Version 11

Specialized Models

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

Marcel Proust
Technology License Notices

- Scintilla - Copyright © 1998-2012 by Neil Hodgson <neilh@scintilla.org>. All Rights Reserved.
  Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.
  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.

- Telerik RadControls: Copyright © 2002-2012, Telerik. Usage of the included Telerik RadControls outside of JMP is not permitted.


- Made with Natural Earth. Free vector and raster map data @ naturalearthdata.com.

- Packages - Copyright © 2009-2010, Stéphane Sudre (s.sudre.free.fr). All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
  Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  Neither the name of the WhiteBox nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGligence OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

• iODBC software - Copyright © 1995-2006, OpenLink Software Inc and Ke Jin
  (www.iodbc.org). All rights reserved.
  Redistribution and use in source and binary forms, with or without modification, are
  permitted provided that the following conditions are met:
  – Redistributions of source code must retain the above copyright notice, this list of
    conditions and the following disclaimer.
  – Redistributions in binary form must reproduce the above copyright notice, this list
    of conditions and the following disclaimer in the documentation and/or other
    materials provided with the distribution.
  – Neither the name of OpenLink Software Inc. nor the names of its contributors may
    be used to endorse or promote products derived from this software without specific
    prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND
  ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
  WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
  DISCLAIMED. IN NO EVENT SHALL OPENLINK OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
  INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
  BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
  LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
  OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
  THE POSSIBILITY OF SUCH DAMAGE.

• bzip2, the associated library “libbzip2”, and all documentation, are Copyright ©
  1996-2010, Julian R Seward. All rights reserved.
  Redistribution and use in source and binary forms, with or without modification, are
  permitted provided that the following conditions are met:
  Redistributions of source code must retain the above copyright notice, this list of
  conditions and the following disclaimer.
  The origin of this software must not be misrepresented; you must not claim that you
  wrote the original software. If you use this software in a product, an acknowledgment
  in the product documentation would be appreciated but is not required.
Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- MATLAB software is Copyright © 1984-2012, The MathWorks, Inc. Protected by U.S. and international patents. See www.mathworks.com/patents. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.
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
• details on registering for JMP training
• 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

## Specialized Models

1. **Learn about JMP**  
   [Documentation and Additional Resources](#) ............................................................. 15  
   - Formatting Conventions ............................................................................................. 17  
   - JMP Documentation ................................................................................................. 17  
     - JMP Documentation Library ................................................................................. 18  
     - JMP Help ............................................................................................................. 22  
   - Additional Resources for Learning JMP .................................................................. 22  
     - Tutorials ................................................................................................................ 23  
     - Sample Data Tables ............................................................................................. 23  
     - Learn about Statistical and JSL Terms ................................................................. 23  
     - Learn JMP Tips and Tricks ................................................................................... 24  
     - Tooltips ............................................................................................................... 24  
     - JMP User Community .......................................................................................... 24  
     - JMPer Cable ......................................................................................................... 24  
     - JMP Books by Users ............................................................................................ 25  
     - The JMP Starter Window ...................................................................................... 25  

2. **Introduction to Specialized Modeling**  
   [Overview of Modeling Techniques](#) ........................................................................ 27  

3. **Partition Models**  
   [Use Decision Trees to Explore and Model Your Data](#) ........................................ 29  
   - Overview of Partition ............................................................................................. 31  
   - Example of Partition .............................................................................................. 31  
   - Launching the Partition Platform ........................................................................... 33  
   - Partition Method .................................................................................................... 34  
     - Decision Tree ..................................................................................................... 34  
     - Bootstrap Forest .................................................................................................. 46  
     - Boosted Tree ...................................................................................................... 50
4 Neural Networks

Fit Nonlinear Models Using Nodes and Layers ........................................ 67
Overview of Neural Networks ................................................................. 69
Launch the Neural Platform ................................................................. 69
The Neural Launch Window ................................................................. 70
The Model Launch ........................................................................ 71
Model Reports ........................................................................ 77
Training and Validation Measures of Fit ........................................ 78
Confusion Statistics ................................................................. 79
Model Options ................................................................. 79
Example of a Neural Network ............................................................. 81
6 Nonlinear Regression with Built-In Models

6.1 Analyze Models with the Fit Curve Platform

6.2 Introduction to the Nonlinear Fit Curve Personality

6.3 Example Using the Fit Curve Personality

6.4 Launch the Nonlinear Platform

6.5 The Fit Curve Report

6.6 Initial Fit Curve Reports

6.7 Fit Curve Options

6.8 Model Formulas

6.9 Test Parallelism

6.10 Compare Parameter Estimates

6.11 Equivalence Test

7 Nonlinear Regression with Custom Models

7.1 Analyze Models That You Create

7.2 Example of Fitting a Custom Model

7.3 Launch the Nonlinear Platform

7.4 The Nonlinear Fit Report

7.5 Nonlinear Platform Options

7.6 Create a Formula Using the Model Library

7.7 Additional Examples

8 Gaussian Process

8.1 Fit Data Using Smoothing Models

8.2 Launching the Platform
9 Time Series Analysis

Fit Time Series Models and Transfer Functions ................................................. 153

Launch the Platform ................................................................. 155
Select Columns into Roles .............................................................. 155
The Time Series Graph ........................................................................... 156

Time Series Commands ............................................................................ 156
Graph .......................................................................................................... 157
Autocorrelation and Partial Autocorrelation ......................................................... 157
Variogram ..................................................................................................... 158
AR Coefficients ............................................................................................ 159
Spectral Density ............................................................................................ 159
Save Spectral Density ..................................................................................... 160
Number of Forecast Periods ........................................................................... 161
Difference ....................................................................................................... 161
Show Lag Plot ................................................................................................. 162

Modeling Reports ......................................................................................... 162
Model Comparison Table .............................................................................. 162
Model Summary Table .................................................................................... 164
Parameter Estimates Table ............................................................................ 166
Forecast Plot ................................................................................................. 167
Residuals ......................................................................................................... 167
Iteration History ............................................................................................. 167
Model Report Options .................................................................................... 168
ARIMA Model ............................................................................................... 168
Seasonal ARIMA ............................................................................................ 170
ARIMA Model Group ..................................................................................... 170
Transfer Functions .......................................................................................... 171
### Specialized Models

- **Report and Menu Structure** .................................................. 171
- **Diagnostics** ................................................................. 173
- **Model Building** .............................................................. 174
- **Transfer Function Model** .................................................. 175
- **Model Reports** ............................................................... 177
- **Model Comparison Table** .................................................. 179
- **Fitting Notes** ................................................................. 179
- **Smoothing Models** .......................................................... 179
  - **Simple Moving Average** .................................................. 180
  - **Smoothing Model Dialog** ............................................... 181
  - **Simple Exponential Smoothing** ....................................... 182
  - **Double (Brown) Exponential Smoothing** .......................... 182
  - **Linear (Holt) Exponential Smoothing** ............................... 183
  - **Damped-Trend Linear Exponential Smoothing** .................... 183
  - **Seasonal Exponential Smoothing** .................................... 183
  - **Winters Method (Additive)** ......................................... 184

### 10 Response Screening

**Screen Large-Scale Data** .................................................. 185

- **Response Screening Platform Overview** ............................. 187
- **Example of Response Screening** ....................................... 188
- **Launch the Response Screening Platform** ........................... 191
- **The Response Screening Report** ....................................... 193
  - **FDR PValue Plot** ....................................................... 193
  - **FDR LogWorth by Effect Size** ..................................... 194
  - **FDR LogWorth by RSquare** .......................................... 195
- **The PValues Data Table** .................................................. 195
  - **PValues Data Table Columns** ....................................... 196
  - **Columns Added for Robust Option** .................................. 197
  - **PValues Data Table Scripts** ........................................ 198
- **Response Screening Platform Options** ............................... 198
- **Means Data Table** .......................................................... 199
- **Compare Means Data Table** ............................................ 200
- **The Response Screening Personality in Fit Model** ............... 202
- **Launch Response Screening in Fit Model** ........................... 202
The Fit Response Screening Report .................................................. 204
PValues Data Table ................................................................. 204
Y Fits Data Table ................................................................. 205
Additional Examples of Response Screening ................................. 206
Example of Tests of Practical Significance and Equivalence ............ 206
Example of the MaxLogWorth Option ........................................... 208
Example of Robust Fit ............................................................. 210
Response Screening Personality ................................................... 213
Statistical Details ..................................................................... 215
The False Discovery Rate ........................................................... 215

A References

B Statistical Details

Specialized Models .................................................................... 221
The Response Models ................................................................ 221
Continuous Responses .............................................................. 221
Nominal Responses ................................................................ 221
Ordinal Responses .................................................................. 223
The Factor Models .................................................................. 224
Continuous Factors ................................................................. 225
Nominal Factors .................................................................... 225
Ordinal Factors ..................................................................... 226
The Usual Assumptions .............................................................. 242
Assumed Model ..................................................................... 242
Relative Significance ............................................................... 242
Multiple Inferences ................................................................. 243
Validity Assessment ................................................................. 243
Alternative Methods ................................................................. 243
Key Statistical Concepts ............................................................ 244
Uncertainty, a Unifying Concept .............................................. 244
The Two Basic Fitting Machines .................................................. 245
Multivariate Details ................................................................. 248
Multivariate Tests .................................................................. 248
Approximate F-Test ................................................................. 249
<table>
<thead>
<tr>
<th>Specialized Models</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canonical Details</td>
<td>249</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>250</td>
</tr>
<tr>
<td>Power Calculations</td>
<td>251</td>
</tr>
<tr>
<td>Computations for the LSN</td>
<td>251</td>
</tr>
<tr>
<td>Computations for the LSV</td>
<td>252</td>
</tr>
<tr>
<td>Computations for the Power</td>
<td>253</td>
</tr>
<tr>
<td>Computations for the Adjusted Power</td>
<td>253</td>
</tr>
<tr>
<td>Inverse Prediction with Confidence Limits</td>
<td>254</td>
</tr>
<tr>
<td>Index</td>
<td>257</td>
</tr>
<tr>
<td>Specialized Models</td>
<td>257</td>
</tr>
</tbody>
</table>
Chapter 1
Learn about JMP
Documentation and Additional Resources

This chapter includes the following information:

- book conventions
- JMP documentation
- JMP Help
- additional resources, such as the following:
  - other JMP documentation
  - tutorials
  - indexes
  - Web resources

Figure 1.1 The JMP Help Home Window on Windows
Contents

Formatting Conventions ........................................................... 17
JMP Documentation ................................................................. 17
  JMP Documentation Library ................................................ 18
  JMP Help ................................................................. 22
Additional Resources for Learning JMP .................................... 22
  Tutorials ............................................................ 23
  Sample Data Tables .......................................................... 23
  Learn about Statistical and JSL Terms .................................... 23
  Learn JMP Tips and Tricks .................................................. 24
  Tooltips ............................................................... 24
  JMP User Community ......................................................... 24
  JMPer Cable .............................................................. 24
  JMP Books by Users .......................................................... 25
  The JMP Starter Window ...................................................... 25
Formatting Conventions

The following conventions help you relate written material to information that you see on your screen.

- Sample data table names, column names, pathnames, filenames, file extensions, and folders appear in Helvetica font.
- Code appears in Lucida Sans Typewriter font.
- Code output appears in *Lucida Sans Typewriter* italic font and is indented farther than the preceding code.
- **Helvetica bold** formatting indicates items that you select to complete a task:
  - buttons
  - check boxes
  - commands
  - list names that are selectable
  - menus
  - options
  - tab names
  - text boxes
- The following items appear in italics:
  - words or phrases that are important or have definitions specific to JMP
  - book titles
  - variables
- Features that are for JMP Pro only are noted with the JMP Pro icon. For an overview of JMP Pro features, visit [http://www.jmp.com/software/pro/](http://www.jmp.com/software/pro/).

**Note:** Special information and limitations appear within a Note.

**Tip:** Helpful information appears within a Tip.

JMP Documentation

JMP offers documentation in various formats, from print books and Portable Document Format (PDF) to electronic books (e-books).

- Open the PDF versions from the Help > Books menu or from the JMP online Help footers.
• All books are also combined into one PDF file, called *JMP Documentation Library*, for convenient searching. Open the *JMP Documentation Library* PDF file from the **Help > Books** menu.

• e-books are available at Amazon, Safari Books Online, and in the Apple iBookstore.

• You can also purchase printed documentation on the SAS website: http://support.sas.com/documentation/onlinedoc/jmp/index.html

### JMP Documentation Library

The following table describes the purpose and content of each book in the JMP library.

<table>
<thead>
<tr>
<th>Document Title</th>
<th>Document Purpose</th>
<th>Document Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovering JMP</td>
<td>If you are not familiar with JMP, start here.</td>
<td>Introduces you to JMP and gets you started creating and analyzing data.</td>
</tr>
<tr>
<td>Using JMP</td>
<td>Learn about JMP data tables and how to perform basic operations.</td>
<td>Covers general JMP concepts and features that span across all of JMP, including importing data, modifying columns properties, sorting data, and connecting to SAS.</td>
</tr>
</tbody>
</table>
| Basic Analysis      | Perform basic analysis using this document.                                     | Describes these Analyze menu platforms:
<pre><code>                                                                                                                             |
</code></pre>
<p>|                     |                                                                                  | • Distribution                                                                                                                                 |
|                     |                                                                                  | • Fit Y by X                                                                                                                                 |
|                     |                                                                                  | • Matched Pairs                                                                                                                                 |
|                     |                                                                                  | • Tabulate                                                                                                                                 |
|                     |                                                                                  | How to approximate sampling distributions using bootstrapping is also included.                                                                 |</p>
<table>
<thead>
<tr>
<th>Document Title</th>
<th>Document Purpose</th>
<th>Document Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Essential Graphing</strong></td>
<td>Find the ideal graph for your data.</td>
<td>Describes these Graph menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Graph Builder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Overlay Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Scatterplot 3D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Contour Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bubble Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parallel Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cell Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Treemap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Scatterplot Matrix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ternary Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Also covers how to create background and custom maps.</td>
</tr>
<tr>
<td><strong>Profilers</strong></td>
<td>Learn how to use interactive profiling tools, which enable you to view cross-sections of any response surface.</td>
<td>Covers all profilers listed in the Graph menu. Analyzing noise factors is included along with running simulations using random inputs.</td>
</tr>
<tr>
<td><strong>Design of Experiments Guide</strong></td>
<td>Learn how to design experiments and determine appropriate sample sizes.</td>
<td>Covers all topics in the <strong>DOE</strong> menu.</td>
</tr>
<tr>
<td>Document Title</td>
<td>Document Purpose</td>
<td>Document Content</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Fitting Linear Models</strong></td>
<td>Learn about Fit Model platform and many of its personalities.</td>
<td>Describes these personalities, all available within the Analyze menu Fit Model platform:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Standard Least Squares</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Stepwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Generalized Regression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Mixed Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MANOVA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Loglinear Variance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Nominal Logistic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ordinal Logistic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Generalized Linear Model</td>
</tr>
<tr>
<td><strong>Specialized Models</strong></td>
<td>Learn about additional modeling techniques.</td>
<td>Describes these Analyze &gt; Modeling menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Neural</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Model Comparison</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Nonlinear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Gaussian Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Time Series</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Response Screening</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Screening platform in the Analyze &gt; Modeling menu is described in <em>Design of Experiments Guide</em>.</td>
</tr>
<tr>
<td><strong>Multivariate Methods</strong></td>
<td>Read about techniques for analyzing several variables simultaneously.</td>
<td>Describes these Analyze &gt; Multivariate Methods menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Multivariate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cluster</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Principal Components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Discriminant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partial Least Squares</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Document Title</th>
<th>Document Purpose</th>
<th>Document Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality and Process Methods</td>
<td>Read about tools for evaluating and improving processes.</td>
<td>Describes these Analyze &gt; Quality and Process menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Control Chart Builder and individual control charts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Measurement Systems Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Variability / Attribute Gauge Charts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Pareto Plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Diagram</td>
</tr>
<tr>
<td>Reliability and Survival Methods</td>
<td>Learn to evaluate and improve reliability in a product or system and analyze survival data for people and products.</td>
<td>Describes these Analyze &gt; Reliability and Survival menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Life Distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fit Life by X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Recurrence Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Degradation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reliability Forecast</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reliability Growth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reliability Block Diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Survival</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fit Parametric Survival</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fit Proportional Hazards</td>
</tr>
<tr>
<td>Consumer Research</td>
<td>Learn about methods for studying consumer preferences and using that insight to create better products and services.</td>
<td>Describes these Analyze &gt; Consumer Research menu platforms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Categorical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Factor Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Choice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Uplift</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Item Analysis</td>
</tr>
</tbody>
</table>
Learn about JMP

Chapter 1

Additional Resources for Learning JMP

Note: The Books menu also contains two reference cards that can be printed: The Menu Card describes JMP menus, and the Quick Reference describes JMP keyboard shortcuts.

Scripting Guide
Learn about taking advantage of the powerful JMP Scripting Language (JSL).
Covers a variety of topics, such as writing and debugging scripts, manipulating data tables, constructing display boxes, and creating JMP applications.

JSL Syntax Reference
Read about many JSL functions on functions and their arguments, and messages that you send to objects and display boxes.
Includes syntax, examples, and notes for JSL commands.

JMP Help
JMP Help is an abbreviated version of the documentation library that provides targeted information. You can open JMP Help in several ways:

• On Windows, press the F1 key to open the Help system window.
• Get help on a specific part of a data table or report window. Select the Help tool from the Tools menu and then click anywhere in a data table or report window to see the Help for that area.
• Within a JMP window, click the Help button.
• Search and view JMP Help on Windows using the Help > Help Contents, Search Help, and Help Index options. On Mac, select Help > JMP Help.
• Search the Help at http://jmp.com/support/help/ (English only).

Additional Resources for Learning JMP

In addition to JMP documentation and JMP Help, you can also learn about JMP using the following resources:

• Tutorials (see “Tutorials” on page 23)
• Sample data (see “Sample Data Tables” on page 23)
• Indexes (see “Learn about Statistical and JSL Terms” on page 23)
Chapter 1
Specialized Models

Learn about JMP

Additional Resources for Learning JMP

• Tip of the Day (see “Learn JMP Tips and Tricks” on page 24)
• Web resources (see “JMP User Community” on page 24)
• JMPer Cable technical publication (see “JMPer Cable” on page 24)
• Books about JMP (see “JMP Books by Users” on page 25)
• JMP Starter (see “The JMP Starter Window” on page 25)

Tutorials

You can access JMP tutorials by selecting Help > Tutorials. The first item on the Tutorials menu is Tutorials Directory. This opens a new window with all the tutorials grouped by category.

If you are not familiar with JMP, then start with the Beginners Tutorial. It steps you through the JMP interface and explains the basics of using JMP.

The rest of the tutorials help you with specific aspects of JMP, such as creating a pie chart, using Graph Builder, and so on.

Sample Data Tables

All of the examples in the JMP documentation suite use sample data. Select Help > Sample Data to do the following actions:

• Open the sample data directory.
• Open an alphabetized list of all sample data tables.
• Find a sample data table within a category.

Sample data tables are installed in the following directory:

On Windows: C:\Program Files\SAS\JMP<version_number>\Samples\Data
On Macintosh: \Library\Application Support\JMP<version_number>\Samples\Data

In JMP Pro, sample data is installed in the JMP.Pro (rather than JMP) directory.

Learn about Statistical and JSL Terms

The Help menu contains the following indexes:

Statistics Index Provides definitions of statistical terms.

Scripting Index Lets you search for information about JSL functions, objects, and display boxes. You can also edit and run sample scripts from the Scripting Index.
Learn JMP Tips and Tricks

When you first start JMP, you see the Tip of the Day window. This window provides tips for using JMP.

To turn off the Tip of the Day, clear the Show tips at startup check box. To view it again, select Help > Tip of the Day. Or, you can turn it off using the Preferences window. See the Using JMP book for details.

Tooltips

JMP provides descriptive tooltips when you place your cursor over items, such as the following:

- Menu or toolbar options
- Labels in graphs
- Text results in the report window (move your cursor in a circle to reveal)
- Files or windows in the Home Window
- Code in the Script Editor

Tip: You can hide tooltips in the JMP Preferences. Select File > Preferences > General (or JMP > Preferences > General on Macintosh) and then deselect Show menu tips.

JMP User Community

The JMP User Community provides a range of options to help you learn more about JMP and connect with other JMP users. The learning library of one-page guides, tutorials, and demos is a good place to start. And you can continue your education by registering for a variety of JMP training courses.

Other resources include a discussion forum, sample data and script file exchange, webcasts, and social networking groups.

To access JMP resources on the website, select Help > JMP User Community.

JMPer Cable

The JMPer Cable is a yearly technical publication targeted to users of JMP. The JMPer Cable is available on the JMP website:

http://www.jmp.com/about/newsletters/jmpercable/
JMP Books by Users

Additional books about using JMP that are written by JMP users are available on the JMP website:

http://www.jmp.com/support/books.shtml

The JMP Starter Window

The JMP Starter window is a good place to begin if you are not familiar with JMP or data analysis. Options are categorized and described, and you launch them by clicking a button. The JMP Starter window covers many of the options found in the Analyze, Graph, Tables, and File menus.

- To open the JMP Starter window, select View (Window on the Macintosh) > JMP Starter.
- To display the JMP Starter automatically when you open JMP on Windows, select File > Preferences > General, and then select JMP Starter from the Initial JMP Window list. On Macintosh, select JMP > Preferences > Initial JMP Starter Window.
Introduction to Specialized Modeling
Overview of Modeling Techniques

Specialized Models provides details about more technical modeling techniques, such as Response Screening, Partitioning, and Neural Networks.

- The Partition platform recursively partitions data according to a relationship between the \( X \) and \( Y \) values, creating a tree of partitions. See Chapter 3, “Partition Models”.
- 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. See Chapter 4, “Neural Networks”.
- The Model Comparison platform lets you compare the predictive ability of different models. Measures of fit are provided for each model along with overlaid diagnostic plots. See Chapter 5, “Model Comparison”.
- The Nonlinear platform’s Fit Curve personality provides predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models. Compare different groups or subjects using a variety of analytical and graphical techniques. See Chapter 6, “Nonlinear Regression with Built-In Models”.
- The Nonlinear platform lets you fit custom nonlinear models, which include a model formula and parameters to be estimated. See Chapter 7, “Nonlinear Regression with Custom Models”.
- The Gaussian Process platform models 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. See Chapter 8, “Gaussian Process”.
- The Time Series platform lets you explore, analyze, and forecast univariate time series. See Chapter 9, “Time Series Analysis”.
- The Response Screening platform automates the process of conducting tests across a large number of responses. Your test results and summary statistics are presented in data tables, rather than reports, to enable data exploration. See Chapter 10, “Response Screening”.

Chapter 2
Specialized Models
Chapter 3

Partition Models

Use Decision Trees to Explore and Model Your Data

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, because it chooses 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
- Boosted Tree

The Bootstrap Forest and Boosted Tree methods are available only in JMP Pro.

**Figure 3.1** Example of a Partition Plot
## Contents

- Overview of Partition ......................................................... 31
- Example of Partition .......................................................... 31
- Launching the Partition Platform .......................................... 33
- Partition Method ................................................................. 34
  - Decision Tree .................................................................. 34
  - Bootstrap Forest ............................................................. 46
  - Boosted Tree ................................................................. 50
- Validation ............................................................................ 53
- Graphs for Goodness of Fit .................................................. 54
  - Actual by Predicted Plot .................................................. 54
  - ROC Curve ..................................................................... 55
  - Lift Curves ...................................................................... 57
- Informative Missing .............................................................. 59
- Examples of Bootstrap Forest, Boosted Tree, and Model Comparison .................................................. 59
  - Decision Tree ................................................................. 60
  - Bootstrap Forest ............................................................. 60
  - Boosted Tree ................................................................. 60
  - Compare Methods ............................................................ 61
  - Model Comparison .......................................................... 62
- Statistical Details .................................................................. 63
Overview of Partition

Variations of partitioning go by many names and brand names: decision trees, CART™, CHAID™, C4.5, C5, and others. The technique is often taught as a data mining technique because:

- it is useful 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. These question help diagnose new patients more quickly.

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 about split criteria, see “Statistical Details” on page 63.

Example of Partition

The Boston Housing.jmp sample data table contains data on the price of homes in Boston area communities, along with demographic information. The Partition platform can be used to model the home values.

1. Open the Boston Housing.jmp sample data table.
2. Select Analyze > Modeling > Partition.
3. Assign mvalue to the Y, Response role.
4. Assign all other variables (crim through lstat) to the X, Factor role.
5. Select the Decision Tree option from the Method menu. If using JMP, the Decision Tree option is the method that is used.
6. Enter 0.2 for the Validation Portion.
7. Click OK.
8. On the platform report window, click Go to perform automatic splitting.
The fit statistics are shown in Figure 3.2. Automatic splitting resulted in five splits. The final RSquare for the Validation set is 0.569.

**Note:** Because you are using a random Validation Portion, your results differ from those in Figure 3.2.

**Figure 3.2** Fit Statistics for Boston Housing Data

```
<table>
<thead>
<tr>
<th></th>
<th>RSquare</th>
<th>RMSE</th>
<th>Number of Splits</th>
<th>AICc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>0.810</td>
<td>3.935937</td>
<td>404</td>
<td>5 2267.38</td>
</tr>
<tr>
<td>Validation</td>
<td>0.569</td>
<td>6.444980</td>
<td>102</td>
<td></td>
</tr>
</tbody>
</table>
```

8. Select Plot Actual by Predicted from the platform red-triangle menu.

A portion of the report is shown in Figure 3.3. Because there are five splits, the tree has six leaves. Therefore, there are six distinct predicted values.

**Figure 3.3** Decision Tree Results
9. Select **Column Contributions** from the report’s red triangle menu.
   
   The report (Figure 3.22) shows that the predictors that contribute the most to the model are rooms and lstat.

10. From the report’s red triangle menu, select **Save Columns > Save Prediction Formula**.
    
    This saves your prediction formula to the data table in a column called *mvalue Predictor*.
    
    See “Examples of Bootstrap Forest, Boosted Tree, and Model Comparison” on page 59 for a continuation of this example.

---

**Launching the Partition Platform**

To launch the Partition platform, select **Analyze > Modeling > Partition**. The Partition launch window is shown in Figure 3.4, using the Boston Housing.jmp data table.

**Figure 3.4** Partition Launch Window

**Table 3.1** Descriptions of Launch Window

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y, Response</td>
<td>Choose the response variable.</td>
</tr>
<tr>
<td>X, Factor</td>
<td>Choose the predictor variables.</td>
</tr>
<tr>
<td>Validation</td>
<td>Enter a validation column here. For more information about validation, see “Validation” on page 53.</td>
</tr>
</tbody>
</table>
The Partition platform provides three methods for producing a final tree:

- For the Decision Tree method, see “Decision Tree” on page 34.
- For the Bootstrap Forest method, see “Bootstrap Forest” on page 46.
- For the Boosted Tree method, see “Boosted Tree” on page 50.

**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. Because the reports for continuous and categorical responses differ, details are presented separately.

**Decision Tree Report for Continuous Responses**

As an example for a continuous response, use the Boston Housing jmp data table. Assign mvalue to the Y, Response role. Assign all the other variables to the X, Factor role. Set the
Validation Portion to 0 so that your results match those shown here. If using JMP Pro, select Decision Tree from the Method menu. Click OK. The initial report displays the partition graph, control buttons, a summary panel, and the first node of the tree (Figure 3.5).

Figure 3.5  Decision Tree Initial Report for a Continuous Response

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.

Summary Panel (Continuous Response)

RSquare  The current value of $R^2$.

RMSE  The root mean square error.

N  The number of observations (if no Freq variable is used).

Number of Splits  The current number of splits.

AICc  The corrected Akaike's Information Criterion.

Node (Continuous Response)

Count  The number of rows in the branch.

Mean  The average response for all rows in that branch.

Std Dev  The standard deviation of the response for all rows in that branch.
Candidates (Continuous Response)

Initially, all rows are in one branch. For each column, the Candidates report gives details about the optimal split. In order to determine the split, each X column, and all possible splits for that column, are considered. The columns of the Candidate report are:

**Term**  Shows the candidate columns.

**Candidate SS**  Sum of squares for the best split. Shown if the response is continuous.

**Candidate G^2**  Likelihood ratio chi-square for the best split. Shown if the response is categorical.

**LogWorth**  The LogWorth statistic, defined as $-\log_{10}(p$-value). The optimal split is the one that maximizes the LogWorth. See “Statistical Details” on page 63 for additional details.

**Cut Point**  For a continuous term, the single value that determines the split is given. For a categorical term, the levels in the left-most split are listed.

As shown in Figure 3.6, the rooms column has the largest LogWorth and therefore defines the optimum split. The Cut Point value of 6.943 indicates that the split is into the nodes: rooms < 6.943 and rooms ≥ 6.943.

**Figure 3.6** Candidates Report (Continuous Response)

The optimum split is noted by an asterisk. However, there are cases where the Candidate 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” on page 63 for more information about LogWorth and SS.

Click the Split button and notice the first split is made on the column rooms, at a value of 6.943. Open the two new candidate reports (Figure 3.7).
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 lstat, which has an SS of 7,311.85. For the right leaf, the next split would happen on the column rooms, which has an SS of 3,060.95. Because the SS for the left leaf is higher, using the Split button again produces a split on the left leaf, on the column lstat.
Click the **Split** button to make the next split (Figure 3.8).

**Figure 3.8** Second Split (Continuous Response)

![Partition for mvalue](image)

The 430 observations from the previous left leaf are now split into two parts:

- A left leaf, corresponding to $lstat \geq 14.43$, has 175 observations.
- A right leaf, corresponding to $lstat < 14.43$, has 255 observations.

The 506 original observations are now split into three parts:

- A leaf corresponding to $rooms < 6.943$ and $lstat \geq 14.43$.
- A leaf corresponding to $rooms < 6.943$ and $lstat < 14.43$.
- 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. Click the Split button to perform splits. Hold the Shift key as you click the Split button to specify multiple splits. If validation is not enabled, Partition is an exploratory platform intended to help you investigate relationships interactively.

When 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” on page 45 for details about the stopping rule.

**Decision Tree Report for Categorical Responses**

As an example for a categorical response, use the Car Poll jmp data table. Assign sex to the Y, Response role. Assign all the other variables to the X, Factor role. Set the Validation Portion to 0 so that your results agree with those shown here. If using JMP Pro, select Decision Tree from the Method menu. Click OK.

In the report, select Display Options > Show Split Prob. Click Split twice. The report is shown in Figure 3.9.

- 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” on page 63 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. (Select Display Options > Show Split Prob.)

- The Prob statistic is the predicted value (a probability) for each response level. See “Statistical Details” on page 63 for more information about the Prob statistic. (Select Display Options > Show Split Prob.)

- 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. If Display Options > Show Points is selected:
  - Points are distributed evenly and randomly in the horizontal direction.
  - In the vertical direction, they are distributed randomly within the box for their category.

- The Color Points button appears. This colors the points on the plot according to the response levels.
**Partition Models**

Chapter 3
Specialized Models

**Figure 3.9** Decision Tree Report for a Categorical Response

**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.
Chapter 3
Specialized Models

Partition Models
Partition Method

Specialized Models Partition Method

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 3.10  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  enables 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” on page 63 for more information about the categorical split statistic $G^2$.
- **Show Split Prob** shows or hides the Rate and Prob statistics. This is for categorical responses only.

JMP automatically shows the Rate and Prob statistics when you select **Show Split Count**. See “Statistical Details” on page 63 for more information about Rate and Prob.

- **Show Split Count** shows or hides each frequency level for all nodes in the tree. This is for categorical responses only.
  
  When you select this option, JMP automatically selects **Show Split Prob**. And when you deselect **Show Split Prob**, the counts do not appear.
  
- **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 does not 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 to define the minimum size split allowed. To specify a number, enter a value greater than or equal to 1. To specify a fraction of the sample size, enter a value less than 1. The default value is set to the minimum of 5 or the floor of the number of rows divided by 10,000.
- **Lock Columns** reveals a check box table to enable 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  Displays a report showing each input column's contribution to the fit. The report also shows how many times it defined a split and the total $C^2$ or Sum of Squares attributed to that column.

Split History  shows a plot of $R^2$ versus 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 about validation, see “Validation” on page 53.

ROC Curve  is described in the section “ROC Curve” on page 55. This is for categorical responses only.

Lift Curve  is described in the section “Lift Curves” on page 57. This is for categorical responses only.

Show Fit Details  Appears only for categorical responses. Shows several measures of fit and a confusion matrix. The confusion matrix is a two-way classification of actual and predicted responses.

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. A category label for a leaf might refer to an inclusive list of categories in a higher tree node. A caret (“^”) appears 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 consists of nested conditional clauses that describe the tree structure. If the response is continuous, the column contains a Predicting property. If the response is categorical, the column contains a Response Probability property.
- **Save Tolerant Prediction Formula** saves a formula that predicts even when there are missing values and when Informative Missing has not been checked. The prediction formula tolerates missing values by randomly allocating response values for missing predictors to a split. If the response is continuous, the column contains a Predicting property. If the response is categorical, the column contains a Response Probability property. If you have checked Informative Missing, you can save the Tolerant Prediction Formula by holding the Shift key as you click on the report’s red triangle.
- **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.

**Specify Profit Matrix** Enables you to specify profit or costs associated with correct or incorrect classification decisions. Only available for categorical responses. You can assign profit and cost values to each combination of actual and predicted response categories. A row labeled Undecided enables you to specify the costs of classifying into an alternative category. Checking **Save to Column as Property** saves your assignments to the response column as a property. Not checking **Save to Column as Property** applies the Profit Matrix only to the current Partition report.

When you select **Save Columns > Save Prediction Formula** from the report’s red triangle menu, additional columns with formulas are saved to the data table. These columns are:

- **Profit for <level>**: For each level of the response, a column gives the expected profit for classifying each observation into that level.
- Most Profitable Prediction for <column name>: For each observation, gives the level of the response with the highest expected profit.
- Expected Profit for <column name>: For each observation, gives the expected profit for the classification defined by the Most Profitable Prediction column.
- Actual Profit for <column name>: For each observation, gives the actual profit for classifying that observation into the level specified by the Most Profitable Prediction column.

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 “Decision Tree Report for Categorical Responses” on page 39).

Script contains options that are available to all platforms. See Using JMP.

Automatic Splitting

The Go button (shown in Figure 3.11) appears when you enable validation. For more information about using validation, see “Validation” on page 53.

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 might 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.
Bootstrap Forest

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 3.12 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 3.12  Bootstrap Forest Fitting Options

![Bootstrap Forest Options Window]

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.

**Maximum Splits Per Tree**  is the maximum number of splits for each tree.

**Minimum Size Split**  is the minimum number of observations needed on a candidate split.

**Early Stopping**  is checked by default. The process stops growing additional trees if adding more trees does not 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.

**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 3.13.

Figure 3.13  Bootstrap Forest

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 about 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 available only 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:

**Plot Actual by Predicted**  Provides a plot of actual versus predicted values. This is only for continuous responses.

**Column Contributions**  Displays a report that shows each input column's contribution to the fit. The report also shows how many times it defined a split and the total $G^2$ or Sum of Squares attributed to that column.

**Show Trees**  Provides a menu of options 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.

**ROC Curve**  is described in the section “ROC Curve” on page 55. This is for categorical responses only.

**Lift Curve**  is described in the section “Lift Curves” on page 57. This is for categorical responses only.

**Save Columns**  Provides a menu of options 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 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. Only available if validation is used.

**Make SAS DATA Step** creates SAS code for scoring a new data set.

**Specify Profit Matrix**  Enables you to specify profit or costs associated with correct or incorrect classification decisions. Only available for categorical responses. See “Specify Profit Matrix” on page 44.

**Script**  contains options that are available to all platforms. See *Using JMP*. 
Boosted Tree

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. Figure 3.14 shows the options window for the Car Poll.jmp sample data table with sex as Y, Response, all other columns as X, Factor, and a Validation Portion of 0.2.

**Figure 3.14  Boosted Tree Options Window**

![Boosted Tree Options Window]

The options on the Boosted Tree options window are:

- **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 protect against fitting probabilities equal to zero. Only appears if the response is categorical.

Minimum Size Split is the minimum number of observations needed on a candidate split.

Early Stopping is checked by default. The boosting process stops fitting additional stages if adding more stages does not improve the validation statistic. If not checked, the boosting process continues until the specified number of stages is reached. This option appears 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 3.15.

Figure 3.15 Boosted Tree Report
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 about 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 available only 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. For details about the options, see “Show Trees” on page 49.

**Plot Actual by Predicted** provides a plot of actual versus predicted values. This is only for continuous responses.

**Column Contributions** Displays a report showing each input column’s contribution to the fit. The report also shows how many times it defined a split and the total $G^2$ or Sum of Squares attributed to that column.

**ROC Curve** is described in the section “ROC Curve” on page 55. This is for categorical responses only.

**Lift Curve** is described in the section “Lift Curves” on page 57. 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 Residuals** saves the residuals to the data table. This is for continuous responses only.
- **Save Offset Estimates** saves the sums of the linear components. These are the logits of the fitted probabilities. 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. Only available is validation is used.

**Make SAS DATA Step** creates SAS code for scoring a new data set.
**Specify Profit Matrix**  Enables you to specify profit or costs associated with correct or incorrect classification decisions. Only available for categorical responses. See “Specify Profit Matrix” on page 44.

**Script**  contains options that are available to all platforms. See Using JMP.

### 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 3.1).

The training, validation, and test sets are created by subsetting the original data into parts. Table 3.2 describes several methods for subsetting a data set.

#### Table 3.2 Validation Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluded Rows</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>For more information about using row states and how to exclude rows, see Using JMP.</td>
</tr>
<tr>
<td>Holdback</td>
<td>Randomly divides the original data into the training and validation data sets. The <strong>Validation Portion</strong> (see Table 3.1) on the platform launch window is used to specify the proportion of the original data to use as the validation data set (holdback).</td>
</tr>
</tbody>
</table>
Graphs for Goodness of Fit Specialized Models

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. When validation is used, plots are shown for both the training and the
validation sets. See Figure 3.16.

When you fit a Decision Tree, observations in a leaf have the same predicted value. If there are \( n \) leaves, then the Actual by Predicted plot shows at most \( n \) distinct predicted values. This gives the plot the appearance of having points arranged on vertical lines. Each of these lines corresponds to a predicted value for some leaf.

**Figure 3.16** Actual by Predicted Plots for Boston Housing Data

---

**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”. 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. Starting at the lower left corner, the curve is drawn 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 Ys where $Y=1$, and the X axis shows the number of Ys 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 3.17 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 crosses 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 (that is 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, the top-rated 10% of fitted probabilities might have a 25% richness of
the chosen response compared with 5% richness over the whole population. Then the lift curve goes 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

When 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 3.18 Lift Curve
Informative Missing

The Informative Missing option enables informative treatment of missing values on the predictors. The model that is fit is deterministic. The option is on by default, in which case the missing values are handled as follows:

- Rows containing missing values for a categorical predictor are entered into the analysis as a separate level of the variable.
- Rows containing missing values for a continuous predictor are assigned to a split as follows. The values of the continuous predictor are sorted. Missing rows are first considered to be on the low end of the sorted values. All splits are constructed. The missing rows are then considered to be on the high end of the sorted values. Again, all splits are constructed. The optimal split is determined using the LogWorth criterion. For further splits on the given predictor, the algorithm commits the missing rows to be treated as high or low values, as determined by the first split induced by that predictor.

If the Informative Missing option is not selected:

- Rows containing missing values for predictor variables are included in the following way. Each row with a missing value on that predictor is randomly assigned to one of the two sides of the split. When this happens using the Decision Tree method, the Imputes message appears, which shows how many times this has happened. See Figure 3.19, where five rows have missing values.

Example of Bootstrap Forest, Boosted Tree, and Model Comparison

The examples in this section use the Boston Housing jmp data table. Your goal is to fit a model to predict the median home value as a function of several demographic characteristics. Build a tree using all three partitioning methods, and compare the results.

Note: Results vary because the Validation Portion option chooses rows at random to use as the training and validation sets.
Decision Tree

In “Example of Partition” on page 31, you fit a Decision Tree to the Boston Housing.jmp data. The prediction formula that you saved was placed in the column mvalue Predictor. For the Validation Portion used, Automatic Splitting resulted in five splits and a Validation RSquare of 0.569.

Bootstrap Forest

Follow the steps below to fit a model using the Bootstrap Forest method:
1. Open the Boston Housing.jmp data table or select it as your active table.
2. Select Analyze > Modeling > Partition.
3. Assign mvalue to the Y, Response role.
4. Assign the other variables (crim through lstat) to the X, Factor role.
5. Select Bootstrap Forest from the Method menu.
6. Enter 0.2 for the Validation Portion.
7. Click OK.
8. Click OK to accept the defaults in the Bootstrap Forest options window.
   The Overall Statistics report is shown in Figure 3.20.

Figure 3.20 Bootstrap Forest Overall Statistics

9. From the report’s red triangle menu, select Save Columns > Save Prediction Formula.
   This saves the prediction formula for the Bootstrap Forest model to the data table in a column called mvalue Predictor 2.

Boosted Tree

Follow the steps below to fit a model using the Boosted Tree method:
1. Open the Boston Housing.jmp data table or select it as your active table.
2. Select **Analyze > Modeling > Partition**.
3. Assign mvalue to the **Y, Response** role.
4. Assign the other variables (crim through lstat) to the **X, Factor** role.
5. Select **Boosted Tree** from the Method menu.
6. Enter 0.2 for the **Validation Portion**.
7. Click **OK**.
8. Click **OK** to accept the defaults in the Boosted Tree options window.
9. Select **Plot Actual by Predicted** on the red-triangle menu.
   The report is shown in Figure 3.21.

**Figure 3.21** Boosted Tree Overall Statistics

![Overall Statistics Table]

<table>
<thead>
<tr>
<th></th>
<th>RSquare</th>
<th>RMSE</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>0.917</td>
<td>2.5961376</td>
<td>404</td>
</tr>
<tr>
<td>Validation</td>
<td>0.792</td>
<td>4.4715792</td>
<td>102</td>
</tr>
</tbody>
</table>

10. From the report’s red triangle menu, select **Save Columns > Save Prediction Formula**.
   This saves the prediction formula for the Bootstrap Forest model to the data table in a column called mvalue Predictor 3.

**Compare Methods**

Based on the particular validation portion chosen in this example, the Decision Tree has the highest Validation RSquare value:

- Decision Tree 0.569
- Bootstrap Forest 0.626
- Boosted Tree 0.792

Compare your RSquare values and your Actual by Predicted plots.

Figure 3.22 shows a summary of the **Column Contributions** report from each method. For Decision Tree and Boosted Tree, rooms and lstat are the major contributors.
Model Comparison

The Model Comparison platform enables you to compare all three models. The comparison is for all observations. In JMP Pro, you can define a validation column. It is often useful to use Model Comparison only for rows in the validation column.

1. Identify the three predictor columns that you have saved in using the Decision Tree, Bootstrap Forest, and Boosted Tree methods.

2. Select **Analyze > Modeling > Model Comparison**.
3. Enter all three prediction formula columns as \textit{Y, Predictors}.

4. Click \textit{OK}.

The report, entitled Measures of Fit for mvalue, is shown in Figure 3.23. The Boosted Tree method has the highest RSquare. It also has the lowest root average square error (RASE) and average absolute error (AAE).

\textbf{Figure 3.23} Measures of Fit for Three Partition Methods

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Creator</th>
<th>.2.4.6.8</th>
<th>RSquare</th>
<th>RASE</th>
<th>AAE</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvalue Predictor</td>
<td>Partition</td>
<td></td>
<td>0.7545</td>
<td>4.5527</td>
<td>3.0132</td>
<td>506</td>
</tr>
<tr>
<td>mvalue Predictor 2</td>
<td>Bootstrap Forest</td>
<td></td>
<td>0.8029</td>
<td>4.0789</td>
<td>2.5104</td>
<td>506</td>
</tr>
<tr>
<td>mvalue Predictor 3</td>
<td>Boosted Tree</td>
<td></td>
<td>0.8885</td>
<td>3.0679</td>
<td>2.0961</td>
<td>506</td>
</tr>
</tbody>
</table>

\textbf{Statistical Details}

This section provides some quantitative details and other information.

\textbf{General}

The response can be either continuous, or categorical (nominal or ordinal). If \textit{Y} is categorical, then it is fitting the probabilities estimated for the response levels, minimizing the residual log-likelihood chi-square \[2\times\text{entropy}\]. 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 \textit{X} is continuous, then the partition is done according to a splitting “cut” value for \textit{X}. If \textit{X} is categorical, then it divides the \textit{X} categories into two groups of levels and considers all possible groupings into two levels.

\textbf{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 \textit{X}s with many levels, and the Bonferroni \(p\)-value, which favors \textit{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.
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

$$ SS_{test} = SS_{parent} - (SS_{right} + SS_{left}) $$

where SS 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

$$ G^2_{test} = G^2_{parent} - (G^2_{left} + G^2_{right}). $$

Partition actually has two rates; one used for training that is the usual ration of count to total, and another that is slightly biased away from zero. By never having attributed probabilities of zero, this allows logs of probabilities to be calculated on validation or excluded sets of data, used in Entropy R Squares.

**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 \( i^{th} \) response level at a given node is as follows:

$$ \text{Prob}_i = \frac{n_i + \text{prior}_i}{\sum (n_j + \text{prior}_j)} $$

where the summation is across all response levels; \( n_i \) is the number of observations at the node for the \( i^{th} \) response level; and \( \text{prior}_i \) is the prior probability for the \( i^{th} \) response level, calculated as
prior_i = \lambda p_i + (1-\lambda)P_i

where p_i is the prior from the parent node, P_i is the Prob 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 nonzero.
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 4.1 Example of a Neural Network
Contents

Overview of Neural Networks ..................................................... 69
Launch the Neural Platform ....................................................... 69
  The Neural Launch Window .................................................. 70
  The Model Launch ............................................................... 71
Model Reports ............................................................................. 77
  Training and Validation Measures of Fit .................................. 78
  Confusion Statistics ................................................................. 79
Model Options ............................................................................ 79
Example of a Neural Network ..................................................... 81
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 4.2 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.

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” on page 74.

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.

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.

Figure 4.3 The Neural Launch Window

Table 4.1 Description of the Neural Launch Window

| **Y, Response** | 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). |
| **X, Factor**   | Choose the input variables. |
| **Freq**        | Choose a frequency variable. |
| **Validation**  | Choose a validation column. For more information, see “Validation Method” on page 73. |
| **By**          | Choose a variable to create separate models for each level of the variable. |
Table 4.1 Description of the Neural Launch Window  (Continued)

| Missing Value Coding | Check this box to enable informative coding of missing values. This coding allows estimation of a predictive model despite the presence of missing values. It is useful in situations where missing data are informative. If this option is not checked, rows with missing values are ignored. For a continuous variable, missing values are replaced by the mean of the variable. Also, a missing value indicator, named \(<\text{colname}> \text{ Is Missing}\), is created and included in the model. If a variable is transformed using the Transform Covariates fitting option on the Model Launch window, missing values are replaced by the mean of the transformed variable. For a categorical variable, missing values are treated as a separate level of that variable. |

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 4.4 The Model Launch Dialog

Table 4.2 Description of the Model Launch Dialog

<table>
<thead>
<tr>
<th>Validation Method</th>
<th>Select the method that you want to use for model validation. For details, see “Validation Method” on page 73.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Layer Structure or Hidden Nodes</td>
<td><strong>Note:</strong> The standard edition of JMP uses only the TanH activation function, and can fit only neural networks with one hidden layer. Specify the number of hidden nodes of each type in each layer. For details, see “Hidden Layer Structure” on page 74.</td>
</tr>
<tr>
<td>Boosting</td>
<td>Specify options for gradient boosting. For details, see “Boosting” on page 75.</td>
</tr>
<tr>
<td>Fitting Options</td>
<td>Specify options for variable transformation and model fitting. For details, see “Fitting Options” on page 76.</td>
</tr>
<tr>
<td>Go</td>
<td>Fits the neural network model and shows the model reports.</td>
</tr>
</tbody>
</table>

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 4.3.

The training, validation, and test sets are created by subsetting the original data into parts. Table 4.3 describes several methods for subsetting a data set.

**Table 4.3 Validation Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluded Rows</td>
<td>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 Using JMP.</td>
</tr>
<tr>
<td>Holdback</td>
<td>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).</td>
</tr>
<tr>
<td>KFold</td>
<td>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.</td>
</tr>
</tbody>
</table>
Chapter 4
Overview of Neural Networks

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 4.4 describes the three types of activation functions.
Chapter 4
Neural Networks

Specialized Models Overview of Neural Networks

Boosting

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.

Table 4.4 Activation Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TanH</td>
<td>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: $\frac{e^{2x} - 1}{e^{2x} + 1}$, where $x$ is a linear combination of the X variables.</td>
</tr>
</tbody>
</table>
| Linear   | The identity function. The linear combination of the X variables is not transformed. 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.

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. |
| Gaussian | 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: $e^{-x^2}$, where $x$ is a linear combination of the X variables. |
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

Table 4.5 describes the model fitting options that you can specify.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transform Covariates</td>
<td>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 “Model Options” on page 79.</td>
</tr>
<tr>
<td>Robust Fit</td>
<td>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.</td>
</tr>
<tr>
<td>Penalty Method</td>
<td>Choose the penalty method. To mitigate the tendency neural networks have to overfit data, the fitting process incorporates a penalty on the likelihood. See “Penalty Method” on page 77.</td>
</tr>
<tr>
<td>Number of Tours</td>
<td>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.</td>
</tr>
</tbody>
</table>
Penalty Method

The penalty is $\lambda p(\beta_i)$, where $\lambda$ is the penalty parameter, and $p(\cdot)$ is a function of the parameter estimates, called the penalty function. Validation is used to find the optimal value of the penalty parameter.

Table 4.6 Descriptions of Penalty Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Penalty Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squared</td>
<td>$\sum \beta_i^2$</td>
<td>Use this method if you think that most of your $X$ variables are contributing to the predictive ability of the model.</td>
</tr>
<tr>
<td>Absolute</td>
<td>$\sum</td>
<td>\beta_i</td>
</tr>
<tr>
<td>Weight Decay</td>
<td>$\sum \frac{\beta_i^2}{1 + \beta_i^2}$</td>
<td></td>
</tr>
<tr>
<td>NoPenalty</td>
<td>none</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 4.5  Example of a Model Report

<table>
<thead>
<tr>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>chas Measures</td>
<td>chas Measures</td>
</tr>
<tr>
<td>0.6415619</td>
<td>0.6153345</td>
</tr>
<tr>
<td>0.4520822</td>
<td>0.4516082</td>
</tr>
<tr>
<td>0.8590396</td>
<td>0.8269821</td>
</tr>
<tr>
<td>0.647777</td>
<td>0.614419</td>
</tr>
<tr>
<td>35.078756</td>
<td>23.743427</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
<th>Confusion Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual  Predicted</td>
<td>Actual  Predicted</td>
</tr>
<tr>
<td>chas</td>
<td>chas</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confusion Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual  Predicted</td>
</tr>
<tr>
<td>0.89363</td>
</tr>
<tr>
<td>0.60870</td>
</tr>
</tbody>
</table>

Training and Validation Measures of Fit

Measures of fit appear for the training and validation sets. See Figure 4.5.

Table 4.7  Descriptions of the Training and Validation Measures of Fit

<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalized RSquare</td>
<td>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.</td>
</tr>
<tr>
<td>Entropy RSquare</td>
<td>Compares the log-likelihoods from the fitted model and the constant probability model. Appears only when the response is nominal or ordinal.</td>
</tr>
<tr>
<td>RSquare</td>
<td>Gives the Rsquare for the model.</td>
</tr>
<tr>
<td>RMSE</td>
<td>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).</td>
</tr>
<tr>
<td>Mean Abs Dev</td>
<td>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).</td>
</tr>
</tbody>
</table>
Table 4.7 Descriptions of the Training and Validation Measures of Fit (Continued)

<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misclassification Rate</td>
<td>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.</td>
</tr>
<tr>
<td>-LogLikelihood</td>
<td>Gives the negative of the log likelihood.</td>
</tr>
<tr>
<td>SSE</td>
<td>Gives the error sums of squares. Available only when the response is continuous.</td>
</tr>
<tr>
<td>Sum Freq</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 4.5. 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 4.8 describes the options in the red triangle menus.

Table 4.8 Model Report Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagram</td>
<td>Shows a diagram representing the hidden layer structure.</td>
</tr>
<tr>
<td>Show Estimates</td>
<td>Shows the parameter estimates in a report.</td>
</tr>
<tr>
<td>Profiler</td>
<td>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 Profiler chapter in the Profilers book.</td>
</tr>
</tbody>
</table>
Table 4.8 Model Report Options (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categorical Profiler</td>
<td>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 Profiler chapter in the Profilers book.</td>
</tr>
<tr>
<td>Contour Profiler</td>
<td>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 Contour Profiler chapter in the Profilers book.</td>
</tr>
<tr>
<td>Surface Profiler</td>
<td>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 Surface Profiler chapter in the Profilers book.</td>
</tr>
<tr>
<td>ROC Curve</td>
<td>Creates an ROC curve. Available only for nominal or ordinal responses. For details about ROC Curves, see “ROC Curve” on page 55 in the “Partition Models” chapter.</td>
</tr>
<tr>
<td>Lift Curve</td>
<td>Creates a lift curve. Available only for nominal or ordinal responses. For details about Lift Curves, see “Lift Curves” on page 57 in the “Partition Models” chapter.</td>
</tr>
<tr>
<td>Plot Actual by Predicted</td>
<td>Plots the actual versus the predicted response. Available only for continuous responses.</td>
</tr>
<tr>
<td>Plot Residual by Predicted</td>
<td>Plots the residuals versus the predicted responses. Available only for continuous responses.</td>
</tr>
<tr>
<td>Save Formulas</td>
<td>Creates new columns in the data table containing formulas for the predicted response and the hidden layer nodes.</td>
</tr>
<tr>
<td>Save Profile Formulas</td>
<td>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.</td>
</tr>
<tr>
<td>Save Fast Formulas</td>
<td>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.</td>
</tr>
<tr>
<td>Make SAS Data Step</td>
<td>Creates SAS code that you can use to score a new data set.</td>
</tr>
</tbody>
</table>
Specialized Models

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 4.6.

Note: Results will vary due to the random nature of choosing a validation set.
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 4.7.

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 4.8.
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 5

Model Comparison

Compare the Predictive Ability of Fitted Models

The Model Comparison Platform is available in JMP Pro only.

The Model Comparison platform in JMP Pro lets you compare the predictive ability of different models. Measures of fit are provided for each model along with overlaid diagnostic plots. Figure 5.1 shows an example of the Model Comparison report.

Figure 5.1 Example of Comparing Models
Contents

Example of Model Comparison ....................................................... 87
Launch the Model Comparison Platform ........................................... 90
The Model Comparison Report ......................................................... 90
Model Comparison Platform Options ............................................... 92
Additional Example of Model Comparison ....................................... 93
Example of Model Comparison

This section provides an example of using the Model Comparison platform. The example uses demographic data to build a model for median home price. A regression model and a bootstrap forest model are compared.

Begin by opening the Boston Housing.jmp sample data table.

Create a Validation Column
1. Create a column called validation.
2. On the Column Info window, select Random from the Initialize Data list.
3. Select the Random Indicator radio button.
4. Click OK.

The rows assigned a 0 are the training set. The rows assigned a 1 are the validation set.

Create the Regression Model and Save the Prediction Formula to a Column
1. Select Analyze > Fit Model.
2. Select mvalue and click Y.
3. Select the other columns (except validation) and click Add.
4. Select Stepwise in the Personality list.
5. Select validation and click Validation.
6. Click the Run button.
7. Click the Go button.
8. Click the Run Model button.

The Fit Group report appears, a portion of which is shown in Figure 5.2.
9. Save the prediction formula to a column by selecting Save Columns > Prediction Formula on the Response red triangle menu.
Figure 5.2  Fit Model Report

Create the Bootstrap Forest Model and Save the Prediction Formula to a Column

1. Select Analyze > Modeling > Partition.
2. Select mvalue and click Y, Response.
3. Select the other columns (except validation) and click X, Factor.
4. Select validation and click Validation.
5. Select Bootstrap Forest in the Method list.
6. Click OK.
7. Select the Early Stopping check box.
8. Select the Multiple Fits over number of terms check box.
9. Click OK.

The Bootstrap Forest report appears, a portion of which is shown in Figure 5.3.

10. Save the prediction formula to a column by selecting Save Columns > Save Prediction Formula on the Bootstrap Forest red triangle menu.
**Figure 5.3** Bootstrap Forest Model

<table>
<thead>
<tr>
<th>Bootstrap Forest for mvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Validation-Set Summaries</strong></td>
</tr>
<tr>
<td>The fit below was the best of these models fit</td>
</tr>
<tr>
<td>N Terms</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

**Specifications**

- Target Column: mvalue
- Validation Column: validation
- Number of trees in the forest: 37
- Number of terms sampled per split: 6
- Training rows: 405
- Validation rows: 101
- Test Rows: 0
- Number of terms: 13
- Bootstrap samples: 405
- Minimum Splits Per Tree: 10
- Minimum Size Split: 5

**Compare the Models**

1. Select **Analyze > Modeling > Model Comparison**.
2. Select the two prediction formula columns and click **Y, Predictors**.
3. Select validation and click **Group**.
4. Click **OK**.

The Model Comparison report appears (Figure 5.4).

**Note:** Your results differ due to the random assignment of training and validation rows.

**Figure 5.4** Model Comparison Report

![Model Comparison Report]

The rows in the training set were used to build the models, so the RSquare statistics for Validation=0 might be artificially inflated. In this case, the statistics are not representative of the models’ future predictive ability. This is especially true for the bootstrap forest model.

Compare the models using the statistics for Validation=1. In this case, the bootstrap forest model predicts better than the regression model.
Launch the Model Comparison Platform

To launch the Model Comparison platform, select Analyze > Modeling > Model Comparison.

Figure 5.5 The Model Comparison Launch Window

**Y, Predictors** The columns that contain the predictions for the models that you want to compare. They can be either formula columns or just data columns.

For a categorical response with $k$ levels, the model fitting platform saves $k$ columns to the data table, each predicting the probability for a level. All $k$ columns need to be specified as **Y, Predictors**.

If you do not specify the **Y, Predictors** columns in the launch window, JMP finds the prediction formula columns in the data table. Prediction formula columns have either the Predicting or Response Probability column property.

**Group** The column that separates the data into groups, which are fit separately.

The other role buttons are common among JMP platforms. See the Using JMP book for details.

The Model Comparison Report

Figure 5.6 shows an example of the initial Model Comparison report for a continuous response.
Figure 5.6 Initial Model Comparison Report

The Predictors report shows all responses and all models being compared for each response. The fitting platform that created the predictor column is also listed.

The Measures of Fit report shows measures of fit for each model. The columns are different for continuous and categorical responses.

Measures of Fit for Continuous Responses

- **RSquare**: The \( r \)-squared statistic. In data tables that contain no missing values, the \( r \)-squared statistics in the Model Comparison report and original models match. However, if there are any missing values, the \( r \)-squared statistics differ.

- **RASE**: The root average squared error, the same value as RMSE except that RMSE adjusts for degrees of freedom.

- **AAE**: The average absolute error.

- **Freq**: The column that contains frequency counts for each row.

Measures of Fit for Categorical Responses

- **Entropy RSquare**: One minus the ratio of the \(-\log\)-likelihoods from the fitted model and the constant probability model. It ranges from 0 to 1.

- **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. The maximum value is 1 for a perfect model. A value of 0 indicates that the model is no better than a constant model.

- **Mean -Log p**: The average of \(-\log(p)\), where \( p \) is the fitted probability associated with the event that occurred.

- **RMSE**: The root mean square error, adjusted for degrees of freedom. For categorical responses, 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. For categorical responses, 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.

N   The number of observations.

Related Information

“Training and Validation Measures of Fit” on page 78 in the “Neural Networks” chapter provides more information about measures of fit for categorical responses.

Model Comparison Platform Options

Some options in the Model Comparison red triangle menu depend on your data.

Continuous and Categorical Responses

Model Averaging  Makes a new column of the arithmetic mean of the predicted values (for continuous responses) or the predicted probabilities (for categorical responses).

Continuous Responses

Plot Actual by Predicted  Shows a scatterplot of the actual versus the predicted values. The plots for the different models are overlaid.

Plot Residual by Row  Shows a plot of the residuals by row number. The plots for the different models are overlaid.

Profiler  Shows a profiler for each response based on prediction formula columns in your data. The profilers have a row for each model being compared.

Categorical Responses

ROC Curve  Shows ROC curves for each level of the response variable. The curves for the different models are overlaid.

AUC Comparison  Provides a comparison of the area under the ROC curve (AUC) from each model. The area under the curve is the indicator of the goodness of fit, with 1 being a perfect fit.

The report includes the following information:

– standard errors and confidence intervals for each AUC
Specialized Models Additional Example of Model Comparison

- standard errors, confidence intervals, and hypothesis tests for the difference between each pair of AUCs
- an overall hypothesis test for testing whether all AUCs are equal

Lift Curve  Shows lift curves for each level of the response variable. The curves for the different models are overlaid.

Cum Gains Curve  Shows cumulative gains curves for each level of the response variable. The curves for the different models are overlaid.

Profiler  Shows a profiler for each response based on prediction formula columns in your data. The profilers have a row for each model being compared.

Related Information
- “ROC Curve” on page 55 in the “Partition Models” chapter
- “Lift Curves” on page 57 in the “Partition Models” chapter

Additional Example of Model Comparison

This example uses automobile data to build a model to predict the size of the purchased car. A logistic regression model and a decision tree model are compared.

Begin by opening the Car Physical Data.jmp sample data table.

Create the Logistic Regression Model
1. Select Analyze > Fit Model.
2. Select Type and click Y.
3. Select the following columns and click Add: Country, Weight, Turning Cycle, Displacement, and Horsepower.
4. Click Run.
   The Nominal Logistic Fit report appears.
5. Save the prediction formulas to columns by selecting Save Probability Formula from the Nominal Logistic red triangle menu.

Create the Decision Tree Model and Save the Prediction Formula to a Column
1. Select Analyze > Modeling > Partition.
2. Select Type and click Y, Response.
4. Make sure that Decision Tree is selected in the Method list.
Model Comparison
Additional Example of Model Comparison

5. Click OK.
   The Partition report appears.
6. Click Split 10 times.
7. Save the prediction formulas to columns by selecting Save Columns > Save Prediction Formula from the Partition red triangle menu.

Compare the Models
1. Select Analyze > Modeling > Model Comparison.
2. Select all columns that begin with Prob and click Y, Predictors.
3. Click OK.
   The Model Comparison report appears (Figure 5.7).

Figure 5.7 Initial Model Comparison Report

<table>
<thead>
<tr>
<th>Creator</th>
<th>Entropy RSquare</th>
<th>Generalized RSquare</th>
<th>Mean Log p</th>
<th>RMSE</th>
<th>Abs Dev</th>
<th>Rate</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit-Nominal Logistic</td>
<td>0.5821</td>
<td>0.6036</td>
<td>0.6557</td>
<td>0.4780</td>
<td>0.3960</td>
<td>0.2109</td>
<td>115</td>
</tr>
<tr>
<td>Partition</td>
<td>0.6248</td>
<td>0.6006</td>
<td>0.5975</td>
<td>0.4575</td>
<td>0.3966</td>
<td>0.2759</td>
<td>115</td>
</tr>
</tbody>
</table>

The report shows that the Partition model has slightly higher values for Entropy RSquare and Generalized RSquare and a slightly lower value for Misclassification Rate.

4. Select ROC Curve from the Model Comparison red triangle menu.
   ROC curves appear for each Type, one of which is shown in Figure 5.8.

Figure 5.8 ROC Curve for Medium
Examining all the ROC curves, you see that the two models are similar in their predictive ability.

5. Select **AUC Comparison** from the Model Comparison red triangle menu.

AUC Comparison reports appear for each **Type**, one of which is shown in Figure 5.9.

**Figure 5.9  AUC Comparison for Medium**

<table>
<thead>
<tr>
<th>Predictor</th>
<th>AUC</th>
<th>Std Error</th>
<th>Lower 95%</th>
<th>Upper 95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob(Medium)</td>
<td>0.8314</td>
<td>0.0218</td>
<td>0.8742</td>
<td>0.8937</td>
</tr>
<tr>
<td>Prob(Type=Medium)</td>
<td>0.9035</td>
<td>0.0255</td>
<td>0.9440</td>
<td>0.9477</td>
</tr>
</tbody>
</table>

The report shows results for a hypothesis test for the difference between the AUC values (area under the ROC curve). Examining the results, you see there is no statistical difference between the values for any level of **Type**.

You conclude that there is no large difference between the predictive abilities of the two models for the following reasons:

- The R Square values and the ROC curves are similar.
- There is no statistically significant difference between AUC values.
Model Comparison
Additional Example of Model Comparison

Chapter 5
Specialized Models
In many situations, especially in the physical and biological sciences, well-known nonlinear equations describe the relationship between variables. For example, bioassay experiments are conducted by pharmacology researchers to understand how the strength of the response to a drug changes as a function of drug concentration. The Logistic family of curves often accurately describes how the response strength is related to drug concentration. Another example is exponential growth curves, which can predict the size of a population over time.

The Nonlinear platform’s Fit Curve personality provides predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models. Fit Curve also enables you to compare different groups or subjects using a variety of analytical and graphical techniques.

You might prefer to create your own nonlinear models, which include a model formula and initial parameter estimates. See “Nonlinear Regression with Custom Models” chapter on page 115 for details.

Figure 6.1 Example of Nonlinear Fit in the Fit Curve Personality
Contents

Introduction to the Nonlinear Fit Curve Personality .............................................. 99
Example Using the Fit Curve Personality ......................................................... 99
Launch the Nonlinear Platform ........................................................................ 101
The Fit Curve Report ....................................................................................... 103
Fit Curve Options ............................................................................................ 106
  Model Formulas ......................................................................................... 106
  Test Parallelism ....................................................................................... 110
  Compare Parameter Estimates ................................................................. 112
  Equivalence Test ....................................................................................... 112
Introduction to the Nonlinear Fit Curve Personality

The Nonlinear platform fits models that are nonlinear in the parameters. The Fit Curve personality of the platform has a variety of predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models.

Specifying a grouping variable lets you estimate separate model parameters for each level of the grouping variable. The fitted models and estimated parameters can be compared across the levels of the grouping variable.

The Nonlinear platform is a good choice for models that are nonlinear in the parameters. Some models are linear in the parameters (for example, a quadratic or other polynomial) or can be transformed to be such (for example, when you use a log transformation of x). The Fit Model or Fit Y by X platforms are more appropriate in these situations. For more information about Fit Model, see Fitting Linear Models. For more information about Fit Y by X, see the Basic Analysis book.

Example Using the Fit Curve Personality

You have data on toxicity levels of a drug. You want to build a model for toxicity as a function of the concentration of the drug. There are four formulations of the drug to compare. Follow the steps below:

1. Open the Bioassay.jmp sample data table.
2. Select Analyze > Modeling > Nonlinear.
3. Assign Toxicity to the Y, Response role.
4. Assign log Conc to the X, Predictor Formula role.
5. Assign Formulation to the Group role.
6. Click OK.

The Fit Curve Report appears. The Plot report contains an overlaid plot of the fitted model of each formulation.
7. Select Sigmoid Curves > Logistic Curves > Fit Logistic 4P from the Fit Curve red triangle menu.

The Logistic 4P report appears (Figure 6.3). There is also a separate plot for each formulation. The plot of the fitted curves suggests that formulation B might be different, because the test B curve starts to rise sooner than the others. This rise is a consequence of the inflection point parameter.

8. Select Compare Parameter Estimates from the Logistic 4P red triangle menu.
A portion of the Parameter Comparison report is shown in Figure 6.4.

**Figure 6.4** Parameter Comparison Report

Notice that the Inflection Point parameter for the test B formulation is significantly lower than the average inflection point. This agrees with what we noticed in the plots of the fitted models.

**Launch the Nonlinear Platform**

To launch the Nonlinear platform, select *Analyze > Modeling > Nonlinear*. The launch window is shown in Figure 6.5.
The Nonlinear platform launch window has the following features:

**Y, Response**  Select the Y variable.

**X, Predictor Formula**  Select either the X variable or a column containing the model formula with parameters.

**Group**  Specify a grouping variable. The fitted model has separate parameters for each level of the grouping variable. The fitted model has separate parameters for each level of the grouping variable. This enables you to compare fitted models and estimated parameters across the levels of the grouping variable.

**Weight**  Specify a variable containing the weights of observations.

**Freq**  Specify a variable representing the frequency of an observation.

**Loss**  Specify a formula column giving a loss function.

**By**  Specify a variable to perform a separate analysis for every level of the variable.

The Model Library, formula, and numerical options are only for classic nonlinear analysis. For more details, see the “Nonlinear Regression with Custom Models” chapter on page 115.
The Nonlinear Fit Curve personality provides several types of built-in nonlinear models: polynomials, logistic, Gompertz, exponential growth, peak, and pharmacokinetic.

The Fit Curve report initially contains only a plot of Y versus X (Figure 6.6). If you specified a Group variable in the launch window, the report includes an overlaid plot and individual plots for each group of the fitted model (shown in Figure 6.6 on the right).

Select any of the following built-in models from the Fit Curve red triangle menu:

**Polynomials**  Fits first degree to fifth degree polynomials.

**Sigmoid Curves**  Fits Logistic and Gompertz models. These models have both upper and lower asymptotes. The Logistic 2P, 3P, and 4P models are symmetric. The Logistic 5P and both Gompertz models are not symmetric. The Logistic 2P is available only when the response is between 0 and 1.

**Exponential Growth and Decay**  Fits Exponential, Biexponential, and Mechanistic Growth models. The Exponential 2P and 3P are similar, but the 3P model has an asymptote. The Biexponential models assume there are two separate growth or decay processes. The Mechanistic Growth and Exponential 3P models always increase, but the rate of growth slows so that the model has an asymptote.

**Peak Models**  Fits Gaussian Peak and Lorentzian Peak models. These models increase up to a mode (peak) then decrease. The Gaussian Peak model is a scaled version of the Gaussian probability density function (pdf). The Lorentzian Peak model is a scaled version of the Cauchy pdf.
**Pharmacokinetic Models**  Fits the One Compartment Oral Dose model, the Two Compartment IV Bolus Dose model, and the Biexponential 4P model. Select this option to model the concentration of the drug in the body.

**Fit Michaelis Menten**  Fits the Michaelis-Menten biochemical kinetics model.

**Script**  Contains options that are available to all platforms. See *Using JMP*.

### Initial Fit Curve Reports

Before fitting a model, the Fit Curve report contains only a plot of \( Y \) versus \( X \). After fitting a model, the fitted model is added to the plot (when no grouping variable is specified on the platform launch window). The report contains the following results:

#### Model Comparison Report

The Model Comparison report shows fit statistics used for comparing multiple models. The statistics are AICc, AICc Weight, BIC, SSE, MSE, RMSE, and \( R^2 \). To create the report shown in Figure 6.7, select **Sigmoid Curves > Logistic Curves > Fit Logistic 4P** and **Sigmoid Curves > Fit Gompertz 4P** and from the Fit Curve red triangle menu.

#### Figure 6.7 Model Comparison Report

The Model Comparison platform provides additional options, such as plotting residual and actual values. See the “Model Comparison” chapter on page 85 for more information.

### Model Reports

A report is created for each fitted model. The red triangle menu for each model report provides the following options.

- **Prediction Model**  Gives the algebraic form of the prediction formula and the parameters.
- **Summary of Fit**  Gives the same fit statistics as the Model Comparison report.
- **Parameter Estimates**  Gives the estimates of the parameters, standard errors, and confidence intervals. The correlations and covariances of the estimates are given also.
- **Plot**  Gives plots of the data with the fitted model. See Figure 6.8. The plots are shown only when you select a Grouping variable on the platform launch window.
Each model report contains a red triangle menu with some or all of the following options: 

**Test Parallelism**  
Gives an analysis for testing if the fitted curves for each group have the same shape. This option is available only when a Group variable is specified on the platform launch window. This option is available for the Sigmoid models and the Fit Linear model. For details, see “Test Parallelism” on page 110.

**Area Under Curve**  
Gives the area under the fitted curve. This option is available only for the following models: One Compartment, Two Compartment, Gaussian Peak, and Lorentzian Peak. The range of integration depends on the type of model and is specified in the report. If a Grouping variable is specified on the platform launch window, an Analysis of Means is performed for comparing the estimates across groups. If the result for a group exceeds a decision limit, the result is considered different from the overall mean of AUC.

**Compare Parameter Estimates**  
Gives an analysis for testing the equality of parameters across levels of the grouping variable. This option is available only when a Group variable is specified on the platform launch window. For details, see “Compare Parameter Estimates” on page 112.

**Equivalence Test**  
Gives an analysis for testing the equivalence of models across levels of the grouping variable. This option is available only when a Group variable is specified on the platform launch window. For details, see “Equivalence Test” on page 112.

**Make Parameter Table**  
Saves the parameter estimates, standard errors, and t-ratios in a data table. This option is available only when a Group variable is specified on the platform launch window. With this data table, you can cluster groups based on parameter estimates.
**Profiler**  Shows or hides a profiler of the fitted prediction function. The derivatives are derivatives of the prediction function with respect to the $X$ variable. For more information about profilers, see the Profiler chapter in the *Profilers* book.

**Save Formulas**  Contains options for saving prediction and derivative equations as formula columns in the data table.

- **Save Prediction Formula** saves the prediction equation.
- **Save Std Error of Predicted** saves the standard error of the predicted values.
- **Save Parametric Prediction Formula** saves the prediction equation with parameters. This is helpful if you want to use the fitted model in the full, interactive personality of the Nonlinear platform.
- **Save Residual Formula** saves the residuals.
- **Save Studentized Residual Formula** saves the studentized residual formula, a standard residual that is divided by its estimated standard deviation.
- **Save First Derivative** saves the derivative of the prediction function with respect to the $X$ variable.
- **Save Std Error of First Derivative** saves the equation of the standard error of the first derivative.
- **Save Inverse Prediction Formula** saves the equation for predicting $X$ from $Y$.

**Custom Inverse Prediction**  Predicts an $X$ value for a specific $Y$ value. For more information about inverse prediction, see “Inverse Prediction with Confidence Limits” on page 254 in the “Statistical Details” appendix.

**Remove Fit**  Removes the model report, the entry from the Model Comparison report, and the fitted line from the plot.

---

**Fit Curve Options**

**Model Formulas**

Table 6.1 provides the formulas for the models on the Fit Curve red triangle menu.
Table 6.1 Fit Curve Model Formulas

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomials</td>
<td>[ \beta_0 + \sum_{i=1}^{k} \beta_i x^i ]</td>
</tr>
<tr>
<td></td>
<td>where ( k ) is the order of the polynomial. These models can also be fit using the Fit Model and Fit Y by X platforms.</td>
</tr>
<tr>
<td>Logistic 2P</td>
<td>[ \frac{1}{1 + \exp(-a(x - b))} ]</td>
</tr>
<tr>
<td></td>
<td>( a = ) Growth Rate</td>
</tr>
<tr>
<td></td>
<td>( b = ) Inflection Point</td>
</tr>
<tr>
<td>Logistic 3P</td>
<td>[ \frac{c}{1 + \exp(-a(x - b))} ]</td>
</tr>
<tr>
<td></td>
<td>( a = ) Growth Rate</td>
</tr>
<tr>
<td></td>
<td>( b = ) Inflection Point</td>
</tr>
<tr>
<td></td>
<td>( c = ) Asymptote</td>
</tr>
<tr>
<td>Logistic 4P</td>
<td>[ c + \frac{d-c}{1 + \exp(-a(x - b))} ]</td>
</tr>
<tr>
<td></td>
<td>( a = ) Growth Rate</td>
</tr>
<tr>
<td></td>
<td>( b = ) Inflection Point</td>
</tr>
<tr>
<td></td>
<td>( c = ) Lower Asymptote</td>
</tr>
<tr>
<td></td>
<td>( d = ) Upper Asymptote</td>
</tr>
<tr>
<td>Logistic 5P</td>
<td>[ c + \frac{d-c}{(1 + \exp(-a(x - b)))^f} ]</td>
</tr>
<tr>
<td></td>
<td>( a = ) Growth Rate</td>
</tr>
<tr>
<td></td>
<td>( b = ) Inflection Point</td>
</tr>
<tr>
<td></td>
<td>( c = ) Asymptote 1</td>
</tr>
<tr>
<td></td>
<td>( d = ) Asymptote 2</td>
</tr>
<tr>
<td></td>
<td>( f = ) Power</td>
</tr>
</tbody>
</table>
## Table 6.1 Fit Curve Model Formulas (Continued)

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gompertz 3P</td>
<td>( a \text{Exp}(-\text{Exp}(-b(x - c))) )</td>
</tr>
<tr>
<td></td>
<td>(a = \text{Asymptote} )</td>
</tr>
<tr>
<td></td>
<td>(b = \text{Growth Rate} )</td>
</tr>
<tr>
<td></td>
<td>(c = \text{Inflection Point} )</td>
</tr>
<tr>
<td>Gompertz 4P</td>
<td>( a + (b - a)\text{Exp}(-\text{Exp}(-c(x - d))) )</td>
</tr>
<tr>
<td></td>
<td>(a = \text{Lower Asymptote} )</td>
</tr>
<tr>
<td></td>
<td>(b = \text{Upper Asymptote} )</td>
</tr>
<tr>
<td></td>
<td>(c = \text{Growth Rate} )</td>
</tr>
<tr>
<td></td>
<td>(d = \text{Inflection Point} )</td>
</tr>
<tr>
<td>Exponential 2P</td>
<td>( a \text{Exp}(bx) )</td>
</tr>
<tr>
<td></td>
<td>(a = \text{Scale} )</td>
</tr>
<tr>
<td></td>
<td>(b = \text{Growth Rate} )</td>
</tr>
<tr>
<td>Exponential 3P</td>
<td>( a + b\text{Exp}(cx) )</td>
</tr>
<tr>
<td></td>
<td>(a = \text{Asymptote} )</td>
</tr>
<tr>
<td></td>
<td>(b = \text{Scale} )</td>
</tr>
<tr>
<td></td>
<td>(c = \text{Growth Rate} )</td>
</tr>
<tr>
<td>Biexponential 4P</td>
<td>( a\text{Exp}(-bx) + c\text{Exp}(-dx) )</td>
</tr>
<tr>
<td></td>
<td>(a = \text{Scale 1} )</td>
</tr>
<tr>
<td></td>
<td>(b = \text{Decay Rate 1} )</td>
</tr>
<tr>
<td></td>
<td>(c = \text{Scale 2} )</td>
</tr>
<tr>
<td></td>
<td>(d = \text{Decay Rate 2} )</td>
</tr>
</tbody>
</table>
### Table 6.1 Fit Curve Model Formulas (Continued)

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biexponential 5P</td>
<td>$a + b \exp(-cx) + d \exp(-fx)$</td>
</tr>
<tr>
<td></td>
<td>$a = \text{Asymptote}$</td>
</tr>
<tr>
<td></td>
<td>$b = \text{Scale 1}$</td>
</tr>
<tr>
<td></td>
<td>$c = \text{Decay Rate 1}$</td>
</tr>
<tr>
<td></td>
<td>$d = \text{Scale 2}$</td>
</tr>
<tr>
<td></td>
<td>$f = \text{Decay Rate 2}$</td>
</tr>
<tr>
<td>Mechanistic Growth</td>
<td>$a(1 - b \exp(-cx))$</td>
</tr>
<tr>
<td></td>
<td>$a = \text{Asymptote}$</td>
</tr>
<tr>
<td></td>
<td>$b = \text{Scale}$</td>
</tr>
<tr>
<td></td>
<td>$c = \text{Growth Rate}$</td>
</tr>
<tr>
<td>Gaussian Peak</td>
<td>$a \exp\left(\frac{1}{2}\left(\frac{x-b}{c}\right)^2\right)$</td>
</tr>
<tr>
<td></td>
<td>$a = \text{Peak Value}$</td>
</tr>
<tr>
<td></td>
<td>$b = \text{Critical Point}$</td>
</tr>
<tr>
<td></td>
<td>$c = \text{Growth Rate}$</td>
</tr>
<tr>
<td>Lorentzian Peak</td>
<td>$\frac{ab^2}{(x-c)^2 + b^2}$</td>
</tr>
<tr>
<td></td>
<td>$a = \text{Peak Value}$</td>
</tr>
<tr>
<td></td>
<td>$b = \text{Growth Rate}$</td>
</tr>
<tr>
<td></td>
<td>$c = \text{Critical Point}$</td>
</tr>
<tr>
<td>One Compartment Oral Dose</td>
<td>$\frac{abc}{c-b}(\exp(-bx) - \exp(-cx))$</td>
</tr>
<tr>
<td></td>
<td>$a = \text{Area Under Curve}$</td>
</tr>
<tr>
<td></td>
<td>$b = \text{Elimination Rate}$</td>
</tr>
<tr>
<td></td>
<td>$c = \text{Absorption Rate}$</td>
</tr>
</tbody>
</table>
Chapter 6
Specialized Models

Nonlinear Regression with Built-In Models

Fit Curve Options

**Test Parallelism**

The Test Parallelism option provides an analysis for testing if the fitted models between groups have the same shape (Figure 6.9). Select **Test Parallelism** from the fitted model’s red triangle menu to add the report.

---

**Table 6.1** Fit Curve Model Formulas  *(Continued)*

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two Compartment IV Bolus Dose</td>
<td>[ \frac{a}{\alpha - \beta}(-\alpha x - (-\beta x)) ] [ \alpha = \frac{1}{2} \left(b + c + d + \sqrt{(b + c + d)^2 - 4bd} \right) ] [ \beta = \frac{1}{2} \left(b + c + d - \sqrt{(b + c + d)^2 - 4bd} \right) ]</td>
</tr>
<tr>
<td></td>
<td>[ a = \text{Initial Concentration} ] [ b = \text{Transfer Rate In} ] [ c = \text{Transfer Rate Out} ] [ d = \text{Elimination Rate} ]</td>
</tr>
<tr>
<td>Michaelis Menten</td>
<td>[ \frac{ax}{b + x} ] [ a = \text{Max Reaction Rate} ] [ b = \text{Inverse Affinity} ]</td>
</tr>
</tbody>
</table>

Figure 6.9 Parallelism Test

<table>
<thead>
<tr>
<th>Parallelism Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Results</td>
</tr>
<tr>
<td>Parallelism F Test</td>
</tr>
<tr>
<td>Fit SSE</td>
</tr>
<tr>
<td>0.0904</td>
</tr>
<tr>
<td>Parallelism Chi-Square Test</td>
</tr>
<tr>
<td>ChiSquare</td>
</tr>
<tr>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parallel Fit Parameter Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Growth Rate</td>
</tr>
<tr>
<td>Lower Asymptote</td>
</tr>
<tr>
<td>Upper Asymptote</td>
</tr>
<tr>
<td>Inflection Point standard</td>
</tr>
<tr>
<td>Inflection Point test A</td>
</tr>
<tr>
<td>Inflection Point test B</td>
</tr>
<tr>
<td>Inflection Point test C</td>
</tr>
</tbody>
</table>

The report gives the following results:

**Test Results**  Gives the results of an F Test and a Chi-Square Test for parallelism. The F Test compares the error sums-of-squares for a full and a reduced model. The full model gives each group different parameters. The reduced model forces the groups to share every parameter except for the inflection point.

**Parallel Fit Parameter Estimates**  Gives the parameter estimates under the reduced model (same parameters, except for inflection point). A plot of the fitted curves under the reduced model is provided.

**Relative Potencies**  Gives the relative potencies. There is a report for each level of the grouping variable. The relative potency is \( \exp(\text{EC}_{50}) \), where \( \text{EC}_{50} \) is the point at which the middle response value is obtained. For the Logistic 2P, 3P, and 4P, the relative potency is \( \exp(\text{inflection point parameter}) \).
Nonlinear Regression with Built-In Models

Chapter 6
Specialized Models

Compare Parameter Estimates

The Compare Parameter Estimates report gives results for testing the equality of parameters across the levels of the grouping variable. There is an Analysis of Means (ANOM) report for each parameter, which tests whether the parameters are equal to an overall mean. If the result for a parameter exceeds the decision limits, then the parameter is different from the overall mean. Figure 6.10 shows the ANOM report for growth rate estimates. Select Compare Parameter Estimates from the fitted model’s red triangle menu to add the report.

Figure 6.10 Parameter Comparison for Growth Rate Estimates

The Analysis of Means red triangle menu has the following options:

Set Alpha Level  Sets the alpha level for the test.

Show Summary Report  Shows or hides a report containing the parameter estimates, the decision limits, and whether the parameter exceeded the limits.

Display Options  Contains options for showing or hiding decision limits, shading, and the center line. Also contains options for changing the appearance of the points.

For more information on the Analysis of Means report, see the Basic Analysis book.

Equivalence Test

The Equivalence Test report gives an analysis for testing the equivalence of models across levels of the grouping variable (Figure 6.11). After selecting the option, you specify the level of the grouping variable that you want to test against every other level. There is a report for every level versus the chosen level. Select Equivalence Test from the fitted model’s red triangle menu to add the report.
The equality of the parameters is tested by analyzing the ratio of the parameters. The default decision lines are placed at ratio values of 0.8 and 1.25, representing a 25% difference.

If all of the confidence intervals are inside the decision lines, then the two groups are practically equal. If a single interval falls outside the lines (as shown in Figure 6.11), then you cannot conclude that the groups are equal.

Figure 6.11 Equivalence Test

The Equivalence red triangle menu has the following options:

- **Set Alpha Level**  Sets the alpha level for the test. The default value is 0.05.

- **Set Decision Lines**  Changes the decision limits for the ratio. The default values are set at 0.8 and 1.25, representing a 25% difference.

- **Show Summary Report**  Shows or hides a report containing the parameter estimates, the decision limits, and whether the parameter exceeded the limits.

- **Display Options**  Contains options for showing or hiding decision limits, shading, and the center line. Also contains options for changing the appearance of the points. For additional formatting options, right-click the graph and select **Customize**.
The Nonlinear platform is a good choice for models that are nonlinear in the parameters. This chapter focuses on custom nonlinear models, which include a model formula and parameters to be estimated. Use the default least squares loss function or a custom loss function to fit models. The platform minimizes the sum of the loss function across the observations.

The Nonlinear platform also provides predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models. See "Nonlinear Regression with Built-In Models" chapter on page 97 for more information.

**Note:** Some models are linear in the parameters (for example, a quadratic or other polynomial) or can be transformed to be such (for example, when you use a log transformation of x). The Fit Model or Fit Y by X platforms are more appropriate in these situations. For more information about these platforms, see the *Fitting Linear Models* and *Basic Analysis* books.
## Contents

- Example of Fitting a Custom Model ....................................................... 117
- Launch the Nonlinear Platform ............................................................... 118
- The Nonlinear Fit Report ........................................................................ 120
- Nonlinear Platform Options .................................................................... 124
- Create a Formula Using the Model Library .............................................. 127
- Additional Examples .................................................................................. 131
  - Example of Maximum Likelihood: Logistic Regression ......................... 131
  - Example of a Probit Model with Binomial Errors: Numerical Derivatives  135
  - Example of a Poisson Loss Function ..................................................... 136
- Statistical Details ..................................................................................... 138
  - Profile Likelihood Confidence Limits .................................................... 138
  - How Custom Loss Functions Work ......................................................... 139
  - Notes Concerning Derivatives ............................................................... 140
- Notes on Effective Nonlinear Modeling ................................................... 142
Example of Fitting a Custom Model

To fit a custom model, you must first create a model column with initial parameter estimates. This method does require a few more steps than fitting a built-in model, but it does allow any nonlinear model to be fit. Also, you can provide a custom loss function, and specify several other options for the fitting process.

This section provides an example of creating the formula column for a model, and fitting the model in the Nonlinear platform. The data is in the US Population.jmp data table. The response variable is the population (in millions) of the United States and the predictor is the year.

1. Open the US Population.jmp sample data table.
2. Create a new column called Model.
3. Right-click the Model column and select Column Properties > Formula.
   The Formula Editor appears.
4. Above the list of columns on the left, select Parameters.
5. Select New Parameter.
6. Use the default name of b0.
7. Type 4 for Value. This is the initial estimate of the parameter.
8. Click OK.
10. Keep the default name and enter 0.02 for Value.
11. Click OK.
12. Enter the model formula using the Formula Editor functions, the column year, and the parameters. Figure 7.2 shows the completed model.

Figure 7.2 Completed Model Formula

\[ b_0 \exp (b_1 \times (year - 1780)) \]

13. Click OK.
15. Assign Model to the X, Predictor Formula role.
16. Assign pop to the Y, Response role.
17. Click OK.
18. Click Go on the Control Panel to fit the model.
   A portion of the report is shown in Figure 7.3.
The final parameter estimates are shown in the Solution report, along with other fit statistics. The fitted model is shown on the plot.

### Parameters for Models with a Grouping Variable

In the formula editor, when you add a parameter, note the check box for **Expand Into Categories, selecting column**. This option 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 enables 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 that you gave the parameter. When you use this parameter in the formula, a Match expression is inserted, containing a separate parameter for each level of the grouping variable.

### Launch the Nonlinear Platform

To launch the Nonlinear platform, select **Analyze > Modeling > Nonlinear**. The launch window is shown in Figure 7.4.
Figure 7.4 Nonlinear Platform Launch Window

The Nonlinear platform launch window has the following features:

**Y, Response**  Select the Y variable.

**X, Predictor Formula**  Select either the X variable or a column containing the model formula with parameters.

**Group**  Specify a grouping variable. The fitted model has separate parameters for each level of the grouping variable. This enables you to compare fitted models and estimated parameters across the levels of the grouping variable.

**Weight**  Specify a variable containing the weights of observations.

**Freq**  Specify a variable representing the frequency of an observation.

**Loss**  Specify a formula column giving a loss function.

**By**  Specify a variable to perform a separate analysis for every level of the variable.

**Model Library**  Launches the Model Library tool, which helps you choose initial values to create a formula column. See “Create a Formula Using the Model Library” on page 127.

**Second Derivatives**  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 useful only if the residuals are unusually large or if you specify a custom loss function and your model is not linear in its parameters. This option is used only when a formula column is provided in the X, Predictor Formula role.
The Nonlinear Fit Report

The initial Nonlinear Fit report includes the following items, shown in Figure 7.5.

**Control Panel**  Provides options for controlling the fitting process.

- **Go** starts the fitting process.
- **Stop** stops the fitting process.
- **Step** proceeds through the fitting process one iteration at a time.
- **Reset** resets the editable values into the formula, reset the iteration values, and calculate the SSE at these new values.
- **Criterion** shows iteration measures from the fitting process and the **Current** values.
- **Stop Limit** sets limits on the Criterion.

**Plot**  Shows a plot of the X and Y variables for models with only one X variable. The model based on the current values is shown on the plot. To change the current values of the parameters, use the sliders or edit boxes beneath the plot.
After you click **Go** to fit a model, the report includes the following additional items, shown in Figure 7.6.

**Save Estimates**  Saves the current parameter values to the parameters in the formula column.

**Confidence Limits**  Computes confidence intervals for all parameters. The intervals are profile-likelihood confidence intervals, and are shown in the Solution report. The confidence limit computations involve a new set of iterations for each limit of each parameter, and the iterations often do not find the limits successfully. The **Edit Alpha** and **Convergence Criterion** options are for the confidence interval computations. For details about the **Goal SSE for CL**, see “Profile Likelihood Confidence Limits” on page 138.
Solution  Shows the parameters estimates and other statistics.

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.

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 might 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 confidence limits for the parameters. They are missing until you click the Confidence Limits on the Control Panel. For more details about the confidence intervals, see “Profile Likelihood Confidence Limits” on page 138.

Excluded Data is a report showing fit statistics for excluded rows. This is useful for validating the model on observations not used to fit the model. You can use this feature in conjunction with the Remember Solution option to change the exclusions, and get a new report reflecting the different exclusions.

Correlation of Estimates  Displays the correlations between the parameter estimates.
Chapter 7
Specialized Models
Nonlinear Regression with Custom Models

Figure 7.6 Fitted Model Report

![Fitted Model Report Diagram]

- **Parameter Estimate Range**: 
  - \( b_0 \): 15.95 to 297.97
  - \( b_1 \): 0.014 to 0.055

- **Solution**:
  - SSE: 2245.57
  - DFE: 20
  - MSE: 112.27
  - RMSE: 10.50
  - Parameters:
    - \( \hat{b}_0 \): 15.95
    - \( \hat{b}_1 \): 0.014

- **Correlation of Estimates**
Nonlinear Platform Options

The Nonlinear Fit red triangle menu has the following options:

**Parameter Bounds**  Sets bounds on the parameters. When the option is selected, editable boxes appear in the Control Panel. Unbounded parameters are signified by leaving the field blank.

**Plot**  Shows or hides a plot of the X and Y variables for models with only one X variable. The model shown on the plot is based on the current values of the parameters. To change the current values of the parameters, use the sliders or edit boxes beneath the plot. If you specify a Group variable at launch, then a curve shows for each group.

**Iteration Options**  Specifies options for the fitting algorithm.

  - **Iteration Log**  records each step of the fitting process in a new window.
  - **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) is 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. See “Notes Concerning Derivatives” on page 140, 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.
Specialized Models

Profilers  Provides various profilers for viewing response surfaces.

  Profiler brings up the Prediction 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.

  Contour Profiler brings up the Contour Profiler. The Contour profiler lets you see two-dimensional contours as well as three dimensional mesh plots.

  Surface Profiler creates a three-dimensional surface plot. This option is available only for models with two or more X variables.

  Parameter Profiler brings up the Prediction Profiler and profiles the SSE or loss as a function of the parameters.

  Parameter Contour Profiler brings up the Contour Profiler and contours the SSE or loss as a function of the parameters.

  Parameter Surface Profiler creates a three-dimensional surface plot and profiles the SSE or loss as a function of the parameters. This option is available only for models with two or more parameters.

SSE Grid  Create a grid of values around the solution estimates and compute the error sum of squares for each value. The solution estimates should have the minimum SSE. When the option is selected, the **Specify Grid for Output** report is shown with these features:

  Parameter lists the parameters in the model.

  Min displays the minimum parameter values 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 the grid of points in a new table. A highlighted row marks the solution estimate row if the solution is in the table.

Revert to Original Parameters  Resets the platform to the original parameter values (the values given in the formula column parameters).

Remember Solution  Creates a report called Remembered Models, which contains the current parameter estimates and summary statistics. Results of multiple models can be remembered and compared. This is useful if you want to compare models based on different parameter restrictions, or models fit using different options. Click on the radio button for a particular model to display that model in the Plot and the parameter estimates in the Control Panel.
Custom Estimate  Gives an estimate of a function of the parameters. You provide an expression involving only parameters. JMP calculates the expression using the current parameter estimates, and also calculates a standard error of the expression using a first-order Taylor series approximation.

Custom Inverse Prediction  Estimates the X value for a given Y value. It also calculates a standard error for the estimated X. JMP must be able to invert the model. 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  Saves asymptotic confidence limits for the model prediction. This is the confidence interval for the average Y at a given X value.

Save Indiv Confid Limits  Saves asymptotic confidence limits for an individual prediction. This is the confidence interval for an individual Y value at a given X value.

Save Formulas  Gives options for saving model results to data table columns:

- **Save Prediction Formula** saves the prediction formula with the current parameter estimates.
- **Save Std Error of Predicted** saves the standard error for a model prediction. This is the standard error for predicting the average Y for a given X. The formula is of the form \( \sqrt{\text{VecQuadratic} \left( \text{matrix1}, \text{vector1} \right)} \). \text{matrix1} is the covariance matrix associated with the parameter estimates, and \text{vector1} is a composition of the partial derivatives of the model with respect to each parameter.
- **Save Std Error of Individual** saves the standard error for an individual prediction. This is the standard error for predicting an individual Y value for a given X value. The formula is of the form \( \sqrt{\text{VecQuadratic} \left( \text{matrix1}, \text{vector1} \right) + \text{mse}} \). \text{matrix1} is the covariance matrix associated with the parameter estimates, \text{vector1} is a composition of the partial derivatives of the model with respect to each parameter, and \text{mse} is the estimate of error variance.
- **Save Residual Formula** saves the formula for computing the residuals.
- **Save Pred Confid Limit Formula** saves the formula to calculate the confidence interval for a model prediction. This is a confidence interval for the average Y for a given X.
- **Save Indiv Confid Limit Formula** saves the formula to calculate the confidence interval for an individual prediction. This is a confidence interval for an individual Y for a given X.
- **Save Inverse Prediction Formula** saves formulas for the inverse of the model, the standard error of an inverse prediction, and the standard error of an individual inverse prediction.
- **Save Specific Solving Formula** is equivalent to Save Inverse Prediction Formula in simple cases. 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 enables you to select the variable to solve for. 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** Shows the prediction model or the loss function at the top of the report.

---

### Create a Formula Using the Model Library

The Model Library can assist you in creating the formula column with parameters and initial values. Click the **Model Library** button on the Nonlinear launch window to open the library. Select a model in the list to see its formula in the **Formula** box (Figure 7.7).
Figure 7.7 Nonlinear Model Library Dialog

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 7.8.
**Figure 7.8** Example Graph in Model Library

![Example Graph in Model Library](image)

Use the slider bars to set starting values for parameters:
- \( \theta_1 = 2 \)
- \( \theta_2 = 1.5 \)
- \( \theta_3 = 2 \)

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 7.9 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 7.9** Select Roles

![Select Roles Dialog](image)

For most models, the starting values are constants. Showing points enables 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 7.10.
Create a Formula Using the Model Library

Specialized Models

Figure 7.10 Show Points

Clicking Make Formula at this point (after using Show Points) creates a new data table column named after the model that you chose from the Model Library. This column has the formula as a function of the latest parameter starting values.

Note: If you click Make Formula before using the Show Graph or Show Points buttons, you are asked to provide the X and Y roles, and an optional Group role. See Figure 7.9. After that, you 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.

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.

Customize the Nonlinear Model Library

The Model Library is created by a built-in script named NonlinLib.jsl, located in the Resources/Builtin folder in the folder that contains JMP (Windows) or in the Application Package (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 Listofmodelist#. 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

```javascript
{//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.

**Additional Examples**

This section provides several examples of the broad usefulness of the Nonlinear platform.

**Example of Maximum Likelihood: Logistic Regression**

In this example, we show several variations of minimizing a loss function. 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 ones for equal-sized samples of x values. The Model Y column has the linear model, and the Loss column has the loss function. In this example, the loss function is the negative log-likelihood for each observation, or the negative log of the probability of getting the observed response.
Run the model by following the steps below:

1. Open Logistic w Loss.jmp from the Nonlinear Examples sample data folder.
2. Select Analyze > Modeling > Nonlinear.
3. Assign Model Y to the X, Predictor Formula role.
4. Assign Loss to the Loss role.

**Figure 7.11 Nonlinear Launch Window**

5. Click OK.

The Nonlinear Fit Control Panel appears.

**Figure 7.12 Nonlinear Fit Control Panel**
6. Click Go.

The parameter estimates are shown in the Solution report. See Figure 7.13.

**Figure 7.13** Solution Report

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>ApproxStdErr</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>-5.51453</td>
<td>2.43720</td>
</tr>
<tr>
<td>b1</td>
<td>0.834553</td>
<td>0.13551</td>
</tr>
</tbody>
</table>

The Loss value in the Solution report is the negative log-likelihood evaluated at the parameter estimates.

The same problem can be handled differently by defining a model column formula that absorbs the logistic function. Also, define a loss function that uses the model to form the probability for a categorical response level. **Model2 Y** holds the model, and the loss function is **Loss2**.

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. Follow these steps to fit the model:

1. Display the Logistic w Loss.jmp sample data table again.
2. Select Analyze > Modeling > Nonlinear.
3. Assign **Model2 Y** to the X, Predictor Formula role.
4. Assign **Loss2** to the Loss role.
5. Select the Second Derivatives option.
6. Click OK.

The Nonlinear Fit Control Panel appears.

7. Type 1000 for the Iteration Stop Limit.

8. Click Go.

The Solution report shows the same Loss and parameter estimates as before.
Figure 7.16 The Standard Error is different

Example of a Probit Model with Binomial Errors: Numerical Derivatives

The Ingots2.jmp sample data table includes the numbers of ingots tested for readiness after different treatments of heating and soaking times. 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 P column contains the model formula:

$\text{Normal Distribution}(b_0 + b_1 \cdot \text{Heat} + b_2 \cdot \text{Soak})$

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

$-\left[N\cdot \log(p) + (N - N\cdot \log(1 - p))\right]$

Follow these steps to fit the model:

1. Select Analyze > Modeling > Nonlinear.
2. Assign P to the X, Predictor Formula role,
3. Assign Loss to the Loss role.
4. Select the Numeric Derivatives Only option.
5. Click OK.
6. Click Go.

The platform used the Numerical SR1 method to obtain the parameter estimates shown in Figure 7.17.
Example of a Poisson Loss Function

A Poisson distribution is often used to model count data.

\[ P(Y = n) = \frac{e^{-\mu} \mu^n}{n!}, \quad n = 0, 1, 2, \ldots \]

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 table. The model formula is in the model column, and the loss function (or –log-likelihood) is in the Poisson column. To fit the model, follow the steps below:

1. Select **Analyze > Modeling > Nonlinear**.
2. Assign model to the **X, Predictor Formula** role.
3. Assign Poisson to the **Loss** role.
4. Click **OK**.
5. Set the **Current Value** (initial value) for b0 to 1, and the other parameters to 0 (Figure 7.18).
6. Click Go.
7. Click the Confidence Limits button.

The Solution report is shown in Figure 7.19. The results include the parameter estimates and confidence intervals, and other summary statistics.

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

\[
\frac{\partial^2 L}{\partial \beta_j \partial \beta_k} = \sum_{i=1}^{n} \frac{\partial^2 p(f(\bar{\beta}))}{\partial \beta_j \partial \beta_k} \frac{\partial f}{\partial \beta_j} \frac{\partial f}{\partial \beta_k}
\]

The second term is probably small if \( \rho \) is the squared residual because the sum of residuals is small. The term is 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 Second Derivative option 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.

Statistical Details

This section provides statistical details and other notes concerning the Nonlinear platform.

Profile Likelihood Confidence Limits

The upper and lower confidence limits for the parameters are based on a search for the value of each parameter after minimizing with respect to the other parameters. The search looks for values that produce an 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).

Profile confidence limits all start with a goal SSE. This 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 increased until the SSE reaches the goal SSE. 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. 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.

Figure 7.20 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.
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.

**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 nonzero 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:
Nonlinear Regression with Custom Models
Statistical Details Specialized Models

If \( f \) 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

\[
\frac{\partial L}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \rho(f(\beta))}{\partial f} \frac{\partial f}{\partial \beta_j}
\]

\[
\frac{\partial^2 L}{\partial \beta_j \partial \beta_k} = \sum_{i=1}^{n} \left[ \frac{\partial^2 \rho(f(\beta))}{\partial f^2} \frac{\partial f}{\partial \beta_j} \frac{\partial f}{\partial \beta_k} + \frac{\partial \rho(f(\beta))}{\partial f} \frac{\partial^2 f}{\partial \beta_j \partial \beta_k} \right]
\]

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

\[
\frac{\partial^2 L}{\partial \beta_j \partial \beta_k} \equiv \sum_{i=1}^{n} \frac{\partial^2 \rho(f(\beta))}{\partial f^2} \frac{\partial f}{\partial \beta_j} \frac{\partial f}{\partial \beta_k}
\]

The second term is probably small if \( \rho \) is the squared residual because the sum of residuals is small. The term is 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 Second Derivative option 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.

**Notes Concerning Derivatives**

The nonlinear platform takes symbolic derivatives for formulas with most common operations. This section shows what type of derivative expressions result.

If you open the Negative Exponential.jmp nonlinear sample data example, the actual formula looks something like this:

\[
\text{Parameter}((b0=0.5, b1=0.5, b0^*(1-\exp(-b1*X)))
\]

The Parameter block in the formula is hidden if you use the formula editor. 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. Then it maps the references to them to the place where they are 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 needs for derivative calculations. The assignments are made to names like T#1, T#2, and so on.
- When the prediction model needs additional subexpressions evaluated, it uses the $\text{First}$ function, which returns the value of the first argument expression, and also evaluates the other arguments. In this case additional assignments are 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 $1-(\text{Exp}(-b1*X))$. The derivative with respect to $b1$ is T#3*b0, which is $-(-1*\text{Exp}(-b1*X)*X)*b0$ if you substitute in the assignments above. Although many optimizations are made, it does not always combine the operations optimally. You can see this by the expression for T#3, which does not remove a double negation.

If you ask for second derivatives, then you get a list of $\frac{(m(m+1))}{2}$ 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 does not know how to take the analytic derivative of a function, then it takes numerical derivatives, using the $\text{NumDeriv}$ function. If this occurs, the platform shows the delta that it used to evaluate the change in the function with respect to a delta change in the arguments. You might 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. 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

\[ a_1 + b_1 x + c_1 x^2 \]

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.

\[
\begin{align*}
a_1 &= a_2 - b_2 \bar{x} + c_2 \bar{x} \\
b_1 &= b_2 - 2c_2 \bar{x} \\
c_1 &= c_2
\end{align*}
\]

If the number of iterations still goes to the maximum, increase the maximum number of iterations or relax one of the convergence criteria.

There is really no one omnibus optimization method that works well on all problems. JMP has options like **Newton**, **QuasiNewton BFGS**, **QuasiNewton SR1**, and **Numeric Derivatives Only** to expand the range of problems that are solvable by the Nonlinear Platform.

If the default settings are unable to converge to the solution for a particular problem, using various combinations of these settings to 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 might be too slow for anything but small data sets.
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 8.1 Example of a Gaussian Process Prediction Surface
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 8.2 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 8.3 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 $X_1$ is the integrated total variation due to $X_1$ alone. In this case, we see that 37.6% of the variation in $Y$ is due to $X_1$. 
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 \( \sigma^2 R \). The \( R \) 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 \( R \).

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 \( \sigma^2 R \). The \( R \) matrix is composed of elements

\[
r_{ij} = \prod_k \rho(d; \theta_k)
\]

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 8.4.

**Figure 8.4** 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 *Profilers* book.

- **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** contains options that are available to all platforms. See *Using JMP*.

In Figure 8.5, we use the saved prediction formula to compare the prediction to the actual data points.
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 8.6.

Figure 8.6 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, \( \log_{10} R_w \). The main effect explains 87.5% of the variation, with 90.5% explained when all second-order interactions are included.

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 \( \log_{10} R_w \) is a highly involved participant in \( Y \)'s variation.
Gaussian Process

Borehole Hypercube Example

Chapter 8
Specialized Models
The Time Series platform lets you explore, analyze, and forecast univariate time series. A time series is a set \( y_1, y_2, \ldots, y_N \) 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.
## Contents

- **Launch the Platform** .................................................. 155
- **Time Series Commands** .............................................. 156
  - Graph ........................................................................... 157
  - Autocorrelation and Partial Autocorrelation ...................... 157
  - Variogram .................................................................... 158
  - AR Coefficients .......................................................... 159
  - Spectral Density .......................................................... 159
  - Save Spectral Density .................................................... 160
  - Number of Forecast Periods .......................................... 161
  - Difference .................................................................... 161
- **Modeling Reports** ..................................................... 162
  - Model Comparison Table .............................................. 162
  - Model Summary Table .................................................. 164
  - Parameter Estimates Table ........................................... 166
  - Forecast Plot ............................................................. 167
  - Residuals ...................................................................... 167
  - Iteration History .......................................................... 167
  - Model Report Options .................................................. 168
- **ARIMA Model** ............................................................ 168
- **Seasonal ARIMA** ....................................................... 170
- **ARIMA Model Group** .................................................. 170
- **Transfer Functions** .................................................... 171
  - Report and Menu Structure .......................................... 171
  - Diagnostics .................................................................. 173
  - Model Building .......................................................... 174
  - Transfer Function Model .............................................. 175
- **Model Reports** ........................................................... 177
  - Model Comparison Table .............................................. 179
  - Fitting Notes .............................................................. 179
- **Smoothing Models** ..................................................... 179
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 9.1). 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 9.1. 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.**
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 9.2). 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 9.2 Time Series Plot of Seriesg (Airline Passenger) Data

If you open *Time Series Basic Diagnostic Tables*, graphs of the autocorrelation and partial autocorrelation (Figure 9.3) 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 9.2. 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 and Partial Autocorrelation

The autocorrelation graph describes the correlation between all the pairs of points in the time series for a given separation in time (lag). Autocorrelation and partial autocorrelation graphs can help you 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.

**Tip:** The autocorrelation graph of the sample is often called the sample autocorrelation function.

Figure 9.3 Autocorrelation (left) and Partial Correlation (right) Graphs

<table>
<thead>
<tr>
<th>Lag</th>
<th>Autocorr</th>
<th>Jang.Box Q</th>
<th>p-Value</th>
<th>Lag</th>
<th>Partial</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>122.14</td>
<td>.0001*</td>
<td>0</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>0.9480</td>
<td>245.86</td>
<td>.0001*</td>
<td>2</td>
<td>-0.2294</td>
</tr>
<tr>
<td>2</td>
<td>0.6758</td>
<td>342.57</td>
<td>.0001*</td>
<td>3</td>
<td>0.0381</td>
</tr>
<tr>
<td>4</td>
<td>0.7520</td>
<td>407.71</td>
<td>.0001*</td>
<td>4</td>
<td>0.0039</td>
</tr>
<tr>
<td>5</td>
<td>0.7130</td>
<td>504.77</td>
<td>.0001*</td>
<td>5</td>
<td>0.0730</td>
</tr>
<tr>
<td>6</td>
<td>0.6817</td>
<td>575.86</td>
<td>.0001*</td>
<td>6</td>
<td>0.0077</td>
</tr>
<tr>
<td>7</td>
<td>0.6529</td>
<td>642.33</td>
<td>.0001*</td>
<td>7</td>
<td>0.1266</td>
</tr>
<tr>
<td>8</td>
<td>0.5956</td>
<td>719.64</td>
<td>.0001*</td>
<td>8</td>
<td>0.0960</td>
</tr>
<tr>
<td>9</td>
<td>0.5760</td>
<td>779.93</td>
<td>.0001*</td>
<td>9</td>
<td>0.2325</td>
</tr>
<tr>
<td>10</td>
<td>0.7927</td>
<td>857.86</td>
<td>.0001*</td>
<td>10</td>
<td>0.1661</td>
</tr>
<tr>
<td>11</td>
<td>0.7432</td>
<td>944.36</td>
<td>.0001*</td>
<td>11</td>
<td>0.1713</td>
</tr>
<tr>
<td>12</td>
<td>0.7504</td>
<td>1036.84</td>
<td>.0001*</td>
<td>12</td>
<td>-0.3154</td>
</tr>
<tr>
<td>13</td>
<td>0.7127</td>
<td>1177.89</td>
<td>.0001*</td>
<td>13</td>
<td>-0.3075</td>
</tr>
<tr>
<td>14</td>
<td>0.6482</td>
<td>1195.65</td>
<td>.0001*</td>
<td>14</td>
<td>-0.266</td>
</tr>
<tr>
<td>15</td>
<td>0.5859</td>
<td>1241.60</td>
<td>.0001*</td>
<td>15</td>
<td>0.0868</td>
</tr>
<tr>
<td>16</td>
<td>0.5560</td>
<td>1385.64</td>
<td>.0001*</td>
<td>16</td>
<td>0.0250</td>
</tr>
<tr>
<td>17</td>
<td>0.4987</td>
<td>1530.30</td>
<td>.0001*</td>
<td>17</td>
<td>0.0355</td>
</tr>
<tr>
<td>18</td>
<td>0.4867</td>
<td>1597.64</td>
<td>.0001*</td>
<td>18</td>
<td>0.0734</td>
</tr>
<tr>
<td>19</td>
<td>0.4490</td>
<td>1401.60</td>
<td>.0001*</td>
<td>19</td>
<td>0.0484</td>
</tr>
<tr>
<td>20</td>
<td>0.4416</td>
<td>1334.15</td>
<td>.0001*</td>
<td>20</td>
<td>0.0460</td>
</tr>
<tr>
<td>21</td>
<td>0.4572</td>
<td>1489.90</td>
<td>.0001*</td>
<td>21</td>
<td>0.0460</td>
</tr>
<tr>
<td>22</td>
<td>0.4825</td>
<td>1518.60</td>
<td>.0001*</td>
<td>22</td>
<td>-0.1962</td>
</tr>
<tr>
<td>23</td>
<td>0.5171</td>
<td>1556.48</td>
<td>.0001*</td>
<td>23</td>
<td>0.0524</td>
</tr>
<tr>
<td>24</td>
<td>0.5322</td>
<td>1606.65</td>
<td>.0001*</td>
<td>24</td>
<td>0.0460</td>
</tr>
<tr>
<td>25</td>
<td>0.4940</td>
<td>1649.18</td>
<td>.0001*</td>
<td>25</td>
<td>-0.1827</td>
</tr>
</tbody>
</table>
Note the following about these graphs:

- Blue lines represent two standard errors.
- For autocorrelation, the Ljung-Box $Q$ and $p$-values appear for each lag. The $Q$-statistic can be 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.
- The number of lags begins with 0 to provide a broader picture of the analysis. To compute correlations beginning with lag 1, modify the JMP preferences before generating the graph. Select File > Preferences > Platforms > Time Series and then select Suppress Lag 0 in ACF and PACF.

Statistical Details for Autocorrelation and Partial Autocorrelation

The autocorrelation for the $k$th lag is computed as follows:

$$ r_k = \frac{c_k}{c_0} \quad \text{where} \quad c_k = \frac{1}{N} \sum_{t=k+1}^{N} (y_t - \bar{y})(y_{t-k} - \bar{y}) $$

where $\bar{y}$ is the mean of the $N$ non-missing 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

$$ SE_k = \frac{1}{\sqrt{N}} \left( \frac{1}{1 + 2 \sum_{i=1}^{k-1} r_i} \right) $$

For partial autocorrelation, the blue lines represent $\pm 2$ standard errors for approximate 95% confidence limits, where the standard error is computed as follows:

$$ SE_k = \frac{1}{\sqrt{n}} \quad \text{for all} \quad k $$

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 9.4 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 9.4 shows the AR coefficients for the Seriesg data.

Figure 9.4 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 9.5).

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 \( f_i = \frac{i}{N} \) 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, \( I(f_i) \), and its average value. The probability of observing a larger Kappa if the null hypothesis is true is given by

\[
Pr(k > \kappa) = 1 - \frac{q}{\sqrt{j} \sum_{j=0}^{q} (-1)^j \left( \begin{array}{c} q \\ j \end{array} \right) \left( \max \left( 1 - \frac{ik}{q}, 0 \right) \right)^q}
\]

where \( q = \frac{N}{2} \) if \( N \) is even, \( q = \frac{N-1}{2} \) 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 \( q > 100 \), 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 9.5** 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 \( f_i = \frac{i}{N} \) (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, \( a_t \).

Cosine is the Fourier cosine coefficients, \( b_t \).

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

\[
wt = (1 - B)^d (1 - B)^s D y_t
\]

where \( t \) is the time index and \( B \) is the backshift operator defined by \( By_t = y_{t-1} \).

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 Differencing Specification 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.

**Show Lag Plot**

The plot show how an observation at time \( t \) is related to another observation at time \( t \pm p \). \( \pm p \) is the lag, because the plot allows negative values and positive values.

**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 9.6 shows the Model Comparison Report.
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 9.7 shows the plots for the ARIMA(1,1,1) model.
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)**

<table>
<thead>
<tr>
<th>Model Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DF</td>
<td>140</td>
</tr>
<tr>
<td>Sum of Squared Errors</td>
<td>127962.119</td>
</tr>
<tr>
<td>Variance Estimate</td>
<td>970.943764</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>31.2990761</td>
</tr>
<tr>
<td>Akaike's Information Criterion</td>
<td>1394.12184</td>
</tr>
<tr>
<td>Schwartz Bayesian Criterion</td>
<td>1403.01968</td>
</tr>
<tr>
<td>R-Square Adj</td>
<td>0.951</td>
</tr>
<tr>
<td>MAPE</td>
<td>8.828</td>
</tr>
<tr>
<td>MAE</td>
<td>24.206</td>
</tr>
<tr>
<td>-2LogLikelihood</td>
<td>1388.12184</td>
</tr>
</tbody>
</table>

**Variance Estimate** is the unconditional sum of squares (SSE) divided by the number of degrees of freedom, \( SSE / (n - k) \). This is the sample estimate of the variance of the random shocks \( a_p \) described in the section “ARIMA Model” on page 168.
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 $\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, $\hat{y}_i$ are the one-step-ahead forecasts, and $\bar{y}_i$ 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

\[
1 - \frac{(n-1)}{(n-k)(1 - R^2)}
\]

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

\[
\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|
\]

$-2\log\text{Likelihood}$  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” on page 168.
Parameter Estimates Table

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” on page 168.
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” on page 156; 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:

\[
\phi(B)(y_t - \mu) = \theta(B)\eta_t
\]

where

\( t \) is the time index

\( B \) is the backshift operator defined as \( By_t = y_{t-1} \)

\( w_t = (1 - B)^d y_t \) is the response series after differencing
μ is the intercept or mean term.

φ(B) and θ(B), respectively, the autoregressive operator and the moving average operator and are written

\[ φ(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \]  and \[ θ(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q \]

\[ a_t \] are the sequence of random shocks.

The \[ a_t \] are assumed to be independent and normally distributed with mean zero and constant variance.

The model can be rewritten as

\[ φ(B) a_t = δ + θ(B) a_t \]  where the constant estimate \( δ \) is given by the relation

\[ δ = φ(B) μ = μ - \phi_1 μ - \phi_2 μ - \cdots - \phi_p μ . \]

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 \( φ(B) \) 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 \( θ(B) \).
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 \( μ \) 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:

\[
   w_t = (1 - B)^d (1 - B^s)^D y_t \\
   \varphi(B) = (1 - \varphi_{1,1}B - \varphi_{1,2}B^2 - \ldots - \varphi_{1,p}B^p)(1 - \varphi_{2,s}B^s - \varphi_{2,2s}B^{2s} - \ldots - \varphi_{2,p_s}B^{ps}) \\
   \theta(B) = (1 - \theta_{1,1}B - \theta_{1,2}B^2 - \ldots - \theta_{1,q}B^q)(1 - \theta_{2,s}B^s - \theta_{2,2s}B^{2s} - \ldots - \theta_{2,q_s}B^{qs})
\]

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 9.8 shows the dialog.
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 9.9.

Figure 9.9 Series J Launch Dialog

When you click OK, the report in Figure 9.10 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.
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 9.11 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.
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 \( 2 \times ACF \) length + 1.
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 Prewhitening 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 9.13.
Figure 9.13 Prewhitening Dialog

Click **Estimate** to reveal the Prewhitening 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

$$Y_t - \mu = \frac{\omega_1(B)}{\delta_1(B)} X_{1,t-d_1} + \cdots + \frac{\omega_m(B)}{\delta_m(B)} X_{m,t-d_m} + \frac{\Theta(B)}{\Phi(B)} \varepsilon_t$$

where

- $Y_t$ denotes the output series
- $X_{1,t}$ to $X_{m,t}$ denote m input series
- $\varepsilon_t$ represents the noise series
\( X_{1,t-d1} \) indicates the series \( X_1 \) is indexed by \( t \) with a \( d1 \)-step lag

\( \mu \) represents the mean level of the model

\( \phi(B) \) 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 9.14**  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 9.14.

### Model Reports

The analysis report is titled Transfer Function Model and is indexed sequentially. Results for the Series J example are shown in Figure 9.15.

**Figure 9.15** Series J Transfer Function Reports

<table>
<thead>
<tr>
<th>Model Summary</th>
<th>Parameter Estimates</th>
<th>Residuals, Iteration History</th>
<th>Interactive Forecasting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Summary</strong> gathers information that is useful for comparing models.</td>
<td><strong>Parameter Estimates</strong> 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.</td>
<td><strong>Residuals, Iteration History</strong> are the same as their ARIMA counterparts.</td>
<td><strong>Interactive Forecasting</strong> provides a forecasting graph based on a specified confidence interval. The functionality changes based on the number entered in the Forecast Periods box.</td>
</tr>
<tr>
<td>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</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

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:

\[ y_t = \mu_t + \beta_t + s(t) + a_t \]

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 \( j = 0, 1, \ldots, s-1 \) 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
- \( \phi \), 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.

**Simple Moving Average**

A simple moving average model (SMA) produces forecasted values that are equal to the average of consecutive observations in a time window. The forecasts can be uncentered or centered in the time window.

To fit a simple moving average model, select Smoothing > Simple Moving Average. A window appears with the following options:

**Enter smoothing window width** Enter the width of the smoothing window.
Centered  Choose whether to center the forecasted values.

The Simple Moving Average report shows a time plot of the data and the fitted model. The red triangle menu has the following options:

**Add Model**  Select this option to fit another model. When additional models are fit, the model is added to the time plot of the data.

**Save to Data Table**  Saves the original data, and forecasts of all moving average models.

**Show Points**  Shows or hides the points on the plot.

**Connecting Lines**  Shows or hides the lines on the plot.

**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 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

$$(1 - B)y_t = (1 - \theta B)\alpha_t \quad \text{with} \quad \theta = 1 - \alpha.$$  

The moving average form of the model is

$$y_t = a_t + \sum_{j=1}^{\infty} \alpha a_{t-j}$$  

**Double (Brown) Exponential Smoothing**

The model for double exponential smoothing is $y_t = \mu_t + \beta_1 t + \alpha_t$.

The smoothing equations, defined in terms of a single smoothing weight $\alpha$ are

$$L_t = \alpha y_t + (1 - \alpha) L_{t-1} \quad \text{and} \quad T_t = \alpha (L_t - L_{t-1}) + (1 - \alpha) T_{t-1}.$$  

This model is equivalent to an ARIMA(0, 1, 1)(0, 1, 1)$_1$ model

$$(1 - B)^2 y_t = (1 - \theta B)^2 a_t \quad \text{where} \quad \theta_{1,1} = \theta_{2,1} \quad \text{with} \quad \theta = 1 - \alpha.$$
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

\[ L_t = \alpha y_t + (1-\alpha)(L_{t-1} + T_{t-1}) \quad \text{and} \quad T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \]

This model is equivalent to an ARIMA\((0,2,2)\) model where

\[ (1-B)^2 y_t = (1-\theta B - \theta_2 B^2) a_t \quad \text{with} \quad \theta = 2 - \alpha - \alpha \gamma \quad \text{and} \quad \theta_2 = \alpha - 1. \]

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

\[ L_t = \alpha y_t + (1-\alpha)(L_{t-1} + \phi T_{t-1}) \quad \text{and} \quad T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)\phi T_{t-1} \]

This model is equivalent to an ARIMA\((1,1,2)\) model where

\[ (1-\phi B)(1-B)y_t = (1-\theta_1 B - \theta_2 B^2) a_t \quad \text{with} \quad \theta_1 = 1 + \phi - \alpha - \alpha \gamma \phi \quad \text{and} \quad \theta_2 = (\alpha - 1)\phi. \]

The moving average form of the model is

\[ y_t = a_t + \sum_{j=1}^{\infty} \left( \frac{\alpha + \alpha \gamma \phi (\phi^j - 1)}{\phi - 1} \right)a_{t-j} \]

**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

\[ L_t = \alpha(y_t - S_{t-s}) + (1-\alpha)L_{t-1} \quad \text{and} \quad S_t = \delta(y_t - L_{t-s}) + (1-\delta)\phi S_{t-s} \]
This model is equivalent to a seasonal ARIMA(0, 1, 1)(0, 1, 0) \_S model where we define
\[ \theta_1 = \theta_{1,1}, \, \theta_2 = \theta_{2,s}, \, \text{and} \, \theta_3 = -\theta_{1,1}\theta_{2,s} \]
so
\[ (1 - B)(1 - B^s)y_t = (1 - \theta_1 B - \theta_2 B^2 - \theta_3 B^{s+1})a_t \]
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
\[ y_t = a_t + \sum_{j=1}^{\infty} \psi_j a_{t-j} \text{ where } \psi = \begin{cases} \alpha \text{ for } j \mod s \neq 0 \\ \alpha + \delta(1 - \alpha) \text{ for } j \mod s = 0 \end{cases} \]

**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
\[ L_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1}), \] \[ T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}, \]\[ S_t = \delta(y_t - L_t) + (1 - \delta)S_{t-s}. \]
This model is equivalent to a seasonal ARIMA(0, 1, s+1)(0, 1, 0)s model
\[ (1 - B)(1 - B^s)y_t = \left(1 - \sum_{i=1}^{s+1} \theta_i B^i \right)a_t \]
The moving average form of the model is
\[ y_t = a_t + \sum_{j=1}^{\infty} \psi_j a_{t-j} \]
where
\[ \psi = \begin{cases} \alpha + j\alpha\gamma, & \text{ for } j \mod s \neq 0 \\ \alpha + j\alpha\gamma + \delta(1 - \alpha), & \text{ for } j \mod s = 0 \end{cases} \]
Chapter 10

Response Screening
Screen Large-Scale Data

The analysis of large-scale data sets, where hundreds or thousands of measurements are taken on a part or an organism, requires innovative approaches. But testing many responses for the effects of factors can be challenging, if not misleading, without appropriate methodology.

Response Screening automates the process of conducting tests across a large number of responses. Your test results and summary statistics are presented in data tables, rather than reports, to enable data exploration. A False-Discovery Rate approach guards against incorrect declarations of significance. Plots of p-values are scaled using the LogWorth, making them easily interpretable.

Because large scale data sets are often messy, Response Screening presents methods that address irregularly distributed and missing data. A robust estimate method allows outliers to remain in the data, but reduces the sensitivity of tests to these outliers. Missing data options allow missing values to be included in the analysis. These features enable you to analyze your data without first conducting an extensive analysis of data quality.

When you have many observations, even differences that are of no practical interest can be statistically significant. Response Screening presents tests of practical difference, where you specify the difference that you are interested in detecting. On the other hand, you might want to know whether differences do not exceed a given magnitude, that is, if the means are equivalent. For this purpose, Response Screening presents equivalence tests.

Figure 10.1 Example of a Response Screening Plot
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Screening Platform Overview</td>
<td>187</td>
</tr>
<tr>
<td>Example of Response Screening</td>
<td>188</td>
</tr>
<tr>
<td>Launch the Response Screening Platform</td>
<td>191</td>
</tr>
<tr>
<td>The Response Screening Report</td>
<td>193</td>
</tr>
<tr>
<td>FDR PValue Plot</td>
<td>193</td>
</tr>
<tr>
<td>FDR LogWorth by Effect Size</td>
<td>194</td>
</tr>
<tr>
<td>FDR LogWorth by RSquare</td>
<td>195</td>
</tr>
<tr>
<td>The PValues Data Table</td>
<td>195</td>
</tr>
<tr>
<td>Response Screening Platform Options</td>
<td>198</td>
</tr>
<tr>
<td>Means Data Table</td>
<td>199</td>
</tr>
<tr>
<td>Compare Means Data Table</td>
<td>200</td>
</tr>
<tr>
<td>The Response Screening Personality in Fit Model</td>
<td>202</td>
</tr>
<tr>
<td>Launch Response Screening in Fit Model</td>
<td>202</td>
</tr>
<tr>
<td>The Fit Response Screening Report</td>
<td>204</td>
</tr>
<tr>
<td>PValues Data Table</td>
<td>204</td>
</tr>
<tr>
<td>Y Fits Data Table</td>
<td>205</td>
</tr>
<tr>
<td>Additional Examples of Response Screening</td>
<td>206</td>
</tr>
<tr>
<td>Example of Tests of Practical Significance and Equivalence</td>
<td>206</td>
</tr>
<tr>
<td>Example of the MaxLogWorth Option</td>
<td>208</td>
</tr>
<tr>
<td>Example of Robust Fit</td>
<td>210</td>
</tr>
<tr>
<td>Response Screening Personality</td>
<td>213</td>
</tr>
<tr>
<td>Statistical Details</td>
<td>215</td>
</tr>
</tbody>
</table>
Response Screening automates the process of conducting tests across a large number of responses. It tests each response that you specify against each factor that you specify. Response screening addresses two main issues connected with large-scale data. These are the need to conduct many tests, and the requirement to deal effectively with outliers and missing values.

Response screening is available as a platform and as a Fit Model personality. In both cases, it performs tests analogous to those found in the Fit Y by X platform, as shown in Table 10.1. As a personality, it performs tests of the response against the individual model effects.

To facilitate and support the multiple inferences that are required, Response Screening provides these features:

**Data Tables** Results are shown in data tables, as well as in a report, to enable you to explore, sort, search, and plot your results. Statistics that facilitate plot interpretation are provided, such as the LogWorth of $p$-values.

**False Discovery Rates** Because you are conducting a large number of tests, you need to control the overall rate of declaring tests significant by chance alone. Response screening controls the *false discovery rate*. The False Discovery Rate (FDR) is the expected proportion of significant tests that are incorrectly declared significant (Benjamini and Hochberg, 1995).

**Tests of Practical Significance** When you have many observations, even small effects that are of no practical consequence can result in statistical significance. To address this issue, you can define an effect size that you consider to be of *practical significance*. You then conduct tests of practical significance, thereby only detecting effects large enough to be of pragmatic interest.

**Equivalence Tests** When you are studying many factors, you are often interested in those that have essentially equivalent effects on the response. In this case, you can specify an effect size that defines practical equivalence and then conduct equivalence tests.

To address issues that arise when dealing with messy data, Response Screening provides features to deal with outliers and missing data. These features enable you to analyze your data directly, without expending effort to address data quality issues:

**Robust Estimation** Outliers in your data increase estimates of standard error, causing tests to be insensitive to real effects. Select the Robust option to conduct Huber M-estimation. Outliers remain in the data, but the sensitivity of tests to these outliers is reduced.

**Missing Value Options** The platform contains an option to treat missing values on categorical predictors in an informative fashion.
Table 10.1 Analyses Performed by Response Screening

<table>
<thead>
<tr>
<th>Response</th>
<th>Factor</th>
<th>Fit Y by X Analysis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td>Categorical</td>
<td>Oneway</td>
<td>Analysis of Variance</td>
</tr>
<tr>
<td>Continuous</td>
<td>Continuous</td>
<td>Bivariate</td>
<td>Simple Linear Regression</td>
</tr>
<tr>
<td>Categorical</td>
<td>Categorical</td>
<td>Contingency</td>
<td>Chi-Square</td>
</tr>
<tr>
<td>Categorical</td>
<td>Continuous</td>
<td>Logistic</td>
<td>Simple Logistic Regression</td>
</tr>
</tbody>
</table>

The Response Screening platform generates a report and a data table: the Response Screening report and the PValues table. The Response Screening personality generates a report and two data tables: the Fit Response Screening report, the PValues table, and the Y Fits table.

The JSL command `Summarize Y by X` performs the same function as the Response Screening platform but without creating a platform window. See the *JSL Syntax Reference* book for details.

**Example of Response Screening**

The *Probe.jmp* sample data table contains 387 characteristics (the *Responses* column group) measured on 5800 wafers. The *Lot ID* and *Wafer Number* columns uniquely identify the wafer. You are interested in which of the characteristics show different values across a process change (*Process*).

1. Open the *Probe.jmp* sample data table.
2. Select **Analyze > Modeling > Response Screening**.
   
   The Response Screening launch window appears.
3. Select the *Responses* column group and click **Y, Response**.
4. Select *Process* and click **X**.
5. Click **OK**.

The Response Screening report appears, along with a data table of supporting information. The report (Figure 10.2) shows the FDR PValue Plot, but also contains two other plot reports. The table contains a row for each of the 387 columns that you entered as **Y, Response**.

The FDR PValue Plot shows two types of *p*-values, FDR PValue and PValue, for each of the 387 tests. These are plotted against Rank Fraction. PValue is the usual *p*-value for the test of a *Y* against *Process*. The FDR PValue is a *p*-value that is adjusted to guarantee a given false discover rate (FDR), here 0.05. The FDR PValues are plotted in blue and the PValues are plotted in red. The Rank Fraction ranks the FDR *p*-values from smallest to largest, in order of decreasing significance.
Both the horizontal blue line and the sloped red line on the plot are thresholds for FDR significance. Tests with FDR $p$-values that fall below the blue line are significant at the 0.05 level when adjusted for the false discovery rate. Tests with ordinary $p$-values that fall below the red line are significant at the 0.05 level when adjusted for the false discovery rate. In this way, the plot enables you to read FDR significance from either set of $p$-values.

**Figure 10.2** Response Screening Report for 387 Tests against Process

The FDR PValue Plot shows that more than 60% of the tests are significant. A handful of tests are significant using the usual $p$-value, but not significant using the FDR $p$-value. These tests correspond to the red points that are above the red line, but below the blue line.

To identify the characteristics that are significantly different across Process, you can drag a rectangle around the appropriate points in the plot. This selects the rows corresponding to these points in the PValues table, where the names of the characteristics are given in the first column. Alternatively, you can select the corresponding rows in the PValues table.

The PValues data table (Figure 10.3) contains 387 rows, one for each response measure in the Responses group. The response is given in the first column, called Y. Each response is tested against the effect in the X column, namely, Process.

The remaining columns give information about the test of Y against X. Here the test is a Oneway Analysis of Variance. In addition to other information, the table gives the test's $p$-value, LogWorth, FDR (False Discovery Rate) $p$-value, and FDR LogWorth. See “The Response Screening Report” on page 193 for details. Use this table to sort by the various
statistics, select rows, or plot quantities of interest.

See “The Response Screening Report” on page 193 for details about the report and PValues table.

Figure 10.3 PValues Data Table, Partial View
Launch the Response Screening Platform

Launch the Response Screening platform by selecting Analyze > Modeling > Response Screening.

**Figure 10.4** Response Screening Launch Window

![Response Screening Launch Window]

**Launch Window Roles**

- **Y, Response**  Identifies the response columns containing the measurements to be analyzed.
- **X**  Identifies the columns against which you want to test the responses.
- **Grouping**  For each level of the specified column, analyzes the corresponding rows separately, but presents the results in a single table and report.
- **Weight**  Identifies a column whose values assign a weight to each row. These values are used as weights in the analysis. For details, see the Weight section in the *Fitting Linear Models* book’s Introduction to the Fit Model Platform chapter.
Launch the Response Screening Platform

**Freq**  Identifies a column whose values assign a frequency to each row. These values enable you to account for pre-summarized data. For details, see the Frequency section in the *Fitting Linear Models* book’s Introduction to the Fit Model Platform chapter.

**By**  For each level of the specified column, analyzes the corresponding Ys and Xs and presents the results in separate tables and reports.

**Launch Window Options**

**Robust**  For continuous responses, uses robust (Huber) estimation to down weight outliers. If there are no outliers, these estimates are close to the least squares estimates. Note that this option increases processing time.

**Poisson Y**  Fits each Y response as a count having a Poisson distribution. The test is only performed for categorical X. This option is appropriate when your responses are counts.

**Kappa**  Adds a new column called Kappa to the data table. If Y and X are both categorical and have the same levels, kappa is provided. This is a measure of agreement between Y and X.

**Corr**  Adds a new column called Corr to the data table. If Y and X are both categorical, Kendall’s Tau-b is provided as a measure of association.

**Same Y Scale**  Aligns all the Y responses to the same scale when you run individual analyses using the report’s Fit Selected Items options.

**Missing is category**  For any categorical X variable, treats missing values on X as a category.

**Force X Categorical**  Ignores the modeling type and treats all X columns as categorical.

**Force X Continuous**  Ignores the modeling type and treats all X columns as continuous.

**Force Y Categorical**  Ignores the modeling type and treats all Y columns as categorical.

**Force Y Continuous**  Ignores the modeling type and treats all Y columns as continuous.

**Paired X and Y**  Performs tests only for Y columns paired with X columns according to their order in the Y, Response and X lists. The first Y is paired with the first X, the second Y with the second X, and so on.

**Unthreaded**  Suppresses multithreading.

**Practical Difference Portion**  The fraction of the specification range, or of an estimated six standard deviation range, that represents a difference that you consider pragmatically meaningful. If Spec Limits is not set as a column property, a range of six standard deviations is estimated for the response. The standard deviation estimate is computed from the interquartile range (IQR), as $\sigma = (IQR)/(1.349795)$.

If no Practical Difference Proportion is specified, its value defaults to 0.10. Tests of practical significance and equivalence tests use this difference to determine the practical difference. See “*Compare Means Data Table*” on page 200.
MaxLogWorth  Use to control the scale of plots involving LogWorth values (-log10 of p-values). LogWorth values that exceed MaxLogWorth are plotted as MaxLogWorth to prevent extreme scales in LogWorth plots. See “Example of the MaxLogWorth Option” on page 208 for an example.

OK   Conducts the analysis and displays the results.

Cancel  Closes the launch window.

Remove  Removes the selected variable from the assigned role.

Recall  Populates the launch window with the previous model specification that you ran.

Help  Opens the Help topics for the Response Screening launch window.

The Response Screening report consists of several Graph Builder plots. These plots focus on False Discovery Rate (FDR) statistics. For details, see “The False Discovery Rate” on page 215.

The default plots are the FDR PValue Plot, the FDR LogWorth by Effect Size, and the FDR LogWorth by RSquare. If you select the Robust option on the launch window, Robust versions of each of these reports are also presented. In addition, a Robust LogWorth by LogWorth plot is presented to help assess the impact of using the robust fit. The standard Graph Builder red triangle options for each plot are available. For details, see the Graph Builder chapter in the Essential Graphing book.

FDR PValue Plot

The FDR PValue Plot report shows a plot of FDR PValues and PValues against the Rank Fraction. The Rank Fraction ranks the PValues in order of decreasing significance. FDR PValues are plotted in blue and PValues in red.

A blue horizontal line shows the 0.05 significance level. (Note that you can change this level in the Graph Builder by selecting Show Control Panel.) A red increasing line provides an FDR threshold for unadjusted p-values. A p-value falls below the red line precisely when the FDR-adjusted p-value falls below the blue line. This enables you to read significance relative to the FDR from either the adjusted or unadjusted p-values.

Figure 10.5 shows the FDR PValue Plot for the Probe.jmp sample data table. Note that some tests are significant according to the usual p-value but not according to the FDR p-value.
When you have large effects, the associated $p$-values are often very small. Visualizing these small values graphically can be challenging. When transformed to the LogWorth ($-\log_{10}(p\text{-value})$) scale, highly significant $p$-values have large LogWorths and nonsignificant $p$-values have low LogWorths. A LogWorth of zero corresponds to a nonsignificant $p$-value of 1. Any LogWorth above 2 corresponds to a $p$-value below 0.01.

In the FDR LogWorth by Effect Size plot, the vertical axis is the FDR LogWorth and the horizontal axis is the Effect Size. Generally, larger effects lead to more significant $p$-values and larger LogWorths. However, this relationship is not necessarily strong because significance also depends on the error variance. In fact, large LogWorths can be associated with small effects, and small LogWorths can be associated with large effects, because of the size of the error variance. The FDR LogWorth by Effect Size plot enables you to explore this relationship.

Figure 10.6 shows the FDR LogWorth by Effect size plot for the Probe.jmp sample data table. Most FDR LogWorth values exceed 2, which indicates that most effects are significant at the 0.01 level. The FDR LogWorth values of about 320 correspond to extremely small $p$-values.
Chapter 10
Response Screening
Specialized Models

The FDR LogWorth by RSquare plot shows the FDR LogWorth on the vertical axis and RSquare values on the horizontal axis. Larger LogWorth values tend to be associated with larger RSquare values, but this relationship also depends on the number of observations.

The PValues Data Table

The PValues data table contains a row for each pair of Y and X variables. If you specified a column for Group, the PValues data table contains a first column called Group. A row appears for each level of the Group column and for each pair of Y and X variables.

Figure 10.7 shows the PValues data table created using the Probe.jmp sample data table.
The PValues data table displays columns containing measures and statistics that are appropriate for the selected fit and combination of Y and X modeling types. The columns in the data table include:

Y  The specified response columns.
X  The specified factor columns.
Count  The number of rows used for testing, or the corresponding sum of the Freq or Weight variable.
PValue  The p-value for the significance test corresponding to the pair of Y and X variables. See the Basic Analysis book for additional details about Fit Y by X statistics.
LogWorth  The quantity \(-\log_{10}(p\text{-value})\). This transformation adjusts p-values to provide an appropriate scale for graphing. A value that exceeds 2 is significant at the 0.01 level (because \(-\log_{10}(0.01) = 2\)).
FDR PValue  The False Discovery Rate p-value calculated using the Benjamini-Hochberg technique. This technique adjusts the p-values to control the false discovery rate for multiple tests. If there is no Group variable, the set of multiple tests includes all tests displayed in the table. If there is a Group variable, the set of multiple tests consists of all tests conducted for each level of the Group variable. For details about the FDR correction, see Benjamini and Hochberg, 1995. For details about the false discovery rate, see “The False Discovery Rate” on page 215.
FDR LogWorth  The quantity \(-\log_{10}(\text{FDR PValue})\). This is the best statistic for plotting and assessing significance. Note that small p-values result in high FDR LogWorth values.
**Effect Size** Indicates the extent to which response values differ across the levels or values of X. Effect sizes are scale invariant.

- When Y is continuous, the effect size is the square root of the average sum of squares for the hypothesis divided by a robust estimate of the response standard deviation. If the interquartile range (IQR) is nonzero, the standard deviation estimate is $IQR / 1.3489795$. If the IQR is zero, the sample standard deviation is used.

- When Y is categorical and X is continuous, the effect size is the square root of the average ChiSquare value for the whole model test.

- When Y and X are both categorical, the effect size is the square root of the average Pearson ChiSquare.

**Rank Fraction** The rank of the FDR LogWorth expressed as a fraction of the number of tests. If the number of tests is $m$, the largest FDR LogWorth value has Rank Fraction $1/m$, and the smallest has Rank Fraction 1. Equivalently, the Rank Fraction ranks the $p$-values in increasing order, as a fraction of the number of tests. The Rank Fraction is used in plotting the PValues and FDR PValues in rank order of decreasing significance.

**YMean** The mean of Y.

**SSE** Appears when Y is continuous. The sum of squares for error.

**DFE** Appears when Y is continuous. The degrees of freedom for error.

**MSE** Appears when Y is continuous. The mean squared error.

**F Ratio** Appears when Y is continuous. The F Ratio for the analysis of variance or regression test.

**RSquare** Appears when Y is continuous. The coefficient of determination, which measures the proportion of total variation explained by the model.

**DF** Appears when Y and X are both categorical. The degrees of freedom for the ChiSquare test.

**LR Chisq** Appears when Y and X are both categorical. The value of the Likelihood Ratio ChiSquare statistic.

**Columns Added for Robust Option**

If you suspect that your data contains outliers, select the Robust option on the launch window to reduce the sensitivity of tests for continuous responses to outliers. With this option, Huber M-estimates (Huber and Ronchetti, 2009) are used in fitting regression and ANOVA models. Huber M-estimates are fairly close to least squares estimates when there are no outliers, but use outlier-downweighting when there are outliers.

The following columns are added to the PValues data table when the Robust option is selected in the launch window. The Robust option only applies when Y is continuous, so Robust column cells are empty when Y is categorical. See the *Basic Analysis* book for additional details.
Robust **PValue**  The p-value for the significance test corresponding to the pair of Y and X variables using a robust.

**Robust LogWorth**  The quantity \(-\log_{10}(\text{Robust PValue})\).

**Robust FDR PValue**  The False Discovery Rate calculated for the Robust PValues using the Benjamini-Hochberg technique. If there is no Group variable, the multiple test adjustment applies to all tests displayed in the table. If there is a Group variable, the multiple test adjustment applies to all tests conducted for each level of the Group variable.

**Robust FDR LogWorth**  The quantity \(-\log_{10}(\text{Robust FDR PValue})\).

**Robust Rank Fraction**  The rank of the Robust FDR LogWorth expressed as a fraction of the number of tests.

**Robust Chisq**  The chi-square value associated with the robust test.

**Robust Sigma**  The robust estimate of the error standard deviation.

**Robust Outlier Portion**  The portion of the values whose distance from the robust mean exceeds three times the Robust Sigma.

**Robust CpuTime**  Time in seconds required to create the Robust report.

**PValues Data Table Scripts**

Relevant scripts are saved to the PValues data table. All but one of these reproduce plots provided in the report. When you select rows in the PValues table, the Fit Selected script produces the appropriate Fit Y by X analyses.

**Response Screening Platform Options**

The Response Screening red triangle menu contains options to customize the display and to compute and save calculated data.

**Fit Selected Items**  For selected relationships, adds the appropriate Fit Y by X reports to the Response Screening report. You can select relationships by selecting rows in the PValue data table or points in the plots.

**Save Means**  For continuous Ys and categorical Xs, creates a data table with the counts, means, and standard deviations for each level of the categorical variable. If the Robust option is selected, the robust mean is included.

**Save Compare Means**  For continuous Ys and categorical Xs, tests all pairwise comparisons across the levels of the categorical variable. For each comparison, the data table gives the usual t-test, a test of practical significance, an equivalence test, and a column that uses color coding to summarize the results. The data table also contains a script that plots
Practical LogWorth by Relative Practical Difference. See “Compare Means Data Table” on page 200 for details and “Example of Tests of Practical Significance and Equivalence” on page 206 for an example.

**Save Std Residuals**  Saves a new group of columns to the original data table and places these in a column group call Residual Group. For each continuous Y and categorical X, a column is constructed containing the residuals divided by their estimated standard deviation. In other words, the column contains standardized residuals. The column is defined by a formula.

If the Robust option is selected, standardized residual columns are constructed using robust fits and robust estimates.

**Save Outlier Indicator**  Saves a new group of columns to the original data table and places these in a column group call Outlier Group. Save Outlier Indicator is most effective when you have selected the Robust option.

For each continuous Y and categorical X, a column that indicates outliers is constructed. An outlier is a point whose distance to the predicted value exceeds three times an estimate of sigma. In other words, an outlier is a point whose standardized residual exceeds three. The column is defined by a formula.

If the Robust option is selected, robust fits and robust estimates are used. An outlier is a point whose distance to the predicted value exceeds three times the robust estimate of sigma.

The Cluster Outliers script is added to the original data table. The script shows outliers on a hierarchical cluster plot of the data.

**Script**  Lists the Script menu options for the platform. See the *Using JMP* book for details.

**Script All By-Groups**  Lists the Script All By-Group menu options for the platform. See the *Using JMP* book for details.

**Means Data Table**

The Means data table contains a row for each combination of response and X level. For the Probe.jmp sample data table, there are 387 response variables, each tested against Process at two levels. The Means table contains $387 \times 2 = 774$ rows (Figure 10.8).
Figure 10.8 Means Data Table

<table>
<thead>
<tr>
<th>Y</th>
<th>X</th>
<th>Level</th>
<th>Count</th>
<th>Mean</th>
<th>StdDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DELL_RPNR</td>
<td>New</td>
<td>3044</td>
<td>0.20338440106</td>
<td>0.1914031302</td>
</tr>
<tr>
<td>2</td>
<td>DELL_RPNR</td>
<td>Old</td>
<td>2750</td>
<td>0.23663294148</td>
<td>0.1778299414</td>
</tr>
<tr>
<td>3</td>
<td>DELL_RPRR</td>
<td>New</td>
<td>3044</td>
<td>-0.068725066</td>
<td>0.1784161107</td>
</tr>
<tr>
<td>4</td>
<td>DELL_RPRR</td>
<td>Old</td>
<td>2750</td>
<td>-0.038321135</td>
<td>1.0165048167</td>
</tr>
<tr>
<td>5</td>
<td>DELW_M1</td>
<td>New</td>
<td>3039</td>
<td>-0.004187197</td>
<td>1.0374346893</td>
</tr>
<tr>
<td>6</td>
<td>DELW_M1</td>
<td>Old</td>
<td>2750</td>
<td>-0.071406861</td>
<td>0.0389746703</td>
</tr>
<tr>
<td>7</td>
<td>DELW_M2</td>
<td>New</td>
<td>3039</td>
<td>0.8014775806</td>
<td>0.0968764759</td>
</tr>
<tr>
<td>8</td>
<td>DELW_M2</td>
<td>Old</td>
<td>2689</td>
<td>0.1476950806</td>
<td>0.0694563527</td>
</tr>
<tr>
<td>9</td>
<td>DELW_NBSE</td>
<td>New</td>
<td>3028</td>
<td>0.0156755669</td>
<td>7.8018912121</td>
</tr>
<tr>
<td>10</td>
<td>DELW_NBSE</td>
<td>Old</td>
<td>2750</td>
<td>1.4765385729</td>
<td>0.1326787074</td>
</tr>
<tr>
<td>11</td>
<td>DELW_NMFT</td>
<td>New</td>
<td>3038</td>
<td>0.3411752522</td>
<td>0.0947626984</td>
</tr>
<tr>
<td>12</td>
<td>DELW_NMFT</td>
<td>Old</td>
<td>2750</td>
<td>0.3813685954</td>
<td>0.097651441</td>
</tr>
<tr>
<td>13</td>
<td>DELW_NMFT</td>
<td>New</td>
<td>3043</td>
<td>3.1727025841</td>
<td>1.1419433350</td>
</tr>
<tr>
<td>14</td>
<td>DELW_NMFT</td>
<td>Old</td>
<td>2740</td>
<td>4.6586228576</td>
<td>1.8539632532</td>
</tr>
<tr>
<td>15</td>
<td>DELW_NMNS</td>
<td>New</td>
<td>3032</td>
<td>9.2876152836</td>
<td>1.2840573681</td>
</tr>
<tr>
<td>16</td>
<td>DELW_NMNS</td>
<td>Old</td>
<td>2750</td>
<td>7.8902297602</td>
<td>0.3781490599</td>
</tr>
<tr>
<td>17</td>
<td>DELW_PBAS</td>
<td>New</td>
<td>3036</td>
<td>1.1651478779</td>
<td>0.1355193171</td>
</tr>
<tr>
<td>18</td>
<td>DELW_PBAS</td>
<td>Old</td>
<td>2750</td>
<td>1.2087969753</td>
<td>0.1046827344</td>
</tr>
</tbody>
</table>

The Means data table includes the following columns:

- **Y**: The continuous response variables.
- **X**: The categorical variables.
- **Level**: The level of the categorical X variable.
- **Count**: The count of values in the corresponding Level.
- **Mean**: The mean of the Y variable for the specified Level.
- **StdDev**: The standard deviation of the Y variable for the specified Level.
- **Robust Mean**: The robust M-estimate of the mean. Appears when you select the Robust option on the launch window.

**Compare Means Data Table**

When your data table consists of a large number of rows (large \( n \)), the standard error used in testing can be very small. As a result, tests might be statistically significant, when in fact, the observed difference is too small to be of practical consequence. Tests of practical significance enable you to specify the size of the difference that you consider worth detecting. This difference is called the practical difference. Instead of testing that the difference is zero, you test whether the difference exceeds the practical difference. As a result, the tests are more meaningful, and fewer tests need to be scrutinized.

Equivalence tests enable you to determine whether two levels have essentially the same effect, from a practical perspective, on the response. In other words, an equivalence test tests whether the difference is smaller than the practical difference.
The Compare Means data table provides results for both tests of practical difference and tests of practical equivalence. Each row compares a response across two levels of a categorical factor. Results of the pairwise comparisons are color-coded to facilitate interpretation. See “Practical Difference” on page 201 for a description of how the practical difference is specified. See “Example of Tests of Practical Significance and Equivalence” on page 206 for an example.

**Figure 10.9 Compare Means Data Table**

The Compare Means data table contains a script that plots Practical LogWorth by Relative Practical Difference. Relative Practical Difference is defined as the actual difference divided by the practical difference.

**Y**  The continuous response variables.

**X**  The categorical variables.

**Leveli**  A level of the categorical X variable.

**Levelj**  The level of the categorical X variable being compared to Leveli.

**Difference**  The estimated difference in means across the two levels. If the Robust option is selected, robust estimates of the means are used.

**Std Err Diff**  The standard error of the difference in means. This is a robust estimate if the Robust option is selected.

**Plain Dif PValue**  The $p$-value for the usual Student’s t-test for a pairwise comparison. This is the robust version of the t-test when the Robust option is selected. Tests that are significant at the 0.05 level are highlighted.

**Practical Difference**  The difference in means that is considered to be of practical interest. If you assign a Spec Limit property to the Y variable, the practical difference is computed as the difference between the specification limits multiplied by the Practical Difference Proportion. If no Practical Difference Proportion has been specified, the Practical Difference is the difference between the specification limits multiplied by 0.10. If you do not assign a Spec Limit property to the Y variable, an estimate of its standard deviation is computed from its interquartile range (IQR). This estimate is $\hat{\sigma} = (IQR)/(1.3489795)$. The Practical Difference is computed as $6\hat{\sigma}$ multiplied by the
Practical Difference Proportion. If no Practical Difference Proportion has been specified, the Practical Difference is computed as $6\sigma$ multiplied by 0.10.

**Practical Dif PValue**  The $p$-value for a test of whether the absolute value of the mean difference in Y between Leveli and Levelj is less than or equal to the Practical Difference. A small $p$-value indicates that the absolute difference exceeds the Practical Difference. This indicates that Leveli and Levelj account for a difference that is of practical consequence.

**Practical Equiv PValue**  Uses the Two One-Sided Tests (TOST) method to test for a practical difference between the means (Schuirmann, 1987). The Practical Difference specifies a threshold difference for which smaller differences are considered practically equivalent. One-sided $t$ tests are constructed for two null hypotheses: the true difference exceeds the Practical Difference; the true difference is less than the negative of the Practical Difference. If both tests reject, this indicates that the absolute difference in the means falls within the Practical Difference. Therefore, the groups are considered practically equivalent.

The Practical Equivalence PValue is the largest $p$-value obtained on the one-sided $t$ tests. A small Practical Equiv PValue indicates that the mean response for Leveli is equivalent, in a practical sense, to the mean for Levelj.

**Practical Result**  A description of the results of the tests for practical difference and equivalence. Values are color-coded to help identify significant results.

- Different (Pink): Indicates that the absolute difference is significantly greater than the practical difference.
- Equivalent (Green): Indicates that the absolute difference is significantly within the practical difference.
- Inconclusive (Gray): Indicates that neither the test for practical difference nor the test for practical equivalence is significant.

---

**The Response Screening Personality in Fit Model**

If you are interested in univariate tests against linear model effects, you can fit the Response Screening personality in Fit Model. The report and tables produced test all responses against all model effects.

**Launch Response Screening in Fit Model**

Select Analyze > Fit Model. Enter your Ys and model effects. Select Response Screening from the Personality list (Figure 10.10).
Figure 10.10 Response Screening from the Fit Model Window

Note that a Robust Fit check box is available. Selecting this option enables robust estimation for tests involving continuous responses. These tests use robust (Huber) estimation to downweight outliers. If there are no outliers, these estimates are close to the least squares estimates. Selecting this option increases processing time.

The Informative Missing option provides a coding system for missing values (Figure 10.11). The Informative Missing coding allows estimation of a predictive model despite the presence of missing values. It is useful in situations where missing data are informative. Select this option from the Model Specification red triangle menu.

Figure 10.11 Informative Missing Option
For details about the Fit Model window, see the *Fitting Linear Models* book’s Introduction to the Fit Model Platform chapter.

**The Fit Response Screening Report**

The Fit Response Screening report shows two plots:

- The FDR PValue Plot
- The FDR LogWorth by Rank Fraction Plot

The FDR PValue Plot is interpreted in the same way as for the platform itself. See “The Response Screening Report” on page 193.

The FDR LogWorth by Rank Fraction plot shows FDR LogWorth values plotted against the ranks of the $p$-values. The plotted points decrease or remain constant as rank fraction increases. The plot gives an indication of what proportion of tests are significant. An example using the Response Screening personality is given in “Response Screening Personality” on page 213.

**Model Dialog**  
Opens a window containing the model dialog that you have run to obtain the given report.

**Save Estimates**  
Opens a data table in which each row corresponds to a response and the columns correspond to the model terms. The entries are the parameter estimates obtained by fitting the specified model.

**Save Prediction Formula**  
Adds columns to the original data table containing prediction equations for all responses.

**Script**  
Lists the Script menu options for the platform. See the *Using JMP* book for details.

**PValues Data Table**

The PValues data table contains a row for each pair consisting of a Y variable and a model Effect. The columns in the table include the following. If you select the Robust option on the launch window, the models are fit using Huber M-estimation.

- **Y**  
The specified response columns.

- **Effect**  
The specified model effects.

- **FRatio**  
The test statistic for a test of the Effect. This is the value found in the Effect Tests report in Least Squares Fit.

- **PValue**  
The $p$-value for the significance test corresponding to the FRatio. See the Standard Least Squares chapter in the *Fitting Linear Models* book for additional details about Effect Tests.
**LogWorth** The quantity $-\log_{10}(p)$-value. This transformation adjusts $p$-values to provide an appropriate scale for graphing. A value that exceeds 2 is significant at the 0.01 level (because $-\log_{10}(0.01) = 2$).

**FDR PValue** The False Discovery Rate $p$-value calculated using the Benjamini-Hochberg technique. This technique adjusts the $p$-values to control the false discovery rate for multiple tests. For details about the FDR correction, see Benjamini and Hochberg, 1995. For details about the false discovery rate, see “The False Discovery Rate” on page 215.

**FDR LogWorth** The quantity $-\log_{10}$(FDR PValue). This is the best statistic for plotting and assessing significance. Note that small $p$-values result in high FDR LogWorth values.

**Rank Fraction** The rank of the FDR LogWorth expressed as a fraction of the number of tests. If the number of tests is $m$, the largest FDR LogWorth value has Rank Fraction $1/m$, and the smallest has Rank Fraction 1. Equivalently, the Rank Fraction ranks the $p$-values in increasing order, as a fraction of the number of tests. The Rank Fraction is used in plotting the PValues and FDR PValues in rank order of decreasing significance.

**Test DF** The degrees of freedom for the effect test.

**Y Fits Data Table**

The Y Fits data table contains a row for Y variable. For each Y, the columns in the table summarize information about the model fit. If you select the Robust option on the launch window, the models are fit using Huber M-estimation.

**Y** The specified response columns.

**RSquare** The multiple correlation coefficient.

**RMSE** The Root Mean Square Error.

**Count** The number of observations (or sum of the Weight variable).

**Overall FRatio** The test statistic for model fit from the Analysis of Variance report in Least Squares Fit.

**Overall PValue** The $p$-value for the overall test of model significance.

**Overall LogWorth** The LogWorth of the $p$-value for the overall test of model significance.

**Overall FDR PValue** The overall $p$-value adjusted for the false discovery rate. (See “The Response Screening Report” on page 193.)

**Overall FDR LogWorth** The LogWorth of the Overall FDR PValue.

**Overall Rank Fraction** The rank of the Overall FDR LogWorth expressed as a fraction of the number of tests. If the number of tests is $m$, the largest Overall FDR LogWorth value has Rank Fraction $1/m$, and the smallest has Rank Fraction 1.
Additional Examples of Response Screening

The following examples illustrate various aspects of Response Screening.

Example of Tests of Practical Significance and Equivalence

This example tests for practical differences using the Probe.jmp sample data table.

1. Open the Probe.jmp sample data table.
2. Select Analyze > Modeling > Response Screening.
   The Response Screening Launch window appears.
3. Select the Responses column group and click Y, Response.
4. Select Process and click X.
5. Type 0.15 in the Practical Difference Portion box.
6. Click OK.
7. From the Response Screening report’s red triangle menu, select Save Compare Means.
   Figure 10.12 shows a portion of the data table. For each response in Y, the corresponding row gives information about tests of the New and the Old levels of Process.
Because specification limits are not saved as column properties in Probe.jmp, JMP calculates a value of the practical difference for each response. The practical difference of 0.15 that you specified is multiplied by an estimate of the 6σ range of the response. This value is used in testing for practical difference and equivalence. It is shown in the Practical Difference column.

The Plain Difference column shows responses whose p-values indicate significance. The Practical Diff PValue and Practical Equiv PValue columns give the p-values for tests of practical difference and practical equivalence. Note that many columns show statistically significant differences, but do not show practically significant differences.

8. Display the Compare Means data table and select Analyze > Distribution.
10. Click OK.

Figure 10.13 shows the distribution of results for practical significance. Only 37 tests are different, as determined by testing for the specified practical difference. For 5 of the responses, the tests were inconclusive. You cannot tell whether the responses result in a practical difference across Process.
The 37 responses can be selected for further study by clicking on the corresponding bar in the plot.

**Example of the MaxLogWorth Option**

When data sets have a large number of observations, $p$-values can be very small. LogWorth values provide a useful way to study $p$-values graphically in these cases. But sometimes $p$-values are so small that the LogWorth scale is distorted by huge values.

1. Open the Probe.jmp sample data table.
2. Select **Analyze > Modeling > Response Screening**.
3. In the Response Screening Launch window, select the Responses column group and click **Y, Response**.
4. Select **Process** and click **X**.
5. Select the **Robust** check box.
6. Click **OK**.

The analysis is numerically intensive and may take some time to complete.

7. In the Response Screening report, open the Robust FDR LogWorth by Effect Size report. The detail in the plot is hard to see, because of the huge Robust FDR LogWorth value of about 58,000 (Figure 10.14). To ensure that your graphs show sufficient detail, you can set a maximum value of the LogWorth.
8. Repeat step 1 through step 5.

9. Type 1000 in the MaxLogWorth box at the bottom of the launch window.

10. Click **OK**.

   The analysis may take some time to complete.

11. In the Response Screening report, open the Robust FDR LogWorth by Effect Size report.

   Now the detail in the plot is apparent (Figure 10.15).
Example of Robust Fit

1. Open the Drosophila Aging.jmp table.
2. Select Analyze > Modeling > Response Screening.
3. Select all of the continuous columns and click Y, Response.
4. Select line and click X.
5. Check Robust.
6. Click OK.

The Robust FDR PValue Plot is shown in Figure 10.16. Note that a number of tests are significant using the unadjusted robust $p$-values, as indicated by the red points that are less than 0.05. However, only two tests are significant according to the robust FDR $p$-values.
These two points are more easily identified in a plot that shows FDR LogWorths.

7. Click the Robust FDR LogWorth by Effect Size disclosure icon.

8. Drag a rectangle around the two points with Robust FDR LogWorth values that exceed 1.5.

9. In the PValues data table, select **Rows > Label/Unlabel**.

The plot shown in Figure 10.17 appears. Points above the red line at 2 have significance levels below 0.01. A horizontal line at about 1.3 corresponds to a 0.05 significance level.
10. Click the Robust LogWorth by LogWorth disclosure icon.

The plot shown in Figure 10.18. If the robust test for a response were identical to the usual test, its corresponding point would fall on the diagonal line in Figure 10.18. The circled point in the plot does not fall near the line, because it has a Robust LogWorth value that exceeds its LogWorth value.
11. Drag a rectangle around this point in the plot.
12. Find the row for this point in the PValues data table.

Note that the response, log2in.CG8237 has PValue 0.9568 and Robust PValue 0.0176.
13. In the Response screening report, select Fit Selected Items from the red triangle menu.

A Fit Selected Items report is displayed containing a Oneway Analysis for the response log2in.CG8237. The plot shows two outliers for the ORE line (Figure 10.19). These outliers indicate why the robust test and the usual test give disparate results. The outliers inflate the error variance for the non-robust test, which makes it more difficult to see a significant effect. In contrast, the robust fit down-weights these outliers, thereby reducing their contribution to the error variance.

Figure 10.19  Oneway Analysis for log2in.CG8237

Response Screening Personality

The Response Screening personality in Fit Model allows you to study tests of multiple responses against linear model effects. This example analyses a model with two main effects and an interaction.

1. Open the Drosophila Aging.jmp table.
2. Select Analyze > Fit Model.
3. Select all the continuous columns and click Y.
4. Select channel and click Add.
5. Select sex, line, and age and select Macros > Full Factorial.
6. Select Response Screening from the Personality list.
7. Click Run.
The Fit Response Screening report appears. Two data tables are also presented: Y Fits summarizes the overall model tests, and PValues tests the individual effects in the model for each Y.

To get a general idea of which effects are important, do the following:

8. Run the FDR LogWorth by Rank Fraction script in the PValues data table.

9. Select Rows > Data Filter.

10. In the Data Filter window, select Effect and click Add.

11. In the Data Filter, click through the list of the model effects, while you view the selected points in the FDR LogWorth by Rank Fraction plot.

Keep in mind that values of LogWorth that exceed 2 are significant at the 0.01 level. The Data Filter helps you see that, with the exception of sex and channel, the model effects are rarely significant at the 0.01 level. Figure 10.20 shows a reference line at 2. The points for tests of the line*age interaction effect are selected. None of these are significant at the 0.01 level.

**Figure 10.20** FDR LogWorth vs Rank Fraction Plot with line*age Tests Selected
Statistical Details

The False Discovery Rate

All of the Response Screening plots involve p-values for tests conducted using the FDR technique described in Benjamini and Hochberg, 1995. JMP uses the following procedure to control the false discovery rate at level \( \alpha \):

1. Conduct the usual hypothesis tests of interest, to obtain p-values \( p_1, p_2, \ldots, p_m \).
2. Rank the p-values from smallest to largest. Denote these by \( p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)} \).
3. Find the largest \( p \)-value for which \( p_{(i)} \leq (i/m)\alpha \). Suppose this first \( p \)-value is the \( k^{th} \) largest, \( p_{(k)} \).
4. Reject the \( k \) hypotheses associated with \( p \)-values less than or equal to \( p_{(k)} \).

This procedure ensures that the expected false discovery rate does not exceed \( \alpha \).

The \( p \)-values adjusted for the false discovery rate, denoted \( p_{(i), FDR} \), are computed as:

\[
p_{(i), FDR} = \text{Minimum}[p_{(i+1)}(m/i)p_{(i)}]
\]

If a hypothesis has an FDR-adjusted \( p \)-value that falls below \( \alpha \), then it is rejected by the procedure.
Appendix A

References


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.
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” on page 242 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.

\[ \log \left( \frac{P(y = j)}{P(y = r)} \right) = X\beta_j \text{ for } j = 1, \ldots, r - 1 \]

**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 $nj$, 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

$$P(y \leq j) = F(\alpha_j + X\beta) \text{ for } j = 1, \ldots, r-1$$

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
Statistical Details

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

$$y_i = \beta_0 + \beta_1 x_{1i} + ... + \beta_k x_{ki} + \epsilon_i$$

where

- $y_i$ is the response
- $x_{ij}$ are functions of the data
- $\epsilon_i$ 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>
<thead>
<tr>
<th>Table B.1 Nominal Factor A</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GLM</td>
<td>JMP</td>
</tr>
<tr>
<td>A</td>
<td>A1</td>
<td>A2</td>
</tr>
<tr>
<td>A1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>A3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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 B.2 Interpreting Parameters**

<table>
<thead>
<tr>
<th>JMP Parameter Report</th>
<th>How to Interpret</th>
<th>Design Column Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>mean over all levels</td>
<td>1'</td>
</tr>
<tr>
<td>A[1]</td>
<td>$\alpha_1 - 1/3(\alpha_1 + \alpha_2 + \alpha_3)$</td>
<td>(A==1) – (A==3)</td>
</tr>
<tr>
<td>A[2]</td>
<td>$\alpha_2 - 1/3(\alpha_1 + \alpha_2 + \alpha_3)$</td>
<td>(A==2) – (A==3)</td>
</tr>
</tbody>
</table>

**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

```plaintext
PROC GLM;
   CLASS A B;
   MODEL A B A*B;
```

yields this design matrix:

**Table B.3 Design Matrix**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>AB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0 1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0 2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0 3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0 4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0 5</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0 6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0 7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0 8</td>
</tr>
</tbody>
</table>
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 B.3** Design Matrix  
(Continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

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 B.5 Three-by-Three Crossed Model**

<table>
<thead>
<tr>
<th></th>
<th>B1</th>
<th></th>
<th></th>
<th>B2</th>
<th></th>
<th></th>
<th>B3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>μ + α₁ + β₁ + αβ₁₁</td>
<td>μ + α₂ + β₁ + αβ₁₂</td>
<td>μ + α₁ - β₁ - β₂ - αβ₁₁ - αβ₁₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>μ + α₂ + β₁ + αβ₂₁</td>
<td>μ + α₂ + β₂ + αβ₂₂</td>
<td>μ + α₂ - β₁ - β₂ - αβ₂₁ - αβ₂₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>μ - α₁ - α₂ + β₁</td>
<td>μ - α₁ - α₂ + β₂ - αβ₁₂</td>
<td>μ + α₁ - α₂ - β₁ - β₂ - αβ₁₁ - αβ₁₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>- αβ₁₁ + αβ₂₁</td>
<td>- αβ₂₂</td>
<td>αβ₁₂ + αβ₂₁ + αβ₂₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 B.6 Nested Effects**

| A | B | A₁ B₁ A₂ B₂ A₁ B₁ A₂ B₂ A₁ B₁ A₂ B₂ A₁ B₁ A₂ B₂ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | |
| A₂ | B₃ | 0 | 1 | 0 | 0 | -1 | -1 | 0 | 0 | |
| A₃ | B₁ | -1 | -1 | 0 | 0 | 0 | 0 | 1 | 0 | |
Appendix B
Specialized Models

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” on page 239. 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.

Table B.6 Nested Effects (Continued)

<table>
<thead>
<tr>
<th>A3</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table B.6 Nested Effects (Continued)
• 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 B.7 Two-Way Model with Interaction |
|-----------------|-----|-----|-----|-----|
| A B | A1 | A2 | B1 | A1B1 | A2B1 |
| A1 B1 | 1 | 0 | 1 | 1 | 0 |
| A2 B1 | 0 | 1 | 1 | 0 | 1 |
| A3 B1 | -1 | -1 | 1 | -1 | -1 |
| A1 B2 | 1 | 0 | -1 | -1 | 0 |
Statistical Details
The Factor Models

Table B.7 Two-Way Model with Interaction (Continued)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A1</th>
<th>A2</th>
<th>B1</th>
<th>A1B1</th>
<th>A2B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2</td>
<td>B2</td>
<td>0</td>
<td>1</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>A3</td>
<td>B2</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Suppose this interaction is missing.

The expected values for each cell are:

Table B.8 Expected Values

<table>
<thead>
<tr>
<th></th>
<th>B1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(\mu + \alpha_1 + \beta_1 + \alpha\beta_{11})</td>
<td>(\mu + \alpha_1 - \beta_1 - \alpha\beta_{11})</td>
</tr>
<tr>
<td>A2</td>
<td>(\mu + \alpha_2 + \beta_1 + \alpha\beta_{21})</td>
<td>(\mu + \alpha_2 - \beta_1 - \alpha\beta_{21})</td>
</tr>
<tr>
<td>A3</td>
<td>(\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21})</td>
<td>(\mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21})</td>
</tr>
</tbody>
</table>

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.

\[\text{LSM}(B1) = (1/3)(\mu + \alpha_1 + \beta_1 + \alpha\beta_{11} + \\
\mu + \alpha_2 + \beta_1 + \alpha\beta_{21} + \\
\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha\beta_{11} - \alpha\beta_{21}) = \mu + \beta_1\]

\[\text{LSM}(B2) = (1/3)(\mu + \alpha_1 - \beta_1 - \alpha\beta_{11} + \\
\mu + \alpha_2 - \beta_1 - \alpha\beta_{21} + \\
\mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha\beta_{11} + \alpha\beta_{21}) = \mu - \beta_1\]

\[\text{LSM}(B1) - \text{LSM}(B2) = 2\beta_1\]

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.
LSM(A1) = \(\frac{1}{2}(\mu + \alpha_1 + \beta_1 + \alpha \beta_{11} + \mu + \alpha_1 - \beta_1 - \alpha \beta_{11})\) 
= \(\mu + \alpha_1\)

LSM(A2) = \(\frac{1}{2}(\mu + \alpha_2 + \beta_1 + \alpha \beta_{21} + \mu + \alpha_2 - \beta_1 - \alpha \beta_{21})\) 
= \(\mu + \alpha_2\)

LSM(A3) = \(\frac{1}{2}(\mu - \alpha_1 - \alpha_2 + \beta_1 - \alpha \beta_{11} - \alpha \beta_{21} + \mu - \alpha_1 - \alpha_2 - \beta_1 + \alpha \beta_{11} + \alpha \beta_{21})\) 
= \(\mu - \alpha_1 - \alpha_2\)

LSM(A1) – LSM(A3) = 2\(\alpha_1 + \alpha_2\)

LSM(A2) – 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.

LSM(A1) – 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 B.9 Verification**

<table>
<thead>
<tr>
<th>parameters</th>
<th>singularity combination</th>
<th>to be tested</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a_1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>a_2</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>b_1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>ab_{11}</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ab_{21}</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

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 B.10 on page 234, 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>
<thead>
<tr>
<th>Parameter</th>
<th>JMP Test for A</th>
<th>GLM-IV Test for A</th>
<th>JMP Rotated Test</th>
<th>GLM-III Test for A</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a1</td>
<td>0.6667</td>
<td>-0.3333</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a2</td>
<td>-0.3333</td>
<td>0.6667</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>a3</td>
<td>-0.3333</td>
<td>-0.3333</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a1b1</td>
<td>0.1667</td>
<td>-0.0833</td>
<td>0.2222</td>
<td>0.25</td>
</tr>
<tr>
<td>a1b2</td>
<td>0.1667</td>
<td>-0.0833</td>
<td>0.3333</td>
<td>0.25</td>
</tr>
<tr>
<td>a1b3</td>
<td>0.1667</td>
<td>-0.0833</td>
<td>0.2222</td>
<td>0.25</td>
</tr>
<tr>
<td>a1b4</td>
<td>0.1667</td>
<td>-0.0833</td>
<td>0.2222</td>
<td>0.25</td>
</tr>
<tr>
<td>a2b1</td>
<td>-0.1667</td>
<td>0.3333</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>a2b2</td>
<td>-0.1667</td>
<td>0.3333</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>a3b1</td>
<td>-0.1111</td>
<td>-0.1111</td>
<td>-0.3333</td>
<td>-0.3333</td>
</tr>
<tr>
<td>a3b2</td>
<td>-0.1111 -0.1111</td>
<td>-0.3333 -0.3333</td>
<td>-0.3333 -0.3333</td>
<td>-0.3333 -0.3333</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>a3b3</td>
<td>-0.1111 -0.1111</td>
<td>-0.3333 -0.3333</td>
<td>-0.3333 -0.3333</td>
<td>-0.3333 -0.3333</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a1b1c1</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a1b1c2</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a1b2c1</td>
<td>0.0556 -0.0278</td>
<td>0.1111 0</td>
<td>0.0833 0</td>
<td>0.0909 0</td>
</tr>
<tr>
<td>a1b2c2</td>
<td>0.0556 -0.0278</td>
<td>0.1111 0</td>
<td>0.0833 0</td>
<td>0.0909 0</td>
</tr>
<tr>
<td>a1b2c3</td>
<td>0.0556 -0.0278</td>
<td>0.1111 0</td>
<td>0.0833 0</td>
<td>0.0909 0</td>
</tr>
<tr>
<td>a1b3c1</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a1b3c2</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a1b4c1</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a1b4c2</td>
<td>0.0833 -0.0417</td>
<td>0.1111 0</td>
<td>0.125 0</td>
<td>0.1212 0</td>
</tr>
<tr>
<td>a2b1c1</td>
<td>-0.0833 0.1667</td>
<td>0 0.25</td>
<td>0 0.25</td>
<td>0 0.25</td>
</tr>
<tr>
<td>a2b1c2</td>
<td>-0.0833 0.1667</td>
<td>0 0.25</td>
<td>0 0.25</td>
<td>0 0.25</td>
</tr>
<tr>
<td>a2b2c1</td>
<td>-0.0833 0.1667</td>
<td>0 0.25</td>
<td>0 0.25</td>
<td>0 0.25</td>
</tr>
<tr>
<td>a2b2c2</td>
<td>-0.0833 0.1667</td>
<td>0 0.25</td>
<td>0 0.25</td>
<td>0 0.25</td>
</tr>
<tr>
<td>a3b1c1</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
<tr>
<td>a3b1c2</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
<tr>
<td>a3b2c1</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
<tr>
<td>a3b2c2</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
<tr>
<td>a3b3c1</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
<tr>
<td>a3b3c2</td>
<td>-0.0556 -0.0556</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
<td>-0.1667 -0.1667</td>
</tr>
</tbody>
</table>
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 B.11  Ordinal Factors

<table>
<thead>
<tr>
<th>Term</th>
<th>Coded Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a2</td>
</tr>
<tr>
<td>A1</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>1</td>
</tr>
<tr>
<td>A3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>a3</td>
</tr>
<tr>
<td>control level, zero dose</td>
</tr>
<tr>
<td>low dose</td>
</tr>
<tr>
<td>higher dose</td>
</tr>
</tbody>
</table>

From the perspective of the JMP parameterization, the tests for A are:

Table B.12  Tests for A

<table>
<thead>
<tr>
<th>parameter</th>
<th>GLM–IV test</th>
<th>JMP test</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a13</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>a23</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a1:b14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a1:b24</td>
<td>0.11111</td>
<td>0</td>
</tr>
<tr>
<td>a1:b34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a2:b12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a3:b13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a3:b23</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>a1b1:c12</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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 B.13 Ordinal Interactions

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A2</th>
<th>A3</th>
<th>B2</th>
<th>B3</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>B1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A1</td>
<td>B2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A1</td>
<td>B3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>B1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>B2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>B3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A3</td>
<td>B1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A3</td>
<td>B2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A3</td>
<td>B3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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 B.14 Expected Cell Means

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>( \mu )</td>
<td>( \mu + \alpha_2 \beta_2 + \alpha_3 \beta_{12} )</td>
</tr>
<tr>
<td>A2</td>
<td>( \mu + \alpha_2 )</td>
<td>( \mu + \alpha_2 \beta_2 + \alpha_3 \beta_{22} )</td>
</tr>
<tr>
<td>A3</td>
<td>( \mu + \alpha_2 + \alpha_3 )</td>
<td>( \mu + \alpha_2 + \alpha_3 + \beta_2 + \alpha_2 \beta_{22} + \alpha_3 \beta_{32} + \alpha_3 \beta_{33} )</td>
</tr>
</tbody>
</table>
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 B.1.

<table>
<thead>
<tr>
<th>Y</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
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.

The singularity details show the linear dependencies (and also identify the missing cell by examining the values).
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.

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 B.5 shows least squares means for an nominal and ordinal fits.
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, 1981). 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 B.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.
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 B.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 B.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 \( \frac{6}{13}, \frac{2}{13}, \text{and} \frac{5}{13} \), respectively, as the pressure against the response partitions balances out to minimize the total energy.

**Figure B.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 B.8 shows these partitions.
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 $E_{n-1}$ 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 B.16** Computations for Multivariate Tests

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wilks’ Lambda</td>
<td>$\Lambda = \frac{\text{det}(E)}{\text{det}(H+E)} = \prod_{i=1}^{n} \left( \frac{1}{1+\lambda_i} \right)$</td>
</tr>
<tr>
<td>Pillai’s Trace</td>
<td>$V = \text{Trace}[H(H+E)^{-1}] = \sum_{i=1}^{n} \frac{\lambda_i}{1+\lambda_i}$</td>
</tr>
<tr>
<td>Hotelling-Lawley Trace</td>
<td>$U = \text{Trace}(E^{-1}H) = \sum_{i=1}^{n} \lambda_i$</td>
</tr>
<tr>
<td>Roy’s Max Root</td>
<td>$\Theta = \lambda_1$, the maximum eigenvalue of $E^{-1}H$.</td>
</tr>
</tbody>
</table>
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 B.17 on page 249, gives the computation of each approximate $F$ from the corresponding test statistic.

<table>
<thead>
<tr>
<th>Test</th>
<th>Approximate $F$</th>
<th>Numerator DF</th>
<th>Denominator DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wilks’ Lambda</td>
<td>$F = \left(1 - \Lambda^{1/t}\right)\left(\frac{rt - 2u}{pq}\right)$</td>
<td>$pq$</td>
<td>$rt - 2u$</td>
</tr>
<tr>
<td>Pillai’s Trace</td>
<td>$F = \left(\frac{V}{s - V}\right)\left(\frac{2n + s + 1}{2m + s + 1}\right)$</td>
<td>$s(2m + s + 1)$</td>
<td>$s(2n + s + 1)$</td>
</tr>
<tr>
<td>Hotelling-Lawley Trace</td>
<td>$F = \frac{2(sn + 1)L}{s^2(2m + s + 1)}$</td>
<td>$s(2m + s + 1)$</td>
<td>$2(sn + 1)$</td>
</tr>
<tr>
<td>Roy’s Max Root</td>
<td>$F = \frac{s\left(v - \max(p, q) + q\right)}{\max(p, q)}$</td>
<td>$\max(p, q)$</td>
<td>$v - \max(p, q) + q$</td>
</tr>
</tbody>
</table>

**Canonical Details**

The canonical correlations are computed as

$$\rho_i = \frac{\lambda_i}{1 + \lambda_i}$$

The canonical Y’s are calculated as

$$\hat{Y} =YMV$$

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

\[ \text{Grand} = \bar{y} \mathbf{MV} \]

where \( \mathbf{V} \) is the matrix of eigenvectors of \( \mathbf{E}^{-1} \mathbf{H} \).

The centroid values for effects are calculated as

\[ \mathbf{m} = (c_1' \bar{x}_j, c_2' \bar{x}_j, \ldots, c_g' \bar{x}_j) \text{ where } \mathbf{c}_i = \left(v_i' \left( \mathbf{E} \left( \mathbf{N} - \mathbf{r} \right) \right)^{-1/2} v_i \right) \]

and the \( v_i \)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

\[ d = \sqrt{\frac{(\chi^2_g(0.95))}{\mathbf{L}'(\mathbf{XX})^{-1}\mathbf{L}}} \]

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

\[ D^2 = (y - \bar{y}_i)' \mathbf{S}^{-1} (y - \bar{y}_i) = y' \mathbf{S}^{-1} y - 2y' \mathbf{S}^{-1} \bar{y}_i + \bar{y}_i' \mathbf{S}^{-1} \bar{y}_i \]

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

\[ \text{SqDist}[0] = y' \mathbf{S}^{-1} y \]

and

\[ \text{SqDist}[i] = \text{SqDist}[0] - 2y' \mathbf{S}^{-1} \bar{y}_i + \bar{y}_i' \mathbf{S}^{-1} \bar{y}_i \]
Appendix B
Specialized Models

Statistical Details

Power Calculations

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 the least significant number (LSN), least significant value (LSV), power, and adjusted power. With the exception of LSV, these computations are provided for each effect, and for a collection of user-specified contrasts (under Custom Test and LS Means Contrast. LSV is only computed for a single linear contrast. In the details below, the hypothesis refers to the collection of contrasts of interest.

#### Computations for the LSN

The LSN solves for $N$ in the equation:

$$\alpha = 1 - FDist\left[\frac{N\delta^2}{df_{Hyp}\sigma^2}, df_{Hyp}, N - df_{Hyp} - 1\right]$$

where

* $FDist$ is the cumulative distribution function of the central $F$ distribution
* $df_{Hyp}$ represents the degrees of freedom for the hypothesis
* $\sigma^2$ is the error variance
* $\delta^2$ is the squared effect size

For retrospective analyses, $\delta^2$ is estimated by the sum of squares for the hypothesis divided by $n$, the size of the current sample. If the test is for an effect, then $\delta^2$ is estimated by the sum of squares for that effect divided by the number of observations in the current study. For retrospective studies, the error variance $\sigma^2$ is estimated by the mean square error. These estimates, along with an $\alpha$ value of 0.05, are entered into the Power Details window as default values.

When you are conducting a prospective analysis to plan a future study, consider determining the sample size that will achieve a specified power (see “Computations for the Power” on page 253.)
Computations for the LSV

The LSV is only computed for a single linear contrast.

Test of a Single Linear Contrast

Consider the one-degree-freedom test \( L\beta = 0 \), where \( L \) is a row vector of constants. The test statistic for a \( t \)-test for this hypothesis is:

\[
\frac{Lb}{s\sqrt{L(X'X)^{-1}L'}}
\]

where \( s \) is the root mean square error. We reject the hypothesis at significance level \( \alpha \) if the absolute value of the test statistic exceeds the \( 1 - \alpha/2 \) quantile of the \( t \) distribution, \( t_{1 - \alpha/2} \), with degrees of freedom equal to those for error.

To find the least significant value, denoted \( (Lb)^{LSV} \), we solve for \( Lb \):

\[
(Lb)^{LSV} = t_{1 - \alpha/2}s\sqrt{L(X'X)^{-1}L'}
\]

Test of a Single Parameter

In the special case where the linear contrast tests a hypothesis setting a single \( \beta_i \) equal to 0, this reduces to:

\[
b_i^{LSV} = t_{1 - \alpha/2}s\sqrt{(X'X)_{ii}} = t_{1 - \alpha/2}\text{StdError}(b_i)
\]

Test of a Difference in Means

In a situation where the test of interest is a comparison of two group means, the literature talks about the least significant difference (LSD). In the special case where the model contains only one nominal variable, the formula for testing a single linear contrast reduces to the formula for the LSD:

\[
\text{LSD} = t_{1 - \alpha/2}s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}
\]

However, in JMP, the parameter associated with a level for a nominal effect measures the difference between the mean of that level and the mean for all levels. So, the LSV for such a comparison is half the LSD for the differences of the means.

Note: If you are testing a contrast across the levels of a nominal effect, keep in mind how JMP codes nominal effects. Namely, the parameter associated with a given level measures the difference to the average for all levels.
Computations for the Power

Suppose that you are interested in computing the power of a test of a linear hypothesis, based on significance level $\alpha$ and a sample size of $N$. You want to detect an effect of size $\delta$.

To calculate the power, begin by finding the critical value for an $\alpha$-level F-test of the linear hypothesis. This is given by solving for $F_C$ in the equation

$$\alpha = 1 - F_{\text{Dist}}[F_{C}, df_{Hyp} - df_{Model} - 1]$$

Here, $df_{Hyp}$ represents the degrees of freedom for the hypothesis, $df_{Model}$ represents the degrees of freedom for the model, and $N$ is the proposed (or actual) sample size.

Then calculate the noncentrality parameter associated with the desired effect size. The noncentrality parameter is given by:

$$\lambda = \frac{(N\delta^2)}{\sigma^2}$$

where $\sigma^2$ is a proposed (or estimated) value of the error variance.

Given an effect of size $\delta$, the test statistic has a noncentral F distribution, with distribution function denoted $F_{\text{Dist}}$ below, with noncentrality parameter $\lambda$. To obtain the power of your test, calculate the probability that the test statistic exceeds the critical value:

$$\text{Power} = 1 - F_{\text{Dist}}[F_C, df_{Hyp} - df_{Model} - 1, \frac{N\delta^2}{\sigma^2}]$$

In obtaining retrospective power for a study with $n$ observations, JMP estimates the noncentrality parameter $\lambda = (n\delta^2)/\sigma^2$ by $\hat{\lambda} = \frac{SS_{Hyp}}{\sigma^2}$, where $SS_{Hyp}$ represents the sum of squares due to the hypothesis.

Computations for the Adjusted Power

The adjusted power calculation (Wright and O’Brien, 1988) is only relevant for retrospective power analysis. Adjusted power calculates power using a noncentrality parameter estimate that has been adjusted to remove the positive bias that occurs when parameters are simply replaced by their sample estimates.

The estimate of the noncentrality parameter, $\hat{\lambda}$, obtained by estimating $\delta$ and $\sigma$ by their sample estimates is

$$\hat{\lambda} = \frac{SS_{Hyp}}{MSE}$$

Wright and O’Brien (1988) explain that an unbiased estimate of the noncentrality parameter is given by
The expression on the right illustrates the calculation of the unbiased noncentrality parameter when a sample size $N$, different from the study size $n$, is proposed for a retrospective power analysis. Here, $df_{Hyp}$ represents the degrees of freedom for the hypothesis and $df_{Model}$ represents the degrees of freedom for the whole model.

Unfortunately, this adjustment to the noncentrality estimate can lead to negative values. Negative values are set to zero, reintroducing some slight bias. The adjusted noncentrality estimate is

$$\hat{\lambda}_{adj} = \max\left[0, \frac{\hat{\lambda}(N - df_{Model} - 1 - 2)}{N - df_{Model} - 1} - df_{Hyp}\right]$$

The adjusted power is

$$\text{Power}_{adj} = 1 - FDist\left[F_C, df_{Hyp}N - df_{Model} - 1, \hat{\lambda}_{adj}\right]$$

Confidence limits for the noncentrality parameter are constructed as described in Dwass (1955):

Lower CL for $\lambda = \max\left[0, \left(\frac{SS_{Hyp}}{MSE} - \frac{df_{Hyp}F_C}{\nu}\right)^2\right]$ 

Upper CL for $\lambda = \left(\frac{SS_{Hyp}}{MSE} - \frac{df_{Hyp}F_C}{\nu}\right)^2$

Confidence limits for the power are obtained by substituting these confidence limits for $\lambda$ into

$$\text{Power} = 1 - FDist\left[F_C, df_{Hyp}N - df_{Model} - 1, \lambda\right]$$

**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 (Fieller, 1954), 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 $b$ estimate the parameters $\beta$ so that we have $b$ distributed as $N(\beta, V)$. 

Appendix B
Specialized Models

Statistical Details
Inverse Prediction with Confidence Limits

- Let \( x \) 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 \( x[i] \) such that \( \beta'x = y \) with all other values of \( x \) given.

The inverse prediction is

\[
x[i] = \frac{y - \beta'(i)x(i)}{\beta[i]}
\]

where the parenthesized subscript \( (i) \) indicates that the \( i^{th} \) component is omitted. A confidence interval can be formed from the relation

\[
(y - b'x)^2 < t^2x'Vx
\]

where \( t \) is the \( t \) value for the specified confidence level.

The equation

\[
(y - b'x)^2 - t^2x'Vx = 0
\]

can be written as a quadratic in terms of \( z = x[i] \):

\[
gz^2 + hz + f = 0
\]

where

\[
g = b[i]^2 - t^2V[i, i]
\]
\[
h = -2yb[i] + 2b[i]b'(i)x(i) - 2t^2V[i, (i)]x(i)
\]
\[
f = y^2 - 2yb'(i)x(i) + (b'(i)x(i))^2 - t^2x'(i)V(i)x(i)
\]

Depending on the values of \( g, h, \) and \( f \), the set of values satisfying the inequality, and hence the confidence interval for the inverse prediction, can have a number of forms:

- an interval of the form \((\phi_1, \phi_2)\), where \( \phi_1 < \phi_2 \)
- two disjoint intervals of the form \((-\infty, \phi_1) \cup (\phi_2, \infty)\), where \( \phi_1 < \phi_2 \)
- the entire real line, \((-\infty, \infty)\)
- only one of \((-\infty, \phi)\) or \((\phi, \infty)\)

When the Fieller confidence interval is the entire real line, Wald intervals are presented.
**Note:** The Fit Y by X logistic platform and the Fit Model Nominal Logistic personalities use $t$ values when computing confidence intervals for inverse prediction. The Fit Model Generalized Linear Model personality, as well as PROC PROBIT in SAS/STAT. Use $z$ values, which give different results.
Index

Specialized Models

Symbols
^, redundant leaf labels 44

Numerics
2D Gaussian Process Example.jmp 145
–2LogLikelihood 165

A
AAE 91
AAE, model comparison 91
Accept Current Estimates 124
Actual by Predicted Plot 54
ADF tests 173
Adjusted Power and Confidence Interval 251
AIC 165
Akaike’s Information Criterion 165
algorithms 219
alternative methods 243
Angular Frequency 161
Annual 155
approximate F test 249
ApproxStdErr 122
AR Coefficients 153, 159
ARIMA 153, 168–170, 180
assess validity 243
assumptions 242–244
AUC Comparison 92
Augmented Dickey-Fuller test 173
Autocorrelation 157, 162
autocorrelation 156, 158
Autocorrelation Lags 155
autoregressive coefficients see AR coefficients
Autoregressive Order 169

B
Bartlett’s Kolmogorov-Smirnov 160
BIC 165
Borehole Latin Hypercube.jmp 150
Boston Housing.jmp 87, 93
Bounded 181
Box-Jenkins model see ARIMA
Brown smoothing 182
By variable 192

calculation details 219
canonical correlation 249
centroid 250
Close All Below (Partition Platform) 41
closing JMP Starter window 25
Color Points 45
Column Contributions 43
Compare Parameter Estimates 112
comparing models 90
computational details 219
Confidence Intervals 169, 181
Confidence Limits 122, 137
Connecting Lines 157
constant estimate 166
Constrain fit 169
continuous response model 221
Contour Profiler 125
contrast M matrix 248
Corr option 192
Correlation Type 145
Cosine 161
Create SAS Job 168
Cross Correlation 173
Cubic 145
Custom 181
custom loss function 139
fitting principles 221–224
Fixed 181
Force X Categorical option 192
Force X Continuous option 192
Force Y Categorical option 192
Force Y Continuous option 192
forecast 153, 161–184
Forecast Periods 155, 167
Forecast plot 167
formulas used in JMP calculations 219
Frequency 161

G
Gaussian 145
Gaussian Process 143
Gaussian Process 145
Gauss-Newton method 137, 140
Generalized RSquare 91
Go 45, 125, 137
goal SSE 138
Graph 157, 162
Grouping 191

H
H matrix 248
Hessian 139
Holt smoothing 183
Hourly 155
hypothesis 246
hypothesis test 229–235, 238

I
Informative Missing 59
Ingots2.jmp 135
interaction effect 226
Intercept 169
intercept 166
inverse prediction 254
Invertible 165
Iter 168
Iteration History 168
Iteration History report 168, 222
Iteration Log 124
Index
Specialized Models

J
JMP Starter 25
JMP tutorials 23

K
K Fold Crossvalidation 43
Kappa option 192
key concepts 244
Kruskal-Wallis 243

L
L matrix 248
Lag 166
Leaf Report 43
least significant difference 252
and LSV 252
least significant number 251
least significant value 251
least squares fit
   introduction 221
least squares means 229, 239
level smoothing weight 180
Lift Curve 57, 93
Lift Curves 57
likelihood confidence intervals 138
limitations of techniques 242–244
linear dependence 230–233
linear exponential smoothing 183
linear rank tests 243
Lock 41
Lock Columns
   Partition platform 42
Logistic platform
   example 131
Logistic w Loss.jmp 131
Loss 132–133
loss function
   custom 139
Lower CL and Upper CL 122
LSN 251
LSV 251

N
Negative Exponential.jmp 140
negative log-likelihood 245
nested effect 228, 234
Newton 124
Newton-Raphson method 137, 140
nominal factor 225, 229–233
nominal response model 221–223
nonestimable 225
nonlinear fit options 124
Nonlinear Model Library 127
Customizing 130
Nonlinear platform 132–133
derivatives 140–142
Fit Curve 106
Nonlinear platform, built-in models 97
Nonlinear platform, custom models 115–142
nonstationary time series 161
normal distribution 246
Nugget parameters 148
Number of Forecast Periods 161
Number of Points 125
NumDeriv 141
Numeric Derivatives Only 120, 124, 135

O
Obj-Criterion 168
opening
JMP Starter window 25
Optimal Value 41
ordinal crossed model 238
ordinal factor 239
ordinal interaction 237
ordinal least squares means 239
ordinal response model 221, 223–224
Output Split Table 41

P
p, d, q parameters 169
Paired X and Y option 192
Parameter 122, 125
Parameter Contour Profiler 125
Parameter Estimates table 166
parameter interpretation 226
Parameter Profiler 125
Parameter Surface Profiler 125
Partial Autocorrelation 157, 162
partial autocorrelation 156
Partition 29
Informative Missing 59
Partition Platform 29–39
Period 160
periodicity 170
Periodogram 161
periodogram 160
Periods Per Season 170, 181
Plot Actual by Predicted 92
Plot Actual by Predicted (Partition) 42
Plot Residual by Row 92
Poisson loss function 136–137
Poisson Y option 192
Power Analysis 251
Practical Difference Portion setting 192
Practical Significance 187
prediction formula 224–238
Predictor role 132–133, 136
Predictors report 91
pressure cylinders fitting machine 246
Prob>|t| 166
probit example 135–136
process disturbance 153
profile confidence limits 138–139
profile likelihood confidence intervals 138
Profiler 125
Prune Below 41
Prune Worst 41–42
PValues data table 195, 204

Q
QuasiNewton BFGS 124
QuasiNewton SR1 124

R
RASE 91
RASE, model comparison 91
relative significance 242
Remember Solution 122
Remove 155
Residual Statistics 168
residuals 167, 221
response models 221–224
Response Screening
equivalence tests 187
False Discovery Rate 187
FDR 187
Practical Significance 187
Response Screening platform
By variable 192
Corr option 192
Index

Specialized Models

Force X Categorical option 192
Force X Continuous option 192
Force Y Categorical option 192
Force Y Continuous option 192
Grouping variable 191
Kappa option 192
MaxLogWorth setting 193
Missing is category option 192
options 198
Paired X and Y option 192
Poisson Y option 192
Practical Difference Portion setting 192
PValues data table 195, 204
Robust option 192
Same Y Scale option 192
Unthreaded option 192
Weight variable 191
X variable 191
Y, Response 191

rho see loss function
RMSE 91
RMSE 122
RMSE, model comparison 91
Robust option 192
ROC Curve 55, 92
ROC Curve 55
ROC Curves 54
R-Square 91
RSquare 165
RSquare Adj 165

S
Same Y Scale option 192
sample autocorrelation function 157
SAS GLM procedure 219, 221–235
Save 162, 243
Save Columns 168
Save Cumulative Details 52
Save Discrim 250
Save Inverse Prediction Formula 126
Save Leaf Label Formula 44
Save Leaf Labels 44
Save Leaf Number Formula 44
Save Leaf Numbers 44
Save Offset Estimates 52
Save Predicted 44, 49, 52
Save Prediction Formula 44, 49, 52, 126
Save Residual Formula 126
Save Residuals 44, 49, 52
Save Specific Solving Formula 126
Save Spectral Density 160
Save Std Error of Individual 126
Save Std Error of Predicted 126
Save Tolerant Prediction Formula 44
Save Tree Details 52
SBC 165
Schwartz’s Bayesian Criterion 165
Seasonal ARIMA 153
Seasonal ARIMA 170
seasonal exponential smoothing 183
seasonal smoothing weight 180
Second 155
Second Deriv Method 119
Second Deriv. Method 137, 140
Second Derivatives 119
Select Rows 41
seriesg.jmp 155
seriesJ.jmp 171
Ship Damage.jmp 136
Show Confidence Interval 167–168
Show Derivatives 124, 140
Show Graph 42
Show Points 42, 129, 157, 167–168
Show Split Bar 42
Show Split Candidates 42
Show Split Count 42
Show Split Prob 42
Show Split Stats 42
Show Tree 42
simple exponential smoothing 182
Sine 161
singularity 230–233, 239
Small Tree View 42
Smoothing Model dialog 181
smoothing models 153, 179–184
smoothing weight 180
Solution table ??–138
Sort Split Candidates 42
Specified Value 41
Specify Profit Matrix 42
Spectral Density 159, 161
spectral density plots 153
Split Best  40, 42
Split Here  41
Split History  43
Split Specific  41
spring fitting machine  245–246
SSE  122
Stable  165
Stable Invertible  181
Standard Deviation  165
statistical details  219
Std Error  166
Step  168
Submit to SAS  168
Sum of Squared Errors  164
Summarize YbyX  188
Surface Profiler  125
symbolic derivative  140

T
t Ratio  166
Term  166
Test Parallelism  110
Time Frequency  155
Time ID role  155
Time Series Graph  157
Time Series platform  153, 155–184
 ARIMA  168–170
 commands  156–162
 example  155–156
 launch  155–156
 modeling report  162–168
 Seasonal ARIMA  170
 smoothing models  179–184
 Time Series Plot  156
 Time Series role  155
tooltips  24
 Transfer Function  176
 Transfer Function Models  171
 Transfer Function Models  153
trend  180
tutorial examples
 logistic regression  131
 probit  135–136
 time series  155–156
tutorials  23
 Types III and IV hypotheses  229

U
uncertainty  244
Unconstrained  181
Unthreaded  124
Unthreaded option  192
usual assumptions  242–244

V
validity  243
Variance Estimate  164
Variogram  153, 158, 162

W-Z
Weekly  155
Weight  191
Wilcoxon rank-sum  243
Winter’s method  184
X  191
X role  132–133, 136, 155
Y role  155
Y, Response variable  191
Zero To One  181
zeroed  225