



Version 9

# Modeling and Multivariate Methods

*“The real voyage of discovery consists not in seeking new landscapes, but in having new eyes.”*

Marcel Proust

JMP, A Business Unit of SAS  
SAS Campus Drive  
Cary, NC 27513

**9.0.2**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *JMP® 9 Modeling and Multivariate Methods*. Cary, NC: SAS Institute Inc.

**JMP® 9 Modeling and Multivariate Methods**

Copyright © 2010, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-60764-595-5

All rights reserved. Produced in the United States of America.

**For a hard-copy book:** 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 the prior written permission of the publisher, SAS Institute Inc.

**For a Web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 2010

2nd printing, January 2011

3rd printing, June 2011

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at [support.sas.com/publishing](http://support.sas.com/publishing) or call 1-800-727-3228.

JMP®, SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## **Get the Most from JMP®**

Whether you are a first-time or a long-time user, there is always something to learn about JMP.

Visit JMP.com to find the following:

- live and recorded Webcasts about how to get started with JMP
- video demos and Webcasts of new features and advanced techniques
- schedules for seminars being held in your area
- success stories showing how others use JMP
- a blog with tips, tricks, and stories from JMP staff
- a forum to discuss JMP with other users

**<http://www.jmp.com/getstarted/>**



# Contents

## JMP Modeling and Multivariate Methods

---

### 1 Introduction to Model Fitting

<b>The Fit Model Platform</b>	1
The Fit Model Dialog: A Quick Example	3
Types of Models	5
The Response Buttons (Y, Weight, Freq, and By)	6
Construct Model Effects Buttons	7
Macros Popup Menu	8
Effect Attributes and Transformations	9
Transformations (Standard Least Squares Only)	12
Fitting Personalities	12
Other Model Dialog Features	14
Emphasis Choices	14
Run	14
Validity Checks	15
Other Model Specification Options	15
Formula Editor Model Features	16
Parametric Models	16
Adding Parameters	18

### 2 Standard Least Squares: Introduction

<b>The Fit Model Platform</b>	21
Launch the Platform: A Simple Example	23
Regression Plot	26
Option Packages for Emphasis	27
Whole-Model Statistical Tables	28
The Summary of Fit Table	28
The Analysis of Variance Table	29
The Lack of Fit Table	31
The Parameter Estimates Table	33

The Effect Test Table .....	34
Saturated Models .....	35
Leverage Plots .....	36
Effect Details .....	40
LSMeans Table .....	40
LSMeans Plot .....	41
LSMeans Contrast .....	43
LSMeans Student's <i>t</i> , LSMeans Tukey's HSD .....	45
Test Slices .....	46
Power Analysis .....	46
Summary of Row Diagnostics and Save Commands .....	46
Row Diagnostics .....	47
Save Columns Command .....	48
Examples with Statistical Details .....	50
One-Way Analysis of Variance with Contrasts .....	50
Analysis of Covariance .....	52
Analysis of Covariance with Separate Slopes .....	54
Singularity Details .....	55

### **3 Standard Least Squares: Perspectives on the Estimates**

Fit Model Platform .....	57
Estimates and Effect Screening Menus .....	59
Show Prediction Expression .....	59
Sorted Estimates .....	60
Expanded Estimates and the Coding of Nominal Terms .....	60
Scaled Estimates and the Coding Of Continuous Terms .....	62
Indicator Parameterization Estimates .....	63
Sequential Tests .....	63
Custom Test .....	63
Joint Factor Tests .....	65
Inverse Prediction .....	65
Cox Mixtures .....	69
Parameter Power .....	70
The Power Analysis Dialog .....	72

Effect Size .....	73
Text Reports for Power Analysis .....	74
Plot of Power by Sample Size .....	75
The Least Significant Value (LSV) .....	75
The Least Significant Number (LSN) .....	76
The Power .....	76
The Adjusted Power and Confidence Intervals .....	76
Prospective Power Analysis .....	77
Correlation of Estimates .....	78
Effect Screening .....	79
Lenth's Method .....	79
Parameter Estimates Population .....	80
Normal Plot .....	82
Half-Normal Plot .....	83
Bayes Plot .....	83
Bayes Plot for Factor Activity .....	84
Pareto Plot .....	85
<b>4 Standard Least Squares: Exploring the Prediction Equation</b>	
<b>The Fit Model Platform</b> .....	87
Exploring the Prediction Equation .....	89
The Profiler .....	89
Contour Profiler .....	90
Mixture Profiler .....	91
Surface Profiler .....	92
Interaction Plots .....	93
Cube Plots .....	94
Response Surface Curvature .....	95
Parameter Estimates .....	96
Canonical Curvature Table .....	97
Box Cox Y Transformations .....	98
<b>5 Standard Least Squares: Random Effects</b>	
<b>The Fit Model Platform</b> .....	101
Topics in Random Effects .....	103
Introduction to Random Effects .....	103
Generalizability .....	104

The REML Method .....	104
Unrestricted Parameterization for Variance Components in JMP .....	104
Negative Variances .....	105
Random Effects <i>BLUP</i> Parameters .....	105
REML and Traditional Methods Agree on the Standard Cases .....	107
<i>F</i> -Tests in Mixed Models .....	107
Specifying Random Effects .....	108
Split Plot Example .....	108
The Model Dialog .....	109
REML Results .....	110
REML Save Menu .....	III
Method of Moments Results .....	112
<b>6 Stepwise Regression</b>	
<b>The Fit Model Platform</b> .....	117
Introduction to Stepwise Regression .....	119
A Multiple Regression Example .....	119
Stepwise Regression Control Panel .....	121
Current Estimates Table .....	122
Step History Table .....	124
Forward Selection Example .....	124
Backwards Selection Example .....	124
Models with Crossed, Interaction, or Polynomial Terms .....	125
Rules for Including Related Terms .....	126
Models with Nominal and Ordinal Terms .....	127
Make Model Command for Hierarchical Terms .....	129
Logistic Stepwise Regression .....	129
All Possible Models .....	130
Model Averaging .....	131
Validation .....	133
K-Fold Crossvalidation .....	133
<b>7 Multiple Response Fitting</b>	
<b>The Fit Model Platform</b> .....	135
Multiple Response Model Specification .....	137
Initial Fit .....	137

Specification of the Response Design .....	140
Multivariate Tests .....	142
The Extended Multivariate Report .....	142
Comparison of Multivariate Tests .....	143
Univariate Tests and the Test for Sphericity .....	144
Multivariate Model with Repeated Measures .....	145
Repeated Measures Example .....	146
A Compound Multivariate Model .....	147
Commands for Response Type and Effects .....	149
Test Details (Canonical Details) .....	150
The Centroid Plot .....	150
Save Canonical Scores (Canonical Correlation) .....	151
Discriminant Analysis .....	152
<b>8 Fitting Dispersion Effects with the LogLinear Variance Model</b>	
<b>The Fit Model Platform</b> .....	155
The Loglinear Variance Model .....	157
Estimation Method .....	157
Loglinear Variance Models in JMP .....	157
Model Specification .....	157
Example .....	157
Displayed Output .....	159
Platform Options .....	160
Examining the Residuals .....	161
Profiling the Fitted Model .....	162
Comments .....	164
<b>9 Logistic Regression for Nominal and Ordinal Response</b>	
<b>The Fit Model Platform</b> .....	165
Introduction to Logistic Models .....	167
The Statistical Report .....	168
Logistic Plot .....	169
Iteration History .....	170
Whole Model Test .....	170
Lack of Fit Test (Goodness of Fit) .....	172
Parameter Estimates .....	173
Likelihood-ratio Tests .....	174

Platform Options .....	175
Plot Options .....	175
Likelihood Ratio Tests .....	175
Wald Tests for Effects .....	175
Confidence Intervals .....	175
Odds Ratios (Nominal Responses Only) .....	176
Inverse Prediction .....	179
Save Commands .....	181
ROC Curve .....	182
Lift Curve .....	183
Confusion Matrix .....	184
Profiler .....	184
Validation .....	184
Nominal Logistic Model Example: The Detergent Data .....	184
Ordinal Logistic Example: The Cheese Data .....	188
Quadratic Ordinal Logistic Example: Salt in Popcorn Data .....	193
What to Do If Your Data Are Counts in Multiple Columns .....	195

## **10 Generalized Linear Models**

The Fit Model Platform .....	197
Generalized Linear Models .....	199
The Generalized Linear Model Personality .....	199
Examples of Generalized Linear Models .....	200
Model Selection and Deviance .....	202
Examples .....	203
Poisson Regression .....	203
Poisson Regression with Offset .....	206
Normal Regression, Log Link .....	208
Platform Commands .....	212

## **11 Analyzing Screening Designs**

The Screening Platform .....	217
The Screening Platform .....	219
Using the Screening Platform .....	219
Comparing Screening and Fit Model .....	219
Launch the Platform .....	222

Report Elements and Commands .....	222
Contrasts .....	222
Half Normal Plot .....	223
Launching a Model .....	223
Tips on Using the Platform .....	223
Statistical Details .....	224
Analyzing a Plackett-Burman Design .....	226
Analyzing a Supersaturated Design .....	227
<b>12 Nonlinear Regression</b>	
<b>The Nonlinear Platform</b> .....	229
The Nonlinear Fitting Process .....	231
A Simple Exponential Example .....	231
Creating a Formula with Parameters .....	231
Launch the Nonlinear Platform .....	232
Drive the Iteration Control Panel .....	233
Using the Model Library .....	235
Customizing the Nonlinear Model Library .....	239
Details for the Formula Editor .....	239
Details of the Iteration Control Panel .....	240
Panel Buttons .....	241
The Current Parameter Estimates .....	241
Save Estimates .....	242
Confidence Limits .....	242
The Nonlinear Fit Popup Menu .....	242
Details of Solution Results .....	247
The Solution Table .....	248
Excluded Points .....	248
Profile Confidence Limits .....	248
Fitted Function Graph .....	249
Chemical Kinetics Example .....	250
How Custom Loss Functions Work .....	251
Maximum Likelihood Example: Logistic Regression .....	253
Iteratively Reweighted Least Squares Example .....	254
Probit Model with Binomial Errors: Numerical Derivatives .....	257

Poisson Loss Function .....	259
Notes Concerning Derivatives .....	261
Notes on Effective Nonlinear Modeling .....	262
Notes Concerning Scripting .....	263
Nonlinear Modeling Templates .....	264
<b>13 Creating Neural Networks</b>	
<b>Using the Neural Platform</b> .....	267
Overview of Neural Networks .....	269
Launch the Neural Platform .....	269
The Neural Launch Window .....	270
The Model Launch .....	271
Model Reports .....	276
Training and Validation Measures of Fit .....	277
Confusion Statistics .....	277
Model Options .....	278
Example of a Neural Network .....	279
<b>14 Gaussian Processes</b>	
<b>Models for Analyzing Computer Experiments</b> .....	283
Launching the Platform .....	285
The Gaussian Process Report .....	286
Actual by Predicted Plot .....	287
Model Report .....	287
Marginal Model Plots .....	288
Platform Options .....	289
Borehole Hypercube Example .....	290
<b>15 Recursive Partitioning</b>	
<b>The Partition Platform</b> .....	293
Introduction to Partitioning .....	295
Launching the Partition Platform .....	295
Partition Method .....	296
Decision Tree .....	296
Bootstrap Forest .....	306
Boosted Tree .....	309

Validation .....	313
Graphs for Goodness of Fit .....	314
Actual by Predicted Plot .....	314
ROC Curve .....	315
Lift Curves .....	317
Missing Values .....	318
Example .....	319
Decision Tree .....	320
Bootstrap Forest .....	321
Boosted Tree .....	323
Compare Methods .....	324
Statistical Details .....	326
<b>16 Time Series Analysis</b>	
<b>The Time Series Platform</b> .....	329
Launch the Platform .....	331
Select Columns into Roles .....	331
The Time Series Graph .....	332
Time Series Commands .....	332
Graph .....	333
Autocorrelation .....	333
Partial Autocorrelation .....	333
Variogram .....	334
AR Coefficients .....	334
Spectral Density .....	335
Save Spectral Density .....	336
Number of Forecast Periods .....	337
Difference .....	337
Modeling Reports .....	338
Model Comparison Table .....	338
Model Summary Table .....	339
Parameter Estimates Table .....	341
Forecast Plot .....	342
Residuals .....	342
Iteration History .....	342
Model Report Options .....	343
ARIMA Model .....	343

Seasonal ARIMA .....	345
ARIMA Model Group .....	345
Transfer Functions .....	346
Report and Menu Structure .....	346
Diagnostics .....	348
Model Building .....	349
Transfer Function Model .....	350
Model Reports .....	352
Model Comparison Table .....	354
Fitting Notes .....	354
Smoothing Models .....	354
Smoothing Model Dialog .....	355
Simple Exponential Smoothing .....	356
Double (Brown) Exponential Smoothing .....	357
Linear (Holt) Exponential Smoothing .....	357
Damped-Trend Linear Exponential Smoothing .....	357
Seasonal Exponential Smoothing .....	358
Winters Method (Additive) .....	358
<b>17 Categorical Response Analysis</b>	
The Categorical Platform .....	361
The Categorical Platform .....	363
Launching the Platform .....	363
Failure Rate Examples .....	366
Response Frequencies .....	366
Indicator Group .....	367
Multiple Delimited .....	367
Multiple Response By ID .....	368
Multiple Response .....	368
Categorical Reports .....	369
Report Content .....	369
Report Format .....	371
Statistical Commands .....	373
Save Tables .....	376
<b>18 Choice Modeling</b>	
Choice Platform .....	379
Introduction to Choice Modeling .....	381

Choice Statistical Model .....	381
Product Design Engineering .....	381
Data for the Choice Platform .....	382
Example: Pizza Choice .....	382
Launch the Choice Platform and Select Data Sets .....	384
Choice Model Output .....	387
Subject Effects .....	389
Utility Grid Optimization .....	391
Platform Options .....	392
Example: Valuing Trade-offs .....	393
One-Table Analysis .....	399
Example: One-Table Pizza Data .....	400
Segmentation .....	402
Special Data Rules .....	406
Default choice set .....	406
Subject Data with Response Data .....	407
Logistic Regression .....	407
Transforming Data .....	411
Transforming Data to Two Analysis Tables .....	411
Transforming Data to One Analysis Table .....	415

## **19 Correlations and Multivariate Techniques**

The Multivariate Platform .....	419
Launch the Platform and Select Options .....	421
Correlations Multivariate .....	422
CI of Correlation .....	423
Inverse Correlations and Partial Correlations .....	423
Set $\alpha$ Level .....	424
Scatterplot Matrix .....	424
Covariance Matrix .....	426
Pairwise Correlations .....	426
Color Maps .....	427
Simple Statistics .....	428
Nonparametric Correlations .....	428
Outlier Analysis .....	429
Principal Components .....	432

Item Reliability .....	432
Parallel Coordinate Plot .....	434
Ellipsoid 3D Plot .....	434
Impute Missing Data .....	435
Computations and Statistical Details .....	436
Pearson Product-Moment Correlation .....	436
Nonparametric Measures of Association .....	436
Inverse Correlation Matrix .....	438
Distance Measures .....	438
Cronbach's $\alpha$ .....	439
<b>20 Clustering</b>	
<b>The Cluster Platform</b> .....	441
Introduction to Clustering Methods .....	443
The Cluster Launch Dialog .....	444
Hierarchical Clustering .....	445
Hierarchical Cluster Options .....	447
Technical Details for Hierarchical Clustering .....	448
K-Means Clustering .....	450
K-Means Control Panel .....	451
K-Means Report .....	453
Normal Mixtures .....	454
Robust Normal Mixtures .....	456
Platform Options .....	458
Details of the Estimation Process .....	458
Self Organizing Maps .....	458
<b>21 Principal Components</b>	
<b>Reducing Dimensionality</b> .....	461
Principal Components .....	463
Launch the Platform .....	463
Report .....	464
Platform Options .....	464
Factor Analysis .....	467

<b>22 Discriminant Analysis</b>	
<b>The Discriminant Platform</b>	471
Introduction	473
Discriminating Groups	473
Discriminant Method	474
Stepwise Selection	475
Canonical Plot	477
Discriminant Scores	477
Commands and Options	478
Validation	483
<b>23 Partial Least Squares</b>	
<b>The PLS Platform</b>	485
PLS	487
Launch the Platform	487
Model Coefficients and PRESS Residuals	492
Validation	494
Platform Options	494
Statistical Details	499
Centering and Scaling	499
Missing Values	499
<b>24 Item Response Theory</b>	
<b>The Item Analysis Platform</b>	501
Introduction to Item Response Theory	503
Launching the Platform	506
Item Analysis Output	508
Characteristic Curves	508
Information Curves	509
Dual Plots	509
Platform Commands	511
Technical Details	512
<b>25 Plotting Surfaces</b>	
<b>The Surface Plot Platform</b>	513
Surface Plots	515

Launching the Platform .....	515
Plotting a Single Mathematical Function .....	516
Plotting Points Only .....	517
Plotting a Formula from a Column .....	518
Isosurfaces .....	520
The Surface Plot Control Panel .....	522
Appearance Controls .....	523
Independent Variables .....	523
Dependent Variables .....	524
Plot Controls and Settings .....	525
Rotate .....	525
Axis Settings .....	526
Lights .....	527
Sheet or Surface Properties .....	527
Other Properties and Commands .....	528
Keyboard Shortcuts .....	529

## 26 Profiling

<b>Response Surface Visualization, Optimization, and Simulation .....</b>	<b>531</b>
Introduction to Profiling .....	533
Profiling Features in JMP .....	533
The Profiler .....	535
Interpreting the Profiles .....	536
Profiler Options .....	540
Desirability Profiling and Optimization .....	545
Special Profiler Topics .....	549
Propagation of Error Bars .....	549
Customized Desirability Functions .....	550
Mixture Designs .....	552
Expanding Intermediate Formulas .....	553
Linear Constraints .....	553
Contour Profiler .....	555
Contour Profiler Pop-up Menu .....	556
Mixtures .....	556
Constraint Shading .....	557
Mixture Profiler .....	557
Explanation of Ternary Plot Axes .....	558

More than Three Mixture Factors .....	559
Mixture Profiler Options .....	560
Linear Constraints .....	561
Examples .....	561
Surface Profiler .....	569
The Custom Profiler .....	569
Custom Profiler Options .....	569
The Simulator .....	570
Specifying Factors .....	571
Specifying the Response .....	573
Run the Simulation .....	573
The Simulator Menu .....	574
Using Specification Limits .....	574
Simulating General Formulas .....	576
The Defect Profiler .....	579
Notes .....	581
Defect Profiler Example .....	581
Stochastic Optimization Example .....	585
Noise Factors (Robust Engineering) .....	593
Profiling Models Stored in Excel .....	599
The Excel Model .....	600
Using the JMP Add-In for Profiling .....	601
Using the Excel Profiler From JMP .....	604
Fit Group .....	604
Statistical Details .....	605

## A Statistical Details

<b>Models in JMP .....</b>	<b>607</b>
The Response Models .....	609
Continuous Responses .....	609
Nominal Responses .....	609
Ordinal Responses .....	611
The Factor Models .....	612
Continuous Factors .....	612
Nominal Factors .....	613
Ordinal Factors .....	622
The Usual Assumptions .....	628

Assumed Model .....	628
Relative Significance .....	628
Multiple Inferences .....	629
Validity Assessment .....	629
Alternative Methods .....	629
Key Statistical Concepts .....	630
Uncertainty, a Unifying Concept .....	630
The Two Basic Fitting Machines .....	631
Leverage Plot Details .....	633
Multivariate Details .....	636
Multivariate Tests .....	636
Approximate <i>F</i> -Test .....	637
Canonical Details .....	637
Discriminant Analysis .....	638
Power Calculations .....	639
Computations for the LSV .....	639
Computations for the LSN .....	640
Computations for the Power .....	640
Computations for Adjusted Power .....	640
Inverse Prediction with Confidence Limits .....	641
Details of Random Effects .....	642
<b>Index</b>	
<b>JMP Modeling and Multivariate Methods .....</b>	<b>655</b>

# Credits and Acknowledgments

---

## Origin

JMP was developed by SAS Institute Inc., Cary, NC. JMP is not a part of the SAS System, though portions of JMP were adapted from routines in the SAS System, particularly for linear algebra and probability calculations. Version 1 of JMP went into production in October 1989.

## Credits

JMP was conceived and started by John Sall. Design and development were done by John Sall, Chung-Wei Ng, Michael Hecht, Richard Potter, Brian Corcoran, Annie Dudley Zangi, Bradley Jones, Craige Hales, Chris Gotwalt, Paul Nelson, Xan Gregg, Jianfeng Ding, Eric Hill, John Schroedl, Laura Lancaster, Scott McQuiggan, Melinda Thielbar, Clay Barker, Peng Liu, Dave Barbour, Jeff Polzin, John Ponte, and Steve Amerige.

In the SAS Institute Technical Support division, Duane Hayes, Wendy Murphrey, Rosemary Lucas, Win LeDinh, Bobby Riggs, Glen Grimme, Sue Walsh, Mike Stockstill, Kathleen Kiernan, and Liz Edwards provide technical support.

Nicole Jones, Kyoko Keener, Hui Di, Joseph Morgan, Wenjun Bao, Fang Chen, Susan Shao, Michael Crotty, Jong-Seok Lee, Tonya Mauldin, Audrey Ventura, Ani Eloyan, Bo Meng, and Sequola McNeill provide ongoing quality assurance. Additional testing and technical support are provided by Noriki Inoue, Kyoko Takenaka, Yusuke Ono, Masakazu Okada, and Naohiro Masukawa from SAS Japan.

Bob Hickey and Jim Borek are the release engineers.

The JMP books were written by Ann Lehman, Lee Creighton, John Sall, Bradley Jones, Erin Vang, Melanie Drake, Meredith Blackwelder, Diane Perhac, Jonathan Gatlin, Susan Conaghan, and Sheila Loring, with contributions from Annie Dudley Zangi and Brian Corcoran. Creative services and production was done by SAS Publications. Melanie Drake implemented the Help system.

Jon Weisz and Jeff Perkinson provided project management. Also thanks to Lou Valente, Ian Cox, Mark Bailey, and Malcolm Moore for technical advice.

Thanks also to Georges Guirguis, Warren Sarle, Gordon Johnston, Duane Hayes, Russell Wolfinger, Randall Tobias, Robert N. Rodriguez, Ying So, Warren Kuhfeld, George MacKensie, Bob Lucas, Warren Kuhfeld, Mike Leonard, and Padraic Neville for statistical R&D support. Thanks are also due to Doug Melzer, Bryan Wolfe, Vincent DelGobbo, Biff Beers, Russell Gonsalves, Mitchel Soltys, Dave Mackie, and Stephanie Smith, who helped us get started with SAS Foundation Services from JMP.

## Acknowledgments

We owe special gratitude to the people that encouraged us to start JMP, to the alpha and beta testers of JMP, and to the reviewers of the documentation. In particular we thank Michael Benson, Howard Yetter (d),

Andy Mauromoustakos, Al Best, Stan Young, Robert Muenchen, Lenore Herzenberg, Ramon Leon, Tom Lange, Homer Hegedus, Skip Weed, Michael Emptage, Pat Spagan, Paul Wenz, Mike Bowen, Lori Gates, Georgia Morgan, David Tanaka, Zoe Jewell, Sky Alibhai, David Coleman, Linda Blazek, Michael Friendly, Joe Hockman, Frank Shen, J.H. Goodman, David Iklé, Barry Hembree, Dan Obermiller, Jeff Sweeney, Lynn Vanatta, and Kris Ghosh.

Also, we thank Dick DeVaux, Gray McQuarrie, Robert Stine, George Fraction, Avigdor Cahaner, José Ramirez, Gudmunder Axelsson, Al Fulmer, Cary Tuckfield, Ron Thisted, Nancy McDermott, Veronica Czitrom, Tom Johnson, Cy Wegman, Paul Dwyer, DaRon Huffaker, Kevin Norwood, Mike Thompson, Jack Reese, Francois Mainville, and John Wass.

We also thank the following individuals for expert advice in their statistical specialties: R. Hocking and P. Spector for advice on effective hypotheses; Robert Mee for screening design generators; Roselinde Kessels for advice on choice experiments; Greg Piepel, Peter Goos, J. Stuart Hunter, Dennis Lin, Doug Montgomery, and Chris Nachtsheim for advice on design of experiments; Jason Hsu for advice on multiple comparisons methods (not all of which we were able to incorporate in JMP); Ralph O'Brien for advice on homogeneity of variance tests; Ralph O'Brien and S. Paul Wright for advice on statistical power; Keith Muller for advice in multivariate methods, Harry Martz, Wayne Nelson, Ramon Leon, Dave Trindade, Paul Tobias, and William Q. Meeker for advice on reliability plots; Lijian Yang and J.S. Marron for bivariate smoothing design; George Milliken and Yurii Bulavski for development of mixed models; Will Potts and Cathy Maahs-Fladung for data mining; Clay Thompson for advice on contour plotting algorithms; and Tom Little, Damon Stoddard, Blanton Godfrey, Tim Clapp, and Joe Ficalora for advice in the area of Six Sigma; and Josef Schmee and Alan Bowman for advice on simulation and tolerance design.

For sample data, thanks to Patrice Strahle for Pareto examples, the Texas air control board for the pollution data, and David Coleman for the pollen (eureka) data.

## **Translations**

Trish O'Grady coordinates localization. Special thanks to Noriki Inoue, Kyoko Takenaka, Masakazu Okada, Naohiro Masukawa, and Yusuke Ono (SAS Japan); and Professor Toshiro Haga (retired, Tokyo University of Science) and Professor Hirohiko Asano (Tokyo Metropolitan University) for reviewing our Japanese translation; François Bergeret for reviewing the French translation; Bertram Schäfer and David Meinstrup (consultants, StatCon) for reviewing the German translation; Patrizia Omodei, Maria Scaccabarozzi, and Letizia Bazzani (SAS Italy) for reviewing the Italian translation; RuiQi Qiao, Rula Li, and Molly Li for reviewing Simplified Chinese translation (SAS R&D Beijing); Finally, thanks to all the members of our outstanding translation and engineering teams.

## **Past Support**

Many people were important in the evolution of JMP. Special thanks to David DeLong, Mary Cole, Kristin Nauta, Aaron Walker, Ike Walker, Eric Gjertsen, Dave Tilley, Ruth Lee, Annette Sanders, Tim Christensen, Eric Wasserman, Charles Soper, Wenjie Bao, and Junji Kishimoto. Thanks to SAS Institute quality assurance by Jeanne Martin, Fouad Younan, and Frank Lassiter. Additional testing for Versions 3 and 4 was done by Li Yang, Brenda Sun, Katrina Hauser, and Andrea Ritter.

Also thanks to Jenny Kendall, John Hansen, Eddie Routten, David Schlitzhauer, and James Mulherin. Thanks to Steve Shack, Greg Weier, and Maura Stokes for testing JMP Version 1.

Thanks for support from Charles Shipp, Harold Gugel (d), Jim Winters, Matthew Lay, Tim Rey, Rubin Gabriel, Brian Ruff, William Lisowski, David Morganstein, Tom Esposito, Susan West, Chris Fehily, Dan Chilko, Jim Shook, Ken Bodner, Rick Blahunka, Dana C. Aultman, and William Fehlner.

### **Technology License Notices**

Scintilla is Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>. NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

XRender is Copyright © 2002 Keith Packard. KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ImageMagick software is Copyright © 1999-2010 ImageMagick Studio LLC, a non-profit organization dedicated to making software imaging solutions freely available.

bzlib software is Copyright © 1991-2009, Thomas G. Lane, Guido Vollbeding. All Rights Reserved.

FreeType software is Copyright © 1996-2002 The FreeType Project ([www.freetype.org](http://www.freetype.org)). All rights reserved.



# Chapter 1

## Introduction to Model Fitting

### The Fit Model Platform

---

**Analyze > Fit Model** launches the *general fitting platform*, which lets you construct linear models that have more complex effects than those assumed by other JMP statistical platforms. **Fit Model** displays the Fit Model dialog that lets you define complex models. You choose specific model effects and error terms and add or remove terms from the model specification as needed.

The Fit Model dialog is a unified launching pad for a variety of fitting *personalities* such as:

- standard least squares fitting of one or more continuous responses (multiple regression)
- screening analysis for experimental data where there are many effects but few observations
- stepwise regression
- multivariate analysis of variance (MANOVA) for multiple continuous responses
- log-linear variance fitting to fit both means and variances
- logistic regression for nominal or ordinal response
- proportional hazard and parametric survival fits for censored survival data
- generalized linear model fitting with various distributions and link functions.

This chapter describes the Fit Model dialog in detail and defines the report surface display options and save commands available with various statistical analyses. The chapters “[Standard Least Squares: Introduction](#),” p. 21, “[Standard Least Squares: Perspectives on the Estimates](#),” p. 57, “[Standard Least Squares: Exploring the Prediction Equation](#),” p. 87, “[Standard Least Squares: Random Effects](#),” p. 101, “[Generalized Linear Models](#),” p. 197, “[Fitting Dispersion Effects with the LogLinear Variance Model](#),” p. 155, “[Stepwise Regression](#),” p. 117, and “[Logistic Regression for Nominal and Ordinal Response](#),” p. 165, and “[Multiple Response Fitting](#),” p. 135, discuss standard least squares and give details and examples for each Fit Model personality.

# Contents

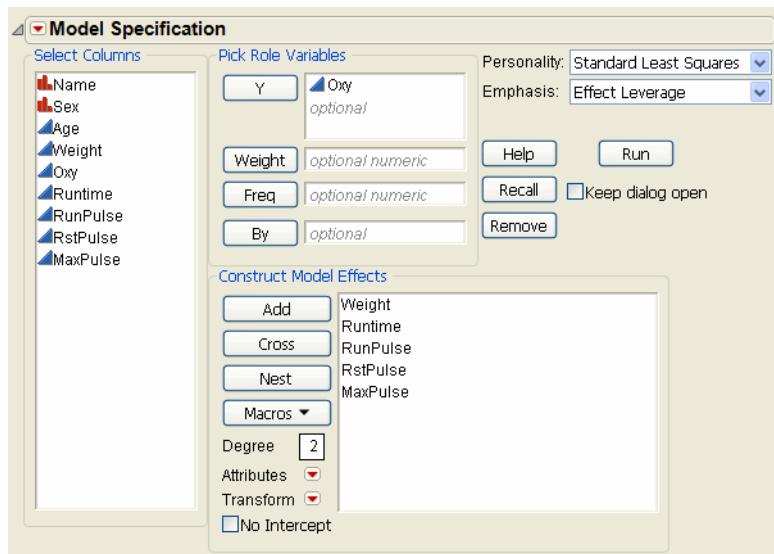
The Fit Model Dialog: A Quick Example .....	3
Types of Models .....	5
The Response Buttons (Y, Weight, Freq, and By) .....	6
Construct Model Effects Buttons .....	7
Macros Popup Menu.....	8
Effect Attributes and Transformations.....	9
Transformations (Standard Least Squares Only) .....	12
Fitting Personalities .....	12
Other Model Dialog Features .....	14
Emphasis Choices .....	14
Run.....	14
Validity Checks.....	15
Other Model Specification Options.....	15
Formula Editor Model Features.....	16
Parametric Models.....	16
Adding Parameters.....	18

## The Fit Model Dialog: A Quick Example

The **Fit Model** command first displays the Fit Model dialog, shown in Figure 1.1. You use this dialog to tailor a model for your data. If you select the **Keep dialog open** check box, this dialog persists until you explicitly close it. This is useful to change the model specification and fit several models before closing the window.

The example in Figure 1.1, uses the **Fitness.jmp** (SAS Institute 1987) data table in the Sample Data folder. The data are results of an aerobic fitness study. The **Oxy** variable is a continuous response (dependent) variable. The variables **Weight**, **Runtime**, **RunPulse**, **RstPulse**, and **MaxPulse** that show in the Construct Model Effects list are continuous effects (also called regressors, factors, or independent variables). The popup menu at the upper-right of the dialog shows **Standard Least Squares**, which defines the fitting method or fitting *personality*. The various fitting personalities are described later in this chapter.

**Figure 1.1** The Fit Model Dialog Completed for a Multiple Regression



This standard least squares example, with a single continuous  $Y$  variable and several continuous  $X$  variables, specifies a multiple regression.

After a model is defined in the Fit Model dialog, click **Run** to perform the analysis and generate a report window with the appropriate tables and supporting graphics.

A standard least squares analysis such as this multiple regression begins by showing you the Summary of Fit table, the Parameter Estimates table, the Effect Tests table, Analysis of Variance, and the Residual by Predicted and Leverage plots. If you want, you can open additional tables and plots, such as those shown in Figure 1.2, to see details of the analysis. Or, if a screening or response surface analysis seems more

## The Fit Model Dialog: A Quick Example

appropriate, you can choose commands from the Effect Screening and Factor Profiling menus at the top of the report.

All tables and graphs available on the Fit Model platform are discussed in detail in the chapters “[Standard Least Squares: Introduction](#),” p. 21, and “[Standard Least Squares: Perspectives on the Estimates](#),” p. 57, and “[Standard Least Squares: Exploring the Prediction Equation](#),” p. 87.

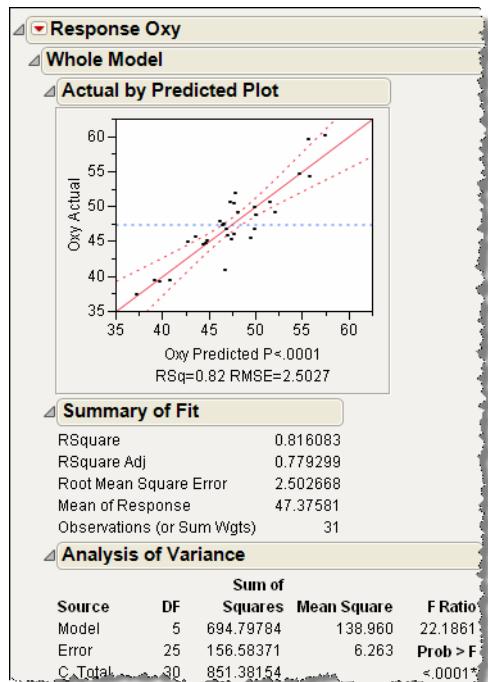
See [Table 1.1 “Types of Models](#),” p. 5, and [Table 1.2 “Clicking Sequences for Selected Models](#),” p. 6, in the next section for a description of models and the clicking sequences needed to enter them into the Fit Model dialog.

### Detailed Example of the Fit Model Dialog

1. Open the **Fitness.jmp** sample data table.
2. Select **Analyze > Fit Model**.
3. Select Oxy and click Y.
4. Select Weight, Runtime, RunPulse, RstPulse, and MaxPulse and click **Add**.
5. Click **Run**.

---

**Figure 1.2** Partial Report for Multiple Regression



## Types of Models

The list in [Table 1.1 “Types of Models,” p. 5](#) is a catalog of model examples that can be defined using the Fit Model dialog, where the effects  $X$  and  $Z$  have continuous values, and  $A$ ,  $B$ , and  $C$  have nominal or ordinal values. This list is not exhaustive.

When you correctly specify the type of model, the model effects show in the **Construct Model Effects** list on the Fit Model dialog. Refer to [Table 1.2 “Clicking Sequences for Selected Models,” p. 6](#) to see the clicking sequences that produce some of these sets of model effects.

**Table 1.1** Types of Models

Type of Model	Model Effects
simple regression	$X$
polynomial ( $x$ ) to Degree 2	$X, X^2$
polynomial ( $x, z$ ) to Degree 2	$X, X^2, Z, Z^2$
cubic polynomial ( $x$ )	$X, X^2, X^3$
multiple regression	$X, Z, \dots$ , other continuous columns
one-way analysis of variance	$A$
two-way main effects analysis of variance	$A, B$
two-way analysis of variance with interaction	$A, B, A^2B$
three-way full factorial	$A, B, A^2B, C, A^2C, B^2C, A^2B^2C$
analysis of covariance, same slopes	$A, X$
analysis of covariance, separate slopes	$A, X, A^2X$
simple nested model	$A, B[A]$
compound nested model	$A, B[A], C[A B]$
simple split-plot or repeated measure	$A, B[A]&Random, C, A^2C$
response surface ( $x$ ) model	$X&RS, X^2$
response surface ( $x, z$ ) model	$X&RS, Z&RS, X^2, Z^2, X^2Z$
MANOVA	multiple $Y$ variables

The following convention is used to specify clicking:

- If a column name is in plain typeface, click the name in the column selection list.

## The Response Buttons (Y, Weight, Freq, and By)

- If a column name is **bold**, then select that column in the dialog model effects list.
- The name of the button to click is shown in all CAPITAL LETTERS.

**Table 1.2** Clicking Sequences for Selected Models

Type of Model	Clicking Sequence
simple regression	X, ADD
polynomial ( <i>x</i> ) to Degree 2	X, <b>Degree 2</b> , select <b>Polynomial to Degree</b> in the <b>Macros</b> popup menu
polynomial ( <i>x, z</i> ) to Degree 3	X, Z, <b>Degree 3</b> , select <b>Polynomial to Degree</b> in the <b>Macros</b> popup menu
multiple regression	X, ADD, Z, ADD,... or X, Z, ..., ADD
one-way analysis of variance	A, ADD
two-way main effects analysis of variance	A, ADD B, ADD, or A, B, ADD
two-way analysis of variance with interaction	A, B, ADD, A, B, CROSS or A, B, select <b>Full Factorial</b> in <b>Macros</b> popup menu
three-way full factorial	A, B, C, select <b>Full Factorial</b> in <b>Macros</b> popup menu
analysis of covariance, same slopes	A, ADD, X, ADD or A, X, ADD
analysis of covariance, separate slopes	A, ADD, X, ADD, A, X, CROSS
simple nested model	A, ADD, B, ADD, A, <b>B</b> , NEST, or A, B, ADD, A, <b>B</b> , NEST
compound nested model	A, B, ADD, A, <b>B</b> , NEST, C, ADD, A, B, <b>C</b> , NEST
simple split-plot or repeated measure	A, ADD, B, ADD, A, <b>B</b> , NEST, select <b>Random</b> from the <b>Effect Attributes</b> popup menu, C, ADD, C, A, CROSS
two-factor response surface	X, Z, select <b>Response Surface</b> in the <b>Macros</b> popup menu

---

**The Response Buttons (Y, Weight, Freq, and By)**

The column selection list in the upper left of the dialog lists the columns in the current data table. When you click a column name, it highlights and responds to the action that you choose with other buttons on the

dialog. Either drag across columns or Shift-click to extend a selection of column names, and Control-click (⌘-click on the Macintosh) to select non-adjacent names.

To assign variables to the **Y**, **Weight**, **Freq**, or **By** roles, select them and click the corresponding role button:

- **Y** identifies one or more response variables (the dependent variables) for the model.
- **Weight** is an optional role that identifies one column whose values supply weights for the response. Weighting affects the importance of each row in the model.
- **Freq** is an optional role that identifies one column whose values assign a frequency to each row for the analysis. The values of this variable determine how degrees of freedom are counted.
- **By** is used to perform a separate analysis for each level of a classification or grouping variable.

If you want to remove variables from roles, highlight them and click **Remove** or hit the backspace key.

---

## Construct Model Effects Buttons

Suppose that a data table contains the variables *X* and *Z* with continuous values, and *A*, *B*, and *C* with nominal values. The following paragraphs describe the buttons in the Fit Model dialog that specify model effects.

### Add

To add a simple regressor (continuous column) or a main effect (nominal or ordinal column) as an *x* effect to any model, select the column from the column selection list and click **Add**. That column name appears in the model effects list. As you add effects, be aware of the modeling type declared for that variable and the consequences that modeling type has when fitting the model:

- Variables with continuous modeling type become simple regressors.
- Variables with nominal modeling types are represented with dummy variables to fit separate coefficients for each level in the variable.
- Nominal and ordinal modeling types are handled alike, but with a slightly different coding. See the appendix “[Statistical Details](#),” p. 607, for details on coding of effects.

If you mistakenly add an effect, select it in the model effects list and click **Remove**.

### Cross

To create a compound effect by crossing two or more variables, Shift-click in the column selection list to select them if they are next to each other in the **Select Columns** list. Control-click (⌘-click on the Macintosh) if the variables are not contiguous in the column selection list. Then click **Cross**:

- Crossed nominal variables become interaction effects.
- Crossed continuous variables become multiplied regressors.
- Crossing a combination of nominal and continuous variables produces special effects suitable for testing homogeneity of slopes.

If you select both a column name in the column selection list and an effect in the model effects list, the **Cross** button crosses the selected column with the selected effect and adds this compound effect to the effects list.

See the appendix “[Statistical Details](#),” p. 607, for a discussion of how crossed effects are parameterized and coded.

## Nest

When levels of an effect B only occur within a single level of an effect A, then B is said to be *nested* within A and A is called the *outside effect*. The notation B[A] is read “B within A” and means that you want to fit a B effect within each level of A. The B[A] effect is like pooling B and A\*B. To specify a nested effect

- select the outside effects in the column selection list and click **Add or Cross**
- select the nested effect in the column selection list and click **Add or Cross**
- select the outside effect in the column selection list
- select the nested variable in the model effects list and click **Nest**

For example, suppose that you want to specify A\*B[C]. Highlight both A and B in the column selection list. Then click **Cross** to see A\*B in the model effects list. Highlight the A\*B term in the model effects list and C in the column selection list. Click **Nest** to see A\*B[C] as a model effect.

**Note:** The nesting terms must be specified in order, from outer to inner. If B is nested within A, and C is nested within B, then the model is specified as: A, B[A], C[B,A].

JMP allows up to ten terms to be combined as crossed and nested.

---

## Macros Popup Menu

Commands from the **Macros** popup menu automatically generate the effects for commonly used models. You can add or remove terms from an automatic model specification as needed.

**Full Factorial** To look at many crossed factors, such as in a factorial design, use **Full Factorial**. It creates the set of effects corresponding to all crossings of all variables you select in the columns list. For example, if you select variables A, B, and C, the **Full Factorial** selection generates A, B, A\*B, C, A\*C, B\*C, A\*B\*C in the effect lists. To remove unwanted model interactions, highlight them and click **Remove**.

**Factorial to Degree** To create a limited factorial, enter the degree of interactions that you want in the **Degree** box, then select **Factorial to a Degree**.

**Factorial Sorted** The **Factorial Sorted** selection creates the same set of effects as **Full Factorial** but lists them in order of degree. All main effects are listed first, followed by all two-way interactions, then all three-way interactions, and so forth.

**Response Surface** In a response surface model, the object is to find the values of the terms that produce a maximum or minimum expected response. This is done by fitting a collection of terms in

a quadratic model. The critical values for the surface are calculated from the parameter estimates and presented with a report on the shape of the surface.

To specify a response surface effect, select two or more variables in the column selection list. When you choose **Response Surface** from the **Effect Macros** popup menu, response surface expressions appear in the model effects list. For example, if you have variables *A* and *B*

Surface(*A*) fits  $\beta_1 A + \beta_{11} A^2$

Surface(*A,B*) fits  $\beta_1 A + \beta_{11} A^2 + \beta_2 B + \beta_{22} B^2 + \beta_{12} AB$

The response surface effect attribute, prefixed with an ampersand (&), is automatically appended to the effect name in the model effects list. The next section discusses the **Attributes** popup menu.

**Mixture Response Surface** The **Mixture Response Surface** design omits the squared terms and the intercept and attaches the &RS effect attribute to the main effects. For more information see the *Design of Experiments* Guide.

**Polynomial to Degree** A polynomial effect is a series of terms that are powers of one variable. To specify a polynomial effect:

- click one or more variables in the column selection list;
- enter the degree of the polynomial in the **Degree** box;
- use the **Polynomial to Degree** command in the **Macros** popup menu.

For example, suppose you select variable *x*. A 2<sup>nd</sup> polynomial in the effects list fits parameters for *x* and *x*<sup>2</sup>. A 3<sup>rd</sup> polynomial in the model dialog fits parameters for *x*, *x*<sup>2</sup>, and *x*<sup>3</sup>.

**Scheffe Cubic** When you fit a 3rd degree polynomial model to a mixture, it turns out that you need to take special care not to introduce even-powered terms, because they are not estimable. When you get up to a cubic model, this means that you can't specify an effect like X1\*X1\*X2. However, it turns out that a complete polynomial specification of the surface should introduce terms of the form:

$$X1*X2*(X1 - X2)$$

which we call *Scheffe Cubic* terms.

In the Fit Model dialog, this macro creates a complete cubic model, including the Scheffe Cubic terms if you either (a) enter a 3 in the Degree box, then do a “Mixture Response Surface” command on a set of mixture columns, or(b) use the **Scheffe Cubic** command in the Macro button.

**Radial** fits a radial-basis smoothing function using the selected variables.

## Effect Attributes and Transformations

The **Attributes** popup menu has five special attributes that you can assign to an effect in a model:

**Random Effect** If you have multiple error terms or random effects, as with a split-plot or repeated measures design, you can highlight them in the model effects list and select **Random Effect** from the **Attributes** popup menu. See the chapter “[Standard Least Squares: Random Effects](#),” p. 101, for a detailed discussion and example of random effects.

**Response Surface Effect** If you have a response surface model, you can use this attribute to identify which factors participate in the response surface. This attribute is automatically assigned if you use the **Response Surface** effects macro. You need only identify the main effects, not the crossed terms, to obtain the additional analysis done for response surface factors.

**Log Variance Effect** Sometimes the goal of an experiment is not just to maximize or minimize the response itself but to aim at a target response and achieve minimum variability. To analyze results from this kind of experiment:

1. Assign the response ( $Y$ ) variable and choose **Loglinear Variance** as the fitting personality
2. Specify loglinear variance effects by highlighting them in the model effects list and select **LogVariance Effect** from the **Attributes** popup menu. The effect appears in the model effects list with **&LogVariance** next to its name.

If you want an effect to be used for both the mean and variance of the response, you must specify it as an effect twice, once with the **LogVariance Effect** attribute.

**Mixture Effect** You can use the **Mixture Effect** attribute to specify a mixture model effect without using the **Mixture Response Surface** effects macro. If you don't use the effects macro you have to add this attribute to the effects yourself so that the model understands which effects are part of the mixture.

**Excluded Effect** Use the **Excluded Effect** attribute when you want to estimate least squares means of an interaction, or include it in lack of fit calculations, but don't want it in the model.

**Knotted Spline Effect** Use the **Knotted Spline Effect** attribute to have JMP fit a segmentation of smooth polynomials to the specified effect. When you select this attribute, a dialog box appears that allows you to specify the number of knot points.

**Note:** Knotted splines are only implemented for main-effect continuous terms.

JMP follows the advice in the literature in positioning the points. The knotted spline is also referred to as a Stone spline or a Stone-Koo spline. See Stone and Koo (1986). If there are 100 or fewer points, the first and last knot are the fifth point inside the minimum and maximum, respectively. Otherwise, the first and last knot are placed at the 0.05 and 0.95 quantile if there are 5 or fewer knots, or the 0.025 and 0.975 quantile for more than 5 knots. The default number of knots is 5 unless there are less than or equal to 30 points, in which case the default is 3 knots.

Knotted splines have the following properties in contrast to smoothing splines:

- Knotted splines work inside of general models with many terms, where smoothing splines are for bivariate regressions.
- The regression basis is not a function of the response.
- Knotted splines are parsimonious, adding only  $k - 2$  terms for curvature for  $k$  knot points.
- Knotted splines are conservative compared to pure polynomials in the sense that the extrapolation outside the range of the data is a straight line, rather than a (curvy) polynomial.
- There is an easy test for curvature.

To test for curvature, select **Estimates > Custom Test** and add a column for each knotted effect, as follows:

1. Open the Growth.jmp sample data table.
2. Select **Analyze > Fit Model**.

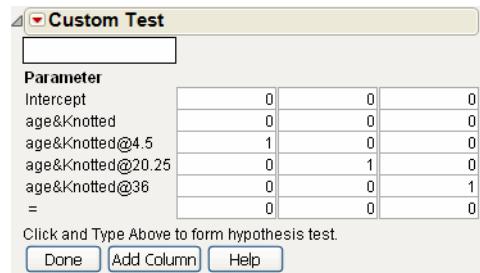
3. Assign ratio to **Y** and add **age** as an effect to the model.
4. Select the **age** variable in the Construct Model Effects box and select **Attributes > Knotted Spline Effect**.
5. Specify the number of knots as 5, and click **OK**.
6. Click **Run**.

When the report appears:

7. Select **Estimates > Custom Test** and notice that there is only one column. Therefore, click the **Add Column** button twice to produce a total of three columns.
8. Fill in the three knotted columns with ones so that they match Figure 1.3.

---

**Figure 1.3** Construction of Custom Test for Curvature



The screenshot shows the 'Custom Test' dialog box. At the top, there is a title bar with the text 'Custom Test'. Below the title bar is a large input field containing a 3x3 matrix of numbers. The matrix is defined by the following parameter names:

Parameter	Intercept	age&Knotted	age&Knotted@4.5	age&Knotted@20.25	age&Knotted@36	=
Intercept	0	0	0	0	0	0
age&Knotted	0	0	0	0	0	0
age&Knotted@4.5	1	0	0	0	0	0
age&Knotted@20.25	0	1	0	0	0	0
age&Knotted@36	0	0	0	1	0	0
=	0	0	0	0	0	0

Below the matrix, there is a note: 'Click and Type Above to form hypothesis test.' At the bottom of the dialog box are three buttons: 'Done', 'Add Column', and 'Help'.

- 
9. Click **Done**.

This produces the report shown in Figure 1.4. The low Prob > F value indicates that there is indeed curvature to the data.

**Figure 1.4** Curvature Report

Custom Test			
Parameter	0	0	0
Intercept	0	0	0
age&Knotted	0	0	0
age&Knotted@4.5	1	0	0
age&Knotted@20.25	0	1	0
age&Knotted@36	0	0	1
=	0	0	0
Value	-1.436874e-5	0.0000365148	-0.000033071
Std Error	1.9803969e-6	6.0647971e-6	8.0723366e-6
t Ratio	-7.255483783	6.020782207	-4.096860562
Prob> t	5.275629e-10	8.1625874e-8	0.0001152566
SS	0.0583271224	0.0401846174	0.0185968833
Sum of Squares	0.112524874		
Numerator DF	3		
F Ratio	33.852401355		
Prob > F	1.955866e-13		

## Transformations (Standard Least Squares Only)

The **Transformations** popup menu has eight functions available to transform selected continuous effects or Y columns for standard least squares analyses only. The available transformations are **None**, **Log**, **Sqrt**, **Square**, **Reciprocal**, **Exp**, **Arrhenius**, and **ArrheniusInv**. These transformations are only supported for single-column continuous effects.

$$\text{The Arrhenius transformation is } T^* = \frac{11605}{\text{Temp} + 273.15}$$

---

## Fitting Personalities

The Fit Model dialog in JMP serves many different fitting methods. Rather than have a separate dialog for each method, there is one dialog with a choice of *fitting personality* for each method. Usually the personality is chosen automatically from the context of the response and factors you enter, but you can change selections from the **Fitting Personality** popup menu to alternative methods.

The following list briefly describes each type of model. Details about text reports, plots, options, special commands, and example analyses are found in the individual chapters for each type of model fit:

**Standard Least Squares** JMP models one or more continuous responses in the usual way through fitting a linear model by least squares. The standard least squares report platform offers two flavors of tables and graphs:

*Traditional statistics* are for situations where the number of error degrees of freedom allows hypothesis testing. These reports include leverage plots, least squares means, contrasts, and output formulas.

*Screening and Response Surface Methodology* analyze experimental data where there are many effects but few observations. Traditional statistical approaches focus on the residual error. However, because

in near-saturated designs there is little or no room for estimating residuals, the focus is on the prediction equation and the effect sizes. Of the many effects that a screening design can fit, you expect a few important terms to stand out in comparison to the others. Another example is when the goal of the experiment is to optimize settings rather than show statistical significance; the factor combinations that optimize the predicted response are of overriding interest.

**Stepwise** Stepwise regression is an approach to selecting a subset of effects for a regression model. The **Stepwise** feature computes estimates that are the same as those of other least squares platforms, but it facilitates searching and selecting among many models. The **Stepwise** personality allows only one continuous  $Y$ .

**Manova** When there is more than one  $Y$  variable specified for a model with the **Manova** fitting personality selected, the Fit Model platform fits the  $Y$ 's to the set of specified effects and provides multivariate tests.

**LogLinear Variance** LogLinear Variance is used when one or more effects model the variance rather than the mean. The **LogLinear Variance** personality must be used with caution. See “[Fitting Dispersion Effects with the LogLinear Variance Model](#),” p. 155 for more information on this feature.

**Nominal Logistic** If the response is nominal, the Fit Model platform fits a linear model to a multilevel logistic response function using maximum likelihood.

**Ordinal Logistic** If the response variable has an ordinal modeling type, the Fit Model platform fits the cumulative response probabilities to the logistic distribution function of a linear model by using maximum likelihood.

**Proportional Hazard** The proportional hazard (Cox) model is a special fitting personality that lets you specify models where the response is time-to-failure. The data may include right-censored observations and time-independent covariates. The covariate of interest may be specified as a grouping variable that defines sub populations or strata.

**Parametric Survival** Parametric Survival performs the same analysis as the **Fit Parametric Survival** command on the **Analyze > Reliability and Survival** menu. See *Quality and Reliability Methods* for details.

**Generalized Linear Model** fits generalized linear models with various distribution and link functions. See the chapter “[Generalized Linear Models](#),” p. 197 for a complete discussion of generalized linear models.

**Table 1.3** Characteristics of Fitting Personalities

Personality	Response (Y) Type	Notes
Standard Least Squares	$\geq 1$ continuous	all effect types
Stepwise	1 continuous	all effect types
MANOVA	$> 1$ continuous	all effect types
LogLinear Variance	1 continuous	variance effects
Nominal Logistic	1 nominal	all effect types

**Table 1.3** Characteristics of Fitting Personalities (*Continued*)

Personality	Response (Y) Type	Notes
Ordinal Logistic	1 ordinal	all effect types
Proportional Hazard	1 continuous	survival models only
Parametric Survival	1 or 2 continuous	survival models only  You can specify two columns in Parametric Survival if the data values are interval-censored. One of the two columns is treated as the lower limit, and the other is treated as the upper limit.  Parametric Survival does not otherwise support a multivariate response model.
Generalized Linear Model	continuous, nominal, ordinal	all effect types

---

## Other Model Dialog Features

### Emphasis Choices

The **Emphasis** popup menu controls the type of plots you see as part of the initial analysis report.

**Effect Leverage** begins with leverage and residual plots for the whole model. You can then request effect details and other statistical reports.

**Effect Screening** shows whole model information followed by a scaled parameter report with graph and the Prediction Profiler.

**Minimal Report** suppresses all plots except the regression plot. You request what you want from the platform popup menu.

### Run

#### **Keep Dialog Open**

Checking this option keeps the Model Specification Dialog open after clicking **Run**.

## Validity Checks

**Fit Model** checks your model for errors such as duplicate effects or missing effects in a hierarchy. If you get an alert message, you can either **Continue** the fitting despite the situation, or click **Cancel** in the alert message to stop the fitting process.

In addition, your data may have missing values. The default behavior is to exclude rows if any  $Y$  or  $X$  value is missing. This can be wasteful of rows for cases when some  $Y$ 's have non-missing values and other  $Y$ 's are missing. Therefore, you may consider fitting each  $Y$  separately.

When this situation occurs, you are alerted with a window that asks: “Missing values different across  $Y$  columns. Fit each  $Y$  separately?” Fitting the  $Y$ 's separately uses all non-missing rows for that particular  $Y$ . Fitting the  $Y$ 's together uses only those rows that are non-missing for both  $Y$ 's.

When  $Y$ 's are fit separately, the results for the individual analyses are given in a **Fit Group** report. This allows all the  $Y$ 's to be profiled in the same Profiler.

---

**Note:** This dialog only appears when the model is run interactively. Scripts continue to use the default behavior, unless **Fit Separately** is placed inside the **Run Model** command.

---

## Other Model Specification Options

The popup menu on the title bar of the Model Specification window gives additional options:

**Center Polynomials** causes a continuous term participating in a crossed term to be centered by its mean. Exceptions to centering are effects with coded or mixture properties. This option is important to make main effects tests be meaningful hypotheses in the presence of continuous crossed effects.

**Set Alpha Level** is used to set the alpha level for confidence intervals in the Fit Model analysis.

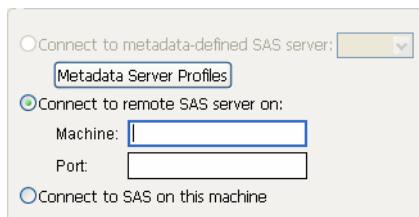
**Save to Data Table** saves the model as a property of the current data table. A **Model** popup menu icon appears in the Tables panel at the left of the data grid with a **Run Script** command. If you then select this **Run Script** command to submit the model, a new completed Fit Model dialog appears. The *JMP Scripting Guide* is the reference for JSL statements.

**Save to Script window** saves the JSL commands for the completed model in a new open script window. You can save the script window and recreate the model at any time by running the script.

**Create SAS job** creates a SAS program in a script window that can recreate the current analysis in SAS. Once created, you have several options for submitting the code to SAS.

1. Copy and Paste the resulting code into the SAS Program Editor. This method is useful if you are running an older version of SAS (pre-version 8.2).
2. Select Edit > Submit to SAS.

**Submit to SAS** brings up a dialog (shown here) that allows you to enter the machine name and port of an accessible SAS server (reached through an integrated object model, or IOM), or you can connect directly to SAS on your personal computer. The dialog allows you to enter profiles for any new SAS servers. Results are returned to JMP and are displayed in a separate log window.

**Figure 1.5** Connect to SAS Server Window

---

3. Save the file and double-click it to open it in a local copy of SAS. This method is useful if you would like to take advantage of SAS ODS options, *e.g.* generating HTML or PDF output from the SAS code.

**Load Version 3 Model** presents an Open File dialog for you to select a text file that contains JMP Version 3 model statements. The model then appears in the Fit Model dialog, and can be saved as a current-version model.

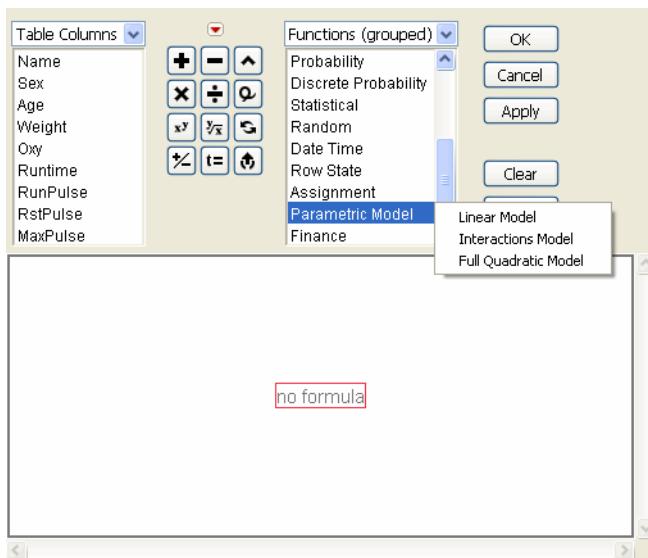
---

## Formula Editor Model Features

There are several features in the Formula Editor that are useful for constructing models.

### Parametric Models

In the Functions List, the Parametric Model group is useful in creating linear regression components. Each presents a dialog that allows you to select the columns involved in the model.

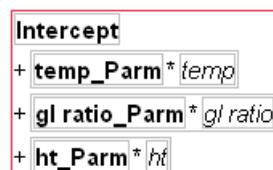
**Figure 1.6** Parametric Model Group

**Linear Model** builds a linear model, with a parameter as the coefficient as each term.

#### Examples of Parametric Models

1. Open the Odor Control Original.jmp sample data table.
2. Double-click in the empty column after odor to create a new column. Name the column Linear Model.
3. Right-click on the Linear Model column and select **Formula**.
4. Scroll down in the Functions list, and select **Parametric Model > Linear Model**.
5. Select the following columns: temp, gl ratio, and ht, and click **OK**.

Figure 1.7 shows the resulting linear model formula.

**Figure 1.7** Linear Model Formula

**Interactions Model** builds a linear model with first-order interactions:

6. Select **Parametric Model > Interactions Model**.

7. Select the following columns: temp, gl ratio, and ht, and click **OK**.

Figure 1.8 shows the resulting interactions model formula.

**Figure 1.8** Interactions Model Formula

```
Intercept
+ temp_Parm * temp
+ gl ratio_Parm * gl ratio
+ ht_Parm * ht
+ gl ratio_temp_Parm * gl ratio * temp
+ ht_temp_Parm * ht * temp
+ ht_gl ratio_Parm * ht * gl ratio
```

**Full Quadratic Model** builds a model with linear, first-order interaction, and quadratic terms.

8. Select **Parametric Model > Full Quadratic Model**.  
 9. Select the following columns: temp, gl ratio, and ht, and click **OK**.

Figure 1.9 shows the resulting full quadratic model formula.

**Figure 1.9** Full Quadratic Model Formula

```
Intercept
+ temp_Parm * temp
+ gl ratio_Parm * gl ratio
+ ht_Parm * ht
+ temp_temp_Parm * temp * temp
+ gl ratio_temp_Parm * gl ratio * temp
+ gl ratio_gl ratio_Parm * gl ratio * gl ratio
+ ht_temp_Parm * ht * temp
+ ht_gl ratio_Parm * ht * gl ratio
+ ht_ht_Parm * ht * ht
```

## Adding Parameters

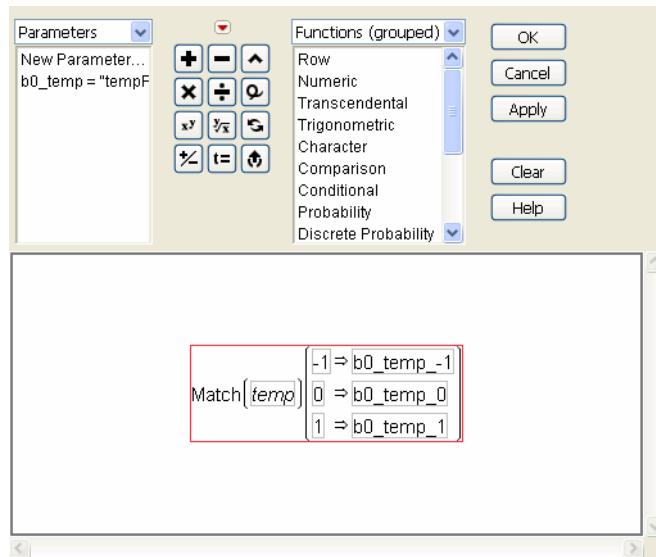
When adding a parameter to a model in the Formula Editor, the **Expand into categories** checkbox is useful for making parametric expressions across categories.

### Example of Adding a Parameter

1. Open the Odor Control Original.jmp sample data table.
2. Double-click in the empty column after **odor** to create a new column. Name the column **Category**.
3. Right-click on the **Category** column and select **Formula**.
4. Click on the arrow next to **Table Columns** and select **Parameters**.
5. Select **New Parameter**.
6. Keep the parameter name **b0**.
7. Next to value, type 0.
8. Select **Expand into categories, selecting columns** and click **OK**.
9. Select **temp** and click **OK**.
10. Click on the new parameter, **b0\_temp = 0**.

---

**Figure 1.10** New Parameter



---

This formula creates new parameters named **b0\_temp\_n** for each level of **temp**.



# Chapter 2

## Standard Least Squares: Introduction The Fit Model Platform

---

The Fit Model platform contains the Standard Least Squares personality. This fitting facility can fit many different types of models (regression, analysis of variance, analysis of covariance, and mixed models), and you can explore the fit in many ways. Even though this is a sophisticated platform, you can do simple tasks very easily.

This Standard Least Squares fitting personality is used for a continuous-response fit to a linear model of factors using least squares. The results are presented in detail, and include leverage plots and least squares means. There are many additional features to consider, such as making contrasts and saving output formulas. More detailed discussions of the Standard Least Squares fitting personality are included in other chapters.

- If you haven't learned how to specify your model in the Model Specification window, you can refer to the chapter "[Introduction to Model Fitting](#)," p. 1.
- If you have response surface effects or want retrospective power calculations, see the chapter "[Standard Least Squares: Perspectives on the Estimates](#)," p. 57.
- For screening applications, see the chapter "[Standard Least Squares: Exploring the Prediction Equation](#)," p. 87.
- If you have random effects, see the chapter "[Standard Least Squares: Random Effects](#)," p. 101.
- If you need the details on how JMP parameterizes its models, see the appendix "[Statistical Details](#)," p. 607.

The Standard Least Squares Personality is just one of many fitting personalities of the Fit Model platform. The other eight personalities are covered in the later chapters "[Standard Least Squares: Perspectives on the Estimates](#)," p. 57, "[Standard Least Squares: Exploring the Prediction Equation](#)," p. 87, "[Standard Least Squares: Random Effects](#)," p. 101, "[Generalized Linear Models](#)," p. 197, "[Fitting Dispersion Effects with the LogLinear Variance Model](#)," p. 155, "[Stepwise Regression](#)," p. 117, and "[Logistic Regression for Nominal and Ordinal Response](#)," p. 165, and "[Multiple Response Fitting](#)," p. 135.

# Contents

Launch the Platform: A Simple Example.....	23
Regression Plot .....	26
Option Packages for Emphasis.....	27
Whole-Model Statistical Tables .....	28
The Summary of Fit Table.....	28
The Analysis of Variance Table .....	29
The Lack of Fit Table .....	31
The Parameter Estimates Table .....	33
The Effect Test Table.....	34
Saturated Models.....	35
Leverage Plots .....	36
Effect Details.....	40
LSMeans Table .....	40
LSMeans Plot .....	41
LSMeans Contrast.....	43
LSMeans Student's t, LSMeans Tukey's HSD .....	45
Test Slices .....	46
Power Analysis.....	46
Summary of Row Diagnostics and Save Commands .....	46
Row Diagnostics .....	47
Save Columns Command .....	48
Examples with Statistical Details .....	50
One-Way Analysis of Variance with Contrasts.....	50
Analysis of Covariance.....	52
Analysis of Covariance with Separate Slopes .....	54
Singularity Details.....	55

## Launch the Platform: A Simple Example

To introduce the Fit Model platform, consider a simple one-way analysis of variance to test if there is a difference in the mean response among three drugs. The example data (Snedecor and Cochran 1967) is a study that measured the response of 30 subjects after treatment by each of three drugs labeled a, d, and f. The results are in the Drug.jmp sample data table.

1. Open the Drug.jmp sample data table.
2. Select Analyze > Fit Model.

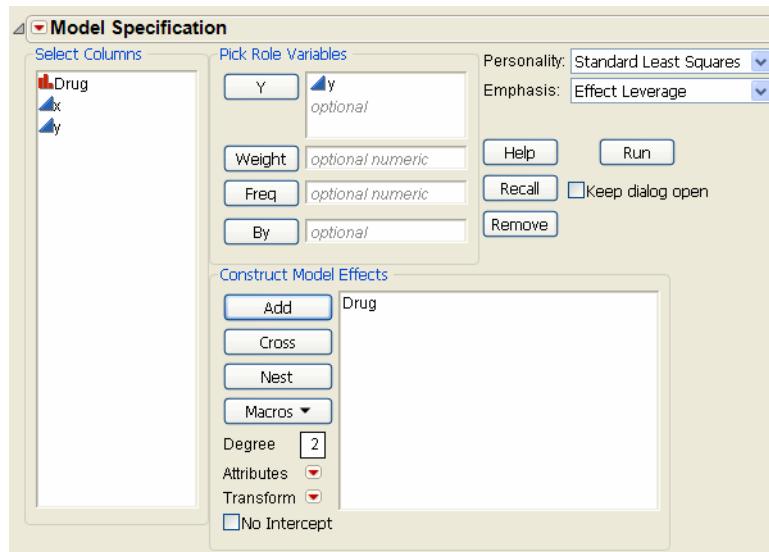
See the chapter “[Introduction to Model Fitting](#),” p. 1, for details about how to use this window.

3. Select y and click Y.

When you select the column y to be the *Y* response, the Fitting Personality becomes **Standard Least Squares** and the Emphasis is **Effect Leverage**. You can change these options in other situations.

4. Select Drug and click Add.

**Figure 2.1** The Fit Model Window For a One-Way Analysis of Variance



5. Click Run.

At the top of the output are the graphs in Figure 2.2 that show how the data fit the model. These graphs are called *leverage plots*, because they convey the idea of the data points pulling on the lines representing the fitted model. Leverage plots have these useful properties:

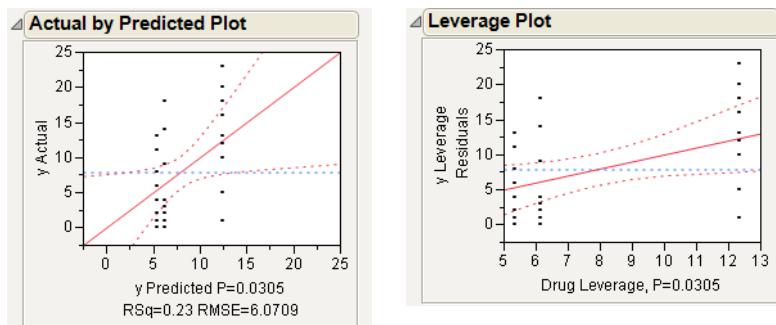
- The distance from each point to the line of fit is the error or residual for that point.

## Launch the Platform: A Simple Example

- The distance from each point to the horizontal line is what the error would be if you took out effects in the model.

Thus, strength of the effect is shown by how strongly the line of fit is suspended away from the horizontal by the points. Confidence curves are on the graph so you can see at a glance whether an effect is significant. In each plot, if the 95% confidence curves cross the horizontal reference line, then the effect is significant. If the curves do not cross, then it is not significant (at the 5% level).

**Figure 2.2** Whole Model Leverage Plot and Drug Effect Leverage Plot



In this simple case where predicted values are simple means, the leverage plot for Drug shows a regression of the actual values on the means for the drug level. Levin, Serlin, and Webne-Behrman (1989) showcase this idea.

Because there is only one effect in the model, the leverage plot for Whole Model and for the effect Drug are equivalent. They differ only in the scaling of the  $x$ -axis. The leverage plot for the Whole Model is a plot of the actual response versus the predicted response. So, the points that fall on the line are those that are perfectly predicted.

In this example, the confidence curve does cross the horizontal line. Thus, the drug effect is marginally significant, even though there is considerable variation around the line of fit (in this case around the group means).

After you examine the fit graphically, you can look below it for textual details. In this case there are three levels of Drug, giving two parameters to characterize the differences among them. Text reports show the estimates and various test statistics and summary statistics concerning the parameter estimates. The Summary of Fit table and the Analysis of Variance table beneath the whole-model leverage plot show the fit as a whole. Because Drug is the only effect, the same test statistic appears in the Analysis of Variance table and in the Effects Tests table. Results for the Summary of Fit and the Analysis of Variance are shown in Figure 2.3.

**Figure 2.3** Summary of Fit and Analysis of Variance

Summary of Fit				
RSquare		0.227826		
RSquare Adj		0.170628		
Root Mean Square Error		6.070878		
Mean of Response		7.9		
Observations (or Sum Wgts)		30		

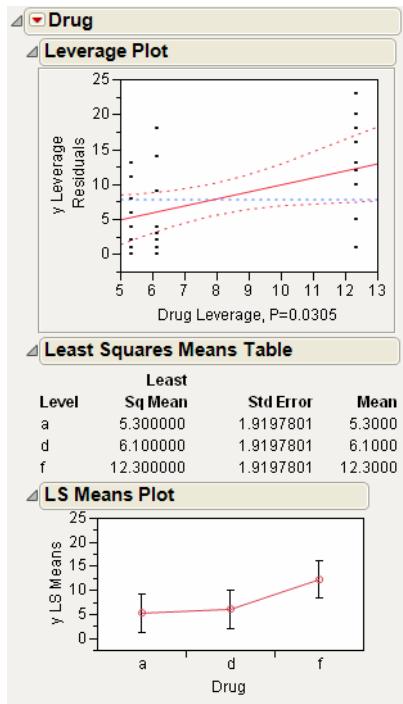
  

Analysis of Variance				
Source	DF	Sum of		
		Squares	Mean Square	F Ratio
Model	2	293.6000	146.800	3.9831
Error	27	995.1000	36.856	Prob > F
C. Total	29	1,288.7000		0.0305*

The Analysis of Variance table shows that the Drug effect has an observed significance probability (Prob > F) of 0.0305, which is significant at a 0.05 level. The RSquare value means that 22.8% of the variation in the response can be absorbed by fitting this model.

Whenever there are nominal effects (such as Drug), it is interesting to compare how well the levels predict the response. Rather than use the parameter estimates directly, it is usually more meaningful to compare the predicted values at the levels of the nominal values. These predicted values are called the *least squares means* (LSMeans), and in this simple case, they are the same as the ordinary means. Least squares means can differ from simple means when there are other effects in the model, as seen in later examples. The Least Squares Means table for a nominal effect shows beneath the effect's leverage plot (Figure 2.4).

The popup menu for an effect also lets you request the **LSMeans Plot**, as shown in Figure 2.4. The plot graphically shows the means and their associated 95% confidence interval.

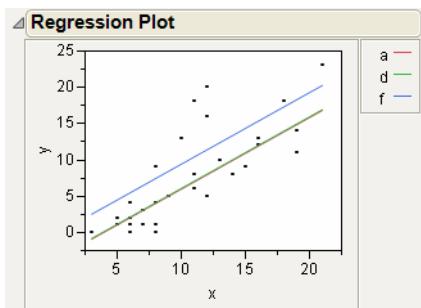
**Figure 2.4** Table of Least Squares Means and LS Means Plot

The later section “[Examples with Statistical Details](#),” p. 50, continues the drug analysis with the Parameter Estimates table and looks at the group means with the **LSMeans Contrast**.

## Regression Plot

If there is exactly one continuous term in a model, and no more than one categorical term, then JMP plots the regression line (or lines). The regression plot for the drug data is shown in Figure 2.5 and is done as follows:

- Specify the model as in “[Launch the Platform: A Simple Example](#),” p. 23 but this time, add x with Drug as a model effect.
- The Regression Plot appears by default. You can elect to not show this plot in the output by selecting **Response y** in the title bar > **Row Diagnostics** > **Plot Regression**. Selecting this command turns the plot off and on.

**Figure 2.5** Regression Plot

## Option Packages for Emphasis

The model fitting process can produce a wide variety of tables, plots, and graphs. For convenience, three standard option packages let you choose the report layout that best corresponds to your needs. Just as automobiles are made available with luxury, sport, and economy option packages, linear model results are made available in choices to adapt to your situation. [Table 2.1 “Standard Least Squares Report Layout Defined by Emphasis,” p. 27](#), describes the types of report layout for the standard least squares analysis.

The chapter [“Introduction to Model Fitting,” p. 1](#), introduces the **Emphasis** menu. The default value of **Emphasis** is based on the number of effects that you specify and the number of rows (observations) in the data table. The Emphasis menu options are summarized in Table 2.1.

**Table 2.1** Standard Least Squares Report Layout Defined by Emphasis

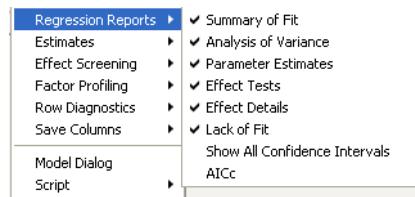
Emphasis	Description of Reports
Effect Leverage	Choose <b>Effect Leverage</b> when you want details on the significance of each effect. The initial reports for this emphasis features leverage plots, and the reports for each effect are arranged horizontally.
Effect Screening	The <b>Effect Screening</b> layout is better when you have many effects, and don't want details until you see which effects are stronger and which are weaker. This is the recommended method for screening designs, where there are many effects and few observations, and the quest is to find the strong effects, rather than to test significance. This arrangement initially displays whole model information followed by effect details, scaled estimates, and the prediction profiler.
Minimal Report	<b>Minimal Report</b> starts simple; you customize the results with the tables and plots you want to see. Choosing <b>Minimal Report</b> suppresses all plots (except the regression plot), and arranges whole model and effect detail tables vertically.

## Whole-Model Statistical Tables

This section starts the details on the standard statistics inside the reports. You might want to skim some of these details sections and come back to them later.

Regression reports can be turned on and off with the Regression Reports menu. This menu, shown in Figure 2.6, is accessible from the report's red triangle menu.

**Figure 2.6** Regression Report Options



The Whole Model section shows how the model fits as a whole. The next sections describe these tables:

- Summary of Fit
- Analysis of Variance
- Lack of Fit
- Parameter Estimates
- Effect Tests

Discussion of tables for effects and leverage plots are described under the sections “[The Effect Test Table](#),” p. 34, and “[Leverage Plots](#),” p. 36.

The **Show All Confidence Intervals** command shows confidence intervals for parameter estimates in the Parameter Estimates report, and also for least squares means in Least Squares Means Tables. The **AICc** command shows AICc and BIC in the Summary of Fit report.

The following examples use the Drug.jmp data table from the Sample Data folder. The model specifies y as the response variable (*Y*) and Drug and *x* as effects, where Drug has the nominal modeling type.

## The Summary of Fit Table

The Summary of Fit table appears first. The numeric summaries of the response for the multiple regression model are shown in Figure 2.7.

**Figure 2.7** Summary of Fit

Summary of Fit	
RSquare	0.676261
RSquare Adj	0.638906
Root Mean Square Error	4.005778
Mean of Response	7.9
Observations (or Sum Wgts)	30

The Summary of Fit for the response includes:

**RSquare** estimates the proportion of the variation in the response around the mean that can be attributed to terms in the model rather than to random error. Using quantities from the corresponding Analysis of Variance table,  $R^2$  is calculated:

$$\frac{\text{Sum of Squares(Model)}}{\text{Sum of Squares(C. Total)}}$$

It is also the square of the correlation between the actual and predicted response. An  $R^2$  of 1 occurs when there is a perfect fit (the errors are all zero). An  $R^2$  of 0 means that the fit predicts the response no better than the overall response mean.

**Rsquare Adj** adjusts  $R^2$  to make it more comparable over models with different numbers of parameters by using the degrees of freedom in its computation. It is a ratio of mean squares instead of sums of squares and is calculated

$$1 - \frac{\text{Mean Square(Error)}}{\text{Mean Square(C. Total)}}$$

where mean square for Error is found in the Analysis of Variance table (shown next under “[The Analysis of Variance Table](#),” p. 29) and the mean square for **C. Total** can be computed as the **C. Total** sum of squares divided by its respective degrees of freedom.

**Root Mean Square Error** estimates the standard deviation of the random error. It is the square root of the mean square for error in the corresponding Analysis of Variance table, and it is commonly denoted as  $s$ .

**Mean of Response** is the overall mean of the response values. It is important as a base model for prediction because all other models are compared to it. The variance measured around this mean is the *Sum of Squares Corrected Total (C. Total)* in the Analysis of Variance table.

**Observations (or Sum of Weights)** records the number of observations used in the fit. If there are no missing values and no excluded rows, this is the same as the number of rows in the data table. If there is a column assigned to the role of weight, this is the sum of the weight column values.

## The Analysis of Variance Table

The Analysis of Variance table is displayed in Figure 2.8 and shows the basic calculations for a linear model.

**Figure 2.8** Analysis of Variance Table

Analysis of Variance				
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	3	871.4974	290.499	18.1039
Error	26	417.2026	16.046	Prob > F
C. Total	29	1,288.7000		<.0001*

The table compares the model fit to a simple fit of a single mean:

**Source** lists the three sources of variation, called **Model**, **Error**, and **C. Total** (Corrected Total).

**DF** records an associated *degrees of freedom* (DF) for each source of variation.

The **C. Total** degrees of freedom is shown for the simple mean model. There is only one degree of freedom used (the estimate of the mean parameter) in the calculation of variation so the **C. Total** DF is always one less than the number of observations.

The total degrees of freedom are partitioned into the **Model** and **Error** terms:

The **Model** degrees of freedom shown is the number of parameters (except for the intercept) used to fit the model.

The **Error** DF is the difference between the **C. Total** DF and the **Model** DF.

**Sum of Squares** records an associated sum of squares (SS for short) for each source of variation. The SS column accounts for the variability measured in the response. It is the sum of squares of the differences between the fitted response and the actual response.

The total (**C. Total**) SS is the sum of squared distances of each response from the sample mean, which is 1288.7 in the example shown at the beginning of this section. That is the base model (or simple mean model) used for comparison with all other models.

The **Error** SS is the sum of squared differences between the fitted values and the actual values, and is 417.2 in the previous example. This sum of squares corresponds to the unexplained **Error** (*residual*) after fitting the regression model.

The total SS less the error SS gives the sum of squares attributed to the **Model**.

One common set of notations for these is SSR, SSE, and SST for sum of squares due to regression (model), error, and total, respectively.

**Mean Square** is a sum of squares divided by its associated degrees of freedom. This computation converts the sum of squares to an average (mean square).

The **Model** mean square is 290.5 in the table at the beginning of this section.

The **Error** Mean Square is 16.046 and estimates the variance of the error term. It is often denoted as MSE or  $s^2$ .

**F Ratio** is the model mean square divided by the error mean square. It tests the hypothesis that all the regression parameters (except the intercept) are zero. Under this whole-model hypothesis, the two mean squares have the same expectation. If the random errors are normal, then under this hypothesis the values reported in the SS column are two independent Chi-squares. The ratio of these two Chi-squares divided by their respective degrees of freedom (reported in the DF column) has an

*F*-distribution. If there is a significant effect in the model, the **F Ratio** is higher than expected by chance alone.

**Prob > F** is the probability of obtaining a greater *F*-value by chance alone if the specified model fits no better than the overall response mean. Significance probabilities of 0.05 or less are often considered evidence that there is at least one significant regression factor in the model.

Note that large values of **Model SS**, relative to small values of **Error SS**, lead to large *F*-ratios and low *p* values. (This is desired if the goal is to declare that terms in the model are significantly different from zero.) Most practitioners check this *F*-test first and make sure that it is significant before delving further into the details of the fit. This significance is also shown graphically by the whole-model leverage plot described in the previous section.

## The Lack of Fit Table

The Lack of Fit table in Figure 2.9 shows a special diagnostic test and appears only when the data and the model provide the opportunity.

**Figure 2.9** Lack of Fit Table

Lack Of Fit				
Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	18	254.86926	14.1594	0.6978
Pure Error	8	162.33333	20.2917	0.7507
Total Error	26	417.20260		Max RSq 0.8740

The idea is that sometimes you can estimate the error variance independently of whether you have the right form of the model. This occurs when observations are exact replicates of each other in terms of the *X* variables. The error that you can measure for these exact replicates is called *pure error*. This is the portion of the sample error that cannot be explained or predicted by the form that the model uses for the *X* variables. However, a lack of fit test is not very useful if it has only a few degrees of freedom (not many replicated *x* values).

The difference between the residual error from the model and the pure error is called *lack of fit error*. A lack of fit error can be significantly greater than pure error if you have the wrong functional form of a regressor, or if you do not have enough interaction effects in an analysis of variance model. In that case, you should consider adding interaction terms, if appropriate, or try to better capture the functional form of a regressor.

There are two common situations where there is no lack of fit test:

1. There are no exactly replicated points with respect to the *X* data, and therefore there are no degrees of freedom for pure error.
2. The model is *saturated*, meaning that the model itself has a degree of freedom for each different *x* value. Therefore, there are no degrees of freedom for lack of fit.

The Lack of Fit table shows information about the error terms:

**Source** lists the three sources of variation called Lack of Fit, Pure Error, and Total Error.

**DF** records an associated *degrees of freedom* (DF) for each source of error.

The Total Error DF is the degrees of freedom found on the Error line of the Analysis of Variance table. It is the difference between the Total DF and the Model DF found in that table. The Error DF is partitioned into degrees of freedom for lack of fit and for pure error.

The Pure Error DF is pooled from each group where there are multiple rows with the same values for each effect. For example, in the sample data, Big Class.jmp, there is one instance where two subjects have the same values of age and weight (Chris and Alfred are both 14 and have a weight of 99). This gives  $1(2 - 1) = 1$  DF for Pure Error. In general, if there are  $g$  groups having multiple rows with identical values for each effect, the pooled DF, denoted  $DF_p$ , is

$$DF_p = \sum_{i=1}^g (n_i - 1)$$

where  $n_i$  is the number of replicates in the  $i$ th group.

The Lack of Fit DF is the difference between the Total Error and Pure Error degrees of freedom.

**Sum of Squares** records an associated sum of squares (SS for short) for each source of error.

The Total Error SS is the sum of squares found on the Error line of the corresponding Analysis of Variance table.

The Pure Error SS is pooled from each group where there are multiple rows with the same values for each effect. This estimates the portion of the true random error that is not explained by model effects. In general, if there are  $g$  groups having multiple rows with like values for each effect, the pooled SS, denoted  $SS_p$ , is written

$$SS_p = \sum_{i=1}^g SS_i$$

where  $SS_i$  is the sum of squares for the  $i$ th group corrected for its mean.

The Lack of Fit SS is the difference between the Total Error and Pure Error sum of squares. If the lack of fit SS is large, it is possible that the model is not appropriate for the data. The  $F$ -ratio described below tests whether the variation due to lack of fit is small enough to be accepted as a negligible portion of the pure error.

**Mean Square** is a sum of squares divided by its associated degrees of freedom. This computation converts the sum of squares to an average (mean square).  $F$ -ratios for statistical tests are the ratios of mean squares.

**F Ratio** is the ratio of mean square for Lack of Fit to mean square for Pure Error. It tests the hypothesis that the lack of fit error is zero.

**Prob > F** is the probability of obtaining a greater  $F$ -value by chance alone if the variation due to lack of fit variance and the pure error variance are the same. This means that an insignificant proportion of error is explained by lack of fit.

**Max RSq** is the maximum  $R^2$  that can be achieved by a model using only the variables in the model. Because Pure Error is invariant to the form of the model and is the minimum possible variance, Max RSq is calculated

$$1 - \frac{\text{SS(Pure Error)}}{\text{SS(Total for whole model)}}$$

## The Parameter Estimates Table

The Parameter Estimates table shows the estimates of the parameters in the linear model and a *t*-test for the hypothesis that each parameter is zero. Simple continuous regressors have only one parameter. Models with complex classification effects have a parameter for each anticipated degree of freedom. The Parameters Estimates table is shown in Figure 2.10.

**Figure 2.10** Parameter Estimates Table

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	-2.695773	1.911085	-1.41	0.1702
Drug[a]	-1.195037	1.060822	-1.12	0.2742
Drug[d]	-1.076065	1.041298	-1.03	0.3109
x	0.9871838	0.164498	6.00	<.0001*

The Parameter Estimates table shows these quantities:

**Term** names the estimated parameter. The first parameter is always the intercept. Simple regressors show as the name of the data table column. Regressors that are *dummy* indicator variables constructed from nominal or ordinal effects are labeled with the names of the levels in brackets. For nominal variables, the dummy variables are coded as 1 except for the last level, which is coded as -1 across all the other dummy variables for that effect. The parameters for ordinally coded indicators measure the difference from each level to the level before it. See “[Interpretation of Parameters](#),” p. 613 in the “Statistical Details” chapter, for additional information.

**Estimate** lists the parameter estimates for each term. They are the coefficients of the linear model found by least squares.

**Std Error** is the standard error, an estimate of the standard deviation of the distribution of the parameter estimate. It is used to construct *t*-tests and confidence intervals for the parameter.

**t Ratio** is a statistic that tests whether the true parameter is zero. It is the ratio of the estimate to its standard error and has a Student’s *t*-distribution under the hypothesis, given the usual assumptions about the model.

**Prob > |t|** is the probability of getting an even greater *t*-statistic (in absolute value), given the hypothesis that the parameter is zero. This is the two-tailed test against the alternatives in each direction. Probabilities less than 0.05 are often considered as significant evidence that the parameter is not zero.

Although initially hidden, the following columns are also available. Right-click (control-click on the Macintosh) and select the desired column from the **Columns** menu. Alternatively, there is a preference that can be set to always show the columns.

**Std Beta** are the parameter estimates that would have resulted from the regression had all the variables been standardized to a mean of 0 and a variance of 1.

**VIF** shows the variance inflation factors. High VIFs indicate a collinearity problem.

Note that the VIF is defined as

$$VIF = \frac{1}{1 - R_i^2}$$

where  $R_i^2$  is the coefficient of multiple determination for the regression of  $x_i$  as a function of the other explanatory variables.

**Lower 95%** is the lower 95% confidence interval for the parameter estimate.

**Upper 95%** is the upper 95% confidence interval for the parameter estimate.

**Design Std Error** is the standard error without being scaled by sigma (RMSE), and is equal to

$$\sqrt{\text{diag}(\mathbf{X}'\mathbf{X})^{-1}}$$

## The Effect Test Table

The effect tests are joint tests that all the parameters for an individual effect are zero. If an effect has only one parameter, as with simple regressors, then the tests are no different from the *t*-tests in the Parameter Estimates table. Parameterization and handling of singularities are different from the SAS GLM procedure. For details, see the appendix “[Statistical Details](#),” p. 607. The Effect Tests table is shown in Figure 2.11.

**Figure 2.11** Effect Tests Table

Effect Tests					
Source	Nparm	DF	Sum of		
			Squares	F Ratio	Prob > F
Drug	2	2	68.55371	2.1381	0.1384
x	1	1	577.89740	36.0145	<.0001*

The Effect Tests table shows the following information for each effect:

**Source** lists the names of the effects in the model.

**Nparm** is the number of parameters associated with the effect. Continuous effects have one parameter. Nominal effects have one less parameter than the number of levels. Crossed effects multiply the number of parameters for each term. Nested effects depend on how levels occur.

**DF** is the degrees of freedom for the effect test. Ordinarily **Nparm** and **DF** are the same. They are different if there are linear combinations found among the regressors such that an effect cannot be tested to its fullest extent. Sometimes the DF is zero, indicating that no part of the effect is testable. Whenever DF is less than Nparm, the note **Lost DFs** appears to the right of the line in the report.

**Sum of Squares** is the sum of squares for the hypothesis that the listed effect is zero.

**F Ratio** is the *F*-statistic for testing that the effect is zero. It is the ratio of the mean square for the effect divided by the mean square for error. The mean square for the effect is the sum of squares for the effect divided by its degrees of freedom.

**Prob > F** is the significance probability for the  $F$ -ratio. It is the probability that if the null hypothesis is true, a larger  $F$ -statistic would occur only due to random error. Values less than 0.0005 appear as <.0001, which is conceptually zero.

Although initially hidden, a column that displays the Mean Square is also available. Right-click (control-click on the Macintosh) and select **Mean Square** from the **Columns** menu.

---

## Saturated Models

Screening experiments often involve fully saturated models, where there are not enough degrees of freedom to estimate error. Because of this, neither standard errors for the estimates, nor  $t$ -ratios, nor  $p$ -values can be calculated in the traditional way.

For these cases, JMP uses the relative standard error, corresponding to a residual standard error of 1. In cases where all the variables are identically coded (say,  $[-1,1]$  for low and high levels), these relative standard errors are identical.

JMP also displays a Pseudo- $t$ -ratio, calculated as

$$\text{Pseudo } t = \frac{\text{estimate}}{\text{relative std error} \times PSE}$$

using Lenth's PSE (pseudo standard-error) and degrees of freedom for error (DFE) equal to one-third the number of parameters. The value for Lenth's PSE is shown at the bottom of the report.

### Example of a Saturated Model

1. Open the Reactor.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select Y and click Y.
4. Select the following five columns: F, Ct, A, T and Cn.
5. Click on the **Macros** button and select **Full Factorial**.
6. Click **Run**.

The parameter estimates are presented in sorted order, with smallest  $p$ -values listed first. The sorted parameter estimates are presented in Figure 2.12.

**Figure 2.12** Saturated Report

Sorted Parameter Estimates			
Term	Estimate	Relative Std Error	t-Ratio
Ct	9.75	0.176777	14.86
Ct*T	6.625	0.176777	10.10
T*Cn	-5.5	0.176777	-8.38
T	5.375	0.176777	8.19
Cn	-3.125	0.176777	-4.76
F*A*Cn	-1.25	0.176777	-1.90
A*T	1.0625	0.176777	1.62
Ct*Cn	1	0.176777	1.52
F*Ct*Cn	-0.9375	0.176777	-1.43
F*Ct*A	0.75	0.176777	1.14
F*Ct*A*Cn	0.75	0.176777	1.14
F	-0.6875	0.176777	-1.05
F*Ct	0.6875	0.176777	1.05
F*Ct*T	0.6875	0.176777	1.05
Ct*A*T	0.5625	0.176777	0.86
F*A*T*Cn	0.5	0.176777	0.76
F*T	-0.4375	0.176777	-0.67
Ct*A	0.4375	0.176777	0.67
A*Cn	0.4375	0.176777	0.67
F*A	0.375	0.176777	0.57
F*A*T	-0.375	0.176777	-0.57
A	-0.3125	0.176777	-0.48
F*T*Cn	0.3125	0.176777	0.48
F*Ct*T*Cn	0.3125	0.176777	0.48
Ct*A*T*Cn	-0.3125	0.176777	-0.48
F*Ct*A*T*Cn	-0.25	0.176777	-0.38
Ct*T*Cn	-0.125	0.176777	-0.19
F*Cn	0.0625	0.176777	0.10
Ct*A*Cn	0.0625	0.176777	0.10
A*T*Cn	0.0625	0.176777	0.10
F*Ct*A*T	0	0.176777	0.00

No error degrees of freedom, so ordinary tests uncomputable.  
 Relative Std Error corresponds to residual standard error of 1.  
 Pseudo t-Ratio and p-Value calculated using Lenth PSE = 0.65625  
 and DFE=10.333

In cases where the relative standard errors are different (perhaps due to unequal scaling), a similar report appears. However, there is a different value for Lenth's PSE for each estimate.

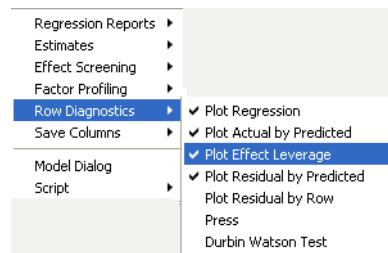
## Leverage Plots

To graphically view the significance of the model or focus attention on whether an effect is significant, you want to display the data by focusing on the hypothesis for that effect. You might say that you want more of an X-ray picture showing the inside of the data rather than a surface view from the outside. The leverage plot gives this view of your data; it offers maximum insight into how the fit carries the data.

The effect in a model is tested for significance by comparing the sum of squared residuals to the sum of squared residuals of the model with that effect removed. Residual errors that are much smaller when the effect is included in the model confirm that the effect is a significant contribution to the fit.

The graphical display of an effect's significance test is called a *leverage plot*. See Sall (1990). This type of plot shows for each point what the residual would be both with and without that effect in the model. Leverage plots are found in the **Row Diagnostics** submenu, shown in Figure 2.13 of the Fit Model report.

**Figure 2.13** Row Diagnostics Submenu

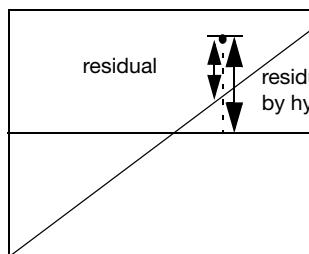


A leverage plot is constructed as illustrated in Figure 2.14. The distance from a point to the line of fit shows the actual residual. The distance from the point to the horizontal line of the mean shows what the residual error would be without the effect in the model. In other words, the mean line in this leverage plot represents the model where the hypothesized value of the parameter (effect) is constrained to zero.

Historically, leverage plots are referred to as a *partial-regression residual leverage plot* by Belsley, Kuh, and Welsch (1980) or an *added variable plot* by Cook and Weisberg (1982).

The term *leverage* is used because a point exerts more influence on the fit if it is farther away from the middle of the plot in the horizontal direction. At the extremes, the differences of the residuals before and after being constrained by the hypothesis are greater and contribute a larger part of the sums of squares for that effect's hypothesis test.

The fitting platform produces a leverage plot for each effect in the model. In addition, there is one special leverage plot titled Whole Model that shows the actual values of the response plotted against the predicted values. This Whole Model leverage plot dramatizes the test that all the parameters (except intercepts) in the model are zero. The same test is reported in the Analysis of Variance report.

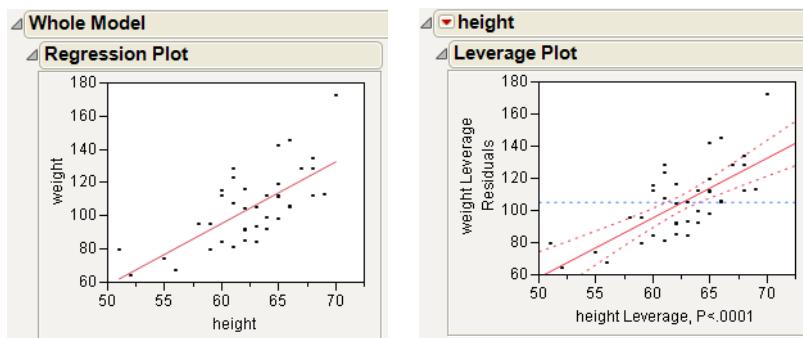
**Figure 2.14** Illustration of a General Leverage Plot

points farther out pull on the line of fit with greater leverage than the points near the middle

The leverage plot for the linear effect in a simple regression is the same as the traditional plot of actual response values and the regressor.

#### Example of a Leverage Plot for a Linear Effect

1. Open the Big Class.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select weight and click **Y**.
4. Select height and click **Add**.
5. Click **Run**.

**Figure 2.15** Whole Model and Effect Leverage Plots

The plot on the left is the Whole Model test for all effects, and the plot on the right is the leverage plot for the effect height.

The points on a leverage plot for simple regression are actual data coordinates, and the horizontal line for the constrained model is the sample mean of the response. But when the leverage plot is for one of multiple effects, the points are no longer actual data values. The horizontal line then represents a partially constrained model instead of a model fully constrained to one mean value. However, the intuitive interpretation of the plot is the same whether for simple or multiple regression. The idea is to judge if the line of fit on the effect's leverage plot carries the points significantly better than does the horizontal line.

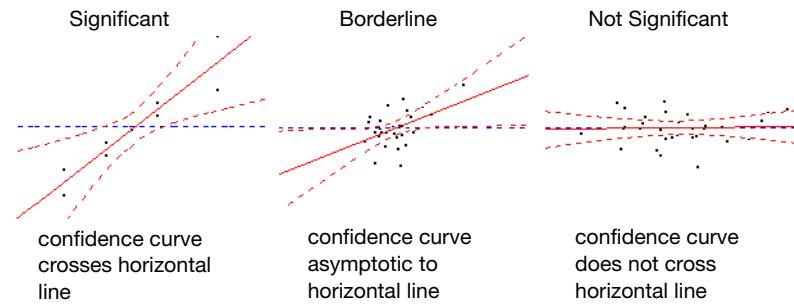
Figure 2.14 is a general diagram of the plots in Figure 2.15. Recall that the distance from a point to the line of fit is the actual residual and that the distance from the point to the mean is the residual error if the regressor is removed from the model.

### **Confidence Curves**

The leverage plots are shown with confidence curves. These indicate whether the test is significant at the 5% level by showing a confidence region for the line of fit. If the confidence region between the curves contains the horizontal line, then the effect is not significant. If the curves cross the line, the effect is significant. Compare the examples shown in Figure 2.16.

---

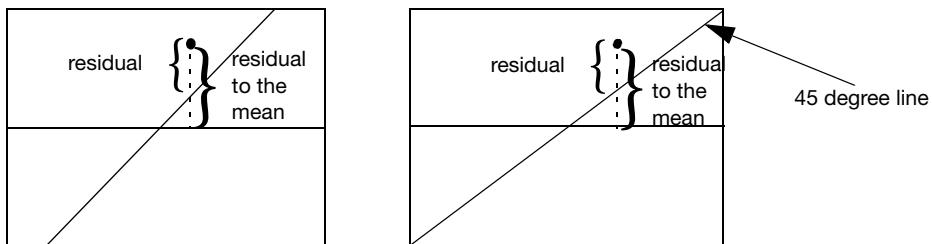
**Figure 2.16** Comparison of Significance Shown in Leverage Plots



### **Interpretation of X Scales**

If the modeling type of the regressor is continuous, then the  $x$ -axis is scaled like the regressor and the slope of the line of fit in the leverage plot is the parameter estimate for the regressor. (See the left illustration in Figure 2.17.)

If the effect is nominal or ordinal, or if a complex effect like an interaction is present instead of a simple regressor, then the  $x$ -axis cannot represent the values of the effect directly. In this case the  $x$ -axis is scaled like the  $y$ -axis, and the line of fit is a diagonal with a slope of 1. The whole model leverage plot is a version of this. The  $x$ -axis turns out to be the predicted response of the whole model, as illustrated by the right-hand plot in Figure 2.17.

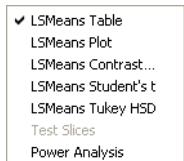
**Figure 2.17** Leverage Plots for Simple Regression and Complex Effects

The influential points in all leverage plots are the ones far out on the  $x$ -axis. If two effects in a model are closely related, then these effects as a whole don't have much leverage. This problem is called *collinearity*. By scaling regressor axes by their original values, collinearity is shown by shrinkage of the points in the  $x$  direction.

See the appendix “[Statistical Details](#),” p. 607, for the details of leverage plot construction.

## Effect Details

Each effect has the popup menu shown Figure 2.18 next to its name. The effect popup menu items let you request tables, plots, and tests for that effect. The commands for an effect append results to the effect report. You can close results or dismiss the results by deselecting the item in the menu.

**Figure 2.18** Effect Submenu

The next sections describe the Effect popup menu commands.

## LSMeans Table

Least squares means are predicted values from the specified model across the levels of a categorical effect where the other model factors are controlled by being set to *neutral* values. The neutral values are the sample means (possibly weighted) for regressors with interval values, and the average coefficient over the levels for unrelated nominal effects.

Least squares means are the values that let you see which levels produce higher or lower responses, holding the other variables in the model constant. Least squares means are also called *adjusted means* or *population marginal means*.

Least squares means are the statistics that are compared when effects are tested. They might not reflect typical real-world values of the response if the values of the factors do not reflect prevalent combinations of values in the real world. Least squares means are useful as comparisons in experimental situations. The Least Squares Mean Table for Big Class.jmp is shown in Figure 2.19. For details on recreating this report, see “[Example of a Leverage Plot for a Linear Effect,](#)” p. 38.

---

**Figure 2.19** Least Squares Mean Table for Big Class.jmp

Least Squares Means Table			
Level	Least Sq Mean	Std Error	Mean
12	58.125000	1.1959150	58.1250
13	60.285714	1.2784869	60.2857
14	64.166667	0.9784605	64.1667
15	64.571429	1.2784869	64.5714
16	64.333333	1.9529210	64.3333
17	66.666667	1.9529210	66.6667

---

A Least Squares Means table with standard errors is produced for all categorical effects in the model. For main effects, the Least Squares Means table also includes the sample mean. It is common for the least squares means to be closer together than the sample means. For further details on least squares means, see the appendix “[Statistical Details,](#)” p. 607.

The Least Squares Means table shows these quantities:

**Level** lists the names of each categorical level.

**Least Sq Mean** lists the least squares mean for each level of the categorical variable.

**Std Error** lists the standard error of the Least Sq Mean for each level of the categorical variable.

**Mean** lists the response sample mean for each level of the categorical variable. This is different from the least squares mean if the values of other effects in the model do not balance out across this effect.

Although initially hidden, columns that display the upper and lower 95% confidence intervals of the mean are also available. Right-click (control-click on the Macintosh) and select **Lower 95%** or **Upper 95%** from the **Columns** menu.

## LSMeans Plot

The **LSMeans Plot** option plots least squares means (LSMeans) plots for nominal and ordinal main effects and two-way interactions. The chapter “[Standard Least Squares: Exploring the Prediction Equation,](#)” p. 87, discusses the **Interaction Plots** command in the **Factor Profiling** menu, which offers interaction plots in a different format.

To see an example of the LSMeans Plot:

1. Open Popcorn.jmp from the sample data directory.

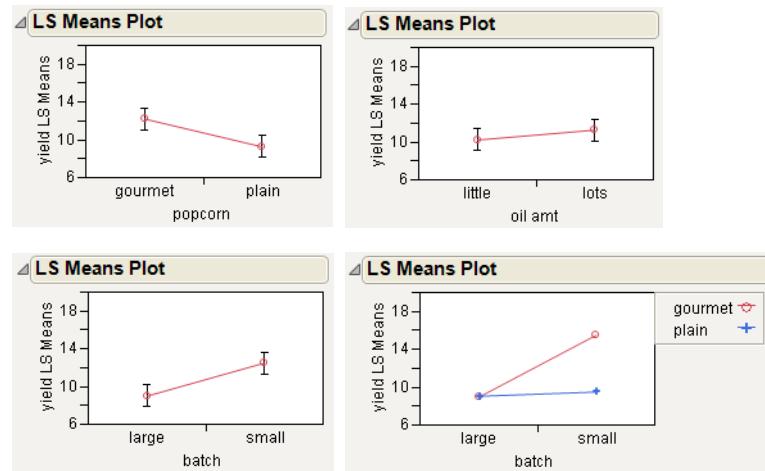
## Effect Details

2. Click on **Analyze > Fit Model**.
3. Select yield as *Y* and add popcorn, oil amt, and batch for the model effects.
4. Select popcorn in the Construct Model Effects section and batch in the Select Columns section.
5. Click on the **Cross** button to obtain the popcorn\*batch interaction.
6. Click on **Run**.
7. Select **LSMeans Plot** from the red-triangle menu for each of the effects.

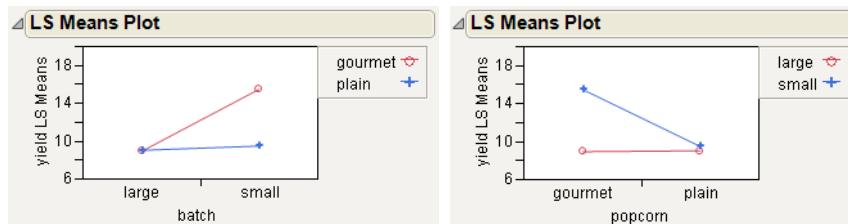
Figure 2.20 shows the effect plots for main effects and the two-way interaction. An interpretation of the data (Popcorn.jmp) is in the chapter “A Factorial Analysis” in the JMP Introductory Guide. In this experiment, popcorn yield measured by volume of popped corn from a given measure of kernels is compared for three conditions:

- type of popcorn (gourmet and plain)
- batch size popped (large and small)
- amount of oil used (little and lots)

**Figure 2.20** LSMeans Plots for Main Effects and Interactions



To transpose the factors of the LSMeans plot for a two-factor interaction, use Shift and select the **LSMeans Plot** option. Figure 2.21 shows both the default and the transposed factors plots.

**Figure 2.21** LSMeans Plot Comparison with Transposed Factors

## LSMeans Contrast

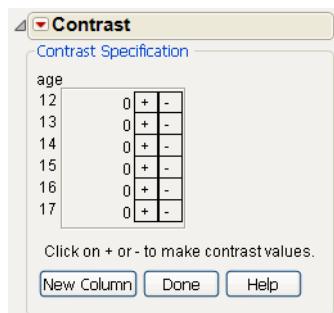
A **contrast** is a set of linear combinations of parameters that you want to jointly test to be zero. JMP builds contrasts in terms of the least squares means of the effect. By convention, each column of the contrast is normalized to have sum zero and an absolute sum equal to two.

If a contrast involves a covariate, you can specify the value of the covariate at which to test the contrast.

To illustrate using the **LSMeans Contrast** command:

1. Open Big Class.jmp from the sample data directory.
2. Click on **Analyze > Fit Model**.
3. Select **height** as the **Y** variable and **age** as the effect variable.
4. Click **Run**.
5. In the red-triangle menu for the **age** effect, select the **LSMeans Contrast** command.

A window is displayed for specifying contrasts with respect to the **age** effect. (This command is enabled only for pure classification effects.) The contrast window for the **age** effect using Big Class.jmp is shown in Figure 2.22.

**Figure 2.22** LSMeans Contrast Specification Window

## Effect Details

This Contrast window shows the name of the effect and the names of the levels in the effect. Beside the levels is an area enclosed in a rectangle that has a column of numbers next to boxes of + and - signs.

To construct a contrast, click the + and - boxes beside the levels that you want to compare. If possible, the window normalizes each time to make the sum for a column zero and the absolute sum equal to two each time you click; it adds to the plus or minus score proportionately.

For example, to form a contrast that compares the first two age levels with the second two levels, click + for the ages 12 and 13, and click - for ages 14 and 15. If you want to do more comparisons, click the **New Column** button for a new column to define the new contrast. The contrast for age is shown in Figure 2.23.

---

**Figure 2.23** LSMeans Contrast Specification for Age

The screenshot shows a software dialog box titled "Contrast". Inside, there's a section labeled "Contrast Specification" with a table for "age". The table has 6 rows (12, 13, 14, 15, 16, 17) and 3 columns. The first column is "age", the second is a numerical value, and the third is a grid of "+" and "-" signs. Below the table is a note: "Click on + or - to make contrast values." At the bottom are three buttons: "New Column", "Done", and "Help".

age	0.5	+	-
12	0.5	+	-
13	0.5	+	-
14	-0.5	+	-
15	-0.5	+	-
16	0	+	-
17	0	+	-

Click on + or - to make contrast values.

New Column Done Help

---

After you are through defining contrasts, click **Done**. The contrast is estimated, and the Contrast table shown in Figure 2.24 is appended to the other tables for that effect.

**Figure 2.24** LSMeans Contrast Results

Contrast	
Test Detail	
12	0.5
13	0.5
14	-0.5
15	-0.5
16	0
17	0
Estimate	-5.164
Std Error	1.1888
t Ratio	-4.344
Prob> t	0.0001
SS	215.88
Parameter Function	
Parameter	
Intercept	0
age[13-12]	-0.5
age[14-13]	-1
age[15-14]	-0.5
age[16-15]	0
age[17-16]	0

The Contrast table shows:

- the contrast s a function of the least squares means
- the estimates and standard errors of the contrast for the least squares means, and *t*-tests for each column of the contrast
- the *F*-test for all columns of the contrast tested jointly
- the Parameter Function table that shows the contrast expressed in terms of the parameters. In this example the parameters are for the ordinal variable, age.

## LSMeans Student's *t*, LSMeans Tukey's HSD

The **LSMeans Student's t** and **LSMeans Tukey's HSD** commands give multiple comparison tests for model effects.

**LSMeans Student's t** computes individual pairwise comparisons of least squares means in the model using Student's *t*-tests. This test is sized for individual comparisons. If you make many pairwise tests, there is no protection across the inferences. Thus, the alpha-size (Type I) error rate across the hypothesis tests is higher than that for individual tests.

**LSMeans Tukey's HSD** gives a test that is sized for all differences among the least squares means. This is the *Tukey* or *Tukey-Kramer HSD* (Honestly Significant Difference) test. (Tukey 1953, Kramer 1956). This test is an exact alpha-level test if the sample sizes are the same and conservative if the sample sizes are different (Hayter 1984).

These tests are discussed in detail in the *Basic Analysis and Graphing* book, which has examples and a description of how to read and interpret the multiple comparison tables.

The reports from both options have menus that allow for the display of additional reports.

**Crosstab Report** shows or hides the crosstab report. This report is a two-way table that highlights significant differences in red.

**Connecting Letters Report** shows or hides a report that illustrates significant differences with letters (similar to traditional SAS GLM output). Levels not connected by the same letter are significantly different.

**Ordered Differences Report** shows or hides a report that ranks the differences from lowest to highest. It also plots the differences on a histogram that has overlaid confidence interval lines. See the *Basic Analysis and Graphing* book for an example of an Ordered Differences report.

**Detailed Comparisons** shows reports and graphs that compare each level of the effect with all other levels in a pairwise fashion. See the *Basic Analysis and Graphing* book for an example of a Detailed Comparison Report.

**Equivalence Test** uses the TOST method to test for practical equivalence. See the *Basic Analysis and Graphing* book for details.

## Test Slices

The **Test Slices** command, which is enabled for interaction effects, is a quick way to do many contrasts at the same time. For each level of each classification column in the interaction, it makes comparisons among all the levels of the other classification columns in the interaction. For example, if an interaction is A\*B\*C, then there is a slice called A=1, which tests all the B\*C levels when A=1. There is another slice called A=2, and so on, for all the levels of B, and C. This is a way to detect the importance of levels inside an interaction.

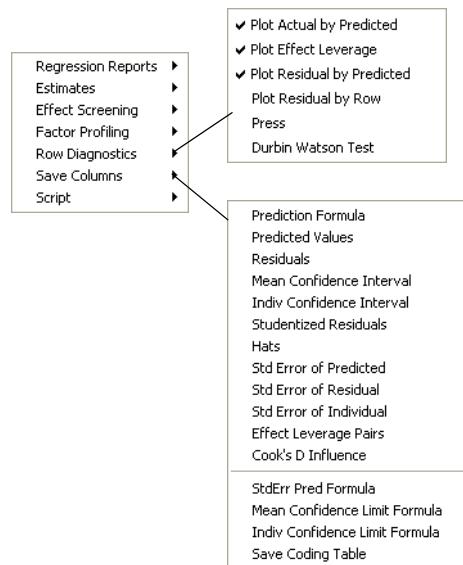
## Power Analysis

Power analysis is discussed in the chapter “[Standard Least Squares: Perspectives on the Estimates](#),” p. 57.

---

## Summary of Row Diagnostics and Save Commands

When you have a continuous response model and click the red triangle next to the response name on the response name title bar, the menu in Figure 2.25 is displayed.

**Figure 2.25** Commands for Least Squares Analysis: Row Diagnostics and Save Commands

The specifics of the Fit Model platform depend on the type of analysis you do and the options and commands you ask for. Menu commands and options are available at each level of the analysis. You always have access to all tables and plots through these menu items. Also, the default arrangement of results can be changed by using preferences or script commands.

The **Estimates** and **Effect Screening** menus are discussed in detail in the chapter “[Standard Least Squares: Perspectives on the Estimates](#),” p. 57. See the chapter “[Standard Least Squares: Exploring the Prediction Equation](#),” p. 87, for a description of the commands in the **Effect Screening** and **Factor Profiling** menus.

The next sections summarize commands in the **Row Diagnostics** and **Save** menus.

## Row Diagnostics

Leverage Plots (the **Plot Actual by Predicted** and **Plot Effect Leverage** commands) are covered previously in this chapter under “[Leverage Plots](#),” p. 36.

**Plot Actual by Predicted** displays the observed values by the predicted values of  $Y$ . This is the leverage plot for the whole model.

**Plot Effect Leverage** produces a leverage plot for each effect in the model showing the point-by-point composition of the test for that effect.

**Plot Residual By Predicted** displays the residual values by the predicted values of  $Y$ . You typically want to see the residual values scattered randomly about zero.

**Plot Residual By Row** displays the residual value by the row number of its observation.

**Press** displays a Press statistic, which computes the residual sum of squares where the residual for each row is computed after dropping that row from the computations. The Press statistic is the total prediction error sum of squares and is given by

$$\text{Press} = \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

$y_i$  is the observed response value of the  $i^{th}$  observation, and

$\hat{y}_i$  is the predicted response value of the omitted observation.

The Press RMSE is defined as  $\sqrt{\text{Press}/n}$ .

The Press statistic is useful when comparing multiple models. Models with lower Press statistics are favored.

**Durbin-Watson Test** displays the Durbin-Watson statistic to test whether the errors have first-order autocorrelation. The autocorrelation of the residuals is also shown. The Durbin-Watson table has a popup command that computes and displays the exact probability associated with the statistic. This Durbin-Watson table is appropriate only for time series data when you suspect that the errors are correlated across time.

**Note:** The computation of the Durbin-Watson exact probability can be time-intensive if there are many observations. The space and time needed for the computation increase with the square and the cube of the number of observations, respectively.

## Save Columns Command

The **Save** submenu offers the following choices. Each selection generates one or more new columns in the current data table titled as shown, where *colname* is the name of the response variable:

**Prediction Formula** creates a new column, called **Pred Formula colname**, containing the predicted values computed by the specified model. It differs from the **Predicted colname** column, because it contains the prediction formula. This is useful for predicting values in new rows or for obtaining a picture of the fitted model.

Use the **Column Info** command and click the **Edit Formula** button to see the prediction formula. The prediction formula can require considerable space if the model is large. If you do not need the formula with the column of predicted values, use the **Save Columns > Predicted Values** option. For information about formulas, see the *JMP User Guide*.

---

**Note:** When using this command, an attempt is first made to find a Response Limits property containing desirability functions. The desirability functions are determined from the profiler, if that option has been used. Otherwise, the desirability functions are determined from the response columns. If you reset the desirabilities later, it affects only subsequent saves. (It does not affect columns that have already been saved.)

**Predicted Values** creates a new column called **Predicted colname** that contain the predicted values computed by the specified model.

**Residuals** creates a new column called **Residual colname** containing the residuals, which are the observed response values minus predicted values.

**Mean Confidence Interval** creates two new columns called Lower 95% Mean colname and Upper 95% Mean colname. The new columns contain the lower and upper 95% confidence limits for the line of fit.

**Note:** If you hold down the Shift key and select the option, you are prompted to enter an  $\alpha$ -level for the computations.

**Indiv Confidence Interval** creates two new columns called Lower 95% Indiv colname and Upper 95% Indiv colname. The new columns contain lower and upper 95% confidence limits for individual response values.

**Note:** If you hold down the Shift key and select the option, you are prompted to enter an  $\alpha$ -level for the computations.

**Studentized Residuals** creates a new column called Studentized Resid colname. The new column values are the residuals divided by their standard error.

**Hats** creates a new column called h colname. The new column values are the diagonal values of the matrix  $X(X'X)^{-1}X'$ , sometimes called hat values.

**Std Error of Predicted** creates a new column, called StdErr Pred colname, containing the standard errors of the predicted values.

**Std Error of Residual** creates a new column called, StdErr Resid colname, containing the standard errors of the residual values.

**Std Error of Individual** creates a new column, called StdErr Indiv colname, containing the standard errors of the individual predicted values.

**Effect Leverage Pairs** creates a set of new columns that contain the values for each leverage plot.

The new columns consist of an X and Y column for each effect in the model. The columns are named as follows. If the response column name is R and the effects are X1 and X2, then the new column names are:

X Leverage of X1 for R      Y Leverage of X1 for R

X Leverage of X2 for R      Y Leverage of X2 for R.

**Cook's D Influence** saves the Cook's D influence statistic. Influential observations are those that, according to various criteria, appear to have a large influence on the parameter estimates.

**StdErr Pred Formula** creates a new column, called PredSE colname, containing the standard error of the predicted values. It is the same as the **Std Error of Predicted** option but saves the formula with the column. Also, it can produce very large formulas.

**Mean Confidence Limit Formula** creates a new column in the data table containing a formula for the mean confidence intervals.

**Note:** If you hold down the Shift key and select the option, you are prompted to enter an  $\alpha$ -level for the computations.

**Indiv Confidence Limit Formula** creates a new column in the data table containing a formula for the individual confidence intervals.

**Note:** If you hold down the Shift key and select the option, you are prompted to enter an  $\alpha$ -level for the computations.

**Save Coding Table** produces a new data table showing the intercept, all continuous terms, and coded values for nominal terms.

**Note:** If you are using the Graph command to invoke the Profiler, then you should first save the columns Prediction Formula and StdErr Pred Formula to the data table. Then, place both of these formulas into the **Y, Prediction Formula** role in the Profiler launch window. The resulting window asks if you want to use the PredSE colname to make confidence intervals for the Pred Formula colname, instead of making a separate profiler plot for the PredSE colname.

## Examples with Statistical Details

This section continues with the example at the beginning of the chapter that uses the Drug.jmp data table Snedecor and Cochran (1967, p. 422). The introduction shows a one-way analysis of variance on three drugs labeled a, d, and f, given to three groups randomly selected from 30 subjects.

Run the example again with response y and effect Drug. The next sections dig deeper into the analysis and also add another effect, x, that has a role to play in the model.

### One-Way Analysis of Variance with Contrasts

In a one-way analysis of variance, a different mean is fit to each of the different sample (response) groups, as identified by a nominal variable. To specify the model for JMP, select a continuous *Y* and a nominal *X* variable such as Drug. In this example Drug has values a, d, and f. The standard least squares fitting method translates this specification into a linear model as follows: The nominal variables define a sequence of dummy variables, which have only values 1, 0, and -1. The linear model is written

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

where

$y_i$  is the observed response in the  $i^{\text{th}}$  trial

$x_{1i}$  is the level of the first predictor variable in the  $i^{\text{th}}$  trial

$x_{2i}$  is the level of the second predictor variable in the  $i^{\text{th}}$  trial

$\beta_0$ ,  $\beta_1$ , and  $\beta_2$  are parameters for the intercept, the first predictor variable, and the second predictor variable, respectively, and

$\varepsilon_i$  are the independent and normally distributed error terms in the  $i^{\text{th}}$  trial

As shown here, the first dummy variable denotes that Drug=a contributes a value 1 and Drug=f contributes a value -1 to the dummy variable.

1	a
0	d
-1	f

The second dummy variable is given values

$$x_{2i} = \begin{bmatrix} 0 & a \\ 1 & d \\ -1 & f \end{bmatrix}$$

The last level does not need a dummy variable because in this model its level is found by subtracting all the other parameters. Therefore, the coefficients sum to zero across all the levels.

The estimates of the means for the three levels in terms of this parameterization are:

$$\mu_1 = \beta_0 + \beta_1$$

$$\mu_2 = \beta_0 + \beta_2$$

$$\mu_3 = \beta_0 - \beta_1 - \beta_2$$

Solving for  $\beta_i$  yields

$$\beta_0 = \frac{(\mu_1 + \mu_2 + \mu_3)}{3} = \mu \text{ (the average over levels)}$$

$$\beta_1 = \mu_1 - \mu$$

$$\beta_2 = \mu_2 - \mu$$

$$\beta_3 = \beta_1 - \beta_2 = \mu_3 - \mu$$

Thus, if regressor variables are coded as indicators for each level minus the indicator for the last level, then the parameter for a level is interpreted as the difference between that level's response and the average response across all levels. See “Nominal Factors,” p. 613 in the “Statistical Details” chapter for additional information about the interpretation of the parameters for nominal factors.

Figure 2.26 shows the Parameter Estimates and the Effect Tests reports from the one-way analysis of the drug data. Figure 2.4, at the beginning of the chapter, shows the Least Squares Means report and LS Means Plot for the Drug effect.

---

**Figure 2.26** Parameter Estimates and Effect Tests for Drug.jmp

The screenshot displays two tables from the JMP software. The first table, titled "Parameter Estimates", lists the estimated values for the intercept and two drug categories (Drug[a] and Drug[d]). The second table, titled "Effect Tests", provides an overall F-test for the Drug effect.

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	7.9	1.108386	7.13	<.0001*
Drug[a]	-2.6	1.567494	-1.66	0.1088
Drug[d]	-1.8	1.567494	-1.15	0.2609

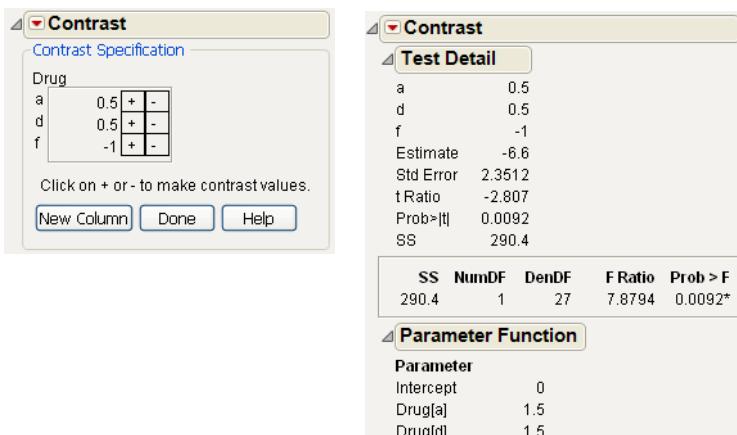
Sum of					
Source	Nparm	DF	Squares	F Ratio	Prob > F
Drug	2	2	293.60000	3.9831	0.0305*

---

The Drug effect can be studied in more detail by using a contrast of the least squares means. To do this, click the red triangle next to the Drug effect title and select **LSMeans Contrast** to obtain the Contrast specification window.

Click the + boxes for drugs a and d, and the - box for drug f to define the contrast that compares the average of drugs a and d to f (shown in Figure 2.27). Then click **Done** to obtain the Contrast report. The report shows that the Drug effect looks more significant using this one-degree-of-freedom comparison test; The LSMean for drug f is clearly significantly different from the average of the LSMeans of the other two drugs.

**Figure 2.27** Contrast Example for the Drug Experiment



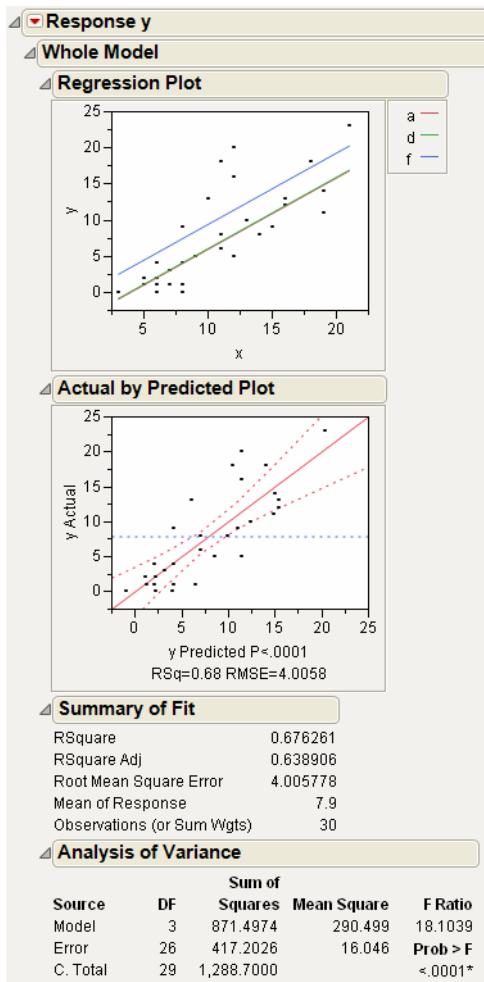
## Analysis of Covariance

An analysis of variance model with an added regressor term is called an *analysis of covariance*. Suppose that the data are the same as above, but with one additional term,  $x_{3i}$ , in the formula as a new regressor. Both  $x_{1i}$  and  $x_{2i}$  continue to be dummy variables that index over the three levels of the nominal effect. The model is written

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i$$

Now there is an intercept plus two effects, one a nominal main effect using two parameters, and the other an interval covariate regressor using one parameter.

Rerun the Snedecor and Cochran Drug.jmp example, but add the x to the model effects as a covariate. Compared with the main effects model (Drug effect only), the  $R^2$  increases from 22.8% to 67.6%, and the standard error of the residual reduces from 6.07 to 4.0. As shown in Figure 2.28, the F-test significance probability for the whole model decreases from 0.03 to less than 0.0001.

**Figure 2.28** ANCOVA Drug Results

Sometimes you can investigate the functional contribution of a covariate. For example, some transformation of the covariate might fit better. If you happen to have data where there are exact duplicate observations for the regressor effects, it is possible to partition the total error into two components. One component estimates error from the data where all the  $x$  values are the same. The other estimates error that can contain effects for unspecified functional forms of covariates, or interactions of nominal effects. This is the basis for a lack of fit test. If the lack of fit error is significant, then the fit model platform warns that there is some effect in your data not explained by your model. Note that there is no significant lack of fit error in this example, as seen by the large probability value of 0.7507.

The covariate,  $x$ , has a substitution effect with respect to Drug. It accounts for much of the variation in the response previously accounted for by the Drug variable. Thus, even though the model is fit with much less

error, the Drug effect is no longer significant. The effect previously observed in the main effects model now appears explainable to some extent in terms of the values of the covariate.

The least squares means are now different from the ordinary mean because they are adjusted for the effect of  $x$ , the covariate, on the response,  $y$ . Now the least squares means are the predicted values that you expect for each of the three values of Drug, given that the covariate,  $x$ , is held at some constant value. The constant value is chosen for convenience to be the mean of the covariate, which is 10.7333.

So, the prediction equation gives the least squares means as follows:

$$\text{fit equation: } -2.696 - 1.185\text{Drug}[a-f] - 1.0761\text{Drug}[d-f] + 0.98718*x$$

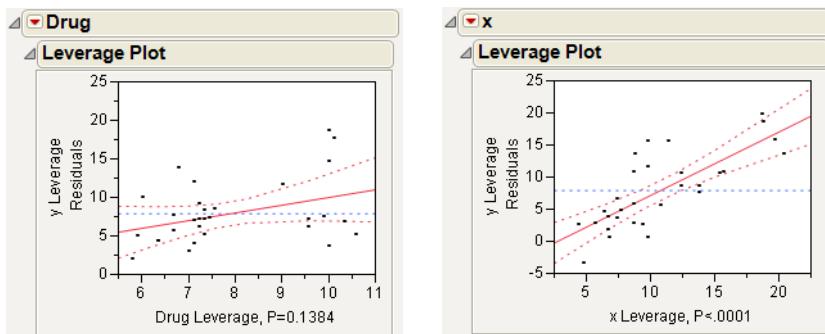
$$\text{for a: } -2.696 - 1.185*(1) - 1.0761*(0) + 0.98718*(10.7333) = 6.71$$

$$\text{for d: } -2.696 - 1.185*(0) - 1.0761*(1) + 0.98718*(10.7333) = 6.82$$

$$\text{for f: } -2.696 - 1.185*(-1) - 1.0761*(-1) + 0.98718*(10.7333) = 10.16$$

Figure 2.29 shows a leverage plot for each effect. Because the covariate is significant, the leverage values for Drug are dispersed somewhat from their least squares means.

**Figure 2.29** Comparison of Leverage Plots for Drug Test Data



## Analysis of Covariance with Separate Slopes

This example is a continuation of the Drug.jmp example presented in the previous section. The example uses data from Snedecor and Cochran (1967, p. 422). A one-way analysis of variance for a variable called Drug, shows a difference in the mean response among the levels a, d, and f, with a significance probability of 0.03.

The lack of fit test for the model with main effect Drug and covariate  $x$  is not significant. However, for the sake of illustration, this example includes the main effects and the Drug\*x effect. This model tests whether the regression on the covariate has separate slopes for different Drug levels.

This specification adds two columns to the linear model (call them  $x_{4i}$  and  $x_{5i}$ ) that allow the slopes for the covariate to be different for each Drug level. The new variables are formed by multiplying the dummy variables for Drug by the covariate values giving

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i} + \beta_5 x_{5i} + \varepsilon_i$$

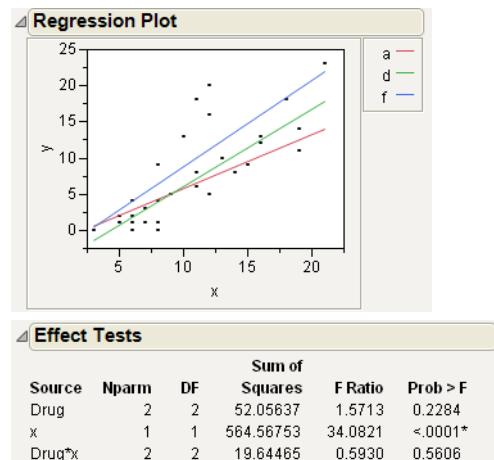
**Table 2.2 “Coding of Analysis of Covariance with Separate Slopes,” p. 55,** shows the coding of this Analysis of Covariance with Separate Slopes. **Note:** The mean of  $X$  is 10.7333.

**Table 2.2** Coding of Analysis of Covariance with Separate Slopes

Regressor	Effect	Values
$X_1$	Drug[a]	+1 if a, 0 if d, -1 if f
$X_2$	Drug[d]	0 if a, +1 if d, -1 if f
$X_3$	X	the values of X
$X_4$	Drug[a] * (X - 10.733)	+X - 10.733 if a, 0 if d, -(X - 10.733) if f
$X_5$	Drug[d] * (X - 10.733)	0 if a, +X - 10.733 if d, -(X - 10.733) if f

A portion of the report is shown in Figure 2.30. The Regression Plot shows fitted lines with different slopes. The Effect Tests report gives a p-value for the interaction of 0.56. This is not significant, indicating the model does not need to have different slopes.

**Figure 2.30** Plot with Interaction



## Singularity Details

When there are linear dependencies between model effects, the Singularity Details report appears.

---

**Figure 2.31** Singularity Report

 **Singularity Details**

```
Intercept = - popcorn[gourmet] - oil amt[little] -
popcorn[gourmet]*oil amt[little] - batch [large] -
popcorn[gourmet]*batch [large] - oil amt[little]*batch [large] -
popcorn[gourmet]*oil amt[little]*batch [large]
```

---

# Chapter 3

## Standard Least Squares: Perspectives on the Estimates Fit Model Platform

---

Though the fitting platform always produces a report on the parameter estimates and tests on the effects, there are many options available to make these more interpretable. The following sections address these questions:

**Table 3.1** Additional Options for Standard Least Squares

- |  |                          |
|--|--------------------------|
| • How do I interpret estimates for nominal factors? How can I get the missing level coefficients?  | Expanded Estimates       |
| • How can I measure the size of an effect in a scale-invariant fashion?  | Scaled Estimates         |
| • If I have a screening design with many effects but few observations, how can I decide which effects are sizable and active?                              | Effect Screening Options |
| • How can I get a series of tests of effects that are independent tests whose sums of squares add up to the total, even though the design is not balanced? | Sequential Tests         |
| • How can I test some specific combination?  | Custom Test              |
| • How can I predict (backwards) which $x$ value led to a given $y$ value?  | Inverse Prediction       |
| • How likely is an effect to be significant if I collect more data, have a different effect size, or have a different error variance?                      | Power Analysis           |
| • How strongly are estimates correlated?   | Correlation of Estimates |

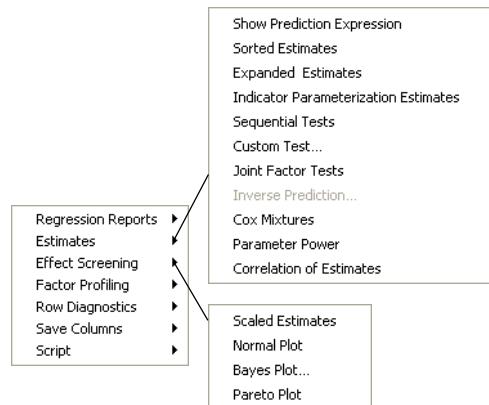
# Contents

Estimates and Effect Screening Menus .....	59
Show Prediction Expression .....	59
Sorted Estimates .....	60
Expanded Estimates and the Coding of Nominal Terms .....	60
Scaled Estimates and the Coding Of Continuous Terms .....	62
Indicator Parameterization Estimates .....	63
Sequential Tests .....	63
Custom Test .....	63
Joint Factor Tests .....	65
Inverse Prediction .....	65
Cox Mixtures .....	69
Parameter Power .....	70
The Power Analysis Dialog .....	72
Effect Size .....	73
Text Reports for Power Analysis .....	74
Plot of Power by Sample Size .....	75
The Least Significant Value (LSV) .....	75
The Least Significant Number (LSN) .....	76
The Power .....	76
The Adjusted Power and Confidence Intervals .....	76
Prospective Power Analysis .....	77
Correlation of Estimates .....	78
Effect Screening .....	79
Lenth's Method .....	79
Parameter Estimates Population .....	80
Normal Plot .....	82
Half-Normal Plot .....	83
Bayes Plot .....	83
Bayes Plot for Factor Activity .....	84
Pareto Plot .....	85

## Estimates and Effect Screening Menus

Most parts of the Model Fit results are optional. When you click the popup icon next to the response name at the topmost outline level, the menu shown in Figure 3.1 lists commands for the continuous response model.

**Figure 3.1** Commands for a Least Squares Analysis: the **Estimates** and **Effect Screening** Menus



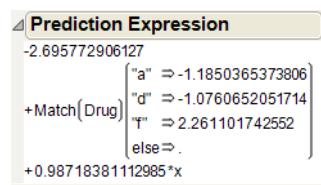
The specifics of the Fit Model platform depend on the type of analysis you do and the options and commands you ask for. The popup icons list commands and options at each level of the analysis. You always have access to all tables and plots through menu items. Also, the default arrangement of results can be changed by using preferences or script commands.

The focus of this chapter is on items in the **Estimates** and **Effect Screening** menus, which includes inverse prediction and a discussion of parameter power.

## Show Prediction Expression

The **Show Prediction Expression** command places the prediction expression in the report. Figure 3.2 shows the equation for the Drug.jmp data table with Drug and x as predictors.

**Figure 3.2** Prediction Expression



---

## Sorted Estimates

The sorted Estimates command produces a different version of the Parameter Estimates report that is more useful in screening situations. This version of the report is especially useful if the design is saturated, when typical reports are less informative.

### Example of a Sorted Estimates Report

1. Open the Drug.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select y and click Y.
4. Add Drug and x as the effects.
5. Click **Run**.
6. From the red triangle menu next to Response y, select **Estimates > Sorted Estimates**.

---

**Figure 3.3** Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob> t
x	0.9871838	0.164498	6.00	<.0001*
Drug[a]	-1.185037	1.060822	-1.12	0.2742
Drug[d]	-1.076065	1.041298	-1.03	0.3109

This report is shown automatically if the emphasis is screening and all the effects have only one parameter. In that case, the Scaled Estimates report is not shown.

Note the following differences between this report and the Parameter Estimates report:

- This report does not show the intercept.
- The effects are sorted by the absolute value of the *t*-ratio, showing the most significant effects at the top.
- A bar graph shows the *t*-ratio, with a line showing the 0.05 significance level.
- If JMP cannot obtain standard errors for the estimates, relative standard errors are used and notated.
- If there are no degrees of freedom for residual error, JMP constructs *t*-ratios and *p*-values using Lenth's Pseudo-Standard Error. These quantities are labeled with Pseudo in their name. A note explains the change and shows the PSE. To calculate *p*-values, JMP uses a DFE of  $m/3$ , where  $m$  is the number of parameter estimates excluding the intercept.

---

## Expanded Estimates and the Coding of Nominal Terms

**Expanded Estimates** is useful when there are categorical (nominal) terms in the model and you want a full set of effect coefficients.

When you have nominal terms in your model, the platform needs to construct a set of dummy columns to represent the levels in the classification. Full details are shown in the appendix “[Statistical Details](#),” p. 607. For  $n$  levels, there are  $n - 1$  dummy columns. Each dummy variable is a zero-or-one indicator for a particular level, except for the last level, which is coded  $-1$  for all dummy variables. For example, if column  $A$  has levels  $A1, A2, A3$ , then the dummy columns for  $A1$  and  $A2$  are as shown here.

$A$	$A1$ dummy	$A2$ dummy
$A1$	1	0
$A2$	0	1
$A3$	-1	-1

These columns are not displayed. They are just for conceptualizing how the fitting is done. The parameter estimates are the coefficients fit to these columns. In this case, there are two of them, labeled  $A[A1]$  and  $A[A2]$ .

This coding causes the parameter estimates to be interpreted as how much the response for each level differs from the average across all levels. Suppose, however, that you want the coefficient for the last level,  $A[A3]$ . The coefficient for the last level is the negative of the sum across the other levels, because the sum across all levels is constrained to be zero. Although many other codings are possible, this coding has proven to be practical and interpretable.

However, you probably don’t want to do hand calculations to get the estimate for the last level. The **Expanded Estimates** command in the **Estimates** menu calculates these missing estimates and shows them in a text report. You can verify that the mean (or sum) of the estimates across a classification is zero.

Keep in mind that the **Expanded Estimates** option with high-degree interactions of two-level factors produces a lengthy report. For example, a five-way interaction of two-level factors produces only one parameter but has  $2^5 = 32$  expanded coefficients, which are all the same except for sign changes.

To recreate the reports in Figure 3.4, follow the steps in “[Example of a Sorted Estimates Report](#),” p. 60, except instead of selecting **Sorted Estimates**, select **Expanded Estimates**.

**Figure 3.4** Comparison of Parameter Estimates and Expanded Estimates

The screenshot displays two tables from a SAS output window. The first table, titled "Parameter Estimates", lists the following coefficients:

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	-2.695773	1.911085	-1.41	0.1702
Drug[a]	-1.185037	1.060822	-1.12	0.2742
Drug[d]	-1.076065	1.041298	-1.03	0.3109
x	0.9871838	0.164498	6.00	<.0001*

The second table, titled "Expanded Estimates", lists the same coefficients with an additional row for the reference level (A3), which is zero for all terms except the intercept:

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	-2.695773	1.911085	-1.41	0.1702
Drug[a]	-1.185037	1.060822	-1.12	0.2742
Drug[d]	-1.076065	1.041298	-1.03	0.3109
Drug[f]	2.2611017	1.093974	2.07	0.0488*
x	0.9871838	0.164498	6.00	<.0001*

## Scaled Estimates and the Coding Of Continuous Terms

The parameter estimates are highly dependent on the scale of the factor. If you convert a factor from grams to kilograms, the parameter estimates change by a multiple of a thousand. If the same change is applied to a squared (quadratic) term, the scale changes by a multiple of a million. If you are interested in the effect size, then you should examine the estimates in a more scale-invariant fashion. This means converting from an arbitrary scale to a meaningful one so that the sizes of the estimates relate to the size of the effect on the response. There are many approaches to doing this. In JMP, the **Scaled Estimates** command on the **Effect Screening** menu gives coefficients corresponding to factors that are scaled to have a mean of zero and a range of two. If the factor is symmetrically distributed in the data then the scaled factor will have a range from  $-1$  to  $1$ . This corresponds to the scaling used in the design of experiments (DOE) tradition. Thus, for a simple regressor, the scaled estimate is half the predicted response change as the regression factor travels its whole range.

Scaled estimates are important in assessing effect sizes for experimental data in which the uncoded values are used. If you use coded values ( $-1$  to  $1$ ), then the scaled estimates are no different than the regular estimates.

Also, you do not need scaled estimates if your factors have the Coding column property. In that case, they are converted to uncoded form when the model is estimated and the results are already in an interpretable form for effect sizes.

To recreate the report in Figure 3.5, follow the steps in “[Example of a Sorted Estimates Report](#),” p. 60, except instead of selecting **Estimates > Expanded Estimates**, select **Effect Screening > Scaled Estimates**.

As noted in the report, the estimates are parameter centered by the mean and scaled by range/2.

**Figure 3.5** Scaled Estimates

Scaled Estimates					
Nominal factors expanded to all levels					
Continuous factors centered by mean, scaled by range/2					
Scaled					
Term	Estimate	Std Error	t Ratio	Prob> t	
Intercept	7.9	0.731352	10.80	<.0001*	
Drug[a]	-1.195037	1.060822	-1.12	0.2742	
Drug[d]	-1.076065	1.041298	-1.03	0.3109	
Drug[f]	2.2611017	1.093974	2.07	0.0488*	
x	8.8846543	1.480478	6.00	<.0001*	

Scaled estimates also take care of the issues for polynomial (crossed continuous) models even if they are not centered by the parameterized **Center Polynomials** default launch option.

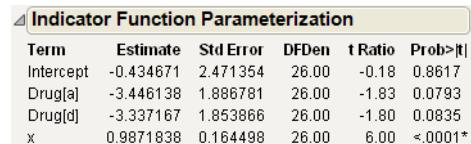
---

## Indicator Parameterization Estimates

This command displays the estimates using the Indicator Variable parameterization. To recreate the report in Figure 3.6, follow the steps in “[Example of a Sorted Estimates Report](#),” p. 60, except instead of selecting **Expanded Estimates**, select **Indicator Parameterization Estimates**.

---

**Figure 3.6** Indicator Parameterization Estimates



Term	Estimate	Std Error	DFDen	t Ratio	Prob> t
Intercept	-0.434671	2.471354	26.00	-0.18	0.8617
Drug[a]	-3.446138	1.886781	26.00	-1.83	0.0793
Drug[d]	-3.337167	1.853866	26.00	-1.80	0.0835
x	0.9871838	0.164498	26.00	6.00	<.0001*

This parameterization is inspired by the PROC GLM parameterization. Some models will match, but others, such as no-intercept models, models with missing cells, and mixture models, will most likely show differences.

---

## Sequential Tests

**Sequential Tests** shows the reduction in residual sum of squares as each effect is entered into the fit. The sequential tests are also called Type I sums of squares (Type I SS). A desirable property of the Type I SS is that they are independent and sum to the regression SS. An undesirable property is that they depend on the order of terms in the model. Each effect is adjusted only for the preceding effects in the model.

The following models are considered appropriate for the Type I hypotheses:

- balanced analysis of variance models specified in proper sequence (that is, interactions do not precede main effects in the effects list, and so forth)
- purely nested models specified in the proper sequence
- polynomial regression models specified in the proper sequence.

---

## Custom Test

If you want to test a custom hypothesis, select **Custom Test** from the **Estimates** popup menu. If you want to jointly test several linear functions, click on **Add Column**. This displays the dialog shown to the left in Figure 3.7. After filling in the test, click **Done**. The dialog then changes to a report of the results, as shown on the right in Figure 3.7.

The space beneath the **Custom Test** title bar is an editable area for entering a test name. Use the Custom Test dialog as follows:

## Custom Test

**Parameter** lists the names of the model parameters. To the right of the list of parameters are columns of zeros corresponding to these parameters. Click a cell here, and enter a new value corresponding to the test you want.

**Add Column** adds a column of zeros so that you can test jointly several linear functions of the parameters. Use the **Add Column** button to add as many columns to the test as you want.

The last line in the Parameter list is labeled **=**. Enter a constant into this box to test the linear constraint against. For example, to test the hypothesis  $\beta_0=1$ , enter a 1 in the **=** box. In Figure 3.7, the constant is equal to zero.

When you finish specifying the test, click **Done** to see the test performed. The results are appended to the bottom of the dialog.

When the custom test is done, the report lists the test name, the function value of the parameters tested, the standard error, and other statistics for each test column in the dialog. A joint *F*-test for all columns shows at the bottom.

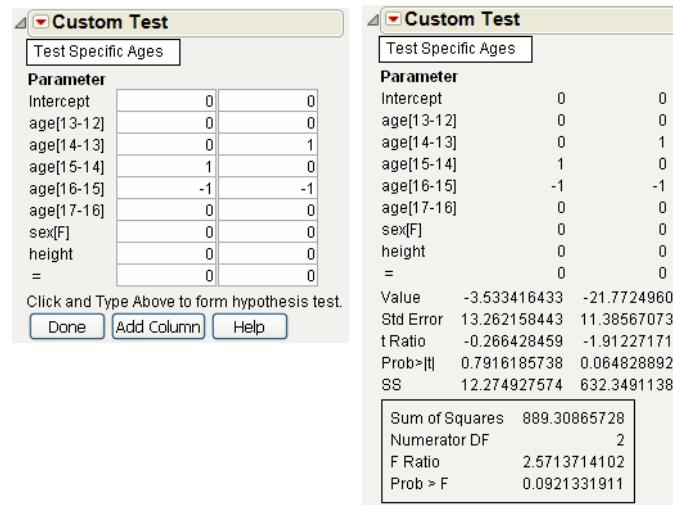
Warning: The test is always done with respect to residual error. If you have random effects in your model, this test may not be appropriate if you use EMS instead of REML.

---

**Note:** If you have a test within a classification effect, consider using the contrast dialog (which tests hypotheses about the least squares means) instead of a custom test.

---

**Figure 3.7** The Custom Test Dialog and Test Results for Age Variable



---

## Joint Factor Tests

This command appears when interaction effects are present. For each main effect in the model, JMP produces a joint test on all the parameters involving that main effect.

### Example of a Joint Factor Tests Report

1. Open the Big Class.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select weight and click Y.
4. Select age, sex, and height and click **Macros > Factorial to degree**.
5. Click **Run**.
6. From the red triangle next to Response weight, select **Estimates > Joint Factor Tests**.

---

**Figure 3.8** Joint Factor Tests for Big Class model

Term	DF	Sum of Squares	F Ratio	Prob > F
age	15	6,116.2127	2.8488	0.0139*
sex	7	2,113.1080	2.1091	0.0879
height	7	9,217.8156	9.2002	<.0001*

Note that age has 15 degrees of freedom because it is testing the five parameters for age, the five parameters for age\*sex, and the five parameters for height\*age, all tested to be zero.

---

## Inverse Prediction

To find the value of  $x$  for a given  $y$  requires *inverse prediction*, sometimes called *calibration*. The **Inverse Prediction** command on the **Estimates** menu displays a dialog (Figure 3.10) that lets you ask for a specific value of one independent ( $X$ ) variable, given a specific value of a dependent variable and other  $x$  values. The inverse prediction computation includes confidence limits (fiducial limits) on the prediction.

### Example of Inverse Prediction

1. Open the Fitness.jmp sample data table.
2. Select **Analyze > Fit Y by X**.
3. Select Oxy and click **Y, Response**.
4. Select Runtime and click **X, Factor**.

When there is only a single  $X$ , as in this example, the Fit Y by X platform can give you a visual approximation of the inverse prediction values.

5. Click **OK**.

## Inverse Prediction

6. From the red triangle menu, select **Fit Line**.

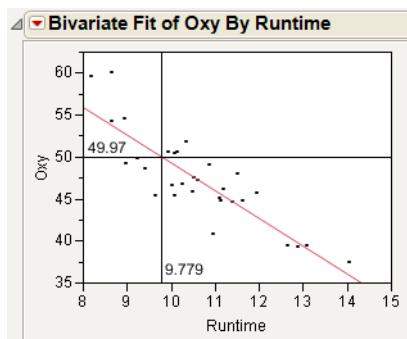
Use the crosshair tool to approximate inverse prediction.

7. Select **Tools > Crosshairs**.

8. Position the crosshair tool with its horizontal line crossing the Oxy axis at about 50, and its intersection with the vertical line positioned on the prediction line.

---

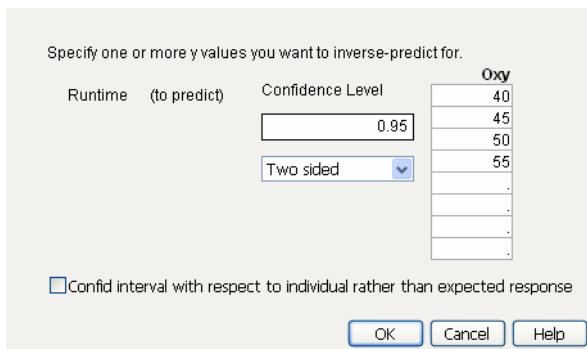
**Figure 3.9** Bivariate Fit for Fitness.jmp



---

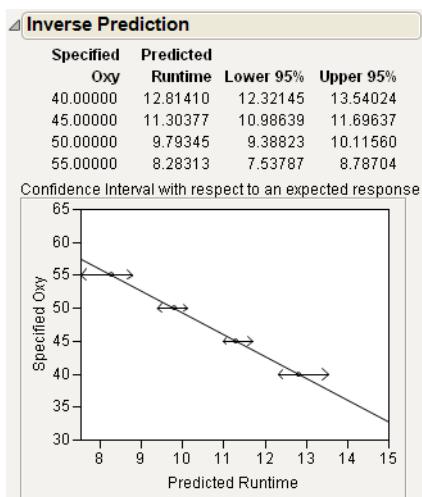
This shows which value of Runtime gives an Oxy value of 50, intersecting the Runtime axis at about 9.779, which is an approximate inverse prediction. However, to see the exact prediction of Runtime, use the Fit Model dialog, as follows:

1. From the Fitness.jmp sample data table, select **Analyze > Fit Model**.
2. Select Oxy and click **Y**.
3. Add Runtime as the single model effect.
4. Click **Run**.
5. From the red triangle menu next to Response Oxy, select **Estimates > Inverse Prediction**.

**Figure 3.10** Inverse Prediction Given by the Fit Model Platform

6. Type the values for Oxy, as shown in Figure 3.10.
7. Click OK.

The dialog disappears and the Inverse Prediction table in Figure 3.11 gives the exact predictions for each Oxy value specified, with upper and lower 95% confidence limits. The exact prediction for Runtime when Oxy is 50 is 9.7935, which is close to the approximate prediction of 9.779 found in Figure 3.9.

**Figure 3.11** Inverse Prediction Given by the Fit Model Platform

**Note:** The fiducial confidence limits are formed by Fieller's method. Sometimes this method results in a degenerate (outside) interval, or an infinite interval, for one or both sides of an interval. When this happens for both sides, Wald intervals are used. If it happens for only one side, the Fieller method is still used and a

## Inverse Prediction

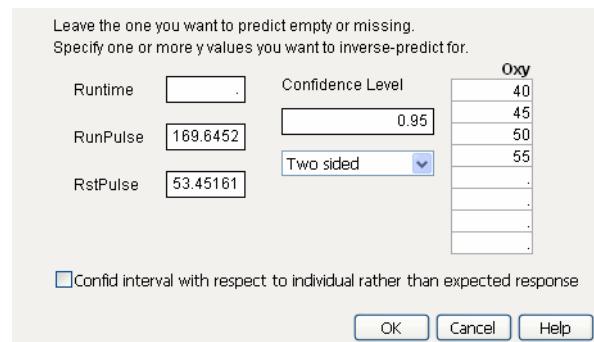
missing value is returned. See the appendix “[Statistical Details](#),” p. 607, for information about computing the confidence limits.

The inverse prediction command also predicts a single  $x$  value when there are multiple effects in the model. To predict a single  $x$ , you supply one or more  $y$  values of interest and set the  $x$  value you want to predict to be missing. By default, the other  $x$  values are set to the regressor’s means but can be changed to any desirable value.

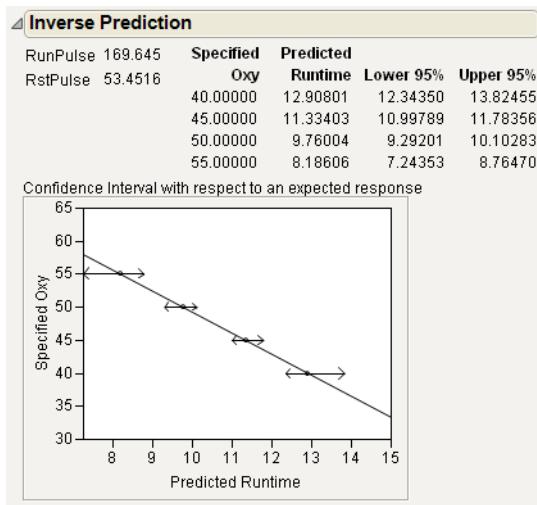
#### Example Predicting a Single X Value with Multiple Model Effects

1. From the **Fitness.jmp** sample data table, select **Analyze > Fit Model**.
2. Select **Oxy** and click **Y**.
3. Add **Runtime**, **RunPulse**, and **RstPulse** as effects.
4. Click **Run**.
5. From the red triangle menu next to Response **Oxy**, select **Estimates > Inverse Prediction**.

**Figure 3.12** Inverse Prediction Dialog for a Multiple Regression Model



6. Type the values for **Oxy**, as shown in Figure 3.12.
7. Delete the value for **Runtime**, since that is the value you want to predict.
8. Click **OK**.

**Figure 3.13** Inverse Prediction for a Multiple Regression Model

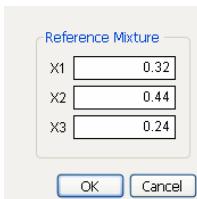
## Cox Mixtures

**Note:** This option is available only for mixture models.

In mixture designs, the model parameters cannot easily be used to judge the effects of the mixture components. The Cox Mixture model (a reparameterized and constrained version of the Scheffe model) produces parameter estimates from which inference can be made about factor effects and the response surface shape, relative to a reference point in the design space. See Cornell (1990) for a complete discussion.

### Example of Cox Mixtures

1. Open the Five Factor Mixture.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select Y1 and click **Y**.
4. Select X1, X2, and X3. Click on **Macros > Mixture Response Surface**.
5. Click **Run**.
6. From the red triangle menu next to Response Y1, select **Estimates > Cox Mixtures**.

**Figure 3.14** Cox Mixtures Dialog

Specify the reference mixture points. Note that if the components of the reference point do not sum to one, then the values are scaled so that they do sum to one.

- Replace the existing values as shown in Figure 3.14, and click **OK**.

**Figure 3.15** Cox Mixtures

Cox Reference Mixture Model						
Parameter	Estimate	Std Error	t Ratio	Prob> t	Factor	Reference Mixture
Intercept	1.25206	1.919933	0.652	0.5163		
X1	0.65808	2.897694	0.227	0.8210	X1	0.3200000
X2	1.83904	1.679614	1.095	0.2771	X2	0.4400000
X3	-4.24902	4.159336	-1.022	0.3103	X3	0.2400000
X1^2	0.82682	4.215683	0.196	0.8450		
X2^2	2.39249	2.842954	0.842	0.4027		
X3^2	5.19189	5.121428	1.014	0.3140		
X1*X2	-1.76705	6.243961	-0.283	0.7780		
X1*X3	1.03474	7.298896	0.142	0.8876		
X2*X3	-6.41642	6.429410	-0.998	0.3215		

The parameter estimates are added to the report window, along with standard errors, hypothesis tests, and the reference mixture.

## Parameter Power

Suppose that you want to know how likely your experiment is to detect some difference at a given  $\alpha$ -level. The probability of getting a significant test result is termed the power. The power is a function of the unknown parameter values tested, the sample size, and the unknown residual error variance.

Or, suppose that you already did an experiment and the effect was not significant. If you think that it might have been significant if you had more data, then you would like to get a good guess of how much more data you need.

JMP offers the following calculations of statistical power and other details relating to a given hypothesis test.

- LSV, the *least significant value*, is the value of some parameter or function of parameters that would produce a certain  $p$ -value alpha.

- LSN, the *least significant number*, is the number of observations that would produce a specified *p*-value alpha if the data has the same structure and estimates as the current sample.
- *Power* is the probability of getting at or below a given *p*-value alpha for a given test.

The LSV, LSN, and power values are important measuring sticks that should be available for all test statistics. They are especially important when the test statistics do not show significance. If a result is not significant, the experimenter should at least know how far from significant the result is in the space of the estimate (rather than in the probability) and know how much additional data is needed to confirm significance for a given value of the parameters.

Sometimes a novice confuses the role of the null hypotheses, thinking that failure to reject the null hypothesis is equivalent to proving it. For this reason, it is recommended that the test be presented in these other aspects (power and LSN) that show how sensitive the test is. If an analysis shows no significant difference, it is useful to know the smallest difference the test is likely to detect (LSV).

The power details provided by JMP can be used for both prospective and retrospective power analyses. A prospective analysis is useful in the planning stages of a study to determine how large your sample size must be in order to obtain a desired power in tests of hypothesis. See the section “[Prospective Power Analysis](#),” [p. 77](#), for more information and a complete example. A retrospective analysis is useful during the data analysis stage to determine the power of hypothesis tests already conducted.

Technical details for power, LSN, and LSV are covered in the section “[Power Calculations](#),” [p. 639](#) in the appendix “[Statistical Details](#)” chapter.

Calculating retrospective power at the actual sample size and estimated effect size is somewhat non-informative, even controversial [Hoenig and Heisey, 2001]. Certainly, it doesn't give additional information to the significance test, but rather shows the test in just a different perspective. However we believe that many studies fail due to insufficient sample size to detect a meaningful effect size, and there should be some facility to help guide for the next study, for specified effect sizes and sample sizes.

For more information, see John M. Hoenig and Dennis M. Heisey, (2001) “The Abuse of Power: The Pervasive Fallacy of Power Calculations for Data Analysis.”, *American Statistician* (v55 No 1, 19-24).

Power commands are available only for continuous-response models. Power and other test details are available in the following contexts:

- If you want the 0.05 level details for all parameter estimates, use the **Parameter Power** command in the **Estimates** menu. This produces the LSV, LSN, and adjusted power for an alpha of 0.05 for each parameter in the linear model.
- If you want the details for an *F*-test for a certain effect, find the **Power Analysis** command in the popup menu beneath the effect details for that effect.
- If you want the details for a Contrast, create the contrast from the popup menu next to the effect's title and select **Power Analysis** in the popup menu next to the contrast.
- If you want the details for a custom test, first create the test you want with the **Custom Test** command from the platform popup menu and then select **Power Analysis** command in the popup menu next to the Custom Test.

In all cases except the first, a Power Analysis dialog lets you enter information for the calculations you want.

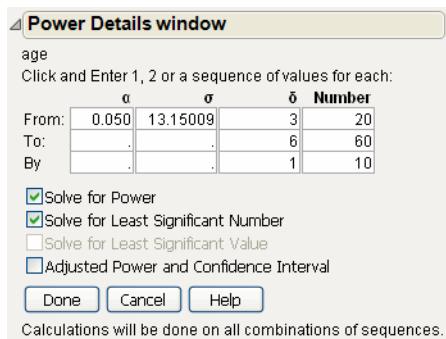
## The Power Analysis Dialog

The Power Analysis dialog (Figure 3.16) displays the contexts and options for test detailing. You fill in the values as described next, and then click **Done**. The results are appended at the end of the report. To create the dialog shown in Figure 3.16, select Power Analysis on the **age** red triangle menu.

### Example of Power Analysis

1. Open the Big Class.jmp sample data table.
2. Select **Analyze > Fit Model**.
3. Select **weight** and click **Y**.
4. Add **age**, **sex**, and **height** as the effects.
5. Click **Run**.
6. From the red triangle next to **age**, select **Power Analysis**.

**Figure 3.16** Power Analysis Dialog



7. Replace the delta values with 3, 6, and 1 as shown in Figure 3.16.
- For details about these columns, see “[Power Details Columns](#),” p. 72.
8. Replace the Number values with 20, 60, and 10 as shown in Figure 3.16.
9. Select **Solve for Power** and **Solve for Least Significant Number**.
10. Click **Done**.

For details about the output window, see “[Text Reports for Power Analysis](#),” p. 74.

### Power Details Columns

For each of four columns **Alpha**, **Sigma**, **Delta**, and **Number**, you can fill in a single value, two values, or the start, stop, and increment for a sequence of values, as shown in the dialog in Figure 3.16. Power calculations are done on all possible combinations of the values you specify.

**Alpha ( $\alpha$ )** is the significance level, between 0 and 1, usually 0.05, 0.01, or 0.10. Initially, **Alpha** automatically has a value of 0.05. Click on one of the three positions to enter or edit one, two, or a sequence of values.

**Sigma ( $\sigma$ )** is the standard error of the residual error in the model. Initially, RMSE, the square root of the mean square error, is supplied here. Click on one of the three positions to enter or edit one, two, or a sequence of values.

**Delta ( $\delta$ )** is the raw effect size. See “[Effect Size](#),” p. 73, for details. The first position is initially set to the square root of the sum of squares for the hypothesis divided by  $n$ . Click on one of the three positions to enter or edit one, two, or a sequence of values.

**Number ( $n$ )** is the sample size. Initially, the actual sample size is in the first position. Click on one of the three positions to enter or edit one, two, or a sequence of values.

Click the following check boxes to request the results you want:

**Solve for Power** Check to solve for the power (the probability of a significant result) as a function of  $\alpha$ ,  $\sigma$ ,  $\delta$ , and  $n$ .

**Solve for Least Significant Number** Check to solve for the number of observations expected to be needed to achieve significance alpha given  $\alpha$ ,  $\sigma$ , and  $\delta$ .

**Solve for Least Significant Value** Check to solve for the value of the parameter or linear test that produces a  $p$ -value of alpha. This is a function of  $\alpha$ ,  $\sigma$ , and  $n$ . This feature is available only for one-degree-of-freedom tests and is used for individual parameters.

**Adjusted Power and Confidence Interval** To look at power retrospectively, you use estimates of the standard error and the test parameters. Adjusted power is the power calculated from a more unbiased estimate of the noncentrality parameter. The confidence interval for the adjusted power is based on the confidence interval for the noncentrality estimate. Adjusted power and confidence limits are computed only for the original  $\delta$ , because that is where the random variation is.

## Effect Size

The power is the probability that an  $F$  achieves its  $\alpha$ -critical value given a noncentrality parameter related to the hypothesis. The noncentrality parameter is zero when the null hypothesis is true—that is, when the effect size is zero. The noncentrality parameter  $\lambda$  can be factored into the three components that you specify in the JMP Power dialog as

$$\lambda = (n\delta^2)/\sigma^2.$$

Power increases with  $\lambda$ , which means that it increases with sample size  $n$  and raw effect size  $\delta$  and decreases with error variance  $\sigma^2$ . Some books (Cohen 1977) use standardized rather than raw Effect Size,  $\Delta = \delta/\sigma$ , which factors the noncentrality into two components  $\lambda = n\Delta^2$ .

Delta ( $\delta$ ) is initially set to the value implied by the square root of  $SSH/n$ , where  $SSH$  is the sum of squares for the hypothesis. If you use this estimate for delta, you might want to correct for bias by asking for the Adjusted Power.

In the special case for a balanced one-way layout with  $k$  levels

## Parameter Power

$$\delta^2 = \frac{\sum(\alpha_i - \bar{\alpha})^2}{k}$$

Because JMP uses parameters of the form

$$\beta_i = (\alpha_i - \bar{\alpha}) \text{ with } \beta_k = - \sum_{m=1}^{k-1} \alpha_m$$

the delta for a two-level balanced layout is

$$\delta^2 = \frac{\beta_1^2 + (-\beta_1)^2}{2} = \beta_1^2$$

## Text Reports for Power Analysis

The power analysis facility calculates power as a function of every combination of  $\alpha$ ,  $\sigma$ ,  $\delta$ , and  $n$  values you specify in the Power Analysis Dialog.

- For every combination of  $\alpha$ ,  $\sigma$ , and  $\delta$  in the Power Analysis dialog, it calculates the least significant number.
- For every combination of  $\alpha$ ,  $\sigma$ , and  $n$  it calculates the least significant value.

For example, if you run the request shown in Figure 3.16, you get the tables shown in Figure 3.17.

**Figure 3.17** The Power Analysis Tables

The screenshot shows two tables generated by JMP's Power Analysis feature. The first table, titled 'Power Details', displays a grid of data for various combinations of parameters. The second table, titled 'Least Significant Number', lists the calculated LSN for each combination.

Power Details				
Test size				
Power				
$\alpha$	$\sigma$	$\delta$	Number	Power
0.0500	13.15009	3	20	0.0828
0.0500	13.15009	3	30	0.1117
0.0500	13.15009	3	40	0.1426
0.0500	13.15009	3	50	0.1755
0.0500	13.15009	3	60	0.2099
0.0500	13.15009	4	20	0.1117
0.0500	13.15009	4	30	0.1694
0.0500	13.15009	4	40	0.2317
0.0500	13.15009	4	50	0.2969
0.0500	13.15009	4	60	0.3630
0.0500	13.15009	5	20	0.1524
0.0500	13.15009	5	30	0.2515
0.0500	13.15009	5	40	0.3554
0.0500	13.15009	5	50	0.4575
0.0500	13.15009	5	60	0.5529
0.0500	13.15009	6	20	0.2063
0.0500	13.15009	6	30	0.3572
0.0500	13.15009	6	40	0.5035
0.0500	13.15009	6	50	0.6320
0.0500	13.15009	6	60	0.7368

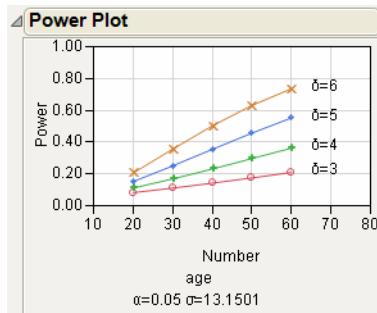
Least Significant Number				
$\alpha$	$\sigma$	$\delta$	Number(LSN)	
0.0500	13.15009	3	216.8578	
0.0500	13.15009	4	123.8892	
0.0500	13.15009	5	80.93391	
0.0500	13.15009	6	57.6814	

If you check **Adjusted Power and Confidence Interval** in the Power Analysis dialog, the Power report includes the **AdjPower**, **LowerCL**, and **UpperCL** columns.

## Plot of Power by Sample Size

The red triangle menu (shown in Figure 3.17) located at the bottom of the Power report gives you the command **Power Plot**, which plots the Power by N columns from the Power table. The plot in Figure 3.18 shows the result when you plot the example table in Figure 3.17. This plot can be enhanced with horizontal and vertical grid lines on the major tick marks, as shown. Double-click on each axis and add grid lines for the major tick marks.

**Figure 3.18** Plot of Power by Sample Size



## The Least Significant Value (LSV)

After a single-degree of freedom hypothesis test is performed, you often want to know how sensitive the test was. Said another way, you want to know how small an effect would be declared significant at some  $p$ -value alpha. The LSV provides a significance measuring stick on the scale of the parameter, rather than on a probability scale. It shows how sensitive the design and data are. It encourages proper statistical intuition concerning the null hypothesis by highlighting how small a value would be detected as significant by the data.

- The LSV is the value that the parameter must be greater than or equal to in absolute value to give the  $p$ -value of the significance test a value less than or equal to alpha.
- The LSV is the radius of the confidence interval for the parameter. A 1-alpha confidence interval is derived by taking the parameter estimate plus or minus the LSV.
- The absolute value of the parameter or function of the parameters tested is equal to the LSV, if and only if the  $p$ -value for its significance test is exactly alpha.

Compare the absolute value of the parameter estimate to the LSV. If the absolute parameter estimate is bigger, it is significantly different from zero. If the LSV is bigger, the parameter is not significantly different from zero.

## The Least Significant Number (LSN)

The LSN or *least significant number* is defined to be the number of observations needed to drive down the variance of the estimates enough to achieve a significant result with the given values of alpha, sigma, and delta (the significance level, the standard deviation of the error, and the effect size, respectively). If you need more data points (a larger sample size) to achieve significance, the LSN helps tell you how many more.

**Note:** LSN is not a recommendation of how large a sample to take because the probability of significance (power) is only about 0.5 at the LSN.

The LSN has these characteristics:

- If the LSN is less than the actual sample size  $n$ , then the effect is significant. This means that you have more data than you need to detect the significance at the given alpha level.
- If the LSN is greater than  $n$ , the effect is not significant. In this case, if you believe that more data will show the same standard errors and structural results as the current sample, the LSN suggests how much data you would need to achieve significance.
- If the LSN is equal to  $n$ , then the  $p$ -value is equal to the significance level alpha. The test is on the border of significance.
- Power (described next) calculated when  $n = \text{LSN}$  is always greater than or equal to 0.5.

## The Power

The power is the probability of getting a significant result. It is a function of the sample size  $n$ , the effect size  $\delta$ , the standard deviation of the error  $\sigma$ , and the significance level  $\alpha$ . The power tells you how likely your experiment is to detect a difference at a given  $\alpha$ -level. Power has the following characteristics:

- If the true value of the parameter is the hypothesized value, the power should be alpha, the size of the test. You do not want to reject the hypothesis when it is true.
- If the true value of the parameters is not the hypothesized value, you want the power to be as great as possible.
- The power increases with the sample size. The power increases as the error variance decreases. The power increases as the true parameter gets farther from the hypothesized value.

## The Adjusted Power and Confidence Intervals

Because power is a function of population quantities that are not known, it is usual practice to substitute sample estimates in power calculations (Wright and O'Brien 1988). If you regard these sample estimates as random, you can adjust them to have a more proper expectation. You can also construct a confidence interval for this adjusted power. However, the confidence interval is often very wide. The adjusted power and confidence intervals can be computed only for the original  $\delta$  because that is where the random variation is. For details about adjusted power see "[Computations for Adjusted Power](#)," p. 640 in the appendix "Statistical Details" chapter.

## Prospective Power Analysis

Prospective analysis helps you answer the question, “Will I detect the group differences I am looking for given my proposed sample size, estimate of within-group variance, and alpha level?” In a prospective power analysis, you must provide estimates of the group means and sample sizes in a data table and an estimate of the within-group standard deviation  $\sigma$  in the Power Analysis dialog.

The **Sample Size and Power** command found on the **DOE** menu offers a facility that computes power, sample size, or the effect size you want to detect, for a given alpha and error standard deviation. You supply two of these values and the Sample Size and Power feature computes the third. If you supply only one of these values, the result is a plot of the other two. This feature assumes equal sample sizes. For unequal sample sizes, you can do as shown in the following example.

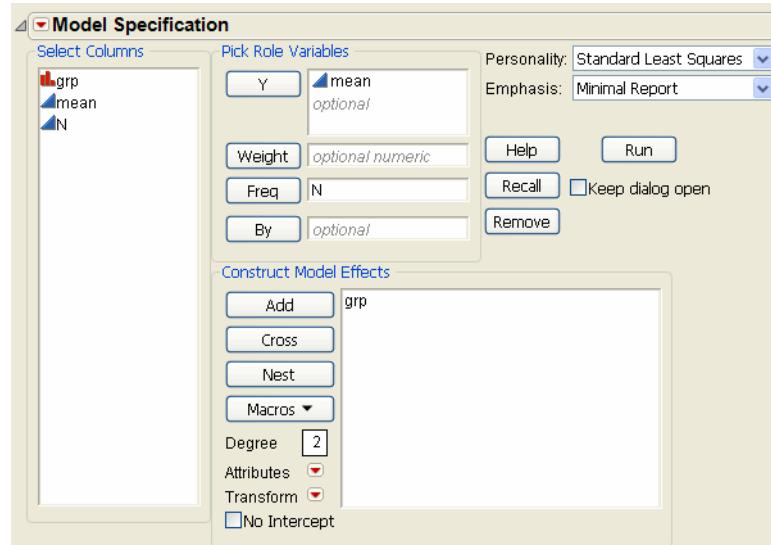
This example is from Wright and O’Brien (1988). Dr. Noah Decay, a dentist, is planning an experiment in which he wishes to compare two treatments that are believed to reduce dental plaque. Dr. Decay believes that the means for the control group and two treatments are 32, 26, and 24, respectively, and that the within-group standard deviation is between 6 and 9. The control group has twice as many patients as each treatment group.

To run a prospective power analysis on the test for the group effect, Dr. Decay must first create the data table with three columns that contain the group names, group means, and proposed group sizes. This example uses the **Noah Decay.jmp** data table.

Dr. Decay uses the Fit Model platform to set up a one-way analysis of variance (as shown here). Note that the sample size variable, N, is declared as **Freq**. Also, the **Minimal Reports** emphasis option is selected.

---

**Figure 3.19** Fit Model Dialog with Noah Decay.jmp variables

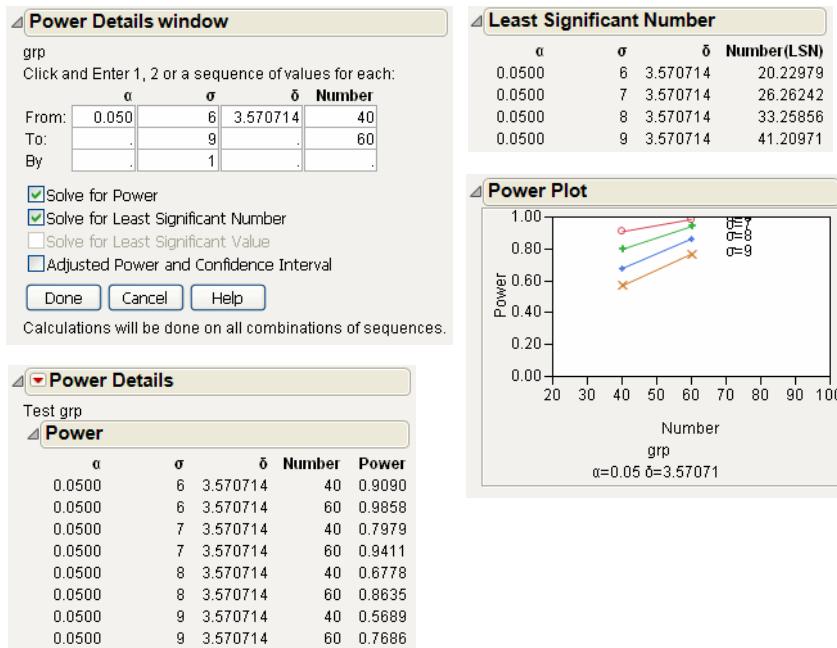


## Correlation of Estimates

The analysis yields zero for the sum of squares error and mean square error. However, Dr. Decay suspects that the standard deviation is between 6 and 9. He now chooses **Power Analysis** from the grp report popup menu beneath Effect Details and checks **Solve for Power** and **Solve for Least Significant Number**. Then, the Power Analysis dialog as shown to the left in Figure 3.20 gives calculations for all combinations of alpha = 0.05, sample sizes of 40 to 60, and standard deviations (sigma) of 6 to 9 by 1.

The **Delta** shown was calculated from the specified mean values. A prospective power analysis uses population values, which give an unbiased noncentrality parameter value, so adjusted power and confidence limits are not relevant.

**Figure 3.20** Power Details and Power by N Plot for Prospective Analysis



## Correlation of Estimates

The **Correlation of Estimates** option on the **Estimates** menu produces a correlation matrix for all parameter estimates. The example shown here uses the **Tiretread.jmp** data table with **ABRASION** as the response, two factors **SILICA** and **SILANE**, and their interaction.

**Figure 3.21** Correlation of Estimates Report

Correlation of Estimates				
Corr		Intercept	SILICA	SILANE (SILICA-1.2)*(SILANE-50)
Intercept	1.0000	-0.4281	-0.8919	0.0000
SILICA	-0.4281	1.0000	-0.0000	0.0000
SILANE	-0.8919	-0.0000	1.0000	-0.0000
(SILICA-1.2)*(SILANE-50)	0.0000	0.0000	-0.0000	1.0000

## Effect Screening

The **Effect Screening** commands help examine the sizes of the effects. The **Scaled Estimates** command was covered earlier under “[Scaled Estimates and the Coding Of Continuous Terms](#),” p. 62, which dealt with scaling issues. The other **Effect Screening** commands are discussed here. These commands correct for scaling and for correlations among the estimates. The features of these **Effect Screening** commands are derived by noticing three things:

1. The process of fitting can be thought of as converting one set of realizations of random values (the response values) into another set of realizations of random values (the parameter estimates). If the design is balanced with an equal number of levels per factor, these estimates are independent and identically distributed, just as the responses are.
2. If you are fitting a screening design with many effects and only a few runs, you expect that only a few effects are active. That is, a few effects have sizable impact and the rest of them are inactive (they are estimating zeroes). This is called the assumption of *effect sparsity*.
3. Given points 1 and 2 above, you can think of screening as a way to determine which effects are inactive with random values around zero and which ones are outliers, not part of the distribution of inactive effects.

Thus, you treat the estimates themselves as a set of data to help you judge which effects are active and which are inactive. If there are few runs, with little or no degrees of freedom for error, then there are no classical significance tests, and this approach is especially needed.

The last three **Effect Screening** commands look at parameters in this way. There are two default reports, Lenth's PSE and the Parameter Estimates Population table with significant factors highlighted. The **Effect Screening** command also has a submenu for looking at the model parameters from different angles using scaled estimates and three plots.

## Lenth's Method

An estimate of standard error is calculated using the method of Lenth (1989) and shows in the Effect Screening table (shown above). This estimate, called the *pseudo standard error*, is formed by taking 1.5 times the median absolute value of the estimates after removing all the estimates greater than 3.75 times the median absolute estimate in the complete set of estimates.

## Parameter Estimates Population

Most inferences about effect size first assume that the estimates are uncorrelated and have equal variances. This is true for fractional factorials and many classical experimental designs. However, for some designs it is not true. The last three **Effect Screening** commands display the Parameter Estimate Population report, which first finds the correlation of the estimates and tells you whether or not the estimates are uncorrelated and have equal variances.

If the estimates are correlated, a normalizing transformation can be applied to make them uncorrelated and have equal variances.

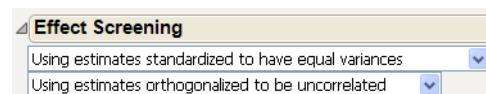
If the estimates are uncorrelated and have equal variances, then the following notes appear and the analysis is straightforward.

- The parameter estimates have equal variances
- The parameter estimates are not correlated

If the estimates are correlated and/or have unequal variances, then each of these two notes may change into a popup menu showing that it has transformed the estimates, and giving you the option to undo the transformation.

---

**Figure 3.22** Options for Transformed Estimates

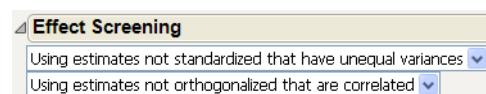


---

If you click these popup menus and undo both transformations, the lines look like this:

---

**Figure 3.23** Notation for Non-Transformed Parameter Estimates



The transformation to make the estimates uncorrelated is the same as that used to calculate sequential sums of squares. The estimates measure the additional contribution of the variable after all previous variables have been entered into the model. An example of a data set with unequal variances and correlated estimates is *Tiretread.jmp*. Using *Tiretread.jmp* with **ABRASION** as *Y* and using the four factors **HARDNESS**, **SILICA**, **SILANE**, and **SULFUR**, with all two-factor interactions included in the model gives the following Parameter Estimate Population table when selecting **Effect Screening > Normal Plot** from the red-triangle menu of the response.

**Figure 3.24** Parameter Estimate table for Tiretread.jmp

Parameter Estimate Population						
Term	Estimate	t Ratio	Orthog		Orthog	
			Coded	t-Ratio	Prob> t	
Intercept	13.6688	0.2024	133.100	114.9700	<.0001*	
HARDNESS	-1.0934	-0.7270	9.360	8.0850	<.0001*	
SILICA	25.5220	6.9519	16.509	14.2607	<.0001*	
SILANE	2.3180	4.0679	6.240	5.3899	0.0004*	
SULFUR	23.6908	4.2235	9.375	8.0981	<.0001*	
(HARDNESS-69.775)*(SILICA-1.2)	1.6930	0.7271	4.411	3.8103	0.0042*	
(HARDNESS-69.775)*(SILANE-50)	-0.0995	-2.7024	-1.517	-1.3103	0.2226	
(HARDNESS-69.775)*(SULFUR-2.3)	-2.9925	-2.2032	2.135	1.8440	0.0983	
(SILICA-1.2)*(SILANE-50)	-0.0965	-0.1100	-3.545	-3.0625	0.0135*	
(SILICA-1.2)*(SULFUR-2.3)	20.7171	2.0714	2.062	1.7811	0.1086	
(SILANE-50)*(SULFUR-2.3)	3.0922	3.9573	4.581	3.9573	0.0033*	

Each Orthog Estimate is conditioned on the effects before it

An example of a data set with equal variances and uncorrelated estimates is Reactor.jmp. The analysis for Reactor.jmp data uses four factors, Ct, A, T, and Cn, with all two-factor interactions included in the model.

The Parameter Estimate Population table for Reactor.jmp is shown below.

**Figure 3.25** Parameter Estimate table for Reactor.jmp

Parameter Estimate Population			
Term	Estimate	t Ratio	Prob> t
Intercept	65.5000	119.6073	<.0001*
Ct	9.7500	17.8041	<.0001*
A	-0.3125	-0.5706	0.5740
T	5.3750	9.8151	<.0001*
Cn	-3.1250	-5.7065	<.0001*
Ct*A	0.4375	0.7989	0.4329
Ct*T	6.6250	12.0977	<.0001*
Ct*Cn	1.0000	1.8261	0.0814
A*T	1.0625	1.9402	0.0653
T*Cn	-5.5000	-10.0434	<.0001*

The Parameter Estimate Population tables shown above contain some of the following columns, which are dependent upon whether or not the estimates have equal variances and whether or not the estimates are correlated:

**Estimate** lists the parameter estimates for the fitted linear model. These estimates can be compared in size only if the  $X$  variables are scaled the same.

**t Ratio** is the t test associated with this parameter.

**Orthog Coded** contains the orthogonalized estimates. They are used in the Pareto plot because this plot partitions the sum of the effects, which requires orthogonality. The orthogonalized values are

## Effect Screening

computed by premultiplying the column vector of the Original estimates by the Cholesky root of  $X'X$ . These estimates depend on the model order unless the original design is orthogonal.

If the design was orthogonal and balanced, then these estimates will be identical to the original estimates. If they are not, then each effect's contribution is measured after it is made orthogonal to the effects before it.

**Orthog t-Ratio** lists the parameter estimates after a transformation that makes them independent and identically distributed. These values are used in the Normal Plot (discussed next), which requires uncorrelated estimates of equal variance. The  $p$ -values associated with Orthog  $t$ -ratio estimates (given in the Prob>|t| column) are equivalent to Type I sequential tests. This means that if the parameters of the model are correlated, the estimates and their  $p$ -values depend on the order of terms in the model. The Orthog  $t$ -ratio estimates are computed by dividing the orthogonalized estimates by their standard errors. The Orthog  $t$ -ratio values let JMP treat the estimates as if they were from a random sample for use in Normal plots or Bayes plots.

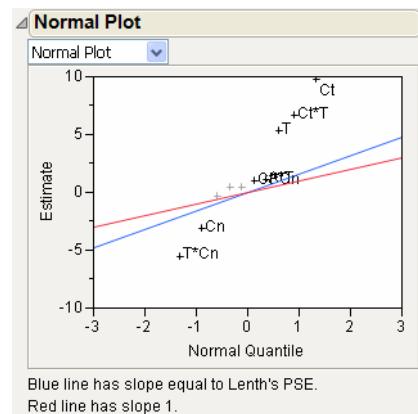
**Prob>|t|** is the significance level or  $p$ -value associated with the values in the Orthog t-Ratio column.

## Normal Plot

The **Normal Plot** command displays the Parameter Estimates Population table (discussed in the previous section) and shows a normal plot of these parameter estimates (Daniel 1959). The estimates are on the vertical axis and the normal quantiles on the horizontal axis. The normal plot for  $Y$  in the **Reactor.jmp** screening model is shown in Figure 3.26. If all effects are due to random noise, they tend to follow a straight line with slope  $\sigma$ , the standard error. The line with slope equal to the Lenth's PSE estimate is shown in blue.

The Normal plot helps you pick out effects that deviate from the normal lines. Estimates that deviate substantially are labeled. In this example, The A factor does not appear important. But not only are the other three factors (T, Ct, and Cn) active, T (temperature) appears to interact with both Ct and Cn.

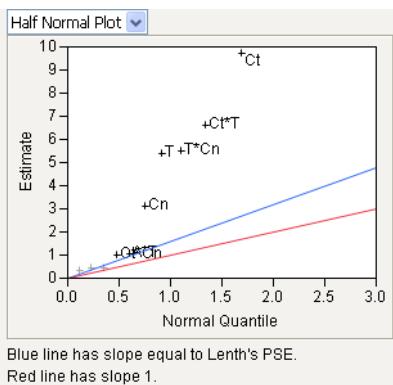
**Figure 3.26** Normal Plot for the Reactor Data with Two-Factor Interactions



## Half-Normal Plot

At the top of the Normal plot is a popup menu. Click this and select **Half Normal Plot** to obtain a plot of the absolute values of the estimates against the normal quantiles for the absolute value normal distribution, as shown in Figure 3.27. Some analysts prefer this to the Normal Plot.

**Figure 3.27** Half Normal Plot

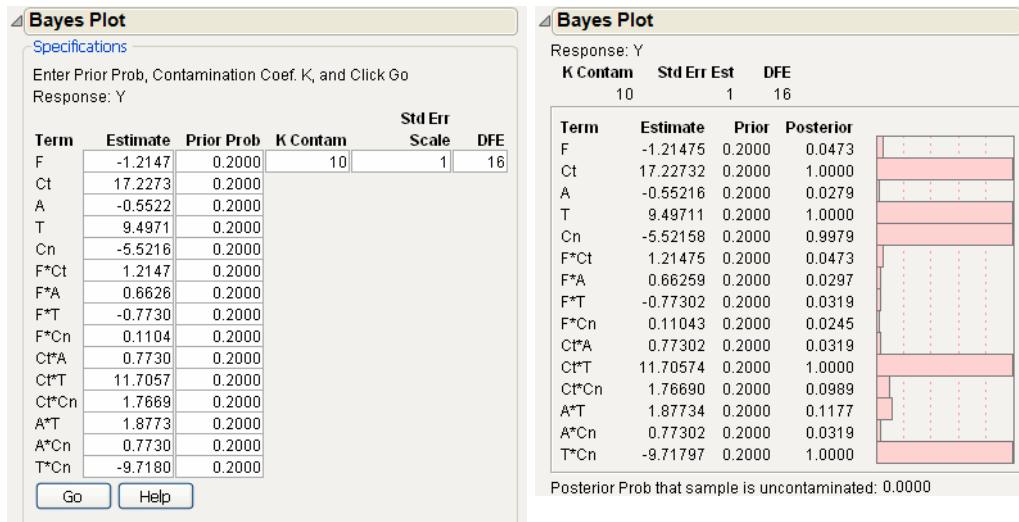


## Bayes Plot

Another approach to resolving which effects are important (sometimes referred to as which contrasts are active) is to compute posterior probabilities using a Bayesian approach. This method, due to Box and Meyer (1986), assumes that the estimates are a mixture from two distributions. Some portion of the effects is assumed to come from pure random noise with a small variance. The remaining terms are assumed to come from a *contaminating* distribution that has a variance  $K$  times larger than the error variance.

The prior probability for an effect is the chance you give that effect of being nonzero, (or being in the contaminating distribution). These priors are usually set to equal values for each effect, and 0.2 is a commonly recommended prior probability value. The  $K$  contamination coefficient is often set at 10, which says the contaminating distribution has a variance that is 10 times the error variance.

The Bayes plot is done with respect to normalized estimates (JMP lists as Orthog  $t$ -Ratio), which have been transformed to be uncorrelated and have equal variance. To see a Box-Meyer Bayesian analysis for the Reactor.jmp example, specify  $Y$  as  $Y$ , and **ADD F, Ct, A, T, and Cn** with all two-factor interactions included in the model. After running the model, select **Bayes Plot** from the **Effect Screening** menu. The dialog panel, shown to the left in Figure 3.28, asks you to fill in the prior probabilities and the  $K$  coefficient. You can also edit the other values. It is not necessary to have a nonzero DFE. Click **Go** to start the calculation of the posterior probabilities.

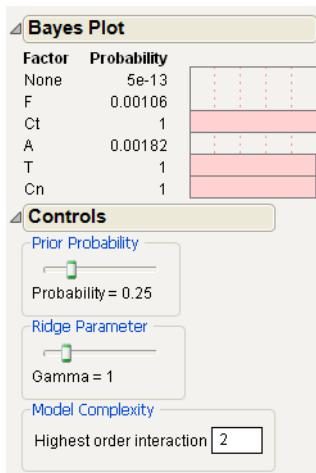
**Figure 3.28** Bayes Plot Dialog and Plot

The **Std Err Scale** field is set to 0 for a saturated model with no estimate of error. If there is an estimate of standard error (the root mean square error), this field is set to 1 because the estimates have already been transformed and scaled to unit variance. If you edit the field to specify a different value, it should be done as a scale factor of the RMSE estimate.

The resulting posterior probabilities are listed and plotted with bars as shown in Figure 3.28. The literature refers to this as the Bayes Plot. An overall posterior probability is also listed for the outcome that the sample is uncontaminated.

## Bayes Plot for Factor Activity

JMP includes a script that allows you to determine which factors are active in the design. Found in the Sample Scripts folder, BayesPlotforFactors.jsl can be used to produce the Factor Activity plot. Open and run the script (**Edit > Run Script**) on the Reactor.jmp data, using Y as *Y* and F, Ct, A, T, and Cn as factors. This produces the following plot.

**Figure 3.29** Bayes Plot for Factor Activity

In this case, we have specified that the highest order interaction to consider is two. Therefore, all possible models that include (up to) second order interactions are constructed, and, based on the value assigned to Prior Probability (see the Controls section of the plot), a posterior probability is computed for each of the possible models. The probability for a factor is the sum of the probabilities for each of the models where it was involved.

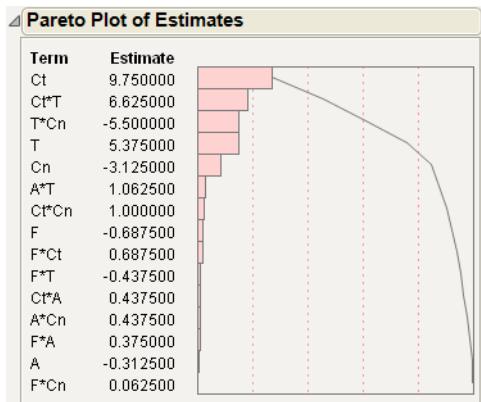
In this example, we see that Ct, T, and Cn are active and that A and F are not. This agrees with Figure 3.28.

**Note:** If the ridge parameter were zero (not allowed), all the models would be fit by least-squares. As the ridge parameter gets large the parameter estimates for any model shrink towards zero. Details on the ridge parameter (and why it cannot be zero) are explained in Box and Meyer (1993).

## Pareto Plot

The **Pareto Plot** selection gives plots of the absolute values of the orthogonalized estimates showing their composition relative to the sum of the absolute values. The estimates are orthogonalized to be uncorrelated and standardized to have equal variances by default. If your data set has estimates that are correlated and/or have unequal variances, then your data is transformed, by default, to have equal variances and to be uncorrelated. However, you have the option of undoing the transformations. (See “[Parameter Estimates Population](#),” p. 80.) In this case, the Pareto Plot represents your selection of equal variances/unequal variances and uncorrelated/correlated estimates.

Figure 3.30 shows a Pareto Plot for the sample data **Reactor.jmp**. For this data set, the estimates have equal variances and are not correlated.

**Figure 3.30** Pareto Plot for Reactor.jmp

# Chapter 4

## Standard Least Squares: Exploring the Prediction Equation The Fit Model Platform

---

Assuming that the prediction equation is estimated well, there is still work in exploring the equation itself to answer a number of questions:

- What kind of curvature does the response surface have?
- What are the predicted values at the corners of the factor space?
- Would a transformation on the response produce a better fit?

The tools described in this chapter explore the prediction equation to answer these questions in the context of assuming that the equation is correct enough to work with.

# Contents

Exploring the Prediction Equation .....	89
The Profiler.....	89
Contour Profiler .....	90
Mixture Profiler.....	91
Surface Profiler .....	92
Interaction Plots .....	93
Cube Plots.....	94
Response Surface Curvature .....	95
Parameter Estimates .....	96
Canonical Curvature Table .....	97
Box Cox Y Transformations.....	98

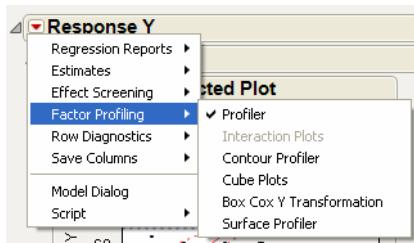
---

## Exploring the Prediction Equation

Figure 4.1 illustrates the relevant commands available for the least squares fitting personality.

---

**Figure 4.1** Commands and Options for a Least Squares Analysis



Here is a quick description of each feature to help you decide which section you need.

**Profiler, Contour Profiler, Mixture Profiler, and Surface Profiler** display cuts through response surfaces for analyzing and optimizing a model. For complete details on using the Profilers, see the “Profiling” chapter.

**Interaction plots** are multiple profile plots across one factor under different settings of another factor. The traces are not parallel when there is a sizable interaction.

**Cube Plots** show predicted values in the corners of the factor space.

**Box Cox Y Transformation** finds a power transformation of the response that fits best.

---

## The Profiler

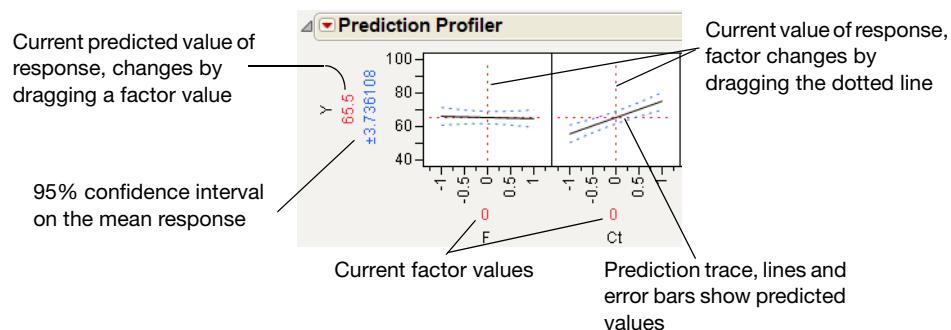
The **Profiler** displays prediction traces (see Figure 4.2) for each  $X$  variable. The vertical dotted line for each  $X$  variable shows its *current value* or *current setting*.

If the variable is nominal, the  $x$ -axis identifies categories.

For each  $X$  variable, the value above the factor name is its current value. You change the current value by clicking in the graph or by dragging the dotted line where you want the new current value to be.

- The horizontal dotted line shows the *current predicted value* of each  $Y$  variable for the current values of the  $X$  variables.
- The black lines within the plots show how the predicted value changes when you change the current value of an individual  $X$  variable. The 95% confidence interval for the predicted values is shown by a dotted curve surrounding the prediction trace (for continuous variables) or an error bar (for categorical variables).

The Prediction Profiler is a way of changing one variable at a time and looking at the effect on the predicted response.

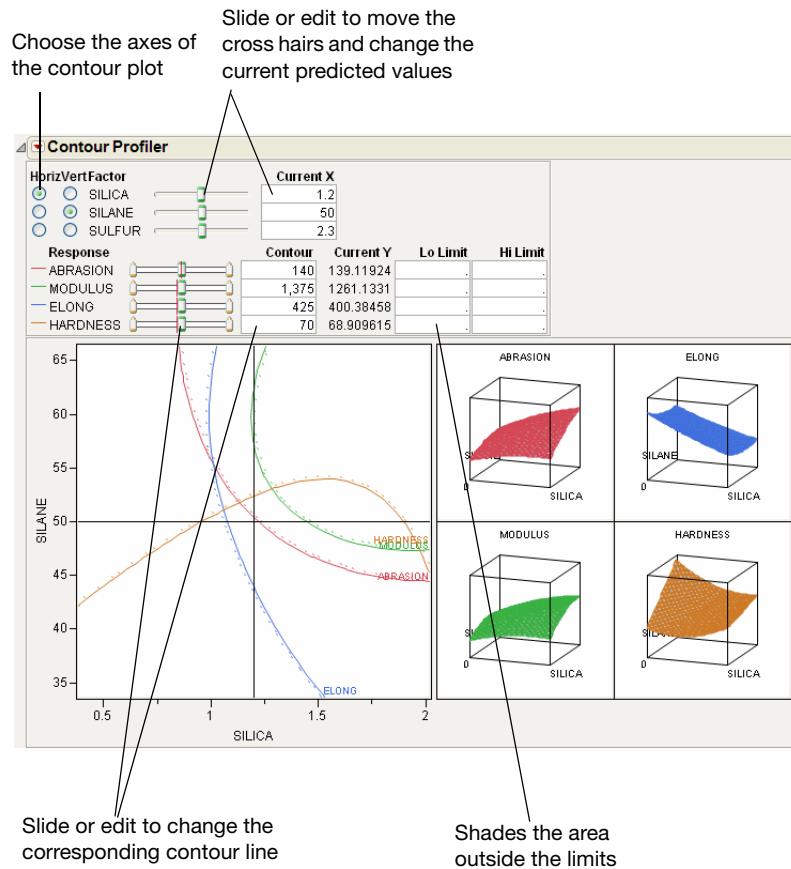
**Figure 4.2** Illustration of Prediction Traces

For details on using the profiler, see the “[Profiling](#)” chapter.

## Contour Profiler

The **Contour Profiler** option in the **Factor Profiling** submenu brings up an interactive contour profiling facility. This is useful for optimizing response surfaces graphically. To use the Contour Profiler:

1. Select Fit Model for Tiretread.jmp.
2. Specify ABRASION, MODULUS, ELONG, and HARDNESS as *Y*.
3. Select SILICA, SILANE, and SULFUR for *X* and select Response Surface from the Macros drop-down menu.
4. Click **Run**.
5. Select **Profilers > Contour Profiler** from the red-triangle menu of the Least Squares Fit report. (See Figure 4.3.)

**Figure 4.3** Contour Profiler for Response Surface Model

Details on the Contour profiler are found in the “[Contour Profiler](#),” p. 555 in the “Profiling” chapter.

## Mixture Profiler

The Mixture Profiler shows response contours of mixture experiment models on a ternary plot. This feature is useful when three or more factors in the experiment are components in a mixture. The Mixture Profiler allows you to visualize and optimize the response surfaces of your experiment.

Figure 4.4 shows an example of a mixture experiment from *Plasticizer.jmp*. To use the Mixture Profiler:

1. Select **Analyze > Fit Model**. (The response and predictor variables for the sample data, *Plasticizer.jmp*, should be automatically populated.)
2. For other data sets, select an appropriate response variable for *Y*.

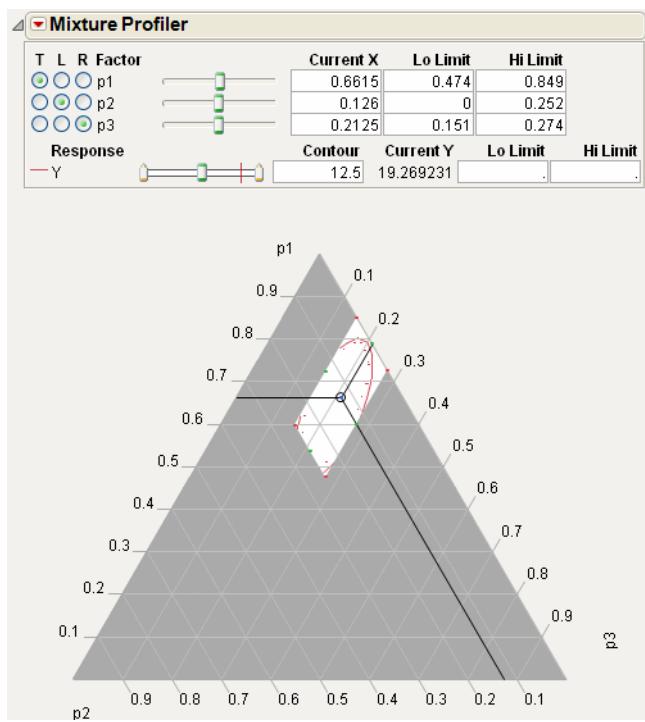
## Surface Profiler

3. Select appropriate  $X$  variables and click Macros > Mixture Response Surface.
4. Click Run.
5. From the red-triangle menu of the response, select **Factor Profiling > Mixture Profiler**.

The radio buttons at the top left of the plot may be used to modify plot axes for the factors and the Lo and Hi Limit columns at the upper right of the plot allow you to enter constraints for both the factors and the response.

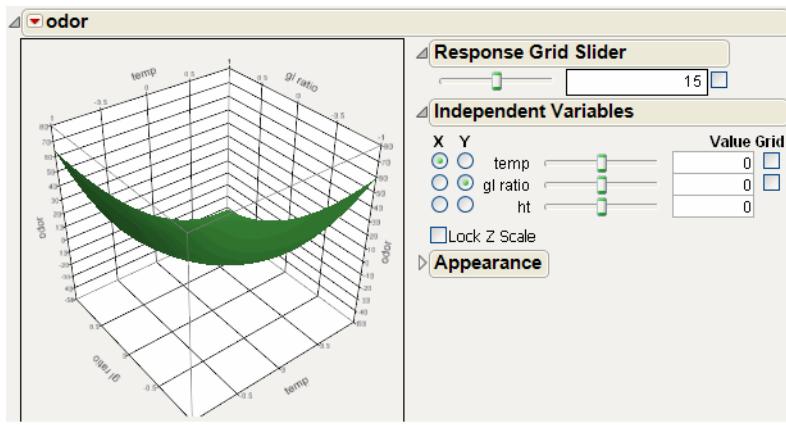
For a detailed explanation of the Mixture Profiler, see “[Mixture Profiler](#),” p. 557 in the “Profiling” chapter.

**Figure 4.4** Mixture Profiler for Mixture Response Surface



## Surface Profiler

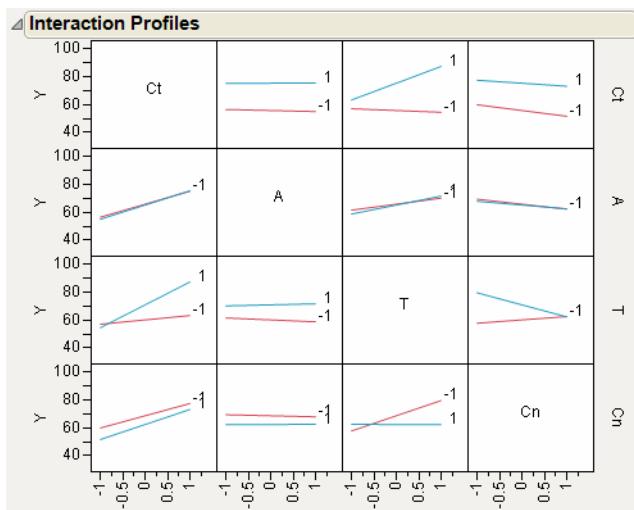
The Surface Profiler shows a three-dimensional surface plot of the response surface. Open the Odor.jmp data table. Run the attached Model script and click Run on the Fit Model Dialog. Select **Factor Profiling > Surface Profiler** to obtain Figure 4.5. Details of Surface Plots are found in the “[Plotting Surfaces](#)” chapter.

**Figure 4.5** Surface Plot

## Interaction Plots

The **Interaction Plots** selection in the **Factor Profiling** submenu shows a matrix of interaction plots when there are interaction effects in the model. As an example, use the **Reactor.jmp** sample data with factors Ct, A, T, and Cn, and all two-factor interactions. You can then see the interaction effects with the **Interaction Plots** command.

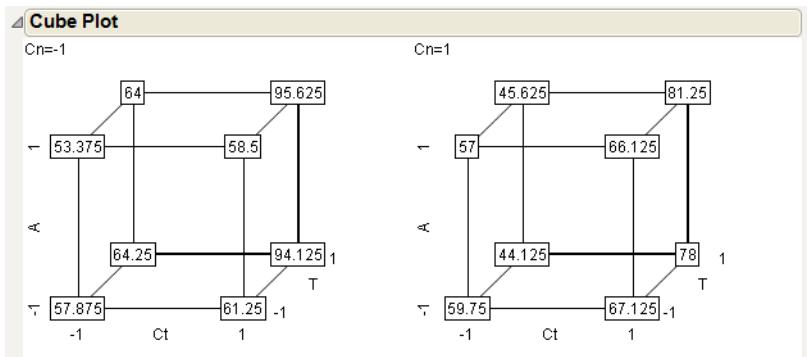
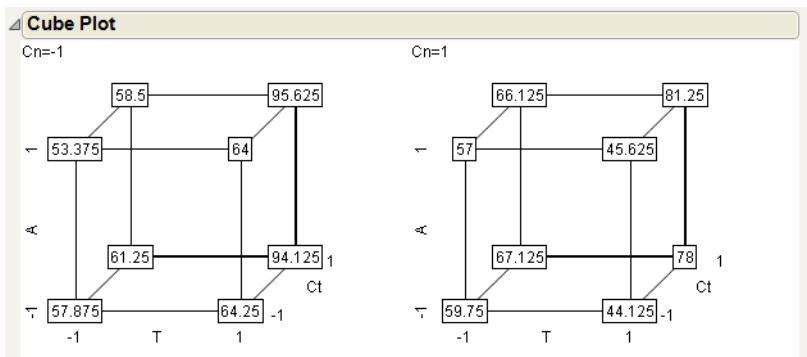
In an interaction plot, evidence of interaction shows as nonparallel lines. For example, in the T\*Cn plot in the bottom row of plots (Figure 4.6) the effect of Cn is very small at the low values of temperature, but it diverges widely for the high values of temperature.

**Figure 4.6** Interaction Plots for the Reactor Example

## Cube Plots

The **Cube Plots** option in the **Factor Profiling** submenu displays a set of predicted values for the extremes of the factor ranges, laid out on the vertices of cubes, as illustrated in Figure 4.7. If a factor is nominal, the vertices are the first and last level.

If you want to change the layout so that the factors are mapped to different cube coordinates, click one of the factor names in the first cube and drag it to the desired axis. For example, if you click T, as shown in Figure 4.7, and drag it over Ct, then T and Ct exchange places, along with their corresponding coordinates, as shown in Figure 4.8. If there is more than one response, the multiple responses are shown stacked at each vertex.

**Figure 4.7** Cube Plots**Figure 4.8** Cube Plot after Exchange

## Response Surface Curvature

Often in industrial experiments, the goal is to find values for one or more factors that maximize or minimize the response. JMP provides surface modeling with special reports that show the critical values, the surface curvature, and a response contour plot.

The **Response Surface** selection in the **Effect Macros** popup menu automatically constructs all the linear, quadratic, and cross product terms needed for a response surface model.

The same model can be specified using the **Add** and **Cross** buttons to create model effects in the Fit Model dialog. You then select a model term and assign it the **Response Surface** effect attribute found in the **Attributes** popup menu. The response surface effects show with &RS after their name in the **Construct Model Effects** list, as shown in Figure 4.9.

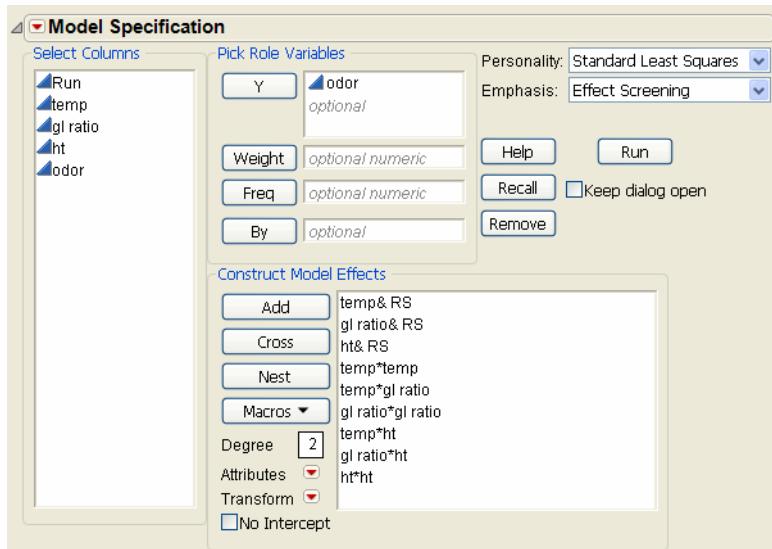
## Response Surface Curvature

**Note:** Curvature analysis is not shown for response surface designs of more than 20 factors in the Fit Model platform. No error message or alert is given. All other analyses contained within the report are valid and are shown. For additional information on response surface designs, see the Design of Experiments Guide Guide, Chapter 5, Response Surface Designs.

After the parameters are estimated, critical values for the factors in the estimated surface can be found. Critical points occur at either maximums, minimums, or saddle points in the surface. The eigenvalues and eigenvectors of the calculated quadratic parameters determine the type of curvature. The eigenvectors show the principal directions of the surface, including the directions of greatest and smallest curvature.

The following example uses the Odor Control Original.jmp file in the Sample Data folder. The example uses data discussed in John (1971). The objective is to find the range of temperature (temp), gas-liquid ratio (gl ratio), and height (ht) values that minimize the odor of a chemical production process. To run this response surface model, select the response variable, odor, from the variable selector list and click Y. Highlight the factors temp, gl ratio, and ht, and choose **Response Surface** from the **Macros** popup menu. Note that the main effects automatically appear in the **Construct Model Effects** list with &RS after their name (see Figure 4.9).

**Figure 4.9** Fit Model Dialog For Response Surface Analysis



## Parameter Estimates

The Parameter Estimates table shows estimates of the model parameters.

Parameter Estimates					
Term	Estimate	Std Error	t Ratio	Prob> t	
Intercept	-30.66667	12.97797	-2.36	0.0645	
temp	-12.125	7.947353	-1.53	0.1876	
gl ratio	-17	7.947353	-2.14	0.0854	
ht	-21.375	7.947353	-2.69	0.0433*	
temp*temp	32.083333	11.69819	2.74	0.0407*	
temp*gl ratio	8.25	11.23925	0.73	0.4959	
gl ratio*gl ratio	47.833333	11.69819	4.09	0.0095*	
temp*ht	1.5	11.23925	0.13	0.8990	
gl ratio*ht	-1.75	11.23925	-0.16	0.8824	
ht*ht	6.0833333	11.69819	0.52	0.6252	

If you use the **Prediction Formula** command in the **Save Columns** submenu, a new data table column called **Pred Formula odor** saves the prediction formula using the coefficients from the Parameter Estimates table.

The probability value of 0.0657 in the Analysis of Variance table indicates that the three-variable response surface model is only marginally better than the sample mean.

Analysis of Variance				
Source	DF	Sum of		
		Squares	Mean Square	F Ratio
Model	9	18,881.983	2,098.00	4.1521
Error	5	2,526.417	505.28	Prob > F
C. Total	14	21,408.400		0.0657

The response surface analysis also displays the Response Surface table and the Solution table (Figure 4.10). The Solution table shows the critical values for the surface variables and indicates that the surface solution point is a minimum for this example.

## Canonical Curvature Table

The Canonical Curvature report, found under the Response Surface title, shows the eigenstructure (Figure 4.10), which is useful for identifying the shape and orientation of the curvature, and results from the eigenvalue decomposition of the matrix of second-order parameter estimates. The eigenvalues (given in the first row of the Canonical Curvature table) are negative if the response surface shape curves back from a maximum. The eigenvalues are positive if the surface shape curves up from a minimum. If the eigenvalues are mixed, the surface is saddle shaped, curving up in one direction and down in another direction.

The eigenvectors listed beneath the eigenvalues show the orientations of the principal axes. In this example the eigenvalues are positive, which indicates that the curvature bends up from a minimum. The direction where the curvature is the greatest corresponds to the largest eigenvalue (48.8588) and the variable with the largest component of the associated eigenvector (gl ratio). The direction with the eigenvalue of 31.1035 is loaded more on temp.

Sometimes a zero eigenvalue occurs. This means that along the direction described by the corresponding eigenvector, the fitted surface is flat.

**Figure 4.10** Basic Reports for Response Surface Model

Response Surface				
Coef	temp	gl ratio	ht	odor
temp	32.083333	8.25	1.5	-12.125
gl ratio	.	47.833333	-1.75	-17
ht	.	.	6.0833333	-21.375

Solution	
Variable	Critical Value
temp	0.1219125
gl ratio	0.1995746
ht	1.7705249

Solution is a Minimum  
 Critical values outside data range  
 Predicted Value at Solution -52.02463

Canonical Curvature			
Eigenvalues and Eigenvectors			
Eigenvalue	48.8588	31.1035	6.0377
temp	0.23809	0.97070	-0.03259
gl ratio	0.97112	-0.23738	0.02413
ht	-0.01569	0.03740	0.99918

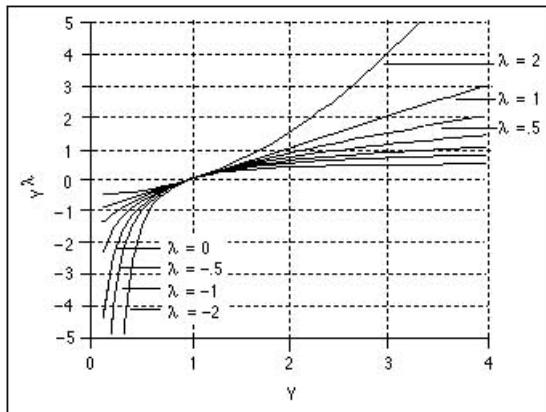
---

## Box Cox Y Transformations

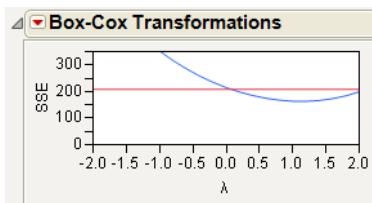
Sometimes a transformation on the response fits the model better than the original response. A commonly used transformation raises the response to some power. Box and Cox (1964) formalized and described this family of power transformations. The **Factor Profiling** menu has the **Box Cox Y Transformation** command. The formula for the transformation is constructed so that it provides a continuous definition and the error sums of squares are comparable.

$$Y^{(\lambda)} = \begin{cases} \frac{y - 1}{\lambda y} & \text{if } \lambda \neq 0 \\ \ln(y) & \text{if } \lambda = 0 \end{cases} \quad \text{where } \bar{y} \text{ is the geometric mean}$$

The plot shown here illustrates the effect of this family of power transformations on  $Y$ .



The **Box-Cox Y Transformation** command fits transformations from  $\lambda = -2$  to 2 in increments of 0.2, and it plots the sum of squares error (SSE) across the  $\lambda$  power. The plot below shows the best fit when  $\lambda$  is between 1.0 and 1.5 for the Reactor.jmp data using the model with effects F, Ct, A, T, and Cn and all two-factor interactions. The best transformation is found on the plot by finding the lowest point on the curve.



The Box-Cox Transformation red triangle menu has the following options:

The **Save Best Transformation** command creates a new column in the data table and saves the formula for the best transformation.

The **Save Specific Transformation** command behaves in the same way, but first prompts for a lambda value.

The **Table of Estimates** command creates a new data table containing parameter estimates and SSE for all values of  $\lambda$  from -2 to 2.



# Chapter 5

## Standard Least Squares: Random Effects

### The Fit Model Platform

---

Random effects are those where the effect levels are chosen randomly from a larger population of levels. These random effects represent a sample from the larger population. In contrast, the levels of fixed effects are of direct interest rather than representative. If you have both random and fixed (nonrandom) effects in a model, it is called a *mixed model*.

---

**Important:** It is very important to declare random effects. Otherwise, the test statistics produced from the fitted model are calculated with the wrong assumptions.

---

Typical random effects are

- subjects in a *repeated measures* experiment, where the subject is measured at several times.
- plots in a *split plot experiment*, where an experimental unit is subdivided and multiple measurements are taken, usually with another treatment applied to the subunits.
- *measurement* studies, where multiple measurements are taken in order to study measurement variation.
- random coefficients models, where the random effect is built with a continuous term crossed with categories.

The Fit Model platform in JMP fits mixed models using modern methods now generally regarded as best practice:

- REML estimation method (REstricted Maximum Likelihood)
- Kenward-Roger tests

For historical interest only, the platform also offers the Method of Moments (EMS), but this is no longer a recommended method except in special cases where it is equivalent to REML.

If you have a model where all effects are random, you can also fit it in the Variability Chart platform.

# Contents

Topics in Random Effects .....	io3
Introduction to Random Effects .....	io3
Generalizability .....	io4
The REML Method .....	io4
Unrestricted Parameterization for Variance Components in JMP .....	io4
Negative Variances. ....	io5
Random Effects BLUP Parameters .....	io5
REML and Traditional Methods Agree on the Standard Cases .....	io7
F-Tests in Mixed Models .....	io7
Specifying Random Effects .....	io8
Split Plot Example. ....	io8
The Model Dialog. ....	io9
REML Results.....	ii0
REML Save Menu. ....	iii
Method of Moments Results.....	ii2

---

## Topics in Random Effects

### Introduction to Random Effects

Levels in random effects are randomly selected from a larger population of levels. For the purpose of testing hypotheses, the distribution of the effect on the response over the levels is assumed to be normal, with mean zero and some variance (called a *variance component*).

In one sense, every model has at least one random effect, which is the effect that makes up the residual error. The units making up individual observations are assumed to be randomly selected from a much larger population, and the effect sizes are assumed to have a mean of zero and some variance,  $\sigma^2$ .

The most common model that has random effects other than residual error is the repeated measures or split plot model. [Table 5.1 “Types of Effects in a Split plot Model,” p. 103](#), lists the types of effects in a split plot model. In these models the experiment has two layers. Some effects are applied on the whole plots or subjects of the experiment. Then these plots are divided or the subjects are measured at different times and other effects are applied within those subunits. The effects describing the whole plots or subjects are one random effect, and the subplots or repeated measures are another random effect. Usually the subunit effect is omitted from the model and absorbed as residual error.

**Table 5.1** Types of Effects in a Split plot Model

Split Plot Model	Type of Effect	Repeated Measures Model
whole plot treatment	fixed effect	across subjects treatment
whole plot ID	random effect	subject ID
subplot treatment	fixed effect	within subject treatment
subplot ID	random effect	repeated measures ID

Each of these cases can be treated as a layered model, and there are several traditional ways to fit them in a fair way. The situation is treated as two different experiments:

1. The whole plot experiment has whole plot or subjects as the experimental unit to form its error term.
2. Subplot treatment has individual measurements for the experimental units to form its error term (left as residual error).

The older, traditional way to test whole plots is to do any one of the following:

- Take means across the measurements and fit these means to the whole plot effects.
- Form an  $F$ -ratio by dividing the whole plot mean squares by the whole plot ID mean squares.
- Organize the data so that the split or repeated measures form different columns and do a MANOVA model, and use the univariate statistics.

While these approaches work if the structure is simple and the data are complete and balanced, there is a more general model that works for any structure of random effects. This more generalized model is called the *mixed model*, because it has both fixed and random effects.

Another common situation that involves multiple random effects is in measurement systems where there are multiple measurements with different parts, different operators, different gauges, and different repetitions. In this situation, all the effects are regarded as random.

## Generalizability

Random effects are randomly selected from a larger population, where the distribution of their effect on the response is assumed to be a realization of a normal distribution with a mean of zero and a variance that can be estimated.

Often, the exact effect sizes are not of direct interest. It is the fact that they represent the larger population that is of interest. What you learn about the mean and variance of the effect tells you something about the general population from which the effect levels were drawn. That is different from fixed effects, where you only know about the levels you actually encounter in the data.

## The REML Method

The REML (REstricted or REsidual Maximum Likelihood) method for fitting mixed models is now the mainstream, state-of-the-art method, supplanting older methods.

In the days before availability of powerful computers, researchers needed to restrict their interest to situations in which there were computational short cuts to obtain estimates of variance components and tests on fixed effects in a mixed model. Most books today introduce mixed models using these short cuts that work on balanced data. See McCulloch, Searle, and Neuhaus (2008), Poduri (1997), and Searle, Casella, and McCulloch(1992). The Method of Moments provided a way to calculate what the expected value of Mean Squares (EMS) were in terms of the variance components, and then back-solve to obtain the variance components. It was also possible using these techniques to obtain expressions for test statistics that had the right expected value under the null hypotheses that were synthesized from mean squares.

If your model satisfies certain conditions (*i.e.*, it has random effects that contain the terms of the fixed effects they provide random structure for) then you can use the EMS choice to produce these traditional analyses. However, since the newer REML method produces identical results to these models, but is considerably more general, the EMS method is never recommended.

The REML approach was pioneered by Patterson and Thompson in 1974. See also Wolfinger, Tobias, and Sall (1994) and Searle, Casella, and McCulloch(1992). The reason to prefer REML is that it works without depending on balanced data, or shortcut approximations, and it gets all the tests right, even contrasts that work across interactions. Most packages that use the traditional EMS method are either not able to test some of these contrasts, or compute incorrect variances for them.

## Unrestricted Parameterization for Variance Components in JMP

---

**Note:** Read this section only if you are concerned about matching the results of certain textbooks.

---

There are two different statistical traditions for parameterizing the variance components: the unrestricted and the restricted approaches. JMP and SAS use the unrestricted approach. In this approach, while the estimated effects always sum to zero, the true effects are not assumed to sum to zero over a particular

random selection made of the random levels. This is the same assumption as for residual error. The estimates make the residual errors have mean zero, and the true mean is zero. But a random draw of data using the true parameters will be some random event that might not have a mean of exactly zero.

You need to know about this assumption because many statistics textbooks use the restricted approach. Both approaches have been widely taught for 50 years. A good source that explains both sides is Cobb (1998, section 13.3).

## Negative Variances

---

**Note:** Read this section only when you are concerned about negative variance components.

---

Though variances are always positive, it is possible to have a situation where the unbiased estimate of the variance is negative. This happens in experiments when an effect is very weak, and by chance the resulting data causes the estimate to be negative. This usually happens when there are few levels of a random effect that correspond to a variance component.

JMP can produce negative estimates for both REML and EMS. For REML, there are two checkboxes in the model launch window: **Unbounded Variance Components** and **Estimate Only Variance Components**. Unchecking the box beside **Unbounded Variance Components** constrains the estimate to be non-negative. We recommend that you do not uncheck this if you are interested in fixed effects. Constraining the variance estimates leads to bias in the tests for the fixed effects. If, however, you are only interested in variance components, and you do not want to see negative variance components, then checking the box beside **Estimate Only Variance Components** is appropriate.

If you remain uncomfortable about negative estimates of variances, please consider that the random effects model is statistically equivalent to the model where the variance components are really covariances across errors within a whole plot. It is not hard to think of situations in which the covariance estimate can be negative, either by random happenstance, or by a real process in which deviations in some observations in one direction would lead to deviations in the other direction in other observations. When random effects are modeled this way, the covariance structure is called *compound symmetry*.

So, consider negative variance estimates as useful information. If the negative value is small, it can be considered happenstance in the case of a small true variance. If the negative value is larger (the variance ratio can get as big as 0.5), it is a troubleshooting sign that the rows are not as independent as you had assumed, and some process worth investigating is happening within blocks.

### Scripting Note

The JSL option for **Unbounded Variance Components** is `No Bounds( Boolean )`. Setting this option to true (1) is equivalent to checking the **Unbounded Variance Components** option.

## Random Effects *BLUP* Parameters

Random effects have a dual character. In one perspective, they appear like residual error, often the error associated with a whole-plot experimental unit. In another respect, they are like fixed effects with a parameter to associate with each effect category level. As parameters, you have extra information about them—they are derived from a normal distribution with mean zero and the variance estimated by the

variance component. The effect of this extra information is that the estimates of the parameters are shrunk towards zero. The parameter estimates associated with random effects are called *BLUPs* (Best Linear Unbiased Predictors). Some researchers consider these BLUPs as parameters of interest, and others consider them by-products of the method that are not interesting in themselves. In JMP, these estimates are available, but in an initially-closed report.

BLUP parameter estimates are used to estimate random-effect least squares means, which are therefore also shrunken towards the grand mean, at least compared to what they would be if the effect were treated as a fixed effect. The degree of shrinkage depends on the variance of the effect and the number of observations per level in the effect. With large variance estimates, there is little shrinkage. If the variance component is small, then more shrinkage takes place. If the variance component is zero, the effect levels are shrunk to exactly zero. It is even possible to obtain highly negative variance components where the shrinkage is reversed. You can consider fixed effects as a special case of random effects where the variance component is very large.

If the number of observations per level is large, the estimate will shrink less. If there are very few observations per level, the estimates will shrink more. If there are infinite observations, there is no shrinkage and the answers are the same as fixed effects.

The REML method balances the information on each individual level with the information on the variances across levels.

As an example, suppose that you have batting averages for different baseball players. The variance component for the batting performance across player describes how much variation is usual between players in their batting averages. If the player only plays a few times and if the batting average is unusually small or large, then you tend not to trust that estimate, because it is based on only a few at-bats; the estimate has a high standard error. But if you mixed it with the grand mean, that is, shrunk the estimate towards the grand mean, you would trust the estimate more. For players that have a long batting record, you would shrink much less than those with a short record.

You can run this example and see the results for yourself. The example batting average data are in the *Baseball.jmp* sample data file. To compare the Method of Moments (EMS) and REML, run the model twice. Assign Batting as **Y** and Player as an effect. Select Player in the Construct Model Effects box, and select **Random Effect** from the Attributes pop-up menu.

Run the model and select **REML (Recommended)** from the **Method** popup menu.

Run the model again with **EMS (Traditional)** as **Method**.

[Table 5.2 “Comparison of Estimates Between Method of Moments and REML,” p. 106](#), summarizes the estimates between Method of Moments and REML across a set of baseball players in this simulated example. Note that Suarez, with only 3 at-bats, is shrunk more than the others with more at-bats.

**Table 5.2** Comparison of Estimates Between Method of Moments and REML

	Method of Moments	REML	N
Variance Component	0.01765	0.019648	

**Table 5.2** Comparison of Estimates Between Method of Moments and REML

	Method of Moments	REML	N
Anderson	0.29500000	0.29640407	6
Jones	0.20227273	0.20389793	11
Mitchell	0.32333333	0.32426295	6
Rodriguez	0.55000000	0.54713393	6
Smith	0.35681818	0.35702094	11
Suarez	0.55000000	0.54436227	3
<b>Least Squares Means</b>	same as ordinary means	shrunken from means	

## REML and Traditional Methods Agree on the Standard Cases

It turns out that in balanced designs, the REML  $F$ -test values for fixed effects will be the same as with the Method of Moments (Expected Means Squares) approach. The degrees of freedom could differ in some cases. There are a number of methods of obtaining the degrees of freedom for REML  $F$ -tests; the one that JMP uses is the smallest degrees of freedom associated with a containing effect.

## F-Tests in Mixed Models

---

**Note:** This section details the tests produced with REML

---

The REML method obtains the variance components and parameter estimates, but there are a few additional steps to obtain tests on fixed effects in the model. The objective is to construct the F statistic and associated degrees of freedom to obtain a  $p$ -value for the significance test.

Historically, in simple models using the Method of Moments (EMS), standard tests were derived by construction of quadratic forms that had the right expectation under the null hypothesis. Where a mean square had to be synthesized from a linear combination of mean squares to have the right expectation, Satterthwaite's method could be used to obtain the degrees of freedom to get the  $p$ -value. Sometimes these were fractional degrees of freedom, just as you might find in a modern (Aspin-Welch) Student's  $t$ -test.

With modern computing power and recent methods, we have much improved techniques to obtain the tests. First, Kackar and Harville (1984) found a way to estimate a bias-correction term for small samples. This was refined by Kenward and Roger (1997) to correct further and obtain the degrees of freedom that gave the closest match of an  $F$ -distribution to the distribution of the test statistic. These are not easy calculations, consequently they can take some time to perform for larger models.

If you have a simple balanced model, the results from REML-Kenward-Roger will agree with the results from the traditional approach, provided that the estimates aren't bounded at zero.

## Specifying Random Effects

- These results do not depend on analyzing the syntactic structure of the model. There are no rules about finding containing effects. The method does not care if your whole plot's fixed effects are nested purely in whole plot random effects. You get the right answer regardless.
- These results do not depend on having categorical factors. It handles continuous (random coefficient) models just as easily.
- These methods will produce different (and better) results than older versions of JMP (that is, earlier than JMP 6) that implemented older, less precise, technology to do these tests.
- These methods do not depend on having positive variance components. Negative variance components are not only supported, but need to be allowed in order for the tests to be unbiased.

Our goal in implementing these methods was not just to handle general cases, but to handle cases without the user needing to know very much about the details. Just declare which effects are random, and everything else is automatic. It is particularly important that engineers learn to declare random effects, because they have a history of performing inadvertent split-plot experiments where the structure is not identified.

## Specifying Random Effects

Models with Random Effects use the same Fit Model dialog as other models. To identify a random effect, highlight it in the model effects list and select **Random Effect** from the Attributes popup menu. This appends &Random to the effect name in the model effect list.

### Split Plot Example

The most common type of layered design is a balanced split plot, often in the form of repeated measures across time. One experimental unit for some of the effects is subdivided, (sometimes by time period) and other effects are applied to these subunits.

As an example, consider the **Animals.jmp** data found in the Sample Data folder (the data are fictional). The study collected information about the difference in seasonal hunting habits of foxes and coyotes. Three foxes and three coyotes were marked and observed periodically (each season) for a year. The average number of miles (rounded to the nearest mile) they wandered from their dens during different seasons of the year was recorded. The model is defined by

- the continuous response variable called **miles**
- the **species** effect with values **fox** or **coyote**
- the **season** effect with values **fall**, **winter**, **spring**, and **summer**
- an animal identification code called **subject**, with nominal values 1, 2, and 3 for both foxes and coyotes.

There are two layers to the model.

1. The top layer is the between-subject layer, in which the effect of being a fox or coyote (**species** effect) is tested with respect to the variation from subject to subject. The bottom layer is the within-subject layer, in which the repeated-measures factor for the four seasons (**season** effect) is tested with respect to the variation from season to season within a subject. The within-subject variability is reflected in the residual error.

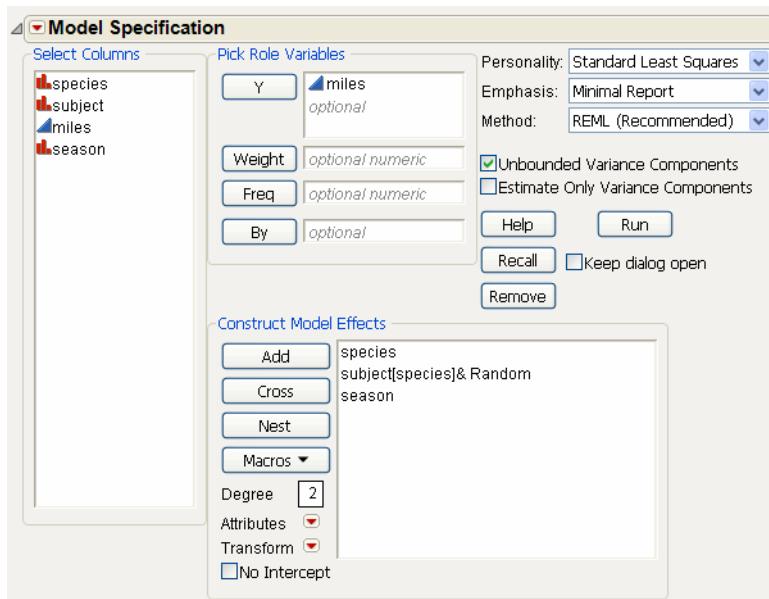
2. The **season** effect can use the residual error for the denominator of its *F*-statistics. However, the between-subject variability is not measured by residual error and must be captured with the subject within species (**subject[species]**) effect in the model. The *F*-statistic for the between-subject effect **species** uses this nested effect instead of residual error for its *F*-ratio denominator.

## The Model Dialog

The **Fit Model** command lets you construct model terms and identify error terms. To fit the nested model to the **Animals.jmp** data with the correct *F*-statistics, specify the response column and add terms to the **Construct Model Effects** list as follows to see the completed window in Figure 5.1.

1. Open the sample data **Animals.jmp**.
2. Select **miles** from the column selector and click **Y**.
3. Select **species** from the column selector list and click **Add**.
4. Select **subject** from the column selector list and click **Add**.
5. Select **species** from the column selector list again.
6. Select **subject** in the **Construct Model Effects** list.
7. Click **Nest** to add the subject within species (**subject[species]**) effect to the model.
8. Select the nested effect, **subject[species]**.
9. Select **Random Effect** from the **Attributes** popup menu. (This nested effect is now identified as an error term for the **species** effect and shows as **subject[species]&Random**.)
10. Select **season** from the column selector list and click **Add**.

The completed dialog is shown Figure 5.1.

**Figure 5.1** Fit Model Dialog

When you assign any effect as random from the **Attributes** popup menu, the Method options (**REML** and **EMS**) appear at the top-right of the dialog, with **REML** selected as the default.

### **Menu Options for Random Effects when REML Method is Used**

When your model includes a **Random Effect** and **REML** is selected as the Method in the launch window, a new menu called **Convergence Settings** becomes available in the red-triangle menu of the Model Specification title bar. This menu includes:

- **Maximum Iterations**
- **Convergence Limit**

When you click on either of these options, you are able to enter new values for these limits. This can be important if you have a very large data set or a complicated model and want to limit the number of iterations, or if your model does not readily converge and you want to either increase the **Maximum Iterations** or increase the **Convergence Limit**.

---

## **REML Results**

A nice feature of REML is that the report doesn't need qualification (Figure 5.2). The estimates are all properly shrunk and the standard errors are properly scaled (SAS Institute Inc. 1996). The variance components are shown as a ratio to the error variance, and as a portion of the total variance.

There is no special table of synthetic test creation, because all the adjustments are automatically taken care of by the model itself. There is no table of expected means squares, because the method does not need this.

If you have random effects in the model, the analysis of variance report is not shown. This is because the variance does not partition in the usual way, nor do the degrees of freedom attribute in the usual way, for REML-based estimates. You can obtain the residual variance estimate from the REML report rather than from the analysis of variance report.

**Figure 5.2** Partial Report of REML Analysis

The screenshot displays a software interface for a REML analysis. The main window is titled "Response miles".

- Summary of Fit:**

RSquare	0.823497
RSquare Adj	0.786338
Root Mean Square Error	1.219062
Mean of Response	4.458333
Observations (or Sum Wgts)	24
- Parameter Estimates:**

Term	Estimate	Std Error	DFDen	t Ratio	Prob> t
Intercept	4.458333	0.42287	4	10.54	0.0005*
species[COYOTE]	1.458333	0.42287	4	3.45	0.0261*
season[fall]	-0.625	0.431003	15	-1.45	0.1676
season[spring]	1.708333	0.431003	15	3.96	0.0012*
season[summer]	0.875	0.431003	15	2.03	0.0605
- Random Effect Predictions:** Not visible in the screenshot.
- REML Variance Component Estimates:**

Random Effect	Var Ratio	Component	Std Error	95% Lower	95% Upper	Pct of Total
subject[species]	0.4719626	0.7013889	0.7707006	-0.809157	2.2119344	32.063
Residual		1.4861111	0.5426511	0.8109483	3.5597535	67.937
Total		2.1875				100.000

-2 LogLikelihood = 78.806486054
- Covariance Matrix of Variance Component Estimates:** Not visible in the screenshot.
- Iterations:** Not visible in the screenshot.
- Fixed Effect Tests:**

Source	Nparm	DF	DFDen	F Ratio	Prob > F
species	1	1	4	11.8932	0.0261*
season	3	3	15	10.6449	0.0005*
- Effect Details:** Not visible in the screenshot.

The Variance Component Estimates table shows 95% confidence intervals for the variance components using the Satterthwaite (1946) approximation. You can right-click on the Variance Components Estimates table to toggle on the Norm KHC (Kackar-Harville correction). This value is an approximation of the magnitude of the increase in the mean squared errors of the estimators for the mixed model. See Kackar and Harville (1984) for a discussion of approximating standard errors in mixed models.

## REML Save Menu

When REML is used, the following commands appear in the Save submenu. All these commands allow the Random effects predicted values to participate in the formulas, rather than their expected value (zero).

**Conditional Pred Formula** saves the prediction formula to a new column in the data table.

**Conditional Pred Values** saves the predicted values to a new column in the data table.

**Conditional Residuals** saves the model residuals to a new column in the data table.

**Conditional Mean CI** saves the mean confidence interval.

**Conditional Indiv CI** saves the confidence interval for an individual.

In addition to these options, a new profiler option becomes available

**Conditional Predictions** includes the random effects predictions in the predicted values and profiles.

## Method of Moments Results

**Note:** This section is only of use in matching historical results

We no longer recommend the Method of Moments, but we understand the need to support it for teaching use, in order to match the results of many textbooks still in use.

You have the option of choosing the **EMS (Traditional)** approach from the **Method** popup menu on the Fit Model dialog. This is also called the Method of Moments method.

Results from the steps for the Method of Moments are as follows:

- For each effect, the coefficients of the expected mean squares for that effect are calculated. This is a linear combination of the variance components and fixed effect values that describes the expected value of the mean square for that effect. All effects also have a unit coefficient on the residual variance.
- The coefficients of expected mean squares for all the random effects, including the residual error, are gathered into a matrix, and this is used to solve for variance components for each random effect.
- For each effect to be tested, a denominator for that effect is synthesized using the terms of the linear combination of mean squares in the numerator that don't contain the effect to be tested or other fixed effects. Thus, the expectation is equal for those terms common to the numerator and denominator. The remaining terms in the numerator then constitute the effect test.
- Degrees of freedom for the synthesized denominator are constructed using Satterthwaite's method.
- The effect tests use the synthetic denominator.

JMP handles random effects like the SAS GLM procedure with a **Random** statement and the **Test** option. Figure 5.3, shows example results.

**Warning:** Standard errors for least squares means and denominators for contrast *F*-tests also use the synthesized denominators. Contrasts using synthetic denominators might not be appropriate, especially in crossed effects compared at common levels. The leverage plots and custom tests are done with respect to the residual, so they might not be appropriate.

**Warning:** Crossed and nested relationships must be declared explicitly. For example, if knowing a subject ID also identifies the group that contains the subject, (that is, if each subject is in only one group), then subject must be declared as nested within group. In that situation, the nesting must be explicitly declared to define the design structure.

**Limitation:** JMP cannot fit a layered design if the effect for a layer's error term cannot be specified under current effect syntax. An example of this is a design with a Latin Square on whole plots for which the error term would be **Row\*Column-Treatment**. Fitting such special cases with JMP requires constructing your own *F*-tests using sequential sums of squares from several model runs.

For the Animals example above, the EMS reports are as follows.

**Figure 5.3** Report of Method of Moments Analysis for Animals Data

**Response miles**

**Summary of Fit**

RSquare	0.838417
RSquare Adj	0.75224
Root Mean Square Error	1.219062
Mean of Response	4.458333
Observations (or Sum Wgts)	24

**Analysis of Variance**

Source	DF	Sum of		F Ratio
		Squares	Mean Square	
Model	8	115.66667	14.4583	9.7290
Error	15	22.29167	1.4861	Prob > F
C. Total	23	137.95833		0.0001*

**Parameter Estimates**

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	4.4583333	0.24884	17.92	<.0001*
species[COYOTE]	1.4583333	0.24884	5.86	<.0001*
season[fall]	-0.625	0.431003	-1.45	0.1676
season[spring]	1.7083333	0.431003	3.96	0.0012*
season[summer]	0.875	0.431003	2.03	0.0605
species[COYOTE]:subject[1]	-0.666667	0.49768	-1.34	0.2003
species[COYOTE]:subject[2]	-0.666667	0.49768	-1.34	0.2003
species[FOX]:subject[1]	-1	0.49768	-2.01	0.0628
species[FOX]:subject[2]	0.25	0.49768	0.50	0.6227

**Expected Mean Squares**

The Mean Square per row by the Variance Component per column

EMS	Intercept	species	subject[species]&Random	season
	Intercept	0	0	0
species	0	12	4	0
subject[species]&Random	0	0	4	0
season	0	0	0	6

plus 1.0 times Residual Error Variance

**Variance Component Estimates**

Component	Var	Percent
	Comp Est	of Total
subject[species]&Random	0.701389	32.063
Residual	1.486111	67.937
Total	2.1875	100.000

These estimates based on equating Mean Squares to Expected Value.

**Test Denominator Synthesis**

Source	MS Den	DF Den	Denom MS Synthesis	
			subject[species]&Random	Residual
species	4.29167	4		
subject[species]&Random	1.486111	15	Residual	
season	1.486111	15	Residual	

**Tests wrt Random Effects**

Source	SS	MS Num	DF Num	F Ratio	Prob > F
species	51.0417	51.0417	1	11.8932	0.0261*
subject[species]&Random	17.1667	4.29167	4	2.8879	0.0588
season	47.4583	15.8194	3	10.6449	0.0005*

**Effect Details**

The random submatrix from the EMS table is inverted and multiplied into the mean squares to obtain variance component estimates. These estimates are usually (but not necessarily) positive. The variance component estimate for the residual is the Mean Square Error.

Note that the CV of the variance components is initially hidden in the Variance Components Estimates report. To reveal it, right-click (Control-click on the Macintosh) and select **Columns > CV** from the menu that appears.



# Chapter 6

## Stepwise Regression The Fit Model Platform

---

In JMP, stepwise regression is a *personality* of the Model Fitting platform; it is one of the selections in the Fitting Personality popup menu on the Fit Model dialog.

Stepwise regression is an approach to selecting a subset of effects for a regression model. It is used when there is little theory to guide the selection of terms for a model and the modeler, in desperation, wants to use whatever seems to provide a good fit.

The approach is somewhat controversial. The significance levels on the statistics for selected models violate the standard statistical assumptions because the model has been selected rather than tested within a fixed model. On the positive side, the approach has been of practical use for 30 years in helping to trim out models to predict many kinds of responses. The book *Subset Selection in Regression*, by A. J. Miller (1990), brings statistical sense to model selection statistics.

This chapter uses the term “significance probability” in a mechanical way to represent that the calculation would be valid in a fixed model, recognizing that the true significance probability could be nowhere near the reported one.

The Stepwise Fit also includes features for looking at all possible models and model averaging.

# Contents

Introduction to Stepwise Regression .....	119
A Multiple Regression Example .....	119
Stepwise Regression Control Panel .....	121
Current Estimates Table .....	122
Step History Table .....	124
Forward Selection Example .....	124
Backwards Selection Example .....	124
Models with Crossed, Interaction, or Polynomial Terms .....	125
Rules for Including Related Terms .....	126
Models with Nominal and Ordinal Terms .....	127
Make Model Command for Hierarchical Terms .....	129
Logistic Stepwise Regression .....	129
All Possible Models .....	130
Model Averaging .....	131
Validation .....	133

---

## Introduction to Stepwise Regression

In JMP, stepwise regression is a *personality* of the Model Fitting platform—it is one of the selections in the Fitting Personality popup menu on the Fit Model dialog (see Figure 6.1). The **Stepwise** feature computes estimates that are the same as those of other least squares platforms, but it facilitates searching and selecting among many models.

---

## A Multiple Regression Example

As an example, consider the **Fitness.jmp** (SAS Institute Inc. 1987) data table in the Sample Data folder, which is the result of an aerobic fitness study.

Aerobic fitness can be evaluated using a special test that measures the oxygen uptake of a person running on a treadmill for a prescribed distance. However, it would be more economical to find a formula that uses simpler measurements that evaluate fitness and predict oxygen uptake. To identify such an equation, measurements of age, weight, runtime, and pulse were taken for 31 participants who ran 1.5 miles.

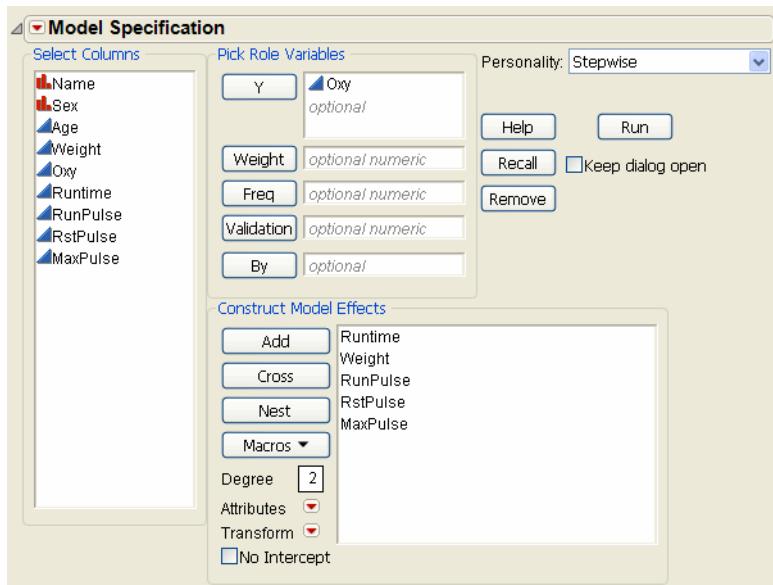
To find a good oxygen uptake prediction equation, you need to compare many different regression models. The Stepwise platform lets you search through models with combinations of effects and choose the model you want.

---

**Note:** For purposes of illustration, certain values of MaxPulse and RunPulse have been changed from data reported by Rawlings (1988, p.105).

---

To do stepwise regression, first select **Fit Model** in the **Analyze** menu. In the Fit Model dialog, choose Oxy as the Y, and Runtime, Weight, RunPulse, RstPulse, MaxPulse as model effects. Then select **Stepwise** from the Personality menu. See Figure 6.1.

**Figure 6.1** Fit Model Dialog for a Stepwise Model

**Note:** Validation is available only in JMP Pro.

Click **Run**.

When the model runs, it displays a window that shows three areas:

- The Stepwise Regression Control panel. See “[Stepwise Regression Control Panel](#),” p. 121.
- The Current Estimates table. See “[Current Estimates Table](#),” p. 122.
- The Step History table. See “[Step History Table](#),” p. 124.

The red triangle menu on the Stepwise Fit report has the following options:

**K-Fold Crossvalidation** is used to perform K-Fold cross-validation in the selection process. When selected, this option enables the Max K-Fold RSquare stopping rule (“[Stepwise Regression Control Panel](#),” p. 121). This is available only for continuous responses. For more information about validation, see “[Validation](#),” p. 133.

**All Possible Models** see “[All Possible Models](#),” p. 130. This is available only for continuous responses.

**Model Averaging** see “[Model Averaging](#),” p. 131. This is available only for continuous responses.

**Plot Criterion History** produces a plot of AICc and BIC versus the number of parameters.

**Plot RSquare History** produces a plot of training and validation R-square versus the number of parameters. This is available only for continuous responses.

**Model Dialog** opens the Model Dialog window with the most recently used settings.

## Stepwise Regression Control Panel

The Stepwise Regression Control Panel (Control Panel for short), shown next, has editable areas, buttons and popup menus. You use these dialog features to limit regressor effect probabilities, determine the method of selecting effects, begin or stop the selection process, and create a model.



The **Stopping Rule** menu has the following options:

**P-value Threshold** uses p-values (significance levels) to enter and remove effects from the model. Two other options appear when P-value Threshold is chosen: **Prob to Enter** is the maximum p-value that an effect must have to be entered into the model during a forward step. **Prob to Leave** is the minimum p-value that an effect must have to be removed from the model during a backward step.

**Minimum AICc** uses the minimum corrected Akaike Information Criterion to choose the best model.

**Minimum BIC** uses the minimum Bayesian Information Criterion to choose the best model.

**Max Validation RSquare** uses the maximum R-square from the validation set to choose the best model. This is available only when a validation column is used, and the validation column has two or three distinct values. For more information about validation, see ["Validation," p. 133](#).

**Max K-Fold RSquare** uses the maximum R-square from K-fold cross-validation to choose the best model. This is available only when K-Fold cross-validation is used. For more information about validation, see ["Validation," p. 133](#).

The **Direction** menu provides options for choosing how effects enter and leave the model, and has the following options:

**Forward** brings in the regressor that most improves the fit, given that term is significant at the level specified by **Prob to Enter**.

**Backward** removes the regressor that affects the fit the least, given that term is not significant at the level specified in **Prob to Leave**.

**Mixed** alternates the forward and backward steps. It includes the most significant term that satisfies **Prob to Enter** and removes the least significant term satisfying **Prob to Leave**. It continues removing terms until the remaining terms are significant and then it changes to the forward direction.

Buttons on the control panel let you control the stepwise processing:

**Go** automates the selection process to completion.

**Stop** stops the selection process.

**Step** increments the selection process one step at a time.

**Enter All** enters all unlocked terms into the model.

**Remove All** removes all terms from the model.

**Make Model** forms a model for the Fit Model Dialog from the model currently showing in the Current Estimates table. In cases where there are nominal or ordinal terms, **Make Model** can create new data table columns to contain terms that are needed for the model.

**Run Model** runs the model currently showing in the Current Estimates table.

The right and left arrow buttons step forward and backward one step in the selection process.

**Tip:** If you have a hierarchy of terms in your model, you can specify their entry rules using the **Rules** drop-down menu. See “[Rules for Including Related Terms](#),” p. 126 for details.

## Current Estimates Table

The Current Estimates table lets you enter, remove, and lock in model effects. The platform begins with no terms in the model except for the intercept, as is shown here. The intercept is permanently locked into the model.

**Figure 6.2** Current Estimates Table

SSE	DFE	RMSE	RSquare	RSquare Adj	Cp	p	AICc	BIC
851.38154	30	5.3272305	0.0000	0.0000	106.93073	1	195.1018	197.5412
<b>Current Estimates</b>								
LockEntered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"		
<input checked="" type="checkbox"/>	Intercept	47.3758065	1	0	0.000	1		
<input type="checkbox"/>	Weight	0	1	22.55181	0.789	0.38169		
<input type="checkbox"/>	Runtime	0	1	632.9001	84.008	4.6e-10		
<input type="checkbox"/>	RunPulse	0	1	134.8447	5.457	0.0266		
<input type="checkbox"/>	RstPulse	0	1	135.7828	5.503	0.02604		
<input type="checkbox"/>	MaxPulse	0	1	47.71646	1.722	0.19975		

You use check boxes to define the stepwise regression process:

**Lock** locks a term in or out of the model. Lock does not permit a term that is checked to be entered or removed from the model.

**Entered** shows whether a term is currently in the model. You can click a term's check box to manually bring an effect into or out of the model.

**Parameter** lists the names of the effects.

**Estimate** is the current parameter estimate. It is zero (0) if the effect is not currently in the model.

**nDF** is the number of degrees of freedom for a term. A term has more than one degree of freedom if its entry into a model also forces other terms into the model.

**SS** is the reduction in the error (residual) SS if the term is entered into the model or the increase in the error SS if the term is removed from the model. If a term is restricted in some fashion, it could have a reported SS of zero.

**"F Ratio"** is the traditional test statistic to test that the term effect is zero. It is the square of a *t*-ratio. It is in quotation marks because it does not have an *F*-distribution for testing the term because the model was selected as it was fit.

**"Prob>F"** is the significance level associated with the *F*-statistic. Like the "F Ratio," it is in quotation marks because it is not to be trusted as a real significance probability.

**R** initially hidden, is the multiple correlation with the other effects in the model.

Statistics for the current model appear above the list of effects:

**SSE, DFE, RMSE** are the sum of squares, degrees of freedom, and root mean square error (residual) of the current model.

**RSquare** is the proportion of the variation in the response that can be attributed to terms in the model rather than to random error.

**RSquare Adj** adjusts  $R^2$  to make it more comparable over models with different numbers of parameters by using the degrees of freedom in its computation. The adjusted  $R^2$  is useful in stepwise procedure because you are looking at many different models and want to adjust for the number of terms in the model.

**Cp** is Mallow's  $C_p$  criterion for selecting a model. It is an alternative measure of total squared error defined as

$$C_p = \left( \frac{SSE_p}{s^2} \right) - (N - 2p)$$

where  $s^2$  is the MSE for the full model and  $SSE_p$  is the sum-of-squares error for a model with  $p$  variables, including the intercept. Note that  $p$  is the number of *x*-variables+1. If  $C_p$  is graphed with  $p$ , Mallows (1973) recommends choosing the model where  $C_p$  first approaches  $p$ .

**p** is the number of parameters in the model, including the intercept.

**AICc** is the corrected Akaike's Information Criterion defined as

$$AICc = -2\text{loglikelihood} + 2k + \frac{2k(k+1)}{n-k-1}$$

where  $k$  is the number of estimated parameters, including intercept and error terms in the model, and  $n$  is the number of observations in the data set. Burnham and Anderson (2004) discuss using  $AIC_c$  for model selection. The best model has the smallest value, as discussed in Akaike (1974).

**BIC** is the Bayesian Information Criterion defined as

$$-2\text{loglikelihood} + k \ln(n)$$

where  $k$  is the number of parameters, and  $n$  is the sample size.

## Step History Table

As each step is taken, the Step History table records the effect of adding a term to the model. The Step History table for the **Fitness** data example shows the order in which the terms entered the model and shows the statistics for each model. Use the radio buttons on the right to choose a model.

Step History										
Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p	AICc	BIC	
1	Runtime	Entered	0.0000	632.9001	0.7434	7.8825	2	155.397	158.81	<input checked="" type="radio"/>
2	RunPulse	Entered	0.1567	15.36208	0.7614	7.4298	3	155.787	159.984	<input type="radio"/>
3	MaxPulse	Entered	0.0140	41.34703	0.8100	2.8284	4	151.592	156.362	<input checked="" type="radio"/>

## Forward Selection Example

The default method of selection is the **Forward** selection. You can proceed with the **Fitness.jmp** data example using the **Step** button on the Control Panel. You see that after one step, the most significant term **Runtime** is entered into the model. Click **Go** to see the stepwise process run to completion. The bottom table in Figure 6.3 shows that all the terms have been added except **RstPulse** and **Weight**.

**Figure 6.3** Current Estimates Table

Current Estimates						
	LockEntered Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
after one step	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Intercept	82.4217727	1	0	0.000	1
	<input type="checkbox"/> <input checked="" type="checkbox"/> Weight	0	1	1.323628	0.171	0.68267
	<input type="checkbox"/> <input checked="" type="checkbox"/> Runtime	-3.3105554	1	632.9001	84.008	4.6e-10
	<input type="checkbox"/> <input checked="" type="checkbox"/> RunPulse	0	1	15.36208	2.118	0.15673
	<input type="checkbox"/> <input checked="" type="checkbox"/> RstPulse	0	1	0.130138	0.017	0.89814
	<input type="checkbox"/> <input checked="" type="checkbox"/> MaxPulse	0	1	1.567361	0.202	0.65632

Current Estimates						
	LockEntered Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
after all steps	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Intercept	80.9007896	1	0	0.000	1
	<input type="checkbox"/> <input checked="" type="checkbox"/> Weight	0	1	4.989591	0.827	0.37137
	<input type="checkbox"/> <input checked="" type="checkbox"/> Runtime	-2.9701867	1	443.2028	73.971	3.25e-9
	<input type="checkbox"/> <input checked="" type="checkbox"/> RunPulse	-0.3751142	1	55.14175	9.203	0.00529
	<input type="checkbox"/> <input checked="" type="checkbox"/> RstPulse	0	1	0.350744	0.056	0.81399
	<input type="checkbox"/> <input checked="" type="checkbox"/> MaxPulse	0.35421891	1	41.34703	6.901	0.01403

## Backwards Selection Example

In backwards selection, terms are entered into the model and then least significant terms are removed until all the remaining terms are significant.

To do a backwards selection stepwise regression in JMP, start by clicking **Enter All**. All effects get entered into the model as shown in Figure 6.4. Then select **Backward** from the Direction popup menu.

**Figure 6.4** Backwards Selection Example in Stepwise Regression

Current Estimates						
LockEntered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/>	Intercept	82.3936054	1	0	0.000	1
<input type="checkbox"/>	Weight	-0.0509071	1	4.83788	0.772	0.38784
<input type="checkbox"/>	Runtime	-2.9518165	1	366.3375	58.489	5.29e-8
<input type="checkbox"/>	RunPulse	-0.3970425	1	59.51519	9.502	0.00495
<input type="checkbox"/>	RstPulse	0.01239004	1	0.199033	0.032	0.85995
<input checked="" type="checkbox"/>	MaxPulse	0.38479281	1	45.83023	7.317	0.01212

Click the **Step** button. The first backward step removes RstPulse and the second step removes Weight. No further terms meet the Prob to Leave probability specified in the Control Panel. The Current Estimates and Step History tables shown in Figure 6.5 summarize the backwards stepwise selection process.

**Figure 6.5** Current Estimates with Terms Removed and Step History Table

Current Estimates						
LockEntered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/>	Intercept	80.9007896	1	0	0.000	1
<input type="checkbox"/>	Weight	0	1	4.989591	0.827	0.37137
<input type="checkbox"/>	Runtime	-2.9701867	1	443.2028	73.971	3.25e-9
<input type="checkbox"/>	RunPulse	-0.3751142	1	55.14175	9.203	0.00529
<input type="checkbox"/>	RstPulse	0	1	0.350744	0.056	0.81399
<input checked="" type="checkbox"/>	MaxPulse	0.35421891	1	41.34703	6.901	0.01403

Step History										
Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p	AICc	BIC	
1	All	Entered	.	.	0.8161	6	6	157.051	162.22	<input type="radio"/>
2	RstPulse	Removed	0.8600	0.199033	0.8158	4.0318	5	153.721	158.825	<input type="radio"/>
3	Weight	Removed	0.3714	4.989591	0.8100	2.8284	4	151.592	156.362	<input checked="" type="radio"/>

## Models with Crossed, Interaction, or Polynomial Terms

Often with models from experimental designs, you have cross-product or interaction terms. For continuous factors, these are simple multiplications. For nominal and ordinal factors, the interactions are outer products of many columns. When there are crossed terms, you usually want to impose rules on the model selection process so that a crossed term cannot be entered unless all its subterms (terms that contain it) are in the model.

The next example uses the Reactor.jmp sample data (Box, Hunter, and Hunter 1978). The response is Y and the variables are F, Ct, A, T, and Cn.

The model shown here is composed of all factorial terms up to two-factor interactions for the five continuous factors. Note that some terms have more than one degree of freedom (nDF) due to the restrictions placed on some of the terms. Under the model selection rules described above, a crossed term cannot be entered into the model until all its subterms are also in the model. For example, if the stepwise process enters F\*Ct, then it must also enter F and Ct, which gives F\*Ct an nDF of 3.

Current Estimates						
LockEntered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/>	Intercept	65.5	1	0	0.000	1
<input type="checkbox"/>	F	0	1	15.125	0.066	0.79972
<input type="checkbox"/>	Ct	0	1	3.042	23.412	3.67e-5
<input type="checkbox"/>	A	0	1	3.125	0.014	0.90823
<input type="checkbox"/>	T	0	1	924.5	4.611	0.03998
<input type="checkbox"/>	Cn	0	1	312.5	1.415	0.24363
<input type="checkbox"/>	F*Ct	0	3	3,072.25	7.414	0.00084
<input type="checkbox"/>	F*A	0	3	22.75	0.031	0.9926
<input type="checkbox"/>	F*T	0	3	945.75	1.473	0.24335
<input type="checkbox"/>	F*Cn	0	3	327.75	0.463	0.71063
<input type="checkbox"/>	Ct*A	0	3	3,051.25	7.323	0.0009
<input type="checkbox"/>	Ct*T	0	3	5,371	31.950	3.51e-9
<input type="checkbox"/>	Ct*Cn	0	3	3,386.5	8.895	0.00027
<input type="checkbox"/>	A*T	0	3	963.75	1.505	0.23482
<input type="checkbox"/>	A*Cn	0	3	321.75	0.454	0.71671
<input type="checkbox"/>	T*Cn	0	3	2,205	4.346	0.01235

The progress of multiterm inclusion is a balance between numerator degrees of freedom and opportunities to improve the fit. When there are significant interaction terms, often several terms enter at the same step. If the **Step** button is clicked once, Ct\*T is entered along with its two contained effects Ct and T. However, a step back is not symmetric because a crossed term can be removed without removing its two component terms. Note that Ct now has 2 degrees of freedom because if Stepwise removes Ct, it also removes Ct\*T.

Current Estimates						
LockEntered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/>	Intercept	65.5	1	0	0.000	1
<input type="checkbox"/>	F	0	1	15.125	0.263	0.61236
<input checked="" type="checkbox"/>	Ct	9.75	2	4,446.5	39.676	6.74e-9
<input type="checkbox"/>	A	0	1	3.125	0.054	0.81819
<input checked="" type="checkbox"/>	T	5.375	2	2,329	20.781	2.93e-6
<input type="checkbox"/>	Cn	0	1	312.5	6.715	0.01524
<input type="checkbox"/>	F*Ct	0	2	30.25	0.256	0.7764
<input type="checkbox"/>	F*A	0	3	22.75	0.123	0.9459
<input type="checkbox"/>	F*T	0	2	21.25	0.178	0.83755
<input type="checkbox"/>	F*Cn	0	3	327.75	2.200	0.11303
<input type="checkbox"/>	Ct*A	0	2	9.25	0.077	0.92801
<input checked="" type="checkbox"/>	Ct*T	6.625	1	1,404.5	25.064	2.72e-5
<input type="checkbox"/>	Ct*Cn	0	2	344.5	3.657	0.03984
<input type="checkbox"/>	A*T	0	2	39.25	0.334	0.7194
<input type="checkbox"/>	A*Cn	0	3	321.75	2.150	0.11924
<input type="checkbox"/>	T*Cn	0	2	1,280.5	57.700	2.7e-10

## Rules for Including Related Terms

You can change the rules that are applied when there is a hierarchy of terms in the model. Notice that when terms are related, an extra popup menu called **Rules** appears.

**Figure 6.6** Rules Menu

The **Rules** choices are used for related terms:

**Combine** groups a term with its precedent terms and calculates the group's significance probability for entry as a joint *F*-test. **Combine** is the default rule.

**Restrict** restricts the terms that have precedents so that they cannot be entered until their precedents are entered.

**No Rules** gives the selection routine complete freedom to choose terms, regardless of whether the routine breaks a hierarchy or not.

**Whole Effects** enters only whole effects, when all terms involving that effect are significant. This rule applies only when categorical variables with more than two levels are entered as possible model effects.

## Models with Nominal and Ordinal Terms

Traditionally, stepwise regression has not addressed the situation when there are categorical terms in the model. When nominal or ordinal terms are in regression models, they are carried as sets of dummy or indicator columns. When there are only two levels, there is no problem because they generate only a single column. However, for more than two levels, multiple columns must be handled. The convention in JMP for nominal variables in standard platforms is to model these terms so that the parameter estimates average out to zero across all the levels.

In the stepwise platform, categorical variables (nominal and ordinal) are coded in a hierarchical fashion, which is different from the other least squares fitting platforms. In hierarchical coding, the levels of the categorical variable are considered in some order and a split is made to make the two groups of levels that most separate the means of the response. Then, each group is further subdivided into its most separated subgroups, and so on, until all the levels are distinguished into  $k - 1$  terms for  $k$  levels.

For nominal terms, the order of levels is determined by the means of the Ys. For ordinal terms, the order is fixed.

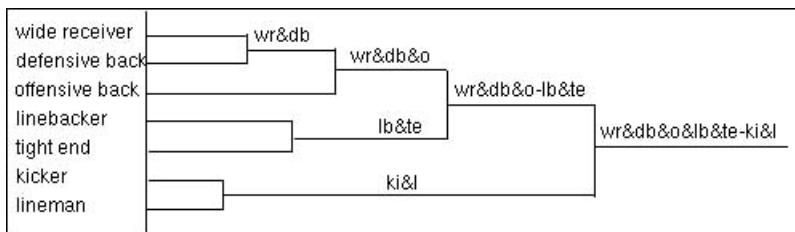
For example, consider the Football.jmp data. Specify Speed as the response, and Weight and Position2 as predictors. Position2 is a nominal variable with values representing football positions. The Current Estimates table is shown in Figure 6.7.

**Figure 6.7** Position Hierarchy

Current Estimates						
LockEnteredParameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Intercept	61.0852252	1	0	0.000	1	
<input type="checkbox"/> <input type="checkbox"/> Weight	0	1	619.6596	77.646	2.2e-14	
<input type="checkbox"/> <input type="checkbox"/> Position2{wr&db&o&lb&te-ki&l}	0	1	723.3538	102.906	2e-17	
<input type="checkbox"/> <input type="checkbox"/> Position2{wr&db&o-lb&te}	0	2	818.9823	65.952	1.9e-19	
<input type="checkbox"/> <input type="checkbox"/> Position2{wr&db-o}	0	3	824.4063	44.207	1.2e-18	
<input type="checkbox"/> <input type="checkbox"/> Position2{wr-db}	0	4	826.0589	32.993	7.4e-18	
<input type="checkbox"/> <input type="checkbox"/> Position2{lb-te}	0	3	819.667	43.642	1.7e-18	
<input type="checkbox"/> <input type="checkbox"/> Position2{ki-l}	0	2	749.098	54.631	4.1e-17	

- The method first splits Position into two groups with the term Position2{wr&db&o&lb&te-ki&l}. One group (the faster group) consists of the wide receivers (wr), defensive backs (db), offensive backs (o), linebackers (lb), and tight ends (te). The other group (the slower group) is the kickers (ki), and linemen (l), which ultimately split as Position2{ki-l}.
- The next split subdivides the faster group into wide receivers (wr), defensive backs (db), and offensive backs(o) versus linebackers (lb) and tight ends (te), shown as Position2{wr&db&o-lb&te}. The linebackers (lb) and tight ends split (te) to form Position2{lb-te}.
- The faster group divides again giving Position2{wr&db-o}, wide receivers (wr) and defensive backs (db) versus offensive backs (o).
- The last subdivision is the wide receivers (wr) and defensive backs (db), Position2{wr-db}.

These terms can be illustrated by the tree hierarchy shown at the top in Figure 6.8.

**Figure 6.8** Tree Structure of Terms and Corresponding Current Estimates Table

Using the default **Combine** rule for terms to enter a model or the **Restrict** rule, a term cannot enter the model unless all the terms above it in the hierarchy have been entered. Thus, it is simple to bring in the term Position2{wr&db&o&lb&te-ki&l} because there is nothing above it in the hierarchy. But to enter Position2{lb-te} requires that the two other terms above it in the hierarchy are entered: Position2{wr&db&o-lb&te}, and Position2{wr&db&o&lb&te-ki&l}.

Reasons to choose a hierarchical coding include:

- The hierarchy leads to natural stepping rules.
- The groupings that make the greatest initial separation enter the model early.

## Make Model Command for Hierarchical Terms

When you click **Make Model** or **Run Model** for a model with nominal or ordinal terms, **Fit Model** creates a new set of columns in the data table that it needs. The model appears in a new Fit Model window for the response variable. The next example uses the Hotdogs2 sample data to illustrate how **Stepwise** constructs a model with hierarchical effects.

A simple model (Figure 6.9) looks at the cost per ounce (\$/oz) as a function of Type (Meat, Beef, Poultry) and Size (Jumbo, Regular, Hors d'oeuvre). Choose **P-value Threshold** on the Stopping Rule menu, and **Restrict** on the Rules menu.

**Figure 6.9** Stepwise Platform for Model with Hierarchical Effects

LockEntered Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Intercept	0.10849381	1	0	0.000	1
<input type="checkbox"/> <input checked="" type="checkbox"/> Type{Poultry&Meat-Beef}	-0.0275278	0	0	.	.
<input type="checkbox"/> <input checked="" type="checkbox"/> Type{Poultry-Meat}	-0.0205527	1	0.013985	11.083	0.00164
<input type="checkbox"/> <input checked="" type="checkbox"/> Size{Hors d'oeuvre-Regular&Jumbo}	-0.0121982	1	0.002596	2.057	0.1577
<input type="checkbox"/> <input type="checkbox"/> Size{Regular-Jumbo}	0	1	0.000125	0.097	0.7566

When you choose **Make Model** in the Stepwise Regression Control Panel, two actions occur:

- Indicator variables are created in the data table for those checked rows in the Current Estimates table that are partial levels of a main effect. In this example, two columns are created for Type and one column is created for Size.
- A new Fit Model dialog opens. The effects are those that were selected in the stepwise process.

## Logistic Stepwise Regression

JMP performs logistic stepwise regression in a similar way to standard least-squares logistic regression. To run a logistic stepwise regression, simply add terms to the model as usual and choose Stepwise from the personality drop-down menu.

The difference in the report when the response is categorical is in the **Current Estimates** section of the report. Wald/Score chi-square statistics appear, and the overall fit of the model is shown as -LogLikelihood. An example is shown in Figure 6.10, using Fitness.jmp with Sex as the response, and Weight, Runtime, RunPulse, RstPulse, and MaxPulse as effects. Click the **Go** button on the Control Panel.

**Figure 6.10** Logistic Stepwise Report

The screenshot displays two tables from a software interface. The first table, titled "Current Estimates", lists parameters with their estimates, degrees of freedom (nDF), ChiSq values, and "Sig Prob". The second table, titled "Step History", shows the step-by-step process of entering variables, including ChiSquare, "Sig Prob", RSquare, p-value, AICc, and BIC.

LockEntered Parameter	Estimate	nDF	Wald/Score	
			ChiSq	"Sig Prob"
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Intercept[M]	35.8078459	1	0	1
<input type="checkbox"/> <input checked="" type="checkbox"/> Weight	-0.2822965	1	5.471429	0.01933
<input type="checkbox"/> <input checked="" type="checkbox"/> Runtime	-1.3268037	1	2.545732	0.11059
<input type="checkbox"/> <input type="checkbox"/> RunPulse	0	1	0.470408	0.4928
<input type="checkbox"/> <input type="checkbox"/> RstPulse	0	1	0.007216	0.9323
<input type="checkbox"/> <input type="checkbox"/> MaxPulse	0	1	0.076497	0.7821

Step	Parameter	Action	ChiSquare	"Sig Prob"	L-R			
					RSquare	p	AICc	BIC
1	Weight	Entered	12.16669	0.0005	0.2833	2	35.2047	37.6441
2	Runtime	Entered	8.017826	0.0046	0.4700	3	29.6472	33.0803
3	RunPulse	Entered	1.440088	0.2301	0.5036	4	30.8567	35.0542
4	MaxPulse	Entered	0.071809	0.7887	0.5052	5	33.6464	38.4164
5	RstPulse	Entered	0.007156	0.9326	0.5054	6	36.7393	41.8432
6	Best	Specific	.	.	0.4700	3	29.6472	33.0803

The enter and remove statistics are calculated using cheap Score or Wald chi-square tests respectively, but the regression estimates and log-likelihood values are based on the full iterative maximum likelihood fit. If you want to compare the Wald/Score values, look at the Step History report.

## All Possible Models

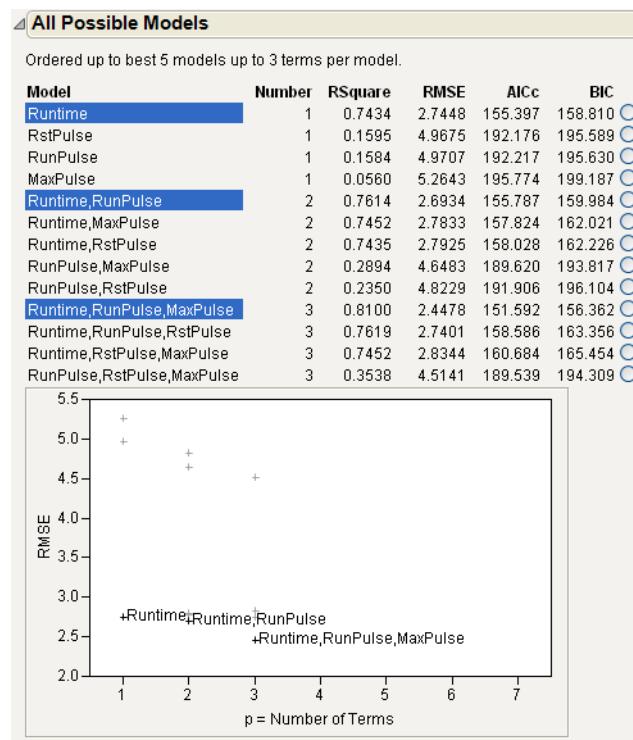
For continuous responses, the Stepwise platform includes an **All Possible Models** command. It is accessible from the red-triangle menu on the Stepwise control panel. When selected, a new popup dialog is shown prompting you to enter values for the maximum number of terms to fit in any one model and for the maximum number of best model results to show for each number of terms in the model.

For an example, use the Fitness.jmp data table. Use Oxy as the response, and Runtime, RunPulse, RstPulse, and MaxPulse as the effects. For this example, the maximum number of terms is 3 and the number of best models is 5. See Figure 6.11.

**Figure 6.11** All Possible Models Popup Dialog

The dialog box has two input fields. The first field, "Maximum number of terms in a model:", contains the value "3". The second field, "Number of best models to see:", contains the value "5".

After clicking **OK**, the platform fits all possible models up to three terms in a model. The results are shown in Figure 6.12.

**Figure 6.12** All Possible Models Report

The models are listed in increasing order of the number of parameters they contain. The model with the highest  $R^2$  for each number of parameters is highlighted. The radio button column at the right of the table allows you to select one model at a time and check the results.

**Note:** The recommended criterion for selecting a model is to choose the one corresponding to the smallest BIC or AICc value. Some analysts also want to see the  $C_p$  statistic. Mallow's  $C_p$  statistic is computed, but initially hidden in the table. To make it visible, Right-click (Control-click on the Macintosh) in the table and select **Columns > Cp** from the menu that appears.

## Model Averaging

For continuous responses, the Stepwise platform includes a **Model Averaging** command, which is found on the red-triangle menu of the Stepwise Control Panel.

Model averaging is a technique which, instead of picking a single best model, allows you to average the fits for a number of models. The result is a model with excellent prediction capability.

This feature is particularly useful for new and unfamiliar models that you do not want to overfit. When many terms are selected into a model, the fit tends to inflate the estimates. Model averaging tends to shrink

the estimates on the weaker terms, yielding better predictions. The models are averaged with respect to the AICc weight, calculated as:

$$\text{AICcWeight} = \exp[-0.5(\text{AICc} - \text{AICcBest})]$$

where AICcBest is the smallest AICc value among the fitted models. The AICc Weights are then sorted in decreasing order. The AICc weights cumulating to less than one minus the cutoff of the total AICc weight are set to zero, allowing the very weak terms to have true zero coefficients instead of extremely small coefficient estimates.

Clicking on **Model Averaging** produces a new popup dialog that asks you to enter the maximum number of terms in any one model and a value that is used as the cutoff of the total AICc weight.

For an example, use the **Fitness.jmp** data table. Use **Oxy** as the response, and **Runtime**, **RunPulse**, **RstPulse**, and **MaxPulse** as the effects. For this example, the maximum number of terms is 3 and the weight cutoff is 0.95. See Figure 6.13.

**Figure 6.13** Model Averaging Dialog



Clicking **OK** yields the results shown in Figure 6.14. Average estimates and standard errors are shown for each parameter. The standard errors shown reflect the bias of the estimates toward zero.

**Figure 6.14** Model Averaging Results

<b>Model Averaging</b>		
Averaging models with 1 to 3 terms, using a cutoff AICc weight quantile of 0.9707, which resulted in using 5 out of 14 models fit		
Parameter	Estimate	Std Error
Intercept	82.4063	.
Runtime	-3.0422	0.3465967
RunPulse	-0.2832	0.1053199
RstPulse	-0.000286	0.0126522
MaxPulse	0.2603	0.1141399
<b>Save Prediction Formula</b>		

The prediction formula can be saved in the original data table by clicking on **Save Prediction Formula** in the report.

---

## Validation

---

**Note:** Validation is available only in JMP Pro.

---

Validation is the process of using part of a data set to estimate model parameters, and using the other part to assess the predictive ability of the model.

- The *training* set is the part that estimates model parameters.
- The *validation* set is the part that assesses or validates the predictive ability of the model.
- The *test* set is a final, independent assessment of the model's predictive ability. The test set is available only when using a validation column.

The training, validation, and test sets are created by subsetting the original data into parts. This is done through the use of a validation column on the Fit Model launch window.

The validation column's values determine how the data is split, and what method is used for validation:

- If the column has two distinct values, then training and validation sets are created.
- If the column has three distinct values, then training, validation, and test sets are created.
- If the column has four or more distinct values, K-Fold crossvalidation is performed.

When validation is used, model fit statistics are given for the training, validation, and test sets.

## K-Fold Crossvalidation

K-Fold crossvalidation divides the original data into K subsets. In turn, each of the K sets is used to validate the model fit on the rest of the data, fitting a total of K models. The model giving the best validation statistic is chosen as the final model. This method is best for small data sets, because it makes efficient use of limited amounts of data.

To use K-Fold crossvalidation, do one of the following:

- Use a validation column with four or more distinct values.
- Choose K-Fold Crossvalidation from the Stepwise Fit red triangle menu.



# Chapter 7

## Multiple Response Fitting

### The Fit Model Platform

---

When more than one  $Y$  is specified for a model in the Fit Model platform, you can choose the **Manova** personality for multivariate fitting. Multivariate models fit several  $Y$ 's to a set of effects. Functions across the  $Y$ 's can be tested with appropriate response designs. In addition to standard MANOVA models, you can do the following techniques with this facility:

- Repeated measures analysis when repeated measurements are taken on each subject and you want to analyze effects both between subjects and within subjects across the measurements. This multivariate approach is especially important when the correlation structure across the measurements is arbitrary.
- Canonical correlation to find the linear combination of the  $X$  and  $Y$  variables that has the highest correlation.
- Discriminant analysis to find distance formulas between points and the multivariate means of various groups so that points can be classified into the groups that they are most likely to be in. A more complete implementation of discriminant analysis is in the **Discriminant** platform.

Multivariate fitting has a different flavor from univariate fitting. The multivariate fit begins with a rudimentary preliminary analysis that shows parameter estimates and least squares means. You can then specify a response design across the  $Y$ 's, and multivariate tests are performed.

# Contents

Multiple Response Model Specification . . . . .	137
Initial Fit . . . . .	137
Specification of the Response Design . . . . .	140
Multivariate Tests . . . . .	142
The Extended Multivariate Report . . . . .	142
Comparison of Multivariate Tests . . . . .	143
Univariate Tests and the Test for Sphericity . . . . .	144
Multivariate Model with Repeated Measures . . . . .	145
Repeated Measures Example . . . . .	146
A Compound Multivariate Model . . . . .	147
Commands for Response Type and Effects . . . . .	149
Test Details (Canonical Details) . . . . .	150
The Centroid Plot . . . . .	150
Save Canonical Scores (Canonical Correlation) . . . . .	151
Discriminant Analysis . . . . .	152

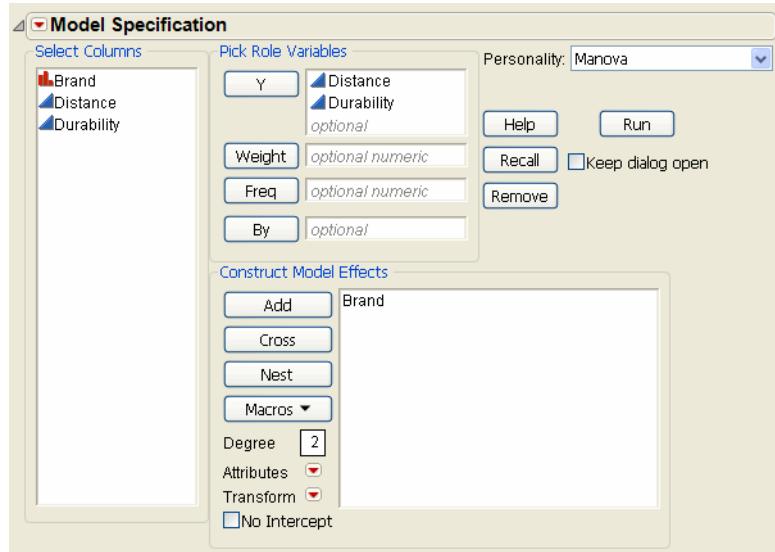
## Multiple Response Model Specification

To form a multivariate model, choose **Fit Model** from the **Analyze** menu and assign multiple Y's in the dialog. Then select **Manova** from the fitting personality popup menu.

This example uses the **Golf Balls.jmp** sample data table (McClave and Dietrich, 1988). The data are a comparison of distances traveled and a measure of durability for three brands of golf balls. A robotic golfer hit a random sample of ten balls for each brand in a random sequence. The hypothesis to test is that distance and durability are the same for the three golf ball brands.

In the Fit Model platform, assign both Distance and Durability to the **Y** role, and assign Brand as an effect. Select **Manova** from the Personality menu. See Figure 7.1.

**Figure 7.1** Manova Setup



## Initial Fit

Click **Run** to perform the initial fit. The results are described in the next several sections. The initial results might not be very interesting in themselves, because no response design has been specified yet. After you specify a response design, the multivariate platform displays tables of multivariate estimates and tests. For details about specifying a response design, see “[Specification of the Response Design](#),” p. 140.

The Manova Fit red-triangle menu has the following options:

**Save Discrim** performs a discriminant analysis and saves the results to the data table. For more details, see “[Discriminant Analysis](#),” p. 152.

**Save Predicted** saves the predicted responses to the data table.

**Save Residuals** saves the residuals to the data table.

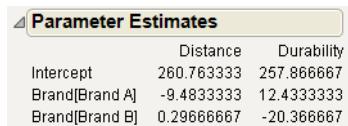
**Model Dialog** opens a Fit Model window populated with the settings of the most recent model.

### **Parameter Estimates Table**

The Parameter Estimates (see Figure 7.2) report includes only the parameter estimates for each response variable, without details like standard errors or *t*-tests. There is a column for each response variable.

---

**Figure 7.2** Parameter Estimates

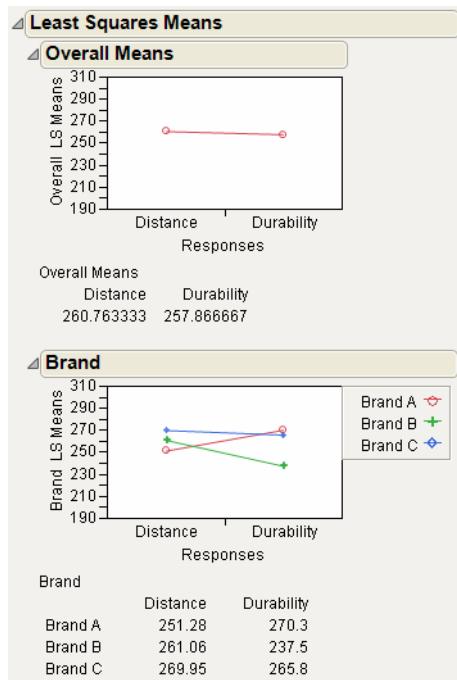


	Distance	Durability
Intercept	260.763333	257.866667
Brand[Brand A]	-9.4833333	12.4333333
Brand[Brand B]	0.29666667	-20.366667

---

### **Least Squares Means Report**

For each pure nominal effect, the Least Squares Means table reports the overall least squares means of all the response columns, least squares means of each nominal level, and LS means plots of the means. Figure 7.3 shows the LS Mean plot of the golf ball brands and the table of least squares means. The same plot and table are available for the overall means.

**Figure 7.3 Least Squares Means Report**

### Partial Covariance and Correlation Tables

The Partial Correlation table (see Figure 7.4) shows the covariance matrix and the partial correlation matrix of residuals from the initial fit, adjusted for the  $X$  effects.

**Figure 7.4 Partial Correlation**

P.Cov		
	Distance	Durability
Distance	23.6424074	9.06518519
Durability	9.06518519	895.859259
P.Corr		
	Distance	Durability
Distance	1.0000	0.0623
Durability	0.0623	1.0000

### Overall E&H Matrices

The main ingredients of multivariate tests are the  $E$  and the  $H$  matrices (see Figure 7.5):

- The elements of the  $E$  matrix are the cross products of the residuals.

## Multiple Response Model Specification

- The **H** matrices correspond to hypothesis sums of squares and cross products.

There is an **H** matrix for the whole model and for each effect in the model. Diagonal elements of the **E** and **H** matrices correspond to the hypothesis (numerator) and error (denominator) sum of squares for the univariate *F* tests. New **E** and **H** matrices for any given response design are formed from these initial matrices, and the multivariate test statistics are computed from them.

**Figure 7.5** E and H Matrices

Overall E&H Matrices		
E		
Distance	Distance	Durability
Distance	638.345	244.76
Durability	244.76	24,188.2
Whole Model H		
	Distance	Durability
Distance	1744.16467	-510.70667
Durability	-510.70667	6323.26667
Intercept		
	Distance	Durability
Distance	2039925.48	2017265.15
Durability	2017265.15	1994856.53
Brand		
	Distance	Durability
Distance	1744.16467	-510.70667
Durability	-510.70667	6323.26667

**Specification of the Response Design**

You use the Response Specification dialog (see Figure 7.6) to specify the response designs for various tests.

**Figure 7.6** Response Specification

Response Specification	
To construct the linear combinations across responses,	<input type="button" value="Choose Response"/>
<input checked="" type="checkbox"/> Univariate Tests Also	<input type="checkbox"/> Test Each Column Separately Also

The dialog has these optional check boxes:

- The **Univariate Tests Also** check box is used in repeated measures models to obtain adjusted and unadjusted univariate repeated measures tests as well as multivariate tests.
- The **Test Each Column Separately Also** check box is used in obtaining univariate ANOVA tests on each response as well as multivariate tests.

The response design forms the M matrix. The columns of an M matrix define a set of transformation variables for the multivariate analysis. The **Choose Response** popup menu lists the choices for M shown below. The popup menu has the following response matrices:

**Repeated Measures** constructs and runs both Sum and Contrast responses.

**Sum** is the sum of the responses, giving a single value.

**Identity** uses each separate response, the identity matrix.

**Contrast** compares each response and the first response.

**Polynomial** constructs a matrix of orthogonal polynomials.

**Helmert** compares each response with the combined responses listed below it.

**Profile** compares each response with the following response.

**Mean** compares each response versus the mean of the others.

**Compound** creates and runs several response functions that are appropriate if the responses are compounded from two effects.

**Custom** uses any custom M matrix you enter.

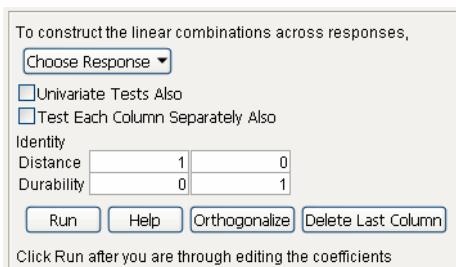
The most typical response designs are **Repeated Measures** and **Identity** for multivariate regression. There is little difference in the tests given by **Contrast**, **Helmert**, **Profile**, and **Mean** because they span the same space. However, the tests and details in the Least Squares means and Parameter Estimates tables for them show correspondingly different highlights.

The **Repeated Measures** and the **Compound** response selections display dialogs to specify response effect names. They then continue on to fit several response functions without waiting for further user input. Otherwise, selections expand the control panel and give you more opportunities to refine the specification.

Figure 7.7 shows the **Identity** dialog selected from the **Response Design** popup menu and the default M matrix for the Y variables distance and durability. The Response Specification dialog stays open to specify further response designs for estimation and testing.

---

**Figure 7.7** Identity Control Panel




---

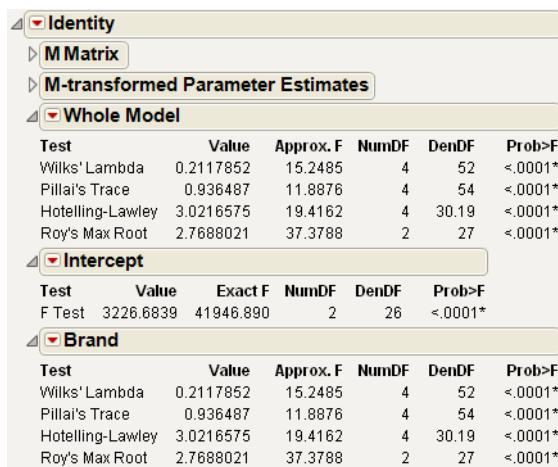
The response function is shown in the form of the M matrix, and control buttons appear at the bottom of the dialog (see Figure 7.7). You use this part of the dialog in the following ways:

- Check the **Univariate Tests Also** check box to see the adjusted and unadjusted univariate tests as well as the multivariate tests in a repeated measures situation. Univariate tests are done with respect to an orthonormalized version of M.
- Click any value in the M matrix text edit boxes to change a value.
- Use the **Orthogonalize** button to orthonormalize the matrix. Orthonormalization is done after the column contrasts (sum to zero) for all response types except **Sum**.
- Use **Delete Last Column** to reduce the dimensionality of the transformation.

## Multivariate Tests

After you complete the Response Specification dialog, click **Run** to see multivariate estimations and tests. Figure 7.8 shows the results of clicking Run on the dialog shown in Figure 7.7.

**Figure 7.8** Multivariate Test Menus



The screenshot displays three tables from the Multivariate Test Menus:

Whole Model					
Test	Value	Approx. F	NumDF	DenDF	Prob>F
Wilks' Lambda	0.2117852	15.2485	4	52	<.0001*
Pillai's Trace	0.936487	11.8876	4	54	<.0001*
Hotelling-Lawley	3.0216575	19.4162	4	30.19	<.0001*
Roy's Max Root	2.7688021	37.3788	2	27	<.0001*

Intercept					
Test	Value	Exact F	NumDF	DenDF	Prob>F
F Test	3226.6839	41946.890	2	26	<.0001*

Brand					
Test	Value	Approx. F	NumDF	DenDF	Prob>F
Wilks' Lambda	0.2117852	15.2485	4	52	<.0001*
Pillai's Trace	0.936487	11.8876	4	54	<.0001*
Hotelling-Lawley	3.0216575	19.4162	4	30.19	<.0001*
Roy's Max Root	2.7688021	37.3788	2	27	<.0001*

The **M Matrix** report gives the response design you specified. The **M-transformed Parameter Estimates** report gives the original parameter estimates matrix multiplied by the transpose of the M matrix.

**Note:** Initially in this chapter the matrix names E and H refer to the error and hypothesis cross products. After specification of a response design, E and H refer to those matrices transformed by the response design, which are actually  $M'EM$  and  $M'HM$ .

## The Extended Multivariate Report

In multivariate fits, the sums of squares due to hypothesis and error are matrices of squares and cross products instead of single numbers. And there are lots of ways to measure how large a value the matrix for the hypothesis sums of squares and cross products (called **H** or **SSCP**) is compared to that matrix for the

residual (called E). JMP reports the four multivariate tests that are commonly described in the literature. If you are looking for a test at an exact significance level, you will have to go hunting for tables in reference books. Fortunately, all four tests can be transformed into an approximate  $F$ -test. If the response design yields a single value, or if the hypothesis is a single degree of freedom, the multivariate tests are equivalent and yield the same exact  $F$ -test. JMP labels the test **Exact F**; otherwise, JMP labels it **Approx. F**.

In the golf balls example, there is only one effect so the Whole Model test and the test for Brand are the same, which show the four multivariate tests with approximate  $F$  tests. There is only a single intercept with two DF (one for each response), so the  $F$ -test for it is exact and is labeled **Exact F**.

The red-triangle menus on the Whole Model, Intercept, and Brand reports give options to generate additional information, which includes eigenvalues, canonical correlations, a list of centroid values, a centroid plot, and a **Save** command that lets you save canonical variates. These options are discussed in the section “[Commands for Response Type and Effects](#),” p. 149.

The effect (Brand in this example) popup menu also includes the option to specify contrasts.

The custom test and contrast features are the same as those for regression with a single response. See the chapters the “[Standard Least Squares: Introduction](#),” p. 21, and “[Standard Least Squares: Perspectives on the Estimates](#),” p. 57, for a description and examples of contrasts and custom tests.

“[Multivariate Details](#),” p. 636 in the appendix “[Statistical Details](#),” p. 607, shows formulas for the MANOVA table tests.

Each MANOVA test table, except the Sphericity Test table, has these elements:

**Test** labels each statistical test in the table. If the number of response function values (columns specified in the M matrix) is 1 or if an effect has only one degree of freedom per response function, the exact  $F$ -test is presented. Otherwise, the standard four multivariate test statistics are given with approximate  $F$  tests: Wilks' Lambda ( $\Lambda$ ), Pillai's Trace, the Hotelling-Lawley Trace, and Roy's Maximum Root.

**Value** the value of each multivariate statistical test in the report.

**Approx. F (or Exact F)** the  $F$ -values corresponding to the multivariate tests. If the response design yields a single value or if the test is one degree of freedom, this will be an exact  $F$ -test.

**NumDF** the numerator degrees of freedom.

**DenDF** the denominator degrees of freedom.

**Prob>F** the significance probability corresponding to the  $F$ -value.

## Comparison of Multivariate Tests

Although the four standard multivariate tests often give similar results, there are situations where they differ, and one may have advantages over another. Unfortunately, there is no clear winner. In general, the order of preference in terms of power is

1. Pillai's Trace
2. Wilks' Lambda
3. Hotelling-Lawley Trace
4. Roy's Maximum Root.

When there is a large deviation from the null hypothesis and the eigenvalues differ widely, the order of preference is the reverse (Seber 1984).

## Univariate Tests and the Test for Sphericity

There are cases, such as a repeated measures model, that allow transformation of a multivariate problem into a univariate problem (Huynh and Feldt 1970). Using univariate tests in a multivariate context is valid

- if the response design matrix  $M$  is orthonormal ( $M'M = \text{Identity}$ ).
- if  $M$  yields more than one response the coefficients of each transformation sum to zero.
- if the *sphericity* condition is met. The sphericity condition means that the  $M$ -transformed responses are uncorrelated and have the same variance.  $M'\Sigma M$  is proportional to an identity matrix, where  $\Sigma$  is the covariance of the  $Y$ 's.

If these conditions hold, the diagonal elements of the  $E$  and  $H$  test matrices sum to make a univariate sums of squares for the denominator and numerator of an  $F$ -test. Note that if the above conditions do not hold, then an error message will be produced. In the case of `Golf Balls.jmp`, an identity matrix is specified as the  $M$ -matrix. Identity matrices cannot be transformed to a full rank matrix after centralization of column vectors and ortho-normalization. So the univariate request is ignored.

To view univariate and sphericity tests, open `Dogs.jmp` from the sample data directory. Click on the MANOVA script or refer to Figure 7.1 to see the model specification dialog.

Check the **Univariate Tests Also** check box in the Response Specification dialog, specify **Repeated Measures** from the **Response Design** popup menu, and enter Time for the **Y Name** and check the box beside **Univariate Tests Also**. You will see a Sphericity Test table, like the one shown in Figure 7.9, and adjusted univariate  $F$ -tests in the multivariate report tables.

---

**Figure 7.9** Sphericity test

<b>Sphericity Test</b>	
Mauchly Criterion	0.1752641
ChiSquare	16.930873
DF	5
Prob >Chisq	0.0046328

---

The sphericity test checks the appropriateness of an unadjusted univariate  $F$ -test for the within-subject effects using the Mauchly criterion to test the sphericity assumption (Anderson 1958). The sphericity test and the univariate tests are always done using an orthonormalized  $M$  matrix. You interpret the sphericity test as follows:

- If the sphericity Chi-square test is not significant, you can use the unadjusted univariate  $F$ -tests.
- If the sphericity test is significant, use the multivariate or the adjusted univariate tests.

The univariate  $F$ -statistic has an approximate  $F$ -distribution even without sphericity, but the degrees of freedom for numerator and denominator are reduced by some fraction epsilon ( $\epsilon$ ). Box (1954), Geisser and Greenhouse (1958), and Huynh-Feldt (1976) offer techniques for estimating the epsilon

degrees-of-freedom adjustment. Muller and Barton (1989) recommend the Geisser-Greenhouse version, based on a study of power.

The epsilon adjusted tests in the multivariate report are labeled G-G (Greenhouse-Geisser) or H-F (Huynh-Feldt), with the epsilon adjustment shown in the value column.

---

## Multivariate Model with Repeated Measures

One common use of multivariate fitting is to analyze data with repeated measures, also called *longitudinal data*. A subject is measured repeatedly across time, and the data are arranged so that each of the time measurements form a variable. Because of correlation between the measurements, data should not be stacked into a single column and analyzed as a univariate model unless the correlations form a pattern termed *sphericity*. See the previous section, “[Univariate Tests and the Test for Sphericity](#),” p. 144, for more details about this topic.

With repeated measures, the analysis is divided into two layers:

- Between-subject (or across-subject) effects are modeled by fitting the sum of the repeated measures columns to the model effects. This corresponds to using the **Sum** response function, which is an M-matrix that is a single vector of 1's.
- Within-subjects effects (repeated effects, or time effects) are modeled with a response function that fits differences in the repeated measures columns. This analysis can be done using the **Contrast** response function or any of the other similar differencing functions: **Polynomial**, **Helmert**, **Profile**, or **Mean**. When you model differences across the repeated measures, think of the differences as being a new within-subjects effect, usually time. When you fit effects in the model, interpret them as the interaction with the within-subjects effect. For example, the effect for Intercept becomes the Time (within-subject) effect, showing overall differences across the repeated measures. If you have an effect A, the within-subjects tests are interpreted to be the tests for the A\*Time interaction, which model how the differences across repeated measures vary across the A effect.

[Table 7.1 “Corresponding Multivariate and Univariate Tests,” p. 146](#), shows the relationship between the response function and the model effects compared with what a univariate model specification would be. Using both the **Sum** (between-subjects) and **Contrast** (within-subjects) models, you should be able to reconstruct the tests that would have resulted from stacking the responses into a single column and obtaining a standard univariate fit.

There is a direct and an indirect way to perform the repeated measures analyses:

- The direct way is to use the popup menu item Repeated Measures. This prompts you to name the effect that represents the within-subject effect across the repeated measures. Then it fits both the **Contrast** and the **Sum** response functions. An advantage of this way is that the effects are labeled appropriately with the within-subjects effect name.
- The indirect way is to specify the two response functions individually. First, do the **Sum** response function and second, do either **Contrast** or one of the other functions that model differences. You will have to remember to associate the within-subjects effect with the model effects in the contrast fit.

## Repeated Measures Example

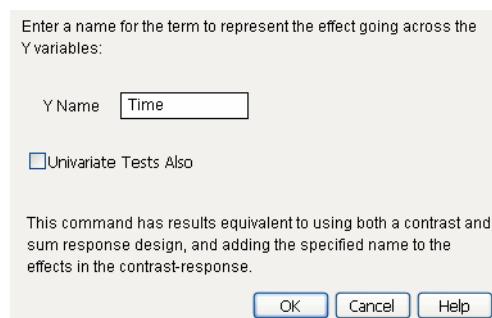
As an example, consider a study by Cole and Grizzle (1966). The results are in the Dogs.jmp table in the sample data folder. Sixteen dogs are assigned to four groups defined by variables drug and dep1, each having two levels. The dependent variable is the blood concentration of histamine at 0, 1, 3, and 5 minutes after injection of the drug. The log of the concentration is used to minimize the correlation between the mean and variance of the data.

Assign LogHist0, LogHist1, LogHist3, and LogHist5 to the **Y** role. Select drug and dep1 and select **Full Factorial** from the Macros pop-up menu. Select **Manova** from the Personality menu, and click **Run**. Select **Repeated Measures** from the Choose Response pop-up menu.

A window (see Figure 7.10) appears for specifying an arbitrary name to represent the effect across the Y variables. This is usually a time effect, so enter Time.

---

**Figure 7.10** Repeated Measures Window




---

If you check the **Univariate Tests Also** check box, the report includes univariate tests, which are calculated as if the responses were stacked into a single column.

[Table 7.1 “Corresponding Multivariate and Univariate Tests,” p. 146](#), shows how the multivariate tests for a **Sum** and **Contrast** response designs correspond to how univariate tests would be labeled if the data for columns LogHist0, LogHist1, LogHist3, and LogHist5 were stacked into a single *Y* column, with the new rows identified with a nominal grouping variable, Time.

**Table 7.1** Corresponding Multivariate and Univariate Tests

Sum M-Matrix Between Subjects		Contrast M-Matrix Within Subjects	
Multivariate Test	Univariate Test	Multivariate Test	Univariate Test
intercept	intercept	intercept	time
drug	drug	drug	time*drug
dep1	dep1	dep1	time*dep1

The between-subjects analysis is produced first. This analysis is the same (except titling) as it would have been if **Sum** had been selected on the popup menu.

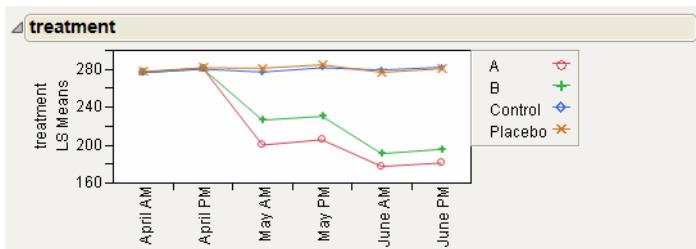
The within-subjects analysis is produced next. This analysis is the same (except titling) as it would have been if **Contrast** had been selected on the popup menu, though the within-subject effect name (**Time**) has been added to the effect names in the report. Note that the position formerly occupied by **Intercept** is **Time**, because the intercept term is estimating overall differences across the repeated measurements.

## A Compound Multivariate Model

JMP can handle data with layers of repeated measures. For example, see the Cholesterol.jmp data table. Groups of five subjects belong to one of four treatment groups called A, B, Control, and Placebo. Cholesterol was measured in the morning and again in the afternoon once a month for three months (the data are fictional). In this example the response columns are arranged chronologically with time of day within month.

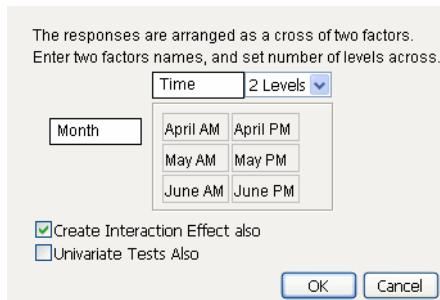
To analyze this experimental design, complete the Fit Model dialog by declaring treatment as the only effect and all six response columns as *Y* variables. Specify the Personality as **Manova**. When you click **Run**, the four tables for the initial fit appear. Part of the report is shown in Figure 7.11.

**Figure 7.11** Graph of Cholesterol



You can see in the Least Squares Means table that the four treatment groups began the study with very similar mean cholesterol values. The A and B treatment groups appear to have lower cholesterol values at the end of the trial period. The control and placebo groups remain unchanged.

Next, choose **Compound** from the response design popup menu. You complete this dialog to tell JMP how the responses are arranged in the data table and the number of levels of each response. In the cholesterol example, the time of day columns are arranged within month. Therefore, you name time of day as one factor and the month effect as the other factor. Note that testing the interaction effect is optional. Complete the window as shown in Figure 7.12.

**Figure 7.12** Compound Dialog

When you click **OK**, the tests for each effect display. Parts of the report are shown in Figure 7.13.

- With a p-value of 0.6038, the interaction between Time and treatment is not significant. This means there is no difference in treatment between AM and PM. Note that since Time has two levels (AM and PM), the exact F-test is given.
- With p-values of <.0001, the interaction between Month and treatment is significant. This suggests that the differences between treatment groups change with Month. The LS Means plot shown above indicates no difference among the groups in April, but the difference between (A,B) and (Control, Placebo) becomes large in May and even larger in June.
- The interaction effect between Month, Time, and treatment is not statistically significant.

**Figure 7.13** Cholesterol Study Results

**Compound**

Time

- M Matrix
- M-transformed Parameter Estimates
- Whole Model
- Intercept
- treatment

Test	Value	Exact F	NumDF	DenDF	Prob>F
F Test	0.1188721	0.6340	3	16	0.6038

**Compound**

Month

- M Matrix
- M-transformed Parameter Estimates
- Whole Model
- Intercept
- treatment

Test	Value	Approx. F	NumDF	DenDF	Prob>F
Wilks' Lambda	0.013025	38.8109	6	30	<.0001*
Pillai's Trace	1.3128917	10.1907	6	32	<.0001*
Hotelling-Lawley	50.753209	123.4368	6	18.326	<.0001*
Roy's Max Root	50.255302	268.0283	3	16	<.0001*

**Compound**

Time\*Month

- M Matrix
- M-transformed Parameter Estimates
- Whole Model
- Intercept
- treatment

Test	Value	Approx. F	NumDF	DenDF	Prob>F
Wilks' Lambda	0.6823742	1.1435	6	30	0.3619
Pillai's Trace	0.3582813	1.1638	6	32	0.3498
Hotelling-Lawley	0.4785668	1.1639	6	18.326	0.3671
Roy's Max Root	0.4008469	2.1378	3	16	0.1355

---

## Commands for Response Type and Effects

The **Custom Test** popup option displays the Custom Test dialog for setting up custom tests of effect levels. See “[Custom Test](#),” p. 63 in the “Standard Least Squares: Perspectives on the Estimates” chapter for a description of how to use the Custom Test dialog to create custom tests.

The popup menu icon beside each effect name gives you the commands shown here, to request additional information about the multivariate fit:

**Test Details** displays the eigenvalues and eigenvectors of the  $E^{-1}H$  matrix used to construct multivariate test statistics.

**Centroid Plot** plots the centroids (multivariate least-squares means) on the first two canonical variables formed from the test space.

**Save Canonical Scores** saves variables called Canon[1], Canon[2], and so on, as columns in the current data table. These columns have both the values and their formulas.

**Contrast** performs the statistical contrasts of treatment levels that you specify in the contrasts dialog.

**Note:** The **Contrast** command is the same as for regression with a single response. See the “[LSMeans Contrast](#),” p. 43 in the “Standard Least Squares: Introduction” chapter, for a description and examples of the **LSMeans Contrast** commands.

## Test Details (Canonical Details)

As an example, open Fisher’s Iris data, Iris.jmp, found in the Sample Data folder (Mardia, Kent, and Bibby 1979). The Iris data have three levels of Species named Virginica, Setosa, and Versicolor. There are four measures (Petal length, Petal width, Sepal length, and Sepal width) taken on each sample. Fit a MANOVA model for the Species effect, with the four petal and sepal measures assigned as responses (Y). Choose the **Identity** response model. Then select the **Test Details** command from the **Species** popup menu.

The eigenvalues, eigenvectors, and canonical correlations appear (see Figure 7.14).

**Figure 7.14** Test Details

The screenshot shows the 'Test Details' report for the 'Species' data. It includes the following sections:

- Test** section: Lists Wilks' Lambda (0.0234386, F=199.1453, p <.0001\*), Pillai's Trace (1.1918988, F=53.4665, p <.0001\*), Hotelling-Lawley (32.47732, F=582.1970, p <.0001\*), and Roy's Max Root (32.191929, F=1166.9574, p <.0001\*).
- Canonical** section: Lists Eigenvalues and Correlations. Eigenvalues are 32.1919292, 0.28539104, 1.235e-15, and -6.174e-16. Corresponding Corrs are 0.98482089, 0.47119702, 0, and 0.
- Eigvec** section: Lists Eigenvectors for Sepal length, Sepal width, Petal length, and Petal width. For Sepal length, the vector is (-0.0684059, 0.00198791, -0.2350196, 0.1176771). For Sepal width, it is (-0.1265612, 0.1785267, 0.21657608, 0.04510419). For Petal length, it is (0.18155288, -0.0768636, 0.23964446, 0.06563465). For Petal width, it is (0.23180286, 0.23417227, -0.2865277, -0.2438953).

**Eigenvalue** lists the eigenvalues of the  $E^{-1}H$  matrix used in computing the multivariate test statistics.

**Canonical Corr** lists the canonical correlations associated with each eigenvalue. This is the canonical correlation of the transformed responses with the effects, corrected for all other effects in the model.

**Eigvec** lists the eigenvectors of the  $E^{-1}H$  matrix, or equivalently of  $(E + H)^{-1}H$ .

## The Centroid Plot

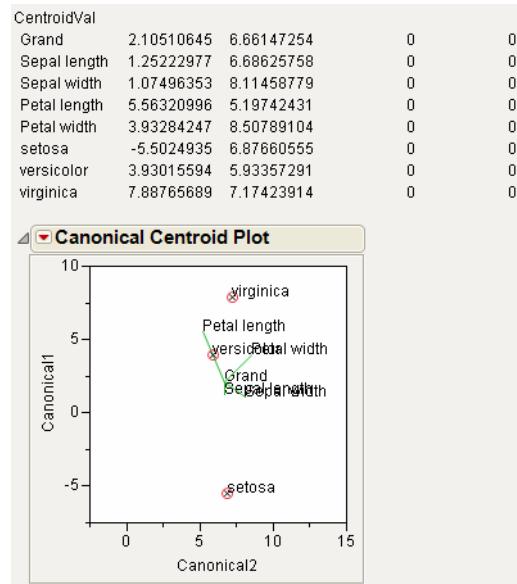
The **Centroid Plot** command (accessed from the red triangle next to Species) plots the centroids (multivariate least-squares means) on the first two canonical variables formed from the test space, as in Figure 7.15. The first canonical axis is the vertical axis so that if the test space is only one dimensional the centroids align on a vertical axis. The centroid points appear with a circle corresponding to the 95%

confidence region (Mardia, Kent, and Bibby, 1980). When centroid plots are created under effect tests, circles corresponding to the effect being tested appear in red. Other circles appear blue. Biplot rays show the directions of the original response variables in the test space.

Click the **Centroid Val** disclosure icon to show additional information, shown in Figure 7.15.

The first canonical axis with an eigenvalue accounts for much more separation than does the second axis. The means are well separated (discriminated), with the first group farther apart than the other two. The first canonical variable seems to load the petal length variables against the petal width variables. Relationships among groups of variables can be verified with Biplot Rays and the associated eigenvectors.

**Figure 7.15** Centroid Plot and Centroid Values



## Save Canonical Scores (Canonical Correlation)

Canonical correlation analysis is not a specific command, but it can be done by a sequence of commands in the multivariate fitting platform:

- Choose the **Fit Model** command.
- Specify Y's and X's, select **Manova** from the **Personality** drop-down menu, and click **Run**.
- Choose the **Identity** from the Choose Response pop-up menu. Click **Run**.
- Select **Test Details** and then **Save Canonical Scores** from the Whole Model popup commands. The details list the canonical correlations (Canonical Corr) next to the eigenvalues. The saved variables are called **Canon[1]**, **Canon[2]**, and so on. These columns contain both the values and their formulas.

- To obtain the canonical variables for the  $X$  side, repeat the same steps but interchange the  $X$  and  $Y$  variables. If you already have the columns Canon[n] appended to the data table, the new columns are called Canon[1] (or another number) that makes the name unique.

For example, try the Linnerud data from Rawlings (1988) called Exercise.jmp found in the Sample Data folder. It has the physiological  $X$  variables weight, waist, and pulse, and the exercise  $Y$  variables chins, situps, and jumps. Figure 7.16 shows how the Whole Model table looks after the details are requested.

**Figure 7.16** Canonical Correlations

Whole Model					
Test	Value	Approx. F	NumDF	DenDF	Prob>F
Wilks' Lambda	0.3503905	2.0482	9	34.223	0.0635
Pillai's Trace	0.6784815	1.5587	9	48	0.1551
Hotelling-Lawley	1.7719415	2.6397	9	19.053	0.0357*
Roy's Max Root	1.7247387	9.1986	3	16	0.0009*
Canonical					
Eigenvalue	Corr				
1.72473874	0.79560815				
0.0419084	0.20055604				
0.00529433	0.07257029				
Eigvec					
chins	0.02503681	-0.016636	0.05641878		
situps	0.00637953	0.0004622	-0.004547		
jumps	-0.0052909	0.0048507	0.0018787		

The output canonical variables use the eigenvectors shown as the linear combination of the  $Y$  variables. For example, the formula for canon[1] is

$$0.02503681 * \text{chins} + 0.00637953 * \text{situps} + -0.0052909 * \text{jumps}$$

This canonical analysis does not produce a standardized variable with mean 0 and standard deviation 1, but it is easy to define a new standardized variable with the calculator that has these features.

## Discriminant Analysis

Discriminant analysis is a method of predicting some level of a one-way classification based on known values of the responses. The technique is based on how close a set of measurement variables are to the multivariate means of the levels being predicted. Discriminant analysis is more fully implemented using the Discriminant Platform (“Discriminant Analysis,” p. 471).

In JMP you specify the measurement variables as  $Y$  effects and the classification variable as a single  $X$  effect. The multivariate fitting platform gives estimates of the means and the covariance matrix for the data, assuming the covariances are the same for each group. You obtain discriminant information with the **Save Discrim** command in the popup menu next to the MANOVA platform name. This command saves distances and probabilities as columns in the current data table using the initial  $E$  and  $H$  matrices.

For a classification variable with  $k$  levels, JMP adds  $k$  distance columns,  $k$  classification probability columns, the predicted classification column, and two columns of other computational information to the current data table.

Again use Fisher's Iris data (*Iris.jmp* in the Sample Data folder) as found in Mardia, Kent, and Bibby. There are  $k = 3$  levels of species and four measures on each sample. The **Save Discrim** command in the Manova Fit drop-down menu adds the following nine columns to the *Iris.jmp* data table.

**SqDist[0]** is the quadratic form needed in the Mahalanobis distance calculations.

**SqDist[setosa]** is the Mahalanobis distance of the observation from the Setosa centroid.

**SqDist[versicolor]** is the Mahalanobis distance of the observation from the Versicolor centroid.

**SqDist[virginica]** is the Mahalanobis distance of the observation from the Virginica centroid.

**Prob[0]** is the sum of the negative exponentials of the Mahalanobis distances, used below.

**Prob[setosa]** is the probability of being in the Setosa category.

**Prob[versicolor]** is the probability of being in the Versicolor category.

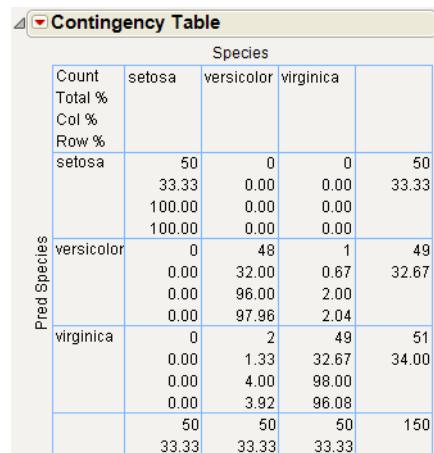
**Prob[virginica]** is the probability of being in the Virginica category.

**Pred Species** is the species that is most likely from the probabilities.

You can use the new columns in the data table with other JMP platforms to summarize the discriminant analysis with reports and graphs. For example, if you use the Fit Y by X platform and specify **Pred Species** as *X* and **Species** as *Y* variable, the Contingency Table report shown in Figure 7.17 summarizes the discriminant classifications. Three misclassifications are identified.

---

**Figure 7.17** Two-Way Table of Predicted and Actual Species



The screenshot shows a JMP Contingency Table report. The title bar says "Contingency Table". The table has "Species" as the column header and "Pred Species" as the row header. The columns are labeled "Count", "setosa", "versicolor", "virginica", and an empty column. The rows are labeled "setosa", "versicolor", and "virginica". The data is as follows:

		Species				
		Count	setosa	versicolor	virginica	
		Total %				
		Col %				
		Row %				
Pred Species	setosa	50	0	0	50	
		33.33	0.00	0.00	33.33	
		100.00	0.00	0.00	100.00	
		100.00	0.00	0.00	100.00	
Pred Species	versicolor	0	48	1	49	
		0.00	32.00	0.67	32.67	
		0.00	96.00	2.00	98.00	
		0.00	97.96	2.04	96.08	
Pred Species	virginica	0	2	49	51	
		0.00	1.33	32.67	34.00	
		0.00	4.00	98.00	99.00	
		0.00	3.92	96.08	96.08	
	50	50	50	150		
	33.33	33.33	33.33	33.33		

---



# Chapter 8

## Fitting Dispersion Effects with the LogLinear Variance Model

### The Fit Model Platform

---

This fitting platform allows you to model both the expected value and the variance of a response using regression models. The log of the variance is fit to one linear model and the expected response is fit to a different linear model simultaneously.

**Note:** The estimates are demanding in their need for a lot of well-designed, well-fitting data. You need more data to fit variances than you do means.

---

For many engineers, the goal of an experiment is not to maximize or minimize the response itself, but to aim at a target response and achieve minimum variability. The loglinear variance model provides a very general and effective way to model variances, and can be used for unreplicated data, as well as data with replications.

Modeling dispersion effects is not very widely covered in textbooks, with the exception of the Taguchi framework. In a Taguchi-style experiment, this is handled by taking multiple measurements across settings of an outer array, constructing a new response which measures the variability off-target across this outer array, and then fitting the model to find out the factors that produce minimum variability. This kind of modeling requires a specialized design that is a complete cartesian product of two designs. The method of this chapter models variances in a more flexible, model-based approach. The particular performance statistic that Taguchi recommends for variability modeling is  $STD = -\log(s)$ . In JMP's methodology, the  $\log(s^2)$  is modeled and combined with a model that has a mean. The two are basically equivalent, since  $\log(s^2) = 2 \log(s)$ .

# Contents

The Loglinear Variance Model .....	157
Estimation Method .....	157
Loglinear Variance Models in JMP .....	157
Model Specification.....	157
Example .....	157
Displayed Output .....	159
Platform Options .....	160
Examining the Residuals .....	161
Profiling the Fitted Model.....	162
Comments .....	164

---

## The Loglinear Variance Model

The loglinear-variance model (Harvey 1976, Cook and Weisberg 1983, Aitken 1987, Carroll and Ruppert 1988) provides a neat way to model the variance simply through a linear model. In addition to having regressor terms to model the mean response, there are regressor terms in a linear model to model the log of the variance:

$$\text{mean model: } E(y) = X\beta$$

$$\text{variance model: } \log(\text{Variance}(y)) = Z\lambda,$$

or equivalently

$$\text{Variance}(y) = \exp(Z\lambda)$$

where the columns of  $X$  are the regressors for the mean of the response, and the columns of  $Z$  are the regressors for the variance of the response. The regular linear model parameters are represented by  $\beta$ , and  $\lambda$  represents the parameters of the variance model.

## Estimation Method

Log-linear variance models are estimated using REML.

---

## Loglinear Variance Models in JMP

This section introduces a new kind of effect, a *dispersion* effect, labeled as a *log-variance* effect, that can model changes in the variance of the response. This is implemented in the Fit Model platform by a fitting personality called the *Loglinear Variance* personality.

## Model Specification

Log-linear variance effects are specified in the Fit Model dialog by highlighting them and selecting **LogVariance Effect** from the **Attributes** drop-down menu. **&LogVariance** appears at the end of the effect. When you use this attribute, it also changes the fitting **Personality** at the top to **LogLinear Variance**. If you want an effect to be used for both the mean and variance of the response, then you must specify it twice, once with the **LogVariance** option.

The effects you specify with the log-variance attribute become the effects that generate the  $Z$ s in the model, and the other effects become the  $X$ s in the model.

## Example

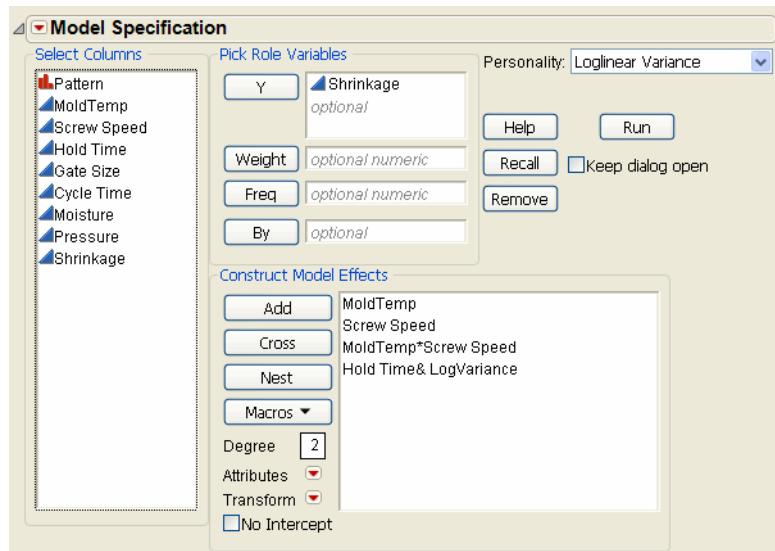
The data table **InjectionMolding.jmp** contains the experimental results from a 7-factor  $2^{7-3}$  fractional factorial design with four added centerpoints [from Myers and Montgomery, 1995, page 519, originally Montgomery, 1991]. Preliminary investigation determined that the mean response only seemed to vary with

the first two factors, Mold Temperature, and Screw Speed, and the variance seemed to be affected by Holding Time.

**Figure 8.1** Injection Molding Data

	Pattern	MoldTemp	Screw Speed	Hold Time	Gate Size	Cycle Time	Moisture	Pressure	Shrinkage
1	-----	-1	-1	-1	-1	-1	-1	-1	6
2	+----++	1	-1	-1	-1	1	-1	1	10
3	-+---+-+	-1	1	-1	-1	1	1	-1	32
4	+++++-	1	1	-1	-1	-1	1	1	60
5	-+---+-	-1	-1	1	-1	1	1	1	4
6	++-++-	1	-1	1	-1	-1	1	-1	15
7	-+---++	-1	1	1	-1	-1	-1	1	26
8	++-+--	1	1	1	-1	1	-1	-1	60
9	-+----+*	-1	-1	-1	1	-1	1	1	8
10	++-+--	1	-1	-1	1	1	1	-1	12
11	-+---+-	-1	1	-1	1	1	-1	1	34
12	++-+--	1	1	-1	1	-1	-1	-1	60
13	-+---+*	-1	-1	1	1	1	-1	-1	16
14	++-+--	1	-1	1	1	-1	-1	1	5
15	-+----+*	-1	1	1	1	-1	1	-1	37
16	++++-++*	1	1	1	1	1	1	1	52
17	00000000	0	0	0	0	0	0	0	25
18	00000000	0	0	0	0	0	0	0	29
19	00000000	0	0	0	0	0	0	0	24
20	00000000	0	0	0	0	0	0	0	27

To proceed with the analysis, select **Analyze > Fit Model** and complete the Fit Model dialog as shown in Figure 8.2. After Hold Time was added to the model, it was selected and changed to a LogVariance effect through the **Attributes** popup menu. This also forced the fitting personality to change to **Loglinear Variance**. Click the **Run** button to start the fitting.

**Figure 8.2** Fit Model Dialog

## Displayed Output

The top portion of the resulting report shows the fitting of the Expected response, with reports similar to standard least squares, though actually derived from restricted maximum likelihood (REML).

**Figure 8.3** Mean Model Output

Loglinear Variance Fit				
Mean Model for Shrinkage				
Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	27.582516	0.380697	72.45	<.0001*
MoldTemp	7.683713	0.392907	19.56	<.0001*
Screw Speed	18.673515	0.392907	47.53	<.0001*
MoldTemp*Screw Speed	5.765297	0.392907	14.67	<.0001*
Fixed Effect Tests				
Source	Nparm	DF	DFDen	F Ratio
MoldTemp	1	1	5.758	382.4387
Screw Speed	1	1	5.758	2258.768
MoldTemp*Screw Speed	1	1	5.758	215.3094
Prob > F				
				<.0001*

**Figure 8.4** Variance Model Output

Variance Model for Shrinkage							
Likelihood Ratio Test for Equal Variance							
Source	-2'LogLikelihood						
Equal Variances Initially	101.76083063						
After Fitted Variances	90.712808548						
Difference: Chi-sq	11.048022086						
Degrees of Freedom	1						
Prob > ChiSquare	0.0008878186						
Variance Parameter Estimates							
Term	Estimate	Std Error	ChiSquare	Prob>ChiSq	Lower	Upper	
Hold Time	1.5788108	0.412603	11.0480	0.0009*	0.7057476	2.37325	
Residual	6.6912718	2.445524	7.4864	0.0062*	3.2689762	13.696373	
Variance Effect Likelihood Ratio Tests							
Source	Test Type	DF	ChiSquare	Prob>ChiSq			
Hold Time	Likelihood	1	11.0480	0.0009*			

The second portion of the report shows the fit of the variance model. The **Variance Parameter Estimates** report shows the estimates and relevant statistics. Two hidden columns are provided:

- The hidden column **exp(Estimate)** is the exponential of the estimate. So, if the factors are coded to have +1 and -1 values, then the +1 level for a factor would have the variance multiplied by the **exp(Estimate)** value and the -1 level would have the variance multiplied by the reciprocal of this column. To see a hidden column, right-click on the report and select the name of the column from the **Columns** menu that appears.
- The hidden column labeled **exp(2|Estimate|)** is the ratio of the higher to the lower variance if the regressor has the range -1 to +1.

The report also shows the standard error, chi-square, *p*-value, and profile likelihood confidence limits of each estimate. The residual parameter is the overall estimate of the variance, given all other regressors are zero.

Does the variance model fit significantly better than the original model? The likelihood ratio test for this question compares the fitted model with the model where all parameters are zero except the intercept, the model of equal-variance. In this case the *p*-value is highly significant. Changes in Hold Time change the variance.

The **Variance Effect Likelihood Ratio Tests** refit the model without each term in turn to create the likelihood ratio tests. These are generally more trusted than Wald tests.

## Platform Options

To access platform options, click on the red triangle menu next to Loglinear Variance Fit.

### Save Columns

Each of these commands creates one or more columns in the data table.

**Prediction Formula** creates a new column, called colname Mean, containing the predicted values for the mean computed by the specified model.

**Variance Formula** creates a new column, called colname Variance, containing the predicted values for the variance computed by the specified model.

**Std Dev Formula** creates a new column, called colname Std Dev, containing the predicted values for the standard deviation computed by the specified model.

**Residuals** creates a new column called colname Residual containing the residuals, which are the observed response values minus predicted values.

**Studentized Residuals** creates a new column called colname Studentized Resid. The new column values are the residuals divided by their standard error.

**Std Error of Predicted** creates a new column, called Std Err Pred colname, containing the standard errors of the predicted values.

**Std Error of Individual** creates a new column, called Std Err Indiv colname, containing the standard errors of the individual predicted values.

**Mean Confidence Interval** creates two new columns, Lower 95% Mean colname and Upper 95% Mean colname that are the bounds for a confidence interval for the prediction mean.

**Indiv Confidence Interval** creates two new columns, Lower 95% Indiv colname and Upper 95% Indiv colname that are the bounds for a confidence interval for the prediction mean.

## Row Diagnostics

**Plot Actual by Predicted** displays the observed values by the predicted values of  $Y$ . This is the leverage plot for the whole model.

**Plot Studentized Residual by Predicted** displays the Studentized residuals by the predicted values of  $Y$ .

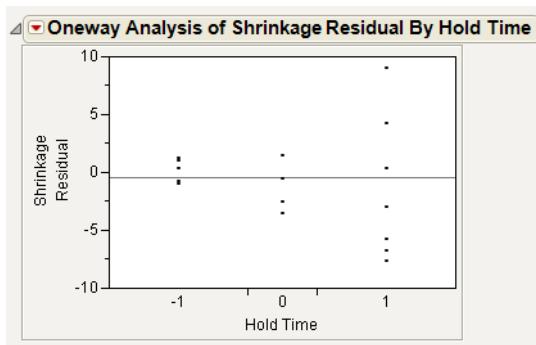
**Plot Studentized Residual by Row** displays the Studentized residuals by row.

## Profilers

Profiler, Contour Profiler, and Surface Profiler are the standard JMP profilers, detailed in the “[Standard Least Squares: Exploring the Prediction Equation](#)” chapter.

## Examining the Residuals

To see the dispersion effect, we invoked the Oneway platform on Shrinkage Residual (produced using the **Save Columns > Residuals** command) by Hold Time: With this plot it is easy to see the variance go up as Hold Time is increased. This is done by treating Hold Time as a nominal factor, though it is originally continuous in the fit above.

**Figure 8.5** Residual by Dispersion Effect

## Profiling the Fitted Model

The **Profiler**, **Contour Profiler**, or **Surface Profiler** can be used to gain more insight on the fitted model. Each can be selected from the platform drop-down menu under the **Profilers** submenu. For example, suppose that the goal was to find the factor settings that achieved a target of 36.35 for the response, but at the smallest variance. Fit the models and choose Profiler from the report menu. For example, Figure 8.6 shows the Profiler set up to match a target value for a mean and to minimize variance.

One of the best ways to see the relationship between the mean and the variance (both modeled with the LogVariance personality) is through looking at the individual prediction confidence intervals about the mean. To see prediction intervals in the Profiler, select **Prediction Intervals** from its drop-down menu. Regular confidence intervals (those shown by default in the Profiler) do not show information about the variance model as well as individual prediction confidence intervals do. Prediction intervals show both the mean and variance model in one graph.

If  $Y$  is the modeled response, and you want a prediction interval for a new observation at  $x_n$ , then

$$s^2|x_n = s^2_Y|x_n + s^2_{\hat{Y}}|x_n$$

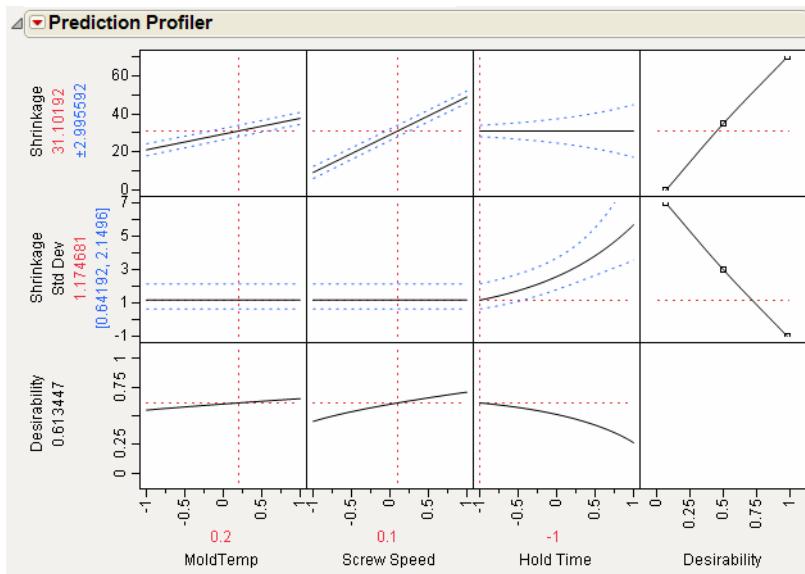
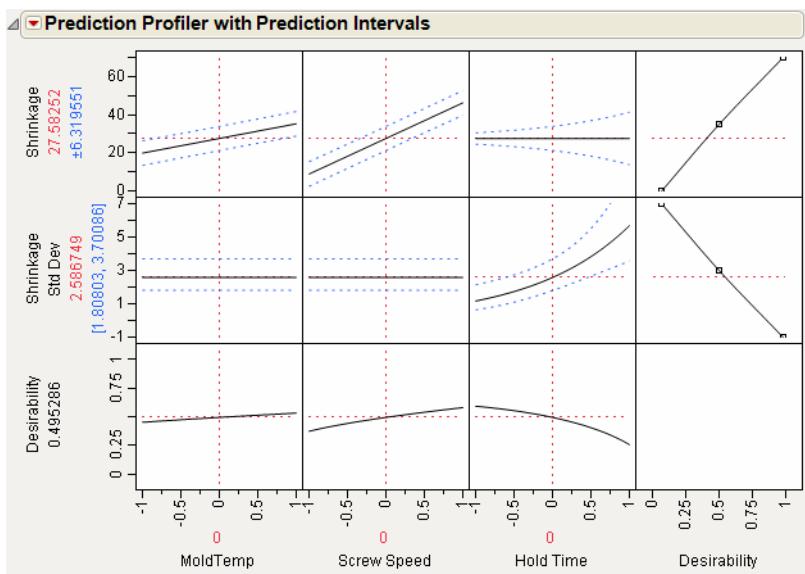
where

$s^2|x_n$  is the variance for the individual prediction at  $x_n$

$s^2_Y|x_n$  is the variance of the distribution of  $Y$  at  $x_n$

$s^2_{\hat{Y}}|x_n$  is the variance of the sampling distribution of  $\hat{Y}$ , and is also the variance for the mean.

Because the variance of the individual prediction contains the variance of the distribution of  $Y$ , the effects of the changing variance for  $Y$  can be seen. Not only are the individual prediction intervals wider, but they can change shape with a change in the variance effects. Figure 8.7 shows prediction intervals for the situation in Figure 8.6.

**Figure 8.6** Profiler to Match Target and Minimize Variance**Figure 8.7** Prediction Intervals

## Comments

Every time another parameter is estimated for the mean model, at least one more observation is needed, and preferably more. But with variance parameters, several more observations for each variance parameter are needed to obtain reasonable estimates. It takes more data to estimate variances than it does means.

The log-linear variance model is a very flexible way to fit dispersion effects, and the method deserves much more attention than it has received so far in the literature.

# Chapter 9

## Logistic Regression for Nominal and Ordinal Response

### The Fit Model Platform

---

If the model response is nominal, the Fit Model platform fits a linear model to a multi-level logistic response function using maximum likelihood. Likelihood-ratio statistics and Lack of Fit tests are computed for the whole model. Likelihood-ratio tests and Wald tests can be computed for each effect in the model. Options include confidence limits for the maximum likelihood parameter estimates. When the response is binary, odds ratios (with confidence intervals) are available.

If the response variable is ordinal, the platform fits the cumulative response probabilities to the logistic distribution function of a linear model using maximum likelihood. Likelihood-ratio test statistics are provided for the whole model and lack of fit.

For simple main effects, you can use the Fit Y by X platform described in the *Basic Analysis and Graphing* book to see a cumulative logistic probability plot for each effect. Details for these models are discussed in the *Basic Analysis and Graphing* book, and in the appendix “[Statistical Details](#),” p. 607.

# Contents

Introduction to Logistic Models .....	167
The Statistical Report .....	168
Logistic Plot .....	169
Iteration History .....	170
Whole Model Test .....	170
Lack of Fit Test (Goodness of Fit) .....	172
Parameter Estimates .....	173
Likelihood-ratio Tests .....	174
Platform Options .....	175
Plot Options .....	175
Likelihood Ratio Tests .....	175
Wald Tests for Effects .....	175
Confidence Intervals .....	175
Odds Ratios (Nominal Responses Only) .....	176
Inverse Prediction .....	179
Save Commands .....	181
ROC Curve .....	182
Lift Curve .....	183
Confusion Matrix .....	184
Profiler .....	184
Validation .....	184
Nominal Logistic Model Example: The Detergent Data .....	184
Ordinal Logistic Example: The Cheese Data .....	188
Quadratic Ordinal Logistic Example: Salt in Popcorn Data .....	193
What to Do If Your Data Are Counts in Multiple Columns .....	195

---

## Introduction to Logistic Models

Logistic regression fits nominal  $Y$  responses to a linear model of  $X$  terms. To be more precise, it fits probabilities for the response levels using a logistic function. For two response levels the function is

$$P(Y = r_1) = (1 + e^{-Xb})^{-1} \text{ where } r_1 \text{ is the first response}$$

or equivalently

$$\log\left(\frac{P(Y = r_1)}{P(Y = r_2)}\right) = Xb \text{ where } r_1 \text{ and } r_2 \text{ are the two responses}$$

For  $r$  nominal responses, where  $r > 2$ , it fits  $r - 1$  sets of linear model parameters of the form

$$\log\left(\frac{P(Y = j)}{P(Y = r)}\right) = X_j b$$

The fitting principle of maximum likelihood means that the  $\beta$ s are chosen to maximize the joint probability attributed by the model to the responses that did occur. This fitting principle is equivalent to minimizing the negative log-likelihood (-LogLikelihood)

$$\text{Loss} = -\text{logLikelihood} = \sum_{i=1}^n -\log(\text{Prob}(i\text{th row has the } y_j\text{-th response}))$$

as attributed by the model.

As an example, consider an experiment that was performed on metal ingots prepared with different heating and soaking times. The ingots were then tested for readiness to roll. See Cox (1970). The Ingots.jmp data table in the Sample Data folder has the experimental results.

	heat	soak	ready	count
1	7	1.0	1	0
2	7	1.0	0	10
3	7	1.7	1	0
4	7	1.7	0	17
5	7	2.2	1	0
6	7	2.2	0	7
7	7	2.8	1	0
8	7	2.8	0	12
9	7	4.0	1	0
10	7	4.0	0	9
11	14	1.0	1	0
12	14	1.0	0	31
13	14	1.7	1	0
14	14	1.7	0	43
15	14	2.2	1	2
16	14	2.2	0	31
17	14	2.8	1	0
18	14	2.8	0	31
19	14	4.0	1	0

The categorical variable called **ready** has values 1 and 0 for readiness and not readiness to roll, respectively.

The Fit Model platform fits the probability of the *not readiness* (0) response to a logistic cumulative distribution function applied to the linear model with regressors **heat** and **soak**:

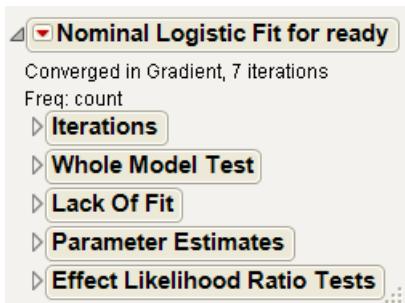
$$\text{Probability (not ready to roll)} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \text{heat} + \beta_2 \text{soak})}}$$

The parameters are estimated by minimizing the sum of the negative logs of the probabilities attributed to the observations by the model (maximum likelihood).

To analyze this model, select **Analyze > Fit Model**. The **ready** variable is **Y**, the response, and **heat** and **soak** are the model effects. The **count** column is the **Freq** variable. When you click **Run**, iterative calculations take place. When the fitting process converges, the nominal/ordinal regression report appears. The following sections discuss the report layout and statistical tables, and show examples.

## The Statistical Report

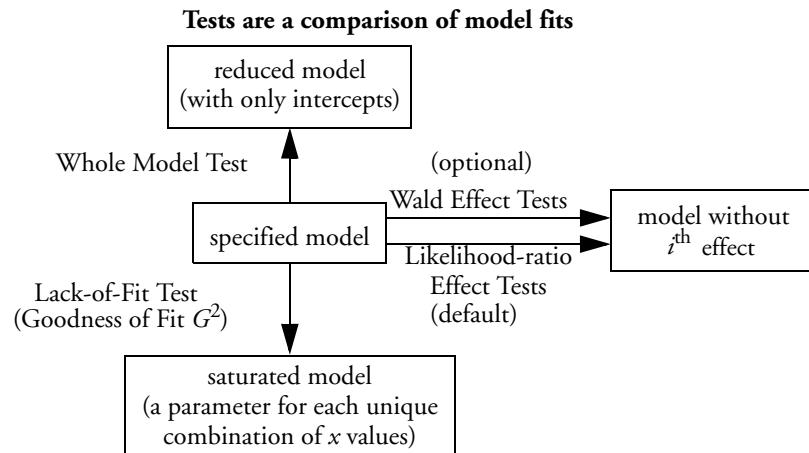
Initially the logistic platform produces the reports shown here. Lack of fit tests show only if they are applicable and Likelihood-ratio tests are done by default; Wald Tests may be requested.



All tests compare the fit of the specified model with subset or superset models, as illustrated in Figure 9.1. If a test shows significance, then the higher order model is justified.

- Whole model tests: if the specified model is significantly better than a reduced model without any effects except the intercepts.
- Lack of Fit tests: if a saturated model is significantly better than the specified model.
- Effect tests: if the specified model is significantly better than a model without a given effect.

**Figure 9.1** Relationship of Statistical Tables



## Logistic Plot

If your model contains a single continuous effect, then a logistic report similar to the one in Fit Y By X appears. See the *Basic Analysis and Graphing* book for an interpretation of these plots.

## Iteration History

After launching Fit Model, an iterative estimation process begins and is reported iteration by iteration. After the fitting process completes, you can open the Iteration History report and see the iteration steps. If the fitting takes too long, you can cancel by pressing the Escape key (⌘-Period on the Macintosh) at any time. Otherwise the iteration process stops when either the log-likelihood doesn't change by more than a very small relative amount (Obj-Criterion), the parameter estimates don't change by a small relative amount (Delta-Criterion), or 15 iterations have been performed.

Iterations			
Iter	Objective	Relative	
		Gradient	Norm Gradient
0	268.24795888	18.501466591	3500.4990148
1	76.294807072	6.140708087	727.80273343
2	53.380329197	2.9100640839	197.03411238
3	48.346085528	1.0892544501	51.063941722
4	47.69181265	0.1931790967	8.9032263493
5	47.67282965	0.0067825247	0.3218631712
6	47.67280663	8.5834561e-6	0.0004112622
7	47.67280663	1.383811e-11	6.618796e-10

## Whole Model Test

The Whole Model table shows tests that compare the whole-model fit to the model that omits all the regressor effects except the intercept parameters. The test is analogous to the Analysis of Variance table for continuous responses. The negative log-likelihood corresponds to the sums of squares, and the Chi-square test corresponds to the *F*-test.

Nominal Logistic Fit for ready

Converged in Gradient, 7 iterations  
Freq: count

**Iterations**

**Whole Model Test**

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	5.821410	2	11.64282	0.0030*
Full	47.672807			
Reduced	53.494217			

RSquare (U) 0.1088  
AICc 101.408  
BIC 113.221  
Observations (or Sum Wgts) 387

Measure	Training	Definition
Entropy RSquare	0.1088	$1 - \text{Loglike}(\text{model})/\text{Loglike}(0)$
Generalized R-Square	0.1227	$(1 - (L(0)/L(\text{model}))^{(2/n)})/(1 - L(0)^{(2/n)})$
Mean -Log p	0.1232	$\sum -\text{Log}(p_{ij})/n$
RMSE	0.1697	$\sqrt{\sum (y_{ij} - p_{ij})^2/n}$
Mean Abs Dev	0.0572	$\sum  y_{ij} - p_{ij} /n$
Misclassification Rate	0.0310	$\sum (p_{ij} \neq p_{\text{Max}})/n$
N	387	n

The Whole Model table shows these quantities:

**Model** lists the model labels called Difference (difference between the Full model and the Reduced model), Full (model that includes the intercepts and all effects), and Reduced (the model that includes only the intercepts).

**-LogLikelihood** records an associated negative log-likelihood for each of the models.

**Difference** is the difference between the Reduced and Full models. It measures the significance of the regressors as a whole to the fit.

**Full** describes the negative log-likelihood for the complete model.

**Reduced** describes the negative log-likelihood that results from a model with only intercept parameters. For the ingot experiment, the -LogLikelihood for the reduced model that includes only the intercepts is 53.49.

**DF** records an associated degrees of freedom (DF) for the Difference between the Full and Reduced model. For the ingots experiment, there are two parameters that represent different heating and soaking times, so there are 2 degrees of freedom.

**Chi-Square** is the Likelihood-ratio Chi-square test for the hypothesis that all regression parameters are zero. It is computed by taking twice the difference in negative log-likelihoods between the fitted model and the reduced model that has only intercepts.

**Prob>ChiSq** is the probability of obtaining a greater Chi-square value by chance alone if the specified model fits no better than the model that includes only intercepts.

**RSquare (U)** shows the  $R^2$ , which is the ratio of the Difference to the Reduced negative log-likelihood values. It is sometimes referred to as  $U$ , the uncertainty coefficient. **RSquare** ranges from zero for no improvement to 1 for a perfect fit. A **Nominal** model rarely has a high **Rsquare**, and it has an **Rsquare** of 1 only when all the probabilities of the events that occur are 1.

**AICc** is the corrected Akaike Information Criterion.

**BIC** is the Bayesian Information Criterion

**Observations** (or **Sum Wgts**) is the total number of observations in the sample.

**Measure** gives several measures of fit to assess model accuracy.

**Entropy RSquare** is the same as R-Square (U) explained above.

**Generalized RSquare** is a generalization of the Rsquare measure that simplifies to the regular Rsquare for continuous normal responses. It is similar to the Entropy RSquare, but instead of using the log-likelihood, it uses the  $\sqrt{2/n}$  root of the likelihood.

**Mean -Log p** is the average of  $-\log(p)$ , where  $p$  is the fitted probability associated with the event that occurred.

**RMSE** is the root mean square error, where the differences are between the response and  $p$  (the fitted probability for the event that actually occurred).

**Mean Abs Dev** is the average of the absolute values of the differences between the response and  $p$  (the fitted probability for the event that actually occurred).

**Misclassification Rate** is the rate for which the response category with the highest fitted probability is not the observed category.

For Entropy RSquare and Generalized RSquare, values closer to 1 indicate a better fit. For Mean -Log p, RMSE, Mean Abs Dev, and Misclassification Rate, smaller values indicate a better fit.

After fitting the full model with two regressors in the ingots example, the **-LogLikelihood** on the Difference line shows a reduction to 5.82 from the Reduced **-LogLikelihood** of 53.49. The ratio of Difference to Reduced (the proportion of the uncertainty attributed to the fit) is 10.9% and is reported as the **Rsquare (U)**.

To test that the regressors as a whole are significant (the Whole Model test), a Chi-square statistic is computed by taking twice the difference in negative log-likelihoods between the fitted model and the reduced model that has only intercepts. In the ingots example, this Chi-square value is  $2 \times 5.82 = 11.64$ , and is significant at 0.003.

## Lack of Fit Test (Goodness of Fit)

The next questions that JMP addresses are whether there is enough information using the variables in the current model or whether more complex terms need to be added. The Lack of Fit test, sometimes called a Goodness of Fit test, provides this information. It calculates a pure-error negative log-likelihood by constructing categories for every combination of the regressor values in the data (Saturated line in the Lack Of Fit table), and it tests whether this log-likelihood is significantly better than the Fitted model.

The screenshot shows a SAS output window for a nominal logistic regression. At the top, it says "Converged in Gradient, 7 iterations" and "Freq: count". Below this, there are three expandable sections: "Iterations", "Whole Model Test", and "Lack Of Fit". The "Lack Of Fit" section contains a table:

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	16	6.876314	13.75263
Saturated	18	40.796493	Prob>ChiSq
Fitted	2	47.672807	0.6171

The Saturated degrees of freedom is  $m-1$ , where  $m$  is the number of unique populations. The Fitted degrees of freedom is the number of parameters not including the intercept. For the Ingots example, these are 18 and 2 DF, respectively. The Lack of Fit DF is the difference between the Saturated and Fitted models, in this case  $18-2=16$ .

The Lack of Fit table lists the negative log-likelihood for error due to Lack of Fit, error in a Saturated model (pure error), and the total error in the Fitted model. Chi-square statistics test for lack of fit.

In this example, the lack of fit Chi-square is not significant ( $\text{Prob}>\text{ChiSq} = 0.617$ ) and supports the conclusion that there is little to be gained by introducing additional variables, such as using polynomials or crossed terms.

## Parameter Estimates

The Parameter Estimates report gives the parameter estimates, standard errors, and associated hypothesis test. The Covariance of Estimates report gives the variances and covariances of the parameter estimates.

Converged in Gradient, 7 iterations  
Freq: count

- ▶ Iterations
- ▶ Whole Model Test
- ▶ Lack Of Fit
- ◀ Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	5.55916646	1.1196947	24.65	<.0001*
heat	-0.0820308	0.0237345	11.95	0.0005*
soak	-0.0567713	0.3312131	0.03	0.8639

For log odds of 0/1

- ◀ Covariance of Estimates

Cov	Intercept	heat	soak
Intercept	1.2537	-0.022	-0.282
heat	-0.022	0.0006	0.0026
soak	-0.282	0.0026	0.1097

## Likelihood-ratio Tests

The **Likelihood Ratio Tests** command produces a table like the one shown here. The Likelihood-ratio Chi-square tests are calculated as twice the difference of the log-likelihoods between the full model and the model constrained by the hypothesis to be tested (the model without the effect). These tests can take time to do because each test requires a separate set of iterations.

This is the default test if the fit took less than ten seconds to complete.

Converged in Gradient, 7 iterations  
Freq: count

- ▶ Iterations
- ▶ Whole Model Test
- ▶ Lack Of Fit
- ◀ Parameter Estimates
- ◀ Effect Likelihood Ratio Tests

Source	Nparm	DF	L-R	
			ChiSquare	Prob>ChiSq
heat	1	1	11.0498622	0.0009*
soak	1	1	0.02894484	0.8649

---

## Platform Options

The red triangle menu next to the analysis name gives you the additional options that are described next.

### Plot Options

These options are described in the *Basic Analysis and Graphing* book.

### Likelihood Ratio Tests

See “[Likelihood-ratio Tests](#),” p. 174.

### Wald Tests for Effects

Effect Wald Tests				
Source	Nparm	DF	Wald	
			ChiSquare	Prob>ChiSq
heat	1	1	11.945226	0.0005*
soak	1	1	0.02937939	0.8639

---

One downside to likelihood ratio tests is that they involve refitting the whole model, which uses another series of iterations. Therefore, they could take a long time for big problems. The logistic fitting platform gives an optional test, which is more straightforward, serving the same function. The Wald Chi-square is a one-step linear approximation to the likelihood-ratio test, and it is a by-product of the calculations. Though Wald tests are considered less trustworthy, they do provide an adequate significance indicator for screening effects. Each parameter estimate and effect is shown with a Wald test. This is the default test if the fit takes more than ten seconds to complete.

Likelihood-ratio tests are the platform default and are discussed under “[Likelihood-ratio Tests](#),” p. 174. We highly recommend using this default option.

### Confidence Intervals

You can also request profile likelihood confidence intervals for the model parameters. When you select the **Confidence Intervals** command, a dialog prompts you to enter  $\alpha$  to compute the  $1 - \alpha$  confidence intervals, or you can use the default of  $\alpha = 0.05$ . Each confidence limit requires a set of iterations in the model fit and can be expensive. Furthermore, the effort does not always succeed in finding limits.

Parameter Estimates							
Term	Estimate	Std Error	ChiSquare	Prob>ChiSq	Lower 95%	Upper 95%	
Intercept	5.55916646	1.1196947	24.65	<.0001*	3.45325628	7.90106513	
heat	-0.0820308	0.0237345	11.95	0.0005*	-0.1292562	-0.0348569	
soak	-0.0567713	0.3312131	0.03	0.8639	-0.6674294	0.66288565	

For log odds of 0/1

## Odds Ratios (Nominal Responses Only)

When you select **Odds Ratios**, a report appears showing **Unit Odds Ratios** and **Range Odds Ratios**, as shown in Figure 9.2.

Figure 9.2 Odds Ratios

Odds Ratios					
For ready odds of 0 versus 1					
Unit Odds Ratios					
Per unit change in regressor					
Term	Odds Ratio	Lower 95%	Upper 95%	Reciprocal	
heat	0.921244	0.878749	0.965744	1.0854892	
soak	0.94481	0.513026	1.940384	1.0584137	
Range Odds Ratios					
Per change in regressor over entire range					
Term	Odds Ratio	Lower 95%	Upper 95%	Reciprocal	
heat	0.027069	0.003389	0.215736	36.942229	
soak	0.8434	0.135026	7.305715	1.185677	

From the introduction (for two response levels), we had

$$\log\left(\frac{\text{Prob}(Y = r_1)}{\text{Prob}(Y = r_2)}\right) = Xb \text{ where } r_1 \text{ and } r_2 \text{ are the two response levels}$$

so the odds ratio

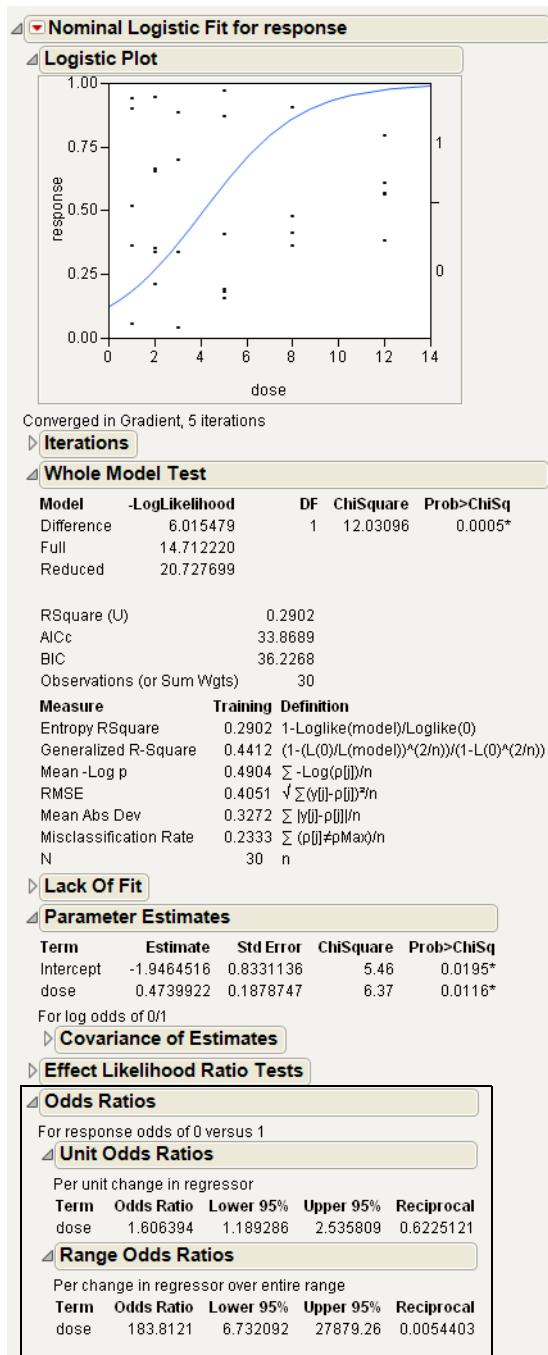
$$\frac{\text{Prob}(Y = r_1)}{\text{Prob}(Y = r_2)} = \exp(Xb) = \exp(\beta_0) \cdot \exp(\beta_1 X_1) \cdots \exp(\beta_i X_i)$$

Note that  $\exp(\beta_i(X_i + 1)) = \exp(\beta_i X_i) \exp(\beta_i)$ . This shows that if  $X_i$  changes by a unit amount, the odds is multiplied by  $\exp(\beta_i)$ , which we label the unit odds ratio. As  $X_i$  changes over its whole range, the odds is multiplied by  $\exp((X_{\text{high}} - X_{\text{low}})\beta_i)$  which we label the range odds ratio. For binary responses, the log odds ratio for flipped response levels involves only changing the sign of the parameter, so you may want the reciprocal of the reported value to focus on the last response level instead of the first.

Two-level nominal effects are coded 1 and -1 for the first and second levels, so range odds ratios or their reciprocals would be of interest.

### **Dose Response Example**

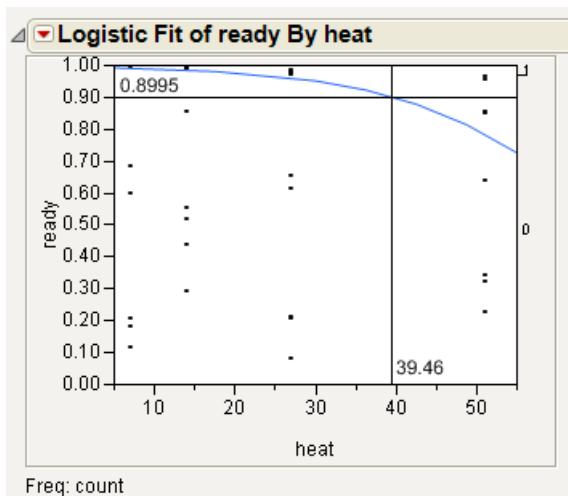
In the Dose Response.jmp sample data table, the dose varies between 1 and 12. Using Fit Model, we got the following report after requesting **Odds Ratio** from the platform drop-down menu.



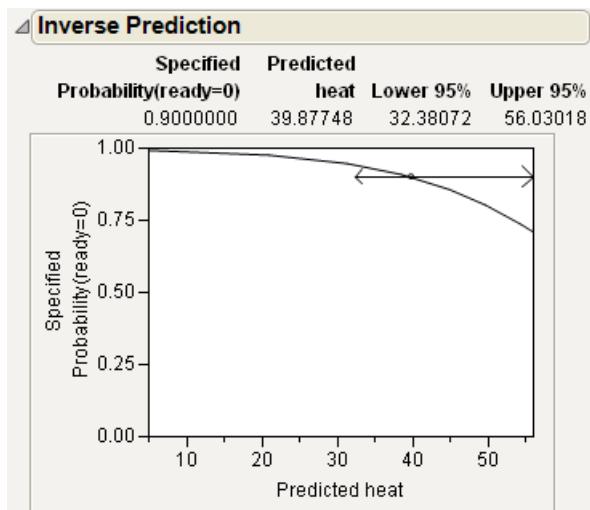
The unit odds ratio for dose is 1.606 (which is  $\exp(0.474)$ ) and indicates that the odds of getting a  $Y = 0$  rather than  $Y = 1$  improves by a factor of 1.606 for each increase of one unit of dose. The range odds ratio for dose is 183.8 ( $\exp((12-1)*0.474)$ ) and indicates that the odds improve by a factor of 183.8 as dose is varied between 1 and 12.

## Inverse Prediction

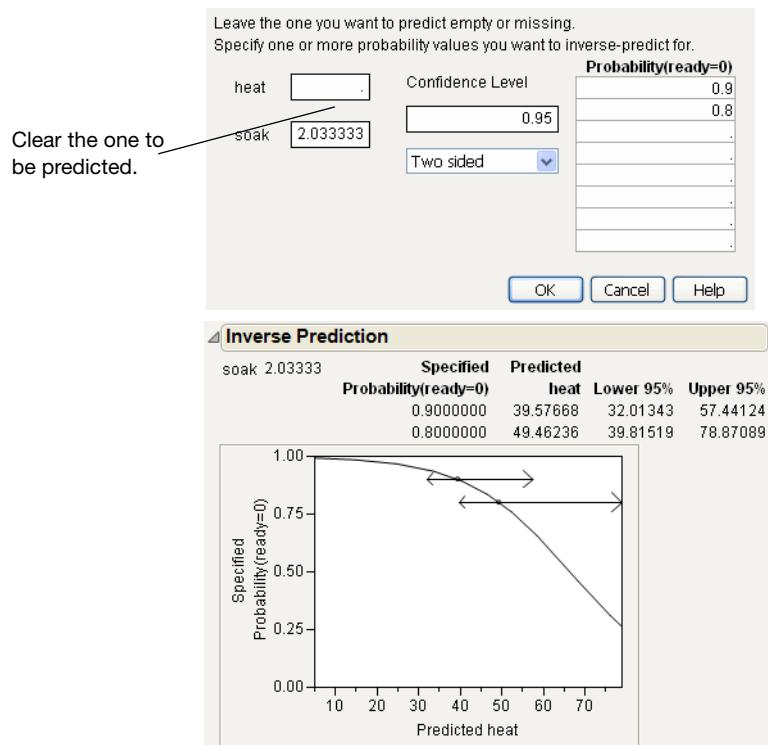
For a two-level response, the **Inverse Prediction** command finds the  $x$  value that results in a specified probability. For example, using the *Ingots.jmp* data, ignore the *Soak* variable, and fit the probability of *ready* by *heat* using the **Fit Y by X** command for a simple logistic regression (Count is **Freq**). The cumulative logistic probability plot shows the result.



Note that the fitted curve crosses the 0.9 probability level at a heat level of about 39.5, which is the inverse prediction. To be more precise and to get a fiducial confidence interval, you can use the **Inverse Prediction** command for **Fit Y by X** (see the *Basic Analysis and Graphing* book for more discussion of the Fit Y by X simple logistic analysis). When you specify exactly 0.9 in the Inverse Prediction dialog, the predicted value (inverse prediction) for heat is 39.8775, as shown in the table below.



However, if you have another regressor variable (Soak), you must use the Fit Model platform. Then the **Inverse Prediction** command displays the Inverse Prediction dialog shown in Figure 9.3, for requesting the probability of obtaining a given value for one independent variable. To complete the dialog, click and type values in the editable X and Probability columns. Enter a value for a single X (heat or soak) and the probabilities you want for the prediction. Set the remaining independent variable to missing by clicking in its X field and deleting. The missing regressor is the value that it will predict.

**Figure 9.3** The Inverse Prediction Dialog and Table

See the appendix “[Statistical Details](#),” p. 607, for more details about inverse prediction.

## Save Commands

If you have ordinal or nominal response models, the **Save Probability Formula** command creates new data table columns.

If the response is numeric and has the ordinal modeling type, the **Save Quantiles** and **Save Expected Values** commands are also available.

The **Save** commands create the following new columns:

**Save Probability Formula** creates columns in the current data table that save formulas for linear combinations of the response levels, prediction formulas for the response levels, and a prediction formula giving the most likely response.

For a nominal response model with  $r$  levels, JMP creates

- columns called  $\text{Lin}[j]$  that contain a linear combination of the regressors for response levels  $j = 1, 2, \dots, r - 1$
- a column called  $\text{Prob}[r]$ , with a formula for the fit to the last level,  $r$

## Platform Options

- columns called `Prob[j]` for  $j < r$  with a formula for the fit to level  $j$
- a column called `Most Likely response` that picks the most likely level of each row based on the computed probabilities.

For an ordinal response model with  $r$  levels, JMP creates

- a column called `Linear` that contains the formula for a linear combination of the regressors without an intercept term
- columns called `Cum[j]`, each with a formula for the cumulative probability that the response is less than or equal to level  $j$ , for levels  $j = 1, 2, \dots, r - 1$ . There is no `Cum[j = 1, 2, \dots, r - 1]` that is 1 for all rows
- columns called `Prob[j = 1, 2, \dots, r - 1]`, for  $1 < j < r$ , each with the formula for the probability that the response is level  $j$ . `Prob[j]` is the difference between `Cum[j]` and `Cum[j-1]`. `Prob[1]` is `Cum[1]`, and `Prob[r]` is  $1 - \text{Cum}[r-1]$ .
- a column called `Most Likely response` that picks the most likely level of each row based on the computed probabilities.

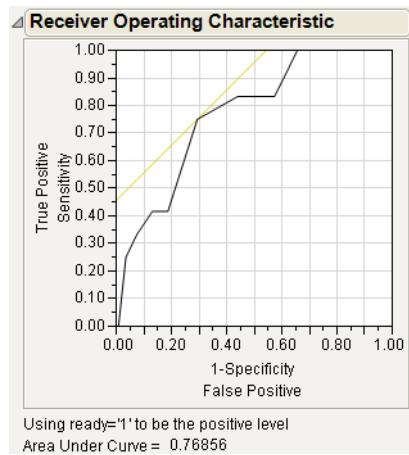
**Save Quantiles** creates columns in the current data table named `OrdQ.05`, `OrdQ.50`, and `OrdQ.95` that fit the quantiles for these three probabilities.

**Save Expected Value** creates a column in the current data table called `Ord Expected` that is the linear combination of the response values with the fitted response probabilities for each row and gives the expected value.

## ROC Curve

Receiver Operating Characteristic (ROC) curves measure the sorting efficiency of the model's fitted probabilities to sort the response levels. ROC curves can also aid in setting criterion points in diagnostic tests. The higher the curve from the diagonal, the better the fit. An introduction to ROC curves is found in the *Basic Analysis and Graphing* book. If the logistic fit has more than two response levels, it produces a generalized ROC curve (identical to the one in the Partition platform). In such a plot, there is a curve for each response level, which is the ROC curve of that level versus all other levels. Details on these ROC curves are found in “[Graphs for Goodness of Fit](#),” p. 314 in the “Recursive Partitioning” chapter.

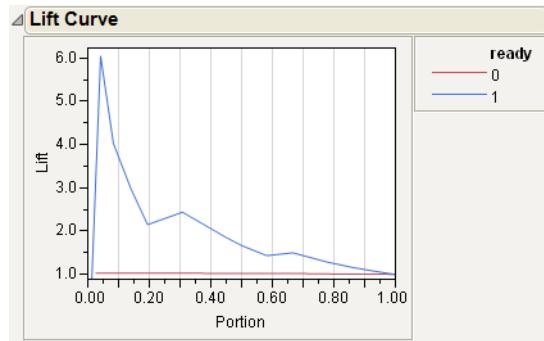
Figure 9.4 shows the ROC Curve using the `Ingots.jmp` data table. Specify both `heat` and `soak` as model effects, `count` as `Freq`, and `ready` as `Y`.

**Figure 9.4** ROC Curve

## Lift Curve

Produces a lift curve for the model. A lift curve shows the same information as an ROC curve, but in a way to dramatize the richness of the ordering at the beginning. The Y-axis shows the ratio of how rich that portion of the population is in the chosen response level compared to the rate of that response level as a whole. See “[Lift Curves](#),” p. 317 in the “Recursive Partitioning” chapter for more details on lift curves.

Figure 9.5 shows the lift curve for the same model specified for the ROC curve (Figure 9.4).

**Figure 9.5** Lift Curve

## Confusion Matrix

A confusion matrix is a two-way classification of the actual response levels and the predicted response levels. For a good model, predicted response levels should be the same as the actual response levels. The confusion matrix gives a way of assessing how the predicted responses aligns with the actual responses.

## Profiler

Brings up the prediction profiler, showing the fitted values for a specified response probability as the values of the factors in the model are changed. This feature is available for both nominal and ordinal responses. For detailed information on profiling features, refer to the “[Profiling](#)” chapter.

---

## Validation

---

**Note:** Validation is available only in JMP Pro.

---

Validation is the process of using part of a data set to estimate model parameters, and using the other part to assess the predictive ability of the model.

- The *training* set is the part that estimates model parameters.
- The *validation* set is the part that assesses or validates the predictive ability of the model.
- The *test* set is a final, independent assessment of the model’s predictive ability. The test set is available only when using a validation column.

The training, validation, and test sets are created by subsetting the original data into parts. This is done through the use of a validation column in the Fit Model launch window.

The validation column’s values determine how the data is split, and what method is used for validation:

- If the column has two distinct values, then training and validation sets are created.
- If the column has three distinct values, then training, validation, and test sets are created.
- If the column has more than three distinct values, or only one, then no validation is performed.

When validation is used, model fit statistics are given for the training, validation, and test sets.

---

## Nominal Logistic Model Example: The Detergent Data

---

A market research study was undertaken to evaluate preference for a brand of detergent (Ries and Smith 1963). The results are in the Detergent.jmp sample data table. The model is defined by

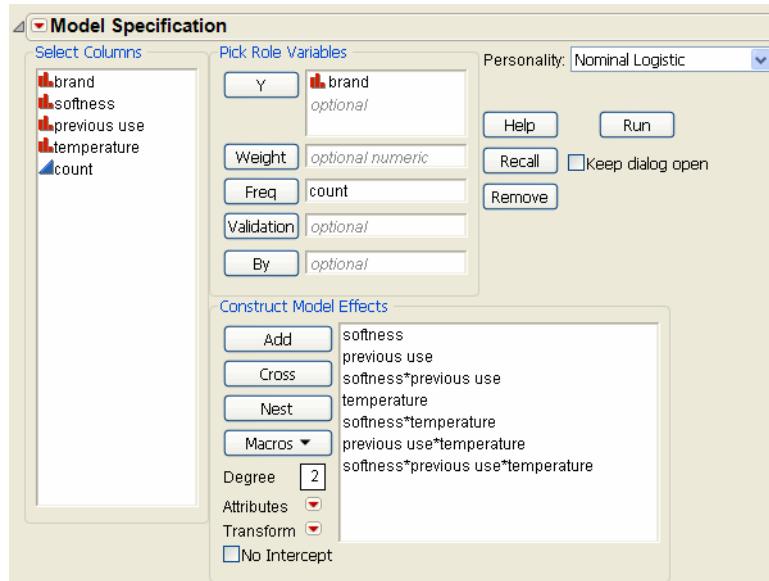
- the response variable, brand with values m and x
- an effect called **softness** (water softness) with values soft, medium, and hard
- an effect called **previous use** with values yes and no
- an effect called **temperature** with values high and low

- a count variable, count, which gives the frequency counts for each combination of effect categories.

The study begins by specifying the full three-factor factorial model as shown by the Fit Model dialog in Figure 9.6. To specify a factorial model, highlight the three main effects in the column selector list. Then select **Full Factorial** from the **Macros** popup menu.

---

**Figure 9.6** A Three-Factor Factorial Model with Nominal Response



---

**Note:** Validation is available only in JMP Pro.

The tables in Figure 9.7 show the three-factor model as a whole to be significant ( $\text{Prob}>\text{ChiSq} = 0.0006$ ) in the Whole Model table. The Effect Likelihood Ratio Tests table shows that the effects which include softness do not contribute significantly to the model fit.

**Figure 9.7** Tables for Nominal Response Three-Factor Factorial

**Nominal Logistic Fit for brand**

Converged in Gradient, 3 iterations  
Freq: count

**Iterations**

**Whole Model Test**

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	16.41281	11	32.82562	0.0006*
Full	682.24780			
Reduced	698.66061			

RSquare (U) 0.0235  
AICc 1388.81  
BIC 1447.48  
Observations (or Sum Wgts) 1008

**Measure** **Training** **Definition**

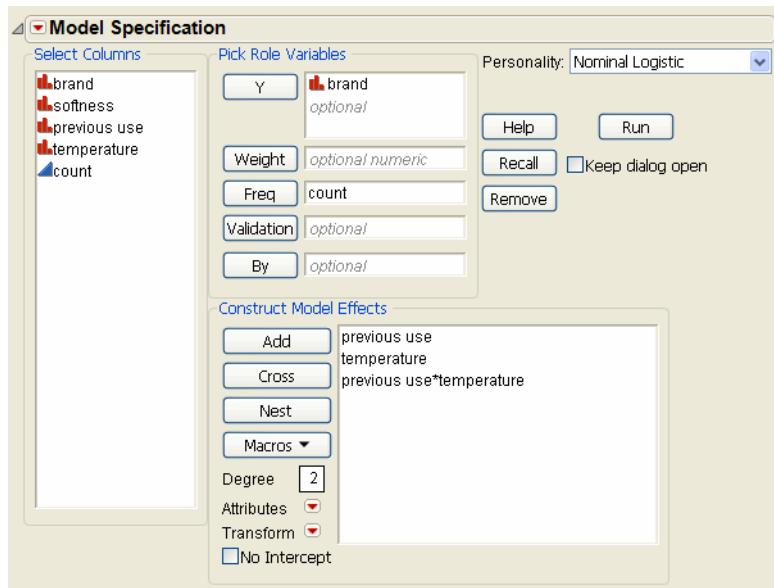
Entropy RSquare	0.0235	$1 - \text{Loglike}(\text{model})/\text{Loglike}(0)$
Generalized R-Square	0.0427	$(1 - (L(0)/L(\text{model})))^{(2/n)} / (1 - L(0))^{(2/n)}$
Mean -Log p	0.6768	$\sum -\text{Log}(p[i])/n$
RMSE	0.4919	$\sqrt{\sum (y[i] - p[i])^2/n}$
Mean Abs Dev	0.4839	$\sum  y[i] - p[i] /n$
Misclassification Rate	0.4206	$\sum (p[i] \neq p\text{Max})/n$
N	1008	n

**Parameter Estimates**

**Effect Likelihood Ratio Tests**

L-R				
Source	Nparm	DF	ChiSquare	Prob>ChiSq
softness	2	2	0.09804239	0.9522
previous use	1	1	22.1316677	<.0001*
softness*previous use	2	2	3.78609668	0.1506
temperature	1	1	3.63914017	0.0564
softness*temperature	2	2	0.19617686	0.9066
previous use*temperature	1	1	2.26089203	0.1327
softness*previous use*temperature	2	2	0.73731691	0.6917

Next, use the Fit Model Dialog again to remove the **softness** factor and its interactions because they don't appear to be significant. You can do this easily by double-clicking the softness factor in the Fit Model dialog. A dialog then appears asking if you want to remove the other factors that involve softness (click **Yes**). This leaves the two-factor factorial model in Figure 9.8.

**Figure 9.8** A Two-factor Factorial Model with Nominal Response

The Whole Model Test table shows that the two-factor model fits as well as the three-factor model. In fact, the three-factor Whole Model table in Figure 9.9 shows a larger Chi-square value (32.83) than the Chi-square value for the two-factor model (27.17). This results from the change in degrees of freedom used to compute the Chi-square values and their probabilities.

Ordinal Logistic Example: The Cheese Data

**Figure 9.9** Two-Factor Model

Converged in Gradient, 3 iterations  
Freq: count  
Iterations

**Whole Model Test**

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	13.58479	3	27.16958	<.0001*
Full	685.07582			
Reduced	698.66061			

RSquare (U) 0.0194  
AICc 1378.19  
BIC 1397.81  
Observations (or Sum Wgts) 1008

**Measure Training Definition**

Measure	Definition
Entropy RSquare	0.0194 $1 - \text{Loglike}(\text{model})/\text{Loglike}(0)$
Generalized R-Square	0.0355 $(1 - (L(0)/L(\text{model}))^{(2/n)})/(1 - L(0)^{(2/n)})$
Mean -Log p	0.6796 $\sum -\text{Log}(p[i])/n$
RMSE	0.4933 $\sqrt{\sum (y[i] - p[i])^2/n}$
Mean Abs Dev	0.4866 $\sum  y[i] - p[i] /n$
Misclassification Rate	0.4286 $\sum (p[i] \neq p\text{Max})/n$
N	1008 n

**Parameter Estimates**

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	0.03043265	0.0666827	0.21	0.6481
previous use[no]	-0.3148498	0.0666827	22.29	<.0001*
temperature[low]	-0.1331196	0.0666827	3.99	0.0459*
previous use[no]*temperature[low]	0.11106673	0.0666827	2.77	0.0958

For log odds of m/x

**Covariance of Estimates**

**Effect Likelihood Ratio Tests**

L-R				
Source	Nparm	DF	ChiSquare	Prob>ChiSq
previous use	1	1	22.6511053	<.0001*
temperature	1	1	3.99625478	0.0456*
previous use*temperature	1	1	2.78794934	0.0950

The report shown in Figure 9.9 supports the conclusion that previous use of a detergent brand, and water temperature, have an effect on detergent preference, and the interaction between temperature and previous use is not statistically significant (the effect of temperature does not depend on previous use).

## Ordinal Logistic Example: The Cheese Data

If the response variable has an ordinal modeling type, the platform fits the cumulative response probabilities to the logistic function of a linear model using maximum likelihood. Likelihood-ratio test statistics are provided for the whole model and lack of fit. Wald test statistics are provided for each effect.

If there is an ordinal response and a single continuous numeric effect, the ordinal logistic platform in Fit Y by X displays a cumulative logistic probability plot.

Details of modeling types are discussed in the *Basic Analysis and Graphing* book. The details of fitting appear in the appendix “[Statistical Details](#),” p. 607. The method is discussed in Walker and Duncan (1967), Nelson (1976), Harrell (1986), and McCullagh and Nelder (1983).

**Note:** If there are many response levels, the ordinal model is much faster to fit and uses less memory than the nominal model.

As an example of ordinal logistic model fitting, McCullagh and Nelder (1983) report an experiment by Newell to test whether various cheese additives (A to D) had an effect on taste. Taste was measured by a tasting panel and recorded on an ordinal scale from 1 (strong dislike) to 9 (excellent taste). The data are in the *Cheese.jmp* sample data table.

To run the model, assign **Response** as **Y**, **Cheese** as the effect, and **Count** as **Freq**.

The method in this example required only seven iterations to reduce the background **-LogLikelihood** of 429.9 to 355.67. This reduction yields a likelihood-ratio Chi-square for the whole model of 148.45 with 3 degrees of freedom, showing the difference in perceived cheese taste to be highly significant.

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	74.22695	3	148.4539	<.0001*
Full	355.67395			
Reduced	429.90090			

Measure	Training	Definition
Entropy RSquare	0.1727	$1 - \text{Loglike}(\text{model})/\text{Loglike}(0)$
Generalized R-Square	0.5185	$(1 - L(0)/L(\text{model}))^{(2/n)} / (1 - L(0)^{(2/n)})$
Mean -Log p	1.7100	$\sum -\text{Log}(p[i])/n$
RMSE	0.7944	$\sqrt{\sum (y[i] - p[i])^2/n}$
Mean Abs Dev	0.7886	$\sum  y[i] - p[i] /n$
Misclassification Rate	0.6635	$\sum (p[i] > p_{\text{Max}})/n$
N	208	n

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	21	10.15410	20.30819
Saturated	24	345.51986	Prob>ChiSq
Fitted	3	355.67395	0.5018

Source	Nparm	DF	ChiSquare	Prob>ChiSq
Cheese	3	3	115.152723	<.0001*

Source	Nparm	DF	ChiSquare	Prob>ChiSq
Cheese	3	3	148.453899	<.0001*

---

The Lack of Fit test happens to be testing the ordinal response model compared to the nominal model. This is because the model is saturated if the response is treated as nominal rather than ordinal, giving 21

## Ordinal Logistic Example: The Cheese Data

additional parameters, which is the Lack of Fit degrees of freedom. The nonsignificance of Lack of Fit leads one to believe that the ordinal model is reasonable.

There are eight intercept parameters because there are nine response categories. There are only three structural parameters. As a nominal problem, there are  $8 \times 3 = 24$  structural parameters.

When there is only one effect, its test is equivalent to the Likelihood-ratio test for the whole model. The Likelihood-ratio Chi-square is 148.45, different than the Wald Chi-square of 115.15, which illustrates the point that Wald tests are to be regarded with some skepticism.

**Note:** Likelihood Ratio Tests is a default table; Wald Tests must be checked as an additional option on the platform drop-down menu.

To see if a cheese additive is preferred, look for the most negative values of the parameters (Cheese D's effect is the negative sum of the others, shown in Figure 9.10.).

**Figure 9.10** Parameter Estimates and Preferences for Cheese Additives in Cheese.jmp

Cheese	Estimate	Preference
A	-0.8622	2nd place
B	2.4896	least liked
C	0.8477	3rd place
D	-2.4750	most liked

You can also use the Fit Y by X platform for this model, which treats ordinal responses like nominal and shows a contingency table analysis. See Figure 9.11. The Fit Model platform can be used, but you must set the ordinal response, Response, to nominal. Nominal Fit Model results are shown in Figure 9.12. The negative log-likelihood values (84.381) and the likelihood chi-square values (168.76) are the same.

**Figure 9.11** Fit Y by X Platform Results for Cheese.jmp

N	DF	-LogLike	RSquare (U)
208	24	84.381046	0.1963

Test	ChiSquare	Prob>ChiSq
Likelihood Ratio	168.762	<.0001*
Pearson	162.482	<.0001*

Warning: 20% of cells have expected count less than 5, ChiSquare suspect.

**Figure 9.12** Fit Model Platform Results Setting Response to Nominal for Cheese.jmp

Converged in Gradient, 18 iterations  
Freq: Count

**Iterations**

**Whole Model Test**

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	84.38105	24	168.7621	<.0001*
Full	345.51986			
Reduced	429.90090			

RSquare (U) 0.1963  
AICc 767.108  
BIC 861.841  
Observations (or Sum Wgts) 208

**Measure Training Definition**

Entropy RSquare	0.1963	1-Loglike(model)/Loglike(0)
Generalized R-Square	0.5648	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	1.6612	$\sum -\text{Log}(p[i])/n$
RMSE	0.7855	$\sqrt{\sum (y[i]-p[i])^2/n}$
Mean Abs Dev	0.7772	$\sum  y[i]-p[i] /n$
Misclassification Rate	0.6635	$\sum (p[i]\neq pMax)/n$
N	208	n

**Parameter Estimates**

**Effect Likelihood Ratio Tests**

L-R				
Source	Nparm	DF	ChiSquare	Prob>ChiSq
Cheese	24	24	168.762091	<.0001*

If you want to see a graph of the response probabilities as a function of the parameter estimates for the four cheeses, you can create a new continuous variable (call it **Score**) with the formula from the Formula Editor shown in Figure 9.13.

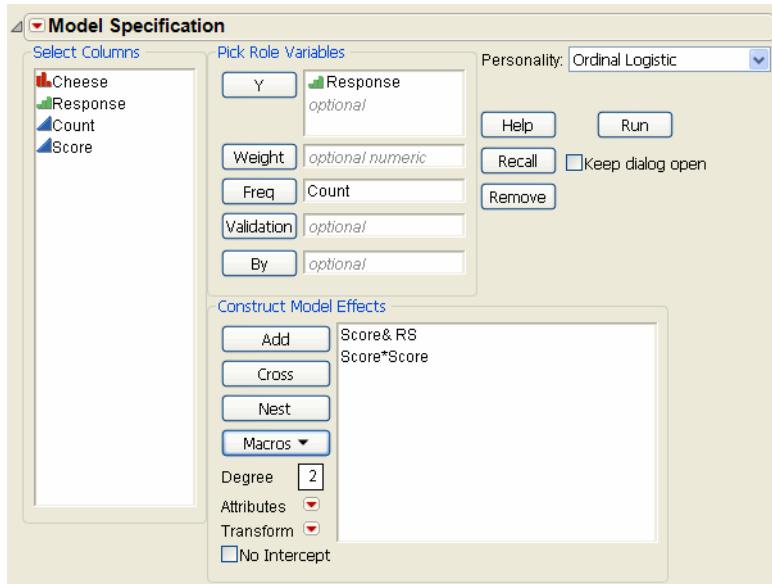
**Figure 9.13** Formula for Score Column

MatchMZ[Cheese]	"A" => -0.8622 "B" => 2.4895 "C" => 0.8477 "D" => -2.475 else =>
-----------------	--

## Ordinal Logistic Example: The Cheese Data

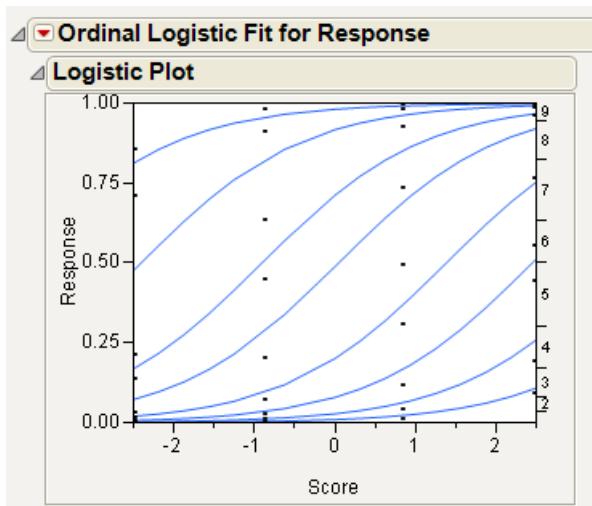
Use the new variable as a response surface effect in the Fit Model dialog as shown. To create the model in Figure 9.14, select Score in the column selector list, and then select **Response Surface** from the Macros popup menu on the Fit Model dialog.

**Figure 9.14** Model Dialog For Ordinal Logistic Regression



**Note:** Validation is available only in JMP Pro.

Click **Run** to see the analysis report and the cumulative logistic probability plot in Figure 9.15. The distance between each curve is the fitted response probability for the levels in the order for the levels on the right axis of the plot.

**Figure 9.15** Cumulative Probability Plot for Ordinal Logistic Regression

## Quadratic Ordinal Logistic Example: Salt in Popcorn Data

The Ordinal Response Model can fit a quadratic surface to optimize the probabilities of the higher or lower responses. The arithmetic in terms of the structural parameters is the same as that for continuous responses. Up to five factors can be used, but this example has only one factor, for which there is a probability plot.

Consider the case of a microwave popcorn manufacturer who wants to find out how much salt consumers like in their popcorn. To do this, the manufacturer looks for the maximum probability of a favorable response as a function of how much salt is added to the popcorn package. An experiment controls salt amount at 0, 1, 2, and 3 teaspoons, and the respondents rate the taste on a scale of 1=low to 5=high. The optimum amount of salt is the amount that maximizes the probability of more favorable responses. The ten observations for each of the salt levels are shown in Table 9.1.

**Table 9.1** Salt in Popcorn

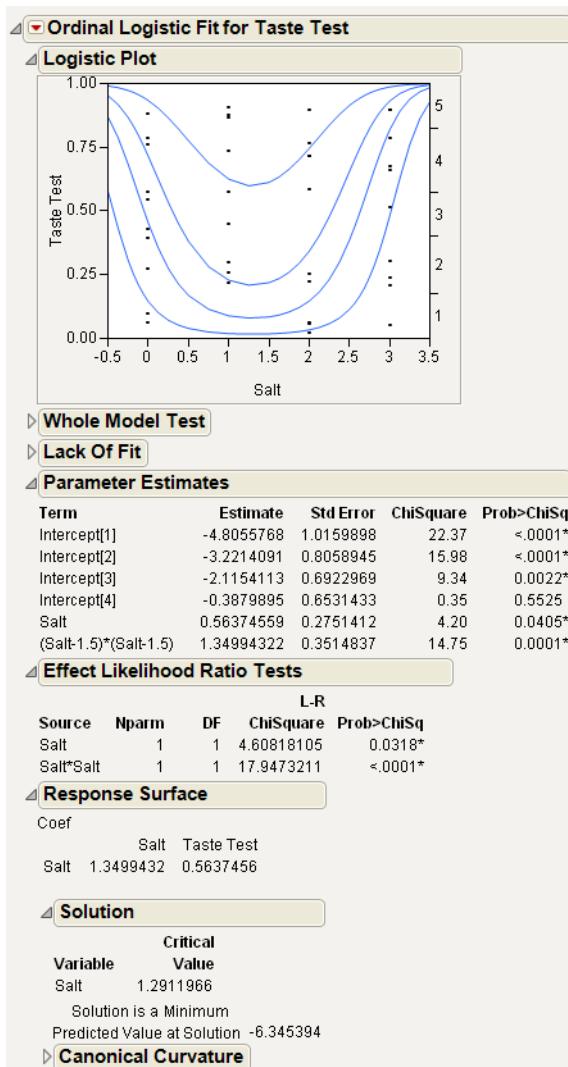
Salt Amount	Salt Rating Response									
no salt	1	3	2	4	2	2	1	4	3	4
1 tsp.	4	5	3	4	5	4	5	5	4	5
2 tsp.	4	3	5	1	4	2	5	4	3	2
3 tsp.	3	1	2	3	1	2	1	2	1	2

## Quadratic Ordinal Logistic Example: Salt in Popcorn Data

Use **Fit Model** with the Salt in Popcorn.jmp sample data to fit the ordinal taste test to the surface effect of salt. Use Taste Test as Y. Highlight Salt in the Select Columns box, and then select **Macros > Response Surface**.

The report shows how the quadratic model fits the response probabilities. The curves, instead of being shifted logistic curves, become a folded pile of curves where each curve achieves its optimum at the same point. The critical value is at  $\text{Mean}(X) - 0.5 * b1/b2$  where  $b1$  is the linear coefficient and  $b2$  is the quadratic coefficient. This formula is for centered  $X$ . From the Parameter Estimates table you can compute the optimum as  $1.5 - 0.5 * (0.5637/1.3499) = 1.29$  teaspoons of salt.

**Figure 9.16** Ordinal Logistic Fit for Salt in Popcorn.jmp



The distance between each curve measures the probability of each of the five response levels. The probability for the highest response level is the distance from the top curve to the top of the plot rectangle. This distance reaches a maximum when the amount of salt is about 1.3 teaspoons. All curves share the same critical point.

The parameter estimates for Salt and Salt\*Salt become the coefficients used to find the critical value. Although it appears as a minimum, it is only a minimum with respect to the probability curves. It is really a maximum in the sense of maximizing the probability of the highest response. The Solution portion of the report is shown under Response Surface in Figure 9.16, where 1.29 is shown under Critical Value.

---

## What to Do If Your Data Are Counts in Multiple Columns

Data that are frequencies (counts) listed in several columns of your data table are not the form you need for logistic regression. For example, the Ingots2.jmp data table in the data folder (see Figure 9.17) has columns Nready and Nnotready that give the number of ready and number of not ready ingots for each combination of Heat and Soak values. To do a logistic regression, you need the data organized like the table in Figure 9.18.

To make a new table, suitable for logistic regression, select the **Stack** command from the **Tables** menu. Complete the Stack dialog by choosing Nready and NNotReady as the columns to stack, and then click **OK** in the Stack dialog. This creates the new table in Figure 9.18. If you use the default column names, Label is the response (*Y*) column and Data is the frequency column.

The example in the section “[Introduction to Logistic Models](#),” p. 167, shows a logistic regression using a sample data table Ingots.jmp. It has a frequency column called count (equivalent to the Data column in the table below) and a response variable called Ready, with values 1 to represent ingots that are ready and 0 for not ready.

---

**Figure 9.17** Original Data Table

	Heat	Soak	Nnotready	Nready	Ntotal	P	Loss
1	7	1	10	0	10	0.5000	6.9315
2	7	1.7	17	0	17	0.5000	11.7835
3	7	2.2	7	0	7	0.5000	4.8520
4	7	2.8	12	0	12	0.5000	8.3178
5	7	4	9	0	9	0.5000	6.2383
6	14	1	31	0	31	0.5000	21.4876
7	14	1.7	43	0	43	0.5000	29.8053
8	14	2.2	31	2	33	0.5000	22.8739
9	14	2.8	31	0	31	0.5000	21.4876
10	14	4	19	0	19	0.5000	13.1698
11	27	1	55	1	56	0.5000	38.8162
12	27	1.7	40	4	44	0.5000	30.4985
13	27	2.2	21	0	21	0.5000	14.5561
14	27	2.8	21	1	22	0.5000	15.2492

---

**Figure 9.18** Stacked Data Table

	Heat	Soak	Ntotal	P	Loss	Label	Data
1	7	1	10	0.5000	6.9315	Nnotready	10
2	7	1	10	0.5000	6.9315	Nready	0
3	7	1.7	17	0.5000	11.7835	Nnotready	17
4	7	1.7	17	0.5000	11.7835	Nready	0
5	7	2.2	7	0.5000	4.8520	Nnotready	7
6	7	2.2	7	0.5000	4.8520	Nready	0
7	7	2.8	12	0.5000	8.3178	Nnotready	12
8	7	2.8	12	0.5000	8.3178	Nready	0
9	7	4	9	0.5000	6.2383	Nnotready	9
10	7	4	9	0.5000	6.2383	Nready	0
11	14	1	31	0.5000	21.4876	Nnotready	31
12	14	1	31	0.5000	21.4876	Nready	0
13	14	1.7	43	0.5000	29.8053	Nnotready	43
14	14	1.7	43	0.5000	29.8053	Nready	0

# Chapter 10

## Generalized Linear Models

### The Fit Model Platform

---

Generalized Linear Models provide a unified way to fit responses that don't fit the usual requirements of least-squares fits. In particular, frequency counts, which are characterized as having a Poisson distribution indexed by a model, are easily fit by a Generalized Linear Model.

The technique, pioneered by Nelder and Wedderburn (1972), involves a set of iteratively reweighted least-squares fits of a transformed response.

Additional features of JMP's Generalized Linear Model personality are

- likelihood ratio statistics for user-defined contrasts, that is, linear functions of the parameters, and  $p$ -values based on their asymptotic chi-square distributions
- estimated values, standard errors, and confidence limits for user-defined contrasts and least-squares means
- graphical profilers for examining the model
- confidence intervals for model parameters based on the profile likelihood function
- optional bias-corrected maximum likelihood estimator discussed by Firth (1993).

# Contents

Generalized Linear Models .....	199
The Generalized Linear Model Personality .....	199
Examples of Generalized Linear Models .....	200
Model Selection and Deviance. ....	202
Examples.....	203
Poisson Regression.....	203
Poisson Regression with Offset .....	206
Normal Regression, Log Link .....	208
Platform Commands.....	212

---

## Generalized Linear Models

While traditional linear models are used extensively in statistical data analysis, there are types of problems for which they are not appropriate.

- It may not be reasonable to assume that data are normally distributed. For example, the normal distribution (which is continuous) may not be adequate for modeling counts or measured proportions.
- If the mean of the data is naturally restricted to a range of values, the traditional linear model may not be appropriate, since the linear predictor can take on any value. For example, the mean of a measured proportion is between 0 and 1, but the linear predictor of the mean in a traditional linear model is not restricted to this range.
- It may not be realistic to assume that the variance of the data is constant for all observations. For example, it is not unusual to observe data where the variance increases with the mean of the data.

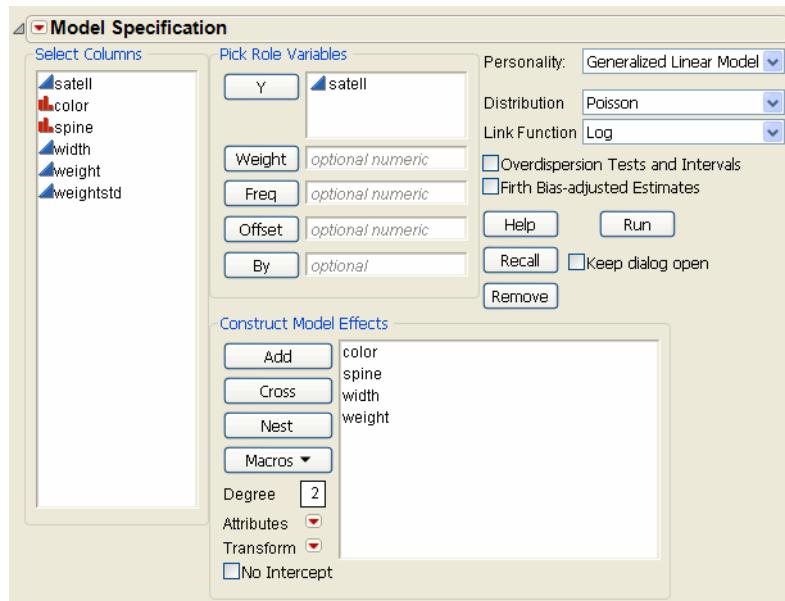
A generalized linear model extends the traditional linear model and is, therefore, applicable to a wider range of data analysis problems. See the section “[Examples of Generalized Linear Models](#),” p. 200 for the form of a probability distribution from the exponential family of distributions.

As in the case of traditional linear models, fitted generalized linear models can be summarized through statistics such as parameter estimates, their standard errors, and goodness-of-fit statistics. You can also make statistical inference about the parameters using confidence intervals and hypothesis tests. However, specific inference procedures are usually based on asymptotic considerations, since exact distribution theory is not available or is not practical for all generalized linear models.

---

## The Generalized Linear Model Personality

Generalized linear models are fit as a personality of the Fit Model Dialog. After selecting **Analyze > Fit Model**, select **Generalized Linear Model** from the drop-down menu before or after assigning the effects to the model.

**Figure 10.1** Generalized Linear Model Launch Dialog

When you specify that you are fitting a generalized linear model, the Fit Model dialog changes to allow you to select a Distribution and a Link Function. In addition, an Offset button, an option for overdispersion tests and intervals, and an option for Firth Bias-adjusted Estimates appears.

## Examples of Generalized Linear Models

You construct a generalized linear model by deciding on response and explanatory variables for your data and choosing an appropriate link function and response probability distribution. Explanatory variables can be any combination of continuous variables, classification variables, and interactions.

**Table 10.1** Examples of Generalized Linear Models

Model	Response Variable	Distribution	Canonical Link Function
Traditional Linear Model	continuous	Normal	identity, $g(\mu) = \mu$
Logistic Regression	a count or a binary random variable	Binomial	logit, $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$

**Table 10.1** Examples of Generalized Linear Models (*Continued*)

Model	Response Variable	Distribution	Canonical Link Function
Poisson Regression in Log Linear Model	a count	Poisson	$\log, g(\mu) = \log(\mu)$
Exponential Regression	positive continuous	Exponential	$\frac{1}{\mu}$

JMP fits a generalized linear model to the data by maximum likelihood estimation of the parameter vector. There is, in general, no closed form solution for the maximum likelihood estimates of the parameters. JMP estimates the parameters of the model numerically through an iterative fitting process. The dispersion parameter  $\phi$  is also estimated by dividing the Pearson goodness-of-fit statistic by its degrees of freedom. Covariances, standard errors, and confidence limits are computed for the estimated parameters based on the asymptotic normality of maximum likelihood estimators.

A number of link functions and probability distributions are available in JMP. The built-in link functions are

$$\text{identity: } g(\mu) = \mu$$

$$\text{logit: } g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

$$\text{probit: } g(\mu) = \Phi^{-1}(\mu), \text{ where } \Phi \text{ is the standard normal cumulative distribution function}$$

$$\text{log: } g(\mu) = \log(\mu)$$

$$\text{reciprocal: } g(\mu) = \frac{1}{\mu}$$

$$\text{power: } g(\mu) = \begin{cases} \mu^\lambda & \text{if } (\lambda \neq 0) \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$$

$$\text{complementary log-log: } g(m) = \log(-\log(1-\mu))$$

When you select the Power link function, a number box appears enabling you to enter the desired power.

The available distributions and associated variance functions are

$$\text{normal: } V(\mu) = 1$$

$$\text{binomial (proportion): } V(\mu) = \mu(1-\mu)$$

$$\text{Poisson: } V(\mu) = \mu$$

$$\text{Exponential: } V(\mu) = \mu^2$$

When you select **Binomial** as the distribution, the response variable must be specified in one of the following ways:

- If your data is not summarized as frequencies of events, specify a single binary column as the response. The response column must be nominal. If your data is summarized as frequencies of events, specify a

single binary column as the response, along with a frequency variable in the Freq role. The response column must be nominal, and the frequency variable gives the count of each response level.

- If your data is summarized as frequencies of events and trials, specify two continuous columns in this order: a count of the number of successes, and a count of the number of trials. Alternatively, you can specify the number of failures instead of successes.

## Model Selection and Deviance

An important aspect of generalized linear modeling is the selection of explanatory variables in the model. Changes in goodness-of-fit statistics are often used to evaluate the contribution of subsets of explanatory variables to a particular model. The *deviance*, defined to be twice the difference between the maximum attainable log likelihood and the log likelihood at the maximum likelihood estimates of the regression parameters, is often used as a measure of goodness of fit. The maximum attainable log likelihood is achieved with a model that has a parameter for every observation. The following table displays the deviance for each of the probability distributions available in JMP.

**Table 10.2** Deviance Functions

Distribution	Deviance
normal	$\sum_i w_i (y_i - \mu_i)^2$
Poisson	$2 \sum_i w_i \left[ y_i \log\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i) \right]$
binomial <sup>a</sup>	$2 \sum_i w_i m_i \left[ y_i \log\left(\frac{y_i}{\mu_i}\right) + (1 - y_i) \log\left(\frac{1 - y_i}{1 - \mu_i}\right) \right]$
exponential	$2 \sum_i w_i \left[ -\log\left(\frac{y_i}{\mu_i}\right) + \left(\frac{y_i - \mu_i}{\mu_i}\right) \right]$

a. In the binomial case,  $y_i = r_i / m_i$ , where  $r_i$  is a binomial count and  $m_i$  is the binomial number of trials parameter

The Pearson chi-square statistic is defined as

$$\chi^2 = \sum_i \frac{w_i (y_i - \mu_i)^2}{V(\mu_i)}$$

where  $y_i$  is the  $i^{th}$  response,  $\mu_i$  is the corresponding predicted mean,  $V(\mu_i)$  is the variance function, and  $w_i$  is a known weight for the  $i^{th}$  observation. If no weight is known,  $w_i = 1$  for all observations.

One strategy for variable selection is to fit a sequence of models, beginning with a simple model with only an intercept term, and then include one additional explanatory variable in each successive model. You can measure the importance of the additional explanatory variable by the difference in deviances or fitted log likelihoods between successive models. Asymptotic tests computed by JMP enable you to assess the statistical significance of the additional term.

## Examples

The following examples illustrate how to use JMP's generalized linear models platform.

### Poisson Regression

This example uses data from a study of nesting horseshoe crabs. Each female crab had a male crab resident in her nest. This study investigated whether there were other males, called satellites, residing nearby. The data set CrabSatellites.jmp contains a response variable listing the number of satellites, as well as variables describing the female crab's color, spine condition, weight, and carapace width. The data are shown in Figure 10.2.

**Figure 10.2** Crab Satellite Data

The screenshot shows the JMP Data Table interface. On the left, the column structure is displayed under 'Model' and 'Columns (6/0)'. The 'Model' section includes 'satell', 'color\*', 'spine\*', 'width', 'weight', and 'weightstd+'. The 'Columns (6/0)' section lists 'satell', 'color\*', 'spine\*', 'width', 'weight', and 'weightstd'. Below this, 'Rows' are listed: All rows (173), Selected (0), Excluded (0), Hidden (0), and Labelled (0). The main table area shows 17 rows of data with the following approximate values:

	satell	color	spine	width	weight	weightstd
1	8	Medium	Both Worn/Broke	28.3	3050	1.06201468
2	0	Dark Med	Both Worn/Broke	22.5	1550	-1.5375251
3	9	Light Med	Both Good	26	2300	-0.2377552
4	0	Dark Med	Both Worn/Broke	24.8	2100	-0.5843605
5	4	Dark Med	Both Worn/Broke	26	2800	0.28215275
6	0	Medium	Both Worn/Broke	23.8	2100	-0.5843605
7	0	Light Med	Both Good	26.5	2350	-0.1511039
8	0	Dark Med	One Worn/Broke	24.7	1900	-0.9309658
9	0	Medium	Both Good	23.7	1950	-0.8443145
10	0	Dark Med	Both Worn/Broke	25.6	2150	-0.4977092
11	0	Dark Med	Both Worn/Broke	24.3	2150	-0.4977092
12	0	Medium	Both Worn/Broke	25.8	2850	0.36880407
13	11	Medium	Both Worn/Broke	28.2	3050	1.06201468
14	0	Dark	One Worn/Broke	21	1850	-1.0176171
15	14	Medium	Both Good	26	2300	-0.2377552
16	8	Light Med	Both Good	27.1	2950	0.88871203
17	1	Medium	Both Worn/Broke	25.2	2000	-0.7576632

To fit the Poisson loglinear model:

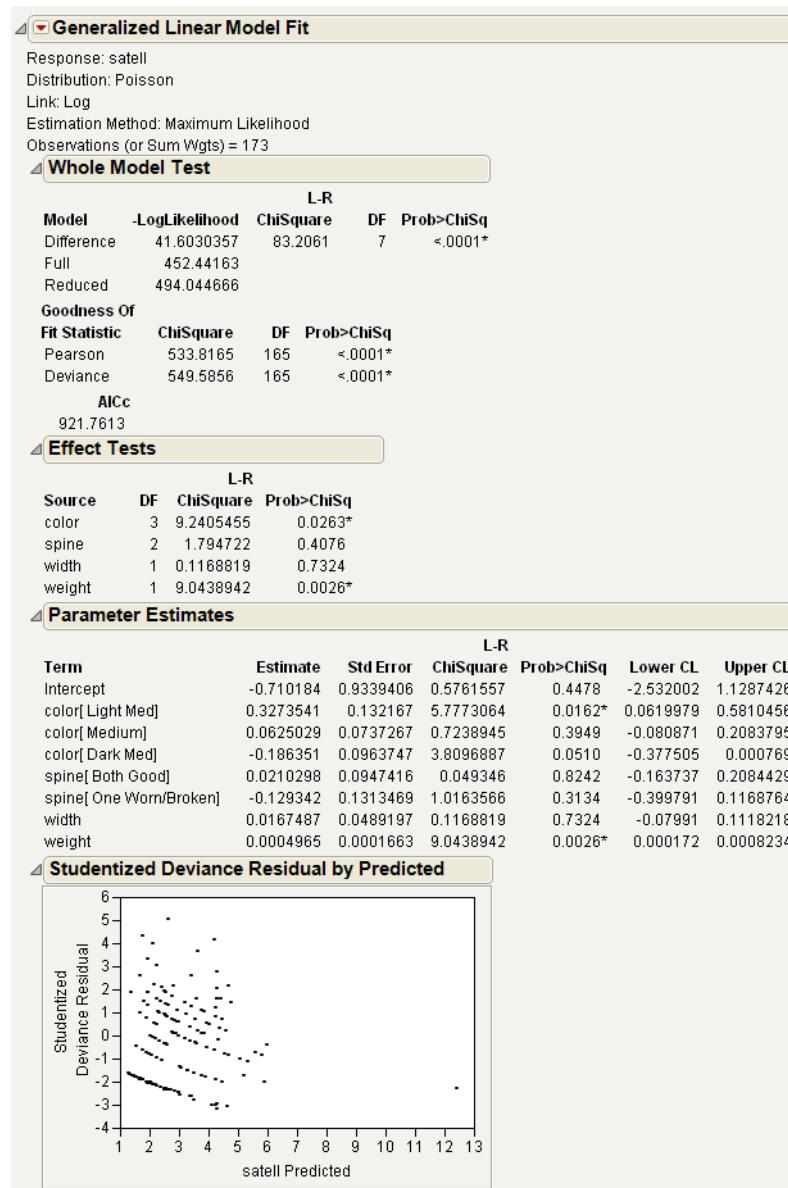
- Select **Analyze > Fit Model**
- Assign **satell** as **Y**
- Assign **color**, **spine**, **width**, and **weight** as **Effects**
- Choose the **Generalized Linear Model** Personality

- Choose the **Poisson** Distribution

The **Log** Link function should be selected for you automatically.

- Click **Run**.

The results are shown in Figure 10.3.

**Figure 10.3** Crab Satellite Results

The Whole Model Test table gives information to compare the whole-model fit to the model that contains only the intercept parameter. The Reduced model is the model containing only an intercept. The Full model contains all of the effects as well as the intercept. The Difference is the difference of the log likelihoods of the full and reduced models. The Prob>Chisq is analogous to a whole-model *F*-test.

Second, goodness-of-fit statistics are presented. Analogous to lack-of-fit tests, they test for adequacy of the model. Low p-values for the ChiSquare goodness-of-fit statistics indicate that you may need to add higher-order terms to the model, add more covariates, change the distribution, or (in Poisson and binomial cases especially) consider adding an overdispersion parameter. AICc is also included and is the corrected Akaike's Information Criterion, where

$$\text{AICc} = -2\text{loglikelihood} + 2k + \frac{2k(k+1)}{n-k-1}$$

and k is the number of estimated parameters in the model and n is the number of observations in the data set. This value may be compared with other models to determine the best-fitting model for the data. The model having the smallest value, as discussed in Akaike (1974), is usually the preferred model.

The Effect Tests table shows joint tests that all the parameters for an individual effect are zero. If an effect has only one parameter, as with simple regressors, then the tests are no different from the tests in the Parameter Estimates table.

The Parameter Estimates table shows the estimates of the parameters in the model and a *test* for the hypothesis that each parameter is zero. Simple continuous regressors have only one parameter. Models with complex classification effects have a parameter for each anticipated degree of freedom. Confidence limits are also displayed.

## Poisson Regression with Offset

The sample data table Ship Damage.JMP is adapted from one found in McCullugh and Nelder (1983). It contains information on a certain type of damage caused by waves to the forward section of the hull. Hull construction engineers are interested in the risk of damage associated with three variables: ship Type, the year the ship was constructed (Yr Made) and the block of years the ship saw service (Yr Used).

In this analysis we use the variable Service, the log of the aggregate months of service, as an *offset variable*. An offset variable is one that is treated like a regression covariate whose parameter is fixed to be 1.0.

These are most often used to scale the modeling of the mean in Poisson regression situations with log link. In this example, we use log(months of service) since one would expect that the number of repairs be proportional to the number of months in service. To see how this works, assume the linear component of the GLM is called eta. Then with a log link function, the model of the mean with the offset included is:

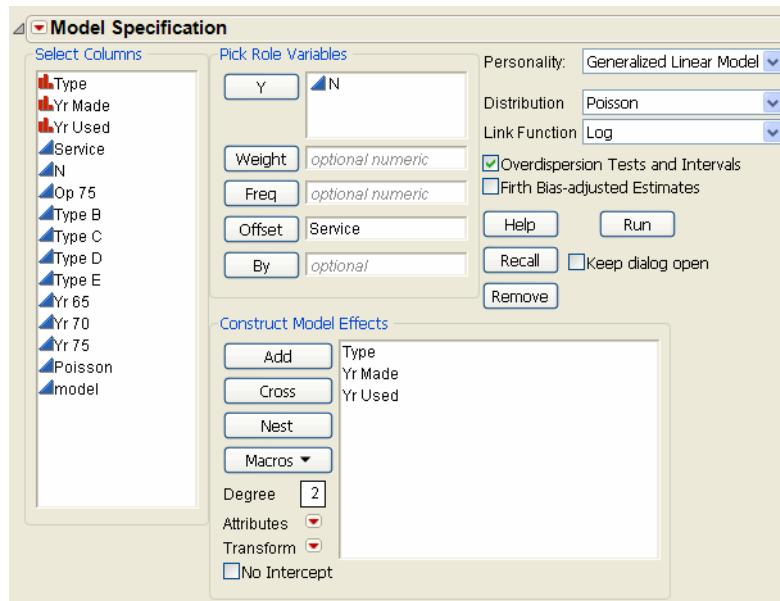
$$\exp[\text{Log}(\text{months of service}) + \text{eta}] = [(\text{months of service}) * \exp(\text{eta})].$$

To run this example, assign

- Generalized Linear Model as the Personality
- Poisson as the Distribution, which automatically selects the Log link function
- N to Y
- Service to Offset
- Type, Yr Made, Yr Used as effects in the model
- Overdispersion Tests and Intervals with a check mark

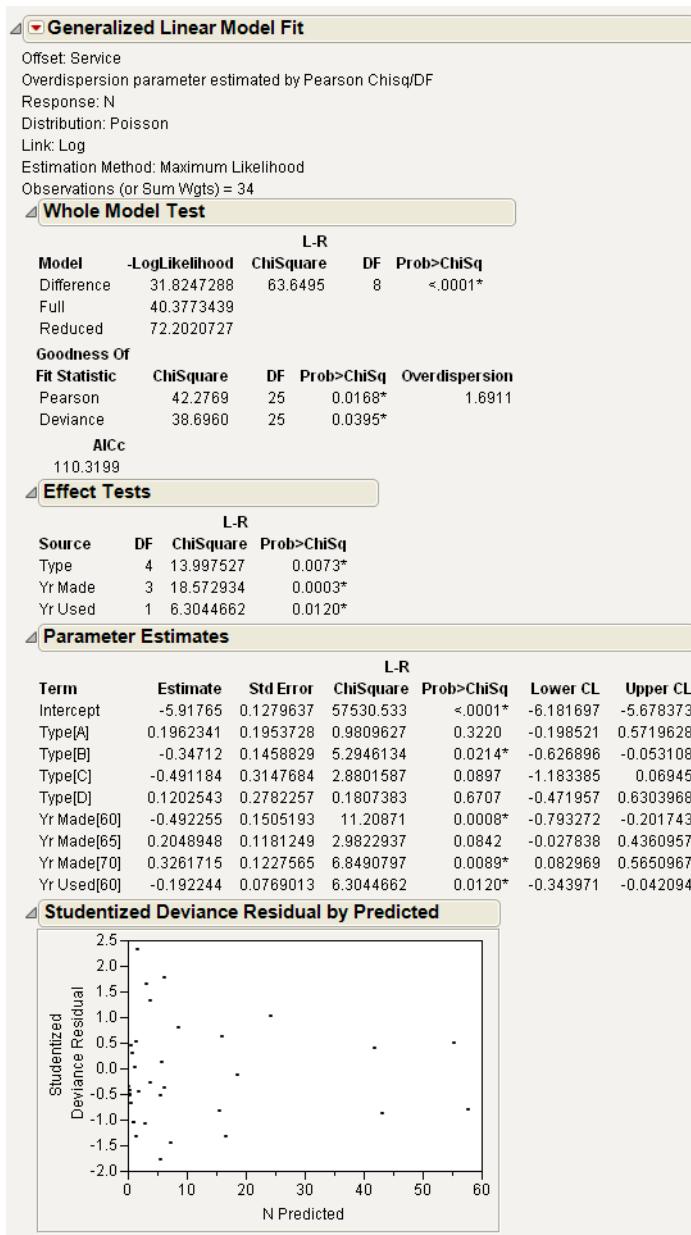
The Fit Model dialog should appear like the one shown in Figure 10.4.

---

**Figure 10.4** Ship Damage Fit Model Dialog

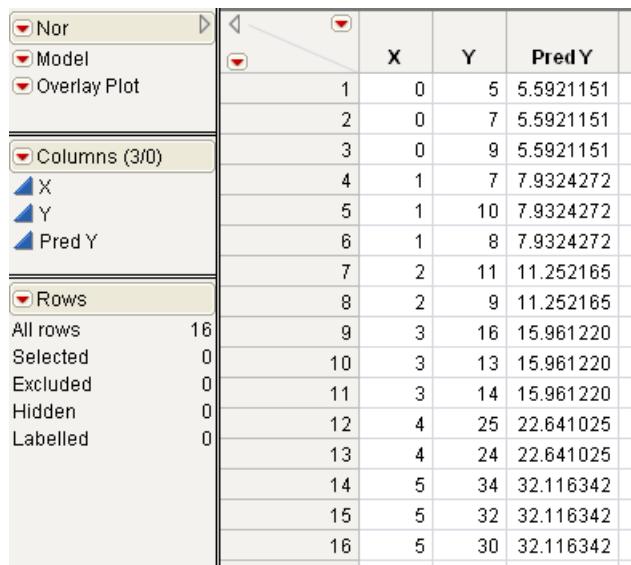
---

When you click Run, you see the report shown in Figure 10.5. Notice that all three effects (Type, Yr Made, Yr Used) are significant.

**Figure 10.5 Ship Damage Report**

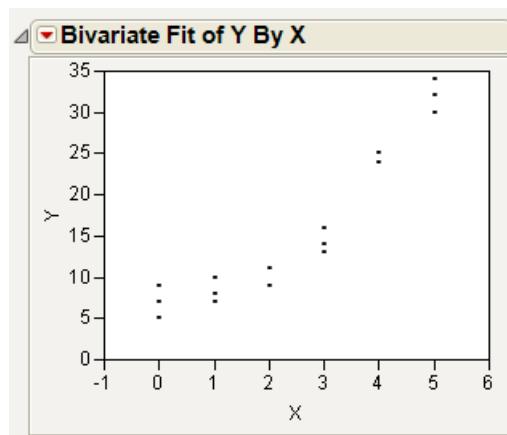
## Normal Regression, Log Link

Consider the following data set, where  $x$  is an explanatory variable and  $y$  is the response variable.

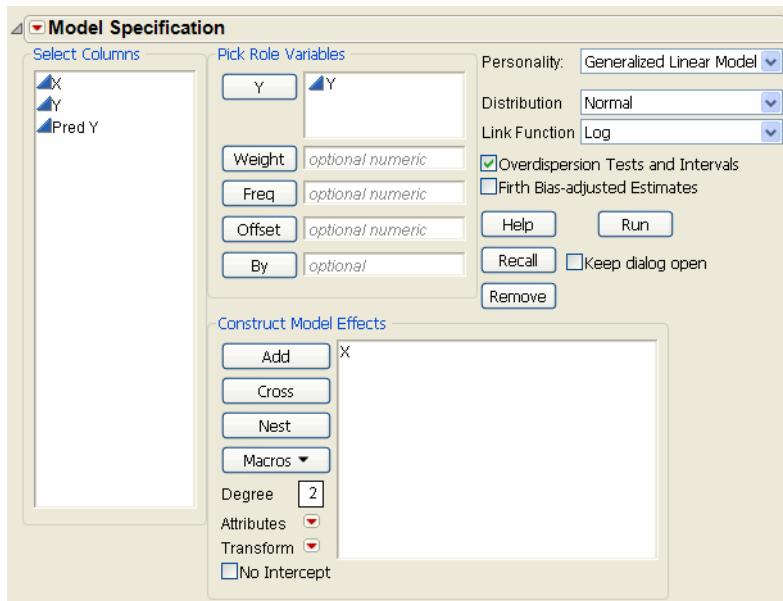
**Figure 10.6** Nor.jmp data set


		X	Y	Pred Y	
1	0	5	5.5921151		
2	0	7	5.5921151		
3	0	9	5.5921151		
4	1	7	7.9324272		
5	1	10	7.9324272		
6	1	8	7.9324272		
7	2	11	11.252165		
8	2	9	11.252165		
All rows	16	9	16	15.981220	
Selected	0	10	3	13	15.981220
Excluded	0	11	3	14	15.981220
Hidden	0	12	4	25	22.641025
Labelled	0	13	4	24	22.641025
		14	5	34	32.116342
		15	5	32	32.116342
		16	5	30	32.116342

Using Fit Y By X, you can easily see that  $y$  varies nonlinearly with  $x$  and that the variance is approximately constant (see Figure 10.7). A normal distribution with a log link function is chosen to model these data; that is,  $\log(\mu_i) = \mathbf{x}_i'\beta$  so that  $\mu_i = \exp(\mathbf{x}_i'\beta)$ . The completed Fit Model dialog is shown in Figure 10.8.

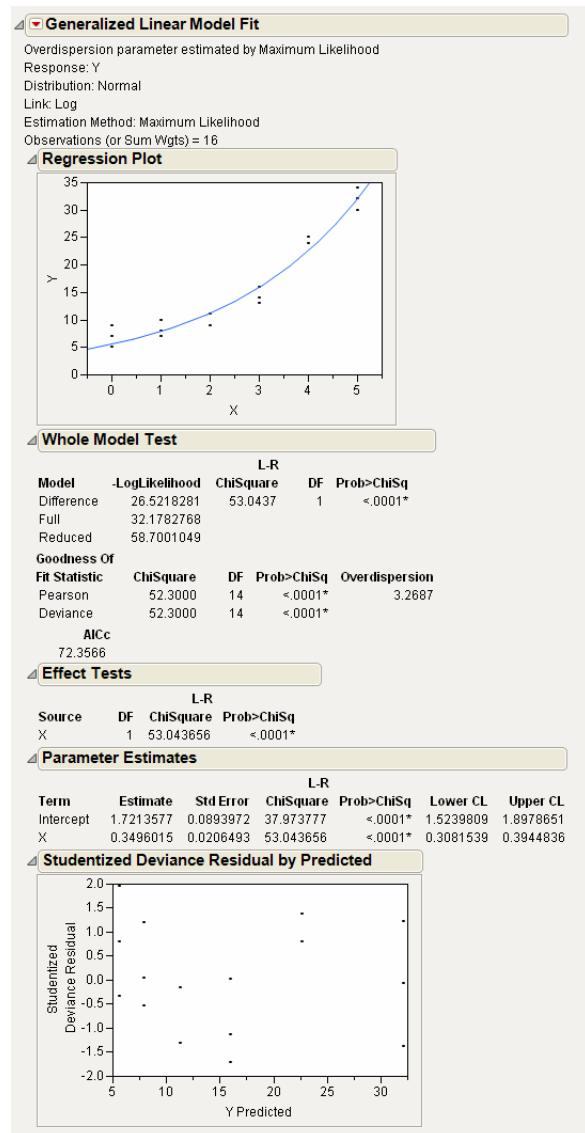
**Figure 10.7** Y by X Results for Nor.jmp

---

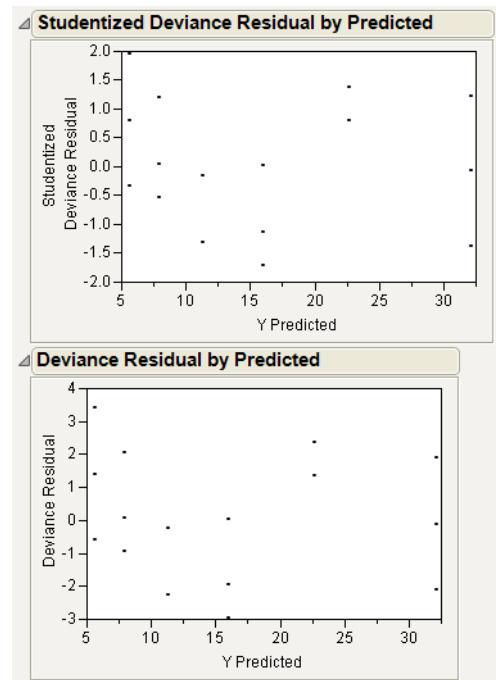
**Figure 10.8** Nor Fit Model Dialog

---

After clicking **Run**, you get the following report.

**Figure 10.9** Nor Results

Because the distribution is normal, the Studentized Deviance residuals and the Deviance residuals are the same. To see this, select **Diagnostic Plots > Deviance Residuals by Predicted** from the platform drop-down menu.



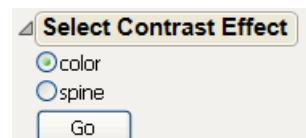
## Platform Commands

The following commands are available in the Generalized Linear Model report.

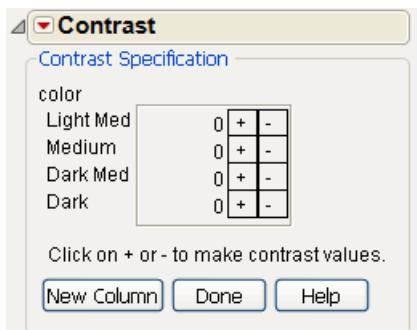
**Custom Test** allows you to test a custom hypothesis. Refer to “[Custom Test](#),” p. 63 in the “Standard Least Squares: Perspectives on the Estimates” chapter for details on custom tests.

**Contrast** allows you to test for differences in levels within a variable. If a contrast involves a covariate, you can specify the value of the covariate at which to test the contrast.

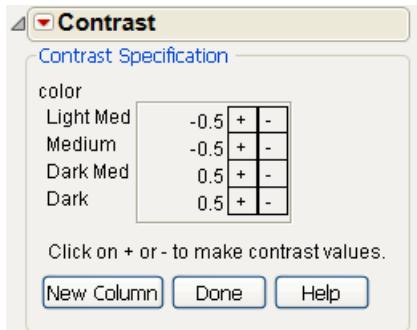
In the Crab Satellite example, suppose you want to test whether the dark-colored crabs attracted a different number of satellites than the medium-colored crabs. Selecting **Contrast** brings up the following dialog.



Here you choose **color**, the variable of interest. When you click **Go**, you are presented with a Contrast Specification dialog.



To compare the dark-colored to the medium-colored, click the + button beside Dark and Dark Med, and the - button beside Medium and Light Medium.



Click **Done** to get the Contrast report shown here.

**Level**

color[ Light Med]	-0.5
color[ Medium]	-0.5
color[ Dark Med]	0.5
color[ Dark]	0.5
Value	-0.389856982
Std Error	0.1324623948
ChiSquare	8.6621445816
Prob>ChiSq	0.003248887
-LogLikelihood	.

**-LogLikelihood**

DF	1
L-R ChiSquare	8.6621445816
Prob>ChiSq	0.003248887

Missing values of -LogLikelihood indicate that a suboptimization step failed to converge. In these cases a Wald test statistic and p-value was provided rather than a likelihood ratio test.

Since the Prob>Chisq is less than 0.05, we have evidence that there is a difference in satellite attraction based on color.

**Inverse Prediction** is used to predict an  $X$  value, given specific values for  $Y$  and the other  $X$  variables. This can be used to predict continuous variables only. For more details about Inverse Prediction, see “[Inverse Prediction](#),” p. 65 in the “Standard Least Squares: Perspectives on the Estimates” chapter.

**Covariance of Estimates** produces a covariance matrix for all the effects in a model. The estimated covariance matrix of the parameter estimator is given by

$$\Sigma = -\mathbf{H}^{-1}$$

where  $\mathbf{H}$  is the Hessian (or second derivative) matrix evaluated using the parameter estimates on the last iteration. Note that the dispersion parameter, whether estimated or specified, is incorporated into  $\mathbf{H}$ . Rows and columns corresponding to aliased parameters are not included in  $\Sigma$ .

**Correlation of Estimates** produces a correlation matrix for all the effects in a model. The correlation matrix is the normalized covariance matrix. That is, if  $\sigma_{ij}$  is an element of  $\Sigma$ , then the corresponding element of the correlation matrix is  $\sigma_{ij}/\sigma_i\sigma_j$ , where  $\sigma_i = \sqrt{\sigma_{ii}}$

**Profiler** brings up the Profiler for examining prediction traces for each X variable. Details on the profiler are found in “[The Profiler](#),” p. 89 in the “Standard Least Squares: Exploring the Prediction Equation” chapter.

**Contour Profiler** brings up an interactive contour profiler. Details are found in “[Contour Profiler](#),” p. 555 in the “Profiling” chapter.

**Surface Profiler** brings up a 3-D surface profiler. Details of Surface Plots are found in the “[Plotting Surfaces](#)” chapter.

**Diagnostic Plots** is a submenu containing commands that allow you to plot combinations of residuals, predicted values, and actual values to search for outliers and determine the adequacy of your model. Deviance is discussed above in “[Model Selection and Deviance](#),” p. 202. The following plots are available:

- **Studentized Deviance Residuals by Predicted**
- **Studentized Pearson Residuals by Predicted**
- **Deviance Residuals by Predicted**
- **Pearson Residuals By Predicted**
- **Actual by Predicted**
- **Regression Plot** is available only when there is one continuous predictor and no more than one categorical predictor.
- **Linear Predictor Plot** is a plot of responses transformed by the inverse link function. This plot is available only when there is one continuous predictor and no more than one categorical predictor.

**Save Columns** is a submenu that lets you save certain quantities as new columns in the data table. Formulas for residuals are shown in Table 10.3.

**Prediction Formula** saves the formula that predicts the current model.

**Predicted Values** saves the values predicted by the current model.

**Mean Confidence Interval** saves the 95% confidence limits for the prediction equation. The confidence limits reflect variation in the parameter estimates.

**Save Indiv Confid Limits** saves the confidence limits for a given individual value. The confidence limits reflect variation in the error and variation in the parameter estimates.

**Deviance Residuals** saves the deviance residuals.

**Pearson Residuals** saves the Pearson residuals.

**Studentized Deviance Residuals** saves the studentized deviance residuals.

**Studentized Pearson Residuals** saves the studentized Pearson residuals.

You can also save the parametric formula using JSL:

```
fit model object <<Parametric Formula( );
```

See the Object Scripting Index for an example.

**Table 10.3** Residual Formulas

Residual Type	Formula
Deviance	$r_{Di} = \sqrt{d_i} \operatorname{sign}(y_i - \mu_i)$
Studentized Deviance	$r_{Di} = \frac{\operatorname{sign}(y_i - \mu_i) \sqrt{d_i}}{\sqrt{\phi(1 - b_i)}}$

**Table 10.3** Residual Formulas

Residual Type	Formula
Pearson	$r_{Pi} = \frac{(y_i - \mu_i)}{\sqrt{V(\mu_i)}}$
Studentized Pearson	$r_{Pi} = \frac{y_i - \mu_i}{\sqrt{V(\mu_i)(1 - h_i)}}$

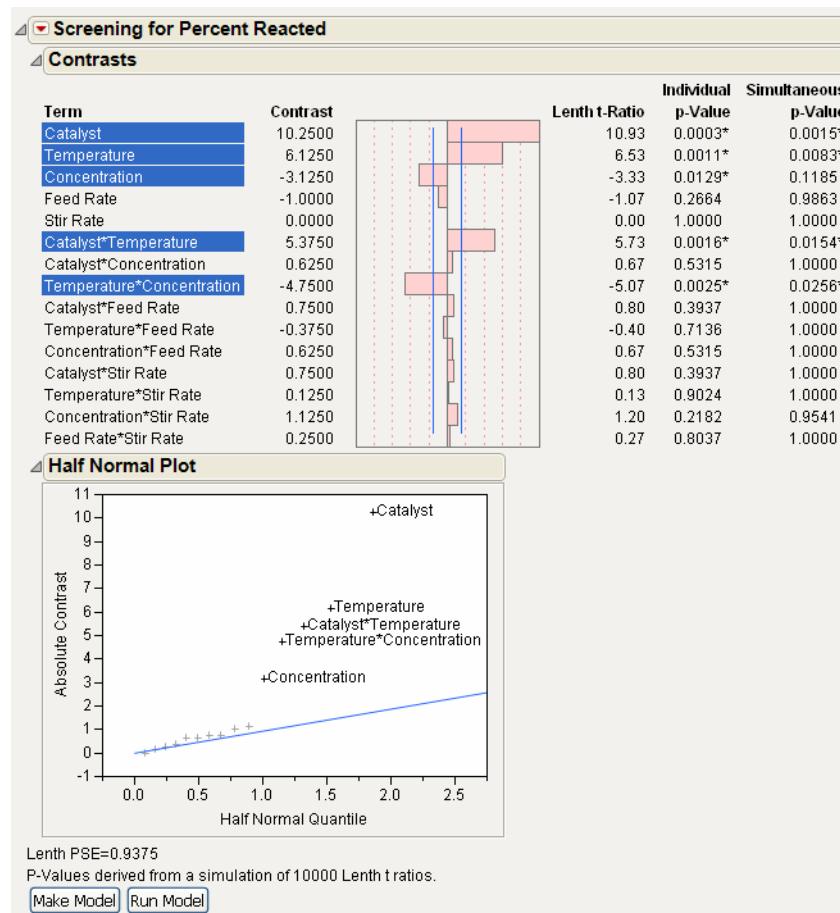
where  $(y_i - \mu_i)$  is the raw residual,  $\text{sign}(y_i - \mu_i)$  is 1 if  $(y_i - \mu_i)$  is positive and -1 if  $(y_i - \mu_i)$  is negative,  $d_i$  is the contribution to the total deviance from observation  $i$ ,  $\phi$  is the dispersion parameter,  $V(\mu_i)$  is the variance function, and  $h_i$  is the  $i^{\text{th}}$  diagonal element of the matrix  $W_e^{(1/2)}X(X'W_eX)^{-1}X'W_e^{(1/2)}$ , where  $W_e$  is the weight matrix used in computing the expected information matrix. For additional information regarding residuals and generalized linear models, see “The GENMOD Procedure” in the SAS/STAT User Guide documentation.

# Chapter 11

## Analyzing Screening Designs The Screening Platform

The Screening platform helps select a model that fits a two-level screening design by indicating which factors have the largest effect on the response. An example of a screening report is shown in Figure 11.1.

**Figure 11.1** Screening Platform Report



# Contents

The Screening Platform .....	219
Using the Screening Platform .....	219
Comparing Screening and Fit Model.....	219
Launch the Platform .....	222
Report Elements and Commands .....	222
Contrasts.....	222
Half Normal Plot .....	223
Launching a Model .....	223
Tips on Using the Platform .....	223
Statistical Details.....	224
Analyzing a Plackett-Burman Design.....	226
Analyzing a Supersaturated Design .....	227

## The Screening Platform

For two-level screening designs, the goal is to search for large effects. The screening platform looks for these big effects, helping you formulate a model for fitting the screening design.

Screening situations depend on *effect sparsity*, where most effects are assumed to be inactive. Using this assumption, the smaller estimates can be used to help estimate the error in the model and determine whether the larger effects are real. Basically, if all the effects are inactive, they should vary randomly, with no effect deviating substantially from the other effects.

## Using the Screening Platform

If your data are all two-level and orthogonal, then all the statistics in this platform should work well.

If categorical terms have more than two levels, then the Screening platform is not appropriate for the design. JMP treats the level numbers as a continuous regressor. The variation across the factor is scattered across main and polynomial effects for that term.

For highly supersaturated main effect designs, the Screening platform is effective in selecting factors, but is not as effective at estimating the error or the significance. The Monte Carlo simulation to produce *p*-values uses assumptions that are valid for this case.

If your data are not orthogonal, then the constructed estimates are different from standard regression estimates. JMP can pick out big effects, but it does not effectively test each effect. This is because later effects are artificially orthogonalized, making earlier effects look more significant.

Note that the Screening platform is not appropriate for mixture designs.

## Comparing Screening and Fit Model

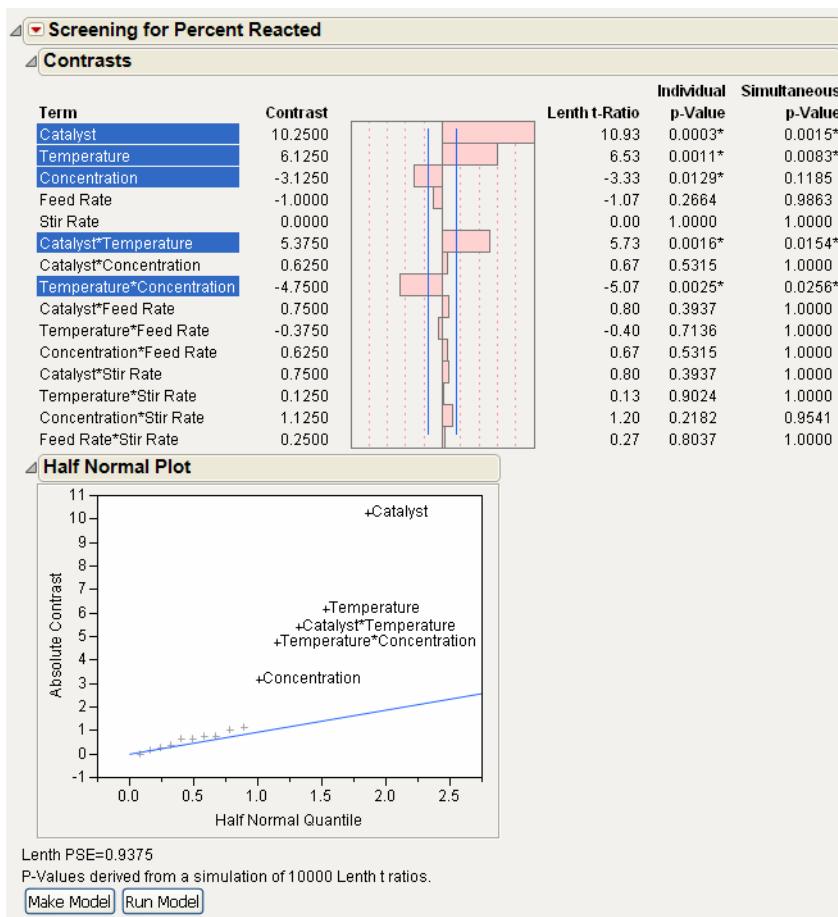
Consider Reactor Half Fraction.jmp, from the Sample Data folder. The data are derived from a design in Box, Hunter, and Hunter (1978). We are interested in a model with main effects and two-way interactions. This example uses a model with fifteen parameters for a design with sixteen runs.

Figure 11.2 shows the result of using the Fit Model platform, where a factorial to degree 2 model is specified. This result illustrates why the Screening platform is needed.

**Figure 11.2** Traditional Saturated Reactor Half Fraction.jmp Design Output

Response Percent Reacted				
Summary of Fit				
RSquare	1			
RSquare Adj				
Root Mean Square Error				
Mean of Response	65.25			
Observations (or Sum Wgts)	16			
Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	65.25	.	.	.
Feed Rate(10,15)	-1	.	.	.
Catalyst(1,2)	10.25	.	.	.
Stir Rate(100,120)	0	.	.	.
Temperature(140,180)	6.125	.	.	.
Concentration(3,6)	-3.125	.	.	.
Feed Rate*Catalyst	0.75	.	.	.
Feed Rate*Stir Rate	0.25	.	.	.
Feed Rate*Temperature	-0.375	.	.	.
Feed Rate*Concentration	0.625	.	.	.
Catalyst*Stir Rate	0.75	.	.	.
Catalyst*Temperature	5.375	.	.	.
Catalyst*Concentration	0.625	.	.	.
Stir Rate*Temperature	0.125	.	.	.

JMP can calculate parameter estimates, but degrees of freedom for error, standard errors, *t*-ratios, and *p*-values are all missing. Rather than use Fit Model, it is better to use the Screening platform, which specializes in getting the most information out of these situations, leading to a better model. The output from the Screening platform for the same data is shown in Figure 11.3.

**Figure 11.3** Reactor Half Fraction.jmp Screening Design Report

Compare the following differences between the Fit Model report and the Screening report.

- Estimates labeled **Contrast**. Effects whose individual *p*-value is less than 0.1 are highlighted.
- A *t*-ratio is calculated using Lenth's PSE (pseudo-standard error). The PSE is shown below the Half Normal Plot.
- Both individual and simultaneous *p*-values are shown. Those that are less than 0.05 are shown with an asterisk.
- A Half Normal plot enables you to quickly examine the effects. Effects initially highlighted in the effects list are also labeled in this plot.
- Buttons at the bottom of the report also operate on the highlighted variables. The **Make Model** button opens the Fit Model window using the current highlighted factors. The **Run Model** button runs the model immediately.

## Launch the Platform

For this example, Catalyst, Temperature, and Concentration, along with two of their two-factor interactions, are selected.

---

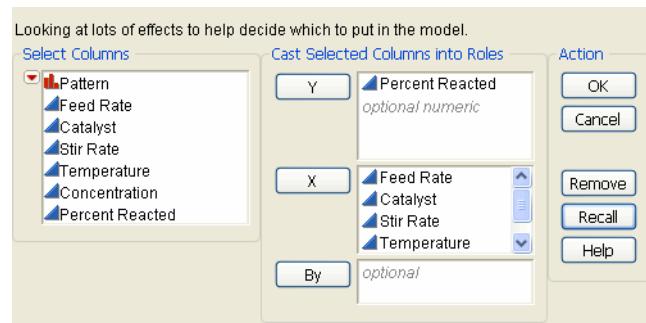
## Launch the Platform

The Screening platform is launched via **Analyze > Modeling > Screening**. The completed launch window is shown in Figure 11.4. This window was used to produce the report in Figure 11.3.

For this example, all continuous factors, except the response factor, are selected as the screening effects, *X*. Percent Reacted is selected as the response *Y*. JMP constructs interactions automatically, unlike Fit Model, where the interactions are added distinctly.

---

**Figure 11.4** Screening Platform Launch Window for Reactor Half Fraction.jmp




---

## Report Elements and Commands

The following information is shown by default in the Screening platform report.

### Contrasts

The **Contrasts** section shows the following columns.

**Term** is the factor name.

**Contrast** is the estimate for the factor. For orthogonal designs, this number is the same as the regression parameter estimate. This is not the case for non-orthogonal designs. An asterisk might appear next to the contrast, indicating lack of orthogonality.

**Bar Chart** shows the *t*-ratios with blue lines marking a critical value at 0.05 significance.

**Lenth t-Ratio** is Lenth's *t*-ratio, calculated as  $\frac{\text{Contrast}}{\text{PSE}}$ , where PSE is Lenth's Pseudo-Standard Error. See “[Lenth's Pseudo-Standard Error](#),” p. 225 for details.

**Individual p-Value** is analogous to the standard *p*-values for a linear model. Small values of this value indicate a significant effect. Refer to “[Statistical Details](#),” p. 224 for details.

**Simultaneous p-Value** is used like the individual  $p$ -value, but is multiple-comparison adjusted.

**Aliases** appears only if there are exact aliases of later effects to earlier effects.

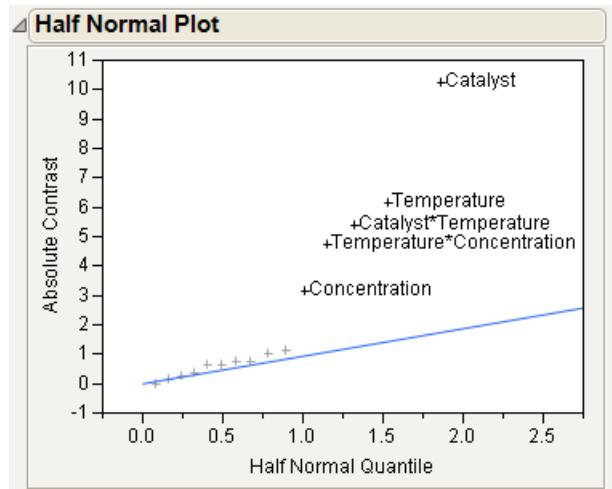
## Half Normal Plot

The Half Normal Plot shows the absolute value of the contrasts against the normal quantiles for the absolute value normal distribution. Significant effects show as being separated from the line toward the upper right of the graph.

Note that this plot is interactive. Select different model effects by dragging a rectangle around the effects of interest. Alternatively, you can Control-click effect names in the report. The Half Normal Plot for Reactor Half Fraction.jmp is shown in Figure 11.5.

---

**Figure 11.5** Half Normal Plot for Reactor Half Fraction.jmp



---

## Launching a Model

The two buttons at the bottom of the plot construct a model using the currently selected effects.

**Make Model** launches the Fit Model window populated with selected effects and responses.

**Run Model** launches the Fit Model window, but also runs the model immediately.

---

## Tips on Using the Platform

**Higher-Order Effects.** Control-click to select more or fewer effects. The effect selection is not constrained by hierarchy. For example, if  $A*B$  is selected, check to make sure the lower-order effects

(the A and B main effects) are also selected. Control-click to select them if they are not already selected.

**Re-running the Analysis.** Do not expect the *p*-values to be exactly the same if the analysis is re-run.

The Monte Carlo method should give similar, but not identical, values if the same analysis is repeated.

## Statistical Details

### Operation

The Screening platform has a carefully defined order of operations.

- First, the main effect terms enter according to the absolute size of their contrast. All effects are orthogonalized to the effects preceding them in the model. The method assures that their order is the same as it would be in a forward stepwise regression. Ordering by main effects also helps in selecting preferred aliased terms later in the process.
- After main effects, all second-order interactions are brought in, followed by third-order interactions, and so on. The second-order interactions cross with all earlier terms before bringing in a new term. For example, with size-ordered main effects A, B, C, and D, B\*C enters before A\*D. If a factor has more than two levels, square and higher-order polynomial terms are also considered.
- An effect that is an exact alias for an effect already in the model shows in the alias column. Effects that are a linear combination of several previous effects are not displayed. If there is partial aliasing (a lack of orthogonality) the effects involved are marked with an asterisk.
- The process continues until *n* effects are obtained, where *n* is the number of rows in the data table, thus fully saturating the model. If complete saturation is not possible with the factors, JMP generates random orthogonalized effects to absorb the rest of the variation. They are labeled Null *n* where *n* is a number. For example, this situation occurs if there are exact replicate rows in the design.

### Screening as an Orthogonal Rotation

Mathematically, the Screening platform takes the  $n$  values in the response vector and rotates them into  $n$  new values. The rotated values are then mapped by the space of the factors and their interactions.

$$\text{Contrasts} = \mathbf{T}' \times \text{Responses}$$

where  $\mathbf{T}$  is an orthonormalized set of values starting with the intercept, main effects of factors, two-way interactions, three-way interactions, and so on, until  $n$  values have been obtained. Since the first column of  $\mathbf{T}$  is an intercept, and all the other columns are orthogonal to it, these other columns are all contrasts, that is, they sum to zero. Since  $\mathbf{T}$  is orthogonal, it can serve as  $\mathbf{X}$  in a linear model. It does not need inversion, since  $\mathbf{T}'$  is also  $\mathbf{T}^{-1}$  and  $(\mathbf{T}'\mathbf{T})\mathbf{T}'$ . The contrasts are the parameters estimated in a linear model.

If no effect in the model is active after the intercept, the contrasts are just an orthogonal rotation of random independent variates into different random independent variates. These newly orthogonally rotated variates have the same variance as the original random independent variates. To the extent that some effects are active, the inactive effects still represent the same variation as the error in the model. The hope is that the effects and the design are strong enough to separate the active effects from the random error effects.

### Lenth's Pseudo-Standard Error

At this point, Lenth's method (Lenth, 1989) identifies inactive effects from which it constructs an estimate of the residual standard error, known as the *Lenth Pseudo Standard Error (PSE)*.

The value for Lenth's PSE is shown at the bottom of the Screening report. From the PSE,  $t$ -ratios are obtained. To generate  $p$ -values, a Monte Carlo simulation of 10,000 runs of  $n - 1$  purely random values is created and Lenth  $t$ -ratios are produced from each set. The  $p$ -value is the interpolated fractional position among these values in descending order. The simultaneous  $p$ -value is the interpolation along the  $\max(|t|)$  of the  $n - 1$  values across the runs. This technique is similar to that in Ye and Hamada (2000).

If you want to run more or less than the 10,000 default runs, you must assign a value to a global JSL variable named `LenthSimN`. As an example, using the sample data `Reactor Half Fraction.jmp`:

1. Open the sample data, `Reactor Half Fraction.jmp`.
2. Select **Analyze > Modeling > Screening**.
3. Select **Percent Reacted** as the response variable,  $Y$ .
4. Select all the other continuous variables as effects,  $X$ .
5. Click **OK**.
6. Select **Script > Save Script to Script Window** from the red-triangle menu of the report.
7. Add `LenthSimN=50000;` to the top of the Script Window (above the code).
8. Highlight `LenthSimN=50000;` and the remaining code.
9. Run the script from the Script Window.

Note that if `LenthSimN=0`, the standard  $t$ -distribution is used (not recommended).

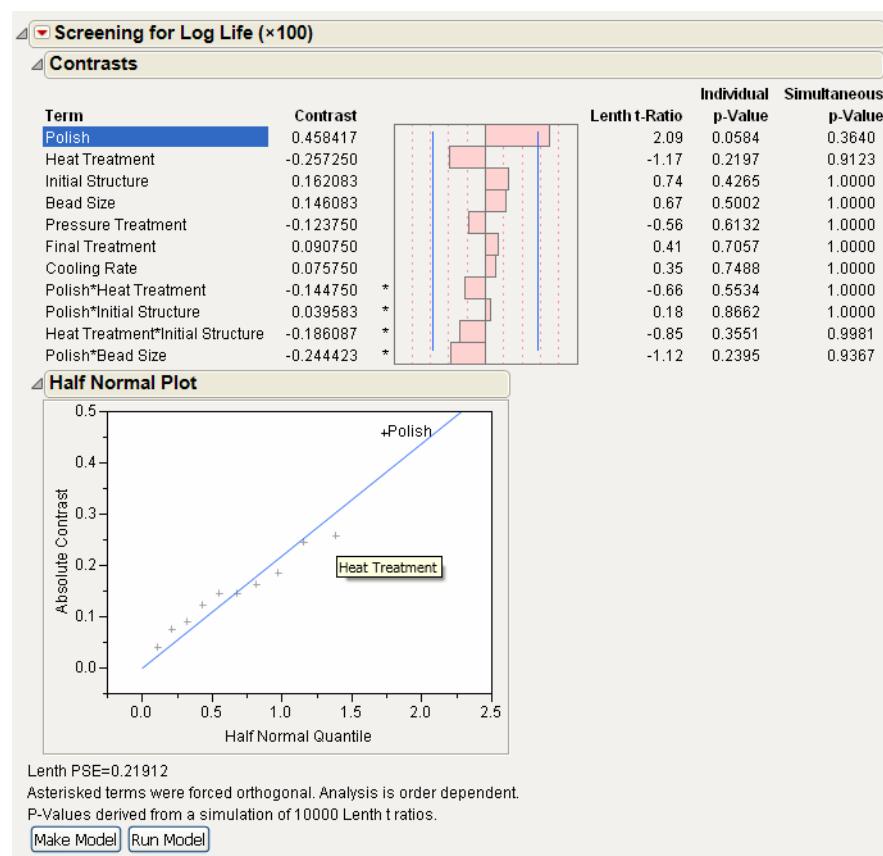
## Analyzing a Plackett-Burman Design

Plackett-Burman designs are an alternative to fractional-factorial screening designs. Two-level fractional factorial designs must, by their nature, have a number of runs that are a power of two. Plackett-Burman designs exist for 12-, 24-, and 28-run designs.

Weld-Repaired Castings.jmp from the Sample Data folder uses a Plackett-Burman design, and is found in textbooks such as Giesbrecht and Gumpertz (2004) and Box, Hunter, and Hunter (2005). Seven factors are thought to be influential on weld quality. The seven factors include Initial Structure, Bead Size, Pressure Treatment, Heat Treatment, Cooling Rate, Polish, and Final Treatment. A Plackett-Burman design with 12 runs is used to investigate the importance of the seven factors. The response is  $100 \times \log(\text{lifetime})$ . (There are also four terms that were used to model error that are not used in this analysis.)

Using the Screening platform, select the seven effects as **X** and Log Life as **Y**. (If terms are automatically populated in the Screening Platform launch window, remove the four error terms listed as effects.) Click **OK**. Figure 11.6 appears, showing only a single significant effect.

**Figure 11.6** Screening Report for Weld-Repaired Castings.jmp



Note asterisks mark four terms, indicating that they are not orthogonal to effects preceding them, and the obtained contrast value was after orthogonalization. So, they would not match corresponding regression estimates.

---

## Analyzing a Supersaturated Design

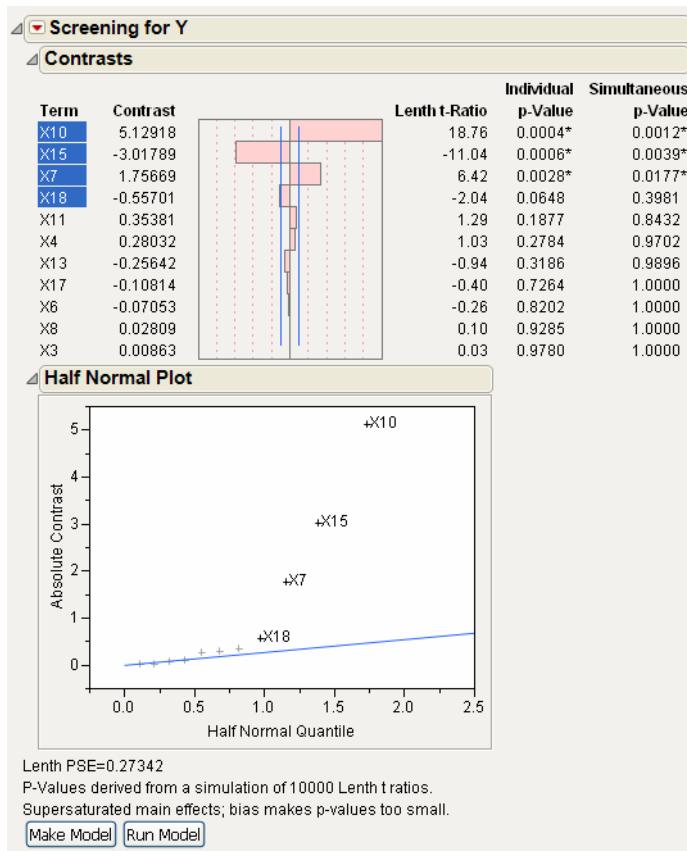
Supersaturated designs have more factors than runs. The objective is to determine which effects are active. They rely heavily on effect sparsity for their analysis, so the Screening platform is ideal for their analysis.

As an example, look at *Supersaturated.jmp*, from the Sample Data folder, a simulated data set with 18 factors but only 12 runs. *Y* is generated by

$$Y = 2(X7) + 5(X10) - 3(X15) + \epsilon$$

where  $\epsilon \sim N(0,1)$ . So, *Y* has been constructed with three active factors.

To detect the active factors, run the Screening platform with *X1–X18* as *X* and *Y* as *Y*. The report shown in Figure 11.7 appears.

**Figure 11.7** Screening Report for Supersaturated.jmp

Note that the three active factors have been highlighted. One other factor, X18, has also been highlighted. It shows in the Half Normal plot close to the blue line, indicating that it is close to the 0.1 cutoff significance value. The 0.1 critical value is generous in its selection of factors so you don't miss those that are possibly active.

The contrasts of 5.1, -3, and 1.8 are close to their simulated values (5, -3, 2). However, the similarity of these values can be increased by using a regression model, without the effect of orthogonalization.

The *p*-values, while useful, are not entirely valid statistically, since they are based on a simulation that assumes orthogonal designs, which is not the case for supersaturated designs.

# Chapter 12

## Nonlinear Regression

### The Nonlinear Platform

---

The **Nonlinear** command fits models that are nonlinear in their parameters, using least-squares or a custom loss function. You do not need to use this platform if the model is nonlinear in the variables but linear in the parameters. Similarly, you can use the linear platforms if you can transform your model to be linear in the parameters.

Nonlinear models are more difficult to fit than linear models. They require more preparation with the specification of the model and initial guesses for parameter values. Iterative methods are used to search for the least-squares estimates, and there is no guarantee that a solution will be found. Indeed it is possible to diverge, or even to converge on a local solution that is not the least-squares solution. Nonlinear fits do not have some of the nice properties that linear models have, and the results must be interpreted with caution. For example, there is no well defined  $R^2$  statistic; the standard errors of the estimates are approximations; and leverage plots are not provided.

Nonlinear fitting begins with a formula for the model that you build in a column using the formula editor or select using JMP's built-in model library. The formula is specified with parameters to be estimated. After entering the formula, select the **Nonlinear** command and work with its interactive iteration Control Panel to do the fitting.

The platform itself is easy to use. The iterative techniques employed in JMP are variations of the Gauss-Newton method with a line search. This method employs derivatives of the model with respect to each parameter, but these derivatives are obtained automatically by the platform. The Newton-Raphson method is also available by requesting second derivatives. There are features for finding profile confidence limits of the parameters and for plotting the fitted function if the model is a function of just one column.

It is also possible to define special loss functions to use instead of least-squares. The loss function can be a function of the model, other variables, and parameters. When a custom loss function is defined, the platform minimizes the sum of the loss function across rows; thus, if you want maximum likelihood, specify something that sums to the negative of the log-likelihood as the loss.

# Contents

The Nonlinear Fitting Process . . . . .	231
A Simple Exponential Example . . . . .	231
Creating a Formula with Parameters . . . . .	231
Launch the Nonlinear Platform . . . . .	232
Drive the Iteration Control Panel . . . . .	233
Using the Model Library . . . . .	235
Customizing the Nonlinear Model Library . . . . .	239
Details for the Formula Editor . . . . .	239
Details of the Iteration Control Panel . . . . .	240
Panel Buttons . . . . .	241
The Current Parameter Estimates . . . . .	241
Save Estimates . . . . .	242
Confidence Limits . . . . .	242
The Nonlinear Fit Popup Menu . . . . .	242
Details of Solution Results . . . . .	247
The Solution Table . . . . .	248
Excluded Points . . . . .	248
Profile Confidence Limits . . . . .	248
Fitted Function Graph . . . . .	249
Chemical Kinetics Example . . . . .	250
How Custom Loss Functions Work . . . . .	251
Maximum Likelihood Example: Logistic Regression . . . . .	253
Iteratively Reweighted Least Squares Example . . . . .	254
Probit Model with Binomial Errors: Numerical Derivatives . . . . .	257
Poisson Loss Function . . . . .	259
Notes Concerning Derivatives . . . . .	261
Notes on Effective Nonlinear Modeling . . . . .	262
Notes Concerning Scripting . . . . .	263
Nonlinear Modeling Templates . . . . .	264

---

## The Nonlinear Fitting Process

The **Nonlinear** command on the **Analyze > Modeling** submenu launches an interactive nonlinear fitting facility. You orchestrate the fitting process as a coordination of three important parts of JMP: the data table, the formula editor, and the Nonlinear platform.

- You define the column and its prediction formula with the formula editor. The formula is specified with parameters to be estimated. Use the formula editor to define parameters and give them initial values.
- Launch the Nonlinear platform with the response variable in the Y role and the model column with fitting formula in the Prediction Column. If no *Y* column is given, then the Prediction Column formula is for residuals. If you have a loss column specified, then you might not need either a Prediction column or *Y* column specification.
- Interact with the platform through the iteration Control Panel.

---

## A Simple Exponential Example

The US Population.jmp table in the Nonlinear Examples of the Sample Data folder illustrates a simple nonlinear exponential model.

The *pop* (population) values are modeled as a nonlinear function of *year*. To see the nonlinear formula, highlight the X-formula column (which contains the model). Then select the **Formula** command in the **Cols** menu. The column formula editor appears with a formula that describes an exponential model for the growth of population in the U. S. between 1790 and 1990.

### Creating a Formula with Parameters

When you fit a nonlinear formula to data, the first step is to create a column in the data table using the formula editor to build a prediction formula, which includes parameters to be estimated. The formula contains the parameters' initial values.

Begin in the formula editor by defining the parameters for the nonlinear formula. Select **Parameters** from the popup menu above the column selector list. The list changes from the column selector list to a parameter selector list. When **New Parameter** appears in the selector list, click on it and respond to the New Parameter definition dialog. Use this dialog to name the parameter and assign it an initial value. When you click **OK**, the new parameter name appears in the selector list. Continue this process to create additional parameters. The population example uses two parameters. You can now build the formula you need by using data table columns and parameters.

The parameter names are arbitrary and in this example were given initial values chosen as follows:

- **B0** is the prediction of population at year 1790, which should be near 3.93, the actual value of the first recorded population value in the data table. Therefore, 3.9 seems to be a reasonable initial value for **B0**.
- The **B1** growth rate parameter is given the initial value of 0.022, which is close to the estimate of the slope you get when you fit the natural log of *pop* to *year* with a straight line (or **Fit Special** and specify

In transformation for pop). This initial value seems reasonable because the nonlinear exponential model can have similar final parameter estimates.

You can now build a formula using data table columns and parameters.

**B0\*Exp(B1\*(year - 1790))**

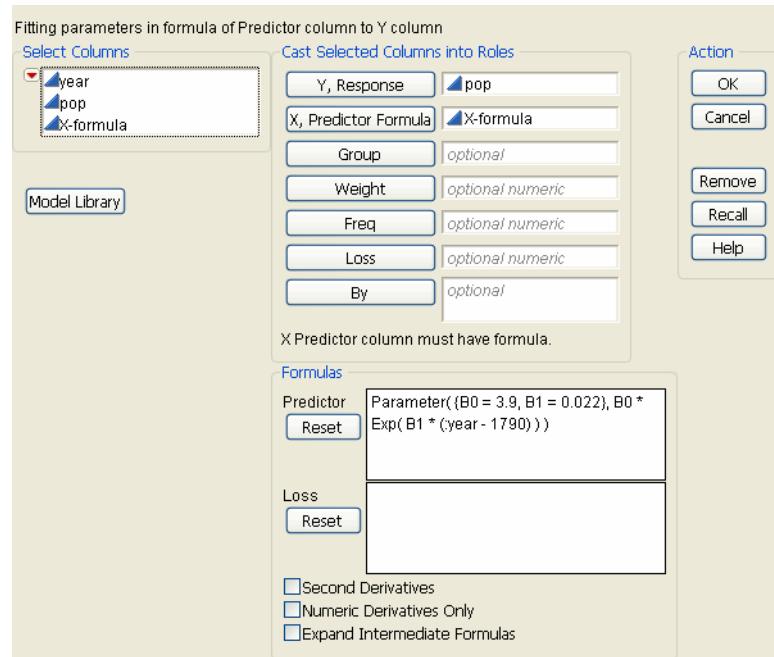
Enter the formula by clicking on column names, parameter names, operator keys, and function list items in the formula editor panel. Alternatively, double-click on the formula in the formula editor to change it into text-edit mode. The formula can be typed in as text.

See the *JMP User Guide* for information about building formulas.

## Launch the Nonlinear Platform

When the formula is complete, choose **Analyze > Modeling > Nonlinear** and complete the Launch dialog as shown in Figure 12.1. Select the pop as **Y, Response**, and the column with the fitting formula (X-formula) as **X, Predictor Formula**.

**Figure 12.1** The Nonlinear Launch Dialog



The model formula (and formula for a loss function if there is one) appear in text form in the **Formulas** area at the bottom of the Nonlinear Launch dialog (see Figure 12.1).

A **Y** variable is not needed if the fitting formula calculates residuals instead of predicted values or if there is a custom loss function. In some cases with loss functions, even the **X, Predictor Formula** column (the model) is not necessary.

**Note:** You can also select **Weight**, **Freq**, **By**, and **Group** columns. If the **Weight** column contains a formula, it is recalculated for each iteration. This also causes the column with the predicted value to update at each iteration. If you specify a **Group** variable, then when you do the plot, JMP overlays a curve for each value of the **Group** column. This assumes that the model is only a function of the **Group** column and one other continuous column.

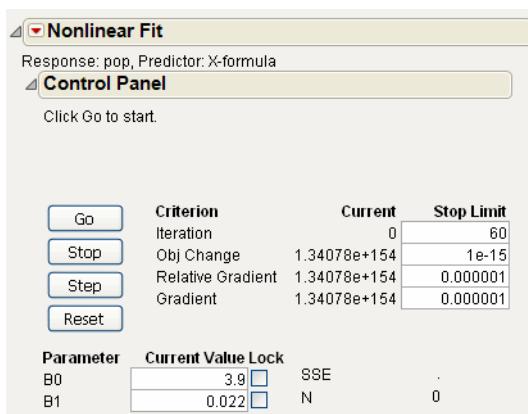
The **Second Derivatives** option uses second derivatives as well as first derivatives in the iterative method to find a solution. With second derivatives, the method is called Newton-Raphson rather than Gauss-Newton. This method is only useful if the residuals are unusually large or if you specify a custom loss function and your model is not linear in its parameters.

The **Numeric Derivatives Only** option is useful when you have a model that is too messy to take analytic derivatives for. It can also be valuable in obtaining convergence in tough cases.

The **Expand Intermediate Formulas** checkbox tells JMP that if an ingredient column to the formula is a column that itself has a formula, to substitute the inner formula, as long as it refers to other columns. To prevent an ingredient column from expanding, use the **Other** column property with a name of "Expand Formula" and a value of 0.

## Drive the Iteration Control Panel

When you complete the Launch dialog, click **OK** to see the Iteration Control Panel. The Control Panel lets you tailor the nonlinear fitting process. The parameters' initial values from the formula construction show in the Control Panel. Edit the **Stop Limit** fields if you need more or less accuracy for the converged solutions. If you edit the **Current Values** of the parameters, click **Reset** to set them for the next fit. Click **Go** to start the fitting iterations. You can watch the iterations progress in the Control Panel.



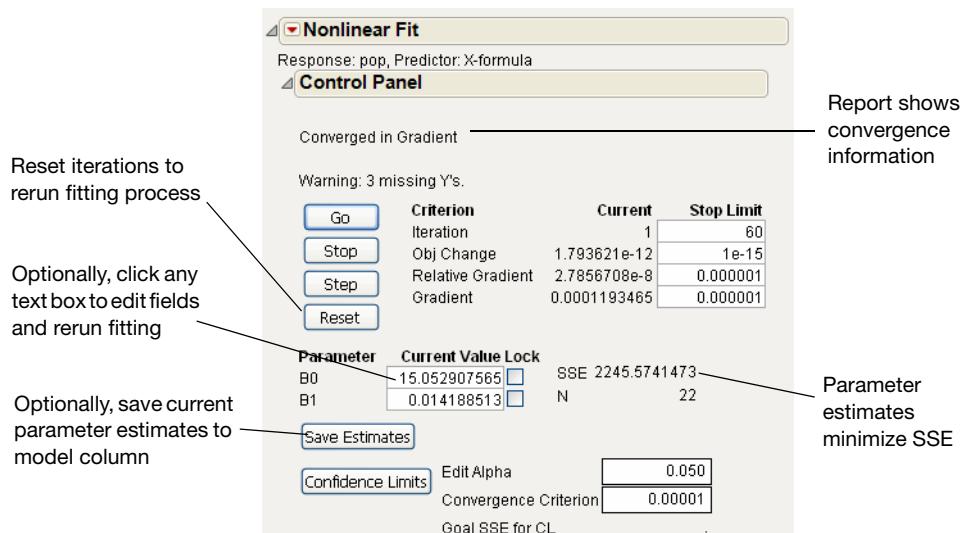
If the iterations do not converge, then try other parameter starting values; each time you click **Reset** and rerun the fit, the sum of squares error is recalculated, so it is easy to try many values. Sometimes locking

some parameters and iterating on the rest can be helpful. If the model is a function of only one column, then use the sliders on the output **Plot** to visually modify the curve to fit better.

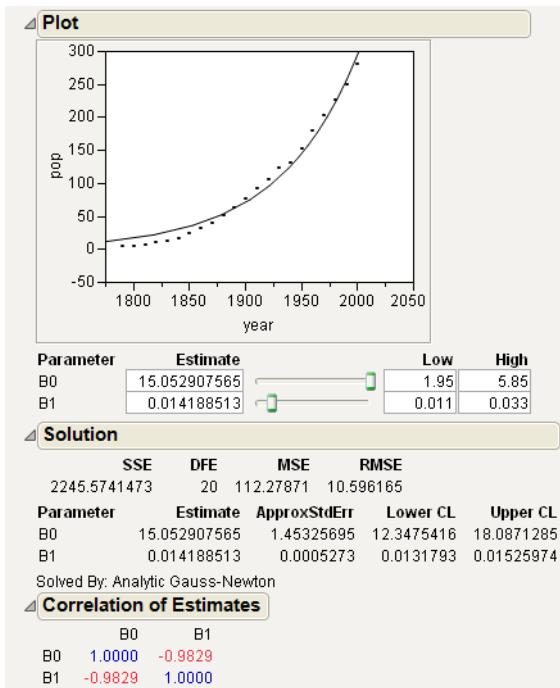
If the iterations converge, then a solution report appears and additional fields appear in the Control Panel, as shown in Figure 12.2. Click **Confidence Limits**, which now shows at the bottom of the Iteration Control Panel, to calculate profile-likelihood confidence intervals on the parameter estimates. These confidence limits show in the Solution table (see Figure 12.3). The confidence limit calculations involve a new set of iterations for each limit of each parameter, and the iterations often do not find the limits successfully.

When there is a single factor, the results include a graph of the response by the factor as shown in Figure 12.3. The solution estimates for the parameters are listed beneath the plot with sliders to alter the values and modify the fitted curve accordingly.

**Figure 12.2** Nonlinear Iteration Control Panel After Fitting Process Is Complete

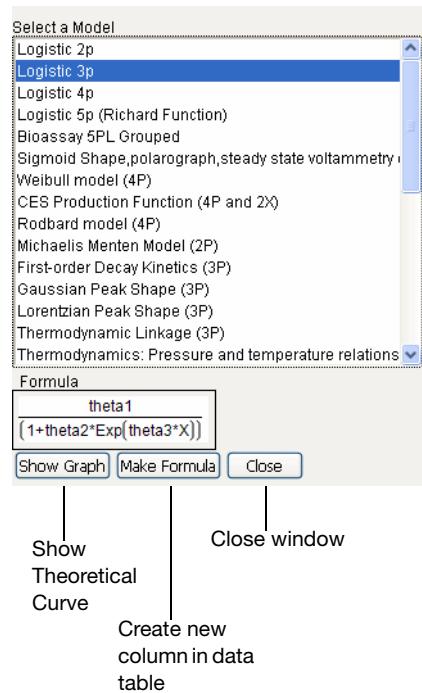


Click the **Confidence Limits** button to generate confidence intervals for the parameters.

**Figure 12.3** Results for Exponential Population Growth Example

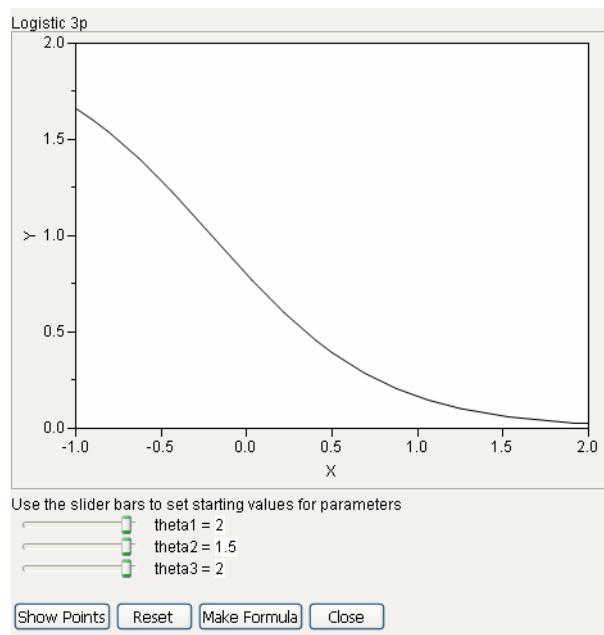
## Using the Model Library

The built-in model library is invoked by clicking the **Model Library** button on the Nonlinear launch dialog. Once it is invoked, a window called Nonlinear Model Library appears.

**Figure 12.4** Nonlinear Model Library Dialog

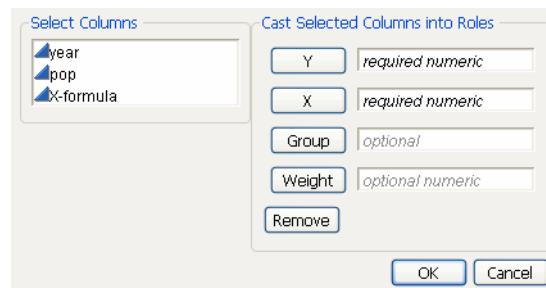
There are about 33 models included in this library. Users can select one of the models and click either **Show Graph** or **Make Formula**.

Click **Show Graph** to show a 2-D theoretical curve for one-parameter models and a 3-D surface plot for two-parameter models. No graph is available for models with more than two explanatory ( $X$ ) variables. On the graph window, change the default initial values of parameters using the slider, or clicking and typing values in directly (see Figure 12.5).

**Figure 12.5** Example Graph in Model Library

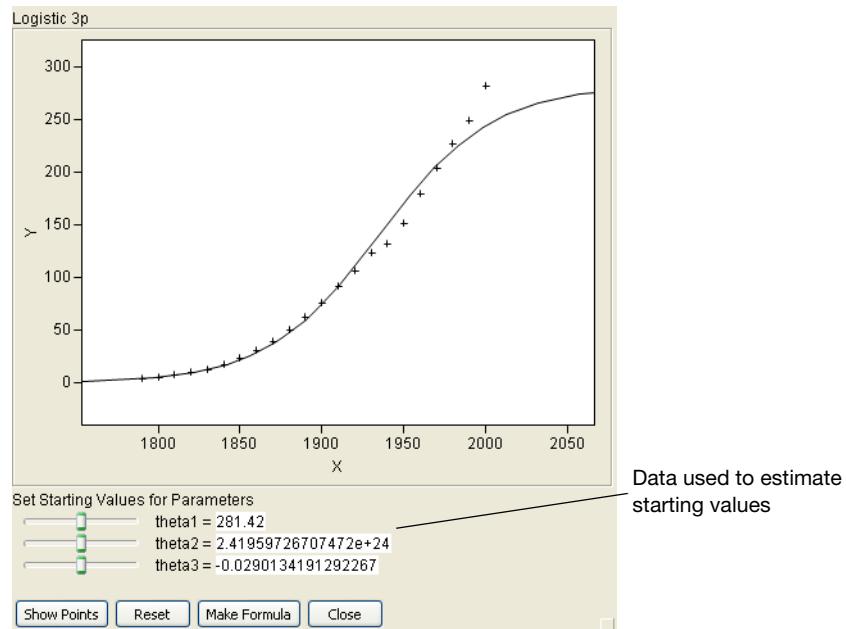
The **Reset** button sets the initial values of parameters back to their default values.

Click **Show Points** to overlay the actual data points to the plot. The dialog in Figure 12.6 opens asking you to assign columns into *X* and *Y* roles, and an optional *Group* role. The *Group* role allows for fitting the model to every level of a categorical variable. If you specify a *Group* role here, also specify the *Group* column on the platform launch window (Figure 12.1).

**Figure 12.6** Roles

The platform uses the actual data to estimate starting values for the parameters. Showing points allows you to adjust the parameter values to see how well the model fits for different values of the parameters. For the US population example, the points are shown in Figure 12.7.

**Figure 12.7** Show Points



Clicking **Make Formula** at this point (after using **Show Points**) creates a new column in the data table which has the formula as a function of the latest parameter starting values.

**Note 1:** If you click **Make Formula** before using the **Show Points** button, you are asked to provide the *X* and *Y* roles, and an optional Group role (see Figure 12.6), and then are brought back to the plot so you have the opportunity to adjust the parameters starting values if desired. At that point click **Make Formula** again to create the new column.

**Note 2:** If you click **Make Formula** from the Nonlinear Model Library window (Figure 12.4) without using the **Show Graph** button, you are first asked to assign *X* and *Y* roles, and an optional Group role (see Figure 12.6). Then you are brought to the plot so you have the opportunity to adjust the parameters starting values. At that point click **Make Formula** again to create the new column.

Once the formula is created in the data table, continue the analysis by assigning the new column as the **X, Predictor Formula** in the Nonlinear launch dialog.

Many of the models in the model library are named Model A, Model B, etc., whose definitions are shown in Table 12.2 “Guide to Nonlinear Modeling Templates,” p. 264.

## Customizing the Nonlinear Model Library

The Model Library is created by a built-in script named `NonlinLib.jsl`, located in the `Resources/Builtin Scripts` folder in the folder that contains JMP (on Windows) or in the Application Package (on Macintosh). You can customize the nonlinear library script by modifying this script.

To add a model, you must add three lines to the list named `Listofmodellist#`. These three lines are actually a list themselves which consists of the following three parts.

- Model name, a quoted string
- Model formula, an expression
- Model scale

For example, suppose you want to add a model called “Simple Exponential Growth” that has the form

$$y = b_1 e^{kx}$$

Add the following lines to the `NonlinLib.jsl` script

```
{//Simple Exponential Growth
  "Simple Exponential Growth",
  Expr(Parameter({b1=2, k=0.5}, b1*exp(k * :X))),
  lowx = -1; highx = 2; lowy = 0; highy = 2},
```

Some things to note:

- The first line is simply an open bracket (starting the list) and an optional comment. The second line is the string that is displayed in the model library window.
- The values of `lowx`, `highx`, `lowy`, and `highy` specify the initial window for the theoretical graph.
- There is a comma as the last character in the example above. If this is the final entry in the `Listofmodellist#` list, the comma can be omitted.
- If the model uses more than two parameters, replace the last line (containing the graph limits) with the quoted string “String Not Available”.

To delete a model, delete the corresponding three-lined list from the `Listofmodellist#` list.

## Details for the Formula Editor

In the formula editor, when you add a parameter, note the checkbox for **Expand Into Categories, selecting column**. This command is used to add several parameters (one for each level of a categorical variable for example) at once. When you select this option, a dialog appears that allows you to select a column. After selection, a new parameter appears in the Parameters list with the name `D_column`, where `D` is the name you gave the parameter. This is, in a sense, a macro that inserts a Match expression containing a separate parameter for each level of the selected column.

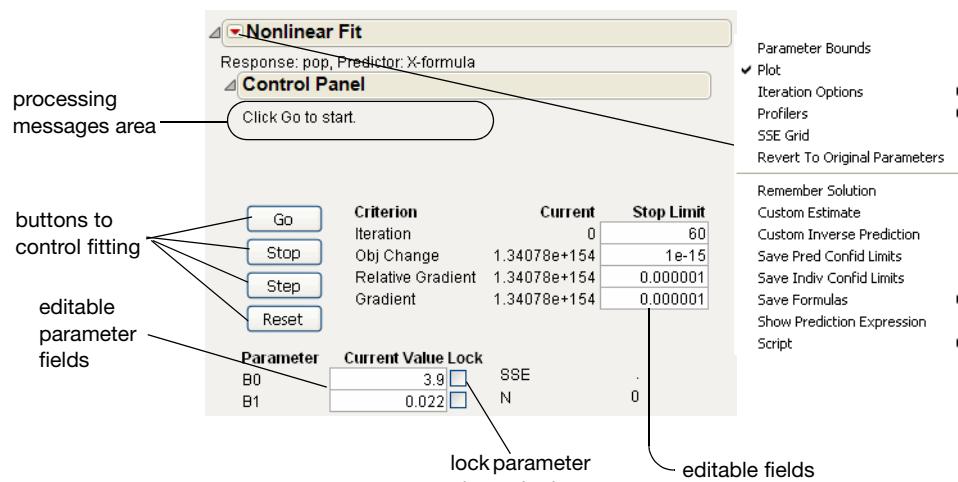
Once you decide where the list of parameters should be inserted in the formula, click the new parameter name to insert the group of parameters into the prediction formula.

## Details of the Iteration Control Panel

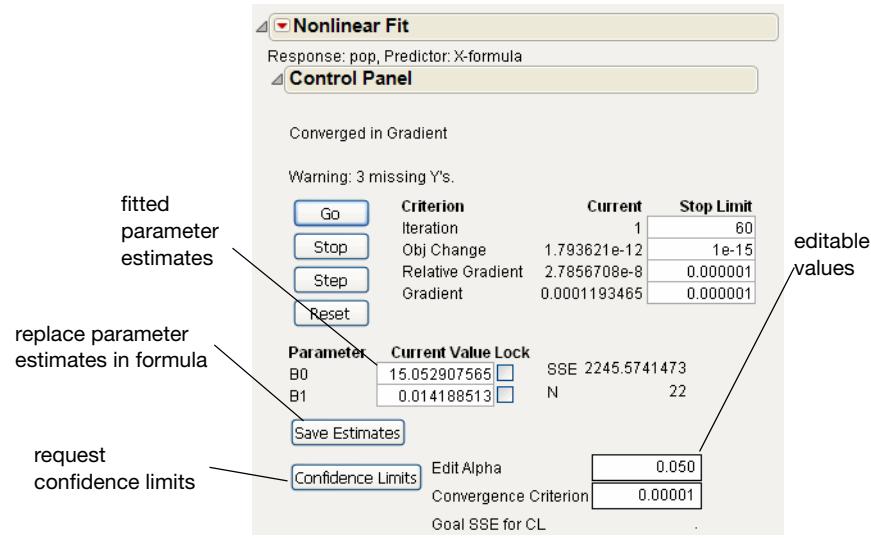
The Nonlinear platform appears with the Iteration Control Panel shown in Figure 12.8. The initial Control Panel has these features:

- a processing messages area, which shows the progress of the fitting processing
- editable parameter fields for starting values for the parameters. Click **Reset** after you edit these.
- buttons to start (**Go**), stop (**Stop**), and step (**Step**) through the fitting process
- a **Reset** button to reset the editable values into the formula, reset the iteration values, and calculate the SSE at these new values.
- initial values for convergence criteria and step counts
- a popup menu with fitting options to specify computational methods and save confidence limits. These popup menu items are described in the next section.

**Figure 12.8** Iteration Control Panel Before Fitting



After the fitting process converges, controls appear at the bottom of the Control Panel that let you specify the alpha level and convergence criterion for computation of confidence limits and predictions (Figure 12.9). When you click **Confidence Limits**, the results appear in the Solution table as shown previously in Figure 12.3.

**Figure 12.9** Control Panel After Fitting

The next sections describe the features of the iteration Control Panel in more detail.

## Panel Buttons

**Go** starts an iterative fitting process. The process runs in the background so that you can do other things while it runs. If you want to re-run the model, you must first click the **Reset** button. A message in the status bar reminds you to do so if you click **Go** multiple times.

**Stop** stops the iterative process. Because the platform only listens to events between steps, it might take a few seconds to get a response from **Stop**. Iterations can be restarted by clicking **Go** again.

**Step** takes a single iteration in the process.

**Reset** resets the iteration counts and convergence criteria, copies the editable values into the internal formulas, and then calculates the SSE for these new values. Use **Reset** if you want to

- take extra steps after the iterations have stopped. For example, extra steps are needed if you set the Stop Limit in order to solve for greater accuracy.
- edit the parameter estimates to clear the previous fitting information and begin iterations using these new starting values.
- calculate the SSE with the current parameter values.

## The Current Parameter Estimates

The Control Panel also shows the **Current Value** of the parameter estimates and the **SSE** values as they compute. At the start, these are the initial values given in the formula. Later, they are values that result from

the iterative process. The SSE is missing before iterations begin and is the first SSE displayed after the nonlinear fit.

If you want to change the parameter values and start or continue the fitting process, click and edit the parameters in the **Current Value** column of the parameter estimates. Then click **Reset**, followed by **Go**.

**Note:** Nonlinear makes its own copy of the column formula, so if you open the formula editor for the model column, the changes you make there will not affect an active Nonlinear platform. The parameter values are not stored back into the column formula until you click **Save Estimates**.

## Save Estimates

After the nonlinear fitting is complete, the **Save Estimates** button appears on the Control Panel. When you click **Save Estimates**, the parameter estimates are stored into the column formula, causing the column to be recalculated by the data table. If you are using By-groups, this does not work because the formula is in the source table, instead of the By-groups.

## Confidence Limits

The **Confidence Limits** button is available above the Solution Table after fitting. It computes confidence intervals for each parameter estimate. These intervals are profile-likelihood confidence limits, and each limit for each parameter involves a new set of iterations, so it can take a lot of time and result in missing values if the iterations fail to find solutions. The calculations do not run in the background but you can cancel them by typing the Escape key (⌘-period key on the Mac). See “[The Solution Table](#),” p. 248, for details about confidence limits.

## The Nonlinear Fit Popup Menu

The popup menu icon on the Nonlinear Fit title bar lets you tailor the fitting process, request additional information about the results, and save confidence limits as new columns in the current data table.

The following list describes each of the popup menu items:

### Parameter Bounds

To set bounds on the parameters, select the **Parameter Bounds** option to reveal new editable fields next to the parameter values, into which you can enter lower and upper bounds to the parameters. Unbounded parameters are signified by leaving the field as a missing value (.).

If you save a script, the bounds are saved in this manner:

```
Parameter Bounds(paramName(lower, upper) . . .)
```

### Plot

If the prediction formula is a function of exactly one other continuous variable and you specify the response separately as the *Y* variable, then **Plot** shows a plot of the function and the observed data (see Figure 12.3). If you specify a Group variable at launch, then a curve shows for each group. The initial estimates of the

parameters show beneath the plot, with sliders to adjust the values. You can use these sliders to adjust the parameters for and obtain an empirical fit of the prediction function and use the resulting parameter estimates and starting values for the nonlinear fitting process.

## Iteration Options

Iteration options let you specify options for the optimization algorithm.

**Iteration Log** opens an additional window (named **Iteration Log**) and records each step of the fitting process. The iteration log lets you see parameter estimates at each step and the objective criterion as they change at each step during the fitting process.

**Numeric Derivatives Only** is useful when you have a model that is too messy to take analytic derivatives for. It can also be valuable in obtaining convergence in tough cases.

**Expand Intermediate Formulas** tells JMP that if an ingredient column to the formula is a column that itself has a formula, to substitute the inner formula, as long as it refers to other columns. To prevent an ingredient column from expanding, use the **Other** column property with a name of "Expand Formula" and a value of 0.

**Newton** chooses whether Gauss-Newton (for regular least squares) or Newton-Raphson (for models with loss functions) as the optimization method.

**QuasiNewton SR1** chooses QuasiNewton SR1 as the optimization method.

**QuasiNewton BFGS** chooses QuasiNewton BFGS as the optimization method.

**Accept Current Estimates** tells JMP to produce the solution report with the current estimates, even if the estimates did not converge.

**Show Derivatives** shows the derivatives of the nonlinear formula in the JMP log. Use the **Log** command in the **View** or **Window** menu to see the JMP log if it is not currently open. See "[Notes Concerning Derivatives](#)," p. 261, for technical information about derivatives.

**Unthreaded** runs the iterations in the main computational thread. In most cases, JMP does the computations in a separate computational thread. This improves the responsiveness of JMP while doing other things during the nonlinear calculations. However, there are some isolated cases (models that have side effects that call display routines, for example) that should be run in the main thread, so this option should be turned on.

## Profilers

The Profilers submenu brings up three facilities for viewing the prediction surface. Details on profiling are discussed in "[Profiling Features in JMP](#)," p. 533.

**Profiler** brings up the JMP Profiler. The Profiler lets you view vertical slices of the surface across each  $x$ -variable in turn, as well as find optimal values of the factors. Details of the Profiler are discussed in "[The Profiler](#)," p. 89 in the "Standard Least Squares: Exploring the Prediction Equation" chapter.

**Contour Profiler** brings up the JMP Contour Profiler. The Contour profiler lets you see two-dimensional contours as well as three dimensional mesh plots. Details of the Contour Profiler are discussed in "[Contour Profiler](#)," p. 555 in the "Profiling" chapter.

**Surface Profiler** creates a three-dimensional surface plot. There must be two continuous factors for this facility to work. Details of surface plots are discussed in the "[Plotting Surfaces](#)" chapter.

## Details of the Iteration Control Panel

**Parameter Profiler** brings up the JMP Profiler and profiles the SSE or loss as a function of the parameters. Details of the Profiler are discussed in “[The Profiler](#),” p. 89 in the “Standard Least Squares: Exploring the Prediction Equation” chapter.

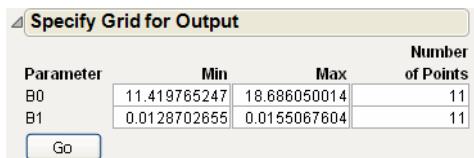
**Parameter Contour Profiler** brings up the JMP Contour Profiler and profiles the SSE or loss as a function of the parameters. Details of the Contour Profiler are discussed in “[Contour Profiler](#),” p. 555 in the “Profiling” chapter.

**Parameter Surface Profiler** creates a three-dimensional surface plot and profiles the SSE or loss as a function of the parameters. Details of surface plots are discussed in the “[Plotting Surfaces](#)” chapter.

### SSE Grid

To explore the sensitivity of the fit with respect to the parameter estimates, you can create a grid around the solution estimates and compute the error sum of squares for each value. The solution estimates should have the minimum SSE.

To create a grid table, use the **SSE Grid** command, which opens the Specify Grid for Output dialog, shown here. Edit the values in the dialog to customize the grid.




---

The Specify Grid for Output table shows these items:

**Parameter** lists the parameters in the fitting model (B0 and B1 in the figure above).

**Min** displays the minimum parameter value used in the grid calculations. By default, Min is the solution estimate minus 2.5 times the ApproxStdErr.

**Max** displays the maximum parameter value used in the grid calculations. By default, Max is the solution estimate plus 2.5 times the ApproxStdErr.

**Number of Points** gives the number of points to create for each parameter. To calculate the total number of points in the new grid table, multiply all the **Number of Points** values. Initially **Number of Points** is 11 for the first two parameters and 3 for the rest. If you specify new values, use odd values to ensure that the grid table includes the solution estimates. Setting **Number of Points** to 0 for any parameter records only the solution estimate in the grid table.

When you click **Go**, JMP creates a new untitled table with grid coordinates, like the table shown in Figure 12.10. A highlighted X marks the solution estimate row if the solution is in the table. You can use the **Scatterplot 3D** command in the **Graph** menu to explore the SSE or likelihood surface.

**Figure 12.10** Data Table with Grid Coordinates

	57	15.052907631	0.013133915	13147.779984
	58	15.052907631	0.0133975645	8682.6549886
	59	15.052907631	0.013661214	5249.9698633
	60	15.052907631	0.0139248635	3034.7016122
solution estimate	61	15.052907631	0.014188513	2245.5741473
	62	15.052907631	0.0144521624	3,117.837666
	63	15.052907631	0.0147158119	5916.3629317
	64	15.052907631	0.0149794814	10939.085746

### Revert to Original Parameters

The **Revert to Original Parameters** command resets the Nonlinear platform to use the originally-specified parameters (*i.e.* the parameters that existed before any fitting).

### Remember Solution

The **Remember Solution** command is used to stage a series of restricted estimates for use in constructing tests on the restrictions. This command is only used after you finish estimating for each model.

For example, suppose you want to test an unrestricted exponential model with a model where one parameter is constrained. First, set the parameter to its restricted value and check the **Lock** box for it. Then, click **Go** to obtain the restricted results. Click the **Remember Solution** command to remember this solution. Now, uncheck the lock and repeat the procedure, clicking **Reset**, then clicking **Go**, then **Remember Solution**. JMP then shows the fitting statistics for each solution, and a test comparing the solutions. Figure 12.11 shows an example using **US Population.jmp**, where the **b1** parameter is constrained to be 1 in the restricted model.

**Figure 12.11** Remembered Models

Model	SSE	DFE	MSE	Restrictions
Unrestricted Model	2245.5741	20	112.2787	
Restricted Model	277552.85	21	13216.8	B1=1

Hypothesized	Alternative	Denominator	SS	NDF	DDF	F Ratio	Prob > F
Restricted Model	Unrestricted Model	Unrestricted Model	275307.28	1	20	2451.999	<.0001*

Parameter	Unrestricted Model	Restricted Model
B0	15.052907631	1.768199e-89
B1	0.014188513	1

### Custom Estimate

The **Custom Estimate** command lets you provide an expression involving only parameters. JMP calculates the expression using the current estimates, and also calculates a standard error of the expression using a first-order Taylor series approximation.

## Custom Inverse Prediction

Given a  $y$ -value, the **Custom Inverse Prediction** command calculates the  $x$ -value yielding this  $y$ -value for the estimated model. It also calculates a standard error. JMP must be able to invert the model to obtain an inverse expression, and it can only do this if the model is a function of one variable mentioned one time in the expression. The standard error is based on the first-order Taylor series approximation using the inverted expression. The confidence interval uses a  $t$ -quantile with the standard error, and is a Wald interval.

## Save Pred Confid Limits and Save Indiv Confid Limits

The **Save** options create new columns in the current data table called LowerM, UpperM, LowerI, and UpperI that contain the estimated lower- and upper-95% confidence limits computed using the asymptotic linear approximation. **Save Pred Confid Limits** saves confidence limits of the nonlinear fit, i.e. the expected or fitted response, which involves the variances in the estimates, but does not involve the variance of the error term itself. **Save Indiv Confid Limits** saves confidence limit values for the individual predicted values, which adds the variance of the error term to the variance of prediction involving the estimates, to form the interval.

## Save Formulas

**Save Prediction Formula** Creates a new column containing the prediction model as its formula, with all the parameters replaced by their estimates. The values produced will be the same as in the Model column after you click the Save Estimates button. The resulting column will be called “Fitted name” where “name” is the name of the column specifying the model.

**Save Std Error of Predicted** Creates a new column containing the formula for the standard error of the prediction, with all the parameters replaced by their estimates. This standard error accounts for all the uncertainty in the parameter estimates, but does not add the uncertainty in predicting individual  $Y$ 's. The resulting column will be called StdError Fitted *name* where *name* is the name of the column specifying the model. The formula is of the form  $\text{Sqrt}(\text{VecQuadratic}(\text{matrix1}, \text{vector1}))$  where *matrix1* is the covariance matrix associated with the parameter estimates, and *vector1* is a composition of the partial derivatives of the model with respect to each parameter.

**Save Std Error of Individual** Creates a new column containing the formula for the standard error of the individual value, with all the parameters replaced by their estimates. This standard error accounts for the uncertainty in both the parameter estimates and in the error for the model. The resulting column will be called StdError Indiv *name* where *name* is the name of the column specifying the model. The formula is of the form  $\text{Sqrt}(\text{VecQuadratic}(\text{matrix1}, \text{vector1}) + \text{mse})$  where *matrix1* is the covariance matrix associated with the parameter estimates, *vector1* is a composition of the partial derivatives of the model with respect to each parameter, and *mse* is the estimate of error variance.

**Save Residual Formula** Creates a column containing a formula to calculate the residuals from the current model. The new column is named Residuals *name* where *name* is the name of the column specifying the model.

**Save Pred Confid Limit Formula** creates a column containing a formula to calculate the confidence interval for a prediction.

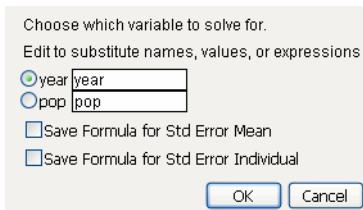
**Save Indiv Confid Limit Formula** creates a column containing a formula to calculate the confidence interval for an individual.

**Save Inverse Prediction Formula** Creates a column containing a formula to predict the  $x$  value given the  $y$  value. The resulting column is called `InvPred name` where `name` is the name of the predicted column. Two additional columns are created with formulas for the standard error of prediction and the standard error of individual, named `StdError InvPred name` and `StdErrIndiv InvPred name` respectively.

The internal implementation of this uses the same routine as the JSL operator `InvertExpr`, and has the same limitations, that the expression contain invertible expressions only for the expression path leading to the inversion target, that the target be in the formula only once, and that it use favorable assumptions to limit functions to an invertible region.

**Save Specific Solving Formula** is, in simple cases, equivalent to **Save Inverse Prediction Formula**. However, this command allows the formula to be a function of several variables and allows expressions to be substituted. This feature only works for solving easily invertible operators and functions that occur just once in the formula.

After selecting this command, a dialog appears that allows you to select the variable to solve for. Here, you can also edit the names of the columns in the resulting table.




---

You can also substitute values for the names in the dialog. In these cases, the formula is solved for those values.

---

**Note:** The standard errors, confidence intervals, and hypothesis tests are correct only if least squares estimation is done, or if maximum likelihood estimation is used with a proper negative log likelihood.

### Show Prediction Expression

The **Show Prediction Expression** command toggles the display of the fitting expression at the top of the report.

---

## Details of Solution Results

When you click **Go** on the iteration Control Panel, the fitting process begins. The iterations run in the background until the method either converges or fails to converge. During fitting, process messages on the Control Panel monitor the iterative process. When the iteration process completes, these messages show whether the convergence criteria are met.

If the convergence criteria are met, reports are appended to the iteration Control Panel. The **Solution** table and **Correlation of Estimates** table always appear. The graph of the fitted function is available only if the prediction formula is a function of one other variable (see Figure 12.3).

## The Solution Table

When the iterations are complete, the results display in the Solution table.

**SSE** shows the residual sum of squares error. SSE is the objective that is to be minimized. If a custom loss function is specified, this is the sum of the loss function.

**DFE** is the degrees of freedom for error, which is the number of observations used minus the number of parameters fitted.

**MSE** shows the mean squared error. It is the estimate of the variance of the residual error, which is the SSE divided by the DFE.

**RMSE** estimates the standard deviation of the residual error, which is square root of the MSE described above.

**Parameter** lists the names that you gave the parameters in the fitting formula.

**Estimate** lists the parameter estimates produced. Keep in mind that with nonlinear regression, there may be problems with this estimate even if everything seems to work.

**ApproxStdErr** lists the approximate standard error, which is computed analogously to linear regression. It is formed by the product of the RMSE and the square root of the diagonals of the derivative cross-products matrix inverse.

**Lower CL and Upper CL** are the lower and upper  $100(1 - \alpha)$  percent confidence limits for the parameters. They are missing until you click the **Confidence Limits** on the Control Panel.

The upper and lower confidence limits are based on a search for the value of each parameter after minimizing with respect to the other parameters, that produces a SSE greater by a certain amount than the solution's minimum SSE. The goal of this difference is based on the *F*-distribution. The intervals are sometimes called *likelihood confidence intervals* or *profile likelihood confidence intervals* (Bates and Watts 1988; Ratkowsky 1990).

## Excluded Points

Sometimes, you want to hold back data as a set for validation of the results. The Nonlinear platform now calculates statistics of fit for excluded data.

## Profile Confidence Limits

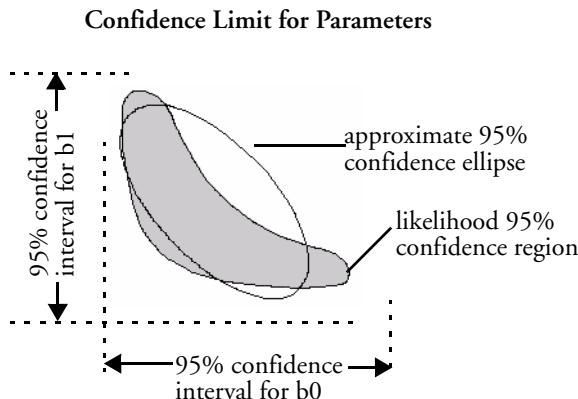
Profile confidence limits all start with a *goal SSE*, which is a sum of squared errors (or sum of loss function) that an F-test considers significantly different from the solution SSE at the given alpha level. If the loss function is specified to be a negative log-likelihood, then a Chi-square quantile is used instead of an *F* quantile. For each parameter's upper confidence limit, the parameter value is moved up until the SSE reaches the goal SSE, but as the parameter value is moved up, all the other parameters are adjusted to be least squares estimates subject to the change in the profiled parameter. Conceptually, this is a compounded

set of nested iterations, but internally there is a way to do this with one set of iterations developed by Johnston and DeLong (see SAS/Stat 9.1 vol. 3 pp. 1666-1667).

The diagram in Figure 12.12, shows the contour of the goal SSE or negative likelihood, with the least squares (or least loss) solution inside the shaded region:

- The asymptotic standard errors produce confidence intervals that approximate the region with an ellipsoid and take the parameter values at the extremes (at the horizontal and vertical tangents).
- Profile confidence limits find the parameter values at the extremes of the true region, rather than the approximating ellipsoid.

**Figure 12.12** Diagram of Confidence Region



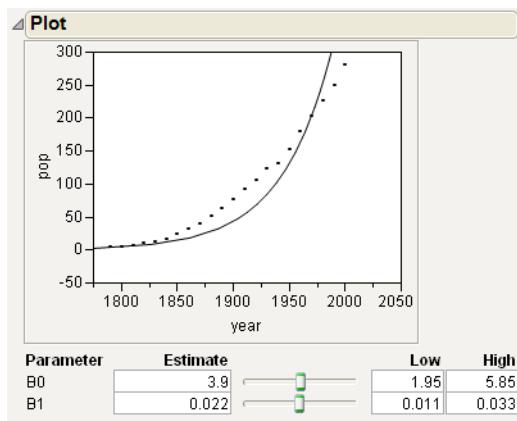
Likelihood confidence intervals are more trustworthy than confidence intervals calculated from approximate standard errors. If a particular limit cannot be found, computations begin for the next limit. When you have difficulty obtaining convergence, try the following:

- use a larger alpha, resulting in a shorter interval, more likely to be better behaved
- use the option for second derivatives
- relax the confidence limit criteria.

## Fitted Function Graph

If the prediction formula is a function of exactly one other variable and you specify the response separately as the *Y* variable, then the **Plot** disclosure button displays a plot of the function and the observed data as shown here.

The sliders to the right of the estimates beneath the graph can be used to experiment with the parameter values and observe the effect on the curve. As you change parameter values, the graph changes accordingly.



## Chemical Kinetics Example

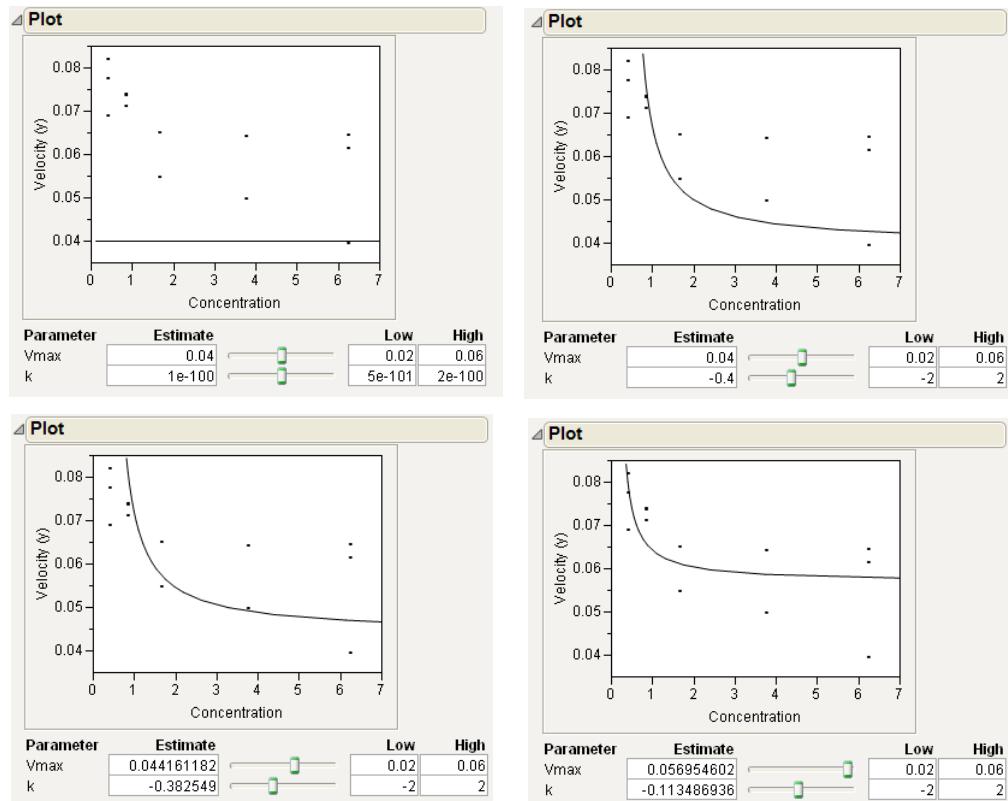
In pharmacology, the relationship between the concentration of an available dissolved organic substrate and the rate of uptake of the substrate is often modeled by the Michaelis-Menten equation, of the form

$$Y = \frac{V_{\max}x}{K + x}$$

where  $Y$  is the velocity of uptake and  $x$  is the concentration. The parameters  $V_{\max}$  and  $K$  estimate the maximum velocity and transport constant, respectively. The Chemical Kinetics.jmp data table (found in the Nonlinear Examples folder of the sample data) records the uptake velocity of a glucose-type substrate in an incubated sediment sample. The data table stores the Michaelis-Menten model with starting parameter values of  $V_{\max}=0.04$  and  $K=0$ .

1. Select **Nonlinear** from the **Analyze > Modeling** menu and assign Model (x) as the **X, Predictor Formula** and Velocity (y) as **Y, Response**. Click **OK** and when the Control Panel appears. The Plot shows the model at the current starting values.
2. Next, adjust the **Low** and **High** values of the k slider to be -2 to 2. Use the slider to change the value of K slightly, so you can get a better starting fit.
3. Click **Go** to continue the iteration steps until convergence.

Figure 12.13 illustrates the fitting process described above.

**Figure 12.13** Example of Plot Option to Observe Nonlinear Fitting Process

The **SSE Grid** option in the Nonlinear Fit popup menu helps you investigate the error sum of squares around the parameter estimates. After you select the **SSE Grid** option, click **Go** in the Specify Grid for Output report to create a new table that gives the SSE for each combination of the parameters. In the data table, a highlighted x marks the solution that the nonlinear platform found. Alternatively, you can view the SSE surface using the Surface Plot platform.

## How Custom Loss Functions Work

The nonlinear facility can minimize or maximize functions other than the default sum of squares residual. This section shows the mathematics of how it is done.

Suppose that  $f(\beta)$  is the model. Then the Nonlinear platform attempts to minimize the sum of the loss functions written as

$$L = \sum_{i=1}^n \rho(f(\beta))$$

The loss function  $\rho(\bullet)$  for each row can be a function of other variables in the data table. It must have non-zero first- and second-order derivatives. The default  $\rho(\bullet)$  function, squared-residuals, is

$$\rho(f(\beta)) = (y - f(\beta))^2$$

To specify a model with a custom loss function, construct a variable in the data table and build the loss function. After launching the Nonlinear platform, select the column containing the loss function as the loss variable.

The nonlinear minimization formula works by taking the first two derivatives of  $\rho(\bullet)$  with respect to the model, and forming the gradient and an approximate Hessian as follows:

$$\begin{aligned} L &= \sum_{i=1}^n \rho(f(\beta)) \\ \frac{\partial L}{\partial \beta_j} &= \sum_{i=1}^n \frac{\partial \rho(f(\beta))}{\partial f} \frac{\partial f}{\partial \beta_j} \\ \frac{\partial^2 L}{\partial \beta_j \partial \beta_k} &= \sum_{i=1}^n \left[ \frac{\partial^2 \rho(f(\beta))}{(\partial f)^2} \frac{\partial f}{\partial \beta_j} \frac{\partial f}{\partial \beta_k} + \frac{\partial \rho(f(\beta))}{\partial f} \frac{\partial^2 f}{\partial \beta_k \partial \beta_j} \right] \end{aligned}$$

If  $f(\bullet)$  is linear in the parameters, the second term in the last equation is zero. If not, you can still hope that its sum is small relative to the first term, and use

$$\frac{\partial^2 L}{\partial \beta_j \partial \beta_k} \approx \sum_{i=1}^n \frac{\partial^2 \rho(f(\beta))}{(\partial f)^2} \frac{\partial f}{\partial \beta_j} \frac{\partial f}{\partial \beta_k}$$

The second term will probably be small if  $\rho$  is the squared residual because the sum of residuals is small—zero if there is an intercept term. For least squares, this is the term that distinguishes Gauss-Newton from Newton-Raphson. In JMP, the second term is calculated only if the option **Second Deriv. Method** is checked.

---

**Note:** The standard errors, confidence intervals, and hypothesis tests are correct only if least squares estimation is done, or if maximum likelihood estimation is used with a proper negative log likelihood.

---

### Parameters in the Loss Function

You can use parameters in loss functions but they might not be the same as the parameters in the model. JMP uses first and second derivatives for these parameters, which means it takes full Newton-Raphson steps for these parameters.

## Maximum Likelihood Example: Logistic Regression

In this example, we show several variations of minimizing a loss function, in which the loss function is the negative of a log-likelihood function, thus producing maximum likelihood estimates.

The Logistic w Loss.jmp data table in the Nonlinear Examples sample data folder has an example for fitting a logistic regression using a loss function. The Y column is the proportion of responses (proportion number of 1s) for equal-sized samples of  $x$  values. The Model Y column has the linear model formula. However, the Loss column has the nonlinear formula shown below. It is the negative log-likelihood for each observation, which is the negative log of the probability of getting the response that you did get.

Model Y

$$b_0 + b_1 * X$$

Loss

$$\text{If } MZ \begin{cases} Y == 1 \Rightarrow -\log \left( \frac{1}{1 + \exp(Model\ Y)} \right) \\ \text{else} \Rightarrow -\log \left( \frac{1}{1 - \exp(Model\ Y)} \right) \end{cases}$$

To run the model, choose **Analyze >Modeling > Nonlinear** and complete the Launch dialog by using Model Y as the **X, Predictor Formula**, and Loss as the **Loss** function. Click **OK** to see the Iteration Control Panel. The initial values of zero (or 1e-100, very close to zero) for the parameters  $b_0$  and  $b_1$  are satisfactory.

Click **Go** to run the model. The results show in the Solution table with the negative log-likelihood as the Loss.

Solution			
		Sqrt	
Loss	DFE	Avg Loss	Avg Loss
7.8070527905	18	0.4337252	0.6595781
Parameter	Estimate	ApproxStdErr	
b0	5.5114510517	2.43728165	
b1	-0.034755348	0.01595111	
Solved By: Analytic NR			

The same problem can be handled differently by defining a model column formula that absorbs the logistic function and a loss function that uses the model to form the probability for a categorical response level. Model2 Y holds the model. The loss function is Loss2.

Model2 Y

$$\frac{1}{1 + \text{Exp}(\mathbf{b}_0 + \mathbf{b}_1 * X)}$$

Loss2

$$\begin{aligned} \text{If } M2 \\ Y == 1 \Rightarrow \text{Log}(\text{Model2 } Y) \\ \text{else } \Rightarrow \text{Log}(1 - \text{Model2 } Y) \end{aligned}$$

The loss function is used in this example, so the second derivative as well as the first one is calculated for the optimization. With least squares, the second derivatives are multiplied by residuals, which are usually near zero. For custom loss functions, second derivatives can play a stronger role.

Select **Nonlinear** again and complete the role assignment dialog with Model2 Y as **X**, **Predictor Formula**, and Loss2 as the loss function. Check the **Second Derivatives** check box on the Launch dialog and click **OK**. Then run the model again. The Solution table shows the same log-likelihood values and same parameter estimates as before.

## Iteratively Reweighted Least Squares Example

Iteratively Reweighted Least Squares (IRLS) is used for robust regression (Holland and Welsch 1977) and to fit maximum likelihoods for many models (McCullagh and Nelder 1983). The Nonlinear platform can use a weight column with a formula and repeatedly reevaluate the formula if it is a function of the model. These methods can serve as automatic outlier down-weighters because, depending on the weight function, large residual values can lead to very small weights. For example, Beaton and Tukey (1974) suggest minimizing the function

$$S_{\text{biweight}} = \sum_{i=1}^n \rho(r_i)$$

$$\rho(r_i) = \begin{cases} \left(\frac{B^2}{2}\right)\left(1 - \left(1 - \left(\frac{r_i}{B}\right)^2\right)^2 & |r_i| \leq B \\ \frac{B^2}{2} & |r_i| > B \end{cases}, \text{ where}$$

$$r = \frac{|R|}{\sigma}$$

$R$  is the residual

$\sigma$  is measure of the error, such as  $1.5 \times \text{med}(|\text{LS residuals}|)$ , (Myers, 1989),

$B$  = a tuning constant, usually between 1 and 3.

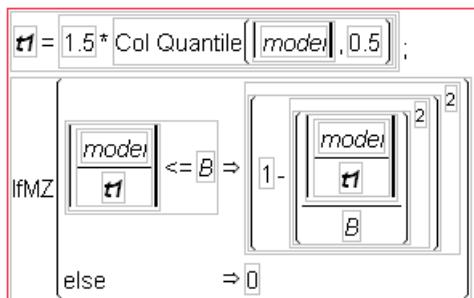
The IRLS Example.jmp sample data table has columns constructed to illustrate an IRLS process.

IRLS Example		pop	year	model	B	weight	weight2
Notes	Iteratively Reweighted Least Squares Example	1	3.93	1790	-3205887.1	1.9	0.8117
Columns (6/0)		2	5.31	1800	-3241795.7	1.9	0.8077
pop		3	7.24	1810	-3277903.8	1.9	0.8036
year		4	9.64	1820	-3314211.4	1.9	0.7995
model		5	12.87	1830	-3350718.1	1.9	0.7953
B		6	17.07	1840	-3387423.9	1.9	0.7910
weight		7	23.19	1850	-3424327.8	1.9	0.7867
weight2		8	31.44	1860	-3461429.6	1.9	0.7824
		9	39.82	1870	-3498731.2	1.9	0.7779
		10	50.16	1880	-3536230.8	1.9	0.7735
All rows	24	11	62.95	1890	-3573928.1	1.9	0.7689
Selected	0	12	75.99	1900	-3611825	1.9	0.7643
Excluded	0	13	91.97	1910	-3649919	1.9	0.7597
Hidden	0	14	105.71	1920	-3688215.3	1.9	0.7550
Labelled	0	15	122.78	1930	-3726708.2	1.9	0.7502
		16	131.67	1940	-3765409.3	1.9	0.7454
		17	151.33	1950	-3804299.7	1.9	0.7405

The IRLS Example table columns use the following formulas:

- the model column (model), with a model formula that uses three parameters b0, b1, and b2  

$$\text{pop} = (\text{b0} + \text{b1} * \text{year} + \text{b2} * \text{year} * \text{year})$$
- B, a tuning constant
- the weight column (weight), with a formula that uses model, B, and a local variable t1, a measure of error:



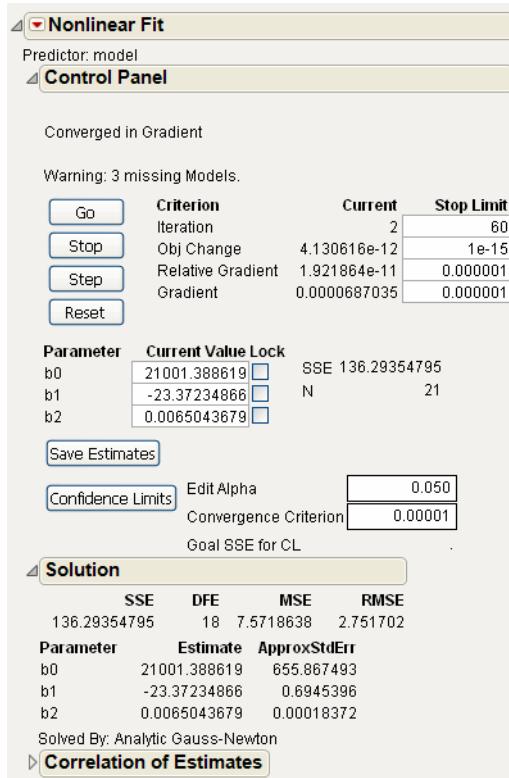
Myers (1989) suggests limiting the tuning constant to a range of 1 to 3. If you're having trouble getting your model to converge, try moving the tuning constant closer to 2. For this example, B (the tuning factor) is equal to 1.9. Myers also proposes using a measure of error computed as  $1.5 * \text{med}(|\text{LS residuals}|)$ . In this example, the formula includes a JSL statement that defines the local variable t1 as  $1.5 * \text{quantiles}_{0.5}(|\text{model}|)$ .

**Note:** The variable that specifies the model must appear in the weight formula so that weights are updated with each iteration. If the actual formula is used, JMP treats the parameters as different sets of parameters and doesn't update them.

To complete the IRLS example, follow these steps:

## Iteratively Reweighted Least Squares Example

1. (Simple Least Squares Step) Select **Analyze >Modeling > Nonlinear**. Choose model as the **X, Predictor Formula** column but don't select a weight yet. The parameter estimates from this least squares analysis provide starting values for the weighted model. Click **OK** on the Launch dialog and then **Go** when the Control Panel appears. You should get the starting values shown here.



At the end of the iterations, click **Save Estimates** on the Control Panel to set the b0, b1, and b2 parameters to these estimates, which are needed for the IRLS step.

2. (IRLS Step) Again, select **Nonlinear**. Assign model as the predictor formula column and weight as the **Weight** variable. Click **OK** to see the Control Panel and then click **Go** to get the final parameter estimates shown in Figure 12.14. The SSE is reduced from 136.2935 for the least squares analysis to 3.0488 for the IRLS analysis.

IRLS gives small weights to extreme values, which tend to reject them from the analysis. Years 1940 and 1950 are easy to identify as outliers because their weights are very small. IRLS reduces the impact of overly influential data points and can thereby better model the rest of the data. Click **Save Estimates** again to see the final data table shown in Figure 12.15.

**Figure 12.14** Final IRLS Solution Table

Solution			
SSE	DFE	MSE	RMSE
3.048855735	13	0.2345274	0.4842803
<b>Parameter</b>	<b>Estimate</b>	<b>ApproxStdErr</b>	
b0	21417.270135	185.573947	
b1	-23.82667788	0.17532257	
b2	0.0066285001	4.6379e-5	

Solved By: Analytic Gauss-Newton

Correlation of Estimates

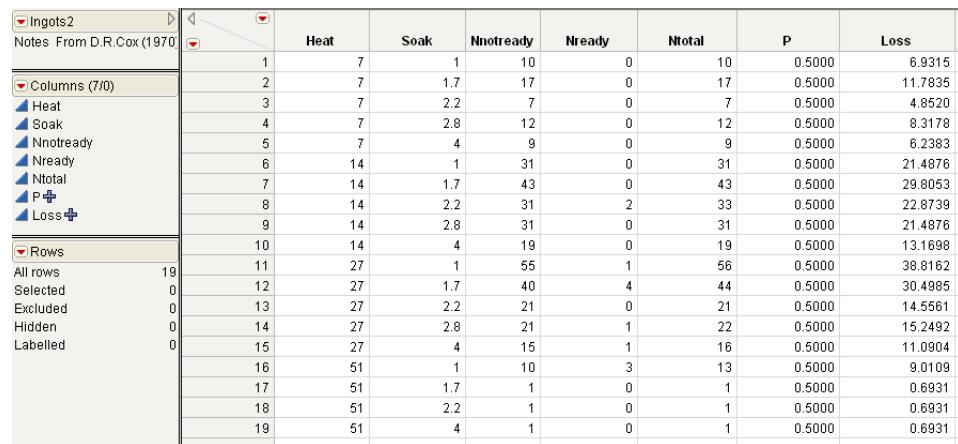
**Figure 12.15** Final Data Table

IRLS Example						
Notes Iteratively Reweigh:						
	pop	year	model	B	weight	weight2
1	3.93	1790	-1.9647773	1.9	0.1232	0.0000
2	5.31	1800	-0.2821508	1.9	0.9734	0.9507
3	7.24	1810	0.62677577	1.9	0.8723	0.7689
4	9.64	1820	0.67800229	1.9	0.8514	0.7326
5	12.87	1830	0.2325288	1.9	0.9819	0.9664
6	17.07	1840	-0.5636447	1.9	0.8960	0.8108
7	23.19	1850	-0.7665182	1.9	0.8122	0.6656
8	31.44	1860	-0.1650917	1.9	0.9909	0.9830
9	39.82	1870	-0.7663653	1.9	0.8123	0.6658
10	50.16	1880	-0.7313388	1.9	0.8282	0.6929
11	62.95	1890	0.43298759	1.9	0.9380	0.8859
12	75.99	1900	0.52661401	1.9	0.9089	0.8337
13	91.97	1910	2.22554042	1.9	0.0280	0.0000
14	105.71	1920	0.35876681	1.9	0.9572	0.9210
15	122.78	1930	0.49329319	1.9	0.9199	0.8533
16	131.67	1940	-8.8688804	1.9	0.0000	0.0000
17	151.33	1950	-8.7947541	1.9	0.0000	0.0000
18	179.32	1960	-1.7043277	1.9	0.2618	0.0080
All rows	24					
Selected	0					
Excluded	0					
Hidden	0					
Labelled	0					
	22	*	2000	*	1.9	0.0000
	23	*	2010	*	1.9	0.0000
	24	*	2020	*	1.9	0.0000

## Probit Model with Binomial Errors: Numerical Derivatives

The Ingots2.jmp file in the Sample Data folder records the numbers of ingots tested for readiness after different treatments of heating and soaking times.

## Probit Model with Binomial Errors: Numerical Derivatives



	Heat	Soak	Nnotready	Nready	Ntotal	P	Loss
1	7	1	10	0	10	0.5000	6.9315
2	7	1.7	17	0	17	0.5000	11.7835
3	7	2.2	7	0	7	0.5000	4.8520
4	7	2.8	12	0	12	0.5000	8.3178
5	7	4	9	0	9	0.5000	6.2383
6	14	1	31	0	31	0.5000	21.4876
7	14	1.7	43	0	43	0.5000	29.8053
8	14	2.2	31	2	33	0.5000	22.8739
9	14	2.8	31	0	31	0.5000	21.4876
10	14	4	19	0	19	0.5000	13.1698
11	27	1	55	1	56	0.5000	38.8162
12	27	1.7	40	4	44	0.5000	30.4985
13	27	2.2	21	0	21	0.5000	14.5561
14	27	2.8	21	1	22	0.5000	15.2492
15	27	4	15	1	16	0.5000	11.0904
16	51	1	10	3	13	0.5000	9.0109
17	51	1.7	1	0	1	0.5000	0.6931
18	51	2.2	1	0	1	0.5000	0.6931
19	51	4	1	0	1	0.5000	0.6931

The response variable, **NReady**, is binomial, depending on the number of ingots tested (**Ntotal**) and the heating and soaking times. Maximum likelihood estimates for parameters from a probit model with binomial errors are obtained using

- numerical derivatives
- the negative log-likelihood as a loss function
- the Newton-Raphson method.

The average number of ingots ready is the product of the number tested and the probability that an ingot is ready for use given the amount of time it was heated and soaked. Using a probit model, the fitting equation for the model variable, **P**, is

$$\text{Normal Distribution}[b_0 + b_1 * \text{Heat} + b_2 * \text{Soak}]$$

The argument to the **Normal Distribution** function is a linear model of the treatments.

To specify binomial errors, the loss function, **Loss**, has the formula

$$-[Nready * \text{Log}[p] + [Ntotal - Nready] * \text{Log}[1 - p]]$$

To complete the Nonlinear Launch dialog, specify **P** as the **X, Predictor Formula** and **Loss** as **Loss**. Click **Second Derivatives** on the Launch dialog. Click **OK**, then click **Go** on the Control Panel.

Default starting values of (near) zero for the parameters should result in the solution shown in Figure 12.16. JMP used the Newton-Raphson, or Second Derivatives method, to obtain the solution.

**Figure 12.16** Solution for the Ingots2 Data Using the Second Derivatives Method

The screenshot shows the JMP Nonlinear Fit interface with the following details:

- Control Panel:**
  - Converged in Gradient
  - Buttons: Go, Stop, Step, Reset.
  - Checkboxes: Loss is Neg LogLikelihood (checked).
- Parameter Estimates:**

Parameter	Current Value	Lock	Value	Lock
b0	-2.893415259	<input type="checkbox"/>	Loss	47.479945327
b1	0.0399554551	<input type="checkbox"/>	N	19
b2	0.0362537787	<input type="checkbox"/>		
- Save Estimates:** Buttons for Save Estimates, Confidence Limits, Edit Alpha (0.050), Convergence Criterion (0.00001), and Goal SSE for CL.
- Solution:**

Loss	DFE	Avg Loss	Avg Loss
47.479945327	16	2.9874986	1.7226423

Parameter	Estimate	ApproxStdErr
b0	-2.893415259	0.51255412
b1	0.0399554551	0.01202293
b2	0.0362537787	0.15016776

Solved By: Analytic NR
- Correlation of Estimates:** A section showing the correlation matrix of the parameter estimates.

## Poisson Loss Function

A Poisson distribution is often used as a model to fit frequency counts.

$$P(Y = n) = \frac{e^{-\mu} \mu^n}{n!}, \quad n = 0, 1, 2, \dots$$

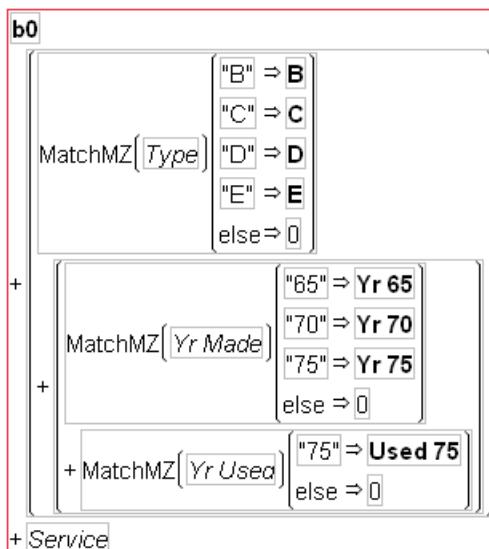
where  $\mu$  can be a single parameter, or a linear model with many parameters. Many texts and papers show how the model can be transformed and fit with iteratively reweighted least squares (Nelder and Wedderburn 1972). However, in JMP it is more straightforward to fit the model directly. For example, McCullagh and Nelder (1989) show how to analyze the number of reported damage incidents caused by waves to cargo-carrying vessels. The data are in the **Ship Damage.jmp** sample data, and includes columns called **model** and **Poisson**.

The formula for the Poisson column is the loss function (or -log-likelihood) for the Poisson distribution, which is the log of the probability distribution function:

$$-(N * \text{model} - \text{Exp}(\text{model}) - \text{Log}(\text{Gamma}(N+1)))$$

where **Gamma** is the  $\Gamma$  function  $\Gamma(n + 1) = n!$  and  $e^{\text{model}}$  represents  $\mu$ .

The formula for model is



To run the model, choose **Analyze > Modeling > Nonlinear**. In the Launch dialog, select **model** as the **X, Predictor Formula** and **Poisson** as **Loss**, and then click **OK**. In the Control Panel, set the initial value for **b0** to 1, the other parameters to 0, and click **Go**. The table in Figure 12.17 shows the results after the model has run and you select **Confidence Limits**.

**Figure 12.17** Solution Table for the Poisson Loss Example

Sqrt					
	Loss	DFE	Avg Loss	Avg Loss	
	68.281234087	25	2.7312494	1.6526492	
Parameter	Estimate	ApproxStdErr	Lower CL	Upper CL	
b0	-6.405914771	0.21744445	-6.8430512	-5.9896833	
B	-0.543353982	0.17758996	-0.881379	-0.183552	
C	-0.687418388	0.32904722	-1.3764541	-0.0745299	
D	-0.075979835	0.2905789	-0.6715296	0.47523936	
E	0.325581683	0.23587934	-0.143451	0.78520431	
Yr 65	0.697149668	0.1496412	0.40752821	0.99512758	
Yr 70	0.8184263427	0.16977314	0.48728046	1.15369087	
Yr 75	0.4534435315	0.23317069	-0.0123098	0.90388895	
Used 75	0.3844880718	0.11827198	0.15341509	0.61742312	
Solved By: Analytic NR					
Correlation of Estimates					

---

## Notes Concerning Derivatives

The nonlinear platform takes symbolic derivatives for formulas with most common operations. This section shows what kind of derivative expressions result.

If you open the Negative Exponential.jmp nonlinear sample data example, the actual formula looks something like this:

```
Parameter({b0=0.5, b1=0.5,}b0*(1-Exp(-b1*X)))
```

The Parameter block in the formula is hidden if you use the formula editor, but that is how it is stored in the column and how it appears in the Nonlinear Launch dialog. Two parameters named **b0** and **b1** are given initial values and used in the formula to be fit.

The Nonlinear platform makes a separate copy of the formula, and edits it to extract the parameters from the expression and maps the references to them to the place where they will be estimated. Nonlinear takes the analytic derivatives of the prediction formula with respect to the parameters. If you use the **Show Derivatives** command, you get the resulting formulas listed in the log, like this:

Prediction Model:

```
b0 * First(T#1=1-(T#2=Exp(-b1*X)), T#3=-(-1*T#2*X))
```

The Derivative of Model with respect to the parameters is:

```
{T#1, T#3*b0}
```

The derivative facility works like this:

- In order to avoid calculating subexpressions repeatedly, the prediction model is threaded with assignments to store the values of subexpressions that it will need for derivative calculations. The assignments are made to names like **T#1**, **T#2**, and so forth.
- When the prediction model needs additional subexpressions evaluated, it uses the **First** function, which returns the value of the first argument expression, and also evaluates the other arguments. In this case additional assignments will be needed for derivatives.
- The derivative table itself is a list of expressions, one expression for each parameter to be fit. For example, the derivative of the model with respect to **b0** is **T#1**; its thread in the prediction model is  $1 - (\text{Exp}(-\text{b1} \cdot X))$ . The derivative with respect to **b1** is **T#3\*b0**, which is  $-(-1 \cdot \text{Exp}(-\text{b1} \cdot X) \cdot X) \cdot \text{b0}$  if you substitute in the assignments above. Although many optimizations are made, it doesn't always combine the operations optimally, as you can see by the expression for **T#3**, which doesn't remove a double negation.

If you ask for second derivatives, then you get a list of  $(m(m + 1))/2$  second derivative expressions in a list, where  $m$  is the number of parameters.

If you specify a loss function, then the formula editor takes derivatives with respect to parameters, if it has any, and it takes first and second derivatives with respect to the model, if there is one.

If the derivative mechanism doesn't know how to take the analytic derivative of a function, then it takes numerical derivatives, using the **NumDeriv** function. If this occurs, then the platform shows the delta it used to evaluate the change in the function with respect to a delta change in the arguments. You may need to experiment with different delta settings to obtain good numerical derivatives.

**Tips**

There are always many ways to represent a given model, and some ways behave much better than other forms. Ratkowsky (1990) covers alternative forms in his text.

If you have repeated subexpressions that occur several places in a formula, then it is better to make an assignment to a temporary variable, and then refer to it later in the formula. For example, one of the model formulas above was this:

```
If(Y==0, Log(1/(1+Exp(model))), Log(1 - 1/(1 + Exp(model))));
```

This could be simplified by factoring out an expression and assigning it to a local variable:

```
temp=1/(1+Exp(model));
If(Y==0, Log(temp), Log(1-temp));
```

The derivative facility can track derivatives across assignments and conditionals.

## Notes on Effective Nonlinear Modeling

We strongly encourage you to *center polynomials*.

Anywhere you have a complete polynomial term that is linear in the parameters, it is always good to center the polynomials. This improves the condition of the numerical surface for optimization. For example, if you have an expression like

$$a_1 + b_1x + c_1x^2$$

you should transform it to

$$a_2 + b_2(x - \bar{x}) + c_2(x - \bar{x})^2$$

The two models are equivalent, apart from a transformation of the parameters, but the second model is far easier to fit if the model is nonlinear.

The transformation of the parameters is easy to solve.

$$a_1 = a_2 - b_2\bar{x} + c_2\bar{x}$$

$$b_1 = b_2 - 2c_2\bar{x}$$

$$c_1 = c_2$$

If the number of iterations still goes to the maximum, increase the maximum number of iterations and select **Second Deriv Method** from the red triangle menu.

There is really no one omnibus optimization method that works well on all problems. JMP has options like **Newton**, **QuasiNewton BFGS**, **QuasiNewton SR1**, **Second Deriv Method**, and **Numeric Derivatives Only** to expand the range of problems that are solvable by the Nonlinear Platform.

So if JMP's defaults are unable to converge to the solution for a particular problem, using various combinations of these settings increase the odds of obtaining convergence.

Some models are very sensitive to starting values of the parameters. Working on new starting values is often effective. Edit the starting values and click **Reset** to see the effect.

The plot often helps. Use the sliders to visually modify the curve to fit better. The parameter profilers can help, but may be too slow for anything but small data sets.

---

## Notes Concerning Scripting

If you are using a JSL script to run the Nonlinear platform, then the following commands could be useful, along with the commands listed under the Nonlinear platform popup menu.

**Table 12.1** JSL commands for Nonlinear

Model(Parameter({name=expr,...},expr)	The model to fit. If not specified, it looks in the <i>X</i> column's formula, if there is an <i>X</i> column.
Loss(expr) or Loss(Parameter({name=expr,...},expr)	The loss function to use. If not specified, it looks in the Loss columns formula, if there is a loss column.
Go	Like the <b>Go</b> button, it starts the iterations, but unlike the <b>Go</b> button, the iterations run in the foreground instead of the background.
Finish	Used instead of the <b>Go</b> command. <b>Finish</b> halts script execution until the fit is complete.
Reset	Same as the <b>Reset</b> button.
Set Parameter(name=expression,...)	To provide new starting values. Expression may even be a Column function like Col Maximum. name is a parameter name.
Lock Parameter(name,...)	To Lock or Unlock. name is a parameter name.
Save Estimates	Same as the <b>Save Estimates</b> button
Confidence Limits	Same as the <b>Confidence Limits</b> button
SSE Grid(name(first,last,),...)	To calculate the SSE on a Grid, on <i>n</i> values in the range specified for each parameter. Unspecified parameters are left at the solution value.
Get SSE	Retrieve the sum of squares error (SSE). (Scripting only.)
Get Parameter Names	Retrieve the parameter names. (Scripting only.)
Get Std Errors	Retrieve the parameter standard errors. (Scripting only.)

**Table 12.1** JSL commands for Nonlinear (Continued)

Get Corr	Retrieve the correlation of the estimates. (Scripting only.)
Get CI	Retrieve the confidence interval about the parameters. Note: you must have already have clicked the <b>Confidence Limits</b> button. (Scripting only.)

The first example below uses a By-group and specifies an expression to evaluate to get starting values. The second example specifies the model itself.

```
Nonlinear(x(Model1),y(pop),By(group),
  Set Parameter(B0=ColMinimum(Pop),B1=.03),
  Go);
Nonlinear(y(pop),
  Model(
    Parameter({B0=3.9,B1=0.022},
      B0*Exp(B1*(year-1790))),
    SSEGrid(B0(2,20,10),B1(.01,.05,6)));
```

## Nonlinear Modeling Templates

This section shows examples of some nonlinear models and sources of data for each model. These models and others are in the Model Library, discussed in the section “[Using the Model Library](#),” p. 235.

**Table 12.2** Guide to Nonlinear Modeling Templates

Data Reference	Formula	Model
Meyers (1988), p. 310	$\frac{\theta_1 x}{\theta_2 + x}$	A Michaelis-Menten
Draper and Smith (1981), p. 522, L	$\theta_1[1 - \exp(-\theta_2 x)]$	B
Draper and Smith (1981), p. 476	$\theta_1 + (0.49 - \theta_1)\exp[-\theta_2(x - 8)]$	C
Draper and Smith (1981), p. 519, H	$\exp\left\{-\theta_1 x_1 \exp\left[-\theta_2 \cdot \left(\frac{1}{x_2} - \frac{1}{620}\right)\right]\right\}$	D

**Table 12.2** Guide to Nonlinear Modeling Templates (*Continued*)

Data Reference	Formula	Model
Draper and Smith (1981), p. 519, H	$\theta_1 x^{\theta_2}$	E
Bates and Watts (1988), p. 310	$\theta_1 + \theta_2 \exp(\theta_3 x)$	F Asymptotic Regression
Bates and Watts (1988), p. 310	$\frac{\theta_1}{1 + \theta_2 \exp(\theta_3 x)}$	G Logistic
Bates and Watts (1988), p. 310	$\theta_1 \exp[-\exp(\theta_2 - \theta_3 x)]$	H Gompertz Growth
Draper and Smith (1981), p. 524, N	$\theta_1 (1 - \theta_2 \exp(-\theta_3 x))$	I
Draper and Smith (1981), p. 524, N	$\theta_1 - \ln[1 + \theta_2 \exp(-\theta_3 x)]$	J Loglogistic
Draper and Smith (1981), p. 524, P	$\theta_1 + \frac{\theta_2}{x^{\theta_3}}$	K
Draper and Smith (1981), p. 524, P	$\ln[\theta_1 \exp(-\theta_2 x) + (1 - \theta_1) \exp(-\theta_3 x)]$	L
Bates and Watts (1988), p. 271	$\frac{\theta_1 \theta_3 \left( x_2 - \frac{x_3}{1.632} \right)}{1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_3}$	M
Bates and Watts (1988), p. 310	$\frac{\theta_2 \theta_3 + \theta_1 x^{\theta_4}}{\theta_3 + x^{\theta_4}}$	O Morgan-Mercer-Florin

**Table 12.2** Guide to Nonlinear Modeling Templates (*Continued*)

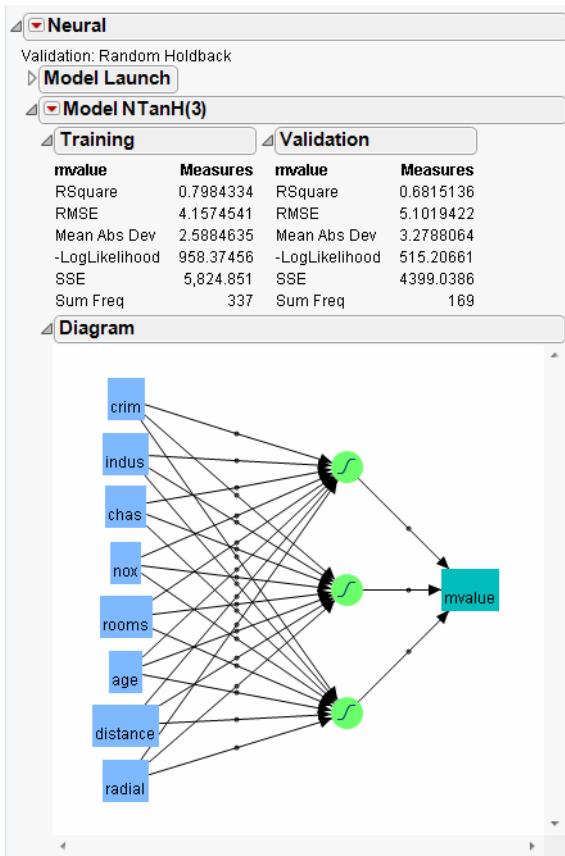
Data Reference	Formula	Model
Bates and Watts (1988), p. 310	$\frac{\theta_1}{[1 + \theta_2 \exp(-\theta_3 x)]^{\frac{1}{\theta_4}}}$	P Richards Growth
Bates and Watts (1988), p. 274	$\frac{\theta_1}{\theta_2 + x_1} + \theta_3 x_2 + \theta_4 x_2^2 + \theta_5 x_2^3 + (\theta_6 + \theta_7 x_2^2) x_2 \exp\left(\frac{-x_1}{\theta_8 + \theta_9 x_2}\right)$	S

# Chapter 13

## Creating Neural Networks Using the Neural Platform

The Neural platform implements a fully-connected multi-layer perceptron with one or two layers. Use neural networks to predict one or more response variables using a flexible function of the input variables. Neural networks can be very good predictors when it is not necessary to describe the functional form of the response surface, or to describe the relationship between the inputs and the response.

**Figure 13.1** Example of a Neural Network



# Contents

Overview of Neural Networks . . . . .	269
Launch the Neural Platform . . . . .	269
The Neural Launch Window . . . . .	270
The Model Launch . . . . .	271
Model Reports . . . . .	276
Training and Validation Measures of Fit . . . . .	277
Confusion Statistics . . . . .	277
Model Options . . . . .	278
Example of a Neural Network . . . . .	279

---

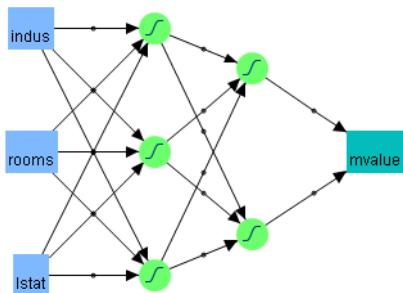
## Overview of Neural Networks

Think of a neural network as a function of a set of derived inputs, called hidden nodes. The hidden nodes are nonlinear functions of the original inputs. You can specify up to two layers of hidden nodes, with each layer containing as many hidden nodes as you want.

Figure 13.2 shows a two-layer neural network with three X variables and one Y variable. In this example, the first layer has two nodes, and each node is a function of all three nodes in the second layer. The second layer has three nodes, and all nodes are a function of the three X variables. The predicted Y variable is a function of both nodes in the first layer.

---

**Figure 13.2** Neural Network Diagram



---

The functions applied at the nodes of the hidden layers are called activation functions. The activation function is a transformation of a linear combination of the X variables. For more details about the activation functions, see “[Hidden Layer Structure](#), p. 273.”

The function applied at the response is a linear combination (for continuous responses), or a logistic transformation (for nominal or ordinal responses).

The main advantage of a neural network model is that it can efficiently model different response surfaces. Given enough hidden nodes and layers, any surface can be approximated to any accuracy. The main disadvantage of a neural network model is that the results are not easily interpretable, since there are intermediate layers rather than a direct path from the X variables to the Y variables, as in the case of regular regression.

---

**Note:** Most features discussed in this chapter are for JMP Pro only, and are designated as such with a note.

---

---

## Launch the Neural Platform

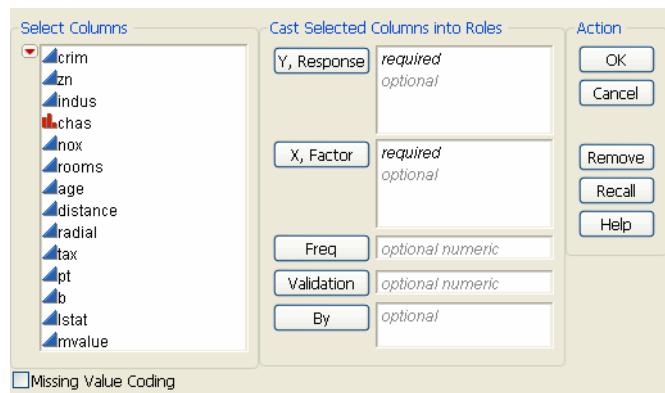
To launch the Neural platform, select **Analyze > Modeling > Neural**.

Launching the Neural platform is a two-step process. First, enter your variables on the Neural launch window. Second, specify your options in the Model Launch.

## The Neural Launch Window

Use the Neural launch window to specify X and Y variables, a validation column, and to enable missing value coding. The Validation button and Missing Value Coding option are available only in JMP Pro.

**Figure 13.3** The Neural Launch Window



**Table 13.1** Description of the Neural Launch Window

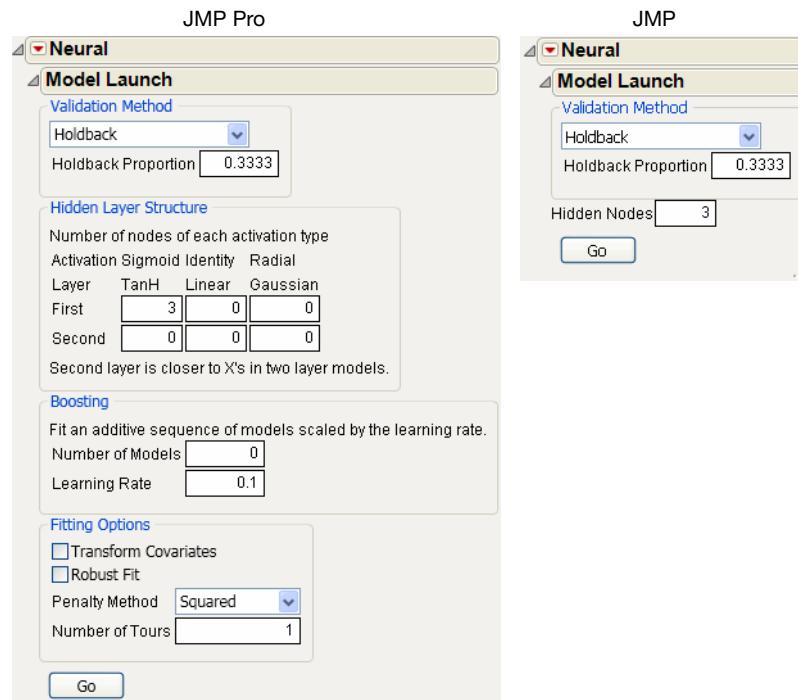
<b>Y, Response</b>	Choose the response variable. When multiple responses are specified, the models for the responses share all parameters in the hidden layers (those parameters not connected to the responses).
<b>X, Factor</b>	Choose the input variables.
<b>Freq</b>	Choose a frequency variable.
<b>Validation</b>	<p><b>Note:</b> This button appears only in JMP Pro.</p> <p>Choose a validation column. For more information, see “<a href="#">Validation Method</a>,” p. 272.</p>
<b>By</b>	Choose a variable to create separate models for each level of the variable.

**Table 13.1** Description of the Neural Launch Window (Continued)

<b>Missing Value Coding</b>	<p><b>Note:</b> This option is available only in JMP Pro.</p> <p>Check this box to enable missing value imputation and coding. If this option is not checked, rows with missing values are ignored.</p> <p>For continuous variables, missing values are replaced by the mean of the variable. Also, a missing value indicator variable is created and included in the model. If a variable is transformed, the imputation occurs after the transformation.</p> <p>For categorical variables, the missing values are not imputed, but are treated as another level of the variable in the model.</p>
-----------------------------	---

## The Model Launch

Use the Model Launch dialog to specify the validation method, the structure of the hidden layer, whether to use gradient boosting, and other fitting options.

**Figure 13.4** The Model Launch Dialog

**Table 13.2** Description of the Model Launch Dialog

Validation Method	Select the method that you want to use for model validation. For details, see “ <a href="#">Validation Method</a> ,” p. 272.
Hidden Layer Structure or Hidden Nodes	<p><b>Note:</b> The standard edition of JMP uses only the TanH activation function, and can fit only neural networks with one hidden layer.</p> <p>Specify the number of hidden nodes of each type in each layer. For details, see “<a href="#">Hidden Layer Structure</a>,” p. 273.</p>
Boosting	<p><b>Note:</b> These options are available only in JMP Pro.</p> <p>Specify options for gradient boosting. For details, see “<a href="#">Boosting</a>,” p. 274.</p>
Fitting Options	<p><b>Note:</b> These options are available only in JMP Pro.</p> <p>Specify options for variable transformation and model fitting. For details, see “<a href="#">Fitting Options</a>,” p. 275.</p>
Go	Fits the neural network model and shows the model reports.

After you click Go to fit a model, you can reopen the Model Launch Dialog and change the settings to fit another model.

### **Validation Method**

Neural networks are very flexible models and have a tendency to overfit data. When that happens, the model predicts the fitted data very well, but predicts future observations poorly. To mitigate overfitting, the Neural platform does the following:

- applies a penalty on the model parameters
- uses an independent data set to assess the predictive power of the model

Validation is the process of using part of a data set to estimate model parameters, and using the other part to assess the predictive ability of the model.

- The *training* set is the part that estimates model parameters.
- The *validation* set is the part that estimates the optimal value of the penalty, and assesses or validates the predictive ability of the model.
- The *test* set is a final, independent assessment of the model’s predictive ability. The test set is available only when using a validation column. See Table 13.3.

The training, validation, and test sets are created by subsetting the original data into parts. Table 13.3 describes several methods for subsetting a data set.

**Table 13.3** Validation Methods

Excluded Rows	Uses row states to subset the data. Rows that are unexcluded are used as the training set, and excluded rows are used as the validation set.  For more information about using row states and how to exclude rows, see <i>Using JMP</i> .
Holdback	Randomly divides the original data into the training and validation sets. You can specify the proportion of the original data to use as the validation set (holdback).
KFold	Divides the original data into K subsets. In turn, each of the K sets is used to validate the model fit on the rest of the data, fitting a total of K models. The model giving the best validation statistic is chosen as the final model.  This method is best for small data sets, because it makes efficient use of limited amounts of data.
Validation Column	<p><b>Note:</b> The use of a validation column is available only in JMP Pro.</p> <p>Uses the column's values to divide the data into parts. The column is assigned using the Validation role on the Neural launch window. See Figure 13.3.</p> <p>The column's values determine how the data is split, and what method is used for validation:</p> <ul style="list-style-type: none"> <li>• If the column has three unique values, then: <ul style="list-style-type: none"> <li>– the smallest value is used for the Training set.</li> <li>– the middle value is used for the Validation set.</li> <li>– the largest value is used for the Test set.</li> </ul> </li> <li>• If the column has two unique values, then only Training and Validation sets are used.</li> <li>• If the column has more than three unique values, then KFold validation is performed.</li> </ul>

### Hidden Layer Structure

---

**Note:** The standard edition of JMP uses only the TanH activation function, and can fit only neural networks with one hidden layer.

---

The Neural platform can fit one or two-layer neural networks. Increasing the number of nodes in the first layer, or adding a second layer, makes the neural network more flexible. You can add an unlimited number

of nodes to either layer. The second layer nodes are functions of the X variables. The first layer nodes are functions of the second layer nodes. The Y variables are functions of the first layer nodes.

The functions applied at the nodes of the hidden layers are called activation functions. An activation function is a transformation of a linear combination of the X variables. Table 13.4 describes the three types of activation functions.

**Table 13.4** Activation Functions

TanH	<p>The hyperbolic tangent function is a sigmoid function. TanH transforms values to be between -1 and 1, and is the centered and scaled version of the logistic function. The hyperbolic tangent function is:</p> $\frac{e^{2x} - 1}{e^{2x} + 1}$ <p>where <math>x</math> is a linear combination of the X variables.</p>
Linear	<p>The identity function. The linear combination of the X variables is not transformed.</p> <p>The Linear activation function is most often used in conjunction with one of the non-linear activation functions. In this case, the Linear activation function is placed in the second layer, and the non-linear activation functions are placed in the first layer. This is useful if you want to first reduce the dimensionality of the X variables, and then have a nonlinear model for the Y variables.</p> <p>For a continuous Y variable, if only Linear activation functions are used, the model for the Y variable reduces to a linear combination of the X variables. For a nominal or ordinal Y variable, the model reduces to a logistic regression.</p>
Gaussian	<p>The Gaussian function. Use this option for radial basis function behavior, or when the response surface is Gaussian (normal) in shape. The Gaussian function is:</p> $e^{-x^2}$ <p>where <math>x</math> is a linear combination of the X variables.</p>

## Boosting

---

**Note:** This feature is available only in JMP Pro.

---

Boosting is the process of building a large additive neural network model by fitting a sequence of smaller models. Each of the smaller models is fit on the scaled residuals of the previous model. The models are combined to form the larger final model. The process uses validation to assess how many component models to fit, not exceeding the specified number of models.

Boosting is often faster than fitting a single large model. However, the base model should be a 1 to 2 node single-layer model, or else the benefit of faster fitting can be lost if a large number of models is specified.

Use the Boosting panel in the Model Launch to specify the number of component models and the learning rate. Use the Hidden Layer Structure panel in the Model Launch to specify the structure of the base model.

The learning rate must be  $0 < r \leq 1$ . Learning rates close to 1 result in faster convergence on a final model, but also have a higher tendency to overfit data. Use learning rates close to 1 when a small Number of Models is specified.

As an example of how boosting works, suppose you specify a base model consisting of one layer and two nodes, with the number of models equal to eight. The first step is to fit a one-layer, two-node model. The predicted values from that model are scaled by the learning rate, then subtracted from the actual values to form a scaled residual. The next step is to fit a different one-layer, two-node model on the scaled residuals of the previous model. This process continues until eight models are fit, or until the addition of another model fails to improve the validation statistic. The component models are combined to form the final, large model. In this example, if six models are fit before stopping, the final model consists of one layer and  $2 \times 6 = 12$  nodes.

## Fitting Options

---

**Note:** These options are available only in JMP Pro.

---

Table 13.5 describes the model fitting options that you can specify.

**Table 13.5** Fitting Options

Transform Covariates	Transforms all continuous variables to near normality using either the Johnson Su or Johnson Sb distribution. Transforming the continuous variables helps to mitigate the negative effects of outliers or heavily skewed distributions.  See the Save Transformed Covariates option in “ <a href="#">Model Options</a> ,” p. 278.
Robust Fit	Trains the model using least absolute deviations instead of least squares. This option is useful if you want to minimize the impact of response outliers. This option is available only for continuous responses.
Penalty Method	Choose the penalty method. To mitigate the tendency neural networks have to overfit data, the fitting process incorporates a penalty on the likelihood. See “ <a href="#">Penalty Method</a> ,” p. 275.
Number of Tours	Specify the number of times to restart the fitting process, with each iteration using different random starting points for the parameter estimates. The iteration with the best validation statistic is chosen as the final model.

### Penalty Method

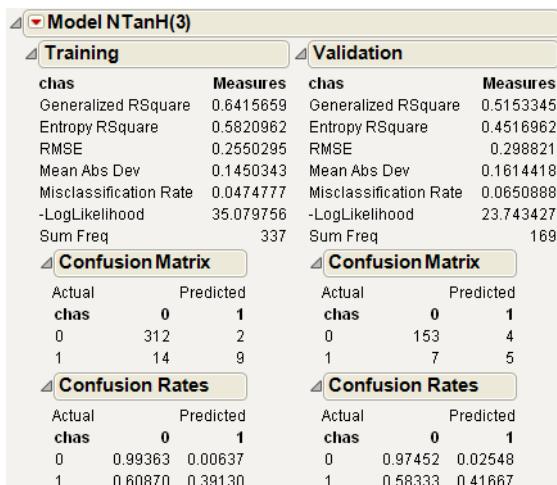
The penalty is  $\lambda p(\beta_i)$ , where  $\lambda$  is the penalty parameter, and  $p()$  is a function of the parameter estimates, called the penalty function. Validation is used to find the optimal value of the penalty parameter.

**Table 13.6** Descriptions of Penalty Methods

Method	Penalty Function	Description
Squared	$\sum \beta_i^2$	Use this method if you think that most of your X variables are contributing to the predictive ability of the model.
Absolute	$\sum  \beta_i $	Use either of these methods if you have a large number of X variables, and you think that a few of them contribute more than others to the predictive ability of the model.
Weight Decay	$\sum \frac{\beta_i^2}{1 + \beta_i^2}$	
NoPenalty	none	Does not use a penalty. You can use this option if you have a large amount of data and you want the fitting process to go quickly. However, this option can lead to models with lower predictive performance than models that use a penalty.

## Model Reports

A model report is created for every neural network model. Measures of fit appear for the training and validation sets. Additionally, confusion statistics appear when the response is nominal or ordinal.

**Figure 13.5** Example of a Model Report

## Training and Validation Measures of Fit

Measures of fit appear for the training and validation sets. See Figure 13.5.

**Table 13.7** Descriptions of the Training and Validation Measures of Fit

Generalized RSquare	A generalization of the Rsquare measure that simplifies to the regular Rsquare for continuous responses. Similar to the Entropy RSquare, but instead of using the log-likelihood, the Generalized RSquare uses the $2/n$ root of the likelihood. It is scaled to have a maximum of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model.
Entropy RSquare	Compares the log-likelihoods from the fitted model and the constant probability model. Appears only when the response is nominal or ordinal.
RSquare	Gives the Rsquare for the model.
RMSE	Gives the root mean square error. When the response is nominal or ordinal, the differences are between 1 and $p$ (the fitted probability for the response level that actually occurred).
Mean Abs Dev	The average of the absolute values of the differences between the response and the predicted response. When the response is nominal or ordinal, the differences are between 1 and $p$ (the fitted probability for the response level that actually occurred).
Misclassification Rate	The rate for which the response category with the highest fitted probability is not the observed category. Appears only when the response is nominal or ordinal.
-LogLikelihood	Gives the negative of the log likelihood.
SSE	Gives the error sums of squares. Available only when the response is continuous.
Sum Freq	Gives the number of observations that are used. If you specified a Freq variable in the Neural launch window, Sum Freq gives the sum of the frequency column.

If there are multiple responses, fit statistics are given for each response, and an overall Generalized Rsquare and -LogLikelihood is given.

## Confusion Statistics

For nominal or ordinal responses, a Confusion Matrix report and Confusion Rates report is given. See Figure 13.5. The Confusion Matrix report shows a two-way classification of the actual response levels and the predicted response levels. For a categorical response, the predicted level is the one with the highest predicted probability. The Confusion Rates report is equal to the Confusion Matrix report, with the numbers divided by the row totals.

## Model Options

Each model report has a red triangle menu containing options for producing additional output or saving results. Table 13.8 describes the options in the red triangle menus.

**Table 13.8** Model Report Options

<b>Diagram</b>	Shows a diagram representing the hidden layer structure.
<b>Show Estimates</b>	Shows the parameter estimates in a report.
<b>Profiler</b>	Launches the Prediction Profiler. For nominal or ordinal responses, each response level is represented by a separate row in the Prediction Profiler. For details about the options in the red triangle menu, see the “ <a href="#">Profiling</a> ” chapter.
<b>Categorical Profiler</b>	Launches the Prediction Profiler. Similar to the Profiler option, except that all categorical probabilities are combined into a single profiler row. Available only for nominal or ordinal responses. For details about the options in the red triangle menu, see the “ <a href="#">Profiling</a> ” chapter.
<b>Contour Profiler</b>	Launches the Contour Profiler. This is available only when the model contains more than one continuous factor. For details about the options in the red triangle menu, see the “ <a href="#">Profiling</a> ” chapter.
<b>Surface Profiler</b>	Launches the Surface Profiler. This is available only when the model contains more than one continuous factor. For details about the options in the red triangle menu, see the “ <a href="#">Profiling</a> ” chapter.
<b>ROC Curve</b>	Creates an ROC curve. Available only for nominal or ordinal responses. For details about ROC Curves, see “ <a href="#">ROC Curve</a> ,” p. 315 in the “Recursive Partitioning” chapter.
<b>Lift Curve</b>	Creates a lift curve. Available only for nominal or ordinal responses. For details about Lift Curves, see “ <a href="#">Lift Curves</a> ,” p. 317 in the “Recursive Partitioning” chapter.
<b>Plot Actual by Predicted</b>	Plots the actual versus the predicted response. Available only for continuous responses.
<b>Plot Residual by Predicted</b>	Plots the residuals versus the predicted responses. Available only for continuous responses.
<b>Save Formulas</b>	Creates new columns in the data table containing formulas for the predicted response and the hidden layer nodes.

**Table 13.8** Model Report Options (*Continued*)

<b>Save Profile Formulas</b>	Creates new columns in the data table containing formulas for the predicted response. Formulas for the hidden layer nodes are embedded in this formula. This option produces formulas that can be used by the Flash version of the Profiler.
<b>Save Fast Formulas</b>	Creates new columns in the data table containing formulas for the predicted response. Formulas for the hidden layer nodes are embedded in this formula. This option produces formulas that evaluate faster than the other options, but cannot be used in the Flash version of the Profiler.
<b>Make SAS Data Step</b>	Creates SAS code that you can use to score a new data set.
<b>Save Validation</b>	Creates a new column in the data table that identifies which rows were used in the training and validation sets. This option is not available when a Validation column is specified on the Neural launch window. See “ <a href="#">The Neural Launch Window</a> ,” p. 270.
<b>Save Transformed Covariates</b>	<p><b>Note:</b> This option is available only in JMP Pro.</p> <p>Creates new columns in the data table showing the transformed covariates. The columns contain formulas that show the transformations. This option is available only when the Transform Covariates option is checked on the Model Launch. See “<a href="#">Fitting Options</a>,” p. 275.</p>
<b>Remove Fit</b>	Removes the entire model report.

---

## Example of a Neural Network

This example uses the **Boston Housing.jmp** data table. Suppose you want to create a model to predict the median home value as a function of several demographic characteristics. Follow the steps below to build the neural network model:

1. Launch the Neural platform by selecting **Analyze > Modeling > Neural**.
2. Assign **mvalue** to the **Y, Response** role.
3. Assign the other columns (crim through lstat) to the **X, Factor** role.
4. Click **OK**.
5. Enter 0.2 for the Holdback Proportion.
6. Enter 3 for the number of TanH nodes in the first layer.
7. Check the **Transform Covariates** option.
8. Click **Go**.

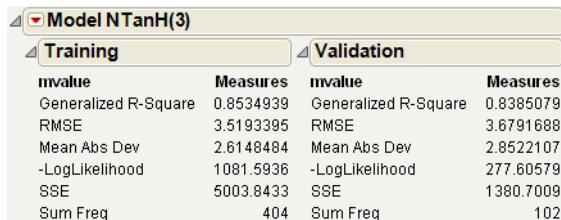
The report is shown in Figure 13.6.

---

**Note:** Results will vary due to the random nature of choosing a validation set.

---

**Figure 13.6** Neural Report



The screenshot shows a software interface for a neural network model named "Model NTanH(3)". It has two tabs: "Training" and "Validation". The "Training" tab displays the following statistics:

Measure	Value
Generalized R-Square	0.8534939
RMSE	3.5193395
Mean Abs Dev	2.6148484
-LogLikelihood	1081.5936
SSE	5003.8433
Sum Freq	404

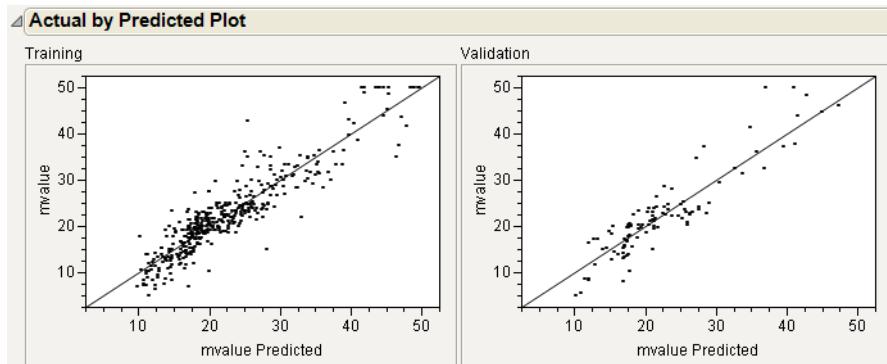
The "Validation" tab displays the following statistics:

Measure	Value
Generalized R-Square	0.8385079
RMSE	3.6791688
Mean Abs Dev	2.8522107
-LogLikelihood	277.60579
SSE	1380.7009
Sum Freq	102

Results are provided for both the training and validation sets. Use the results of the validation set as a representation of the model's predictive power on future observations.

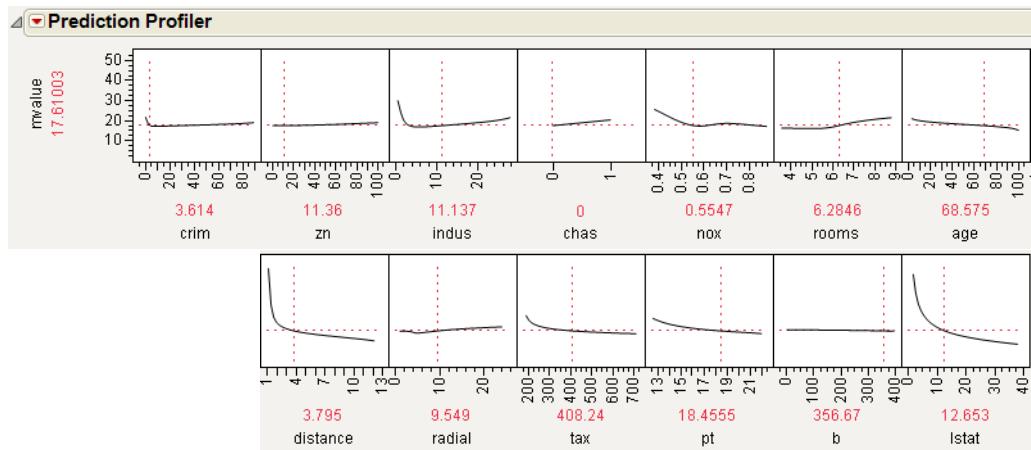
The R-Square statistic for the Validation set is 0.838, signifying that the model is predicting well on data not used to train the model. As an additional assessment of model fit, select **Plot Actual by Predicted** from the Model red-triangle menu. The plot is shown in Figure 13.7.

**Figure 13.7** Actual by Predicted Plot



The points fall along the line, signifying that the predicted values are similar to the actual values.

To get a general understanding of how the  $X$  variables are impacting the predicted values, select **Profiler** from the Model red-triangle menu. The profiler is shown in Figure 13.8.

**Figure 13.8** Profiler

Some of the variables have profiles with positive slopes, and some negative. For example, *rooms* has a positive slope. This indicates that the more rooms a home has, the higher the predicted median value. The variable *age* is the proportion of owner-occupied units built prior to 1940. This variable has a negative slope, indicating that the more older homes there are in the area, the lower the median value.



# Chapter 14

## Gaussian Processes Models for Analyzing Computer Experiments

---

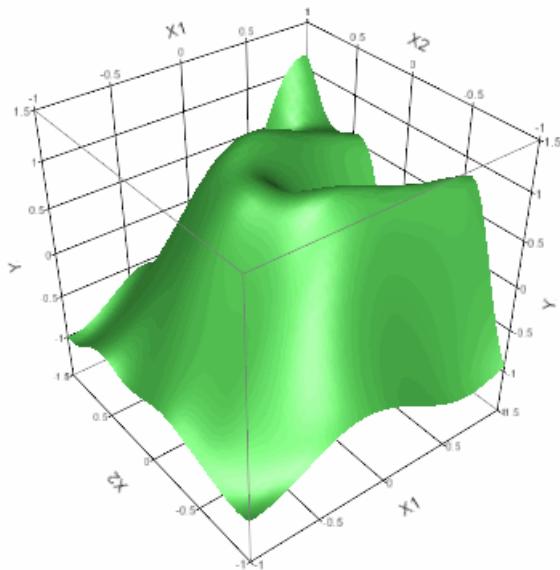
The Gaussian Process platform is used to model the relationship between a continuous response and one or more continuous predictors. These models are common in areas like computer simulation experiments, such as the output of finite element codes, and they often perfectly interpolate the data. Gaussian processes can deal with these no-error-term models, in which the same input values always result in the same output value.

The Gaussian Process platform fits a spatial correlation model to the data, where the correlation of the response between two observations decreases as the values of the independent variables become more distant.

The main purpose for using this platform is to obtain a prediction formula that can be used for further analysis and optimization.

---

**Figure 14.1** Example of a Gaussian Process Prediction Surface



---

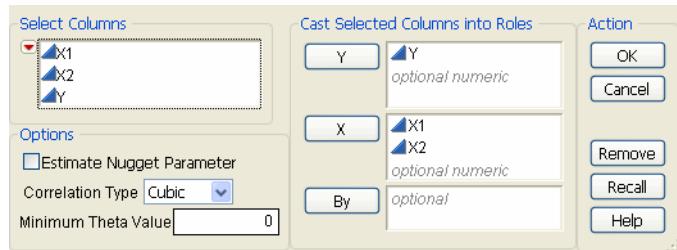
# Contents

Launching the Platform .....	285
The Gaussian Process Report .....	286
Actual by Predicted Plot .....	287
Model Report .....	287
Marginal Model Plots .....	288
Platform Options .....	289
Borehole Hypercube Example .....	290

## Launching the Platform

To launch the Gaussian Process platform, choose **Analyze > Modeling > Gaussian Process** from the main menu bar. Here, we illustrate the platform with 2D Gaussian Process Example.jmp data set, found in the Sample Data folder.

**Figure 14.2** Gaussian Process Launch Dialog



The launch dialog has the following options:

**Estimate Nugget Parameter** introduces a ridge parameter into the estimation procedure. This is useful if there is noise or randomness in the response, and you would like the prediction model to smooth over the noise instead of perfectly interpolating.

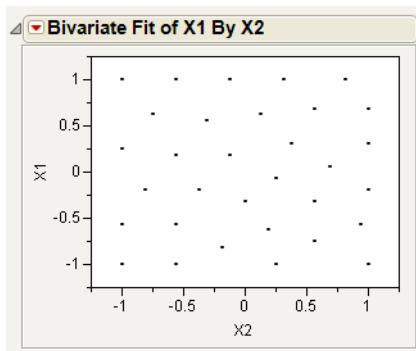
**Correlation Type** lets you choose the correlation structure used in the model. The platform fits a spatial correlation model to the data, where the correlation of the response between two observations decreases as the values of the independent variables become more distant.

**Gaussian** restricts the correlation between two points to always be non-zero, no matter the distance between the points.

**Cubic** lets the correlation between two points to be zero for points far enough apart. This method can be considered a generalization of a cubic spline.

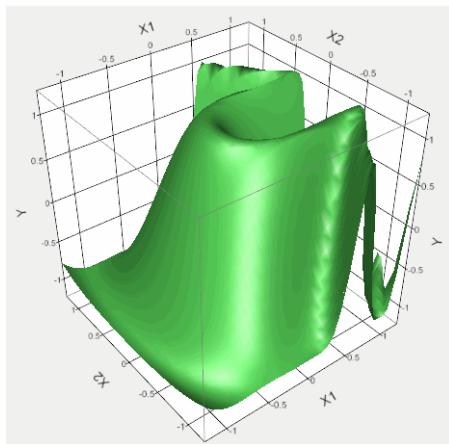
**Minimum Theta Value** lets you set the minimum theta value used in the fitted model. The default is 0. The theta values are analogous to a slope parameter in regular regression models. If a theta value is 0 in the fitted model, then that *X* variable has no influence on the predicted values.

In this example, we are interested in finding the explanatory power of the two *x*-variables (*X1* and *X2*) on *Y*. A plot of *X1* and *X2* shows their even dispersal in the factor space.



---

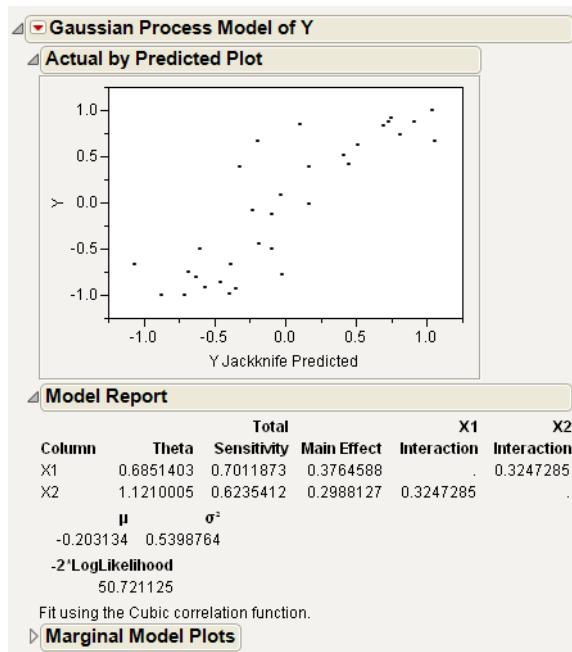
Since this is generated data, we can look at the function that generates the Y values. It is this function that we want to estimate.



---

## The Gaussian Process Report

After clicking OK from the launch dialog, the following report appears.

**Figure 14.3** Gaussian Process Default Report

## Actual by Predicted Plot

The Actual by Predicted plot shows the actual Y values on the *y*-axis and the jackknife predicted values on the *x*-axis. One measure of goodness-of-fit is how well the points lie along the 45 degree diagonal line.

The jackknife values are really pseudo-jackknife values because they are not refit unless the row is excluded. Therefore, the correlation parameters still have the contribution of that row in them, but the prediction formula does not. If the row is excluded, neither the correlation parameters nor the prediction formula have the contribution.

## Model Report

The Model Report shows a functional ANOVA table for the model parameters that the platform estimates. Specifically, it is an analysis of variance table, but the variation is computed using a function-driven method.

The Total Variation is the integrated variability over the entire experimental space.

For each covariate, we can create a marginal prediction formula by averaging the overall prediction formula over the values of all the other factors. The functional main effect of X1 is the integrated total variation due to X1 alone. In this case, we see that 37.6% of the variation in Y is due to X1.

The ratio of (Functional X1 effect)/(Total Variation) is the value listed as the Main Effect in the Model report. A similar ratio exists for each factor in the model.

Functional interaction effects, computed in a similar way, are also listed in the Model Report table.

Summing the value for main effect and all interaction terms gives the Total Sensitivity, the amount of influence a factor and all its two-way interactions have on the response variable.

### **Mu, Theta, and Sigma**

The Gaussian correlation structure uses the product exponential correlation function with a power of 2 as the estimated model. This comes with the assumptions that  $Y$  is Normally distributed with mean  $\mu$  and covariance matrix  $\sigma^2 \mathbf{R}$ . The  $\mathbf{R}$  matrix is composed of elements

$$r_{ij} = \exp\left(-\sum_k \theta_k (x_{ik} - x_{jk})^2\right)$$

In the Model report,  $\mu$  is the Normal distribution mean,  $\sigma^2$  is the Normal Distribution parameter, and the Theta column corresponds to the values of  $\theta_k$  in the definition of  $\mathbf{R}$ .

These parameters are all fitted via maximum likelihood.

**Note:** If you see **Nugget parameters set to avoid singular variance matrix**, JMP has added a ridge parameter to the variance matrix so that it is invertible.

The Cubic correlation structure also assumes that  $Y$  is Normally distributed with mean  $\mu$  and covariance matrix  $\sigma^2 \mathbf{R}$ . The  $\mathbf{R}$  matrix is composed of elements

$$r_{ij} = \prod_k \rho(d; \theta_k) \quad d = x_{ik} - x_{jk}$$

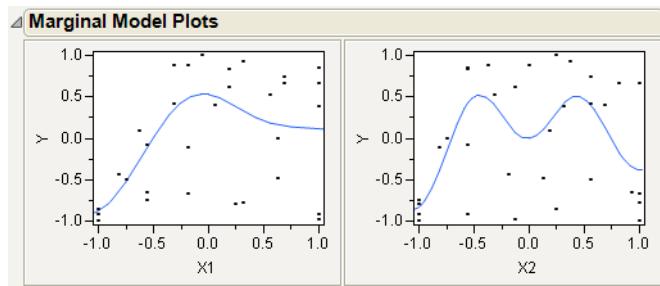
where

$$\rho(d; \theta) = \begin{cases} 1 - 6(d/\theta)^2 + 6(|d/\theta|^3), & |d| \leq \frac{1}{2\theta} \\ 2(1 - |d/\theta|^3), & \frac{1}{2\theta} < |d| \leq \frac{1}{\theta} \\ 0, & \frac{1}{\theta} < |d| \end{cases}$$

For more information see Santer (2003). The theta parameter used in the cubic correlation is the reciprocal of the parameter used in the literature. The reason is so that when a parameter (theta) has no effect on the model, then it has a value of zero, instead of infinity.

### **Marginal Model Plots**

The Marginal Model plots are shown in Figure 14.4.

**Figure 14.4** Marginal Model Plots

These plots show the average value of each factor across all other factors. In this two-dimensional example, we examine slices of  $X_1$  from  $-1$  to  $1$ , and plot the average value at each point.

## Platform Options

The Gaussian Process platform has the following options:

**Profiler** brings up the standard Profiler.

**Contour Profiler** brings up the Contour Profiler.

**Surface Profiler** brings up the Surface Profiler.

Details on Profiling are found in the “[Profiling](#)” chapter.

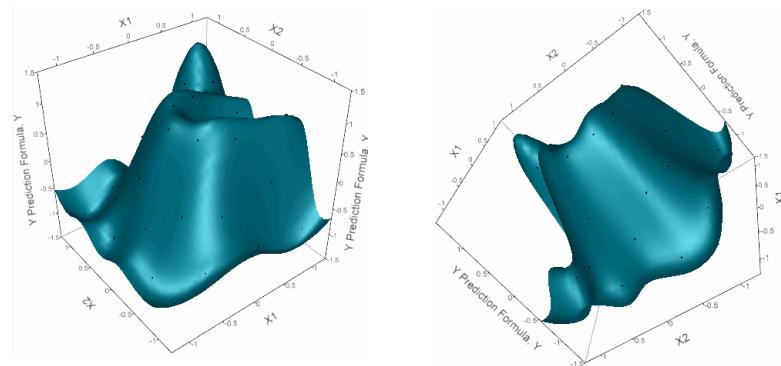
**Save Prediction Formula** creates a new column in the table containing the prediction formula.

**Save Variance Formula** creates a new column in the table containing the variance formula.

**Save Jackknife Predicted Values** stores the jackknife predicted values to the data table. These are the  $x$ -axis values for the Actual by Predicted Plot.

**Script** the standard JMP Script menu.

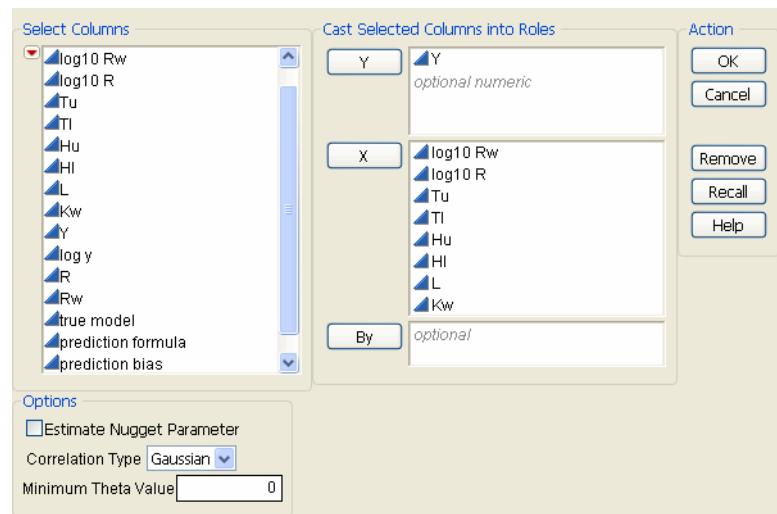
In Figure 14.5, we use the saved prediction formula to compare the prediction to the actual data points.

**Figure 14.5** Two Views of the Prediction Surface and Actual Ys

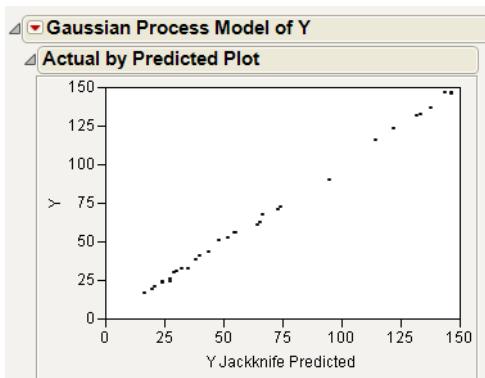
## Borehole Hypercube Example

A more complicated model is seen using Borehole Latin Hypercube.jmp, found in the Design Experiment folder.

To launch the analysis, fill out the Gaussian Process dialog as shown in Figure 14.6.

**Figure 14.6** Borehole Latin Hypercube Launch Dialog

When you click **OK**, the following Actual by Predicted plot appears.



Since the points are close to the 45 degree diagonal line, we can be confident that the Gaussian process prediction model is a good approximation to the true function that generated the data.

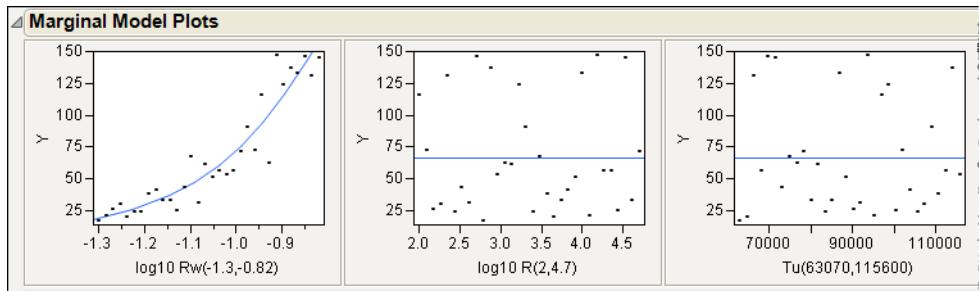
The Model Report shows us that this is mainly due to one factor,  $\log_{10} \text{Rw}$ . The main effect explains 87.5% of the variation, with 90.5% explained when all second-order interactions are included.

Most of the variation is  
explained by  $\log_{10}\text{Rw}$

Model Report											
Column	Theta	Total Sensitivity	Main Effect	$\log_{10} \text{Rw}$	$\log_{10} \text{R}$	Tu	TI	Hu	HI	Kw	
				Interaction	Interaction	Interaction	Interaction	Interaction	Interaction	Interaction	
log10 Rw	4.6447919	0.9047126	0.8751062		0	0	0	0.0092619	0.0102189	0.0084618	0.0016638
log10 R	0	0	0	0	0	0	0	0	0	0	0
Tu	0	0	0	0	0	0	0	0	0	0	0
TI	0	0	0	0	0	0	0	0	0	0	0
Hu	2.1669e-6	0.0400141	0.0304916	0.0092619	0	0	0	3.2173e-7	0.0002417	1.8471e-5	
HI	1.9013e-6	0.0424113	0.0320668	0.0102189	0	0	0	3.2173e-7	0.000121	4.4258e-6	
L	1.4549e-7	0.0349244	0.0260815	0.0084618	0	0	0	0.0002417	0.000121	1.8352e-5	
Kw	1.4577e-9	0.0079433	0.0062382	0.0016638	0	0	0	1.8471e-5	4.4258e-6	1.8352e-5	
<b>Nugget</b>											
<b><math>\mu</math></b>											
151.25532 12454.793											
<b><math>\sigma^2</math></b>											
0.0001											
<b>-2 LogLikelihood</b>											
205.56463											
Fit using the Gaussian correlation function.											
Nugget parameter set to avoid singular variance matrix.											
<b>Marginal Model Plots</b>											

Factors with a theta value of 0 do not impact the prediction formula at all. It is as if they have been dropped from the model.

The Marginal Model plots confirm that  $\log_{10} \text{Rw}$  is a highly involved participant in Y's variation.



# Chapter 15

## Recursive Partitioning The Partition Platform

---

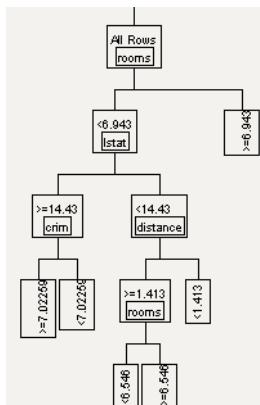
The **Partition** platform recursively partitions data according to a relationship between the  $X$  and  $Y$  values, creating a tree of partitions. It finds a set of cuts or groupings of  $X$  values that best predict a  $Y$  value. It does this by exhaustively searching all possible cuts or groupings. These splits (or *partitions*) of the data are done recursively forming a tree of decision rules until the desired fit is reached. This is a powerful platform, since it picks the optimum splits from a large number of possible splits.

The platform offers three methods for growing the final predictive tree:

- Decision Tree
- Bootstrap Forest (JMP Pro only)
- Boosted Tree (JMP Pro only)

---

**Figure 15.1** Example of Partition



# Contents

Introduction to Partitioning .....	295
Launching the Partition Platform .....	295
Partition Method.....	296
Decision Tree .....	296
Bootstrap Forest .....	306
Boosted Tree .....	309
Validation .....	313
Graphs for Goodness of Fit .....	314
Actual by Predicted Plot .....	314
ROC Curve.....	315
Lift Curves .....	317
Missing Values.....	318
Example .....	319
Decision Tree .....	320
Bootstrap Forest .....	321
Boosted Tree .....	323
Compare Methods.....	324
Statistical Details.....	326

## Introduction to Partitioning

Variations of partitioning go by many names and brand names: decision trees, CART<sup>TM</sup>, CHAID<sup>TM</sup>, C4.5, C5, and others. The technique is often taught as a data mining technique because

- it is good for exploring relationships without having a good prior model,
- it handles large problems easily, and
- the results are very interpretable.

A classic application is where you want to turn a data table of symptoms and diagnoses of a certain illness into a hierarchy of questions to ask new patients in order to make a quick initial diagnosis.

The factor columns ( $X$ 's) can be either continuous or categorical (nominal or ordinal). If an  $X$  is continuous, then the splits (partitions) are created by a *cutting value*. The sample is divided into values below and above this cutting value. If the  $X$  is categorical, then the sample is divided into two groups of levels.

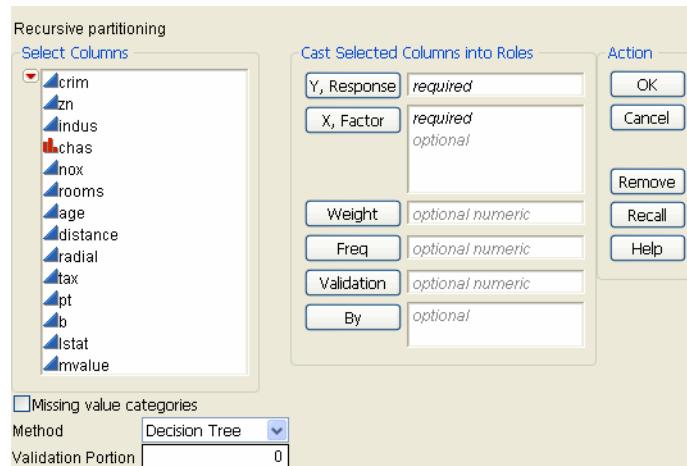
The response column ( $Y$ ) can also be either continuous or categorical (nominal or ordinal). If  $Y$  is continuous, then the platform fits means. If  $Y$  is categorical, then the fitted value is a probability. In either case, the split is chosen to maximize the difference in the responses between the two branches of the split.

For more information on split criteria, see “Statistical Details,” p. 326.

## Launching the Partition Platform

To launch the Partition platform, select **Analyze > Modeling > Partition**. The Partition launch window is shown in Figure 15.2, using the Boston Housing.jmp data table.

**Figure 15.2** Partition Launch Window



**Table 15.1** Descriptions of Launch Window

Option	Description
Y, Response	Choose the response variable.
X, Factor	Choose the predictor variables.
Validation	<p><b>Note:</b> This option is available only in JMP Pro.</p> <p>Enter a validation column here. For more information on validation, see “<a href="#">Validation</a>,” p. 313.</p>
Missing Value Categories	<p>Check this box to enable missing value categorization for nominal or ordinal predictors and responses. This option does not impact continuous predictors or responses.</p> <p>For more details on this option, or for complete details on how the Partition platform handles missing values, see “<a href="#">Missing Values</a>,” p. 318.</p>
Method	<p><b>Note:</b> This option is available only in JMP Pro.</p> <p>Select the partition method:</p> <ul style="list-style-type: none"> <li>• Decision Tree</li> <li>• Bootstrap Forest (JMP Pro only)</li> <li>• Boosted Tree (JMP Pro only)</li> </ul> <p>For more information on the methods, see “<a href="#">Partition Method</a>,” p. 296.</p>
Validation Portion	Enter the portion of the data to be used as the validation set. For more information on validation, see “ <a href="#">Validation</a> ,” p. 313.

---

## Partition Method

The Partition platform provides three methods for producing a final tree:

- For the Decision Tree method, see “[Decision Tree](#),” p. 296.
- For the Bootstrap Forest method, see “[Bootstrap Forest](#),” p. 306.
- For the Boosted Tree method, see “[Boosted Tree](#),” p. 309.

---

**Note:** Bootstrap Forest and Boosted Tree are available only in JMP Pro.

---

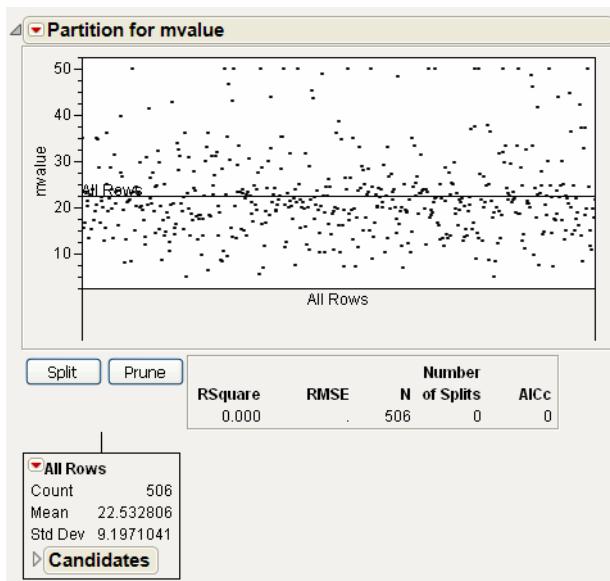
## Decision Tree

The Decision Tree method makes a single pass through the data and produces a single tree. You can interactively grow the tree one split at a time, or grow the tree automatically if validation is used.

## Decision Tree Report

As an example, use the Boston Housing.jmp data table. Launch the Partition platform and assign mvalue to the Y, Response role. Assign all the other variables to the X, Factor role. If using JMP Pro, select **Decision Tree** from the Method menu, then click **OK**. The initial report is shown in Figure 15.3.

**Figure 15.3** Decision Tree Initial Report



The **Split** button is used to partition the data, creating a tree of partitions. Repeatedly splitting the data results in branches and leaves of the tree. This can be thought of as growing the tree. The **Prune** button is used to combine the most recent split back into one group.

Note the reported statistics:

**RSquare** is the current  $R^2$  value.

**N** is the number of observations (if no Freq variable is used).

**Number of Splits** is the current number of splits.

**AICc** is the corrected Akaike's Information Criterion.

**Count** gives the number of rows in the branch.

**Mean** gives the average response for all rows in that branch.

**Std Dev** gives the standard deviation of the response for all rows in that branch.

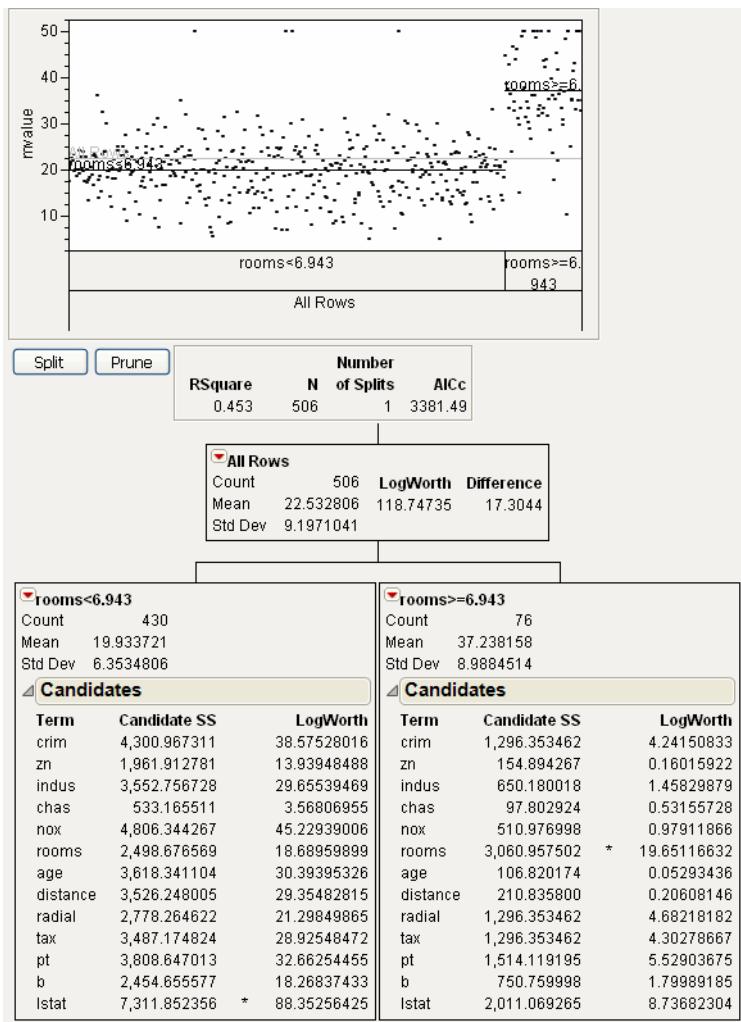
Initially, all rows are in one branch. In order to determine the first split, each  $X$  column must be considered. The candidate columns are shown in the **Candidates** report. As shown in Figure 15.4, the rooms column has the largest LogWorth, and is therefore the optimum split to make. See “[Statistical Details](#),” p. 326 for more information about LogWorth.

**Figure 15.4** Candidates Columns

All Rows		
Count	506	
Mean	22.532806	
Std Dev	9.1971041	
Candidates		
Term	Candidate SS	LogWorth
crim	8,266.17273	32.6638216
zn	6,689.06251	24.9773486
Indus	11,083.22547	48.7519537
chas	1,312.07927	4.1110954
nox	9,536.22405	39.5670978
rooms	19,339.55503 *	118.7473483
age	5,573.64765	19.8751451
distance	4,994.54054	17.1453361
radial	6,708.64333	24.6205659
tax	8,618.08428	34.5266980
pt	10,438.69478	44.8775094
b	5,259.31980	18.2910466
Istat	18,896.19401	113.7427826

The optimum split is noted by an asterisk. However, there are cases where the SS is higher for one variable, but the Logworth is higher for a different variable. In this case > and < are used to point in the best direction for each variable. The asterisk corresponds to the condition where they agree. See “[Statistical Details](#),” p. 326 for more information about LogWorth and SS.

Click the **Split** button and notice the first split was made on the column **rooms**, at a value of 6.943. Open the two new candidate reports. See Figure 15.5.

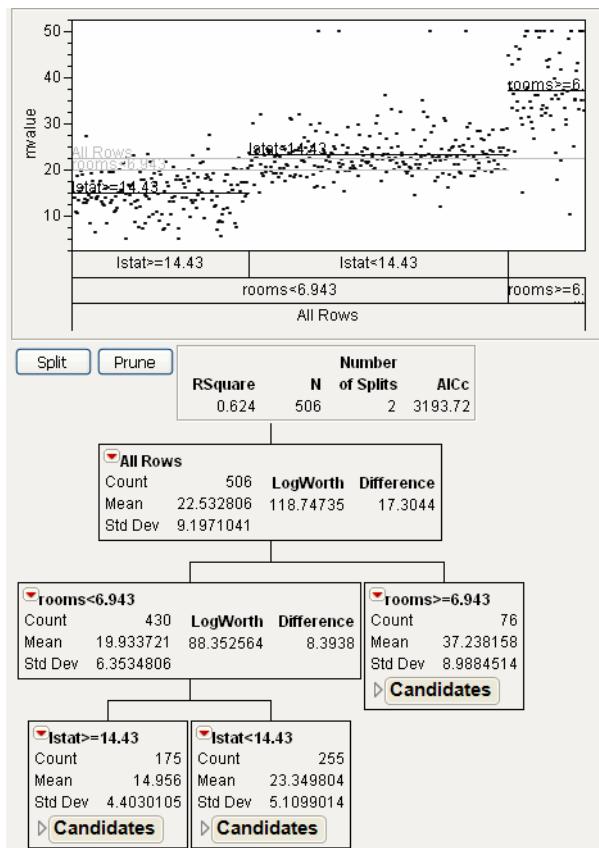
**Figure 15.5** First Split

The original 506 observations are now split into two parts:

- A left leaf, corresponding to **rooms < 6.943**, has 430 observations.
- A right leaf, corresponding to **rooms ≥ 6.943**, has 76 observations.

For the left leaf, the next split would happen on the column **Istat**, which has a SS of 7,311.85. For the right leaf, the next split would happen on the column **rooms**, which has a SS of 3,060.95. Since the SS for the left leaf is higher, using the **Split** button again will produce a split on the left leaf, on the column **Istat**.

Click the **Split** button to make the next split. See Figure 15.6.

**Figure 15.6** Second Split

The 430 observations from the previous left leaf are now split into two parts:

- A left leaf, corresponding to  $Istat \geq 14.43$ , has 175 observations.
- A right leaf, corresponding to  $Istat < 14.43$ , has 255 observations.

The 506 original observations are now split into three parts:

- A leaf corresponding to  $rooms < 6.943$  and  $Istat \geq 14.43$ .
- A leaf corresponding to  $rooms < 6.943$  and  $Istat < 14.43$ .
- A leaf corresponding to  $rooms \geq 6.943$ .

The predicted value for the observations in each leaf is the average response. The plot is divided into three sections, corresponding to the three leaves. These predicted values are shown on the plot with black lines. The points are put into random horizontal positions in each section. The vertical position is based on the response.

### Stopping Rules

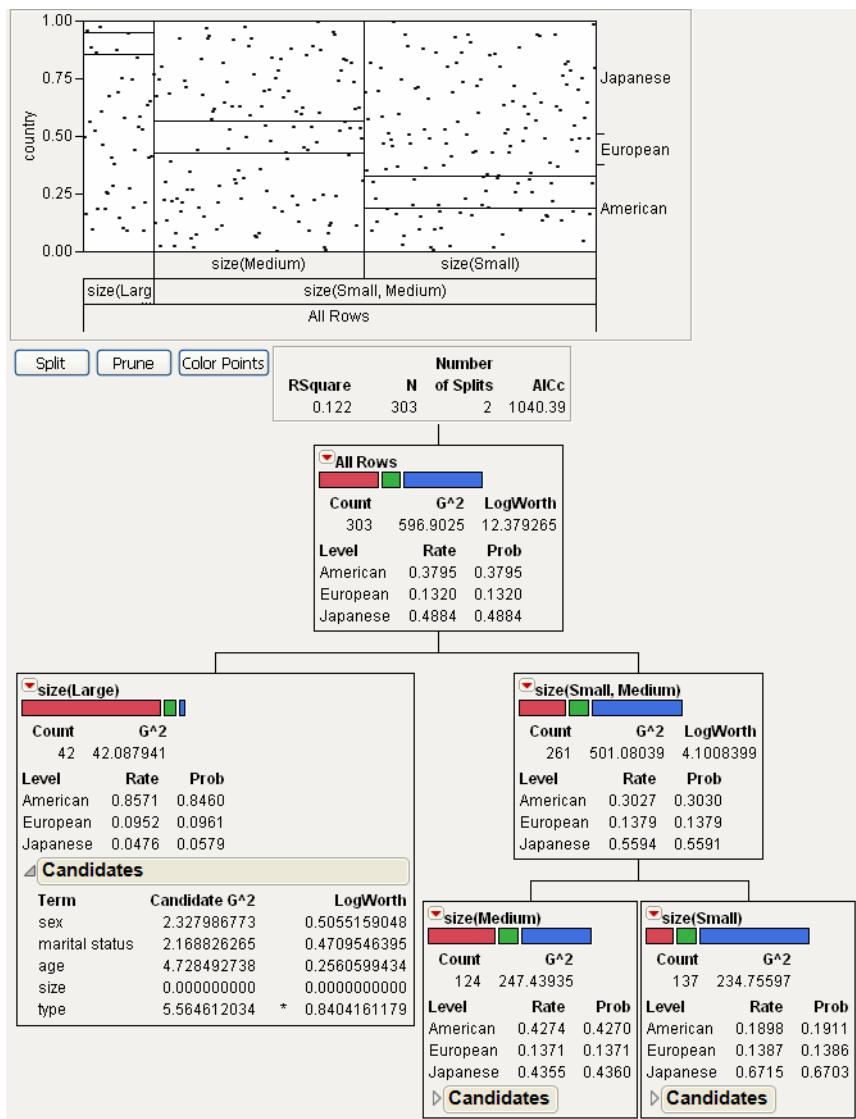
If validation is not used, the platform is purely interactive. Keep pushing the **Split** button until the result is satisfactory. Without cross-validation enabled, Partition is an exploratory platform intended to help you investigate relationships interactively.

When cross-validation is used, the user has the option to perform automatic splitting. This allows for repeated splitting without having to repeatedly click the **Split** button. See “[Automatic Splitting](#),” p. 306 for details on the stopping rule.

### Categorical Responses

When the response is categorical, the report differs in several ways (see Figure 15.7). The differences are described here, using the **Car Poll.jmp** data table:

- The  $G^2$  statistic is given instead of the Mean and Std Dev at the top of each leaf, and instead of SS in the Candidates report. See “[Statistical Details](#),” p. 326 for more information about  $G^2$ .
- The Rate statistic gives the proportion of observations in the leaf that are in each response level. The colored bars represent those proportions.
- The Prob statistic is the predicted value (a probability) for each response level. The Y axis of the plot is divided into sections corresponding to the predicted probabilities of the response levels for each leaf. The predicted probabilities always sum to one across the response levels. Random jitter is added to points in the X and Y direction in a leaf. See “[Statistical Details](#),” p. 326 for more information about the Prob statistic.
- For the plot, the vertical position is random.
- The **Color Points** button appears. This colors the points on the plot according to the response levels.

**Figure 15.7** Categorical Report

### Node Options

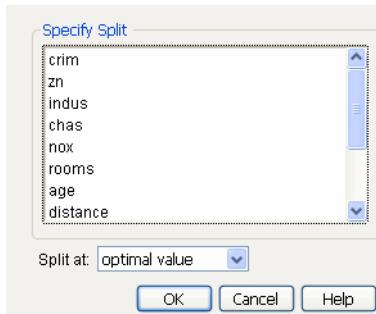
This section describes the options on the red triangle menu of each node.

**Split Best** finds and executes the best split at or below this node.

**Split Here** splits at the selected node on the best column to split by.

**Split Specific** lets you specify where a split takes place. This is useful in showing what the criterion is as a function of the cut point, as well as in determining custom cut points. After selecting this command, the following window appears.

**Figure 15.8** Window for the Split Specific Command



The **Split at** menu has the following options:

**Optimal Value** splits at the optimal value of the selected variable.

**Specified Value** allows you to specify the level where the split takes place.

**Output Split Table** produces a data table showing all possible splits and their associated split value.

**Prune Below** eliminates the splits below the selected node.

**Prune Worst** finds and removes the worst split below the selected node.

**Select Rows** selects the data table rows corresponding to this leaf. You can extend the selection by holding down the Shift key and choosing this command from another node.

**Show Details** produces a data table that shows the split criterion for a selected variable. The data table, composed of split intervals and their associated criterion values, has an attached script that produces a graph for the criterion.

**Lock** prevents a node or its subnodes from being chosen for a split. When checked, a lock icon is shown in the node title.

### Platform Options

The section describes the options on the platform red triangle menu.

**Display Options** gives a submenu consisting of items that toggle report elements on and off.

**Show Points** shows or hides the points. For categorical responses, this option shows the points or colored panels.

**Show Tree** shows or hides the large tree of partitions.

**Show Graph** shows or hides the partition graph.

**Show Split Bar** shows or hides the colored bars showing the split proportions in each leaf. This is for categorical responses only.

**Show Split Stats** shows or hides the split statistics. See “[Statistical Details](#),” p. 326 for more information on the categorical split statistic  $G^2$ .

**Show Split Prob** shows or hides the Rate and Prob statistics. This is for categorical responses only. See “[Statistical Details](#),” p. 326 for more information on Rate and Prob.

**Show Split Candidates** shows or hides the Candidates report.

**Sort Split Candidates** sorts the candidates report by the statistic or the log(worth), whichever is appropriate. This option can be turned on and off. When off, it doesn’t change any reports, but new candidate reports are sorted in the order the  $X$  terms are specified, rather than by a statistic.

**Split Best** splits the tree at the optimal split point. This is the same action as the **Split** button.

**Prune Worst** removes the terminal split that has the least discrimination ability. This is equivalent to hitting the Prune Button.

**Minimum Size Split** presents a dialog box where you enter a number or a fractional portion of the total sample size which becomes the minimum size split allowed. The default is 5. To specify a fraction of the sample size, enter a value less than 1. To specify an actual number, enter a value greater than or equal to 1.

**Lock Columns** reveals a check box table to allow you to interactively lock columns so that they are not considered for splitting. You can toggle the display without affecting the individual locks.

**Plot Actual by Predicted** produces a plot of actual values by predicted values. This is for continuous responses only.

**Small Tree View** displays a smaller version of the partition tree to the right of the scatterplot.

**Tree 3D** Shows or hides a 3D plot of the tree structure. To access this option, hold down the Shift key and click the red-triangle menu.

**Leaf Report** gives the mean and count or rates for the bottom-level leaves of the report.

**Column Contributions** brings up a report showing how each input column contributed to the fit, including how many times it was split and the total  $G^2$  or Sum of Squares attributed to that column.

**Split History** shows a plot of  $R^2$  vs. the number of splits. If you use validation, separate curves are drawn for training and validation  $R^2$ .

**K Fold Crossvalidation** shows a Crossvalidation report, giving fit statistics for both the training and folded sets. For more information on validation, see “[Validation](#),” p. 313.

**ROC Curve** is described in the section “[ROC Curve](#),” p. 315. This is for categorical responses only.

**Lift Curve** is described in the section “[Lift Curves](#),” p. 317. This is for categorical responses only.

**Show Fit Details** shows several measures of fit and a confusion matrix. The confusion matrix is a two-way classification of actual and predicted response. This is for categorical responses only.

**Entropy RSquare** compares the log-likelihoods from the fitted model and the constant probability model.

**Generalized RSquare** is a generalization of the Rsquare measure that simplifies to the regular Rsquare for continuous normal responses. It is similar to the Entropy RSquare, but instead of using the log-likelihood, it uses the  $2/n$  root of the likelihood. It is scaled to have a maximum of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model.

**Mean -Log p** is the average of  $-\log(p)$ , where  $p$  is the fitted probability associated with the event that occurred.

**RMSE** is the root mean square error, where the differences are between the response and  $p$  (the fitted probability for the event that actually occurred).

**Mean Abs Dev** is the average of the absolute values of the differences between the response and  $p$  (the fitted probability for the event that actually occurred).

**Misclassification Rate** is the rate for which the response category with the highest fitted probability is not the observed category.

For Entropy RSquare and Generalized RSquare, values closer to 1 indicate a better fit. For Mean -Log p, RMSE, Mean Abs Dev, and Misclassification Rate, smaller values indicate a better fit.

**Save Columns** is a submenu for saving model and tree results, and creating SAS code.

**Save Residuals** saves the residual values from the model to the data table.

**Save Predicteds** saves the predicted values from the model to the data table.

**Save Leaf Numbers** saves the leaf numbers of the tree to a column in the data table.

**Save Leaf Labels** saves leaf labels of the tree to the data table. The labels document each branch that the row would trace along the tree, with each branch separated by “&”. An example label could be “size(Small,Medium)&size(Small)”. However, JMP does not include redundant information in the form of category labels that are repeated. When a category label for a leaf references an inclusive list of categories in a higher tree node, JMP places a caret (^) where the tree node with redundant labels occurs. Therefore, “size(Small,Medium)&size(Small)” is presented as ^&size(Small).

**Save Prediction Formula** saves the prediction formula to a column in the data table. The formula is made up of nested conditional clauses that describe the tree structure.

**Save Leaf Number Formula** saves a column containing a formula in the data table that computes the leaf number.

**Save Leaf Label Formula** saves a column containing a formula in the data table that computes the leaf label.

**Make SAS DATA Step** creates SAS code for scoring a new data set.

**Make Tolerant SAS DATA Step** creates SAS code that can score a data set with missing values.

**Color Points** colors the points based on their response level. This is for categorical responses only, and does the same thing as the Color Points button (see “[Categorical Responses](#),” p. 301).

**Script** is the typical JMP script submenu, used to repeat the analysis or save a scripts.

**Note:** JMP 8 had a Partition Preference, and a Partition platform option, called Missing Value Rule. These have both been removed for JMP 9. The new method for handling missing values is described in “[Missing Values](#),” p. 318.

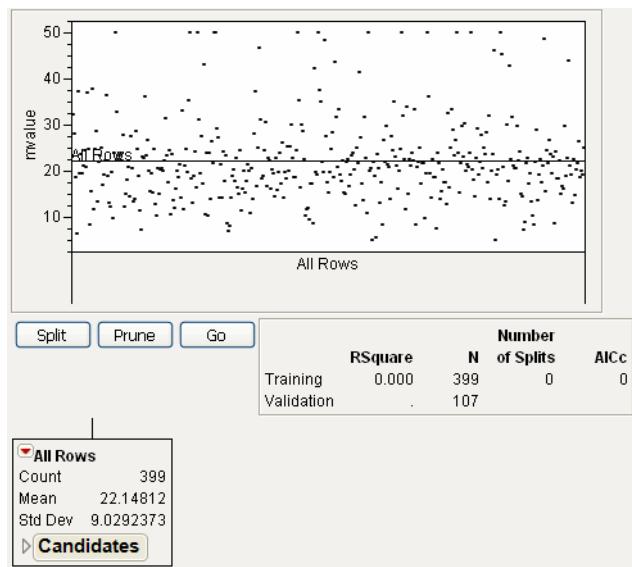
### Automatic Splitting

The **Go** button (shown in Figure 15.9) appears when you have validation enabled. For more information on using validation, see “[Validation](#),” p. 313.

The **Go** button provides for repeated splitting without having to repeatedly click the **Split** button. When you click the **Go** button, the platform performs repeated splitting until the validation R-Square is better than what the next 10 splits would obtain. This rule may produce complex trees that are not very interpretable, but have good predictive power.

Using the **Go** button turns on the **Split History** command. If using the **Go** button results in a tree with more than 40 nodes, the **Show Tree** command is turned off.

**Figure 15.9** The Go Button



## Bootstrap Forest

**Note:** The Bootstrap Forest method is available only in JMP Pro.

The Bootstrap Forest method makes many trees, and averages the predicted values to get the final predicted value. Each tree is grown on a different random sample (with replacement) of observations, and each split

on each tree considers only a random sample of candidate columns for splitting. The process can use validation to assess how many trees to grow, not to exceed the specified number of trees.

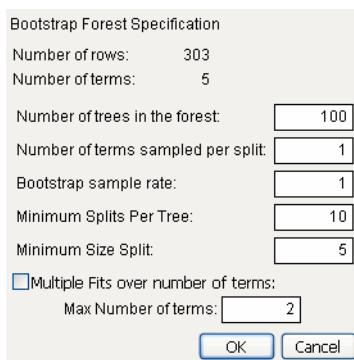
Another word for bootstrap-averaging is *bagging*. Those observations included in the growing of a tree are called the in-bag sample, abbreviated IB. Those not included are called the out-of-bag sample, abbreviated OOB.

### **Bootstrap Forest Fitting Options**

If the Bootstrap Forest method is selected on the platform launch window, the Bootstrap Forest options window appears after clicking **OK**. Figure 15.10 shows the window using the Car Poll.jmp data table. The column **sex** is used as the response, and the other columns are used as the predictors.

---

**Figure 15.10** Bootstrap Forest Fitting Options




---

The options on the Bootstrap Forest options window are described here:

**Number of rows** gives the number of observations in the data table.

**Number of terms** gives the number of columns specified as predictors.

**Number of trees in the forest** is the number of trees to grow, and then average together.

**Number of terms sampled per split** is the number of columns to consider as splitting candidates at each split. For each split, a new random sample of columns is taken as the candidate set.

**Bootstrap sample rate** is the proportion of observations to sample (with replacement) for growing each tree. A new random sample is generated for each tree.

**Minimum Splits Per Tree** is the minimum number of splits for each tree.

**Minimum Size Split** is the minimum number of observations needed on a candidate split.

**Early Stopping** is checked to perform early stopping. If checked, the process stops growing additional trees if adding more trees doesn't improve the validation statistic. If not checked, the process continues until the specified number of trees is reached. This option appears only if validation is used.

## Partition Method

**Multiple Fits over number of terms** is checked to create a bootstrap forest for several values of Number of terms sampled per split. The lower value is specified above by the Number of terms samples per split option. The upper value is specified by the following option:

**Max Number of terms** is the maximum number of terms to consider for a split.

**Bootstrap Forest Report**

The Bootstrap Forest report is shown in Figure 15.11.

**Figure 15.11** Bootstrap Forest

The screenshot shows a software interface for a Bootstrap Forest analysis. The main window title is "Bootstrap Forest for sex".

- Model Validation-Set Summaries:** Shows fit statistics for two models (N Terms = 2, 3) across various metrics: Entropy, Misclassification, Rate, Avg -Log p, RMS Error, and Avg Abs Error.
- Specifications:** Details about the model setup:
 

Target Column:	sex	Training rows:	239
		Validation rows:	64
Number of trees in the forest:	60	Test Rows:	0
Number of terms sampled per split:	3	Number of terms:	5
		Bootstrap samples:	239
		Minimum Splits Per Tree:	10
		Minimum Size Split:	5
- Overall Statistics:** Describes various performance measures:
 

Measure	Training	Validation	Definition
Entropy RSquare	0.1438	-0.008	$1 - \text{Loglike}(\text{model})/\text{LogLike(null)}$
Generalized R-Square	0.1795	-0.011	$1 - (\text{Like(null)}/\text{Like(model)})^n/(2/n)$
Mean -Log p	0.5888	0.7007	$\sum -\text{Log}(p_{ij})/n$
RMSE	0.4475	0.5029	$\sqrt{\sum (y_{ij} - p_{ij})^2/n}$
Mean Abs Dev	0.4339	0.4849	$\sum  y_{ij} - p_{ij} /n$
Misclassification Rate	0.2803	0.4688	$\sum (p_{ij} \neq p_{iMax})/n$
N	239	64	n
- Confusion Matrix:**

Actual		Predicted			
Training	Female	Male	Validation		
Female	59	48	Female	8	23
Male	19	113	Male	7	26
- Cumulative Validation**
- Per-Tree Summaries**

The results on the report are described here:

**Model Validation - Set Summaries** provides fit statistics for all the models fit if you selected the Multiple Fits option on the options window.

**Specifications** provides information on the partitioning process.

**Overall Statistics** provides fit statistics for both the training and validation sets.

**Confusion Matrix** provides two-way classifications of actual and predicted response levels for both the training and validation sets. This is available only with categorical responses.

**Cumulative Validation** provides a plot of the fit statistics versus the number of trees. The Cumulative Details report below the plot is a tabulation of the data on the plot. This is only available when validation is used.

**Per-Tree Summaries** gives summary statistics for each tree.

### Bootstrap Forest Platform Options

The Bootstrap Forest report red-triangle menu has the following options:

**Show Trees** is a submenu for displaying the Tree Views report. The report produces a picture of each component tree.

**None** does not display the Tree Views Report.

**Show names** displays the trees labeled with the splitting columns.

**Show names categories** displays the trees labeled with the splitting columns and splitting values.

**Show names categories estimates** displays the trees labeled with the splitting columns, splitting values, and summary statistics for each node.

**Plot Actual by Predicted** provides a plot of actual versus predicted values. This is only for continuous responses.

**Column Contributions** brings up a report showing how each input column contributed to the fit, including how many times it was split and the total  $G^2$  or Sum of Squares attributed to that column.

**ROC Curve** is described in the section “[ROC Curve](#),” p. 315. This is for categorical responses only.

**Lift Curve** is described in the section “[Lift Curves](#),” p. 317. This is for categorical responses only.

**Save Columns** is a submenu for saving model and tree results, and creating SAS code.

**Save Predicteds** saves the predicted values from the model to the data table.

**Save Prediction Formula** saves the prediction formula to a column in the data table.

**Save Tolerant Prediction Formula** saves the prediction formula to a column in the data. This formula can predict even with missing values.

**Save Residuals** saves the residuals to the data table. This is for continuous responses only.

**Save Cumulative Details** creates a data table containing the fit statistics for each tree.

**Make SAS DATA Step** creates SAS code for scoring a new data set.

**Make Tolerant SAS DATA Step** creates SAS code that can score a data set with missing values.

**Script** is the typical JMP script submenu, used to repeat the analysis or save a scripts.

### Boosted Tree

---

**Note:** The Boosted Tree method is available only in JMP Pro.

---

**Partition Method**

Boosting is the process of building a large, additive decision tree by fitting a sequence of smaller trees. Each of the smaller trees is fit on the scaled residuals of the previous tree. The trees are combined to form the larger final tree. The process can use validation to assess how many stages to fit, not to exceed the specified number of stages.

The tree at each stage is short, typically 1-5 splits. After the initial tree, each stage fits the residuals from the previous stage. The process continues until the specified number of stages is reached, or, if validation is used, until fitting an additional stage no longer improves the validation statistic. The final prediction is the sum of the estimates for each terminal node over all the stages.

If the response is categorical, the residuals fit at each stage are offsets of linear logits. The final prediction is a logistic transformation of the sum of the linear logits over all the stages.

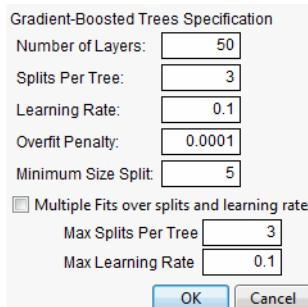
For categorical responses, only those with two levels are supported.

**Boosted Tree Fitting Options**

If the Boosted Tree method is selected on the platform launch window, the Boosted Tree options window appears after clicking **OK**. See Figure 15.12.

---

**Figure 15.12** Boosted Tree Options Window




---

The options on the Boosted Tree options window are described here:

**Number of Layers** is the maximum number of stages to include in the final tree.

**Splits per Tree** is the number of splits for each stage

**Learning Rate** is a number such that  $0 < r \leq 1$ . Learning rates close to 1 result in faster convergence on a final tree, but also have a higher tendency to overfit data. Use learning rates closer to 1 when a small Number of Layers is specified.

**Overfit Penalty** is a biasing parameter that helps to protect against fitting probabilities equal to zero.

**Minimum Size Split** is the minimum number of observations needed on a candidate split.

**Early Stopping** is checked to perform early stopping. If checked, the boosting process stops fitting additional stages if adding more stages doesn't improve the validation statistic. If not checked, the boosting process continues until the specified number of stages is reached. This option is available only if validation is used.

**Multiple Fits over splits and learning rate** is checked to create a boosted tree for every combination of Splits per Tree and Learning Rate. The lower ends of the combinations are specified by the Splits per Tree and Learning Rate options. The upper ends of the combinations are specified by the following options:

**Max Splits Per Tree** is the upper end for Splits per Tree.

**Max Learning Rate** is the upper end for Learning Rate.

### Boosted Tree Report

The Boosted Tree report is shown in Figure 15.13.

**Figure 15.13** Boosted Tree Report

The figure shows a detailed Boosted Tree report for the 'sex' column. It includes sections for Model Validation-Set Summaries, Specifications, Overall Statistics, and a Confusion Matrix, along with a Cumulative Validation section at the bottom.

**Model Validation-Set Summaries:**

		Learning	Entropy	Misclassification			
N	Splits	Rate	RSquare	Rate	Avg -Log p	RMS Error	Avg Abs Error
3	20	0.1	-0.020	0.4464	0.6986	0.5015	0.4926
4	20	0.1	0.0157	0.3425	0.6758	0.4909	0.4808
3	16	0.2	-0.045	0.4706	0.7190	0.5067	0.4910
4	12	0.2	-0.104	0.4800	0.7580	0.5206	0.4997

**Specifications:**

- Target Column: sex
- Number of training rows: 230
- Number of Layers: 20
- Number of validation rows: 73
- Splits Per Tree: 4
- Learning Rate: 0.1
- Overfit Penalty: 0.0001

**Overall Statistics:**

Measure	Training	Validation	Definition
Entropy RSquare	0.1224	0.0157	$1 - \text{Loglik}(\text{model})/\text{Loglik}(\text{null})$
Generalized R-Square	0.1555	0.0213	$1 - (\text{Like}(\text{null})/\text{Like}(\text{model}))^{(2/n)}$
Mean -Log p	0.6056	0.6758	$\sum -\text{Log}(p_{ij})/n$
RMSE	0.4568	0.4909	$\sqrt{\sum (y_{ij} - p_{ij})^2/n}$
Mean Abs Dev	0.4449	0.4808	$\sum  y_{ij} - p_{ij} /n$
Misclassification Rate	0.3261	0.3425	$\sum (p_{ij} \neq p_{\text{Max}})/n$
N	230	73	n

**Confusion Matrix:**

Actual		Predicted		Actual		Predicted		
Training	Female	Male	Validation	Female	Male	Validation	Female	Male
Female	46	60	Female	17	15	Female	17	15
Male	15	109	Male	10	31	Male	10	31

**Cumulative Validation:**

The results on the report are described here:

**Model Validation - Set Summaries** provides fit statistics for all the models fit if you selected the Multiple Splits option on the options window.

**Specifications** provides information on the partitioning process.

**Overall Statistics** provides fit statistics for both the training and validation sets.

**Confusion Matrix** provides confusion statistics for both the training and validation sets. This is available only with categorical responses.

**Cumulative Validation** provides a plot of the fit statistics versus the number of stages. The Cumulative Details report below the plot is a tabulation of the data on the plot. This is only available when validation is used.

### **Boosted Tree Platform Options**

The Boosted Tree report red-triangle menu has the following options:

**Show Trees** is a submenu for displaying the Tree Views report. The report produces a picture of the tree at each stage of the boosting process.

**None** does not display the Tree Views Report.

**Show names** displays the trees labeled with the splitting columns.

**Show names categories** displays the trees labeled with the splitting columns and splitting values.

**Show names categories estimates** displays the trees labeled with the splitting columns, splitting values, and summary statistics for each node.

**Plot Actual by Predicted** provides a plot of actual versus predicted values. This is only for continuous responses.

**Column Contributions** brings up a report showing how each input column contributed to the fit, including how many times it was split and the total  $G^2$  or Sum of Squares attributed to that column.

**ROC Curve** is described in the section “[ROC Curve](#),” p. 315. This is for categorical responses only.

**Lift Curve** is described in the section “[Lift Curves](#),” p. 317. This is for categorical responses only.

**Save Columns** is a submenu for saving model and tree results, and creating SAS code.

**Save Predicteds** saves the predicted values from the model to the data table.

**Save Prediction Formula** saves the prediction formula to a column in the data table.

**Save Tolerant Prediction Formula** saves the prediction formula to a column in the data. This formula can predict even with missing values.

**Save Residuals** saves the residuals to the data table. This is for continuous responses only.

**Save Offset Estimates** saves the offsets from the linear logits. This is for categorical responses only.

**Save Tree Details** creates a data table containing split details and estimates for each stage.

**Save Cumulative Details** creates a data table containing the fit statistics for each stage.

**Make SAS DATA Step** creates SAS code for scoring a new data set.

**Make Tolerant SAS DATA Step** creates SAS code that can score a data set with missing values.

**Script** is the typical JMP script submenu, used to repeat the analysis or save a scripts.

---

## Validation

If you grow a tree with enough splits, partitioning can overfit data. When this happens, the model predicts the fitted data very well, but predicts future observations poorly. Validation is the process of using part of a data set to estimate model parameters, and using the other part to assess the predictive ability of the model.

- The *training* set is the part that estimates model parameters.
- The *validation* set is the part that assesses or validates the predictive ability of the model.
- The *test* set is a final, independent assessment of the model's predictive ability. The test set is available only when using a validation column (see Table 15.1).

The training, validation, and test sets are created by subsetting the original data into parts. Table 15.2 describes several methods for subsetting a data set.

**Table 15.2** Validation Methods

Excluded Rows	Uses row states to subset the data. Rows that are unexcluded are used as the training set, and excluded rows are used as the validation set.  For more information about using row states and how to exclude rows, see <i>Using JMP</i> .
Holdback	Randomly divides the original data into the training and validation data sets. The <b>Validation Portion</b> (see Table 15.1) on the platform launch window is used to specify the proportion of the original data to use as the validation data set (holdback).
KFold	Divides the original data into K subsets. In turn, each of the K sets is used to validate the model fit on the rest of the data, fitting a total of K models. The model giving the best validation statistic is chosen as the final model.  KFold validation can be used only with the Decision Tree method. To use KFold, select <b>K Fold Crossvalidation</b> from the platform red-triangle menu, see “ <a href="#">Platform Options</a> ,” p. 303.  This method is best for small data sets, because it makes efficient use of limited amounts of data.

**Table 15.2** Validation Methods (*Continued*)

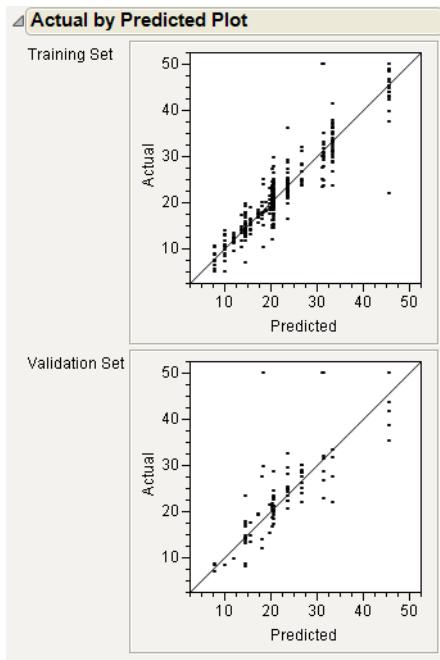
<b>Validation Column</b>	<p><b>Note:</b> The use of a validation column is available only in JMP Pro.</p> <p>Uses a column's values to divide the data into parts. The column is assigned using the Validation role on the Partition launch window. See Table 15.1.</p> <p>The column's values determine how the data is split, and what method is used for validation:</p> <ul style="list-style-type: none"> <li>• If the column's values are 0, 1 and 2, then:           <ul style="list-style-type: none"> <li>– Rows with 0 are assigned to the Training set</li> <li>– Rows with 1 are assigned to the Validation set</li> <li>– Rows with 2 are assigned to the Test set</li> </ul> </li> <li>• If the column's values are 0 and 1, then only Training and Validation sets are used.</li> </ul>
--------------------------	---

## Graphs for Goodness of Fit

The graph for goodness of fit depends on which type of response you use. The Actual by Predicted plot is for continuous responses, and the ROC Curve and Lift Curve are for categorical responses.

### Actual by Predicted Plot

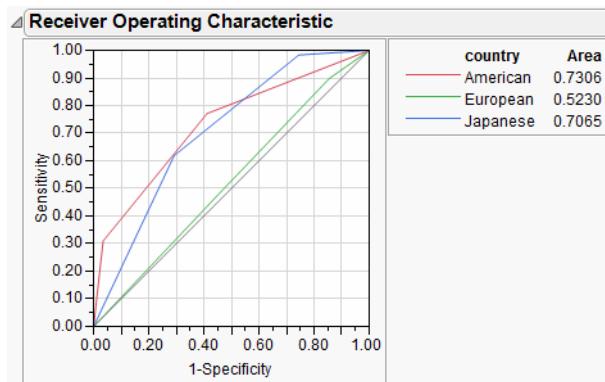
For continuous responses, the Actual by Predicted plot shows how well the model fits the data. Each leaf is predicted with its mean, so the x-coordinates are these means. The actual values form a scatter of points around each leaf mean. A diagonal line represents the locus of where predicted and actual values are the same. For a perfect fit, all the points would be on this diagonal.



## ROC Curve

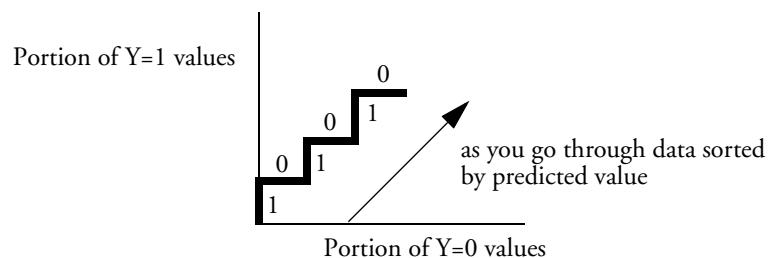
The ROC curve is for categorical responses. The classical definition of ROC curve involves the count of True Positives by False Positives as you accumulate the frequencies across a rank ordering. The True Positive  $y$ -axis is labeled “Sensitivity” and the False Positive  $X$ -axis is labeled “1-Specificity”. The idea is that if you slide across the rank ordered predictor and classify everything to the left as positive and to the right as negative, this traces the trade-off across the predictor's values.

To generalize for polytomous cases (more than 2 response levels), Partition creates an ROC curve for each response level versus the other levels. If there are only two levels, one is the diagonal reflection of the other, representing the different curves based on which is regarded as the “positive” response level.



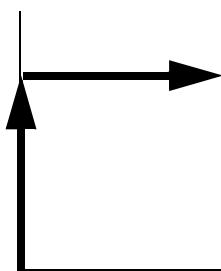
ROC curves are nothing more than a curve of the sorting efficiency of the model. The model rank-orders the fitted probabilities for a given  $Y$ -value, then starting at the lower left corner, draws the curve *up* when the row comes from that category, and to the right when the  $Y$  is another category.

In the following picture, the Y axis shows the number of  $Y=1$ 's where  $Y=1$  and the X axis shows the number of  $Y=0$ 's where  $Y=0$ .



If the model perfectly rank-orders the response values, then the sorted data has all the targeted values first, followed by all the other values. The curve moves all the way to the top before it moves at all to the right.

**Figure 15.14** ROC for Perfect Fit



If the model does not predict well, it wanders more or less diagonally from the bottom left to top right.



In practice, the curve lifts off the diagonal. The area under the curve is the indicator of the goodness of fit, with 1 being a perfect fit.

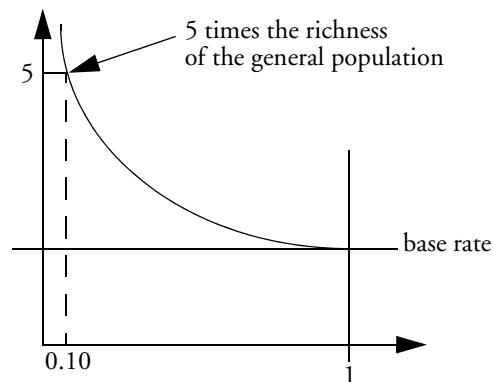
If a partition contains a section that is all or almost all one response level, then the curve lifts almost vertically at the left for a while. This means that a sample is almost completely sensitive to detecting that level. If a partition contains none or almost none of a response level, the curve at the top will cross almost horizontally for a while. This means that there is a sample that is almost completely specific to not having that response level.

Because partitions contain clumps of rows with the same (*i.e.* tied) predicted rates, the curve actually goes slanted, rather than purely up or down.

For polytomous cases, you get to see which response categories lift off the diagonal the most. In the CarPoll example above, the European cars are being identified much less than the other two categories. The American's start out with the most sensitive response (Size(Large)) and the Japanese with the most negative specific (Size(Large)'s small share for Japanese).

## Lift Curves

A lift curve shows the same information as an ROC curve, but in a way to dramatize the richness of the ordering at the beginning. The Y-axis shows the ratio of how rich that portion of the population is in the chosen response level compared to the rate of that response level as a whole. For example, if the top-rated 10% of fitted probabilities have a 25% richness of the chosen response compared with 5% richness over the whole population, the lift curve would go through the X-coordinate of 0.10 at a Y-coordinate of 25% / 5%, or 5. All lift curves reach (1,1) at the right, as the population as a whole has the general response rate.

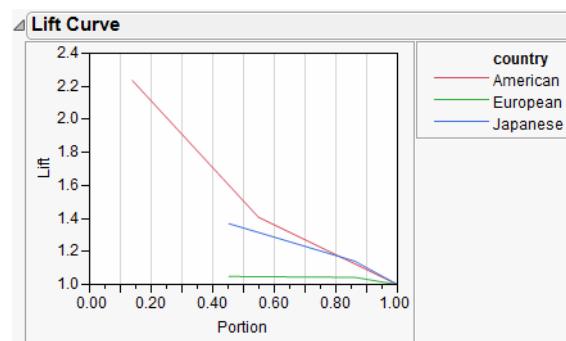


Sorted from highest predicted rate to lowest predicted rate

---

In problem situations where the response rate for a category is very low anyway (for example, a direct mail response rate), the lift curve explains things with more detail than the ROC curve.

**Figure 15.15** Lift Curve



## Missing Values

The Partition platform has methods for handling missing values in both  $Y$  and  $X$  variables.

### Missing Responses

The handling of missing values for responses depends on if the response is categorical or continuous.

### Categorical

If the Missing Value Categories checkbox (see Figure 15.2) is not selected, missing rows are not included in the analysis. If the Missing Value Categories checkbox is selected, the missing rows are entered into the analysis as another level of the variable.

### Continuous

Rows with missing values are not included in the analysis.

## Missing Predictors

The handling of missing values for predictors depends on if it is categorical or continuous.

### Categorical

If the variable is used as a splitting variable, and if the Missing Value Categories checkbox (see Figure 15.2) is not selected, then each missing row gets randomly assigned to one of the two sides of the split. When this happens using the Decision Tree method, the **Imputes** message appears showing how many times this has happened. See Figure 15.16.

If the Missing Value Categories checkbox is selected, the missing values are entered into the analysis as another level of the variable.

### Continuous

If the variable is used as a splitting variable, then each missing row gets randomly assigned to one of the two sides of the split. When this happens using the Decision Tree method, the **Imputes** message appears showing how many times this has happened. See Figure 15.16.

---

**Figure 15.16** Impute Message

RSquare	N	Number of Splits	Imputes	AICc
0.036	303	2	10	953.672

Ten observations were missing and randomly assigned to a split

---

## Example

The examples in this section use the **Boston Housing.jmp** data table. Suppose you are interested in creating a model to predict the median home value as a function of several demographic characteristics.

The Partition platform can be used to quickly assess if there is any relationship between the response and the potential predictor variables. Build a tree using all three partitioning methods, and compare the results.

---

**Note:** Results will vary because the Validation Portion option chooses rows at random to use as the training and validation sets.

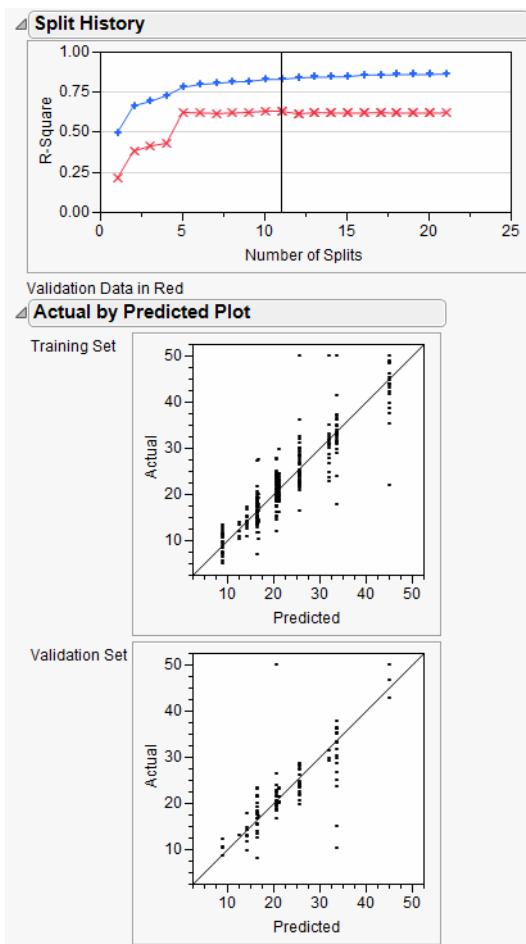
---

## Decision Tree

Follow the steps below to grow a tree using the Decision Tree method:

1. Select **Analyze > Modeling > Partition**.
2. Assign **mvalue** to the **Y, Response** role.
3. Assign the other variables (crim through lstat) to the **X, Factor** role.
4. If using JMP Pro, select the **Decision Tree** option from the Method menu. If using JMP, the Decision Tree option is the method that gets used.
5. Enter 0.2 for the **Validation Portion**.
6. Click **OK**.
7. On the platform report window, click **Go** to perform automatic splitting.
8. Select **Plot Actual by Predicted** from the platform red-triangle menu.

A portion of the report is shown in Figure 15.17.

**Figure 15.17** Decision Tree Results

## Bootstrap Forest

Follow the steps below to grow a tree using the Bootstrap Forest method:

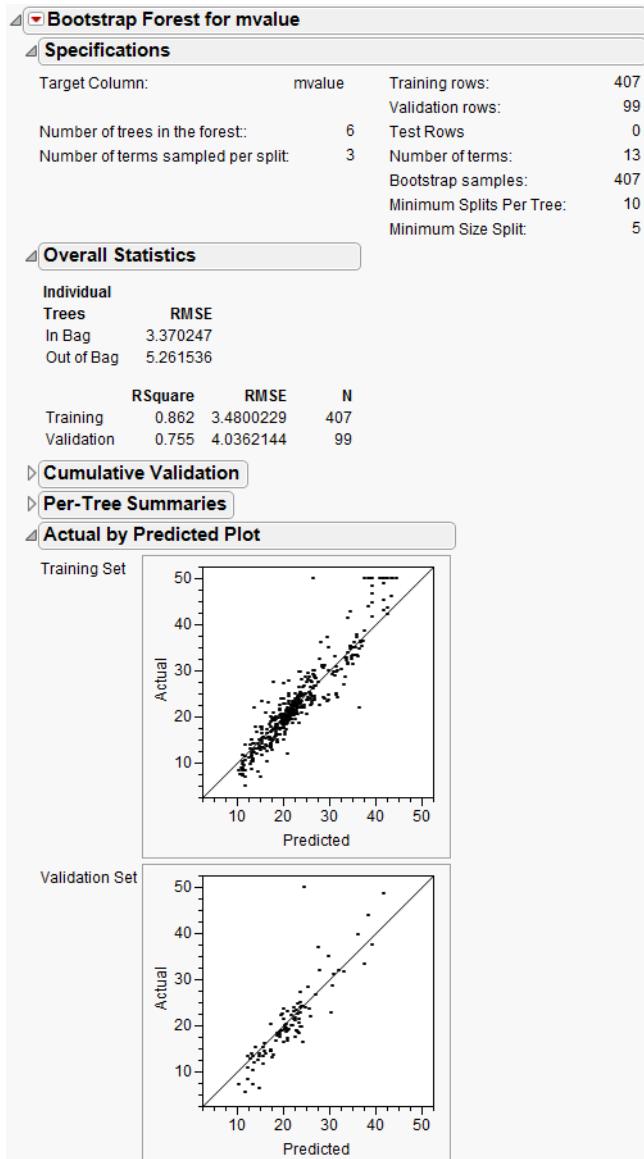
1. Select **Analyze > Modeling > Partition**.
2. Assign mvalue to the **Y, Response** role.
3. Assign the other variables (crim through lstat) to the **X, Factor** role.
4. Select **Bootstrap Forest** from the Method menu.
5. Enter 0.2 for the **Validation Portion**.
6. Click **OK**.
7. Check the **Early Stopping** option on the Bootstrap Forest options window.

8. Click **OK**.

9. Select **Plot Actual by Predicted** from the red-triangle menu.

The report is shown in Figure 15.18.

**Figure 15.18** Bootstrap Forest Report

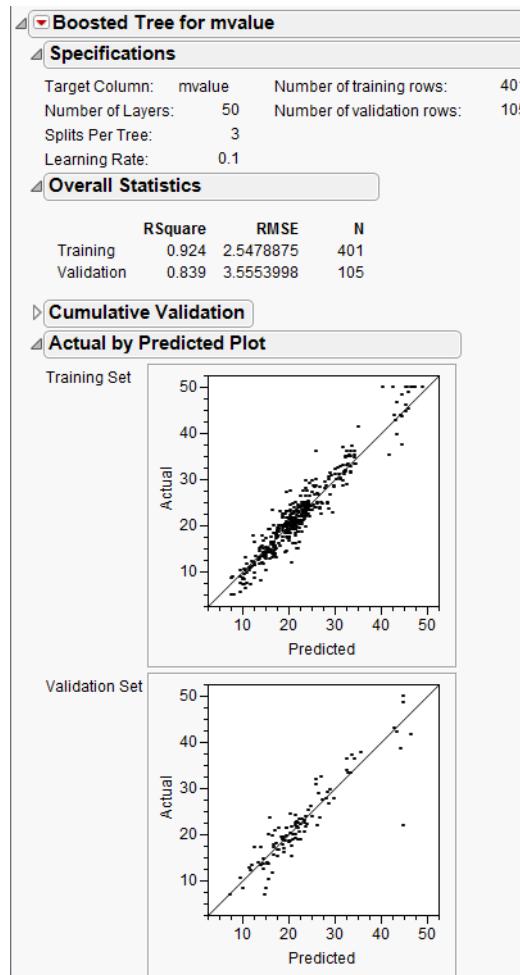


## Boosted Tree

Follow the steps below to grow a tree using the Boosted Tree method:

1. Select **Analyze > Modeling > Partition**.
2. Assign mvalue to the **Y, Response** role.
3. Assign the other variables (crim through lstat) to the **X, Factor** role.
4. Select **Boosted Tree** from the Method menu.
5. Enter 0.2 for the **Validation Portion**.
6. Click **OK**.
7. Check the **Early Stopping** option on the Boosted Tree options window.
8. Click **OK**.
9. Select **Plot Actual by Predicted** on the red-triangle menu.

The report is shown in Figure 15.19.

**Figure 15.19** Boosted Tree Report

## Compare Methods

The Decision Tree method produced a tree with 7 splits. The Bootstrap Forest method produced a final tree with 20 component trees. The Boosted Tree method produced a final tree with 50 component trees.

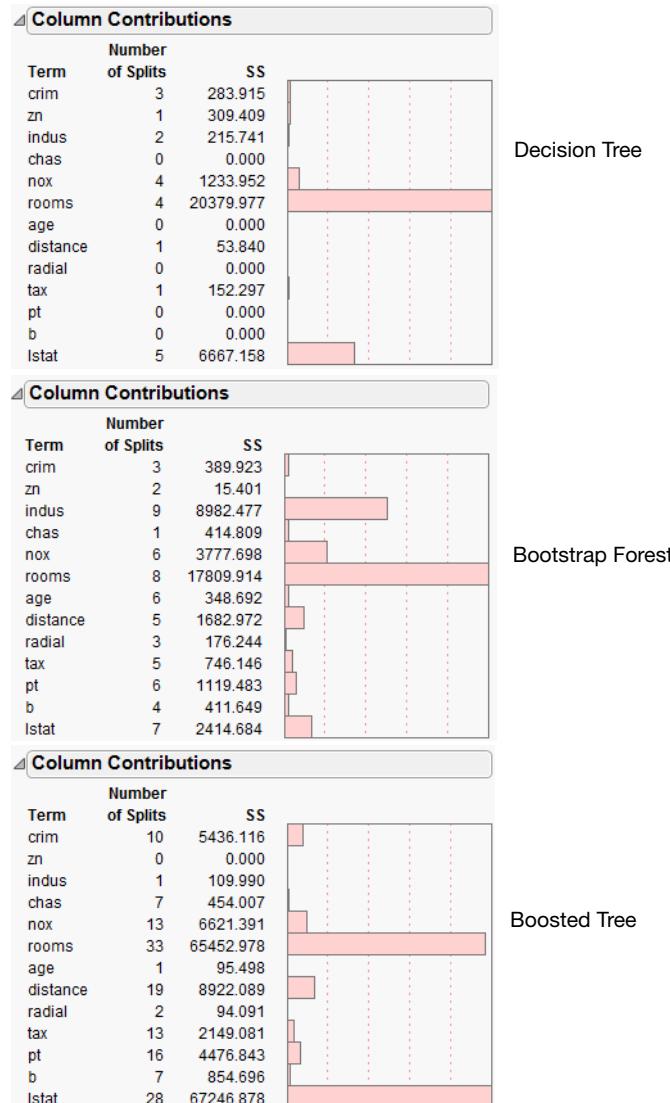
The Validation R-Square from the three methods are different, with the Boosted Tree having the best:

- Decision Tree 0.647
- Bootstrap Forest 0.716
- Boosted Tree 0.847

The actual-by-predicted plots show that the Boosted Tree method is best at predicting the actual median home values. The points on the plot are closer to the line for the Boosted Tree method.

Figure 15.20 shows a summary of the **Column Contributions** report from each method.

**Figure 15.20** Column Contributions



All three methods indicate that rooms and Istat are the most important variables for predicting median home value.

## Statistical Details

This section provides some quantitative details and other information.

### General

The response can be either continuous, or categorical (nominal or ordinal). If  $Y$  is categorical, then it is fitting the probabilities estimated for the response levels, minimizing the residual log-likelihood chi-square [ $2^* \text{entropy}$ ]. If the response is continuous, then the platform fits means, minimizing the sum of squared errors.

The factors can be either continuous, or categorical (nominal or ordinal). If an  $X$  is continuous, then the partition is done according to a splitting “cut” value for  $X$ . If  $X$  is categorical, then it divides the  $X$  categories into two groups of levels and considers all possible groupings into two levels.

### Splitting Criterion

Node splitting is based on the LogWorth statistic, which is reported in node Candidate reports. LogWorth is calculated as:

$$-\log_{10}(p\text{-value})$$

where the adjusted  $p$ -value is calculated in a complex manner that takes into account the number of different ways splits can occur. This calculation is very fair compared to the unadjusted  $p$ -value, which favors  $X$ 's with many levels, and the Bonferroni  $p$ -value, which favors  $X$ 's with small numbers of levels. Details on the method are discussed in a white paper “Monte Carlo Calibration of Distributions of Partition Statistics” found on the jmp website [www.jmp.com](http://www.jmp.com).

For continuous responses, the Sum of Squares (SS) is reported in node reports. This is the change in the error sum-of-squares due to the split.

A candidate SS that has been chosen is

$$SS_{\text{test}} = SS_{\text{parent}} - (SS_{\text{right}} + SS_{\text{left}}) \text{ where } SS \text{ in a node is just } s^2(n - 1).$$

Also reported for continuous responses is the Difference statistic. This is the difference between the predicted values for the two child nodes of a parent node.

For categorical responses, the  $G^2$  (likelihood-ratio chi-square) is shown in the report. This is actually twice the [natural log] entropy or twice the change in the entropy. Entropy is  $\Sigma -\log(p)$  for each observation, where  $p$  is the probability attributed to the response that occurred.

A candidate  $G^2$  that has been chosen is

$$G^2_{\text{test}} = G^2_{\text{parent}} - (G^2_{\text{left}} + G^2_{\text{right}}).$$

When Partition calculates a  $G^2$  or  $R^2$  on excluded data for a categorical response, it uses the rate value 0.25/ $m$  when it encounters a zero rate in a group with  $m$  rows. Otherwise, a missing statistic would be reported, since the logarithm of zero is undefined.

### Predicted Probabilities in Decision Tree and Bootstrap Forest

The predicted probabilities for the Decision Tree and Bootstrap Forest methods are calculated as described below by the Prob statistic.

For categorical responses in Decision Tree, the Show Split Prob command shows the following statistics:

**Rate** is the proportion of observations at the node for each response level.

**Prob** is the predicted probability for that node of the tree. The method for calculating Prob for the  $i^{\text{th}}$  response level at a given node is as follows:

$$\text{Prob}_i = \frac{n_i + \text{prior}_i}{\sum(n_i + \text{prior}_i)}$$

where the summation is across all response levels,  $n_i$  is the number of observations at the node for the  $i^{\text{th}}$  response level, and  $\text{prior}_i$  is the prior probability for the  $i^{\text{th}}$  response level, calculated as

$$\text{prior}_i = \lambda p_i + (1-\lambda)P_i$$

where  $p_i$  is the  $\text{prior}_i$  from the parent node,  $P_i$  is the  $\text{Prob}_i$  from the parent node, and  $\lambda$  is a weighting factor currently set at 0.9.

The estimate, Prob, is the same that would be obtained for a Bayesian estimate of a multinomial probability parameter with a conjugate Dirichlet prior.

The method for calculating Prob assures that the predicted probabilities are always non-zero.



# Chapter 16

## Time Series Analysis

### The Time Series Platform

---

The Time Series platform lets you explore, analyze, and forecast univariate time series. A time series is a set  $y_1, y_2, \dots, y_N$  of observations taken over a series of equally-spaced time periods. The analysis begins with a plot of the points in the time series. In addition, the platform displays graphs of the autocorrelations and partial autocorrelations of the series. These indicate how and to what degree each point in the series is correlated with earlier values in the series. You can interactively add:

**Variograms** a characterization of process disturbances

**AR coefficients** autoregressive coefficients

**Spectral Density Plots** versus period and frequency, with white noise tests.

These graphs can be used to identify the type of model appropriate for describing and predicting (forecasting) the evolution of the time series. The model types include:

**ARIMA** autoregressive integrated moving-average, often called Box-Jenkins models

**Seasonal ARIMA** ARIMA models with a seasonal component

**Smoothing Models** several forms of exponential smoothing and Winter's method

**Transfer Function Models** for modeling with input series.

---

**Note:** The Time Series Launch dialog requires that one or more continuous variables be assigned as the time series. Optionally, you can specify a time ID variable, which is used to label the time axis, or one or more input series. If a time ID variable is specified, it must be continuous, sorted ascending, and without missing values.

---

# Contents

Launch the Platform . . . . .	331
Time Series Commands . . . . .	332
Graph . . . . .	333
Autocorrelation . . . . .	333
Partial Autocorrelation . . . . .	333
Variogram . . . . .	334
AR Coefficients . . . . .	334
Spectral Density . . . . .	335
Save Spectral Density . . . . .	336
Number of Forecast Periods . . . . .	337
Difference . . . . .	337
Modeling Reports . . . . .	338
Model Comparison Table . . . . .	338
Model Summary Table . . . . .	339
Parameter Estimates Table . . . . .	341
Forecast Plot . . . . .	342
Residuals . . . . .	342
Iteration History . . . . .	342
Model Report Options . . . . .	343
ARIMA Model . . . . .	343
Seasonal ARIMA . . . . .	345
ARIMA Model Group . . . . .	345
Transfer Functions . . . . .	346
Report and Menu Structure . . . . .	346
Diagnostics . . . . .	348
Model Building . . . . .	349
Transfer Function Model . . . . .	350
Model Reports . . . . .	352
Model Comparison Table . . . . .	354
Fitting Notes . . . . .	354
Smoothing Models . . . . .	354

---

## Launch the Platform

To begin a time series analysis, choose the **Time Series** command from the **Analyze > Modeling** submenu to display the Time Series Launch dialog (Figure 16.1). This dialog allows you to specify the number of lags to use in computing the autocorrelations and partial autocorrelations. It also lets you specify the number of future periods to forecast using each model fitted to the data. After you select analysis variables and click **OK** on this dialog, a platform launches with plots and accompanying text reports for each of the time series (*Y*) variables you specified.

### Select Columns into Roles

You assign columns for analysis with the dialog in Figure 16.1. The selector list at the left of the dialog shows all columns in the current table. To cast a column into a role, select one or more columns in the column selector list and click a role button. Or, drag variables from the column selector list to one of the following role boxes:

**X, Time ID** for the *x*-axis, one variable used for labeling the time axis

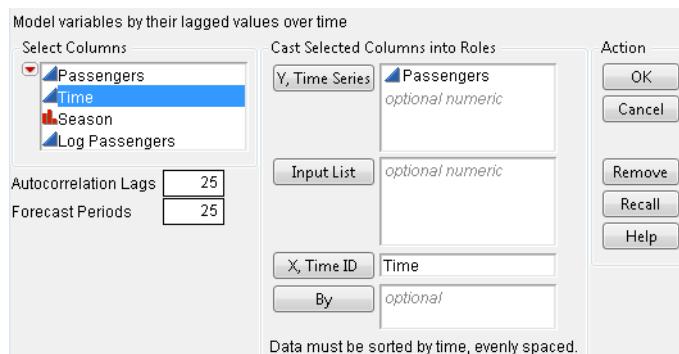
**Y, Time Series** for the *y*-axis, one or more time series variables.

If you use a **X, Time ID** variable, you can specify the time frequency by using the **Time Frequency** column property. The choices are **Annual**, **Monthly**, **Weekly**, **Daily**, **Hourly**, **Minute**, and **Second**. This lets JMP take things like leap years and leap days into account. If no frequency is specified, the data is treated as equally spaced numeric data.

To remove an unwanted variable from an assigned role, select it in the role box and click **Remove**. After assigning roles, click **OK** to see the analysis for each time series variable versus the time ID.

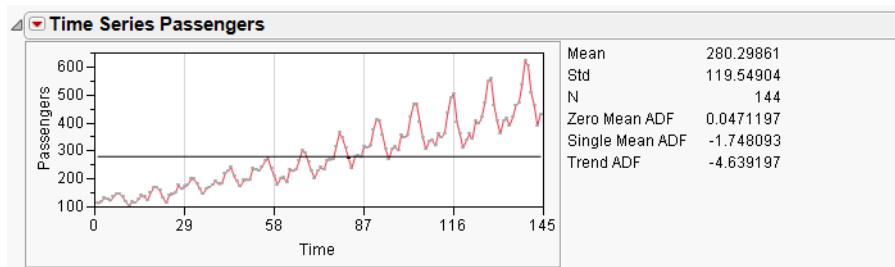
You set the number of lags for the autocorrelation and partial autocorrelation plots in the **Autocorrelation Lags** box. This is the maximum number of periods between points used in the computation of the correlations. It must be more than one but less than the number of rows. A commonly used rule of thumb for the maximum number of lags is  $n/4$ , where  $n$  is the number of observations. The **Forecast Periods** box allows you to set the number of periods into the future that the fitted models are forecast. By default, JMP uses 25 lags and 25 forecast periods.

The data for the next examples are in the *Seriesg.jmp* table found in the Time Series sample data folder (Box and Jenkins 1976). The time series variable is **Passengers** and the Time ID is **Time**.

**Figure 16.1** The Time Series Launch Dialog

## The Time Series Graph

The Time Series platform begins with a plot of each times series by the time ID, or row number if no time ID is specified (Figure 16.2). The plot, like others in JMP, has features to resize the graph, highlight points with the cursor or brush tool, and label points. See the *Using JMP* for a discussion of these features.

**Figure 16.2** Time Series Plot of Seriesg (Airline Passenger) Data

If you open **Time Series Basic Diagnostic Tables**, graphs of the autocorrelation and partial autocorrelation (Figure 16.3) of the time series are shown.

The platform popup menu, discussed next, also has fitting commands and options for displaying additional graphs and statistical tables.

---

## Time Series Commands

The platform red-triangle menu has the options described in the following sections.

## Graph

The Time Series platform begins by showing a time series plot, like the one shown previously in Figure 16.2. The **Graph** command on the platform popup menu has a submenu of controls for the time series plot with the following commands.

**Time Series Graph** hides or displays the time series graph.

**Show Points** hides or displays the points in the time series graph.

**Connecting Lines** hides or displays the lines connecting the points in the time series graph.

**Mean Line** hides or displays a horizontal line in the time series graph that depicts the mean of the time series.

## Autocorrelation

The **Autocorrelation** command alternately hides or displays the autocorrelation graph of the sample, often called the *sample autocorrelation function*. This graph describes the correlation between all the pairs of points in the time series with a given separation in time or lag. The autocorrelation for the  $k$ th lag is

$$r_k = \frac{c_k}{c_0} \text{ where } c_k = \frac{1}{N} \sum_{t=k+1}^N (y_t - \bar{y})(y_{t-k} - \bar{y})$$

where  $\bar{y}$  is the mean of the  $N$  nonmissing points in the time series. The bars graphically depict the autocorrelations.

By definition, the first autocorrelation (lag 0) always has length 1. The curves show twice the large-lag standard error ( $\pm 2$  standard errors), computed as

$$\text{SE}_k = \sqrt{\frac{1}{N} \left( 1 + 2 \sum_{i=1}^{k-1} r_i^2 \right)}$$

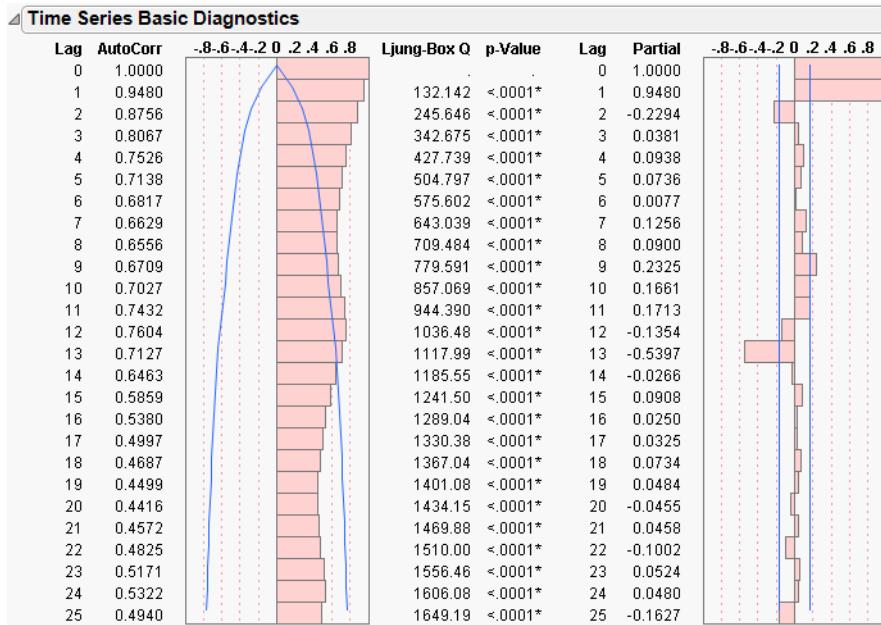
The autocorrelation plot for the `Seriesg` data is shown on the left in Figure 16.3. You can examine the autocorrelation and partial autocorrelations plots to determine whether the time series is stationary (meaning it has a fixed mean and standard deviation over time) and what model might be appropriate to fit the time series.

In addition, the Ljung-Box  $Q$  and  $p$ -values are shown for each lag. The  $Q$ -statistic is used to test whether a group of autocorrelations is significantly different from zero or to test that the residuals from a model can be distinguished from white-noise.

## Partial Autocorrelation

The **Partial Autocorrelation** command alternately hides or displays the graph of the sample partial autocorrelations. The plot on the right in Figure 16.3 shows the partial autocorrelation function for the `Seriesg` data. The solid blue lines represent  $\pm 2$  standard errors for approximate 95% confidence limits, where the standard error is computed

$$\text{SE}_k = \frac{1}{\sqrt{n}} \text{ for all } k$$

**Figure 16.3** Autocorrelation and Partial Correlation Plots

## Variogram

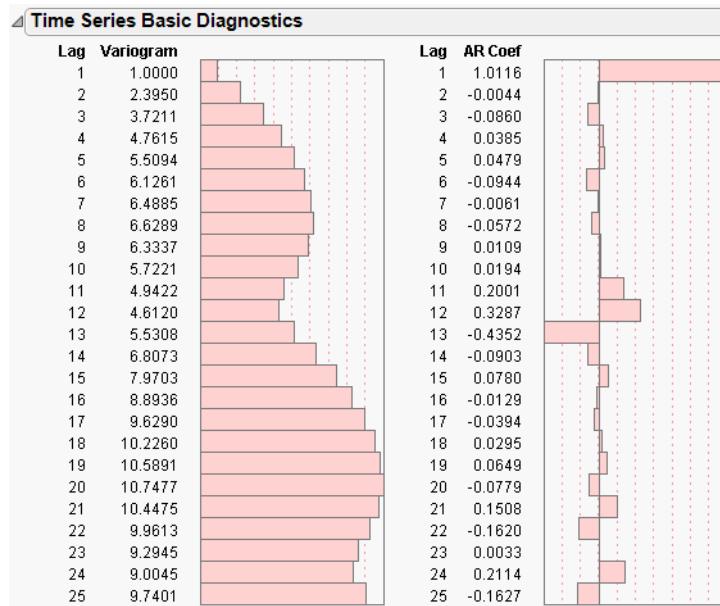
The **Variogram** command alternately displays or hides the graph of the variogram. The variogram measures the variance of the differences of points  $k$  lags apart and compares it to that for points one lag apart. The variogram is computed from the autocorrelations as

$$V_k = \frac{1 - r_{k+1}}{1 - r_1}$$

where  $r_k$  is the autocorrelation at lag  $k$ . The plot on the left in Figure 16.4 shows the Variogram graph for the Seriesg data.

## AR Coefficients

The **AR Coefficients** command alternately displays or hides the graph of the least squares estimates of the autoregressive (AR) coefficients. The definition of these coefficients is given below. These coefficients approximate those that you would obtain from fitting a high-order, purely autoregressive model. The right-hand graph in Figure 16.4 shows the AR coefficients for the Seriesg data.

**Figure 16.4** Variogram Graph (left) and AR Coefficient Graph (right)

## Spectral Density

The **Spectral Density** command alternately displays or hides the graphs of the spectral density as a function of period and frequency (Figure 16.5).

The least squares estimates of the coefficients of the Fourier series

$$a_t = \frac{2}{N} \sum_{i=1}^N y_t \cos(2\pi f_i t)$$

and

$$b_t = \frac{2}{N} \sum_{i=1}^N y_t \sin(2\pi f_i t)$$

where  $f_i = i/N$  are combined to form the periodogram  $I(f_i) = \frac{N}{2}(a_i^2 + b_i^2)$ , which represents the intensity at frequency  $f_i$ .

The periodogram is smoothed and scaled by  $1/(4\pi)$  to form the spectral density.

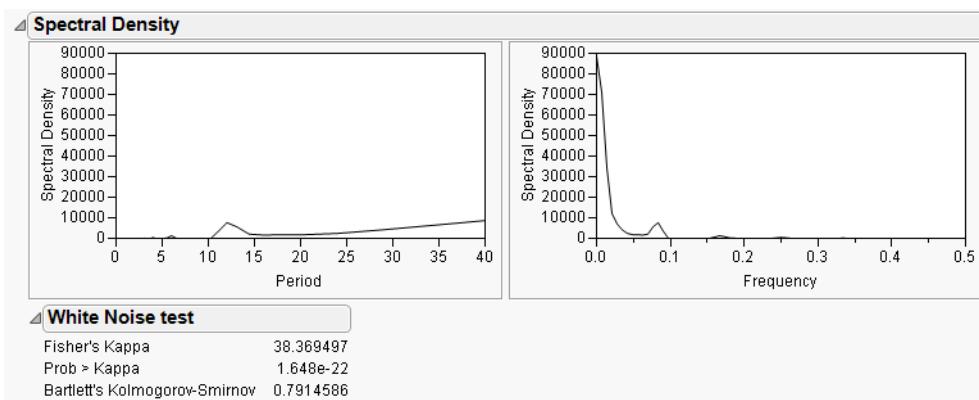
The *Fisher's Kappa* statistic tests the null hypothesis that the values in the series are drawn from a normal distribution with variance 1 against the alternative hypothesis that the series has some periodic component. Kappa is the ratio of the maximum value of the periodogram,  $I(f)$ , and its average value. The probability of observing a larger Kappa if the null hypothesis is true is given by

$$(k > \kappa) = 1 - \sum_{j=0}^q (-1)^j \binom{q}{j} \left[ \max\left(1 - \frac{jk}{q}, 0\right) \right]^{q-1}$$

where  $q = N / 2$  if  $N$  is even,  $q = (N - 1) / 2$  if  $N$  is odd, and  $\kappa$  is the observed value of Kappa. The null hypothesis is rejected if this probability is less than the significance level  $\alpha$ .

For  $q - 1 > 100$ , *Bartlett's Kolmogorov-Smirnov* compares the normalized cumulative periodogram to the cumulative distribution function of the uniform distribution on the interval  $(0, 1)$ . The test statistic equals the maximum absolute difference of the cumulative periodogram and the uniform CDF. If it exceeds  $a / (\sqrt{q})$ , then reject the hypothesis that the series comes from a normal distribution. The values  $a = 1.36$  and  $a = 1.63$  correspond to significance levels 5% and 1% respectively.

**Figure 16.5** Spectral Density Plots



## Save Spectral Density

**Save Spectral Density** creates a new table containing the spectral density and periodogram where the  $(i+1)$ th row corresponds to the frequency  $f_i = i / N$  (that is, the  $i$ th harmonic of  $1 / N$ ).

The new data table has these columns:

**Period** is the period of the  $i$ th harmonic,  $1 / f_i$

**Frequency** is the frequency of the harmonic,  $f_i$

**Angular Frequency** is the angular frequency of the harmonic,  $2\pi f_i$

**Sine** is the Fourier sine coefficients,  $a_i$

**Cosine** is the Fourier cosine coefficients,  $b_i$

**Periodogram** is the periodogram,  $I(f_i)$

**Spectral Density** is the spectral density, a smoothed version of the periodogram.

## Number of Forecast Periods

The **Number of Forecast Periods** command displays a dialog for you to reset the number of periods into the future that the fitted models will forecast. The initial value is set in the Time Series Launch dialog. All existing and future forecast results will show the new number of periods with this command.

## Difference

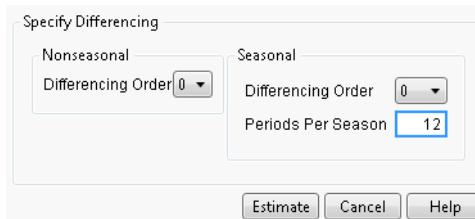
Many time series do not exhibit a fixed mean. Such nonstationary series are not suitable for description by some time series models such as those with only autoregressive and moving average terms (ARMA models). However, these series can often be made stationary by differencing the values in the series. The differenced series is given by

$$w_t = (1 - B)^d (1 - B^s)^D y_t$$

where  $t$  is the time index and  $B$  is the backshift operator defined by  $B y_t = y_{t-1}$ .

The **Difference** command computes the differenced series and produces graphs of the autocorrelations and partial autocorrelations of the differenced series. These graphs can be used to determine if the differenced series is stationary.

Several of the time series models described in the next sections accommodate a differencing operation (the ARIMA, Seasonal ARIMA models, and some of the smoothing models). The **Difference** command is useful for determining the order of differencing that should be specified in these models.




---

The Specify Differencing dialog appears in the report window when you select the Difference command. It allows you to specify the differencing operation you want to apply to the time series. Click **Estimate** to see the results of the differencing operation. The Specify Differencing dialog allows you to specify the Nonseasonal Differencing Order,  $d$ , the Seasonal Differencing Order,  $D$ , and the number of Periods Per Season,  $s$ . Selecting zero for the value of the differencing order is equivalent to no differencing of that kind.

The red triangle menu on the Difference plot has the following options:

**Graph** controls the plot of the differenced series and behaves the same as those under the Time Series Graph menu.

**Autocorrelation** alternately displays or hides the autocorrelation of the differenced series.

**Partial Autocorrelation** alternately hides or displays the partial autocorrelations of differenced series.

**Variogram** alternately hides or displays the variogram of the differenced series.

**Save** appends the differenced series to the original data table. The leading  $d + sD$  elements are lost in the differencing process. They are represented as missing values in the saved series.

## Modeling Reports

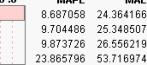
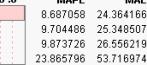
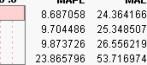
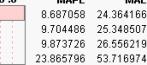
The time series modeling commands are used to fit theoretical models to the series and use the fitted model to predict (forecast) future values of the series. These commands also produce statistics and residuals that allow you to ascertain the adequacy of the model you have elected to use. You can select the modeling commands repeatedly. Each time you select a model, a report of the results of the fit and a forecast is added to the platform results.

The fit of each model begins with a dialog that lets you specify the details of the model being fit as well as how it will be fit. Each general class of models has its own dialog, as discussed in their respective sections. The models are fit by maximizing the likelihood function, using a Kalman filter to compute the likelihood function. The ARIMA, seasonal ARIMA, and smoothing models begin with the following report tables.

## Model Comparison Table

Figure 16.6 shows the Model Comparison Report.

**Figure 16.6** Model Comparison

Report	Graph	Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	Weights	2, 4, 6, 8	MAPE	MAE
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ARIMA(1,1,1)	140	979.9437	1394.1215	1403.0101	0.932	1389.1215	0.999902		8.607059	24.364166
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ARMA(1,1)	141	998.13035	1407.7493	1416.6577	0.919	1401.7493	0.001088		9.704406	25.349507
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AR(1)	142	1134.3871	1426.1794	1432.1190	0.809	1422.1794	0.000000		9.873726	26.556219
<input checked="" type="checkbox"/>	<input type="checkbox"/>	MA(1)	142	4264.6121	1616.8626	1622.8022	0.693	1612.8626	0.000000		23.865796	53.716974

The Model Comparison table summarizes the fit statistics for each model. You can use it to compare several models fitted to the same time series. Each row corresponds to a different model. The models are sorted by the AIC statistic. The Model Comparison table shown above summarizes the ARIMA models (1, 0, 0), (0, 0, 1), (1, 0, 1), and (1, 1, 1) respectively. Use the **Report** checkbox to show or hide the Model Report for a model.

The Model Comparison report has red-triangle menus for each model, with the following options:

**Fit New** opens a window giving the settings of the model. You can change the settings to fit a different model.

**Simulate Once** provides one simulation of the model out  $k$  time periods. The simulation is shown on the Model Comparison time series plot. To change  $k$ , use the Number of Forecast Periods option on the platform red-triangle menu.

**Simulate More** provides the specified number of simulations of the model out  $k$  time periods. The simulations are shown on the Model Comparison time series plot. To change  $k$ , use the Number of Forecast Periods option on the platform red-triangle menu.

**Remove Model Simulation** removes the simulations for the given model.

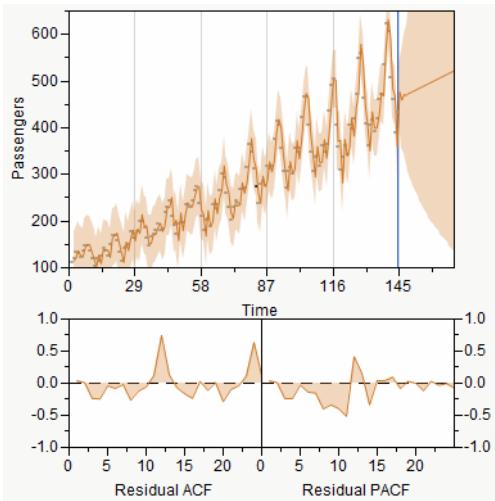
**Remove All Simulation** removes the simulations for all models.

**Generate Simulation** generates simulations for the given model, and stores the results in a data table. You specify the random seed, number of simulations, and the number of forecast periods.

**Set Seed** is used to specify the seed for generating the next forecasts.

The Model Comparison report provides plots for a model when the **Graph** checkbox is selected. Figure 16.7 shows the plots for the ARIMA(1,1,1) model.

**Figure 16.7** Model Plots



The top plot is a time series plot of the data, forecasts, and confidence limits. Below that are plots of the autocorrelation and partial autocorrelation functions.

## Model Summary Table

Each model fit generates a Model Summary table, which summarizes the statistics of the fit. In the formulae below,  $n$  is the length of the series and  $k$  is the number of fitted parameters in the model.

**DF** is the number of degrees of freedom in the fit,  $n - k$ .

**Sum of Squared Errors** is the sum of the squared residuals.

Model: ARIMA(1, 1, 1)			
Model Summary			
DF	140	Stable	Yes
Sum of Squared Errors	137052.119	Invertible	Yes
Variance Estimate	978.943704		
Standard Deviation	31.2880761		
Akaike's 'A' Information Criterion	1394.12154		
Schwarz's Bayesian Criterion	1403.01008		
RSquare	0.93245111		
RSquare Adj	0.93148612		
MAPE	8.68705796		
MAE	24.3641658		
-2LogLikelihood	1388.12154		

**Variance Estimate** the unconditional sum of squares (SSE) divided by the number of degrees of freedom,  $\text{SSE} / (n - k)$ . This is the sample estimate of the variance of the random shocks  $a_t$ , described in the section “ARIMA Model,” p. 343.

**Standard Deviation** is the square root of the variance estimate. This is a sample estimate of the standard deviation of  $a_t$ , the random shocks.

**Akaike's Information Criterion [AIC], Schwarz's Bayesian Criterion [SBC or BIC]** Smaller values of these criteria indicate better fit. They are computed:

$$\text{AIC} = -2\log\text{likelihood} + 2k$$

$$\text{SBC} = -2\log\text{likelihood} + k\ln(n)$$

**RSquare** RSquare is computed

$$1 - \frac{\text{SSE}}{\text{SST}}$$

where  $\text{SST} = \sum_{i=1}^n (y_i - \bar{y})^2$  and  $\text{SSE} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$ ,  $\hat{y}_i$  are the one-step-ahead forecasts, and  $\bar{y}$  is the mean  $y_i$ .

If the model fits the series badly, the model error sum of squares, SSE might be larger than the total sum of squares, SST and  $R^2$  will be negative.

**RSquare Adj** The adjusted  $R^2$  is

$$1 - \left[ \frac{(n-1)}{(n-k)} (1 - R^2) \right]$$

**MAPE** is the Mean Absolute Percentage Error, and is computed

$$\frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

**MAE** is the Mean Absolute Error, and is computed

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

**-2LogLikelihood** is minus two times the natural log of the likelihood function evaluated at the best-fit parameter estimates. Smaller values are better fits.

**Stable** indicates whether the autoregressive operator is stable. That is, whether all the roots of  $\phi(z) = 0$  lie outside the unit circle.

**Invertible** indicates whether the moving average operator is invertible. That is, whether all the roots of  $\theta(z) = 0$  lie outside the unit circle.

**Note:** The  $\phi$  and  $\theta$  operators are defined in the section “[ARIMA Model](#),” p. 343.

## Parameter Estimates Table

Term	Lag	Estimate	Std Error	t Ratio	Prob> t	Constant Estimate
AR1	1	-0.476762	0.115264	-4.14	<.0001*	3.61907305
MA1	1	-0.864542	0.071380	-12.11	<.0001*	
Intercept	0	2.450682	3.258443	0.75	0.4533	

There is a Parameter Estimates table for each selected fit, which gives the estimates for the time series model parameters. Each type of model has its own set of parameters. They are described in the sections on specific time series models. The Parameter Estimates table has these terms:

**Term** lists the name of the parameter. These are described below for each model type. Some models contain an *intercept* or mean term. In those models, the related *constant estimate* is also shown. The definition of the constant estimate is given under the description of ARIMA models.

**Factor (Seasonal ARIMA only)** lists the factor of the model that contains the parameter. This is only shown for multiplicative models. In the multiplicative seasonal models, Factor 1 is nonseasonal and Factor 2 is seasonal.

**Lag (ARIMA and Seasonal ARIMA only)** lists the degree of the lag or backshift operator that is applied to the term to which the parameter is multiplied.

**Estimate** lists the parameter estimates of the time series model.

**Std Error** lists the estimates of the standard errors of the parameter estimates. They are used in constructing tests and confidence intervals.

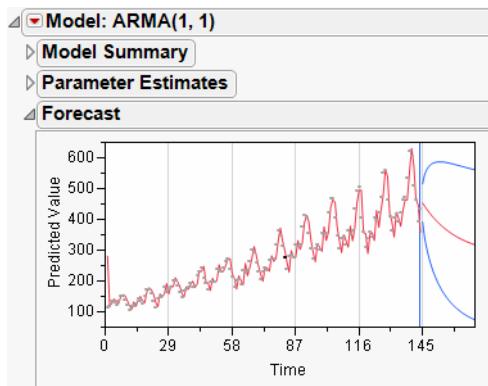
**t Ratio** lists the test statistics for the hypotheses that each parameter is zero. It is the ratio of the parameter estimate to its standard error. If the hypothesis is true, then this statistic has an approximate Student's *t*-distribution. Looking for a *t*-ratio greater than 2 in absolute value is a common rule of thumb for judging significance because it approximates the 0.05 significance level.

**Prob>|t|** lists the observed significance probability calculated from each *t*-ratio. It is the probability of getting, by chance alone, a *t*-ratio greater (in absolute value) than the computed value, given a true

hypothesis. Often, a value below 0.05 (or sometimes 0.01) is interpreted as evidence that the parameter is significantly different from zero.

The Parameter Estimates table also gives the Constant Estimate, for models that contain an intercept or mean term. The definition of the constant estimate is given under “[ARIMA Model](#),” p. 343.

## Forecast Plot




---

Each model has its own Forecast plot. The Forecast plot shows the values that the model predicts for the time series. It is divided by a vertical line into two regions. To the left of the separating line the one-step-ahead forecasts are shown overlaid with the input data points. To the right of the line are the future values forecast by the model and the confidence intervals for the forecasts.

You can control the number of forecast values by changing the setting of the **Forecast Periods** box in the platform launch dialog or by selecting **Number of Forecast Periods** from the Time Series drop-down menu. The data and confidence intervals can be toggled on and off using the **Show Points** and **Show Confidence Interval** commands on the model’s popup menu.

## Residuals

The graphs under the residuals section of the output show the values of the residuals based on the fitted model. These are the actual values minus the one-step-ahead predicted values. In addition, the autocorrelation and partial autocorrelation of these residuals are shown. These can be used to determine whether the fitted model is adequate to describe the data. If it is, the points in the residual plot should be normally distributed about the zero line and the autocorrelation and partial autocorrelation of the residuals should not have any significant components for lags greater than zero.

## Iteration History

The model parameter estimation is an iterative procedure by which the log-likelihood is maximized by adjusting the estimates of the parameters. The iteration history for each model you request shows the value of the objective function for each iteration. This can be useful for diagnosing problems with the fitting

procedure. Attempting to fit a model which is poorly suited to the data can result in a large number of iterations that fail to converge on an optimum value for the likelihood. The Iteration History table shows the following quantities:

**Iter** lists the iteration number.

**Iteration History** lists the objective function value for each step.

**Step** lists the type of iteration step.

**Obj-Criterion** lists the norm of the gradient of the objective function.

## Model Report Options

The title bar for each model you request has a popup menu, with the following options for that model:

**Show Points** hides or shows the data points in the forecast graph.

**Show Confidence Interval** hides or shows the confidence intervals in the forecast graph.

**Save Columns** creates a new data table with columns representing the results of the model.

**Save Prediction Formula** saves the data and prediction formula to a new data table.

**Create SAS Job** creates SAS code that duplicates the model analysis in SAS.

**Submit to SAS** submits code to SAS that duplicates the model analysis. If you are not connected to a SAS server, prompts guide you through the connection process.

**Residual Statistics** controls which displays of residual statistics are shown for the model. These displays are described in the section “[Time Series Commands](#),” p. 332; however, they are applied to the residual series.

## ARIMA Model

An **AutoRegressive Integrated Moving Average** (ARIMA) model predicts future values of a time series by a linear combination of its past values and a series of errors (also known as *random shocks* or *innovations*). The **ARIMA** command performs a maximum likelihood fit of the specified ARIMA model to the time series.

For a response series  $\{y_t\}$ , the general form for the ARIMA model is:

$$\phi(B)(w_t - \mu) = \theta(B)a_t$$

where

$t$  is the time index

$B$  is the backshift operator defined as  $B y_t = y_{t-1}$

$w_t = (1 - B)^d y_t$  is the response series after differencing

$\mu$  is the intercept or mean term.

$\phi(B)$  and  $\theta(B)$ , respectively, the autoregressive operator and the moving average operator and are written

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \text{ and } \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$$

$\alpha_t$  are the sequence of random shocks.

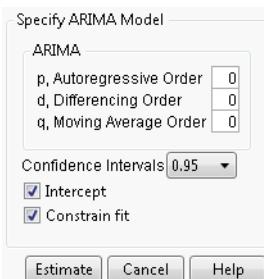
The  $\alpha_t$  are assumed to be independent and normally distributed with mean zero and constant variance.

The model can be rewritten as

$$\phi(B)\omega_t = \delta + \theta(B)\alpha_t \text{ where the constant estimate } \delta \text{ is given by the relation}$$

$$\delta = \phi(B)\mu = \mu - \phi_1\mu - \phi_2\mu - \dots - \phi_p\mu .$$

The ARIMA command displays the Specify ARIMA Model dialog, which allows you to specify the ARIMA model you want to fit. The results appear when you click **Estimate**.



Use the Specify ARIMA Model dialog for the following three orders that can be specified for an ARIMA model:

1. The **Autoregressive Order** is the order ( $p$ ) of the polynomial  $\phi(B)$  operator.
2. The **Differencing Order** is the order ( $d$ ) of the differencing operator.
3. The **Moving Average Order** is the order ( $q$ ) of the differencing operator  $\theta(B)$  .
4. An ARIMA model is commonly denoted ARIMA( $p,d,q$ ). If any of  $p,d$ , or  $q$  are zero, the corresponding letters are often dropped. For example, if  $p$  and  $d$  are zero, then model would be denoted MA( $q$ ).

The **Confidence Intervals** box allows you to set the confidence level between 0 and 1 for the forecast confidence bands. The **Intercept** check box determines whether the intercept term  $\mu$  will be part of the model. If the **Constrain fit** check box is checked, the fitting procedure will constrain the autoregressive parameters to always remain within the stable region and the moving average parameters within the invertible region. You might want to uncheck this box if the fitter is having difficulty finding the true optimum or if you want to speed up the fit. You can check the Model Summary table to see if the resulting fitted model is stable and invertible.

---

## Seasonal ARIMA

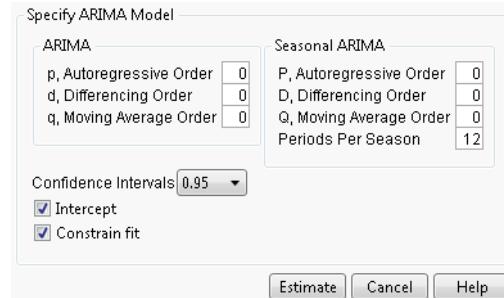
In the case of **Seasonal ARIMA** modeling, the differencing, autoregressive, and moving average operators are the product of seasonal and nonseasonal polynomials:

$$w_t = (1 - B)^d (1 - B^s)^D y_t$$

$$\varphi(B) = (1 - \varphi_{1,1}B - \varphi_{1,2}B^2 - \dots - \varphi_{1,p}B^p)(1 - \varphi_{2,s}B^s - \varphi_{2,2s}B^{2s} - \dots - \varphi_{2,P_s}B^{P_s})$$

$$\theta(B) = (1 - \theta_{1,1}B - \theta_{1,2}B^2 - \dots - \theta_{1,q}B^q)(1 - \theta_{2,s}B^s - \theta_{2,2s}B^{2s} - \dots - \theta_{2,Q_s}B^{Q_s})$$

where  $s$  is the number of periods in a season. The first index on the coefficients is the factor number (1 indicates nonseasonal, 2 indicates seasonal) and the second is the lag of the term.



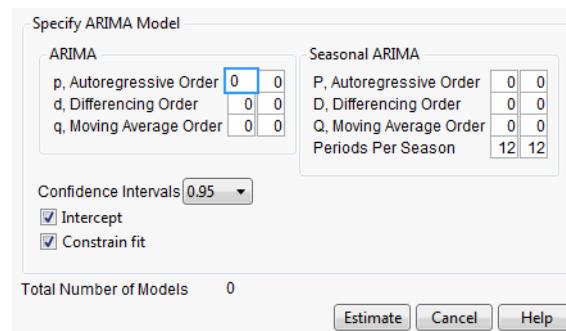

---

The Seasonal ARIMA dialog appears when you select the **Seasonal ARIMA** command. It has the same elements as the ARIMA dialog and adds elements for specifying the seasonal autoregressive order ( $P$ ), seasonal differencing order ( $D$ ), and seasonal moving average order ( $Q$ ). Also, the **Periods Per Season** box lets you specify the number of periods per season ( $s$ ). The seasonal ARIMA models are denoted as Seasonal ARIMA( $p,d,q$ )( $P,D,Q$ ) $s$ .

---

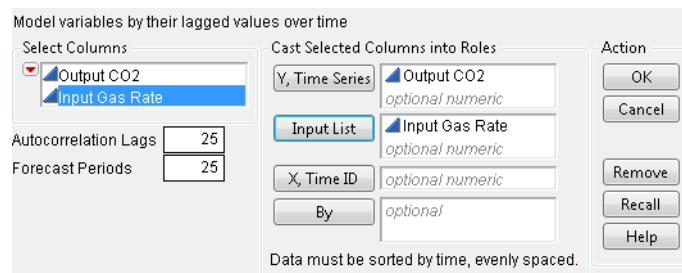
## ARIMA Model Group

The ARIMA Model Group option on the platform red-triangle menu allows the user to fit a range of ARIMA or Seasonal ARIMA models by specifying the range of orders. Figure 16.8 shows the dialog.

**Figure 16.8** ARIMA Group

## Transfer Functions

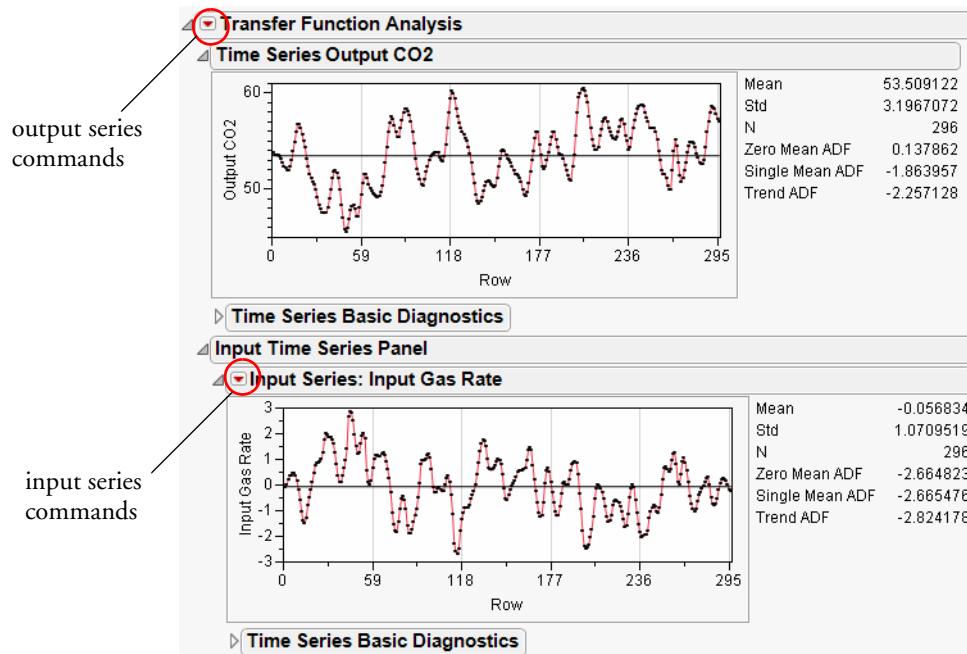
This example analyzes the gas furnace data (`seriesJ.jmp`) from Box and Jenkins. To begin the analysis, select Input Gas Rate as the **Input List** and Output CO2 as the **Y, Time Series**. The launch dialog should appear as in Figure 16.9.

**Figure 16.9** Series J Launch Dialog

When you click **OK**, the report in Figure 16.10 appears.

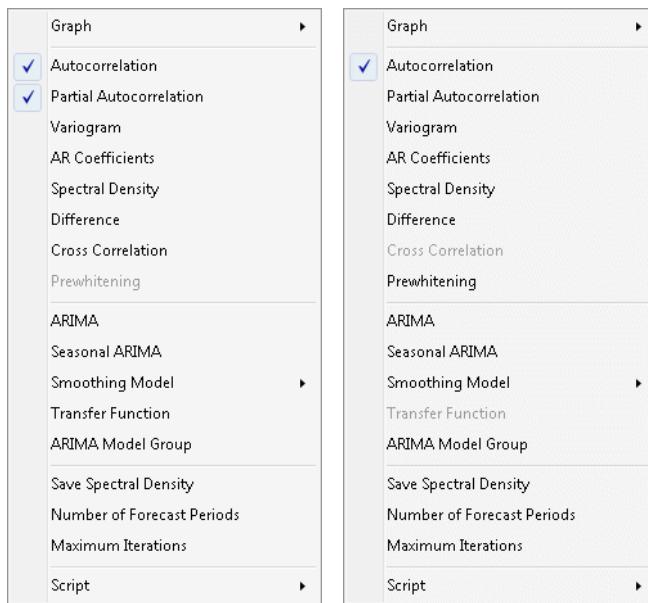
## Report and Menu Structure

This preliminary report shows diagnostic information and groups the analysis in two main parts. The first part, under Time Series Output CO2, contains analyses of the output series, while the Input Time Series Panel, contains analyses on the input series. The latter may include more than one series.

**Figure 16.10** Series J Report

Each report section has its own set of commands. For the output (top) series, the commands are accessible from the red triangle on the outermost outline bar (**Transfer Function Analysis**). For the input (bottom) series, the red triangle is located on the inner outline bar (**Input Series: Input Gas Rate**).

Figure 16.11 shows these two command sets. Note their organization. Both start with a **Graph** command. The next set of commands are for exploration. The third set is for model building. The fourth set includes functions that control the platform.

**Figure 16.11** Output and Input Series Menus

## Diagnostics

Both parts give basic diagnostics, including the sample mean (**Mean**), sample standard deviation (**Std**), and series length (**N**).

In addition, the platform tests for stationarity using *Augmented Dickey-Fuller* (ADF) tests.

**Zero Mean ADF** tests against a random walk with zero mean, *i.e.*

$$x_t = \phi x_{t-1} + e_t$$

**Single Mean ADF** tests against a random walk with a non-zero mean, *i.e.*

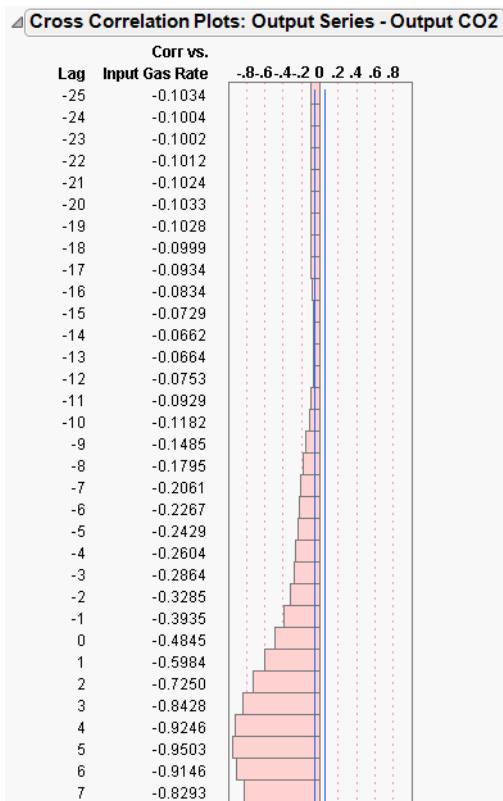
$$x_t - \mu = \phi(x_{t-1} - \mu) + e_t$$

**Trend ADF** tests against a random walk with a non-zero mean and a linear trend, *i.e.*

$$x_t - \mu - \beta t = \phi[x_{t-1} - \mu - \beta(t-1)] + e_t$$

Basic diagnostics also include the autocorrelation and partial autocorrelation functions, as well as the Ljung-Box *Q*-statistic and *p*-values, found under the **Time Series Basic Diagnostics** outline node.

The **Cross Correlation** command adds a cross-correlation plot to the report. The length of the plot is twice that of an autocorrelation plot, or  $2 \times \text{ACF length} + 1$ .

**Figure 16.12** Cross Correlation Plot

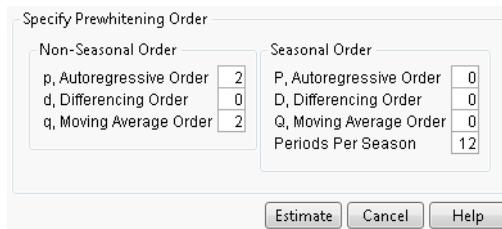
The plot includes plots of the output series versus all input series, in both numerical and graphical forms. The blue lines indicate standard errors for the statistics.

## Model Building

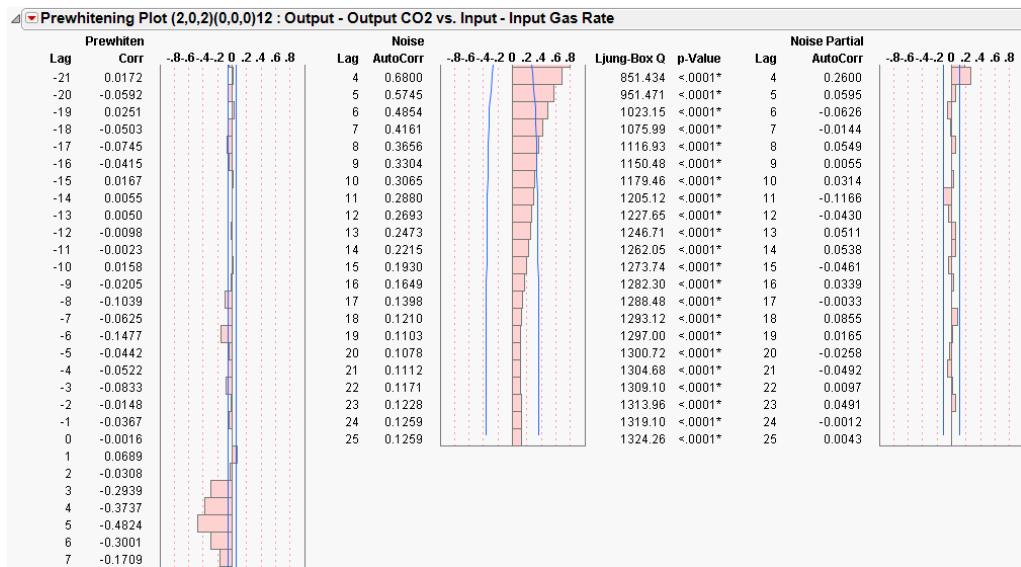
Building a transfer function model is quite similar to building an ARIMA model, in that it is an iterative process of exploring, fitting, and comparing.

Before building a model and during the data exploration process, it is sometimes useful to prewhiten the data. This means find an adequate model for the input series, apply the model to the output, and get residuals from both series. Compute cross-correlations from residual series and identify the proper orders for the transfer function polynomials.

To prewhiten the input series, select the **Prewhitenning** command. This brings up a dialog similar to the ARIMA dialog where you specify a stochastic model for the input series. For our SeriesJ example, we use an ARMA(2,2) prewhitening model, as shown in Figure 16.13.

**Figure 16.13** Prewhitenning Dialog

Click **Estimate** to reveal the Prewhitenning plot.



Patterns in these plots suggest terms in the transfer function model.

## Transfer Function Model

A typical transfer function model with  $m$  inputs can be represented as

$$Y_t - \mu = \frac{\omega_1(B)}{\delta_1(B)} X_{1,t-d1} + \dots + \frac{\omega_m(B)}{\delta_m(B)} X_{m,m-dm} + \frac{\theta(B)}{\phi(B)} e_t$$

where

$Y_t$  denotes the output series

$X_1$  to  $X_m$  denote  $m$  input series

$e_t$  represents the noise series

$X_{1, t-d_1}$  indicates the series  $X_1$  is indexed by  $t$  with a  $d_1$ -step lag

$\mu$  represents the mean level of the model

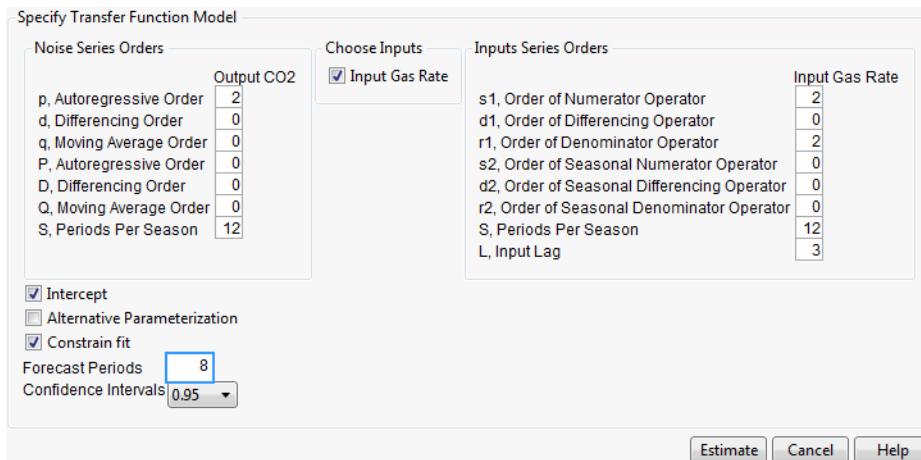
$\varphi(B)$  and  $\theta(B)$  represent autoregressive and moving average polynomials from an ARIMA model

$\omega_k(B)$  and  $\delta_k(B)$  represent numerator and denominator factors (or polynomials) for individual transfer functions, with  $k$  representing an index for the 1 to  $m$  individual inputs.

Each polynomial in the above model can contain two parts, either nonseasonal, seasonal, or a product of the two as in seasonal ARIMA. When specifying a model, leave the default 0 for any part that you do not want in the model.

Select **Transfer Function** to bring up the model specification dialog.

**Figure 16.14** Transfer Function Specification Dialog



The dialog consists of several parts.

**Noise Series Orders** contains specifications for the noise series. Lowercase letters are coefficients for non-seasonal polynomials, and uppercase letters for seasonal ones.

**Choose Inputs** lets you select the input series for the model.

**Input Series Orders** specifies polynomials related to the input series. The first three orders deal with non-seasonal polynomials. The next four are for seasonal polynomials. The final is for an input lag.

In addition, there are three options that control model fitting.

**Intercept** specifies whether  $\mu$  is zero or not.

**Alternative Parameterization** specifies whether the general regression coefficient is factored out of the numerator polynomials.

**Constrain Fit** toggles constraining of the AR and MA coefficients.

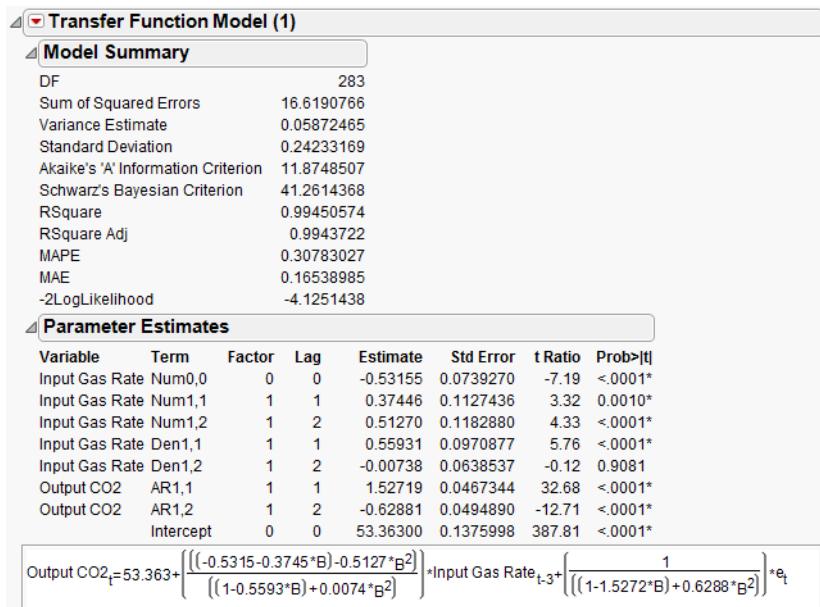
**Forecast Periods** specifies the number of forecasting periods for forecasting.

Using the information from prewhitening, we specify the model as shown in Figure 16.14.

## Model Reports

The analysis report is titled Transfer Function Model and is indexed sequentially. Results for the Series J example are shown in Figure 16.15.

**Figure 16.15** Series J Transfer Function Reports



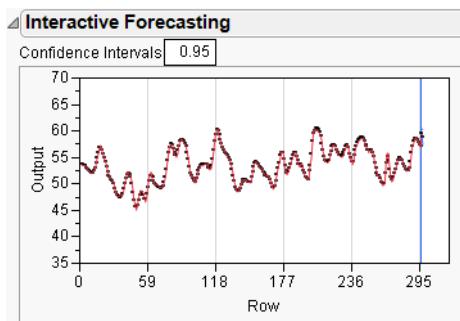
**Model Summary** gathers information that is useful for comparing models.

**Parameter Estimates** shows the parameter estimates and is similar to the ARIMA version. In addition, the Variable column shows the correspondence between series names and parameters. The table is followed by the formula of the model. Note the notation **B** is for the backshift operator.

**Residuals, Iteration History** are the same as their ARIMA counterparts.

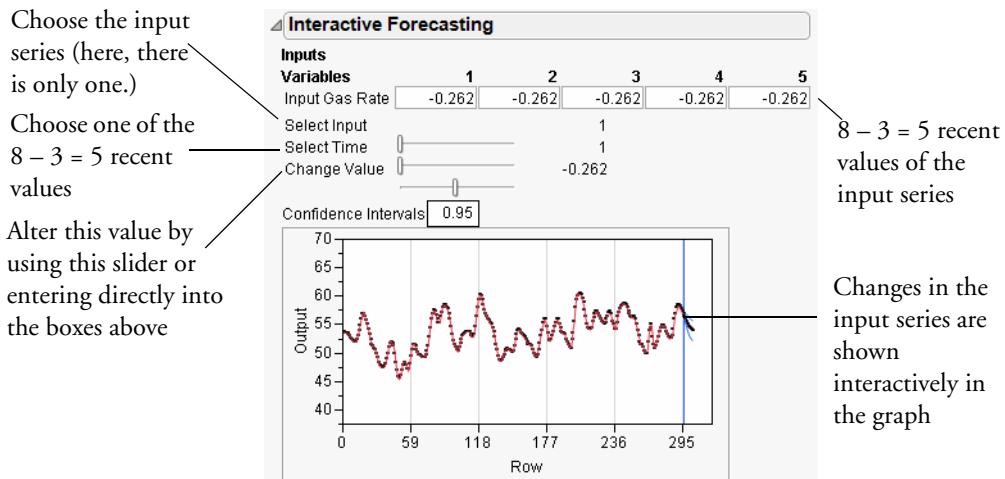
**Interactive Forecasting** provides a forecasting graph based on a specified confidence interval. The functionality changes based on the number entered in the Forecast Periods box.

If the number of Forecast Periods is less than or equal to the Input Lag, the forecasting box shows the forecast for the number of periods. A confidence interval around the prediction is shown in blue, and this confidence interval can be changed by entering a number in the Confidence Interval box above the graph.



If the number of forecast periods is larger than the number of lags (say, eight in our example), the presentation is a little different.

8 forecast periods with an input lag of 3



Here, you manipulate lagged values of the series by entering values into the edit boxes next to the series, or by manipulating the sliders. As before, the confidence interval can also be changed. The results of your changes are reflected in real time in the Interactive Forecasting graph.

The following commands are available from the report drop-down menu.

**Save Columns** creates a new data table containing the input and output series, a time column, predicted output with standard errors, residuals, and 95% confidence limits.

**Create SAS Job** creates PROC ARIMA code that can reproduce this model.

**Submit to SAS** submits PROC ARIMA code to SAS that reproduces the model.

## Model Comparison Table

The model comparison table works like its ARIMA counterpart by accumulating statistics on the models you specify.

ReportGraph	Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	Weights	.2 .4 .6 .8	MAPE	MAE
<input checked="" type="checkbox"/>	Transfer Function Model (1)	283	0.0587246	11.874851	41.261437	0.995	-4.125149	0.696769		0.307830	0.165390
<input checked="" type="checkbox"/>	Transfer Function Model (2)	282	0.0588643	13.538767	46.598676	0.995	-4.461233	0.303231		0.308171	0.165563

## Fitting Notes

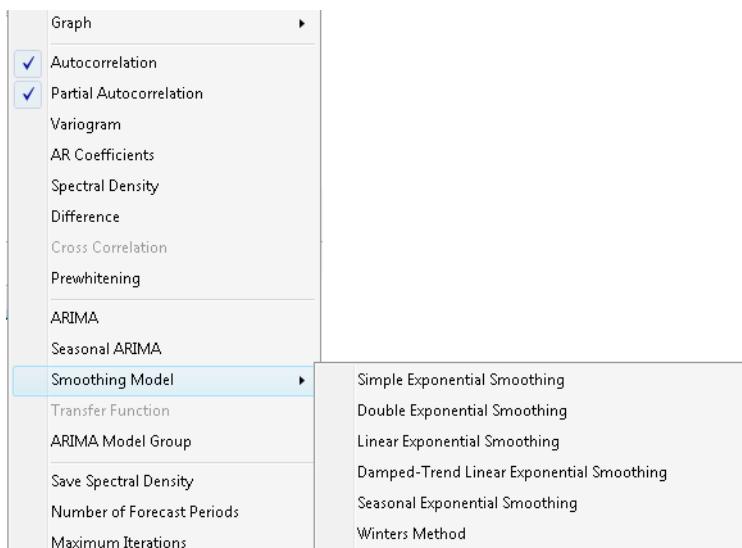
A regression model with serially correlated errors can be specified by including regressors in the model and not specifying any polynomial orders.

Intervention analysis can also be conducted, but prewhitening is no longer meaningful.

Currently, the transfer function model platform has limited capability of supporting missing values.

## Smoothing Models

JMP offers a variety of smoothing techniques.



Smoothing models represent the evolution of a time series by the model:

$$y_t = \mu_t + \beta_t t + s(t) + \alpha_t \text{ where}$$

$\mu_t$  is the time-varying mean term,

$\beta_t$  is the time-varying slope term,

$s(t)$  is one of the  $s$  time-varying seasonal terms,

$a_t$  are the random shocks.

Models without a trend have  $\beta_t = 0$  and nonseasonal models have  $s(t) = 0$ . The estimators for these time-varying terms are

$L_t$  smoothed level that estimates  $\mu_t$

$T_t$  is a smoothed trend that estimates  $\beta_t$

$S_{t-j}$  for  $j = 0, 1, \dots, s-1$  are the estimates of the  $s(t)$ .

Each smoothing model defines a set of recursive smoothing equations that describes the evolution of these estimators. The smoothing equations are written in terms of model parameters called *smoothing weights*. They are

$\alpha$ , the level smoothing weight

$\gamma$ , the trend smoothing weight

$\varphi$ , the trend damping weight

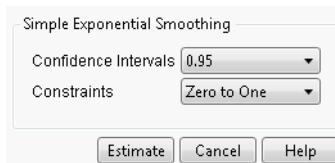
$\delta$ , the seasonal smoothing weight.

While these parameters enter each model in a different way (or not at all), they have the common property that larger weights give more influence to recent data while smaller weights give less influence to recent data.

Each smoothing model has an ARIMA model equivalent. You may not be able to specify the equivalent ARIMA model using the **ARIMA** command because some smoothing models intrinsically constrain the ARIMA model parameters in ways the ARIMA command will not allow.

## Smoothing Model Dialog

The Smoothing Model dialog appears in the report window when you select one of the smoothing model commands.




---

The **Confidence Intervals** popup list allows you to set the confidence level for the forecast confidence bands. The dialogs for seasonal smoothing models include a **Periods Per Season** box for setting the

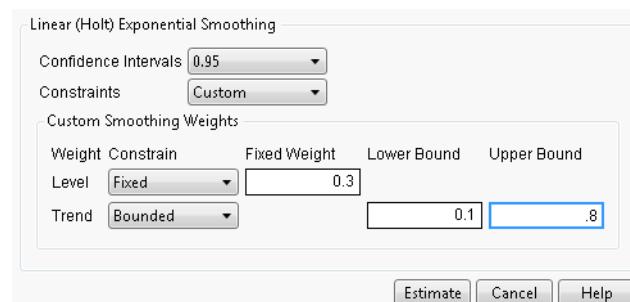
number of periods in a season. The **Constraints** popup list lets you to specify what type of constraint you want to enforce on the smoothing weights during the fit. The constraints are:

**Zero To One** keeps the values of the smoothing weights in the range zero to one.

**Unconstrained** allows the parameters to range freely.

**Stable Invertible** constrains the parameters such that the equivalent ARIMA model is stable and invertible.

**Custom** expands the dialog to allow you to set constraints on individual smoothing weights. Each smoothing weight can be **Bounded**, **Fixed**, or **Unconstrained** as determined by the setting of the the popup menu next to the weight's name. When entering values for fixed or bounded weights, the values can be positive or negative real numbers.



The example shown here has the Level weight ( $\alpha$ ) fixed at a value of 0.3 and the Trend weight ( $\gamma$ ) bounded by 0.1 and 0.8. In this case, the value of the Trend weight is allowed to move within the range 0.1 to 0.8 while the Level weight is held at 0.3. Note that you can specify all the smoothing weights in advance by using these custom constraints. In that case, none of the weights would be estimated from the data although forecasts and residuals would still be computed. When you click **Estimate**, the results of the fit appear in place of the dialog.

## Simple Exponential Smoothing

The model for simple exponential smoothing is  $y_t = \mu_t + \alpha_t$ .

The smoothing equation,  $L_t = \alpha y_t + (1 - \alpha)L_{t-1}$ , is defined in terms of a single smoothing weight  $\alpha$ . This model is equivalent to an ARIMA(0, 1, 1) model where

$$(1 - B)y_t = (1 - \theta B)\alpha_t \text{ with } \theta = 1 - \alpha.$$

The moving average form of the model is

$$y_t = \mu_t + \sum_{j=1}^{\infty} \alpha \alpha_{t-j}$$

## Double (Brown) Exponential Smoothing

The model for double exponential smoothing is  $y_t = \mu_t + \beta_1 t + a_t$ .

The smoothing equations, defined in terms of a single smoothing weight  $\alpha$  are

$$L_t = \alpha y_t + (1 - \alpha)L_{t-1} \text{ and } T_t = \alpha(L_t - L_{t-1}) + (1 - \alpha)T_{t-1}.$$

This model is equivalent to an ARIMA(0, 1, 1)(0, 1, 1)1 model

$$(1 - B)^2 y_t = (1 - \theta B)^2 a_t \text{ where } \theta_{1,1} = \theta_{2,1} \text{ with } \theta = 1 - \alpha.$$

The moving average form of the model is

$$y_t = a_t + \sum_{j=1}^{\infty} (2\alpha + (j-1)\alpha^2) a_{t-j}$$

## Linear (Holt) Exponential Smoothing

The model for linear exponential smoothing is  $y_t = \mu_t + \beta_t t + a_t$ .

The smoothing equations defined in terms of smoothing weights  $\alpha$  and  $\gamma$  are

$$L_t = \alpha y_t + (1 - \alpha)(L_{t-1} + T_{t-1}) \text{ and } T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$$

This model is equivalent to an ARIMA(0, 2, 2) model where

$$(1 - B)^2 y_t = (1 - \theta B - \theta_2 B^2) a_t \text{ with } \theta = 2 - \alpha - \alpha\gamma \text{ and } \theta_2 = \alpha - 1.$$

The moving average form of the model is

$$y_t = a_t + \sum_{j=1}^{\infty} (\alpha + j\alpha\gamma) a_{t-j}$$

## Damped-Trend Linear Exponential Smoothing

The model for damped-trend linear exponential smoothing is  $y_t = \mu_t + \beta_t t + a_t$ .

The smoothing equations in terms of smoothing weights  $\alpha$ ,  $\gamma$ , and  $\phi$  are

$$L_t = \alpha y_t + (1 - \alpha)(L_{t-1} + \phi T_{t-1}) \text{ and } T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)\phi T_{t-1}$$

This model is equivalent to an ARIMA(1, 1, 2) model where

$$(1 - \phi B)(1 - B)y_t = (1 - \theta_1 B - \theta_2 B^2)a_t \text{ with } \theta_1 = 1 + \phi - \alpha - \alpha\gamma\phi \text{ and } \theta_2 = (\alpha - 1)\phi.$$

The moving average form of the model is

$$y_t = \alpha_t + \sum_{j=1}^{\infty} \left( \frac{\alpha + \alpha\gamma\varphi(\varphi^j - 1)}{\varphi - 1} \right) \alpha_{t-j}$$

## Seasonal Exponential Smoothing

The model for seasonal exponential smoothing is  $y_t = \mu_t + s(t) + a_t$ .

The smoothing equations in terms of smoothing weights  $\alpha$  and  $\delta$  are

$$L_t = \alpha(y_t - S_{t-s}) + (1-\alpha)L_{t-1} \text{ and } S_t = \delta(y_t - L_{t-s}) + (1-\delta)\varphi S_{t-s}$$

This model is equivalent to a seasonal ARIMA(0, 1, 1)(0, 1, 0)s model where we define

$$\theta_1 = \theta_{1,1}, \theta_2 = \theta_{2,s} = \theta_{2,s}, \text{ and } \theta_3 = -\theta_{1,1}\theta_{2,s}$$

so

$$(1-B)(1-B^s)y_t = (1-\theta_1B - \theta_2B^2 - \theta_3B^{s+1})a_t$$

with

$$\theta_1 = 1-\alpha, \theta_2 = \delta(1-\alpha), \text{ and } \theta_3 = (1-\alpha)(\delta-1).$$

The moving average form of the model is

$$y_t = a_t + \sum_{j=1}^{\infty} \psi_j a_{t-j} \text{ where } \psi = \begin{cases} \alpha & \text{for } j \bmod s \neq 0 \\ \alpha + \delta(1-\alpha) & \text{for } j \bmod s = 0 \end{cases}$$

## Winters Method (Additive)

The model for the additive version of the Winters method is  $y_t = \mu_t + \beta_t t + s(t) + a_t$ .

The smoothing equations in terms of weights  $\alpha$ ,  $\gamma$ , and  $\delta$  are

$$L_t = \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + T_{t-1}), \quad T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1}, \text{ and} \\ S_t = \delta(y_t - L_t) + (1-\delta)S_{t-s}.$$

This model is equivalent to a seasonal ARIMA(0, 1, s+1)(0, 1, 0)s model

$$(1-B)(1-B^2)y_t = \left( 1 - \sum_{i=1}^{s+1} \theta_i B^i \right) a_t$$

The moving average form of the model is

$$\gamma_t = \alpha_t + \sum_{j=1}^{\infty} \Psi_j \alpha_{t-j}$$

where

$$\Psi = \begin{cases} \alpha + j\alpha\gamma, & j \bmod s \neq 0 \\ \alpha + j\alpha\gamma + \delta(1 - \alpha), & j \bmod s = 0 \end{cases}$$



# Chapter 17

## Categorical Response Analysis The Categorical Platform

The Categorical platform does tabulation and summarization of categorical response data, including multiple response data, and calculates test statistics. It is designed to handle survey and other categorical response data, such as defect records, side effects, etc.

**Figure 17.1** Example of a Categorical Analysis

**Categorical**

Sample Size: SampleSize

**Freq Group By clean**

Frequency Columns

**Frequency**

clean	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	30	6	1	14	3	4	5	63	150
before	51	6	6	7	7	17	3	97	150
-All-	81	12	7	21	10	21	8	160	300

**Share of Responses**

clean	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	0.4762	0.0952	0.0159	0.2222	0.0476	0.0635	0.0794	63	150
before	0.5258	0.0619	0.0619	0.0722	0.0722	0.1753	0.0309	97	150
-All-	0.5063	0.0750	0.0438	0.1313	0.0625	0.1313	0.0500	160	300

**Share Chart**

clean	Freq Group	Responses	Cases
after		63	150
before		97	150
-All-		160	300

**Frequency Chart**

clean	Freq Group	Responses	Cases
after		63	150
before		97	150
-All-		160	300

**Test Each Response**

Freq Group	ChiSquare	Prob>ChiSq
contamination	5.5071	0.0189*
corrosion	0.0000	1.0000
doping	3.9624	0.0465*
metallization	2.3786	0.1230
miscellaneous	1.6457	0.1996
oxide defect	8.6618	0.0032*
silicon defect	0.5053	0.4772

# Contents

The Categorical Platform .....	363
Launching the Platform.....	363
Failure Rate Examples .....	366
Response Frequencies .....	366
Indicator Group .....	367
Multiple Delimited .....	367
Multiple Response By ID .....	368
Multiple Response.....	368
Categorical Reports.....	369
Report Content.....	369
Report Format.....	371
Statistical Commands .....	373
Save Tables .....	376

---

## The Categorical Platform

The Categorical platform has capabilities similar to other platforms. The choice of platforms depends on your focus, the shape of your data, and the desired level of detail. Table 17.1 shows several of JMP's analysis platforms, and their strengths.

**Table 17.1** Comparing JMP's Categorical Analyses

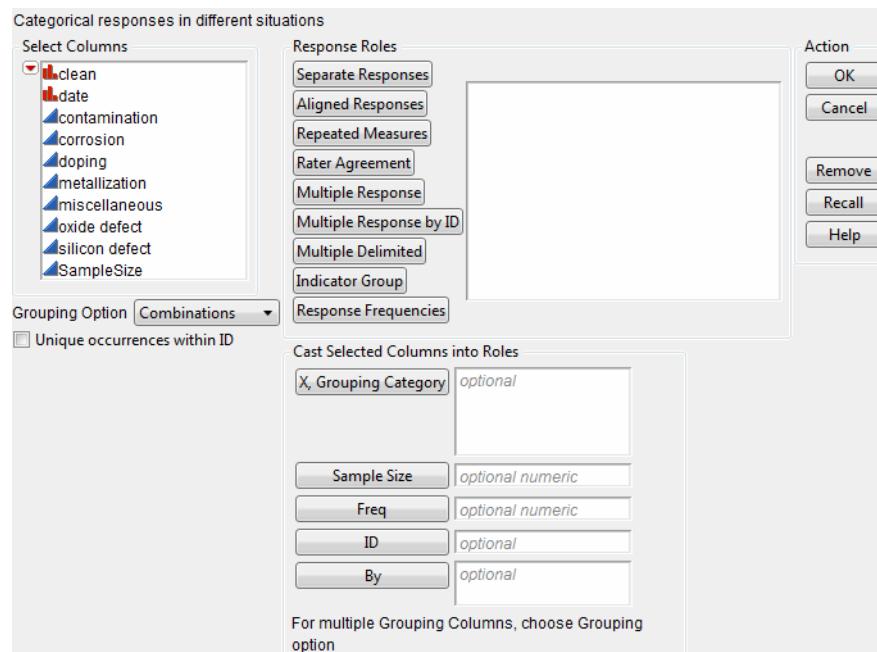
Platform	Specialty
Distribution	Separate, ungrouped categorical responses.
Fit Y By X: Contingency	Two-way situations, including chi-square tests, correspondence analysis, agreement.
Pareto Plot	Graphical analysis of multiple-response data, especially multiple-response defect data, with more rate tests than Fit Y By X.
Variability Chart: Attribute	Attribute gauge studies, with more detail on rater agreement.
Fit Model	Logistic categorical responses and generalized linear models.
Partition, Neural Net	Specific categorical response models.

The strength of the categorical platform is that it can handle responses in a wide variety of formats without needing to reshape the data.

---

## Launching the Platform

Open Failure3Freq.jmp in the Quality Control folder located in the sample data. Click on **Analyze > Modeling > Categorical** to obtain the launch dialog. The platform has a unique launch dialog. The buttons select groups of columns (as opposed to single columns like other JMP dialogs) for each of the situations.

**Figure 17.2** Categorical Platform Launch Dialog

## Response Roles

Use the response roles buttons to choose selected columns as responses with specified roles. The response roles are summarized in Table 17.2, with the first four roles describing single response data and the last five roles describing multiple response data.

**Table 17.2** Response Roles

Response Role	Description	Example Data
<b>Separate Responses</b>	Separate responses are in each column, resulting in a separate analysis for each column.	ID      Drink      Entrée John    Coffee    Chicken Jane    Tea        Veggie
<b>Aligned Responses</b>	Responses share common categories across columns, resulting in better-organized reports.	ID      Coffee      Tea John    Like        Dislike Jane    Dislike     Like
<b>Repeated Measures</b>	Aligned responses from an individual across different times or situations.	ID      Morning      Noon      Night John    Coffee      Coffee    Water Jane    Tea        Water    Tea

**Table 17.2** Response Roles (*Continued*)

Response Role	Description	Example Data															
<b>Rater Agreement</b>	Aligned responses from different raters evaluating the same unit, to study agreement across raters.	<table> <thead> <tr> <th>Drink</th> <th>John</th> <th>Jane</th> </tr> </thead> <tbody> <tr> <td>Coffee</td> <td>Like</td> <td>Dislike</td> </tr> <tr> <td>Tea</td> <td>Dislike</td> <td>Like</td> </tr> <tr> <td>Water</td> <td>Like</td> <td>Like</td> </tr> </tbody> </table>	Drink	John	Jane	Coffee	Like	Dislike	Tea	Dislike	Like	Water	Like	Like			
Drink	John	Jane															
Coffee	Like	Dislike															
Tea	Dislike	Like															
Water	Like	Like															
<b>Multiple Response</b>	Aligned responses, where multiple responses are entered across several columns, but treated as one grouped response.	<table> <thead> <tr> <th>ID</th> <th>Drink 1</th> <th>Drink 2</th> <th>Drink 3</th> </tr> </thead> <tbody> <tr> <td>John</td> <td>Coffee</td> <td>Milk</td> <td>Water</td> </tr> <tr> <td>Jane</td> <td>Tea</td> <td>Water</td> <td></td> </tr> </tbody> </table>	ID	Drink 1	Drink 2	Drink 3	John	Coffee	Milk	Water	Jane	Tea	Water				
ID	Drink 1	Drink 2	Drink 3														
John	Coffee	Milk	Water														
Jane	Tea	Water															
<b>Multiple Response by ID</b>	Multiple responses across rows that have the same ID values.	<table> <thead> <tr> <th>ID</th> <th>Drinks</th> </tr> </thead> <tbody> <tr> <td>John</td> <td>Coffee</td> </tr> <tr> <td>John</td> <td>Milk</td> </tr> <tr> <td>John</td> <td>Water</td> </tr> <tr> <td>Jane</td> <td>Tea</td> </tr> <tr> <td>Jane</td> <td>Water</td> </tr> </tbody> </table>	ID	Drinks	John	Coffee	John	Milk	John	Water	Jane	Tea	Jane	Water			
ID	Drinks																
John	Coffee																
John	Milk																
John	Water																
Jane	Tea																
Jane	Water																
<b>Multiple Delimited</b>	Several responses in a single cell, separated by commas.	<table> <thead> <tr> <th>ID</th> <th>Drinks</th> </tr> </thead> <tbody> <tr> <td>John</td> <td>Coffee, Milk, Water</td> </tr> <tr> <td>Jane</td> <td>Tea, Water</td> </tr> </tbody> </table>	ID	Drinks	John	Coffee, Milk, Water	Jane	Tea, Water									
ID	Drinks																
John	Coffee, Milk, Water																
Jane	Tea, Water																
<b>Indicator Group</b>	Binary responses across columns, like checked or not, yes or no, but all in a related group.	<table> <thead> <tr> <th>ID</th> <th>Coffee</th> <th>Milk</th> <th>Tea</th> <th>Water</th> </tr> </thead> <tbody> <tr> <td>John</td> <td>Y</td> <td>Y</td> <td>N</td> <td>Y</td> </tr> <tr> <td>Jane</td> <td>N</td> <td>N</td> <td>Y</td> <td>Y</td> </tr> </tbody> </table>	ID	Coffee	Milk	Tea	Water	John	Y	Y	N	Y	Jane	N	N	Y	Y
ID	Coffee	Milk	Tea	Water													
John	Y	Y	N	Y													
Jane	N	N	Y	Y													
<b>Response Frequencies</b>	Columns containing frequency counts for each response level, all in a related group.	<table> <thead> <tr> <th>Group</th> <th>Coffee</th> <th>Milk</th> <th>Tea</th> <th>Water</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>12</td> <td>15</td> <td>8</td> <td>19</td> </tr> <tr> <td>B</td> <td>9</td> <td>20</td> <td>6</td> <td>22</td> </tr> </tbody> </table>	Group	Coffee	Milk	Tea	Water	A	12	15	8	19	B	9	20	6	22
Group	Coffee	Milk	Tea	Water													
A	12	15	8	19													
B	9	20	6	22													

### Other Launch Dialog Options

The Launch dialog has standard JMP options (like **By** variables) and additionally:

**X, Grouping Categories** defines sample groups to break the counts into.

**Sample Size** For multiple response roles with summarized data, defines the number of individual units in the group for which that frequency is applicable to. For example, a Freq column might say there were 50 defects, where the sample size variable would say that they reflect the defects for a batch of 100 units.

**Freq** is for presummarized data, specifying the column containing frequency counts for each row.

**ID** is only required and used when Multiple Response by ID is used.

**Unique occurrences within ID** allows duplicate response levels within a subject to be counted only once. An ID variable must be specified.

**Grouping Option** is used to give results for individual X variables, or for combinations of the variables.

**Combinations** gives frequency results for combinations of the X variables.

**Each Individually** gives frequency results for each X variable individually.

**Both** gives frequency results for combinations of the X variables, and individually.

## Failure Rate Examples

The following examples come from testing a fabrication line on three different occasions under two different conditions. Each set of operating conditions yielded 50 data points. Inspectors recorded the following kinds of defects:

1. contamination
2. corrosion
3. doping
4. metallization
5. miscellaneous
6. oxide defect
7. silicon defect

Each unit could have several defects or even several defects of the same kind. We illustrate the data in a variety of different shapes all supported directly by the Categorical platform.

## Response Frequencies

Suppose the data have columns containing frequency counts for each batch and a column showing the total number of units of the batch (as in Failure3Freq.jmp).

	clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	SampleSize
1	after	OCT 1	12	2	0	4	2	1	2	50
2	after	OCT 2	10	1	1	5	1	2	3	50
3	after	OCT 3	8	3	0	5	0	1	0	50
4	before	OCT 1	14	2	1	2	3	8	1	50
5	before	OCT 2	15	2	2	1	4	6	0	50
6	before	OCT 3	22	2	3	4	0	3	2	50

To generate the appropriate analysis, highlight the frequency variables and click **Response Frequencies**. The **X, Grouping Category** columns are clean and date, and SampleSize is **Sample Size**.

In the resulting output, a frequency count table shows the total number of defects for each defect type. There is a separate row for each of the six batches. The last two columns show the total number of defects (**Responses**) and the total number of units (**Cases**).

**Figure 17.3** Defect Rate Output

Frequency										
								silicon defect	Responses	Cases
after	OCT 1		12	2	0	4	2	1	23	50
	OCT 2		10	1	1	5	1	2	23	50
	OCT 3		8	3	0	5	0	1	17	50
before	OCT 1		14	2	1	2	3	8	31	50
	OCT 2		15	2	2	1	4	6	30	50
	OCT 3		22	2	3	4	0	3	36	50
-All-	-All-		81	12	7	21	10	21	160	300

## Indicator Group

In some cases, the data is not yet summarized, so there are individual records for each unit. We illustrate this situation in Failures3Indicators.jmp.

	clean	date	ID	ID Label	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect
1	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
2	before	OCT 1	1	OCT 1 before	0	0	0	0	0	1	0
3	before	OCT 1	1	OCT 1 before	1	0	0	0	0	1	0
4	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
5	before	OCT 1	1	OCT 1 before	1	0	0	0	0	0	0
6	before	OCT 1	1	OCT 1 before	0	0	0	0	0	1	0
7	before	OCT 1	1	OCT 1 before	1	0	0	0	0	0	0
8	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
9	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
10	before	OCT 1	1	OCT 1 before	1	0	0	0	1	0	0
11	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
12	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
13	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
14	before	OCT 1	1	OCT 1 before	1	0	0	0	0	0	0
15	before	OCT 1	1	OCT 1 before	0	0	0	0	0	0	0
16	before	OCT 1	1	OCT 1 before	0	0	0	0	1	0	0

With data like this, specify all the defect columns with the **Indicator Group** button. The **X, Grouping Category** columns are clean and date. When you click **OK**, you get the same output as in the Response Group example (Figure 17.3).

## Multiple Delimited

Suppose that the inspector entered the observed defects for each unit. The defects are listed in a single column, delimited by a comma. (Failures3Delimited.jmp). Note in the partial data table, shown below, that some units did not have any observed defects, so the failureS column is empty.

## Failure Rate Examples

	failureS	clean	date	ID	ID Label
1		before	OCT 1	1	OCT 1 before
2	oxide defect	before	OCT 1	1	OCT 1 before
3	contamination,oxide defect	before	OCT 1	1	OCT 1 before
4		before	OCT 1	1	OCT 1 before
5	contamination	before	OCT 1	1	OCT 1 before
6	oxide defect	before	OCT 1	1	OCT 1 before
7	contamination	before	OCT 1	1	OCT 1 before
8		before	OCT 1	1	OCT 1 before
9		before	OCT 1	1	OCT 1 before
10	metallization,contamination	before	OCT 1	1	OCT 1 before
11		before	OCT 1	1	OCT 1 before
12		before	OCT 1	1	OCT 1 before
13		before	OCT 1	1	OCT 1 before
14	contamination	before	OCT 1	1	OCT 1 before

To get the appropriate analysis, specify the delimited defect columns with the **Multiple Delimited** button. The results are identical to those in Figure 17.3.

**Note:** If more than one delimited column is specified, separate analyses are produced for each column.

## Multiple Response By ID

Suppose each failure type is a separate record, with an ID column that can be used to link together different defect types for each unit, as in Failure3ID.jmp.

	failure	N	clean	date	SampleSize	ID
1	contamination	14	before	OCT 1	50	OCT 1 before
2	corrosion	2	before	OCT 1	50	OCT 1 before
3	doping	1	before	OCT 1	50	OCT 1 before
4	metallization	2	before	OCT 1	50	OCT 1 before
5	miscellaneous	3	before	OCT 1	50	OCT 1 before
6	oxide defect	8	before	OCT 1	50	OCT 1 before
7	silicon defect	1	before	OCT 1	50	OCT 1 before
8	doping	0	after	OCT 1	50	OCT 1 after
9	corrosion	2	after	OCT 1	50	OCT 1 after
10	metallization	4	after	OCT 1	50	OCT 1 after

Launch the Categorical platform. Specify failure as **Multiple Response by ID**, N as **Freq**, SampleSize as **Sample Size**, and clean and date as **X, Grouping** variables.

Results are identical to those in Figure 17.3.

## Multiple Response

Suppose that the defects for each unit are entered via a web page, but since each unit rarely has more than three defect types, the form has three fields to enter any of the defect types for a unit, as in Failure3MultipleField.jmp.

	clean	date	ID	ID Label	Failure1	Failure2	Failure3
1	before	OCT 1	1	OCT 1 before			
2	before	OCT 1	1	OCT 1 before	oxide defect		
3	before	OCT 1	1	OCT 1 before	contamination	oxide defect	
4	before	OCT 1	1	OCT 1 before			
5	before	OCT 1	1	OCT 1 before	contamination		
6	before	OCT 1	1	OCT 1 before	oxide defect		
7	before	OCT 1	1	OCT 1 before	contamination		
8	before	OCT 1	1	OCT 1 before			
9	before	OCT 1	1	OCT 1 before			
10	before	OCT 1	1	OCT 1 before	metallization	contamination	

Select the three columns containing defect types (Failure1, Failure2, and Failure3), and click the **Multiple Response** button. After specifying clean and date as X, **Grouping** variables, click **OK** to see the analysis results. Results are identical to those in Figure 17.3.

## Categorical Reports

### Report Content

The Categorical platform produces a report with several tables and charts. Frequency, Share of Responses, Rate Per Case tables, and a Share Chart are included by default. You can also select a Frequency Chart. A Crosstab Format, instead of a Table Format can also be selected. The defect data for Failure3Freq.jmp is illustrated below.

The topmost table is a **Frequency** count table, showing the frequency counts for each category with the total frequency (**Responses**) and total units (**Cases**) on the right. In this example, we have total defects for each defect type by the 6 batches. The last two columns show the total number of defects (**Responses**) and the total number of units (**Cases**).

Frequency										
clean	date	silicon								
		contamination	corrosion	doping	metallization	miscellaneous	oxide defect	defect	Responses	Cases
after	OCT 1	12	2	0	4	2	1	2	23	50
	OCT 2	10	1	1	5	1	2	3	23	50
	OCT 3	8	3	0	5	0	1	0	17	50
before	OCT 1	14	2	1	2	3	8	1	31	50
	OCT 2	15	2	2	1	4	6	0	30	50
	OCT 3	22	2	3	4	0	3	2	36	50
-All-	-All-	81	12	7	21	10	21	8	160	300

The **Share of Responses** table is built by dividing each count by the total number of responses.

**Share of Responses**

clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	OCT 1	0.5217	0.0870	0.0000	0.1739	0.0870	0.0435	0.0870	23	50
	OCT 2	0.4348	0.0435	0.0435	0.2174	0.0435	0.0870	0.1304	23	50
	OCT 3	0.4706	0.1765	0.0000	0.2941	0.0000	0.0588	0.0000	17	50
before	OCT 1	0.4516	0.0645	0.0323	0.0645	0.0968	0.2581	0.0323	31	50
	OCT 2	0.5000	0.0667	0.0667	0.0333	0.1333	0.2000	0.0000	30	50
	OCT 3	0.6111	0.0556	0.0833	0.1111	0.0000	0.0833	0.0556	36	50
-All-	-All-	0.5063	0.0750	0.0438	0.1313	0.0625	0.1313	0.0500	160	300

As an example, examine the second row of the table (corresponding to October 2, after cleaning). The 10 contamination defects were  $(10/23)=43.5\%$  of all defects.

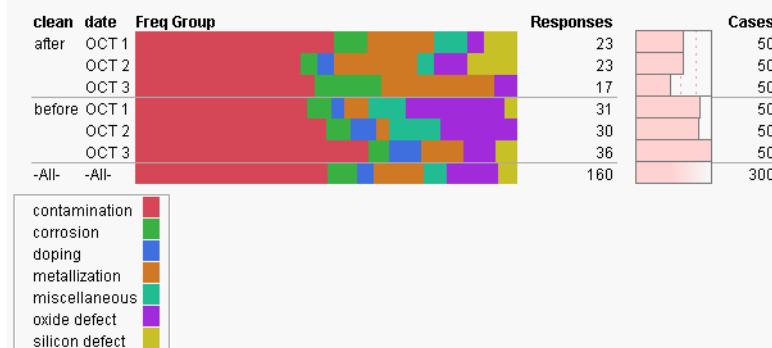
The **Rate Per Case** table divides each count in the frequency table by the total number of cases.

**Rate Per Case**

clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	OCT 1	0.2400	0.0400	0.0000	0.0800	0.0400	0.0200	0.0400	23	50
	OCT 2	0.2000	0.0200	0.0200	0.1000	0.0200	0.0400	0.0600	23	50
	OCT 3	0.1600	0.0600	0.0000	0.1000	0.0000	0.0200	0.0000	17	50
before	OCT 1	0.2800	0.0400	0.0200	0.0400	0.0600	0.1600	0.0200	31	50
	OCT 2	0.3000	0.0400	0.0400	0.0200	0.0800	0.1200	0.0000	30	50
	OCT 3	0.4400	0.0400	0.0600	0.0800	0.0000	0.0600	0.0400	36	50
-All-	-All-	0.2700	0.0400	0.0233	0.0700	0.0333	0.0700	0.0267	160	300

For example, in the first row of the table (October 1, after cleaning), the 12 contamination defects are from 50 units, making the rate per unit 24%.

The **Share Chart** presents two bar chart columns. The bar chart on the left shows the **Share of Responses** as a divided bar chart. The bar length is proportional to the percentage of responses for each type. Since a bar in one sample might reflect a much higher frequency than a bar in another sample, the bar chart on the right shows the number of responses. The bar filled with a gradient means the the number of responses for that bar is more than the width of the bar chart allows.

**Share Chart**

The **Frequency Chart** shows response frequencies. The bars in the chart on the left reflect the frequency count on the same scale. The bar chart on the right shows the number of units.



Note again that the gradient-filled bars are used when the number of responses or units is more than the width of the bar chart allows.

The **Transposed Freq Chart** option produces a transposed version of the Frequency Chart. Marginal totals are given for each response, as opposed to each *X* variable.

## Report Format

The default report format is the **Table Format**.

Freq Group By clean, date										
Frequency										
clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
-All-	-All-	81	12	7	21	10	21	8	160	300
Share of Responses										
clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	OCT 1	0.5217	0.0870	0.0000	0.1739	0.0870	0.0435	0.0870	23	50
	OCT 2	0.4348	0.0435	0.0435	0.2174	0.0435	0.0870	0.1304	23	50
	OCT 3	0.4706	0.1765	0.0000	0.2941	0.0000	0.0588	0.0000	17	50
before	OCT 1	0.4516	0.0645	0.0323	0.0645	0.0968	0.2581	0.0323	31	50
	OCT 2	0.5000	0.0667	0.0667	0.0333	0.1333	0.2000	0.0000	30	50
	OCT 3	0.6111	0.0556	0.0833	0.1111	0.0000	0.0833	0.0556	36	50
-All-	-All-	0.5063	0.0750	0.0438	0.1313	0.0625	0.1313	0.0500	160	300
Rate Per Case										
clean	date	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect	Responses	Cases
after	OCT 1	0.2400	0.0400	0.0000	0.0800	0.0400	0.0200	0.0400	23	50
	OCT 2	0.2000	0.0200	0.0200	0.1000	0.0200	0.0400	0.0600	23	50
	OCT 3	0.1600	0.0600	0.0000	0.1000	0.0000	0.0200	0.0000	17	50
before	OCT 1	0.2800	0.0400	0.0200	0.0400	0.0600	0.1600	0.0200	31	50
	OCT 2	0.3000	0.0400	0.0400	0.0200	0.0800	0.1200	0.0000	30	50
	OCT 3	0.4400	0.0400	0.0600	0.0800	0.0000	0.0600	0.0400	36	50
-All-	-All-	0.2700	0.0400	0.0233	0.0700	0.0333	0.0700	0.0267	160	300

To gather all three statistics for each sample and response together, use the **Crosstab** format.

Freq Group									
Freq Share Rate	contamination	corrosion	doping	metallization	miscellaneous	oxide defect	silicon defect		
clean, date	after,OCT 1	12	2	0	4	2	1	2	23
		0.522	0.087	0.000	0.174	0.087	0.043	0.087	50.000
		0.240	0.040	0.000	0.080	0.040	0.020	0.040	
	after,OCT 2	10	1	1	5	1	2	3	23
		0.435	0.043	0.043	0.217	0.043	0.087	0.130	50.000
		0.200	0.020	0.020	0.100	0.020	0.040	0.060	
	after,OCT 3	8	3	0	5	0	1	0	17
		0.471	0.176	0.000	0.294	0.000	0.059	0.000	50.000
		0.160	0.060	0.000	0.100	0.000	0.020	0.000	
	before,OCT 1	14	2	1	2	3	8	1	31
		0.452	0.065	0.032	0.065	0.097	0.258	0.032	50.000
		0.280	0.040	0.020	0.040	0.060	0.160	0.020	
	before,OCT 2	15	2	2	1	4	6	0	30
		0.500	0.067	0.067	0.033	0.133	0.200	0.000	50.000
		0.300	0.040	0.040	0.020	0.080	0.120	0.000	
	before,OCT 3	22	2	3	4	0	3	2	36
		0.611	0.056	0.083	0.111	0.000	0.083	0.056	50.000
		0.440	0.040	0.060	0.080	0.000	0.060	0.040	

Both **Table Format** and **Crosstab Format** have transposed versions (**Table Transposed** and **Crosstab Transposed**). These are useful when there are a lot of response categories but not a lot of samples.

## Statistical Commands

The following commands appear in the platform menu depending on context.

**Table 17.3** Categorical Platform Commands

	Command	Supported Response Contexts	Question	Details
These require multiple X columns	Test Response Homogeneity	<ul style="list-style-type: none"> <li>• Separate Responses</li> <li>• Aligned Responses</li> <li>• Repeated Measures</li> <li>• Response Frequencies (if no Sample Size)</li> </ul>	Are the probabilities across the response categories the same across sample categories?	Marginal Homogeneity (Independence) Test, both Pearson and Chi-square likelihood ratio chi-square
	Test Each Response	<ul style="list-style-type: none"> <li>• Multiple Response</li> <li>• Multiple Response by ID (with Sample Size)</li> <li>• Multiple Delimited</li> <li>• Response Frequencies with Sample Size</li> </ul>	For each response category, are the rates the same across sample categories?	Poisson regression on sample for each defect frequency.
	Agreement Statistic	<ul style="list-style-type: none"> <li>• Rater Agreement</li> </ul>	How closely do raters agree, and is the lack of agreement symmetrical?	Kappa for agreement, Bowker/McNemar for symmetry.
	Transition Report	<ul style="list-style-type: none"> <li>• Repeated Measures</li> </ul>	How have the categories changed across time?	Transition counts and rates matrices.

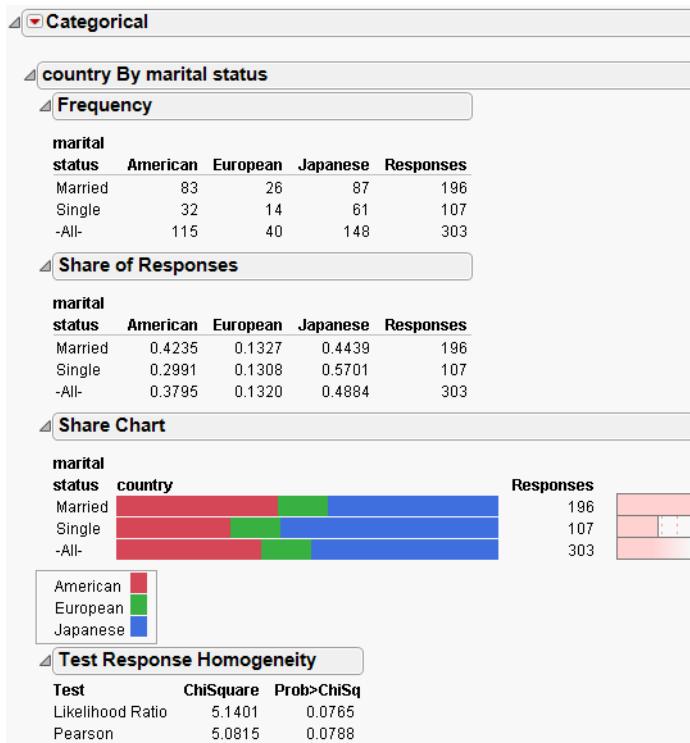
### Test Response Homogeneity

In this situation, there is typically one categorical response variable and one categorical sample variable. Multiple sample variables are treated as a single variable.

## Categorical Reports

The test is the chi-square test for marginal homogeneity of response patterns, testing that the response probabilities are the same across samples. This is equivalent to a test for independence when the sample category is like a response. There are two versions of this test, the Pearson form and the Likelihood Ratio form, both with chi-square statistics. The **Test Options** menu is used to show or hide the Likelihood Ratio or Pearson tests.

As an example, open Car Poll.jmp and launch the Categorical platform. Choose country as a **Separate Response** and marital status as an **X, Grouping Category**. When the report appears, select **Test Response Homogeneity** from the platform menu.



The Share Chart seems to indicate that the married group has higher probability to buy American cars, and the single group has higher probability to buy Japanese cars, but the statistical test only shows a significance of 0.08. Therefore, the difference in response probabilities across marital status is not statistically significant at an alpha level of 0.05.

**Test Each Response**

When there are multiple responses, each response category can be modeled separately. The question is whether the response rates are the same across samples. For each response category, we assume the frequency count has a random Poisson distribution. The rate test is obtained using a Poisson regression (through generalized linear models) of the frequency per unit modeled by the sample categorical variable. The result is a likelihood ratio chi-square test of whether the rates are different across samples.

This test can also be done by the **Pareto** platform, as well as in the **Generalized Linear Model** personality of the **Fit Model** platform.

As an example, open Failure3Freq.jmp. We want to compare the samples across the **clean** treatment variable, so launch the Categorical platform and assign all the defect columns as **Response Frequencies**, with **clean** as an **X, Grouping** variable and **SampleSize** as **Sample Size**. Click **OK**. Select **Test Each Response** from the platform menu.



For which defects are the rates significantly different across the **clean** treatments? The *p*-values show that **oxide defect** is the most significantly different, followed by **contamination**, then **doping**. The other defects are not significantly different with this amount of data.

### Relative Risk

The **Relative Risk** option is used to compute relative risks for different responses. The risk of responses is computed for each level of the **X, Grouping** variable. The risks are compared to get a relative risk. This option is available only when the **Unique occurrences within ID** box is checked on the platform launch window.

A common application of this analysis is when the responses represent adverse events (side effects), and the **X** variable represents a treatment (drug vs. placebo). The risk for getting each side effect is computed for both the drug and placebo. The relative risk is the ratio of the two risks.

### Conditional Association

The **Conditional Association** option is used to compute the conditional probability of one response given a different response. A table and color map of the conditional probabilities are given. This option is available only when the **Unique occurrences within ID** box is checked on the platform launch window.

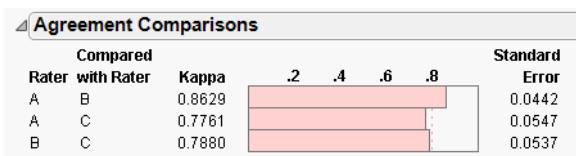
## Categorical Reports

A common application of this analysis is when the responses represent adverse events (side effects) from a drug. The computations represent the conditional probability of one side effect given the presence of another side effect.

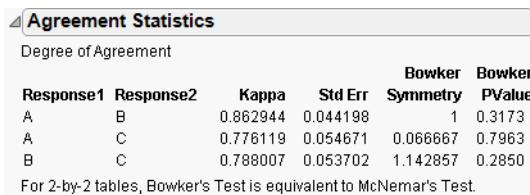
**Rater Agreement**

The Rater Agreement analysis answers the questions of how closely raters agree with one another and if the lack of agreement is symmetrical.

As an example, open Attribute Gauge.jmp. The Attribute Chart script runs the Variability Chart platform, which has a test for agreement among raters.



Launch the Categorical platform and designate the three raters (A, B, and C) as Rater Agreement responses in the launch dialog. In the resulting report, you have a similar test for agreement that is augmented by a symmetry test that the lack of agreement is symmetric.

**Save Tables**

The Save Tables menu on the platform red-triangle menu has the following options:

**Save Frequencies** saves the Frequency report to a new data table, without the marginal totals.

**Save Share of Responses** saves the Share of Responses report to a new data table, without the marginal totals.

**Save Rate Per Case** saves the Rate Per Case report to a new data table, without the marginal totals.

**Save Transposed Frequencies** saves the Transposed Freq Chart report to a new data table, without the marginal totals.

**Save Transposed Share of Responses** saves a transposed version of the Share of Responses report to a new data table.

**Save Transposed Rate Per Case** saves a transposed version of the Rate Per Case report to a new data table.

**Save Test Rates** saves the results of the Test Each Response option to a new data table.

**Save Test Homogeneity** saves the results of the Test Response Homogeneity option to a new data table. This option is available only when using Aligned Responses.



# Chapter **18**

## **Choice Modeling**

### **Choice Platform**

---

The Choice platform is designed for use in market research experiments, where the ultimate goal is to discover the preference structure of consumers. Then, this information is used to design products or services that have the attributes most desired by consumers.

Features provided in the Choice platform include:

- ability to use information about consumer traits as well as product attributes.
- integration of data from one, two, or three sources.
- ability to use the integrated profiler to understand, visualize, and optimize the response (utility) surface.
- provides subject-level scores for segmenting or clustering your data.
- uses a special default bias-corrected maximum likelihood estimator described by Firth (1993). This method has been shown to produce better estimates and tests than MLEs without bias correction. In addition, bias-corrected MLEs ameliorate separation problems that tend to occur in logistic-type models. Refer to Heinze and Schemper (2002) for a discussion of the separation problem in logistic regression.

The Choice platform is not appropriate to use for fitting models that involve:

- ranking or scoring.
- nested hierarchical choices. (PROC MDC in SAS/ETS can be used for such analysis.)

# Contents

Introduction to Choice Modeling . . . . .	381
Choice Statistical Model . . . . .	381
Product Design Engineering . . . . .	381
Data for the Choice Platform . . . . .	382
Example: Pizza Choice . . . . .	382
Launch the Choice Platform and Select Data Sets . . . . .	384
Choice Model Output . . . . .	387
Subject Effects . . . . .	389
Utility Grid Optimization . . . . .	391
Platform Options . . . . .	392
Example: Valuing Trade-offs . . . . .	393
One-Table Analysis . . . . .	399
Example: One-Table Pizza Data . . . . .	400
Segmentation . . . . .	402
Special Data Rules . . . . .	406
Default choice set . . . . .	406
Subject Data with Response Data . . . . .	407
Logistic Regression . . . . .	407
Transforming Data . . . . .	411
Transforming Data to Two Analysis Tables . . . . .	411
Transforming Data to One Analysis Table . . . . .	415

---

## Introduction to Choice Modeling

Choice modeling, pioneered by McFadden (1974), is a powerful analytic method used to estimate the probability of individuals making a particular choice from presented alternatives. Choice modeling is also called conjoint modeling, discrete choice analysis, and conditional logistic regression.

The Choice Modeling platform uses a form of conditional logistic regression. Unlike simple logistic regression, choice modeling uses a linear model to model choices based on response attributes and not solely upon subject characteristics. For example, in logistic regression, the response might be whether you buy brand A or brand B as a function of ten factors or characteristics that describe you such as your age, gender, income, education, etc. However, in choice modeling, you might be choosing between two cars that are a compound of ten attributes such as price, passenger load, number of cup holders, color, GPS device, gas mileage, anti-theft system, removable-seats, number of safety features, and insurance cost.

### Choice Statistical Model

Parameter estimates from the choice model identify consumer *utility*, or marginal utilities in the case of a linear utility function. Utility is the level of satisfaction consumers receive from products with specific attributes and is determined from the parameter estimates in the model.

The choice statistical model is expressed as follows.

Let  $X[k]$  represent a subject attribute design row, with intercept

Let  $Z[j]$  represent a choice attribute design row, without intercept

Then, the probability of a given choice for the  $k$ 'th subject to the  $j$ 'th choice of  $m$  choices is

$$P_i[jk] = \frac{\exp(\beta'(X[k] \otimes Z[j]))}{\sum_{l=1}^m \exp(\beta'(X[k] \otimes Z[l]))}$$

where

$\otimes$  is the Kronecker row-wise product

the numerator calculates for the  $j$ 'th alternative actually chosen, and

the denominator sums over the  $m$  choices presented to the subject for that trial.

---

## Product Design Engineering

When engineers design a product, they routinely make hundreds or thousands of small design decisions. Most of these decisions are not tested by prospective customers. Consequently, these products are not optimally designed. However, if customer testing is not too costly and test subjects (prospective customers) are readily available, it is worthwhile to test more of these decisions via consumer choice experiments.

Modeling costs have recently decreased with improved product and process development techniques and methodologies. Prototyping, including pure digital prototyping, is becoming less expensive, so it is possible

to evaluate the attributes and consequences of more alternatives. Another important advancement is the use of the Internet to deliver choice experiments to a wide audience. You can now inform your customers that they can have input into the design of the next product edition by completing a web survey.

Choice modeling can be added to Six Sigma programs to improve consumer products. Six Sigma aims at making products better by improving the manufacturing process and ensuring greater performance and durability. But, Six Sigma programs have not addressed one very important aspect of product improvement—making the products that people actually want. Six Sigma programs often consider the Voice of the Customer and can use customer satisfaction surveys. However, while these surveys can disclose what is wrong with the product, they fail to identify consumer preferences with regard to specific product attributes. Choice experiments provide a tool that enables companies to gain insight for actual customer preferences. Choice modeling analysis can reveal such preferences.

Market research experiments have a long history of success, but performing these experiments has been expensive, and research has previously focused on price elasticity and competitive situations. It is by using these same techniques for product design engineering where choice modeling can have the most impact.

---

## Data for the Choice Platform

The Choice platform is unique because it is designed to use data from one, two or three different data tables.

**Profile data** describe the attributes associated with each choice. Each choice can comprise many different attributes, and each attribute is listed as a column in the data table. There is a row for each possible choice, and each possible choice contains a unique ID.

**Response data** contain the experimental results and have the choice set IDs for each trial as well as the actual choice selected by the subject. Each subject usually has several trials, or *choice sets*, to cover several choice possibilities. There can be more than one row of data for each subject. For example, an experiment might have 100 subjects with each subject making 12 choice decisions, resulting in 1200 rows in this data table. The Response data are linked to the Profile data through the choice set columns and the actual choice response column. Choice set refers to the set of alternatives from which the subject makes a choice. Grouping variables are sometimes used to align choice indices when more than one group is contained within the data.

**Subject data** are optional, depending on whether subject effects are to be modeled. This source contains one or more attributes or characteristics of each subject and a subject identifier. The Subject data table contains the same number of rows as subjects and has an identifier column that matches a similar column in the Response data table. You can also put Subject data in the Response data table, but it is still specified as a subject table.

If all your data are contained in one table, you can use the Choice platform, but additional effort is necessary. See the section “[One-Table Analysis](#),” p. 399.

---

## Example: Pizza Choice

Suppose that you are supplying pizza for an airline. You want to find pizza attributes that are optimal for the flying population. So, you have a group of frequent flyers complete a choice survey. In order to weigh the

importance of each attribute and to determine whether there are any interactions between the different attributes, you give them a series of choices that require them to state their preference between each pair of choices. One pair of choices might be between two types of pizza that they like, or between two types of pizza that they do not like. Hence, the choice might not always be easy.

This example examines pizza choices where three attributes, each with two levels, are presented to the subjects:

- crust (thick or thin),
- cheese (mozzarella or Monterey Jack),
- topping (pepperoni or none).

Suppose a subject likes thin crust with mozzarella cheese and no topping, but the choices given to the subject are either a thick crust with mozzarella cheese and pepperoni topping, or a thin crust with Monterey Jack cheese and no topping. Since neither of these pizzas is ideal, the subject has to weigh which of the attributes are more important.

The profile data table lists all the pizza choice combinations that you want to present to the subjects. Each choice combination is given an ID. The profile data table is shown in Figure 18.1.

---

**Figure 18.1** Pizza Profile Data Table



	Crust	Cheese	Topping	ID
1	Thick	Mozzarella	Pepperoni	ThickOni
2	Thick	Mozzarella	None	ThickElla
3	Thick	Jack	Pepperoni	ThickJackoni
4	Thick	Jack	None	ThickJack
5	Thin	Mozzarella	Pepperoni	TrimOni
6	Thin	Mozzarella	None	Trimella
7	Thin	Jack	Pepperoni	TrimPepperjack
8	Thin	Jack	None	TrimJack

---

For the actual survey or experiment, each subject is given four trials, where each trial consists of stating his or her preference between two choice profiles (Choice1 and Choice2). The choice profiles given for each trial are referred to as a choice set. One subject's choice trials can be different from another subject's trials. Refer to the DOE Choice Design platform for generating optimal choice designs. Twelve runs from the first three subjects are shown in Figure 18.2.

**Figure 18.2** Pizza Response Data Table Segment

	Subject	Choice1	Choice2	Choice
1	1	ThickJack	TrimPepperjack	TrimPepperjack
2	1	TrimPepperjack	ThickElla	ThickElla
3	1	TrimOni	Trimella	TrimOni
4	1	ThickElla	ThickJack	ThickElla
5	2	Trimella	ThickJacksoni	Trimella
6	2	TrimJack	ThickElla	ThickElla
7	2	Trimella	TrimPepperjack	Trimella
8	2	TrimPepperjack	TrimOni	TrimOni
9	3	TrimOni	ThickJacksoni	TrimOni
10	3	TrimPepperjack	ThickElla	ThickElla
11	3	ThickJacksoni	TrimPepperjack	ThickJacksoni
12	3	ThickOni	Trimella	ThickOni
13	4	ThickElla	ThickOni	ThickElla
14	4	TrimPepperjack	ThickJack	ThickJack

Notice that each choice value refers to an ID value in the Profile data table which has the attribute information.

If data about the subject are to be used, a separate Subject data table is needed. This table includes a subject ID column and characteristics of the subject. In the pizza example, the only characteristic or attribute about the Subject is Gender. Subject data for the first four subjects are shown in Figure 18.3. Notice that the response choices and choice sets in the response data table use the ID names given in the profile data set. Similarly, the subject identifications in the response data table match those in the subject data table.

**Figure 18.3** Pizza Subject Data Table Segment

	Subject	Gender
1	1	M
2	2	F
3	3	M
4	4	F

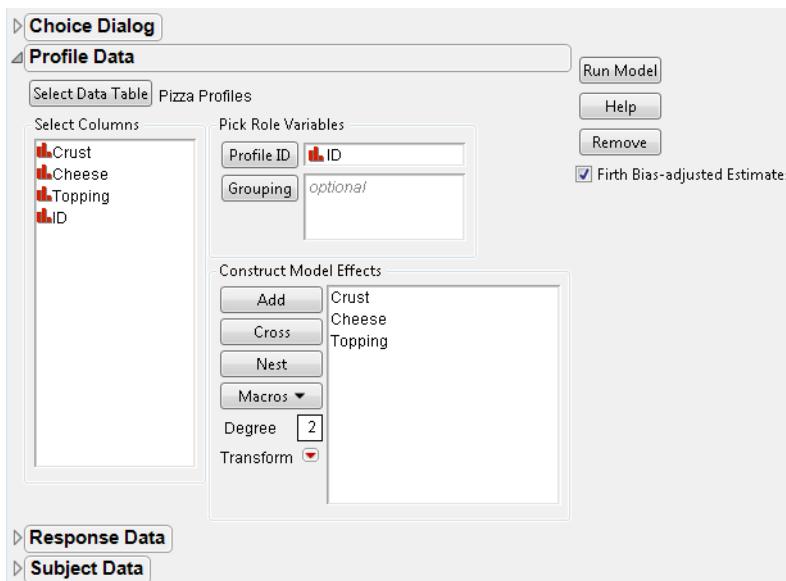
## Launch the Choice Platform and Select Data Sets

Since the Choice platform can use several data tables, no initial assumption is made about using the current data table—as is the case with other JMP platforms. You must select the data table for each of the three choice data sources. You are prompted to select the profile data set and the response data set. If you want to model subject attributes, then a subject data set must also be selected. You can expand or collapse each section of the Choice dialog box, as needed.

To illustrate the Choice platform, three data sets from the pizza example are used and are found in the sample data directory. The first data set is entered into the Profile Data section of the Choice Dialog Box, as shown in Figure 18.4.

- Select **Analyze > Modeling > Choice** to open the launch dialog box. You see three separate sections for each of the data sources.
- Click on **Select Data Table** under Profile Data. A new dialog box opens, which prompts you to specify the data table for the profile data. You can select from any of the data sets already open in the current JMP session, or you can select **Other**. Selecting **Other** allows you to open a file that is not currently open.
- Select **Pizza Profiles.jmp**. The columns from this table now populate the field under **Select Columns** in the Choice Dialog box.
- Select **ID** for **Profile ID** under **Pick Role Variables** and **Add Crust, Cheese, and Topping** under **Construct Model Effects**. If the **Profile ID** column does not uniquely identify each row in the profile data table, you need to add **Grouping** columns until the combination of **Grouping** and **Profile ID** columns uniquely identify the row, or profile. For example, if **Profile ID = 1** for **Survey = A**, and a different **Profile ID = 1** for **Survey = B**, then **Survey** would be used as a **Grouping** column. In this simple experiment, all eight combinations of the three two-level factors were used.

**Figure 18.4** Profile Data Set Dialog Box

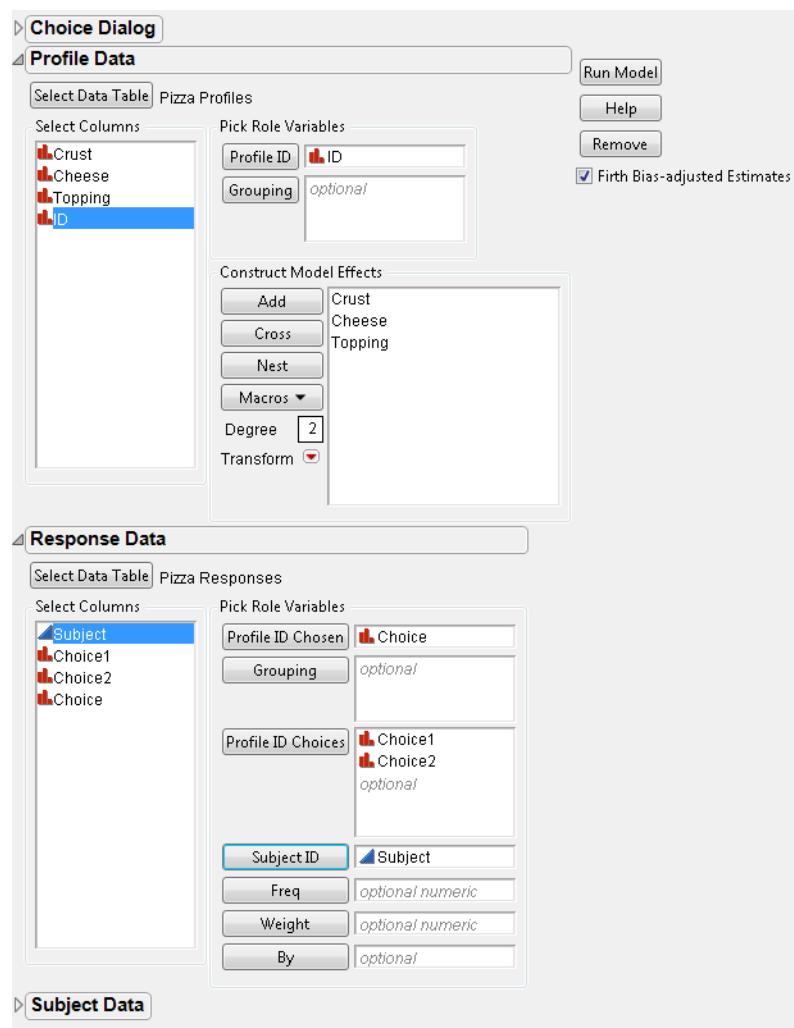


The second data set is the Response Data containing the experimental results. For the pizza example, **Choice1** and **Choice2** are the profile ID choices given to a subject on each of four trials. The **Choice** column contains the chosen preference between **Choice1** and **Choice2**.

- Open the Response Data section of the dialog box. Click **Select Data Table**. When the Response Data Table dialog box opens, select **Pizza Responses.jmp**.
- Select **Choice** for the **Profile ID Chosen**, and **Choice1** and **Choice2** for the **Profile ID Choices**.

## Launch the Choice Platform and Select Data Sets

- The **Subject ID** is optional, but is needed if you later want to model subject effects or if you later want to cluster subjects into groups.
- Freq** and **Weight** are used to weight the analysis. For example, if you have summarized a set that had the same trial **Freq** is used for the count in that row of the response data table.
- Select **Subject** for **Subject ID** to identify individual subjects for later analysis. If you are not interested in assessing subjects at this time, it is not necessary to enter the subject data set into the model. The completed dialog box, without the subject data set, is shown in Figure 18.5.

**Figure 18.5** Response Data Set Dialog Box

If you are scripting the Choice platform, you can also set the acceptable criterion for convergence when estimating the parameters by adding this command to the `Choice()` specification:

```
Choice( ..., Convergence Criterion( fraction ), ... )
```

See the Object Scripting Index for an example.

## Choice Model Output

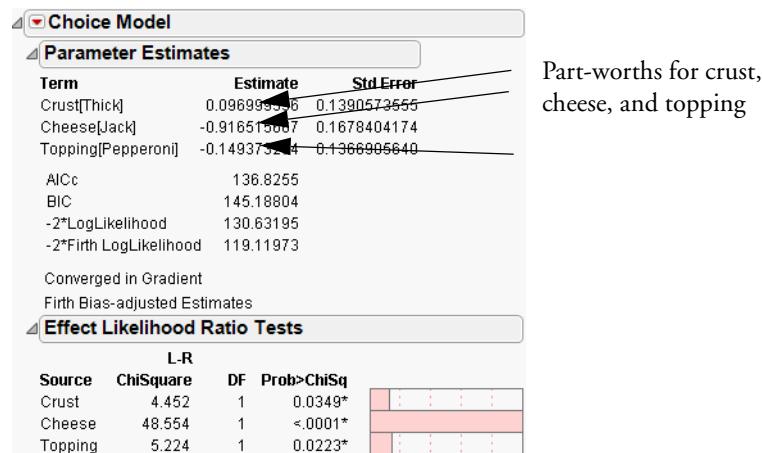
Click on **Run Model** to obtain the results. These results are shown in Figure 18.6.

- The resulting parameter estimates are sometimes referred to as *part-worths*. Each part-worth is the coefficient of utility associated with that attribute. By default, these estimates are based upon the Firth bias-corrected maximum likelihood estimators, and are, therefore, considered to be more accurate than MLEs without bias correction.
- Comparison criteria are used to help determine the better-fitting model(s) when more than one model is investigated for your data. The model with the lower or lowest criterion value is believed to be the better or best model. Three criteria are shown in the Choice Model output and include AICc (corrected Akaike's Information Criterion), -2\*LogLikelihood, and -2\*Firth Loglikelihood. The AICc formula is:

$$\text{AICc} = -2\log\text{likelihood} + 2k + \frac{2k(k+1)}{n-k-1}$$

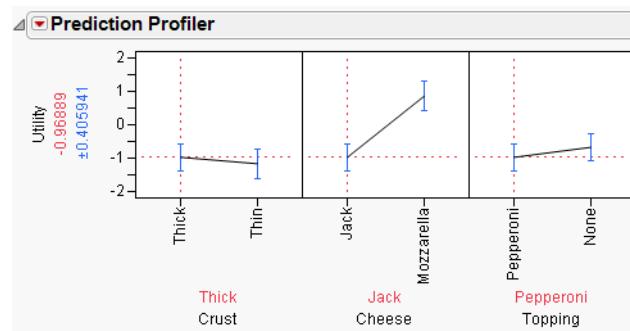
where  $k$  is the number of estimated parameters in the model and  $n$  is the number of observations in the dataset. Note that the -2\*Firth Loglikelihood result is only included in the report when the Firth Bias-adjusted Estimates checkbox is checked in the launch window. (See Figure 18.5.) This option is checked by default. The decision to use or not use the Firth Bias-adjusted Estimates does not affect the AICc score or the -2\*LogLikelihood results.

- Likelihood ratio tests appear for each effect in the model. These results are obtained by default if the model is fit quickly (less than five seconds); otherwise, you can click on the Choice Model drop down menu and select **Likelihood Ratio Tests**.

**Figure 18.6** Choice Model Results with No Subject Data for Pizza Example

The profiler option is particularly valuable in understanding the model. It shows the value of the linear model, or the utility, as you change each factor, one at a time.

- You specify the profiler option by selecting **Profiler** in the platform menu. The Prediction Profiler is shown in Figure 18.7.
- In this example involving only main effects, the factor showing the greatest difference is Cheese, favoring Mozzarella.
- If there were interactions in the model, more exploration of the profiler would be needed in order to understand the response surface.

**Figure 18.7** Prediction Profiler with No Subject Data for Pizza Example

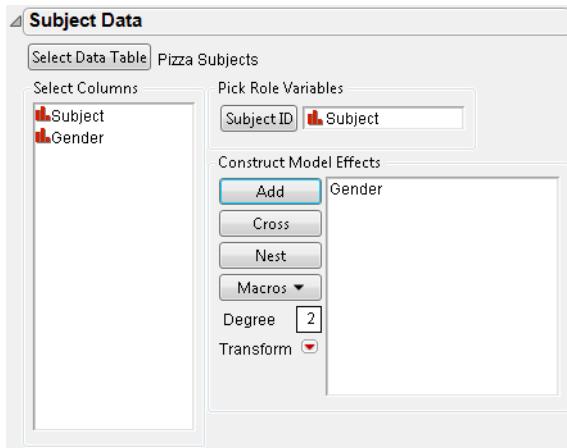
## Subject Effects

If you want to include subject effects in the model, you need to open the Subject data table section of the Choice dialog box. Suppose you are interested in Gender main effects.

- Open the Subject Data section of the launch dialog box. Click **Select Data Table**, and when the Subject Data Table opens, select Pizza Subjects.jmp.
- Specify Subject as **Subject ID**, and add Gender under **Construct Model Effects**. The Subject data dialog box section for the pizza example is shown in Figure 18.8.
- Click on **Run Model**.

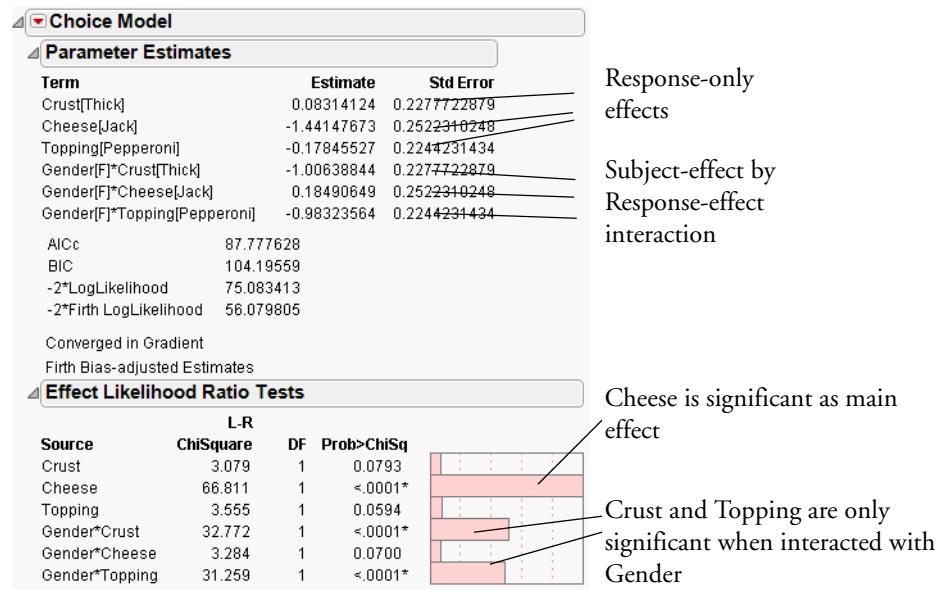
---

**Figure 18.8** Choice Model Subject Data Dialog Box for Pizza Example



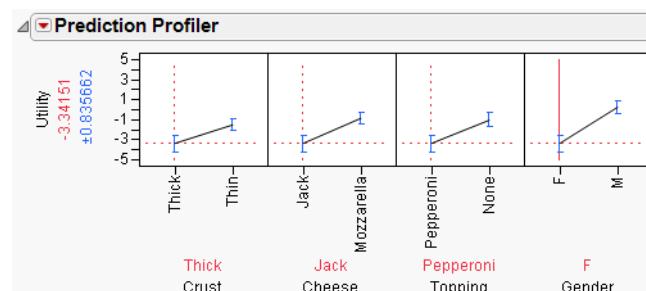
---

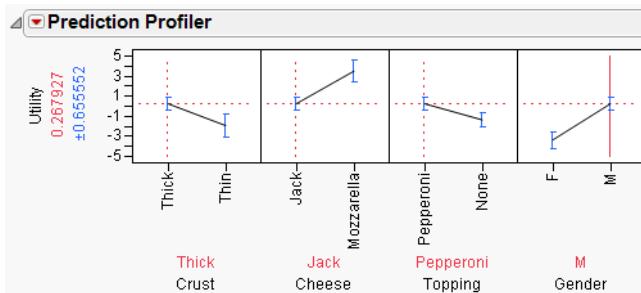
Figure 18.9 shows the parameter estimates and the likelihood ratio tests for the Choice Model with subject effects included. Strong interactions are seen between Gender and Crust and between Gender and Topping. When the Crust and Topping factors are assessed for the entire population, the effects are not significant. However, the effects of Crust and Topping are strong when they are evaluated between Gender groups.

**Figure 18.9** Choice Model Results with Subject Effects for Pizza Example

The profiler is used to explore the response surface of the model. Click on the Choice Model drop-down menu and select the **Profiler**. You can Alt-click on any segment of the Profiler graphics to lock that particular factor level. (F is locked by default in this example.) A solid vertical line appears in place of the dotted line. Other factor settings can then be assessed easily for the locked factor level.

- As shown in Figure 18.10, when the female (F) level of Gender is locked, the Profiler shows that females prefer pizza with thin crust, mozzarella cheese, and no topping. For example, the Utility measure is higher for Crust equals Thin, meaning that females prefer thin crust.
- Now, switch Gender to M to assess male pizza preferences. As shown in Figure 18.11, males prefer thick crust, mozzarella cheese, and pepperoni topping.

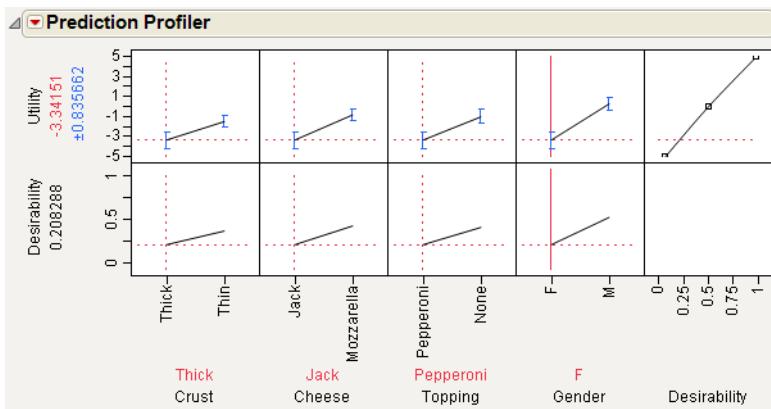
**Figure 18.10** Prediction Profiler with Subject Data and Female Level Factor Setting

**Figure 18.11** Prediction Profiler with Subject Data and Male Level Factor Setting

## Utility Grid Optimization

The Prediction Profiler allows you to optimize the utility function over a grid of fixed subject factor settings without having to manually manipulate profiler settings:

- Click on the platform drop-down-menu, select **Profiler**, and verify that one of the subject factors is locked. If not, Alt-click within the Profiler plot to lock the desired factor level. Set the Utility function to the maximum values for the other factors by sliding the red dotted vertical line.
- Click on the red-triangle menu of the Prediction Profiler and select **Desirability Functions**. A new row is added to the Prediction Profiler, displaying overall desirability traces and measures. Utility and Desirability Functions are shown together in Figure 18.12.

**Figure 18.12** Utility and Desirability Functions

- From the red-triangle menu of the Prediction Profiler, select **Maximize for each Grid Point**. A Remembered Settings table containing the grid settings with the maximum utility and desirability functions, and a table of differences between grids is displayed. See Figure 18.13. As illustrated, this

feature can be a very quick and useful tool for selecting the most desirable attribute combinations for a factor.

**Figure 18.13** Utility and Desirability Settings

The screenshot shows a software interface for choice modeling. It displays two tables: 'Remembered Settings' and 'Differences'.

**Remembered Settings:**

Setting	Crust	Cheese	Topping	Gender	Utility	Desirability
Grid1	Thin	Mozzarella	None	F	3.3415083	0.820201
Grid2	Thick	Mozzarella	Pepperoni	M	3.5206933	0.837771

**Differences:**

Setting A - Setting B	Utility	Desirability
Grid2 - Grid1	0.1791849	0.0175696

The grid setting for females shows that the greatest utility and desirability values are obtained when the pizza attributes are thin crust, mozzarella cheese, and no topping. For males, the grid setting shows that the highest utility and desirability values are obtained when the pizza attributes are thick crust, mozzarella cheese, and pepperoni topping.

## Platform Options

The Choice Modeling platform has many available options. To access these options, click on the platform drop-down menu.

**Likelihood Ratio Tests** tests the significance of each effect in the model. These are done by default if the estimate of cpu time is less than five seconds.

**Joint Factor Tests** tests each factor in the model by constructing a likelihood ratio test for all the effects involving that factor.

**Confidence Intervals** produces a 95% confidence interval for each parameter (by default), using the profile-likelihood method. Shift-click on the platform drop-down menu and select **Confidence Intervals** to input alpha values other than 0.05.

**Correlation of Estimates** shows the correlations of the parameter estimates.

**Effect Marginals** shows the fitted utility values for different levels in the effects, with neutral values used for unrelated factors.

**Profiler** produces a response surface viewer that takes vertical cross-sections across each factor, one at a time.

**Save Utility Formula** makes a new column with a formula for the utility, or linear model, that is estimated. This is in the profile data table, except if there are subject effects. In that case, it makes a new data table for the formula. This formula can be used with various profilers with subsequent analyses.

**Save Gradients by Subject** constructs a new table that has a row for each subject containing the average (Hessian-scaled-gradient) steps on each parameter. This corresponds to using a Lagrangian multiplier test for separating that subject from the remaining subjects. These values can later be clustered, using the built-in-script, to indicate unique market segments represented in the data.

**Model Dialog** shows the Choice dialog box, which can be used to modify and re-fit the model. You can specify new data sets, new IDs, and new model effects.

---

## Example: Valuing Trade-offs

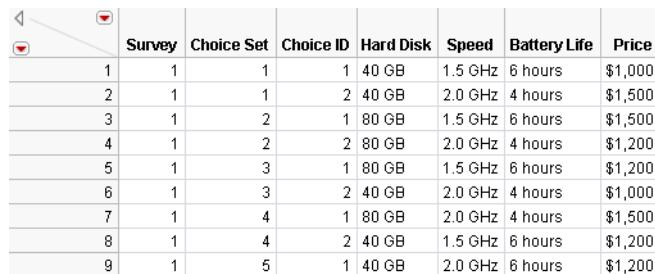
The Choice Modeling platform is also useful for determining the relative importance of product attributes. Even if the attributes of a particular product that are important to the consumer are known, information about preference trade-offs with regard to these attributes might be unknown. By gaining such information, a market researcher or product designer is able to incorporate product features that represent the optimal trade-off from the perspective of the consumer.

The advantages of this approach to product design can be found in the following example. It is already known that four attributes are important for laptop design--hard-disk size, processor speed, battery life, and selling price. The data gathered for this study are used to determine which of four laptop attributes (Hard Disk, Speed, Battery Life, and Price) are most important. It also assesses whether or not there are Gender or Job differences seen with these attributes.

- Select **Analyze > Modeling > Choice** to open the launch dialog box.
- Open Laptop Profile.jmp from the sample data directory and **Select Data Table** under Profile Data. Select Laptop Profile.jmp. A partial listing of the Profile Data table is shown in Figure 18.14. The complete data set consists of 24 rows, 12 for Survey 1 and 12 for Survey 2. Survey and Choice Set define the grouping columns and Choice ID represents the four attributes of laptops: Hard Disk, Speed, Battery Life, and Price.

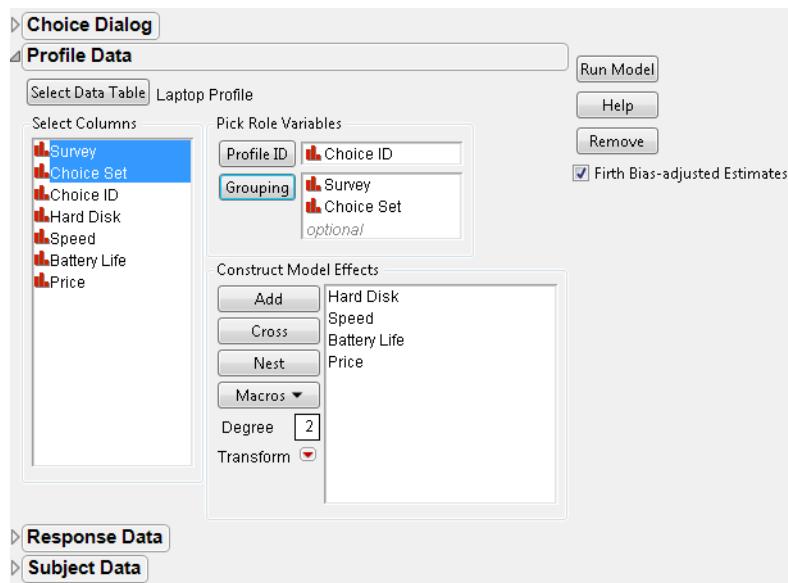
---

**Figure 18.14** Profile Data Set for the Laptop Example

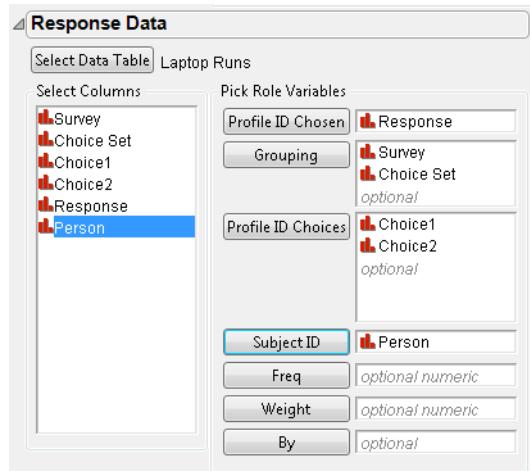


	Survey	Choice Set	Choice ID	Hard Disk	Speed	Battery Life	Price
1	1	1	1	40 GB	1.5 GHz	6 hours	\$1,000
2	1	1	2	40 GB	2.0 GHz	4 hours	\$1,500
3	1	2	1	80 GB	1.5 GHz	6 hours	\$1,500
4	1	2	2	80 GB	2.0 GHz	4 hours	\$1,200
5	1	3	1	80 GB	1.5 GHz	6 hours	\$1,200
6	1	3	2	40 GB	2.0 GHz	4 hours	\$1,000
7	1	4	1	80 GB	2.0 GHz	4 hours	\$1,500
8	1	4	2	40 GB	1.5 GHz	6 hours	\$1,200
9	1	5	1	40 GB	2.0 GHz	6 hours	\$1,200

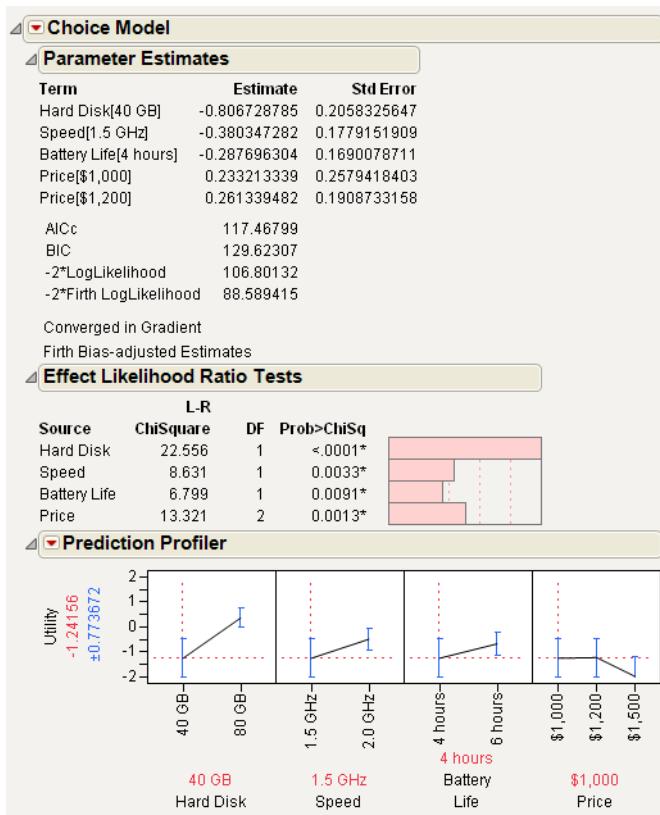
- 
- Select Choice ID for Profile ID, and **ADD** Hard Disk, Speed, Battery Life, and Price for the model effects.
  - Select Survey and Choice Set as the **Grouping** columns. The Profile Data dialog box is shown in Figure 18.15.

**Figure 18.15** Profile Data Dialog Box for Laptop Study

- Click on **Response Data > Select Data Table > Other > OK** and select Laptop Runs.jmp from the sample data directory.
- Select **Response** as the **Profile ID Chosen**, **Choice1** and **Choice2** as the **Profile ID Choices**, **Survey** and **Choice Set** as the **Grouping** columns, and **Person** as **Subject ID**. The Response Data dialog box is shown in Figure 18.16.

**Figure 18.16** Response Data Dialog Box for Laptop Study

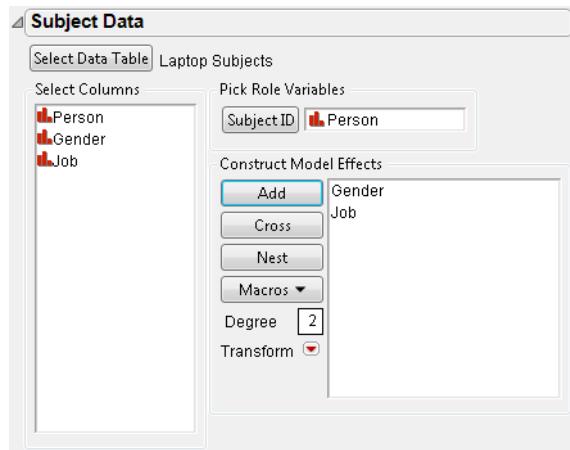
- To run the model without subject effects, click on **Run Model**.
- Choose **Profiler** from the red-triangle menu.

**Figure 18.17** Laptop Results Without Subject Effects

Results of this study show that while all the factors are important, the most important factor in the laptop study is Hard Disk. The respondents prefer the larger size. Note that respondents did not think a price increase from \$1000 to \$1200 was important, but an increase from \$1200 to \$1500 was considered important. This effect is easily visualized by examining the factors interactively with the Prediction Profiler. Such a finding can have implications for pricing policies, depending on external market forces.

To include subject effect for the laptop study, simply add to the Choice Modeling dialog box:

- Under Subject Data, **Select Data Table > Other > OK > Laptop Subjects.jmp**.
- Select Person as **Subject ID** and ADD Gender and Job as the model effects. The Subject Data dialog box is shown in Figure 18.18.

**Figure 18.18** Subject Dialog Box for Laptop Study

- Click on **Run Model**.

Results are shown in Figure 18.19, Figure 18.20, and Figure 18.23.

**Figure 18.19** Laptop Parameter Estimate Results With Subject Data

Term	Estimate	Std Error
Hard Disk[40 GB]	-0.887027357	0.2421091374
Speed[1.5 GHz]	-0.362097199	0.1907670046
Battery Life[4 hours]	-0.394080806	0.2025662868
Price[\$1,000]	0.344799986	0.3094758460
Price[\$1,200]	0.305771815	0.2112382529
Gender[F]*Hard Disk[40 GB]	-0.389344918	0.2594789312
Gender[F]*Speed[1.5 GHz]	0.017754116	0.2229164234
Gender[F]*Battery Life[4 hours]	-0.250255571	0.2306512647
Gender[F]*Price[\$1,000]	0.327081163	0.3293378134
Gender[F]*Price[\$1,200]	0.103980693	0.2219937705
Job[Development]*Hard Disk[40 GB]	0.083419819	0.2335285785
Job[Development]*Speed[1.5 GHz]	-0.150191535	0.2167139358
Job[Development]*Battery Life[4 hours]	-0.0565889926	0.2115342604
Job[Development]*Price[\$1,000]	0.042732902	0.2947913725
Job[Development]*Price[\$1,200]	0.134544817	0.2100258497
AICc	129.35431	
BIC	161.81953	
-2*LogLikelihood	93.354306	
-2*Firth LogLikelihood	41.678087	
Converged in Gradient		
Firth Bias-adjusted Estimates		

**Figure 18.20** Laptop Likelihood Ratio Test Results With Subject Data

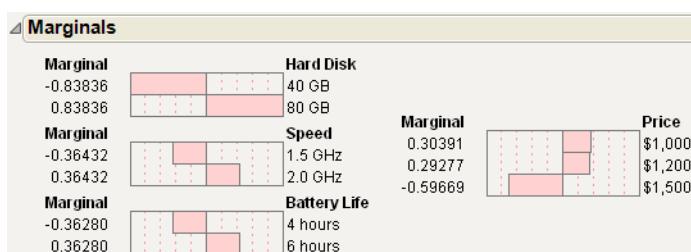
Effect Likelihood Ratio Tests			
Source	ChiSquare	DF	Prob>ChiSq
Hard Disk	23.197	1	<.0001*
Speed	7.417	1	0.0065*
Battery Life	8.039	1	0.0046*
Price	14.624	2	0.0007*
Gender*Hard Disk	5.433	1	0.0198*
Gender*Speed	3.003	1	0.0831
Gender*Battery Life	4.422	1	0.0355*
Gender*Price	7.889	2	0.0194*
Job*Hard Disk	3.046	1	0.0810
Job*Speed	3.549	1	0.0596
Job*Battery Life	3.188	1	0.0742
Job*Price	6.512	2	0.0385*

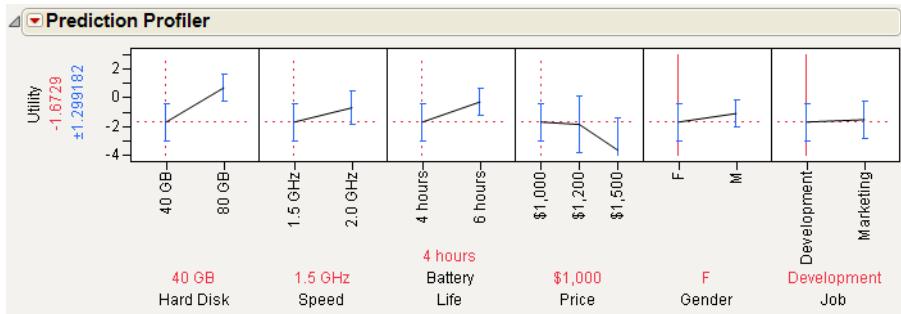
Selecting **Joint Factor Tests** from the platform menu gives the table shown in Figure 18.21. Notice that both Gender and Job are significant.

**Figure 18.21** Joint Factor Test for Laptop

Joint Factor Tests			
Source	ChiSquare	DF	Prob>ChiSq
Hard Disk	31.085	3	<.0001*
Speed	15.714	3	0.0013*
Battery Life	15.012	3	0.0018*
Price	27.709	6	0.0001*
Gender	23.686	5	0.0002*
Job	20.188	5	0.0012*

Selecting **Effect Marginals** from the platform menu displays the table shown in Figure 18.22. The marginal effects of each level for each factor are displayed. Notice that the marginal effects for each factor across all levels sum to zero.

**Figure 18.22** Marginal Effects for Laptop

**Figure 18.23** Laptop Profiler Results for Females with Subject Data**Figure 18.24** Laptop Profiler Results for Males with Subject Data

Some interaction effects between subject effects and laptop attributes are significant statistically. For example, the interaction effect between **Gender** and **Hard Disk** is significant (see Figure 18.20). When you check the result in Prediction Profiler, you find that a larger slope is shown for **Hard Disk** in the Prediction Profiler for females.

## One-Table Analysis

The Choice Modeling platform can also be used if all of your data are in one table. For this one-table scenario, you use only the Profile Data section of the Choice Dialog box. Subject-specific terms can be used in the model, but not as main effects. Two advantages, both offering more model-effect flexibility than the three-table specification, are realized by using a one-table analysis:

- Interactions can be selectively chosen instead of automatically getting all possible interactions between subject and profile effects as seen when using three tables.
- Unusual combinations of choice sets are allowed. This means, for example, that the first trial can have a choice set of two, the second trial can consist of a choice set of three, the third trial can have a choice set

of five, and so on. With multiple tables, in contrast, it is assumed that the number of choices for each trial is fixed.

A choice response consists of a set of rows, uniquely identified by the Grouping columns. An indicator column is specified for Profile ID in the Choice Dialog box. This indicator variable uses the value of 1 for the chosen profile row and 0 elsewhere. There must be exactly one “1” for each Grouping combination.

## Example: One-Table Pizza Data

This example illustrates how the pizza data are organized for the one-table situation. Figure 18.25 shows a subset of the combined pizza data. Open *Pizza Combined.jmp* from the sample data directory to see the complete table. Each subject completes four choice sets, with each choice set or trial consisting of two choices. For this example, each subject has eight rows in the data set. The indicator variable specifies the chosen profile for each choice set. The columns Subject and Trial together identify the choice set, so they are the **Grouping** columns.

---

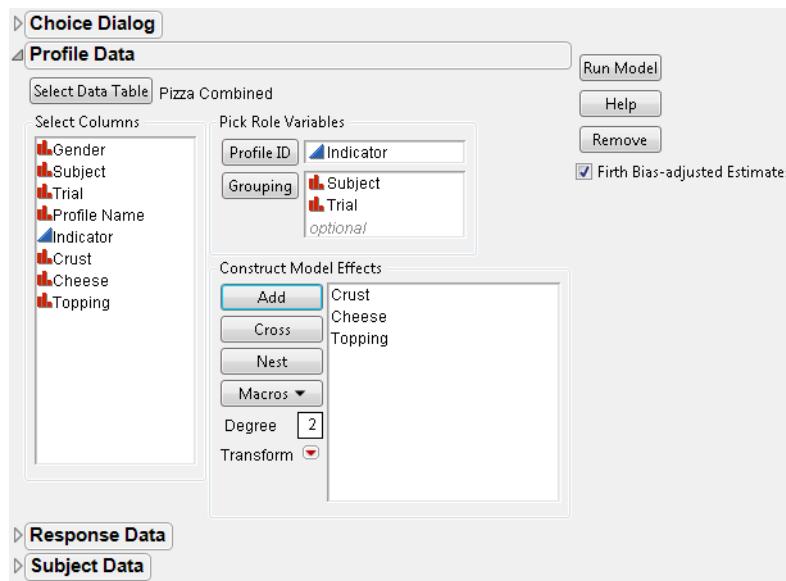
**Figure 18.25** Partial Listing of Combined Pizza Data for One-Table Analysis

	Gender	Subject	Trial	Profile Name	Indicator	Crust	Cheese	Topping
1	M		1	1 ThickJack	0	Thick	Jack	None
2	M		1	1 TrimPepperjack	1	Thin	Jack	Pepperoni
3	M		1	2 TrimPepperjack	0	Thin	Jack	Pepperoni
4	M		1	2 ThickElla	1	Thick	Mozzarella	None
5	M		1	3 TrimOni	1	Thin	Mozzarella	Pepperoni
6	M		1	3 Trimella	0	Thin	Mozzarella	None
7	M		1	4 ThickElla	1	Thick	Mozzarella	None
8	M		1	4 ThickJack	0	Thick	Jack	None

---

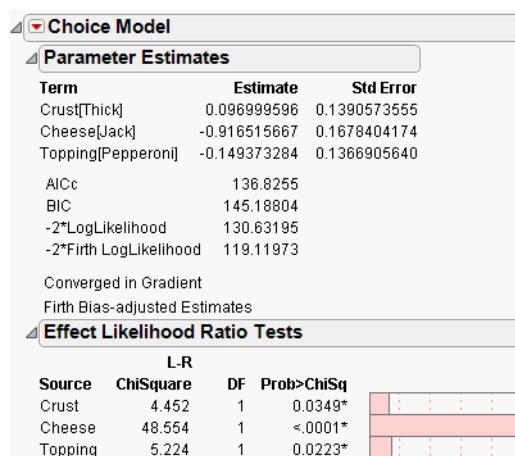
To analyze the data in this format, open the Profile Data section in the Choice Dialog box, shown in Figure 18.26.

- Specify *Pizza Combined.jmp* as the data set.
- Specify Indicator as the **Profile ID**, Subject and Trial as the **Grouping** variables, and add Crust, Cheese, and Topping as the main effects.
- Click **Run Model**.

**Figure 18.26** Choice Dialog Box for Pizza Data One-Table Analysis

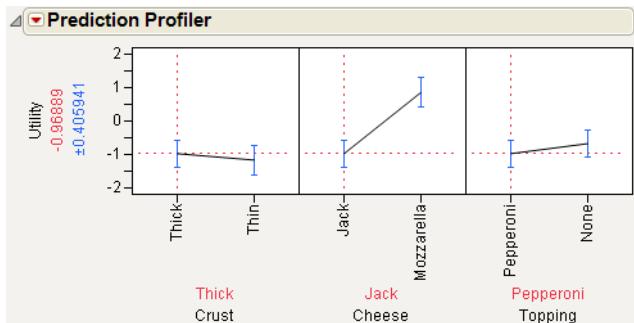
A new dialog box appears asking if this is a one-table analysis with all of the data in the Profile Table.

- Click **Yes** to fit the model, as shown in Figure 18.27.

**Figure 18.27** Choice Model for Pizza Data One-Table Analysis

- Select **Profiler** from the drop-down menu to obtain the results shown in Figure 18.28. Notice that the parameter estimates and the likelihood ratio test results are identical to the results obtained for the Choice Model with only two tables, shown in Figure 18.6 and Figure 18.7.

**Figure 18.28** Prediction Profiler for Pizza Data One-Table Analysis



## Segmentation

Market researchers sometimes want to analyze the preference structure for each subject separately in order to see whether there are groups of subjects that behave differently. However, there are usually not enough data to do this with ordinary estimates. If there are sufficient data, you can specify “By groups” in the Response Data or you could introduce a Subject identifier as a subject-side model term. This approach, however, is costly if the number of subjects is large. Other segmentation techniques discussed in the literature include Bayesian and mixture methods.

You can also use JMP to segment by clustering subjects using response data. For example, after running the model using the **Pizza Profiles.jmp**, **Pizza Responses.jmp**, and the optional **Pizza Subjects.jmp** data sets, click on the drop-down menu for the Choice Model platform and select **Save Gradients by Subject**. A new data table is created containing the average Hessian-scaled gradient on each parameter, and there is one row for each subject.

**Note:** This feature is regarded as an experimental method, since, in practice, little research has been conducted on its effectiveness.

These gradient values are the subject-aggregated Newton-Raphson steps from the optimization used to produce the estimates. At the estimates, the total gradient is zero, and

$$\Delta = H^{-1}g = 0 \text{ where } g \text{ is the total gradient of the log-likelihood evaluated at the MLE, and}$$

$H^{-1}$  is the inverse Hessian function or the inverse of the negative of the second partial derivative of the log-likelihood.

But, the disaggregation of  $\Delta$  results in

$$\Delta = \sum_{ij} \Delta_{ij} = \Sigma H^{-1} g_{ij} = 0$$

where  $i$  is the subject index,  $j$  is the choice response index for each subject,

$\Delta_{ij}$  are the partial Newton-Raphson steps for each run, and

$g_{ij}$  is the gradient of the log-likelihood by run.

The mean gradient step for each subject is then calculated as:

$$\bar{\Delta}_i = \sum_j \frac{\Delta_{ij}}{n_i} \text{ where } n_i \text{ is the number of runs per subject.}$$

These  $\bar{\Delta}_i$  are related to the force that subject  $i$  is applying to the parameters.

If groups of subjects have truly different preference structures, these forces are strong, and they can be used to cluster the subjects.

The  $\bar{\Delta}_i$  are the gradient forces that are saved.

A partial data table with these subject forces is shown in Figure 18.29.

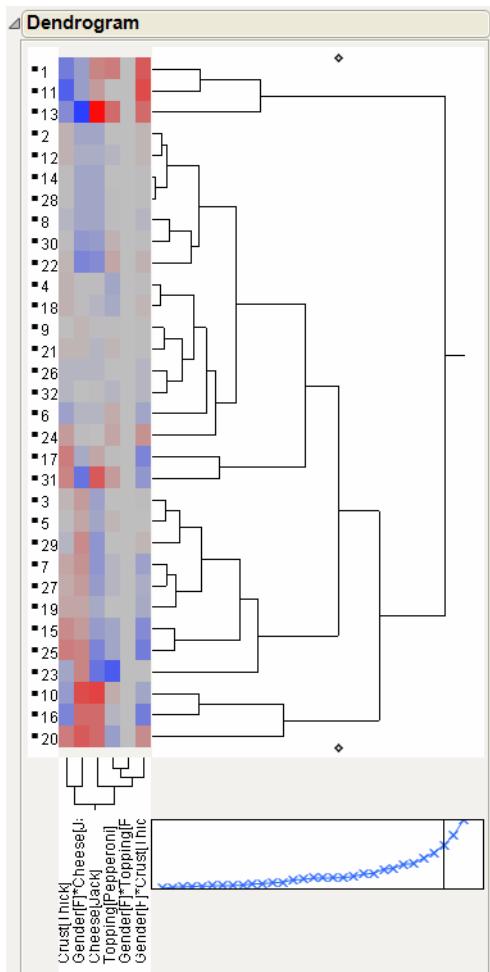
**Figure 18.29** Gradients by Subject for Pizza Data



	Subject	Crust[Thick]	Cheese[Jack]	Topping[Pepperoni]	Gender[F]Cru	Gender[F]Cheese[Jack]
1	1	-0.01256	0.009336	0.011205	0.015894	-0.00687
2	2	0.002347	-0.0048	-2.35e-5	0.002341	-0.0048
3	3	0.001321	-0.00615	-0.00045	-0.00145	0.006054
4	4	0.002459	-0.00053	-0.00488	0.001185	0.000643
5	5	0.000605	-0.00473	0.001945	-2.67e-5	0.005156
6	6	-0.0064	-0.00173	0.003591	-0.00547	-0.0026

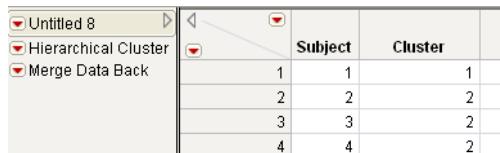
You can cluster these values by clicking on the drop-down menu of **Hierarchical Clustering** in the new data table and selecting **Run Script**. The resulting dendrogram of the clusters is shown in Figure 18.30.

**Figure 18.30** Dendrogram of Subject Clusters for Pizza Data



Now, select the number of clusters desired by moving the diamond indicator at the top or bottom of the dendrogram. Alternatively, you can select **Number of Clusters** in the platform drop-down menu and enter a number. You can save the cluster IDs by clicking on the drop-down menu of Hierarchical Clustering and selecting **Save Clusters**. A new column called **Cluster** is created in the data table containing the gradients. Each subject has been assigned a **Cluster** value that is associated with other subjects having similar gradient forces. Refer to “[Hierarchical Clustering](#),” p. 445 in the “Clustering” chapter for a discussion of other available options of Hierarchical Clustering. The gradient columns can be deleted since they were used only to obtain the clusters. Your data table then contains only **Subject** and **Cluster** variables.

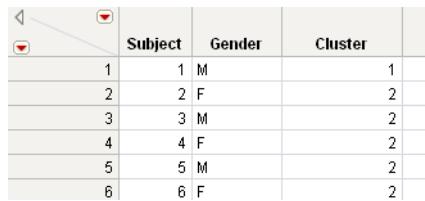
If you click on **Run Script** under the **Merge Data Back** menu, as shown in the partial gradient-by-subject table in Figure 18.31, the cluster information becomes a part of the Subject data table.

**Figure 18.31** Merge Clusters Back into Original Table

The screenshot shows the SPSS interface with the 'Merge Data Back' operation selected in the 'Hierarchical Cluster' dialog. The table on the right contains four rows of data with columns 'Subject' and 'Cluster'.

Subject	Cluster
1	1
2	2
3	2
4	2

The columns in the Subject data table are now Subject, Gender, and Cluster, as shown in Figure 18.32.

**Figure 18.32** Subject Data with Cluster Column

The screenshot shows the SPSS interface with the merged Subject data table. The table includes columns for Subject, Gender, and Cluster, with 6 rows of data.

Subject	Gender	Cluster
1	M	1
2	F	2
3	M	2
4	F	2
5	M	2
6	F	2

This table can then be used for further analysis. For example, select **Analyze > Fit Y by X**. Then, specify Gender as the **Y, Response** and Cluster as **X, Factor**. For the pizza example, this analysis is depicted in Figure 18.33.

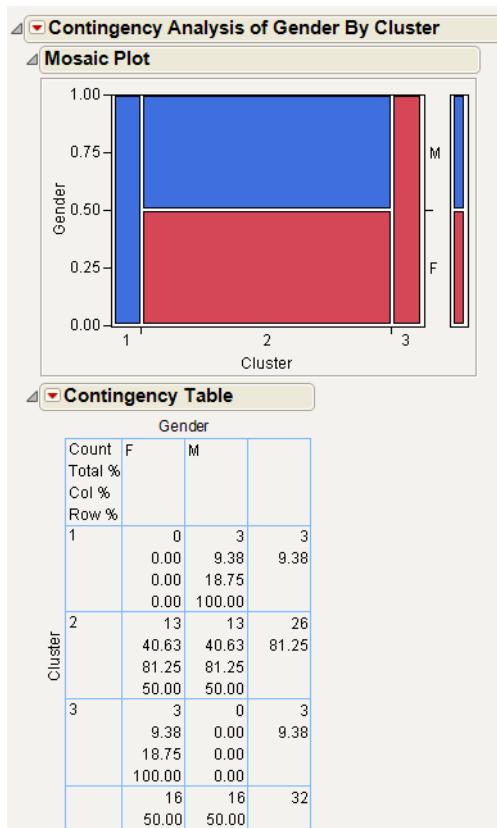
**Figure 18.33** Contingency Analysis of Gender by Cluster for Pizza Example

Figure 18.33 shows that Cluster 1 contains only males, Cluster 2 is half male and female, and Cluster 3 is all female. If desired, you could now refit and analyze the model with the addition of the **Cluster** variable.

---

## Special Data Rules

### Default choice set

If in every trial you can choose any of the response profiles, you can omit the **Profile ID Choices** selection under **Pick Role Variables** in the Response Data section of the Choice Dialog Box. Then the Choice Model platform assumes that all choice profiles are available on each run.

## Subject Data with Response Data

If you have subject data in the Response data table, just select this table as the **Select Data Table** under the Subject Data. In this case, a **Subject ID** column does not need to be specified. In fact, it is not used. It is generally assumed that the subject data repeats consistently in multiple runs for each subject.

## Logistic Regression

Ordinary logistic regression can be done with the Choice Modeling platform.

---

**Note:** The Fit Y by X and Fit Model platforms are more convenient to use than the Choice Modeling platform for logistic regression modeling. This section is used only to demonstrate that the Choice Modeling platform can be used for logistic regression, if desired.

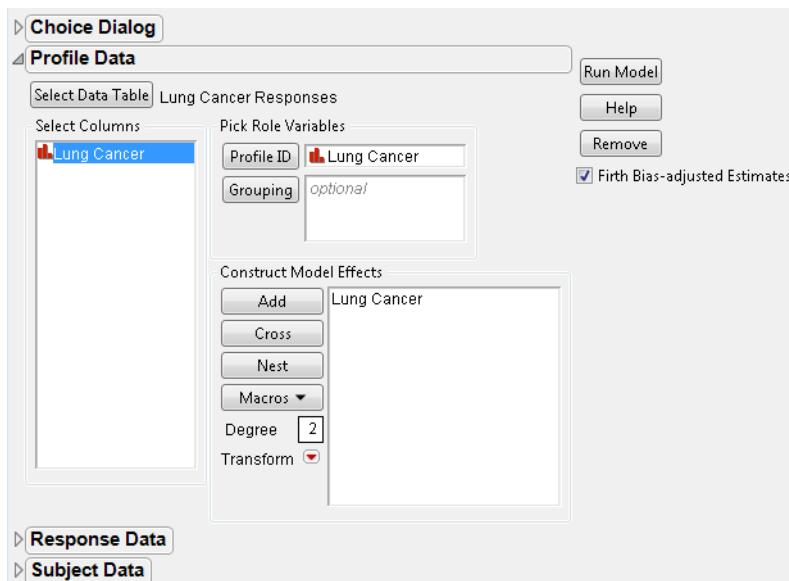
---

If your data are already in the choice-model format, you might want to use the steps given below for logistic regression analysis. However, three steps are needed:

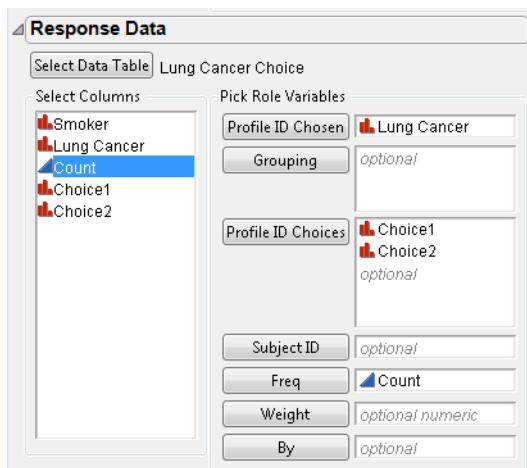
- Create a trivial Profile data table with a row for each response level.
- Put the explanatory variables into the Response data.
- Specify the Response data table, again, for the Subject data table.

An example of using the Choice Modeling platform for logistic regression follows:

1. Select **Analyze > Modeling > Choice > Select Data Table > Other > OK.**
2. Open the sample data set Lung Cancer Responses.jmp. Notice this data table has only one column (Lung Cancer) with two rows (Cancer and NoCancer).
3. Select Lung Cancer as the **Profile ID** and Add Lung Cancer as the model effect. The Profile Data dialog box is shown in Figure 18.34.

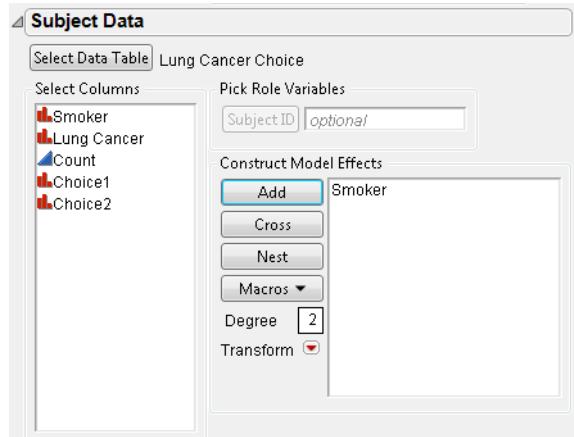
**Figure 18.34** Profile Data for Lung Cancer Example

4. Click on the disclosure button for **Response Data** > **Select Data Table** > Other > OK.
5. Open the sample data set Lung Cancer Choice.jmp.
6. Select Lung Cancer for **Profile ID Chosen**, Choice1 and Choice2 for **Profile ID Choices**, and Count for **Freq**. The Response Data launch dialog box is shown in Figure 18.35.

**Figure 18.35** Response Data for Lung Cancer Example

7. Click on the disclosure button for **Subject Data > Select Data Table > Lung Cancer Choice.jmp > OK.**
8. Add Smoker as the model effect. The Subject Data launch dialog box is shown in Figure 18.36.

**Figure 18.36** Subject Data for Lung Cancer Example



9. Uncheck **Firth Bias-adjusted Estimates** and **Run Model**.

Choice Modeling results are shown in Figure 18.37.

**Figure 18.37** Choice Modeling Logistic Regression Results for the Cancer Data

Term	Estimate	Std Error
Lung Cancer[Cancer]	-0.244049633	0.0649808386
Smoker[NonSmoker]*Lung Cancer[Cancer]	-0.272457870	0.0649808386

Source	ChiSquare	DF	Prob>ChiSq
Lung Cancer	15.803	1	<.0001*
Smoker*Lung Cancer	19.878	1	<.0001*

Compare these results with those of logistic regression under the Fit Model platform:

- Open Lung Cancer.jmp in the sample data directory.

- Select **Analyze > Fit Model**. Automatic specification of the columns is: Lung Cancer for **Y**, Count for **Freq**, and Smoker for **Add** under **Construct Model Effects**. The **Nominal Logistic** personality is automatically selected.
- Click on **Run Model**. The nominal logistic fit for the data is shown in Figure 18.38.

**Figure 18.38** Fit Model Nominal Logistic Regression Results for the Cancer Data

The screenshot displays the SAS Fit Model Nominal Logistic Regression Results for the Cancer Data. The results are organized into several sections:

- Iterations:** Converged in Gradient, 4 iterations. Freq: Count.
- Whole Model Test:**

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	9.93901	1	19.87802	<.0001*
Full	972.94369			
Reduced	982.88270			
- Measures:**

Measure	Training	Definition
Entropy RSquare	0.0101	$1 - \text{Loglike}(\text{model})/\text{Loglike}(0)$
Generalized R-Square	0.0186	$(1 - (L(0)/L(\text{model}))^{(2/n)})/(1 - L(0)^{(2/n)})$
Mean -Log p	0.6861	$\sum -\text{Log}(p[i])/n$
RMSE	0.4966	$\sqrt{\sum (y[i] - p[i])^2/n}$
Mean Abs Dev	0.4933	$\sum  y[i] - p[i] /n$
Misclassification Rate	0.4732	$\sum (p[i] \neq \text{pMax})/n$
N	1,418	n
- Parameter Estimates:**

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-0.4880993	0.1299617	14.11	0.0002*
Smoker[NonSmoker]	-0.5449157	0.1299617	17.58	<.0001*

For log odds of Cancer/NoCancer
- Covariance of Estimates:**
- Effect Likelihood Ratio Tests:**

L-R	Source	Nparm	DF	ChiSquare	Prob>ChiSq
	Smoker	1	1	19.878016	<.0001*

Notice that the likelihood ratio chi-square test for Smoker\*Lung Cancer in the Choice model matches the likelihood ratio chi-square test for Smoker in the Logistic model. The reports shown in Figure 18.37 and Figure 18.38 support the conclusion that smoking has a strong effect on developing lung cancer. See the “[Logistic Regression for Nominal and Ordinal Response](#)” chapter for details.

## Transforming Data

### Transforming Data to Two Analysis Tables

Although data are often in the Response/Profile/Subject form, the data are sometimes specified in another format that must be manipulated into the normalized form needed for choice analysis. For example, consider the data from Daganzo, found in *Daganzo Trip.jmp*. This data set contains the travel time for three transportation alternatives and the preferred transportation alternative for each subject. A partial listing of the data set is shown in Figure 18.39.

**Figure 18.39** Partial Daganzo Travel Time Table for Three Alternatives

	Subway	Bus	Car	Choice
1	16.481	16.196	23.89	2
2	15.123	11.373	14.182	2
3	19.469	8.822	20.819	2
4	18.847	15.649	21.28	2
5	12.578	10.871	18.335	2

Each choice number listed must first be converted to one of the travel mode names. This transformation is easily done by using the **Choose** function in the formula editor, as follows.

- Create a new column labeled **Choice Mode**. Specify the modeling type as **Nominal**. Right-click on the **Choice Mode** column and select **Formula**.
- Click on **Conditional** under the **Functions (grouped)** command, select **Choose**, and press the comma key twice to obtain additional arguments for the function.
- Click on **Choice** for the **Choose expression (expr)**, and double click on each clause entry box to enter “Subway”, “Bus”, and “Car” (with the quotation marks) as shown in Figure 18.40.

**Figure 18.40** Choose Function for Choice Mode Column of Daganzo Data

```
Choose[Choice]
  1  = "Subway"
  2  = "Bus"
  else= "Car"
```

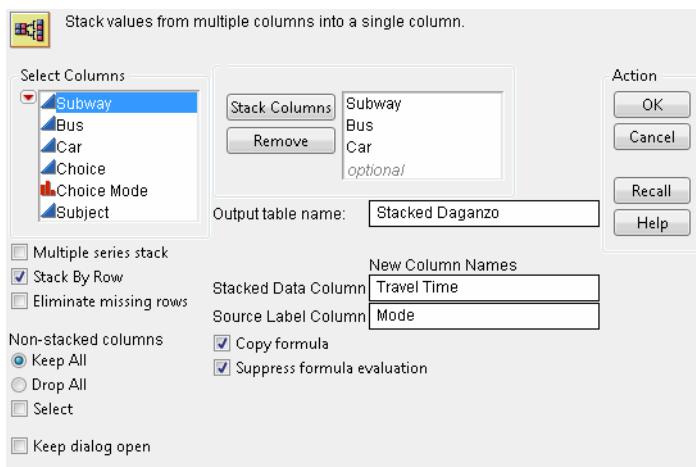
The choice response is now in the correct format. Since each row contains a choice made by each subject, another column containing a sequence of numbers should be created to identify the subjects. This can be done by creating a column with the **Subject** label. Then, enter a 1 in the first row of the column, a 2 in the second row of the column. Finally, highlight the first and second rows of the column, right click, and select **Fill > Continue sequence to end of table**. A partial listing of the modified table is shown in Figure 18.41.

**Figure 18.41** Daganzo Data with New Choice Mode and Subject Columns

	Subway	Bus	Car	Choice	Choice Mode	Subject
1	16.481	16.196	23.89	2	Bus	1
2	15.123	11.373	14.182	2	Bus	2
3	19.469	8.822	20.819	2	Bus	3
4	18.847	15.649	21.28	2	Bus	4
5	12.578	10.871	18.335	2	Bus	5
6	11.513	20.582	27.838	1	Subway	6

In order to construct the Profile data, each alternative needs to be expressed in a separate row.

- Use the Stack operation by clicking on **Tables > Stack** and filling in the entry fields as shown in Figure 18.42. Give this new data table a name, such as **Stacked Daganzo.jmp**, so that you can use this table for future analyses. Click **OK**. A partial view of the resulting table is shown in Figure 18.43.

**Figure 18.42** Stack Operation for Daganzo Data**Figure 18.43** Partial Stacked Daganzo Table

	Choice	Choice Mode	Subject	Mode	Travel Time
1	2	Bus		1	Subway
2	2	Bus		1	Bus
3	2	Bus		1	Car
4	2	Bus		2	Subway
5	2	Bus		2	Bus
6	2	Bus		2	Car
7	2	Bus		3	Subway

- Make a subset of the stacked data with just Subject, Mode, and Travel Time by selecting these columns and selecting **Tables > Subset > OK**. A partial data table is shown in Figure 18.44.

**Figure 18.44** Partial Subset Table of Daganzo Data

	Subject	Mode	Travel Time
1	1	Subway	16.481
2	1	Bus	16.196
3	1	Car	23.89
4	2	Subway	15.123
5	2	Bus	11.373
6	2	Car	14.182
7	3	Subway	19.469

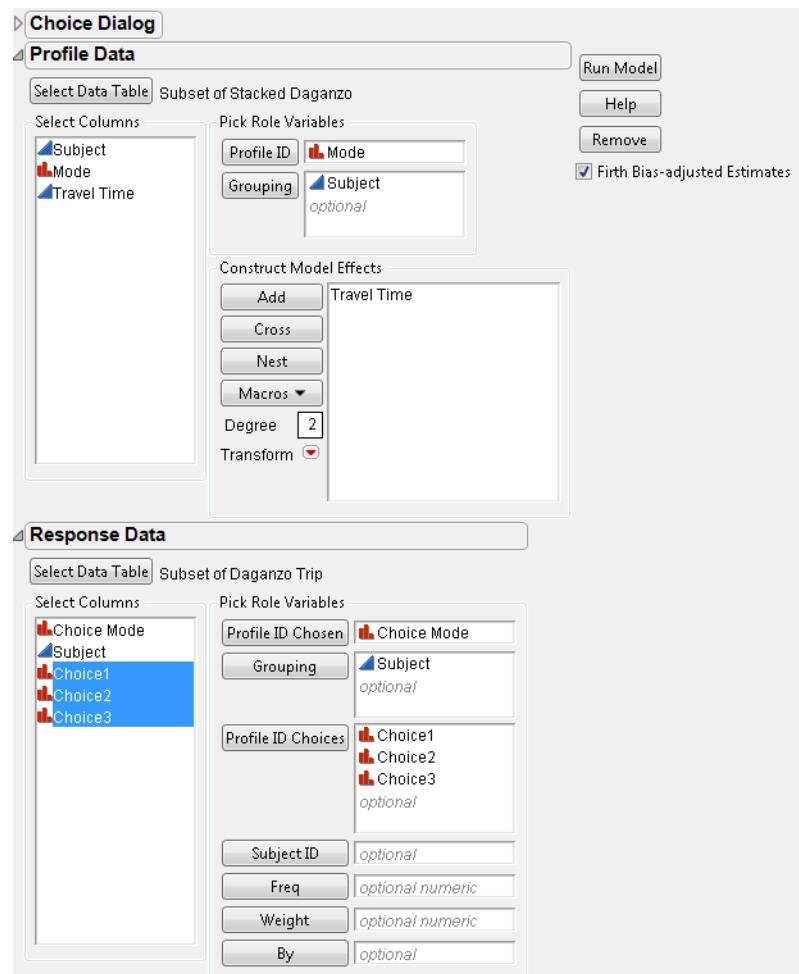
- Make another subset of the original data with just Subject and Choice Mode. Then, add three constant columns for the choice set: Choice1, Choice2, and Choice3, as shown in Figure 18.45.

**Figure 18.45** Partial Subset Table of Daganzo Data with Choice Set

	Choice Mode	Subject	Choice1	Choice2	Choice3
1	Bus	1	Bus	Subway	Car
2	Bus	2	Bus	Subway	Car
3	Bus	3	Bus	Subway	Car
4	Bus	4	Bus	Subway	Car
5	Bus	5	Bus	Subway	Car
6	Subway	6	Bus	Subway	Car
7	Subway	7	Bus	Subway	Car

- Specify the model, as shown in Figure 18.46.

**Figure 18.46** Choice Dialog Box for Subset of Daganzo Data



- Run the model. The resulting parameter estimate now expresses the utility coefficient for Travel Time and is shown in Figure 18.47.

**Figure 18.47** Parameter Estimate for Travel Time of Daganzo Data

Term	Estimate	Std Error
Travel Time	-0.341768586	0.0745222259

Source	ChiSquare	DF	Prob>ChiSq
Travel Time	48.371	1	<.0001*

The negative coefficient implies that increased travel time has a negative effect on consumer utility or satisfaction. The likelihood ratio test result indicates that the Choice model with the effect of Travel Time is significant.

## Transforming Data to One Analysis Table

Rather than creating two or three tables, it can be more practical to transform the data so that only one table is used. For the one-table format, the subject effect is added as above. A response indicator column is added instead of using three different columns for the choice sets (Choice1, Choice2, Choice3). The transformation steps for the one-table scenario include:

- Create or open Stacked Daganzo.jmp from the steps shown in *Transforming Data to Two Analysis Tables*.
- Add a new column labeled **Response** and right-click on the column. Select **Formula**.
- Select **Conditional > If** from the formula editor and select the column **Choice Mode** for the expression.
- Type “=” and select **Mode**.
- Type 1 for the **Then Clause** and 0 for the **Else Clause**. Click **OK**. The completed formula should look like Figure 18.48.

**Figure 18.48** Formula for Response Indicator for Stacked Daganzo Data

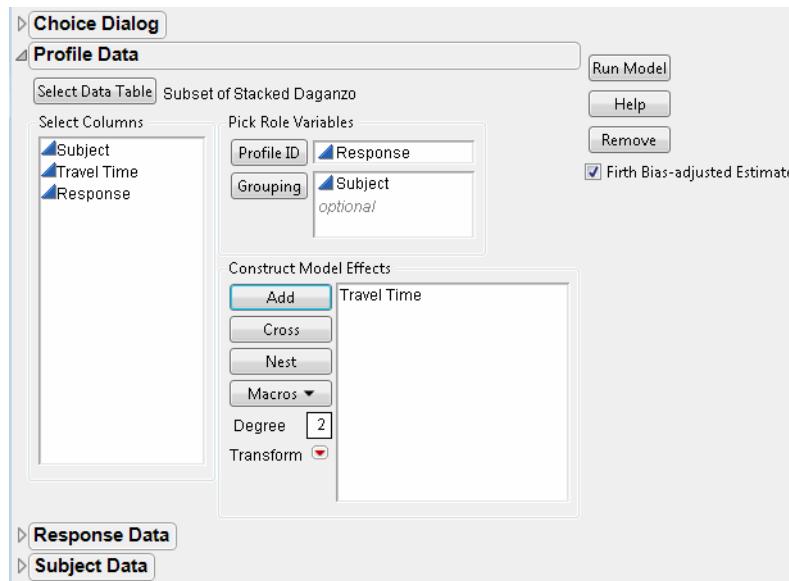
```
If [Choice Mode == Mode => 1]
[else => 0]
```

- Subset the data table by selecting **Subject**, **Travel Time**, and **Response** and then select **Tables > Subset > OK**. A partial listing of the new data table is shown in Figure 18.49.

**Figure 18.49** Partial Table of Stacked Daganzo Data Subset

	Subject	Travel Time	Response
1	1	16.481	0
2	1	16.196	1
3	1	23.89	0
4	2	15.123	0
5	2	11.373	1
6	2	14.182	0
7	3	19.469	0

- Select **Analyze > Modeling > Choice** to open the launch dialog box and specify the model as shown in Figure 18.50.

**Figure 18.50** Choice Dialog Box for Subset of Stacked Daganzo Data for One-Table Analysis

- Select **Run Model**. A pop-up dialog window asks if this is a one-table analysis with all the data in the Profile Table. Select **Yes** to obtain the parameter estimate expressing the utility Travel Time coefficient, shown in Figure 18.51.

**Figure 18.51** Parameter Estimate for Travel Time of Daganzo Data from One-Table Analysis

The screenshot shows the 'Choice Model' software interface with the following data:

**Parameter Estimates**

Term	Estimate	Std Error
Travel Time	-0.341768586	0.074522259

AICc: 68.766653  
BIC: 70.595342  
-2\*LogLikelihood: 66.683319  
-2\*Firth LogLikelihood: 61.490004

Converged in Gradient  
Firth Bias-adjusted Estimates

**Effect Likelihood Ratio Tests**

Source	L-R		
	ChiSquare	DF	Prob>ChiSq
Travel Time	48.371	1	<.0001*

Notice that the result is identical to that obtained for the two-table model, shown earlier in Figure 18.47.

This chapter illustrates the use of the Choice Modeling platform with simple examples. This platform can also be used for more complex models, such as those involving more complicated transformations and interaction terms.



# Chapter **19**

## Correlations and Multivariate Techniques

### The Multivariate Platform

---

The **Multivariate** platform specializes in exploring how many variables relate to each other. The platform begins by showing a standard correlation matrix. The Multivariate platform popup menu gives the additional correlations options and other techniques for looking at multiple variables such as

- a scatterplot matrix with normal density ellipses.
- inverse, partial, and pairwise correlations.
- a covariance matrix.
- nonparametric measures of association.
- simple statistics (such as mean and standard deviation).
- an outlier analysis that shows how far a row is from the center, respecting the correlation structure.
- a principal components facility with several options for rotated components and a table showing rotated factor patterns and communalities.
- imputation of missing data.

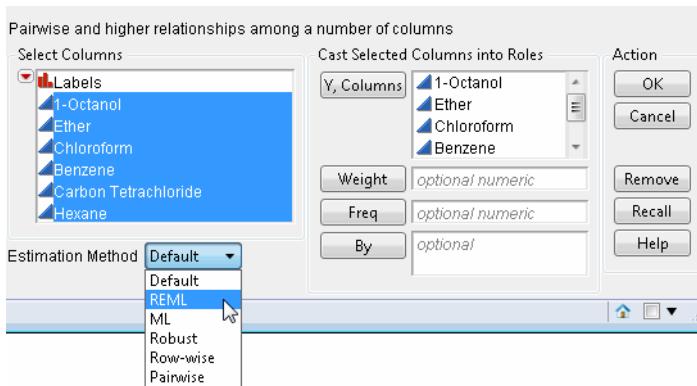
All plots and the current data table are linked. You can highlight points on any scatterplot in the scatterplot matrix, or the outlier distance plot. The points are highlighted on all other plots and are selected in the data table.

# Contents

Launch the Platform and Select Options . . . . .	421
Correlations Multivariate . . . . .	422
CI of Correlation . . . . .	423
Inverse Correlations and Partial Correlations . . . . .	423
Set a Level . . . . .	424
Scatterplot Matrix . . . . .	424
Covariance Matrix . . . . .	426
Pairwise Correlations . . . . .	426
Color Maps . . . . .	427
Simple Statistics . . . . .	428
Nonparametric Correlations . . . . .	428
Outlier Analysis . . . . .	429
Principal Components . . . . .	432
Item Reliability . . . . .	432
Parallel Coordinate Plot . . . . .	434
Ellipsoid 3D Plot . . . . .	434
Impute Missing Data . . . . .	435
Computations and Statistical Details . . . . .	436
Pearson Product-Moment Correlation . . . . .	436
Nonparametric Measures of Association . . . . .	436
Inverse Correlation Matrix . . . . .	438
Distance Measures . . . . .	438
Cronbach's $\alpha$ . . . . .	439

## Launch the Platform and Select Options

When you open a data table and select **Analyze > Multivariate Methods > Multivariate**, you can choose from several estimation methods for the correlations, including a **Default** option, in the Launch window. These Estimation Method options are available to provide flexibility with various data sets and to accommodate personal preferences, as shown below:



- The **Default** option enables the JMP platform to choose between the **REML** or **Pairwise** methods, depending upon the number of rows and columns in your dataset, and whether or not missing data is present. If your dataset has more than 10 columns or more than 5000 rows, **Pairwise** will be chosen, otherwise, **REML** will be used as the **Default** method.
- REML** (restricted maximum likelihood) estimates are less biased than the **ML** (maximum likelihood) estimation method. The REML method maximizes marginal likelihoods based upon error contrasts. The REML method is often used for estimating variances and covariances. The REML method in the Multivariate platform is the same as the REML estimation of mixed models for repeated measures data with an unstructured covariance matrix. See the documentation for SAS PROC MIXED about REML estimation of mixed models. **REML** uses all of your data, even if missing cells are present, and is most useful for smaller datasets. Because of the bias-correction factor, this method is slow if your dataset is large and there are many missing data values. If there are no missing cells in the data, then the **REML** estimate is equivalent to the sample covariance matrix.
- For large datasets with missing cells, **ML** is useful. The **ML** estimates are similar to the **REML** estimates, but the **ML** estimates are generated faster. Observations with missing values are not excluded. For small datasets, **REML** is preferred over **ML** because **REML**'s variance and covariance estimates are less biased.
- The **Robust** method is particularly useful if you suspect that your data have one or more outliers. **Robust** essentially ignores the outlying values by substantially down-weighting them. A sequence of iteratively reweighted fits of the data is done using the weight:

$$w_i = 1.0 \text{ if } Q < K \text{ and } w_i = K/Q \text{ otherwise,}$$

where  $K$  is a constant equal to the 0.9 quantile of a chi-square distribution with the degrees of freedom equal to the number of columns in the dataset, and

Launch the Platform and Select Options

$$Q = (y_i - \mu)^T (S^2)^{-1} (y_i - \mu)$$

where  $y_i$  = the response for the  $i^{th}$  observation,  $\mu$  = current estimate of the mean vector,  $S^2$  = current estimate of the covariance matrix, and  $T$  = the transpose matrix operation. The final step is a bias reduction of the variance matrix. The tradeoff of this method is that you can have higher variance estimates when the data do not have many outliers, but can have a much more precise estimate of the variances when the data do have outliers.

- **Row-wise** estimation does not use observations containing missing cells. The **Row-wise** option is useful if you want to check compatibility with earlier JMP versions (before JMP 8), or if you want to exclude any observations that have missing data. (**Row-wise** estimation was the only estimation method available before JMP 8.)
- **Pairwise** performs correlations for all rows for each pair of columns having nonmissing values.

**REML** and **Pairwise** are the methods used most frequently. You can also estimate missing values by using the estimated covariance matrix and then using the **Impute Missing Data** command, shown later in this chapter. See “[Impute Missing Data](#),” p. 435.

The multivariate report first shows the standard correlation matrix and scatterplot matrix. The red triangle menu for Multivariate lists additional correlation options and other techniques for looking at multiple variables. The following sections describe the tables and plots offered by the Multivariate platform.

In most of the following analysis options, a missing value in an observation does not cause the entire observation to be deleted. However, the **Pairwise Correlations** command excludes rows that are missing for either of the variables under consideration. The **Simple Statistics > Univariate** command calculates its statistics column-by-column, without regard to missing values in other columns.

Many of the following examples use the **Solubility.jmp** sample data table.

## Correlations Multivariate

The **Correlations Multivariate** option gives the Correlations table, which is a matrix of correlation coefficients that summarizes the strength of the linear relationships between each pair of response ( $Y$ ) variables. This correlation matrix is calculated by the method you select in the launch dialog.



A screenshot of the JMP Multivariate launch dialog. The 'Correlations' option is checked under the 'Multivariate' section. The resulting Correlations table is displayed below:

	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane
1-Octanol	1.0000	0.9343	0.5976	0.7197	0.6151	0.6046
Ether	0.9343	1.0000	0.5146	0.6489	0.5376	0.5494
Chloroform	0.5976	0.5146	1.0000	0.9411	0.9210	0.8719
Benzene	0.7197	0.6489	0.9411	1.0000	0.9573	0.9091
Carbon Tetrachloride	0.6151	0.5376	0.9210	0.9573	1.0000	0.9498
Hexane	0.6046	0.5494	0.8719	0.9091	0.9498	1.0000

## CI of Correlation

To obtain the two-tailed confidence intervals of the correlations, select **CI of Correlation** from the red-triangle of the Multivariate report. The default confidence coefficient is 95%, but you can change the value by using the **Set a Level** command. Confidence intervals for the correlations of the Solubility.jmp data are shown in Figure 19.1.

**Figure 19.1** Multivariate CI of Correlation

CI of Correlation				
Variable	by Variable	Correlation	Lower 95%	Upper 95%
Ether	1-Octanol	0.9343	0.8967	0.9585
Chloroform	1-Octanol	0.5976	0.4247	0.7284
Chloroform	Ether	0.5146	0.3212	0.6668
Benzene	1-Octanol	0.7197	0.5857	0.8154
Benzene	Ether	0.6489	0.4911	0.7655
Benzene	Chloroform	0.9411	0.9072	0.9628
Carbon Tetrachloride	1-Octanol	0.6151	0.4472	0.7412
Carbon Tetrachloride	Ether	0.5376	0.3495	0.6841
Carbon Tetrachloride	Chloroform	0.9210	0.8763	0.9500
Carbon Tetrachloride	Benzene	0.9573	0.9325	0.9732
Hexane	1-Octanol	0.6046	0.4337	0.7335
Hexane	Ether	0.5494	0.3640	0.6929
Hexane	Chloroform	0.8719	0.8023	0.9181
Hexane	Benzene	0.9091	0.8581	0.9423
Hexane	Carbon Tetrachloride	0.9498	0.9208	0.9684

## Inverse Correlations and Partial Correlations

The inverse correlation matrix (**Inverse Corr** table), shown at the top in the next figure, provides useful multivariate information. The diagonal elements of the matrix are a function of how closely the variable is a linear function of the other variables. In the inverse correlation, the diagonal is  $1/(1 - R^2)$  for the fit of that variable by all the other variables. If the multiple correlation is zero, the diagonal inverse element is 1. If the multiple correlation is 1, then the inverse element becomes infinite and is reported missing.

Launch the Platform and Select Options

 Multivariate

 Inverse Corr

	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane
1-Octanol	9.6893	-7.7565	0.1280	-2.8052	-0.2165	1.0470
Ether	-7.7565	8.5445	1.4972	-2.6483	2.9732	-1.7262
Chloroform	0.1280	1.4972	10.2337	-9.8884	-0.5151	-0.3445
Benzene	-2.8052	-2.6483	-9.8884	27.6656	-15.5648	1.4066
Carbon Tetrachloride	-0.2165	2.9732	-0.5151	-15.5648	25.2505	-10.8871
Hexane	1.0470	-1.7262	-0.3445	1.4066	-10.8871	10.6777

 Partial Corr

	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane
1-Octanol	0.8525	-0.0129	0.1713	0.0138	-0.1029	
Ether	0.8525	. .	-0.1601	0.1722	-0.2024	0.1807
Chloroform	-0.0129	-0.1601	. .	0.5877	0.0320	0.0330
Benzene	0.1713	0.1722	0.5877	. .	0.5889	-0.0818
Carbon Tetrachloride	0.0138	-0.2024	0.0320	0.5889	. .	0.6630
Hexane	-0.1029	0.1807	0.0330	-0.0818	0.6630	. .

partialed with respect to all other variables

The partial correlation table (**Partial Corr** table) shows the partial correlations of each pair of variables after adjusting for all the other variables. This is the negative of the inverse correlation matrix scaled to unit diagonal.

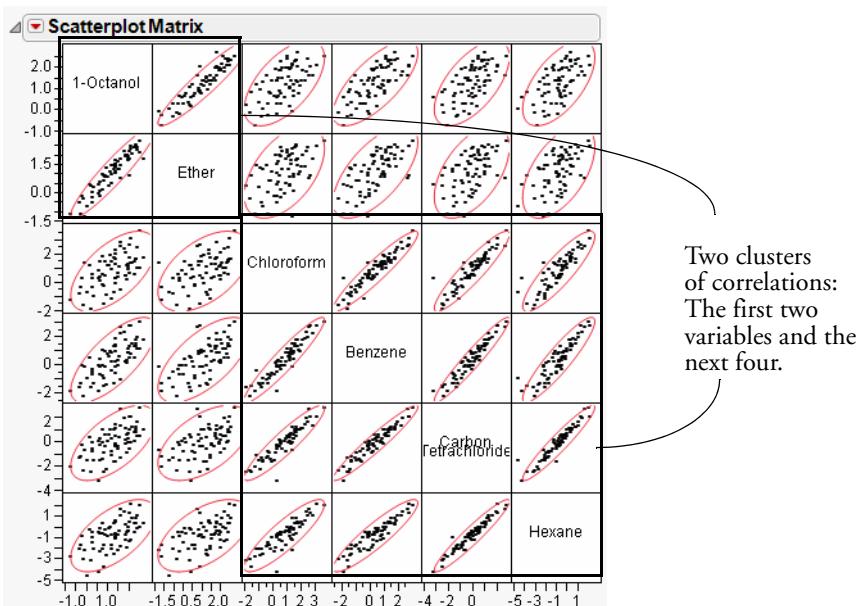
## Set $\alpha$ Level

The Multivariate platform allows you to interactively specify any alpha value for the correlation confidence intervals. Four alpha values are listed: **0.01**, **0.05**, **0.10**, and **0.50**. Selecting the **Other** command brings up a dialog where you can enter any other value. The default value of alpha is 0.05.

## Scatterplot Matrix

To help you visualize the correlations, a scatterplot for each pair of response variables displays in a matrix arrangement, as shown in Figure 19.2. The scatterplot matrix is shown by default. If the scatterplots are not showing, select **Scatterplot Matrix** from the platform popup menu. The cells of the scatterplot matrix are size-linked so that stretching a plot from any cell resizes all the scatterplot cells.

By default, a 95% bivariate normal density ellipse is imposed on each scatterplot. If the variables are bivariate normally distributed, this ellipse encloses approximately 95% of the points. The correlation of the variables is seen by the collapsing of the ellipse along the diagonal axis. If the ellipse is fairly round and is not diagonally oriented, the variables are uncorrelated.

**Figure 19.2** Example of a Scatterplot Matrix

The popup menu next on the **Scatterplot Matrix** title bar button lets you tailor the matrix with color and density ellipses and by setting the  $\alpha$ -level.

**Show Points** toggles the display of points on and off in the scatterplot matrix.

**Density Ellipses** toggles the display of the density ellipses on the scatterplots constructed by the  $\alpha$  level that you choose. By default they are 95% ellipses.

**Shaded Ellipses** provides color inside each ellipse.

**Show Correlations** shows the correlation of each histogram in the upper left corner of each scatterplot.

**Show Histogram** draws histograms in the diagonal of the scatterplot matrix. These histograms can be specified as **Horizontal** or **Vertical**. In addition, you can toggle the counts that label each bar with the **Show Counts** command.

**Ellipse  $\alpha$**  lets you select from a submenu of standard  $\alpha$ -levels or select the **Other** command and specifically set the  $\alpha$  level for the density ellipses.

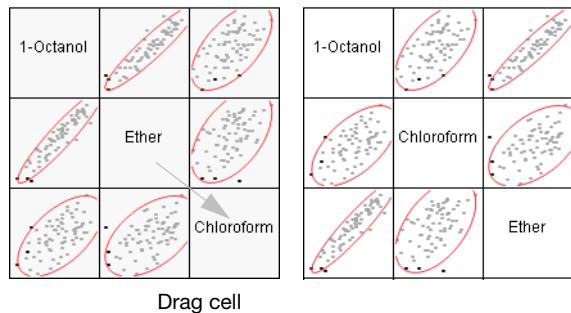
**Ellipses Transparency** lets you select a transparency value from a submenu or select the **Other** command to specify any desired transparency value. The default transparency value is 0.2.

**Ellipse Color** lets you select from a palette of colors to change the color of the ellipses.

You can reorder the scatterplot matrix columns by dragging a diagonal (label) cell to another position on the diagonal. For example, if you drag the cell of the column labeled Ether diagonally down one cell, the columns reorder as shown in Figure 19.3.

When you look for patterns in the whole scatterplot matrix with reordered columns, you clearly see the variables cluster into groups based on their correlations, as illustrated previously by the two groups showing in Figure 19.2.

**Figure 19.3 Reorder Scatterplot Matrix**



## Covariance Matrix

The **Covariance Matrix** command displays the covariance matrix for the analysis.

	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane
1-Octanol	0.67553	0.76870	0.57055	0.72776	0.64207	0.69393
Ether	0.76870	1.00217	0.59845	0.79922	0.68356	0.76796
Chloroform	0.57055	0.59845	1.34948	1.34496	1.35876	1.41442
Benzene	0.72776	0.79922	1.34496	1.51350	1.49577	1.56172
Carbon Tetrachloride	0.64207	0.68356	1.35876	1.49577	1.61295	1.68446
Hexane	0.69393	0.76796	1.41442	1.56172	1.68446	1.94995

## Pairwise Correlations

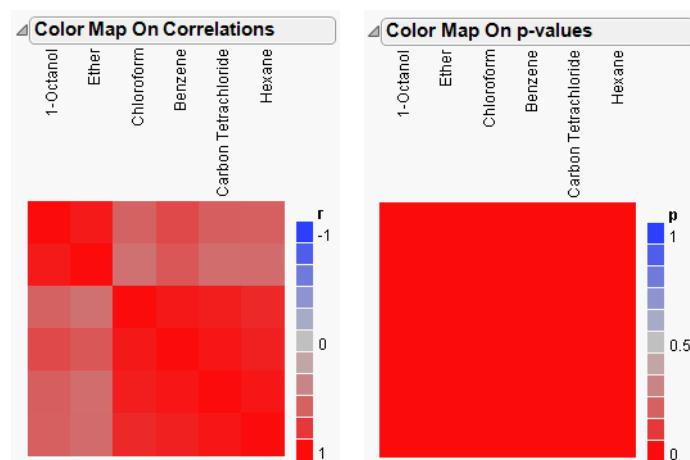
The Pairwise Correlations table is shown by selecting the **Pairwise Correlations** option from the red-triangle menu of the Multivariate title bar. It lists the Pearson product-moment correlations for each pair of Y variables. The correlations are calculated by the pairwise deletion method. The count values differ if any pair has a missing value for either variable. The Pairwise Correlations report also shows significance probabilities and compares the correlations with a bar chart. (See Figure 19.4.) All results like significance probabilities, confidence limits, and so on, under this outline, are based on the pairwise method.

**Figure 19.4** Pairwise Correlations Report

Pairwise Correlations															
Variable	by Variable	Correlation	Count	Lower 95%	Upper 95%	Signif Prob	.8	.6	.4	.2	0	.2	.4	.6	.8
Ether	1-Octanol	0.9343	72	0.8967	0.9585	<.0001*									
Chloroform	1-Octanol	0.5976	72	0.4247	0.7284	<.0001*									
Chloroform	Ether	0.5146	72	0.3212	0.6668	<.0001*									
Benzene	1-Octanol	0.7197	72	0.5857	0.8154	<.0001*									
Benzene	Ether	0.6489	72	0.4911	0.7655	<.0001*									
Benzene	Chloroform	0.9411	72	0.9072	0.9628	<.0001*									
Carbon Tetrachloride	1-Octanol	0.6151	72	0.4472	0.7412	<.0001*									
Carbon Tetrachloride	Ether	0.5376	72	0.3495	0.6841	<.0001*									
Carbon Tetrachloride	Chloroform	0.9210	72	0.8763	0.9500	<.0001*									
Carbon Tetrachloride	Benzene	0.9573	72	0.9325	0.9732	<.0001*									
Hexane	1-Octanol	0.6046	72	0.4337	0.7335	<.0001*									
Hexane	Ether	0.5494	72	0.3640	0.6929	<.0001*									
Hexane	Chloroform	0.8719	72	0.8023	0.9181	<.0001*									
Hexane	Benzene	0.9091	72	0.8581	0.9423	<.0001*									
Hexane	Carbon Tetrachloride	0.9498	72	0.9208	0.9684	<.0001*									

## Color Maps

The Color Map submenu gives you three choices: **Color Map On Correlations**, **Color Map On p-values**, and **Cluster the Correlations**. The first option produces the cell plot on the left of Figure 19.5, showing the correlations among variables on a scale from red (+1) to blue (-1). **Color Map on p-values** shows the significance of the correlations on a scale from  $p = 0$  (red) to  $p = 1$  (blue). **Cluster the Correlations** groups variables that have similar correlations together, which (in the case of Figure 19.5 looks exactly like the color map on correlations.)

**Figure 19.5** Cell Plots

## Simple Statistics

The Simple Statistics submenu allows you to display simple statistics (mean, standard deviation, and so on) for each column. The univariate and multivariate simple statistics can differ when there are missing values present, or when the Robust method is used.

**Univariate Simple Statistics** are calculated on each column, regardless of values in other columns.

These values match the ones that would be produced using the Distribution platform.

Univariate Simple Statistics							
Column	N	DF	Mean	Std Dev	Sum	Minimum	Maximum
1-Octanol	72	71.00	1.2665	0.8219	91.1900	-0.7700	2.7000
Ether	72	71.00	1.1010	1.0011	79.2700	-1.3000	2.7000
Chloroform	72	71.00	0.8065	1.1617	58.0700	-1.9200	3.6700
Benzene	72	71.00	0.3949	1.2302	28.4300	-2.2600	3.0300
Carbon Tetrachloride	72	71.00	-0.0746	1.2700	-5.3700	-3.3000	2.7400
Hexane	72	71.00	-0.7817	1.3964	-56.280	-4.6000	2.1000

Note: Statistics were calculated for each column independently without regard for missing values in other columns.

**Multivariate Simple Statistics** These are the statistics corresponding to the estimation method selected in the launch dialog. If REML, ML, or Robust methods are selected, the mean vector and covariance matrix are estimated by that selected method. If the Row-wise method is selected, all rows with at least one missing value are excluded from the calculation of means and variances. Finally, if the Pairwise method is selected, the mean and variance are calculated for each column.

Multivariate Simple Statistics							
Column	N	DF	Mean	Std Dev	Sum	Minimum	Maximum
1-Octanol	72	71.00	1.2665	0.8219	91.1900	-0.7700	2.7000
Ether	72	71.00	1.1010	1.0011	79.2700	-1.3000	2.7000
Chloroform	72	71.00	0.8065	1.1617	58.0700	-1.9200	3.6700
Benzene	72	71.00	0.3949	1.2302	28.4300	-2.2600	3.0300
Carbon Tetrachloride	72	71.00	-0.0746	1.2700	-5.3700	-3.3000	2.7400
Hexane	72	71.00	-0.7817	1.3964	-56.280	-4.6000	2.1000

## Nonparametric Correlations

When you select **Nonparametric Correlations** from the platform popup menu, the Nonparametric Measures of Association table is shown. The Nonparametric submenu offers these three nonparametric measures:

**Spearman's Rho** is a correlation coefficient computed on the ranks of the data values instead of on the values themselves.

**Kendall's Tau** is based on the number of concordant and discordant pairs of observations. A pair is *concordant* if the observation with the larger value of X also has the larger value of Y. A pair is *discordant* if the observation with the larger value of X has the smaller value of Y. There is a correction for tied pairs (pairs of observations that have equal values of X or equal values of Y).

**Hoeffding's D** is a statistical scale that ranges from -0.5 to 1, with large positive values indicating dependence. The statistic approximates a weighted sum over observations of chi-square statistics for

two-by-two classification tables. The two-by-two tables are made by setting each data value as the threshold. This statistic detects more general departures from independence.

The Nonparametric Measures of Association report also shows significance probabilities for all measures and compares them with a bar chart similar to the one in Figure 19.4.

See “[Computations and Statistical Details](#),” p. 436, for computational information.

---

**Note:** The nonparametric correlations are always calculated by the Pairwise method, even if other methods have been chosen in the launch dialog.

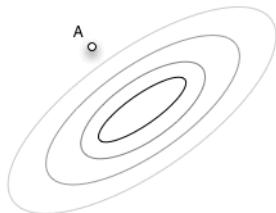
---

## Outlier Analysis

The **Outlier Analysis** submenu toggles the display of three plots:

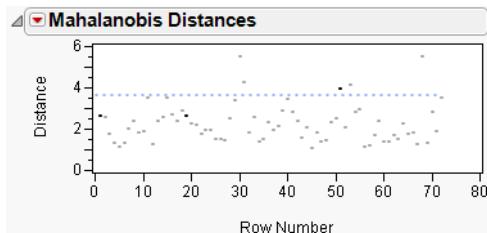
- Mahalanobis Distance
- Jackknife Distances
- $T^2$  Statistic.

These all measure distance in the multivariate sense (with respect to the correlation structure), as in the normal ellipse contours shown here. Point A is an outlier because it is outside the correlation structure rather than because it is an outlier in any of the coordinate directions.

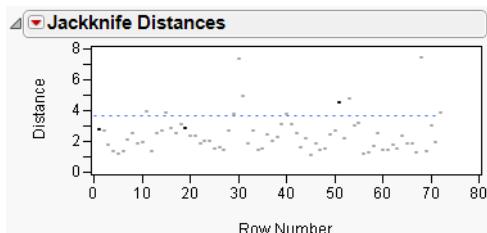


---

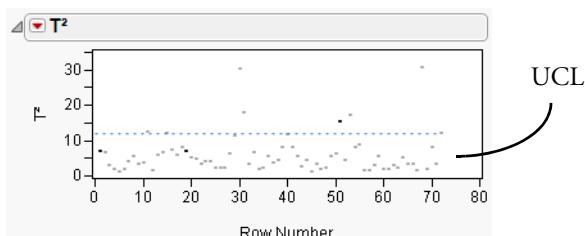
Figure 19.6 shows the *Mahalanobis distance* of each point from the multivariate mean (centroid). The standard Mahalanobis distance depends on estimates of the mean, standard deviation, and correlation for the data. The distance is plotted for each observation number. Extreme multivariate outliers can be identified by highlighting the points with the largest distance values. See “[Computations and Statistical Details](#),” p. 436, for more information.

**Figure 19.6** Mahalanobis Outlier Distance Plot

JMP can also calculate an alternate distance using a *jackknife* technique. The distance for each observation is calculated with estimates of the mean, standard deviation, and correlation matrix that do not include the observation itself. The jackknifed distances are useful when there is an outlier. In this case, the Mahalanobis distance is distorted and tends to disguise the outlier or make other points look more outlying than they are.

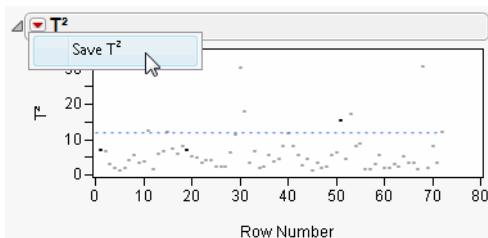
**Figure 19.7** Jackknifed Distances Plot

A third option is the  $T^2$  statistic. This is the square of the Mahalanobis distance, and is preferred for multivariate control charts. The plot includes the value of the calculated  $T^2$  statistic, as well as its upper control limit. Values that fall outside this limit may be an outlier.

**Figure 19.8**  $T^2$  Statistic Plot

## Saving Distances and Values

Each of the three outlier methods allows you to save distances back to the data table. To save the distances, use the popup menu beside the name of the distance you want to save. For example, the following figure saves values for the  $T^2$  statistic.



There is no formula saved with the distance column. This means the distance is not recomputed if you modify the data table. If you add or delete columns or change values in the data table, you should select **Analyze > Multivariate Methods > Multivariate** again to compute new distances.

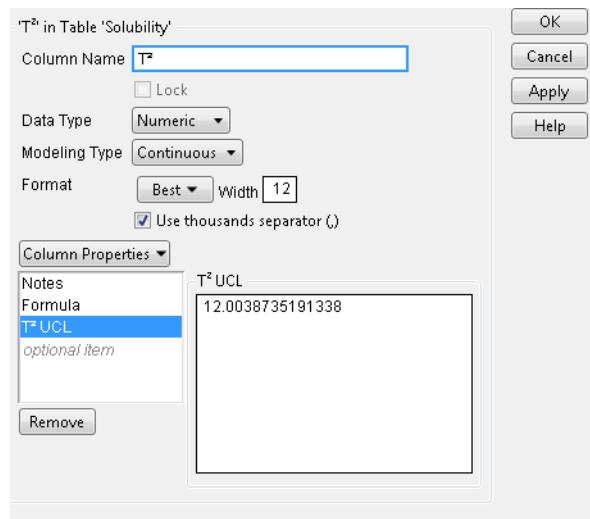
**Figure 19.9** Distances Saved in Data Table

	Labels	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane	$T^2$
1	METHANOL	-0.770	-1.150	-1.260	-1.890	-2.100	-2.800	6.782222533
2	ETHANOL	-0.310	-0.570	-0.850	-1.620	-1.400	-2.100	6.4301318549
3	PROPANOL	0.250	-0.020	-0.400	-0.700	-0.820	-1.520	2.9818194418
4	BUTANOL	0.880	0.890	0.450	-0.120	-0.400	-0.700	1.7123207338
5	PENTANOL	1.560	1.200	1.050	0.620	0.400	-0.400	1.2717763323
6	HEXANOL	2.030	1.800	1.690	1.300	0.990	0.460	1.712655569

---

In addition to saving the distance values for each row, a column property is created that holds the distance value (for Mahalanobis Distance and Jackknife distance) or a list containing the UCL of the  $T^2$  statistic.

## Launch the Platform and Select Options



## Principal Components

*Principal components* is a technique to take linear combinations of the original variables such that the first principal component has maximum variation, the second principal component has the next most variation subject to being orthogonal to the first, and so on. See the chapter “[Principal Components](#),” p. 461 for details on principal components.

## Item Reliability

Item reliability indicates how consistently a set of instruments measures an overall response. Cronbach’s  $\alpha$  (Cronbach 1951) is one measure of reliability. Two primary applications for Cronbach’s  $\alpha$  are industrial instrument reliability and questionnaire analysis.

Cronbach’s  $\alpha$  is based on the average correlation of items in a measurement scale. It is equivalent to computing the average of all split-half correlations in the data set. The **Standardized  $\alpha$**  can be requested if the items have variances that vary widely.

Cronbach’s  $\alpha$  is not related to a significance level  $\alpha$ . Also, item reliability is unrelated to survival time reliability analysis.

As an example, consider the Danger.jmp data in the Sample Data folder. This table lists 30 items having some level of inherent danger. Three groups of people (students, nonstudents, and experts) ranked the items according to perceived level of danger. Note that Nuclear power is rated as very dangerous (1) by both students and nonstudents but was ranked low (20) by experts. On the other hand, motorcycles are ranked fifth or sixth by the three judging groups.

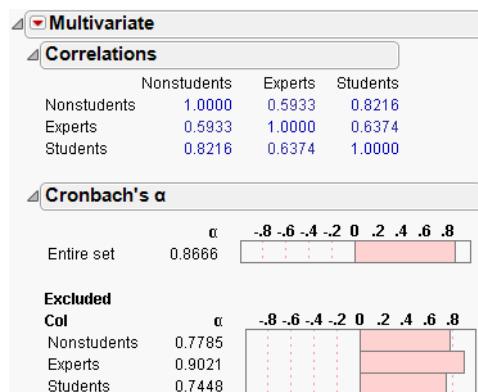
Danger	Activity	Nonstudents	Experts
Notes Rankings of the risks	1 Nuclear power	1	20
	2 Motor vehicles	2	1
	3 Handguns	3	4
	4 Smoking	4	2
	5 Motorcycles	5	6
	6 Alcoholic beverages	6	3
	7 Private aviation	7	12
	8 Police work	8	17
	9 Pesticides	9	8
	10 Surgery	10	5
	11 Fire fighting	11	18
	12 Large construction	12	13
	13 Hunting	13	23

You can use Cronbach's  $\alpha$  to evaluate the agreement in the perceived way the groups ranked the items. The results at the bottom in Figure 19.10 show an overall  $\alpha$  of 0.8666, which indicates a high correlation of the ranked values among the three groups. Further, when you remove the experts from the analysis, the Nonstudents and Students ranked the dangers nearly the same, with Cronbach's  $\alpha$  scores of 0.7785 and 0.7448, respectively. Nunnally (1979) suggests a Cronbach's  $\alpha$  of 0.7 as a rule-of-thumb acceptable level of agreement.

To look at the influence of an individual item, JMP excludes it from the computations and shows the effect of the Cronbach's  $\alpha$  value. If  $\alpha$  increases when you exclude a variable (item), that variable is not highly correlated with the other variables. If the  $\alpha$  decreases, you can conclude that the variable is correlated with the other items in the scale. Computations for Cronbach's  $\alpha$  are given in the next section, "[Computations and Statistical Details](#)," p. 436.

Note that in this kind of example, where the values are the same set of ranks for each group, standardizing the data has no effect.

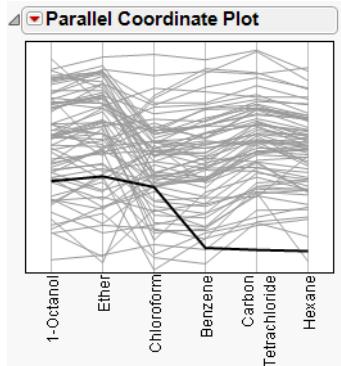
**Figure 19.10** Cronbach's  $\alpha$  Report



## Parallel Coordinate Plot

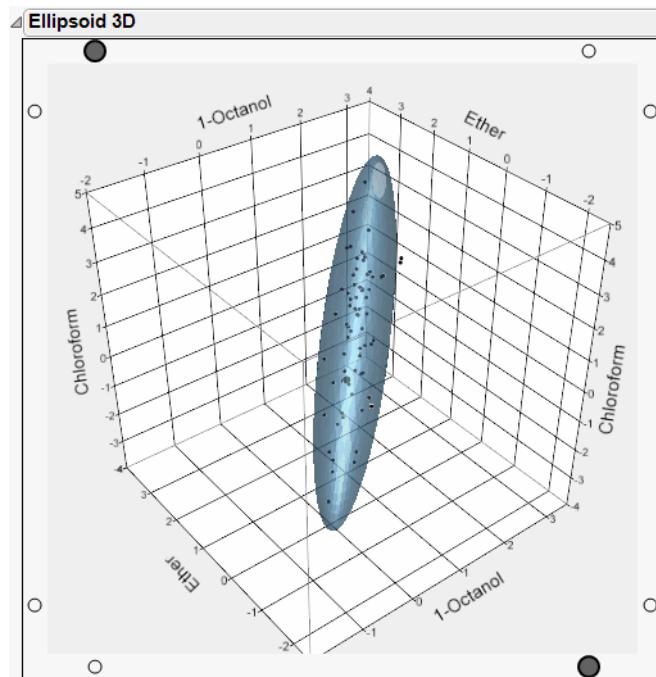
The **Parallel Coordinate Plot** option shows or hides a parallel coordinate plot of the variables.

**Figure 19.11** Parallel Coordinate Plot



## Ellipsoid 3D Plot

The Ellipsoid 3D Plot toggles a 95% confidence ellipsoid around three chosen variables. When the command is first invoked, a dialog asks which three variables to include in the plot.

**Figure 19.12** Ellipsoid 3D Plot

## Impute Missing Data

The platform has a feature to impute missing data. For example, delete several of the values within the first ten rows of Solubility.jmp, as shown in Figure 19.13.

**Figure 19.13** Partial Solubility.jmp with missing data

	Labels	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane
1	METHANOL	*	-1.150	-1.260	-1.890	-2.100	-2.800
2	ETHANOL	-0.310	*	-0.850	*	-1.400	*
3	PROPANOL	0.250	-0.020	-0.400	-0.700	*	-1.520

Click on **Analyze > Multivariate Methods > Multivariate** and select all of the continuous columns as **Y, Columns**. Click on the Multivariate platform drop-down menu and select **Impute Missing Data**. All missing data values will be replaced by estimated values, as shown in Figure 19.14.

**Figure 19.14** Imputed Data for Solubility.jmp

	1-Octanol	Ether	Chloroform	Benzene	Carbon Tetrachloride	Hexane	T <sup>2</sup>
1	-0.507168171	-1.15	-1.26	-1.89	-2.1	-2.8	8.3518967351
2	-0.31	-0.707244489	-0.85	-1.316461137	-1.4	-2.215688986	6.7032132235
3	0.25	-0.02	-0.4	-0.7	-0.951717531	-1.52	5.4029758288

Imputed values are expectations conditional on the non-missing values for each row. The mean and covariance matrix, which is estimated by the method chosen in the launch dialog, is used for the imputation calculation.

## Computations and Statistical Details

### Pearson Product-Moment Correlation

The Pearson product-moment correlation coefficient measures the strength of the linear relationship between two variables. For response variables  $X$  and  $Y$ , it is denoted as  $r$  and computed as

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2} \sqrt{\sum (y - \bar{y})^2}}.$$

If there is an exact linear relationship between two variables, the correlation is 1 or -1, depending on whether the variables are positively or negatively related. If there is no linear relationship, the correlation tends toward zero.

### Nonparametric Measures of Association

For the Spearman, Kendall, or Hoeffding correlations, the data are first ranked. Computations are then performed on the ranks of the data values. Average ranks are used in case of ties.

#### Spearman's $\rho$ (rho) Coefficients

Spearman's  $\rho$  correlation coefficient is computed on the ranks of the data using the formula for the Pearson's correlation previously described.

#### Kendall's $\tau_b$ Coefficients

Kendall's  $\tau_b$  coefficients are based on the number of concordant and discordant pairs. A pair of rows for two variables is *concordant* if they agree in which variable is greater. Otherwise they are discordant, or tied.

The formula

$$\tau_b = \frac{\sum_{i < j} \operatorname{sgn}(x_i - x_j) \operatorname{sgn}(y_i - y_j)}{\sqrt{(T_0 - T_1)(T_0 - T_2)}}$$

computes Kendall's  $\tau_b$  where

$$T_0 = (n(n-1))/2,$$

$$T_1 = \sum((t_i)(t_i-1))/2, \text{ and}$$

$$T_2 = \sum((u_i)(u_i-1))/2,$$

Note that  $\operatorname{sgn}(z)$  is equal to 1 if  $z > 0$ , 0 if  $z = 0$ , and -1 if  $z < 0$ .

The  $t_i$  (the  $u_i$ ) are the number of tied  $x$  (respectively  $y$ ) values in the  $i$ th group of tied  $x$  (respectively  $y$ ) values,  $n$  is the number of observations, and Kendall's  $\tau_b$  ranges from -1 to 1. If a weight variable is specified, it is ignored.

Computations proceed in the following way:

- Observations are ranked in order according to the value of the first variable.
- The observations are then re-ranked according to the values of the second variable.
- The number of interchanges of the first variable is used to compute Kendall's  $\tau_b$ .

### **Hoeffding's D Statistic**

The formula for Hoeffding's  $D$  (1948) is

$$D = 30 \left( \frac{(n-2)(n-3)D_1 + D_2 - 2(n-2)D_3}{n(n-1)(n-2)(n-3)(n-4)} \right)$$

where:

$$D_1 = S_i(Q_i-1)(Q_i-2)$$

$$D_2 = S_i(R_i-1)(S_i-1)(S_i-2)$$

$$D_3 = (R_i-1)(S_i-2)(Q_i-1)$$

The  $R_i$  and  $S_i$  are ranks of the  $x$  and  $y$  values, and the  $Q_i$  (sometimes called bivariate ranks) are one plus the number of points that have both  $x$  and  $y$  values less than the  $i$ th points. A point that is tied on its  $x$  value or  $y$  value, but not on both, contributes 1/2 to  $Q_i$  if the other value is less than the corresponding value for the  $i$ th point. A point tied on both  $x$  and  $y$  contributes 1/4 to  $Q_i$ .

When there are no ties among observations, the  $D$  statistic has values between -0.5 and 1, with 1 indicating complete dependence. If a weight variable is specified, it is ignored.

## Inverse Correlation Matrix

The inverse correlation matrix provides useful multivariate information. The diagonal elements of the inverse correlation matrix, sometimes called the variance inflation factors (VIF), are a function of how closely the variable is a linear function of the other variables. Specifically, if the correlation matrix is denoted  $R$  and the inverse correlation matrix is denoted  $R^{-1}$ , the diagonal element is denoted  $r^{ii}$  and is computed as

$$r^{ii} = \text{VIF}_i = \frac{1}{1 - R_i^2}$$

where  $R_i^2$  is the coefficient of variation from the model regressing the  $i^{\text{th}}$  explanatory variable on the other explanatory variables. Thus, a large  $r^{ii}$  indicates that the  $i^{\text{th}}$  variable is highly correlated with any number of the other variables.

## Distance Measures

The Outlier Distance plot shows the Mahalanobis distance of each point from the multivariate mean (centroid). The Mahalanobis distance takes into account the correlation structure of the data as well as the individual scales. For each value, the distance is denoted  $d_i$  and is computed as

$$d_i = \sqrt{(Y_i - \bar{Y})' S^{-1} (Y_i - \bar{Y})}$$

where:

$Y_i$  is the data for the  $i^{\text{th}}$  row

$\bar{Y}$  is the row of means

$S$  is the estimated covariance matrix for the data

The reference line drawn on the Mahalanobis Distance plot is computed as  $\sqrt{F \times \text{nvrs}}$  where  $\text{nvrs}$  is the number of variables and the computation for  $F$  in formula editor notation is:

`F Quantile(0.95, nvrs, n-nvrs-1, centered at 0)`

If a variable is an exact linear combination of other variables, then the correlation matrix is singular and the row and the column for that variable are zeroed out. The generalized inverse that results is still valid for forming the distances.

The  $T^2$  distance is just the square of the Mahalanobis distance, so  $T_i^2 = d_i^2$ . The upper control limit on the  $T^2$  is

$$UCL = \frac{(n-1)^2}{n} \beta_{\left[ \frac{n-p}{2}, \frac{n-p-1}{2} \right]}$$

where

$n$  = number of observations

$p$  = number of variables (columns)

$\alpha = \alpha^{\text{th}}$  quantile

$\beta =$  beta distribution

Multivariate distances are useful for spotting outliers in many dimensions. However, if the variables are highly correlated in a multivariate sense, then a point can be seen as an outlier in multivariate space without looking unusual along any subset of dimensions. Said another way, when data are correlated, it is possible for a point to be unremarkable when seen along one or two axes but still be an outlier by violating the correlation.

---

**Statistical Warning:** This outlier distance is not particularly robust in the sense that outlying points themselves can distort the estimate of the covariances and means in such a way that outliers are disguised. You might want to use the alternate distance command so that distances are computed with a jackknife method. The alternate distance for each observation uses estimates of the mean, standard deviation, and correlation matrix that do not include the observation itself.

---

## Cronbach's $\alpha$

Cronbach's  $\alpha$  is defined as

$$\alpha = \frac{k \sum \frac{c}{v}}{1 + (k - 1) \sum \frac{c}{v}}$$

where

$k$  = the number of items in the scale

$c$  = the average covariance between items

$v$  = the average variance between items.

If the items are standardized to have a constant variance, the formula becomes

$$\alpha = \frac{k(r)}{1 + (k - 1)r} \text{ where}$$

$r$  = the average correlation between items.

The larger the overall  $\alpha$  coefficient, the more confident you can feel that your items contribute to a reliable scale or test. The coefficient can approach 1.0 if you have many highly correlated items.



# Chapter **20**

## **Clustering** The Cluster Platform

---

Clustering is the technique of grouping rows together that share similar values across a number of variables. It is a wonderful exploratory technique to help you understand the clumping structure of your data. JMP provides three different clustering methods:

- Hierarchical clustering is appropriate for small tables, up to several thousand rows. It combines rows in an hierarchical sequence portrayed as a tree. In JMP, the tree, also called a dendrogram, is a dynamic, responding graph. You can choose the number of clusters you like after the tree is built.
- $K$ -means clustering is appropriate for larger tables, up to hundreds of thousands of rows. It makes a fairly good guess at cluster seed points. It then starts an iteration of alternately assigning points to clusters and recalculating cluster centers. You have to specify the number of clusters before you start the process.
- Normal mixtures are appropriate when data is assumed to come from a mixture of multivariate normal distributions that overlap. Maximum likelihood is used to estimate the mixture proportions and the means, standard deviations, and correlations jointly. This approach is particularly good at estimating the total counts in each group. However each point, rather than being classified into one group, is assigned a probability of being in each group. The EM algorithm is used to obtain estimates.

After the clustering process is complete, you can save the cluster assignments to the data table or use them to set colors and markers for the rows.

# Contents

Introduction to Clustering Methods .....	443
The Cluster Launch Dialog.....	444
Hierarchical Clustering .....	445
Hierarchical Cluster Options.....	447
Technical Details for Hierarchical Clustering.....	448
K-Means Clustering .....	450
K-Means Control Panel.....	451
K-Means Report .....	453
Normal Mixtures.....	454
Robust Normal Mixtures.....	456
Platform Options .....	458
Details of the Estimation Process.....	458
Self Organizing Maps .....	458

---

## Introduction to Clustering Methods

Clustering is a multivariate technique of grouping rows together that share similar values. It can use any number of variables. The variables must be numeric variables for which numerical differences makes sense. The common situation is that data are not scattered evenly through  $n$ -dimensional space, but rather they form clumps, locally dense areas, modes, or clusters. The identification of these clusters goes a long way towards characterizing the distribution of values.

JMP provides two approaches to clustering:

- *hierarchical clustering* for small tables, up to several thousand rows
- *k-means* and *normal mixtures* clustering for large tables, up to hundreds of thousands of rows.

Hierarchical clustering is also called *agglomerative clustering* because it is a combining process. The method starts with each point (row) as its own cluster. At each step the clustering process calculates the distance between each cluster, and combines the two clusters that are closest together. This combining continues until all the points are in one final cluster. The user then chooses the number of clusters that seems right and cuts the clustering tree at that point. The combining record is portrayed as a tree, called a *dendrogram*, with the single points as leaves, the final single cluster of all points as the trunk, and the intermediate cluster combinations as branches. Since the process starts with  $n(n + 1)/2$  distances for  $n$  points, this method becomes too expensive in memory and time when  $n$  is large.

Hierarchical clustering also supports character columns. If the column is ordinal, then the data value used for clustering is just the index of the ordered category, treated as if it were continuous data. If the column is nominal, then the categories must match to contribute a distance of zero. They contribute a distance of 1 otherwise.

JMP offers five rules for defining distances between clusters: Average, Centroid, Ward, Single, and Complete. Each rule can generate a different sequence of clusters.

*K*-means clustering is an iterative follow-the-leader strategy. First, the user must specify the number of clusters,  $k$ . Then a search algorithm goes out and finds  $k$  points in the data, called *seeds*, that are not close to each other. Each seed is then treated as a cluster center. The routine goes through the points (rows) and assigns each point to the cluster it is closest to. For each cluster, a new cluster center is formed as the means (centroid) of the points currently in the cluster. This process continues as an alternation between assigning points to clusters and recalculating cluster centers until the clusters become stable.

Normal mixtures clustering, like *k*-means clustering, begins with a user-defined number of clusters and then selects distance seeds. JMP uses the cluster centers chosen by *k*-means as seeds. However, each point, rather than being classified into one group, is assigned a probability of being in each group.

SOMs are a variation on *k*-means where the cluster centers are laid out on a grid. Clusters and points close together on the grid are meant to be close together in the multivariate space. See “[Self Organizing Maps](#),” p. 458.

*K*-means, normal mixtures, and SOM clustering are doubly-iterative processes. The clustering process iterates between two steps in a particular implementation of the EM algorithm:

- The *expectation step* of mixture clustering assigns each observation a probability of belonging to each cluster.

- For each cluster, a new center is formed using every observation with its probability of membership as a weight. This is the *maximization step*.

This process continues alternating between the expectation and maximization steps until the clusters become stable.

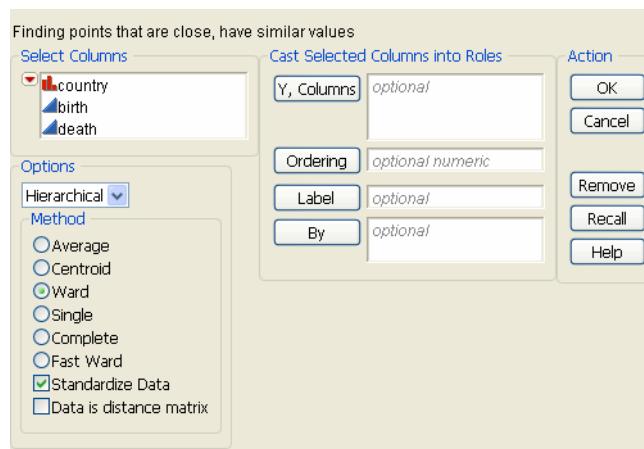
**Note:** For  $k$ -means clustering, you can choose a variable whose values form preset fixed centers for clusters, instead of using the default random seeds for clusters.

## The Cluster Launch Dialog

When you choose **Cluster** from the **Analyze > Multivariate Methods** submenu, the Hierarchical Cluster Launch dialog shown in Figure 20.1 appears. The data table used is Birth Death Subset.jmp.

Choose **KMeans** from the Options menu to see the KMeans launch dialog. See “[K-Means Clustering](#),” p. 450 for more information about the KMeans method.

**Figure 20.1** Hierarchical Cluster Launch Dialog



You can specify as many  $Y$  variables as you want by selecting the variables in the column selector list and clicking **Y, Columns**.

For Hierarchical clustering, you must also choose one of the clustering distance options: Average, Centroid, Ward, Single, and Complete, and Fast Ward. The clustering methods differ in how the distance between two clusters is computed. These clustering methods are discussed under “[Technical Details for Hierarchical Clustering](#),” p. 448.

By default, data are first standardized by the column mean and standard deviation. Uncheck the **Standardize Data** check box if you do not want the cluster distances computed on standardized values.

You can use the **Data is distance matrix** option if you have a data table of distances instead of raw data. If your raw data consists of  $n$  observations, the distance table should have  $n$  rows and  $n$  columns, with the

values being the distances between the observations. The distance table needs to have an additional column giving a unique identifier (such as row number) that matches the column names of the other  $n$  columns. The diagonal elements of the table should be zero, since the distance between a point and itself is zero. For an example of what the distance table should look like, use the option “[Save Distance Matrix](#),” p. 448.

An **Ordering** column can (optionally) be specified in the Hierarchical Clustering launch dialog. In the ordering column, clusters are sorted by their mean value. One way to utilize this feature is to complete a Principal Components analysis (using **Multivariate**) and save the first principal component to use as an **Ordering** column. The clusters are then sorted by these values.

Hierarchical clustering supports character columns as follows. K-Means clustering only supports numeric columns.

- For Ordinal columns, the data value used for clustering is just the index of the ordered category, treated as if it were continuous data. These data values are standardized like continuous columns.
- For Nominal columns, the categories must either match to contribute a distance of zero, or contribute a standardized distance of 1.

---

## Hierarchical Clustering

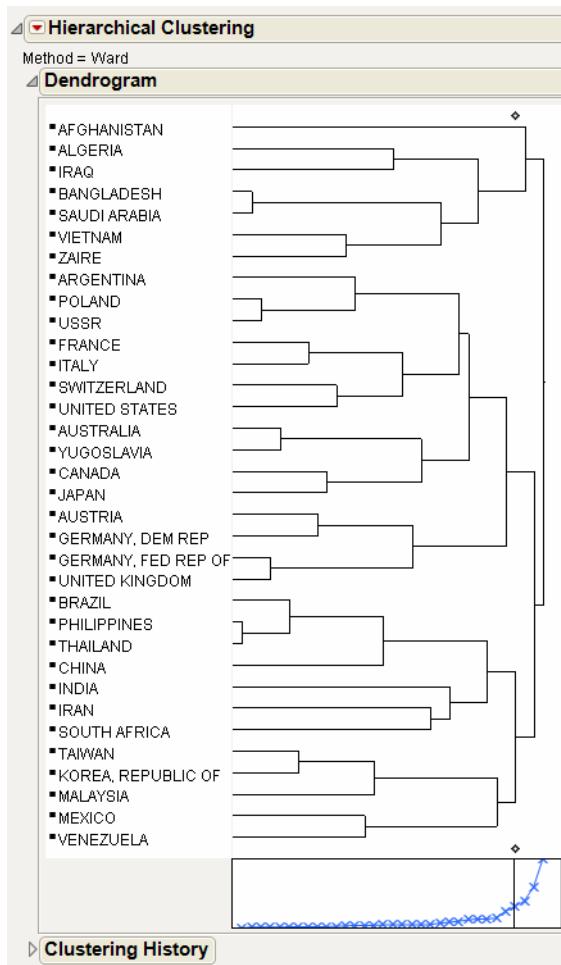
The **Hierarchical** option clusters rows that group the points (rows) of a JMP table into clusters whose values are close to each other relative to those of other clusters. Hierarchical clustering is a process that starts with each point in its own cluster. At each step, the two clusters that are closest together are combined into a single cluster. This process continues until there is only one cluster containing all the points. This kind of clustering is good for smaller data sets (a few hundred observations).

To see a simple example of hierarchical clustering, select the *Birth Death Subset.jmp* data table. The data are the 1976 crude birth and death rates per 100,000 people.

When you select the **Cluster** command, the Cluster Launch dialog (shown previously in Figure 20.1) appears. In this example the **birth** and **death** columns are used as cluster variables in the default method, Ward’s minimum variance, for hierarchical clustering.

In clustering problems, it is especially useful to assign a label variable so that you can determine which observations cluster together. In this example, **country** is the label variable.

When you click **OK**, the clustering process proceeds. The Hierarchical Cluster platform report (see Figure 20.2) consists of a Clustering History table, a dendrogram tree diagram, and a plot of the distances between the clusters. If you assigned a variable the label role, its values identify each observation on the dendrogram.

**Figure 20.2** Hierarchical Cluster Report

The clustering sequence is easily visualized with the help of the *dendrogram*, shown in Figure 20.2. A dendrogram is a tree diagram that lists each observation, and shows which cluster it is in and when it entered its cluster.

You can drag the small diamond-shaped handle at either the top or bottom of the dendrogram to identify a given number of clusters. If you click on any cluster stem, all the members of the cluster highlight in the dendrogram and in the data table.

The scree plot beneath the dendrogram has a point for each cluster join. The ordinate is the distance that was bridged to join the clusters at each step. Often there is a natural break where the distance jumps up suddenly. These breaks suggest natural cutting points to determine the number of clusters.

Open the Clustering History table to see the results shown in Figure 20.3.

**Figure 20.3** Clustering History

Clustering History			
Number of Clusters	Distance	Leader	Joiner
33	0.000000000	PHILIPPINES	THAILAND
32	0.104271097	BANGLADESH	SAUDI ARABIA
31	0.104271097	POLAND	USSR
30	0.104271097	GERMANY, FED REP OF	UNITED KINGDOM
29	0.104271097	AUSTRALIA	YUGOSLAVIA
28	0.120401892	BRAZIL	PHILIPPINES
27	0.140113736	TAIWAN	KOREA, REPUBLIC OF
26	0.140113736	FRANCE	ITALY
25	0.140113736	AUSTRIA	GERMANY, DEM REP
24	0.140113736	CANADA	JAPAN
23	0.156406646	SWITZERLAND	UNITED STATES
22	0.209987852	VIETNAM	ZAIRE
21	0.242473085	ARGENTINA	POLAND
20	0.251240335	MEXICO	VENEZUELA
19	0.254028376	TAIWAN	MALAYSIA
18	0.290281099	BRAZIL	CHINA
17	0.298998156	ALGERIA	IRAQ
16	0.299503627	FRANCE	SWITZERLAND
15	0.306234544	AUSTRIA	GERMANY, FED REP OF
14	0.306234544	AUSTRALIA	CANADA
13	0.312813292	IRAN	SOUTH AFRICA
12	0.445451502	BANGLADESH	VIETNAM
11	0.517879998	INDIA	IRAN
10	0.612959173	ARGENTINA	FRANCE
9	0.733942725	ARGENTINA	AUSTRALIA
8	0.793425680	ALGERIA	BANGLADESH
7	0.823015833	BRAZIL	INDIA
6	0.880310450	TAIWAN	MEXICO
5	1.501738614	ARGENTINA	AUSTRIA
4	1.903914282	BRAZIL	TAIWAN
3	2.380570164	AFGHANISTAN	ALGERIA
2	3.590473842	ARGENTINA	BRAZIL
1	6.095747490	AFGHANISTAN	ARGENTINA

The number of clusters begins with 33, which is the number of rows in the data table minus one. You can see that the two closest points, the Philippines and Thailand, are joined to reduce the number of existing clusters to 32. They show as the first Leader and Joiner in the Clustering History table. The next two closest points are Bangladesh and Saudi Arabia, followed by Poland and USSR. When Brazil is joined by the Philippines in the sixth line, the Philippines had already been joined by Thailand, making it the first cluster with three points. The last single point to be joined to others is Afghanistan, which reduces the number of clusters from four to three at that join. At the very end a cluster of seven points led by Argentina is joined by the rest of the points, led by Argentina. The order of the clusters at each join is unimportant, essentially an accident of the way the data was sorted.

## Hierarchical Cluster Options

Hierarchical clustering has the popup menu with the following commands.

**Color Clusters** assigns colors to the rows of the data table corresponding to the cluster the row belongs to. Also colors the dendrogram according to the clusters. The colors automatically update if you change the number of clusters.

**Mark Clusters** assigns markers to the rows of the data table corresponding to the cluster the row belongs to. The markers automatically update if you change the number of clusters.

**Number of Clusters** prompts you to enter a number of clusters and positions the dendrogram slider to that number.

**Show Dendogram** shows or hides the dendogram.

**Dendrogram Scale** contains options for scaling the dendrogram. **Even Spacing** shows the distance between each join point as equal. **Geometric Spacing** is useful when there are many clusters and you want the clusters near the top of the tree to be more visible than those at the bottom. (This option is the default for more than 256 rows). **Distance Scale** shows the actual joining distance between each join point, and is the same scale used on the plot produced by the **Distance Graph** command.

**Distance Graph** shows or hides the scree plot at the bottom of the histogram.

**Show NCluster Handle** shows or hides the handles on the dendrogram used to manually change the number of clusters.

**Zoom to Selected Rows** is used to zoom the dendrogram to a particular cluster after selecting the cluster on the dendrogram. Alternatively, you can double-click on a cluster to zoom in on it.

**Release Zoom** returns the dendrogram to original view after zooming.

**Pivot on Selected Cluster** reverses the order of the two sub-clusters of the currently selected cluster.

**Color Map** is an option to add a color map showing the values of all the data colored across its value range. There are several color theme choices in a submenu. Another term for this feature is *heat map*.

**Two way clustering** adds clustering by column. A color map is automatically added with the column dendrogram at its base. The columns must be measured on the same scale.

**Positioning** provides options for changing the positions of dendograms and labels.

**Legend** shows or hides a legend for the colors used in a color map. This option is available only if a color map is enabled.

**More Color Map Columns** adds a color map for specified columns.

**Save Clusters** creates a data table column containing the cluster number.

**Save Display Order** creates a data table column containing the order the row is presented in the dendrogram.

**Save Cluster Hierarchy** saves information needed if you are going to do a custom dendrogram with scripting. For each clustering it outputs three rows, the joiner, the leader, and the result, with the cluster centers, size, and other information.

**Save Distance Matrix** makes a new data table containing the distances between the observations.

**Parallel Coord Plots** creates a parallel coordinate plot for each cluster. For details about the plots, see *Basic Analysis and Graphing*.

**Script** contains options common to all platforms used for repeating analyses and saving scripts.

## Technical Details for Hierarchical Clustering

The following description of hierarchical clustering methods gives distance formulas that use the following notation. Lowercase symbols generally pertain to observations and uppercase symbols to clusters.

$n$  is the number of observations

$v$  is the number of variables

$x_i$  is the  $i$ th observation

$C_K$  is the  $K$ th cluster, subset of  $\{1, 2, \dots, n\}$

$N_K$  is the number of observations in  $C_K$

$\bar{x}$  is the sample mean vector

$\bar{x}_K$  is the mean vector for cluster  $C_K$

$\|\mathbf{x}\|$  is the square root of the sum of the squares of the elements of  $\mathbf{x}$  (the Euclidean length of the vector  $\mathbf{x}$ )

$d(x_i, x_j)$  is  $\|\mathbf{x}\|^2$

**Average Linkage** In average linkage, the distance between two clusters is the average distance between pairs of observations, or one in each cluster. Average linkage tends to join clusters with small variances and is slightly biased toward producing clusters with the same variance. See Sokal and Michener (1958).

Distance for the average linkage cluster method is

$$D_{KL} = \sum_{i \in C_K} \sum_{j \in C_L} \frac{d(x_i, x_j)}{N_K N_L}$$

**Centroid Method** In the centroid method, the distance between two clusters is defined as the squared Euclidean distance between their means. The centroid method is more robust to outliers than most other hierarchical methods but in other respects might not perform as well as Ward's method or average linkage. See Milligan (1980).

Distance for the centroid method of clustering is

$$D_{KL} = \|\bar{x}_K - \bar{x}_L\|^2$$

**Ward's** In Ward's minimum variance method, the distance between two clusters is the ANOVA sum of squares between the two clusters added up over all the variables. At each generation, the within-cluster sum of squares is minimized over all partitions obtainable by merging two clusters from the previous generation. The sums of squares are easier to interpret when they are divided by the total sum of squares to give the proportions of variance (squared semipartial correlations).

Ward's method joins clusters to maximize the likelihood at each level of the hierarchy under the assumptions of multivariate normal mixtures, spherical covariance matrices, and equal sampling probabilities.

Ward's method tends to join clusters with a small number of observations and is strongly biased toward producing clusters with roughly the same number of observations. It is also very sensitive to outliers. See Milligan (1980).

Distance for Ward's method is

$$D_{KL} = \frac{\left\| \bar{x}_K - \bar{x}_L \right\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

**Single Linkage** In single linkage the distance between two clusters is the minimum distance between an observation in one cluster and an observation in the other cluster. Single linkage has many desirable theoretical properties. See Jardine and Sibson (1976), Fisher and Van Ness (1971), and Hartigan (1981). Single linkage has, however, fared poorly in Monte Carlo studies. See Milligan (1980). By imposing no constraints on the shape of clusters, single linkage sacrifices performance in the recovery of compact clusters in return for the ability to detect elongated and irregular clusters. Single linkage tends to chop off the tails of distributions before separating the main clusters. See Hartigan (1981). Single linkage was originated by Florek et al. (1951a, 1951b) and later reinvented by McQuitty (1957) and Sneath (1957).

Distance for the single linkage cluster method is

$$D_{KL} = \min_{i \in C_K} \min_{j \in C_L} d(x_i, x_j)$$

**Complete Linkage** In complete linkage, the distance between two clusters is the maximum distance between an observation in one cluster and an observation in the other cluster. Complete linkage is strongly biased toward producing clusters with roughly equal diameters and can be severely distorted by moderate outliers. See Milligan (1980).

Distance for the Complete linkage cluster method is

$$D_{KL} = \max_{i \in C_K} \max_{j \in C_L} d(x_i, x_j)$$

**Fast Ward** is a method of applying the Ward's method more quickly for large numbers of rows. It is used automatically whenever there are more than 2000 rows.

## K-Means Clustering

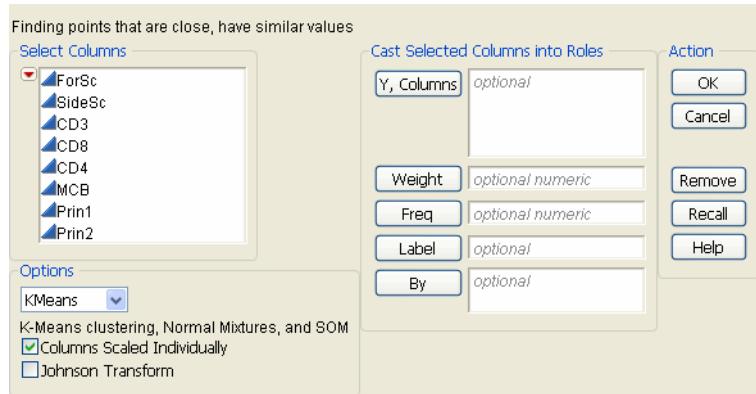
The  $k$ -means approach to clustering performs an iterative alternating fitting process to form the number of specified clusters. The  $k$ -means method first selects a set of  $n$  points called *cluster seeds* as a first guess of the means of the clusters. Each observation is assigned to the nearest seed to form a set of temporary clusters. The seeds are then replaced by the cluster means, the points are reassigned, and the process continues until no further changes occur in the clusters. When the clustering process is finished, you see tables showing brief summaries of the clusters. The  $k$ -means approach is a special case of a general approach called the *EM algorithm*, where E stands for Expectation (the cluster means in this case) and the M stands for maximization, which means assigning points to closest clusters in this case.

The  $k$ -means method is intended for use with larger data tables, from approximately 200 to 100,000 observations. With smaller data tables, the results can be highly sensitive to the order of the observations in the data table.

*K*-Means clustering only supports numeric columns. *K*-Means clustering ignores model types (nominal and ordinal), and treat all numeric columns as continuous columns.

To see the KMeans cluster launch dialog (see Figure 20.4), select **KMeans** from the Options menu on the platform launch dialog. The figure uses the Cytometry.jmp data table.

**Figure 20.4** KMeans Launch Dialog



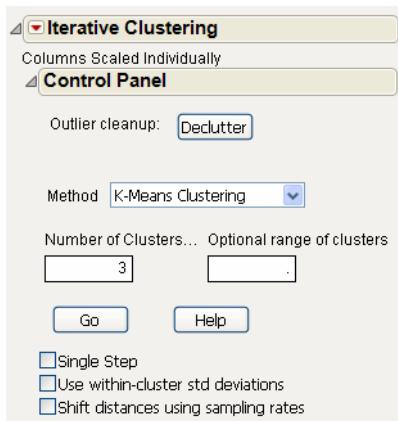
The dialog has the following options:

**Columns Scaled Individually** is used when variables don't share a common measurement scale, and you don't want one variable to dominate the clustering process. For example, one variable may have values that are between 0-1000, and another variable may have values between 0-10. In this situation, you can use the option so the clustering process is not dominated by the first variable.

**Johnson Transform** can be used to balance highly skewed variables, or to bring outliers closer to the center of the rest of the values.

## K-Means Control Panel

As an example of KMeans clustering, use the Cytometry.jmp sample data table. Add the variables CD3 and CD8 as **Y, Columns** variables. Click **OK**. The Control Panel appears, and is shown in Figure 20.5.

**Figure 20.5** Iterative Clustering Control Panel

The Iterative Clustering red-triangle menu has the **Save Transformed** option. This saves the Johnson transformed variables to the data table. This option is available only if the Johnson Transform option is selected on the launch dialog (Figure 20.4).

The Control Panel has these options:

**Declutter** is used to locate outliers in the multivariate sense. Plots are produced giving distances between each point and that points nearest neighbor, the second nearest neighbor, up to the  $k^{\text{th}}$  nearest neighbor. You are prompted to enter  $k$ . Beneath the plots are options to create a scatterplot matrix, and save the distances to the data table. If an outlier is identified, you may want to exclude the row from the clustering process.

**Method** is used to choose the clustering method. The available methods are:

**KMeans Clustering** is described in this section.

**Normal Mixtures** is described in “[Normal Mixtures](#),” p. 454.

**Robust Normal Mixtures** is described in “[Normal Mixtures](#),” p. 454.

**Self Organizing Map** is described in “[Self Organizing Maps](#),” p. 458.

**Number of Clusters** is the number of clusters to form.

**Optional range of clusters** is an upper bound for the number of clusters to form. If a number is entered here, the platform creates separate analyses for every integer between **Number of clusters** and this one.

**Single Step** enables you to step through the clustering process one iteration at a time using a **Step** button, or automate the process using a **Go** button.

**Use within-cluster std deviations** If you don’t use this option, all distances are scaled by an overall estimate of the standard deviation of each variable. If you use this option, then it will calculate distances scaled by the standard deviation estimated for each cluster.

**Shift distances using sampling rates** assumes that you have a mix of unequally sized clusters, and points should give preference to being assigned to larger clusters because there is a greater prior

probability that it is from a larger cluster. This option is an advanced feature. The calculations for this option are implied, but not shown for normal mixtures.

## K-Means Report

Clicking **Go** in the Control Panel in Figure 20.5 produces the K-Means report, shown in Figure 20.6.

**Figure 20.6** K-Means Report

K Means NCluster=3			
Columns Scaled Individually			
Cluster Summary			
Cluster	Count	Step	Criterion
1	2,145	4	0
2	1,788		
3	1,067		
Cluster Means			
Cluster	CD3	CD8	
1	186.811189	119.101632	
2	318.097315	303.381432	
3	332.606373	99.0824742	
Cluster Standard Deviations			
Cluster	CD3	CD8	
1	51.3596512	37.8377585	
2	22.8409179	25.7223588	
3	32.591227	37.0441205	

The report gives summary statistics for each cluster:

- count of number of observations
- means for each variable
- standard deviations for each variable.

## K-Means Platform Options

These options are accessed from the red-triangle menus, and apply to KMeans, Normal Mixtures, Robust Normal Mixtures, and Self-Organizing Map methods.

**Biplot** shows a plot of the points and clusters in the first two principal components of the data. Circles are drawn around the cluster centers with area proportional to the number of points in the cluster. Below the plot is an option to save the cluster colors to the data table.

**Biplot Options** contains options for controlling the Biplot.

**Show Biplot Rays** allows you to show or hide the biplot rays.

**Biplot Ray Position** allows you to position the biplot ray display. This is viable since biplot rays only signify the directions of the original variables in canonical space, and there is no special significance to where they are placed in the graph.

**Mark Clusters** assigns markers to the rows of the data table corresponding to the clusters.

- Biplot 3D** shows a three-dimensional biplot of the data. Three variables are needed to use this option.
- Parallel Coord Plots** creates a parallel coordinate plot for each cluster. For details about the plots, see *Basic Analysis and Graphing*. The plot report has options for showing and hiding the data and means.
- Scatterplot Matrix** creates a scatterplot matrix using all the variables.
- Save Colors to Table** colors each row with a color corresponding to the cluster it is in.
- Save Clusters** creates a new column with the cluster number that each row is assigned to. For normal mixtures, this is the cluster that is most likely.
- Save Cluster Formula** creates a new column with a formula to evaluate which cluster the row belongs to.
- Save Mixture Probabilities** creates a column for each cluster and saves the probability an observation belongs to that cluster in the column. This is available for Normal Mixtures and Robust Normal Mixtures clustering only.
- Save Mixture Formulas** creates columns with mixture probabilities, but stores their formulas in the column and needs additional columns to hold intermediate results for the formulas. Use this feature if you want to score probabilities for excluded data, or data you add to the table. This is available for Normal Mixtures and Robust Normal Mixtures clustering only.
- Save Density Formula** saves the density formula in the data table. This is available for Normal Mixtures clustering only.
- Simulate Clusters** creates a new data table containing simulated clusters using the mixing probabilities, means, and standard deviations.
- Remove** removes the clustering report.

---

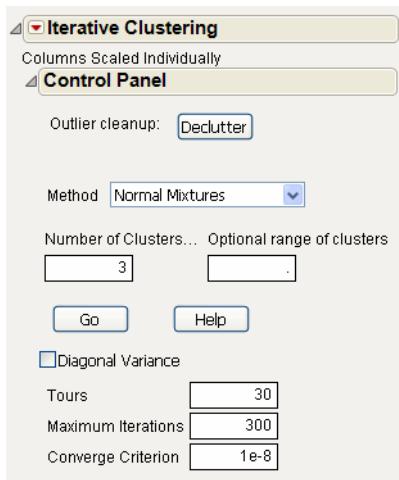
## Normal Mixtures

Normal mixtures is an iterative technique, but rather than being a clustering method to group rows, it is more of an estimation method to characterize the cluster groups. Rather than classifying each row into a cluster, it estimates the probability that a row is in each cluster. See McLachlan and Krishnan (1997).

The normal mixtures approach to clustering predicts the proportion of responses expected within each cluster. The assumption is that the joint probability distribution of the measurement columns can be approximated using a mixture of multivariate normal distributions, which represent different clusters. The distributions have mean vectors and covariance matrices for each cluster.

Hierarchical and  $k$ -means clustering methods work well when clusters are well separated, but when clusters overlap, assigning each point to one cluster is problematic. In the overlap areas, there are points from several clusters sharing the same space. It is especially important to use normal mixtures rather than  $k$ -means clustering if you want an accurate estimate of the total population in each group, because it is based on membership probabilities, rather than arbitrary cluster assignments based on borders.

To perform Normal Mixtures, select that option on the Method menu of the Iterative Clustering Control Panel (Figure 20.5). After selecting Normal Mixtures, the control panel looks like Figure 20.7.

**Figure 20.7** Normal Mixtures Control Panel

Some of the options on the panel are described in “[K-Means Control Panel](#),” p. 451. The other options are described below:

**Diagonal Variance** is used to constrain the off-diagonal elements of the covariance matrix to zero. In this case, the platform fits multivariate normal distributions that have no correlations between the variables.

This is sometimes necessary in order to avoid getting a singular covariance matrix, when there are fewer observations than columns.

**Tours** is the number of independent restarts of estimation process, each with different starting values. This helps to guard against finding local solutions.

**Maximum Iterations** is the maximum number of iterations of the convergence stage of the EM algorithm.

**Converge Criteria** is the difference in the likelihood at which the EM iterations stop.

For an example of **Normal Mixtures**, open the **Iris.jmp** sample data table. This data set was first introduced by Fisher (1936), and includes four different measurements: sepal length, sepal width, petal length, and petal width, performed on samples of 50 each for three species of iris.

**Note:** Your results may not exactly match these results due to the random selection of initial centers.

On the Cluster launch dialog, assign all four variables to the **Y, Columns** role, select **KMeans** from Method menu, and click **OK**. Select **Normal Mixtures** from the Method menu, specify 3 for the Number of Clusters, and click **Go**. The report is shown in Figure 20.8.

**Figure 20.8** Normal Mixtures Report

Normal Mixtures NCluster=3				
Cluster Summary				
Cluster	Count	Proportion		
1	35	0.2309928		
2	50	0.3332438		
3	65	0.4357634		
Cluster Means				
Cluster	Sepal length	Sepal width	Petal length	
1	6.37566247	2.99671394	5.33216719	
2	5.00630333	3.42822867	1.4625027	
3	6.20125076	2.80582105	4.67897146	
			Petal width	
1	2.10613675			
Cluster Standard Deviations				
Cluster	Sepal length	Sepal width	Petal length	
1	0.46266289	0.2358482	0.40899603	
2	0.32344967	0.3992495	0.23133759	
3	0.65953715	0.31679811	0.83378953	
			Petal width	
1	0.22566686			
LogLikelihood				
	197.41345			
Correlations for Normal Mixtures				
1	Sepal length	Sepal width	Petal length	Petal width
Sepal length	1.0000	0.5099	0.7351	0.5766
Sepal width	0.5099	1.0000	0.4542	0.5385
Petal length	0.7351	0.4542	1.0000	0.7971
Petal width	0.5766	0.5385	0.7971	1.0000
2	Sepal length	Sepal width	Petal length	Petal width
Sepal length	1.0000	0.7096	0.3343	0.3381
Sepal width	0.7096	1.0000	0.0767	0.1523
Petal length	0.3343	0.0767	1.0000	0.6196
Petal width	0.3381	0.1523	0.6196	1.0000
3	Sepal length	Sepal width	Petal length	Petal width
Sepal length	1.0000	0.5387	0.8784	0.8045
Sepal width	0.5387	1.0000	0.4362	0.5134
Petal length	0.8784	0.4362	1.0000	0.9178
Petal width	0.8045	0.5134	0.9178	1.0000

The report gives summary statistics for each cluster:

- count of number of observations and proportions
- means for each variable
- standard deviations for each variable.
- correlations between variables

## Robust Normal Mixtures

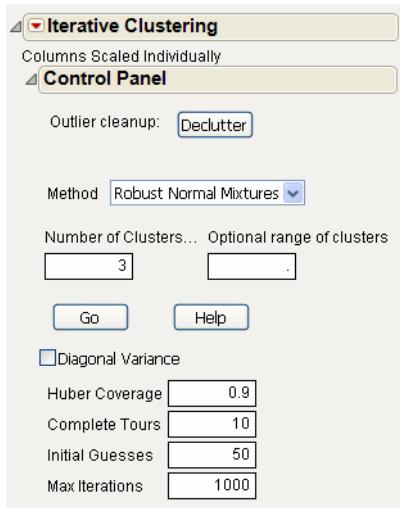
The Robust Normal Mixtures option is available if you suspect you may have outliers in the multivariate sense. Since regular Normal Mixtures is sensitive to outliers, the Robust Normal Mixtures option uses a

more robust method for estimating the parameters. For details, see “[Additional Details for Robust Normal Mixtures](#),” p. 458.

To perform Robust Normal Mixtures, select that option on the Method menu of the Iterative Clustering Control Panel (Figure 20.5). After selecting Robust Normal Mixtures, the control panel looks like Figure 20.9.

---

**Figure 20.9** Normal Mixtures Control Panel



---

Some of the options on the panel are described in “[K-Means Control Panel](#),” p. 451. The other options are described below:

**Diagonal Variance** is used to constrain the off-diagonal elements of the covariance matrix to zero. In this case, the platform fits multivariate normal distributions that have no correlations between the variables.

This is sometimes necessary in order to avoid getting a singular covariance matrix, when there are fewer observations than columns.

**Huber Coverage** is a number between 0 and 1. Robust Normal Mixtures protects against outliers by down-weighting them. Huber Coverage can be loosely thought of as the proportion of the data that is not considered outliers, and not down-weighted. Values closer to 1 result in a larger proportion of the data not being down-weighted. In other words, values closer to 1 protect only against the most extreme outliers. Values closer to 0 result in a smaller proportion of the data not being down-weighted, and may falsely consider less extreme data points to be outliers.

**Complete Tours** is the number of times to restart the estimation process. This helps to guard against the process finding a local solution.

**Initial Guesses** is the number of random starts within each tour. Random starting values for the parameters are used for each new start.

**Max Iterations** is the maximum number of iterations during the convergence stage. The convergence stage starts after all tours are complete. It begins at the optimal result out of all the starts and tours, and from there converges to a final solution.

## Platform Options

For details on the red-triangle options for Normal Mixtures and Robust Normal Mixtures, see “[K-Means Platform Options](#),” p. 453.

## Details of the Estimation Process

Normal Mixtures uses the EM algorithm to do fitting because it is more stable than the Newton-Raphson algorithm. Additionally we're using a Bayesian regularized version of the EM algorithm, which allows us to smoothly handle cases where the covariance matrix is singular. Since the estimates are heavily dependent on initial guesses, the platform will go through a number of tours, each with randomly selected points as initial centers.

Doing multiple tours makes the estimation process somewhat expensive, so considerable patience is required for large problems. Controls allow you to specify the tour and iteration limits.

### Additional Details for Robust Normal Mixtures

Because Normal Mixtures is sensitive to outliers, JMP offers an outlier robust alternative called Robust Normal Mixtures. This uses a robust method of estimating the normal parameters. JMP computes the estimates via maximum likelihood with respect to a mixture of Huberized normal distributions (a class of modified normal distributions that was tailor-made to be more outlier resistant than the normal distribution).

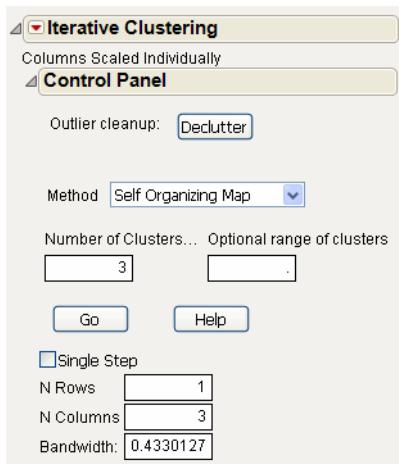
---

## Self Organizing Maps

The *Self-Organizing Maps* (SOMs) technique was developed by Teuvo Kohonen (1989) and further extended by a number of other neural network enthusiasts and statisticians. The original SOM was cast as a learning process, like the original neural net algorithms, but the version implemented here is done in a much more straightforward way as a simple variation on  $k$ -means clustering. In the SOM literature, this would be called a *batch algorithm* using a *locally weighted linear smoother*.

The goal of a SOM is to not only form clusters, but form them in a particular layout on a cluster grid, such that points in clusters that are near each other in the SOM grid are also near each other in multivariate space. In classical  $k$ -means clustering, the structure of the clusters is arbitrary, but in SOMs the clusters have the grid structure. This grid structure helps to interpret the clusters in two dimensions: clusters that are close are more similar than distant clusters.

To create a Self Organizing Map, select that option on the Method menu of the Iterative Clustering Control Panel (Figure 20.5). After selecting Self Organizing Map, the control panel looks like Figure 20.10.

**Figure 20.10** Self Organizing Map Control Panel

Some of the options on the panel are described in “[K-Means Control Panel](#),” p. 451. The other options are described below:

**N Rows** is the number of rows in the cluster grid.

**N Columns** is the number of columns in the cluster grid.

**Bandwidth** determines the effect of neighboring clusters for predicting centroids. A higher bandwidth results in a smoother fitting of the data.

As an example of a SOM, use the Iris.jmp sample data table to follow the steps below:

1. Select **Analyze > Multivariate Methods > Cluster**.
2. Assign all four columns as **Y, Column** variables.
3. Select **K Means** on the Options menu.
4. Uncheck **Columns Scaled Individually**.
5. Click **OK**.
6. Select **Self Organizing Map** from the Method menu on the Control Panel.
7. Since we know the data consists of three species, set **Number of Clusters** equal to 3.
8. Set **N Rows** equal to 1 and **N Columns** equal to 3.
9. Click **Go**. The report is shown in Figure 20.11.

**Figure 20.11** Self Organizing Map Report

SOM Grid 3 by 1				
Columns have common scale				
Bandwidth: 0.4330127				
Cluster Summary				
Cluster	Count	Step	Criterion	
1	38	4	0	
2	62			
3	50			
Cluster Means				
Cluster	Sepal length	Sepal width	Petal length	Petal width
1	6.84999838	3.07365342	5.74215376	2.0710695
2	5.90394147	2.79272704	4.32370157	1.40957274
3	5.00599907	3.42798221	1.46202802	0.24600975
Cluster Standard Deviations				
Cluster	Sepal length	Sepal width	Petal length	Petal width
1	0.48760964	0.28824999	0.48211791	0.27616537
2	0.46263925	0.29721103	0.50958381	0.29608946
3	0.34894699	0.37525458	0.17191859	0.10432641

The report gives summary statistics for each cluster:

- count of number of observations
- means for each variable
- standard deviations for each variable.

For details on the red-triangle options for Self Organizing Maps, see “[K-Means Platform Options](#),” p. 453.

### Implementation Technical Details

The SOM implementation in JMP proceeds as follows:

- The first step is to obtain good initial cluster seeds that provide a good coverage of the multidimensional space. JMP uses principal components to determine the two directions which capture the most variation in the data.
- JMP then lays out a grid in this principal component space with its edges 2.5 standard deviations from the middle in each direction. The clusters seeds are formed by translating this grid back into the original space of the variables.
- The cluster assignment proceeds as with  $k$ -means, with each point assigned to the cluster closest to it.
- The means are estimated for each cluster as in  $k$ -means. JMP then uses these means to set up a weighted regression with each variable as the response in the regression, and the SOM grid coordinates as the regressors. The weighting function uses a ‘kernel’ function that gives large weight to the cluster whose center is being estimated, with smaller weights given to clusters farther away from the cluster in the SOM grid. The new cluster means are the predicted values from this regression.
- These iterations proceed until the process has converged.

# Chapter 21

## Principal Components Reducing Dimensionality

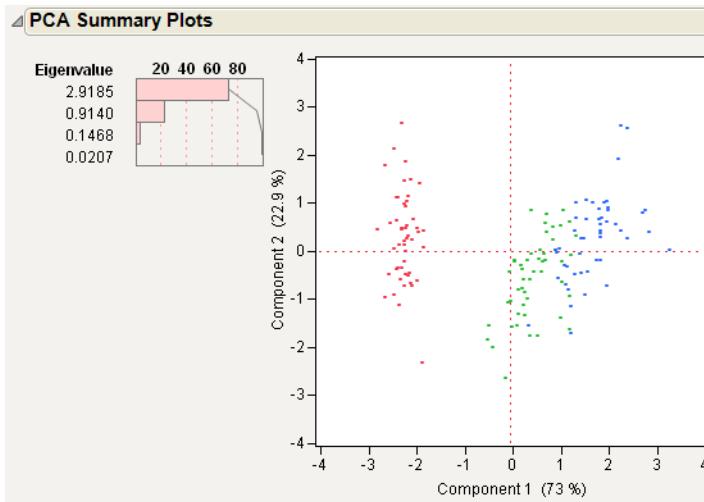
---

The purpose of principal component analysis is to derive a small number of independent linear combinations (principal components) of a set of variables that capture as much of the variability in the original variables as possible.

JMP also offers several types of orthogonal and oblique Factor-Analysis-Style rotations to help interpret the extracted components. The platform also supports factor analysis.

Principal components can be accessed through the **Multivariate** platform, through the **Scatterplot 3D** platform, or through the **Principal Components** command on the **Analyze > Multivariate Methods** menu. All map to the same routines, documented in this chapter.

**Figure 21.1** Example of Principal Components



---

# Contents

Principal Components . . . . .	463
Launch the Platform . . . . .	463
Report . . . . .	464
Platform Options . . . . .	464
Factor Analysis . . . . .	467

---

## Principal Components

If you want to see the arrangement of points across many correlated variables, you can use principal component analysis to show the most prominent directions of the high-dimensional data. Using principal component analysis reduces the *dimensionality* of a set of data. Principal components is a way to picture the structure of the data as completely as possible by using as few variables as possible.

For  $n$  original variables,  $n$  principal components are formed as follows:

- The first principal component is the linear combination of the variables that has the greatest possible variance.
- Each subsequent principal component is the linear combination of the variables that has the greatest possible variance and is uncorrelated with all previously defined components.

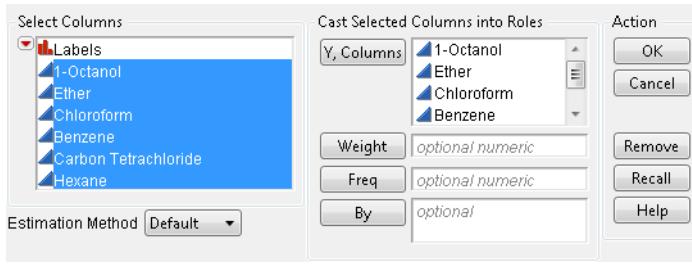
Each principal component is calculated by taking a linear combination of an eigenvector of the correlation matrix (covariance matrix or SSCP matrix) with a variable. The eigenvalues show the variance of each component.

Principal components representation is important in visualizing multivariate data by reducing it to dimensionalities that are graphable.

---

## Launch the Platform

Select **Analyze > Multivariate Methods > Principal Components** to launch the platform. JMP presents you with a dialog box to specify the variables involved in the analysis. In this example, we use all the continuous variables from the Solubility.jmp data set.



---

The **Estimation Method** list provides different methods for calculating the correlations. For details on the methods, see the “[Correlations and Multivariate Techniques](#)” chapter.

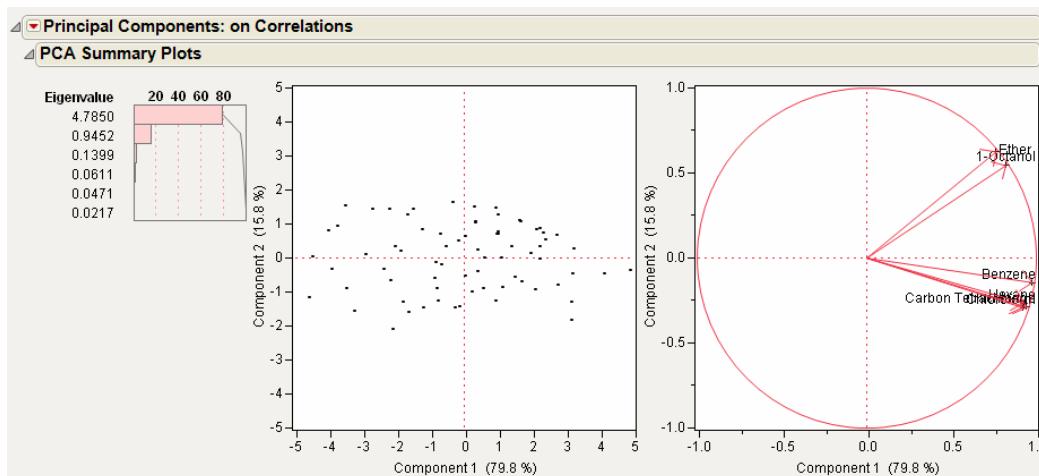
Principal components analysis is also available in the Multivariate and Scatterplot 3D platforms.

## Report

The initial principal components report (Figure 21.2) summarizes the variation of the specified  $Y$  variables with principal components. The principal components are derived from an eigenvalue decomposition of the correlation matrix, the covariance matrix, or on the unscaled and uncentered data.

The details in the report show how the principal components absorb the variation in the data. The principal component points are derived from the eigenvector linear combination of the variables.

**Figure 21.2** Principal Components/Factor Analysis Report



The report gives the eigenvalues and a bar chart of the percent of the variation accounted for by each principal component. There is a Score plot and a Loadings plot as well.

## Platform Options

The platform red-triangle menu has the following options:

**Principal Components** allows you to choose to create the principal components based on **Correlations**, **Covariances**, or **Unscaled**.

**Correlations** gives the correlations between the variables.

**Covariance Matrix** gives the variances and covariances of the variables.

**Eigenvalues** lists the eigenvalue that corresponds to each principal component in order from largest to smallest (first principal component, second principal component, and so on). The eigenvalues represent a partition of the total variation in the multivariate sample. They sum to the number of variables when the principal components analysis is done on the correlation matrix.

**Eigenvectors** shows columns of values that correspond to the eigenvectors for each of the principal components, in order, from left to right. Using these coefficients to form a linear combination of the original variables produces the principal component variables.

**Loading Matrix** shows columns corresponding to the factor loading for each component.

**Summary Plots** shows or hides the summary information produced in the initial report. This information is shown in Figure 21.2.

**Biplot** shows a plot which overlays the Score Plot and the Loading Plot.

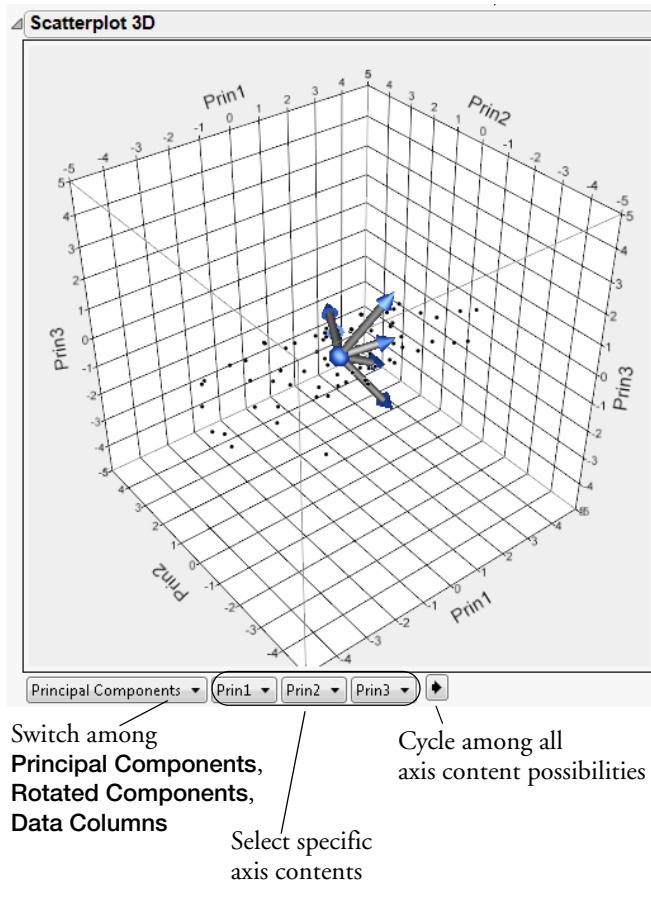
**Scree Plot** shows a scree plot of the eigenvalues vs. the number of components. This plot is useful for visualizing the dimensionality of the data space.

**Score Plot** shows a matrix of scatterplots of the scores for pairs of principal components.

**Loading Plot** shows a matrix of two-dimensional representations of factor loadings. The loading plot labels variables if the number of variables is 30 or fewer. If there are more than 30 variables, the labels are off by default.

**Score Plot with Imputation** imputes any missing values and creates a score plot. This option is available only if there are missing values.

**3D Score Plot** shows a 3D scatterplot of any principal component scores. When you first invoke the command, the first three principal components are presented.

**Figure 21.3** 3D Score Plot

The variables show as rays in the plot. These rays, called *biplot rays*, approximate the variables as a function of the principal components on the axes. If there are only two or three variables, the rays represent the variables exactly. The length of the ray corresponds to the eigenvalue or variance of the principal component.

**Factor Analysis** performs factor-analytic-style rotations of the principal components, or factor analysis. For more information, see “[Factor Analysis](#),” p. 467.

**Save Principal Components** saves the principal component to the data table, with a formula for computing the components. The formula can not evaluate rows with any missing values.

**Save Rotated Components** saves the rotated components to the data table, with a formula for computing the components. This option appears after the Factor Analysis option is used. The formula can not evaluate rows with missing values.

**Save Principal Components with Imputation** imputes missing values, and saves the principal components to the data table. The column contains a formula for doing the imputation, and computing the principal components. This option is available only if there are missing values.

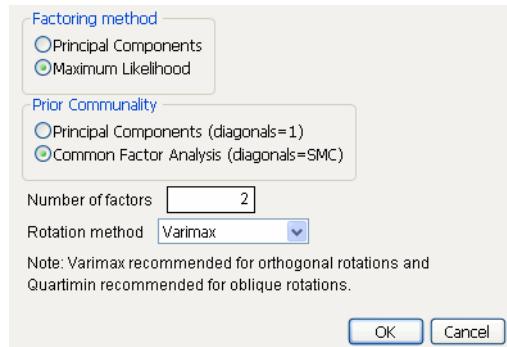
**Save Rotated Components with Imputation** imputes missing values, and saves the rotated components to the data table. The column contains a formula for doing the imputation, and computing the rotated components. This option appears after the Factor Analysis option is used, and if there are missing values.

**Script** a menu common to all platforms allowing you to reproduce results or save scripts.

## Factor Analysis

The Factor Analysis option on the platform red-triangle menu performs rotations of the principal components, or factor analysis. When the option is selected, you are presented with the window shown in Figure 21.4.

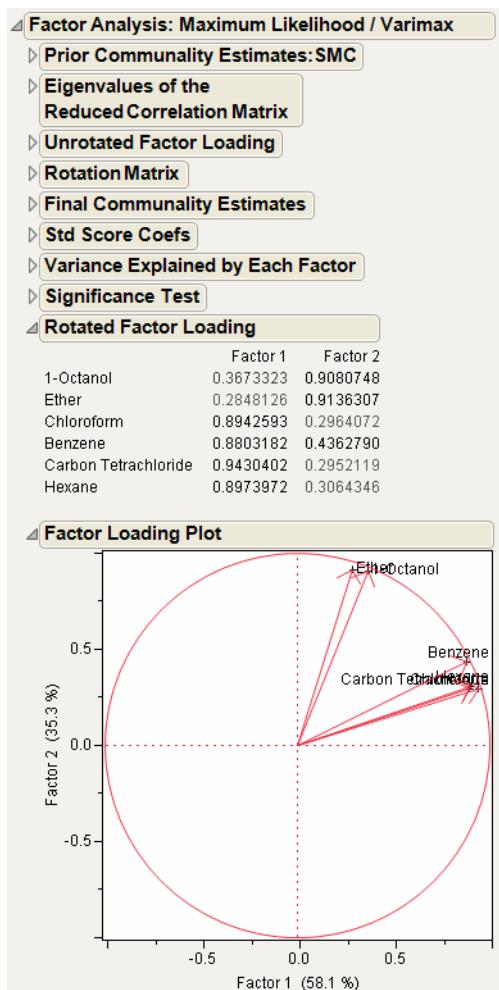
**Figure 21.4** Factor Analysis Window



Use this dialog to specify:

1. Factoring Method, either **Principal Components** or **Maximum Likelihood**. If you select the Maximum Likelihood method, a factor analytic model is estimated.
2. Prior Communality, **Principal Components** (where the diagonals are 1) or **Common Factor Analysis** (where the diagonals are SMC, the squared multiple correlation between  $X_i$  and the other  $p - 1$  variables).
3. The number of principal components or factors to rotate.
4. The rotation method. See “[Rotation Methods](#),” p. 468.

After all selections are made, click OK. The report is shown in Figure 21.5.

**Figure 21.5** Factor Analysis Report

Note that the first factor loads on the Carbon Tetrachloride-Chloroform-Benzene-Hexane cluster and that the second factor loads on the Ether-1-octanol cluster.

### Rotation Methods

Rotations are used to better align the directions of the factors with the original variables so that the factors may be more interpretable. You hope for clusters of variables that are highly correlated to define the rotated factors.

After the initial extraction, the factors are uncorrelated with each other. If the factors are rotated by an orthogonal transformation, the rotated factors are also uncorrelated. If the factors are rotated by an oblique

transformation, the rotated factors become correlated. Oblique rotations often produce more useful patterns than do orthogonal rotations. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable.

Available orthogonal rotation methods are:

**Varimax** specifies the orthogonal varimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=1.

**Biquartimax** specifies the orthogonal biquartimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=0.5.

**Equamax** specifies the orthogonal equamax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=number of factors/2.

**Factorparsimax** specifies the orthogonal factor parsimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=number of variables.

**Orthomax** specifies the orthogonal orthomax rotation with orthomax weight  $p$ . The corresponding SAS Proc Factor option specification is either ROTATE=ORTHOMAX or ROTATE=ORTHOMAX( $p$ ), which specifies  $p$  as the orthomax weight or the GAMMA=value. The default  $p$  value is 1 unless specified otherwise in the GAMMA= option. For additional information on orthomax weight, see the SAS documentation, “Simplicity Functions for Rotations.”

**Parsimax** specifies the orthogonal parsimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=[(nvar(nfact - 1))/(nvar + nfact - 2)] where nvar is the number of variables, and nfact is the number of factors.

**Quartimax** specifies the orthogonal quartimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=ORTHOMAX with GAMMA=0.

Available oblique rotation methods are:

**Biquartimin** specifies the oblique biquartimin rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBBLIMIN(.5) or ROTATE=OBBLIMIN with TAU=.5.

**Covarimin** specifies the oblique covarimin rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBBLIMIN(1) or ROTATE=OBBLIMIN with TAU=1.

**Obbiquartimax** specifies the oblique biquartimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBBIQUARTIMAX.

**Obequamax** specifies the oblique equamax rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBEQUAMAX.

**Obfactorparsimax** specifies the oblique factor parsimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBFACTOPARSIMAX.

**Oblimin** specifies the oblique oblimin rotation with oblimin weight  $p$ . The corresponding SAS Proc Factor option specification is ROTATE=OBBLIMIN, where the default  $p$  value is zero, unless specified otherwise in the TAU= option. ROTATE=OBBLIMIN( $p$ ) specifies  $p$  as the oblimin weight or the TAU= value. For additional information on oblimin weight, see the SAS documentation, “Simplicity Functions for Rotations.”

**Obparsimax** specifies the oblique parsimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBPARSIMAX.

**Obquartimax** specifies the oblique quartimax rotation. This is equivalent to the QUARTIMIN method. The corresponding SAS Proc Factor option specification is ROTATE=OBQUARTIMAX.

**Obvarimax** specifies the oblique varimax rotation. The corresponding SAS Proc Factor option specification is ROTATE=OBVARIMAX.

**Quartimin** specifies the oblique quartimin rotation. It is equivalent to the oblique quartimax method. The corresponding SAS Proc Factor option specification is ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

**Promax** specifies oblique promax rotation. The corresponding SAS Proc Factor option specification is ROTATE=PROMAX.

# Chapter **22**

## **Discriminant Analysis**

### The Discriminant Platform

---

Discriminant Analysis seeks to find a way to predict a classification ( $X$ ) variable (nominal or ordinal) based on known continuous responses ( $Y$ ). It can be regarded as inverse prediction from a multivariate analysis of variance (MANOVA). In fact, the **Manova** personality of Fit Model also provides some discriminant features.

Features include:

- A stepwise selection dialog to help choose variables that discriminate well.
- Choice among Linear, Quadratic, or Regularized-Parameter analyses.
- The discriminant scores, showing which groups each point is near.
- A misclassification summary.
- Plots of the points and means in a canonical space, which is the space that shows the best separation of means.
- Options to save prediction distances and probabilities to the data table.

# Contents

Introduction .....	473
Discriminating Groups .....	473
Discriminant Method .....	474
Stepwise Selection .....	475
Canonical Plot .....	477
Discriminant Scores .....	477
Commands and Options .....	478
Validation .....	483

---

## Introduction

Discriminant Analysis is an alternative to logistic regression. In logistic regression, the classification variable is random and predicted by the continuous variables, whereas in discriminant analysis the classifications are fixed, and the  $Y$  variables are realizations of random variables. However, in both cases, the categorical value is predicted by the continuous.

There are several varieties of discriminant analysis. JMP implements linear and quadratic discriminant analysis, along with a method that blends both types. In linear discriminant analysis, it is assumed that the  $Y$  variables are normally distributed with the same variances and covariances, but that there are different means for each group defined by  $X$ . In quadratic discriminant analysis, the covariances can be different across groups. Both methods measure the distance from each point in the data set to each group's multivariate mean (often called a *centroid*) and classify the point to the closest group. The distance measure used is the Mahalanobis distance, which takes into account the variances and covariances between the variables.

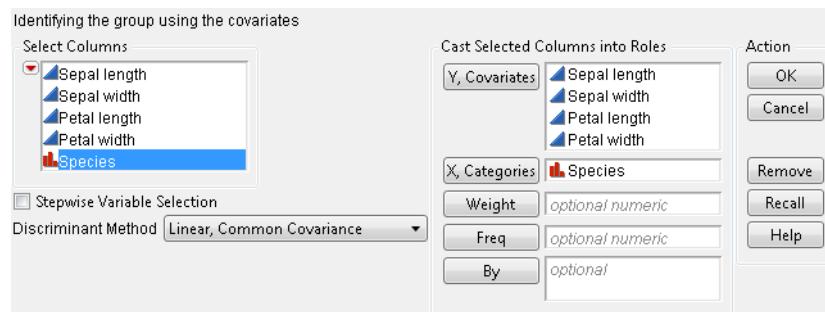
---

## Discriminating Groups

Fisher's Iris data set is the classic example of discriminant analysis. Four measurements are taken from a sample consisting of three different species. The goal is to identify the species accurately using the values of the four measurements. Open Iris.jmp, and select **Analyze > Multivariate Methods > Discriminant** to launch the Discriminant Analysis platform. The launch dialog in Figure 22.1 appears.

---

**Figure 22.1** Discriminant Launch Dialog



---

If you want to find which variables discriminate well, click the checkbox for **Stepwise Variable Selection**. Otherwise, the platform uses all the variables you specify. In this example, specify the four continuous variables as **Y, Covariates** and **Species** as **X, Categories**.

## Discriminant Method

JMP offers three kinds of Discriminant Analysis. All three calculate distances as the Mahalanobis distance from each point to each group's multivariate mean. The difference in the methods is only in the covariance matrix used in the computations.

**Linear Discriminant Analysis** uses a common (within-) covariance matrix for all groups.

**Quadratic Discriminant Analysis** uses a separate covariance matrix for each group.

Quadratic discriminant suffers in small data sets because it does not have enough data to make nicely invertible and stable covariance matrices. Regularized discriminant ameliorates these problems and still allows for differences among groups.

**Regularized Discriminant Analysis** is a compromise between the linear and quadratic methods, governed by two arguments. When you choose **Regularized Discriminant Analysis**, a dialog appears allowing specification of these two parameters.

The first parameter (**Lambda, Shrinkage to Common Covariance**) specifies how to mix the individual and group covariance matrices. For this parameter, 1 corresponds to Linear Discriminant Analysis and 0 corresponds to Quadratic Discriminant Analysis.

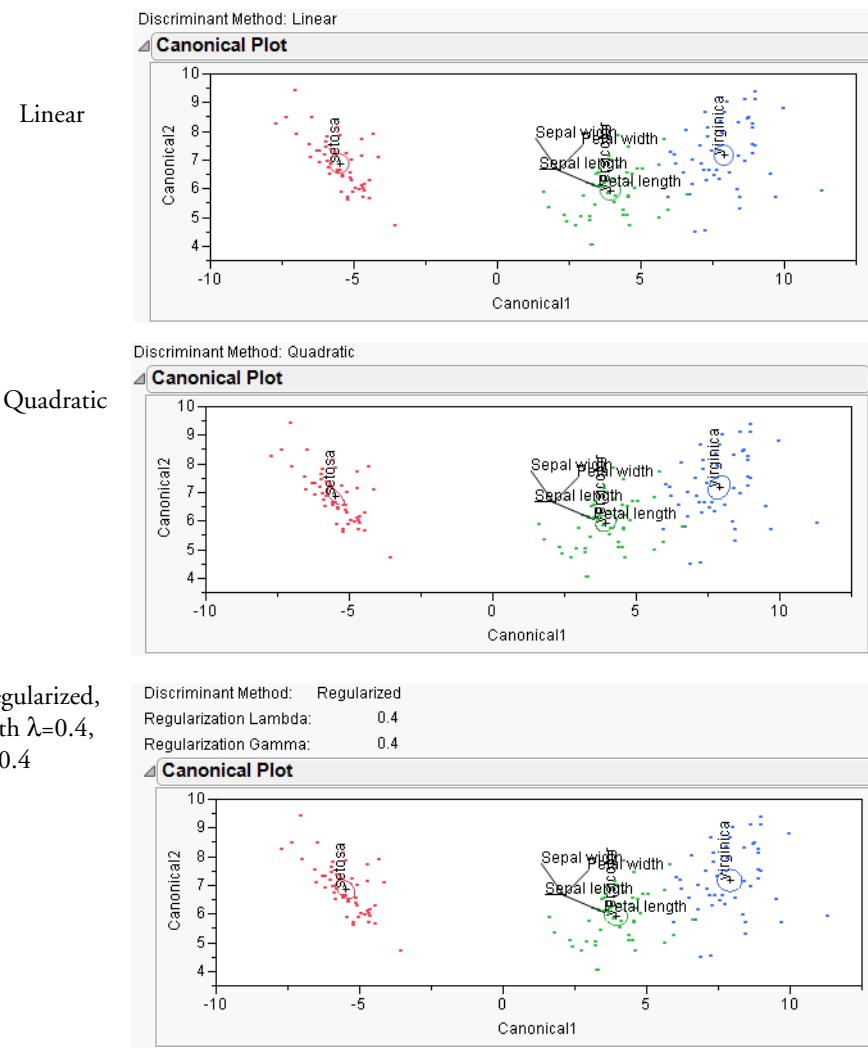
The second parameter (**Gamma, Shrinkage to Diagonal**) specifies whether to deflate the non-diagonal elements, that is, the covariances across variables. If you choose 1, then the covariance matrix is forced to be diagonal.

Therefore, assigning 0,0 to these parameters is identical to requesting quadratic discriminant analysis. Similarly, a 1,0 assignment requests linear discriminant analysis. These cases, along with a Regularized Discriminant Analysis example with  $l=0.4$  and  $g=0.4$  are shown in Figure 22.2.

Use Table 22.1 to help decide on the regularization.

**Table 22.1** Regularized Discriminant Analysis

Use lower Lambda (Gamma) when:	Use higher Lambda (Gamma) when:
Covariances are different (Variables are correlated)	Covariances are the same (Variables are uncorrelated)
lots of data	little data
small number of variables	many variables

**Figure 22.2** Linear, Quadratic, and Regularized Discriminant Analysis

## Stepwise Selection

If you choose **Stepwise Variable Selection**, a dialog appears (Figure 22.3) to select variables. You can review which columns have large  $F$  ratios or small  $p$ -values and control which columns are entered into the discriminant model. In addition, the dialog displays how many columns are currently in and out of the model, and the largest and smallest  $p$ -values to be entered or removed.

**Figure 22.3** Stepwise Control Panel

The screenshot shows the 'Column Selection' panel of the SPSS Stepwise Control Panel. It includes the following sections:

- Click to select columns into discriminant model:** Displays 'Columns In 0' and 'Smallest P to Enter 0.0000000'.
- Buttons:** Step Forward, Enter All, Step Backward, Remove All, Apply This Model.
- LockEntered Column F Ratio Prob>F:** A table showing the current state of the model. Sepal length and Sepal width are entered (checked in the Lock column). Petal length and Petal width are not yet entered (unchecked).

LockEntered Column	F Ratio	Prob>F
<input type="checkbox"/> <input checked="" type="checkbox"/> Sepal length	119.265	0.0000000
<input type="checkbox"/> <input checked="" type="checkbox"/> Sepal width	49.160	0.0000000
<input type="checkbox"/> <input type="checkbox"/> Petal length	1180.16	0.0000000
<input type="checkbox"/> <input type="checkbox"/> Petal width	960.007	0.0000000

**Entered** checkboxes show which columns are currently in the model. You can manually click columns in or out of the model.

**Lock** checkboxes are used when you want to force a column to stay in its current state regardless of any stepping by the buttons.

**Step Forward** adds the most significant column not already in the model.

**Step Backward** removes the least significant column already in the model.

**Enter All** enters all the columns into the model.

**Remove All** removes all the columns from the model.

**Apply This Model** is used when you are finished deciding the columns to include in the analysis, and want to proceed to estimation and scoring.

Figure 22.4 shows three forward steps, which add all the columns to the model except Sepal length.

**Figure 22.4** Stepped Model

The screenshot shows the 'Column Selection' panel of the SPSS Stepwise Control Panel. It includes the following sections:

- Click to select columns into discriminant model:** Displays 'Columns In 3' and 'Smallest P to Enter 0.0103288'.
- Buttons:** Step Forward, Enter All, Step Backward, Remove All, Apply This Model.
- LockEntered Column F Ratio Prob>F:** A table showing the current state of the model. Sepal width, Petal length, and Petal width are entered (checked in the Lock column). Sepal length is not yet entered (unchecked).

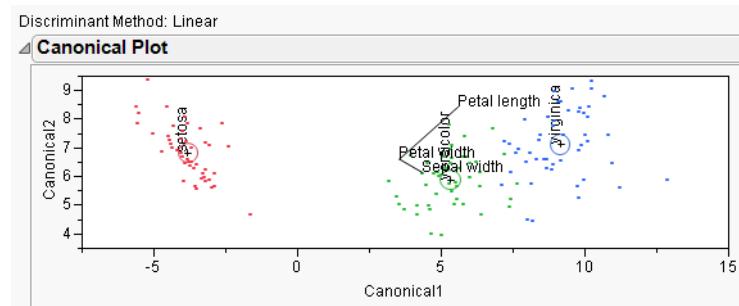
LockEntered Column	F Ratio	Prob>F
<input type="checkbox"/> <input checked="" type="checkbox"/> Sepal length	4.721	0.0103288
<input type="checkbox"/> <input checked="" type="checkbox"/> Sepal width	54.577	0.0000000
<input type="checkbox"/> <input checked="" type="checkbox"/> Petal length	38.724	0.0000000
<input type="checkbox"/> <input checked="" type="checkbox"/> Petal width	34.569	0.0000000

Click **Apply This Model** to estimate the model. After estimation and scoring are done, two reports are produced: a Canonical Plot (Figure 22.5), and a Scoring Report.

## Canonical Plot

The Canonical Plot shows the points and multivariate means in the two dimensions that best separate the groups.

**Figure 22.5** Canonical Plot



- Each row in the data set is a point, controlled by the **Canonical Options > Show Points** option.
- Each multivariate mean is a labeled circle. The size of the circle corresponds to a 95% confidence limit for the mean. Groups that are significantly different tend to have non-intersecting circles. This is controlled by the **Canonical Options > Show Means CL Ellipses** option.
- The directions of the variables in the canonical space is shown by labeled rays emanating from the grand mean. This is controlled by the **Canonical Options > Show Biplot Rays** option. You can drag the center of the biplot rays to other places in the graph.
- The option **Show Normal 50% Contours** shows areas that contain roughly 50% of the points for that group if the assumptions are correct. Under linear discriminant analysis, they are all the same size and shape.

In order to have the points color-coded like the centroid circles, use the **Color Points** option or button. This is equivalent to **Rows > Color or Mark by Column**, coloring by the classification column.

The canonical plot can also be referred to as a *biplot* when both the points and the variable direction rays are shown together, as in Figure 22.5. It is identical to the Centroid plot produced in the **Manova** personality of the Fit Model platform.

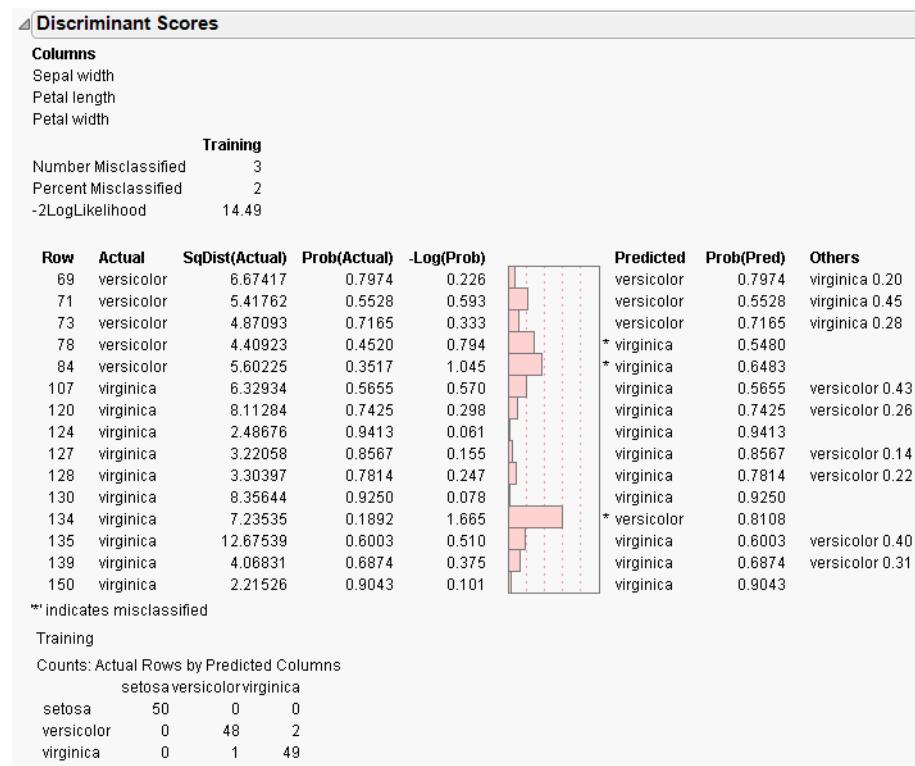
## Discriminant Scores

The scores report shows how well each point is classified. The first five columns of the report represent the actual (observed) data values, showing row numbers, the actual classification, the distance to the mean of that classification, and the associated probability. JMP graphs -Log(Prob) to show the loss in log-likelihood when a point is predicted poorly. When the red bar is large, the point is being poorly predicted.

On the right of the graph, the report shows the category with the highest prediction probability, and, if they exist, other categories that have a predicted probability of over 0.1. An asterisk (\*) indicates which points are misclassified.

In Figure 22.6, the option **Show Interesting Rows Only** option is set so that only those rows that have fitted probabilities between 0.05 and 0.95 or are misclassified are shown.

**Figure 22.6** Show Interesting Rows Only



The Counts report appears at the bottom of Figure 22.6. The counts are zero off the diagonal if everything is classified correctly.

## Commands and Options

The following commands are available from the platform popup menu.

**Stepwise Variable Selection** returns to the stepwise control panel.

**Discriminant Method** chooses the discriminant method. Details are shown in the section “[Discriminant Method](#),” p. 474.

**Linear, Common Covariance** assumes a common covariance.

**Quadratic, Different Covariances** which estimates different covariances.

**Regularized, Compromise Method** a compromise method.

**Score Data** shows or hides the listing of the scores by row in the **Discriminant Scores** portion of the report.

**Score Options** deal with the scoring of the observations and includes the following:

**Show Interesting Rows Only** shows rows that are misclassified and those where  $p>.05$  and  $p<0.95$  for any  $p$ , the attributed probability.

**Show Classification Counts** shows a matrix of actual by predicted counts for each category. When the data are perfectly predicted, the off-diagonal elements are zero. If there are excluded rows, a separate matrix is given for the excluded rows. For more information, see “[Validation](#),” p. 483.

**Show Distances to each group** appends a table to show the squared distance to each group.

**Show Probabilities to each group** appends a table to show the probabilities to each group.

**ROC Curve** appends an ROC curve to the report.

**Select Misclassified Rows** selects the misclassified rows in the original data table.

**Select Uncertain Rows** selects the rows which have uncertain classifications in the data table. When this option is selected, a dialog box appears so you can specify the difference (0.1 is the default) to be marked as uncertain.

**Save Formulas** saves formulas to the data table. The distance formulas are `Dist[0]`, needed in the Mahalanobis distance calculations, and a `Dist[ ]` column for each  $X$ -level's Mahalanobis distance. Probability formulas are `Prob[0]`, the sum of the exponentials of -0.5 times the Mahalanobis distances, and a `Prob[ ]` column for each  $X$ -level's posterior probability of being in that category. The `Pred` column holds the most likely level for each row.

**Canonical Plot** shows or hides the Canonical Plot.

**Canonical Options** provides commands that affect the Canonical Plot and include the following:

**Show Points** shows or hides the points in the plot.

**Show Means CL Ellipses** shows or hides the 95% confidence ellipse of each mean. The ellipses appear as a circle because of scaling. Categories with more observations have smaller circles.

**Show Normal 50% Contours** shows or hides the normal contours which estimate the region where 50% of the level's points lie.

**Show Biplot Rays** shows or hides the biplot rays. These rays indicate the directions of the variables in the canonical space.

**Biplot Ray Position** allows you to specify the position and radius scaling (default = 1) of the biplot rays in the canonical plot.

**Color Points** colors the points based on levels of the  $X$  variable. This statement is equivalent to selecting **Rows > Color or Mark by Column** and selecting the  $X$  variable.

**Show Canonical Details** shows or hides the Canonical details. Details for the `Iris` data set are shown in Figure 22.7. The matrices at the bottom of the report are opened by clicking on the disclosure button beside their name and closed by clicking on the name of the matrix.

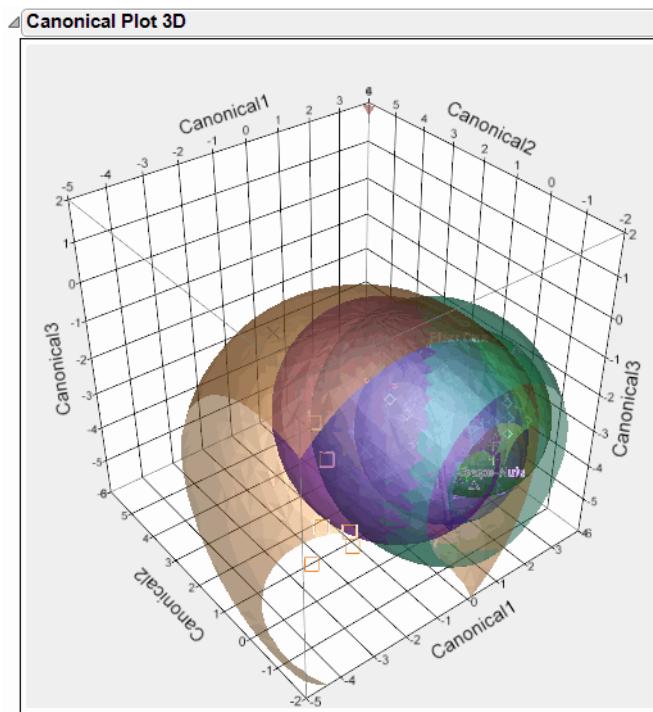
**Figure 22.7** Canonical Details

Canonical					
Eigenvalue	Percent	Cum Percent	Corr		
30.1498095	99.0624	99.0624	0.9838176		
0.28537471	0.9376	100.0000	0.47118653		
Test					
Wilks' Lambda	0.0249755	257.5032	6	290	<.0001*
Pillai's Trace	1.1899138	71.4852	6	292	<.0001*
Hotelling-Lawley	30.435184	733.0249	6	191.57	<.0001*
Roy's Max Root	30.149809	1467.2907	3	146	<.0001*
▶ Within Matrix					
Between Matrix					
Sepal width	Sepal width	Petal length	Petal width		
Sepal width	0.0771764	-0.389385	-0.156005		
Petal length	-0.389385	2.9734884	1.2705714		
Petal width	-0.156005	1.2705714	0.5470295		
Closed matrix					
▶ Scoring Coefficients					

**Save Canonical Scores** creates columns in the data table holding the canonical score for each observation. The new columns are named `Canon[ ]`.

**Save to New Data Table** saves the group means and the biplot rays on the canonical variables, together with the canonical scores to a new data table.

**Canonical 3D Plot** is available only when there are four or more groups (levels). An example of this plot is shown in Figure 22.8. It shows a three-dimensional version of the Canonical Plot and respects other Canonical Options.

**Figure 22.8** Canonical 3D Plot

The example in Figure 22.8 is displayed by:

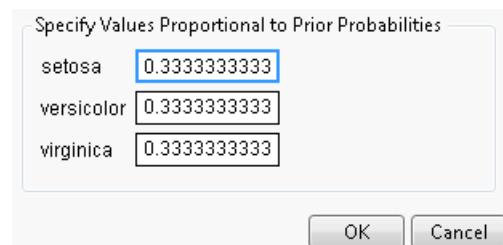
1. Open cereal.jmp.
2. Click on **Analyze > Multivariate Methods > Discriminant**.
3. Specify Calories, Protein, Fat, and Sodium as **Y, Covariates**.
4. Specify Manufacturer as **X, Categories** and click **OK**.
5. Click on the red triangle in the Discriminant Analysis title bar and select **Canonical 3D Plot**.
6. Click inside the plot and rotate the plot until you see the desired effect.

**Specify Priors** lets you specify the prior probabilities for each level of the *X* variable:

**Equal Probabilities** assigns an equal probability to each level.

**Proportional to Occurrence** assigns probabilities to each level according to their frequency in the data.

**Other** brings up a dialog to allow custom specification of the priors, shown in Figure 22.9. By default, each level is assigned equal probabilities.

**Figure 22.9** Specify Prior Probabilities Dialog

**Consider New Levels** is used when you have some points that may not fit any known group, but instead may be from an unscored, new group.

**Save Discrim Matrices** creates a global list (`DiscrimResults`) for use in the JMP scripting language. The list contains a list of `YNames`, a list of `XNames`, a list of `XValues`, a matrix of `YMeans`, and a matrix of `YPartialCov` (covariances). An example from the iris data `DiscrimResults` is

```
{YNames = {"Sepal length", "Sepal width", "Petal length", "Petal Width"},  
 XName = "Species", XValues = {"setosa", "versicolor", "virginica"}, YMeans  
 = [5.005999999999999 3.428000000000001 1.462 0.2459999999999999, 5.936 2.77  
 4.26 1.326, 6.58799999999998 2.974 5.552 2.026], YPartialCov =  
 [0.2650081632653061 0.09272108843537413 0.1675142857142858  
 0.03840136054421769, 0.09272108843537413 0.1153877551020407  
 0.055243537414966 0.03271020408163266, 0.1675142857142858 0.055243537414966  
 0.1851877551020409 0.04266530612244898, 0.03840136054421769  
 0.03271020408163266 0.04266530612244898 0.04188163265306122]}
```

**Get Discrim Matrices**, only available through scripting, obtains the same values as **Save Discrim Matrices**, but returns them to the caller rather than storing them in the data table.

**Show Within Covariances** shows or hides the Covariance Matrix report. The report for this example is shown in Figure 22.10.

**Figure 22.10** Covariance Matrix Report

Covariance Matrices				
Within Cov				
	Sepal length	Sepal width	Petal length	Petal width
Sepal length	0.2650082	0.0927211	0.1675143	0.0384014
Sepal width	0.0927211	0.1153878	0.0552435	0.0327102
Petal length	0.1675143	0.0552435	0.1851878	0.0426653
Petal width	0.0384014	0.0327102	0.0426653	0.0418816
Within Corr				
	Sepal length	Sepal width	Petal length	Petal width
Sepal length	1	0.5302358	0.7561642	0.3645064
Sepal width	0.5302358	1	0.3779162	0.4705346
Petal length	0.7561642	0.3779162	1	0.4844589
Petal width	0.3645064	0.4705346	0.4844589	1

**Show Group Means** shows or hides a table with the means of each variable. Means are shown for each level and for all levels of the  $X$  variable. Figure 22.11 shows the Group Means table for this example.

**Figure 22.11** Group Means Table

Group Means					
Species	Count	Sepal length	Sepal width	Petal length	Petal width
setosa	50	5.0060000	3.4280000	1.4620000	0.2460000
versicolor	50	5.9360000	2.7700000	4.2600000	1.3260000
virginica	50	6.5880000	2.9740000	5.5520000	2.0260000
All	150	5.8433333	3.0573333	3.7580000	1.1993333

## Validation

Validation is the process of using part of a data set to build a model, and using the other part to assess the predictive ability of the model.

- The *training* set is the part that estimates model parameters.
- The *validation* set is the part that validates the predictive ability of the model.

Excluded rows are not included in the training set. When the Show Classification Counts option is chosen on the platform red-triangle menu, results are given for both training and validation (excluded) data. This helps you assess if the discriminant function has any predictive ability on data that was not used to build the function.

In the Discriminant Scores report, there is an indicator for excluded rows.



# Chapter **23**

## Partial Least Squares The PLS Platform

---

The PLS platform fits models using partial least squares (PLS). PLS balances the two objectives of explaining response variation and explaining predictor variation. It is especially useful since it is appropriate in analysis that have more  $x$ -variables than observations.

The number of factors to extract depends on the data. Basing the model on more extracted factors (successive linear combinations of the predictors also called *components* or *latent vectors*) optimally addresses one or both of two goals: explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation. Remember that extracting too many factors can cause over-fitting, (tailoring the model too much to the current data,) to the detriment of future predictions. The PLS platform enables you to choose the number of extracted factors by *cross-validation*, that is, fitting the model to part of the data and minimizing the prediction error for the unfitted part. It also allows for interactive exploration of the extracted factors through the use of the Profiler, which greatly simplifies the identification of the factors' meaning.

# Contents

PLS .....	487
Launch the Platform .....	487
Model Coefficients and PRESS Residuals .....	492
Validation .....	494
Platform Options .....	494
Statistical Details.....	499
Centering and Scaling.....	499
Missing Values.....	499

---

## PLS

Ordinary least squares regression, as implemented in JMP platforms such as Fit Model and Fit Y by X, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS platform have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for new observations when the predictors are highly correlated. The techniques implemented in the PLS platform work by extracting successive linear combinations of the predictors, called *factors* (also called *components* or *latent vectors*), which optimally address the combined goals of explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking factors that explain both response and predictor variation.

The most important point, however, is that PLS works in cases where OLS does not. If the data have more  $x$ -variables than observations, OLS cannot produce estimates, where PLS can.

---

## Launch the Platform

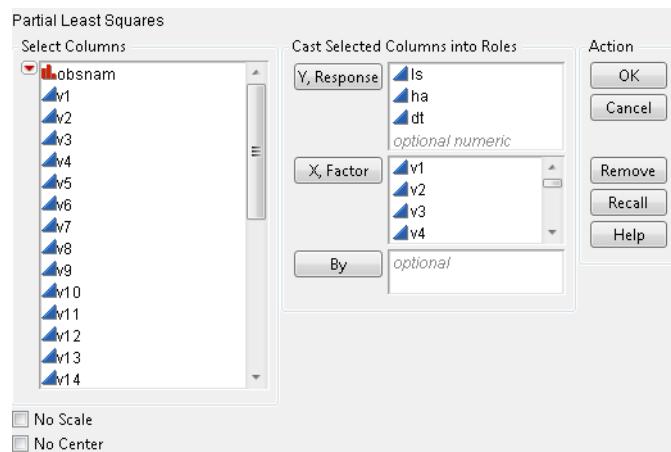
As an example, open the **Baltic.jmp** data table. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose that you are researching pollution in the Baltic Sea, and you would like to use the spectra of samples of sea water to determine the amounts of three compounds present in samples from the Baltic Sea: lignin sulfonate (pulp industry pollution), humic acids (natural forest products), and detergent (optical whitener). Spectrometric calibration is a type of problem in which partial least squares can be very effective. The predictors are the spectra emission intensities at different frequencies in sample spectrum (v1–v27), and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples with known compositions are used. The calibration data consist of 16 samples of known concentrations of lignin sulfonate, humic acids, and detergent, with spectra based on 27 frequencies (or, equivalently, wavelengths).

Launch the PLS platform by selecting **Analyze > Multivariate Methods > PLS**. Assign ls, ha, and dt as **Y, Response** and v1–v27 as **X, Factors**. See launch dialog in Figure 23.1.

Launch the Platform

**Figure 23.1** PLS Launch Dialog

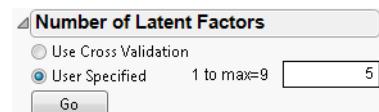


**No Scale** does not scale the data. The data will not be divided by the standard deviation.

**No Center** does not center the data. The mean will not be subtracted from the data.

When you click **OK**, the initial PLS report appears. The PLS model is incomplete until you specify the number of factors to use in the model. The platform suggests a number in the Number of Latent Factors control panel as shown in Figure 23.2.

**Figure 23.2** Number of Latent Factors Control Panel



**Use Cross Validation** uses cross validation to choose the number of factors.

**User Specified** uses the number of latent factors specified in the number entry box. JMP provides the range of factors that can be used. In this example, the user can specify any number between 1 and 9.

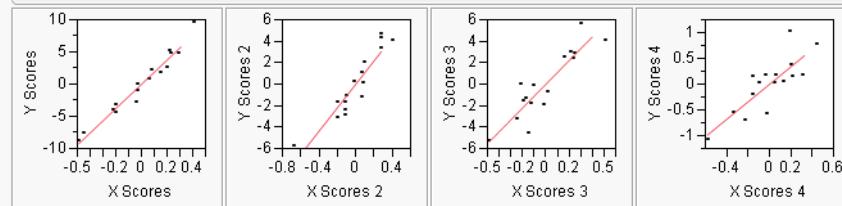
For this example you use cross validation. Select **Use Cross Validation** and click **Go**. The cross validation procedure uses the *one-at-a-time* method. This method fits the model to all data points except one, then you check the capability of the model to predict response for the point omitted. Repeating this for each data point, you then can measure the overall capability of a given form of the model. JMP displays the results using the prediction root mean square error for each number of factors. See Figure 23.3.

**Figure 23.3** Cross-Validation Report**Cross Validation**

Number	Prediction RMSE
1	0.865
2	0.798
3	0.57
4	0.505
5	0.638
6	0.464
7	0.413
8	0.46
9	0.458

The model with the lowest prediction RMSE has seven factors. However, notice that this is almost the same as a model with four factors. A model with four factors is preferable, so the prediction reports are based on the four-model factors.

There are plots of each  $X$  score against its corresponding  $Y$  score, as seen in Figure 23.4. As your analysis continues, this plot updates according to the number of scores you specify.

**Figure 23.4** Scores Plot**X-Y Scores Plots**

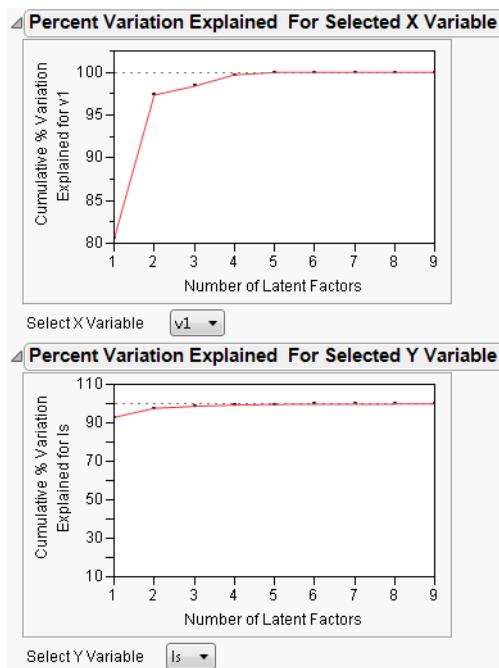
JMP determines the default number of dimensions as the smallest dimension ( $k$ ) which gives a slope ratio (slope of  $Y$  Score –  $X$  Score chart/slope of  $Y$  Score  $k$  –  $X$  Score  $k$  chart) greater than 30. In the Baltic example, five latent vectors are extracted by default. Before selecting cross validation and clicking **Go**, five plots are shown. After running cross validation, the report changes to four plots as shown in Figure 23.4.

For the Baltic data, the percent of variation explained by each latent factor is shown in the Percent Variation Explained report (Figure 23.5). Note that most of the variation in  $X$  is explained by three factors; most of the variation in  $Y$  is explained by six or seven. Note that in this case, “factors” refers to latent variables, not columns in the data table.

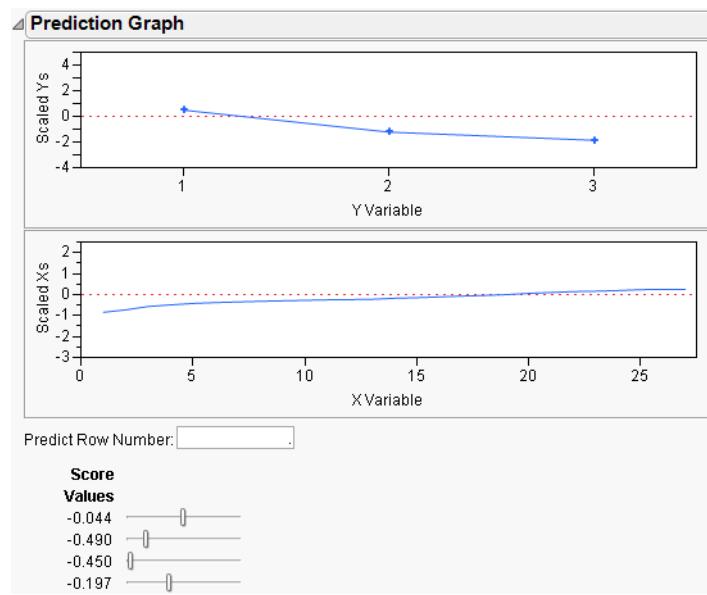
**Figure 23.5** Percent Variation Explained Report

		Percent Variation Explained			
Number	X	20	40	60	80
1	97.46				
2	2.183				
3	0.178				
4	0.12				
5	0.042				
6	0.011				
7	0.002				
8	0.001				
9	0.001				
Cumulative X	Y	20	40	60	80
97.46	41.92				
99.64	24.24				
99.82	24.56				
99.94	3.77				
99.98	0.991				
99.99	2.298				
100	1.158				
100	0.501				
100	0.123				
Cumulative Y					
41.92					
66.16					
90.72					
94.49					
95.48					
97.78					
98.93					
99.43					
99.56					

The report also includes the Percent Variation Explained for Selected X Variables, and the same for the Y variables. See Figure 23.6. A combo box allows you to select different X and Y variables.

**Figure 23.6** Percent Variation Explained for Selected X and Y

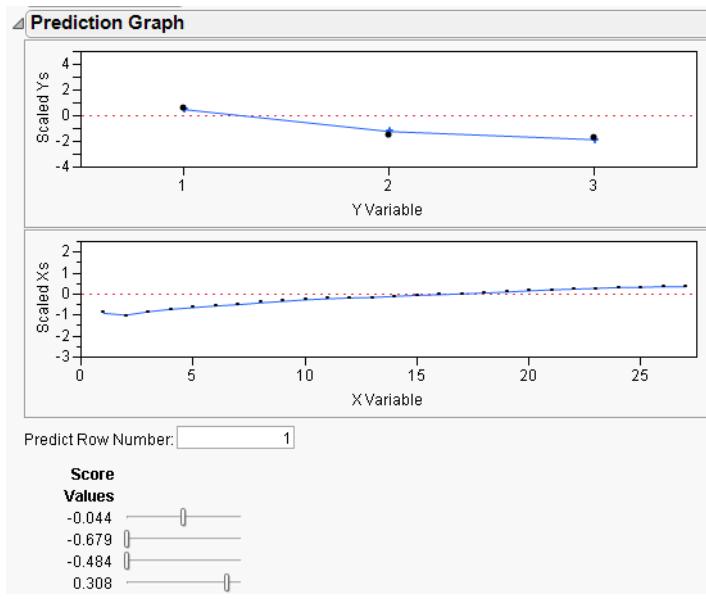
You also see plots of the predicted  $Y$ 's and predicted  $X$ 's, as shown in Figure 23.7. Note that these are the scaled values of  $X$  and  $Y$ , not the original values. Below these plots are a text box to enter a value to predict and four slider boxes to adjust the values of each latent factor.

**Figure 23.7** Prediction Graph and Sliders

Move the sliders to dynamically see each factor's effect on the predictions. For example, moving the fourth slider in this example shows that this latent variable has a great effect on the second Y-variable (humic acid). This factor obviously represents a quantity that affects humic acid far more than lignin sulfonate or detergent. It may represent something that increases production of natural (rather than man-made) pollution.

Similarly, moving the second slider affects the amount of the 11th and 21st spectral values. Therefore, the physical quantity that this represents manifests itself through variations in those frequencies far more than the others, which remain essentially constant.

To predict the values of a specific row number, enter the row number in the text box above the score value sliders. For example, to predict the values of the first row, enter a 1 in the box. Predicted values appear on the graphs of both the *X* and *Y* factors (Figure 23.8).

**Figure 23.8** Prediction of Row 1

## Model Coefficients and PRESS Residuals

Both Model Coefficients reports and the PRESS Residuals report are closed initially. The Model Coefficients appear under the Prediction Graph, and are shown for this example in Figure 23.9.

**Figure 23.9** Model Coefficients for Centered and Scaled Data

Coefficient	ls	ha	dt
Intercept	0	0	0
v1	-0.175741	-0.547917	1.1280725
v2	-0.090812	-0.00915	0.5048483
v3	0.0613942	-0.533759	0.2773249
v4	0.1150536	-0.6908	0.1790379
v5	0.1867027	-0.969957	0.0911426
v6	0.1809188	-0.810213	0.0133251
v7	0.1260524	-0.366151	-0.061916
v8	0.0504472	0.1979488	-0.136548
v9	-0.053093	0.9253383	-0.210766
v10	-0.138579	1.4361835	-0.217802
v11	-0.173922	1.5885429	-0.185641
v12	-0.225789	1.775383	-0.114994
v13	-0.211533	1.6225463	-0.073716
v14	-0.148941	1.1836956	-0.032546
v15	-0.148078	1.1109669	0.007108
v16	-0.053698	0.5403725	0.0158036
v17	-0.008904	0.2756775	0.014912
v18	0.0068338	0.1925671	0.0077631
v19	0.04643	-0.06297	0.0199237
v20	0.1145131	-0.398902	-0.021742
v21	0.1390668	-0.545465	-0.022841
v22	0.1724696	-0.678813	-0.06397
v23	0.1603875	-0.578768	-0.081955
v24	0.1788087	-0.700671	-0.075431
v25	0.2868351	-1.267959	-0.117824
v26	0.288625	-1.260972	-0.129322
v27	0.2830179	-1.237202	-0.123853

**Note:** If you checked the **No Scale** and **No Center** boxes on the launch dialog, the reported model coefficients are for uncentered and unscaled data, and the title changes to Model Coefficients.

The PRESS Residuals (Figure 23.10) appear above the prediction graph.

**Figure 23.10** PRESS Residuals

ls	ha	dt
0.2549951	-0.440404	53.113284
0.3911943	-0.055201	-17.26102
0.0993813	-0.011916	30.923776
-0.106683	0.0833134	-1.968457
-0.049514	-0.030001	-4.005078
-0.082972	0.0128508	-11.2758
-0.102874	0.0595321	-11.64102
0.0906734	-0.081541	22.584916
-0.141984	-0.003241	4.2547091
-0.105445	-0.040669	14.006062
0.1158819	-0.118052	5.9682233
0.3316283	-0.057635	21.608066
-0.088832	0.026665	-9.744934
0.0302446	0.0548227	-28.91227
-0.019586	0.069231	-24.48254
-0.263461	0.1814668	10.738664

---

## Validation

The quality of the model fit to the observed data cannot be used to choose the number of factors to extract; the number of extracted factors must be chosen on the basis of how well the model fits observations not involved in the modeling procedure itself.

One way of choosing the number of extracted factors is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted factors fit the other part of the data (the *validation set*). This is called *validation*.

To do this in JMP, select the validation set and exclude them from the analysis. Then, fit a PLS model and save the prediction formula to the data table. This adds a column to the data table that contains predicted values for both the training set and validation set.

---

## Platform Options

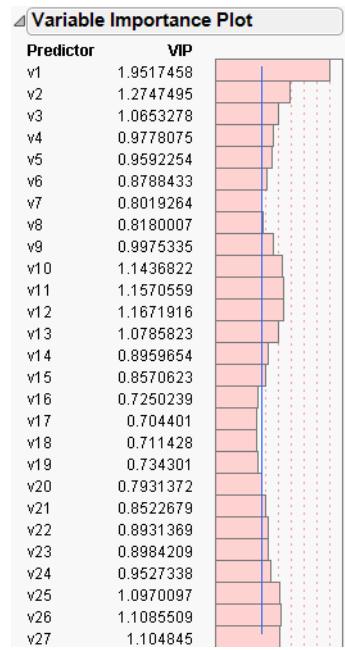
The PLS platform has several options available on the platform popup menu:

**Show Points** shows or hides the points on the  $Y$  scores vs.  $X$  scores scatterplots.

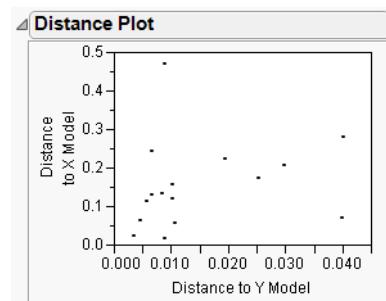
**Show Confidence Lines** shows or hides the confidence lines on the  $Y$  scores vs.  $X$  scores scatterplots.

Note that these should only be used for outlier detection, since the platform doesn't compute traditional confidence lines.

**Variable Importance Plot** shows or hides a Variable Importance Plot. The statistics given are VIP statistics, see Wold (1994). The statistics summarize the contribution a variable makes to the model. If a variable has a small coefficient and a small VIP, then it is a candidate for deletion from the model. Wold in Umetrics (1995) considers a value of 0.8 to be a small VIP. The blue line on the VIP plot is drawn at 0.8.

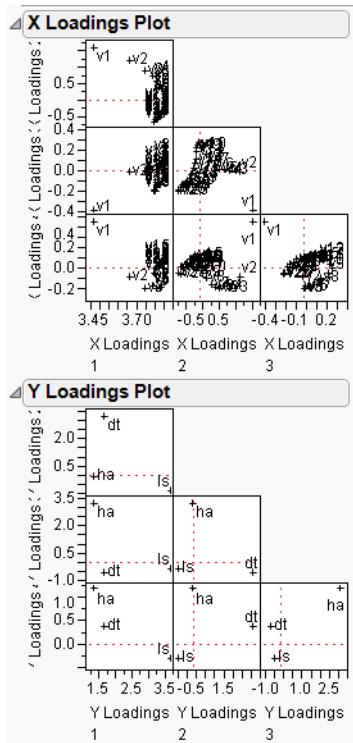
**Figure 23.11** Variables Importance Plot

**Distance Plot** shows or hides a Distance Plot. The *x* axis gives the distance from each point to the model with respect to the *Y*'s. The *y* axis gives the distance from each point to the model with respect to the *X*'s. In a good model, both *X* and *Y* distances are small, and the points are close to the origin (0,0). Use the plot to look for outliers in either direction. If a group of points are separate from the rest, and cluster together, then they might have something in common and could be analyzed separately. To output the plot data to a table, select **Save > Save Scores and Distance** from the platform menu. The table contains two columns, STDXSSE and STDYSSE, which are the residual sums of squares for standardized *X* and *Y*.

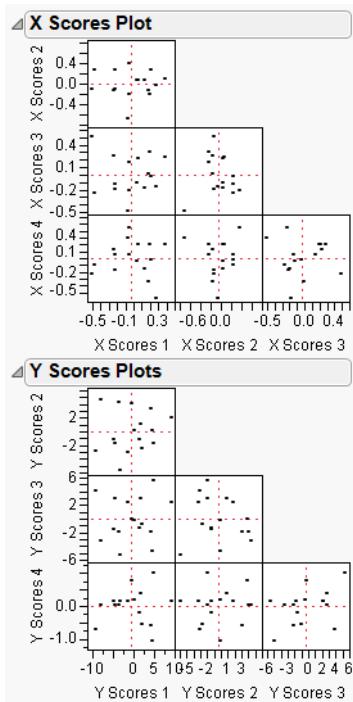
**Figure 23.12** Distance Plot

**Loadings Plots** shows or hides X Loadings Plot and Y Loadings Plot. To output the plot data to a table, select **Save > Save Loadings** from the platform menu.

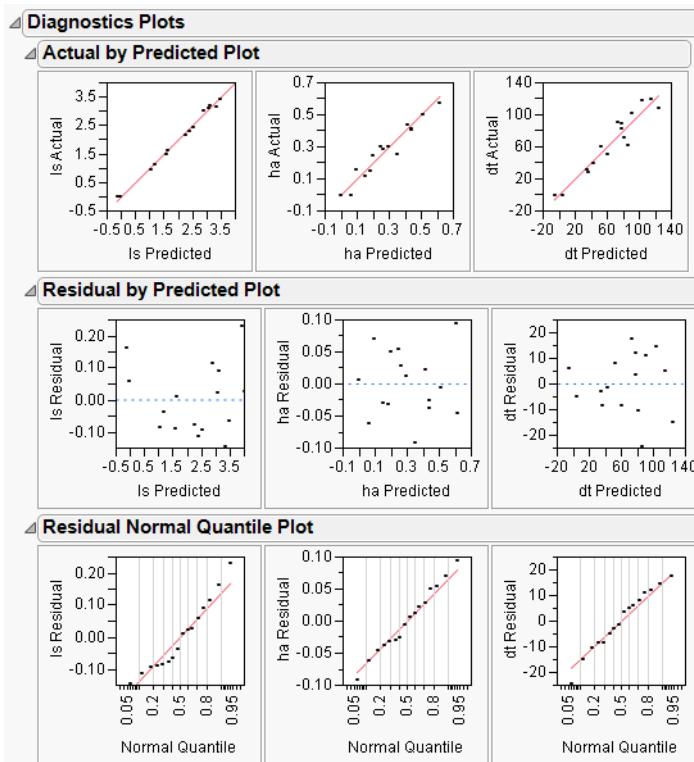
**Figure 23.13** Loadings Plots



**Scores Plots** shows or hides X Scores Plot and Y Scores Plot. To output the plot data to a table, select **Save > Save Scores and Distance** from the platform menu.

**Figure 23.14 Scores Plots**

**Diagnostics Plots** shows various diagnostic plots for each response.

**Figure 23.15** Diagnostic Plots

**Save** is a submenu of commands for saving model results.

**Save Prediction Formula** saves the prediction formula for each  $Y$  to the data table.

**Save Y Residuals** saves the residuals for each  $Y$  to the data table.

**Save X Residuals** saves the residuals for each  $X$  to the data table.

**Save Percent Variation Explained for X Variables** saves to a new table the percent variation explained for each  $X$  and number of factors. This is the information used to create the  $X$  plot of Figure 23.6.

**Save Percent Variation Explained for Y Variables** saves to a new table the percent variation explained for each  $Y$  and number of factors. This is the information used to create the  $Y$  plot of Figure 23.6.

**Save Scores and Distance** saves to a new table the scores (used to create Figure 23.14) and the distances (used to create Figure 23.12). Also creates a Table Property/Script called PLS Scores, and attaches it to the original data table. The script contains JSL matrices of the scores.

**Save Loadings** saves the  $X$  and  $Y$  loadings in two new data tables. This is the information used to create Figure 23.13. Also creates a script called PLS Loadings, and attaches it to the original data table. The script contains JSL matrices of the loadings.

---

## Statistical Details

Partial least squares (PLS) works by extracting one factor at a time. Let  $X = X_0$  be the centered and scaled matrix of predictors and  $Y = Y_0$  the centered and scaled matrix of response values. The PLS method starts with a linear combination  $\mathbf{t} = X_0\mathbf{w}$  of the predictors, where  $\mathbf{t}$  is called a *score vector* and  $\mathbf{w}$  is its associated *weight vector*. The PLS method predicts both  $X_0$  and  $Y_0$  by regression on  $\mathbf{t}$ :

$$\hat{X}_0 = \mathbf{tp}', \text{ where } \mathbf{p}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'X_0$$

$$\hat{Y}_0 = \mathbf{tc}', \text{ where } \mathbf{c}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'Y_0$$

The vectors  $\mathbf{p}$  and  $\mathbf{c}$  are called the *X*- and *Y-loadings*, respectively.

The specific linear combination  $\mathbf{t} = X_0\mathbf{w}$  is the one that has maximum covariance  $\mathbf{t}'\mathbf{u}$  with some response linear combination  $\mathbf{u} = Y_0\mathbf{q}$ . Another characterization is that the *X*- and *Y*-weights  $\mathbf{w}$  and  $\mathbf{q}$  are proportional to the first left and right singular vectors of the covariance matrix  $X_0'Y_0$  or, equivalently, the first eigenvectors of  $X_0'Y_0Y_0'X_0$  and  $Y_0'X_0X_0'Y_0$  respectively.

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing  $X_0$  and  $Y_0$  with the *X*- and *Y*-residuals from the first factor

$$\hat{X}_1 = X_0 - \hat{X}_0$$

$$\hat{Y}_1 = Y_0 - \hat{Y}_0$$

These residuals are also called the *deflated X* and *Y* blocks. The process of extracting a score vector and deflating the data matrices is repeated for as many extracted factors as are desired.

## Centering and Scaling

By default, the predictors and the responses are centered and scaled to have mean 0 and standard deviation 1. Centering the predictors and the responses ensures that the criterion for choosing successive factors is based on how much variation they explain, in either the predictors or the responses or both. Without centering, both the mean variable value and the variation around that mean are involved in selecting factors. Scaling serves to place all predictors and responses on an equal footing relative to their variation in the data. For example, if `Time` and `Temp` are two of the predictors, then scaling says that a change of `std(Time)` in `Time` is roughly equivalent to a change of `std(Temp)` in `Temp`.

## Missing Values

Observations with any missing independent variables are excluded from the analysis, and no predictions are computed for such observations. Observations with no missing independent variables but any missing dependent variables are also excluded from the analysis, but predictions are computed.



# Chapter **24**

## **Item Response Theory**

### The Item Analysis Platform

---

Item Response Theory (IRT) is a method of scoring tests. Although classical test theory methods have been widely used for a century, IRT provides a better and more scientifically-based scoring procedure. Its advantages include:

- scoring tests at the item level, giving insight into the contributions of each item on the total test score
- producing scores of both the test takers and the test items on the same scale
- fitting nonlinear logistic curves, more representative of actual test performance than classical linear statistics.

# Contents

Introduction to Item Response Theory .....	503
Launching the Platform.....	506
Item Analysis Output .....	508
Characteristic Curves.....	508
Information Curves.....	509
Dual Plots .....	509
Platform Commands.....	511
Technical Details.....	512

---

## Introduction to Item Response Theory

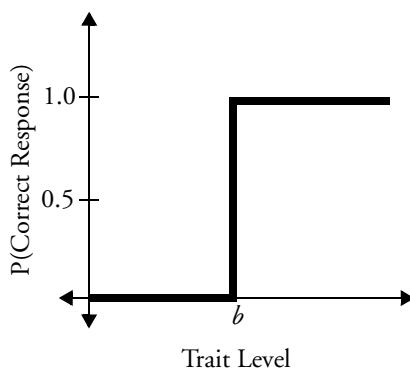
*Psychological measurement* is the process of assigning quantitative values as representations of characteristics of individuals or objects, so-called *psychological constructs*. *Measurement theories* consist of the rules by which those quantitative values are assigned. Item response theory (IRT) is a measurement theory.

IRT utilizes a mathematical function to relate an individual's probability of correctly responding to an item to a trait of that individual. Frequently, this trait is not directly measurable, and is therefore called a *latent trait*.

To see how IRT relates traits to probabilities, first examine a test question that follows the Guttman "perfect scale" as shown in Figure 24.1. The horizontal axis represents the amount of the theoretical trait that the examinee has. The vertical axis represents the probability that the examinee will get the item correct. (A missing value for a test question is treated as an incorrect response.) The curve in Figure 24.1 is called an *item characteristic curve* (ICC).

---

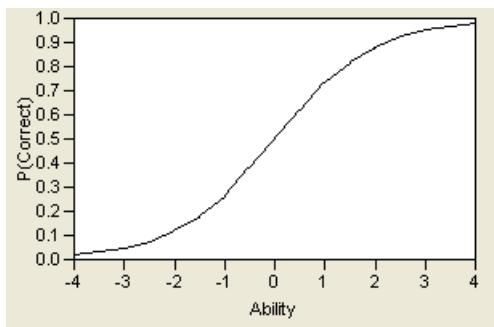
**Figure 24.1** Item characteristic curve of a perfect scale item



---

This figure shows that a person who has ability less than the value  $b$  has a zero percent chance of getting the item correct. A person with trait level higher than  $b$  has a 100 percent chance of getting the item correct.

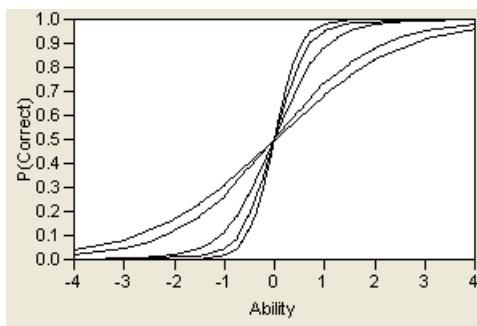
Of course, this is an unrealistic item, but it is illustrative in showing how a trait and a question probability relate to each other. More typical is a curve that allows probabilities that vary from zero to one. A typical curve found empirically is the S-shaped logistic function with a lower asymptote at zero and upper asymptote at one. It is markedly nonlinear. An example curve is shown in Figure 24.2.

**Figure 24.2** Example item response curve

The logistic model is the best choice to model this curve, since it has desirable asymptotic properties, yet is easier to deal with computationally than other proposed models (such as the cumulative normal density function). The model itself is

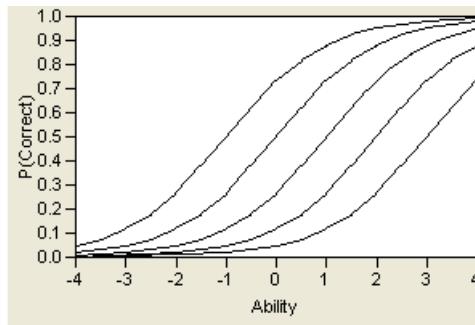
$$P(\theta) = c + \frac{1-c}{1+e^{-(a)(\theta-b)}}$$

In this model, referred to as a *3PL* (three-parameter logistic) model, the variable  $a$  represents the steepness of the curve at its inflection point. Curves with varying values of  $a$  are shown in Figure 24.3. This parameter can be interpreted as a measure of the discrimination of an item—that is, how much more difficult the item is for people with high levels of the trait than for those with low levels of the trait. Very large values of  $a$  make the model practically the step function shown in Figure 24.1. It is generally assumed that an examinee will have a higher probability of getting an item correct as their level of the trait increases. Therefore,  $a$  is assumed to be positive and the ICC is monotonically increasing. Some use this positive-increasing property of the curve as a test of the appropriateness of the item. Items whose curves do not have this shape should be considered as candidates to be dropped from the test.

**Figure 24.3** Logistic model for several values of  $a$ 

Changing the value of  $b$  merely shifts the curve from left to right, as shown in Figure 24.4. It corresponds to the value of  $\theta$  at the point where  $P(\theta)=0.5$ . The parameter  $b$  can therefore be interpreted as item difficulty where (graphically), the more difficult items have their inflection points farther to the right along their  $x$ -coordinate.

**Figure 24.4** Logistic curve for several values of  $b$

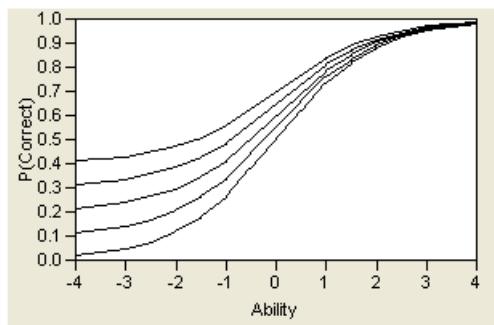


Notice that

$$\lim_{\theta \rightarrow -\infty} P(\theta) = c$$

and therefore  $c$  represents the lower asymptote, which can be non-zero. ICCs for several values of  $c$  are shown graphically in Figure 24.5. The  $c$  parameter is theoretically pleasing, since a person with no ability of the trait may have a non-zero chance of getting an item right. Therefore,  $c$  is sometimes called the *pseudo-guessing parameter*.

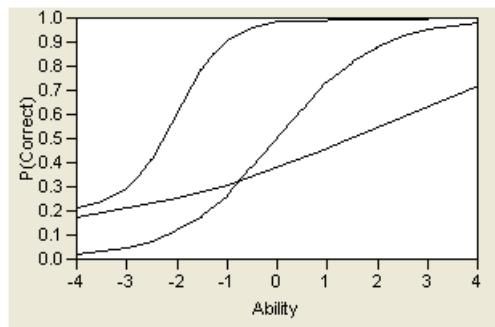
**Figure 24.5** Logistic model for several values of  $c$



By varying these three parameters, a wide variety of probability curves are available for modeling. A sample of three different ICCs is shown in Figure 24.6. Note that the lower asymptote varies, but the upper

asymptote does not. This is because of the assumption that there may be a lower guessing parameter, but as the trait level increases, there is always a theoretical chance of 100% probability of correctly answering the item.

**Figure 24.6** Three item characteristic curves



Note, however, that the 3PL model may be unnecessarily complex for many situations. If, for example, the  $c$  parameter is restricted to be zero (in practice, a reasonable restriction), there are fewer parameters to predict. This model, where only  $a$  and  $b$  parameters are estimated, is called the 2PL model.

Another advantage of the 2PL model (aside from its greater stability than the 3PL) is that  $b$  can be interpreted as the point where an examinee has a 50 percent chance of getting an item correct. This interpretation is not true for 3PL models.

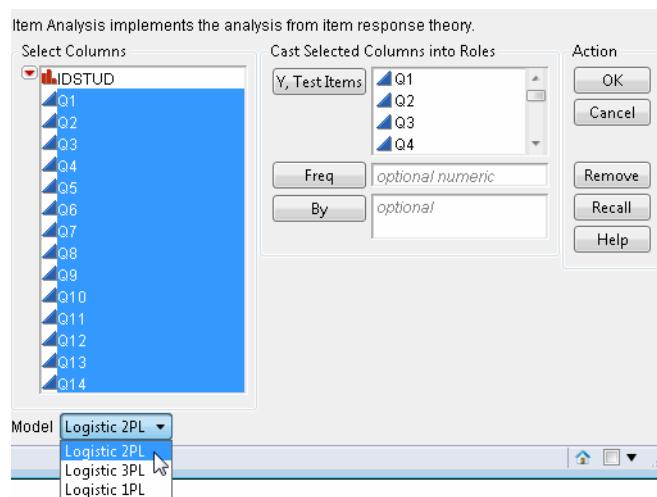
A further restriction can be imposed on the general model when a researcher can assume that test items have equal discriminating power. In these cases, the parameter  $b$  is set equal to zero (essentially dropped from the model), leaving a single parameter to be estimated. This *1PL* model is frequently called the *Rasch model*, named after Danish mathematician Georg Rasch, the developer of the model. The Rasch model is quite elegant, and is the least expensive to use computationally.

**Caution:** You must have a lot of data to produce stable parameter estimates using a 3PL model. 2PL models are frequently sufficient for tests that intuitively deserve a guessing parameter. Therefore, the 2PL model is the default and recommended model.

## Launching the Platform

As an example, open the sample data file `MathScienceTest.jmp`. These data are a subset of the data from the Third International Mathematics and Science Study (TIMSS) conducted in 1996.

To launch the Item Analysis platform, select **Analyze > Multivariate Methods > Item Analysis**. This shows the dialog in Figure 24.7.

**Figure 24.7** Item Analysis Launch Dialog

**Y, Test Items** are the questions from the test instrument.

**Freq** optionally specifies a variable used to specify the number of times each response pattern appears.

**By** performs a separate analysis for each level of the specified variable.

Specify the desired model (1PL, 2PL, or 3PL) by selecting it from the **Model** drop-down menu.

For this example, specify all fourteen continuous questions (Q1, Q2,..., Q14) as **Y, Test Items** and click **OK**. This accepts the default 2PL model.

### Special note on 3PL Models

If you select the 3PL model, a dialog pops up asking for a penalty for the  $c$  parameters (thresholds). This is not asking for the threshold itself. The penalty it requests is similar to the type of penalty parameter that you would see in ridge regression, or in neural networks.

The penalty is on the sample variance of the estimated thresholds, so that large values of the penalty force the estimated thresholds' values to be closer together. This has the effect of speeding up the computations, and reducing the variability of the threshold (at the expense of some bias).

In cases where the items are questions on a multiple choice test where there are the same number of possible responses for each question, there is often reason to believe (*a priori*) that the threshold parameters would be similar across items. For example, if you are analyzing the results of a 20-question multiple choice test where each question had four possible responses, it is reasonable to believe that the guessing, or threshold, parameters would all be near 0.25. So, in some cases, applying a penalty like this has some “physical intuition” to support it, in addition to its computational advantages.

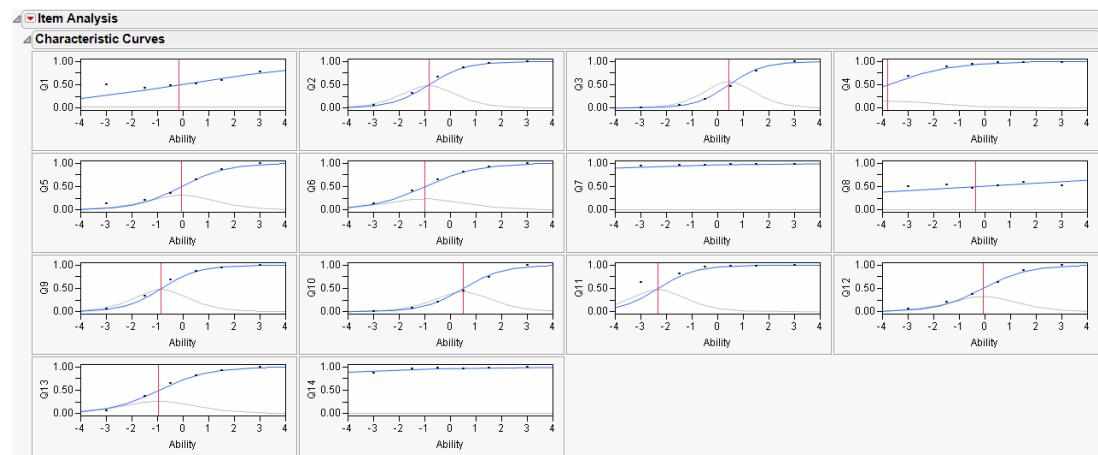
## Item Analysis Output

The following plots appear in Item Analysis reports.

### Characteristic Curves

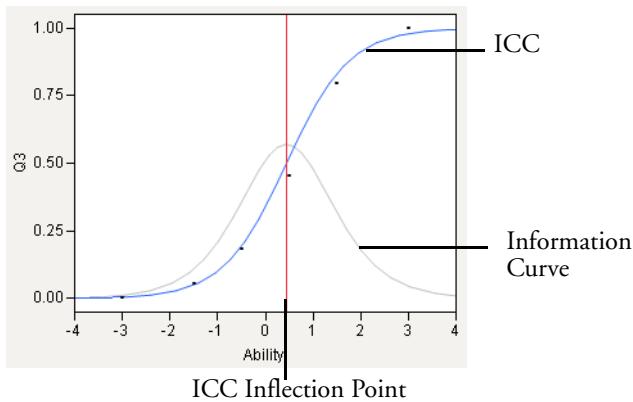
Item characteristic curves for each question appear in the top section of the output. Initially, all curves are shown stacked in a single column. They can be rearranged using the **Number of Plots Across** command, found in the drop down menu of the report title bar. For Figure 24.8, four plots across are displayed.

**Figure 24.8** Component Curves



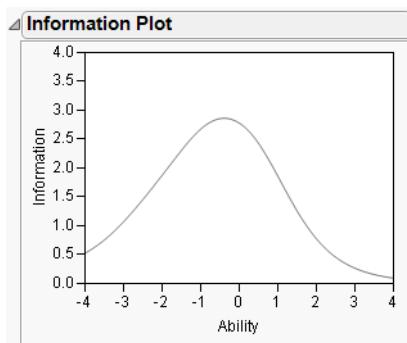
A vertical red line is drawn at the inflection point of each curve. In addition, dots are drawn at the actual proportion correct for each ability level, providing a graphical method of judging goodness-of-fit.

Gray information curves show the amount of information each question contributes to the overall information of the test. The information curve is the slope of the ICC curve, which is maximized at the inflection point.

**Figure 24.9** Elements of the ICC Display

## Information Curves

Questions provide varying levels of information for different ability levels. The gray information curves for each item show the amount of information that each question contributes to the total information of the test. The total information of the test for the entire range of abilities is shown in the Information Plot section of the report (Figure 24.10).

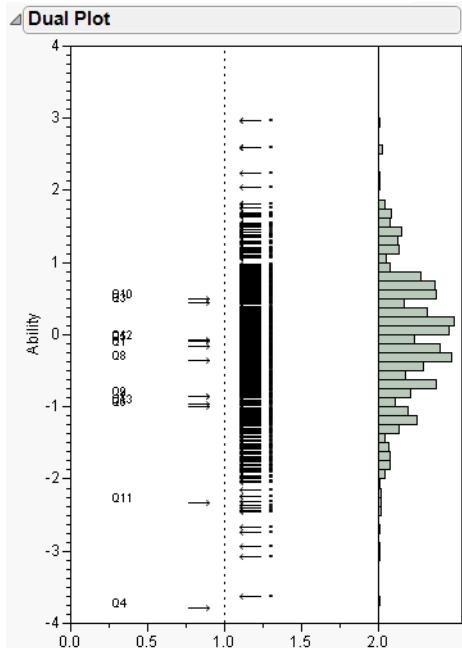
**Figure 24.10** Information Plot

## Dual Plots

The information gained from item difficulty parameters in IRT models can be used to construct an increasing scale of questions, from easiest to hardest, on the same scale as the examinees. This structure gives information on which items are associated with low levels of the trait, and which are associated with high levels of the trait.

JMP shows this correspondence with a *dual plot*. The dual plot for this example is shown in Figure 24.11.

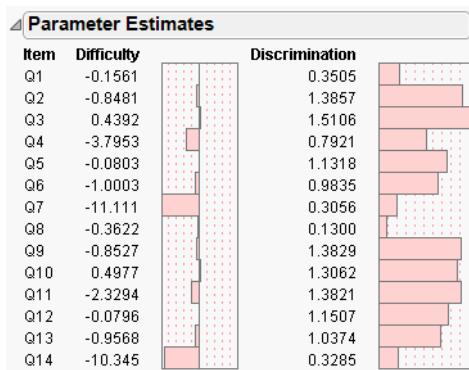
**Figure 24.11** Dual Plot



Questions are plotted to the left of the vertical dotted line, examinees on the right. In addition, a histogram of ability levels is appended to the right side of the plot.

This example shows a wide range of abilities. Q10 is rated as difficult, with an examinee needing to be around half a standard deviation above the mean in order to have a 50% chance of correctly answering the question. Other questions are distributed at lower ability levels, with Q11 and Q4 appearing as easier. There are some questions that are off the displayed scale (Q7 and Q14).

The estimated parameter estimates appear below the Dual Plot, as shown in Figure 24.12.

**Figure 24.12** Parameter Estimates

**Item** identifies the test item.

**Difficulty** is the  $b$  parameter from the model. A histogram of the difficulty parameters is shown beside the difficulty estimates.

**Discrimination** is the  $a$  parameter from the model, shown only for 2PL and 3PL models. A histogram of the discrimination parameters is shown beside the discrimination estimates.

**Threshold** is the  $c$  parameter from the model, shown only for 3PL models.

## Platform Commands

The following three commands are available from the drop-down menu on the title bar of the report.

**Number of Plots Across** brings up a dialog to specify how many plots should be grouped together on a single line. Initially, plots are stacked one-across. [Figure 24.8 “Component Curves,” p. 508](#) shows four plots across.

**Save Ability Formula** creates a new column in the data table containing a formula for calculating ability levels. Since the ability levels are stored as a formula, you can add rows to the data table and have them scored using the stored ability estimates. In addition, you can run several models and store several estimates of ability in the same data table.

The ability is computed using the **IRT Ability** function. The function has the following form

**IRT Ability** ( $Q1, Q2, \dots, Qn, [a1, a2, \dots, an, b1, b2, \dots, bn, c1, c2, \dots, cn]$ );

where  $Q1, Q2, \dots, Qn$  are columns from the data table containing items,  $a1, a2, \dots, an$  are the corresponding discrimination parameters,  $b1, b2, \dots, bn$  are the corresponding difficulty parameters for the items, and  $c1, c2, \dots, cn$  are the corresponding threshold parameters. Note that the parameters are entered as a matrix, surrounded by square brackets.

**Script** accesses the standard script menu for saving and re-computing an analysis.

---

## Technical Details

Note that  $P(\theta)$  does not necessarily represent the probability of a positive response from a *particular* individual. It is certainly feasible that an examinee might definitely select an incorrect answer, or that an examinee may know an answer for sure, based on the prior experiences and knowledge of the examinee, apart from the trait level. It is more correct to think of  $P(\theta)$  as the probability of response for a set of individuals with ability level  $\theta$ . Said another way, if a large group of individuals with equal trait levels answered the item,  $P(\theta)$  predicts the proportion that would answer the item correctly. This implies that IRT models are item-invariant; theoretically, they would have the same parameters regardless of the group tested.

An assumption of these IRT models is that the underlying trait is unidimensional. That is to say, there is a single underlying trait that the questions measure which can be theoretically measured on a continuum. This continuum is the horizontal axis in the plots of the curves. If there are several traits being measured, each of which have complex interactions with each other, then these unidimensional models are not appropriate.

# Chapter **25**

## Plotting Surfaces

### The Surface Plot Platform

---

The Surface Plot platform functions both as a separate platform and as an option in model fitting platforms. Up to four dependent surfaces can be displayed in the same plot. The dependent variables section, below the plot, has four rows that correspond to the four surfaces. Depending on what you choose to view (sheets, points, isosurfaces, or density grids) and whether you supply a formula variable, different options appear in the dependent variables section.

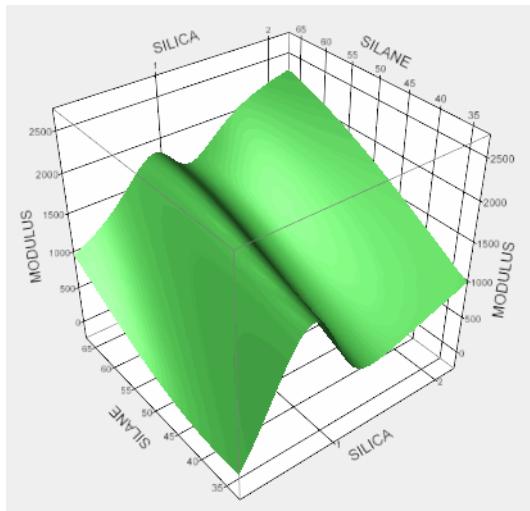
Surface Plot is built using the 3-D scene commands from JMP's scripting language. Complete documentation of the OpenGL-style scene commands is found in the *JMP Scripting Guide*.

In this platform, you can

- use the mouse to drag the surface to a new position;
- Right-click on the surface to change the background color or show the virtual ArcBall (which helps position the surface);
- enable hardware acceleration, which may increase performance if it is supported on your system;
- drag lights to different positions, assign them colors, and turn them on and off.

---

**Figure 25.1** Example of a Surface Plot



# Contents

Surface Plots . . . . .	515
Launching the Platform. . . . .	515
Plotting a Single Mathematical Function. . . . .	516
Plotting Points Only . . . . .	517
Plotting a Formula from a Column . . . . .	518
Isosurfaces . . . . .	520
The Surface Plot Control Panel . . . . .	522
Appearance Controls . . . . .	523
Independent Variables . . . . .	523
Dependent Variables . . . . .	524
Plot Controls and Settings . . . . .	525
Rotate . . . . .	525
Axis Settings . . . . .	526
Lights . . . . .	527
Sheet or Surface Properties . . . . .	527
Other Properties and Commands . . . . .	528
Keyboard Shortcuts . . . . .	529

## Surface Plots

The Surface Plot platform is used to plot points and surfaces in three dimensions.

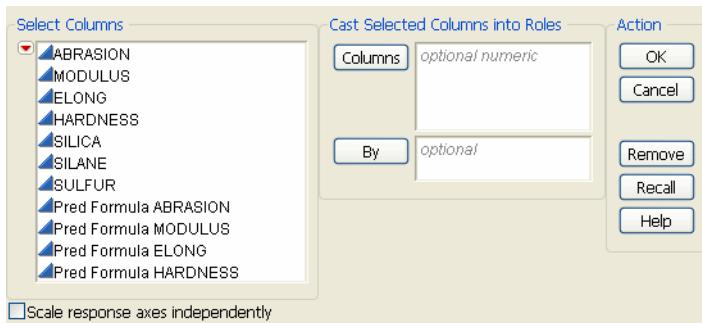
Surface plots are available as a separate platform (**Graph > Surface Plot**) and as options in many reports (known as the **Surface Profiler**). Its functionality is similar wherever it appears.

The plots can be of points or surfaces. When the surface plot is used as a separate platform (that is, not as a profiler), the points are linked to the data table—they are clickable, respond to the brush tool, and reflect the colors and markers assigned in the data table. Surfaces can be defined by a mathematical equation, or through a set of points defining a polygonal surface. These surfaces can be displayed smoothly or as a mesh, with or without contour lines. Labels, axes, and lighting are fully customizable.

## Launching the Platform

To launch the platform, select **Surface Plot** from the **Graph** menu. If there is a data table open, this displays the dialog in Figure 25.2. If you do not want to use data table for drawing surface plots, click **OK** without specifying columns. You are presented with the default surface plot shown in Figure 25.3. If there is no data table open, the default surface plot shown in Figure 25.3 appears immediately, with no launch window.

**Figure 25.2** Surface Plot Launch Dialog



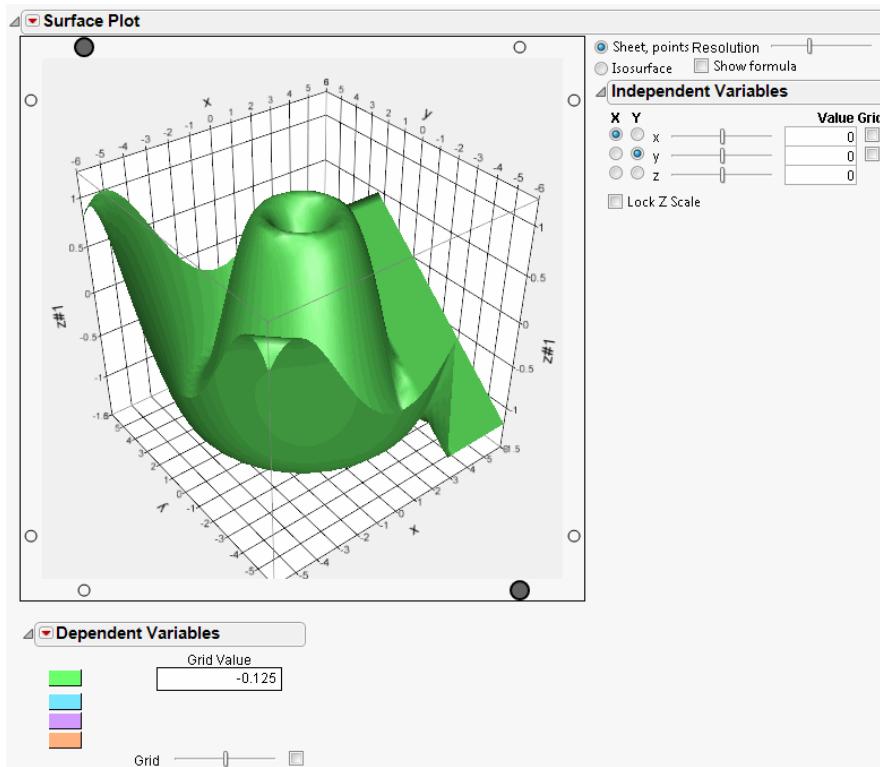
Specify the columns you want to plot by putting them in the **Columns** role. Only numeric variables can be assigned to the **Columns** role. Variables in the **By** role produce a separate surface plot for each level of the **By** variable.

When selected, the **Scale response axes independently** option gives a separate scale to each response on the plot. When not selected, the axis scale for all responses is the same as the scale for the first item entered in the **Columns** role.

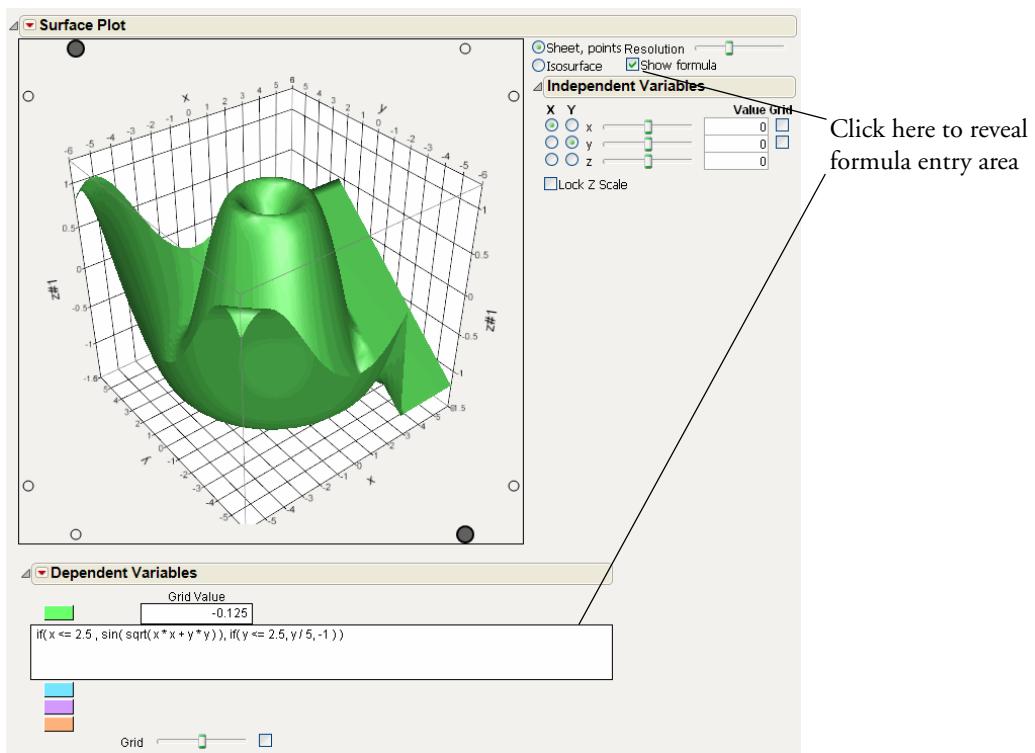
## Plotting a Single Mathematical Function

To produce the graph of a mathematical function without any data points, do not fill in any of the roles on the launch dialog. Simply click **OK** to get a default plot, as shown in Figure 25.3.

**Figure 25.3** Default Surface Plot



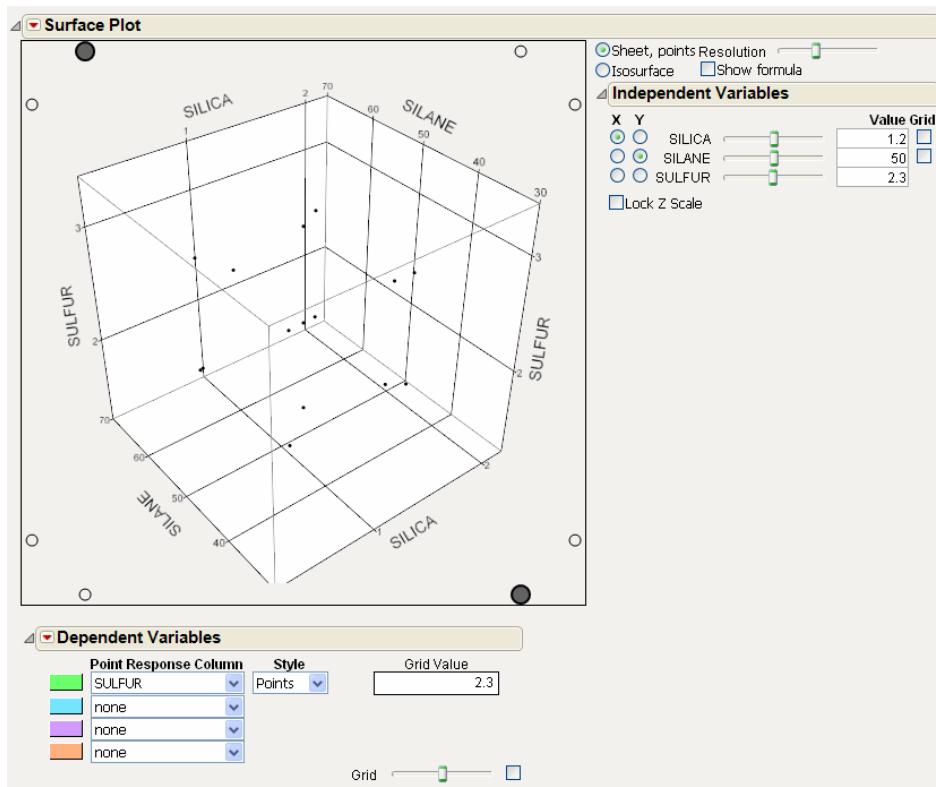
Select the **Show Formula** checkbox to show the formula space.



The default function shows in the box. To plot your own function, enter it in this box.

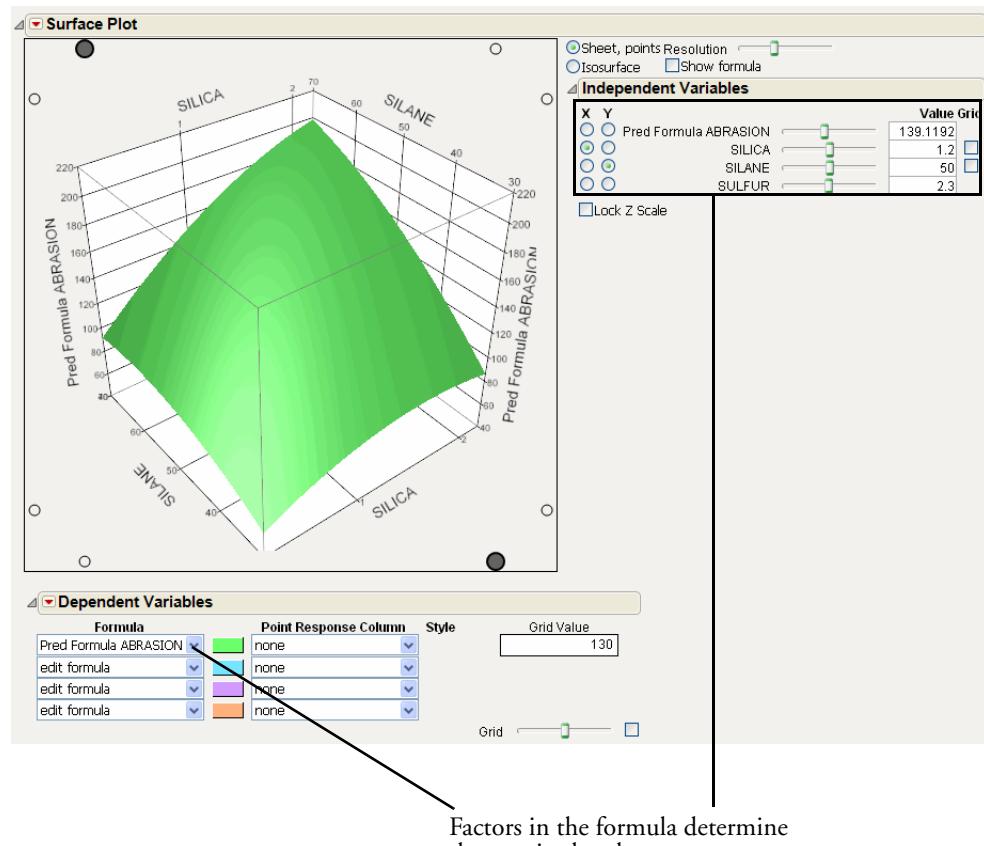
## Plotting Points Only

To produce a 3-D scatterplot of points, place the  $x$ -,  $y$ -, and  $z$ -columns in the **Columns** box. For example, using the Tiretread.jmp data, first select **Rows > Clear Row States**. Then select **Graph > Surface Plot**. Assign Silica, Silane, and Sulfur to the **Columns** role. Click **OK**.

**Figure 25.4** 3-D Scatterplot Launch and Results

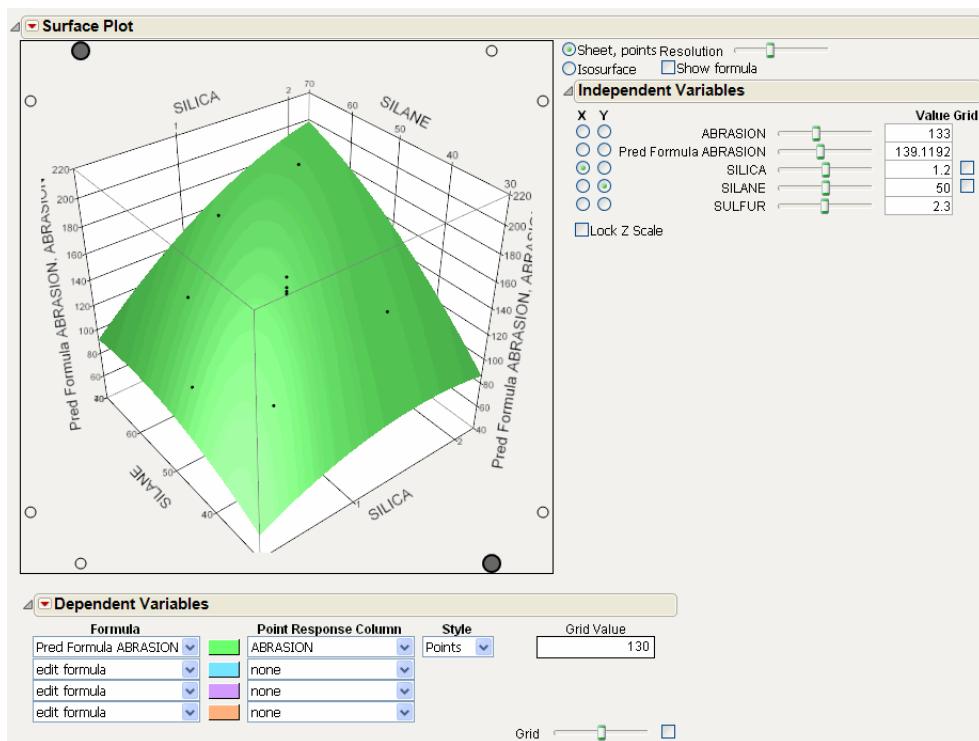
## Plotting a Formula from a Column

To plot a formula (*i.e.* a formula from a column in the data table), place the column in the **Columns** box. For example, use the **Tiretread.jmp** data table and select **Graph > Surface Plot**. Assign **Pred Formula ABRASION** to the **Columns** role. Click **OK**. You do not have to specify the factors for the plot, since the platform automatically extracts them from the formula.

**Figure 25.5** Formula Launch and Output

Note that this only plots the prediction surface. To plot the actual values in addition to the formula, assign the ABRASION and Pred Formula ABRASION to the **Columns** role.

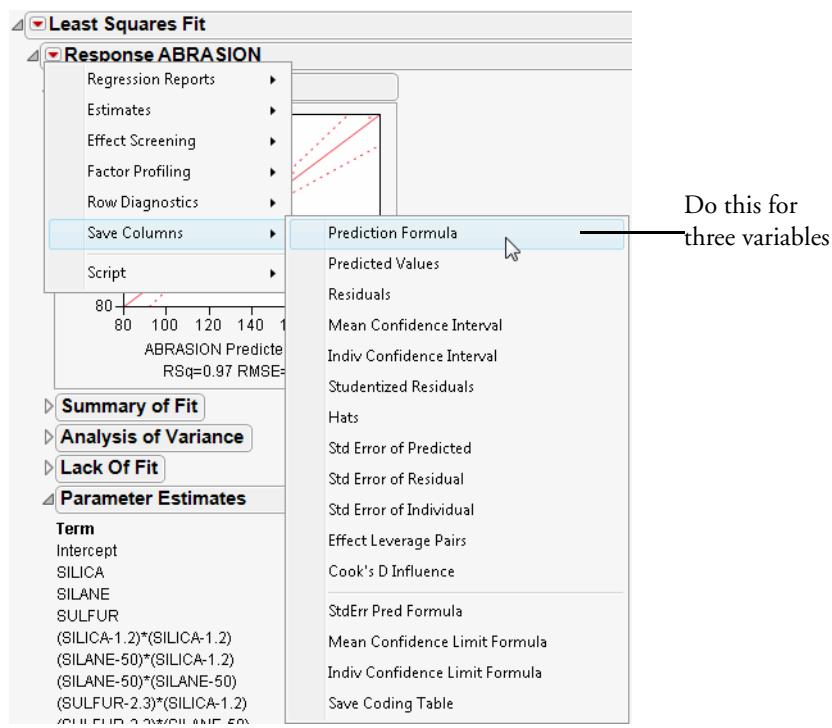
Figure 25.6 shows the launch dialog and the completed results.

**Figure 25.6** Formula and Data Points Launch and Output

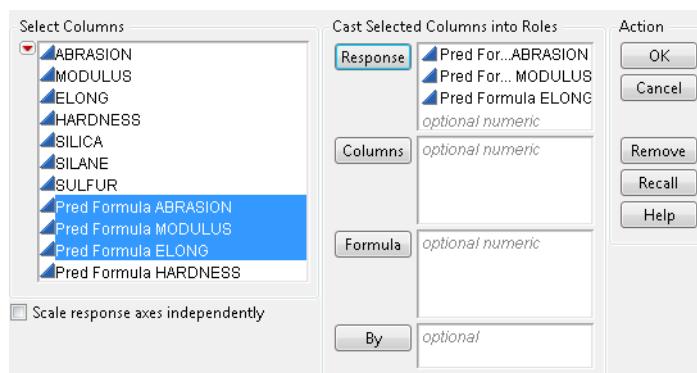
## Isosurfaces

Isosurfaces are the 3-D analogy to a 2-D contour plot. An isosurface requires a formula with three independent variables. The Resolution slider determines the  $n \times n \times n$  cube of points that the formula is evaluated over. The Value slider in the Dependent Variable section picks the isosurface (that is, the contour level) value.

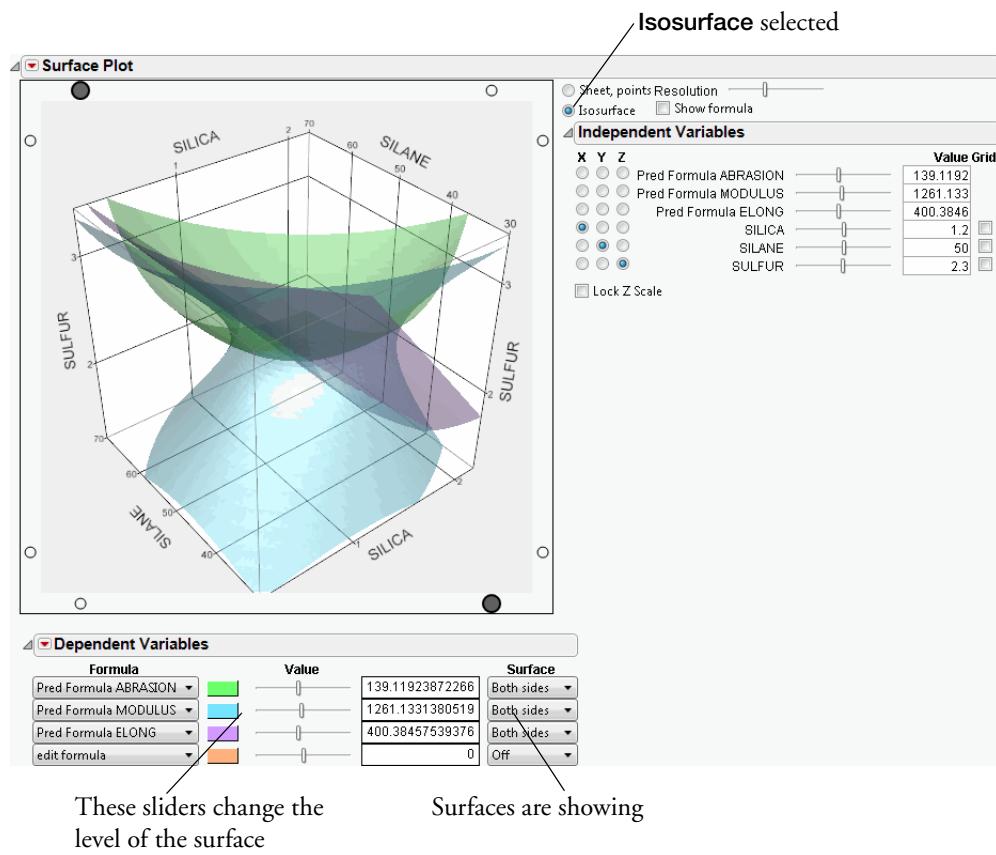
As an example, open the Tiretread.jmp data table and run the RSM for 4 Responses script. This produces a response surface model with dependent variables ABRASION, MODULUS, ELONG, and HARDNESS. Since isosurfaces require formulas, select **Save Columns > Prediction Formula** for ABRASION, MODULUS, and ELONG.



Now launch Surface Plot and designate the three prediction columns as those to be plotted.



When the report appears, select the Isosurface radio button. Under the **Dependent Variables** outline node, select **Both Sides** for all three variables.

**Figure 25.7** Isosurface of Three Variables

For the tire tread data, one might set the hardness at a fixed minimum setting and the elongation at a fixed maximum setting, then use the slider for modulus to see which values of modulus are inside the limits set by the other two surfaces.

## The Surface Plot Control Panel

The drop-down menu in the main Surface Plot title bar has three entries.

**Control Panel** shows or hides the Control Panel.

**Scale response axes independently** scales response axes independently. See explanation of Figure 25.2 on p. 515.

**Script** the standard JMP Script menu.

The Control Panel consists of the following groups of options.

## Appearance Controls

The first set of controls allows you to specify the overall appearance of the surface plot.



**Sheet, points** is the setting for displaying sheets, points, and lines.

**Isosurface** changes the display to show isosurfaces, described in “[Isosurfaces](#),” p. 520.

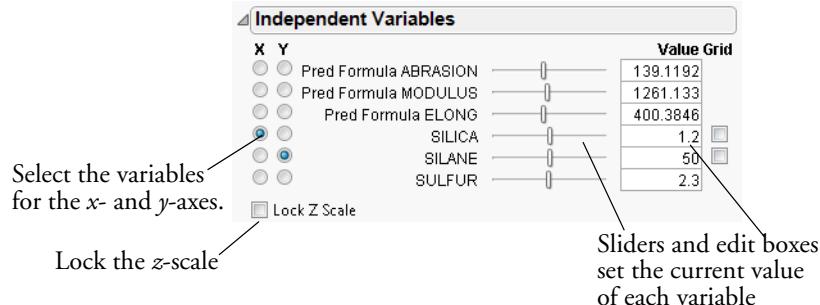
**Show formula** shows the formula edit box, allowing you to enter a formula to be plotted.

The **Resolution** slider affects how many points are evaluated for a formula. Too coarse a resolution means a function with a sharp change might not be represented very well, but setting the resolution high makes evaluating and displaying the surface slower.

## Independent Variables

The independent variables controls are displayed in Figure 25.8.

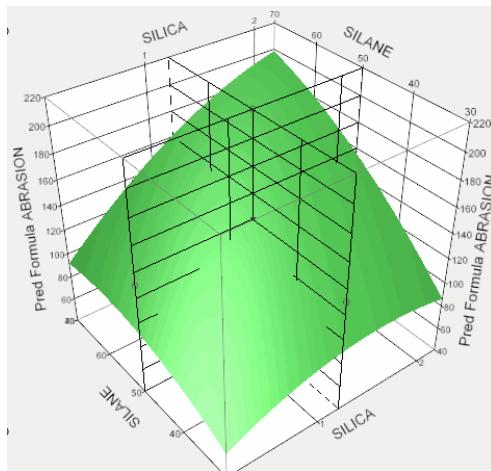
**Figure 25.8** Variables Controls



When there are more than two independent variables, you can select which two are displayed on the *x*- and *y*-axes using the radio buttons in this panel. The sliders and text boxes set the current values of each variable, which is most important for the variables that are not displayed on the axes. In essence, the plot shows the three-dimensional slice of the surface at the value shown in the text box. Move the slider to see different slices.

**Lock Z Scale** locks the *z*-axis to its current values. This is useful when moving the sliders that are not on an axis.

**Grid** check boxes activate a grid that is parallel to each axis. The sliders allow you to adjust the placement of each grid. The resolution of each grid can be controlled by adjusting axis settings. As an example, Figure 25.9 shows a surface with the *X* and *Y* grids activated.

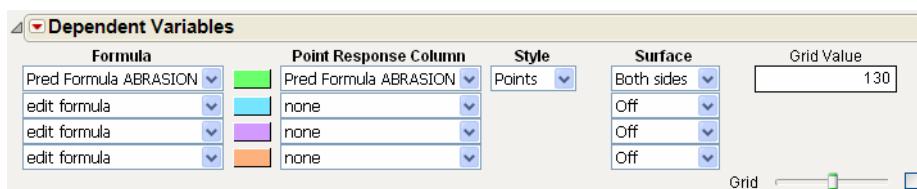
**Figure 25.9** Activated X and Y Grids

## Dependent Variables

The dependent variables controls change depending on whether you have selected **Sheet, points** or **Isosurface** in the Appearance Controls.

### Controls for Sheet, Points

The Dependent Variables controls are shown in Figure 25.10 with its default menus.

**Figure 25.10** Dependent Variable Controls

**Formula** lets you pick the formula(s) to be displayed in the plot as surfaces.

**Point Response Column** lets you pick the column that holds values to be plotted as points.

**Style** menus appear after you have selected a **Point Response Column**. The style menu lets you choose how those points are displayed, as **Points**, **Needles**, a **Mesh**, **Surface**, or **Off** (not at all).

**Points** shows individual points, which change according to the color and marker settings of the row in the data table. **Needles** draws lines from the *x-y* plane to the points, or, if a surface is also plotted,

connects the surface to the points. **Mesh** connects the points into a triangular mesh. **Surface** overlays a smooth, reflective surface on the points.

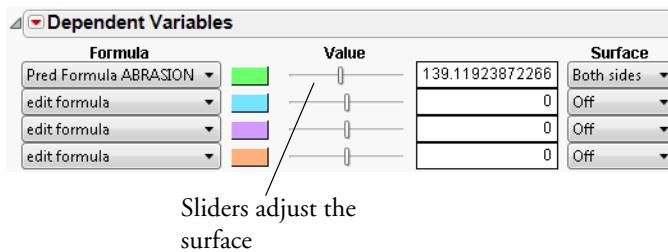
**Surface** enables you to show or hide the top or bottom of a surface. If **Above only** or **Below only** is selected, the opposite side of the surface is darkened.

**Grid** slider and checkbox activate a grid for the dependent variable. Use the slider to adjust the value where the grid is drawn, or type the value into the Grid Value box above the slider.

### Controls for Isosurface

Most of the controls for **Isosurface** are identical to those of **Sheet, points**. Figure 25.11 shows the default controls, illustrating the slightly different presentation.

**Figure 25.11** Dependent Variable Controls for Isosurfaces



### Dependent Variable Menu Options

There are several options for the Dependent Variable, accessed through the popup menu.

**Formula** reveals or hides the Formula drop-down list.

**Surface** reveals or hides the Surface drop-down list.

**Points** reveals or hides the Point Response Column drop-down list.

**Response Grid** reveals or hides the Grid controls.

---

## Plot Controls and Settings

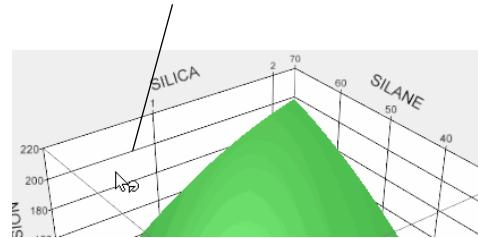
### Rotate

The plot can be rotated in any direction by dragging it. Place the cursor anywhere on the plot where the cursor has a circular arrow attached to it, then click and drag the plot to rotate.

---

**Figure 25.12** Example of Cursor Position for Rotating Plot

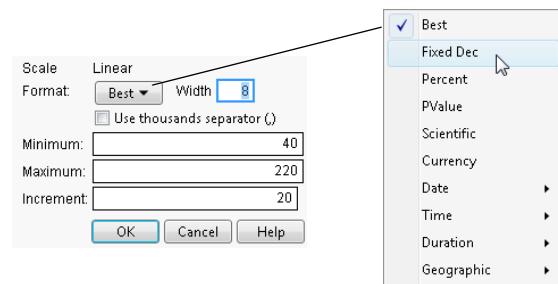
Ready to rotate plot when  
cursor looks like this



The Up, Down, Left, and Right arrow keys can also be used to rotate the plot.

## Axis Settings

Double-click on an axis to reveal the axis control dialog shown below. The dialog allows you to change the Minimum, Maximum, Increment, and tick mark label Format.




---

Like other JMP graphs, the axes can be adjusted, stretched, and compressed using the grabber tool. Place the cursor over an axis to change it to the grabber.

**Figure 25.13** Grabber Tools

Place grabber in the middle of axis to adjust.



Place grabber at the end of axis to stretch or compress.



Notice the orientation of grabber changes from vertical to horizontal

## Lights

By default, the plot has lights shining on it. There are eight control knobs on the plot for changing the position and color of the lights. This is useful for highlighting different parts of a plot and creating contrast. Four of the eight knobs are show below.

**Figure 25.14** Control Knobs for Lights

Activated light

Right-click in border to Reset lights

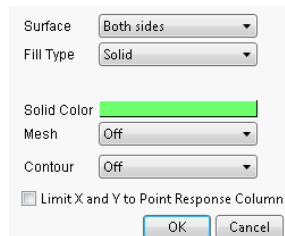
Light is off. Right-click on knob to turn on



Right-click on a knob to turn that light on or off. More lights turned on brighten a plot, and fewer lights darken it. Drag a knob to change the position of a light. Change the color of a light by right-clicking on the knob. The default color is white.

## Sheet or Surface Properties

If you are plotting a **Sheet**, **points**, right-click on the sheet and select **Sheet Properties** to reveal a dialog for changing the sheet properties.

**Figure 25.15** Sheet Properties Dialog

**Surface** lets you fill the surface with a solid color (the default), or a gradient of colors according to another dependent variable. If you choose a solid color, you may also change the color by clicking on the **Solid Color** bar. If you choose a gradient, options appear for changing the gradient.

**Fill Type** allows you to color the surface using a solid color, or continuous or discrete gradients. If a gradient is chosen, the Show Legend option appears when you right click on the surface.

**Mesh** allows you to turn on or off a surface mesh, for either the *X* or *Y* directions or both. If turned on, the **Mesh Color** option is revealed allowing you to change the color.

**Contour** allows you to turn on or off a contour grid, either above, below, or on the surface. If turned on, the **Contour Color** option is revealed allowing you to change the color.

**Limit X and Y to Point Response Column** limits the range of the plot to the range of the data in the Point Response Column, if one is activated. If checked, this essentially restricts the plot from extrapolating outside the range of the data in the Point Response Column.

The equivalent JSL command for this option is `Clip Sheet( Boolean )`. You can send this message to a particular response column by appending the number of the response column. For example, `Clip Sheet2( 1 )` limits the range of the plot to the range of the data of the second response column. See the Object Scripting Index for an example.

If you are plotting a **Isosurface**, right-click on the surface and select **Surface Properties** to reveal a similar dialog. You can modify the surface color, opacity, and toggle a mesh.

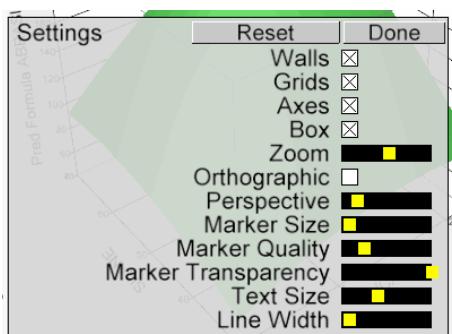
## Other Properties and Commands

Right-click anywhere in the plot area to reveal the following options:

**Show Legend** shows a legend when the surface is colored using gradients.

**Reset** resets the plot to the original viewpoint. Changes in wall and background color are not affected.

**Settings** opens a dialog for changing many plot settings.



**Hide Lights Border** shows or hides lighting controls.

**Wall Color** allows you to change the plot wall color.

**Background Color** allows you to change the plot background color.

**Rows** allows you to change row colors or markers, and also exclude, hide, and label points. These options only have affect if points are on the plot.

**Use Hardware Acceleration** provides for faster rendering of the display. For example, if the plot redraws slowly when rotating, this option may help it to redraw faster.

**Show ArcBall** provides options for using the ArcBall. The ArcBall is a sphere drawn around the plot to help visualize the directions of rotation.

## Keyboard Shortcuts

The following keyboard shortcuts can be used to manipulate the surface plot. To get the plot back to the original viewpoint, right-click on the plot and select **Reset**.

**Table 25.1** Surface Plot Keyboard Shortcuts

Key	Function
left, right, up, and down arrows	spin
Home, End	diagonally spin
Enter (Return)	toggles ArcBall appearance
Delete	roll counterclockwise
Control	boost spin speed 10X
Shift	allows continual spinning



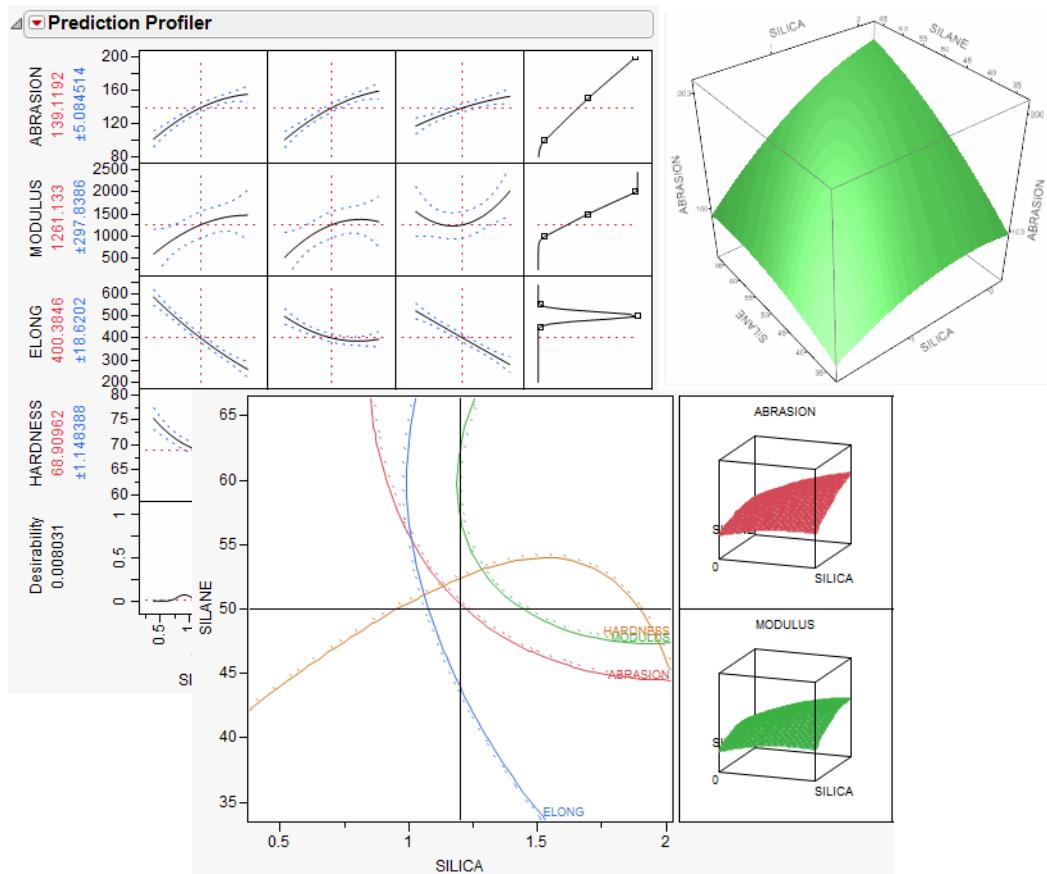
# Chapter 26

## Profiling

### Response Surface Visualization, Optimization, and Simulation

Profiling is an approach to visualizing response surfaces by seeing what would happen if you change just one or two factors at a time. Essentially, a profile is a cross-section view.

**Figure 26.1** Examples of Profilers



# Contents

Introduction to Profiling .....	533
The Profiler .....	535
Interpreting the Profiles .....	536
Profiler Options .....	540
Desirability Profiling and Optimization.....	545
Special Profiler Topics .....	549
Propagation of Error Bars .....	549
Customized Desirability Functions .....	550
Mixture Designs .....	552
Expanding Intermediate Formulas.....	553
Linear Constraints.....	553
Contour Profiler .....	555
Mixture Profiler.....	557
Surface Profiler .....	569
The Custom Profiler .....	569
The Simulator .....	570
Specifying Factors .....	571
Specifying the Response.....	573
Run the Simulation .....	573
The Simulator Menu.....	574
Using Specification Limits.....	574
Simulating General Formulas .....	576
The Defect Profiler .....	579
Noise Factors (Robust Engineering).....	593
Profiling Models Stored in Excel .....	599
The Excel Model.....	600
Using the JMP Add-In for Profiling .....	601
Using the Excel Profiler From JMP .....	604
Fit Group .....	604
Statistical Details.....	605

## Introduction to Profiling

It is easy to visualize a response surface with one input factor  $X$  and one output factor  $Y$ . It becomes harder as more factors and responses are added. The profilers provide a number of highly interactive cross-sectional views of any response surface.

Desirability profiling and optimization features are available to help find good factor settings and produce desirable responses.

Simulation and defect profiling features are available for when you need to make responses that are robust and high-quality when the factors have variation.

## Profiling Features in JMP

There are five profiler facilities in JMP, accessible from a number of fitting platforms and the main menu. They are used to profile data column formulas.

**Table 26.1** Profiler Features Summary

	Description	Features
<b>Profiler</b>	Shows vertical slices across each factor, holding other factors at current values	Desirability, Optimization, Simulator, Propagation of Error
<b>Contour Profiler</b>	Horizontal slices show contour lines for two factors at a time	Simulator
<b>Surface Profiler</b>	3-D plots of responses for 2 factors at a time, or a contour surface plot for 3 factors at a time	Surface Visualization
<b>Custom Profiler</b>	A non-graphical profiler and numerical optimizer	General Optimization, Simulator
<b>Mixture Profiler</b>	A contour profiler for mixture factors	

Profiler availability is shown in Table 15.2.

**Table 26.2** Where to Find JMP Profilers

Location	Profiler	Contour Profiler	Surface Profiler	Mixture Profiler	Custom Profiler
Graph Menu (as a Platform)	Yes	Yes	Yes	Yes	Yes
Fit Model: Least Squares	Yes	Yes	Yes	Yes	
Fit Model: Logistic	Yes				
Fit Model: LogVariance	Yes	Yes	Yes		

**Table 26.2** Where to Find JMP Profilers (*Continued*)

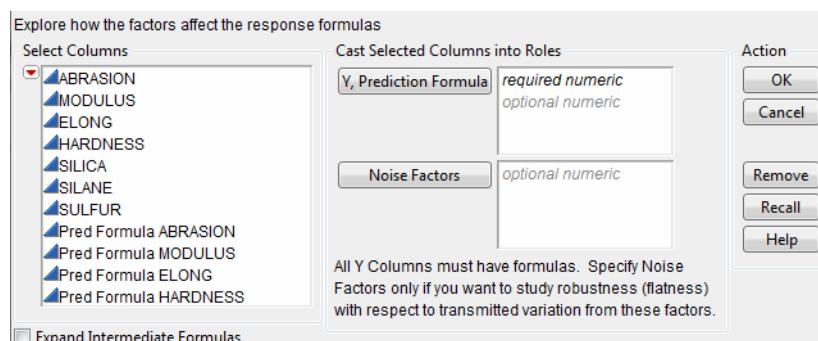
Location	Profiler	Contour Profiler	Surface Profiler	Mixture Profiler	Custom Profiler
Fit Model: Generalized Linear	Yes	Yes	Yes		
Nonlinear: Factors and Response	Yes	Yes	Yes		
Nonlinear: Parameters and SSE	Yes	Yes	Yes		
Neural Net	Yes	Yes	Yes		
Gaussian Process	Yes	Yes	Yes		
Custom Design Prediction Variance	Yes		Yes		
Life Distribution	Yes				
Fit Life by X	Yes		Yes		
Choice	Yes				

**Note:** In this chapter, we use the following terms interchangeably:

- factor, input variable,  $X$  column, independent variable, setting
- response, output variable,  $Y$  column, dependent variable, outcome

The **Profiler** (with a capital P) is one of several profilers (lowercase p). Sometimes, to distinguish the **Profiler** from other profilers, we call it the **Prediction Profiler**.

When the profiler is invoked as (main menu) a platform, rather than through a fitting platform, you provide columns with formulas as the **Y, Prediction Formula** columns. These formulas could have been saved from the fitting platforms.

**Figure 26.2** Profiler Launch Window

The columns referenced in the formulas become the *X* columns (unless the column is also a *Y*).

**Y, Prediction Formula** are the response columns containing formulas.

**Noise Factors** are only used in special cases for modeling derivatives. Details are in “[Noise Factors \(Robust Engineering\)](#),” p. 593.

**Expand Intermediate Formulas** tells JMP that if an ingredient column to a formula is a column that itself has a formula, to substitute the inner formula, as long as it refers to other columns. To prevent an ingredient column from expanding, use the **Other** column property with a name of “Expand Formula” and a value of 0.

The **Surface Plot** platform is discussed in a separate chapter. The **Surface Profiler** is very similar to the **Surface Plot** platform, except **Surface Plot** has more modes of operation. Neither the **Surface Plot** platform nor the **Surface Profiler** have some of the capabilities common to other profilers.

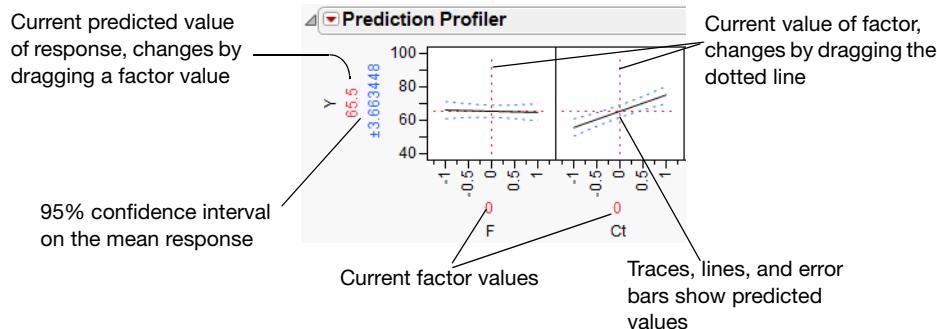
---

## The Profiler

The **Profiler** displays profile traces (see Figure 26.3) for each *X* variable. A *profile trace* is the predicted response as one variable is changed while the others are held constant at the current values. The **Profiler** recomputes the profiles and predicted responses (in real time) as you vary the value of an *X* variable.

- The vertical dotted line for each *X* variable shows its *current value* or *current setting*. If the variable is nominal, the *x*-axis identifies categories. See “[Interpreting the Profiles](#),” p. 536, for more details.  
For each *X* variable, the value above the factor name is its current value. You change the current value by clicking in the graph or by dragging the dotted line where you want the new current value to be.
- The horizontal dotted line shows the *current predicted value* of each *Y* variable for the current values of the *X* variables.
- The black lines within the plots show how the predicted value changes when you change the current value of an individual *X* variable. In fitting platforms, the 95% confidence interval for the predicted values is shown by a dotted blue curve surrounding the prediction trace (for continuous variables) or the context of an error bar (for categorical variables).

The **Profiler** is a way of changing one variable at a time and looking at the effect on the predicted response.

**Figure 26.3** Illustration of Traces

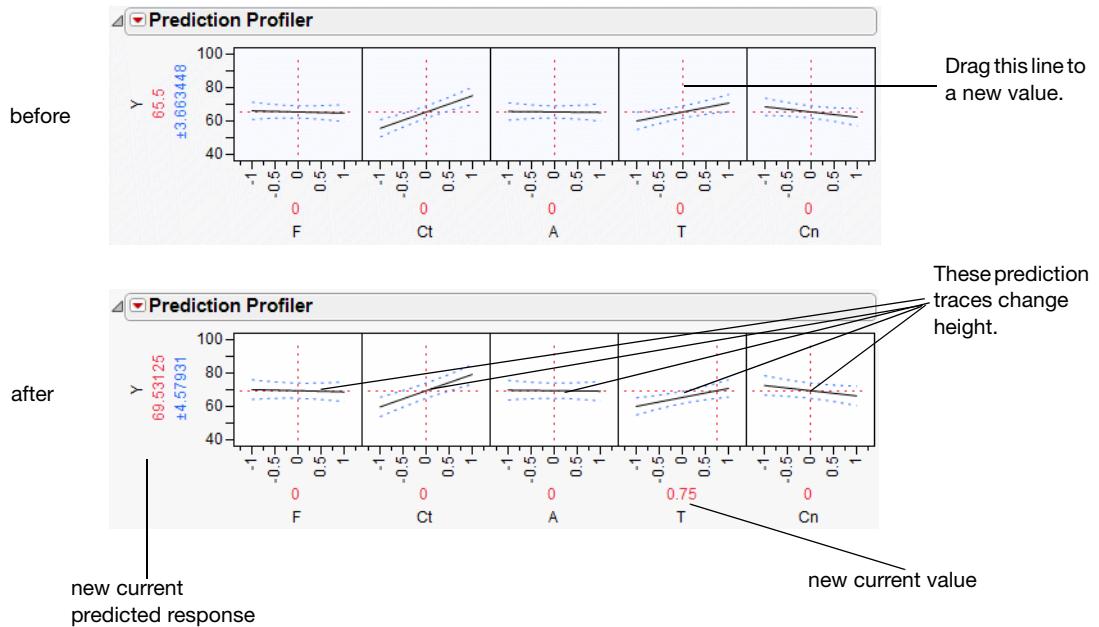
The **Profiler** in some situations computes confidence intervals for each profiled column. If you have saved both a standard error formula and a prediction formula for the same column, the **Profiler** offers to use the standard errors to produce the confidence intervals rather than profiling them as a separate column.

## Interpreting the Profiles

The illustration in Figure 26.4 describes how to use the components of the **Profiler**. There are several important points to note when interpreting a prediction profile:

- The importance of a factor can be assessed to some extent by the steepness of the prediction trace. If the model has curvature terms (such as squared terms), then the traces may be curved.
- If you change a factor's value, then its prediction trace is not affected, but the prediction traces of all the other factors can change. The  $Y$  response line must cross the intersection points of the prediction traces with their current value lines.

**Note:** If there are interaction effects or cross-product effects in the model, the prediction traces can shift their slope and curvature as you change current values of other terms. That is what interaction is all about. If there are no interaction effects, the traces only change in height, not slope or shape.

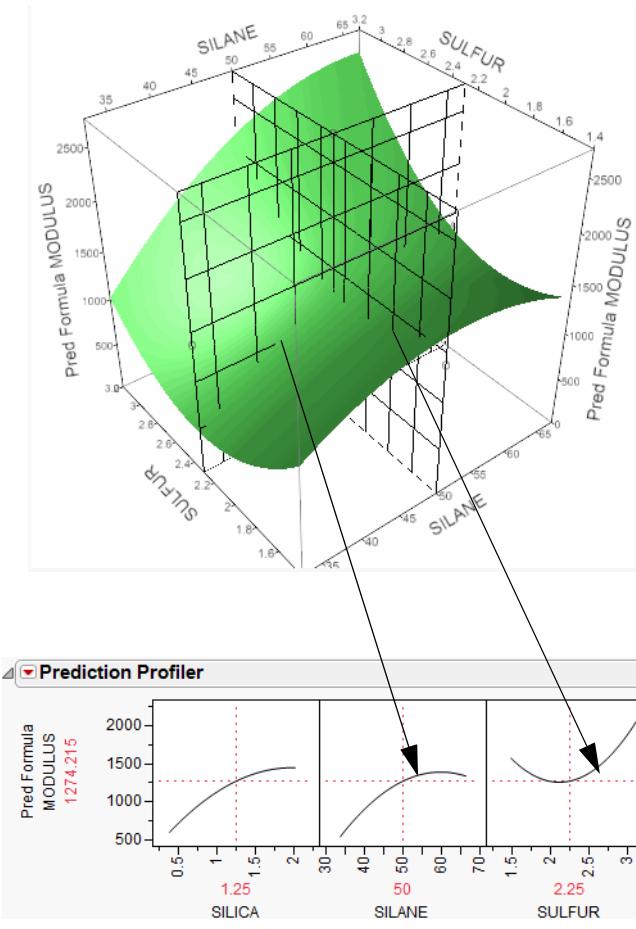
**Figure 26.4** Changing one Factor From 0 to 0.75

Prediction profiles are especially useful in multiple-response models to help judge which factor values can optimize a complex set of criteria.

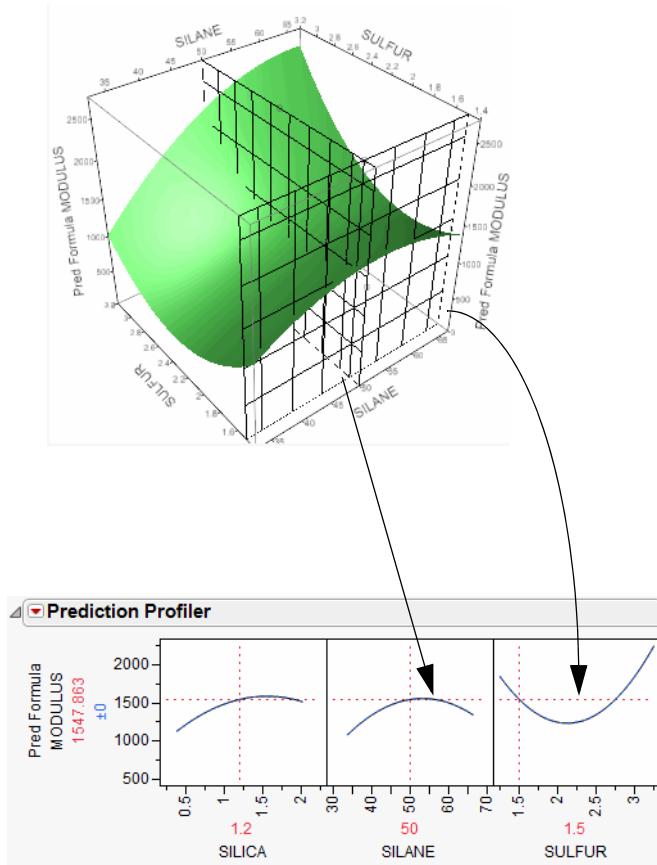
Click on a graph or drag the current value line right or left to change the factor's current value. The response values change as shown by a horizontal reference line in the body of the graph. When you click in the vertical axis, the  $X$  value at that point displays. Double-click in an axis to bring up a dialog that changes its settings.

### Thinking about Profiling as Cross-Sectioning

In the following example using *Tiretread.jmp*, look at the response surface of the expression for MODULUS as a function of SULFUR and SILANE (holding SILICA constant). Now look at how a grid that cuts across SILANE at the SULFUR value of 2.3. Note how the slice intersects the surface. If you transfer that down below, it becomes the profile for SILANE. Similarly, note the grid across SULFUR at the SILANE value of 50. The intersection when transferred down to the SULFUR graph becomes the profile for SULFUR.

**Figure 26.5** Profiler as a Cross-Section

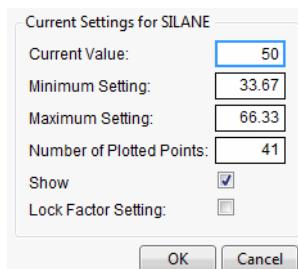
Now consider changing the current value of **SULFUR** from 2.3 down to 1.5.

**Figure 26.6** Profiler as a Cross-Section

In the **Profiler**, note the new value just moves along the same curve for SULFUR, the SULFUR curve itself doesn't change. But the profile for SILANE is now taken at a different cut for SULFUR, and is a little higher and reaches its peak in the different place, closer to the current SILANE value of 50.

### Setting or Locking a Factor's Values

If you Alt-click (Option-click on the Macintosh) in a graph, a dialog prompts you to enter specific settings for the factor.

**Figure 26.7** Continuous Factor Settings Dialog

For continuous variables, you can specify

**Current Value** is the value used to calculate displayed values in the profiler, equivalent to the red vertical line in the graph.

**Minimum Setting** is the minimum value of the factor's axis.

**Maximum Value** is the maximum value of the factor's axis.

**Number of Plotted Points** specifies the number of points used in plotting the factor's prediction traces.

**Show** is a checkbox that allows you to show or hide the factor in the profiler.

**Lock Factor Setting** locks the value of the factor at its current setting.

## Profiler Options

The popup menu on the **Profiler** title bar has the following options:

**Profiler** shows or hides the Profiler.

**Contour Profiler** shows or hides the Contour Profiler.

**Custom Profiler** shows or hides the Custom Profiler.

**Mixture Profiler** shows or hides the Mixture Profiler.

**Surface Profiler** shows or hides the Surface Profiler.

**Save As Flash (SWF)** allows you to save the Profiler (with reduced functionality) as an Adobe Flash file, which can be imported into presentation and web applications. An HTML page can be saved for viewing the Profiler in a browser. The **Save as Flash (SWF)** command is not available for categorical responses. For information on importing the Flash version of the Profiler into Microsoft PowerPoint, go to [www.jmp.com/support/swfhelp/en/powerpoint.shtml](http://www.jmp.com/support/swfhelp/en/powerpoint.shtml).

The Profiler will accept any JMP function, but the Flash Profiler only accepts the following functions: Add, Subtract, Multiply, Divide, Minus, Power, Root, Sqrt, Abs, Floor, Ceiling, Min, Max, Equal, Not Equal, Greater, Less, GreaterEqual, LesserEqual, Or, And, Not, Exp, Log, Log10, Sine, Cosine, Tangent, SinH, CosH, TanH, ArcSine, ArcCosine, ArcTangent, ArcSineH, ArcCosH, ArcTanH, Squish, If, Match, Choose.

**Note:** Some platforms create column formulas that are not supported by the Save As Flash option.

**Show Formulas** opens a JSL window showing all formulas being profiled.

**Formulas for OPTMODEL** Creates code for the OPTMODEL SAS procedure.

**Script** has a submenu of commands available to all platforms that let you redo the analysis, or save the JSL commands for the analysis to a window or file.

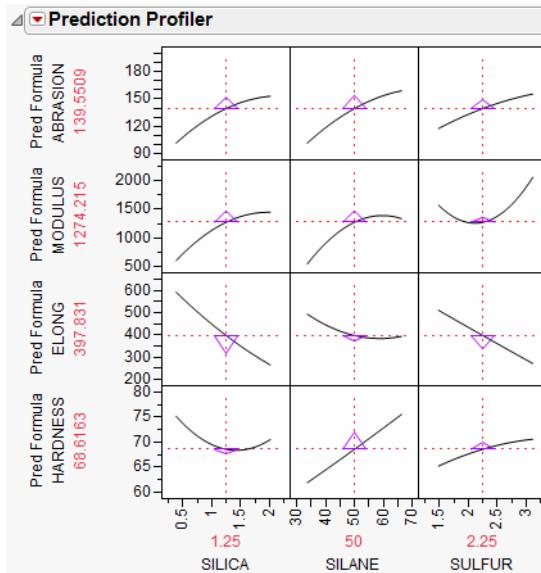
The popup menu on the **Prediction Profiler** title bar has the following options:

**Prop of Error Bars** appears under certain situations. See “[Propagation of Error Bars](#),” p. 549.

**Confidence Intervals** shows or hides the confidence intervals. The intervals are drawn by bars for categorical factors, and curves for continuous factors. These are available only when the profiler is used inside certain fitting platforms.

**Sensitivity Indicator** shows or hides a purple triangle whose height and direction correspond to the value of the partial derivative of the profile function at its current value. This is useful in large profiles to be able to quickly spot the sensitive cells.

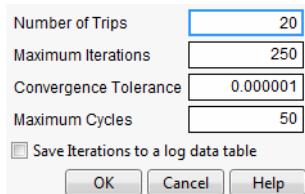
**Figure 26.8** Sensitivity Indicators



**Desirability Functions** shows or hides the desirability functions, as illustrated by Figure 26.17 and Figure 26.18. Desirability is discussed in “[Desirability Profiling and Optimization](#),” p. 545.

**Maximize Desirability** sets the current factor values to maximize the desirability functions.

**Maximization Options** allows you to refine the optimization settings through a dialog.

**Figure 26.9** Maximization Options Window

**Maximize for Each Grid Point** can only be used if one or more factors are locked. The ranges of the locked factors are divided into a grid, and the desirability is maximized at each grid point. This is useful if the model you are profiling has categorical factors; then the optimal condition can be found for each combination of the categorical factors.

**Save Desirabilities** saves the three desirability function settings for each response, and the associated desirability values, as a Response Limits column property in the data table. These correspond to the coordinates of the handles in the desirability plots.

**Set Desirabilities** brings up a dialog where specific desirability values can be set.

**Figure 26.10** Response Grid Window

Maximize		
Pred Formula ABRASION Values	Desirability	
High:	200	0.9819
Middle:	145	0.5
Low:	90	0.066
Importance:	1	

OK    Cancel    Help

**Save Desirability Formula** creates a column in the data table with a formula for Desirability. The formula uses the fitting formula when it can, or the response variables when it can't access the fitting formula.

**Reset Factor Grid** displays a dialog for each value allowing you to enter specific values for a factor's current settings. See the section "Setting or Locking a Factor's Values," p. 539 for details on these dialog boxes.

**Factor Settings** is a submenu that consists of the following options:

**Remember Settings** adds an outline node to the report that accumulates the values of the current settings each time the **Remember Settings** command is invoked. Each remembered setting is preceded by a radio button that is used to reset to those settings.

**Set To Data in Row** assigns the values of a data table row to the Profiler.

**Copy Settings Script** and **Paste Settings Script** allow you to move the current Profiler's settings to a Profiler in another report.

**Append Settings to Table** appends the current profiler's settings to the end of the data table. This is useful if you have a combination of settings in the Profiler that you want to add to an experiment in order to do another run.

**Link Profilers** links all the profilers together, so that a change in a factor in one profiler causes that factor to change to that value in all other profilers, including Surface Plot. This is a global option, set or unset for all profilers.

**Set Script** sets a script that is called each time a factor changes. The set script receives a list of arguments of the form

```
{factor1 = n1, factor2 = n2, ...}
```

For example, to write this list to the log, first define a function

```
ProfileCallbackLog = Function({arg}, show(arg));
```

Then enter **ProfileCallbackLog** in the **Set Script** dialog.

Similar functions convert the factor values to global values:

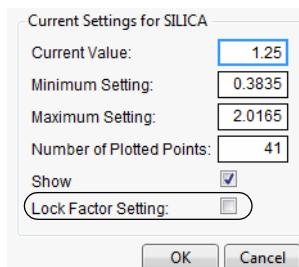
```
ProfileCallbackAssign = Function({arg}, evalList(arg));
```

Or access the values one at a time:

```
ProfileCallbackAccess =
Function({arg}, f1=arg["factor1"]; f2=arg["factor2"]);
```

**Output Grid Table** produces a new data table with columns for the factors that contain grid values, columns for each of the responses with computed values at each grid point, and the desirability computation at each grid point.

If you have a lot of factors, it is impractical to use the **Output Grid Table** command, since it produces a large table. In such cases, you should lock some of the factors, which are held at locked, constant values. To get the dialog to specify locked columns, Alt- or Option-click inside the profiler graph to get a dialog that has a **Lock Factor Setting** checkbox.

**Figure 26.11** Factor Settings Window

**Output Random Table** prompts for a number of runs and creates an output table with that many rows, with random factor settings and predicted values over those settings. This is equivalent to (but much simpler than) opening the Simulator, resetting all the factors to a random uniform distribution, then simulating output. This command is similar to **Output Grid Table**, except it results in a random table rather than a sequenced one.

The prime reason to make uniform random factor tables is to explore the factor space in a multivariate way using graphical queries. This technique is called *Filtered Monte Carlo*.

Suppose you want to see the locus of all factor settings that produce a given range to desirable response settings. By selecting and hiding the points that don't qualify (using graphical brushing or the Data Filter), you see the possibilities of what is left: the opportunity space yielding the result you want.

**Alter Linear Constraints** allows you to add, change, or delete linear constraints. The constraints are incorporated into the operation of **Prediction Profiler**. See “[Linear Constraints](#),” p. 553.

**Save Linear Constraints** allows you to save existing linear constraints to a Table Property/Script called **Constraint**. See “[Linear Constraints](#),” p. 553.

**Default N Levels** allows you to set the default number of levels for each continuous factor. This option is useful when the Profiler is especially large. When calculating the traces for the first time, JMP measures how long it takes. If this time is greater than three seconds, you are alerted that decreasing the Default N Levels speeds up the calculations.

**Conditional Predictions** appears when random effects are included in the model. The random effects are shown and used in formulating the predicted value.

**Simulator** launches the Simulator. The Simulator enables you to create Monte Carlo simulations using random noise added to factors and predictions for the model. A typical use is to set fixed factors at their optimal settings, and uncontrolled factors and model noise to random values and find out the rate that the responses are outside the specification limits. For details see “[The Simulator](#),” p. 570.

**Interaction Profiler** brings up interaction plots that are interactive with respect to the profiler values. This option can help visualize third degree interactions by seeing how the plot changes as current values for the terms are changed. The cells that change for a given term are the cells that do not involve that term directly.

## Desirability Profiling and Optimization

Often there are multiple responses measured and the desirability of the outcome involves several or all of these responses. For example, you might want to maximize one response, minimize another, and keep a third response close to some target value. In desirability profiling, you specify a desirability function for each response. The overall desirability can be defined as the geometric mean of the desirability for each response.

To use desirability profiling, select **Desirability Functions** from the **Prediction Profiler** red triangle menu.

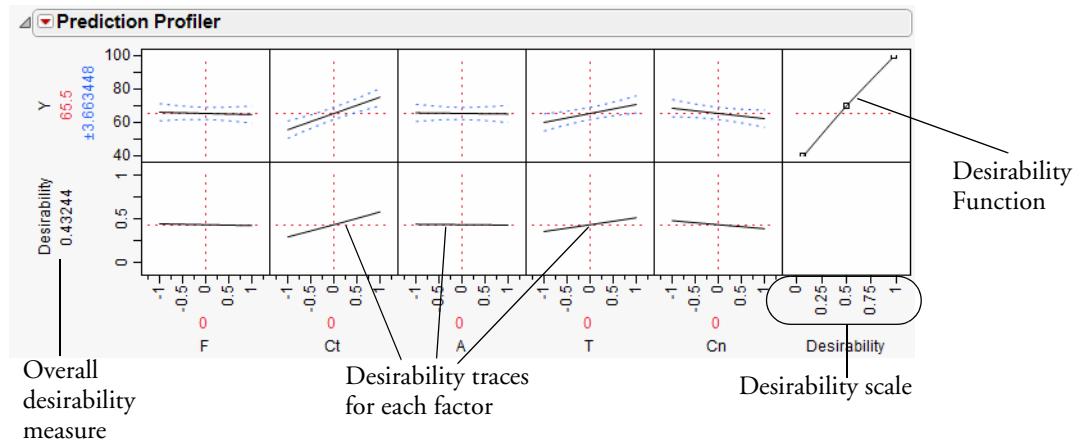
---

**Note:** If the response column has a Response Limits property, desirability functions are turned on by default.

---

This command appends a new row to the bottom of the plot matrix, dedicated to graphing desirability. The row has a plot for each factor showing its *desirability trace*, as illustrated in Figure 26.12. It also adds a column that has an adjustable desirability function for each *Y* variable. The overall desirability measure shows on a scale of zero to one at the left of the row of desirability traces.

**Figure 26.12** The Desirability Profiler



### About Desirability Functions

The desirability functions are smooth piecewise functions that are crafted to fit the control points.

- The minimize and maximize functions are three-part piecewise smooth functions that have exponential tails and a cubic middle.
- The target function is a piecewise function that is a scale multiple of a normal density on either side of the target (with different curves on each side), which is also piecewise smooth and fit to the control points.

These choices give the functions good behavior as the desirability values switch between the maximize, target, and minimize values. For completeness, we implemented the upside-down target also.

JMP doesn't use the Derringer and Suich functional forms. Since they are not smooth, they do not always work well with JMP's optimization algorithm.

The control points are not allowed to reach all the way to zero or one at the tail control points.

### Using the Desirability Function

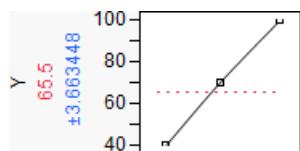
To use a variable's desirability function, drag the function handles to represent a response value.

As you drag these handles, the changing response value shows in the area labeled Desirability to the left of the plots. The dotted line is the response for the current factor settings. The overall desirability shows to the left of the row of desirability traces. Alternatively, you can select **Set Desirabilities** to enter specific values for the points.

The next illustration shows steps to create desirability settings.

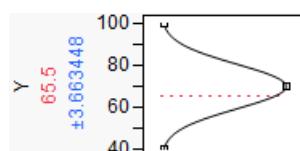
**Maximize** The default desirability function setting is maximize ("higher is better"). The top function handle is positioned at the maximum Y value and aligned at the high desirability, close to 1. The bottom function handle is positioned at the minimum Y value and aligned at a low desirability, close to 0.

**Figure 26.13** Maximizing Desirability

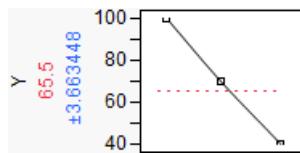


**Target** You can designate a target value as "best." In this example, the middle function handle is positioned at  $Y = 70$  and aligned with the maximum desirability of 1.  $Y$  becomes less desirable as its value approaches either 45 or 95. The top and bottom function handles at  $Y = 45$  and  $Y = 95$  are positioned at the minimum desirability close to 0.

**Figure 26.14** Defining a Target Desirability



**Minimize** The minimize ("smaller is better") desirability function associates high response values with low desirability and low response values with high desirability. The curve is the maximization curve flipped around a horizontal line at the center of plot.

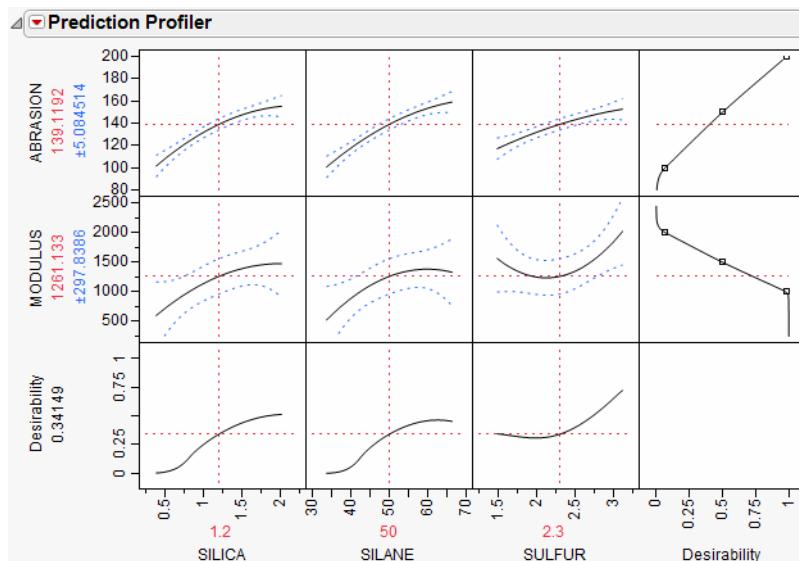
**Figure 26.15** Minimizing Desirability

**Note:** Dragging the top or bottom point of a maximize or minimize desirability function across the  $y$ -value of the middle point results in the opposite point reflecting, so that a Minimize becomes a Maximize, and vice versa.

### The Desirability Profile

The last row of plots shows the desirability trace for each factor. The numerical value beside the word Desirability on the vertical axis is the geometric mean of the desirability measures. This row of plots shows both the current desirability and the trace of desirabilities that result from changing one factor at a time.

For example, Figure 26.16 shows desirability functions for two responses. You want to maximize ABRASION and minimize MODULUS. The desirability plots indicate that you could increase the desirability by increasing any of the factors.

**Figure 26.16** Prediction Profile Plot with Adjusted Desirability and Factor Values

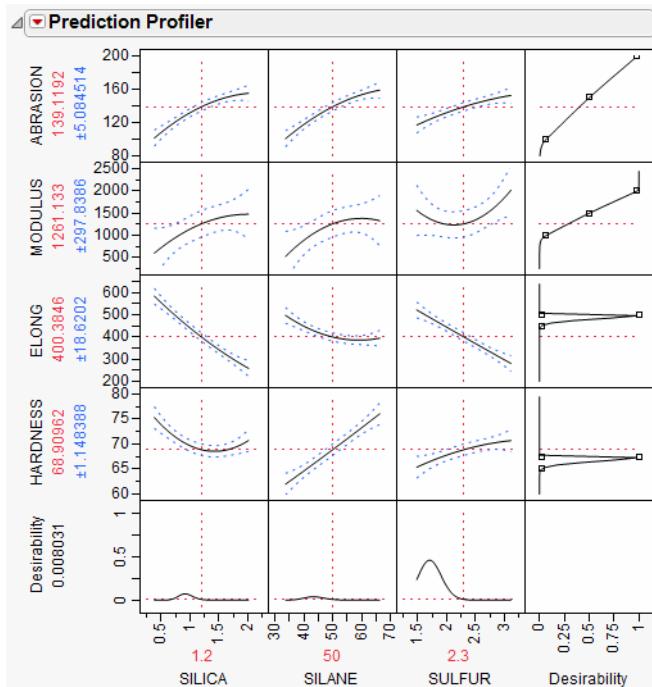
### Desirability Profiling for Multiple Responses

A desirability index becomes especially useful when there are multiple responses. The idea was pioneered by Derringer and Suich (1980), who give the following example. Suppose there are four responses, ABRASION, MODULUS, ELONG, and HARDNESS. Three factors, SILICA, SILANE, and SULFUR, were used in a central composite design.

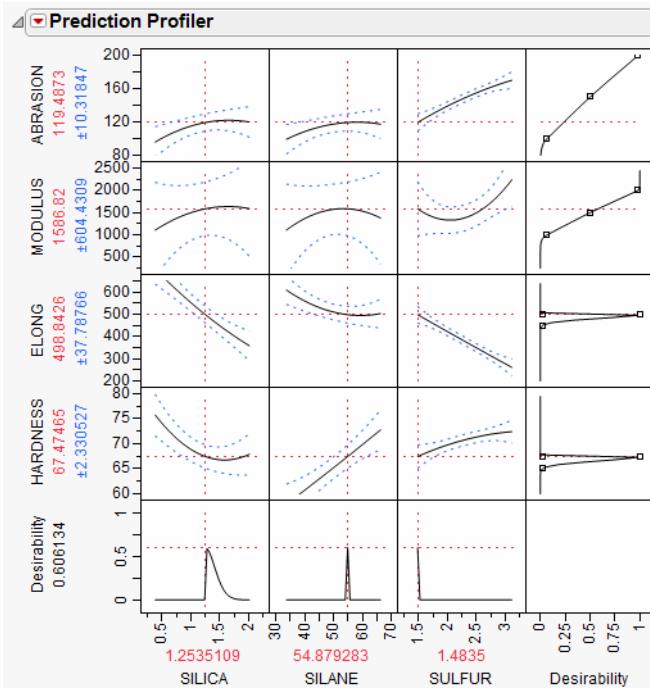
The data are in the Tiretread.jmp table in the Sample Data folder. Use the **RSM For 4 responses** script in the data table, which defines a model for the four responses with a full quadratic response surface. The summary tables and effect information appear for all the responses, followed by the prediction profiler shown in Figure 26.17. The desirability functions are as follows:

1. Maximum ABRASION and maximum MODULUS are most desirable.
2. ELONG target of 500 is most desirable.
3. HARDNESS target of 67.5 is most desirable.

**Figure 26.17** Profiler for Multiple Responses Before Optimization



Select **Maximize Desirability** from the **Prediction Profiler** pop-up menu to maximize desirability. The results are shown in Figure 26.18. The desirability traces at the bottom decrease everywhere except the current values of the effects, which indicates that any further adjustment could decrease the overall desirability.

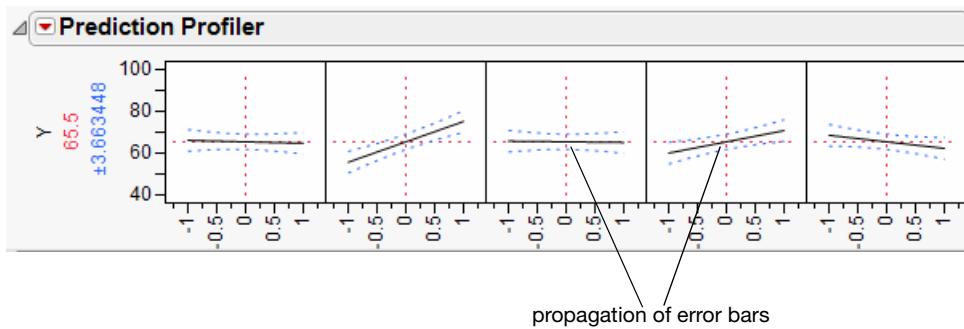
**Figure 26.18** Profiler After Optimization

## Special Profiler Topics

### Propagation of Error Bars

Propagation of error (POE) is important when attributing the variation of the response in terms of variation in the factor values when the factor values are not very controllable.

In JMP's implementation, the **Profiler** first looks at the factor and response variables to see if there is a **Sigma** column property (a specification for the standard deviation of the column, accessed through the **Cols > Column Info** dialog box). If the property exists, then the **Prop of Error Bars** command becomes accessible in the **Prediction Profiler** drop-down menu. This displays the  $3\sigma$  interval that is implied on the response due to the variation in the factor.

**Figure 26.19** Propagation of Errors Bars in the Prediction Profiler

The interval is calculated (assuming that the variation among the factors is uncorrelated) by

$$\sum_{i=1}^N \left( \sigma_{x_i}^2 \times \left( \frac{\partial f}{\partial x_i} \right)^2 \right) + \sigma_y^2$$

where  $f$  is the prediction function,  $x_i$  is the  $i^{\text{th}}$  factor, and  $N$  is the number of factors.

Currently, these partial derivatives are calculated by numerical derivatives:

centered, with  $\delta = \text{xrange}/10000$

POE limits increase dramatically in response surface models when you are over a more sloped part of the response surface. One of the goals of robust processes is to operate in flat areas of the response surface so that variations in the factors do not amplify in their effect on the response.

## Customized Desirability Functions

It is possible to use a customized desirability function. For example, suppose you want to maximize using the following function.

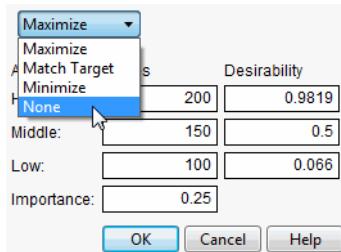
**Figure 26.20** Maximizing Desirability Based on a Function

$$\begin{aligned}
 & \text{Pred Formula ABRASION} \\
 & \quad 96 \\
 & + \text{Pred Formula MODULUS} \\
 & \quad 700 \\
 & \quad 33 \\
 & + \frac{[(\text{Pred Formula ELONG}-450)+1]}{2} \\
 & + \frac{[(\text{Pred Formula HARDNESS}-67)+1]}{2}
 \end{aligned}$$

First, create a column called MyDesire that contains the above formula. Then, launch the **Profiler** using **Graph > Profiler** and include all the Pred Formula columns and the MyDesire column. Turn on the desirability functions by selecting **Desirability Functions** from the red-triangle menu. All the desirability functions for the individual effects must be turned off. To do this, first double-click in a desirability plot window, then select **None** in the dialog that appears (Figure 26.21). Set the desirability for MyDesire to be maximized.

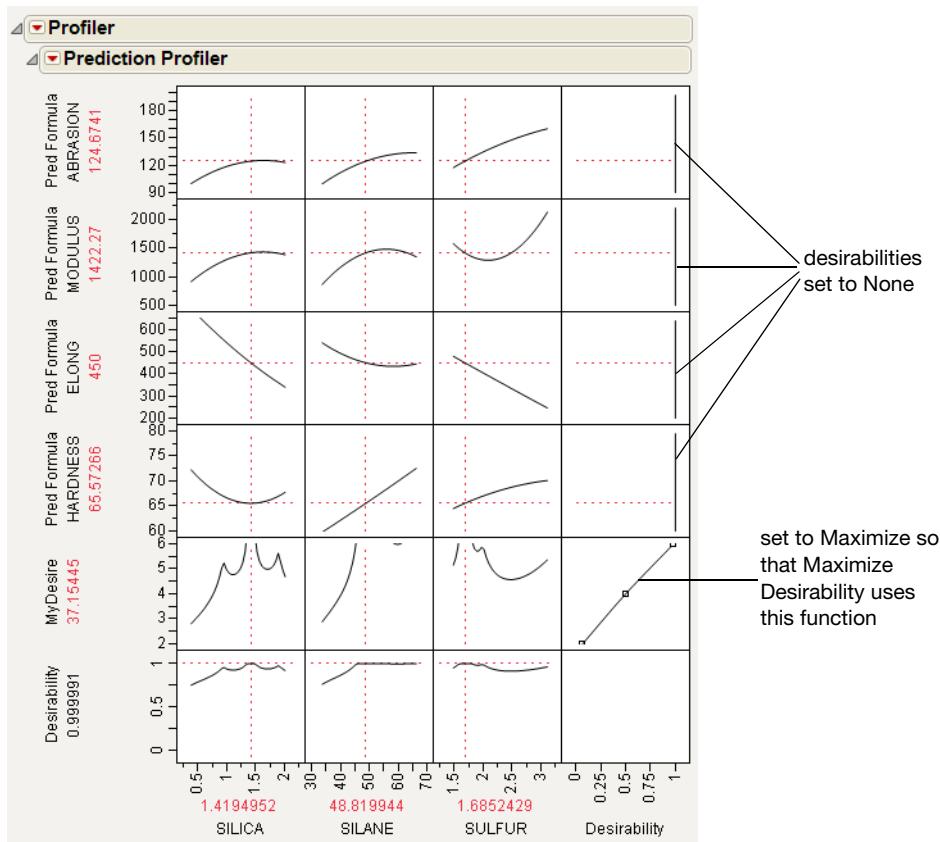
---

**Figure 26.21** Selecting No Desirability Goal



---

At this point, selecting **Maximize Desirability** uses only the custom MyDesire function.

**Figure 26.22** Maximized Custom Desirability

## Mixture Designs

When analyzing a mixture design, JMP constrains the ranges of the factors so that settings outside the mixture constraints are not possible. This is why, in some mixture designs, the profile traces appear to turn abruptly.

When there are mixture components that have constraints, other than the usual zero-to-one constraint, a new submenu, called **Profile at Boundary**, appears on the **Prediction Profiler** popup menu. It has the following two options:

**Turn At Boundaries** lets the settings continue along the boundary of the restraint condition.

**Stop At Boundaries** truncates the prediction traces to the region where strict proportionality is maintained.

## Expanding Intermediate Formulas

The **Profiler** launch dialog has an **Expand Intermediate Formulas** checkbox. When this is checked, then when the formula is examined for profiling, if it references another column that has a formula containing references to other columns, then it substitutes that formula and profiles with respect to the end references—not the intermediate column references.

For example, when **Fit Model** fits a logistic regression for two levels (say A and B), the end formulas ( $\text{Prob}[A]$  and  $\text{Prob}[B]$ ) are functions of the  $\text{Lin}[x]$  column, which itself is a function of another column  $x$ . If **Expand Intermediate Formulas** is selected, then when  $\text{Prob}[A]$  is profiled, it is with reference to  $x$ , not  $\text{Lin}[x]$ .

In addition, using the **Expand Intermediate Formulas** checkbox enables the **Save Expanded Formulas** command in the platform red triangle menu. This creates a new column with a formula, which is the formula being profiled as a function of the end columns, not the intermediate columns.

## Linear Constraints

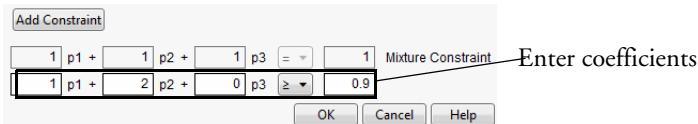
The **Prediction Profiler**, **Custom Profiler** and **Mixture Profiler** can incorporate linear constraints into their operations. Linear constraints can be entered in two ways, described in the following sections.

### Pop-up Menu Item

To enter linear constraints via the pop-up menu, select **Alter Linear Constraints** from either the **Prediction Profiler** or **Custom Profiler** pop-up menu.

Choose **Add Constraint** from the resulting dialog, and enter the coefficients into the appropriate boxes. For example, to enter the constraint  $p1 + 2*p2 \leq 0.9$ , enter the coefficients as shown in Figure 26.23. As shown, if you are profiling factors from a mixture design, the mixture constraint is present by default and cannot be modified.

**Figure 26.23** Enter Coefficients



After you click **OK**, the **Profiler** updates the profile traces, and the constraint is incorporated into subsequent analyses and optimizations.

If you attempt to add a constraint for which there is no feasible solution, a message is written to the log and the constraint is not added. To delete a constraint, enter zeros for all the coefficients.

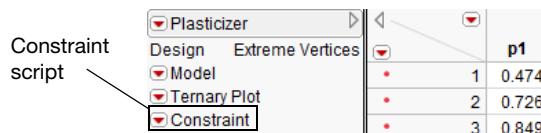
Constraints added in one profiler are not accessible by other profilers until saved. For example, if constraints are added under the **Prediction Profiler**, they are not accessible to the **Custom Profiler**. To use the

constraint, you can either add it under the **Custom Profiler** pop-up menu, or use the **Save Linear Constraints** command described in the next section.

### Constraint Table Property/Script

If you add constraints in one profiler and want to make them accessible by other profilers, use the **Save Linear Constraints** command, accessible through the platform pop-up menu. For example, if you created constraints in the **Prediction Profiler**, choose **Save Linear Constraints** under the **Prediction Profiler** pop-up menu. The **Save Linear Constraints** command creates or alters a Table Script called **Constraint**. An example of the Table Property is shown in Figure 26.24.

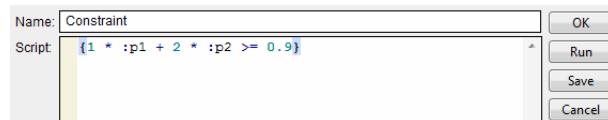
**Figure 26.24** Constraint Table Script



Design	Extreme Vertices	p1
•	1	0.474
•	2	0.726
•	3	0.849

The **Constraint** Table Property is a list of the constraints, and is editable. It is accessible to other profilers, and negates the need to enter the constraints in other profilers. To view or edit **Constraint**, right click on the pop-up menu and select **Edit**. The contents of the constraint from Figure 26.23 is shown below in Figure 26.25.

**Figure 26.25** Example Constraint



The **Constraint** Table Script can be created manually by choosing **New Script** from the pop-up menu beside a table name.

**Note:** When creating the **Constraint** Table Script manually, the spelling must be exactly “Constraint”. Also, the constraint variables are case sensitive and must match the column name. For example, in Figure 26.25, the constraint variables are p1 and p2, not P1 and P2.

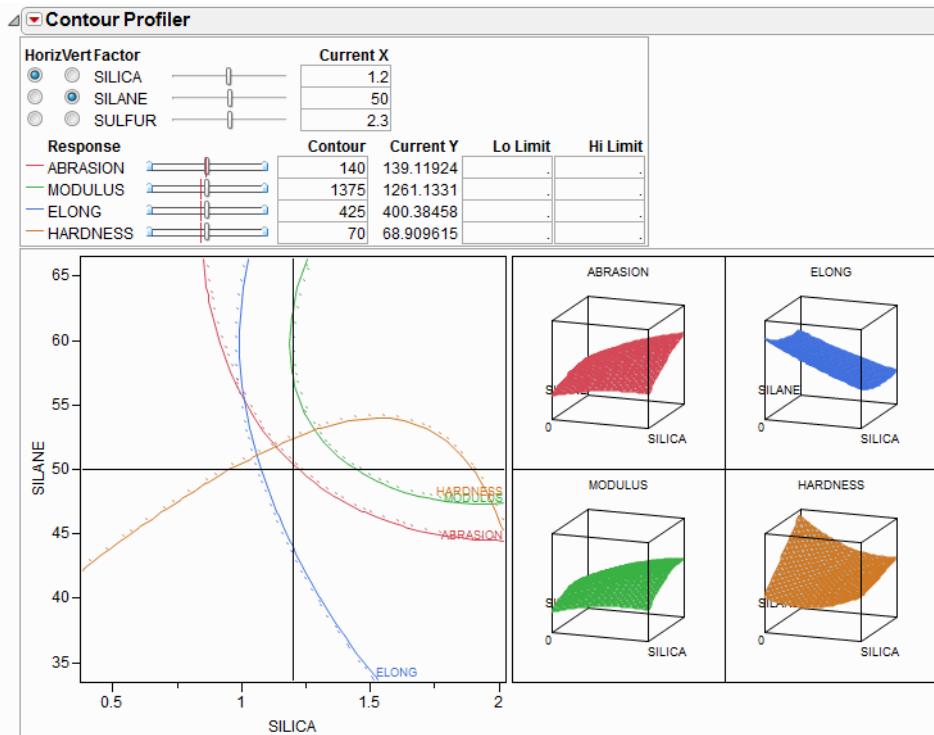
The **Constraint** Table Script is also created when specifying linear constraints when designing an experiment.

The **Alter Linear Constraints** and **Save Linear Constraints** commands are not available in the **Mixture Profiler**. To incorporate linear constraints into the operations of the **Mixture Profiler**, the **Constraint** Table Script must be created by one of the methods discussed in this section.

## Contour Profiler

The **Contour Profiler** shows response contours for two factors at a time. The interactive contour profiling facility is useful for optimizing response surfaces graphically. Figure 26.26 shows an example of the Contour Profiler for the Tiretread sample data.

**Figure 26.26** Contour Profiler



- There are slider controls and edit fields for both the *X* and *Y* variables.
- The Current *X* values generate the Current *Y* values. The Current *X* location is shown by the crosshair lines on the graph. The Current *Y* values are shown by the small red lines in the slider control.
- The other lines on the graph are the contours for the responses set by the *Y* slider controls or by entering values in the Contour column. There is a separately colored contour for each response (4 in this example).
- You can enter low and high limits to the responses, which results in a shaded region. To set the limits, you can click and drag from the side zones of the *Y* sliders or enter values in the Lo Limit or Hi Limit columns. If a response column's **Spec Limits** column property has values for Lower Spec Limit or Upper Spec Limit, those values are used as the initial values for Lo Limit and Hi Limit.

- If you have more than two factors, use the radio buttons in the upper left of the report to switch the graph to other factors.
- Right-click on the slider control and select **Rescale Slider** to change the scale of the slider (and the plot for an active X variable).
- For each contour, there is a dotted line in the direction of higher response values, so that you get a sense of direction.
- Right-click on the color legend for a response (under **Response**) to change the color for that response.

## Contour Profiler Pop-up Menu

**Grid Density** lets you set the density of the mesh plots (Surface Plots).

**Graph Updating** gives you the options to update the Contour Profiler **Per Mouse Move**, which updates continuously, or **Per Mouse Up**, which waits for the mouse to be released to update. (The difference might not be noticeable on a fast machine.)

**Surface Plot** hides or shows mesh plots.

**Contour Label** hides or shows a label for the contour lines. The label colors match the contour colors.

**Contour Grid** draws contours on the Contour Profiler plot at intervals you specify.

**Factor Settings** is a submenu of commands that allows you to save and transfer the Contour Profiler's settings to other parts of JMP. Details are in the section “[Factor Settings](#),” p. 543.

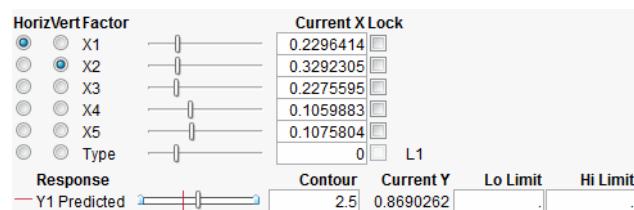
**Simulator** launches the **Simulator**. See “[The Simulator](#),” p. 570.

**Up Dots** shows or hides dotted lines corresponding to each contour. The dotted lines show the direction of increasing response values, so that you get a sense of direction.

## Mixtures

For mixture designs, a Lock column appears in the **Contour Profiler** (Figure 26.27). This column allows you to lock settings for mixture values so that they are not changed when the mixture needs to be adjusted due to other mixture effects being changed. When locked columns exist, the shaded area for a mixture recognizes the newly restricted area.

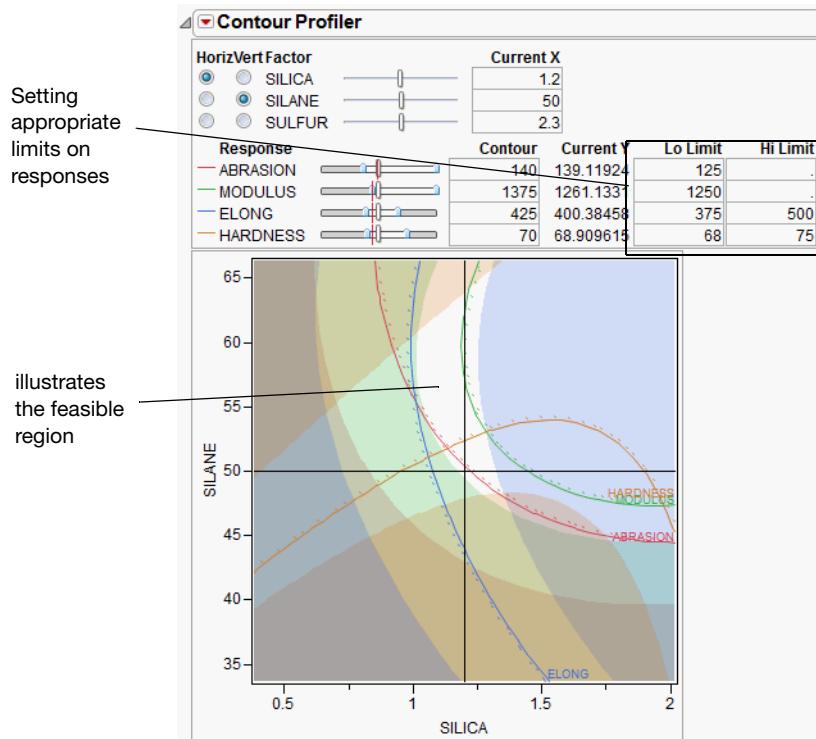
**Figure 26.27** Boxes to Lock Columns



## Constraint Shading

Specifying limits to the Y's shades the areas outside the limits as shown in Figure 26.28. The unshaded white area becomes the feasible region.

**Figure 26.28** Settings for Contour Shading



If a response column's **Spec Limits** column property has values for Lower Spec Limit or Upper Spec Limit, those values are used as the initial values for Lo Limit and Hi Limit.

## Mixture Profiler

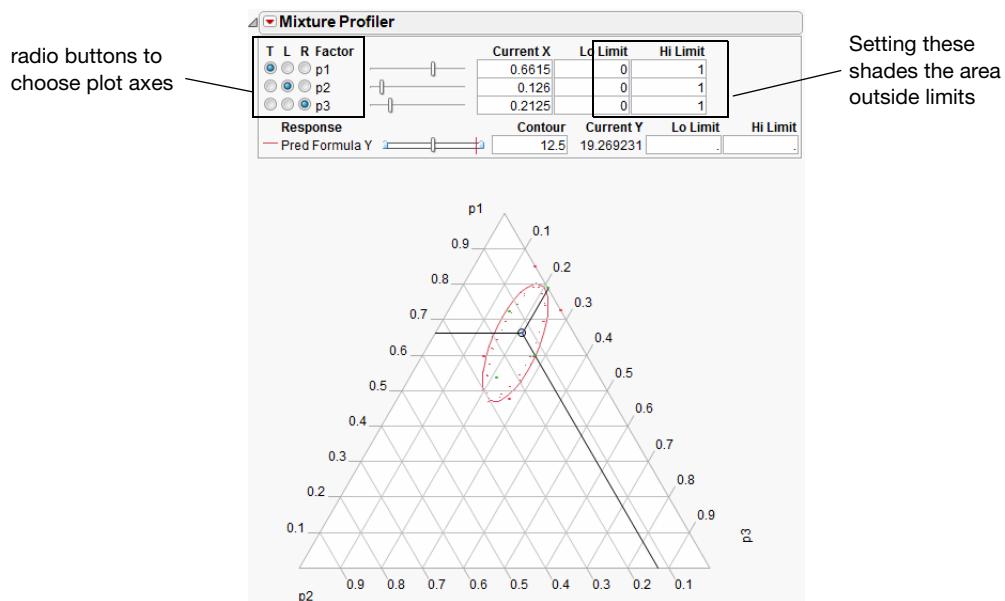
The **Mixture Profiler** shows response contours for mixture experiment models, where three or more factors in the experiment are components (ingredients) in a mixture. The **Mixture Profiler** is useful for visualizing and optimizing response surfaces resulting from mixture experiments.

Figure 26.29 shows an example of the Mixture Profiler for the sample data in Plasticizer.jmp. To generate the graph shown, select **Mixture Profiler** from the **Graph** menu. In the resulting Mixture Profiler launch dialog, assign Pred Formula Y to the **Y, Prediction Formula** role and click **OK**. Delete the Lo and Hi limits from p1, p2, and p3.

Many of the features shown are the same as those of the **Contour Profiler** and are described on p. 556. Some of the features unique to the **Mixture Profiler** include:

- A ternary plot is used instead of a Cartesian plot. A ternary plot enables you to view three mixture factors at a time.
- If you have more than three factors, use the radio buttons at the top left of the **Mixture Profiler** window to graph other factors. For detailed explanation of radio buttons and plot axes, see “[Explanation of Ternary Plot Axes](#),” p. 558
- If the factors have constraints, you can enter their low and high limits in the Lo Limit and Hi Limit columns. This shades non-feasible regions in the profiler. As in **Contour Plot**, low and high limits can also be set for the responses.

**Figure 26.29** Mixture Profiler



## Explanation of Ternary Plot Axes

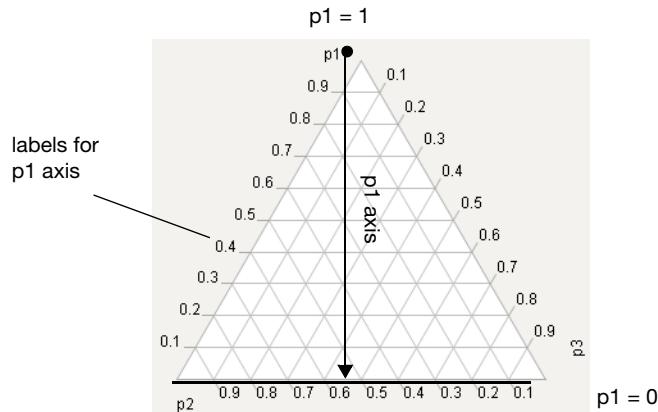
The sum of all mixture factor values in a mixture experiment is a constant, usually, and henceforth assumed to be 1. Each individual factor's value can range between 0 and 1, and *three* are represented on the axes of the ternary plot.

For a three factor mixture experiment in which the factors sum to 1, the plot axes run from a vertex (where a factor's value is 1 and the other two are 0) perpendicular to the other side (where that factor is 0 and the sum of the other two factors is 1). See Figure 26.30.

For example, in Figure 26.30, the proportion of  $p_1$  is 1 at the top vertex and 0 along the bottom edge. The tick mark labels are read along the left side of the plot. Similar explanations hold for  $p_2$  and  $p_3$ .

For an explanation of ternary plot axes for experiments with more than three mixture factors, see “[More than Three Mixture Factors](#),” p. 559.

**Figure 26.30** Explanation of p1 Axis.

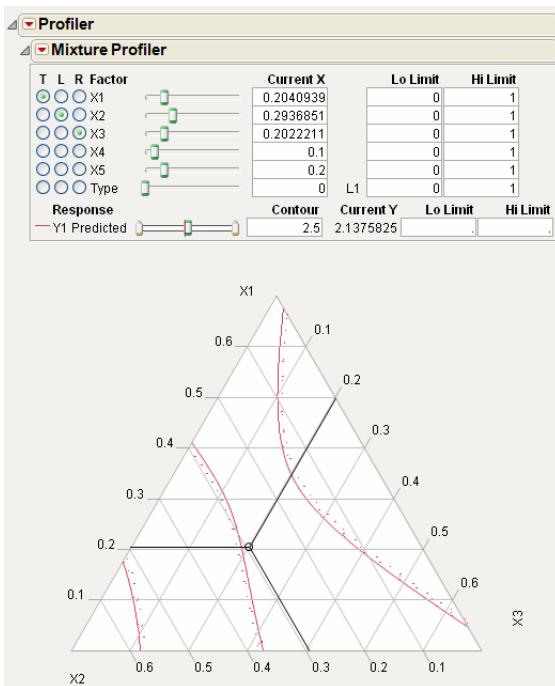


## More than Three Mixture Factors

The ternary plot can only show three factors at a time. If there are more than three factors in the model you are profiling, the total of the three on-axis (displayed) factors is 1 minus the total of the off-axis (non-displayed) factors. Also, the plot axes are scaled such that the maximum value a factor can attain is 1 minus the total for the off-axis factors.

For example Figure 26.31 shows the **Mixture Profiler** for an experiment with 5 factors. The Five Factor Mixture.jmp data table is being used, with the Y1 Predicted column as the formula. The on-axis factors are x1, x2 and x3, while x4 and x5 are off-axis. The value for x4 is 0.1 and the value for x5 is 0.2, for a total of 0.3. This means the sum of x1, x2 and x3 has to equal  $1 - 0.3 = 0.7$ . In fact, their **Current X** values add to 0.7. Also, note that the maximum value for a plot axis is now 0.7, not 1.

If you change the value for either x4 or x5, then the values for x1, x2 and x3 change, keeping their relative proportions, to accommodate the constraint that factor values sum to 1.

**Figure 26.31** Scaled Axes to Account for Off-Axis Factors Total

## Mixture Profiler Options

The commands under the Mixture Profiler popup menu are explained below.

**Ref Labels** shows or hides the labels on the plot axes.

**Ref Lines** shows or hides the grid lines on the plot.

**Show Points** shows or hides the design points on the plot. This feature is only available if there are no more than three mixture factors.

**Show Current Value** shows or hides the three-way crosshairs on the plot. The intersection of the crosshairs represents the current factor values. The Current X values above the plot give the exact coordinates of the crosshairs.

**Show Constraints** shows or hides the shading resulting from any constraints on the factors. Those constraints can be entered in the Lo Limits and Hi Limits columns above the plot, or in the Mixture Column Property for the factors.

**Up Dots** shows or hides dotted line corresponding to each contour. The dotted lines show the direction of increasing response values, so that you get a sense of direction.

**Contour Grid** draws contours on the plot at intervals you specify.

**Remove Contour Grid** removes the contour grid if one is on the plot.

**Factor Settings** is a submenu of commands that allows you to save and transfer the **Mixture Profiler** settings to other parts of JMP. Details on this submenu are found in the discussion of the profiler on [p. 543](#).

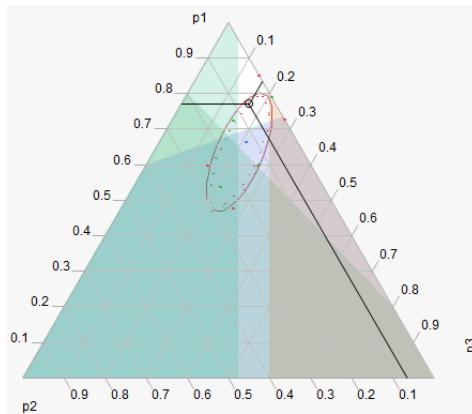
## Linear Constraints

The **Mixture Profiler** can incorporate linear constraints into its operations. To do this, a **Constraint Table Script** must be part of the data table. See “[Linear Constraints](#),” p. 553 for details on creating the Table Script.

When using constraints, unfeasible regions are shaded in the profiler. Figure 26.32 shows an example of a mixture profiler with shaded regions due to four constraints. The unshaded portion is the resulting feasible region. The constraints are below:

- $4*p2 + p3 \leq 0.8$
- $p2 + 1.5*p3 \leq 0.4$
- $p1 + 2*p2 \geq 0.8$
- $p1 + 2*p2 \leq 0.95$

**Figure 26.32** Shaded Regions Due to Linear Constraints



## Examples

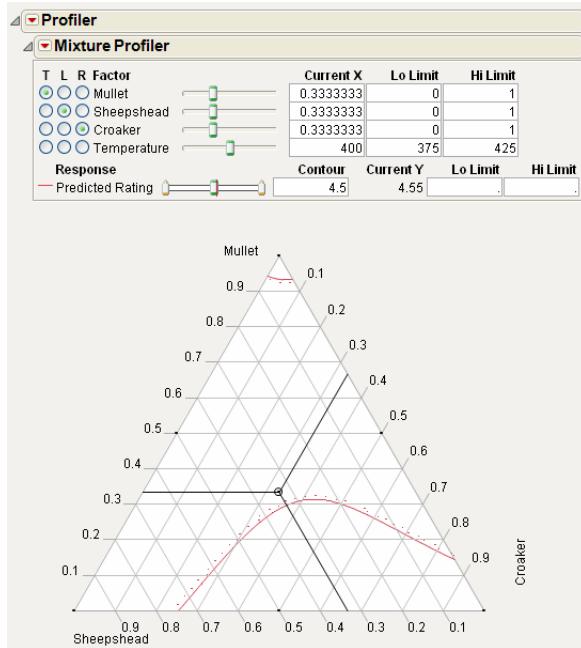
### Single Response

This example, adapted from Cornell (1990), comes from an experiment to optimize the texture of fish patties. The data is in **Fish Patty.jmp**. The columns **Mullet**, **Sheepshead** and **Croaker** represent what proportion of the patty came from that fish type. The column **Temperature** represents the oven temperature used to bake the patties. The column **Rating** is the response and is a measure of texture acceptability, where

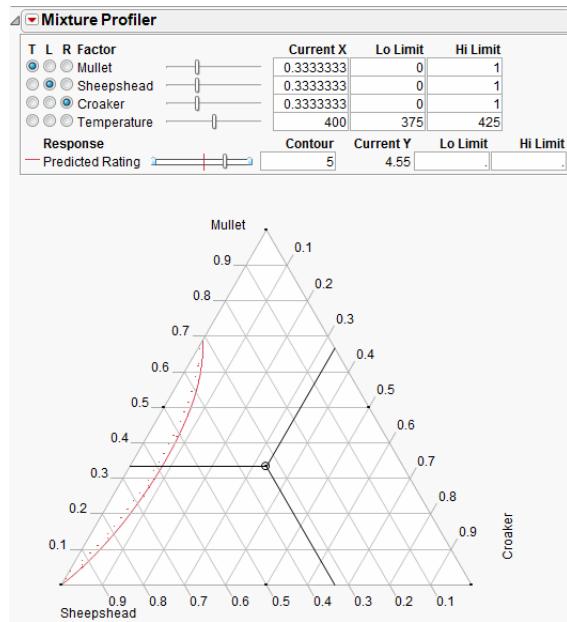
higher is better. A response surface model was fit to the data and the prediction formula was stored in the column Predicted Rating.

To launch the **Mixture Profiler**, select **Graph > Mixture Profiler**. Assign Predicted Rating to **Y, Prediction Formula** and click **OK**. The output should appear as in Figure 26.33.

**Figure 26.33** Initial Output for Mixture Profiler.

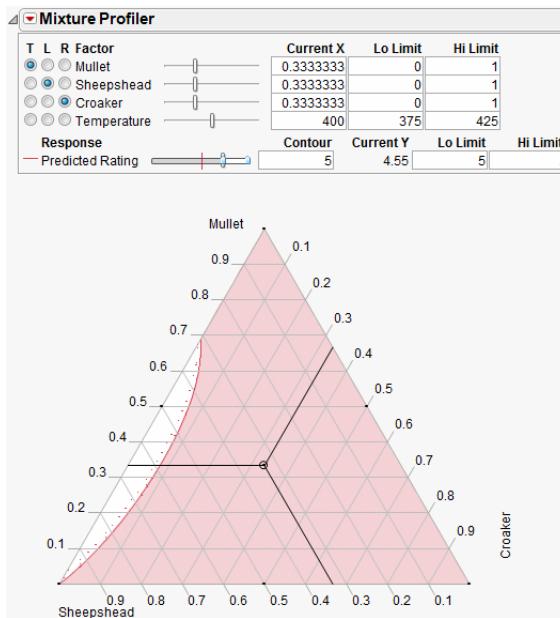


The manufacturer wants the rating to be at least 5. Use the slider control for **Predicted Rating** to move the contour close to 5. Alternatively, you can enter 5 in the **Contour** edit box to bring the contour to a value of 5. Figure 26.34 shows the resulting contour.

**Figure 26.34** Contour Showing a Predicted Rating of 5

The Up Dots shown in Figure 26.34 represent the direction of increasing Predicted Rating. Enter 5 in the Lo Limit edit box. The resulting shaded region shown in Figure 26.35 represents factor combinations that will yield a rating less than 5. To produce patties with at least a rating of 5, the manufacturer can set the factors values anywhere in the feasible (unshaded) region.

The feasible region represents the factor combinations predicted to yield a rating of 5 or more. Notice the region has small proportions of Croaker (<10%), mid to low proportions of Mullet (<70%) and mid to high proportions of Sheepshead (>30%).

**Figure 26.35** Contour Shading Showing Predicted Rating of 5 or more.


Up to this point the fourth factor, Temperature, has been held at 400 degrees. Move the slide control for Temperature and watch the feasible region change.

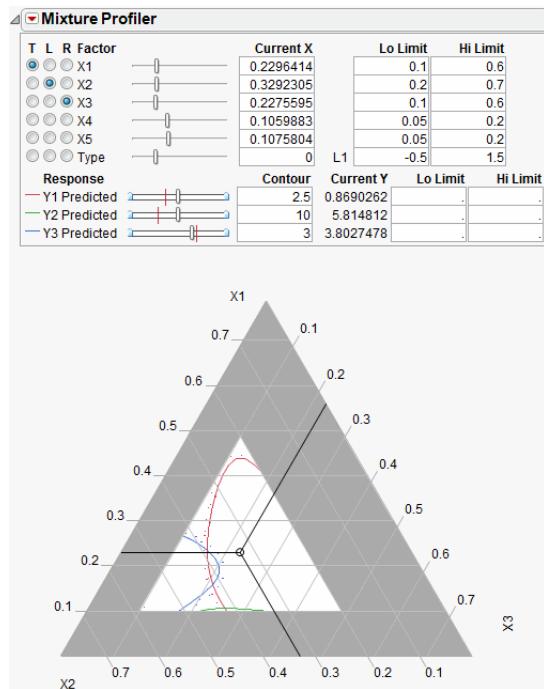
Additional analyses may include:

- Optimize the response across all four factors simultaneously. See “[The Custom Profiler](#),” p. 569 or “[Desirability Profiling and Optimization](#),” p. 545.
- Simulate the response as a function of the random variation in the factors and model noise. See “[The Simulator](#),” p. 570.

### Multiple Responses

This example uses data from **Five Factor Mixture.jmp**. There are five continuous factors ( $x_1$ – $x_5$ ), one categorical factor (Type), and three responses, Y1, Y2 and Y3. A response surface model is fit to each response and the prediction equations are saved in Y1 Predicted, Y2 Predicted and Y3 Predicted.

Launch the **Mixture Profiler** and assign the three prediction formula columns to the **Y, Prediction Formula** role, then click **OK**. Enter 3 in the **Contour** edit box for Y3 Predicted so the contour shows on the plot. The output appears in Figure 26.36.

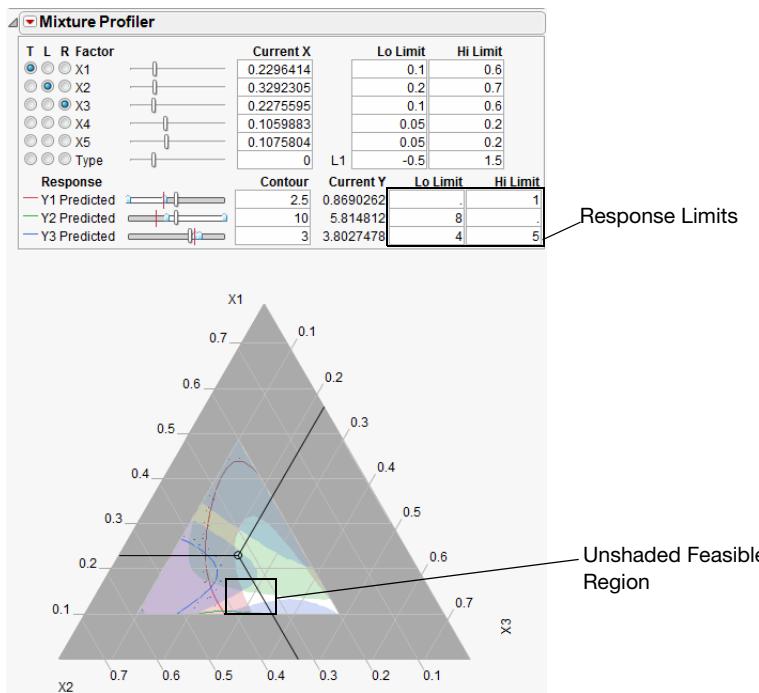
**Figure 26.36** Initial Output Window for Five Factor Mixture

A few items to note about the output in Figure 26.36.

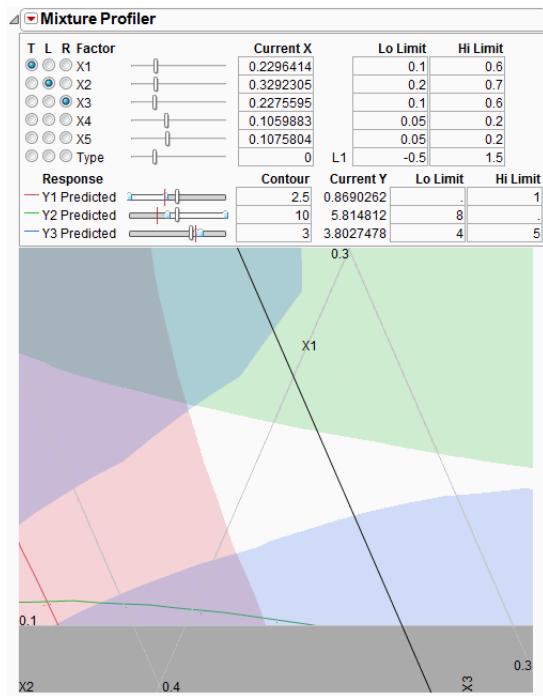
- All the factors appear at the top of the window. The mixture factors have low and high limits, which were entered previously as a Column Property. See *Using JMP* for more information about entering column properties. Alternatively, you can enter the low and high limits directly by entering them in the **Lo Limit** and **Hi Limit** boxes.
- Certain regions of the plot are shaded in gray to account for the factor limits.
- The on-axis factors,  $x_1$ ,  $x_2$  and  $x_3$ , have radio buttons selected.
- The categorical factor, **Type**, has a radio button, but it cannot be assigned to the plot. The current value for **Type** is L1, which is listed immediately to the right of the **Current X** box. The **Current X** box for **Type** uses a 0 to represent L1.
- All three prediction equations have contours on the plot and are differentiated by color.

A manufacturer desires to hold  $Y_1$  less than 1, hold  $Y_2$  greater than 8 and hold  $Y_3$  between 4 and 5, with a target of 4.5. Furthermore, the low and high limits on the factors need to be respected. The **Mixture Profiler** can help you investigate the response surface and find optimal factor settings.

Start by entering the response constraints into the **Lo Limit** and **Hi Limit** boxes, as shown in Figure 26.37. Colored shading appears on the plot and designates unfeasible regions. The feasible region remains white (unshaded). Use the **Response** slider controls to position the contours in the feasible region.

**Figure 26.37** Response Limits and Shading

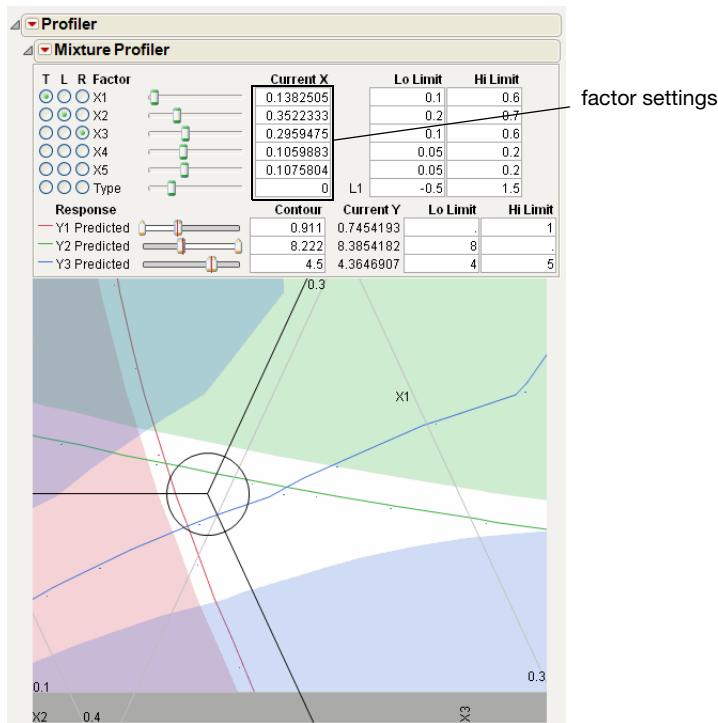
The feasible region is small. Use the magnifier tool to zoom in on the feasible region shown with a box in Figure 26.37. The enlarged feasible region is shown in Figure 26.38.

**Figure 26.38** Feasible Region Enlarged

The manufacturer wants to maximize Y1, minimize Y2 and have Y3 at 4.5.

- Use the slider controls or **Contour** edit boxes for Y1 Predicted to maximize the red contour within the feasible region. Keep in mind the Up Dots show direction of increasing predicted response.
- Use the slider controls or **Contour** edit boxes for Y2 Predicted to minimize the green contour within the unshaded region.
- Enter 4.5 in the **Contour** edit box for Y3 Predicted to bring the blue contour to the target value.

The resulting three contours don't all intersect at one spot, so you will have to compromise. Position the three-way crosshairs in the middle of the contours to understand the factor levels that produce those response values.

**Figure 26.39** Factor Settings

As shown in Figure 26.39, the optimal factor settings can be read from the **Current X** boxes.

The factor values above hold for the current settings of  $x_4$ ,  $x_5$  and **Type**. Select **Factor Settings > Remember Settings** from the **Mixture Profiler** pop-up menu to save the current settings. The settings are appended to the bottom of the report window and appear as shown below.

**Figure 26.40** Remembered Settings

Remembered Settings						
Setting	X1	X2	X3	X4	X5	Type
Type L1	0.1382505	0.3522333	0.2959475	0.1059883	0.1075804	L1

With the current settings saved, you can now change the values of  $x_4$ ,  $x_5$  and **Type** to see what happens to the feasible region. You can compare the factor settings and response values for each level of **Type** by referring to the **Remembered Settings** report.

## Surface Profiler

The **Surface Profiler** shows a three-dimensional surface plot of the response surface. The functionality of **Surface Profiler** is the same as the **Surface Plot** platform, but with fewer options. Details of Surface Plots are found in the “[Plotting Surfaces](#)” chapter.

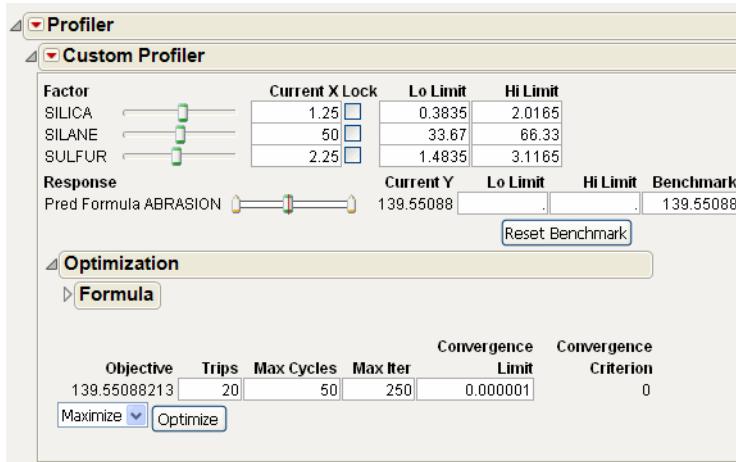
## The Custom Profiler

The **Custom Profiler** allows you to optimize factor settings computationally, without graphical output. This is used for large problems that would have too many graphs to visualize well.

It has many fields in common with other profilers. The **Benchmark** field represents the value of the prediction formula based on current settings. Click **Reset Benchmark** to update the results.

The **Optimization** outline node allows you to specify the formula to be optimized and specifications about the optimization iterations. Click the **Optimize** button to optimize based on current settings.

**Figure 26.41** Custom Profiler



## Custom Profiler Options

**Factor Settings** is a submenu identical to the one covered on [p. 543](#).

**Log Iterations** outputs iterations to a table.

**Alter Linear Constraints** allows you to add, change, or delete linear constraints. The constraints are incorporated into the operation of **Custom Profiler**. See “[Linear Constraints](#),” [p. 553](#).

**Save Linear Constraints** allows you to save existing linear constraints to a Table Property/Script called **Constraint**. See “[Linear Constraints](#),” [p. 553](#).

**Simulator** launches the Simulator. See “[The Simulator](#),” p. 570.

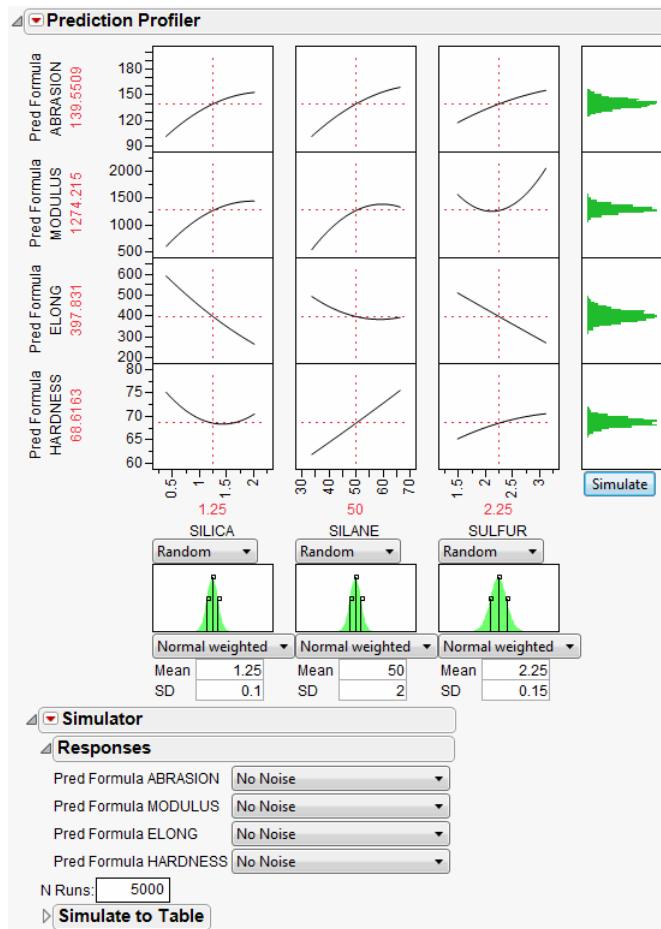
---

## The Simulator

Simulation allows you to discover the distribution of model outputs as a function of the random variation in the factors and model noise. The simulation facility in the profilers provides a way to set up the random inputs and run the simulations, producing an output table of simulated values.

An example of this facility’s use would be to find out the defect rate of a process that has been fit, and see if it is robust with respect to variation in the factors. If specification limits have been set in the responses, they are carried over into the simulation output, allowing a prospective capability analysis of the simulated model using new factors settings.

In the **Profiler**, the **Simulator** is integrated into the graphical layout. Factor specifications are aligned below each factor’s profile. A simulation histogram is shown on the right for each response.

**Figure 26.42 Profiler with Simulator**

In the other profilers, the **Simulator** is less graphical, and kept separate. There are no integrated histograms, and the interface is textual. However, the internals and output tables are the same.

## Specifying Factors

Factors (inputs) and responses (outputs) are already given roles by being in the **Profiler**. Additional specifications for the simulator are on how to give random values to the factors, and add random noise to the responses.

For each factor, the choices of how to give values are as follows:

**Fixed** fixes the factor at the specified value. The initial value is the current value in the profiler, which may be a value obtained through optimization.

**Random** gives the factor the specified distribution and distributional parameters.

See the *JMP User's Guide* for descriptions of most of these random functions. If the factor is categorical, then the distribution is characterized by probabilities specified for each category, with the values normalized to sum to 1.

**Normal weighted** is Normally distributed with the given mean and standard deviation, but a special stratified and weighted sampling system is used to simulate very rare events far out into the tails of the distribution. This is a good choice when you want to measure very low defect rates accurately. See “Statistical Details,” p. 605.

**Normal truncated** is a normal distribution limited by lower and upper limits. Any random realization that exceeds these limits is discarded and the next variate within the limits is chosen. This is used to simulate an inspection system where inputs that do not satisfy specification limits are discarded or sent back.

**Normal censored** is a normal distribution limited by lower and upper limits. Any random realization that exceeds a limit is just set to that limit, putting a density mass on the limits. This is used to simulate a re-work system where inputs that do not satisfy specification limits are reworked until they are at that limit.

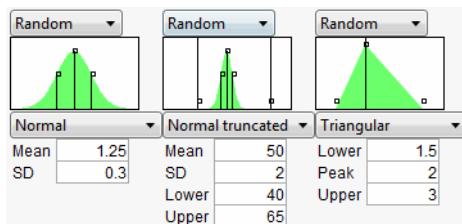
**Sampled** means that JMP picks values at random from that column in the data table.

**External** means that JMP picks values at random from a column in another table. You are prompted to choose the table and column.

The **Aligned** checkbox is used for two or more Sampled or External sources. When checked, the random draws come from the same row of the table. This is useful for maintaining the correlation structure between two columns. If the Aligned option is used to associate two columns in different tables, the columns must have equal number of rows.

In the **Profiler**, a graphical specification shows the form of the density for the continuous distributions, and provides control points that can be dragged to change the distribution. The drag points for the Normal are the mean and the mean plus or minus one standard deviation. The Normal truncated and censored add points for the lower and upper limits. The Uniform and Triangular have limit control points, and the Triangular adds the mode.

**Figure 26.43** Distributions



**Expression** allows you to write your own expression in JMP Scripting Language (JSL) form into a field. This gives you flexibility to make up a new random distribution. For example, you could create a censored normal distribution that guaranteed non-negative values with an expression like

`Max(0, RandomNormal(5, 2))`. In addition, character results are supported, so `If(Random Uniform() < 0.2, "M", "F")` works fine. After entering the expression, click the **Reset** button to submit the expression.

**Multivariate** allows you to generate a multivariate normal for when you have correlated factors. Specify the mean and standard deviation with the factor, and a correlation matrix separately.

**Figure 26.44** Using a Correlation Matrix

The screenshot shows the JMP Profiler interface. At the top left, there is a dropdown menu set to 'Multivariate'. Below it, two input fields show 'Mean' as 2.3 and 'SD' as 0.3. To the right, a section titled 'X Correlation Specification' displays a correlation matrix for factors SILICA, SILANE, and SULFUR. The matrix is as follows:

Factor	SILANE	SULFUR
SILICA	0	0.82
SILANE		0

## Specifying the Response

If the model is only partly a function of the factors, and the rest of the variation of the response is attributed to random noise, then you will want to specify this with the responses. The choices are:

**No Noise** just evaluates the response from the model, with no additional random noise added.

**Add Random Noise** obtains the response by adding a normal random number with the specified standard deviation to the evaluated model.

**Add Random Weighted Noise** is distributed like Add Random Noise, but with weighted sampling to enable good extreme tail estimates.

**Add Multivariate Noise** yields a response as follows: A multivariate random normal vector is obtained using a specified correlation structure, and it is scaled by the specified standard deviation and added to the value obtained by the model.

## Run the Simulation

Specify the number of runs in the simulation by entering it in the **N Runs** box.

After factor and response distributions are set, click the **Simulate** button to run the simulation.

Or, use the **Make Table** button to generate a simulation and output the results to a data table.

The table contains **N Runs** rows, simulated factor values from the specified distributions, and the corresponding response values. If spec limits are given, the table also contains a column specifying whether a row is in or out of spec.

## The Simulator Menu

**Automatic Histogram Update** toggles histogram update, which sends changes to all histograms shown in the Profiler, so that histograms update with new simulated values when you drag distribution handles.

**Defect Profiler** shows the defect rate as an isolated function of each factor. This command is enabled when spec limits are available, as described below.

**Defect Parametric Profile** shows the defect rate as an isolated function of the parameters of each factor's distribution. It is enabled when the Defect Profiler is launched.

**Simulation Experiment** is used to run a designed simulation experiment on the locations of the factor distributions. A dialog appears, allowing you to specify the number of design points, the portion of the factor space to be used in the experiment, and which factors to include in the experiment. For factors not included in the experiment, the current value shown in the Profiler is the one used in the experiment.

The experimental design is a Latin Hypercube. The output has one row for each design point. The responses include the defect rate for each response with spec limits, and an overall defect rate. After the experiment, it would be appropriate to fit a Gaussian Process model on the overall defect rate, or a root or a logarithm of it.

A simulation experiment does not sample the factor levels from the specified distributions. As noted above, the design is a Latin Hypercube. At each design point, **N Runs** random draws are generated with the design point serving as the center of the random draws, and the shape and variability coming from the specified distributions.

**Spec Limits** shows or edits specification limits.

**N Strata** is a hidden option accessible by holding down the Shift key before clicking the Simulator popup menu. This option allows you to specify the number of strata in Normal Weighted. For more information also see “[Statistical Details](#),” p. 605.

**Set Random Seed** is a hidden option accessible by holding down the Shift key before clicking the Simulator popup menu. This option allows you to specify a seed for the simulation starting point. This enables the simulation results to be reproducible, unless the seed is set to zero. The seed is set to zero by default. If the seed is non-zero, then the latest simulation results are output if the **Make Table** button is clicked.

## Using Specification Limits

The profilers support specification limits on the responses, providing a number of features

- In the **Profiler**, if you don't have the **Response Limits** property set up in the input data table to provide desirability coordinates, JMP looks for a **Spec Limits** property and constructs desirability functions appropriate to those **Spec Limits**.
- If you use the Simulator to output simulation tables, JMP copies **Spec Limits** to the output data tables, making accounting for defect rates and capability indices easy.
- Adding **Spec Limits** enables a feature called the **Defect Profiler**.

In the following example, we assume that the following Spec Limits have been specified.

**Table 26.3** Spec Limits for Tiretread.jmp data table

Response	LSL	USL
Abrasion	110	
Modulus		2000
Elong	350	550
Hardness	66	74

To set these limits in the data table, highlight a column and select **Cols > Column Info**. Then, click the **Column Properties** button and select the **Spec Limits** property.

If you are already in the **Simulator** in a profiler, another way to enter them is to use the **Spec Limits** command in the **Simulator** pop-up menu.

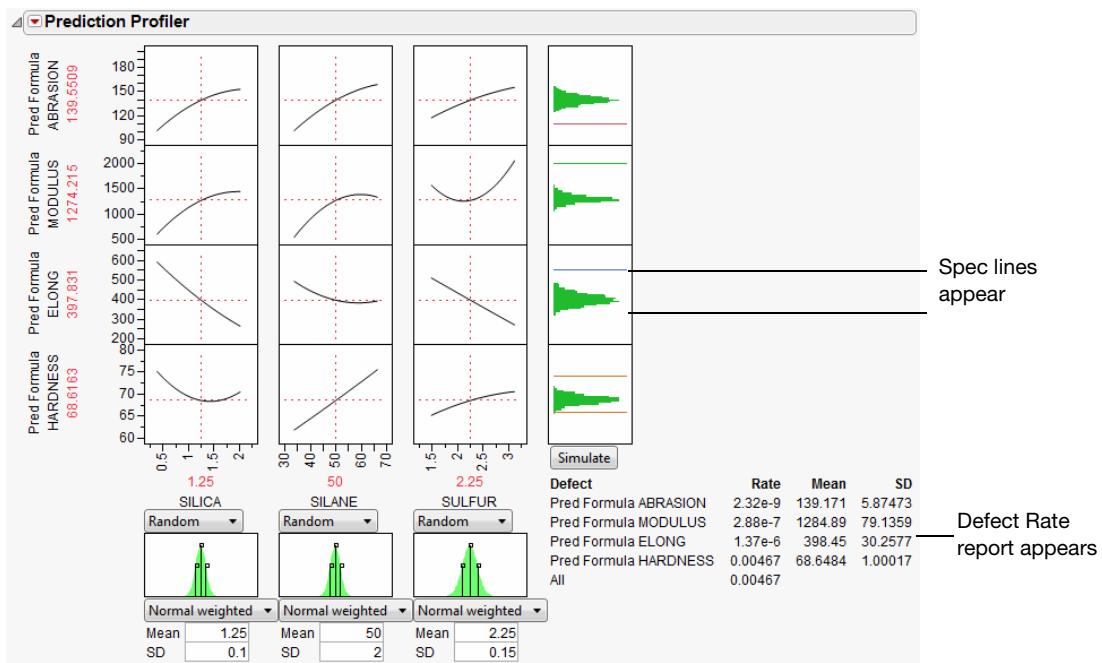
**Figure 26.45** Spec Limits

The dialog box has a title bar labeled "Spec Limits". It contains a table with four rows, each representing a response. The columns are labeled "Response", "LSL", and "USL". A "Save" button is located at the bottom right of the dialog. The data entries are:

Response	LSL	USL
Pred Formula ABRASION	110	.
Pred Formula MODULUS	.	2000
Pred Formula ELONG	350	550
Pred Formula HARDNESS	66	74

After entering the spec limits, they are incorporated into the profilers. Click the **Save** button if you want the spec limits saved back to the data table as a column property.

With these specification limits, and the distributions shown in Figure 26.42, click the **Simulate** button. Notice the spec limit lines in the output histograms.

**Figure 26.46** Spec Limits in the Prediction Profiler

Look at the histogram for Abrasion. The lower spec limit is far above the distribution, yet the **Simulator** is able to estimate a defect rate for it. This despite only having 5000 runs in the simulation. It can do this rare-event estimation when you use a **Normal weighted** distribution.

Note that the Overall defect rate is close to the defect rate for Hardness, indicating that most of the defects are in the Hardness variable.

To see this weighted simulation in action, click the **Make Table** button and examine the Weight column.

JMP generates extreme values for the later observations, using very small weights to compensate. Since the **Distribution** platform handles frequencies better than weights, there is also a column of frequencies, which is simply the weights multiplied by  $10^{12}$ .

The output data set contains a **Distribution** script appropriate to analyze the simulation data completely with a capability analysis.

## Simulating General Formulas

Though the profiler and simulator are designed to work from formulas stored in a model fit, they work for any formula that can be stored in a column. A typical application of simulation is to exercise financial models under certain probability scenarios to obtain the distribution of the objectives. This can be done in JMP—the key is to store the formulas into columns, set up ranges, and then conduct the simulation.

**Table 26.4** Factors and Responses for a Financial Simulation

Inputs (Factors)	Unit Sales	random uniform between 1000 and 2000
	Unit Price	fixed
	Unit Cost	random normal with mean 2.25 and std dev 0.1
Outputs (Responses)	Revenue	formula: Unit Sales*Unit Price
	Total Cost	formula: Unit Sales*Unit Cost + 1200
	Profit	formula: Revenue – Total Cost

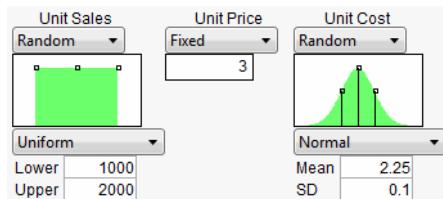
The following JSL script creates the data table below with some initial scaling data and stores formulas into the output variables. It also launches the Profiler.

```
dt = newTable("Sales Model");
dt<<newColumn("Unit Sales",Values({1000,2000}));
dt<<newColumn("Unit Price",Values({2,4}));
dt<<newColumn("Unit Cost",Values({2,2.5}));
dt<<newColumn("Revenue",Formula(:Unit Sales*:Unit Price));
dt<<newColumn("Total Cost",Formula(:Unit Sales*:Unit Cost + 1200));
dt<<newColumn("Profit",Formula(:Revenue-:Total Cost), Set Property("Spec
Limits",{LSL(0)}));
Profiler(Y(:Revenue,:Total Cost,:Profit), Objective Formula(Profit));
```

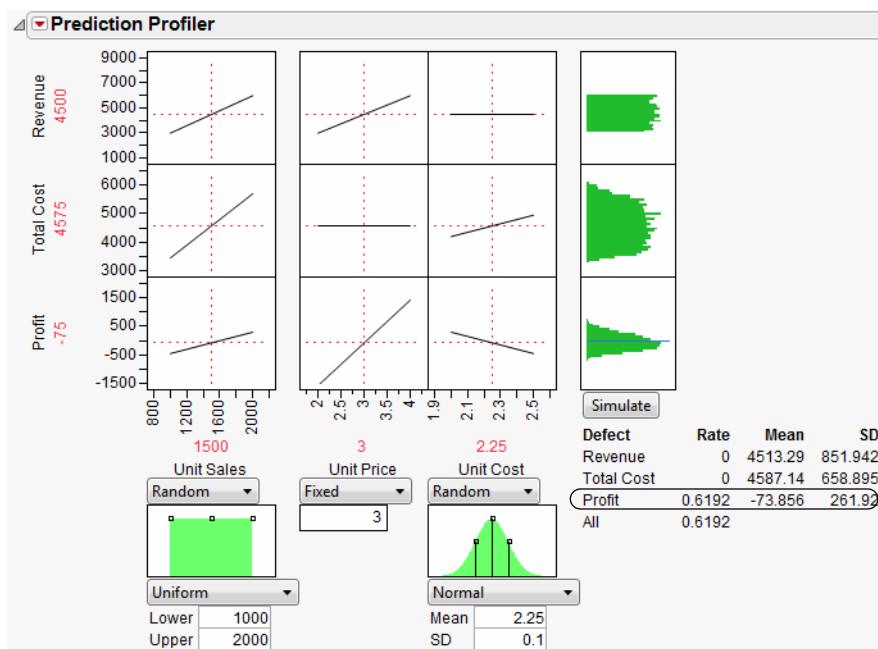
**Figure 26.47** Data Table Created from Script

Sales Model					
	Unit Sales	Unit Price	Unit Cost	Revenue	Total Cost
1	1000	2	2	2000	3200
2	2000	4	2.5	8000	6200

Once they are created, select the **Simulator** from the **Prediction Profiler**. Use the specifications from Table 26.4 “Factors and Responses for a Financial Simulation,” p. 577 to fill in the Simulator.

**Figure 26.48** Profiler Using the Data Table

Now, run the simulation which produces the following histograms in the **Profiler**.

**Figure 26.49** Simulator

It looks like we are not very likely to be profitable. By putting a lower specification limit of zero on Profit, the defect report would say that the probability of being unprofitable is 62%.

So we raise the Unit Price to \$3.25 and rerun the simulation. Now the probability of being unprofitable is down to about 21%.

**Figure 26.50** Results

Defect	Rate	Mean	SD
Revenue	0	4889.4	922.937
Total Cost	0	4587.14	658.895
Profit	0.2006	302.252	321.969
All	0.2006		

If unit price can't be raised anymore, you should now investigate lowering your cost, or increasing sales, if you want to further decrease the probability of being unprofitable.

## The Defect Profiler

The defect profiler shows the probability of an out-of-spec output defect as a function of each factor, while the other factors vary randomly. This is used to help visualize which factor's distributional changes the process is most sensitive to, in the quest to improve quality and decrease cost.

Specification limits define what is a defect, and random factors provide the variation to produce defects in the simulation. Both need to be present for a Defect Profile to be meaningful.

At least one of the Factors must be declared **Random** for a defect simulation to be meaningful, otherwise the simulation outputs would be constant. These are specified though the simulator Factor specifications.

**Important:** If you need to estimate very small defect rates, use **Normal weighted** instead of just **Normal**. This allows defect rates of just a few parts per million to be estimated well with only a few thousand simulation runs.

### About Tolerance Design

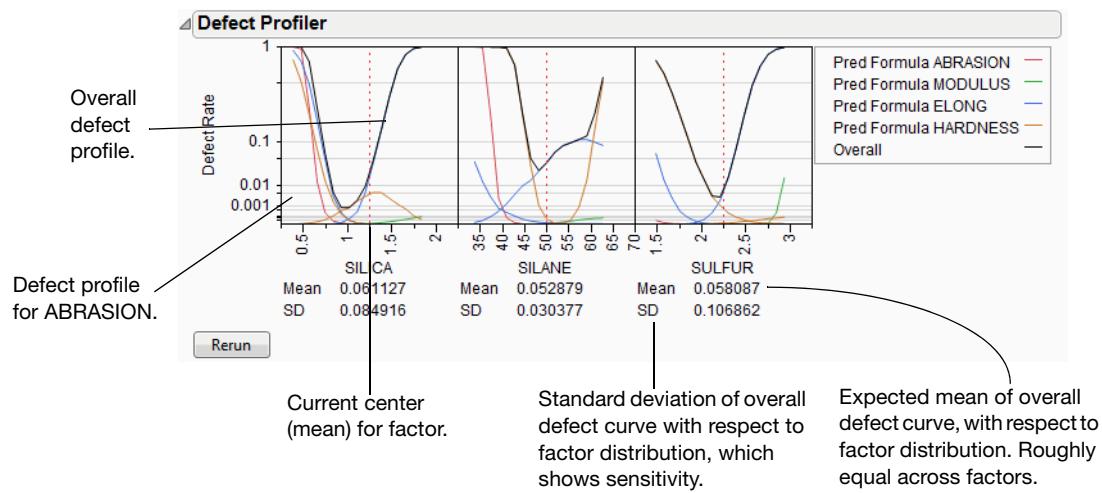
*Tolerance Design* is the investigation of how defect rates on the outputs can be controlled by controlling variability in the input factors.

The input factors have variation. Specification limits are used to tell the supplier of the input what range of values are acceptable. These input factors then go into a process producing outputs, and the customer of the outputs then judges if these outputs are within an acceptable range.

Sometimes, a Tolerance Design study shows that spec limits on input are unnecessarily tight, and loosening these limits results in cheaper product without a meaningful sacrifice in quality. In these cases, Tolerance Design can save money.

In other cases, a Tolerance Design study may find that either tighter limits or different targets result in higher quality. In all cases, it is valuable to learn which inputs the defect rate in the outputs are most sensitive to.

This graph shows the defect rate as a function of each factor as if it were a constant, but all the other factors varied according to their random specification. If there are multiple outputs with Spec Limits, then there is a defect rate curve color-coded for each output. A black curve shows the overall defect rate—this curve is above all the colored curves.

**Figure 26.51** Defect Profiler

## Graph Scale

Defect rates are shown on a cubic root scale, so that small defect rates are shown in some detail even though large defect rates may be visible. A log scale is not used because zero rates are not uncommon and need to be shown.

## Expected Defects

Reported below each defect profile plot is the mean and standard deviation (SD). The mean is the overall defect rate, calculated by integrating the defect profile curve with the specified factor distribution.

In this case, the defect rate that is reported below all the factors is estimating the same quantity, the rate estimated for the overall simulation below the histograms (*i.e.* if you clicked the **Simulate** button). Since each estimate of the rate is obtained in a different way, they may be a little different. If they are very different, you may need to use more simulation runs. In addition, check that the range of the factor scale is wide enough so the integration covers the distribution well.

The standard deviation is a good measure of the sensitivity of the defect rates to the factor. It is quite small if either the factor profile were flat, or the factor distribution has a very small variance. Comparing SD's across factors is a good way to know which factor should get more attention to reducing variation.

The mean and SD are updated when you change the factor distribution. This is one way to explore how to reduce defects as a function of one particular factor at a time. You can click and drag a handle point on the factor distribution, and watch the mean and SD change as you drag. However, changes are not updated across all factors until you click the **Rerun** button to do another set of simulation runs.

### Simulation Method and Details

Assume we want a defect profile for factor X1, in the presence of random variation in X2 and X3. A series of  $n=N$  Runs simulation runs is done at each of  $k$  points in a grid of equally spaced values of X1. ( $k$  is generally set at 17). At each grid point, suppose that there are  $m$  defects due to the specification limits. At that grid point, the defect rate is  $m/n$ . With normal weighted, these are done in a weighted fashion. These defect rates are connected and plotted as a continuous function of X1.

### Notes

**Recalculation** The profile curve is not recalculated automatically when distributions change, though the expected value is. It is done this way because the simulations could take a while to run.

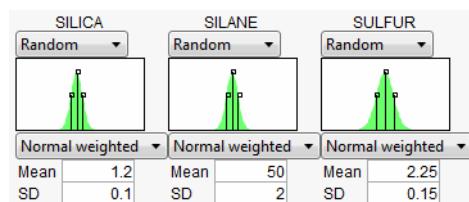
**Limited goals** Profiling does not address the general optimization problem, that of optimizing quality against cost, given functions that represent all aspects of the problem. This more general problem would benefit from a surrogate model and space filling design to explore this space and optimize to it.

**Jagged Defect Profiles** The defect profiles tend to get uneven when they are low. This is due to exaggerating the differences for low values of the cubic scale. If the overall defect curve (black line) is smooth, and the defect rates are somewhat consistent, then you are probably taking enough runs. If the Black line is jagged and not very low, then increase the number of runs. 20,000 runs is often enough to stabilize the curves.

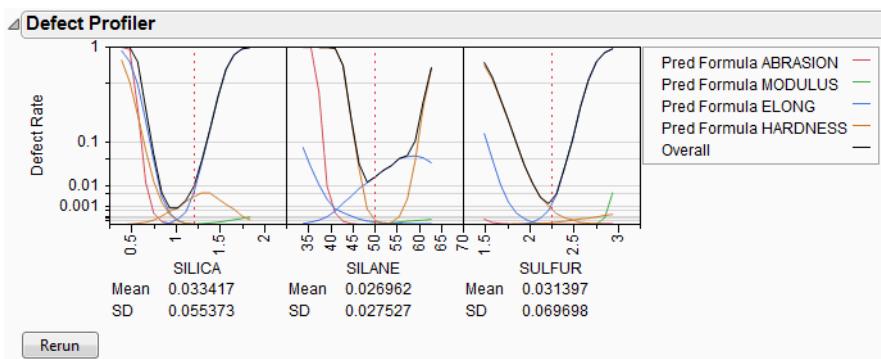
### Defect Profiler Example

To show a common workflow with the Defect profiler, we use Tiretread.jmp with the specification limits from Table 26.3. We also give the following random specifications to the three factors.

**Figure 26.52** Profiler



Select **Defect Profiler** to see the defect profiles. The curves, Means, and SDs will change from simulation to simulation, but will be relatively consistent.

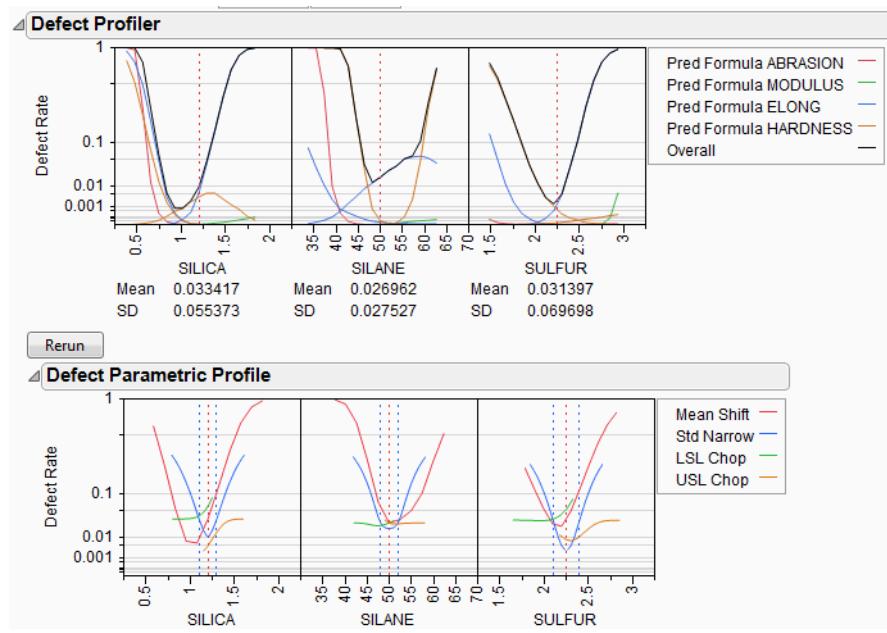
**Figure 26.53** Defect Profiler

The black curve on each factor shows the defect rate if you could fix that factor at the  $x$ -axis value, but leave the other features random.

Look at the curve for **SILICA**. As its values vary, its defect rate goes from the lowest 0.001 at **SILICA**=0.95, quickly up to a defect rate of 1 at **SILICA**=0.4 or 1.8. However, **SILICA** is itself random. If you imagine integrating the density curve of **SILICA** with its defect profile curve, you could estimate the average defect rate 0.033, also shown as the Mean for **SILICA**. This is estimating the overall defect rate shown under the simulation histograms, but by numerically integrating, rather than by the overall simulation. The Means for the other factors are similar. The numbers are not exactly the same. However, we now also get an estimate of the standard deviation of the defect rate with respect to the variation in **SILICA**. This value (labeled SD) is 0.055. The standard deviation is intimately related to the sensitivity of the defect rate with respect to the distribution of that factor.

Looking at the SDs across the three factors, we see that the SD for **SULFUR** is higher than the SD for **SILICA**, which is in turn much higher than the SD for **SILANE**. This means that to improve the defect rate, improving the distribution in **SULFUR** should have the greatest effect. A distribution can be improved in three ways: changing its mean, changing its standard deviation, or by chopping off the distribution by rejecting parts that don't meet certain specification limits.

In order to visualize all these changes, there is another command in the **Simulator** pop-up menu, **Defect Parametric Profile**, which shows how single changes in the factor distribution parameters affect the defect rate.

**Figure 26.54** Defect Parametric Profile

Let's look closely at the **SULFUR** situation. You may need to enlarge the graph to see more detail.

First, note that the current defect rate (0.03) is represented in four ways corresponding to each of the four curves.

For the red curve, **Mean Shift**, the current rate is where the red solid line intersects the vertical red dotted line. The Mean Shift curve represents the change in overall defect rate by changing the mean of SULFUR. One opportunity to reduce the defect rate is to shift the mean slightly to the left. If you use the crosshair tool on this plot, you see that a mean shift reduces the defect rate to about 0.02.

For the blue curve, **Std Narrow**, the current rate represents where the solid blue line intersects the two dotted blue lines. The Std Narrow curves represent the change in defect rate by changing the standard deviation of the factor. The dotted blue lines represent one standard deviation below and above the current mean. The solid blue lines are drawn symmetrically around the center. At the center, the blue line typically reaches a minimum, representing the defect rate for a standard deviation of zero. That is, if we totally eliminate variation in SULFUR, the defect rate is still around 0.003. This is much better than 0.03. If you look at the other Defect parametric profile curves, you can see that this is better than reducing variation in the other factors, something we suspected by the SD value for SULFUR.

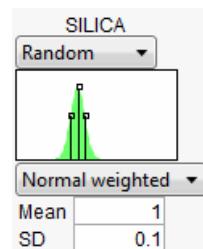
For the green curve, **LSL Chop**, there are no interesting opportunities in this example, because the green curve is above current defect rates for the whole curve. This means that reducing the variation by rejecting parts with too-small values for SULFUR will not help.

For the orange curve, **USL Chop**, there are good opportunities. Reading the curve from the right, the curve starts out at the current defect rate (0.03), then as you start rejecting more parts by decreasing the USL for

SULFUR, the defect rate improves. However, moving a spec limit to the center is equivalent to throwing away half the parts, which may not be a practical solution.

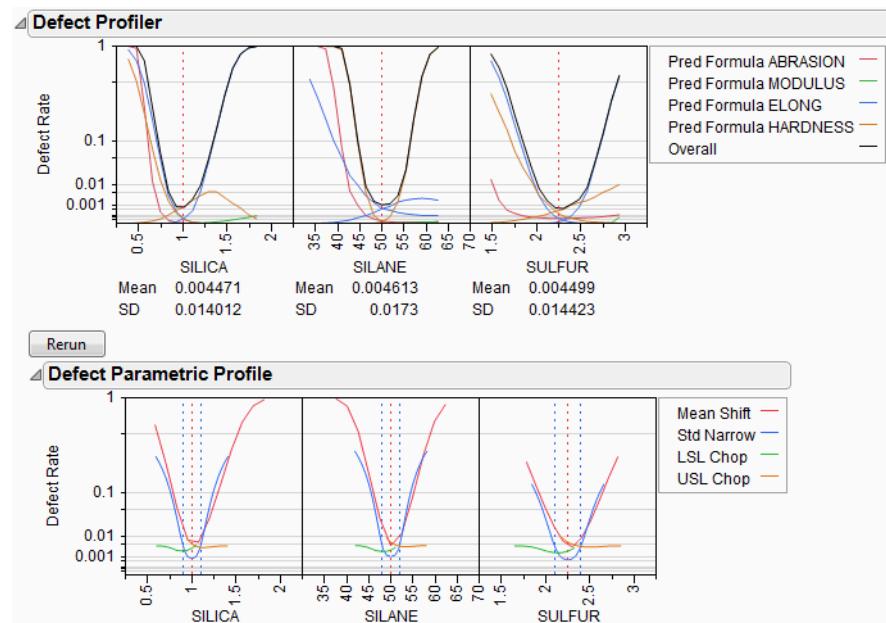
Looking at all the opportunities over all the factors, it now looks like there are two good options for a first move: change the mean of SILICA to about 1, or reduce the variation in SULFUR. Since it is generally easier in practice to change a process mean than process variation, the best first move is to change the mean of SILICA to 1.

**Figure 26.55** Adjusting the Mean of Silica



After changing the mean of SILICA, all the defect curves become invalid and need to be rerun. After clicking **Rerun**, we get a new perspective on defect rates.

**Figure 26.56** Adjusted Defect Rates



Now, the defect rate is down to about 0.004, much improved. Further reduction in the defect rate can occur by continued investigation of the parametric profiles, making changes to the distributions, and rerunning the simulations.

As the defect rate is decreased further, the mean defect rates across the factors become relatively less reliable. The accuracy could be improved by reducing the ranges of the factors in the **Profiler** a little so that it integrates the distributions better.

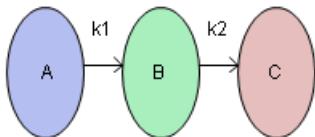
This level of fine-tuning is probably not practical, because the experiment that estimated the response surface is probably not at this high level of accuracy. Once the ranges have been refined, you may need to conduct another experiment focusing on the area that you know is closer to the optimum.

## Stochastic Optimization Example

This example is adapted from Box and Draper (1987) and uses **Stochastic Optimization.jmp**. A chemical reaction converts chemical “A” into chemical “B”. The resulting amount of chemical “B” is a function of reaction time and reaction temperature. A longer time and hotter temperature result in a greater amount of “B”. But, a longer time and hotter temperature also result in some of chemical “B” getting converted to a third chemical “C”. What reaction time and reaction temperature will maximize the resulting amount of “B” and minimize the amount of “A” and “C”? Should the reaction be fast and hot, or slow and cool?

---

**Figure 26.57** Chemical Reaction



---

The goal is to maximize the resulting amount of chemical “B”. One approach is to conduct an experiment and fit a response surface model for reaction yield (amount of chemical “B”) as a function of time and temperature. But, due to well known chemical reaction models, based on the Arrhenius laws, the reaction yield can be directly computed. The column **Yield** contains the formula for yield. The formula is a function of **Time** (hours) and reaction rates **k1** and **k2**. The reaction rates are a function of reaction **Temperature** (degrees Kelvin) and known physical constants  $\theta_1, \theta_2, \theta_3, \theta_4$ . Therefore, **Yield** is a function of **Time** and **Temperature**.

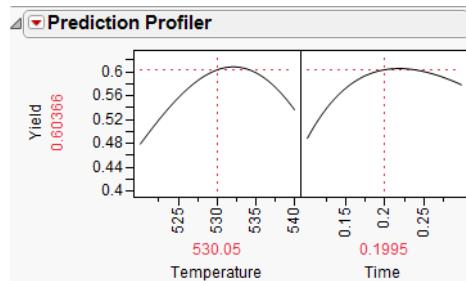
---

**Figure 26.58** Formula for Yield

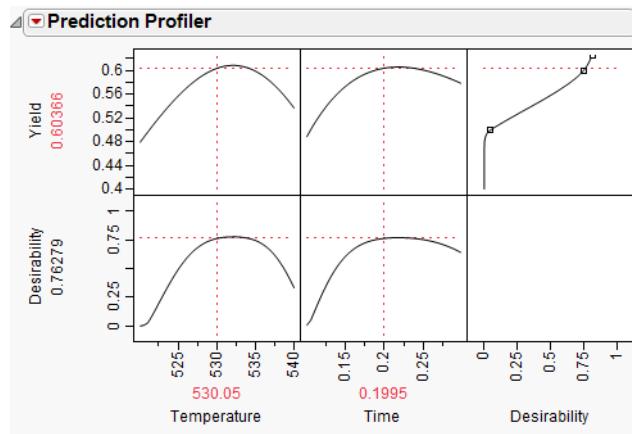
$$\frac{(k1 * (\text{Exp}(-k1 * \text{Time}) - \text{Exp}(-k2 * \text{Time})))}{(k2 - k1)}$$

---

You can use the **Profiler** to find the maximum **Yield**. Open **Stochastic Optimization.jmp** and run the attached script called **Profiler**. Profiles of the response are generated as follows.

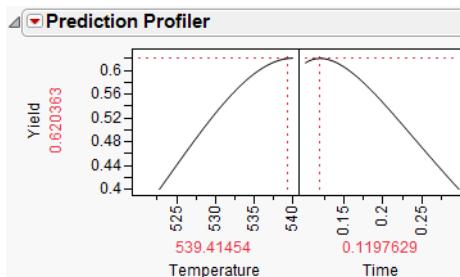
**Figure 26.59** Profile of Yield

To maximize Yield use a desirability function. See “Desirability Profiling and Optimization,” p. 545. One possible desirability function was incorporated in the script. To view the function choose **Desirability Functions** from the **Prediction Profiler** pop-up menu.

**Figure 26.60** Prediction Profiler

From this point on, the desirability function will be hidden.

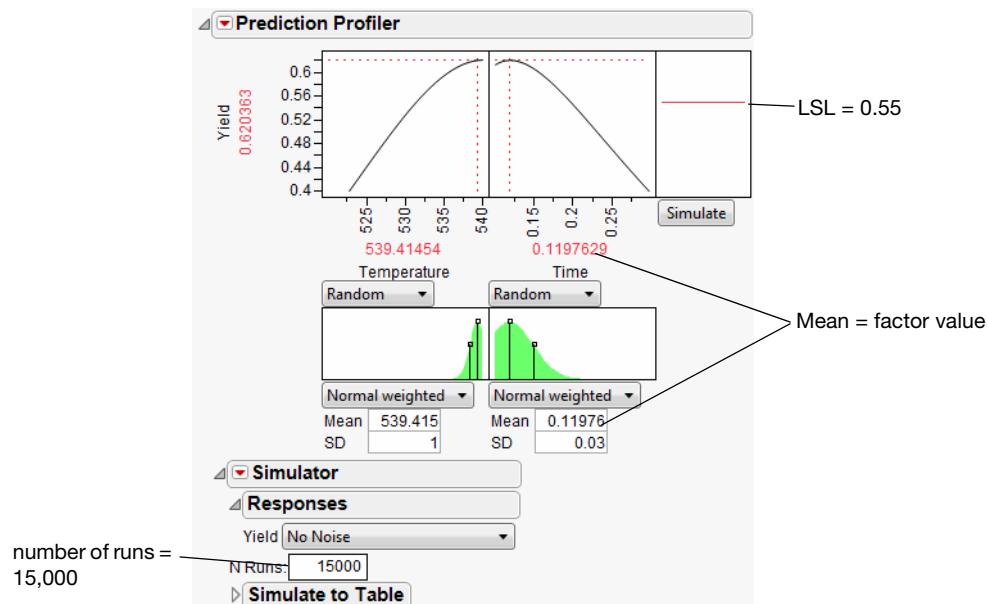
To maximize Yield, select **Maximize Desirability** from the **Prediction Profiler** pop-up menu. The Profiler then maximizes Yield and sets the graphs to the optimum value of Time and Temperature.

**Figure 26.61** Yield Maximum

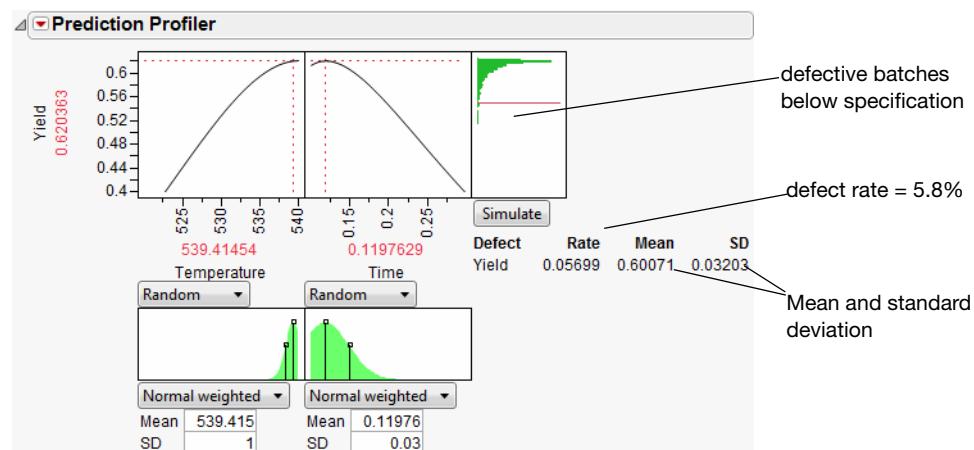
The maximum Yield is approximately 0.62 at a Time of 0.116 hours and Temperature of 539.94 degrees Kelvin, or hot and fast. [Your results may differ slightly due to random starting values in the optimization process. Optimization settings can be modified (made more stringent) by selecting **Maximization Options** from the **Prediction Profiler** pop-up menu. Decreasing the **Convergence Tolerance** will enable the solution to be reproducible.]

In a production environment, process inputs can't always be controlled exactly. What happens to Yield if the inputs (Time and Temperature) have random variation? Furthermore, if Yield has a spec limit, what percent of batches will be out of spec and need to be discarded? The Simulator can help us investigate the variation and defect rate for Yield, given variation in Time and Temperature.

Select **Simulator** from the **Prediction Profiler** pop-up menu. As shown in Figure 26.62, fill in the factor parameters so that Temperature is **Normal weighted** with standard deviation of 1, and Time is **Normal weighted** with standard deviation of 0.03. The **Mean** parameters default to the current factor values. Change the number of runs to 15,000. Yield has a lower spec limit of 0.55, set as a column property, and shows on the chart as a red line. If it doesn't show by default, enter it by selecting **Spec Limits** on the **Simulator** pop-up menu.

**Figure 26.62** Initial Simulator Settings

With the random variation set for the input factors, you are ready to run a simulation to study the resulting variation and defect rate for Yield. Click the **Simulate** button.

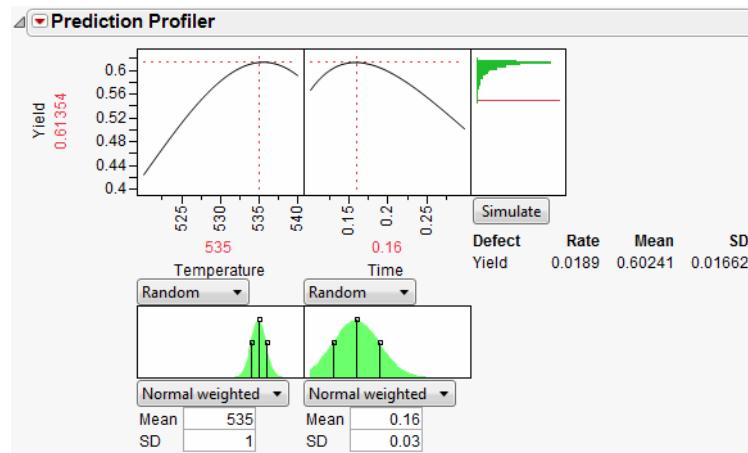
**Figure 26.63** Simulation Results

The predicted Yield is 0.62, but if the factors have the given variation, the average Yield is 0.60 with a standard deviation of 0.03.

The defect rate is about 5.7%, meaning that about 5.7% of batches are discarded. A defect rate this high is not acceptable.

What is the defect rate for other settings of Temperature and Time? Suppose you change the Temperature to 535, then set Time to the value that maximizes Yield? Enter settings as shown in Figure 26.64 then click **Simulate**.

**Figure 26.64** Defect Rate for Temperature of 535



The defect rate decreases to about 1.8%, which is much better than 5.8%. So, what you see is that the fixed (no variability) settings that maximize Yield are not the same settings that minimize the defect rate in the presence of factor variation.

By running a **Simulation Experiment** you can find the settings of Temperature and Time that minimize the defect rate. To do this you simulate the defect rate at each point of a Temperature and Time design, then fit a predictive model for the defect rate and minimize it.

Before running the Simulation Experiment, save the factor settings that maximize Yield so you can reference them later. To do this, re-enter the factor settings (Mean and SD) from Figure 26.62 and select **Factor Settings > Remember Settings** from **Prediction Profiler** pop-up menu. A dialog prompts you to name the settings then click **OK**. The settings are appended to the report window.

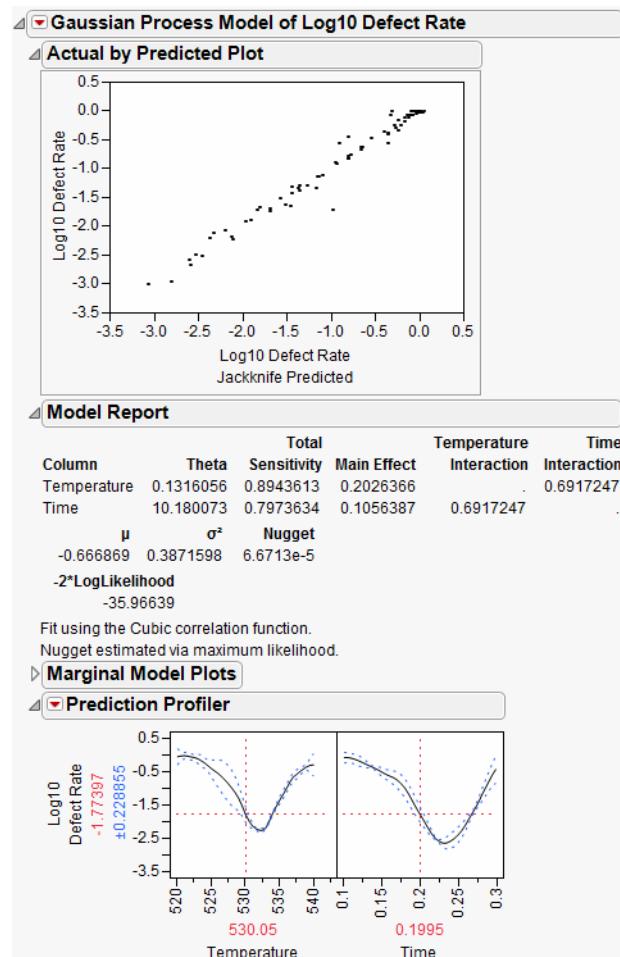
**Figure 26.65** Remembered Settings

Remembered Settings				
Setting	Temperature	Time	Yield	Desirability
Max Yield	539.415	0.11976	0.6203633	.

Select **Simulation Experiment** from the **Simulator** pop-up menu. Enter 80 runs, and 1 to use the whole factor space in the experiment. A Latin Hypercube design with 80 design points is chosen within the specified factor space, and **N Runs** random draws are taken at each of the design points. The design point are the center of the random draws, and the shape and variance of the random draws coming from the factor distributions.

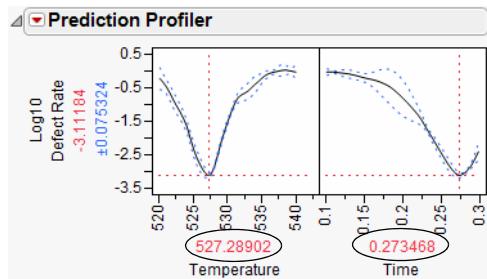
A table is created with the results of the experiment. The Overall Defect Rate is given at each design point. You can now fit a model that predicts the defect rate as a function of **Temperature** and **Time**. To do this, run the attached **Gaussian Process** script and wait for the results. The results are shown below. Your results will be slightly different due to the random draws in the simulation.

**Figure 26.66** Results of Gaussian Process Model Fit



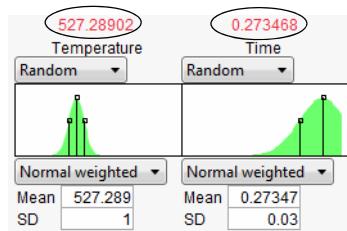
The **Gaussian Process** platform automatically starts the **Prediction Profiler**. The desirability function is already set up to minimize the defect rate. To find the settings of Temperature and Time that minimizes the defect rate, select **Maximize Desirability** from the **Prediction Profiler** pop-up menu.

**Figure 26.67** Settings for Minimum Defect Rate

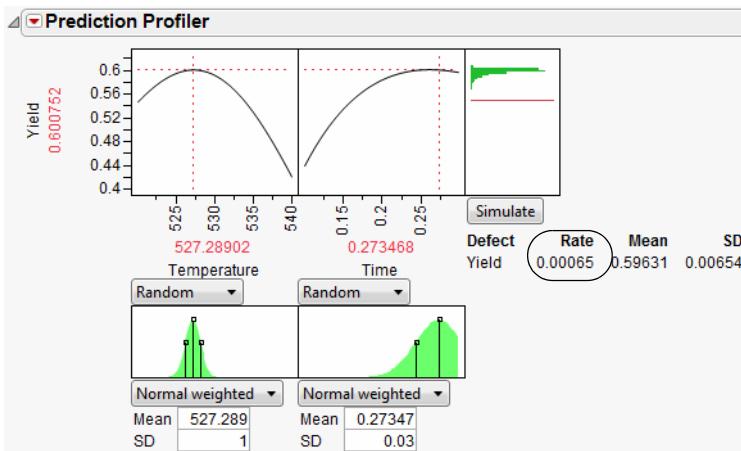


The settings that minimize the defect rate are approximately Temperature = 527 and Time = 0.27. Select **Factor Settings > Copy Settings Script** from the **Prediction Profiler** pop-up menu. Return to the original **Profiler** report window and select **Factor Settings > Paste Settings Script**. This sets **Temperature** and **Time** to those settings that minimize the defect rate. Use **Remember Settings** as before to save these new settings.

**Figure 26.68** Minimum Defect Settings

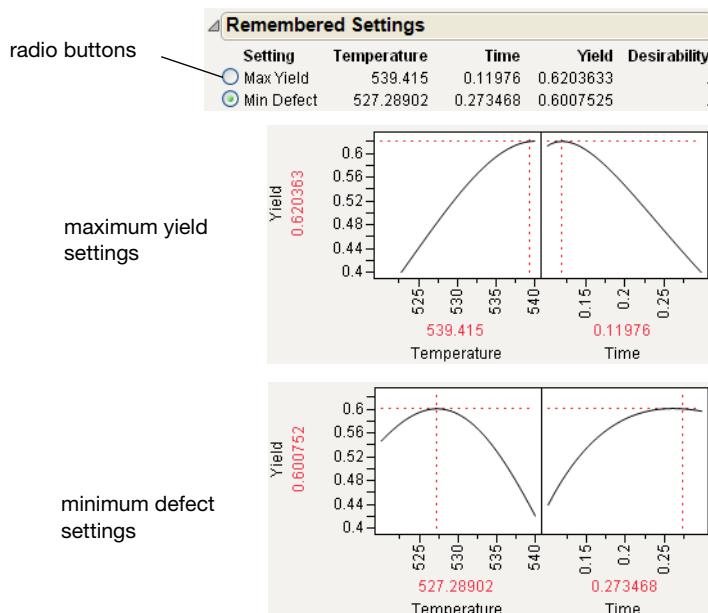


With the new settings in place, click the **Simulate** button to estimate the defect rate at the new settings.

**Figure 26.69** Lower Defect Rate

At the new settings the defect rate is 0.065%, much better than the 5.7% for the settings that maximize Yield. That is a reduction of about 100x. Recall the average Yield from the first settings is 0.60 and the new average is 0.59. The decrease in average Yield of 0.01 is very acceptable when the defect rate decreases by 100x.

Because we saved the settings using **Remember Settings**, we can easily compare the old and new settings by selecting the radio buttons.

**Figure 26.70** Settings Comparison

The chemist now knows what settings to use for a quality process. If the factors have no variation, the settings for maximum Yield are hot and fast. But, if the process inputs have variation similar to what we've simulated, the settings for maximum Yield produce a high defect rate. Therefore, to minimize the defect rate in the presence of factor variation, the settings should be cool and slow.

## Noise Factors (Robust Engineering)

Robust process engineering enables you to produce acceptable products reliably, despite variation in the process variables. Even when your experiment has controllable factors, there is a certain amount of uncontrollable variation in the factors that affects the response. This is called *transmitted variation*. Factors with this variation are called *noise factors*. Some factors you can't control at all, like environmental noise factors. Some factors can have their mean controlled, but not their standard deviation. This is often the case for intermediate factors that are output from a different process or manufacturing step.

A good approach to making the process robust is to match the target at the flattest place of the noise response surface so that the noise has little influence on the process. Mathematically, this is the value where the first derivatives of each response with respect to each noise factor are zero. JMP, of course, computes the derivatives for you.

To analyze a model with noise factors,

- Fit the appropriate model (using, for example, the **Fit Model** platform).
- Save the model to the data table with the **Save > Prediction Formula** command.

- Launch the **Profiler** (from the **Graph** menu).
- Assign the prediction formula to the **Y, Prediction Formula** role and the noise factors to the **Noise Factor** role.
- Click **OK**.

The resulting profiler shows response functions and their appropriate derivatives with respect to the noise factors, with the derivatives set to have maximum desirability at zero.

- Select **Maximize Desirability** from the **Profiler** menu.

This finds the best settings of the factors, balanced with respect to minimizing transmitted variation from the noise factors.

### **Example**

As an example, use the *Tiretread.jmp* sample data set. This data set shows the results of a tire manufacturer's experiment whose objective is to match a target value of HARDNESS= 70 based on three factors: SILICA, SILANE, and SULFUR content. Suppose the SILANE and SULFUR content are easily (and precisely) controllable, but SILICA expresses variability that is worth considering.

For comparison, first optimize the factors for hardness without considering variation from the noise factor.

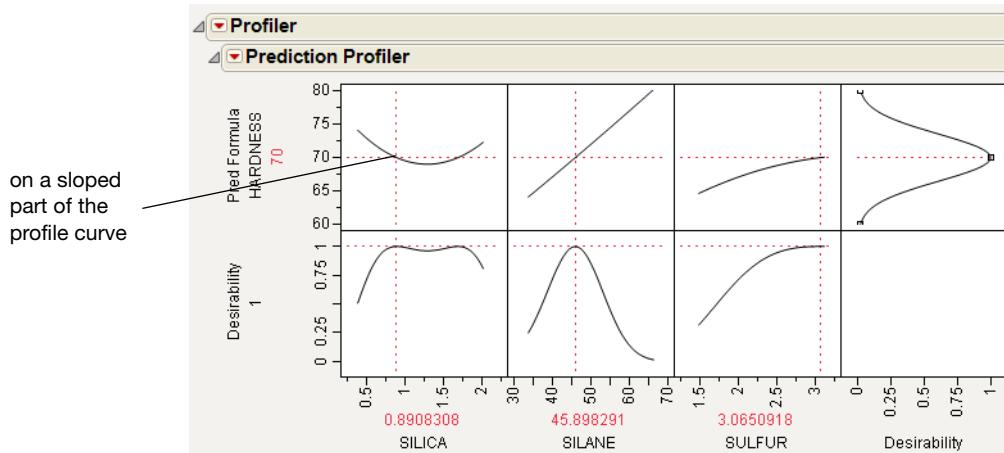
- Select **Graph > Profiler** to launch the **Profiler**.
- Assign Pred Formula HARDNESS to the **Y, Prediction Formula** role. Click **OK**.
- Select **Desirability Functions** in the Prediction Profiler menu.
- Double-click in the Desirability plot to open the Response Goal window. Change the pull-down menu to **Match Target**.
- Select **Maximize Desirability** to find the optimum factor settings for our target value of HARDNESS.

We get the following **Profiler** display. Notice that the SILICA factor's optimum value is on a sloped part of a profile curve. This means that variations in SILICA are transmitted to become variations in the response, HARDNESS.

---

**Note:** You may get different results from these because different combinations of factor values can all hit the target.

---

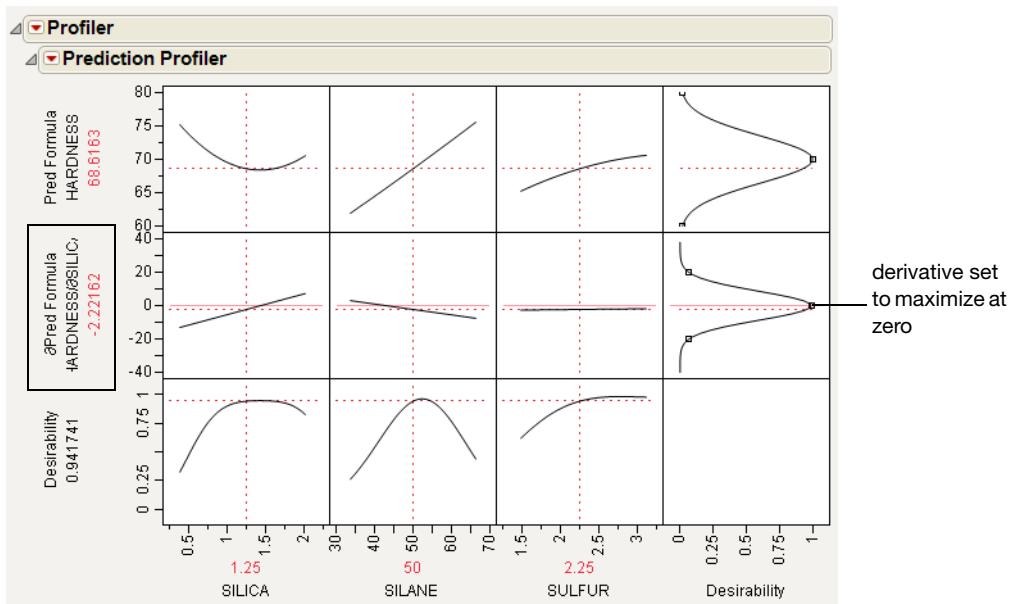
**Figure 26.71** Maximizing Desirability for HARDNESS

Now, we would like to not just optimize for a specific target value of HARDNESS, but also be on a flat part of the curve with respect to Silica. So, repeat the process and add SILICA as a noise factor.

1. Select **Graph > Profiler**.
2. Select **Pred Formula HARDNESS** and click **Y, Prediction Formula**.
3. Select **SILICA** and click **Noise Factors**.
4. Click **OK**.
5. Change the **Pred Formula Hardness** desirability function as before.

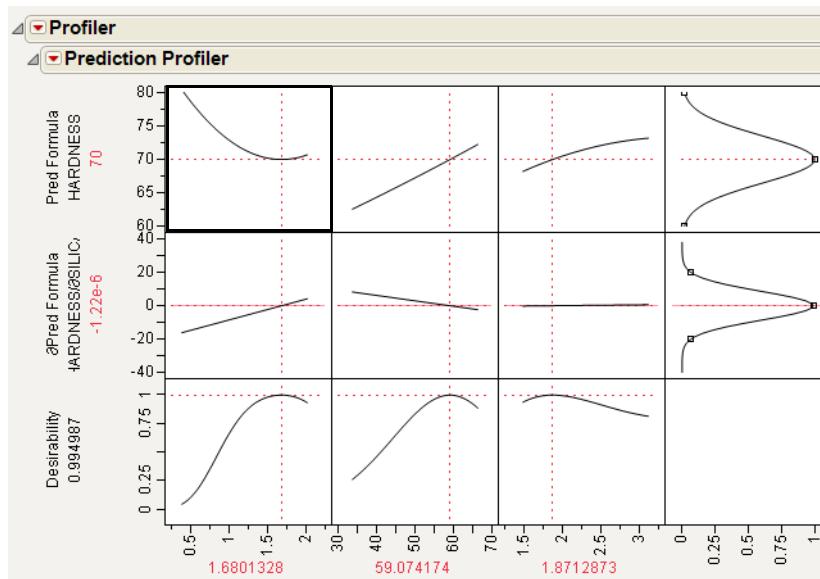
The resulting profiler has the appropriate derivative of the fitted model with respect to the noise factor, set to be maximized at zero, its flattest point.

**Figure 26.72** Derivative of the Prediction Formula with Respect to Silica



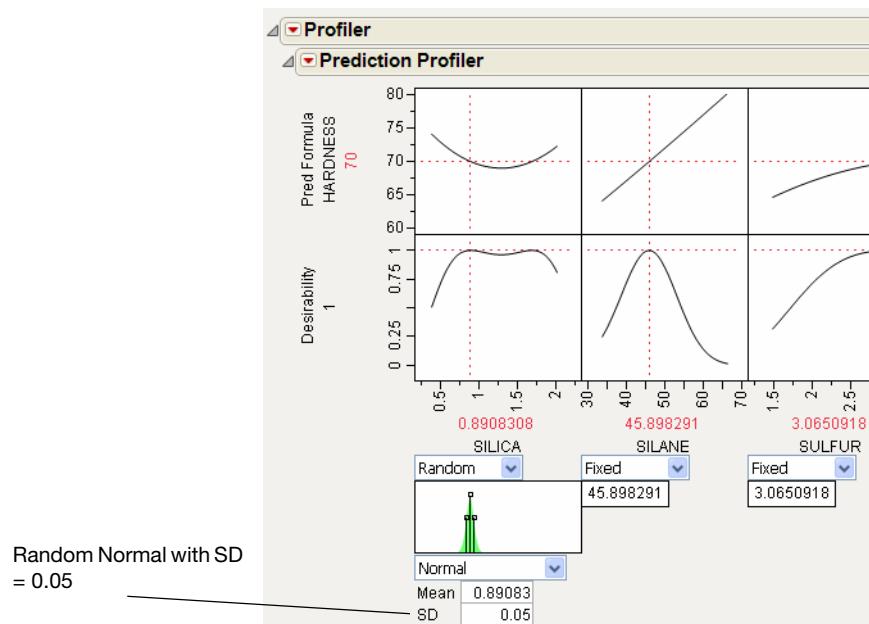
6. Select **Maximize Desirability** to find the optimum values for the process factor, balancing for the noise factor.

This time, we have also hit the targeted value of HARDNESS, but our value of SILICA is on its flatter region. This means variation in SILICA will not transmit as much variation to HARDNESS.

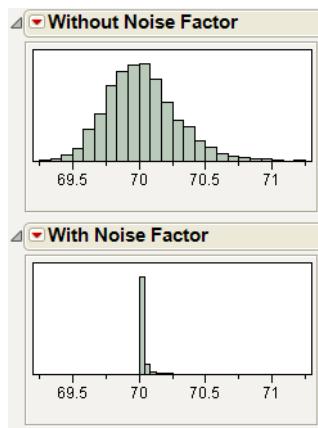
**Figure 26.73** Maximize Desirability

You can easily see the effect this has on the variance of the predictions by following these steps for each profiler (one without the noise factor, and one with the noise factor):

1. Select **Simulator** from the platform menu.
2. Assign SILICA to have a random Normal distribution with a standard deviation of 0.05.

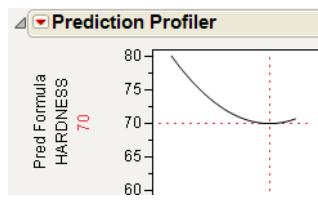
**Figure 26.74** Setting a Random Normal Distribution

3. Click **Simulate**.
  4. Click the **Make Table** button under the **Simulate to Table** node.
- Doing these steps for both the original and noise-factor-optimal simulations results in two similar data tables, each holding a simulation. In order to make two comparable histograms of the predictions, we need the two prediction columns in a single data table.
5. Copy the **Pred Formula HARDNESS** column from one of the simulation tables into the other table. They must have different names, like **Without Noise Factor** and **With Noise Factor**.
  6. Select **Analyze > Distribution** and assign both prediction columns as **Y**.
  7. When the histograms appear, select **Uniform Scaling** from the **Distribution** main title bar.

**Figure 26.75** Comparison of Distributions With and Without Noise Factors

The histograms show that there is much more variation in Hardness when the noise factor was not included in the analysis.

It is also interesting to note the shape of the histogram when the noise factor was included. In the comparison histograms above, note that the **With Noise Factor** distribution only has data trailing off in one direction. The predictions are skewed because Hardness is at a minimum with respect to SILICA, as shown in Figure 26.76. Therefore, variation in SILICA can only make HARDNESS increase. When the non-robust solution is used, the variation could be transmitted either way.

**Figure 26.76** Profiler Showing the Minima of HARDNESS by SILICA

### Other Platforms

Noise factor optimization is also available in the **Contour Profiler**, **Custom Profiler**, and **Mixture Profiler**.

---

## Profiling Models Stored in Excel

The JMP Add-In for Excel lets you use the JMP Profiler to visualize models (formulas) stored in Excel worksheets. The add-in is automatically installed when you install JMP.

## The Excel Model

An Excel model is simply one or more Excel formulas. Each formula must be a function of one or more other cells. As a simple example of an Excel formula, open the Demand.xls file from the Sample Import Data folder, usually located at C:\Program Files\SAS\JMP\9\Support Files English.

**Figure 26.77** Demand Model in Excel

JMP_Profiler						
Label	Inputs	Min	Max	Mean		
Amount Stocked	5	1	10	5		
Demand	6	1	10	1		
Air Freight	150	25	175	150		
Expiration Cost	50	25	100	50		
Label	Outputs	Min	Max	Mean		
Overall Cost	150	0	1000	100		

The formula is in cell B8, and is a calculation of the Overall Cost associated with having different amounts of product in stock. The formula can be seen in the Formula Bar, and is a function of four cells:

- Amount Stocked is the amount of product in stock.
- Demand is the customer demand for the product.
- Air Freight is the cost per unit to ship additional product by air when the demand exceeds the amount in stock.
- Expiration Cost is the cost per unit of disposing of unused product when the demand is less than the amount in stock.

The calculations of the formula are as follows:

- If Amount Stocked is less than Demand, then the company has to ship additional units, at a cost of (Demand-Amount Stocked) x Air Freight. For example, if the demand is 8, but the company has only 6 in stock, then it has to ship 8-6=2 units at a cost of  $2 \times 150 = 300$ .
- If Amount Stocked is greater than Demand, then the company has to dispose of unused product, at a cost of (Amount Stocked-Demand) x Expiration Cost. For example, if the demand is 5, but the company has 8 in stock, then it has to dispose of 8-5=3 units at a cost of  $3 \times 50 = 150$ .
- If Amount Stocked is equal to Demand, then there is no shipping cost or disposal cost.
- There is never both a shipping cost and a disposal cost at the same time.

Using the model in Excel, you can get the cost for only a given set of inputs at once. It is difficult to visualize how changing the value of one input affects the output. You can choose a different combination of the inputs to see how the cost is affected, but doing so for many combinations can take a long time.

Use the JMP Profiler to simultaneously see the effect of all inputs on the output. Also, you can quickly simulate a range of input combinations to see the resulting range of output values.

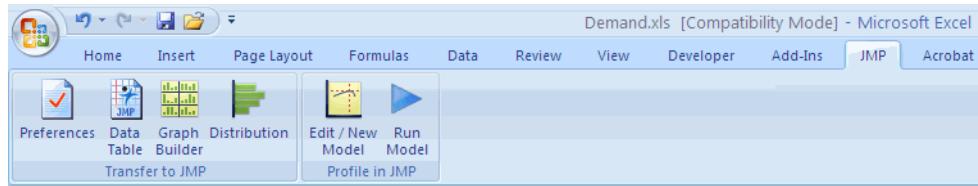
## Using the JMP Add-In for Profiling

The JMP Add-In for Excel is installed in the following areas of Excel, depending on the version of Office that you have:

- In Excel 2010, all options are on the JMP ribbon.
- In Excel 2007, all options are on the JMP ribbon. (See Figure 26.78.)
- In Excel 2003, the menu is in the toolbar on a JMP button that produces a menu of options.

---

**Figure 26.78** JMP Add-In for Excel for Windows 2007



Profiling an Excel model is a two step process:

1. Use the **Edit / New Model** button to enter information about the model that JMP needs. This needs to be done only once per model. For details, see “[Entering the Model Information](#),” p. 601.
2. Click the **Run Model** button to launch the JMP Profiler and run the Excel model. For details, see “[Running the JMP Profiler](#),” p. 603.

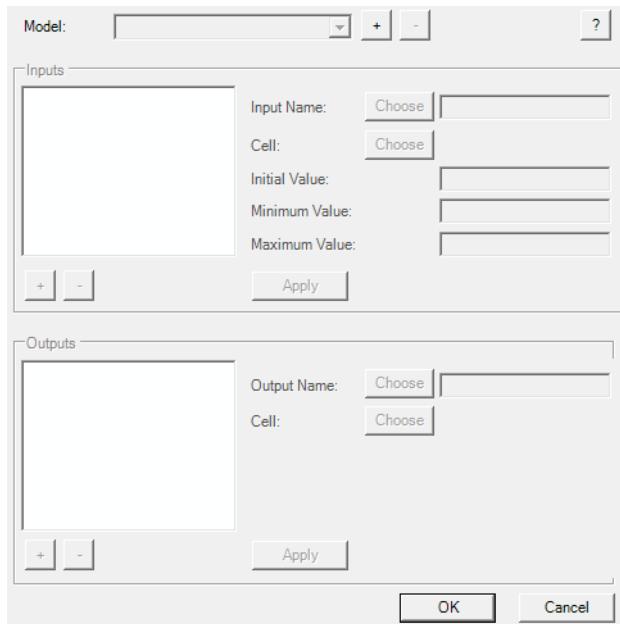
---

**Note:** The Preferences, Data Table, Graph Builder, and Distribution buttons are not needed to profile an Excel model. Full details about these features are given in the *Using JMP* book.

---

### **Entering the Model Information**

Before you can profile an Excel model, you must provide the information about the model that JMP needs. To do so, click the **Edit / New Model** button to open the Define Model window shown in Figure 26.79.

**Figure 26.79** Define Model Window

### Start and Name the Model

1. Click the + button at the top of the window.
2. Enter the name of the model.

### Add the Inputs

1. Click the + button in the Inputs section.
2. Enter the name in the **Input Name** box, or click **Choose** to select a cell from the worksheet.
3. Click **Choose** next to Cell to select the cell that is an input to the formula.
4. Change the **Minimum**, **Maximum**, and **Initial** values, if you need to. These values define the initial plot ranges shown in the profiler.
5. Click **Apply** in the Inputs section.
6. Repeat steps 1-5 until you are finished adding inputs.

### Add the Outputs

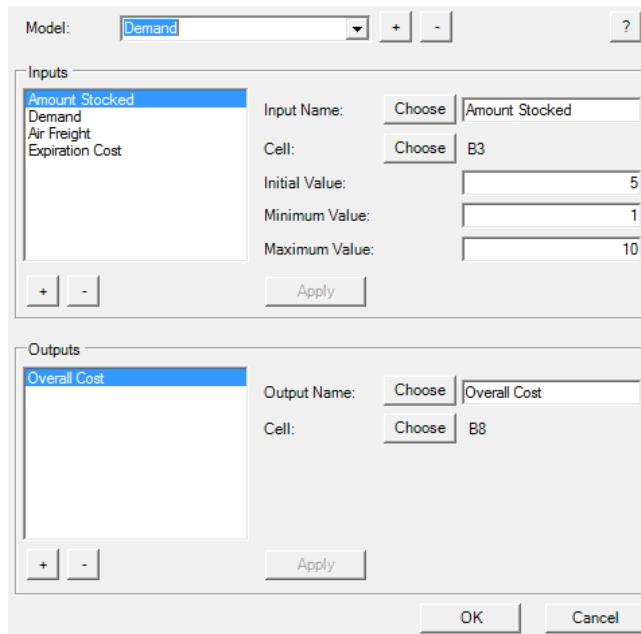
1. Click the + button in the Outputs section.
2. Enter the name in the **Output Name** box, or click **Choose** to select a cell from the worksheet.
3. Click **Choose** next to Cell to select the cell that contains the formula.
4. Click **Apply** in the Outputs section.

5. Repeat steps 1-4 until you are finished adding outputs.

Figure 26.80 shows a completed Define Model window.

---

**Figure 26.80** Completed Define Model Window



---

After all information is entered for the inputs and outputs, click **OK**.

---

**Note:** When you save the .xls file in Excel 2007, you might see a compatibility error. If so, click **Continue** to save the file.

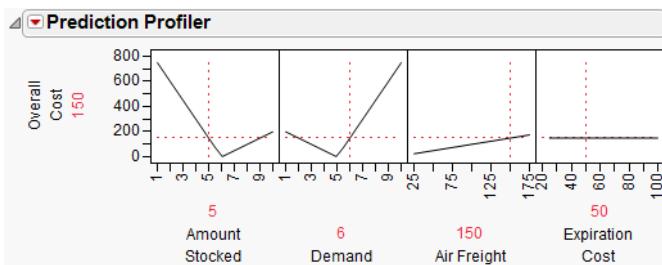
### **Running the JMP Profiler**

Follow these steps to run the model in the JMP Profiler.

1. Click **Run Model**.
2. Select the model that you want to use from the Model list.
3. Click **Profile in JMP**

As shown in Figure 26.81, the model is run in the Profiler.

**Figure 26.81** Example of the Profiler Using Excel Models



JMP runs a hidden copy of Excel in the background to drive all the Profiler calculations. Using a hidden copy ensures that the original Excel spreadsheet is not changed.

The Default N Levels option on the red-triangle menu affects the resolution of the profile lines. This option defaults to 17 when the Profiler runs a model stored in Excel, and to 41 when the model is stored directly in JMP. If the same model is stored in both Excel and JMP, the profile lines can appear different when the models are profiled. Increasing this value when using the Excel Profiler makes it slower.

## Using the Excel Profiler From JMP

Once an Excel file has the model inputs and outputs defined, you can profile the model from within JMP.

1. Select **Graph > Excel Profiler**.
2. Locate the Excel file containing the model and click **Open**.
3. If the Excel file contains multiple models, you are prompted to select the model that you want to Profile.

The Excel Profiler is also scriptable:

```
Excel Profiler( "path to workbook", <"model name"> ) ;
```

If more than one model exists, and no model is specified, a window with the list of available models appears.

For more information about scripting the Excel Profiler, see the *Scripting Guide*.

---

## Fit Group

For the REML and Stepwise personalities of the Fit Model platform, if models are fit to multiple  $Y$ 's, the results are combined into a **Fit Group** report. This enables the different  $Y$ 's to be profiled in the same Profiler. The Fit Group red-triangle menu has options for launching the joint Profiler. Profilers for the individual  $Y$ 's can still be used in the respective Fit Model reports.

Fit Group reports are also created when a By variable is specified for a Stepwise analysis. This allows for the separate models to be profiled in the same Profiler.

The Fit Group scripting command can be used to fit models in different platforms, and have the individual models profiled in the Profiler. For more details, see the *Scripting Guide*.

## Statistical Details

### Normal Weighted Distribution

JMP uses the multivariate radial strata method for each factor that uses the **Normal Weighted** distribution. This seems to work better than a number of Importance Sampling methods, as a multivariate Normal Integrator accurate in the extreme tails.

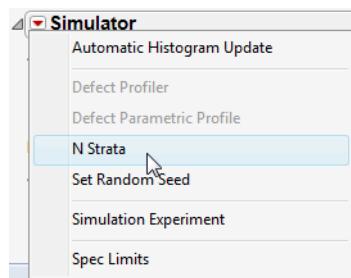
First, define strata and calculate corresponding probabilities and weights. For  $d$  random factors, the strata are radial intervals as follows.

**Table 26.5** Strata Intervals

Strata Number	Inside Distance	Outside Distance
0	0	$\sqrt{d}$
1	$\sqrt{d}$	$\sqrt{d + \sqrt{2}d}$
2	$\sqrt{d + \sqrt{2}d}$	$\sqrt{d + 2\sqrt{2}d}$
$i$	$\sqrt{d + (i - 1)\sqrt{2}d}$	$\sqrt{d + i\sqrt{2}d}$
$N_{Strata} - 1$	previous	$\infty$

The default number of strata is 12. To change the number of strata, a hidden command **N Strata** is available if you hold the Shift key down while clicking on the red triangle next to **Simulator**.

**Figure 26.82** Showing the N Strata Menu Option



Increase the sample size as needed to maintain an even number of strata.

For each simulation run,

1. Select a strata as  $\text{mod}(i - 1, N_{\text{Strata}})$  for run  $i$ .
2. Determine a random  $n$ -dimensional direction by scaling multivariate Normal (0,1) deviates to unit norm.
3. Determine a random distance using a chi-square quantile appropriate for the strata of a random uniform argument.
4. Scale the variates so the norm is the random distance.
5. Scale and re-center the variates individually to be as specified for each factor.

The resulting factor distributions are multivariate normal with the appropriate means and standard deviations when estimated with the right weights. Note that you cannot use the Distribution standard deviation with weights, because it does not estimate the desired value. However, multiplying the weight by a large value, like  $10^{12}$ , and using that as a Freq value results in the correct standard deviation.

# Appendix **A**

## **Statistical Details** **Models in JMP**

---

This appendix discusses the different types of response models, their factors, their design coding, and parameterization. It also includes many other details of methods described in the main text.

The JMP system fits linear models to three different types of response models that are labeled continuous, ordinal, and nominal. Many details on the factor side are the same between the different response models, but JMP only supports graphics and marginal profiles on continuous responses—not on ordinal and nominal.

Different computer programs use different design-matrix codings, and thus parameterizations, to fit effects and construct hypothesis tests. JMP uses a different coding than the GLM procedure in the SAS system, although in most cases JMP and SAS GLM procedure produce the same results. The following sections describe the details of JMP coding and highlight those cases when it differs from that of the SAS GLM procedure, which is frequently cited as the industry standard.

# Contents

The Response Models . . . . .	609
Continuous Responses . . . . .	609
Nominal Responses . . . . .	609
Ordinal Responses . . . . .	611
The Factor Models . . . . .	612
Continuous Factors . . . . .	612
Nominal Factors . . . . .	613
Ordinal Factors . . . . .	622
The Usual Assumptions . . . . .	628
Assumed Model . . . . .	628
Relative Significance . . . . .	628
Multiple Inferences . . . . .	629
Validity Assessment . . . . .	629
Alternative Methods . . . . .	629
Key Statistical Concepts . . . . .	630
Uncertainty, a Unifying Concept . . . . .	630
The Two Basic Fitting Machines . . . . .	631
Leverage Plot Details . . . . .	633
Multivariate Details . . . . .	636
Multivariate Tests . . . . .	636
Approximate F-Test . . . . .	637
Canonical Details . . . . .	637
Discriminant Analysis . . . . .	638
Power Calculations . . . . .	639
Computations for the LSV . . . . .	639
Computations for the LSN . . . . .	640
Computations for the Power . . . . .	640
Computations for Adjusted Power . . . . .	640
Inverse Prediction with Confidence Limits . . . . .	641
Details of Random Effects . . . . .	642

---

## The Response Models

JMP fits linear models to three different kinds of responses: continuous, nominal, and ordinal. The models and methods available in JMP are practical, are widely used, and suit the need for a general approach in a statistical software tool. As with all statistical software, you are responsible for learning the assumptions of the models you choose to use, and the consequences if the assumptions are not met. For more information see “[The Usual Assumptions](#),” p. 628 in this chapter.

### Continuous Responses

When the response column (column assigned the Y role) is continuous, JMP fits the value of the response directly. The basic model is that for each observation,

$$Y = (\text{some function of the } X\text{'s and parameters}) + \text{error}$$

Statistical tests are based on the assumption that the error term in the model is normally distributed.

#### **Fitting Principle for Continuous Response**

The Fitting principle is called *least squares*. The least squares method estimates the parameters in the model to minimize the sum of squared errors. The errors in the fitted model, called *residuals*, are the difference between the actual value of each observation and the value predicted by the fitted model.

The least squares method is equivalent to the maximum likelihood method of estimation if the errors have a normal distribution. This means that the analysis estimates the model that gives the most likely residuals. The log-likelihood is a scale multiple of the sum of squared errors for the normal distribution.

#### **Base Model**

The simplest model for continuous measurement fits just one value to predict all the response values. This value is the estimate of the *mean*. The mean is just the arithmetic average of the response values. All other models are compared to this base model.

### Nominal Responses

Nominal responses are analyzed with a straightforward extension of the logit model. For a binary (two-level) response, a logit response model is

$$\log\left(\frac{P(y=1)}{P(y=2)}\right) = X\beta$$

which can be written

$$P(y=1) = F(X\beta)$$

where  $F(x)$  is the cumulative distribution function of the standard logistic distribution

$$F(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

For  $r$  response levels, JMP fits the probabilities that the response is one of  $r$  different response levels given by the data values. The probability estimates must all be positive. For a given configuration of  $X$ 's, the probability estimates must sum to 1 over the response levels. The function that JMP uses to predict probabilities is a composition of a linear model and a multi-response logistic function. This is sometimes called a *log-linear* model because the logs of ratios of probabilities are linear models. JMP relates each response probability to the  $r$ th probability and fit a separate set of design parameters to these  $r - 1$  models.

$$\log\left(\frac{P(y=j)}{P(y=r)}\right) = X\beta_{(j)} \text{ for } j = 1, \dots, r-1$$

### Fitting Principle For Nominal Response

The fitting principle is called *maximum likelihood*. It estimates the parameters such that the joint probability for all the responses given by the data is the greatest obtainable by the model. Rather than reporting the joint probability (likelihood) directly, it is more manageable to report the total of the negative logs of the likelihood.

The uncertainty ( $-\log$ -likelihood) is the sum of the negative logs of the probabilities attributed by the model to the responses that actually occurred in the sample data. For a sample of size  $n$ , it is often denoted as  $H$  and written

$$H = \sum_{i=1}^n -\log(P(y=y_i))$$

If you attribute a probability of 1 to each event that did occur, then the sum of the negative logs is zero for a perfect fit.

The nominal model can take a lot of time and memory to fit, especially if there are many response levels. JMP tracks the progress of its calculations with an *iteration history*, which shows the  $-\log$ -likelihood values becoming smaller as they converge to the estimates.

### Base Model

The simplest model for a nominal response is a set of constant response probabilities fitted as the occurrence rates for each response level across the whole data table. In other words, the probability that  $y$  is response level  $j$  is estimated by dividing the total sample count  $n$  into the total of each response level  $n_j$ , and is written

$$p_j = \frac{n_j}{n}$$

All other models are compared to this base model. The base model serves the same role for a nominal response as the sample mean does for continuous models.

The  $R^2$  statistic measures the portion of the uncertainty accounted for by the model, which is

$$1 - \frac{H(\text{full model})}{H(\text{base model})}$$

However, it is rare in practice to get an  $R^2$  near 1 for categorical models.

## Ordinal Responses

With an ordinal response ( $Y$ ), as with nominal responses, JMP fits probabilities that the response is one of  $r$  different response levels given by the data.

Ordinal data have an order like continuous data. The order is used in the analysis but the spacing or distance between the ordered levels is not used. If you have a numeric response but want your model to ignore the spacing of the values, you can assign the ordinal level to that response column. If you have a classification variable and the levels are in some natural order such as low, medium, and high, you can use the ordinal modeling type.

Ordinal responses are modeled by fitting a series of parallel logistic curves to the cumulative probabilities. Each curve has the same design parameters but a different intercept and is written

$$P(y \leq j) = F(\alpha_j + X\beta) \quad \text{for } j = 1, \dots, r-1$$

where  $r$  response levels are present and  $F(x)$  is the standard logistic cumulative distribution function

$$F(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

Another way to write this is in terms of an unobserved continuous variable,  $z$ , that causes the ordinal response to change as it crosses various thresholds

$$y = \begin{cases} r & \alpha_{r-1} \leq z \\ j & \alpha_{j-1} \leq z < \alpha_j \\ 1 & z \leq \alpha_1 \end{cases}$$

where  $z$  is an unobservable function of the linear model and error

$$z = X\beta + \varepsilon$$

and  $\varepsilon$  has the logistic distribution.

These models are attractive in that they recognize the ordinal character of the response, they need far fewer parameters than nominal models, and the computations are fast even though they involve iterative maximum likelihood calculation.

A different but mathematically equivalent way to envision an ordinal model is to think of a nominal model where, instead of modeling the odds, you model the cumulative probability. Instead of fitting functions for all but the last level, you fit only one function and slide it to fit each cumulative response probability.

### **Fitting Principle For Ordinal Response**

The maximum likelihood fitting principle for an ordinal response model is the same as for a nominal response model. It estimates the parameters such that the joint probability for all the responses that occur is the greatest obtainable by the model. It uses an iterative method that is faster and uses less memory than nominal fitting.

#### **Base Model**

The simplest model for an ordinal response, like a nominal response, is a set of response probabilities fitted as the occurrence rates of the response in the whole data table.

## **The Factor Models**

The way the  $x$ -variables (factors) are modeled to predict an expected value or probability is the subject of the factor side of the model.

The factors enter the prediction equation as a linear combination of  $x$  values and the parameters to be estimated. For a continuous response model, where  $i$  indexes the observations and  $j$  indexes the parameters, the assumed model for a typical observation,  $y_i$ , is written

$$y_i = \beta_0 + \beta_1 x_{1i} + \dots + \beta_k x_{ki} + \varepsilon_i \text{ where}$$

$y_i$  is the response

$x_{ij}$  are functions of the data

$\varepsilon_i$  is an unobservable realization of the random error

$\beta_j$  are unknown parameters to be estimated.

The way the  $x$ 's in the linear model are formed from the factor terms is different for each modeling type. The linear model  $x$ 's can also be complex effects such as interactions or nested effects. Complex effects are discussed in detail later.

## **Continuous Factors**

Continuous factors are placed directly into the design matrix as regressors. If a column is a linear function of other columns, then the parameter for this column is marked *zeroed* or *nonestimable*. Continuous factors are centered by their mean when they are crossed with other factors (interactions and polynomial terms). Centering is suppressed if the factor has a Column Property of **Mixture** or **Coding**, or if the centered polynomials option is turned off when specifying the model. If there is a coding column property, the factor is coded before fitting.

## Nominal Factors

Nominal factors are transformed into indicator variables for the design matrix. SAS GLM constructs an indicator column for each nominal level. JMP constructs the same indicator columns for each nominal level except the last level. When the last nominal level occurs, a one is subtracted from all the other columns of the factor. For example, consider a nominal factor A with three levels coded for GLM and for JMP as shown below.

**Table A.1** Nominal Factor A

A	GLM			JMP	
	A1	A2	A3	A13	A23
A1	1	0	0	1	0
A2	0	1	0	0	1
A3	0	0	1	-1	-1

In GLM, the linear model design matrix has linear dependencies among the columns, and the least squares solution employs a generalized inverse. The solution chosen happens to be such that the A3 parameter is set to zero.

In JMP, the linear model design matrix is coded so that it achieves full rank unless there are missing cells or other incidental collinearity. The parameter for the A effect for the last level is the negative sum of the other levels, which makes the parameters sum to zero over all the effect levels.

## Interpretation of Parameters

**Note:** The parameter for a nominal level is interpreted as the differences in the predicted response for that level from the average predicted response over all levels.

The design column for a factor level is constructed as the zero-one indicator of that factor level minus the indicator of the last level. This is the coding that leads to the parameter interpretation above.

**Table A.2** Interpreting Parameters

JMP Parameter Report	How to Interpret	Design Column Coding
Intercept	mean over all levels	1'
A[1]	$\alpha_1 - 1/3(\alpha_1 + \alpha_2 + \alpha_3)$	(A==1) - (A==3)
A[2]	$\alpha_2 - 1/3(\alpha_1 + \alpha_2 + \alpha_3)$	(A==2) - (A==3)

## Interactions and Crossed Effects

Interaction effects with both GLM and JMP are constructed by taking a direct product over the rows of the design columns of the factors being crossed. For example, the GLM code

```
PROC GLM;
  CLASS A B;
  MODEL A B A*B;
```

yields this design matrix:

**Table A.3** Design Matrix

	A			B			AB									
A	B	1	2	3	1	2	3	11	12	13	21	22	23	31	32	33
A1	B1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
A1	B2	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
A1	B3	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
A2	B1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
A2	B2	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
A2	B3	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
A3	B1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
A3	B2	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
A3	B3	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1

Using the JMP **Fit Model** command and requesting a factorial model for columns A and B produces the following design matrix. Note that A13 in this matrix is A1–A3 in the previous matrix. However, A13B13 is A13\*B13 in the current matrix.

**Table A.4** Current Matrix

	A		B		A13 B13				A13 B23				A23 B13				A23 B23				
A	B	13	23	13	23	A13	B13	A13	B23	A23	B13	A23	B23	A13	B13	A13	B23	A23	B13	A23	B23
A1	B1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A1	B2	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	
A1	B3	1	0	-1	-1	-1	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	
A2	B1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
A2	B2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
A2	B3	0	1	-1	-1	0	0	0	0	-1	0	0	0	0	0	0	0	-1	-1	0	
A3	B1	-1	-1	1	0	-1	0	0	0	-1	0	0	0	-1	0	-1	0	0	0	0	

**Table A.4** Current Matrix (*Continued*)

A3	B2	-1	-1	0	1	0	-1	0	-1
A3	B3	-1	-1	-1	-1	1	1	1	1

The JMP coding saves memory and some computing time for problems with interactions of factors with few levels.

The expected values of the cells in terms of the parameters for a three-by-three crossed model are:

**Table A.5** Three-by-Three Crossed Model

	B1	B2	B3
A1	$\mu + \alpha_1 + \beta_1 + \alpha\beta_{11}$	$\mu + \alpha_1 + \beta_2 + \alpha\beta_{12}$	$\mu + \alpha_1 - \beta_1 - \beta_2 - \alpha\beta_{11} - \alpha\beta_{12}$
A2	$\mu + \alpha_2 + \beta_1 + \alpha\beta_{21}$	$\mu + \alpha_2 + \beta_2 + \alpha\beta_{22}$	$\mu + \alpha_2 - \beta_1 - \beta_2 - \alpha\beta_{21} - \alpha\beta_{22}$
A3	$\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21}$	$\mu - \alpha_1 - \alpha_2 + \beta_2 - \alpha\beta_{12} - \alpha\beta_{22}$	$\mu + \alpha_1 - \alpha_2 - \beta_1 - \beta_2 - \alpha\beta_{11} - \alpha\beta_{12} + \beta_{21} + \alpha\beta_{22}$

### Nested Effects

Nested effects in GLM are coded the same as interaction effects because GLM determines the right test by what isn't in the model. Any effect not included in the model can have its effect soaked up by a containing interaction (or, equivalently, nested) effect.

Nested effects in JMP are coded differently. JMP uses the terms inside the parentheses as grouping terms for each group. For each combination of levels of the nesting terms, JMP constructs the effect on the outside of the parentheses. The levels of the outside term need not line up across the levels of the nesting terms. Each level of nest is considered separately with regard to the construction of design columns and parameters.

**Table A.6** Nested Effects

A	B			B(A)					
		A13	A23	A1	A1	A2	A2	A3	A3
		B13	B23	B13	B23	B13	B23		
A1	B1	1	0	1	0	0	0	0	0
A1	B2	1	0	0	1	0	0	0	0
A1	B3	1	0	-1	-1	0	0	0	0
A2	B1	0	1	0	0	1	0	0	0
A2	B2	0	1	0	0	0	1	0	0

**Table A.6** Nested Effects (*Continued*)

A2	B3	0	1	0	0	-1	-1	0	0
A3	B1	-1	-1	0	0	0	0	1	0
A3	B2	-1	-1	0	0	0	0	0	1
A3	B3	-1	-1	0	0	0	0	-1	-1

### Least Squares Means across Nominal Factors

Least squares means are the predicted values corresponding to some combination of levels, after setting all the other factors to some neutral value. The neutral value for direct continuous regressors is defined as the sample mean. The neutral value for an effect with uninvolved nominal factors is defined as the average effect taken over the levels (which happens to result in all zeroes in our coding). Ordinal factors use a different neutral value in “[Ordinal Least Squares Means](#),” p. 625. The least squares means might not be estimable, and if not, they are marked nonestimable. JMP’s least squares means agree with GLM’s (Goodnight and Harvey 1978) in all cases except when a weight is used, where JMP uses a weighted mean and GLM uses an unweighted mean for its neutral values.

### Effective Hypothesis Tests

Generally, the hypothesis tests produced by JMP agree with the hypothesis tests of most other trusted programs, such as SAS PROC GLM (Hypothesis types III and IV). The following two sections describe where there are differences.

In the SAS GLM procedure, the hypothesis tests for Types III and IV are constructed by looking at the general form of estimable functions and finding functions that involve only the effects of interest and effects contained by the effects of interest (Goodnight 1978).

In JMP, the same tests are constructed, but because there is a different parameterization, an effect can be tested (assuming full rank for now) by doing a joint test on all the parameters for that effect. The tests do not involve containing interaction parameters because the coding has made them uninvolved with the tests on their contained effects.

If there are missing cells or other singularities, the JMP tests are different than GLM tests. There are several ways to describe them:

- JMP tests are equivalent to testing that the least squares means are different, at least for main effects. If the least squares means are nonestimable, then the test cannot include some comparisons and, therefore, loses degrees of freedom. For interactions, JMP is testing that the least squares means differ by more than just the marginal pattern described by the containing effects in the model.
- JMP tests an effect by comparing the SSE for the model with that effect with the SSE for the model without that effect (at least if there are no nested terms, which complicate the logic slightly). JMP parameterizes so that this method makes sense.
- JMP implements the *effective hypothesis tests* described by Hocking (1985, 80–89, 163–166), although JMP uses structural rather than cell-means parameterization. Effective hypothesis tests start with the

hypothesis desired for the effect and include “as much as possible” of that test. Of course, if there are containing effects with missing cells, then this test will have to drop part of the hypothesis because the complete hypothesis would not be estimable. The effective hypothesis drops as little of the complete hypothesis as possible.

- The differences among hypothesis tests in JMP and GLM (and other programs) that relate to the presence of missing cells are not considered interesting tests anyway. If an interaction is significant, the test for the contained main effects are not interesting. If the interaction is not significant, then it can always be dropped from the model. Some tests are not even unique. If you relabel the levels in a missing cell design, then the GLM Type IV tests can change.

The following section continues this topic in finer detail.

### **Singularities and Missing Cells in Nominal Effects**

Consider the case of linear dependencies among the design columns. With JMP coding, this does not occur unless there is insufficient data to fill out the combinations that need estimating, or unless there is some kind of confounding or collinearity of the effects.

With linear dependencies, a least squares solution for the parameters might not be unique and some tests of hypotheses cannot be tested. The strategy chosen for JMP is to set parameter estimates to zero in sequence as their design columns are found to be linearly dependent on previous effects in the model. A special column in the report shows what parameter estimates are zeroed and which parameter estimates are estimable. A separate *singularities* report shows what the linear dependencies are.

In cases of singularities the hypotheses tested by JMP can differ from those selected by GLM. Generally, JMP finds fewer degrees of freedom to test than GLM because it holds its tests to a higher standard of marginality. In other words, JMP tests always correspond to tests across least squares means for that effect, but GLM tests do not always have this property.

For example, consider a two-way model with interaction and one missing cell where A has three levels, B has two levels, and the A3B2 cell is missing.

**Table A.7** Two-Way Model with Interaction

A B	A1	A2	B1	A1B1	A2B1	
<b>A1 B1</b>	1	0	1	1	0	
<b>A2 B1</b>	0	1	1	0	1	
<b>A3 B1</b>	-1	-1	1	-1	-1	
<b>A1 B2</b>	1	0	-1	-1	0	
<b>A2 B2</b>	0	1	-1	0	-1	
<b>A3 B2</b>	-1	-1	-1	1	1	suppose this is missing

The expected values for each cell are:

**Table A.8** Expected Values

	B1	B2
A1	$\mu + \alpha_1 + \beta_1 + \alpha\beta_{11}$	$\mu + \alpha_1 - \beta_1 - \alpha\beta_{11}$
A2	$\mu + \alpha_2 + \beta_1 + \alpha\beta_{21}$	$\mu + \alpha_2 - \beta_1 - \alpha\beta_{21}$
A3	$\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21}$	$\mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21}$

Obviously, any cell with data has an expectation that is estimable. The cell that is missing has an expectation that is nonestimable. In fact, its expectation is precisely that linear combination of the design columns that is in the singularity report

$$\mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21}$$

Suppose that you want to construct a test that compares the least squares means of B1 and B2. In this example, the average of the rows in the above table give these least squares means.

$$\begin{aligned} \text{LSM(B1)} &= (1/3)(\mu + \alpha_1 + \beta_1 + \alpha\beta_{11} + \\ &\quad \mu + \alpha_2 + \beta_1 + \alpha\beta_{21} + \\ &\quad \mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21}) \\ &= \mu + \beta_1 \end{aligned}$$

$$\begin{aligned} \text{LSM(B2)} &= (1/3)(\mu + \alpha_1 + -\beta_1 - \alpha\beta_{11} + \\ &\quad \mu + \alpha_2 + -\beta_1 - \alpha\beta_{21} + \\ &\quad \mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21}) \\ &= \mu - \beta_1 \end{aligned}$$

$$\text{LSM(B1)} - \text{LSM(B2)} = 2\beta_1$$

Note that this shows that a test on the  $\beta_1$  parameter is equivalent to testing that the least squares means are the same. But because  $\beta_1$  is not estimable, the test is not testable, meaning there are no degrees of freedom for it.

Now, construct the test for the least squares means across the A levels.

$$\begin{aligned} \text{LSM(A1)} &= (1/2)(\mu + \alpha_1 + \beta_1 + \alpha\beta_{11} + \mu + \alpha_1 - \beta_1 - \alpha\beta_{11}) \\ &= \mu + \alpha_1 \end{aligned}$$

$$\begin{aligned} \text{LSM(A2)} &= (1/2)(\mu + \alpha_2 + \beta_1 + \alpha\beta_{21} + \mu + \alpha_2 - \beta_1 - \alpha\beta_{21}) \\ &= \mu + \alpha_2 \end{aligned}$$

$$\begin{aligned} \text{LSM(A3)} &= (1/2)(\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21} + \\ &\quad \mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21}) \\ &= \mu - \alpha_1 - \alpha_2 \end{aligned}$$

$$\text{LSM(A1)} - \text{LSM(A3)} = 2\alpha_1 + \alpha_2$$

$$\text{LSM(A2)} - \text{LSM(A3)} = 2\alpha_2 + \alpha_1$$

Neither of these turn out to be estimable, but there is another comparison that is estimable; namely comparing the two A columns that have no missing cells.

$$\text{LSM(A1)} - \text{LSM(A2)} = \alpha_1 - \alpha_2$$

This combination is indeed tested by JMP using a test with 1 degree of freedom, although there are two parameters in the effect.

The estimability can be verified by taking its inner product with the singularity combination, and checking that it is zero:

**Table A.9** Verification

parameters	singularity	combination
	combination	to be tested
<b>m</b>	1	0
<b>a<sub>1</sub></b>	-1	1
<b>a<sub>2</sub></b>	-1	-1
<b>b<sub>1</sub></b>	-1	0
<b>ab<sub>11</sub></b>	1	0
<b>ab<sub>21</sub></b>	1	0

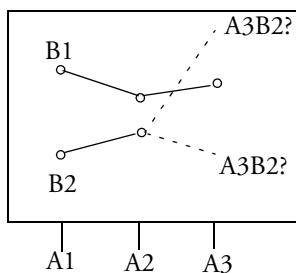
It turns out that the design columns for missing cells for any interaction will always knock out degrees of freedom for the main effect (for nominal factors). Thus, there is a direct relation between the nonestimability of least squares means and the loss of degrees of freedom for testing the effect corresponding to these least squares means.

How does this compare with what GLM does? GLM and JMP do the same test when there are no missing cells. That is, they effectively test that the least squares means are equal. But when GLM encounters singularities, it focuses out these cells in different ways, depending on whether they are Type III or Type IV. For Type IV, it looks for estimable combinations that it can find. These might not be unique, and if you reorder the levels, you might get a different result. For Type III, it does some orthogonalization of the estimable functions to obtain a unique test. But the test might not be very interpretable in terms of the cell means.

The JMP approach has several points in its favor, although at first it might seem distressing that you might lose more degrees of freedom than with GLM:

1. The tests are philosophically linked to LSMs.
2. The tests are easy computationally, using reduction sum of squares for reparameterized models.
3. The tests agree with Hocking's "Effective Hypothesis Tests".
4. The tests are *whole marginal tests*, meaning they always go completely across other effects in interactions.

The last point needs some elaboration: Consider a graph of the expected values of the cell means in the previous example with a missing cell for A3B2.



The graph shows expected cell means with a missing cell. The means of the A1 and A2 cells are profiled across the B levels. The JMP approach says you can't test the B main effect with a missing A3B2 cell, because the mean of the missing cell could be anything, as allowed by the interaction term. If the mean of the missing cell were the higher value shown, the B effect would likely test significant. If it were the lower, it would likely test nonsignificant. The point is that you don't know. That is what the least squares means are saying when they are declared nonestimable. That is what the hypotheses for the effects should be saying too—that you don't know.

If you want to test hypotheses involving margins for subsets of cells, then that is what GLM Type IV does. In JMP you would have to construct these tests yourself by partitioning the effects with a lot of calculations or by using contrasts.

### JMP and GLM Hypotheses

GLM works differently than JMP and produces different hypothesis tests in situations where there are missing cells. In particular, GLM does not recognize any difference between a nesting and a crossing in an effect, but JMP does. Suppose that you have a three-layer nesting of A, B(A), and C(A B) with different numbers of levels as you go down the nested design.

**Table A.10 “Comparison of GLM and JMP Hypotheses,” p. 620,** shows the test of the main effect A in terms of the GLM parameters. The first set of columns is the test done by JMP. The second set of columns is the test done by GLM Type IV. The third set of columns is the test equivalent to that by JMP; it is the first two columns that have been multiplied by a matrix

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

to be comparable to the GLM test. The last set of columns is the GLM Type III test. The difference is in how the test distributes across the containing effects. In JMP, it seems more top-down hierarchical. In GLM Type IV, the test seems more bottom-up. In practice, the test statistics are often similar.

**Table A.10** Comparison of GLM and JMP Hypotheses

Parameter	JMP Test for A	GLM-IV Test for A	JMP Rotated Test	GLM-III Test for A
-----------	----------------	-------------------	------------------	--------------------

**Table A.10** Comparison of GLM and JMP Hypotheses (*Continued*)

<b>u</b>	0	0	0	0	0	0	0	0
<b>a1</b>	0.6667	-0.3333	1	0	1	0	1	0
<b>a2</b>	-0.3333	0.6667	0	1	0	1	0	1
<b>a3</b>	-0.3333	-0.3333	-1	-1	-1	-1	-1	-1
<b>a1b1</b>	0.1667	-0.0833	0.2222	0	0.25	0	0.2424	0
<b>a1b2</b>	0.1667	-0.0833	0.3333	0	0.25	0	0.2727	0
<b>a1b3</b>	0.1667	-0.0833	0.2222	0	0.25	0	0.2424	0
<b>a1b4</b>	0.1667	-0.0833	0.2222	0	0.25	0	0.2424	0
<b>a2b1</b>	-0.1667	0.3333	0	0.5	0	0.5	0	.5
<b>a2b2</b>	-0.1667	0.3333	0	0.5	0	0.5	0	.5
<b>a3b1</b>	-0.1111	-0.1111	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333
<b>a3b2</b>	-0.1111	-0.1111	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333
<b>a3b3</b>	-0.1111	-0.1111	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333	-0.3333
<b>a1b1c1</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0
<b>a1b1c2</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0
<b>a1b2c1</b>	0.0556	-0.0278	0.1111	0	0.0833	0	0.0909	0
<b>a1b2c2</b>	0.0556	-0.0278	0.1111	0	0.0833	0	0.0909	0
<b>a1b2c3</b>	0.0556	-0.0278	0.1111	0	0.0833	0	0.0909	0
<b>a1b3c1</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0
<b>a1b3c2</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0
<b>a1b4c1</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0
<b>a1b4c2</b>	0.0833	-0.0417	0.1111	0	0.125	0	0.1212	0

**Table A.10** Comparison of GLM and JMP Hypotheses (*Continued*)

a2b1c1	-0.0833	0.1667	0	0.25	0	0.25	0	0.25
a2b1c2	-0.0833	0.1667	0	0.25	0	0.25	0	0.25
a2b2c1	-0.0833	0.1667	0	0.25	0	0.25	0	0.25
a2b2c2	-0.0833	0.1667	0	0.25	0	0.25	0	0.25
a3b1c1	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667
a3b1c2	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667
a3b2c1	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667
a3b2c2	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667
a3b3c1	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667
a3b3c2	-0.0556	-0.0556	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667	-0.1667

## Ordinal Factors

Factors marked with the ordinal modeling type are coded differently than nominal factors. The parameters estimates are interpreted differently, the tests are different, and the least squares means are different.

The theme for ordinal factors is that the first level of the factor is a control or baseline level, and the parameters measure the effect on the response as the ordinal factor is set to each succeeding level. The coding is appropriate for factors with levels representing various doses, where the first dose is zero:

**Table A.11** Ordinal Factors

Term	Coded Column		
A	a2	a3	
A1	0	0	control level, zero dose
A2	1	0	low dose
A3	1	1	higher dose

From the perspective of the JMP parameterization, the tests for A are:

**Table A.12** Tests for A

parameter	GLM-IV test		JMP test	
m	0	0	0	0

**Table A.12** Tests for A (Continued)

a13	2	1	1	0
a23	1	2	0	1
a1:b14	0	0	0	0
a1:b24	0.11111	0	0	0
a1:b34	0	0	0	0
a2:b12	0	0	0	0
a3:b13	0	0	0	0
a3:b23	0	0	0	0
a1b1:c12	0	0	0	0
a1b2:c13	0	0	0	0
a1b2:c23	0	0	0	0
a1b3:c12	0	0	0	0
a1b4:c12	0	0	0	0
a2b1:c13	0	0	0	0
a2b2:c12	0	0	0	0
a3b1:c12	0	0	0	0
a3b2:c12	0	0	0	0
a3b3:c12	0	0	0	0

So from JMP's perspective, the GLM test looks a little strange, putting a coefficient on the a1b24 parameter.

The pattern for the design is such that the lower triangle is ones with zeros elsewhere.

For a simple main-effects model, this can be written

$$y = \mu + \alpha_2 X_{(a \leq 2)} + \alpha_3 X_{(a \leq 3)} + \varepsilon$$

noting that  $\mu$  is the expected response at  $A = 1$ ,  $\mu + \alpha_2$  is the expected response at  $A = 2$ , and  $\mu + \alpha_2 + \alpha_3$  is the expected response at  $A = 3$ . Thus,  $\alpha_2$  estimates the effect moving from  $A = 1$  to  $A = 2$  and  $\alpha_3$  estimates the effect moving from  $A = 2$  to  $A = 3$ .

If all the parameters for an ordinal main effect have the same sign, then the response effect is monotonic across the ordinal levels.

**Ordinal Interactions**

The ordinal interactions, as with nominal effects, are produced with a horizontal direct product of the columns of the factors. Consider an example with two ordinal factors A and B, each with three levels. JMP's ordinal coding produces the design matrix shown next. The pattern for the interaction is a block lower-triangular matrix of lower-triangular matrices of ones.

**Table A.13** Ordinal Interactions

		A*B			
		A2		A3	
A	B	A2	A3	B2	B3
A1	B1	0	0	0	0
A1	B2	0	0	1	0
A1	B3	0	0	1	1
A2	B1	1	0	0	0
A2	B2	1	0	1	0
A2	B3	1	0	1	1
A3	B1	1	1	0	0
A3	B2	1	1	1	0
A3	B3	1	1	1	1

---

**Note:** When you test to see if there is no effect, there is not much difference between nominal and ordinal factors for simple models. However, there are major differences when interactions are specified. We recommend that you use nominal rather than ordinal factors for most models.

---

**Hypothesis Tests for Ordinal Crossed Models**

To see what the parameters mean, examine this table of the expected cell means in terms of the parameters, where  $\mu$  is the intercept,  $\alpha_2$  is the parameter for level A2, and so forth.

**Table A.14** Expected Cell Means

	B1	B2	B3
A1	$\mu$	$\mu + \alpha\beta_2 + \alpha\beta_{12}$	$\mu + \beta_2 + \beta_3$
A2	$\mu + \alpha_2$	$\mu + \alpha_2 + \beta_2 + \alpha\beta_{22}$	$\mu + \alpha_2 + \beta_2 + \beta_3 + \alpha\beta_{22} + \alpha\beta_{23}$

**Table A.14** Expected Cell Means

A3	$\mu + \alpha_2 + \alpha_3$	$\mu + \alpha_2 + \alpha_3 + \beta_2 + \alpha\beta_{22} + \alpha\beta_{32} + \alpha\beta_{23} + \alpha\beta_{32} + \alpha\beta_{33}$	$\mu + \alpha_2 + \alpha_3 + \beta_2 + \beta_3 + \alpha\beta_{22} + \alpha\beta_{23} + \beta_{32} + \alpha\beta_{33}$
----	-----------------------------	--	---

Note that the main effect test for A is really testing the A levels holding B at the first level. Similarly, the main effect test for B is testing across the top row for the various levels of B holding A at the first level. This is the appropriate test for an experiment where the two factors are both doses of different treatments. The main question is the efficacy of each treatment by itself, with fewer points devoted to looking for *drug interactions* when doses of both drugs are applied. In some cases it may even be dangerous to apply large doses of each drug.

Note that each cell's expectation can be obtained by adding all the parameters associated with each cell that is to the left and above it, inclusive of the current row and column. The expected value for the last cell is the sum of all the parameters.

Though the hypothesis tests for effects contained by other effects differs with ordinal and nominal codings, the test of effects not contained by other effects is the same. In the crossed design above, the test for the interaction would be the same no matter whether A and B were fit nominally or ordinally.

### Ordinal Least Squares Means

As stated previously, least squares means are the predicted values corresponding to some combination of levels, after setting all the other factors to some neutral value. JMP defines the neutral value for an effect with uninvolved ordinal factors as the effect at the first level, meaning the *control* or *baseline* level.

This definition of least squares means for ordinal factors maintains the idea that the hypothesis tests for contained effects are equivalent to tests that the least squares means are equal.

### Singularities and Missing Cells in Ordinal Effects

With the ordinal coding, you are saying that the first level of the ordinal effect is the baseline. It is thus possible to get good tests on the main effects even when there are missing cells in the interactions—even if you have no data for the interaction.

### Example with Missing Cell

The example is the same as above, with two observations per cell except that the A3B2 cell has no data. You can now compare the results when the factors are coded nominally with results when they are coded ordinally. The model as a whole fits the same as seen in tables shown in Figure A.1.

**Table A.15** Observations

Y	A	B
12	1	1
14	1	

**Table A.15** Observations

15	1	2
16	1	2
17	2	1
17	2	1
18	2	2
19	2	2
20	3	1
24	3	1

**Figure A.1** Comparison of Summary Information for Nominal and Ordinal Fits

Summary of Fit (Nominal)					Summary of Fit (Ordinal)				
RSquare		0.891732			RSquare		0.891732		
RSquare Adj		0.805118			RSquare Adj		0.805118		
Root Mean Square Error		1.48324			Root Mean Square Error		1.48324		
Mean of Response		17.2			Mean of Response		17.2		
Observations (or Sum Wgts)		10			Observations (or Sum Wgts)		10		
Analysis of Variance (Nominal)					Analysis of Variance (Ordinal)				
Source	DF	Sum of Squares	Mean Square	F Ratio	Source	DF	Sum of Squares	Mean Square	F Ratio
Model	4	90.60000	22.6500	10.2955	Model	4	90.60000	22.6500	10.2955
Error	5	11.00000	2.2000	Prob > F	Error	5	11.00000	2.2000	Prob > F
C. Total	9	101.60000		0.0125*	C. Total	9	101.60000		0.0125*

The parameter estimates are very different because of the different coding. Note that the missing cell affects estimability for some nominal parameters but for none of the ordinal parameters.

**Figure A.2** Comparison of Parameter Estimates for Nominal and Ordinal Fits

Parameter Estimates (Nominal)					Parameter Estimates (Ordinal)				
Term	Estimate	Std Error	t Ratio	Prob> t	Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	17.333333	0.60553	28.63	<.0001*	Intercept	13	1.048809	12.40	<.0001*
A[1]	-4.333333	0.856349	-5.06	0.0039*	A[2-1]	4	1.48324	2.70	0.0429*
A[2]	-0.333333	0.856349	-0.39	0.7131	A[3-2]	5	1.48324	3.37	0.0199*
B[2-1] Biased	1.5	1.48324	1.01	0.3583	B[2-1]	2.5	1.48324	1.89	0.1527
A[1]*B[2-1] Biased	1	2.097618	0.48	0.6537	A[2-1]*B[2-1]	-1	2.097618	-0.48	0.6537
A[2]*B[2-1] Zeroed	0	0	.	.	A[3-2]*B[2-1] Zeroed	0	0	.	.

The singularity details show the linear dependencies (and also identify the missing cell by examining the values).

**Figure A.3** Comparison of Singularity Details for Nominal and Ordinal Fits

Singularity Details (Nominal)	
$B[2-1] = A[1]*B[2-1] + A[2]*B[2-1]$	
Singularity Details (Ordinal)	
$A[3-2]*B[2-1] = 0$	

The effect tests lose degrees of freedom for nominal. In the case of B, there is no test. For ordinal, there is no loss because there is no missing cell for the *base* first level.

**Figure A.4** Comparison of Effects Tests for Nominal and Ordinal Fits

Effect Tests (Nominal)					
Source	Nparm	DF	Sum of	F Ratio	Prob > F
			Squares		
A	2	2	81.333333	18.4848	0.0049*
B	1	0	0.000000	.	.
A*B	2	1	0.500000	0.2273	0.6537

Effect Tests (Ordinal)					
Source	Nparm	DF	Sum of	F Ratio	Prob > F
			Squares		
A	2	2	81.333333	18.4848	0.0049*
B	1	1	6.250000	2.8409	0.1527
A*B	2	1	0.500000	0.2273	0.6537

The least squares means are also different. The nominal LSMS are not all estimable, but the ordinal LSMS are. You can verify the values by looking at the cell means. Note that the A\*B LSMS are the same for the two. Figure A.5 shows least squares means for an nominal and ordinal fits.

**Figure A.5** Least Squares Means for Nominal and Ordinal Fits

The figure displays four tables of Least Squares Means extracted from a statistical software interface. The tables are organized into two columns: 'Effect Details (Nominal)' on the left and 'Effect Details (Ordinal)' on the right. Each column contains two tables: one for factor A and one for factor B.

**Effect Details (Nominal) - Factor A:**

Level	Least Sq Mean	Std Error	Mean
1	13.000000	1.0488088	14.2500
2	17.000000	1.0488088	17.7500
3	22.000000	1.0488088	22.0000

**Effect Details (Nominal) - Factor B:**

Level	Least Sq Mean	Std Error	Mean
1	17.333333	0.60553007	17.3333
2	NonEstimable		17.0000

**Effect Details (Nominal) - Interaction A\*B:**

Level	Least Sq Mean	Std Error
1,1	13.000000	1.0488088
1,2	15.500000	1.0488088
2,1	17.000000	1.0488088
2,2	18.500000	1.0488088
3,1	22.000000	1.0488088
3,2	NonEstimable	

**Effect Details (Ordinal) - Factor A:**

Level	Least Sq Mean	Std Error	Mean
1	13.000000	1.0488088	14.2500
2	17.000000	1.0488088	17.7500
3	22.000000	1.0488088	22.0000

**Effect Details (Ordinal) - Factor B:**

Level	Least Sq Mean	Std Error	Mean
1	13.000000	1.0488088	17.3333
2	15.500000	1.0488088	17.0000

**Effect Details (Ordinal) - Interaction A\*B:**

Level	Least Sq Mean	Std Error
1,1	13.000000	1.0488088
1,2	15.500000	1.0488088
2,1	17.000000	1.0488088
2,2	18.500000	1.0488088
3,1	22.000000	1.0488088
3,2	NonEstimable	

## The Usual Assumptions

Before you put your faith in statistics, reassure yourself that you know both the value and the limitations of the techniques you use. Statistical methods are just tools—they cannot guard you from incorrect science (invalid statistical assumptions) or bad data.

### Assumed Model

Most statistics are based on the assumption that the model is correct. To the extent that your model may not be correct, you must attenuate your credibility in the statistical reports that result from the model.

### Relative Significance

Many statistical tests do not evaluate the model in an absolute sense. Significant test statistics might only be saying that the model fits better than some reduced model, such as the mean. The model can appear to fit the data but might not describe the underlying physical model well at all.

## Multiple Inferences

Often the value of the statistical results is not that you believe in them directly, but rather that they provide a key to some discovery. To confirm the discovery, you may need to conduct further studies. Otherwise, you might just be sifting through the data.

For instance, if you conduct enough analyses you can find 5% significant effects in five percent of your studies by chance alone, even if the factors have no predictive value. Similarly, to the extent that you use your data to shape your model (instead of testing the correct model for the data), you are corrupting the significance levels in your report. The random error then influences your model selection and leads you to believe that your model is better than it really is.

## Validity Assessment

Some of the various techniques and patterns to look for in assessing the validity of the model are as follows:

- Model validity can be checked against a saturated version of the factors with Lack of Fit tests. The Fit Model platform presents these tests automatically if you have replicated  $x$  data in a nonsaturated model.
- You can check the distribution assumptions for a continuous response by looking at plots of residuals and studentized residuals from the Fit Model platform. Or, use the **Save** commands in the platform popup menu to save the residuals in data table columns. Then use the **Analyze > Distribution** on these columns to look at a histogram with its normal curve and the normal quantile plot. The residuals are not quite independent, but you can informally identify severely non-normal distributions.
- The best all-around diagnostic tool for continuous responses is the leverage plot because it shows the influence of each point on each hypothesis test. If you suspect that there is a mistaken value in your data, this plot helps determine if a statistical test is heavily influenced by a single point.
- It is a good idea to scan your data for outlying values and examine them to see if they are valid observations. You can spot univariate outliers in the Distribution platform reports and plots. Bivariate outliers appear in Fit Y by X scatterplots and in the Multivariate scatterplot matrix. You can see trivariate outliers in a three-dimensional plot produced by the **Graph > Scatterplot 3D**. Higher dimensional outliers can be found with Principal Components or Scatterplot 3D, and with Mahalanobis and jack-knifed distances computed and plotted in the Multivariate platform.

## Alternative Methods

The statistical literature describes special nonparametric and robust methods, but JMP implements only a few of them at this time. These methods require fewer distributional assumptions (nonparametric), and then are more resistant to contamination (robust). However, they are less conducive to a general methodological approach, and the small sample probabilities on the test statistics can be time consuming to compute.

If you are interested in linear rank tests and need only normal large sample significance approximations, you can analyze the ranks of your data to perform the equivalent of a Wilcoxon rank-sum or Kruskal-Wallis one-way test.

If you are uncertain that a continuous response adequately meets normal assumptions, you can change the modeling type from continuous to ordinal and then analyze safely, even though this approach sacrifices some richness in the presentations and some statistical power as well.

## **Key Statistical Concepts**

There are two key concepts that unify classical statistics and encapsulate statistical properties and fitting principles into forms you can visualize:

- a unifying concept of uncertainty
- two basic fitting machines.

These two ideas help unlock the understanding of statistics with intuitive concepts that are based on the foundation laid by mathematical statistics.

Statistics is to science what accounting is to business. It is the craft of weighing and balancing observational evidence. Statistical tests are like credibility audits. But statistical tools can do more than that. They are instruments of discovery that can show unexpected things about data and lead to interesting new ideas. Before using these powerful tools, you need to understand a bit about how they work.

### **Uncertainty, a Unifying Concept**

When you do accounting, you total money amounts to get summaries. When you look at scientific observations in the presence of uncertainty or noise, you need some statistical measurement to summarize the data. Just as money is additive, uncertainty is additive if you choose the right measure for it.

The best measure is not the direct probability because to get a joint probability you have to assume that the observations are independent and then multiply probabilities rather than add them. It is easier to take the log of each probability because then you can sum them and the total is the log of the joint probability.

However, the log of a probability is negative because it is the log of a number between 0 and 1. In order to keep the numbers positive, JMP uses the negative log of the probability. As the probability becomes smaller, its negative log becomes larger. This measure is called uncertainty, and it is measured in reverse fashion from probability.

In business, you want to maximize revenues and minimize costs. In science you want to minimize uncertainty. Uncertainty in science plays the same role as cost plays in business. All statistical methods fit models such that uncertainty is minimized.

It is not difficult to visualize uncertainty. Just think of flipping a series of coins where each toss is independent. The probability of tossing a head is 0.5, and  $-\log(0.5)$  is 1 for base 2 logarithms. The probability of tossing  $h$  heads in a row is simply

$$p = \left(\frac{1}{2}\right)^h$$

Solving for  $h$  produces

$$h = -\log_2 p$$

You can think of the uncertainty of some event as the number of consecutive “head” tosses you have to flip to get an equally rare event.

Almost everything we do statistically has uncertainty,  $-\log p$ , at the core. Statistical literature refers to uncertainty as *negative log-likelihood*.

## The Two Basic Fitting Machines

An amazing fact about statistical fitting is that most of the classical methods reduce to using two simple machines, the spring and the pressure cylinder.

### **Springs**

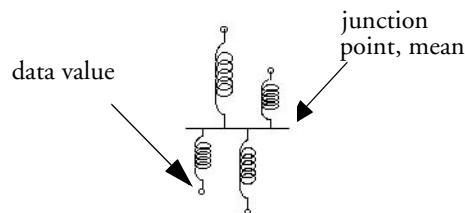
First, springs are the machine of fit for a continuous response model (Farebrother, 1987). Suppose that you have  $n$  points and that you want to know the expected value (mean) of the points. Envision what happens when you lay the points out on a scale and connect them to a common junction with springs (see Figure A.6). When you let go, the springs wiggle the junction point up and down and then bring it to rest at the mean. This is what must happen according to physics.

If the data are normally distributed with a mean at the junction point where springs are attached, then the physical energy in each point’s spring is proportional to the uncertainty of the data point. All you have to do to calculate the energy in the springs (the uncertainty) is to compute the sum of squared distances of each point to the mean.

To choose an estimate that attributes the least uncertainty to the observed data, the spring settling point is chosen as the estimate of the mean. That is the point that requires the least energy to stretch the springs and is equivalent to the least squares fit.

---

**Figure A.6** Connect Springs to Data Points



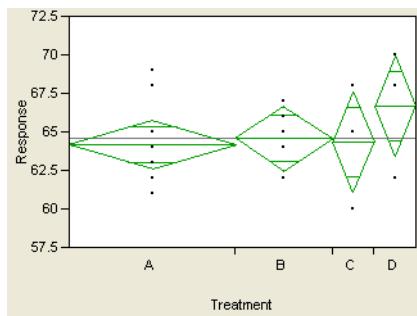

---

That is how you fit one mean or fit several means. That is how you fit a line, or a plane, or a hyperplane. That is how you fit almost any model to continuous data. You measure the energy or uncertainty by the sum of squares of the distances you must stretch the springs.

Statisticians put faith in the normal distribution because it is the one that requires the least faith. It is, in a sense, the most random. It has the most non-informative shape for a distribution. It is the one distribution that has the most expected uncertainty for a given variance. It is the distribution whose uncertainty is

measured in squared distance. In many cases it is the limiting distribution when you have a mixture of distributions or a sum of independent quantities. It is the distribution that leads to test statistics that can be measured fairly easily.

When the fit is constrained by hypotheses, you test the hypotheses by measuring this same spring energy. Suppose you have responses from four different treatments in an experiment, and you want to test if the means are significantly different. First, envision your data plotted in groups as shown here, but with springs connected to a separate mean for each treatment. Then exert pressure against the spring force to move the individual means to the common mean. Presto! The amount of energy that constrains the means to be the same is the test statistic you need. That energy is the main ingredient in the *F*-test for the hypothesis that tests whether the means are the same.



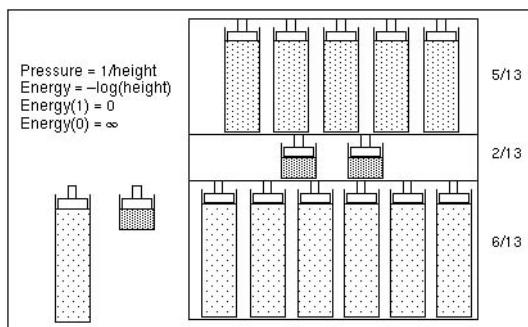
### Pressure Cylinders

What if your response is categorical instead of continuous? For example, suppose that the response is the country of origin for a sample of cars. For your sample, there are probabilities for the three response levels, American, European, and Japanese. You can set these probabilities for country of origin to some estimate and then evaluate the uncertainty in your data. This uncertainty is found by summing the negative logs of the probabilities of the responses given by the data. It is written

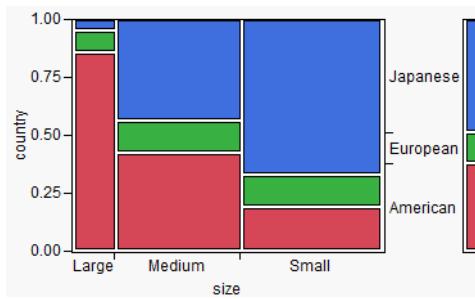
$$H = \sum h_{y(i)} = -\sum \log p_{y(i)}$$

The idea of springs illustrates how a mean is fit to continuous data. When the response is categorical, statistical methods estimate the response probabilities directly and choose the estimates that minimize the total uncertainty of the data. The probability estimates must be nonnegative and sum to 1. You can picture the response probabilities as the composition along a scale whose total length is 1. For each response observation, load into its response area a gas pressure cylinder, for example, a tire pump. Let the partitions between the response levels vary until an equilibrium of lowest potential energy is reached. The sizes of the partitions that result then estimate the response probabilities.

Figure A.7 shows what the situation looks like for a single category such as the medium size cars (see the mosaic column from *Carpoll.jmp* labeled *medium* in Figure A.8). Suppose there are thirteen responses (cars). The first level (American) has six responses, the next has two, and the last has five responses. The response probabilities become  $6/13$ ,  $2/13$ , and  $5/13$ , respectively, as the pressure against the response partitions balances out to minimize the total energy.

**Figure A.7** Effect of Pressure Cylinders in Partitions

As with springs for continuous data, you can divide your sample by some factor and fit separate sets of partitions. Then test that the response rates are the same across the groups by measuring how much additional energy you need to push the partitions to be equal. Imagine the pressure cylinders for car origin probabilities grouped by the size of the car. The energy required to force the partitions in each group to align horizontally tests whether the variables have the same probabilities. Figure A.8 shows these partitions.

**Figure A.8** A Mosaic Plot for Categorical Data

## Leverage Plot Details

Leverage plots for general linear hypotheses were introduced by Sall (1980). This section presents the details from that paper. Leverage plots generalize the partial regression residual leverage plots of Belsley, Kuh, and Welsch (1980) to apply to any linear hypothesis. Suppose that the estimable hypothesis of interest is

$$L\beta = 0$$

The leverage plot characterizes this test by plotting points so that the distance of each point to the sloped regression line displays the unconstrained residual, and the distance to the  $x$ -axis displays the residual when the fit is constrained by the hypothesis.

Of course the difference between the sums of squares of these two groups of residuals is the sum of squares due to the hypothesis, which becomes the main ingredient of the  $F$ -test.

The parameter estimates constrained by the hypothesis can be written

$$\mathbf{b}_0 = \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'\lambda$$

where  $\mathbf{b}$  is the least squares estimate

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

and where  $\lambda$  is the Lagrangian multiplier for the hypothesis constraint, which is calculated

$$\lambda = (\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}\mathbf{L}\mathbf{b}$$

Compare the residuals for the unconstrained and hypothesis-constrained residuals, respectively.

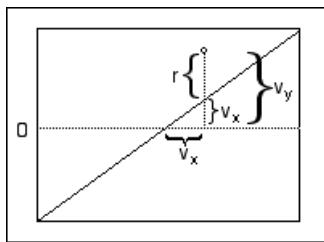
$$\mathbf{r} = \mathbf{y} - \mathbf{X}\mathbf{b}$$

$$\mathbf{r}_0 = \mathbf{r} + \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'\lambda$$

To get a leverage plot, the  $x$ -axis values  $v_x$  of the points are the differences in the residuals due to the hypothesis, so that the distance from the line of fit (with slope 1) to the  $x$ -axis is this difference. The  $y$ -axis values  $v_y$  are just the  $x$ -axis values plus the residuals under the full model as illustrated in Figure A.9. Thus, the leverage plot is composed of the points

$$v_x = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'\lambda \text{ and } v_y = \mathbf{r} + v_x$$

**Figure A.9** Construction of Leverage Plot



### **Superimposing a Test on the Leverage Plot**

In simple linear regression, you can plot the confidence limits for the expected value as a smooth function of the regressor variable  $x$

$$\text{Upper}(x) = \mathbf{x}\mathbf{b} + t_{\alpha/2} s \sqrt{\mathbf{x}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}'}$$

$$\text{Lower } (x) = \mathbf{x}\mathbf{b} - t_{\alpha/2} s \sqrt{\mathbf{x}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{x}'}$$

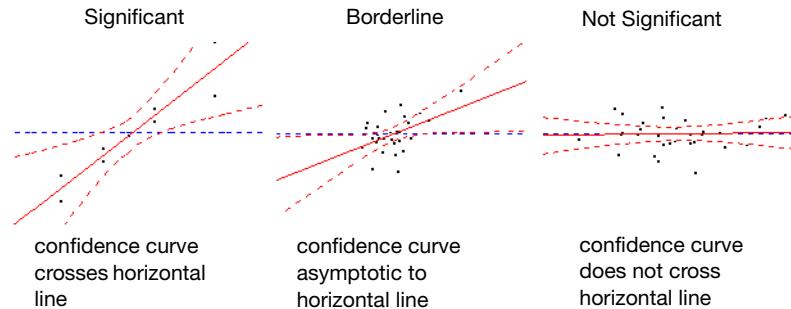
where  $\mathbf{x} = [1 \ x]$  is the 2-vector of regressors.

This hyperbola is a useful significance-measuring instrument because it has the following nice properties:

- If the slope parameter is significantly different from zero, the confidence curve will cross the horizontal line of the response mean (left plot in Figure A.10).
- If the slope parameter is not significantly different from zero, the confidence curve will not cross the horizontal line of the response mean (plot at right in Figure A.10).
- If the  $t$ -test for the slope parameter is sitting right on the margin of significance, the confidence curve will have the horizontal line of the response mean as an asymptote (middle plot in Figure A.10).

This property can be verified algebraically or it can be understood by considering what the confidence region has to be when the regressor value goes off to plus or minus infinity.

**Figure A.10** Cases of Significant, Borderline, and Nonsignificant Confidence Curves.



Leverage plots make use of the same device by calculating a confidence function with respect to one variable while holding all other variables constant at their sample mean value.

For general hypothesis tests, JMP can display a curve with the same properties: the confidence function shows at the mean value in the middle, with adjusted curvature so that it crosses the horizon if the  $F$ -test is significant at some  $\alpha$ -level like 0.05.

Consider the functions

$$\text{Upper}(z) = z + \sqrt{s^2 t_{\alpha/2}^2 \bar{h} + \frac{F_\alpha}{F} z^2}$$

and

$$\text{Lower } (z) = z - \sqrt{s^2 t_{\alpha/2}^2 \bar{h} + \frac{F_\alpha}{F} z^2}$$

where  $F$  is the  $F$ -statistic for the hypothesis,  $F_\alpha$  is the reference value for significance  $\alpha$ , and

$$\bar{b} = \bar{x}(X'X)^{-1}\bar{x}', \text{ where } \bar{x} \text{ is the regressors at a suitable middle value, such as the mean.}$$

These functions serve the same properties listed above. If the  $F$ -statistic is greater than the reference value, the confidence functions cross the  $x$ -axis. If the  $F$ -statistic is less than the reference value, the confidence functions do not cross. If the  $F$ -statistic is equal to the reference value, the confidence functions have the  $x$ -axis as an asymptote. And the range between the confidence functions at the middle value is a valid confidence region of the predicted value at the point.

## Multivariate Details

The following sections show computations used for multivariate tests and related, exact and approximate  $F$ -statistics, canonical details, and discriminant functions. In the following sections,  $E$  is the residual cross product matrix and  $\frac{E}{n-1}$  estimates the residual covariance matrix. Diagonal elements of  $E$  are the sum of squares for each variable. In discriminant analysis literature, this is often called  $W$ , where  $W$  stands for *within*.

## Multivariate Tests

Test statistics in the multivariate results tables are functions of the eigenvalues  $\lambda$  of  $E^{-1}H$ . The following list describes the computation of each test statistic.

**Note:** After specification of a response design, the initial  $E$  and  $H$  matrices are premultiplied by  $M'$  and postmultiplied by  $M$ .

**Table A.16** Computations for Multivariate Tests

Wilks' Lambda	$\Lambda = \frac{\det(E)}{\det(H+E)} = \prod_{i=1}^n \left( \frac{1}{1+\lambda_i} \right)$
Pillai's Trace	$V = \text{Trace}[H(H+E)^{-1}] = \sum_{i=1}^n \frac{\lambda_i}{1+\lambda_i}$
Hotelling-Lawley Trace	$U = \text{Trace}(E^{-1}H) = \sum_{i=1}^n \lambda_i$
Roy's Max Root	$\Theta = \lambda_1$ , the maximum eigenvalue of $E^{-1}H$ .

The whole model L is a column of zeros (for the intercept) concatenated with an identity matrix having the number of rows and columns equal to the number of parameters in the model. L matrices for effects are subsets of rows from the whole model L matrix.

## Approximate F-Test

To compute  $F$ -values and degrees of freedom, let  $p$  be the rank of  $H + E$ . Let  $q$  be the rank of  $L(X'X)^{-1}L'$ , where the L matrix identifies elements of  $X'X$  associated with the effect being tested. Let  $v$  be the error degrees of freedom and  $s$  be the minimum of  $p$  and  $q$ . Also let  $m = 0.5(|p - q| - 1)$  and  $n = 0.5(v - p - 1)$ .

[Table A.17 “Approximate F-statistics,” p. 637](#), gives the computation of each approximate F from the corresponding test statistic.

**Table A.17** Approximate  $F$ -statistics

Test	Approximate F	Numerator DF	Denominator DF
Wilks' Lambda	$F = \left( \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \right) \left( \frac{rt - 2u}{pq} \right)$	$pq$	$rt - 2u$
Pillai's Trace	$F = \left( \frac{V}{s - V} \right) \left( \frac{2n + s + 1}{2m + s + 1} \right)$	$s(2m + s + 1)$	$s(2n + s + 1)$
Hotelling-Lawley Trace	$F = \frac{2(sn + 1)U}{s^2(2m + s + 1)}$	$s(2m + s + 1)$	$2(sn + 1)$
Roy's Max Root	$F = \frac{\Theta(v - \max(p, q) + q)}{\max(p, q)}$	$\max(p, q)$	$v - \max(p, q) + q$

## Canonical Details

The canonical correlations are computed as

$$\rho_i = \sqrt{\frac{\lambda_i}{1 + \lambda_i}}$$

The canonical Y's are calculated as

$$Y = YMV$$

where Y is the matrix of response variables, M is the response design matrix, and V is the matrix of eigenvectors of  $E^{-1}H$ . Canonical Y's are saved for eigenvectors corresponding to eigenvalues larger than zero.

The total sample centroid is computed as

$$\text{Grand} = \bar{y}MV$$

where  $\mathbf{V}$  is the matrix of eigenvectors of  $\mathbf{E}^{-1}\mathbf{H}$ .

The centroid values for effects are calculated as

$$\mathbf{m} = (\mathbf{c}'_1 \bar{x}_j, \mathbf{c}'_2 \bar{x}_j, \dots, \mathbf{c}'_g \bar{x}_j) \quad \text{where } c_i = \left( v'_i \left( \frac{\mathbf{E}}{N-r} \right) v_i \right)^{-1/2} v_i$$

and the  $v$ s are columns of  $\mathbf{V}$ , the eigenvector matrix of  $\mathbf{E}^{-1}\mathbf{H}$ ,  $\bar{x}_j$  refers to the multivariate least squares mean for the  $j$ th effect,  $g$  is the number of eigenvalues of  $\mathbf{E}^{-1}\mathbf{H}$  greater than 0, and  $r$  is the rank of the  $\mathbf{X}$  matrix.

The centroid radii for effects are calculated as

$$d = \sqrt{\frac{\chi_g^2(0.95)}{\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'}}$$

where  $g$  is the number of eigenvalues of  $\mathbf{E}^{-1}\mathbf{H}$  greater than 0 and the denominator  $\mathbf{L}$ 's are from the multivariate least squares means calculations.

## Discriminant Analysis

The distance from an observation to the multivariate mean of the  $i$ th group is the Mahalanobis distance,  $D^2$ , and computed as

$$D^2 = (\mathbf{y} - \bar{\mathbf{y}}_i)' \mathbf{S}^{-1} (\mathbf{y} - \bar{\mathbf{y}}_i) = \mathbf{y}' \mathbf{S}^{-1} \mathbf{y} - 2\mathbf{y}' \mathbf{S}^{-1} \bar{\mathbf{y}}_i + \bar{\mathbf{y}}_i' \mathbf{S}^{-1} \bar{\mathbf{y}}_i$$

where

$$\mathbf{S} = \frac{\mathbf{E}}{N-1}$$

In saving discriminant columns,  $N$  is the number of observations and  $M$  is the identity matrix.

The **Save Discrim** command in the popup menu on the platform title bar saves discriminant scores with their formulas as columns in the current data table.  $\text{SqDist}[0]$  is a quadratic form needed in all the distance calculations. It is the portion of the Mahalanobis distance formula that does not vary across groups.  $\text{SqDist}[i]$  is the Mahalanobis distance of an observation from the  $i$ th centroid.  $\text{SqDist}[0]$  and  $\text{SqDist}[i]$  are calculated as

$$\text{SqDist}[0] = \mathbf{y}' \mathbf{S}^{-1} \mathbf{y}$$

and

$$\text{SqDist}[i] = \text{SqDist}[0] - 2\mathbf{y}' \mathbf{S}^{-1} \bar{\mathbf{y}}_i + \bar{\mathbf{y}}_i' \mathbf{S}^{-1} \bar{\mathbf{y}}_i$$

Assuming that each group has a multivariate normal distribution, the posterior probability that an observation belongs to the  $i$ th group is

$$\text{Prob}[i] = \frac{\exp(\text{Dist}[i])}{\text{Prob}[0]}$$

where

$$\text{Prob}[0] = \sum e^{-0.5 \text{Dist}[i]}$$

## Power Calculations

The next sections give formulas for computing LSV, LSN, power, and adjusted power.

### Computations for the LSV

For one-degree-freedom tests, the LSV is easy to calculate. The formula for the  $F$ -test for the hypothesis  $\mathbf{L}\beta = 0$  is:

$$F = \frac{(\mathbf{L}\mathbf{b})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{L}\mathbf{b})/r}{s^2}$$

Solving for  $\mathbf{L}\mathbf{b}$ , a scalar, given an  $F$  for some  $\alpha$ -level, like 0.05, and using a  $t$  as the square root of a one-degree-of-freedom  $F$ , making it properly two-sided, gives

$$(\mathbf{L}\mathbf{b})^{\text{LSV}} = t_{\alpha/2} s \sqrt{\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'}$$

For  $\mathbf{L}$  testing some  $\beta_i$ , this is

$$b_i^{\text{LSV}} = t_{\alpha/2} s \sqrt{((\mathbf{X}'\mathbf{X})^{-1})_{ii}}$$

which can be written with respect to the standard error as

$$b_i^{\text{LSV}} = t_{\alpha/2} \text{stderr}(b_i)$$

If you have a simple model of two means where the parameter of interest measures the difference between the means, this formula is the same as the LSD, least significant difference, from the literature

$$\text{LSD} = t_{\alpha/2} s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

In the JMP Fit Model platform, the parameter for a nominal effect measures the difference to the average of the levels, not to the other mean. So, the LSV for the parameter is half the LSD for the differences of the means.

## Computations for the LSN

The LSN solves for  $n$  in the equation:

$$\alpha = 1 - \text{ProbF} \left[ \frac{n\delta^2}{dfH\sigma^2}, dfH, n - dfR - 1 \right]$$

where

$\text{ProbF}$  is the central  $F$ -distribution function

$dfH$  is the degrees of freedom for the hypothesis

$dfR$  is the degrees of freedom for regression in the whole model

$\sigma^2$  is the error variance

$\delta^2$  is the squared effect size, which can be estimated by  $\frac{\text{SS}(H)}{n}$

When planning a study, the LSN serves as the lower bound.

## Computations for the Power

To calculate power, first get the critical value for the central  $F$  by solving for  $F_C$  in the equation

$$\alpha = 1 - \text{ProbF}[F_C, dfH, n - dfR - 1]$$

Then obtain the probability of getting this critical value or higher

$$\text{Power} = 1 - \text{ProbF} \left[ F_C, dfH, n - dfR - 1, \frac{n\delta^2}{\sigma^2} \right]$$

The last argument to  $\text{ProbF}$  is the noncentrality value  $\lambda = \frac{n\delta^2}{\sigma^2}$ , which can be estimated as  $\frac{\text{SS}(H)}{\hat{\sigma}^2}$  for retrospective power.

## Computations for Adjusted Power

The adjusted power is a function a noncentrality estimate that has been adjusted to remove positive bias in the original estimate (Wright and O'Brien 1988). Unfortunately, the adjustment to the noncentrality estimate can lead to negative values. Negative values are handled by letting the adjusted noncentrality estimate be

$$\hat{\lambda}_{\text{adj}} = \text{Max} \left[ 0, \frac{\hat{\lambda}(N - dfR - 1 - 2)}{N - dfR - 1} - dfH \right]$$

where  $N$  is the actual sample size,  $dfH$  is the degrees of freedom for the hypothesis,  $dfR$  is the degrees of freedom for regression in the whole model,  $F_s$  is a  $F$ -value calculated from data, and  $F_C$  is the critical value for the  $F$ -test.

The adjusted power is

$$\text{Power}_{\text{adj}} = 1 - \text{ProbF}[F_C, dfH, n - dfR - 1, \hat{\lambda}_{\text{adj}}].$$

Confidence limits for the noncentrality parameter are constructed according to Dwass (1955) as

$$\text{Lower CL for } \lambda = dfH(\text{Max}[0, \sqrt{F_s} - \sqrt{F_C}])^2$$

$$\text{Upper CL for } \lambda = dfH(\sqrt{F_s} - \sqrt{F_C})^2$$

Power confidence limits are obtained by substituting confidence limits for  $\lambda$  into

$$\text{Power} = 1 - \text{ProbF}[F_C, dfH, n - dfR - 1, \lambda]$$

## Inverse Prediction with Confidence Limits

Inverse prediction estimates a value of an independent variable from a response value. In bioassay problems, inverse prediction with confidence limits is especially useful. In JMP, you can request inverse prediction estimates for continuous and binary response models. If the response is continuous, you can request confidence limits for an individual response or an expected response.

The confidence limits are computed using Fieller's theorem, which is based on the following logic. The goal is predicting the value of a single regressor and its confidence limits given the values of the other regressors and the response.

Let  $\mathbf{b}$  estimate the parameters  $\beta$  so that we have  $\mathbf{b}$  distributed as  $N(\beta, V)$ .

Let  $\mathbf{x}$  be the regressor values of interest, with the  $i$ th value to be estimated.

Let  $y$  be the response value.

We desire a confidence region on the value of  $\mathbf{x}[i]$  such that  $\beta' \mathbf{x} = y$  with all other values of  $\mathbf{x}$  given.

The inverse prediction is just

$$x[i] = \frac{y - \beta'_{(i)} x_{(i)}}{\beta[i]}$$

where the parenthesized-subscript " $(i)$ " means that the  $i$ th component is omitted. A confidence interval can be formed from the relation:

$$(y - b'x)^2 < t^2 x'Vx$$

with specified confidence if  $y = \beta'x$ . A quadratic equation results of the form

$$z^2 g + zh + f = 0$$

where

$$g = b[i]^2 - t^2 V[i, i]$$

$$h = -2yb[i] + 2b[i]b'_{(i)}x_{(i)} - 2t^2V[i, (i)]'x_{(i)}$$

$$f = y^2 - 2yb'_{(i)}x_{(i)} + (b'_{(i)}x_{(i)})^2 - t^2x_{(i)}'V_{(i)}x_{(i)}$$

It is possible for the quadratic equation to have only imaginary roots, in which case the confidence interval becomes the entire real line. In this situation, Wald intervals are used. If only one side of an interval is imaginary, the Fieller method is still used, but the imaginary side is returned as missing.

**Note:** The Logistic platform in JMP uses  $t$  values when computing the confidence intervals for inverse prediction. PROC PROBIT in SAS/STAT and the Generalized Linear Model platform in JMP use  $z$  values, which give slightly different results.

## Details of Random Effects

The variance matrix of the fixed effects is always modified to include a Kackar-Harville correction. The variance matrix of the BLUPs and the covariances between the BLUPs and the fixed effects are not Kackar-Harville corrected because the corrections for these can be quite computationally intensive and use lots of memory when there are lots of levels of the random effects. In SAS, the Kackar-Harville correction is done for both fixed effects and BLUPs only when the `DDFM=KR` is set, so the standard errors from PROC MIXED with this option will differ from all the other options.

This implies:

- Standard errors for linear combinations involving only fixed effects parameters will match PROC MIXED `DDFM=KR`, assuming that one has taken care to transform between the different parameterizations used by PROC MIXED and JMP.
- Standard errors for linear combinations involving only BLUP parameters will match PROC MIXED `DDFM=SATTERTH`.
- Standard errors for linear combinations involving both fixed effects and BLUPS do not match PROC MIXED for any `DDFM` option if the data are unbalanced. However, these standard errors are between what you get with the `DDFM=SATTERTH` and `DDFM=KR` options. If the data are balanced, JMP matches SAS for balanced data, regardless of the `DDFM` option, since the Kackar-Harville correction is null.

The degrees of freedom for tests involving only linear combinations of fixed effects parameters are calculated using the Kenward and Roger correction, so JMP's results for these tests match PROC MIXED using the `DDFM=KR` option. If there are BLUPs in the linear combination, JMP uses a Satterthwaite approximation to get the degrees of freedom. So, the results follow a pattern similar to what is described for standard errors in the preceding paragraph.

For more details of the Kackar-Harville correction and the Kenward-Roger DF approach, see Kenward and Roger(1997). The Satterthwaite method is described in detail in the SAS PROC MIXED documentation.

## References

---

- Abernethy, Robert B. (1996) *The New Weibull Handbook*. Published by the author: 536 Oyster Road North Palm Beach, Florida 33408.
- Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley and Sons, Inc.
- Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley and Sons, Inc.
- Agresti, A., and Coull, B. (1998), “Approximate is Better Than ‘Exact’ for Interval Estimation of Binomial Proportions,” *The American Statistician*, 52, 119–126
- Aitken, M. (1987) “Modelling Variance Heterogeneity in Normal Regression Using GLIM,” *Applied Statistics* 36:3, 332–339.
- Akaike, H. (1974), “Factor Analysis and AIC,” *Psychometrika*, 52, 317–332.
- Akaike, H. (1987), “A new Look at the Statistical Identification Model,” *IEEE Transactions on Automatic Control*, 19, 716–723.
- American Society for Quality Statistics Division (2004), *Glossary and Tables for Statistical Quality Control*, Fourth Edition, Milwaukee: Quality Press.
- Anderson, T.W. (1971), *The Statistical Analysis of Time Series*, New York: John Wiley and Sons.
- Anderson, T. W. (1958) *An Introduction to Multivariate Statistical Analysis*. New York: John Wiley & Sons.
- Andrews, D.F. and A. M. Herzberg (1985), *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York: Springer-Verlag.
- Ashton, W.D. (1972), “The Logit Transformation,” *Griffin’s Statistical Monographs*, New York: Hafner Publishing.
- Atkinson, A.C. (1985), *Plots, Transformations and Regression*, Oxford: Clarendon Press.
- Barlow, R.E., Bartholomew, D.J., Bremner, J.M., and Brunk, H.D. (1972), *Statistical Inference under Order Restrictions*, New York: John Wiley and Sons, Inc.
- Barrentine (1991), *Concepts for R&R Studies*, Milwaukee, WI: ASQC Quality Press.
- Bartlett, M.S. and D.G. Kendall (1946), “The Statistical Analysis of Variances—Heterogeneity and the Logarithmic Transformation,” *JRSS Suppl* 8, 128–138.
- Bartlett, M.S. (1966), *An Introduction to Stochastic Processes*, Second Edition, Cambridge: Cambridge University Press.
- Bates, D.M. and Watts, D.G. (1988), *Nonlinear Regression Analysis & its Applications*. New York, John Wiley and Sons.
- Beaton, A.E. and Tukey, J.W. (1974), “The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data,” *Technometrics* 16, 147–180.

- Becker, R.A. and Cleveland, W.S. (1987), "Brushing Scatterplots," *Technometrics*, 29, 2.
- Berger, R.L., and Hsu, J.C. (1996), "Bioequivalence Trails, Intersection-Union Tests and Equivalence Confidence Sets," *Statistical Science*, 11, 283–319.
- Belsley, D.A., Kuh, E., and Welsch, R.E. (1980), *Regression Diagnostics*, New York: John Wiley and Sons.
- Ben-Akiva, M. and Lerman, S.R. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*, Cambridge: MIT Press.
- Benzecri, J.P. (1973), "L'Analyse des Donnees," *l'analyse des Correspondances*, Paris: Dunod.
- Bissell, A. F. (1990), "How Reliable is Your Capability Index?", *Applied Statistics*, 30, 331-340.
- Bowman, A. and Foster, P. (1992) "Density Based Exploration of Bivariate Data," Dept. of Statistics, Univ. of Glasgow, Tech Report No 92-1.
- Bowman, A. and Schmee, J. (2004) "Estimating Sensitivity of Process Capability Modeled by a Transfer Function" *Journal of Quality Technology*, v36, no.2 (April)
- Box, G. E. P. (1954). "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, II: Effects of Inequality of Variance and Correlation Between Errors in the Two-Way Classification". *Annals of Mathematical Statistics*, 1, 69-82.
- Box, G.E.P. (1988), "Signal-to-Noise Ratio, Performance Criteria, and Transformations," *Technometrics* 30, 1–40.
- Box, G.E.P. and Cox, D.R. (1964), "An Analysis of Transformations," *JRSS B*26, 211–243.
- Box, G.E.P. and Draper, N.R. (1969), *Evolutionary Operation: A Statistical Method for Process Improvement*, New York: John Wiley and Sons.
- Box, G.E.P. and Draper, N.R. (1987), *Empirical Model-Building and Response Surfaces*, New York: John Wiley and Sons.
- Box, G.E.P. and Meyer, R.D. (1986), "An analysis of Unreplicated Fractional Factorials," *Technometrics* 28, 11–18.
- Box, G.E.P. and Meyer, R. D. (1993), "Finding the Active Factors in Fractionated Screening Experiments.", *Journal of Quality Technology* Vol.25 #2: 94–105.
- Box, G.E.P., Hunter,W.G., and Hunter, J.S. (1978), *Statistics for Experimenters*, New York: John Wiley and Sons, Inc.
- Brown, M.B. and Forsythe, A.B. (1974a), "The Small Sample Behavior of Some Statistics Which Test the Equality of Several Means," *Technometrics* 16:1, 129–132.
- Brown, M.B. and Forsythe, A.B. (1974), "Robust tests for the equality of variances" *Journal of the American Statistical Association*, 69, 364–367.
- Byrne, D.M. and Taguchi, G. (1986), *ASQC 40th Anniversary Quality Control Congress Transactions*, Milwaukee, WI: American Society of Quality Control, 168–177.
- Carroll, R.J. and Ruppert, D. (1988), *Transformation and Weighting in Regression*, New York: Chapman and Hall.
- Carroll, R.J., Ruppert, D. and Stefanski, L.A. (1995), *Measurement Error in Nonlinear Models*, New York: Chapman and Hall.
- Cobb, G.W. (1998), *Introduction to Design and Analysis of Experiments*, Springer-Verlag: New York.

- Cohen, J. (1960), "A coefficient of agreement for nominal scales," *Education Psychological Measurement*, 20: 37–46.
- Cole, J.W.L. and Grizzle, J.E. (1966), "Applications of Multivariate Analysis of Variance to Repeated Measures Experiments," *Biometrics*, 22, 810–828.
- Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley and Sons.
- Conover, W. J. (1972). "A Kolmogorov Goodness-of-fit Test for Discontinuous Distributions". *Journal of the American Statistical Association* 67: 591–596.
- Conover, W.J. (1980), *Practical Nonparametric Statistics*, New York: John Wiley and Sons, Inc.
- Cook, R.D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman and Hall.
- Cook, R.D. and Weisberg, S. (1983), "Diagnostics for heteroscedasticity in regression" *Biometrika* 70, 1–10.
- Cornell, J.A. (1990), *Experiments with Mixtures*, Second Edition, New York: John Wiley and Sons.
- Cox, D.R. (1970), *The Analysis of Binary Data*, London: Methuen.
- Cox, D.R. (1972), "Regression Models And Life-tables", *Journal Of The Royal Statistical Society Series B-statistical Methodology*. 34 (2): 187–220, 1972.
- Cronbach, L.J. (1951), "Coefficient Alpha and the Internal Structure of Tests," *Psychometrika*, 16, 297–334.
- Daniel C. and Wood, F. (1980), *Fitting Equations to Data*, Revised Edition, New York: John Wiley and Sons, Inc.
- Daniel, C. (1959), "Use of Half-normal Plots in Interpreting Factorial Two-level Experiments," *Technometrics*, 1, 311–314.
- Davis, H.T. (1941), *The Analysis of Economic Time Series*, Bloomington, IN: Principia Press.
- DeLong, E., Delong, D, and Clarke-Pearson, D.L. (1988), "Comparing the Areas Under Two or more Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach," *Biometrics* 44, 837–845.
- Derringer, D. and Suich, R. (1980), "Simultaneous Optimization of Several Response Variables," *Journal of Quality Technology*, 12:4, 214–219.
- Devore, J. L. (1995), *Probability and Statistics for Engineering and the Sciences*, Duxbury Press, CA.
- Do, K-A, and McLachlan G.J. (1984). Estimation of mixing proportions: a case study. *Journal of the Royal Statistical Society, Series C*, 33: 134-140.
- Draper, N. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley and Sons, Inc.
- Dunnett, C.W. (1955), "A multiple comparison procedure for comparing several treatments with a control" *Journal of the American Statistical Association*, 50, 1096–1121.
- Dwass, M. (1955), "A Note on Simultaneous Confidence Intervals," *Annals of Mathematical Statistics* 26: 146–147.
- Eppright, E.S., Fox, H.M., Fryer, B.A., Lamkin, G.H., Vivian, V.M., and Fuller, E.S. (1972), "Nutrition of Infants and Preschool Children in the North Central Region of the United States of America," *World Review of Nutrition and Dietetics*, 14.

- Eubank, R.L. (1999), *Nonparametric Regression and Spline Smoothing*, Second Edition, Boca Raton, Florida: CRC.
- Farebrother, R.W. (1981), "Mechanical Representations of the L1 and L2 Estimation Problems," *Statistical Data Analysis*, 2nd Edition, Amsterdam, North Holland: edited by Y. Dodge.
- Firth, D. (1993), "Bias Reduction of Maximum Likelihood Estimates," *Biometrika* 80:1, 27–38.
- Fisher, L. and Van Ness, J.W. (1971), "Admissible Clustering Procedures," *Biometrika*, 58, 91–104.
- Fisherkeller, M.A., Friedman, J.H., and Tukey, J.W. (1974), "PRIM-9: An Interactive Multidimensional Data Display and Analysis System," SLAC-PUB-1408, Stanford, California: Stanford Linear Accelerator Center.
- Fleis, J.L., Cohen J., and Everitt, B.S. (1969), "Large-sample standard errors of kappa and weighted kappa," *Psychological Bulletin*, 72: 323–327.
- Fleiss, J. L. (1981). Statistical Methods for Rates and Proportions. New York: John Wiley and Sons.
- Florek, K., Lukaszewicz, J., Perkal, J., and Zubrzycki, S. (1951a), "Sur La Liaison et la Division des Points d'un Ensemble Fini," *Colloquium Mathematicae*, 2, 282–285.
- Foster, D.P., Stine, R.A., and Waterman, R.P. (1997), *Business Analysis Using Regression*, New York, Springer-Verlag.
- Foster, D.P., Stine, R.A., and Waterman, R.P. (1997), *Basic Business Statistics*, New York, Springer-Verlag.
- Friendly, M. (1991), "Mosaic Displays for Multiway Contingency Tables," *New York University Department of Psychology Reports*: 195.
- Fuller, W.A. (1976), *Introduction to Statistical Time Series*, New York, John Wiley and Sons.
- Fuller, W.A. (1987), *Measurement Error Models*, New York, John Wiley and Sons.
- Gabriel, K.R. (1982), "Biplot," *Encyclopedia of Statistical Sciences*, Volume 1, eds. N.L.Johnson and S. Kotz, New York: John Wiley and Sons, Inc., 263–271.
- Gallant, A.R. (1987), *Nonlinear Statistical Models*, New York, John Wiley and Sons.
- Giesbrecht, F.G. and Gumpertz, M.L. (2004). *Planning, Construction, and Statistical Analysis of Comparative Experiments*. New York: John Wiley & Sons.
- Goodnight, J.H. (1978), "Tests of Hypotheses in Fixed Effects Linear Models," *SAS Technical Report R-101*, Cary: SAS Institute Inc, also in *Communications in Statistics* (1980), A9 167–180.
- Goodnight, J.H. and W.R. Harvey (1978), "Least Square Means in the Fixed Effect General Linear Model," *SAS Technical Report R-103*, Cary NC: SAS Institute Inc.
- Greenacre, M.J. (1984), *Theory and Applications of Correspondence Analysis*, London: Academic Press.
- Greenhouse, S. W. and Geiser, S. (1959). "On Methods in the Analysis of Profile Data." *Psychometrika*, 32, 95–112.
- Gupta, S.S. (1965), On Some Multiple Decision (selection and ranking), Rules., *Technometrics* 7, 225–245.
- Haaland, P.D. (1989), *Experimental Design in Biotechnology*, New York: Marcel Dekker, Inc.
- Hahn, G.J. (1976), "Process Improvement Through Simplex EVOP," *Chemical Technology* 6, 243–345.
- Hahn, G. J. and Meeker, W. Q. (1991) *Statistical Intervals: A Guide for Practitioners*. New York: Wiley.
- Hajek, J. (1969), *A Course in Nonparametric Statistics*, San Francisco: Holden-Day.

- Harrell, F. (1986), "The Logist Procedure," *SUGI Supplemental Library User's Guide*, Version 5 Edition, Cary, NC: SAS Institute Inc.
- Harris, R.J. (1975), *A Primer of Multivariate Statistics*, New York: Academic Press.
- Hartigan, J.A. (1975), *Clustering Algorithms*, New York, John Wiley and Sons.
- Hartigan, J.A. (1981), "Consistence of Single Linkage for High-Density Clusters," *Journal of the American Statistical Association*, 76, 388–394.
- Hartigan, J.A. and Kleiner, B. (1981), "Mosaics for Contingency Tables," *Proceedings of the 13th Symposium on the Interface between Computer Science and Statistics*, Ed. Eddy, W. F., New York: Springer–Verlag, 268–273.
- Harvey, A.C. (1976), "Estimating Regression Models with Multiplicative Heteroscedasticity," *Econometrica* 44–3 461–465.
- Hauck, W.W. and Donner, A. (1977), "Wald's Test as Applied to Hypotheses in Logit Analysis," *Journal of the American Statistical Association*, 72, 851–863.
- Hawkins, D.M. (1974), "The Detection of Errors in Multivariate Data Using Principal Components," *Journal of the American Statistical Association*, 69.
- Hayashi, C. (1950), "On the Quantification of Qualitative Data from the Mathematico–Statistical Point of View," *Annals of the Institute of Statistical Mathematics*, 2:1, 35–47.
- Hayter, A.J. (1984), "A proof of the conjecture that the Tukey–Kramer multiple comparisons procedure is conservative," *Annals of Mathematical Statistics*, 12 61–75.
- Heinze, G. and Schemper, M. (2002), "A Solution to the Problem of Separation in Logistic Regression," *Statistics in Medicine* 21:16, 2409–2419.
- Henderson, C.R. (1984), *Applications of Linear Models in Animal Breeding*, Univ. of Guelph.
- Hocking, R.R. (1985), *The Analysis of Linear Models*, Monterey: Brooks–Cole.
- Hoeffding, W (1948), "A Non-Parametric Test of Independence", *Annals of Mathematical Statistics*, 19, 546–557.
- Hoffman, Heike (2001). "Generalized Odds Ratios for Visual Modeling," *Journal of Computational and Graphical Statistics*, 10:4 pp. 628–640.
- Holland, P.W. and Welsch, R.E. (1977), "Robust Regression Using Iteratively Reweighted Least Squares," *Communications Statistics: Theory and Methods*, 6, 813–827.
- Hooper, J. H. and Amster, S. J. (1990), "Analysis and Presentation of Reliability Data," *Handbook of Statistical Methods for Engineers and Scientists*, Harrison M. Wadsworth, editor. New York: McGraw Hill.
- Hosmer, D.W. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley and Sons.
- "Hot Dogs," (1986), *Consumer Reports* (June), 364–367.
- Hsu, J. (1981), "Simultaneous confidence intervals for all distances from the 'best,'" *Annals of Statistics*, 9, 1026–1034.
- Hsu, J. (1984), "Constrained two-sided simultaneous confidence intervals for multiple comparisons with the 'best,'" *Annals of Statistics*, 12, 1136–1144.
- Hsu, J. (1989), "Multiple Comparison Procedures" ASA Short Course notes, Columbus OH: Ohio State University.

- Hsu, J. (1989), *Tutorial Notes on Multiple Comparisons*, American Statistical Association, Washington, DC.
- Hunter, J.S. (1985), "Statistical Design Applied to Product Design," *Journal of Quality Technology*, 17, 210–221.
- Huynh, H. and Feldt, L. S. (1970). "Conditions under which Mean Square Ratios in Repeated Measurements Designs have Exact *F*-Distributions." *Journal of the American Statistical Association*, 65, 1582–1589.
- Huynh, H. and Feldt, L. S. (1976). "Estimation of the Box Correction for Degrees of Freedom from Sample Data in the Randomized Block Split Plot Designs." *Journal of Educational Statistics*, 1, 69–82.
- Iman, R.L. (1974), "Use of a t-statistic as an Approximation to the Exact Distribution of Wilcoxon Signed Ranks Test Statistic," *Communications in Statistics—Simulation and Computation*, 795–806.
- Inselberg, A. (1985) "The Plane with Parallel Coordinates." *Visual Computing* 1. pp 69–91.
- Jardine, N. and Sibson, R. (1971), *Mathematical Taxonomy*, New York: John Wiley and Sons.
- John, P.W.M. (1971), *Statistical Design and Analysis of Experiments*, New York: Macmillan Publishing Company, Inc.
- Johnson, M.E. and Nachtsheim, C.J. (1983), "Some Guidelines for Constructing Exact D–Optimal Designs on Convex Design Spaces," *Technometrics* 25, 271–277.
- Johnson, N.L. (1949). *Biometrika*, 36, 149-176.
- Judge, G.G., Griffiths,W.E., Hill,R.C., and Lee, Tsoung–Chao (1980), *The Theory and Practice of Econometrics*, New York: John Wiley and Sons.
- Kalbfleisch, J.D. and Prentice, R.L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley and Sons.
- Kalbfleisch, J.D. and Prentice, R.L. (2002), 2nd Edition, *The Statistical Analysis of Failure Time Data*, New York: John Wiley and Sons, pp 71-73.
- Kackar, R.N. and Harville, D.A. (1984), Approximations for standard errors of estimators of fixed and random effects in mixed linear models, *Journal of the American Statistical Association*, 79, 853-862.
- Kaiser, H.F. (1958), "The varimax criterion for analytic rotation in factor analysis" *Psychometrika*, 23, 187–200.
- Kenward, M.G. and Roger, J.H. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 53, 983-997.
- Khuri, A.I. and Cornell J.A. (1987), *Response Surfaces: Design and Analysis*, New York: Marcel Dekker.
- Koehler, M.G., Grigoras, S., and Dunn, J.D. (1988), "The Relationship Between Chemical Structure and the Logarithm of the Partition Coefficient," *Quantitative Structure Activity Relationships*, 7.
- Kohonen, Teuvo. (1989) *Self-Organization and Associative Memory*. Berlin: Springer.
- Kramer, C.Y. (1956), "Extension of multiple range tests to group means with unequal numbers of replications," *Biometrics*, 12, 309–310.
- Lawless, J.F. (1982), *Statistical Models and Methods for Lifetime Data*, New York: John Wiley and Sons.
- Lawless, J.F. (2003), *Statistical Models and Methods for Lifetime Data, 2nd Edition*, pp. 33–34. New York: John Wiley and Sons.
- Lebart, L., Morineau, A., and Tabaard, N. (1977), *Techniques de la Description Statistique*, Paris: Dunod.

- Lee, E.T. (1980), *Statistical Methods for Survival Data Analysis*, Belmont CA, Lifetime Learning Publications, a Division of Wadsworth, Inc.
- Lenth, R.V. (1989), "Quick and Easy Analysis of Unreplicated Fractional Factorials," *Technometrics*, 31, 469–473.
- Levene, J.R., Serlin, R.C., and Webne-Behrman, L. (1989), "Analysis of Variance Through Simple Correlation," *American Statistician*, 43, (1), 32–34.
- Levene, H. (1960), "Robust tests for the equality of variances" In I. Olkin (ed), *Contributions to probability and statistics*, Stanford Univ. Press.
- Linnerud: see Rawlings (1988).
- Louviere, J.J., Hensher, D.A. and Swait, J.D. (2000), *Stated Choice Methods: Analysis and Application*, Cambridge: Cambridge University Press.
- Lucas, J.M. (1976), "The Design and Use of V-Mask Control Schemes," *Journal of Quality Technology*, 8, 1–12.
- MacQueen, J.B. (1967) (1967) "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.
- Mallows, C.L. (1967), "Choosing a Subset Regression," unpublished report, Bell Telephone Laboratories.
- Mallows, C.L. (1973), "Some Comments on Cp," *Technometrics*, 15, 661–675.
- Mardia, K.V., Kent, J.T., and Bibby J.M. (1979). *Multivariate Analysis*, New York: Academic Press.
- Marsaglia, G. (1996) DIEHARD: A Battery of Tests of Randomness". <http://stat.fsu.edu/~geo>.
- Mason, R.L. and Young, J.C. (2002), *Multivariate Statistical Process Control with Industrial Applications*, Philadelphia: ASA-SIAM.
- Matsumoto, M. and Nishimura, T. (1998)"Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator", *ACM Transactions on Modeling and Computer Simulation*, Vol. 8, No. 1, January 1998, 3–f30.
- McFadden, D. (1974), "Conditional Logit Analysis of Qualitative Choice Behavior," in P. Zarembka, ed., *Frontiers in Econometrics*, pp. 105–142.
- McLachlan, G.J. and Krishnan, T. (1997), *The EM Algorithm and Extensions*, New York: John Wiley and Sons.
- McCullagh, P. and Nelder, J.A. (1983), *Generalized Linear Models*, London: Chapman and Hall Ltd.
- McCulloch, C.E., Searle, S.R., and Neuhaus, J.M. (2008), *Generalized, Linear, and Mixed Models*, New York: John Wiley and Sons.
- McQuitty, L.L. (1957), "Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies," *Educational and Psychological Measurement*, 17, 207–229.
- Meeker, W.Q. and Escobar, L.A. (1998), *Statistical Methods for Reliability Data*, pp. 60–62, New York: John Wiley and Sons.
- Myers, R.H. (1976), *Response Surface Methodology*, Boston: Allyn and Bacon.
- Myers, R.H. (1988), *Response Surface Methodology*, Virginia Polytechnic and State University.
- Myers, R.H. (1989), *Classical and Modern Regression with Applications*, Boston: PWS-KENT.

- Meyer, R.D., Steinberg, D.M., and Box, G. (1996), "Follow-up Designs to Resolve Confounding in Multifactor Experiments," *Technometrics*, Vol. 38, #4, p307
- Miller, A.J. (1990), *Subset Selection in Regression*, New York: Chapman and Hall.
- Milligan, G.W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, 45, 325–342.
- Milliken, G.A. and Johnson, E.J. (1984), *Analysis of Messy Data Volume I: Design of Experiments*, New York: Van Nostrand Reinhold Company.
- Montgomery, D.C. and Peck, E.A. (1982), *Introduction to Linear Regression Analysis*, New York: John Wiley and Sons.
- Montgomery, D. C. (1991), "Using Fractional Factorial Designs for Robust Process Development," *Quality Engineering*, 3, 193–205.
- Montgomery, D. C. (1996) *Introduction to Statistical Quality Control*, 3rd edition. New York: John Wiley.
- Montgomery, D.C. (2001), *Introduction to Statistical Quality Control*, 4th Edition New York: John Wiley and Sons.
- Moore, D.S. and McCabe, G. P. (1989), *Introduction to the Practice of Statistics*, New York and London: W. H. Freeman and Company.
- Mosteller, F. and Tukey, J.W. (1977), *Data Analysis and Regression*, Reading, MA: Addison–Wesley.
- Muller, K.E. and Barton, C.N. (1989), "Approximate Power for Repeated-measures ANOVA Lacking Sphericity," *Journal of the American Statistical Association*, 84, 549–555.
- Myers, R. H. and Montgomery, D. C. (1995), *Response Surface Methodology*, New York: John Wiley and Sons.
- Nelder, J.A. and Mead, R. (1965), "A Simplex Method for Function Minimization," *The Computer Journal*, 7, 308–313.
- Nelder, J.A. and Wedderburn, R.W.M. (1983), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 370–384.
- Nelson, F. (1976), "On a General Computer Algorithm for the Analysis of Model with Limited Dependent Variables," *Annals of Economic and Social Measurement*, 5/4.
- Nelson, L. (1984), "The Shewhart Control Chart—Tests for Special Causes," *Journal of Quality Technology*, 15, 237–239.
- Nelson, L. (1985), "Interpreting Shewhart X Control Charts," *Journal of Quality Technology*, 17, 114–116.
- Nelson, W.B. (1982), *Applied Life Data Analysis*, New York: John Wiley and Sons.
- Nelson, W.B. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data analysis*, New York: John Wiley and Sons.
- Nelson, W.B. (2003), *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*, Philadelphia: Society for Industrial Mathematics.
- Neter, J., Wasserman, W. and Kutner, M.H. (1990), *Applied Linear Statistical Models*, Third Edition, Boston: Irwin, Inc.
- Nunnaly, J. C. (1978) *Psychometric Theory*, 2nd Ed., New York: McGraw-Hill.

- O'Brien, R.G. (1979), "A general ANOVA method for robust tests of additive models for variances," *Journal of the American Statistical Association*, 74, 877–880.
- O'Brien, R., and Lohr, V. (1984), "Power Analysis For Linear Models: The Time Has Come," *Proceedings of the Ninth Annual SAS User's Group International Conference*, 840–846.
- Odeh, R. E. and Owen, D. B. (1980) *Tables for Normal Tolerance Limits, Sampling Plans, and Screening*. New York: Marcel Dekker, Inc.
- Olejnik, S.F. and Algina, J. (1987), "Type I Error Rates and Power Estimates of Selected Parametric and Nonparametric Tests of Scale," *Journal of Educational Statistics* 12, 45–61.
- Olson, C.L. (1976), "On Choosing a Test Statistic in MANOVA," *Psychological Bulletin* 83, 579–586.
- Patterson, H. D. and Thompson, R. (1974). Maximum likelihood estimation of components of variance. *Proc. Eighth International Biochem. Conf.*, 197–209.
- Piepel, G.F. (1988), "Programs for Generating Extreme Vertices and Centroids of Linearly Constrained Experimental Regions," *Journal of Quality Technology* 20:2, 125–139.
- Plackett, R.L. and Burman, J.P. (1947), "The Design of Optimum Multifactorial Experiments," *Biometrika*, 33, 305–325.
- Poduri, S.R.S. Rao (1997), *Variance Components: Mixed Models, Methodologies and Applications (Monographs on Statistics and Applied Probability)*, New York, Chapman & Hall.
- Portnoy, Stephen (1971), "Formal Bayes Estimation with Application to a Random Effects Model", *The Annals of Mathematical Statistics*, Vol. 42, No. 4, pp. 1379–1402.
- Prentice, R.L. (1973), "Exponential survivals with censoring and explanatory variables," *Biometrika*, 60:2, 279–288.
- Ratkowsky, D.A. (1990), *Handbook of Nonlinear Regression Models*, New York, Marcel Dekker, Inc.
- Rawlings, J.O. (1988), *Applied Regression Analysis: A Research Tool*, Pacific Grove CA: Wadsworth and Books/Cole.
- Reinsch, C.H. (1967), *Smoothing by Spline Functions*, Numerische Mathematik, 10, 177–183.
- Robertson, T., Wright, F.T., and R.L. Dykstra, R.L (1988), *Order Restricted Statistical Inference*, New York: John Wiley and Sons, Inc.
- Rodriguez, R.N. (1990), "Selected SAS/QC Software Examples, Release 6.06," SAS Institute Inc., Cary, NC.
- Rodriguez, R.N. (1991), "Applications of Computer Graphics to Two Basic Statistical Quality Improvement Methods," National Computer Graphics Association Proceedings, 17–26.
- Rosenbaum, P.R. (1989), "Exploratory Plots for Paired Data," *American Statistician*, 108–109.
- Rousseeuw, P.J. and Leroy, A.M. (1987), *Robust Regression and Outlier Detection*, New York: John Wiley and Sons.
- Royston, J.P. (1982), "An Extension of Shapiro and Wilk's W Test for Normality to Large Samples," *Applied Statistics* 31, 115–124.
- Sahai, Hardeo (1974), "Some Formal Bayes Estimators of Variance Components in the Balanced Three-Stage Nested Random Effects Model", *Communication in Statistics – Simulation and Computation*, 3:3, 233–242.

- Sall, J.P. (1990), "Leverage Plots for General Linear Hypotheses," *American Statistician*, 44, (4), 308–315.
- Santer, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, New York, Springer-Verlag New York, Inc.
- SAS Institute Inc. (1995), *SAS/QC Software: Usage and References, Version 6*, 1st Ed., Vol. 1, SAS Institute Inc., Cary, NC.
- SAS Institute Inc. (1986), *SAS/QC User's Guide*, Version 5 Edition, SAS Institute Inc., Cary, NC.
- SAS Institute Inc. (1987), *SAS/STAT Guide for Personal Computers*, Version 6 Edition, Cary NC: SAS Institute Inc.
- SAS Institute Inc. (1999), *SAS/ETS User's Guide, Version 8*, Cary NC: SAS Institute Inc.
- SAS Institute Inc. (1989), "SAS/ Technical Report P-188: SAS/QC Software Examples, Version 6 Edition," SAS Institute Inc., Cary, NC.
- SAS Institute Inc. (1999), *SAS/STAT User's Guide, Version 8*, Cary, NC: SAS Institute Inc.
- SAS Institute Inc. (1996), "SAS/STAT Software, Changes and Enhancements through Version 6.11, The Mixed Procedure, Cary, NC: SAS Institute Inc.
- SAS Institute Inc. (2004), *SAS/STAT User's Guide, Version 9.1*, Cary, NC: SAS Institute Inc.
- Satterthwaite, F.E., (1946), "An approximate distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.
- Scheffé, H. (1958) "Experiments with Mixtures", *Journal of the Royal Statistical Society B* v20, 344–360.
- Searle, S. R, Casella, G. and McCulloch, C. E. (1992) *Variance Components*. New York: John Wiley and Sons.
- Seber, G.A.F. (1984), *Multivariate Observations*, New York: John Wiley and Sons, 413–416.
- Seder, L.A. (1950) "Diagnosis with Diagrams, Part I and Part II, Industrial Quality Control," 6, 11–19, 7–11 reprinted in Quality Engineering 2, 505–530 (1990).
- Shapiro, S.S. and Wilk, M.B. (1965), "An Analysis of Variance Test for Normality (complete samples)," *Biometrika* 52, 591–611.
- Slifker, J. F. and Shapiro, S. S. (1980). *Technometrics*, 22, 239-246.
- Sneath, P.H.A. (1957) "The Application of Computers to Taxonomy," *Journal of General Microbiology*, 17, 201–226.
- Snedecor, G.W. and Cochran, W.G. (1967), *Statistical Methods*, Ames, Iowa: Iowa State University Press.
- Snee, R.D. (1979), "Experimental Designs for Mixture Systems with Multicomponent Constraints," *Commun. Statistics, A8*(4), 303–326.
- Snee, R.D. and Marquardt, D.W. (1974), "Extreme Vertices Designs for Linear Mixture Models," *Technometrics* 16, 391–408.
- Snee, R.D. and Marquardt D.W. (1975), "Extreme vertices designs for linear mixture models," *Technometrics* 16 399–408.
- Sokal, R.R. and Michener, C.D. (1958), "A Statistical Method for Evaluating Systematic Relationships," *University of Kansas Science Bulletin*, 38, 1409–1438.
- Spendley, W., Hext, G.R., and Minsworth, F.R. (1962), "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation," *Technometrics* 4, 441–461.

- Stevens, J.P. (1979), "Comment on Olson: Choosing a Test Statistic in Multivariate Analysis of Variance," *Psychological Bulletin*, 86, 355–360.
- Stevens, J.P. (1986), *Applied Multivariate Statistics for the Social Sciences*, New Jersey: Laurence Erlbaum Associates.
- Stone, C. and Koo, C.Y. (1986). "Additive Splines in Statistics," *In Proceedings of the Statistical Computing Section*, 45-48, Amer. Statist. Assoc., Washington, DC.
- Sullivan, J.H. and Woodall, W.H. (2000), "Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations," *IIE Transactions*, 32, 537-549.
- Tan, Charles Y., and Iglewicz, Boris (1999), "Measurement-methods Comparisons and Linear Statistical Relationship," *Technometrics*, 41:3, 192–201.
- Taguchi, G. (1976), "An Introduction to Quality Control," Nagoya, Japan: Central Japan Quality Control Association.
- Tobin, J. (1958), "Estimation of Relationships for Limited Dependent Variables," *Econometrica*, 26 24–36.
- Tamhane, A. C. and Dunlop, D. D. (2000) Statistics and Data Analysis. Prentice Hall.
- Tobias, P.A. and Trindade, D.C. (1995), *Applied Reliability, 2nd Edition*. New York: Van Nostrand Reinhold Company.
- Train, K.E. (2003), *Discrete Choice Methods with Simulation*, Cambridge: Cambridge University Press.
- Tukey, J. (1953), "A problem of multiple comparisons," Dittoed manuscript of 396 pages, Princeton University.
- Tukey, J. (1991), "The Philosophy of Multiple Comparisons," *Statistical Science*, 6, 100–116.
- Umetrics (1995), *Multivariate Analysis (3-day course)*, Winchester, MA.
- Wadsworth, H. M., Stephens, K., Godfrey, A.B. (1986) *Modern Methods for Quality Control and Improvement*. John Wiley & Sons.
- Walker, S.H. and Duncan, D.B. (1967), "Estimation of the Probability of an Event as a Function of Several Independent Variables," *Biometrika* 54.
- Wegman, E. J. (1990) "Hyperdimensional Data Analysis using Parallel Coordinates." *Journal of the American Statistical Association* 85, pp. 664–675.
- Welch, B.L. (1951), "On the comparison of several mean values: an alternative approach," *Biometrika* 38, 330–336.
- Western Electric Company (1956), *Statistical Quality Control Handbook*, currently referred to as the *AT&T Statistical Quality Control Handbook*.
- Westgard, J.O. (2002), *Basic QC Practices, 2nd Edition*. Madison, Wisconsin: Westgard QC Inc.
- Winer, B.J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill, Inc.
- Wold, S. (1994), "PLS for Multivariate Linear Modeling", *QSAR: Chemometric Methods in Molecular Design. Methods and Principles in Medicinal Chemistry*.
- Wolfinger, R., Tobias, R., and Sall, J. (1994). Computing Gaussian likelihoods and their derivatives for general linear mixed models. *SIAM J. Sci. Comput.* 15, 6 (Nov. 1994), 1294-1310.

- Wright, S.P. and R.G. O'Brien (1988), "Power Analysis in an Enhanced GLM Procedure: What it Might Look Like," *SUGI 1988, Proceedings of the Thirteenth Annual Conference*, 1097–1102, Cary NC: SAS Institute Inc.
- Ye, K. and Hamada, M. (2000) "Critical Values of the Lenth Method for Unreplicated Factorial Designs," *Journal of Quality Technology*, 32, 57-66.
- Ye, K. Q., and Hamada, M.(2001) "A Step-Down Lenth Method for Analyzing Unreplicated Factorial Designs." *Journal of Quality Technology*, 33, 140-152.

# Index

## JMP Modeling and Multivariate Methods

---

### Symbols

&LogVariance 157  
&Random 108  
^, redundant leaf labels 305

### Numerics

2D Gaussian Process Example.jmp 285  
-2LogLikelihood 341  
95% bivariate normal density ellipse 424

### A

Accept Current Estimates 243  
Actual by Predicted 215  
Actual by Predicted Plot 314  
actual vs. regressor plot *see* Leverage Plot  
Add 7–8, 95  
Add Column 64  
Add Multivariate Noise 573  
Add Random Noise 573  
Add Random Weighted Noise 573  
added variable plot 37  
ADF tests 348  
Adjusted Power and Confidence Interval 73, 75–76, 639–641  
agglomerative clustering 443  
Agreement Statistic 373  
AIC 340  
AIC 123  
Akaike's Information Criterion 340  
algorithms 607–642  
Aligned Responses 364  
Alpha 73  
Alt-click 539  
Alter Linear Constraints 544, 569  
alternative methods 629  
analysis of covariance example 52  
analysis of variance example 50  
Analysis of Variance table 25, 29–31, 97, 170  
Angular Frequency 336

Animals.jmp 108  
Annual 331  
Append Settings to Table 543  
Approx. F 143  
approximate F test 637  
ApproxStdErr 248  
AR Coefficients 329, 334  
ArcBall 529  
ARIMA 329, 343–345, 355  
Arrhenius 12  
ArrheniusInv 12  
assess validity 629  
assumptions 628–630  
AttributeGage.jmp 376  
Augmented Dickey-Fuller test 348  
Autocorrelation 333, 337  
autocorrelation 332–333  
Autocorrelation Lags 331  
Automatic Histogram Update 574  
autoregressive coefficients *see* AR coefficients  
Autoregressive Order 344  
Average Linkage 449

### B

Backward 121, 124  
Baltic.jmp 487  
bar chart of correlations 426  
Bartlett's Kolmogorov-Smirnov 336  
Baseball.jmp 106  
Bayes Plot 83–85  
between-subject 108, 147  
bibliographic references 643–654  
BIC 340  
Big Class.jmp 32  
Biplot 453  
Biplot 3D 454  
Biplot Options 453  
Biplot Ray Position 453, 479  
biplot rays 466  
Biquartimax 469  
Biquartimin 469

Birth Death Subset.jmp 445  
 bivariate normal density ellipse 424  
 Borehole Latin Hypercube.jmp 290  
**Bounded** 356  
**Box Cox Y Transformation** 89, 98–99  
 Box-Jenkins model *see ARIMA*  
 Box-Meyer Bayesian analysis 83  
 Brown smoothing 357  
 By role 242, 264

## C

**C. Total** 29–30  
 calculation details 607–642  
 calibration 65–69  
 Candidates report, Partition platform 297  
**Canonical 3D Plot** 480  
 canonical axis 151  
**Canonical Corr** 150  
 canonical correlation 135, 151–152, 637  
 Canonical Curvature report 97  
**Canonical Options** 479  
 canonical variables 149  
 Car Poll.jmp 374  
**Center Polynomials** 15, 62  
 centroid 429, 438, 638  
**Centroid Method** 449  
**Centroid Plot** 149–150  
**Centroid Val** 151  
 Cheese.jmp 189  
 Chemical Kinetics.jmp 250  
 Chi Square 172  
**Chi-Square** 171  
 Choice Modeling 379–417  
 Choice Platform  
     Launch Platform 384  
     Response Data 385  
     Segmentation 402  
     Single Table Analysis 399  
     subject effects 389  
 Cholesterol.jmp 147  
 citations 643–654  
 classification variable 152  
**Close All Below** (Partition Platform) 303  
 Cluster platform 441–458  
     compare methods 443  
     example 445  
     hierarchical 445–450  
     introduction 443–444  
     k-means 450–458

launch 444–445  
 normal mixtures 454–458  
**Cluster the Correlations** 427  
 Clustering History table 445–446  
 Coding column property 62  
 collinearity 40  
**Color Clusters and Mark Clusters** 447  
**Color Map** 448  
**Color Map On Correlations** 427  
**Color Map On p-values** 427  
**Color Points** 305, 479  
**Column Contributions** 304  
**Column Info** 48  
 column property  
     Coding 62  
**Combine** 127–128  
**Common Factor Analysis** 467  
**Complete Linkage** 450  
**Compound** 141, 147  
 compound multivariate example 147–149  
 computational details 607–642  
**Conditional Confidence CI** 112  
**Conditional Mean CI** 112  
**Conditional Pred Formula** 112  
**Conditional Pred Values** 112  
**Conditional Predictions** 112, 544  
**Conditional Residuals** 112  
 Confidence Intervals 234  
**Confidence Intervals** 175, 344, 355, 392, 541  
**Confidence Limits** 234, 242, 248, 260, 263  
**Confidence Limits Button** 234  
**Connecting Lines** 333  
**Connecting Lines Report** 46  
 constant estimate 341  
**Constrain fit** 344  
 Constraints 553  
**Construct Model Effects** 5  
 contaminating 83  
**Continue** 15  
 continuous response model 609  
**Contour Grid** 556, 560  
**Contour Label** 556  
**Contour Profiler** 89–90, 161–162, 214, 243, 555–593  
**Contrast** 221  
**Contrast** 141, 145, 147, 150, 212  
 contrast 43  
 contrast M matrix 141, 146, 636  
 Control Panel 120–122, 231, 233  
 Control-click 7, 115  
 converge *see* iteration

- Cook's D Influence 49  
Copy Settings Script 543  
corrected Akaike's Information Criterion 123  
correlation 419–439  
correlation matrix 422  
Correlation of Estimates 78–79, 214, 392  
Correlation Type 285  
Correlations Multivariate 422  
Cosine 336  
count data 195  
covariance 419–439  
Covariance Matrix 426  
Covariance of Estimates 214  
Covarimin 469  
 $C_p$  123  
 $C_p$  123  
CrabSattellites.jmp 203  
Create SAS Job 343  
Cronbach's alpha 432–433, 439  
Cross 7–8, 95  
Cross Correlation 348  
Cross Validation 488  
crossed term 112  
cross-product term 125, 536  
Crosstab 372  
Crosstab Report 46  
Crosstab Transposed 372  
cross-validation 485  
cross-validation report 489  
Cube Plots 89, 94–95  
Cubic 285  
cumulative logistic probability plot 179  
Current Estimates table 120, 122  
current predicted value 89, 535  
Current Value 241, 540  
current value 89, 535  
Custom 141, 356  
Custom Estimate 245  
Custom Inverse Prediction 246  
custom loss function 229, 233, 251–252  
Custom Profiler 569  
Custom Test 10, 63–64, 71, 149, 212  
CV for variance components 115  
Cytometry.jmp 451
- D**
- Daganzo Trip.jmp 411  
Daily 331  
damped-trend linear exponential smoothing 357  
Danger.jmp 432  
Data Filter 544  
data table of frequencies 195  
decision trees 295  
Default N Levels 544  
Defect Parametric Profile 574, 582  
Defect Profiler 579  
Defect Profiler 574, 581  
degenerate interval 67  
Degree 6  
Degree box 8  
degrees of freedom 30  
Delete Last Column 142  
Delta 73  
DenDF 143  
dendrogram 443, 445–446  
Dendrogram Scale command 448  
Density Ellipse 424–425  
derivative 261–262  
design code 607  
design matrix 624  
desirability confidence curve 89, 535  
Desirability Functions 541, 545  
desirability trace 545  
Detailed Comparisons Report 46  
Detergent.jmp 184  
deviance 202  
Deviance Residuals by Predicted 215  
DF 30, 32, 34, 171, 339  
DFE 123, 248  
diagnostic for SLS fit 31  
Diagnostics Plots 497  
Difference 171, 337  
Differencing Order 344  
dimensionality 463  
discriminant analysis 135, 152–153, 638  
Dispersion Effects 155–164  
distance column 152  
Distance Graph 448  
Distance Scale 448  
Distribution platform 629  
DOE menu overview 77  
Dogs.jmp 146  
Dose Response.jmp 177  
double exponential smoothing 357  
double-click 75  
drag 89, 94, 331, 425, 446, 535, 537, 546, 555  
Drug.jmp 23, 50, 52, 54  
dummy coding 33, 54, 61, 613  
Durbin-Watson Test 48

**E**

E matrix 139, 142, 152, 636  
**Edit Formula** 48  
 effect 612–628  
**Effect Attributes** 6  
**Effect Attributes and Transformations** 9–12  
**Effect Leverage** 14, 23, 27  
**Effect Leverage Pairs** 49  
 Effect Leverage Plot 24  
**Effect Macros** 6–9, 95  
**Effect Marginals** 392, 398  
**Effect Screening** 14, 27, 47, 57, 62, 79–81, 83–86  
 Effect Screening table 79  
 effect sparsity 79, 219  
 Effect Test table 3, 34  
 effective hypothesis tests 616  
**EigenValue** 150  
 eigenvalue decomposition 97, 463–464, 637  
**Eigenvectors** 465  
**Eigvec** 150  
**Ellipse alpha** 425  
**Ellipse Color** 425  
**Ellipses Transparency** 425  
 EM algorithm 450  
**Emphasis** 14, 27  
**EMS (Traditional)** 106  
 EMS method 64  
**Enter All** 122, 124  
**Entered** 122  
 epsilon adjustment 145  
**Equamax** 469  
**Equivalence Test** 46  
**Error** 30–31  
 error matrix 142  
 error terms 109  
**Estimate** 33, 81, 122, 248, 341, 344, 356  
**Estimate Nugget Parameter** 285  
**Estimates** 47, 57, 61, 63, 65, 71, 78–86  
**Even Spacing** 448  
 evolution 355  
**Exact F** 143  
 Excel  
   profiling models stored in 599  
**Excluded Effect** 10  
 excluded rows 29  
 Exercise.jmp 152  
**Expand Intermediate Formulas** 535, 553  
**Expand Into Categories, selecting column** 239  
**Expanded Estimates** 60–61  
 Expectation Maximization algorithm 450

expectation step 443  
 exponential regression example 231–235  
 exponential smoothing *see Smoothing Models*  
 Expression 572

**F**

**F Ratio** 30–32, 34  
**F Ratio**  
   in quotes 123  
 F test, joint 64  
**Factor** 341  
 factor analysis 432, 463–470  
 factor model 612–628  
**Factor Profiling** 41, 47, 90, 93–94, 98–99  
**Factor Rotation** 466  
**Factor Settings** 543, 569  
**Factorial Sorted** 8  
**Factorial to Degree** 8  
**Factorparsimax** 469  
 failure3Delimited.jmp 367  
 Failure3Freq.jmp 366  
 failure3Freq.jmp 375  
 failure3ID.jmp 368  
 failure3Indicators.jmp 367  
 failure3MultipleField.jmp 368  
 fiducial confidence limits 67  
 Fieller's method 67  
 Filtered Monte Carlo 544  
**First** 261  
**Firth Bias-adjusted Estimates** 409  
 Fish Patty.jmp 561  
 Fisher's Kappa 335  
 Fit Group 15, 604  
 Fit Model  
   Transformations 12  
 Fit Model dialog 1–16  
 Fit Model platform 1, 109, 119, 129, 137, 151–153, 194, 614  
   analysis of covariance 5–6, 52  
   analysis of variance 5–6, 23–26, 50  
   dialog overview 1–16  
   effects attributes 9–12  
   effects buttons 7–8  
   effects macros 8  
   Emphasis 14, 27  
   example 3–4, 65–69, 77–78, 90, 96–98, 106–107, 548–549, 555–557  
   examples 50  
   expanded estimates 60

fitting personality 12–13  
launch 3–16  
logistic regression 165  
Manova 5  
multiple regression 5–6  
multiple response 135–153  
nested model 5–6  
other options 15–16  
polynomial regression 5–6  
power analysis 70  
prediction equation 99  
random effects 103  
repeated measures 5–6  
response buttons 7  
response surface 5–6  
row diagnostics 46–48  
**Save** 48–49  
simple regression 5–6  
SLS estimates 57–86  
SLS introduction 21  
SLS prediction equation 87  
split-plot 5–6  
stepwise regression 117  
    categorical terms 127–128  
validity check 15  
**Fit Separately** 15  
**Fit Special** 231  
Fit Y by X platform 153, 165, 190  
Fitness.jmp 3, 119, 124  
fitting machines 630  
fitting personality 1, 3, 12–13  
fitting principles 609–612  
Five Factor Mixture.jmp 564  
Fixed 571  
**Fixed** 356  
Flash object in Profiler 540  
Football.jmp 127  
forecast 329, 337–359  
**Forecast Periods** 331, 342  
Forecast plot 342  
formula  
    editor 231–232  
    loss function 232  
    model 232  
    nonlinear fit 229  
formulas used in JMP calculations 607–642  
**Forward** 121, 124  
Freq role 7, 77, 168, 179, 233  
**Frequency** 336  
frequency data table 195

**Full** 171  
**Full Factorial** 6, 8, 185  
full factorial Anova *see* Fit Model platform  
**Full Quadratic Model** 18  
**G**  
**Gaussian** 285  
Gaussian Process 283  
**Gaussian Process** 285  
Gauss-Newton method 229, 252  
general fitting platform 1  
**Generalized Linear Model** 13  
**Geometric Spacing** 448  
G-G 145  
**Go** 83, 122, 124, 233, 241–242, 244, 247, 250, 260, 263, 306  
goal SSE 248  
Golf Balls.jmp 137  
Goodness of Fit test 172  
**Graph** 333, 337  
**Graph Updating** 556  
Greenhouse-Geisser 145  
**Grid Density** 556  
group similar rows *see* Cluster platform

## H

H matrix 140, 142, 152, 636  
Half Normal plot 221, 223  
Hardware Acceleration 529  
**Hats** 49  
**Helmert** 141, 145  
Hessian 252  
H-F 145  
hidden column 160  
**Hierarchical** 444  
hierarchical effects 129  
**Hoeffding's D** 428, 437  
Holt smoothing 357  
homogeneity of slopes 54  
Hotelling-Lawley Trace 143  
**Hourly** 331  
Huynh-Feldt 145  
hypothesis 632  
hypothesis SSCP matrix 142  
hypothesis test 616–622, 624

## I

**Identity** 141, 151

**Indicator Group** 365  
**Indiv Confidence Interval** 161  
**Individ Confidence Interval** 49  
**Ingots.jmp** 167, 179, 195  
**Ingots2.jmp** 195, 257  
**InjectionMolding.jmp** 157  
 interaction effect 31, 125, 536, 613  
**Interaction Plots** 41, 93–94  
**Interaction plots** 89  
 Interaction Profiler 544  
**Intercept** 344  
 intercept 341  
 Inverse Corr table 423  
 inverse correlation 423, 438  
**Inverse Prediction** 65–66, 68–69, 179  
 inverse prediction 641–642  
**InvertExpr** 247  
**Invertible** 341  
**InvPred** 247  
**Iris.jmp** 150, 153, 455, 473  
**IRLS** 254–257  
**IRLS Example.jmp** 254  
 Isosurface 520  
 item reliability 432–433  
**Iter** 343  
 Iteration Control Panel 231, 233  
**Iteration History** 343  
 Iteration History report 170, 343, 610  
**Iteration Log** 243  
 iteratively reweighted least squares 254–257

**J**

jackknife 430  
 Jackknife Distances 429  
 joint F test 64  
**Joint Factor Tests** 392, 398  
**JSL** 15, 263

**K**

**K Fold Crossvalidation** 304  
**Kendall's Tau** 428  
 Kendall's tau-b 436  
 key concepts 630  
**KMeans** 450  
 K-Means Clustering Platform  
   SOMs 458  
     Technical Details 460  
**Knotted Spline Effect** 10  
 Knotted Splines

test for curvature 10  
 Kruskal-Wallis 629

**L**

L matrix 636  
 lack of fit error 31, 53  
 Lack of Fit table 31–33, 172, 189  
**Lag** 341  
**Laptop Profile.jmp** 393  
**Laptop Runs.jmp** 394  
**Laptop Subjects.jmp** 396  
 Latin Square design 113  
 layered design 103, 113  
**Leaf Report** 304  
 least significant number 70, 76, 639–641  
 least significant value 70, 75–76, 639–641  
**Least Sq Mean** 41  
 least squares fit  
   estimates 57–86  
   introduction 21, 609  
   prediction equation 87, 99  
 least squares means 25, 616, 625  
 Least Squares Means table 26, 41, 138  
**Legend** 448  
 Lenth's method 79  
 Lenth's PSE 221, 225  
 Lenth's t-ratio 222  
**Level** 41  
 level smoothing weight 355  
**Leverage Plot** 23, 36–40, 54, 633–636  
   confidence curves 39  
   interpret 39–40  
**Lift Curve** 317  
**Lift Curves** 317  
 likelihood confidence intervals 248  
 Likelihood Ratio test 174, 190  
 Likelihood Ratio Tests 387  
**Likelihood Ratio Tests** 174, 392  
 limitations of techniques 628–630  
 linear combination 463  
 Linear Constraints 561  
**Linear Constraints** 553  
 linear dependence 617–620  
 linear exponential smoothing 357  
**Linear Model** 17  
 linear rank tests 629  
**Link Profilers** 543  
**Load Version 3 Model** 16  
**Loading Plot** 465

- L**
- Loading Plots** 496
  - Lock** 122, 303
  - Lock Columns**
    - Partition platform 304
  - Lock Factor Setting** 540, 543
  - Lock Z Scale** 523
  - Log** 243
  - Log Iterations** 569
  - Log Variance Effect** 10
  - Logistic platform 165
    - example 184–195, 253–254
    - response function 165
    - also see* Fit Model platform, Fit Y by X platform
  - Logistic Stepwise Regression 129
  - Logistic w Loss.jmp 253
  - LogLikelihood** 171
  - Log-Linear Variance 155
  - LogLinear Variance** 13
  - Loglinear Variance 157
  - LogLinear Variance Model 155, 157–164
  - log-variance effect 157
  - longitudinal data 145
  - Loss** 253
    - loss function 229, 232
      - custom 229, 251–252
  - Lost DFs** 34
  - Lower CL and Upper CL** 248
  - LS Means Plot 26
  - LSL Chop 583
  - LSMeans** 42
  - LSMeans Contrast** 26, 43–45, 51
  - LSMeans Plot** 25, 41
  - LSMeans Student's t** 45
  - LSMeans Tukey's HSD** 45
  - LSN 70, 76, 639–641
  - LSV 70, 75–76, 639–641
  - Lung Cancer Responses.jmp 407
  - Lung Cancer.jmp 409
- M**
- M matrix 141–142, 146, 636
  - machines of fit 630
  - Macros** 8, 96, 185
  - MAE** 340
  - Mahalanobis distance 429, 438
    - plot 429
  - Make Formula** 236
  - Make Model** 122, 129, 221
  - Make Table** 573, 576
- M**
- Mallow's Cp criterion 123
  - Manova** 13, 135
  - MANOVA 5, 13, 103, 143
  - Manova test tables 143
  - MAPE** 340
  - Mauchly criterion 144
  - Max** 244
  - Max RSq** 32
  - Maximization Options** 541
  - maximization step 444
  - Maximize** 546
  - Maximize Desirability** 541
  - Maximize for Each Grid Point** 542
  - Maximize for each Grid Point** 391
  - maximum likelihood 167, 610
  - Maximum Value** 540
  - Mean** 41, 141, 145
  - mean 609
  - Mean Confidence Interval** 49, 161
  - Mean Line** 333
  - mean model 157, 164
  - Mean of Response** 29
  - mean shift 583
  - Mean Square** 30, 32
  - measurement study 103
  - Mesh 528
  - Method** 106
  - Min** 244
  - Minimal Report** 14, 27
  - Minimize** 546
  - Minimum Setting** 540
  - Minimum Size Split** 304
  - Minimum Theta Value** 285
  - Minute** 331
  - missing cells 617–620, 625
    - nominal vs. ordinal factor 625–628
  - missing value 422
  - missing values 426
  - missing values (Fit Model) 15
  - Mixed** 121
    - mixed model 101, 103
  - Mixture Effect** 10
  - Mixture Profiler 557
  - Mixture Response Surface** 9
  - Model** 15, 30–31, 171
  - Model Comparison table 338
  - Model Dialog** 393
  - Model effects 3
  - model effects buttons 7–8
  - model formula 232

**Model Library** 235  
 model significance 36  
**Model Summary table** 339, 344  
**Monte Carlo simulation** 219  
**Monthly** 331  
**Moving Average Order** 344  
**MSE** 123, 248  
**Multiple Delimited** 365  
 multiple inference 629  
 multiple regression example 119  
**Multiple Response** 365  
**Multiple Response by ID** 365  
 multiple response fitting 135–153, 537, 548  
**Multivariate** 573  
**Multivariate** 419, 421  
 multivariate analysis of variance *see* Fit Model platform  
 multivariate estimates 142  
 multivariate least-squares means 149  
 multivariate mean 135, 429, 438  
 multivariate outliers 429  
 Multivariate platform 419–439, 636–639  
     example 432–433  
     principal components 432–470  
 multivariate regression 135–153

**N**

**N Runs** 573  
**N Strata** 574, 605  
**nDF** 123  
**Negative Exponential.jmp** 261  
 negative log-likelihood 170, 631  
 negative variance components 105  
**Nest** 7–8  
 nested effect 109, 112, 615, 620  
 neutral values 40  
**New Column** 44  
 New Parameter dialog 231  
**Newton** 243  
 Newton-Raphson method 229, 252  
**No Center** 488  
**No Noise** 573  
**No Rules** 127  
**No Scale** 488  
**Noah Decay.jmp** 77  
 Noise Factors 593–599  
**Noise Factors** 535  
**Nominal** 172  
 nominal factor 613, 616–620

**Nominal Logistic** 13  
 nominal response model 609–611  
 nonestimable 612  
**Nonlinear** 234  
**Nonlinear Model Library** 235  
     Customizing 239  
**Nonlinear platform** 229, 250, 253, 256, 260–266  
     derivatives 261–262  
     example 250  
     launch 232  
**Nonparametric Correlations** 428  
**Nonparametric Measures of Association table** 428  
 nonstationary time series 337  
**Norm KHC** 111  
**Normal censored** 572  
 normal density ellipse 424  
 normal distribution 104, 631  
**Normal Plot** 82  
**Normal Truncated** 572  
**Normal Weighted** 605  
**Normal weighted** 572, 576  
**Nparm** 34  
 Nugget parameters 288  
 null hypothesis 71  
**Number** 73  
**Number of Clusters** 448  
**Number of Forecast Periods** 337  
**Number of Plots Across** 508  
**Number of Plotted Points** 540  
**Number of Points** 244  
**NumDeriv** 261  
**Numeric Derivatives Only** 243

**O**

**Obbiquartimax** 469  
**Obequamax** 469  
**Obfactorparsimax** 469  
**Obj-Criterion** 343  
**Oblimin** 469  
**Obparsimax** 469  
**Obquartimax** 470  
**Observations** 29, 172  
**Obvarimax** 470  
**Odds Ratio** 177  
 Odds Ratio Example 177  
**Odor Control Original.jmp** 96  
**Offset** 206  
 offset variable 206  
**Optimal Value** 303

Option-click 539  
OPTMODEL formulas 541  
**Ordered Differences Report** 46  
ordinal crossed model 624  
ordinal factor 625  
ordinal interaction 624  
ordinal least squares means 625  
**Ordinal Logistic** 13  
ordinal logistic regression 188  
ordinal response model 609, 611–612  
**Orthog Coded** 81  
**Orthog t-Ratio** 82–83  
**Orthogonalize** 142  
**Orthomax** 469  
orthonormal response design matrix 144  
**Other** 425  
**Outlier Analysis** 429–432  
Outlier Distance plot 438  
**Output Grid Table** 543–544  
**Output Random Table** 544  
**Output Split Table** 303  
outside interval 67

## P

p, d, q parameters 344  
**Pairwise Correlations** 422  
Pairwise Correlations table 426  
**Parallel Coord Plots** 454  
**Parallel Coordinate Plot** 434  
**Parameter** 64, 122, 244, 248  
**Parameter Bounds** 242  
**Parameter Contour Profiler** 244  
Parameter Estimate Population table 80–82  
Parameter Estimates Population table 82  
Parameter Estimates table 3, 33, 96, 138, 206, 341  
parameter interpretation 613  
**Parameter Power** 70–71  
**Parameter Profiler** 244  
**Parameter Surface Profiler** 244  
**Parameters** 231  
parameters in loss function 252  
**Parametric Survival** 13  
**Pareto Plot** 85  
**Parsimax** 469  
**Partial Autocorrelation** 333, 337  
partial autocorrelation 332  
**Partial Corr table** 139, 424  
partial correlation 424  
partial-regression residual leverage plot 37

**Partition** 293  
Partition Platform 293–301  
**Paste Settings Script** 543  
PCA 432, 463–470  
Pearson correlation 426, 436  
**Pearson Residuals By Predicted** 215  
**Per Mouse Move** 556  
**Per Mouse Up** 556  
Percent Variance Explained 489  
**Period** 336  
periodicity 345  
**Periodogram** 336  
periodogram 335  
**Periods Per Season** 345, 355  
personality 1, 3, 119  
Pillai's Trace 143  
Pizza Choice Example 382–406  
Pizza Combined.jmp 400  
Pizza Profiles.jmp 385, 402  
Pizza Responses.jmp 385, 402  
Pizza Subjects.jmp 389, 402  
**Plot** 242  
**Plot Actual by Predicted** 47, 161  
**Plot Actual by Predicted (Partition)** 304  
**Plot Effect Leverage** 47  
**Plot Residual By Predicted** 47  
**Plot Residual By Row** 47  
**Plot Studentized Residual by Predicted** 161  
**Plot Studentized Residual by Row** 161  
**PLS** 485–499  
    components 485  
    latent vectors 485  
    Missing Values 499  
    Platform Options 494  
    Statistical Details 499  
Poisson loss function 259–260  
**Polynomial** 141, 145  
**Polynomial to Degree** 6, 9  
Popcorn.jmp 42  
population of levels 101  
posterior probability 84  
power 70, 76  
**Power Analysis** 70–71, 77–86, 639–641  
**Power Plot** 75  
**Predicted Values** 48  
Prediction Column role 231  
prediction equation 54, 87, 99  
**Prediction Formula** 48, 97, 161  
prediction formula 181, 231, 612–624  
**Prediction Intervals** 162

**Prediction Profiler** 89, 535  
**Predictor role** 253–254, 256, 260  
**Press** 48  
 Press Residuals 492  
 pressure cylinders fitting machine 632  
**Principal Components** 467  
 principal components analysis 432, 463–470  
 prior probability 83  
**Prob to Enter** 121  
**Prob to Leave** 121  
**Prob>|t|** 33, 82, 341  
**Prob>ChiSq** 171  
**Prob>F** 31–32, 35, 143  
**Prob>F**  
 in quotes 123  
 probit example 257–259  
 process disturbance 329  
 product-moment correlation 426, 436  
**Profile** 141, 145  
 profile confidence limits 248–249  
 Profile data 382  
 profile likelihood confidence intervals 248  
 profile trace 535  
**Profiler**  
 and saved standard error formulas 50  
**Profiler** 89, 161–162, 184, 214, 243, 392, 535  
**Profilers** 531–599  
**Prop of Error Bars** 541, 549  
**Proportional Hazards** 13  
**Proportional Hazards**  
*also see Fit Model platform*  
 prospective power analysis 77–78  
**Prune Below** 303  
**Prune Worst** 303–304  
 pseudo standard error 79  
 pure error 31

**Q**

quadratic equation 642  
 quadratic ordinal logistic regression 193  
**Quartimax** 469  
**Quartimin** 470  
**QuasiNewton BFGS** 243  
**QuasiNewton SR1** 243  
 questionnaire analysis 432–433

**R**

**R** 123  
 &Random 108

Random 572  
**Random** 6  
**Random Effect** 9, 108–109, 119  
 random effects 64, 103  
 introduction 103  
**Range Odds Ratios** 176  
**Rater Agreement** 365  
**Reactor.jmp** 81–82, 93, 99, 125  
 reduce dimensions 463  
**Reduced** 171  
**Ref Labels** 560  
**Ref Lines** 560  
 references 643–654  
 relative significance 628  
 reliability analysis 432–433  
*also see Survival platform*  
**Remember Settings** 543, 568  
**Remember Solution** 245  
**REML (Recommended)** 106  
 REML method 64  
 results 110–111  
**Remove** 7, 331  
**Remove All** 122  
**Remove Contour Grid** 560  
**Repeated Measures** 141, 364  
 repeated measures design 103, 135  
 example 145, 147–149  
 replicated points 31  
 report layout emphasis 27  
**Reset** 233, 241–242, 263  
**Reset Factor Grid** 542  
 residual 30  
 residual matrix 142  
**Residual Statistics** 343  
 residual/restricted maximum likelihood 110–111  
**Residuals** 48, 161  
 residuals 103, 342, 609  
 Response data 382  
**Response Frequencies** 365  
**Response Limits** 574  
 response models 609–612  
 Response role 232  
 Response Specification dialog 140  
**Response Surface** 6, 8, 95, 192  
 response surface 95–98  
**Response Surface Effect** 10  
 Response Surface table 97  
**Restrict** 127–128  
 restricted vs. unrestricted parameterization 104  
 retrospective power analysis 70–71

rho *see* loss function  
right-click 111, 115  
RMSE 73, 84  
**RMSE** 248  
Robust Engineering 593–599  
**ROC** 182  
ROC Curve 315  
**ROC Curve** 315, 479  
ROC Curves 314  
**Root Mean Square Error** 29  
**Row Diagnostics** 47  
row diagnostics 46–48  
Roy's Maximum Root 143  
**RSquare** 29, 123, 340  
**RSquare (U)** 172  
**RSquare Adj** 29, 123, 340  
Rules 126  
**Run Model** 3, 66, 68, 120, 147, 151, 168, 192, 221  
**Run Script** 15

## S

Salt in Popcorn.jmp 194  
sample autocorrelation function 333  
sample of effects 101  
**Sample Size and Power** 77  
**Sampled** 572  
SAS GLM procedure 607, 609–622  
saturated 31  
**Save** 47, 181, 246, 338, 629  
**Save As Flash (SWF)** 540  
**Save Best Transformation** 99  
**Save Canonical Scores** 150–151, 480  
**Save Cluster Hierarchy** 448  
**Save Clusters** 448, 454  
**Save Coding Table** 49  
**Save Columns** 97, 215, 343  
**Save Density Formula** 454  
**Save Desirabilities** 542  
**Save Desirability Formula** 542  
**Save Discrim** 152, 638  
**Save Display Order** 448  
**Save Estimates** 242, 256, 263  
**Save Expanded Formulas** 553  
**Save Expected Values** 181–182  
**Save Gradients by Subject** 392, 402  
**Save Indiv Confid Limits** 246  
**Save Inverse Prediction Formula** 247  
**Save Leaf Label Formula** 305  
**Save Leaf Labels** 305

**Save Leaf Number Formula** 305  
**Save Leaf Numbers** 305  
**Save Linear Constraints** 544, 569  
**Save Loadings** 498  
**Save Mixture Formulas** 454  
**Save Mixture Probabilities** 454  
**Save Percent Variation Explained for X Variables** 498  
**Save Percent Variation Explained for Y Variables** 498  
**Save Pred Confid Limits** 246  
**Save Predicted** 305, 309, 312  
**Save Predicted Values** 48  
**Save Prediction Formula** 246, 305, 309, 312, 498  
**Save Probability Formula** 181  
**Save Quantiles** 181–182  
**Save Residual Formula** 246  
**Save Residuals** 305, 309, 312  
**Save Scores and Distance** 498  
**Save Specific Solving Formula** 247  
**Save Specific Transformation** 99  
**Save Spectral Density** 336  
**Save Std Error of Individual** 246  
**Save Std Error of Predicted** 49, 246  
**Save to Data Table** 15  
**Save to Script window** 15  
**Save Utility Formula** 392  
**Save X Residuals** 498  
**Save Y Residuals** 498  
**SBC** 340  
**Scaled Estimates** 62, 79  
**Scatterplot Matrix** 424  
scatterplot matrix 422  
**Schwartz's Bayesian Criterion** 340  
**Score Options** 479  
**Score Plot** 465  
**Scores Plot** 489  
**Scores Plots** 496  
**Scree Plot** 465  
screening analysis *see* Fit Model platform  
**Screening and Response Surface Methodology** 12  
screening design 79  
Seasonal ARIMA 329  
**Seasonal ARIMA** 345  
seasonal exponential smoothing 358  
seasonal smoothing weight 355  
**Second** 331  
**Second Deriv Method** 233  
**Second Deriv. Method** 252  
**Second Derivatives** 254, 258

**Select Rows** 303  
 select rows 446  
 select rows in data table 419  
**Sensitivity Indicator** 541  
**Separate Response** 374  
**Separate Responses** 364  
**Sequential Tests** 63  
*Seriesg.jmp* 331  
*seriesJ.jmp* 346  
**Set Desirabilities** 542, 546  
**Set Random Seed** 574  
**Set Script** 543  
**Set To Data in Row** 543  
**Shaded Ellipses** 425  
 Shift-click 7  
 Ship 206  
*Ship Damage.JMP* 206  
*Ship Damage.jmp* 259  
**Show** 540  
**Show Biplot Rays** 453, 479  
**Show Canonical Details** 479  
**Show Confidence Interval** 342–343  
**Show Confidence Lines** 494  
**Show Constraints** 560  
**Show Correlations** 425  
**Show Current Value** 560  
**Show Derivatives** 243, 261  
**Show Distances to each group** 479  
**Show Formulas** 541  
**Show Graph** 236, 304  
**Show Histogram** 425  
**Show Means CL Ellipses** 479  
**Show Normal 50% Contours** 479  
**Show Points** 237, 304, 333, 342–343, 425, 479, 494, 560  
**Show Prediction Expression** 59, 247  
**Show Probabilities to each group** 479  
**Show Split Bar** 304  
**Show Split Candidates** 304  
**Show Split Prob** 304  
**Show Split Stats** 304  
**Show Tree** 304  
 ShrinkageResidual 161  
**Sigma** 549  
**Sigma** 73  
 significance of model 36  
 significance probability 426  
     stepwise regression 117  
 simple exponential smoothing 356  
**Simulate** 580  
**Simulation Experiment** 574  
**Simulator** 544, 570  
**Simulator** 544  
**Sine** 336  
**Single Linkage** 450  
 singularity 617–620, 625  
 SLS fit *see* Fit Model platform  
**Small Tree View** 304  
**Smoothing Model dialog** 355  
 smoothing models 329, 354–359  
 smoothing weight 355  
*Solubility.jmp* 463  
 Solution table 97, 242, 248  
**Solve for Least Significant Number** 73  
**Solve for Least Significant Value** 73  
**Solve for Power** 73  
**SOM Technical Details** 460  
**SOMs** 458  
**Sort Split Candidates** 304  
**Source** 30–31, 34  
 sources 643–654  
 Spearman's Rho 436  
**Spearman's Rho** 428  
**Spec Limits** 574–575  
**Specified Value** 303  
**Specify Differencing dialog** 337  
**Spectral Density** 335–336  
 spectral density plots 329  
 sphericity 144–145  
 Sphericity Test table 144  
**Split Best** 302, 304  
**Split Here** 302  
**Split History** 304  
 split plot design 103  
     example 108–110  
**Split Specific** 303  
 spring fitting machine 631–632  
**SS** 123  
**SSE** 30, 99, 123, 241, 248  
**SSE Grid** 244, 251  
**SSR** 30  
**SST** 30  
**Stable** 341  
**Stable Invertible** 356  
**Stack** 195  
*Stacked Daganzo.jmp* 412  
**Standard Deviation** 340  
 standard least squares 12, 23  
     estimates 57–86  
     introduction 21

prediction equation 87, 99  
*also see* Fit Model platform

**Standardize Data** 444  
statistical details 607–642

**Std Beta** 33–34

**Std Dev Formula** 161

**Std Err Scale** 84

**Std Error** 33, 41, 341

**Std Error of Individual** 49, 161

**Std Error of Predicted** 49, 161

**Std Error of Residual** 49

**Std Narrow** 583

**StdErr Pred Formula** 49

**StdError Fitted** 246

**StdError Indiv** 246

**Step** 122, 124, 241, 343  
Step History table 120  
stepwise regression 13, 117, 119, 129  
    categorical terms 127–128  
    Control panel 120–122  
    example 119, 124–126  
    Logistic 129

Stochastic Optimization.jmp 585

**Stop** 122, 241

**Stop At Boundaries** 552

**Studentized Deviance Residuals by Predicted** 215

**Studentized Pearson Residuals by Predicted** 215

**Studentized Residuals** 49, 161

Subject data 382

**Submit to SAS** 15, 343

substitution effect 53

subunit effect 103

**Sum** 141–142, 145, 147

sum M matrix 146

**Sum of Squared Errors** 339

**Sum of Squares** 30, 32, 34  
sum of squares 30  
Sum of Squares Corrected Total 29

**Sum of Weights** 29

**Sum Wgts** 172

Summary of Fit table 3, 28–29

Surface Fill 528

Surface Plot 513  
    Constants 529  
    Control Panel 522  
    Dependent Variables 524  
    Variables 523

**Surface Plot** 556

Surface Profiler 569

**Surface Profiler** 89, 161–162, 214, 243, 515

survival regression *see* Fit Model platform  
symbolic derivative 261  
synthetic denominator, effect 112

## T

**t Ratio** 33, 81, 341

**T<sup>2</sup> Statistic** 429

**Table Format** 371

**Table of Estimates** 99

**Table Transposed** 372

Taguchi 155

**Target** 546

**Term** 33, 341

**Test** 143

**Test Details** 149–151

**Test Each Column Separately Also** 140

**Test Each Response** 373, 375

**Test Reponse Homogeneity** 374

**Test Response Homogeneity** 373

**Test Slices** 46

**Time Frequency** 331

Time ID role 331

**Time Series Graph** 333

Time Series platform 329, 331–359  
    **ARIMA** 343–345  
        commands 332–338  
        example 331–332  
        launch 331–332  
        modeling report 338–343  
    **Seasonal ARIMA** 345  
        smoothing models 354–359

Time Series Plot 332

Time Series role 331

Tiretread.jmp 517–518, 537, 581, 594

tiretread.jmp 548

**Traditional statistics** 12

training set 494

**Transfer Function** 351

Transfer Function Models 346

**Transfer Function Models** 329

transform covariate 53

Transformations 9–12

**Transformations** 12

**Transition Report** 373

transmitted variation 593

tree diagram 445

trend 355

**Turn At Boundaries** 552

tutorial examples

analysis of covariance 52  
 analysis of variance 23–26  
 compound multivariate model 147–149  
 contour profiler 90, 555–557  
 correlation 432–433  
 desirability profile 548–549  
 exponential regression 231–235  
**Fit Model** 3–4, 50  
 hierarchical clustering 445  
 inverse prediction 65–69  
**IRLS** 254–257  
 logistic regression 184–195, 253–254  
 multiple regression 119  
 nonlinear regression 250  
 one-way Anova 50  
 probit 257–259  
 random effects 106–107  
 repeated measures 145, 147–149  
 response surface 96–98  
 retrospective power 77–78  
 split plot design 108–110  
 stepwise regression 119, 124–126  
 time series 331–332  
**Two way clustering** 448  
 Type I sums of squares 63  
 Types III and IV hypotheses 616

**U**

uncertainty 630  
**Unconstrained** 356  
**Unit Odds Ratios** 176  
**Univariate** 422  
**Univariate Tests Also** 140, 144, 146  
 unrestricted vs. restricted parameterization 104  
**Unthreaded** 243  
**Up Dots** 556, 560  
 US Population.jmp 231, 245  
**Use Cross Validation** 488  
**User Specified** 488  
 USL Chop 583  
 usual assumptions 628–630

**V**

validation set 494  
 validity 629  
**Value** 143  
 Variability Chart platform 101  
 variance components 103–104  
**Variance Effect Likelihood Ratio Tests** 160

**Variance Estimate** 340  
**Variance Formula** 161  
 Variance Parameter Estimates 160  
**Varimax** 469  
**Variogram** 329, 334, 337  
**VIF** 34

**W-Z**

Wald test 174, 190  
**Ward's** 449  
**Weekly** 331  
 Weight role 7, 233, 254, 256  
 Weld-Repaired Castings.jmp 226  
**Whole Effects** 127  
 Whole Model Leverage Plot 24, 37  
 Whole Model table 28, 143, 170  
 Wilcoxon rank-sum 629  
 Wilks' Lambda 143  
 Winter's method 358  
 within-subject 108, 144, 147  
 X role 253–254, 256, 260, 331  
**X, Predictor Formula** 232  
 Y role 3, 7, 231–232, 331, 444  
**Y, Prediction Formula** 535  
 zero eigenvalue 97  
**Zero To One** 356  
 zeroed 612