smoother. *Technical Report No. 5, Laboratory for Computational Statistics*. Stanford University, California.
- Games, P.A., and Howell, J.F. 1976. Pairwise Multiple Comparison Procedures with Unequal n 's and/or Variances: A Monte Carlo Study. *Journal of Educational Statistics*. 1. 113–125.
- Games, P.A., Keselman, H.J., and Rogan, J.C. 1981. Simultaneous Pairwise Multiple Comparison Procedures for Means When Sample Sizes are Unequal. *Psychological Bulletin*. 90. 594–98.
- Goodnight, J.H. 1979. A Tutorial on the SWEEP Operator. *The American Statistician*. 33. 149–158.
- Hocking, R.R. 1985. *The Analysis of Linear Models*. Brooks/Cole, Monterey, California.
- Hollander, M. and Wolfe, D. 1973. *Nonparametric Statistical Methods*. John Wiley & Sons, New York.
- Jaccard, J., Becker, M.A., and Wood, G. 1984. Pairwise Multiple Comparison Procedures: A Review. *Psychological Bulletin*. 96. 589–596.
- Kendall, M. and Stuart, A. 1977. *Volume 1: Advanced Theory of Statistics*. Charles Griffin & Company, London.
- Keselman, H.J., and Rogan, J.C. 1978. A Comparison of the Modified-Tukey and Scheffé Methods of Multiple Comparisons for Pairwise Contrasts. *Journal of the American Statistical Association*. 73. 47–51.

- Kleinbaum, D.G. & Kupper, L.L. 1978. *Applied Regression Analysis and Other Multivariate Methods*. Duxbury Press, Wadsworth Publishing Company, Belmont, California.
- Kramer, C. Y. 1956. Extension of Multiple Range Tests to Group Means With Unequal Numbers of Replications. *Biometrics*. 12. 307–10.
- Milliken, G.A. and Johnson, D.E. 1984. *Analysis of Messy Data Volume 1: Designed Experiments*. Lifetime Learning Publications, Belmont, California.
- Montgomery, D. & Peck, E. 1982. *Introduction to Linear Regression Analysis*. John Wiley & Sons, New York.
- Neter, J., Wasserman, W., and Whitmore, G.A. 1988. *Applied Statistics*. 3rd edition. Allyn & Bacon, New York. 975–6.
- Satterwaite, F.E. 1946. An Approximate Distribution of Estimates of Variance Components. *Biometrics Bulletin*. 2. 110–14.
- Searle, S.R. 1971. *Linear Models*. John Wiley & Sons, New York.
- Scheffé, H. 1953. A Method for Judging All Contrasts in the Analysis of Variance. *Biometrika*. 40. 87–104.
- Siegel, S. 1956. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York.
- Simpson, G.G., Roe, A., and Lewontin, R.C. 1960. *Quantitative Zoology*. revised ed. Harcourt, Brace & Co., New York.
- Smith, H.F. 1936. The Problem of Comparing the Results of Two Experiments with Unequal Errors. *Journal of Scientific and Industrial Research*. 9. 211–12.
- Snedecor, G. and Cochran, W. 1980. *Statistical Methods*. Iowa State University Press, Ames, Iowa.
- Sokal, R.R. and Rohlf, F.J. 1981. *Biometry*. W.H. Freeman and Company, New York.
- Welch, B.L. 1949. Further Note on Mrs. Aspins Tables and on Certain Approximations to the Tabled Functions. *Biometrika*. 36. 293–96.
- Winer, B.J. 1971. *Statistical Principles in Experimental Design*. McGraw-Hill, New York.

Factor analysis

- Armstrong, J.S. and Soelberg, P. 1968. On the interpretation of factor analysis. *Psychological Bulletin*. 70(5):361.
- Bartlett, M.S. 1951. A further note on tests of significance in factor analysis. *British Journal of Psychology*. 4(1):1.
- Carroll, J.B. 1953. Approximating simple structure in factor analysis. *Psychometrika*. 18:23.
- Cattell, R.B. and Jaspers, J.A. 1967. A general plasmode (No. 30-10-5-2) for factor analytic exercises and research. *Multivariate Behavioral Research Monographs*. 67(3): 211.

- Cattell, R.B. 1966. The scree test for the number of factors. *Multivariate Behavioral Research*. 1(2):245.
- Gorsuch, R. 1983. *Factor Analysis*. Lawrence Erlbaum Publishers, Hillsdale, NJ.
- Guttman, L. 1954. Some necessary conditions for factor analysis. *Psychometrika*. 19:149.
- Harman, H. 1976. *Modern Factor Analysis*. 3rd ed. University of Chicago Press, Chicago.
- Harris, C.W. 1962. Some Rao-Guttman relationships. *Psychometrika*. 27:247.
- Harris, C.W. 1967. On factors and factor scores. *Psychometrika*. 32:363.
- Hofmann, R.J. 1975. Brief report: on the proportionate contributions of transformed factors to common variance. *Multivariate Behavioral Research*. 10(4):507.
- Hofmann, R.J. 1978. Complexity and simplicity as objective indices descriptive of factor solutions. *Multivariate Behavioral Research*. 13(1):247.
- Hofmann, R.J. Indices descriptive of factor complexity. *The Journal of Psychology*. 96:103, 107.
- Hofmann, R.J. 1978. The orthotran solution. *Multivariate Behavioral Research*. 13(1):99.
- Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*. 24: 417 and 498.
- Kaiser, H.F. 1970. A second generation little jiffy. *Psychometrika*. 35:401.
- Kaiser, H.F. 1965. Psychometric approaches to factor analysis. *Proceedings of the 1964 Invitational Conference on Testing Problems*. Educational Testing Service, 37, Princeton, NJ.
- Kaiser, H.F. 1958. The varimax criterion for varimax rotation in factor analysis. *Psychometrika*. 23: 187.
- Mulaik, S. 1972. *The Foundations of Factor Analysis*. McGraw Hill, New York.
- Saunders, D.R. 1962. Trans-varimax. *American Psychologist*. 17:395.
- Thurstone, L.L. 1947. *Multiple Factor Analysis*. University of Chicago Press, Chicago.
- Timm, N. 1975. *Multivariate Analysis with Applications in Education and Psychology*. Brooks/Cole, New York.
- Wilkinson, J.H. 1965. *The Algebraic Eigenvalue Problem*. Oxford University Press, London.

Logistic regression

- Grossman, S., M.Milos, I.S. Tekawa, and N.P. Jewell. 1989. Colonoscopic screening of persons with suspected risk factors for colon cancer: II. Past history of colorectal neoplasms. *Gastroenterology*. 96. 299–306.
- Hosmer, D. and S. Lemeshow. 1989. *Applied Logistic Regression*. John Wiley & Sons, Inc., New York.

MacMahon, B., S. Yen, D. Trichopoulos, K. Warren, and G. Nardi. 1981. Coffee and cancer of the pancreas. *New England Journal of Medicine*. 304(11): 630–33.

Survival analysis

Cox, D. R., and D. Oakes. 1984. *Analysis of Survival Data*. Chapman and Hall.

Embury, S. H., Elias, L., Heller, P.H., Hood, C. E., Greenberg, P. L., and Schrier, S.L. 1977. Remission maintenance therapy in acute myelogenous leukemia. *Western Journal of Medicine*. 126: 267–272.

Fleming, T. R., and Harrington, D.P. 1991. *Counting Processes and Survival Analysis*. John Wiley & Sons, New York.

Kalbfleisch, J. D., and Prentice, R.L. 1980. *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, New York.

Lawless, J.F. 1982. *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, New York.

Lee, E.T. 1992. *Statistical Methods for Survival Data Analysis*. John Wiley & Sons, New York.

Miller, R.G. 1981. *Survival Analysis*. John Wiley & Sons, New York.

Nelson, W. 1982. *Applied Life Data Analysis*. John Wiley & Sons, New York.

Peto, R. and Pike, M.C. 1973. Conservatism in the approximation $\sum (O - E)^2 / E$ in the logrank test for survival data or tumor incidence data. *Biometrics*. 29: 579–584.

Ragland, D.R. and Brand, R.J. 1988. Coronary heart disease mortality in the Western Collaborative Group study. *American Journal of Epidemiology*. 1217: 462–475.

Rosenman, R. H., Brand, R. J., Jenkins, C.D. *et al.* 1975. Coronary heart disease in the Western Collaborative Group study. *Journal of the American Medical Association*. 223: 872–877.

QC analysis

Grant, E. L., and Leavenworth, R.S. 1988. *Statistical Quality Control*. Sixth ed. McGraw-Hill, Inc., New York.

Harter, H.L. 1960. Tables of range and studentized range. *The Annals of Mathematical Statistics* 31.4: 1122–1147.

Nelson, L.S. 1984. The Shewhart control chart—tests for special causes. *Journal of Quality Technology* 16.4: 237–239.

Nelson, L.S. 1985. Interpreting Shewhart Xbar control charts. *Journal of Quality Technology* 17.2: 114–116.

Ryan, T.P. 1989. *Statistical Methods for Quality Improvement*. John Wiley & Sons, New York.

Western Electric Co., Inc. 1956. *Statistical Quality Control Handbook*. Western Electric Co., Inc., Newark, New Jersey.

Westgard, J. O., and Barry, P.L. 1986. *Cost-effective Quality Control: Managing the Quality and Productivity of Analytical Processes*. AACC Press, Washington, DC.

Glossary

accelerated failure time model Fully parametric survival regression models of the form $\log(T) = \mu + \beta'Z + \sigma W$. These are termed “accelerated” failure time models because a unit change in the covariate Z_j induces a multiplicative change of $e^{\beta_j Z_j}$ in the time to failure. Perhaps more accurately, this change should be termed an acceleration when β_j is negative and a deceleration when β_j is positive.

alpha The Type I probability that any individual value of a process statistic exceeds the control limits; more generally, the probability of Type I error, where a null hypothesis that should be accepted is rejected.

alphanumeric Comprised of letters and/or numerals.

analysis of variance (ANOVA) Analysis of variance determines the significance of the effects or factors in a model by calculating how much of the variability in the dependent variable can be explained by the effect in question. Other model types are ANCOVA (analysis of covariance), which includes continuous independent variables called covariates, MANOVA (multivariate analysis of variance), which includes more than one dependent variable, and MANCOVA (multivariate analysis of covariance), which includes more than one dependent variable and continuous independent variables.

argument A value on which a function operates. The arguments to a function can be constants, column names or formulas.

assignable causes Any nonrandom factors that affect the results of a process. Generally, the identification and elimination of these is one of the primary goals of any quality improvement program.

attribute Any descriptive characteristic of an item (e.g., an item’s color or texture). In spc analyses, the most common attribute of interest is whether or not an item is defective.

baseline The descriptor given to survival and hazard function estimates from a survival regression model for the case in which all covariates are equal to 0.

beta The probability of type II error (the error of accepting a null hypothesis when it is actually false).

between factor In repeated measures ANOVA, an independent variable to test for differences among groups, as opposed to a within factor, which is used to compare several measurements of the same quantity under different conditions or at various times.

binary Having two possible values.

bivariate graph A graph that plots the relationship between an X and Y variable. Can be displayed as a scattergram or line chart. (See also univariate graph.) Can include fitted lines: cubic spline, lowess, super-smoother, or linear regression.

case-control studies In many settings, it can be expensive or impossible to obtain random samples of the dependent variable, even at pre-specified values of the independent variables. To overcome these obstacles, case control studies sample separately a set of cases (e.g., individuals with the disease of interest) and a set of controls (e.g., individuals who do not have the disease).

cell A subset of your data. Specifically, the intersection of the groups in your data when several nominal variables are considered. A cell is defined by the group labels of

the nominal variables. Multiply the number of distinct group labels in all nominal variables to get the maximum number of cells in the resulting analysis.

cell plot A plot which compares the means or sums of related variables or groups. You can depict data as bar charts (often referred to as side-by-side bar charts), line charts, or point charts to compare variable to variable and group to group.

censor time The time elapsed from the onset of observation of a subject until the subject is censored. Censoring occurs when the subject can no longer be observed because, for instance, the subject drops out of the study or the study ends.

censored observation Any subject in a survival analysis for whom the event time measurement is incompletely known. StatView supports only right censoring. A censored observation is sometimes referred to as an incomplete observation.

censoring The exclusion of a subject from the risk set at time t because that subject's survival status is unknown at time t .

colinearity The condition of relatively high correlation among variables. In an extreme case of colinearity, a straight line describes perfectly the relationship between two variables. Colinearity between independent variables in a regression makes it impossible to discern their individual effect on a dependent variable.

compact variable An alternative to entering each observation's group label in a column, a compact variable is a way to use individual columns to identify the groups of a nominal variable. Created by selecting the columns and clicking the Compact button in the dataset or the variable browser. In a compact variable, all the data in a column must belong to the same group. This structure is required to define the within factor of a repeated measures model.

confidence interval A range of values such that there is a known probability that the true value of some quantity lies within that range. This probability is known as the confidence level, and must be stated before the confidence interval is calculated. For example, the 95% confidence interval for a mean represents a range of values within which we expect to find the true value of the mean 95% of the time.

continuous selection Selection (using the mouse) of cells, rows, columns or elements that are next to each other. Selected by clicking the first item and then dragging the mouse to the last item.

continuous data Continuous data can assume any numerical value over a given interval, e.g., data that describe persons' weights or height.

control limits Maximum and minimum values of a particular process statistic if the process is in control. Values of the process statistic beyond these limits are regarded as evidence that the process is out of control.

convergence criterion The minimum relative difference in likelihood functions for successive iterations of the model-fitting procedure. When the relative difference in likelihood functions for successive iterations falls below this value, the fitting procedure stops. Note that the fitting procedure also stops if the maximum number of iterations is reached before the convergence criterion is met.

covariate A covariate is an independent variable in a [M]AN[C]OVA model that is a continuous variable. A covariate is expected to behave as a regressor (where its values might be thought to predict the values of the dependent variable) and is added to the model to remove its effect so that the influence of the factors can be more accurately measured. In survival regression models, covariates are used to model the variation in the event time variable.

cubic spline A smoothing method for bivariate scattergrams that connects a series of third order polynomial regressions in a moving window of four data points.

degrees of freedom The degrees of freedom (often denoted “df” or “DF”) associated with a statistical calculation are the total number of parameters minus the number of “fixed” parameters in the calculation. For example, a statistic based on the sample mean for a dataset with n observations has $n - 1$ degrees of freedom. One of the n observations is considered “fixed,” because more than one observation is required to calculate the variance for the mean. The estimate of variance is required because without it one cannot estimate measures of certainty, and thus p values, about test statistics (such as t , F , chi-square, etc.).

dependent variable The dependent is the variable whose variation you want to explain through a relationship with the assigned independent variable(s). Dependent variables are often called “Y variables,” “response variables,” or “outcome variables.”

deviance In logistic regression, a multiple of the log of the likelihood function: $-2\log l(b_0, b_1)$.

dichotomous Having two possible values.

effect A term in an ANOVA, ANCOVA, MANOVA, or MANCOVA model. A main effect is a term that consists of a single variable treated as a factor. An interaction effect is a term that consists of a factor crossed with one or more other factors or covariates.

eigenvalue A value of lambda (λ) for which $Ax = \lambda x$ for $x \neq 0$ where A is a square matrix and x is a vector. Each eigenvalue is associated with a corresponding eigenvector. The eigenvectors corresponding to large eigenvalues are usually the most useful.

eigenvector A vector $x \neq 0$ for which $Ax = \lambda x$, where A is a square matrix and λ (lambda) is an eigenvalue of A . The eigenvectors of a correlation matrix are useful in determining which variables explain the variability seen in a dataset.

error bar The extension of a single point on a graph to reflect the variability of the quantity being estimated.

event An occurrence of special interest that marks the endpoint of an event time. In a survival analysis, the event is often failure or death of the subject.

event time The time that elapses from the onset of observation of a subject to the occurrence of the event of interest.

excess risk The absolute difference in risk or probability of an outcome when comparing exposed to unexposed individuals.

explanatory variable Another name for independent variable.

exponential distribution A special case of the Weibull distribution, with the scale parameter equal to one. For the exponential distribution to be applied to a parametric model, the hazard function must be constant.

exponential regression A nonlinear transformation of the basic linear regression model in the form $Y = b_0 + e^{b_1 X}$.

factor A factor is a single nominal variable in a linear model, such as an ANOVA. Factors by themselves or crossed with other factors comprise “effects” or “terms” in such models.

failure The term used as a synonym for **event** in a typical medical or engineering survival analysis in which subjects “fail” or die.

false signal Any indication, usually from control charts or tests for special causes, that a process is out of control when, in fact, the process is in control. See Type I error, below. The QC analyst tries to mini-

mize the occurrence of such signals, while maximizing the detection of true signals.

fitted value The values of the dependent variable generated by a regression equation when you calculate it using the values of the independent variables in your data.

frequency plots Graphical displays of the frequency distribution of a variable. StatView produces regular histograms, *z*-score histograms, and pie charts.

grid lines Lines that run across a graph perpendicular to an axis and mark the major and minor intervals along the axis.

group A collection of cases in a dataset that share the value of a nominal variable. All observations with the same value for the nominal variable are said to be in the same group level. For example, a nominal variable describing a person's gender divides the data into two group levels: male and female.

group label A name that identifies the distinct groups of a nominal variable. A label is also used to identify the groups of a category.

grouping variable A nominal variable that has a distinct value for each group in the dataset and thereby identifies the different groups in the data when the variable is used in an analysis.

growth regression A nonlinear transformation of the basic linear regression model in the form $Y = e^{b_0 + b_1 X}$.

hazard function The rate of occurrence of events at time *t*.

hazard function, cumulative Measures the cumulative risk to which an individual is exposed up to time *t*. The cumulative hazard function is equal to the negative log of the cumulative survival function.

histogram A bar chart that plots the distribution of a variable.

hypothesis testing A statistical technique for collecting data to answer questions

through the use of a statistical model. Each question is stated in the form of a null hypothesis, and the answer takes the form of either acceptance or rejection of the null hypothesis according to whether the *p* value of a test statistic is greater than or less than an appropriate significance level.

hypothesized value A value you suspect a particular statistic to have in the population you are studying. You can construct a hypothesis test to see if your hypothesized value is reasonable considering the data you collected.

in control The description for any process that produces items that vary within the limits proscribed by a particular statistical distribution.

independent variable The independent variable is used to explain the linear variation in the dependent variable. Note that multiple and stepwise regressions take more than one independent variable. Independent variables are sometimes called "X variables," "predictor variables," "design variables," or "explanatory variables."

informative data An informative variable is used for identification purposes only, e.g., a column containing names of patients in a study.

interaction effect An interaction effect is a term in an [M]AN[C]OVA model that consists of a factor crossed with one or more other factors or covariates, as opposed to a main effect, which is a single variable treated as a factor.

joint significance tests Statistical procedures for evaluating the probability that two or more covariates *together* (thus, "joint") make a significant contribution to a statistical model. These same procedures can be used to evaluate the contribution to a model of individual covariates as well, though in such cases, these procedures would be more accurately called individual significance tests.

k The constant by which sigma is multiplied to calculate the normal approximation of the control limits.

***k*-sigma** An expression that indicates that *k* multiples of sigma are used to compute upper and lower control limits about a center line.

lambda A quantity used to compute Power, Lambda is sometimes called “partial eta squared” or “noncentrality value.”

LCL The Lower Control Limit, the minimum value for a particular process statistic if the process is in control.

level A cell or group within a factor, interaction, categorical variable, or any other nominal or grouping variable. A collection of cases in a dataset that share the value of a nominal variable. All observations with the same value for the nominal variable are said to be in the same group level. For example, a nominal variable describing a person's gender divides the data into two groups or levels: male and female.

likelihood ratio test In logistic regression, a test of the relationship between an independent and dependent variable based on comparing the likelihood or deviance of a model including the independent variable of interest with that of a model excluding it.

log odds The log of a probability divided by one minus the same probability:
 $\log(p/(1 - p))$.

logarithmic regression A nonlinear transformation of the basic linear regression model in the form $Y = b_0 + b_1 \ln X$.

logistic regression A modeling technique analogous to linear regression that examines the relationship between a nominal outcome (or dependent) variable with one or more nominal or continuous independent variables.

loglogistic distribution The frequency distribution of a variable whose logarithm fol-

lows a logistic distribution. The hazard function of a parametric model based on a loglogistic distribution either always decreases with time or initially increases to a maximum and then decreases.

lognormal distribution The frequency distribution of a variable whose logarithm follows a normal distribution. For the lognormal distribution to be applied to a parametric model, the hazard function must be initially increasing and then decreasing.

lowess A locally weighted regression method for smoothing bivariate scattergrams. Its tension parameter indicates what percentage of the dataset's values should be included each the window for the smoothing. A higher number produces a tighter smooth (with less response to local variances); a lower number produces a looser smooth (that is more strongly influenced by local variances).

(M)AN(C)OVA [M]AN[C]OVA is shorthand for [Multivariate] Analysis of [CO]Variance; that is, it denotes ANOVA (analysis of variance), ANCOVA (analysis of covariance), MANOVA (multivariate analysis of variance), and MANCOVA (multivariate analysis of covariance). All are specific types of models within the general linear model (along with linear regression, etc.). All are models predicting the values of one or more dependent continuous variables from combinations of one or more factors (independent nominal variables) and/or covariates (independent continuous variables).

main effect A main effect is a term in a [M]AN[C]OVA model that consists of a single variable treated as a factor, as opposed to an interaction effect, which consists of a factor crossed with one or more other factors or covariates.

missing cell An intersection of the groups in combined nominal variables for which there is no data. You get missing cells in

your data when one of the combinations of groups among the nominal variables does not exist in the data.

missing value A case which has no value for a variable, either because none was available or because data were lost. A missing value is represented by a period (.).

model In statistics, any mathematical expression used to account for or “explain” the variation in at least one other variable.

nominal data Nominal data identify which the groups to which each observation belongs.

nonconforming item An item that is defective, i.e., an item that does not meet minimum standards of acceptability.

nonconformity Any attribute of an item that is unacceptable (e.g., scratches, dents, discolorations).

nonlinear regression Nonlinear regression analysis estimates a nonlinear (exponential, logarithmic, power, or growth) transformation of the linear regression model.

nonparametric Statistical procedures which make less restrictive assumptions about the population(s) from which the data were sampled.

null hypothesis A statement that a quantity has a particular value, or that several quantities are equal. The null hypothesis is the statement you are evaluating through your analysis of the data. It provides a basis for hypothesizing a known distribution for a statistic. You compare an observed value to the hypothesized value to see if the data supports the null hypothesis. If the test statistic seems unreasonable under the assumption of the null hypothesis, you can reject the null hypothesis in favor of some alternative, usually a statement which is the opposite of the null hypothesis. For example, the null hypothesis for an unpaired t -test is that there is no difference between the means of the two groups you are comparing. So, a rejection of the null hypothe-

sis means that the means of these two groups are not the same.

one-sided test A statistical test which considers the possibility of change or difference in only one direction. For example, a test of the hypothesis that one mean is equal to another mean versus an alternative hypothesis that the first mean is greater than the other mean. This is in opposition to the two-sided test which has an alternative hypothesis that the means are simply not equal. One-sided tests should only be performed when you have secure knowledge that a change in the other direction is physically impossible. The option to perform a one-sided test is available in the one sample inference, paired comparison and unpaired comparison.

out of control The description for any process that produces items that deviate from the expected patterns of variation that are consistent with a particular statistical distribution.

outcome variable Another name for dependent variable.

p value A value indicating the likelihood that the data used to carry out a statistical test would occur under a specified hypothesis. A p value represents the probability that a statistic would have a value at least as extreme as the one observed, assuming the hypothesis in question is true. Thus, with a low p value (less than 0.05, for example) it is unlikely that the hypothesis is reasonable; similarly a high p value indicates that the data does not contradict the null hypothesis. A low p value leads you to reject the null hypothesis.

paired comparison A comparison of two variables, both measured on each of several subjects.

polytomous Having many possible values.

population The collection of all possible units similar to the ones you are studying. The population is usually the group to

which you extend your results after your analysis is performed. A sample is a subset of a population.

post hoc test A post hoc test is used as a follow-up to a [M]AN[C]OVA analysis in which one or more effects are found to be significant (in other words, after the null hypothesis has been rejected) to determine how the groups of that effect differ.

power The ability of a statistical test to declare a true difference “statistically significant.” Its value is equal to 1 minus Beta, and its computation for [M]AN[C]OVA is based on a quantity called Lambda.

power regression A nonlinear transformation of the basic linear regression model in the form $Y = b_0X^{\beta_1}$.

process Any action or series of actions that generates a measurable result.

process statistic Computations that are used to infer characteristics of a process which typically are based on measured results of this process.

proportional hazards The assumption that, for different covariate values, the ratio of hazard functions is constant across all failure times.

raw data Raw data consists of the information originally obtained from a test, experiment or survey, before it has been summarized or condensed by any method.

recode To describe any change in the representation of a variable’s values by recoding continuous values to levels of a category or substituting computed values to replace missing values.

regression Regression analysis determines whether the values of one or more independent variables in a model can predict the values of a dependent variable.

regression line The line that describes the position of values predicted from a regression equation, with the independent variable plotted on the vertical axis and the

dependent variable plotted on the horizontal axis.

relative risk The risk or probability of an outcome for an exposed individual divided by the risk of an unexposed individual (e.g., a relative risk of ten in a smoking/lung cancer study would indicate that subjects who smoke are ten times more likely to develop lung cancer than subjects who don’t).

repeated measures analysis of variance An ANOVA model in which one or more of the independent variables are used to compare several measurements of the same quantity under different conditions or at various times. This independent variable is called a within factor. A repeated measures ANOVA can also contain one or more between factors to test for differences among groups.

residual The difference between the fitted value of the dependent variable in a regression and its actual value.

response variable Another name for dependent variable.

right censoring The exclusion of subjects from the risk set at times beyond t_c , because their survival status is unknown beyond t_c .

risk The probability that any individual will experience an event.

risk set The group of all subjects vulnerable to an event at time t . Excludes all subjects that have experienced the event or have been censored before time t .

sample The specific collection of units from which a dataset is derived. The units of a sample are usually a subset of the population.

scattergram A graph that represents data points as unconnected marks or dots on an X-Y plane (Cartesian coordinate system).

sigma The estimate of the standard deviation about a particular process statistic.

significance level A preset value, expressed as a probability between zero and one

(p value), used as a cutoff value in determining whether to reject a null hypothesis. Essentially, the significance level is an estimate of how often you will err by rejecting a hypothesis which is in fact true. A common significance level is 0.05, which means you are willing to be wrong one out of twenty times ($1/20 = 0.05$) when you reject the null hypothesis.

statistical process control (SPC) The use of statistics to identify the occurrence of assignable causes in processes.

strata In proportional hazards models, strata define groups having different baseline survival functions. In nonparametric models, strata define groups for which separate survival functions are estimated. In general, strata define groups whose behavior must be accounted for to maintain the validity of the model, but do not provide a basis for tests of significance. In this sense, strata define “nuisance” groups.

subgroup A natural division of observations from a population, with no observations repeated among equivalent divisions in the same population.

supersmoother Supersmoother is a smoothing method for bivariate scattergrams that uses a local cross-validation technique to determine how much smoothing is needed in each region along the X axis. It uses less smoothing in areas of greater curvature or lesser variance, and it uses more smoothing in areas of lesser curvature or greater variance.

survival function, cumulative The estimate of the proportion of individuals that have not experienced the event from time 0 to time t . Often referred to simply as the survival function.

survival status The state of subjects with respect to whether or not they have experienced the event.

tail The extreme region of a distribution curve for a particular variable or statistic. If

there are extreme values spread out over a large range, the distribution has long tails. The upper tail of a distribution refers to extremely large values; the lower tail refers to extremely small values.

tolerance A criterion for abortion of a model-fitting procedure that is dependent on colinearity among independent variables (covariates). The greater the colinearity among these variables, the more likely it is that they will fail to satisfy the conditions established by a given tolerance, and that the model-fitting procedure will be aborted.

Type I error The rejection of a true null hypothesis. For QC analysis: in SPC analyses, a Type I error occurs when a process is mistakenly identified as being out of control.

Type II error The acceptance of a null hypothesis when it is false.

UCL The Upper Control Limit, the maximum value for a particular process statistic if the process is in control.

unbalanced design A balanced ANOVA design is one in which the cells of each factor or combination of factors have an equal number of cases. An unbalanced ANOVA design is one in which the cells of each factor or combination of factors have differing numbers of cases.

uncensored observation Any subject in a survival analysis for which the entire duration of the event time measurement is known. This is sometimes referred to as a complete observation.

univariate graph A graph that presents one-dimensional data, with only a Y axis. Each individual observation is plotted.

unpaired comparison A comparison of the measurements of two distinct groups of equal or unequal size.

Wald test In logistic regression, a test of the relationship between an independent

and dependent variable based on comparing the estimate of the slope coefficient to its standard error.

Weibull distribution The generalization of the exponential distribution, with both scale and intercept parameters, to accommodate non-constant hazard functions. For the Weibull distribution to be applied to a

parametric model the hazard function must be a power of T .

within factor In repeated measures ANOVA, an independent variable used to compare several measurements of the same quantity under different conditions or at various times, as opposed to a between factor, which tests for differences among groups.

Index

SR=StatView Reference, US=Using StatView

Symbols

– [SR333](#), [SR335](#)
 # [US64](#)
 () [SR336–SR337](#)
 [] [SR337](#)
 * [SR79](#), [SR333](#)
 ** [SR334](#)
 + [SR333](#), [SR335](#)
 . *see missing values*
 / [SR334](#)
 /* [SR329](#)
 < [SR337](#), [SR340](#)
 <> [SR341](#)
 = [US255](#), [SR340](#)
 > [SR337](#), [SR341](#)
 ? [US113](#)
 [] [SR337](#)
 [] [SR337](#)
 ^ [SR334](#)
 { } [SR336](#)
 ≠ [US255](#), [SR341](#)
 ≤ [SR337](#), [SR340](#)
 ≥ [SR337](#), [SR341](#)
 π [SR400](#)
 ⊃ [SR345](#)
 ... [US64](#), [US113](#)
 ÷ [SR334](#)

Numerics

0 [SR345](#)
 1 [SR345](#)

A

abbreviations *see syntax*
 abort calculations [US138](#)
 abscissa *see X axis*
 absolute value [SR348](#)
 accelerated failure time [SRI72](#), [SR487](#)
 actuarial analyses [SRI48–SRI49](#),
 [SRI52–SRI53](#), [SRI61](#)
 Add Multiple Columns [US62](#)
 add results [US152](#)
 add variables [US132](#), [US144](#)
 multiple vs. compound
 results [US135–US137](#), [US170](#), [US229](#)
 tutorial example [US21](#)
 also see assign variables
 add vertex [US208](#), [US210](#)
 addition [SR333](#), [SR368](#), [SR374](#), [SR423–SR424](#)
 unary [SR335](#)
 adjusted R squared [SR56](#)
 adopt variable assignments [US131–US133](#)
 tutorial example [US29–US30](#)
 alert messages [US222](#), [US225](#)
 algorithms [SR433–SR480](#)
 Align Objects [US218](#)
 Align to Grid [US218](#)
 alignment [US185](#), [US199](#), [US214](#),
 [US217–US218](#)
 allow changes *see Unlock*
 AllRows [US108](#), [US124](#), [SR325–SR326](#)
 alpha
 control limits
 c/u analyses [SR299](#)
 individual measurements [SR279](#)
 p/np analysis [SR288](#)
 range or moving range [SR475](#)

- subgroup measurements [SR262](#)
 - error *see type I error, significance level*
 - alphabetize *see Sort*
 - alternate mouse button [USVII](#)
 - ambiguous data class [US51](#)
 - analyses
 - clone [US146](#)
 - compact variables [US95–US97](#)
 - example [US95](#), [SR91–SR93](#), [SR97–SR99](#), [SR101](#), [SR105](#), [SR107](#)
 - create [US131–US133](#)
 - by hand or with templates [US131](#)
 - exercises [US150–US156](#)
 - tutorial example [US20–US27](#)
 - dialog box hints [US222](#)
 - multiple vs. compound
 - results [US135–US137](#), [US170](#), [US229](#)
 - objects [US21–US22](#)
 - overview [US131](#), [SRIV–SRX](#)
 - parameters [US24](#), [US134–US135](#), [US142](#)
 - tutorial example [US31](#)
 - template exercises [US175](#)
 - tutorial example [US14–US40](#)
 - variable requirements [US145](#)
 - variable requirements *also see data re-requirements under specific analysis*
- analysis browser [US141–US143](#)
 - exercises [US150–US156](#)
 - open, close [US43](#)
- analysis generated variables
 - correlation matrix [SR46](#)
 - data source [US77](#)
 - factor scores [SR135](#)
 - regression [SR61](#)
 - survival regression analysis [SR177](#)
- analysis of variance [US235](#)
 - analysis of covariance models [SR78](#), [SR80–SR81](#), [SR99–SR101](#)
 - analysis of variance models [SR77](#), [SR79](#), [SR96–SR97](#)
 - data requirements [SR90–SR95](#)
 - dialog box [SR89–SR90](#)
 - discussion [SR73–SR89](#)
 - exercises [SR96–SR110](#)
 - hypothesis testing [SR74–SR76](#)
 - interaction plots [SR90](#), [SR96–SR99](#), [SR104](#), [SR107](#)
 - Latin square [SR105](#), [SR107](#)
 - means tables [SR78](#)
 - model building [SR76–SR78](#)
 - multivariate models [SR81–SR82](#), [SR108–SR110](#)
 - nonparametric [SR121–SR122](#)
 - post hoc tests [SR103](#)
 - randomized complete
 - block [SR101–SR105](#)
 - repeated measures models [US85–US86](#), [SR82–SR83](#), [SR88–SR89](#), [SR91–SR93](#), [SR97–SR99](#)
 - results [SR95](#)
 - templates [SR96](#)
 - tutorial example [US38–US40](#)
- analysis windows [US139–US148](#)
 - View, Window menus [US148–US149](#)
also see analysis browser, results browser, variable browser, views
- Analyze menu
 - New View [US132](#), [US139](#), [US150](#)
 - rearrange [US168–US169](#)
 - Rebuild Template List [US168](#)
 - templates [US162](#), [US174](#)
- analyze subsets [US149–US150](#)
- ANCOVA *see analysis of variance*
- AND [SR345](#)
- angle [US185](#), [US199](#)
- ANOVA *see analysis of variance*
- Apple Guide [US48](#), [US221](#), [US223](#)
- application preferences [US225–US226](#)
- Arabic characters [SR379](#), [SR384](#), [SR423](#)
- arc functions [US112](#)
- arc tool [US205–US207](#)
- ArcCos [SR349](#)
- ArcCosh [SR349](#)
- ArcCot [SR350](#)
- ArcCsc [SR351](#)
- ArcSec [SR352](#)
- ArcSin [SR353](#)
- ArcSinh [SR354](#)
- ArcTan [SR355](#)
- ArcTanh [SR356](#)
- arguments [US113](#), [SR323–SR326](#), [SR331](#)
- arithmetic operators [US112](#)
- arrange results [US43](#), [US213–US214](#)
- arrow tool [US212](#)

tutorial example [US45](#)
also see selection tool
 ascending sort *see Sort*
 assign variables [US144](#)
 dialog box [US158–US159](#), [US222](#)
 exercises [US150–US156](#)
 from other analyses *see adopt*
 templates [US162](#), [US164](#)
 variables
 assign from other analyses *see adopt*
 assignable causes [SR252](#), [SR254](#)
 association *see correlation, covariance*
 asterisk [SR79](#), [SR329](#), [SR333](#)
 double [SR334](#)
 attribute pane [US58–US60](#), [US73–US80](#),
 [US255](#)
 change attributes [US59](#)
 control [US5](#), [US60](#)
 set attributes [US58–US59](#)
 show [US5](#)
 tutorial example [US5](#)
 autocorrelation [SR363](#)
 automate analyses *see templates*
 Average [SR356](#)
 average [SRI](#), [SR388](#), [SR391](#)
 AverageIgnoreMissing [SR357](#)
 avoid errors [US162](#)
 axis
 bounds [US185](#), [US188](#), [US191](#)
 cell dialog box [US192](#)
 colors [US197](#)
 decimal places [US188](#), [US192](#)
 frames [US183](#), [US187](#), [US197](#), [US229](#)
 grid lines [US185](#), [US192](#)
 labels [US183](#), [US186](#)
 logarithmic and linear scales [US192](#)
 move [US186](#)
 numeric dialog box [US190](#)
 numeric formats [US192](#), [US229](#)
 ordinal dialog box [US193](#)
 rotate text [US190](#)
 select [US185](#)
 three types [US190](#)
 tick marks [US185](#), [US191–US193](#)
 transpose [US188](#)
 values [US183](#)

B

background calculation [US138](#)
 background colors [US212](#)
 backward stepwise regression *see regression*
 Balloon help [US48](#), [US221–US222](#), [US224](#)
 bar charts
 fill patterns [US195](#)
 frequency distribution [SRI3](#)
 univariate plots [SR217](#)
 also see cell plots
 Bartlett's chi-square [SRI38](#)
 Bartlett's test of sphericity [SR44](#)
 Bartlett's test template [US235–US237](#)
 baseline cumulative hazard plot [SRI89](#)
 baseline cumulative survival plot [SRI87](#)
 baseline estimates, Kaplan-Meier [SRI75](#)
 baseline hazard [SRI69](#)
 baseline ln cumulative hazard plot [SRI89](#)
 baseline survival table [SRI88](#)
 Basic Statistics [US142](#)
 batch mode *see templates*
 beep for error messages [US225](#)
 bell-shaped curve [SRI7](#)
 Bernoulli distribution [SR401](#)
 beta distribution [SR406](#)
 beta *see type II error*
 between subjects [SR83](#)
 Bezier curves *see spline tool*
 bimodal distribution [US32](#)
 binary logistic regression [SRI99](#)
 binary operators *see operators*
 binomial distribution [SR287](#), [SR401](#), [SR406](#)
 BinomialCoeffs [SR358](#)
 bivariate plots [SR221–SR236](#)
 axis types [US190](#)
 confidence intervals [US242](#), [SR221](#),
 [SR224](#), [SR228](#), [SR232](#)
 correlation [SR31](#), [SR48](#)
 cubic spline [SR221](#), [SR225](#), [SR227–SR228](#),
 [SR233](#)
 data requirements [SR229](#)
 dialog box [SR228](#)
 discussion [SR221–SR228](#)
 error bars [US242](#)
 exercises [SR35](#), [SR230–SR236](#)
 fitted lines [SR221–SR228](#)

- interaction plots [SR99](#)
 - lowess [SR221](#), [SR225–SR226](#), [SR235](#)
 - multiple variables [SR222](#)
 - nominal data [SR232](#)
 - results [SR229](#)
 - split-by variables [SR222](#)
 - strategy [SR222–SR223](#)
 - supersmoother [SR221](#), [SR225–SR227](#), [SR236](#)
 - templates [SR230](#)
 - exercise [US171](#)
 - black selection handles [US21](#), [US184](#), [US186](#), [US198–US199](#), [US204](#), [US207](#), [US209–US210](#), [US214](#)
 - blocking factor [SR101](#), [SR105](#)
 - Bonferroni/Dunn [SR86–SR87](#), [SR104](#)
 - Boolean operators [US127](#)
 - Boolean variables [SR338](#)
 - borders
 - dialog box [US201](#)
 - tables [US199](#), [US201](#), [US231](#)
 - box plots [US96](#), [US133](#), [US135](#), [US137](#), [SR243](#)
 - axis labels vs. legend text [SR245](#)
 - change style [US188](#)
 - data requirements [SR244](#)
 - dialog box [SR243](#)
 - discussion [SR243](#)
 - exercise [SR244](#)
 - results [SR244](#)
 - subgroups [US244](#)
 - templates [SR244](#)
 - tutorial example [US29](#)
 - BoxCox [SR358](#)
 - braces [SR336](#)
 - brackets [SR337](#)
 - breakpoints *see Recode*
 - Breslow-Gehan-Wilcoxon test [SR151](#), [SR156](#)
 - browser *see analysis browser, formula browser, results browser, triangle controls, variable browser*
- C**
- C class marker *see class marker*
 - C usage marker *see usage markers*
 - c/u analyses *see QC c/u analysis*
 - calculations
 - background [US138](#)
 - cancel [US138](#)
 - control [US138–US139](#)
 - precision [US73](#)
 - save results with view [US141](#)
 - calculator keypad [US112](#)
 - cancel calculations [US138](#)
 - Candy Bars Data [US2–US48](#)
 - capability analysis [SR252](#), [SR254–SR255](#), [SR261](#)
 - CAPA dialog box [SR263](#), [SR267](#)
 - CAPA table [SR272](#)
 - capability indices [SR262](#)
 - C_{pk} [SR261](#)
 - C_{pm} [SR261](#)
 - example [SR275](#)
 - indices [US244](#), [SR254](#), [SR476](#)
 - individual measurements analysis [SR279](#)
 - k (centering index) [SR262](#)
 - parameters [SR267](#)
 - caret [SR334](#)
 - case number [SR417](#)
 - case-control studies [SR203](#)
 - case-sensitive [US255](#)
 - casewise operation [SR321–SR323](#), [SR331–SR332](#)
 - categories [US74](#), [US80–US84](#)
 - add [US91](#)
 - advantages [US80](#)
 - compact variables [US91](#)
 - create [US81](#)
 - example [US92](#)
 - data type [US73](#), [SR320](#)
 - delete [US84](#), [US91](#)
 - disadvantages [US80](#)
 - edit [US83–US84](#)
 - enter data [US83](#)
 - how StatView uses order [US238–US240](#)
 - import [US104](#), [US254](#)
 - multiple [US254](#)
 - nominal data class [US74](#), [US78](#)
 - problems from editing [US256](#)
 - recode [US118](#), [US238](#), [US255](#)
 - reorder levels [US238–US240](#)
 - required [US81](#), [US91](#), [US118](#)
 - tutorial example [US33](#)
- CDF

- Bernoulli [SR401](#)
- binomial [SR401](#)
- chi-square [SR402](#)
- F [SR402](#)
- inverse
 - chi-square [SR414](#)
 - F [SR415](#)
 - normal [SR415](#)
 - t [SR416](#)
- normal [SR403](#)
- t [SR404](#)
- Ceil [SR359](#)
- cell axes [US190](#)
- dialog box [US192](#)
- labels [US192](#)
- tick marks [US192](#)
- cell normality [SR84](#)
- cell plots [SR237](#)
 - axis labels vs. legend text [SR240](#)
 - cell bar chart picture [SR240](#)
 - cell point chart picture [SR241](#)
 - data requirements [SR238](#)
 - dialog box [SR237](#)
 - discussion [SR237](#)
 - exercise [SR239](#)
 - results [SR239](#)
 - templates [SR239](#)
- censor [SR488](#)
 - nonparametric analyses [SRI47](#), [SRI57](#)
 - pattern plot [SRI61](#)
 - regression methods [SRI75](#), [SRI84](#)
- center-justify shapes [US205](#), [US218](#)
- central limit theorem [SR258](#)
- central tendency
 - Average [SR357](#)
 - AverageIgnoreMissing [SR357](#)
 - GeometricMean [SR380](#)
 - HarmonicMean [SR382](#)
 - Mean [SR388](#)
 - Median [SR389](#)
 - Mode [SR391](#)
 - TrimmedMean [SR428](#)
- change
 - analysis parameters [US134–US135](#)
 - appearance of results [US135](#)
 - criteria [US240](#)
 - data class [US51](#), [US78](#)
 - data source [US77](#)
 - data type [US75–US76](#)
 - formulas [US240](#)
 - templates [US171](#)
 - variable names [US58](#)
- characteristic roots [SRI32](#)
- Chinese characters [SR379](#), [SR384](#), [SR423](#)
- chi-square [SR24](#)
 - contingency tables [SRI12](#)
 - data requirements [SR25](#)
 - distribution [SR402](#), [SR407](#)
 - results [SR26](#)
- choose group(s) *see Criteria*
- ChooseArg [SR359](#)
- chords [USVI](#)
- class markers [US110](#), [US143](#), [US163–US164](#)
 - compact variables [US89](#), [US93](#)
 - tutorial example [US21](#)
- class *see data class*
- classify results *see Split By*
- Clean Up Items [US159](#), [US213](#)
 - tutorial example [US43](#)
- Clear [US65](#), [US181–US182](#)
- Clipboard [US66](#), [US181](#)
 - import pictures, text [US210](#)
 - transfer data [US66](#)
- clone analyses [US131–US133](#), [US146](#)
 - tutorial example [US26–US27](#)
- closed interval [SR337](#)
- closed polygon [US207](#)
- closed spline [US208](#)
- coded raw data [US51](#), [US84](#), [SRI14](#)
- coded summary data [US52–US53](#), [SRI15](#)
- coefficient correlations table [SRI89](#)
- coefficient covariances table [SRI89](#)
- coefficient of determination *see R squared*
- coefficient of variation [US60](#), [SR4](#)
- CoeffOfVariation [SR360](#)
- colinearity [SR53](#), [SR58](#)
- collection of analyses *see templates*
- color palette preferences [US226–US227](#)
- colors
 - axis [US197](#)
 - graph frame [US185](#)
 - graph text [US197](#)
 - graphs [US229](#)
 - grid [US217](#)

- page break [US217](#)
- plots [US197](#)
- preferences [US226–US227](#)
- shapes [US212](#)
- table text [US199](#)
- tables [US199](#), [US202](#)
- tutorial example [US43](#)
- view background [US212](#)
- column attributes *see attributes*
- column charts *see cell plots*
- columns
 - add multiple columns [US62](#)
 - insert [US62](#)
 - labels [US199](#)
 - selecting [US64](#)
 - transpose into rows [US68](#)
 - vs. variables [US53](#)
 - widths in tables [US199](#)
- columnwise operation [US108](#), [US124](#),
[SR321–SR323](#), [SR325–SR326](#)
- Combinations [SR361](#)
- combinations [SR358](#)
- combine analyses [US162](#)
- combine datasets *see merge*
- combine functions [SR323](#)
- combine levels [SR360](#)
- combine strings [SR361](#)
- command syntax *see syntax*
- commas [SR324](#)
- comment [SR329](#)
- common intercepts test [SR81](#)
- common problems *see troubleshoot*
- common questions [US237–US250](#)
 - dataset [US237–US240](#)
 - formulas [US240–US243](#)
 - QC analysis [US244–US245](#)
 - survival analysis [US245–US250](#)
- common slopes test [SR81](#)
- communality summary [SR140](#)
- Compact [US57](#), [US88](#), [US90](#)
- compact variables [US57](#), [US81](#), [US84–US97](#),
[US111](#), [US117](#), [US143](#), [US163–US164](#)
 - advantages [US85](#)
 - analyses [US95–US97](#)
 - example [US95](#), [SR91–SR93](#),
[SR97–SR99](#), [SR101](#), [SR105–SR107](#)
 - categories [US91](#)
 - compact [US57](#)
 - create [US86](#), [US245](#)
 - complex example [US89–US93](#)
 - simple example [US87–US89](#)
 - disadvantages [US85–US86](#)
 - expand [US57](#), [US94–US95](#)
 - QC analyses [US244–US245](#)
 - repeated measures analysis of
variance [US85](#), [SR91](#)
 - triangle controls [US88](#), [US95](#)
- compare distributions
 - box plots [SR243](#)
- compare percentile plots [SR247](#)
- data requirements [SR247](#)
- dialog box [SR247](#)
- discussion [SR247](#)
- exercise [SR248](#)
- results [SR248](#)
- templates [SR248](#)
- comparison operators [US126](#)
- complete [SR147](#)
- complex criteria [US125](#), [SR409](#)
- compound vs. multiple
 - results [US135–US137](#), [US170](#), [US229](#)
- Concat [SR361](#)
- conditional transformation [SR341](#)
- confidence intervals [US241–US242](#)
 - bivariate plots [SR221](#), [SR224](#), [SR228](#),
[SR232](#)
 - chi-square test [SR24](#)
 - interaction plots [SR90](#)
 - logistic regression [SR205](#), [SR209](#),
[SR214–SR215](#)
 - mean difference [SR30](#), [SR37–SR38](#), [SR44](#)
 - one sample t-test [SR23](#)
 - proportional hazards models [SR170](#)
 - survival regression analyses [SR188](#)
 - univariate plots [SR217](#)
 - unpaired comparisons [SR37](#)
- connect lines [US188](#)
- constants [SR324–SR325](#)
 - π [SR400](#)
 - e [SR375](#)
- consultant [US162](#)
- contingency coefficient [SR113](#)
- contingency tables [SR111](#)
 - data requirements [SR114–SR116](#)

- dialog box [SRI14](#)
- discussion [SRI11](#)
- exercise [SRI17](#)
- results [SRI16](#)
- templates [SRI16](#)
- continuous data class [US78](#)
- control charts [SR283–SR284](#)
 - lines [SR266–SR267](#)
- control limits
 - 3-sigma rule [SR258](#)
 - QC subgroup measurements
 - analysis [SR262](#)
 - violations [SR251–SR252](#)
- convert data types [SR331](#)
- convert values *see Recode*
- coprocessor [US73](#)
- Copy [US65–US66](#), [US104](#), [US181–US182](#)
 - as text and picture [US233](#), [US236](#)
 - unusual selection shapes [US68–US70](#)
- copy analysis with new variables *see clone*
- copy variable assignments *see adopt*
- corner/center control [US205](#)
- correct errors [SR330](#)
- Correlation [SR362](#)
- correlation [SR31](#), [SR43](#), [SR53](#)
 - data requirements [SR47](#)
 - dialog box [SR46](#)
 - discussion [SR43](#)
 - exercise [US155](#), [SR35](#), [SR48](#)
 - factor analysis [SRI31](#)
 - Kendall rank [SRI21](#)
 - matrix [US53](#)
 - results [SR47](#)
 - Spearman rank [SRI21](#)
 - templates [SR47](#)
- Cos [SR363](#)
- Cosh [SR364](#)
- Cot [SR364](#)
- Count [SR365](#)
- count [US60](#)
- Covariance [SR365](#)
- covariance [SR43](#), [SR45](#)
 - data requirements [SR47](#)
 - dialog box [SR46](#)
 - discussion [SR43](#)
 - exercise [US155](#), [SR48](#)
 - matrix [US53](#)
 - results [SR47](#)
 - templates [SR47](#)
- covariates [SR78](#)
 - proportional hazards models [SRI69](#)
 - survival functions [US245](#), [US248](#)
- Cramer's V [SRI13](#)
- crash [US251](#)
- create
 - analyses [US131–US133](#)
 - by hand or with templates [US131](#)
 - templates [US161–US177](#)
 - tutorial example [US20–US27](#)
 - category
 - tutorial example [US5–US6](#)
 - compact variables [US86](#)
 - criteria [US124–US128](#)
 - graphs [US131–US133](#)
 - tables [US131–US133](#)
 - templates [US169–US174](#)
 - exercise [US171–US177](#)
- Create Analysis [US132](#), [US143](#)
 - exercises [US150–US156](#)
 - tutorial example [US20](#)
- Create Criteria [US124–US128](#), [SR317](#)
- Criteria [US124–US129](#), [SR325–SR326](#),
[SR336–SR339](#), [SR343–SR344](#), [SR409](#)
 - analyses [US149–US150](#)
 - Boolean operators [US127](#)
 - choose level(s) [US127](#)
 - compare results with different [US182](#)
 - complex [US125](#), [SR409](#)
 - delete [US129](#)
 - Edit/Apply [US129](#), [US240](#)
 - example [US130](#)
 - hints [US222](#)
 - names [US125](#)
 - pop-up menu [US28](#), [US124](#), [US128–US129](#)
 - print definitions [US125](#)
 - random [US128](#)
 - set values [US126](#)
 - subtitles [US29](#)
 - troubleshoot [US254–US256](#)
 - turn off [US129](#)
 - tutorial example [US28–US29](#)
 - vs. Include/Exclude Row [US108](#)
 - windows at Open [US254](#), [US256](#)
also see row inclusion

- critical values *see* CDF, inverse
 - cross-hair cursor *see* Recode
 - cross-platform compatibility [US70](#), [US99](#)
 - crosstabs *see* contingency tables
 - Csc [SR366](#)
 - CSCC [SR261](#)
 - cube root [SR334](#)
 - cubic spline [US208](#), [SR221](#), [SR225](#),
[SR227–SR228](#), [SR233](#)
 - CubicSeries [SR367](#)
 - CumProduct [SR367](#)
 - CumSum [SR368](#)
 - CumSumSquares [SR368](#)
 - cumulative distribution function *see* CDF
 - cumulative hazard function [SR165](#)
 - cumulative hazard plot [SR150](#), [SR160](#)
 - cumulative survival plot [SR160](#)
 - currency
 - data type [US73](#), [SR320–SR321](#)
 - formats [USVI](#), [US79](#)
 - cursor movement *see* dataset preferences
 - curve tool *see* spline tool
 - Custom Rulers [US217](#)
 - custom templates [US169–US177](#)
 - exercise [US171–US177](#)
 - custom tests for special causes [SR260](#)
 - dialog box [SR264](#)
 - save as template [SR261](#)
 - customize
 - graphs [US183–US197](#), [US203–US212](#)
 - results [US179–US202](#)
 - shapes [US203–US212](#)
 - tables [US197–US202](#)
 - text [US203–US212](#)
 - CUSUM analysis
 - charts [SR255](#), [SR261](#), [SR271](#), [SR283](#)
 - individual measurement analyses [SR279](#)
 - results [SR263](#), [SR265](#)
 - Cut [US65](#), [US181–US182](#)
 - unusual selection shapes [US68–US70](#)
 - cutpoints [SR342](#)
also see Recode
- D**
- D usage marker *see* usage markers
 - data
 - Copy [US66](#)
 - Cut, Clear, Delete [US65](#)
 - enter [US61](#)
 - manage [US107–US130](#)
 - select [US64](#)
 - subsets *see* Criteria, Include Row, Exclude Row, Sort, row inclusion
also see dataset
 - data class [US6](#), [US50](#), [SR332](#)
 - change [US51](#), [US78](#)
 - continuous [US78](#)
 - discussion [US78](#)
 - example [US49](#)
 - in examples [USVI](#)
 - informative [US78](#), [US117](#)
 - nominal [US78](#), [US238–US240](#)
 - data format
 - currency [US79](#)
 - date/time [US79](#)
 - engineering [US79](#)
 - enhanced free fixed [US79](#)
 - fixed places [US79](#)
 - free format [US79](#)
 - free format fixed [US79](#)
 - in examples [USVI](#)
 - scientific [US79](#)
 - data loss
 - change type [US75](#)
 - when pasting [US67](#)
 - data organization [US3](#), [US49–US53](#)
 - arrangement [US50](#)
 - class [US49–US50](#)
 - compact variables [US84](#)
 - example [US49](#)
 - structure [US49–US53](#)
 - Data pop-up menu
 - Assign variables dialog box [US163](#)
 - variable browser [US56](#), [US143](#), [US146](#)
 - data source [US6](#), [US77](#), [SR329–SR330](#)
 - analysis generated [US77](#)
 - change [US77](#)
 - dynamic formula [US77](#)
 - in examples [USVI](#)
 - static formula [US77](#)
 - user entered [US77](#)
 - data type [US51](#), [US73–US76](#), [SR316](#),
[SR318–SR321](#), [SR332](#), [SR359](#)

- category [US73–US74](#), [SR320](#)
- change [US75–US76](#)
- convert [SR331](#)
- currency [US73](#), [SR320](#)
- date/time [US73](#), [SR321](#)
- import [US103](#), [US254](#)
- in examples [USVI](#)
- integer [US73](#), [SR319](#)
- long integer [US73](#)
- real [US73](#), [SR318](#)
- string [US73](#), [SR319–SR320](#), [SR338](#)
- dataset
 - add columns [US54](#), [US62](#)
 - close [US72](#)
 - common questions [US237–US240](#)
 - copy [US65](#)
 - cut [US65](#)
 - delete [US66](#)
 - edit [US64–US70](#)
 - insert columns [US62](#)
 - paste [US66](#)
 - preferences [US227](#)
 - print [US72](#)
 - renamed [US158](#)
 - save [US70](#)
 - scroll [US64](#)
 - split pane control [US55](#)
 - summary pane *see attribute pane*
 - transfer between Windows and
 - Macintosh [US70](#), [US99](#)
 - troubleshoot [US252](#)
 - window [US4](#)
 - windows [US54–US57](#)
- Dataset Templates [US233–US237](#)
 - custom [US240](#)
- Date [SR369](#)
- date/time
 - data type [US73](#), [SR321](#)
 - fix imported values [SR369](#), [SR427](#)
 - format [US79](#)
 - formats [USVI](#), [US79](#), [SR330–SR331](#)
 - functions [SR330–SR331](#)
 - Date [SR369](#)
 - DateDifference [SR370](#)
 - Day [SR371](#)
 - DayOfWeek [SR372](#)
 - DayOfYear [SR372](#)
 - Hour [SR382](#)
 - Minute [SR389](#)
 - Month [SR391](#)
 - Now [SR393](#)
 - Second [SR419](#)
 - Time [SR427](#)
 - Weekday [SR430](#)
 - WeekOfYear [SR431](#)
 - Year [SR431](#)
 - group by month [SR370](#)
 - missing values [SR321](#)
 - valid data range [US74](#), [SR321](#), [SR330–SR331](#)
- DateDifference [SR370](#)
- Day [SR371](#)
- DayOfWeek [SR372](#)
- DayOfYear [SR372](#)
- decimal characters [USVI](#)
- decimal places [US80](#)
 - graphs [US188](#)
 - in examples [USVI](#)
 - see dataset preferences*
 - tables [US201](#), [US231](#)
- defaults *see preferences*
- defect variable [SR310](#)
- degrees [SR372](#), [SR405](#)
- degrees of freedom [SR73](#)
- DegToRad [SR372](#)
- Delete [US63](#), [US65](#)
 - categories [US84](#)
 - Criteria [US129](#)
 - variables [US116](#)
- delimiters [US101](#), [US252–US253](#)
- denominator df [SR41](#)
- density plot [SR161](#)
- Dependent [US144](#)
- dependent variables [SR76–SR77](#)
- descending Sort *see Sort*
- descriptive statistics [US96](#), [US135–US136](#)
 - data requirements [SR9](#)
 - dialog box [SR7](#)
 - discussion [SR1](#)
 - exercise [SR9](#)
 - results [SR9](#)
 - template exercise [US165–US166](#)
 - templates [US165](#), [SR9](#)
 - tutorial example [US20](#)

- design of StatView [US1](#)
 - determine whether results are
 - selected [US133–US134](#)
 - deviance [SR203](#)
 - deviance residuals [SR170](#), [SR178](#), [SR190](#)
 - df *see degrees of freedom*
 - dichotomous logistic regression [SR199](#)
 - Difference [SR373](#)
 - difference [SR333](#), [SR370](#), [SR373](#)
 - unary [SR335](#)
 - differing results [USVI](#)
 - dimensionality reduction [SR131](#)
 - direction of operation [SR321–SR323](#),
[SR331–SR332](#)
 - directory of results *see results browser*
 - disk space [US170](#), [US232](#)
 - distribute space [US218](#)
 - distributions *see CDF, random numbers functions*
 - Div [SR374](#)
 - divide continuous into groups *see Recode*
 - division [SR334](#), [SR374](#), [SR413](#)
 - document formulas [SR329](#)
 - document size [US213](#)
 - limit [US232](#)
 - documents vs. templates [US161–US162](#)
 - dose response [SR202](#)
 - DotProduct [SR374](#)
 - dotted lines [US185](#)
 - dotted red line *see page breaks*
 - double asterisk [SR334](#)
 - double-byte strings
 - manipulating [SR379](#), [SR384](#), [SR423](#)
 - double-click row numbers *see row inclusion*
 - double-spacing *see line spacing*
 - Draw palette [US185](#), [US195–US197](#), [US199](#),
[US202–US212](#)
 - arc tool [US205–US207](#)
 - arrow tool [US212](#)
 - tutorial example [US45](#)
 - also see selection tool*
 - corner/center control [US205](#)
 - curve tool *see spline tool*
 - ellipse tool [US205](#)
 - fill color [US202](#)
 - tutorial example [US43](#)
 - fill pattern [US211](#)
 - grid [US217](#)
 - line tool [US205](#)
 - tutorial example [US45](#)
 - line widths [US211](#)
 - pen color [US202](#)
 - tutorial example [US43](#)
 - pen pattern [US211](#)
 - polygon tool [US205](#), [US207–US208](#)
 - rectangle tool [US205](#)
 - rounded rectangle tool [US205–US206](#)
 - selection tool [US185](#), [US197](#), [US204](#)
 - spline tool [US208–US210](#)
 - tear-off menu [US204](#)
 - text tool [US204–US205](#)
 - tutorial example [US45](#)
 - drawing and layout [US203–US212](#)
 - Drawing Size [US159](#), [US213](#)
 - dialog box [US213](#)
 - DS Transfer file format [US70](#), [US99](#)
 - Dunnett's [SR87](#)
 - Duplicate [US181–US182](#)
 - Durbin-Watson [SR59](#)
 - dynamic formulas [US116](#), [SR329–SR330](#),
[SR393](#)
 - data source [US77](#)
 - also see Formula*
 - dynamic links
 - analysis objects [US133](#)
 - Analyze menu [US168](#)
 - formulas [US109](#), [US116](#)
 - graph text [US187](#)
 - reopen views [US161](#)
 - results and data [US138–US139](#)
 - tables [US201](#)
- ## E
- e [SR375](#), [SR385](#)
 - also see hyperbolic functions*
 - edit
 - criteria [US240](#)
 - data [US64–US70](#)
 - tutorial example [US15](#)
 - formulas [US240](#)
 - table text [US200](#)
 - Edit Analysis [US134–US135](#), [US140](#), [US142](#),
[US233](#)

- shortcut [US141](#)
 - tutorial example [US24–US25](#)
 - Edit Categories [US83–US84](#)
 - Edit Display [US135](#), [US140–US141](#),
[US180–US182](#), [US185](#), [US187](#), [US195](#),
[US197](#), [US199](#), [US201](#), [US233](#)
 - dialog boxes [US180–US181](#)
 - Edit/Apply Criteria [US124](#), [US129](#), [SR317](#)
 - effects [SR73](#)
 - eigenvalues [SR132](#)
 - table [SR139](#)
 - ElementOf [US28](#), [US126](#), [SR345](#)
 - ellipse tool [US205](#)
 - ellipses [US64](#), [US113](#)
 - empty cell *see missing values*
 - empty graphs [US138](#), [US164](#)
 - empty tables [US138](#), [US151](#), [US164](#)
 - engineering format [US79](#)
 - enhanced free fixed format [US79](#)
 - enter data [US57–US64](#)
 - tutorial example [US3](#), [US6–US10](#)
 - values [US61–US62](#)
 - equal [SR340](#)
 - equamax [SR135](#)
 - Erf [SR376](#)
 - error bars [US241–US242](#)
 - cell plots [SR237](#)
 - interaction plots [SR90](#)
 - error function [SR376](#)
 - error messages [US222](#), [US225](#), [US251](#)
 - beep [US225](#)
 - Formula [US116](#)
 - error of intercept [SR56](#)
 - error-free analyses [US162](#)
 - Euclidean norm [SR392](#)
 - evaluation [SR326–SR328](#)
 - event time variable [US243](#), [SR147](#), [SR149](#)
 - discrete vs. continuous [SR148](#)
 - nonparametric analyses [SR157](#)
 - pattern plot [SR162](#)
 - regression model survival plots [SR175](#)
 - regression models [SR182](#), [SR184](#)
 - survival regression models [SR184](#)
 - Example Views and Datasets [US233](#)
 - examples [USVI](#), [SR316–SR317](#)
 - Excel import/export [US99–US100](#), [US254](#)
 - tutorial example [US11–US12](#)
 - excess risk [SR201](#)
 - Exclude Row [US108–US109](#)
 - analyses [US149–US150](#)
 - compare results [US182](#)
 - subtitles [US29](#)
 - vs. Criteria [US108](#)
also see row inclusion
 - exclusion *see Criteria, Include Row, Exclude Row, row inclusion*
 - exclusive OR [SR347](#)
 - Exercises [SR96](#)
 - Expand [US57](#), [US94](#)
 - expand compact variables [US94–US95](#)
 - expected value [SR112](#)
 - exponential distribution [SR407](#)
 - exponential function [SR375](#), [SR385](#)
 - exponential model [SR172–SR173](#), [SR175](#)
 - exponential regression [SR54](#), [SR68](#)
 - ExponentialSeries [SR376](#)
 - exponentiation [SR334](#)
 - export
 - EMF [US157](#)
 - Excel [US11](#), [US99–US100](#)
 - missing values [US102](#)
 - PICT [US157](#)
 - previous StatView versions [US106](#)
 - SuperANOVA [US106](#)
 - text [US100–US102](#), [US105](#)
 - WMF [US157](#)
 - expression [SR325](#), [SR338–SR339](#)
 - expression language [US113](#)
 - extended precision [US73](#)
 - extra-Binomial variation [SR202](#)
 - extract text [SR423](#)
- ## F
- F* distribution [SR402](#), [SR407](#)
 - factor [SR77](#)
 - factor analysis [SR131](#)
 - basic output [SR138](#)
 - data requirements [SR137](#)
 - dialog box [SR136](#)
 - discussion [SR131](#)
 - exercise [SR138](#)
 - factor extraction methods [SR132](#)
 - factor loadings [SR133](#), [SR139](#)

- factor scores [SR133](#)
- oblique solution [SR142](#)
- results [SR138](#)
- save factor scores [SR133](#)
- summary table [SR138](#)
- templates [SR138](#)
- transformation methods [SR135](#)
- unrotated solution [SR141](#)
- Factorial [SR377](#)
- factorial design [SR91](#)
- factors *see category, nominal data*
- false [SR338](#), [SR345](#)
- features [US1](#)
- FibonacciSeries [SR378](#)
- file formats [US70](#), [US72](#), [US99](#), [US156–US157](#)
- file size [US170](#), [US232](#)
- filename [US99](#)
- fill patterns
 - color [US197](#)
 - colors [US202](#), [US212](#)
 - graph interior [US185](#)
 - graphs [US196](#), [US229](#)
 - shapes [US211](#)
- final communality estimate [SR140](#)
- find and replace [US241](#)
- Fisher's exact test [SR113](#)
- Fisher's PLSD [SR86](#), [SR103](#)
 - tutorial example [US39](#)
- Fisher's r to z transformation [SR32](#), [SR44](#)
- fitted lines *see bivariate plots*
- fitted values [SR60–SR61](#)
- fix page breaks
 - tutorial example [US43](#)
- fix page breaks *see Clean Up Items*
- fixed places [US79](#)
- flip *see transpose*
- Floor [SR380](#)
- fonts
 - graphs [US185](#)
 - tables [US199–US200](#)
 - views [US205](#), [US232](#)
- Food Guide Pyramid [US2](#)
- Force button [US144](#)
 - stepwise regression [SR62](#)
 - survival regression models [SR185](#)
- force recalculation [US138–US139](#)
- foreign versions [US256](#)
- format [SR16](#)
 - date/time data [SR330–SR331](#)
 - graphs [US183–US197](#)
 - multiple columns
 - tutorial example [US9](#)
 - numeric data [SR332](#)
 - tables [US197–US202](#), [US231](#)
 - templates [US170](#), [US233](#)
 - also see data format*
- Formula [US109–US116](#), [SR318](#), [SR330](#), [SR338–SR339](#)
 - analysis generated variables [SR61](#)
 - build definition [US113](#)
 - common questions [US240–US243](#)
 - compute [US113](#)
 - date/time data [US74](#)
 - dialog box [US109](#)
 - dynamic links [US109](#), [US116](#)
 - edit [US113](#), [US240](#)
 - errors [US116](#)
 - examples [US114–US115](#)
 - hints [US222](#)
 - import from SuperANOVA [US256](#)
 - missing values [US255](#)
 - preferences [US228](#)
 - print definitions [US110](#)
 - shortcuts [US115](#)
 - troubleshoot [US116](#), [US254–US256](#)
 - tutorial example [US17–US20](#)
 - variable attributes [US110](#)
 - windows at Open [US254](#), [US256](#)
- fractional values [SR359](#), [SR374](#), [SR380](#), [SR390](#), [SR413](#), [SR416](#), [SR429](#)
- F-ratio [SR74](#), [SR84](#)
- free format [US79](#)
- free format fixed [US79](#)
- free-form curves *see spline tool*
- frequency distribution [US233](#), [SR13](#)
 - data requirements [SR16](#)
 - dialog box [SR14](#)
 - discussion [SR13](#)
 - exercise [SR17](#)
 - interval settings [SR14](#)
 - results [SR16](#)
 - templates [SR16](#)
 - tutorial example [US31](#)
- Frequency Summary Table [US136–US137](#)

Friedman test [SRI22](#)
F-statistic [SR73](#)
F-test *see unpaired comparisons*
function browser [US111](#), [US113](#)

G

G usage marker *see usage markers*
Games-Howell [SR88](#)
gamma distribution [SR408](#)
Gaussian distribution [SR408](#)
generate data [SR338](#), [SR347](#)
 also see Series, Random Number, Formula
generator seed *see random numbers*
generic variable names for templates [US170](#)
geometric mean [SR3](#)
GeometricMean [SR380](#)
GeometricSeries [SR381](#)
global null hypothesis tests table [SRI86](#)
golden ratio [SR378](#)
graph defaults *see preferences*
Graph dialog box [US187](#)
graphs
 align [US214](#)
 arrange [US213](#)–[US214](#)
 axes [US183](#)
 axis bounds [US185](#)
 axis frame [US185](#)
 axis frames [US185](#), [US187](#), [US229](#)
 axis labels [US183](#)
 axis values [US183](#)
 bivariate plots [SR221](#)–[SR236](#)
 box plots [SR243](#)
 cell plots [SR237](#)
 chart of types [SRV](#)
 colors [US229](#)
 compare percentile plots [SR247](#)
 create [US131](#)–[US133](#)
 by hand or with templates [US131](#)
 customize [US183](#)–[US197](#)
 decimal places [US192](#)
 Edit commands [US181](#)
 edit text [US187](#)
 fill color [US197](#)
 fill patterns [US196](#), [US229](#)
 fonts [US185](#)
 format [US135](#)

frames [US183](#), [US197](#)
grid lines [US183](#), [US185](#), [US192](#)
group [US215](#)
height [US188](#), [US229](#)
interior [US183](#), [US185](#), [US195](#)
layers [US215](#)
legends [US183](#), [US186](#), [US188](#),
 [US194](#)–[US195](#), [US197](#)
line patterns [US185](#)
line widths [US185](#), [US196](#)
list in analysis browser [US141](#)
lock [US214](#)
move [US186](#), [US214](#)
move components [US186](#)
notes [US184](#), [US186](#)
numeric formats [US229](#)
overlay [US186](#)
pen color [US197](#)
pen patterns [US185](#), [US196](#)
percentile plots [SRI9](#)
plots [US183](#), [US197](#)
plotted lines [US183](#), [US197](#)
point colors [US195](#)–[US196](#)
point types [US195](#)
point types and sizes [US229](#)
preferences [US179](#)–[US180](#), [US228](#)–[US229](#)
reference lines [US184](#)
resize [US186](#)
select [US184](#)
select components [US185](#)
template exercise [US171](#)
text color [US185](#), [US197](#)
tick marks [US183](#), [US185](#)
titles [US183](#), [US186](#), [US188](#)
ungroup [US215](#)
univariate plots [SR217](#)
unlock [US214](#)
width [US188](#), [US229](#)
X axis [US183](#)
Y axis [US183](#)
Graphs Only [US142](#)
greater than [SR341](#)
greater than or equal to [SR337](#), [SR341](#)
gremlins [US252](#)
grid [US186](#), [US200](#), [US217](#)
 colors [US217](#)
 spacing [US217](#)

grid lines [US183](#), [US185](#), [US192](#)
 Group [US215](#)
 unexpected results [US251](#)
 group labels *see category*
 group variable
 nonparametric analyses [SRI57](#)
 grouped regression [SR81](#)
 Groups [SR381](#)
 groups [US50–US51](#), [US80](#), [US190](#), [SR327](#),
 [SR336](#), [SR349](#)
 choose *see Criteria*
 nonparametric analyses [SRI47](#)
 also see compact variables, Split By
 growth regression [SR55](#), [SR67](#)
 G-statistic [SRI13](#)

H

hairlines [US232](#)
 half-open interval [SR337](#)
 harmonic mean [SR3](#)
 HarmonicMean [SR382](#)
 Harrington-Fleming test [SRI51](#), [SRI56](#)
 Harris image analysis [SRI32](#)
 hashed red lines *see page breaks*
 hatch marks *see tick marks*
 hazard function [SRI78](#)
 hazard plot [SRI50](#), [SRI61](#)
 height [US229](#)
 graphs [US188](#)
 tables [US199–US200](#)
 help [US48](#), [US221](#), [US223–US225](#)
 helpful hints *see troubleshoot*
 heterogeneous variances [SR86](#)
 Hide Grid Lines [US217](#)
 Hide Page Breaks [US217](#)
 Hide Rulers [US217](#)
 hierarchical Analyze menu [US169](#)
 Hints [US48](#), [US221–US222](#)
 balloon [US222](#)
 formulas [US113](#)
 informational [US222](#)
 interface [US222](#)
 preferences [US222](#), [US230](#)
 templates (Assign Variables) [US163](#)
 window [US4](#)
 Histogram [US132](#), [US136–US137](#), [US233](#)

capability indices [US244](#)
 normal curve [US244](#)
 tutorial example [US31](#)
 histograms *see frequency distribution*
 historical values [SR256](#)
 homogeneity of slopes [SR81](#)
 homogeneity of variances [US235](#), [SR84](#)
 horizontal *see casewise*
 Hotelling-Lawley trace [SR82](#)
 Hour [SR382](#)
 HYP [US112](#)
 hyperbolic arccosine [SR349](#)
 hyperbolic arcsine [SR354](#)
 hyperbolic arctangent [SR356](#)
 hyperbolic cosine [SR364](#)
 hyperbolic sine [SR419](#)
 hyperbolic tangent [SR426](#)
 hyperbolic trig functions [US112](#)
 hypothesis testing [SR74–SR76](#)
 hypothesized mean [SR23](#)
 hypothesized variance [SR24](#)

I

I charts [SR277](#), [SR281](#), [SR283](#)
 I class marker *see class marker*
 if...then...else [US114](#), [SR341](#)
 tutorial example [US19](#)
 ignore characters [SR329](#)
 illustrations in manual [USVI](#)
 impact [SR201](#)
 import
 categories [US104](#), [US254](#)
 data type [US103](#), [US254](#)
 date/time values [SR369–SR370](#),
 [SR427–SR428](#)
 dialog box [US101](#)
 examples [US104–US105](#)
 Excel [US254](#)
 tutorial example [US11–US12](#)
 missing values [US103](#), [US253–US254](#)
 non-numeric data as type String [US254](#)
 pictures [US210](#)
 previous StatView versions [US106](#)
 separator characters [US252–US253](#)
 SuperANOVA [US106](#), [US256](#)
 text [US100–US102](#)

tutorial example [US12–US13](#)
 troubleshoot [US252–US254](#)
 variable names [US102](#)
 in control [SR251, SR253](#)
 Include Row [US108–US109](#)
 analyses [US149–US150](#)
 compare results [US182](#)
 subtitles [US29](#)
 vs. Criteria [US108](#)
 inclusion *see* *Criteria, Include Row, Exclude Row, row inclusion*
 incomplete [SR147](#)
 incorrect results [US250](#)
 Independent [US144](#)
 independent variables [SR77–SR78](#)
 index of results *see* *results browser*
 individual measurements analysis *see* *QC individual measurements analysis*
 inequality [SR338–SR341](#)
 informational hints [US222](#)
 informative data class [US78, US117, SR332](#)
 input column [US4, US54, US61](#)
 input row [US61](#)
 insert columns [US62](#)
 insert rows [US62–US63](#)
 integer data type [US73, SR319](#)
 interaction effects [SR78, SR90](#)
 interaction plots [SR78, SR90, SR96–SR99, SR104, SR107](#)
 tutorial example [US40](#)
 also see *cell plots*
 intercept [SR55–SR56, SR80–SR81, SR200](#)
 Interface hints [US222](#)
 interior
 graphs [US183](#)
 tables [US198](#)
 international datasets [US256, SR324](#)
 international system configurations [USVI](#)
 interquartile range [SR5](#)
 intervals [SR336–SR338, SR345](#)
 frequency distribution [SR14](#)
 INV [US112](#)
 inverse CDF *see* *CDF*
 inverse functions *see* *arc functions*
 inverting matrices [SR433](#)
 invisible lines [US202](#)
 IQR *see* *interquartile range*

IS [US255, SR346](#)
 IsMissing [US255, SR343](#)
 ISNOT [US255, SR347](#)
 IsRowExcluded [SR343](#)
 IsRowIncluded [SR344](#)
 item count variable [SR290](#)
 iterated principal axis factor
 extraction [SR132](#)
 iteration history table [SR189](#)

J

Japanese characters [SR379, SR384, SR423](#)
 joint significance tests table [SR190](#)
 jump point [SR149](#)

K

Kaiser image analysis [SR132](#)
 Kaplan-Meier [SR149, SR163](#)
 Kendall rank correlation [SR121](#)
 data [SR123](#)
 data requirements [SR123](#)
 exercise [SR127](#)
 Kendall's tau [SR121](#)
 key *see* *legends*
 keyboard shortcuts [USVI, US64](#)
 draw shapes [US205](#)
 Edit Analysis, Edit Display [US233](#)
 move graphs [US186](#)
 move tables [US200](#)
 see *StatView Shortcuts card*
 Kolmogorov-Smirnov test [US86, SR120](#)
 data requirements [SR123](#)
 template [US234](#)
 Kruskal-Wallis test [SR121](#)
 data requirements [SR123](#)
 exercise [SR128](#)
 kurtosis [US234, SR6](#)

L

Lag [SR382](#)
 lag [SR363](#)
 lambda [SR74, SR76](#)
 landscape page [US213](#)
 Latin square design [SR105–SR107](#)

- layers [US215](#)
 - exercise [US215](#)
- Layout tools [US212–US219](#)
 - tutorial example [US43](#)
- least significant difference (Fisher's protected) [SR86](#)
- Left Justify [US205](#), [US218](#)
- left to right evaluation [SR328](#)
- legends [US182–US183](#), [US194–US195](#)
 - color [US197](#)
 - frame [US194](#)
 - layout [US195](#)
 - move [US186](#)
 - orientation [US186](#)
 - show [US188](#)
 - symbols [US186](#), [US194](#)
 - text [US195](#)
- Len [SR383](#)
- length of string [SR383](#)
- length of vector [SR392](#)
- leptokurtic [SR7](#)
- less than [SR337](#), [SR340](#)
- less than or equal to [SR337](#), [SR340](#)
- levels *see categories*
- Library [US91](#), [US225](#), [US233](#), [US251](#)
- life table method [SR152](#)
- likelihood ratio test [SR170](#), [SR179](#), [SR209](#)
- Likert scale, reverse [SR348](#)
- limit document size [US232](#)
- line charts
 - cell plots [SR237](#)
 - connect lines [US188](#)
 - Line Plot dialog box [SR217](#)
 - univariate plots [SR217](#)*also see bivariate plots and univariate plots*
- line patterns
 - graphs [US185](#)
 - tables [US202](#)
- line spacing [US199](#)
 - tables [US201](#), [US231](#)
- line tool [US205](#)
 - tutorial example [US45](#)
- line widths [US232](#)
 - graphs [US185](#), [US196](#)
 - shapes [US211](#)
 - tables [US199](#), [US202](#)
- linear algebra
 - DotProduct [SR374](#)
 - Norm [SR392](#)
- linear axis scale [US192](#)
- linear predictor [SR170](#), [SR178](#)
 - standard error [SR178](#)
- LinearSeries [SR384](#)
- listwise deletion [SR45](#)
- Ln [SR385](#)
- In cumulative hazard function [SR150](#), [SR161](#), [SR166](#)
- localized versions [US256](#)
- locally weighted scatterplot smoother *see lowess*
- locate results *see results browser*
- Lock [US214](#)
- Log [SR385](#)
- log odds [SR200](#)
- logarithmic axis scale [US192](#)
- logarithmic regression [SR54](#)
- logarithms [SR375](#), [SR385–SR386](#)
also see hyperbolic functions
- LogB [SR386](#)
- logical expressions *see Criteria, Formula*
- logical operators [SR328](#), [SR338–SR340](#)
 - AND [SR345](#)
 - ElementOf [SR345](#)
 - equal [SR340](#)
 - exclusive OR [SR347](#)
 - false [SR345](#)
 - greater than [SR341](#)
 - greater than or equal to [SR341](#)
 - if...then...else [SR341](#)
 - is [SR346](#)
 - IsMissing [SR343](#)
 - ISNOT [SR347](#)
 - IsRowExcluded [SR343](#)
 - IsRowIncluded [SR344](#)
 - less than [SR340](#)
 - less than or equal to [SR340](#)
 - NOT [SR345](#)
 - not equal [SR341](#)
 - OR [SR347](#)
 - true [SR345](#)
- logistic regression [SR199–SR215](#)
 - assumptions [SR202](#)
 - binary [SR199](#)
 - case-control studies [SR203](#)

confidence intervals [SR205](#), [SR209](#),
 [SR214–SR215](#)
 data requirements [SR205](#)
 dialog box [SR204](#)
 dichotomous [SR199](#)
 discussion [SR199–SR204](#)
 estimating coefficients [SR203](#)
 exercises [SR207](#), [SR215](#)
 iterations [SR205](#)
 multiple [SR201–SR202](#), [SR212](#)
 nominal data coding [SR206](#)
 polytomous [SR199](#), [SR204](#), [SR214](#)
 random samples [SR203](#)
 reference level [SR206](#)
 results [SR207](#)
 simple [SR200–SR201](#), [SR207](#)
 templates [SR207](#)
 loglogistic model [SR172](#)
 lognormal model [SR172](#), [SR175](#)
 LogOdds [SR386](#)
 logrank (Mantel-Cox) test [SR151](#), [SR156](#)
 long integer data type [US73](#)
 lowess [SR221](#), [SR225–SR226](#), [SR235](#)
 tension [SR226](#), [SR228](#)
 LSD *see Fisher's PLSD*

M

M usage marker [US163](#)
 MacDraw II
 size limit [US232](#)
 macros *see templates*
 MAD *see median absolute deviation*
 magnitude [SR348](#)
 main effects [SR78](#)
 manage data [US107–US130](#)
 tutorial example [US2–US13](#)
 Manage menu
 commands [US107](#), [SR315](#), [SR317–SR318](#)
 Preferences [US225](#)
 manage templates [US167–US169](#)
 MANCOVA *see analysis of variance*
 manipulate columns and rows [US62–US64](#)
 Mann-Whitney U test [SR120](#)
 data [SR123](#)
 exercise [SR125](#)
 MANOVA *see analysis of variance*

Mantel-Cox test [SR151](#), [SR156](#)
 Mantel-Haenszel test [SR151](#), [SR156](#)
 marquee select [US184](#), [US198](#)
 martingale residuals [SR170](#), [SR178](#), [SR190](#)
 mathematical expression language [US113](#)
 mathematical functions
 absolute value [SR348](#)
 addition [SR333](#)
 Average [SR356](#)
 AverageIgnoreMissing [SR357](#)
 Ceil [SR359](#)
 Combinations [SR361](#)
 CumProduct [SR367](#)
 CumSum [SR368](#)
 CumSumSquares [SR368](#)
 difference [SR373](#)
 Div [SR374](#)
 division [SR334](#)
 DotProduct [SR374](#)
 e [SR375](#)
 Erf [SR376](#)
 exponentiation [SR334](#)
 Factorial [SR377](#)
 Floor [SR380](#)
 Lag [SR382](#)
 Ln [SR385](#)
 Log [SR385](#)
 log [SR386](#)
 LogOdds [SR386](#)
 Mod [SR390](#)
 MovingAverage [SR391](#)
 multiplication [SR333](#)
 negative [SR335](#)
 Norm [SR392](#)
 parentheses [SR336](#)
 Percentages [SR397](#)
 Percentile [SR398](#)
 Permutations [SR399](#)
 Pi [SR400](#)
 positive [SR335](#)
 Remainder [SR413](#)
 Round [SR416](#)
 Sqrt [SR420](#)
 subtraction [SR333](#)
 Sum [SR423](#)
 SumIgnoreMissing [SR424](#)
 SumOfColumn [SR424](#)

- SumOfSquares [SR425](#)
 - Trunc [SR429](#)
 - matrix inversion [SR433](#)
 - Maximum [SR387](#)
 - maximum [US60](#), [US255](#), [SR3](#)
 - Mean [SR388](#)
 - mean [US60](#), [US184](#), [SR1](#), [SR356–SR357](#), [SR388](#), [SR391](#), [SR428](#)
 - confidence interval around [SR217](#)
 - one sample *t*-test [SR23](#)
 - mean difference confidence interval [SR30](#)
 - mean square [SR73](#)
 - means tables [SR78](#), [SR90](#)
 - tutorial example [US39](#)
 - measure string values [SR383](#)
 - measurement units [US217](#)
 - Median [SR388](#)
 - median [SR2](#), [SR398](#)
 - median absolute deviation [SR6](#), [SR387](#)
 - memory requirements [US232](#), [US251](#)
 - merge
 - datasets [US66](#)
 - files [US99](#)
 - graphs [US186](#)
 - mesokurtic [SR7](#)
 - message area [US140](#)
 - method default [SR135](#)
 - Microsoft Excel *see Excel*
 - Minimum [SR389](#)
 - minimum [US60](#), [US255](#), [SR3](#)
 - minus [SR333](#), [SR335](#), [SR370](#), [SR373](#)
 - Minute [SR389](#)
 - Missing Cells [US255](#)
 - also see missing values*
 - missing values [US60–US62](#), [US118](#), [SR326](#), [SR335](#), [SR338–SR339](#), [SR343](#), [SR346–SR347](#), [SR357](#), [SR365](#), [SR393–SR394](#), [SR424](#)
 - date/time data [SR321](#), [SR330](#)
 - formulas [US255](#)
 - import [US103](#), [US253–US254](#)
 - in criteria [US127](#)
 - multiple datasets [US146](#)
 - Recode [US120](#), [US255](#)
 - recode [SR341](#)
 - Mod [SR390](#)
 - Mode [SR390](#)
 - mode [SR2](#)
 - model building [SR76](#), [SR78](#)
 - model coefficients table [SR186](#)
 - modify templates [US171](#)
 - modulus [SR390](#)
 - Month [SR391](#)
 - Most [SR324](#)
 - mouse shortcuts [USV1](#)
 - also see StatView Shortcuts card*
 - move
 - graphs [US186](#)
 - objects [US214](#)
 - table components [US200](#)
 - tables [US200](#)
 - Move Backward [US215–US216](#)
 - Move Forward [US215–US216](#)
 - Move to Back [US186](#), [US215](#)
 - Move to Front [US186](#), [US215](#)
 - moving range [SR477](#)
 - MovingAverage [SR391](#)
 - MR charts [SR278](#), [SR281](#), [SR283](#)
 - multiple [SR52](#)
 - multiple categories [US254](#)
 - multiple comparisons *see post hoc tests*
 - multiple logistic regression [SR201–SR202](#), [SR212](#)
 - multiple regression [SR52](#), [SR69](#), [SR77](#)
 - multiple vs. compound
 - results [US135–US137](#), [US170](#), [US229](#)
 - multiplication [SR333](#), [SR367](#), [SR374](#)
- ## N
- N class marker *see class marker*
 - name variables *see variables*
 - natural logarithm [SR385](#)
 - negation [SR345](#)
 - negative [SR335](#)
 - also see absolute value*
 - nest functions [SR323](#)
 - nested groups [SR381](#)
 - new dataset
 - tutorial example [US4](#)
 - New View [US132](#), [US139](#), [US150](#)
 - nominal data [SR204](#)
 - bivariate plots [SR232](#)
 - coding [SR206](#)

nominal data class [US74](#), [US78](#),
[US238](#)–[US240](#), [SR332](#)
also see category, Split By
nonconformity variable
 c/u analyses [SR302](#)
 p/np analyses [SR290](#)
nonlinear regression [SR54](#)–[SR55](#), [SR67](#)–[SR68](#)
 also see logistic regression
nonparametric tests [SRI19](#)
 data [SRI23](#)
 data requirements [SRI23](#)
 dialog box [SRI22](#)
 discussion [SRI19](#)
 exercises [SRI25](#)
 Friedman test [SRI22](#)
 Kendall's tau [SRI21](#)
 Kolmogorov-Smirnov test [SRI20](#)
 Kruskal-Wallis test [SRI21](#)
 Mann-Whitney U test [SRI20](#)
 one sample sign test [SRI19](#)
 paired sign test [SRI21](#)
 results [SRI24](#)
 Spearman rank correlation
 coefficient [SRI21](#)
 templates [SRI24](#)
 ties [SRI24](#)
 Wald-Wolfowitz runs test [SRI20](#)
 Wilcoxon signed rank test [SRI20](#)
Norm [SR392](#)
normal count [SRI5](#)
normal curve [US244](#)
normal distribution [SR403](#), [SR410](#)
 curve on histogram [SRI5](#)
 definition [SR4](#)
normality [SRI3](#), [SR416](#)
Normality Test [US86](#), [US233](#)
NOT [SR345](#)
not enough memory [US232](#), [US251](#)
not equal [SR341](#)
notation
 keyboard/mouse shortcuts [USVI](#)
notation *see syntax*
notched box plots [US188](#)
notes [US184](#)
 move [US186](#)
 tables [US199](#)
Now [SR393](#)

np charts [SR266](#), [SR288](#), [SR293](#)
*n*th root [SR334](#)
null hypothesis [SR74](#)–[SR75](#)
number of cases [SR365](#)
number of seconds *see dateltime functions*
NumberMissing [SR393](#)
NumberOfRows [SR394](#)
numerator df [SR41](#)
numeric axes [US190](#)
 bounds [US191](#)
 dialog boxes [US190](#)
 tick marks [US191](#)
numeric data [US66](#)
numeric formats
 axes [US192](#)
 graphs [US229](#)
 tables [US201](#), [US231](#)
numeric intervals [SR377](#)
numeric precision [US73](#)
nutrition labels [US2](#)

0

object-oriented technology [US133](#), [US161](#)
objects
 align [US214](#)
 clean up [US213](#)–[US214](#)
 group [US215](#)
 layers [US215](#)
 lock [US214](#)
 move [US214](#)
 ungroup [US215](#)
 unlock [US214](#)
oblique factor scores [SRI35](#), [SRI37](#), [SRI40](#),
 [SRI42](#)
odds [SR200](#)
off-diagonal [SRI33](#)
old StatView data [US106](#)
omnibus tests [SR74](#)
one [SR345](#)
one sample analysis [SR23](#)
 data requirements [SR25](#)
 dialog box [SR24](#)
 discussion [SR23](#)–[SR24](#)
 exercise [SR26](#)
 nonparametric [SRI19](#)
 results [SR26](#)

templates [SR26](#)
 t -test [SR23](#)
 one-case-per-row data [US51–US52](#)
 OneGroupChiSquare [SR394](#)
 OnlyExcludedRows [US108](#), [US124](#),
 [SR325–SR326](#)
 OnlyIncludedRows [US108](#), [US124](#),
 [SR325–SR326](#)
 Open [US72](#), [US99–US100](#), [US157](#)
 dataset
 tutorial example [US13](#)
 view
 use different variables [US158](#)
 views [US157–US159](#)
 use different dataset [US159](#)
 use original variables [US157](#)
 open interval [SR337](#)
 open polygon [US207](#)
 open spline [US208](#)
 Open View As [US157–US158](#), [US161](#)
 operators [SR332–SR336](#)
 addition [SR333](#)
 division [SR334](#)
 exponentiation [SR334](#)
 multiplication [SR333](#)
 negative [SR335](#)
 parentheses [SR336](#)
 positive [SR335](#)
 subtraction [SR333](#)
 OR [SR347](#)
 order of operations [SR326–SR328](#), [SR336](#)
 Order pop-up menu
 analysis browser [US142](#)
 Assign variables dialog box [US163](#)
 function browser [US111](#)
 results browser [US147](#)
 variable browser [US56](#), [US110](#), [US143](#)
 ordinal axes [US190](#), [US193](#)
 ordinate *see Y axis*
 orientation
 page [US213](#)
 also see transpose
 orthogonal factor solution [SR135](#), [SR137](#)
 out of control [SR253](#)
 out of memory [US232](#), [US251](#)
 outliers [SR57](#), [SR391](#)
 and variance [SR4](#)

 box plots [SR243](#)
 in descriptive statistics [SR1](#)
 output list in analysis browser [US141](#)
 ovals [US206](#)
 overlay
 graphs [US186](#)
 graphs, tables [US215](#)

P

p value
 analysis of variance [SR74](#)
 contingency tables [SR112](#)
 correlation [SR44](#)
 one sample t -test [SR23](#)
 paired comparisons [SR29](#)
 regression [SR57](#)
 regression intercept [SR55](#)
 unpaired comparisons [SR37](#)
 z test [SR31](#)
p/np analyses see QC p/np analysis
 page breaks [US27](#), [US159](#)
 color [US217](#)
 print [US214](#)
 show/hide [US217](#)
 page orientation [US213](#)
 paired comparisons [SR29](#)
 data [SR33](#)
 data requirements [SR33](#)
 dialog box [SR32](#)
 discussion [SR29–SR32](#)
 exercise [SR34–SR36](#)
 nonparametric [SR120–SR121](#)
 paired t -test [SR29](#)
 results [SR33](#)
 templates [SR34](#)
 z -test [SR31](#)
 z-test [SR31](#)
 pairwise deletion [SR45](#)
 parametric models [SR171–SR172](#)
 parentheses [SR327](#), [SR336](#), [SR349](#)
 intervals [SR337](#)
 Pareto analysis [US238](#), [SR256](#), [SR311](#)
 data requirements [SR310–SR311](#)
 dialog box [SR309–SR310](#)
 discussion [SR309](#)
 exercise [SR311–SR313](#)

- results [SR311](#)
- partial correlation [SR45](#), [SR205](#)
- partial *F*-ratio [SR53](#)
- Pascal's triangle [SR358](#)
- Paste [US66](#), [US104](#), [US181–US182](#)
 - imported pictures and text [US210](#)
 - results into datasets [US233](#)
 - unusual selection shapes [US68–US70](#)
- Paste Transposed [US68](#)
- patterns *see pen patterns*
- Pearson correlation [SR131](#), [SR362](#)
- pen color [US202](#)
- pen patterns
 - colors [US197](#), [US212](#)
 - graphs [US185](#), [US196](#)
 - shapes [US211](#)
 - tables [US199](#), [US202](#)
- Percentages [SR397](#)
- Percentile [SR398](#)
- percentile plots [SR19](#)
- percentiles
 - convert raw scores to [SR399](#)
 - data requirements [SR19](#)
 - dialog box [SR19](#)
 - exercise [SR20](#)
 - find several at once [SR398](#)
 - results [SR20](#)
 - templates [SR20](#)
- percents of column totals table [SR112](#)
- percents of row totals table [SR112](#)
- period *see missing values*
- Permutations [SR399](#)
- permutations
 - ordered [SR377](#), [SR400](#)
 - unordered [SR361](#)
- Peto-Peto-Wilcoxon test [SR151](#), [SR156](#)
- phi coefficient [SR113](#)
- Pi [SR400](#)
- Pillai's trace [SR82](#)
- placeholders [US138](#), [US151](#), [US164](#), [SR324](#)
 - formulas [US115](#)
- plateau [SR79](#)
- platykurtic [SR7](#)
- plots [US183](#)
 - colors [US197](#)
- plotted lines [US183](#)
 - color [US197](#)
- PLSD *see Fisher's* PLSD
- plus [SR333](#), [SR335](#), [SR368](#), [SR423–SR424](#)
- point charts *see scattergrams*
- point colors [US195–US196](#)
- point sizes [US195](#), [US229](#)
- point types [US195](#), [US229](#)
- Poisson distribution [SR410](#)
- polygon tool [US205](#), [US207–US208](#)
- polynomial regression [SR52](#), [SR64–SR65](#)
- polytomous logistic regression [SR199](#),
[SR204](#), [SR214](#)
- population statistics [SR1](#)
- portrait page [US213](#)
- positive [SR335](#)
- post hoc tests [SR74](#), [SR84–SR90](#), [SR103](#)
 - assumptions [SR84](#)
 - Bonferroni/Dunn [SR86](#)
 - cell contributions table [SR112](#)
 - Dunnett's [SR87](#)
 - Fisher's PLSD [SR86](#)
 - Games-Howell [SR88](#)
 - interaction effects [SR89](#)
 - purposes [SR84](#)
 - repeated measures [SR88](#)
 - Scheffé's *F* [SR86](#), [SR104](#)
 - Student Newman Keuls [SR88](#)
 - Tukey-Kramer [SR87](#)
 - type I errors [SR85](#)
- pound signs [US64](#)
- power [SR74](#), [SR76](#), [SR90](#), [SR334](#)
- power regression [SR55](#)
- precision [US73](#)
- predicted values [SR60–SR61](#), [SR66](#)
- preferences [US225–US233](#)
 - application [US225–US226](#)
 - color palette [US226–US227](#)
 - dataset [US227](#)
 - formula [US228](#)
 - graph [US228–US229](#)
 - graphs [US179–US180](#)
 - hints [US222](#), [US230](#)
 - Survival Analysis [US230–US231](#)
 - table [US179–US180](#), [US231](#), [US250](#)
 - view [US141](#), [US171](#), [US179–US180](#),
[US232–US233](#)
- presentation [US161](#), [US169](#)
 - tutorial example [US43–US47](#)

prevent changes *see* *Lock*
 prevent errors [US162](#)
 prevent recalculation [US138–US139](#), [US170](#)
 preview format changes [US181](#)
 previous versions' data [US106](#)
 primary pattern solution [SR140](#)
 principal components analysis [SR132](#)
 principal values *see* *arc functions*
 print
 Criteria definitions [US125](#)
 dataset [US72](#)
 Formula definitions [US110](#)
 line widths [US159](#), [US232](#)
 presentation
 tutorial example [US47](#)
 Random Numbers definitions [US123](#)
 Recode definitions [US119–US120](#)
 Series definitions [US122](#)
 troubleshoot [US254](#)
 views [US159–US160](#)
 prior probability [SR211](#)
 probabilities functions
 ProbBinomial [SR401](#)
 ProbChiSquare [SR402](#)
 ProbF [SR402](#)
 ProbNormal [SR403](#)
 Probt [SR404](#)
 ReturnChiSquare [SR414](#)
 ReturnF [SR415](#)
 ReturnNormal [SR415](#)
 ReturnT [SR416](#)
 probability value *see* *p value*
 ProbBinomial [SR401](#)
 ProbChiSquare [SR402](#)
 ProbF [SR402](#)
 problems *see* *troubleshoot*
 ProbNormal [SR403](#)
 Probt [SR404](#)
 process [SR251](#)
 process capability analysis [SR261](#)
 product [SR333](#), [SR367](#), [SR374](#)
 product limit method (Kaplan-Meier) [SR149](#)
 product-limit method (Kaplan-Meier) [SR152](#)
 progress bar [US11](#)
 proportional hazards models [SR167](#), [SR175](#),

[SR191](#)
 baseline hazard [SR169](#)
 coefficients [SR169](#)
 confidence intervals [SR170](#)
 covariate values [SR169](#)
 residuals plots [SR170](#)
 significance tests [SR169](#)
 stratification [SR171](#)
 stratified [SR169](#)
 protected least significant difference *see*
 Fisher's PLSD

Q

QC analysis [US142](#)
 common questions [US244–US245](#)
 example [SR254](#)
 introduction [SR251–SR256](#)
 QC *c/u* analysis
 c/u charts [SR300](#), [SR303](#), [SR306](#)
 control limits [SR299](#)
 data requirements [SR299](#), [SR302–SR303](#)
 dialog boxes [SR301–SR302](#)
 discussion [SR299–SR301](#)
 exercise [SR305–SR307](#)
 nonconformity variable [SR302](#)
 results [SR303–SR305](#)
 standardize inspection criteria [SR301](#)
 subgroups [SR299](#)
 templates [SR305](#)
 QC individual measurements analysis
 capability analysis [SR279](#)
 CUSUM [SR279](#)
 data requirements [SR280](#)
 dialog boxes [SR279–SR280](#)
 discussion [SR277–SR279](#)
 exercise [SR283–SR284](#)
 results [SR280–SR282](#)
 templates [SR283](#)
 tests for special causes [SR278](#)
 QC *p/np* analysis [SR255](#)
 control limits [SR288](#)
 data requirements [SR287](#), [SR290–SR292](#)
 dialog boxes [SR290](#)
 discussion [SR287–SR289](#)
 exercises [SR294–SR297](#)
 p charts [SR256](#), [SR266](#), [SR288](#), [SR292](#),

[SR295](#)
 results [SR292–SR294](#)
 standardize inspection criteria [SR289](#)
 subgroup variables [SR287](#)
 templates [SR294](#)
 QC subgroup measurements analysis
 data requirements [SR268–SR269](#)
 dialog boxes [SR262–SR268](#)
 discussion [SR257–SR262](#)
 exercises [SR273–SR276](#)
 results [SR269–SR272](#)
 templates [SR273](#)
 tests for special causes [SR259](#)
 QuadraticSeries [SR404](#)
 quantile plots [SR174](#)
 QuarticSeries [SR405](#)
 quartiles [SR398](#)
 quartimax [SR135](#)
 question marks [US113](#), [SR324](#)
 questions *see common questions*
 quotation marks [US255](#), [SR325](#), [SR331](#)
 quotient [SR334](#)

R

R (partial correlation coefficient) [SR205](#)
 R charts [SR259](#), [SR266](#), [SR270](#)
 R squared [SR56](#)
 radians [SR372](#), [SR400](#), [SR405](#)
 radius for round corners [US206](#)
 RadToDeg [SR405](#)
 raise to powers [SR334](#)
 random criteria [US124](#), [US128](#), [SR409](#)
 Random Numbers [US123–US124](#), [SR317](#),
 [SR330](#)
 hints [US222](#)
 print definitions [US123](#)
 unique [SR412](#)
 RandomBeta [SR406](#)
 RandomBinomial [SR406](#)
 RandomChiSquare [SR407](#)
 RandomExponential [SR407](#)
 RandomF [SR407](#)
 RandomGamma [SR408](#)
 RandomGaussian [SR408](#)
 RandomInclusion [SR409](#)
 randomized complete block design [SR101](#),

[SR105](#)
 RandomNormal [SR410](#)
 RandomPoisson [SR410](#)
 RandomT [SR411](#)
 RandomUniform [SR411](#)
 RandomUniformInteger [SR412](#)
 Range [SR259](#), [SR412](#)
 range [US60](#), [SR3](#)
 ranges [SR336–SR338](#), [SR345](#)
 Rank [SR413](#)
 rank tests [SR150–SR151](#), [SR162](#), [SR164](#), [SR166](#)
 raw data
 contingency tables exercise [SR117](#)
 factor analysis [SR131](#)
 real data type [US73](#), [SR318](#)
 rearrange templates [US168–US169](#)
 Rebuild Template List [US168–US169](#)
 exercise [US176](#)
 Recalculate [US138–US140](#), [US164](#), [US181](#)
 background [US138](#)
 templates [US170](#)
 recalculate *see dynamic formulas*
 reciprocal powers [SR334](#)
 Recode [US117–US121](#), [SR71](#), [SR317](#), [SR330](#),
 [SR338](#), [SR359](#)
 categories [US118](#), [US238](#)
 dialog boxes [US117](#), [US120](#)
 example [US120](#)
 examples [SR342–SR343](#)
 hints [US222](#)
 missing to specified value [US120](#)
 missing values [US255](#)
 print definitions [US119–US120](#)
 troubleshoot [US255](#)
 tutorial example [US33–US35](#)
 record macros *see templates*
 rectangle tool [US205](#)
 recycle formulas [US240](#)
 recycle results *see template*
 reference level [SR206](#)
 reference lines [US184](#)
 reference structure solution [SR140](#)
 regression [SR51–SR71](#)
 data requirements [SR61](#)
 dialog boxes [SR59](#), [SR204](#)
 discussion [SR51](#)
 error distribution [SR51](#)

- error of intercept [SR56](#)
- exercises [US151](#), [SR64](#), [SR69–SR71](#)
- exponential [SR54](#), [SR68](#)
- growth [SR55](#), [SR67](#)
- line equation [US184](#)
- lines *see bivariate plots*
- logarithmic [SR54](#)
- models [SR77](#)
- multiple [SR52](#), [SR69](#), [SR77](#)
- nonlinear [SR54–SR55](#), [SR67–SR68](#)
also see logistic regression
- plots [SR231](#)
- polynomial [SR52](#), [SR64–SR65](#)
- power [SR55](#)
- residual plots [SR57](#)
- residuals [SR57](#)
- results [SR62](#), [SR207](#)
- simple [SR52](#), [SR64](#), [SR77](#)
- stepwise [US144](#), [SR52–SR54](#), [SR69](#)
- t* value [SR57](#)
- templates [US166](#), [US175](#), [SR63](#), [SR207](#)
 exercise [US166–US167](#), [US175](#)
 with ANOVA procedure [SR80](#)
 also see logistic regression
- regroup [SR360](#), [SR381](#)
- relations [SR338–SR340](#)
 ElementOf [SR345](#)
 equal [SR340](#)
 greater than [SR341](#)
 greater than or equal to [SR341](#)
 is [SR346](#)
 ISNOT [SR347](#)
 less than [SR340](#)
 less than or equal to [SR340](#)
 not equal [SR341](#)
- relative frequencies [SR15](#)
- relative risk [SR201](#)
- Remainder [SR413](#)
- remainder [SR390](#)
- remark [SR329](#)
- Remove [US144](#)
- remove variables [US63](#)
 templates [US164](#)
 tutorial example [US22](#)
- rename datasets [US70](#)
- reopen view [US157–US159](#)
- reorder category variable [US238–US240](#)
- repeat analyses *see templates*
- repeated measures analysis of variance *see analysis of variance*
- reserved words [US255–US256](#)
- Reshape [US206](#), [US208–US209](#)
 spline curves [US209](#)
- residual mean square [SR73](#)
- residuals [SR57](#), [SR59–SR61](#)
 plots [SR58](#), [SR68](#)
 proportional hazards models [SR171](#)
 saving and plotting [SR171](#)
- resize
 columns [US63](#)
 graphs [US186](#)
 imported pictures [US210](#)
 pasted object [US182](#)
 shapes [US206](#)
 tables [US199–US200](#)
 text [US204](#)
- restrict computations *see Criteria, Include Row, Exclude Row, row inclusion*
- results
 accuracy [USVI](#)
 align [US214](#)
 clean up [US213–US214](#)
 group [US215](#)
 incorrect [US250](#)
 layers [US215](#)
 list in analysis browser [US141](#)
 lock [US214](#)
 move [US214](#)
 selected [US133](#)
 unexpected [US251](#)
 ungroup [US215](#)
 unlock [US214](#)
 validation [US250](#)
- results browser [US147–US148](#)
 selected results [US133–US134](#)
 tutorial example [US43](#)
- Results Selected note [US133–US134](#), [US140](#)
- resume work [US157–US159](#), [US161](#)
- ReturnChiSquare [SR414](#)
- ReturnF [SR415](#)
- ReturnNormal [SR415](#)
- ReturnT [SR416](#)
- reuse formulas [US240](#)
- reuse results *see template*

reverse-code Likert scale [SR348](#)
rho *see* *Harrington-Fleming, Spearman*
Right Justify [US205](#)
right mouse button [USVII](#)
right to left evaluation [SR328](#)
right-justify shapes [US218](#)
root
 nth [SR334](#)
 square [SR420](#)
root curve [SRI34](#)
roots greater than one [SRI34](#)
Rotate Left/Right [US190](#)
rotate text [US205](#)
 axis values [US190](#)
rotation methods [SRI33](#)
Round [SR416](#)
Round Corners dialog box [US206](#)
rounded rectangle tool [US205–US206](#)
rounded squares [US205](#)
row and column organization *see* *data organization*
row exclusion *see* *Criteria, Include Row, Exclude Row, row inclusion*
row heights [US199](#), [US231](#)
row inclusion [US108](#), [US124](#), [SR325–SR326](#),
 [SR339](#), [SR343–SR344](#), [SR409](#)
 multiple datasets [US146](#)
 subtitles [US29](#)
 also see *Criteria, Include Row, Exclude Row*
row labels [US199](#)
row numbers, dimmed [US28](#), [US108](#), [US124](#),
 [SR326](#), [SR339](#)
RowNumber [SR417](#)
rows
 selecting [US64](#)
 transpose into columns [US68](#)
row-wise *see* *casewise*
Roy's Greatest Root [SR82](#)
rs (range span) [SR477](#)
rulers [US217](#)

S

S charts [SR255](#), [SR259](#), [SR266](#), [SR271](#)
 compared to R charts [SR259](#)
S usage marker *see* *usage markers*

sample size [SR365](#)
sample statistics [SRI](#)
save
 analysis results with view [US141](#),
 [US170–US171](#), [US232](#)
 datasets [US70](#)
 tutorial example [US41](#)
 Excel [US100](#)
 file formats [US70](#), [US156–US157](#)
 template, tutorial example [US41](#), [US47](#)
 text [US102](#)
 views [US156–US157](#)
 tutorial example [US41](#), [US47](#)
Save As *see* *save*
scattergrams
 cell plots [SR237](#)
 compare percentiles plot [SR247](#)
 confidence intervals [US242](#)
 error bars [US242](#)
 factor plots [SRI41](#)
 format [US195](#)
 regression plots [SR57](#)
 residual plots [SR57](#)
 scree plot [SRI34](#)
 templates [US171](#)
 example [US174](#)
 univariate plots [SR217](#)
 also see *bivariate plots, univariate plots*
Scheffé's *F* [SR86](#), [SR104](#)
scientific format [US79](#)
score residuals [SRI71](#), [SRI78](#)
score test [SRI70](#), [SRI79](#)
Scrapbook [US240](#)
scree plot [SRI34](#)
search [SR379](#)
Sec [SR418](#)
Second [SR419](#)
second mouse button [USVII](#)
seed *see* *random numbers*
Select [US148](#)
select
 graph components [US185](#)
 graphs [US184](#)
 rows and columns [US64](#)
 shapes [US204](#)
 table components [US198](#)
 tables [US197](#)

- variables [US57](#)
- Select a Dataset dialog box [US107](#)
- selected results [US133–US134](#)
- selection handles [US21](#), [US184–US186](#),
[US198–US199](#), [US204](#), [US207](#),
[US209–US210](#), [US214](#)
- selection tool [US185](#), [US197](#)
- SEM [SR421](#)
- semi-colons [SR324](#)
- separator characters [US100–US101](#),
[US252–US253](#)
 - importing [US252](#)
- serial autocorrelation [SR59](#)
- Series [US121–US123](#), [SR317](#), [SR330](#)
 - example [US122](#)
 - hints [US222](#)
 - print definition [US122](#)
- series functions
 - BinomialCoeffs [SR358](#)
 - CubicSeries [SR367](#)
 - ExponentialSeries [SR376](#)
 - FibonacciSeries [SR378](#)
 - GeometricSeries [SR381](#)
 - LinearSeries [SR384](#)
 - QuadraticSeries [SR404](#)
 - QuarticSeries [SR405](#)
- sets [SR336–SR338](#), [SR345](#)
 - braces [SR336](#)
- 75% variance rule [SR134](#)
- shapes
 - arcs [US205–US207](#)
 - colors [US212](#)
 - corner/center control [US205](#)
 - curves *see spline tool*
 - ellipses [US205](#)
 - fill patterns [US211](#)
 - line widths [US211](#)
 - lines [US205](#)
 - ovals [US206](#)
 - pen patterns [US211](#)
 - polygons [US205](#), [US207–US208](#)
 - rectangles [US205](#)
 - reshape [US206](#)
 - resize [US206](#)
 - rounded rectangles [US205–US206](#)
 - rounded squares [US205](#)
 - select [US204](#)
 - spline curves [US208–US210](#)
 - squares [US205](#)
 - starting point [US205](#)
 - text [US204–US205](#)
- shortcuts *see StatView Shortcuts card*
- Show [US57](#), [US181](#)
- Show Balloons [US224](#)
- Show Definition [US121](#)
- Show Grid Lines [US217](#)
- Show Page Breaks [US217](#)
- Show pop-up menu
 - analysis browser [US142](#)
 - results browser [US148](#)
- Show Rulers [US217](#)
- Show Selection [US64](#), [US133–US134](#)
- side by side column charts [SR237](#)
- sigma limits [SR258](#)
- sign of coefficients [SR54](#)
- sign test [SR119](#), [SR121](#)
 - exercise [SR125](#)
- significance level [SR90](#)
 - discussion [SR75–SR76](#)
 - post hoc tests [SR84](#)
 - also see p value*
- simple logistic regression [SR200–SR201](#),
[SR207](#)
- simple regression [SR52](#), [SR64](#), [SR77](#)
- Sin [SR419](#)
- single-byte strings
 - manipulating [SR379](#), [SR384](#), [SR423](#)
- single-spacing *see line spacing*
- singular matrix [SR43](#)
- Sinh [SR419](#)
- Size [US205](#)
- skewness [US234](#), [SR6](#), [SR17](#)
- slash [SR329](#), [SR334](#)
- slope [SR80](#), [SR200](#)
- slots for variables [US163](#)
- SMC *see squared multiple correlation*
- smooth [SR391](#)
- smoothing *see bivariate plots*
- snap to grid [US217–US218](#)
- solve problems *see troubleshoot*
- Sort [US116–US117](#), [SR413](#), [SR418](#)
 - analyses [US149–US150](#)
 - turn off [SR418](#)
 - tutorial example [US14](#)

- Undo [US117](#)
- source *see data source*
- space [SR333](#)
- SPC [SR251](#)
- Spearman rank correlation coefficient
 - data [SR123](#)
- Spearman rank correlation coefficient (rho) [SR121](#), [SR123](#)
- Special Causes Definitions table [SR270](#)
- special purpose functions
 - (), [], (), [], [] [SR337](#)
 - <, >, ≤, ≥ [SR337](#)
 - { } [SR336](#)
 - ChooseArg [SR359](#)
 - VariableElement [SR429](#)
- specification limits [SR253](#)
- sphericity [SR44](#)
- spline tool [US208–US210](#)
- Split By [US97](#), [US136](#), [US144](#), [US146](#)
 - tutorial example [US25–US26](#)
- split pane control [US55](#), [US141](#)
 - Recode [US119](#)
- Sqrt [SR420](#)
- square root [SR420](#)
- squared multiple correlation [SR133](#), [SR140](#)
- squares [US205](#)
- SS[e(i) – e(i–1)] [SR59](#)
- stabilize variance [SR386](#)
- stack order of objects [US215](#)
- tagger tick marks [US192](#)
- standard deviation [US60](#), [US184](#), [SR4](#)
 - bars on interaction plot [SR90](#)
 - lines on univariate plot [SR217](#)
- standard error [US60](#)
 - bars on interaction plot [SR90](#)
 - descriptive statistics table [SR5](#)
 - lines on univariate plot [SR217](#)
- standard error of the mean [SR5](#), [SR421](#)
- StandardDeviation [SR420](#)
- StandardError [SR421](#)
- standardize [SR397](#), [SR422](#)
- standardized regression coefficients [SR56](#)
- StandardScores [SR422](#)
- Static Formula [US116](#), [SR329–SR330](#)
 - data source [US77](#)
 - reason to use [US256](#), [SR344](#), [SR418](#)
 - also see Formula*
- stationery *see templates*
- statistical functions
 - BoxCox [SR358](#)
 - CoeffOfVariation [SR360](#)
 - Correlation [SR362](#)
 - Count [SR365](#)
 - Covariance [SR365](#)
 - GeometricMean [SR380](#)
 - Groups [SR381](#)
 - HarmonicMean [SR382](#)
 - MAD [SR387](#)
 - Maximum [SR387](#)
 - Mean [SR388](#)
 - Median [SR388](#)
 - Minimum [SR389](#)
 - Mode [SR390](#)
 - NumberMissing [SR393](#)
 - NumberOfRows [SR394](#)
 - OneGroupChiSquare [SR394](#)
 - Range [SR412](#)
 - Rank [SR413](#)
 - RowNumber [SR417](#)
 - StandardDeviation [SR420](#)
 - StandardError [SR421](#)
 - StandardScores [SR422](#)
 - TrimmedMean [SR428](#)
 - Variance [SR430](#)
- statistics texts, recommended [SR481](#)
- status bars [US140](#), [US221](#), [US223](#)
- StatView 4.x data [US70](#)
- StatView Guide [US223](#)
- StatView II/SE+Graphics file format [US105](#)
- StatView Library [US225](#), [US233](#), [US251](#)
 - categories [US91](#)
- StatView Templates folder [US161](#), [US174](#)
- step function plots [US249](#)
- stepwise regression [US144](#), [SR52–SR54](#), [SR69](#)
 - F-to-enter [SR53](#)
 - F-to-remove [SR53](#)
 - survival analysis [SR176](#), [SR186](#)
- Strata button
 - survival regression models [SR185](#)
- stratification variable
 - nonparametric analyses [SR153](#), [SR157](#)
 - regression models [SR183](#)
- strike-through text [SR323](#)
- string data type [US73](#), [SR319–SR320](#), [SR338](#)

- string functions *see text functions*
 - Student-Newman-Keuls [SR88](#)
 - Style [US205](#)
 - style sheets *see templates, preferences*
 - subgroup measurements analysis *see QC subgroup measurements analysis*
 - subgroup variables [US242](#), [SR290](#)
 - differences [US244](#)
 - formulas [SR291](#)
 - subject (group) [SR83](#)
 - subsets of data *see Criteria, Include Row, Exclude Row, row inclusion*
 - Substring [SR422](#)
 - subtitles for inclusion [US29](#)
 - subtraction [SR333](#), [SR370](#), [SR373](#)
 - unary [SR335](#)
 - Sum [SR423](#)
 - sum [US60](#), [SR333](#), [SR368](#), [SR424](#)
 - sum of squares [US60](#), [SR368](#), [SR425](#)
 - algorithms [SR433](#)
 - SumIgnoreMissing [SR424](#)
 - summary data [US52](#)
 - summary pane *see attribute pane*
 - summary statistics [US15](#), [US255](#)
 - tutorial example [US8](#)
 - SumOfColumn [SR424](#)
 - SumOfSquares [SR425](#)
 - SuperANOVA
 - file format [US70](#)
 - formulas [US256](#)
 - import/export [US106](#), [US256](#)
 - superimpose graphs [US186](#)
 - supersmoother [SR221](#), [SR225–SR227](#), [SR236](#)
 - suppress recalculation [US138–US139](#), [US170](#)
 - survival analysis [US142](#)
 - common questions [US245–US250](#)
 - example [SR146–SR147](#)
 - functions [US245](#), [SR150](#), [SR165](#), [SR178](#)
 - introduction [SR143](#)
 - nonparametric methods
 - data requirements [SR157–SR162](#)
 - dialog boxes [SR152–SR156](#)
 - discussion [SR147–SR151](#)
 - exercise [SR163–SR166](#)
 - results [SR159](#)
 - templates [SR163](#)
 - preferences [US230–US231](#)
 - regression methods
 - data requirements [SR183–SR185](#)
 - dialog boxes [SR175–SR183](#)
 - discussion [SR167](#), [SR175](#)
 - exercise [SR191–SR198](#)
 - results [SR185–SR191](#)
 - stepwise [SR176](#)
 - templates [SR191](#)
 - symbols *see point types*
 - syntax [SR339](#)
 - arguments [SR323–SR326](#), [SR331](#)
 - combine functions [SR323](#)
 - constants [SR324–SR325](#)
 - expression [SR325](#)
 - order of operations [SR326–SR328](#)
 - placeholders [SR324](#)
 - quotation marks [SR325](#), [SR331](#)
 - row inclusion [SR325–SR326](#)
 - variables [SR324–SR325](#)
 - system configuration [US74](#)
 - troubleshoot [US252](#)
 - system crash [US251](#)
 - system software [SR317](#), [SR321](#), [SR330–SR331](#)
- ## T
- t* distribution [SR404](#), [SR411](#)
 - T usage marker *see usage markers*
 - t* value
 - one sample *t*-test [SR23](#)
 - paired *t*-test [SR30](#)
 - regression [SR57](#)
 - unpaired *t*-test [SR37](#)
 - table defaults *see preferences*
 - Table dialog box [US201](#)
 - tables
 - align [US214](#)
 - arrange [US213–US214](#)
 - borders [US199](#), [US201](#), [US231](#)
 - colors [US199](#), [US202](#)
 - column widths [US199](#)
 - components [US197](#)
 - create [US131–US133](#)
 - by hand or with templates [US131](#)
 - customize [US197–US202](#)
 - decimal places [US201](#), [US231](#)
 - Edit commands [US181](#)

- edit text [US200](#)
- fonts [US199–US200](#)
- format [US135](#), [US201](#), [US231](#)
- group [US215](#)
- height and width [US199–US200](#)
- interior [US198](#)
- layers [US215](#)
- line spacing [US199](#), [US201](#), [US231](#)
- line widths [US199](#), [US202](#)
- list in analysis browser [US141](#)
- lock [US214](#)
- move [US200](#), [US214](#)
- move components [US200](#)
- notes [US199](#)
- number format [US201](#)
- numeric formats [US231](#)
- pen patterns [US199](#), [US202](#)
- preferences [US179–US180](#), [US231](#), [US250](#)
- resize [US199–US200](#)
- row and column labels [US199](#)
- row heights [US199](#), [US231](#)
- select [US197](#)
- select components [US198](#)
- structure [US201](#)
- text alignment [US199–US200](#)
- text angles [US199–US200](#)
- text colors [US199](#)
- titles [US199](#)
- transpose [US199](#), [US201](#)
- ungroup [US215](#)
- unlock [US214](#)
- tails
 - F-test [SR38](#)
 - one sample analysis [SR24](#)
 - paired comparisons [SR30](#)
 - unpaired *t*-test [SR38](#)
- Tan [SR426](#)
- Tanh [SR426](#)
- target value [SR254](#)
- Tarone-Ware test [SR151](#), [SR156](#)
- Template folder [US161](#), [US174](#)
- templates [US161–US177](#)
 - assign variables
 - dialog box [US162](#)
 - combine analyses [US162](#)
 - create [US169–US174](#)
 - exercise [US171–US177](#)
- dataset [US240](#)
- exercises [US165–US167](#), [US171](#), [US175](#)
- formats [US170](#), [US233](#)
- generic variable names [US170](#)
- graph formats [US171](#)
- manage [US167–US169](#)
- modify [US165](#), [US171](#)
 - exercise [US175](#)
- open [US157](#)
- pre-assigned variables [US170](#)
- rearrange [US168–US169](#)
- repeat analyses [US162](#)
- save [US157](#)
- save views [US157](#)
- tips [US169](#)
- variable slots [US163](#)
- vs. views [US161–US162](#)
- temporary files [US233](#)
- tension [SR226](#), [SR228](#)
- test differences among covariate levels [US248](#)
- test normality [US86](#), [US233](#), [SR13](#), [SR416](#)
- tests for special causes [SR259–SR260](#)
 - c/u* analyses [SR260](#), [SR300](#)
 - false signal [SR259](#)
 - I analyses [SR260](#)
 - individual measurements [SR278](#)
 - individual measurements
 - analyses [SR260](#)
 - p/np* analyses [SR260](#), [SR289–SR290](#)
 - subgroup measurement analyses [SR262](#)
 - subgroup measurements analysis [SR260](#)
- text
 - attributes [US205](#)
 - colors [US185](#), [US197](#), [US202](#), [US212](#)
 - edit graph text [US187](#)
 - edit table text [US200](#)
 - import/export [US100–US102](#)
 - tutorial example [US12–US13](#)
 - resize [US204](#)
 - rotate [US205](#)
 - Save As [US105](#)
 - views and templates [US233](#)
 - see text tool*
- text editor [US100](#)
- text functions [SR331–SR332](#)
 - ChooseArg [SR359](#)

- Concat [SR361](#)
- Find [SR379](#)
- Len [SR383](#)
- Substring [SR422](#)
- Text menu [US185](#), [US199](#)
- text tool [US200](#), [US204–US205](#)
 - tutorial example [US45](#)
- thickness *see line widths*
- tick marks [US183](#), [US185](#), [US191–US193](#)
 - stagger [US192](#)
- ties [SR124](#)
- Time [SR184](#), [SR427](#)
- time functions *see datetime functions*
- time series functions
 - Correlation [SR363](#)
 - Difference [SR373](#)
 - Lag [SR383](#)
 - MovingAverage [SR391](#)
- times *see multiplication, date/time*
- titles [US182](#)
 - graphs [US183](#)
 - inclusion subtitles [US29](#)
 - move [US186](#)
 - show [US188](#)
 - tables [US199](#)
- tool bar [US221](#)
- Tool tips [US221](#), [US223](#)
- transform data *see Formula, Recode*
- transformations
 - BoxCox [SR358](#)
 - Difference [SR373](#)
 - Ln [SR385](#)
 - Log [SR385](#)
 - LogB [SR386](#)
 - LogOdds [SR386](#)
- transpose
 - axes [US188](#)
 - page [US213](#)
 - rows and columns [US68](#)
 - tables [US199](#), [US201](#)
 - tutorial example [US23](#)
- trend [SR373](#)
- triangle controls
 - analysis browser [US141](#)
 - compact variables [US57](#), [US88](#), [US95](#), [US143](#)
 - Formula dialog box [US110–US111](#)
 - templates [US163](#)
 - tutorial example [US31](#)
- trigonometric functions [US112](#)
 - ArcCos [SR349](#)
 - ArcCosh [SR349](#)
 - ArcCot [SR350](#)
 - ArcCsc [SR351](#)
 - ArcSec [SR352](#)
 - ArcSin [SR353](#)
 - ArcSinh [SR354](#)
 - ArcTan [SR355](#)
 - ArcTanh [SR356](#)
 - Cos [SR363](#)
 - Cosh [SR364](#)
 - Cot [SR364](#)
 - Csc [SR366](#)
 - DegToRad [SR372](#)
 - example dataset [SR400](#)
 - RadToDeg [SR405](#)
 - Sec [SR418](#)
 - Sin [SR419](#)
 - Sinh [SR419](#)
 - Tan [SR426](#)
 - Tanh [SR426](#)
- trimmed mean [SR2](#)
- TrimmedMean [SR428](#)
- troubleshoot [US250–US256](#)
 - formulas and criteria [US116](#), [US254–US256](#)
 - general problems [US250–US251](#)
 - import [US252–US254](#)
 - print [US254](#)
 - Recode [US255](#)
 - system configuration [US252](#)
- true [SR338](#), [SR345](#)
- Trunc [SR429](#)
- truth tables [SR339–SR340](#)
- t*-test *see paired comparisons, unpaired comparisons, one sample analysis*
- Tukey-Kramer [SR87](#)
- Turn Grid On/Off [US217](#)
- turn off Criteria [US129](#)
- tutorial [US1–US48](#)
- two-way table [US53](#), [SR115](#)
- type I error [SR75–SR76](#), [SR85](#)
- type II error [SR75–SR76](#)

U

u charts [SR256](#), [SR266](#), [SR300](#), [SR304](#), [SR306](#)

UE in variable name [SR47](#)

uncensored [SRI47](#)

Undo [US95](#)

graph/table formats [USI8I](#)

Sort [USII7](#)

unexpected results [US25I](#)

Ungroup [US2I5](#)

uniform distribution [SR4II–SR4I2](#)

unique random integers [SR4I2](#)

univariate plots [SR2I7](#)

axis types [USI90](#)

confidence intervals [SR2I7](#)

connect lines [USI88](#)

data requirements [SR2I8](#)

dialog box [SR2I7](#)

discussion [SR2I7](#)

exercise [SR2I9](#)

Line Plot dialog box [SR2I7](#)

ordinal axes [USI93](#)

results [SR2I8](#)

templates [SR2I9](#)

Unlock [US2I4](#)

unpaired comparisons [SR37](#)

data [SR39](#)

data requirements [SR39](#)

dialog box [SR39](#)

discussion [SR37](#)

exercise [SR4I](#)

nonparametric [SRI20](#)

nonparametric test [SRI20](#)

results [SR4I](#)

templates [SR4I](#)

unrotated factor solution [SRI39](#), [SRI4I](#)

unsort data [SR4I8](#)

update Analyze menu [USI69](#)

update *see dynamic formulas*

usage markers [US57](#), [USI45](#), [USI63](#), [USI65](#)

tutorial example [US22](#)

user entered data source [US77](#)

V

valid range

date/time data [US74](#), [SR32I](#), [SR330–SR33I](#)

each data type [US73](#)

integer data [SR3I9](#)

validation of StatView results [US250](#)

variability

CoeffOfVariation [SR36I](#)

MAD [SR387](#)

StandardDeviation [SR42I](#)

StandardError [SR42I](#)

Variance [SR430](#)

variable attributes *see attribute pane*

variable attributes in examples [USVI](#)

variable browser [US2I](#), [US56–US57](#),

[USI43–USI47](#)

buttons [USI44](#)

exercises [USI50–USI56](#)

Force button [SR53](#)

Formula dialog box [USIIO](#)

keyboard shortcuts *see StatView Shortcuts card*

X Variable button [SR229](#)

Y Variable button [SR229](#)

variable names, rules [SR3I7](#)

variable summary pane *see attribute pane*

variable symbols *see usage markers, class markers*

variable types *see data type*

VariableElement [SR429](#)

variables [SR324–SR325](#)

assign [USI44](#)

delete [USII6](#)

names [US58](#)

change [US58](#)

generic for templates [USI70](#)

tutorial example [US4](#)

requirements *also see data requirements under specific analysis*

slots for templates [USI63](#)

vs. columns [US53](#)

Variables dialog box [SR267](#)

variable-wise *see columnwise*

Variance [SR430](#)

variance [US60](#), [SR4](#)

chi-square test [SR24](#)

comparison [SR4I](#)

test homogeneity [US235](#)

varimax [SRI35](#)

vectors [SR374](#)

also see variables
 velocity handle [US209](#)
 vertical alignment [US218](#)
 vertical *see columnwise*
 View menu [US148–US149](#)
 View pop-up menu [US147](#)
 views
 background color [US212](#)
 clean up [US213–US214](#)
 document limits [US213](#)
 documents vs. templates [US161](#)
 Edit commands [US181](#)
 file formats [US156–US157](#)
 fonts [US232](#)
 grid lines [US217](#)
 hairlines [US232](#)
 Open [US157–US159](#)
 use different datasets [US159](#)
 use different variables [US158](#)
 use original variables [US157](#)
 preferences [US141](#), [US171](#), [US179–US180](#),
 [US232–US233](#)
 print [US159–US160](#)
 rulers [US217](#)
 save results [US171](#), [US232](#)
 save templates [US157](#)
 save views as views [US156](#)
 vs. templates [US161–US162](#)
 window [US20](#), [US139–US141](#)
 Results Selected note [US133](#)
 violations of control limits [SR251](#)

W

Wald test [SR170](#), [SR179](#), [SR209](#)
 Wald-Wolfowitz runs test [SR120](#), [SR123](#)
 Weekday [SR430](#)
 WeekOfYear [SR431](#)

Weibull model [SR172–SR173](#), [SR175](#), [SR196](#)
 Welch's test template [US235–US237](#)
 Western Electric rules [SR259](#)
 Westgard rules [SR278](#)
 width [US229](#)
 graphs [US188](#)
 tables [US199–US200](#)
 Wilcoxon signed rank test [SR120](#)
 data [SR123](#)
 exercise [SR126](#)
 Wilks' Lambda [SR82](#)
 Window menu [US148–US149](#)
 within subjects [SR83](#)

X

X axis [US183](#)
 X boxes [US138](#), [US164](#)
 X usage marker *see usage markers*
 X Variable [US144](#)
 Xbar charts [SR255](#), [SR257](#), [SR266](#), [SR269](#),
 [SR274](#)
 XOR [SR347](#)

Y

Y axis [US183](#)
 Y usage marker *see usage markers*
 Y Variable [US144](#)
 Year [SR431](#)
 yin-yang cursor [US11](#), [US66](#), [US138](#)

Z

zero [SR345](#)
 z-scores [SR13](#), [SR422](#)
 z-test *see paired comparisons*