Statistical Discovery.™ From SAS.

Version 13

Predictive and Specialized Modeling

Second Edition

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

Marcel Proust

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

13.2

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *JMP® 13 Predictive and Specialized Modeling, Second Edition*. Cary, NC: SAS Institute Inc.

JMP® 13 Predictive and Specialized Modeling, Second Edition

ISBN 978-1-62960-964-5 (Hardcopy)
ISBN 978-1-62960-965-2 (EPUB)
ISBN 978-1-62960-966- (MOBI)
ISBN 978-1-62960-967-6 (Web PDF)

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

February 2017

August 2017

Technology License Notices

  - 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 Florian Reuter 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.

- libxml2 - Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

  Copyright © 1998 - 2003 Daniel Veillard. All Rights Reserved.

  Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

  The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

  THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

  Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

Contents

Specialized and Predictive Modeling

13 Nonlinear Regression

14 Gaussian Process

15 Time Series Analysis

16 Matched Pairs Analysis

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
  - technical support options

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
  - script output
- Features that are for JMP Pro only are noted with the JMP Pro icon JMP PRO. For an overview of JMP Pro features, visit 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.
- 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.
- You can also purchase printed documentation and e-books on the SAS website:

  http://www.sas.com/store/search.ep?keyWords=JMP

JMP Documentation Library

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

Document Title	Document Purpose	Document Content
Discovering JMP	If you are not familiar with JMP, start here.	Introduces you to JMP and gets you started creating and analyzing data.
Using JMP	Learn about JMP data tables and how to perform basic operations.	Covers general JMP concepts and features that span across all of JMP, including importing data, modifying columns properties, sorting data, and connecting to SAS.
Basic Analysis	Perform basic analysis using this document.	Describes these Analyze menu platforms: • Distribution • Fit Y by X • Tabulate • Text Explorer Covers how to perform bivariate, one-way ANOVA, and contingency analyses through Analyze > Fit Y by X. How to approximate sampling distributions using bootstrapping and how to perform parametric resampling with the Simulate platform are also included.

Document Title	Document Purpose	Document Content
Essential Graphing	Find the ideal graph for your data.	Describes these Graph menu platforms: • Graph Builder • Overlay Plot • Scatterplot 3D • Contour Plot • Bubble Plot • Parallel Plot • Cell Plot • Treemap • Scatterplot Matrix • Ternary Plot • Chart The book also covers how to create background and custom maps.
Profilers	Learn how to use interactive profiling tools, which enable you to view cross-sections of any response surface.	Covers all profilers listed in the Graph menu. Analyzing noise factors is included along with running simulations using random inputs.
Design of Experiments Guide	Learn how to design experiments and determine appropriate sample sizes.	Covers all topics in the DOE menu and the Specialized DOE Models menu item in the Analyze > Specialized Modeling menu.

Document Title	Document Purpose	Document Content
Fitting Linear Models	Learn about Fit Model platform and many of its personalities.	Describes these personalities, all available within the Analyze menu Fit Model platform: • Standard Least Squares • Stepwise • Generalized Regression • Mixed Model • MANOVA • Loglinear Variance • Nominal Logistic • Ordinal Logistic • Generalized Linear Model

Document Title	Document Purpose	Document Content
Predictive and Specialized Modeling	Learn about additional modeling techniques.	Describes these Analyze > Predictive Modeling menu platforms: • Modeling Utilities • Neural • Partition • Bootstrap Forest • Boosted Tree • K Nearest Neighbors • Naive Bayes • Model Comparison • Formula Depot Describes these Analyze > Specialized Modeling menu platforms: • Fit Curve • Nonlinear • Gaussian Process • Time Series • Matched Pairs Describes these Analyze > Screening menu platforms: • Response Screening • Process Screening • Predictor Screening • Association Analysis The platforms in the Analyze > Specialized Modeling > Specialized DOE Models menu are described in *Design of Experiments Guide*.

Document Title	Document Purpose	Document Content
Multivariate Methods	Read about techniques for analyzing several variables simultaneously.	Describes these Analyze > Multivariate Methods menu platforms: • Multivariate • Principal Components • Discriminant • Partial Least Squares Describes these Analyze > Clustering menu platforms: • Hierarchical Cluster • K Means Cluster • Normal Mixtures • Latent Class Analysis • Cluster Variables
Quality and Process Methods	Read about tools for evaluating and improving processes.	Describes these Analyze > Quality and Process menu platforms: • Control Chart Builder and individual control charts • Measurement Systems Analysis • Variability / Attribute Gauge Charts • Process Capability • Pareto Plot • Diagram

Document Title	Document Purpose	Document Content
Reliability and Survival Methods	Learn to evaluate and improve reliability in a product or system and analyze survival data for people and products.	Describes these Analyze > Reliability and Survival menu platforms: • Life Distribution • Fit Life by X • Cumulative Damage • Recurrence Analysis • Degradation and Destructive Degradation • Reliability Forecast • Reliability Growth • Reliability Block Diagram • Repairable Systems Simulation • Survival • Fit Parametric Survival • Fit Proportional Hazards
Consumer Research	Learn about methods for studying consumer preferences and using that insight to create better products and services.	Describes these Analyze > Consumer Research menu platforms: • Categorical • Multiple Correspondence Analysis • Multidimensional Scaling • Factor Analysis • Choice • MaxDiff • Uplift • Item Analysis
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.

Document Title	Document Purpose	Document Content
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.

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.

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 26)
- Sample data (see "Sample Data Tables" on page 26)
- Indexes (see "Learn about Statistical and JSL Terms" on page 26)
- Tip of the Day (see "Learn JMP Tips and Tricks" on page 26)
- Web resources (see "JMP User Community" on page 27)
- JMPer Cable technical publication (see "JMPer Cable" on page 27)
- Books about JMP (see "JMP Books by Users" on page 28)
- JMP Starter (see "The JMP Starter Window" on page 28)

- Teaching Resources (see "Sample Data Tables" on page 26)

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 designing an experiment and comparing a sample mean to a constant.

Sample Data Tables

All of the examples in the JMP documentation suite use sample data. Select **Help > Sample Data Library** to open the sample data directory.

To view an alphabetized list of sample data tables or view sample data within categories, select **Help > Sample Data.**

Sample data tables are installed in the following directory:

On Windows: C:\Program Files\SAS\JMP\13\Samples\Data

On Macintosh: \Library\Application Support\JMP\13\Samples\Data

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

To view examples using sample data, select **Help > Sample Data** and navigate to the Teaching Resources section. To learn more about the teaching resources, visit http://jmp.com/tools.

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: On Windows, you can hide tooltips in the JMP Preferences. Select **File > Preferences > General** and then deselect **Show menu tips**. This option is not available on Macintosh.

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** or visit https://community.jmp.com/.

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/en_us/software/books.html

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. The window also lists JMP Pro features and platforms.

- 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**.

Technical Support

JMP technical support is provided by statisticians and engineers educated in SAS and JMP, many of whom have graduate degrees in statistics or other technical disciplines.

Many technical support options are provided at http://www.jmp.com/support, including the technical support phone number.

Chapter 2

Introduction to Predictive and Specialized Modeling

Overview of Modeling Techniques

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

- The Modeling Utilities assist in the data cleaning and pre-processing stages of data analysis. Each utility has exploratory tools to give you a more thorough understanding of your data. See Chapter 3, “Modeling Utilities”.
- 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 Partition platform recursively partitions data according to a relationship between the *X* and *Y* values, creating a decision tree of partitions. See Chapter 5, “Partition Models”.
- JMP PRO The Bootstrap Forest platform enables you to fit an ensemble model by averaging many decision trees each of which is fit to a random subset of the training data. See Chapter 6, “Bootstrap Forest”.
- JMP PRO The Boosted Tree platform produces an additive decision tree model that is composed of many smaller decision trees that are constructed in layers. The tree in each layer consists of a small number of splits, typically five or fewer. Each layer is fit using the recursive fitting methodology. See Chapter 7, “Boosted Tree”.
- JMP PRO The K Nearest Neighbors platform predicts a response value for a given observation using the responses of the observations in that observation’s local neighborhood. It can be used with a categorical response for classification and with a continuous response for prediction. See Chapter 8, “K Nearest Neighbors”.
- JMP PRO The Naive Bayes platform classifies observations into groups that are defined by the levels of a categorical response variable. The variables (or factors) that are used for classification are often called features in the data mining literature. See Chapter 9, “Naive Bayes”.
- JMP PRO 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 10, “Model Comparison”.
- JMP PRO The Formula Depot platform enables you to organize, compare, profile, and score models for deployment. For model exploration work, you can use the Formula Depot to store candidate models outside of your JMP data table. See Chapter 11, “Formula Depot”.

- The Fit Curve platform 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 12, "Fit Curve".
- The Nonlinear platform lets you fit custom nonlinear models, which include a model formula and parameters to be estimated. See Chapter 13, "Nonlinear Regression".
- 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 14, "Gaussian Process".
- The Time Series platform lets you explore, analyze, and forecast univariate time series. See Chapter 15, "Time Series Analysis".
- The Matched Pairs platform compares the means between two or more correlated variables and assesses the differences. See Chapter 16, "Matched Pairs 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 17, "Response Screening".
- The Process Screening platform enables you to explore a large number of processes across time. The platform calculates control chart, process stability, and process capability metrics, and detects large process shifts. See Chapter 18, "Process Screening".
- The Predictor Screening platform enables you to screen a data set for significant predictors. See Chapter 19, "Predictor Screening".
- JMP PRO The Association Analysis platform enables you to identify items that have an affinity for each other. It is frequently used to analyze transactional data (also called market baskets) to identify items that often appear together in transactions. See Chapter 20, "Association Analysis".

Chapter 3

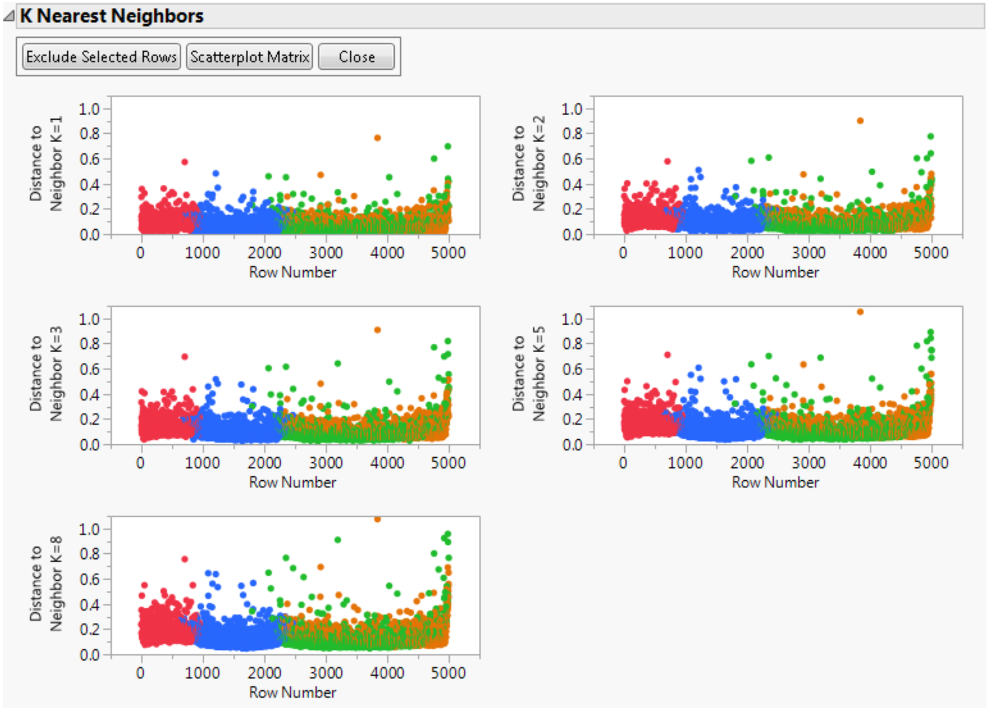
Modeling Utilities

Exploring Data for Outliers, Missing Values, and Strong Predictors

Modeling Utilities is a collection of utilities that are designed to assist in the data cleaning and pre-processing stages of data analysis. Each utility has exploratory tools to give you a more thorough understanding of your data. With Modeling Utilities, you can do the following:

- Explore outliers in both the univariate and multivariate cases.
- Explore and impute missing values in your data.
- JMP PRO Create a validation column that divides the data into training, validation, and test sets.
- Screen high-dimensional data for strong predictors.

Figure 3.1 Multivariate k-Nearest Neighbor Outlier Example



Explore Outliers Utility

Exploring and understanding outliers in your data is an important part of analysis. Outliers in data can be due to mistakes in data collection or reporting, measurement systems failure, or the inclusion of error or missing value codes in the data set. The presence of outliers can distort estimates. Therefore, any analyses that are conducted are biased toward those outliers. Outliers also inflate the sample variance. Sometimes retaining outliers in data is necessary, however, and removing them could underestimate the sample variance and bias the data in the opposite direction.

Whether you remove or retain outliers, you must locate them. There are many ways to visually inspect for outliers. For example, box plots, histograms, and scatter plots can sometimes easily display these extreme values. See the Visualizing Your Data in the *Discovering JMP* book for more information.

The Explore Outliers tool provides four different options to identify, explore, and manage outliers in your univariate or multivariate data.

Quantile Range Outliers Uses the quantile distribution of each column to identify outliers as extreme values. This tool is useful for discovering missing value or error codes within the data. This is the recommended method to begin exploring outliers in your data. See "Quantile Range Outliers" on page 36.

Robust Fit Outliers Finds robust estimates of the center and spread of each column and identifies outliers as those far from those values. See "Robust Fit Outliers" on page 39.

Multivariate Robust Outliers Uses the Multivariate platform with Robust option to find outliers based on the Mahalanobis distance from the estimated robust center. See "Multivariate Robust Outliers" on page 40.

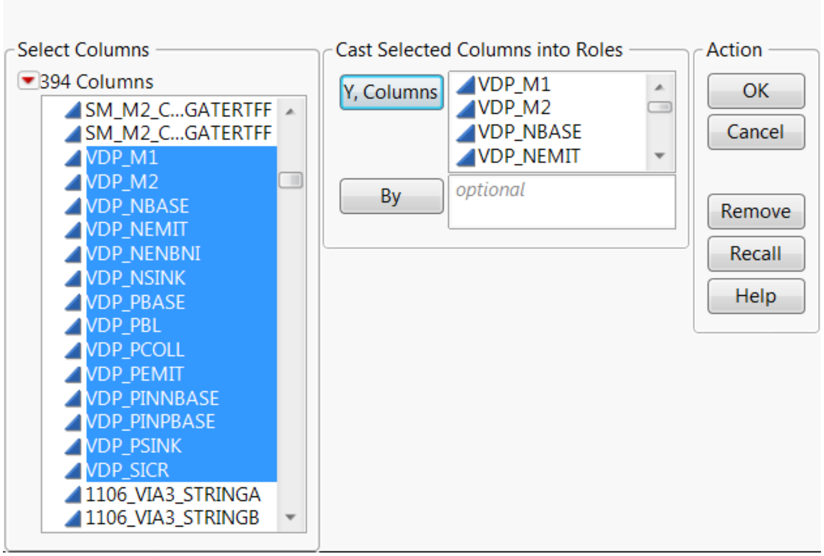
Multivariate k-Nearest Neighbor Outliers Finds outliers as values far from their k-nearest neighbors. See "Multivariate k-Nearest Neighbor Outliers" on page 41.

Example of the Explore Outliers Utility

The Probe.jmp sample data table contains 387 characteristics (the Responses column group) measured on 5800 semiconductor wafers. The Lot ID and Wafer Number columns uniquely identify the wafer. You are interested in identifying outliers within a select group of columns of the data set. Use the Explore Outliers utility to identify outliers that can then be examined using the Distribution platform.

1. Select **Help > Sample Data Library** and open the Probe.jmp sample data table.
2. Select **Analyze > Screening > Explore Outliers.**
3. Select columns VDP_M1 through VDP_SICR and click **Y, Columns**. You should have 14 columns selected (see Figure 3.2).

Figure 3.2 Explore Outliers Launch Window



Y, Columns: VDP_M1, VDP_M2, VDP_NBASE, VDP_NEMIT

4. Click **OK**.

5. Click **Quantile Range Outliers**.

   The Quantile Range Outliers report shows each column and lists the number and identity of the outliers found.

6. In the Quantile Range Outliers report, select the check box named **Show only columns with outliers.** This limits the list of columns to only those that contain outliers.

   Note that several columns contain outlier values of 9999. Many industries use nines as a missing value code.

7. In the Nines report, select each column.

8. Click **Add Highest Nines to Missing Value Codes**.

   A JMP Alert indicates that you should use the **Save As** command to preserve your original data.

9. Click **OK**.

10. In the Quantile Range Outliers report, click **Rescan**.

11. Select the check box named **Restrict search to integers**.

    In most cases of continuous data, integer values are often error codes or other coded data values. Notice that no additional error codes are included in this set of columns.

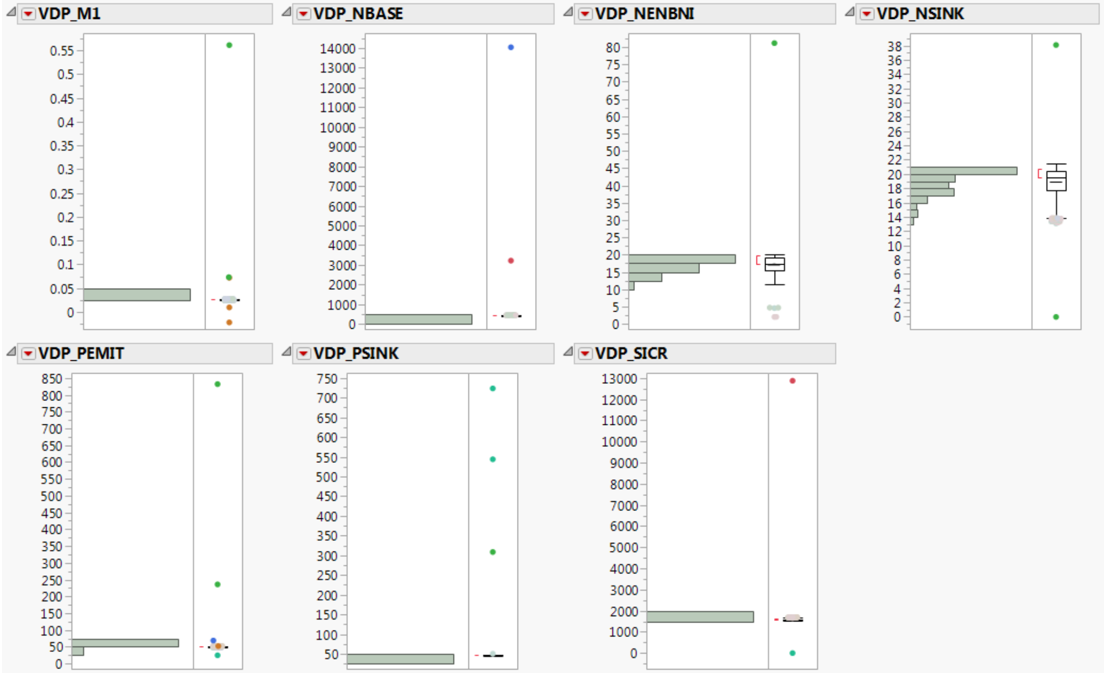
12. Deselect **Restrict search to integers**.

Examine the Data

1. Select all of the remaining columns in the Quantile Range Outliers report.
2. Click **Select Rows**.
3. Select **Analyze > Distribution**.
4. Assign the selected columns to the **Y, Columns** role. Because you selected these column names in the Quantile Range Outliers report, they are already selected in the Distribution launch window.
5. Click **OK**.

   Figure 3.3 shows a simplified version of the report.

Figure 3.3 Distribution of Columns with Outliers Selected



In columns VDP_M1 and VDP_PEMIT, notice that the selected outliers are somewhat close to the majority of data. For the rest of the columns, the selected outliers appear distant enough to exclude them from your analyses.

Refine Excluded Outliers

1. In the Quantile Range Outliers report, hold Ctrl and deselect columns VDP_M1 and VDP_PEMIT.
2. With the remaining columns selected in the report, click **Exclude Rows**.

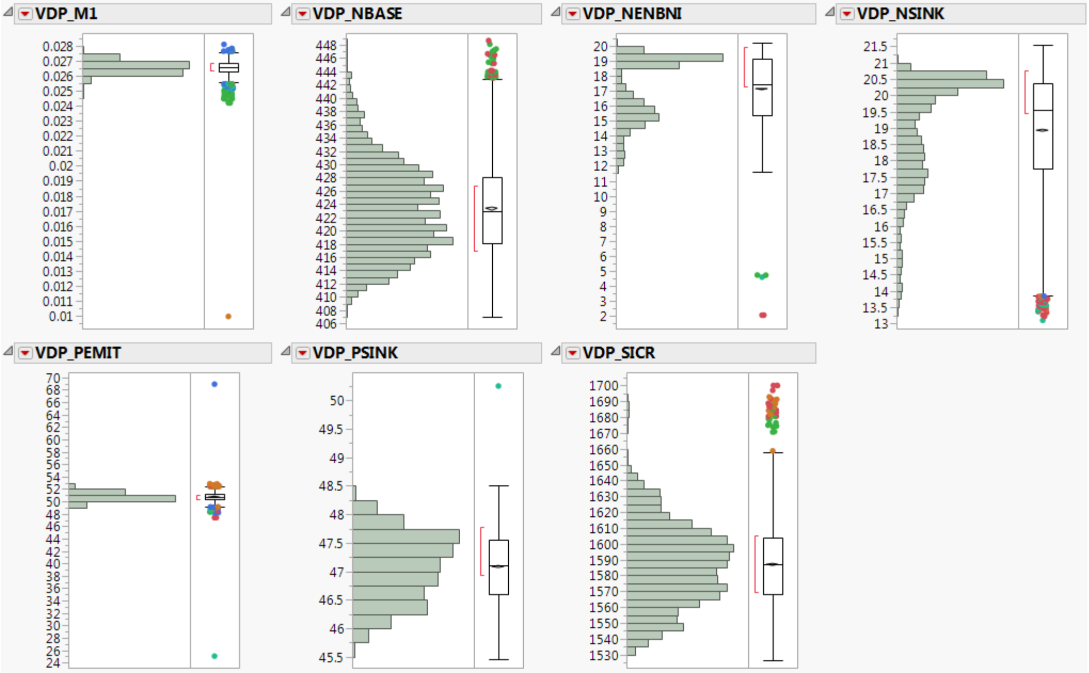
3. Change Q to 20.
4. Click **Rescan**.
5. Select columns VDP_M1 and VDP_PEMIT in the report. Click **Select Rows**.

Reexamine the Data

1. Examine the Distributions report again. Notice the selected outliers are now separate enough from the majority of the data to select and exclude them from your analyses.
2. In the Quantile Range Outliers report, click **Exclude Rows**.
3. In the Distributions report, click the red triangle menu next to Distributions.
4. Select **Redo > Redo Analysis**.

   Figure 3.4 shows a simplified version of the report.

Figure 3.4 Distributions of Columns with Outliers Excluded



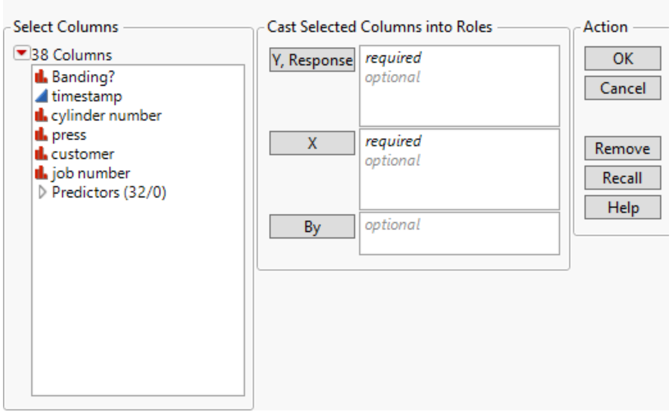
The displays of the distributions of the data are now more informative without the outliers.

Launch the Explore Outliers Utility

Note: The Explore Outliers commands only analyze columns with a Continuous modeling type. Other columns can be entered in the launch window but are ignored.

To launch Explore Outliers, select **Analyze** > **Screening** > **Explore Outliers**. The launch window appears.

Figure 3.5 Explore Outliers Utility Launch Window



In the launch window, select the analysis columns as **Y, Columns**. You can also specify a **By** variable. After you click **OK**, the Explore Outliers report appears. You are presented with the following four outlier analysis commands:

- "Quantile Range Outliers" on page 36
- "Robust Fit Outliers" on page 39
- "Multivariate Robust Outliers" on page 40
- "Multivariate k-Nearest Neighbor Outliers" on page 41

Quantile Range Outliers

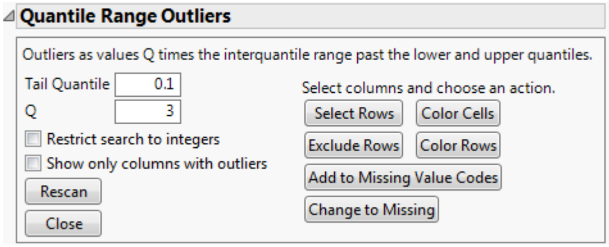
The Quantile Range Outliers method of outlier detection uses the quantile distribution of the values in a column to locate the extreme values. Quantiles are useful for detecting outliers because there is no distributional assumption associated with them. Data are simply sorted from smallest to largest. For example, the 20th quantile is the value at which 20% of values are smaller. Extreme values are found using a multiplier of the interquantile range, the distance between two specified quantiles. For more details about how quantiles are computed, see the "Distributions" chapter in the *Basic Analysis* book.

The Quantile Range Outliers utility is also useful for identifying missing value codes stored within the data. As noted earlier, in some industries, missing values are entered as nines (such as 999 and 9999). This utility finds any nines greater than the upper quartile as suspected missing value codes. The utility then enables you to add those missing value codes as a column property in the data table.

Quantile Range Outliers Options

The Quantile Range Outliers panel enables you to specify how outliers are to be calculated and how you want to manage them. Figure 3.6 shows the default Quantile Range Outliers window.

Figure 3.6 Quantile Range Outliers Window



Quantile Range Outliers

Outliers as values Q times the interquantile range past the lower and upper quantiles.

Tail Quantile 0.1

Q 3

□ Restrict search to integers

□ Show only columns with outliers

Select columns and choose an action.

An outlier is considered any value more than *Q* times the interquantile range from the lower and upper quantiles. You can adjust the value of *Q* and the size of the interquantile range.

Tail Quantile The probability for the lower quantile that is used to calculate the interquantile range. The probability of the upper quantile is considered 1 – Tail Quantile . For example, a Tail Quantile value of 0.1 means that the interquantile range is between the 0.1 and 0.9 quantiles of the data. The default value is 0.1.

Q The multiplier that helps determine values as outliers. Outliers are considered *Q* times the interquantile range past the Tail Quantile and 1 – Tail Quantile values. Large values of *Q* provide a more conservative set of outliers than small values. The default is 3.

Restrict search to integers Restricts outlier values to only integer values. This setting limits the search for outliers in order to find industry-specific missing value codes and error codes.

Show only columns with outliers Limits the list of columns in the report to those that contain outliers.

After the report is displayed using your specifications, there are many ways to act on these extreme values. You can select the outliers in a column by selecting the specified column in the Quantile Range Outliers report.

Select Rows Selects the rows of outliers in the selected columns in the data table.

Exclude Rows Turns on the exclude row state for the selected rows. Click **Rescan** to update the Quantile Range Outliers report.

Color Cells Colors the cells of the selected outliers in the data table.

Color Rows Colors the rows containing outliers for the selected columns in the data table

Add to Missing Value Codes Adds the selected outliers to the missing value codes column property. Use this option to identify known missing value or error codes within the data. Missing value and error codes are often integers and are sometimes either a positive or negative series of nines. Click **Rescan** to update the Quantile Range Outliers report.

Change to Missing Changes the outlier value to a missing value in the data table. Use caution when changing values to missing. Change values to missing only if the data are known to be invalid or inaccurate. Click **Rescan** to update the Quantile Range Outliers report.

Rescan Rescans the data after outlier actions have been taken.

Close Closes the Quantile Range Outliers panel.

Quantile Range Outliers Report

The Quantile Range Outliers report lists all columns with the outliers found using the specified options. The report shows values for the upper and lower quantiles along with their low and high thresholds. Values outside of these threshold limits are considered outliers. The number of outliers in each column is indicated. The values of each outlier are listed in the last column of the report. Outliers that occur more than once in a column are listed with their count in parentheses. To remove columns without outliers from the report, select **Show only columns with outliers**.

There are several things to look for when reading this report.

- Error codes. For some continuous data, suspiciously high integer values are likely to be error codes. For example, if your upper and lower quantile values are all less than 0.5, outliers such as 1049 or -777 are likely to be error codes.
- Zeros. Sometimes zeros can indicate missing values. If the majority of your data is reasonably large and you notice zeros as outliers, they are likely to be due to missing data.

Nines Report

The Nines report within the Quantile Range Outliers window shows a list of columns that contain probable missing value codes. These missing value codes are a series of nines (usually 9999) and are the highest number that is all nines and also higher than the upper quantile. If the count is high, it is likely that these outliers are actually missing value codes. If the count is very low, you should explore further to determine whether the value is an outlier or a missing value code. The Nines Report includes the upper quantile value.

This report is displayed only when probable missing value codes are identified.

Add Highest Nines to Missing Value Codes Adds the selected outlier values to the missing value codes column property. You must click **Rescan** to update the Quantile Range Outliers report.

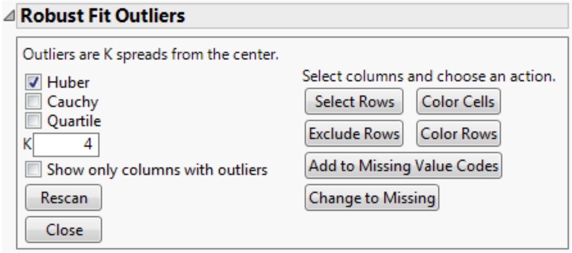
Change Highest Nines to Missing Replaces the selected outlier values with missing values in the data table.

Note: The first time you use choose an action (such as **Change to Missing** or **Exclude Rows**) to change your data, the alert window warns you to use the **Save As** command to save your data table as a new file to preserve a copy of your original data. When this window appears, click **OK**. If you decide to save your new data file, select **File > Save As** and save the file with a new name.

Robust Fit Outliers

Robust estimates of parameters are less sensitive to outliers than non-robust estimates. Robust Fit Outliers provides several types of robust estimates of the center and spread of your data to determine those values that can be considered extreme. Figure 3.7 shows the default Robust Fit Outliers window.

Figure 3.7 Robust Fit Outliers Window



Outliers are K spreads from the center.

☑ Huber
□ Cauchy
□ Quartile
K 4
□ Show only columns with outliers

Select columns and choose an action.

Robust Fit Outliers Options

Given a robust estimate of the center and spread, outliers are defined as those values that are *K* times the robust spread from the robust center. The Robust Fit Outliers window provides several options for calculating the robust estimates and multiplier *K* as well as provides tools to manage the outliers found.

Huber Uses Huber M-Estimation to estimate center and spread. This option is the default. See Huber and Ronchetti (2009).

Cauchy Assumes a Cauchy distribution to calculate estimates for the center and spread. Cauchy estimates have a high breakdown point and are typically more robust than Huber estimates. However, if your data are separated into clusters, the Cauchy distribution tends to consider only the half of the data that makes closer clusters, ignoring the rest.

Quartile Uses the interquartile range (IQR) to estimate the spread. The estimate for the center is the median. The estimate for spread is the IQR divided by 1.34898. Dividing the IQR by this factor makes the spread correspond to one standard deviation if it was normally distributed data.

K The multiplier that determines outliers as *K* times the spread away from the center. Large values of *K* provide a more conservative set of outliers than small values. The default is 4.

Show only columns with outliers Limits the list of columns in the report to those that contain outliers.

Once the report is displayed using your specifications, there are many ways to explore these extreme values. You can select the outliers in a row by selecting the specified row in the Robust Estimates and Outliers report.

Select Rows Selects the rows containing outliers for the selected columns in the data table.

Exclude Rows Sets the Exclude Row state for outliers in the selected columns in the data table. Click **Rescan** to update the Robust Estimates and Outliers report.

Color Cells Colors the cells of the selected outliers in the data table.

Color Rows Colors the rows containing outliers for the selected columns in the data table.

Add to Missing Value Codes Adds the selected outliers to the missing value codes column property for the selected columns. Use this option to identify known missing value or error codes within the data. Click **Rescan** to update the Robust Estimates and Outliers report.

Change to Missing Changes the outlier value to a missing value in the data table. Click **Rescan** to update the Robust Estimates and Outliers report.

Rescan Rescans the data after outlier actions have been taken.

Close Closes the Robust Fit Outliers panel.

Multivariate Robust Outliers

The Multivariate Robust Fit Outliers tool uses the Robust option in the Multivariate platform to examine the relationships between multiple variables. For more information about how the Multivariate platform works, see Correlations and Multivariate Techniques in the *Multivariate Methods* book.

Outlier Analysis

The Outlier Analysis calculates the Mahalanobis distances from each point to the center of the multivariate normal distribution. This measure relates to contours of the multivariate normal density with respect to the correlation structure. The greater the distance from the center, the higher the probability that it is an outlier. For more information about the Mahalanobis distance and other distance measures, see the Correlations and Multivariate Techniques in the *Multivariate Methods* book.

After the rows are excluded, you are given the option to either rerun the analysis or close the utility. Rerunning the analysis recalculates the center of the multivariate distribution without those excluded rows. Note that unless you hide the excluded rows in the data table, they still appear in the graph.

You can save the distances to the data table by selecting the **Save** option from the Mahalanobis Distances red triangle menu.

Figure 3.8 Multivariate Robust Outliers Mahalanobis Distance Plot

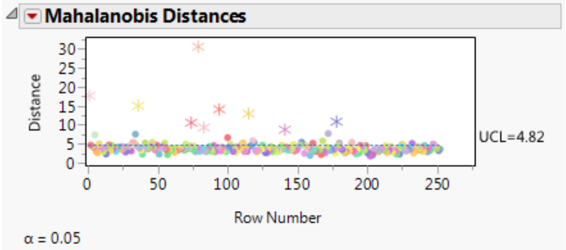


Figure 3.8 shows the Mahalanobis distances of 16 different columns. The plot contains an upper control limit (UCL) of 4.82.This UCL is meant to be a helpful guide to show where potential outliers might be. However, you should use your own discretion to determine which values are outliers. For more details about this upper control limit (UCL), see Mason and Young (2002).

Multivariate with Robust Estimates Options

The red triangle menu for Multivariate with Robust Estimates contains numerous options to analyze your multivariate data. For a list and description of these options, see the Correlations and Multivariate Techniques in the *Multivariate Methods* book.

Multivariate k-Nearest Neighbor Outliers

The basic approach of outlier detection is to consider points distant from other points as outliers. One way of determining the distance of a point to other clusters of points is explore the distance to its nearest neighbors. For each value of *K*, the Multivariate *k*-Nearest Neighbor Outliers utility displays a plot of the Euclidean distance from each point to it's *K*th nearest neighbor. You specify the largest value of *K*, denoted as *k*. Plots are provided for $K = 1, 2, 3, \ldots, k$, skipping values by the Fibonacci sequence to avoid displaying too many plots.

This approach is sensitive to the specified value of *k*. A small value of *k* can miss identifying points as outliers and a large value of *k* can falsely classify points as outliers:

- Suppose that the specified *k* is small, so that you are only studying a few neighbors. If there is a cluster of more than *k* points that is far from the rest of the points, then the points within the cluster will have small distances to their nearest neighbors. You may be unable to detect the cluster of outliers.

- Suppose that the specified *k* is large, so that you are studying a large number of neighbors. If there are clusters with fewer than *k* data points, then the points within these clusters may appear to be outliers. You may overlook the fact that the points form a cluster, interpreting the individual cluster members as outliers instead.

K-Nearest Neighbor Report

When you select Multivariate k-Nearest Neighbor Outliers from the list of commands, you are asked to specify the value of *k* to use as an upper bound for the furthest neighbor to be considered. Notice that the default value is set to 8.

The report shows plots for select values of *K* up to the value *k*. The value of *K* for each plot is displayed in its vertical axis label, which is of the form Distance to Neighbor K = <a>, where a is an integer denoting the a^{th} closest neighbor. Each plot shows the distance from the point in the i^{th} row to its a^{th} nearest neighbor. The points that have large distances from their neighbors, across multiple values of *K*, are likely to be outliers.

The buttons above the plots do the following:

Exclude Selected Rows Excludes rows corresponding to selected points from further analysis. The rows are assigned the Excluded row state in the data table. You are asked if you want to rerun or close the K Nearest Neighbors report. Rerunning the analysis identifies new nearest neighbors. The plots are updated and the excluded points are not shown.

Scatterplot Matrix Opens a separate window containing a scatterplot matrix for all columns in the analysis. You can explore potential outliers by selecting them in the K Nearest Neighbors plots and viewing them in the scatterplot matrix.

Close Closes the K Nearest Neighbors report.

Explore Outliers Utility Options

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

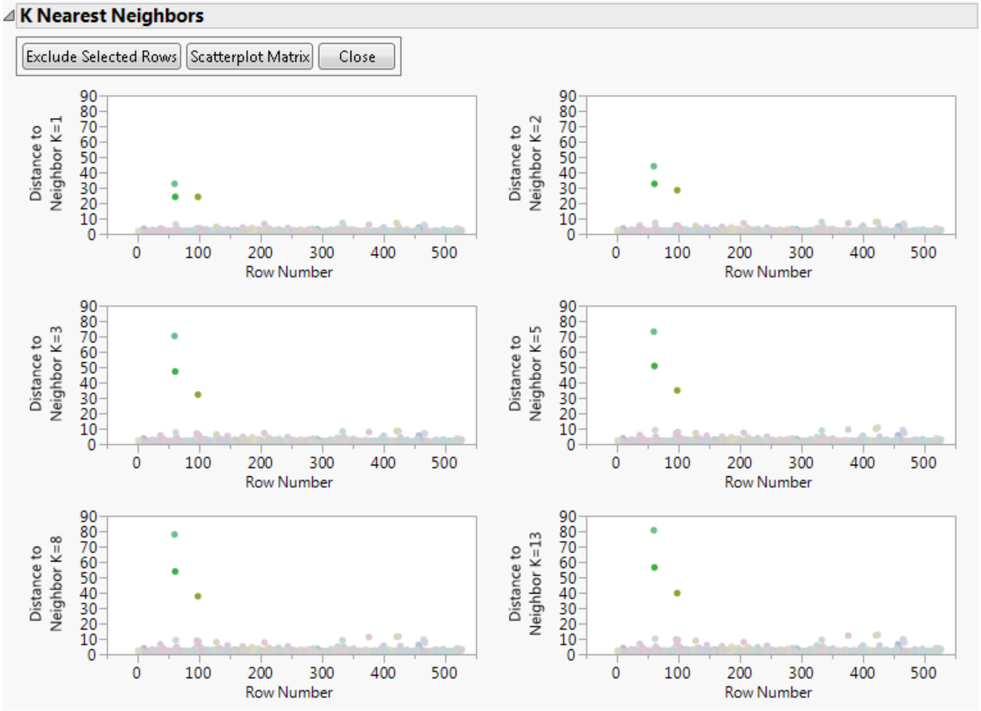
Additional Examples of the Explore Outliers Utility

Multivariate k-Nearest Neighbor Outliers Example

The Water Treatment.jmp data set contains daily measurement values of 38 sensors in an urban waste water treatment plant. You are interested in exploring these data for potential outliers. Potential outliers could include sensor failures, storms, and other situations.

1. Select **Help > Sample Data Library** and open Water Treatment.jmp.
2. Select **Analyze > Screening > Explore Outliers.**
3. Select the Sensor Measurements column group and click **Y, Columns**.
4. Click **OK**.
5. Select **Multivariate k-Nearest Neighbor Outliers**.
6. Enter 13 for k-nearest neighbors.
7. Click **OK**.

Figure 3.9 Outliers in Multivariate k-Nearest Neighbor Outliers Example



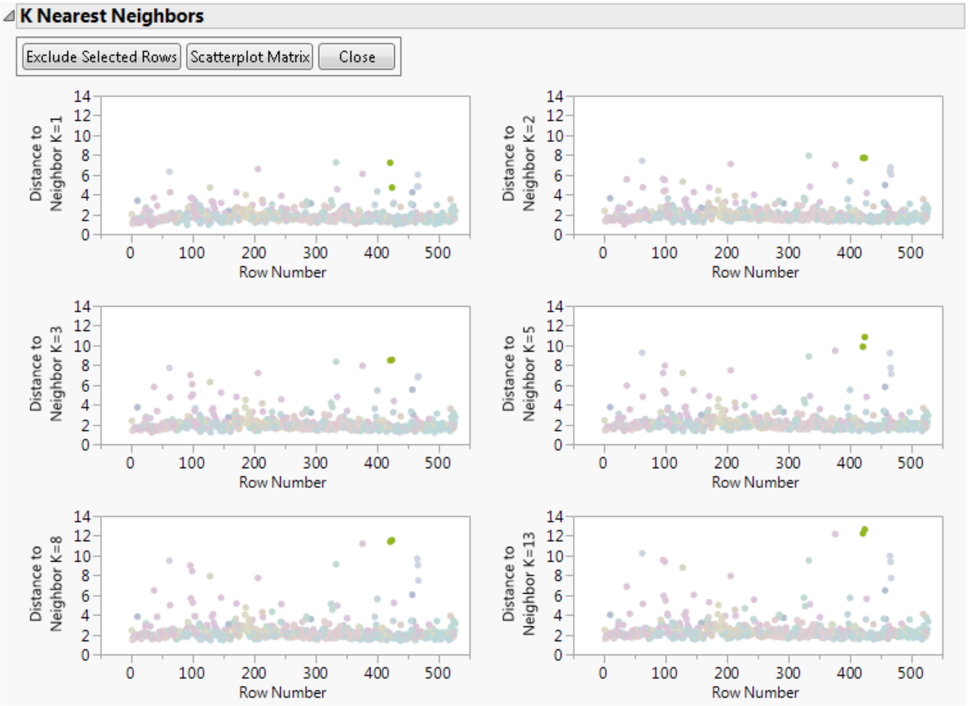
Notice the three extreme outliers selected in the K Nearest Neighbors plots in Figure 3.9. Each of these three rows corresponds to a date when the secondary settler in the water treatment plant was reported as malfunctioning. Because these three data points are due to faulty equipment, exclude them from future analyses.

8. Select the three extreme outliers and click **Exclude Selected Rows**.

   You are prompted to Rerun the utility or Close the window.

9. Click **Rerun**.
10. Type 13 for k-nearest neighbors.
11. Click **OK**.

Figure 3.10 Outliers in Multivariate k-Nearest Neighbors Example



Now locate the two light-green outliers close to row 400. Notice how they tend to stay close to each other as *k* increases. These two rows correspond to dates when solids overloads were experienced by the water treatment plant. Even though these data points have a relatively high Distance to Neighbor K=13, because they are due to a situation that you want to include in your study, you do not exclude them. Instead, you keep them in mind as you conduct further analyses.

Explore Missing Values Utility

The presence of missing values in a data set can affect the conclusions made using the data. If, for example, several healthy participants dropped out of a longitudinal study and their data continued on as missing, the results of the study can be biased toward those unhealthy individuals who remained. Missing data values must not only be identified, they must also be understood before further analysis can be conducted.

The Explore Missing Values utility provides several ways to identify and understand the missing values in your data. It also provides methods for conducting multivariate normal imputation for missing values. These methods assume that data are *missing at random*, which means that the probability that an observation is missing depends only on the values of the other variables in the study. If you suspect that missing values are *not* missing at random, then consider using the Informative Missing procedure, which is available in a number of platforms. For more information, see the Model Specification chapter in the *Fitting Linear Models* book.

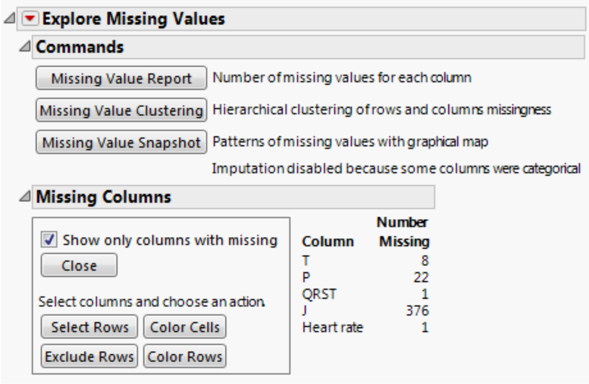
Example of the Explore Missing Values Utility

The Arrhythmia.jmp sample data table contains information from 452 patient electrocardiograms (ECGs). The data was originally collected to classify different patterns of ECGs as cardiac arrhythmia. However, there are missing values in this data table. You are primarily interested in exploring these missing values and imputing them when necessary. Since you can only conduct missing value imputation for columns that have a continuous modeling type, you will conduct your analysis in two stages.

Examine Missing Values

1. Select **Help > Sample Data Library** and open Arrhythmia.jmp.
2. Select **Analyze > Screening > Explore Missing Values**.
3. Select all columns (280 in all) and click **Y, Columns**.
4. Click **OK**. Select the **Show only columns with missing** checkbox.

Figure 3.11 Missing Value Report



Explore Missing Values

Commands

Missing Value Report — Number of missing values for each column

Missing Value Clustering — Hierarchical clustering of rows and columns missingness

Missing Value Snapshot — Patterns of missing values with graphical map

Imputation disabled because some columns were categorical

Missing Columns

☑ Show only columns with missing

Close

Select columns and choose an action

Select Rows | Color Cells | Exclude Rows | Color Rows

Column	Number Missing
T	8
P	22
QRST	1
J	376
Heart rate	1

The Missing Columns report shown in Figure 3.11 indicates that only five columns have missing data. Out of a total of 452 rows, Column J has 376 missing values. Because it is largely missing, it is not useful for data analysis, even with imputed values. However, it might be useful to model Column J using the Informative Missing option in a platform that supports this option to see if values are perhaps not missing at random.

Note that the two Imputation options, Multivariate Imputation and Multivariate SVD Imputation, are not shown. A message indicates that imputation is disabled because some columns included in the analysis were categorical. The data table contains several columns that are numeric, but have a nominal modeling type. These cannot be used for imputation.

Impute Missing Values

The five columns that have missing values are continuous. You proceed to impute values for the four columns other than Column J using multivariate imputation for the continuous columns in your data table. By doing so, you tacitly assume that the probabilities that values are missing depend only on the values of the continuous variables and not on the values of excluded nominal variables. To conduct this new analysis, you need to launch the Explore Missing Values utility again.

1. Select **Analyze > Screening > Explore Missing Values**.
2. In the launch window, click the red triangle next to **280 Columns**.

   You will use the columns filter menu to view only the columns with a Continuous modeling type in the Select Columns list.
3. Select **Modeling Type** > **Uncheck All**.

   This removes all columns from the Select Columns list.
4. Select **Modeling Type** > **Continuous**.

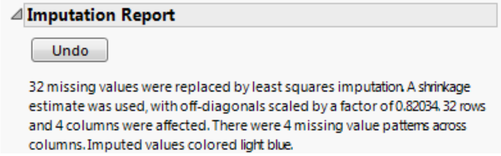
   The Select Columns list now contains only the 207 columns that are Continuous.

5. Select all 207 columns. Then Ctrl-click the J column (to deselect it) and click **Y, Columns**.
6. Click **OK**.
7. Click **Multivariate Normal Imputation.**

   A window appears and asks whether you want to use a Shrinkage estimator for covariances.
8. Click **Yes Shrinkage**.

   A JMP Alert appears, informing you that you should use the **Save As** command to preserve your original data.
9. Click **OK**.

Figure 3.12 Imputation Report



32 missing values were replaced by least squares imputation. A shrinkage estimate was used, with off-diagonals scaled by a factor of 0.82034. 32 rows and 4 columns were affected. There were 4 missing value patterns across columns. Imputed values colored light blue.

The Imputation Report in Figure 3.12 indicates how many missing values were imputed and the specific imputation details. No missing data remain in the four columns that had missing values.

Launch the Explore Missing Values Utility

Launch the Explore Missing Values modeling utility by selecting **Analyze > Screening > Explore Missing Values**. Enter the columns of interest into the Y, Columns list.

Note: You can enter only columns with a Numeric modeling type in the Explore Missing Values utility.

The Missing Value Report

After you click OK in the launch window, the report opens to show a Commands outline and a Missing Columns report. The commands are the following:

- "Missing Value Report" on page 48
- "Missing Value Clustering" on page 48
- "Missing Value Snapshot" on page 48
- "Multivariate Normal Imputation" on page 49 (Not available if you entered a Numeric column with a Nominal or Ordinal modeling type in the launch window.)

- "Multivariate SVD Imputation" on page 49 (Not available if you entered a Numeric column with a Nominal or Ordinal modeling type in the launch window.)

Missing Value Report

The Missing Value Report opens the Missing Columns report, which lists the name of each column and the number of missing values in that column.

Show only columns with missing Removes columns from the list that do not have missing values.

Close Closes the Missing Columns report.

Select Rows Selects the rows in the data table that contain missing values for the column(s) that you select in the Missing Columns report.

Exclude Rows Applies the excluded row state for rows in the data table that contain missing values for the column(s) that you select in the Missing Columns report.

Color Cells Colors the cells in the data table that contain missing values for the column(s) that you select in the Missing Columns report.

Color Rows Colors the rows in the data table that contain missing values for the column(s) that you select in the Missing Columns report.

Missing Value Clustering

Missing Value Clustering provides a hierarchical clustering analysis of the missing data.

- The dendrogram to the right of the plot shows clusters of missing data pattern rows. These are the rows that you would obtain by using Tables > Missing Data Pattern.
- The dendrogram beneath the plot shows clusters of variables.

Use this report to determine if certain groups of columns tend to have similar patterns of missing values.

The rows of the plot are defined by the missing data patterns; there is a row for each pattern. The columns correspond to the variables. Each red cell indicates a group of missing values for the column listed beneath the plot. Place your cursor in a cell to see the list of values represented. Click in the plot to select missing data pattern rows. Vertical bars appear to indicate the selected patterns.

Missing Value Snapshot

The Missing Value Snapshot shows a cell plot for the missing values. The columns represent the variables. Black cells indicate a missing value. This plot is especially useful in understanding missingness for longitudinal data, where subjects may withdraw from a study before the end of the data collection period.

Multivariate Normal Imputation

The Multivariate Normal Imputation utility imputes missing values based on the multivariate normal distribution. The procedure requires that all variables have a Continuous modeling type. The algorithm uses least squares imputation. The covariance matrix is constructed using pairwise covariances. The diagonal entries (variances) are computed using all non-missing values for each variable. The off-diagonal entries for any two variables are computed using all observations that are non-missing for both variables. In cases where the covariance matrix is singular, the algorithm uses minimum norm least squares imputation based on the Moore-Penrose pseudo-inverse.

Multivariate Normal Imputation allows the option to use a shrinkage estimator for the covariances. The use of shrinkage estimators is a way of improving the estimation of the covariance matrix. For more information about shrinkage estimators, see Schafer and Strimmer (2005).

Note: If a validation column is specified, the covariance matrices are computed using observations from the Training set.

Multivariate Normal Imputation Report

The imputation report explains the results of the multivariate imputation process. Results include the following:

- Method of imputation (either least squares or minimum-norm least squares)
- How many values were replaced
- Shrinkage estimator on/off
- Factor by which the off-diagonals were scaled
- How many rows and columns were affected
- How many different missing value patterns there were

Once the imputation is complete, the cells corresponding to imputed values in the data table are colored in light blue. If the Missing Columns report is open, it is updated to show no missing values.

Click **Undo** to undo the imputation and replace the imputed data with missing values.

Multivariate SVD Imputation

The Multivariate SVD Imputation utility imputes missing values using the singular value decomposition (SVD). This utility is useful for data with hundreds or thousands of variables. Because SVD calculations do not require calculation of a covariance matrix, the SVD method is recommended for wide problems that contain large numbers of variables. The procedure requires that all variables have a Continuous modeling type.

The singular value decomposition represents a matrix of observations **X** as **X** = **UDV'**, where **U** and **V** are orthogonal matrices and **D** is a diagonal matrix.

The SVD algorithm used by default in the Multivariate SVD Imputation utility is the sparse Lanczos method, also known as the *implicitly restarted Lanczos bidiagonalization method* (IRLBA). See Baglama and Reichel (2005). The algorithm does the following:

1. Each missing value is replaced with its column's mean.
2. An SVD decomposition is performed on the matrix of observations, **X**.
3. Each cell that had a missing value is replaced by the corresponding element of the **UDV'** matrix obtained from the SVD decomposition.
4. Steps 2 and 3 are repeated until the SVD converges to the matrix **X**.

Imputation Method Window

When you click Multivariate SVD Imputation, the Imputation Method window opens to show recommended settings.

Number of Singular Vectors Number of singular vectors that are computed and used in the imputation.

> **Note:** It is important not to specify too many singular vectors, otherwise the SVD and the imputations do not change from iteration to iteration.

Maximum Iterations The number of iterations used in imputing the missing values.

Show Iteration Log Opens a Details report that shows the number of iterations and gives details on the criteria.

For large problems, a progress bar shows how many dimensions the SVD has completed. You can stop the imputation and use that number of dimensions at any time.

Multivariate SVD Imputation Report

The imputation report explains the results of the multiple imputation process.

- Method of imputation
- How many values were replaced
- How many rows and columns were affected

Once the imputation is complete, the Missing Columns report is automatically shown indicating no missing values in the columns that were imputed. Imputed values are displayed in light blue.

Click **Undo** to undo the imputation and replace the imputed data with missing values.

Explore Missing Values Utility Options

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

JMP PRO Make Validation Column Utility

Validation is the process of using part of a data set to estimate model parameters and using another part to assess the predictive ability of a model. With complex data, this can reduce the risk of model overfitting.

A validation column partitions the data into two or three parts.

- The training set is used to estimate the model parameters.
- The validation set is used to help choose a model with good predictive ability.
- The testing set checks the model's predictive ability after a model has been chosen.

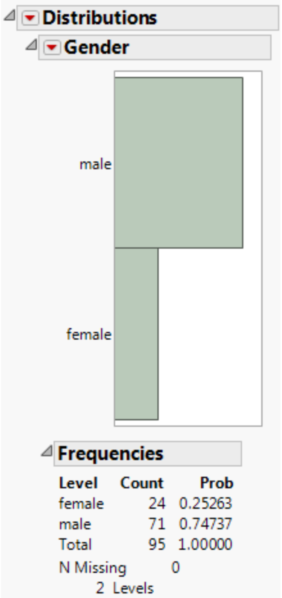
A validation column can be used as a validation method in the Fit Model platform.

JMP PRO Example of the Make Validation Column Utility

The Lipid Data.jmp data table contains blood measurements, physical measurements, and questionnaire data from 95 subjects at a California hospital. You are interested in using a validation column as a way of validation during future analyses.

1. Select **Help > Sample Data Library** and open Lipid Data.jmp.
2. Select **Analyze > Distribution**.
3. Assign Gender to the Y, Columns role. Click **OK**.

Figure 3.13 Distribution of Gender in Lipid Data.jmp



Level	Count	Prob
female	24	0.25263
male	71	0.74737
Total	95	1.00000
N Missing	0	
2 Levels		

Figure 3.13 illustrates the distribution of Gender in the data set. Notice that there is not an equal proportion of males and females represented. Because there is a scarcity of females within the data, you want to be sure to balance the genders across the validation and training sets.

4. Select **Analyze > Predictive Modeling > Make Validation Column**.
5. Click **Stratified Random**.
6. Select Gender as the column used for validation holdback.
7. Click **OK**.

   A Validation column is added to the data table. You can explore the distribution of the validation and training sets by creating a Mosaic Plot.

8. Select **Analyze > Fit Y by X**.
9. Assign Validation to Y, Response, and Gender to the X, Factor.
10. Click **OK**.

Figure 3.14 Distribution of Gender across Validation and Training Sets

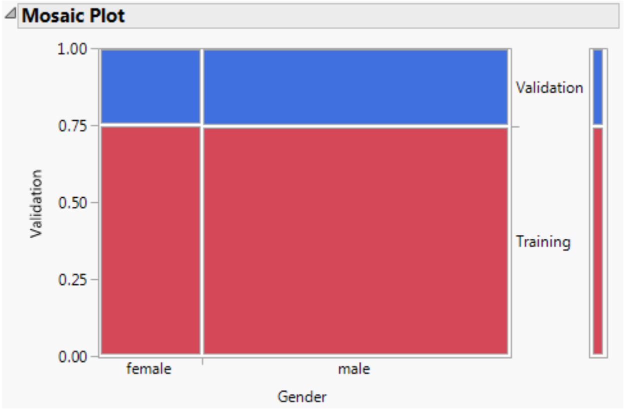


Figure 3.14 illustrates the distribution of Gender across each of the validation and training sets. Note that about 75% of both females and males are in the training set and about 25% of both females and males are in the validation set.

Launch the Make Validation Column Utility

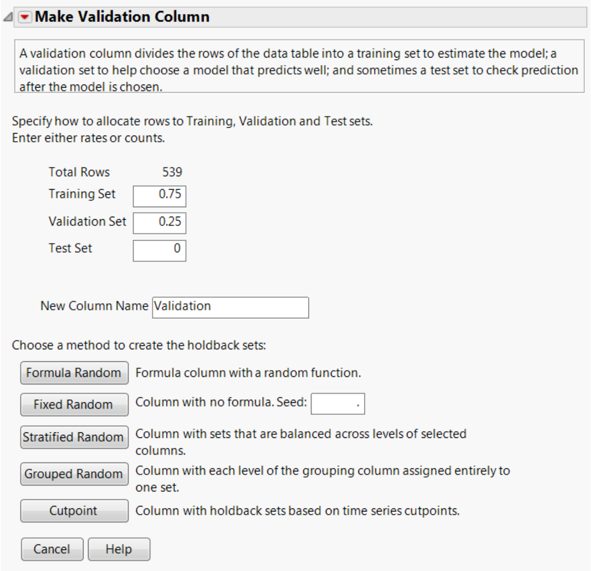
You can launch the Make Validation Column utility in two ways:

- Select Analyze > Predictive Modeling > Make Validation Column. See “Make Validation Column Window” on page 53.
- Click Validation in a platform launch window. See“Click Validation in a Platform Launch” on page 55.

Make Validation Column Window

In the Make Validation Column window, you specify the proportion or number of rows for each of your holdback sets and then you select a method for constructing the holdback sets.

Figure 3.15 Make Validation Column Window



Make Validation Column

A validation column divides the rows of the data table into a training set to estimate the model; a validation set to help choose a model that predicts well; and sometimes a test set to check prediction after the model is chosen.

Specify how to allocate rows to Training, Validation and Test sets.
Enter either rates or counts.

Total Rows	539
Training Set	0.75
Validation Set	0.25
Test Set	0

New Column Name: Validation

Choose a method to create the holdback sets:

Formula Random: Formula column with a random function.
Fixed Random: Column with no formula. Seed: .
Stratified Random: Column with sets that are balanced across levels of selected columns.
Grouped Random: Column with each level of the grouping column assigned entirely to one set.
Cutpoint: Column with holdback sets based on time series cutpoints.

- Next to Training Set, Validation Set, and Test Set, enter values that represent the proportions or numbers of rows that you would like to include in each of these sets. The default values construct a training set that contains about 75% of the rows and a validation set that contains about 25% of the rows.
- Enter a name for your validation column next to New Column Name.

There are five methods available to create the holdback sets.

Formula Random Partitions the data into sets based on the allocations entered. For example, if the default values are entered, each row has a probability of 0.75 to be included in the training set and 0.25 probability of being included in the validation set. The formula is saved to the column. To see it, click on the plus icon to the right of the column name in the Columns panel.

Fixed Random Partitions the data into sets based on the allocations entered. For example, if the default values are entered, each row has a probability of 0.75 to be included in the training set and 0.25 probability of being included in the validation set. You can specify a random seed that enables you to reproduce the allocations in the future. No formula is saved to the column.

Stratified Random Partitions the data into balanced sets based on levels of columns that you specify. Use this option when you want a balanced representation of a column's levels in each of the training, validation, and testing sets.

When you click Stratified Random, a window appears that enables you to select one or more columns by which to stratify the data. When you click OK, the validation column is added to the data table. As in the Fixed Random case, rows are randomly assigned to the holdback sets based on the specified allocations. However, this is done at each level or combination of levels of the stratifying columns.

A column is added to the data table with a Notes property that gives the stratifying variables.

Grouped Random Partitions the data into sets in such a way that entire levels of a specified column or combinations of levels of two or more columns are placed in the same holdback set. Use this option when splitting levels across holdback sets is not desirable.

When you click Grouped Random, a window appears that enables you to select one or more columns to be grouping columns. When you click OK, the levels are randomly assigned to holdback sets. When a level is larger than the proportion or number of rows you specify, it stays in its assigned holdback set. However, fewer rows are allocated into the training set. Because of this, the sizes of the resulting sets vary slightly from the sizes that you specified.

Cutpoint Partitions the data into sets based on time series cutpoints. Use this option when you want to assign your data to holdback sets based on time periods.

When you click Cutpoint, a window appears that enables you to select one or more columns to define time periods. When you click OK, a JMP Alert appears that shows the assigned cutpoints. A column that reflects this assignment is added to the data table. The training set consists of rows between the first cutpoint and the second cutpoint. The validation set consists of rows between the second and third cutpoints. The test set consists of the remaining rows. These sets are chosen to reflect the proportions or numbers of rows that you specified.

Click Validation in a Platform Launch

Use this method if you are in a platform launch window and need to construct a validation column quickly. Note the following:

- The platform must support a Validation column.
- No columns must be selected in the Select Columns list.

Click the Validation button in the platform launch window. A Make Validation Column window appears with default settings of 0.7 for the Training Set, 0.3 for the Validation Set, and 0.0 for the Test Set.

1. Enter your desired proportions or numbers next to Training Set, Validation Set, and Test Set.
2. Type a name for the new column next to New Column Name.
3. Click OK.

The new column appears in the data table with a formula. In the launch window, the new column is assigned to the Validation role.

Note: Launching the Make Validation Column utility through a platform launch window is equivalent to selecting the Formula Random method from Analyze > Predictive Modeling > Make Validation Column.The Fixed Random, Stratified Random, Grouped Random, and Cutpoint methods are not available.

Chapter 4

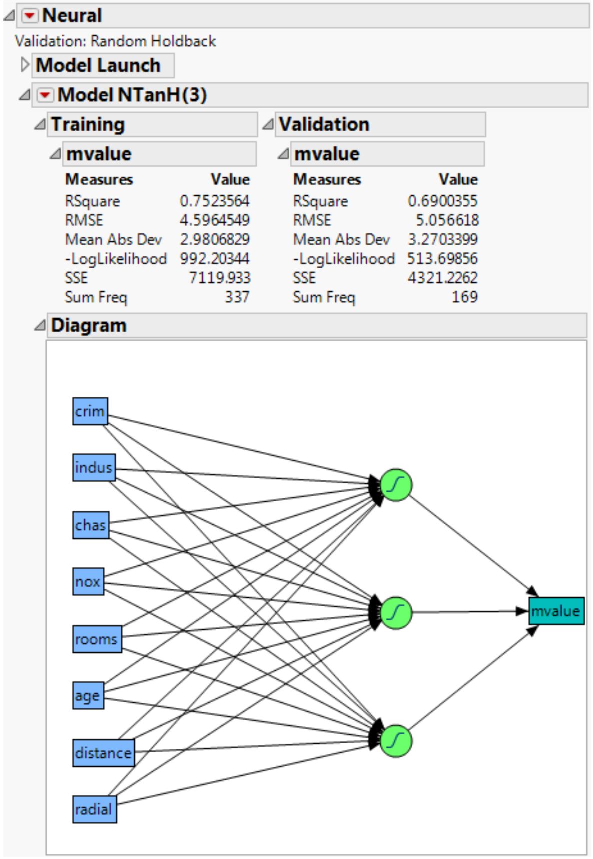
Neural Networks

Fit Nonlinear Models Using Nodes and Layers

Most features in this platform are available only in JMP Pro and noted with this icon.

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



Neural

Validation: Random Holdback

Model Launch

Model NTanH(3)

Training — mvalue

Measures	Value
RSquare	0.7523564
RMSE	4.5964549
Mean Abs Dev	2.9806829
-LogLikelihood	992.20344
SSE	7119.933
Sum Freq	337

Validation — mvalue

Measures	Value
RSquare	0.6900355
RMSE	5.056618
Mean Abs Dev	3.2703399
-LogLikelihood	513.69856
SSE	4321.2262
Sum Freq	169

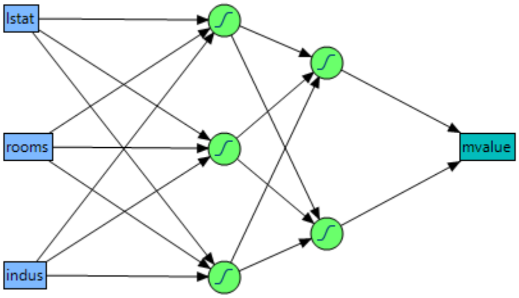
Diagram

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.

Figure 4.2 Neural Network Diagram



The functions applied at the nodes of the hidden layers are called activation functions. The activation function is a transformation of a linear combination of the X variables. For more details about the activation functions, see "Hidden Layer Structure" on page 62.

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

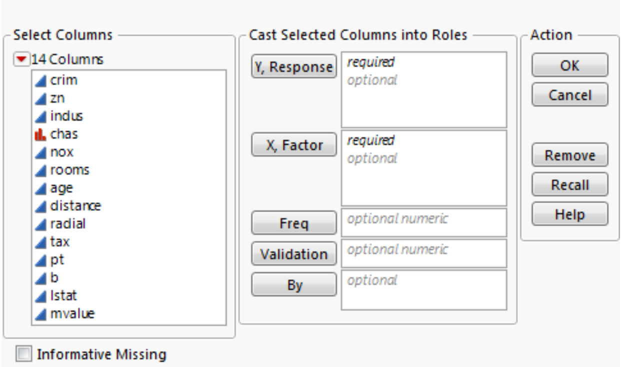
To launch the Neural platform, select **Analyze** > **Predictive Modeling** > **Neural**.

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 control panel.

The Neural Launch Window

Use the Neural launch window to specify X and Y variables, a validation column, and to enable Informative Missing value coding.

Figure 4.3 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 61. If you click the Validation button with no columns selected in the Select Columns list, you can add a validation column to your data table. For more information about the Make Validation Column utility, see the Modeling Utilities chapter in the *Predictive and Specialized Modeling* book.

By Choose a variable to create separate models for each level of the variable.

Informative Missing 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 <colname> Is Missing, is created and included in the model. If a variable is transformed using the Transform Covariates fitting option on the Model Launch control panel, missing values are replaced by the mean of the transformed variable.

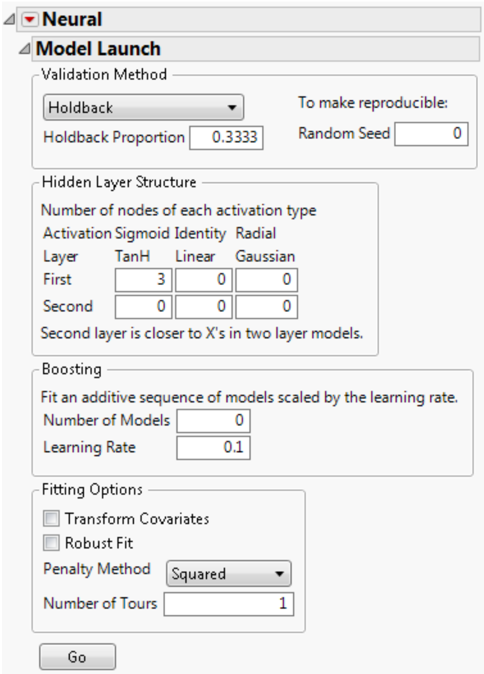
For a categorical variable, missing values are treated as a separate level of that variable.

The Model Launch Control Panel

Use the Model Launch control panel 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 Control Panel

JMP Pro

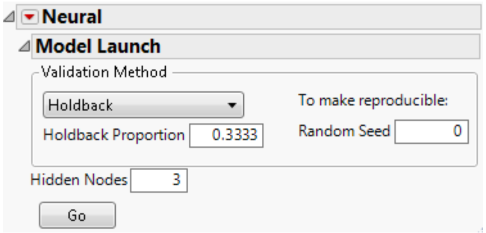


Neural
Model Launch
Validation Method
Holdback
Holdback Proportion 0.3333
To make reproducible:
Random Seed 0
Hidden Layer Structure
Number of nodes of each activation type

Activation	Sigmoid	Identity	Radial
Layer	TanH	Linear	Gaussian
First	3	0	0
Second	0	0	0

Second layer is closer to X's in two layer models.
Boosting
Fit an additive sequence of models scaled by the learning rate.
Number of Models 0
Learning Rate 0.1
Fitting Options
□ Transform Covariates
□ Robust Fit
Penalty Method Squared
Number of Tours 1
Go

JMP



Neural
Model Launch
Validation Method
Holdback
Holdback Proportion 0.3333
To make reproducible:
Random Seed 0
Hidden Nodes 3
Go

Validation Method Select the method that you want to use for model validation. For details, see “Validation Method” on page 61.

Random Seed Specify a nonzero numeric random seed if you want to reproduce the validation assignment for future launches of the Neural platform. By default, the Random Seed is set to zero, which does not produce reproducible results. When you save the analysis to a script, the random seed that you enter is saved to the script.

Hidden Layer Structure or Hidden Nodes Specify the number of hidden nodes of each type in each layer. For details, see “Hidden Layer Structure” on page 62.

Note: The standard edition of JMP uses only the TanH activation function, and can fit only neural networks with one hidden layer.

JMP PRO **Boosting** Specify options for gradient boosting. For details, see “Boosting” on page 63.

JMP PRO **Fitting Options** Specify options for variable transformation and model fitting. For details, see “Fitting Options” on page 63.

Go Fits the neural network model and shows the model reports.

After you click **Go** to fit a model, you can reopen the Model Launch Control Panel 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.

The training, validation, and test sets are created by subsetting the original data into parts. Select one of the following methods to subset a data set.

Excluded Rows Holdback 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 the Enter and Edit Data chapter in the *Using JMP* book.

Holdback Randomly divides the original data into the training and validation sets. You can specify the proportion of the original data to use as the validation set (holdback).

KFold Divides the original data into K subsets. In turn, each of the K sets is used to validate the model fit on the rest of the data, fitting a total of K models. The model giving the best validation statistic is chosen as the final model.

This method is best for small data sets, because it makes efficient use of limited amounts of data.

JMP PRO **Validation Column** Uses the column’s values to divide the data into parts. The column is assigned using the Validation role on the Neural launch window. See Figure 4.3.

The column’s values determine how the data is split, and what method is used for validation:

- If the column has three unique values, then:

  the smallest value is used for the Training set.

  the middle value is used for the Validation set.

  the largest value is used for the Test set.
- If the column has two unique values, then only Training and Validation sets are used.
- If the column has more than three unique values, then K-Fold validation is performed.

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. The following activation functions are available:

TanH 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.

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.

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.

Use the Boosting panel in the Model Launch control panel to specify the number of component models and the learning rate. Use the Hidden Layer Structure panel in the Model Launch control panel 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 x 6 = 12 nodes.

Fitting Options

The following model fitting options are available:

Transform Covariates Transforms all continuous variables to near normality using either the Johnson Su or Johnson Sb distribution. Transforming the continuous variables helps mitigate the negative effects of outliers or heavily skewed distributions.

See the Save Transformed Covariates option in “Model Options” on page 66.

Robust Fit Trains the model using least absolute deviations instead of least squares. This option is useful if you want to minimize the impact of response outliers. This option is available only for continuous responses.

Penalty Method Choose the penalty method. To mitigate the tendency neural networks have to overfit data, the fitting process incorporates a penalty on the likelihood. See "Penalty Method" on page 64.

Number of Tours Specify the number of times to restart the fitting process, with each iteration using different random starting points for the parameter estimates. The iteration with the best validation statistic is chosen as the final model.

Penalty Method

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

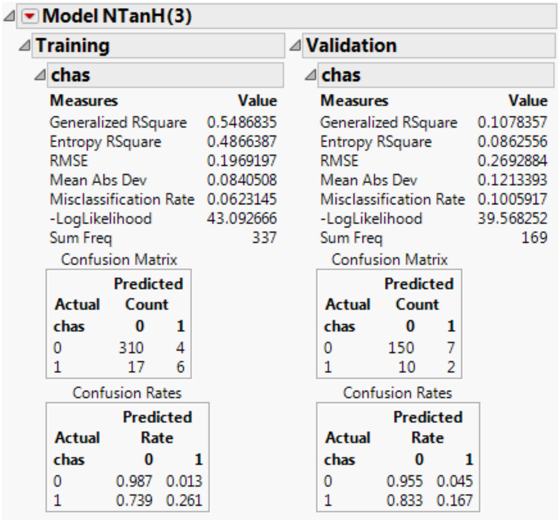
Table 4.1 Descriptions of Penalty Methods

Method	Penalty Function	Description
Squared	$\sum \beta_i^2$	Use this method if you think that most of your X variables are contributing to the predictive ability of the model.
Absolute	$\sum \lvert \beta_i \rvert$	Use either of these methods if you have a large number of X variables, and you think that a few of them contribute more than others to the predictive ability of the model.
Weight Decay	$\sum \frac{\beta_i^2}{1 + \beta_i^2}$	
NoPenalty	none	Does not use a penalty. You can use this option if you have a large amount of data and you want the fitting process to go quickly. However, this option can lead to models with lower predictive performance than models that use a penalty.

Model Reports

A model report is created for every neural network model. Measures of fit appear for the training and validation sets. Additionally, confusion statistics appear when the response is nominal or ordinal.

Figure 4.5 Example of a Neural Model Report



Model NTanH(3)

Training

chas

Measures	Value
Generalized RSquare	0.5486835
Entropy RSquare	0.4866387
RMSE	0.1969197
Mean Abs Dev	0.0840508
Misclassification Rate	0.0623145
-LogLikelihood	43.092666
Sum Freq	337

Confusion Matrix

Actual	Predicted Count	
chas	0	1
0	310	4
1	17	6

Confusion Rates

Actual	Predicted Rate	
chas	0	1
0	0.987	0.013
1	0.739	0.261

Validation

chas

Measures	Value
Generalized RSquare	0.1078357
Entropy RSquare	0.0862556
RMSE	0.2692884
Mean Abs Dev	0.1213393
Misclassification Rate	0.1005917
-LogLikelihood	39.568252
Sum Freq	169

Confusion Matrix

Actual	Predicted Count	
chas	0	1
0	150	7
1	10	2

Confusion Rates

Actual	Predicted Rate	
chas	0	1
0	0.955	0.045
1	0.833	0.167

Training and Validation Measures of Fit

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

Generalized RSquare A measure that can be applied to general regression models. It is based on the likelihood function L and is scaled to have a maximum value of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model. The Generalized RSquare measure simplifies to the traditional RSquare for continuous normal responses in the standard least squares setting. Generalized RSquare is also known as the Nagelkerke or Craig and Uhler R^2, which is a normalized version of Cox and Snell's pseudo R^2. See Nagelkerke (1991).

Entropy RSquare Compares the log-likelihoods from the fitted model and the constant probability model. Appears only when the response is nominal or ordinal.

RSquare Gives the RSquare for the model.

RMSE Gives the root mean square error. When the response is nominal or ordinal, the differences are between 1 and p (the fitted probability for the response level that actually occurred).

Mean Abs Dev The average of the absolute values of the differences between the response and the predicted response. When the response is nominal or ordinal, the differences are between 1 and p (the fitted probability for the response level that actually occurred).

Misclassification Rate The rate for which the response category with the highest fitted probability is not the observed category. Appears only when the response is nominal or ordinal.

-LogLikelihood Gives the negative of the log-likelihood. See the *Fitting Linear Models* book.

SSE Gives the error sums of squares. Available only when the response is continuous.

Sum Freq Gives the number of observations that are used. If you specified a Freq variable in the Neural launch window, Sum Freq gives the sum of the frequency column.

If there are multiple responses, fit statistics are given for each response, and an overall Generalized RSquare and negative Log-Likelihood 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. The model report red triangle menu provides the following options:

Diagram Shows a diagram representing the hidden layer structure.

Show Estimates Shows the parameter estimates in a report.

Profiler 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.

Categorical Profiler 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

Contour Profiler 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

Surface Profiler 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 Plot chapter in the *Profilers* book.

ROC Curve Creates an ROC curve. Available only for nominal or ordinal responses. For details about ROC Curves, see "ROC Curve" on page 90 in the "Partition Models" chapter.

Lift Curve Creates a lift curve. Available only for nominal or ordinal responses. For details about Lift Curves, see “Lift Curve” on page 91 in the “Partition Models” chapter.

Plot Actual by Predicted Plots the actual versus the predicted response. Available only for continuous responses.

Plot Residual by Predicted Plots the residuals versus the predicted responses. Available only for continuous responses.

Save Formulas Creates new columns in the data table containing formulas for the predicted response and the hidden layer nodes.

Save Profile Formulas 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.

Save Fast Formulas 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.

JMP PRO **Publish Prediction Formula** Creates prediction formulas and saves them as formula column scripts in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the “Formula Depot” chapter on page 169.

Make SAS Data Step Creates SAS code that you can use to score a new data set.

Save Validation Creates a new column in the data table that identifies which rows were used in the training and validation sets. This option is not available when a Validation column is specified on the Neural launch window. See “The Neural Launch Window” on page 59.

JMP PRO **Save Transformed Covariates** Creates new columns in the data table showing the transformed covariates. The columns contain formulas that show the transformations. This option is available only when the Transform Covariates option is checked on the Model Launch control panel. See “Fitting Options” on page 63.

Remove Fit Removes the entire model report.

Example of a Neural Network

This example uses the Boston Housing.jmp data table. Suppose you want to create a model to predict the median home value as a function of several demographic characteristics. Follow the steps below to build the neural network model:

1. Launch the Neural platform by selecting **Analyze** > **Predictive Modeling** > **Neural**.
2. Assign mvalue to the **Y, Response** role.
3. Assign the other columns (crim through lstat) to the **X, Factor** role.

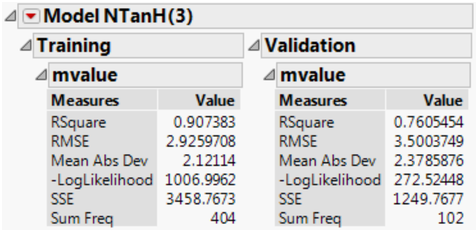
4. Click **OK**.
5. Enter 0.2 for the **Holdback Proportion**.
6. Enter 1234 for the **Random Seed**.

Note: In general, results vary due to the random nature of choosing a validation set. Entering the seed above enable you to reproduce the results shown in this example.

7. Enter 3 for the number of TanH nodes in the first layer.
8. Check the **Transform Covariates** option.
9. Click **Go**.

The report is shown in Figure 4.6.

Figure 4.6 Neural Report



Model NTanH(3)

Training — mvalue

Measures	Value
RSquare	0.907383
RMSE	2.9259708
Mean Abs Dev	2.12114
-LogLikelihood	1006.9962
SSE	3458.7673
Sum Freq	404

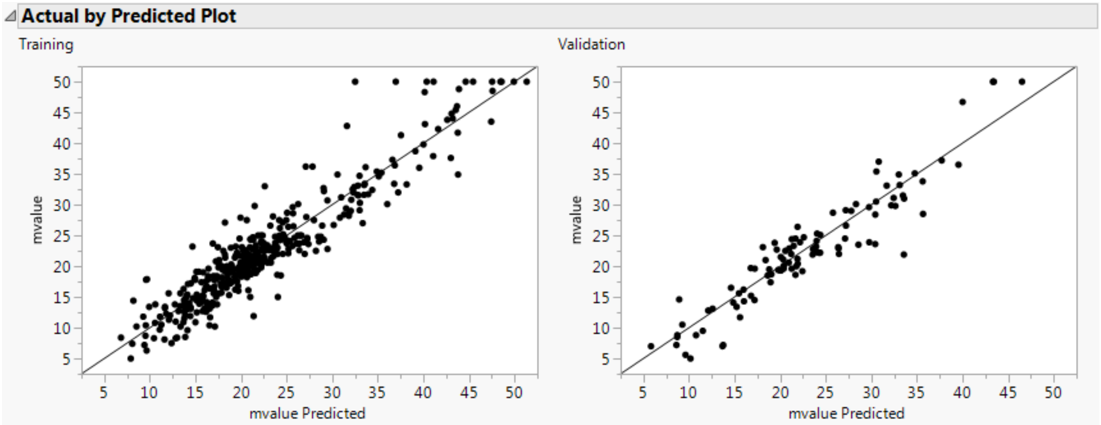
Validation — mvalue

Measures	Value
RSquare	0.7605454
RMSE	3.5003749
Mean Abs Dev	2.3785876
-LogLikelihood	272.52448
SSE	1249.7677
Sum Freq	102

Results are provided for both the training and validation sets. Use the results of the validation set as a representation of the model's predictive power on future observations.

The R-Square statistic for the Validation set is 0.819, 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.

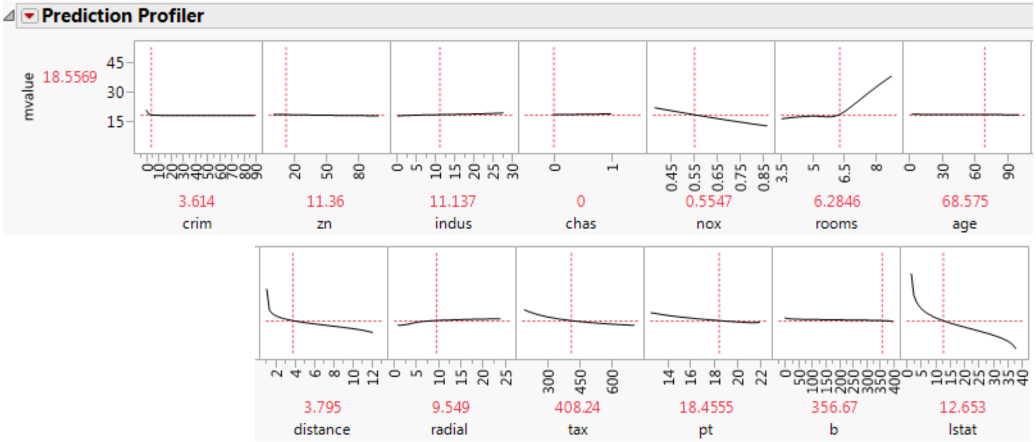
Figure 4.7 Actual by Predicted Plot



The points fall along the line, signifying that the predicted values are similar to the actual values.

To get a general understanding of how the *X* variables are impacting the predicted values, select **Profiler** from the Model red-triangle menu. The profiler is shown in Figure 4.8.

Figure 4.8 Profiler



Some of the variables have profiles with positive slopes, and some negative. For example, rooms has a positive slope. This indicates that the more rooms a home has, the higher the predicted median value. The variable pt is the pupil teacher ratio by town. This variable has a negative slope, indicating that the higher the pupil to teacher ratio, the lower the median value.

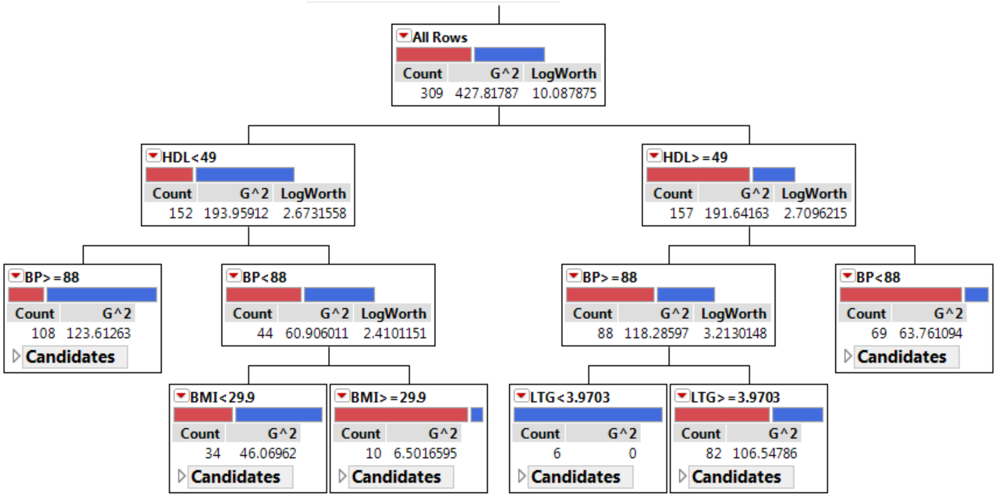
Chapter 5

Partition Models

Use Decision Trees to Explore and Model Your Data

The Partition platform recursively partitions data according to a relationship between the predictors and response values, creating a decision tree. The partition algorithm searches all possible splits of predictors to best predict the response. These splits (or *partitions*) of the data are done recursively to form a tree of decision rules. The splits continue until the desired fit is reached. The partition algorithm chooses optimum splits from a large number of possible splits, making it a powerful modeling, and data discovery tool.

Figure 5.1 Example of a Decision Tree


All Rows Count 309 G^2 427.81787 LogWorth 10.087875
HDL<49 Count 152 G^2 193.95912 LogWorth 2.6731558
HDL>=49 Count 157 G^2 191.64163 LogWorth 2.7096215
BP>=88 Count 108 G^2 123.61263 Candidates
BP<88 Count 44 G^2 60.906011 LogWorth 2.4101151
BP>=88 Count 88 G^2 118.28597 LogWorth 3.2130148
BP<88 Count 69 G^2 63.761094 Candidates
BMI<29.9 Count 34 G^2 46.06962 Candidates
BMI>=29.9 Count 10 G^2 6.5016595 Candidates
LTG<3.9703 Count 6 G^2 0 Candidates
LTG>=3.9703 Count 82 G^2 106.54786 Candidates

Partition Platform Overview

The Partition platform recursively partitions data according to a relationship between the predictors and response values, creating a decision tree. Variations of partitioning go by many names and brand names: decision trees, CART^{TM}, CHAID^{TM}, C4.5, C5, and others. The technique is often considered as a data mining technique for the following reasons:

- it is useful for exploring relationships without having a good prior model
- it handles large problems easily
- the results are interpretable

A classic application of partitioning is to create a diagnostic heuristic for a disease. Given symptoms and outcomes for a number of subjects, partitioning can be used to generate a hierarchy of questions to help diagnose new patients.

Predictors can be either continuous or categorical (nominal or ordinal). If a predictor is continuous, then the splits are created by a *cutting value*. The sample is divided into values below and above this cutting value. If a predictor is categorical, then the sample is divided into two groups of levels.

The response can also be either continuous or categorical (nominal or ordinal). If the response is continuous, then the platform fits the means of the response values. If the response is categorical, then the fitted value is a probability for the levels of the response. In either case, the split is chosen to maximize the difference in the responses between the two nodes of the split.

For more information about split criteria, see “Statistical Details” on page 104.

For more information about recursive partitioning, see Hawkins, D. M., and Kass, G. V. (1982) and Kass, G. B. (1980).

Example of the Partition Platform

In this example, you use the Partition platform to construct a decision tree that predicts the one-year disease progression (low or high) of patients with diabetes.

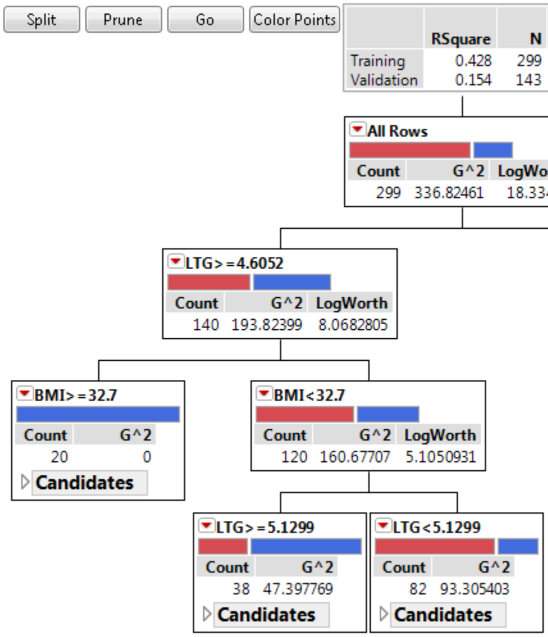
1. Select **Help > Sample Data Library** and Diabetes.jmp.
2. Select **Analyze** > **Predictive Modeling** > **Partition**.
3. Select Y Binary and click **Y, Response**.
4. Select Age through Glucose and click **X, Factor**.
5. Enter 0.33 for the **Validation Portion**.

Note: In JMP Pro, a validation column can be used for validation. Select **Validation** and click **Validation**. Set the **Validation Portion** to 0.

6. Click **OK**.
7. On the platform report window, click **Go** to perform automatic splitting.

Note: Because you are using a random Validation Portion, your results differ from those in Figure 5.2.

Figure 5.2 Partition Report for Diabetes



	RSquare	N	Number of Splits
Training	0.428	299	4
Validation	0.154	143	

All Rows

Count	G^2	LogWorth
299	336.82461	18.33404

LTG>=4.6052

Count	G^2	LogWorth
140	193.82399	8.0682805

LTG<4.6052

Count	G^2	LogWorth
159	63.421961	3.4514003

BMI>=32.7

Count	G^2
20	0

BMI<32.7

Count	G^2	LogWorth
120	160.67707	5.1050931

BMI>=27.3

Count	G^2
30	30.024145

BMI<27.3

Count	G^2
129	20.635492

LTG>=5.1299

Count	G^2
38	47.397769

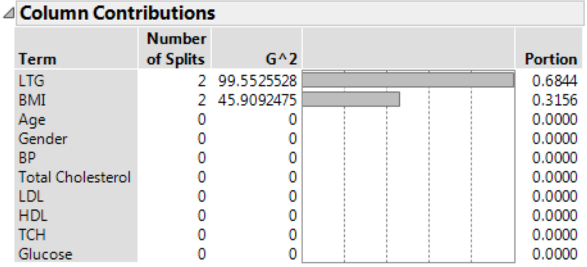
LTG<5.1299

Count	G^2
82	93.305403

Automatic splitting resulted in four splits. The final RSquare for the Validation set is 0.154. The decision tree shows the four splits and the counts of observations in each split.

8. Click the red triangle next to Partition for Y Binary and select **Column Contributions**.

Figure 5.3 Column Contributions Report



Column Contributions

Term	Number of Splits	G^2	Portion
LTG	2	99.5525528	0.6844
BMI	2	45.9092475	0.3156
Age	0	0	0.0000
Gender	0	0	0.0000
BP	0	0	0.0000
Total Cholesterol	0	0	0.0000
LDL	0	0	0.0000
HDL	0	0	0.0000
TCH	0	0	0.0000
Glucose	0	0	0.0000

The Column Contributions report shows that LTG and BMI are the only predictors in the decision tree model. Each column is used in two splits. Your results can differ. When the Validation Portion is used, the validation set is selected at random from the data table. If you redo your analysis, a new random validation set is selected and your results can differ from your first run.

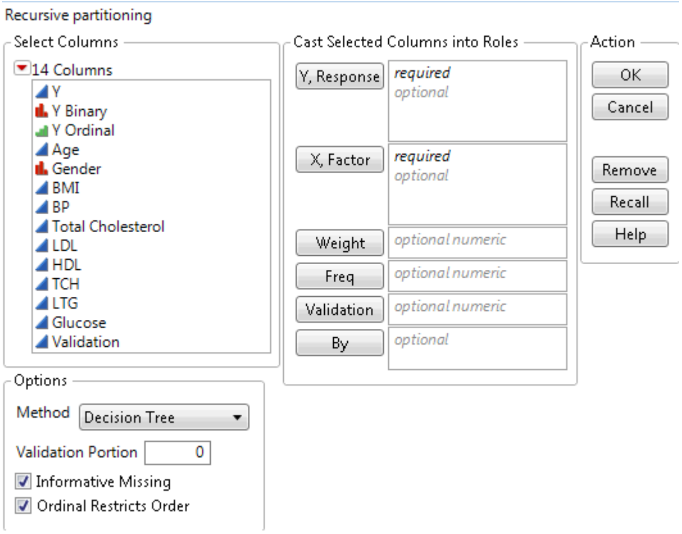
9. Click the red triangle next to Partition for Y Binary and select **Save Columns > Save Prediction Formula**.

   In the Diabetes.jmp data table, columns called Prob(Y Binary==Low), Prob(Y Binary==High), and Most Likely Y Binary are added. To see how these response probabilities are calculated, in the Columns panel, next to each column, double-click the Formula icon ✚ .

Launch the Partition Platform

Launch the Partition platform by selecting **Analyze** > **Predictive Modeling** > **Partition**.

Figure 5.4 Partition Launch Window



Method: Decision Tree

Validation Portion: 0

☑ Informative Missing

☑ Ordinal Restricts Order

Y, Response The response variable or variables that you want to analyze.

X, Factor The predictor variables.

Weight A column whose numeric values assign a weight to each row in the analysis.

Freq A column whose numeric values assign a frequency to each row in the analysis.

Validation A numeric column that contains at most three distinct values. See "Validation" on page 93.

By A column or columns whose levels define separate analyses. For each level of the column, the corresponding rows are analyzed using the other variables that you specify. The results appear in separate reports. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

Method Enables you to select the partition method (Bootstrap Forest, Boosted Tree, K Nearest Neighbors, or Naive Bayes).

For more details about these methods, see Chapter 6, "Bootstrap Forest", Chapter 7, "Boosted Tree", Chapter 8, "K Nearest Neighbors", and Chapter 9, "Naive Bayes".

Validation Portion The portion of the data to be used as the validation set. See "Validation" on page 93.

Informative Missing If selected, enables missing value categorization for categorical predictors and informative treatment of missing values for continuous predictors. See "Informative Missing" on page 88.

Ordinal Restricts Order If selected, restricts consideration of splits to those that preserve the ordering.

The Partition Report

The initial Partition report shows a partition plot, control buttons, a summary panel, and a decision tree. The partition plot and decision tree are initialized without any splits. The reports details are different for categorical and continuous responses.

Control Buttons

Use the control buttons to interact with the decision tree.

Split Creates a partition of the data using the optimal split. To specify multiple splits, hold the Shift key as you click **Split**.

Prune Removes the most recent split.

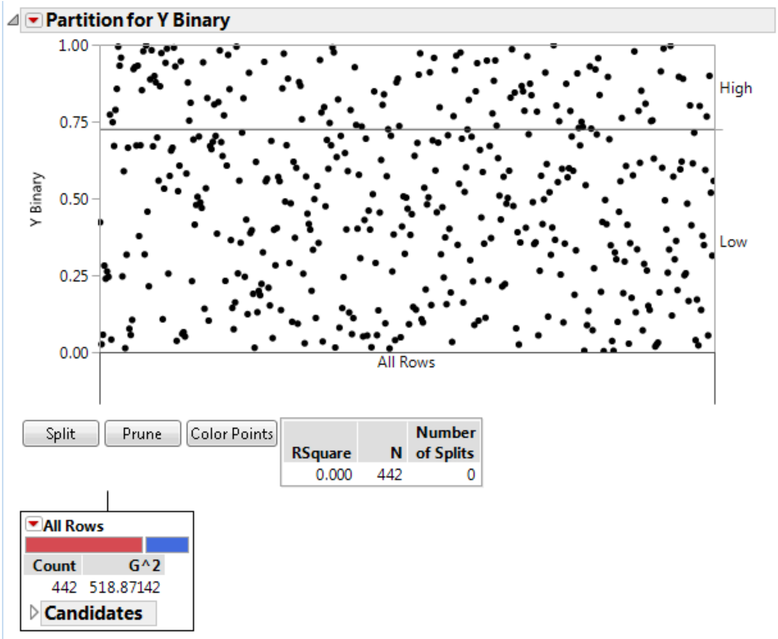
Go (Available when you are using validation.) Automatically adds splits to the decision tree until the validation statistic is optimized. See “Validation” on page 93. Without validation, you simply decide the number of splits to use in the partition model.

Color Points For categorical responses, colors observations according to response level. These colors are added to the data table.

Report for Categorical Responses

The sample data table Diabetes.jmp was used to create a report for the categorical response Y Binary.

Figure 5.5 Partition Report for a Categorical Response



RSquare	N	Number of Splits
0.000	442	0

Count	G^2
442	518.87142

Partition Plot

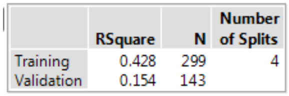
Each point in the Partition Plot represents an observation in the data table. If validation is used, the plot is only for the training data. The initial partition plot does not show splits.

Notice the following:

- The left vertical axis is the proportion of each response outcome.
- The right vertical axis shows the order in which the response levels are plotted.
- Horizontal lines divide each split by the response variable. The initial horizontal line shows the overall proportion of the first plotted response in the data set.
- Splits are shown below the x-axis with a text description and a vertical line that splits the observations in the plot. The vertical lines extend into the plot and indicate the boundaries for each node. The most recent split appears directly below the horizontal axis and on top of existing splits. The plot is updated with each split or prune of the decision tree.

Summary Report

Figure 5.6 Summary Report for a Categorical Response



	RSquare	N	Number of Splits
Training	0.428	299	4
Validation	0.154	143	

The Summary Report provides fit statistics for the training data and validation and test data (if used). The fit statistics in the Summary Panel update as you add splits or prune the decision tree.

RSquare The current value of R^2.

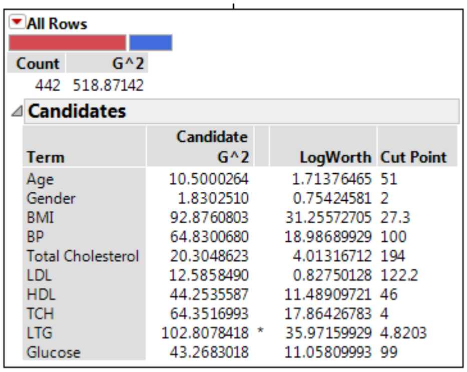
N Number of observations.

Number of Splits Current number of splits in the decision tree.

Node Reports

Each node in the tree has a report and a red triangle menu with additional options. Terminal nodes also have a Candidates report.

Figure 5.7 Terminal Node Report for a Categorical Response



All Rows

Count	G^2
442	518.87142

Candidates

Term	Candidate G^2		LogWorth	Cut Point
Age	10.5000264		1.71376465	51
Gender	1.8302510		0.75424581	2
BMI	92.8760803		31.25572705	27.3
BP	64.8300680		18.98689929	100
Total Cholesterol	20.3048623		4.01316712	194
LDL	12.5858490		0.82750128	122.2
HDL	44.2535587		11.48909721	46
TCH	64.3516993		17.86426783	4
LTG	102.8078418	*	35.97159929	4.8203
Glucose	43.2683018		11.05809993	99

Count Number of training observations that are characterized by the node.

G^2 A fit statistic used for categorical responses (instead of sum of squares that is used for continuous responses). Lower values indicate a better fit. See "Statistical Details" on page 104.

Candidates For each column, the Candidates report provides details about the optimal split for that column. The optimal split over all terms is marked with an asterisk.

Term Shows the candidate columns.

Candidate G^2 Likelihood ratio chi-square for the best split. Splitting on the predictor with the largest G^2 maximizes the reduction in the model G^2.

LogWorth The LogWorth statistic, defined as $-\log_{10}(p\text{-value})$. The optimal split is the one that maximizes the LogWorth. See "Statistical Details" on page 104 for additional details.

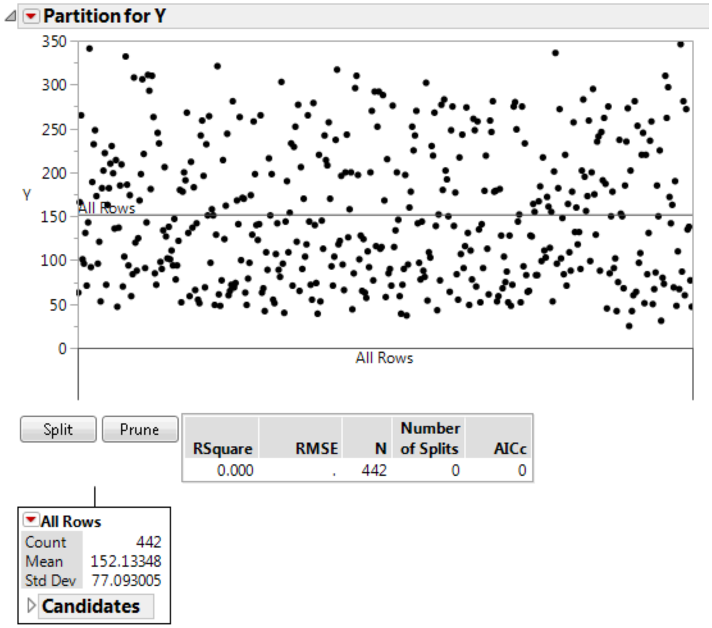
Cut Point The value of the predictor that determines the split. For a categorical term, the levels in the left-most split are listed.

The optimal split is noted by an asterisk. However, there are cases where the Candidate G^2 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 104 for details.

Report for Continuous Responses

The sample data table Diabetes.jmp was used to create a report for the continuous response Y.

Figure 5.8 Partition Report for a Continuous Response



RSquare	RMSE	N	Number of Splits	AICc
0.000	.	442	0	0

All Rows

Count	442
Mean	152.13348
Std Dev	77.093005

Candidates

Partition Plot

The partition plot is initialized without any splits. Each point represents an observation in the data table. If validation is used, the plot is only for the training data.

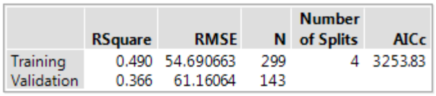
Notice the following:

- The vertical axis represents the response value of the observations.
- Horizontal lines show the mean response value for each node of the decision tree. The initial horizontal line is at the overall mean of the response.
- Vertical axis divisions represent splits in the decision tree. A text description of the most recent split appears below the horizontal axis. Observations are reorganized into their respective nodes as splits are created or removed.

Tip: To see tooltips for narrow partitions, place your cursor over the labels on the horizontal axis of the partition plot.

Summary Report

Figure 5.9 Summary Report for a Continuous Response



	RSquare	RMSE	N	Number of Splits	AICc
Training	0.490	54.690663	299	4	3253.83
Validation	0.366	61.16064	143		

The Summary Report provides fit statistics for the training data and validation and test data (if used). The fit statistics in the Summary Panel update as you add splits or prune the decision tree.

RSquare The current value of R^2.

RMSE The root mean square error.

N The number of observations.

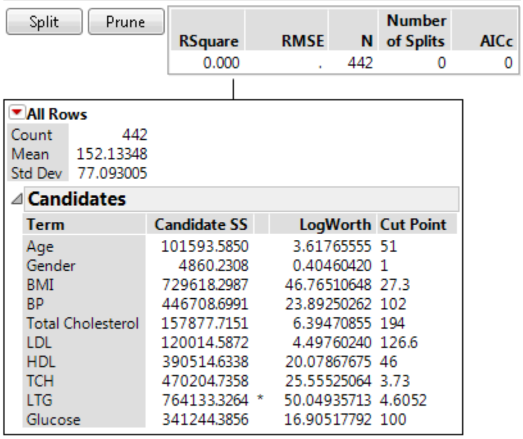
Number of Splits The current number of splits in the decision tree.

AICc The corrected Akaike’s Information Criterion. For more details, see the Statistical Details appendix in the *Fitting Linear Models* book.

Node Reports

Each node in the tree has a report and a red triangle menu with additional options. Terminal nodes also have a Candidates report.

Figure 5.10 Terminal Node Report for a Continuous Response



RSquare	RMSE	N	Number of Splits	AICc
0.000	.	442	0	0

All Rows

Count	442
Mean	152.13348
Std Dev	77.093005

Candidates

Term	Candidate SS		LogWorth	Cut Point
Age	101593.5850		3.61765555	51
Gender	4860.2308		0.40460420	1
BMI	729618.2987		46.76510648	27.3
BP	446708.6991		23.89250262	102
Total Cholesterol	157877.7151		6.39470855	194
LDL	120014.5872		4.49760240	126.6
HDL	390514.6338		20.07867675	46
TCH	470204.7358		25.55525064	3.73
LTG	764133.3264	*	50.04935713	4.6052
Glucose	341244.3856		16.90517792	100

Count The number of observations (rows) in the branch.

Mean The average response for all observations in that branch.

Std Dev The standard deviation of the response for all observations in that branch.

Candidates For each column, the Candidates report provides details about the optimal split for that column. The optimal split over all columns is marked with an asterisk.

Term Shows the candidate columns.

Candidate SS Sum of squares for the best split.

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 104 for additional details.

Cut Point The value of the predictor that determines the split. For a categorical term, the levels in the left-most split are listed.

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 104 for details.

Partition Platform Options

The Partition red triangle menu options give you the ability to customize reports according to your needs. The available options are determined by the type of data that you use for your analysis.

Display Options Contains options that show or hide report elements.

Show Points Shows the points. For categorical responses, this option shows the points or colored panels.

Show Tree Shows the large tree of partitions.

Show Graph Shows the partition graph.

Show Split Bar (Categorical responses only) Shows the colored bars that indicate the split proportions in each leaf.

Show Split Stats Shows the split statistics. For more information about the categorical split statistic G^2, see "Statistical Details" on page 104.

Show Split Prob (Categorical responses only) Shows the Rate and Prob statistics in the node reports.

JMP automatically shows the Rate and Prob statistics when you select **Show Split Count**. For more information about Rate and Prob, see "Statistical Details" on page 104.

Show Split Count (Categorical responses only) Shows frequency counts in the node reports. When you select this option, JMP automatically selects **Show Split Prob**. And when you de-select **Show Split Prob**, the counts do not appear.

Show Split Candidates Shows the Candidates report.

Sort Split Candidates Sorts the Candidates reports by the statistic or the log(worth), whichever is appropriate.

Split Best Splits the tree at the optimal split point. This is equivalent to clicking the **Split** button.

Prune Worst Removes the terminal split that has the least discrimination ability. This is equivalent to clicking the **Prune** button.

Minimum Size Split Define the minimum size split allowed by entering a number or a fractional portion of the total sample size. 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 maximum of 5, or the floor of the number of rows divided by 10,000.

Lock Columns Interactively lock columns so that they are not considered for splitting. You can turn the display off or back on without affecting the individual locks.

Plot Actual by Predicted (Continuous responses only) Shows a plot of actual values by predicted values. See "Actual by Predicted Plot" on page 89.

Small Tree View Shows a small version of the partition tree to the right of the partition plot.

Tree 3D Shows a 3-D plot of the tree structure. To access this option, hold down the Shift key and click the red triangle menu.

Leaf Report Shows the mean and count or rates for the bottom-level leaves of the report.

Column Contributions Shows a report indicating 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.

Split History Shows a plot of RSquare versus the number of splits. If you use excluded row validation, holdback validation, or a validation column, separate curves are drawn for training and validation RSquare values. The RSquare curve is blue for the training set and red for the validation set. If you select K Fold Crossvalidation, the RSquare curve for all of the data is blue, and the curve for the crossvalidation RSquare is green.

K Fold Crossvalidation Shows a Crossvalidation report that gives fit statistics for both the training and folded sets. For more information about validation, see "K-Fold Crossvalidation" on page 94.

ROC Curve (Categorical responses only) Receiver Operating Characteristic (ROC) curves display the efficiency of a model's fitted probabilities to sort the response levels. See "ROC Curve" on page 90.

Lift Curve (Categorical responses only) Lift curves display the predictive ability of a partition model. See "Lift Curve" on page 91.

Show Fit Details (Appears only for categorical responses.) The Fit Details report shows several measures of fit and provides a Confusion Matrix report. See "Show Fit Details" on page 85

Save Columns Contains options 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. Each branch is separated by "&". An example label might 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 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.

JMP PRO **Publish Prediction Formula** Creates a prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169.

JMP PRO **Publish Tolerant Prediction Formula** Creates a tolerant prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169. If you have checked Informative Missing, you can use this option by holding the Shift key as you click on the report's red triangle.

Specify Profit Matrix (Available only for categorical responses.) Enables you to specify profits or costs associated with correct or incorrect classification decisions. For a nominal response, you can specify the profit matrix entries using a probability threshold. See "Show Fit Details" on page 85.

Profiler Shows an interactive profiler report. Changes in the factor values are reflected in the estimated classification probabilities. See the Profiler chapter in the *Profilers* book.

Color Points (Categorical responses only) Colors points based on their response level. This is equivalent to clicking the Color Points button. See "Report for Categorical Responses" on page 76.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

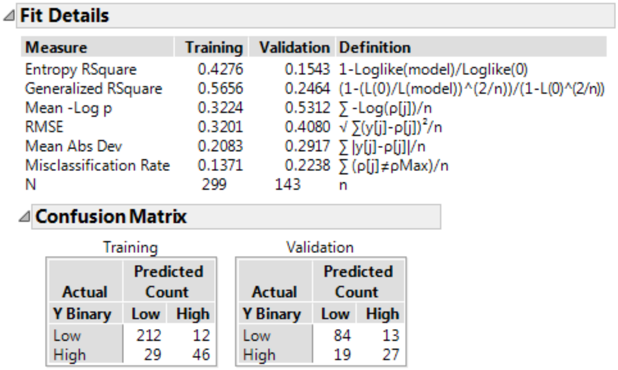
Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Show Fit Details

Figure 5.11 Fit Details for Categorical Response (Y Binary from Diabetes.jmp)



Fit Details

Measure	Training	Validation	Definition
Entropy RSquare	0.4276	0.1543	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5656	0.2464	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3224	0.5312	∑ -Log(ρ[j])/n
RMSE	0.3201	0.4080	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2083	0.2917	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1371	0.2238	∑ (ρ[j]≠ρMax)/n
N	299	143	n

Confusion Matrix

Training

Actual	Predicted Count	
Y Binary	Low	High
Low	212	12
High	29	46

Validation

Actual	Predicted Count	
Y Binary	Low	High
Low	84	13
High	19	27

Entropy RSquare Compares the log-likelihoods from the fitted model and the constant probability model. Values closer to 1 indicate a better fit.

Generalized RSquare A measure that can be applied to general regression models. It is based on the likelihood function L and is scaled to have a maximum value of 1. The Generalized RSquare measure simplifies to the traditional RSquare for continuous normal responses in the standard least squares setting. Generalized RSquare is also known as the Nagelkerke or Craig and Uhler R^2, which is a normalized version of Cox and Snell's pseudo R^2. See Nagelkerke (1991). Values closer to 1 indicate a better fit.

Mean -Log p The average of -log(p), where p is the fitted probability associated with the event that occurred. Smaller values indicate a better fit.

RMSE The root mean square error, where the differences are between the response and p (the fitted probability for the event that actually occurred). Smaller values indicate a better fit.

Mean Abs Dev The average of the absolute values of the differences between the response and p (the fitted probability for the event that actually occurred). Smaller values indicate a better fit.

Misclassification Rate The rate for which the response category with the highest fitted probability is not the observed category. Smaller values indicate a better fit.

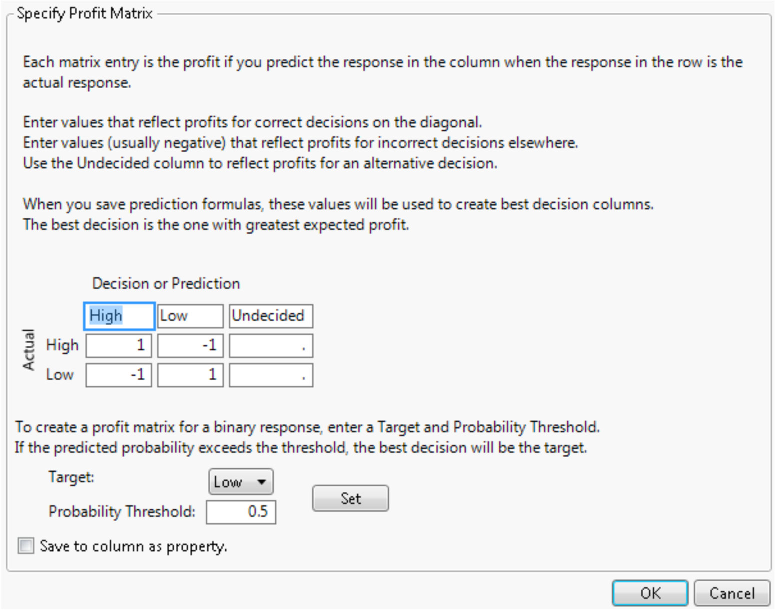
The Confusion Matrix report shows matrices for the training set and for the validation and test sets (if defined). The Confusion Matrix is a two-way classification of actual and predicted responses.

If the response has a Profit Matrix column property, or if you specify costs using the Specify Profit Matrix option, then a Decision Matrix report appears. See "Decision Matrix Report" on page 87.

Specify Profit Matrix

A profit matrix can be used with categorical responses. A profit matrix is used to assign costs to undesirable outcomes and profits to desirable outcomes.

Figure 5.12 Specify Profit Matrix Window



Specify Profit Matrix

Each matrix entry is the profit if you predict the response in the column when the response in the row is the actual response.

Enter values that reflect profits for correct decisions on the diagonal.
Enter values (usually negative) that reflect profits for incorrect decisions elsewhere.
Use the Undecided column to reflect profits for an alternative decision.

When you save prediction formulas, these values will be used to create best decision columns.
The best decision is the one with greatest expected profit.

Decision or Prediction

Actual	High	Low	Undecided
High	1	-1	.
Low	-1	1	.

To create a profit matrix for a binary response, enter a Target and Probability Threshold.
If the predicted probability exceeds the threshold, the best decision will be the target.

Target: Low

Probability Threshold: 0.5

□ Save to column as property.

You can assign profit and cost values to each combination of actual and predicted response categories. To specify the costs of classifying into an alternative category, enter values in the Undecided column. To save your assignments to the response column as a property, check **Save to column as property.** Leaving this option unchecked applies the Profit Matrix only to the current Partition report.

Probability Threshold Specification for Profit Matrix

When the response is binary, instead of entering weights into the profit matrix, you can specify a probability threshold in the Profit Matrix window. For details about how values are calculated for the profit matrix, see The Column Info Window chapter in the *Using JMP* book.

Target The level whose probability is modeled.

Probability Threshold A threshold for the probability of the target level. If the probability that an observation falls into the target level exceeds the probability threshold, the observation is classified into that level.

When you define costs using the Specify Profit Matrix option and then select Show Fit Details, a Decision Matrix report appears. See "Decision Matrix Report" on page 87.

When you specify a profit matrix and save the model prediction formula, the formula columns saved to the data table include the following:

– Profit for <level>: For each level of the response, a column gives the expected profit for classifying each observation into that level.

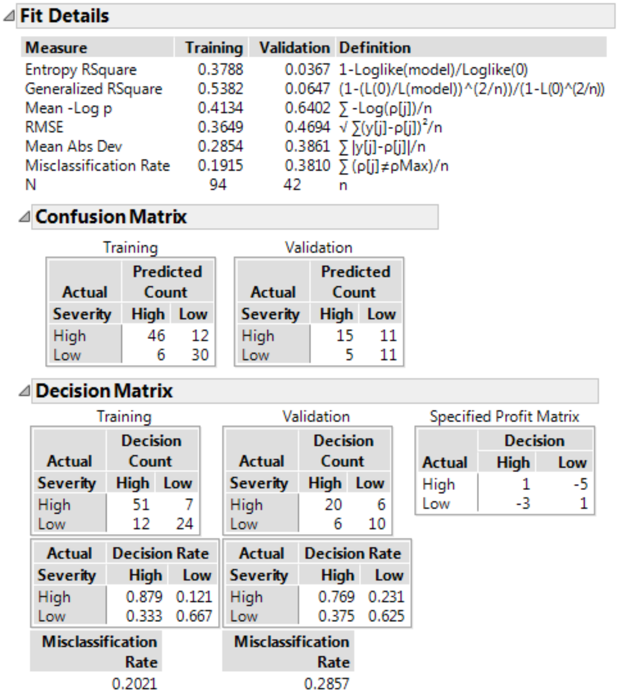
– Most Profitable Prediction for <column name>: For each observation, gives the level of the response with the highest expected profit.

– 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.

Decision Matrix Report

Figure 5.13 Fit Details Report with Decision Matrix Report



Fit Details

Measure	Training	Validation	Definition
Entropy RSquare	0.3788	0.0367	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5382	0.0647	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4134	0.6402	∑ -Log(ρ[j])/n
RMSE	0.3649	0.4694	√ ∑(y[j]-ρ[j])^2/n
Mean Abs Dev	0.2854	0.3861	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1915	0.3810	∑ (ρ[j]≠ρMax)/n
N	94	42	n

Confusion Matrix

Training

Actual	Predicted Count	
Severity	High	Low
High	46	12
Low	6	30

Validation

Actual	Predicted Count	
Severity	High	Low
High	15	11
Low	5	11

Decision Matrix

Training

Actual	Decision Count	
Severity	High	Low
High	51	7
Low	12	24

Actual	Decision Rate	
Severity	High	Low
High	0.879	0.121
Low	0.333	0.667

Misclassification Rate
0.2021

Validation

Actual	Decision Count	
Severity	High	Low
High	20	6
Low	6	10

Actual	Decision Rate	
Severity	High	Low
High	0.769	0.231
Low	0.375	0.625

Misclassification Rate
0.2857

Specified Profit Matrix

	Decision	
Actual	High	Low
High	1	-5
Low	-3	1

Note: This report is available only if the response has a Profit Matrix column property or if you specify costs using the Specify Profit Matrix option. The report is part of the Fit Details report.

When a profit matrix is defined, the partition algorithm uses the values in the matrix to calculate the profit for each decision. When you select Show Fit Details, a Decision Matrix report appears.

In the Decision Matrix report, the decision counts reflect the most profitable prediction decisions based on the weighting in the profit matrix. The report gives Decision Count and Decision Rate matrices for the training set and for validation and test sets (if defined). For reference, the profit matrix is also shown.

Note: If you change the weights in your Profit Matrix using the Specify Profit Matrix option, the Decision Matrix report automatically updates to reflect your changes.

Decision Count Matrix Shows a two-way classification with actual responses in rows and classification counts in columns.

Specified Profit Matrix Gives the weights that define the Profit Matrix.

Decision Rate Matrix Shows rate values corresponding to the proportion of a given row's observations that are classified into each category. If all observations are correctly classified, the rates on the diagonal are all equal to one.

Tip: You can obtain a decision rate matrices for a response using the default profit matrix with costs of 1 and -1. Select **Specify Profit Matrix** from the red triangle menu, make no changes to the default values, and click **OK**.

The matrices are arranged in two rows:

- The Decision Count matrices are in the first row.
- The Specified Profit Matrix is to the right in the first row.
- The Decision Rate matrices are in the second row.

Informative Missing

The Informative Missing option enables informative treatment of missing values on the predictors. The model that is fit is deterministic. The Informative Missing option is found on the launch window and is selected by default. When informative missing is selected 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

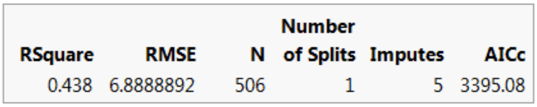
further splits on the given predictor, the algorithm commits the missing rows to high or low values, as determined by the first split induced by that predictor.

If the Informative Missing option is not selected, the missing values are handled as follows:

- When a predictor with missing values is used as a splitting variable, each row with a missing value on that predictor is randomly assigned to one of the two sides of the split.
- The first time a predictor with missing values is used as a splitting variable an *Imputes* column is added to the Summary Report showing the number of imputations. As additional imputations are made, the Imputes column updates. See Figure 5.14, where five imputations were performed.

Note: The number of Imputes can be greater than the number of rows that contain missing values. The imputation occurs at each split. A row with missing values can be randomly assigned multiple times. Each time a row is randomly assigned it increments the imputation count.

Figure 5.14 Impute Message in Summary Report



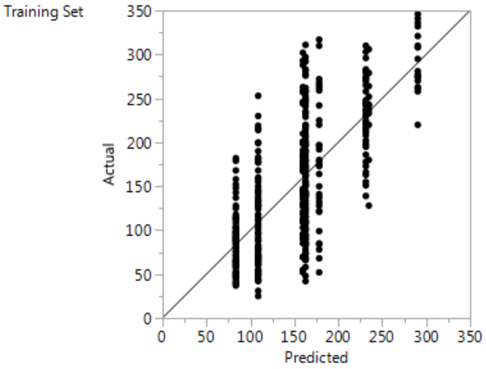
RSquare	RMSE	N	Number of Splits	Imputes	AICc
0.438	6.8888892	506	1	5	3395.08

Actual by Predicted Plot

For continuous responses, the Actual by Predicted plot is the typical plot of the actual response versus the predicted response. When you fit a Decision Tree, all 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. The actual values form a scatter of points around each leaf mean on *n* vertical lines.

The diagonal line is the Y = X line. 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 5.15.

Figure 5.15 Actual by Predicted Plots for a Continuous Response



ROC Curve

The ROC Curve option is available only for categorical responses. Receiver Operating Characteristic (ROC) curves display the efficiency of a model's fitted probabilities in sorting the response levels. An introduction to ROC curves is found in the Logistic Analysis chapter in the *Basic Analysis* book.

The predicted response for each observation in a partition model is a value between 0 and 1. To use the predicted response to classify observations as positive or negative, a *cut point* is used. For example, if the cut point is 0.5, an observation with a predicted response at or above 0.5 would be classified as positive, and an observation below 0.5 as negative. There are trade offs in classification as the cut point is varied.

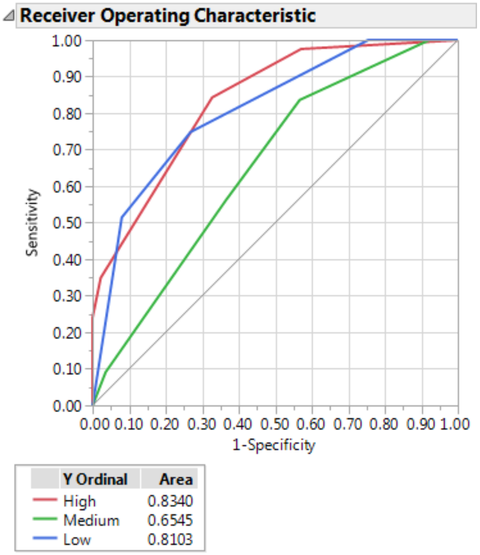
To generate a ROC curve, each predicted response level is considered as a possible cut point and the following values are computed for each possible cut point:

- The *sensitivity* is the proportion of true positives or the percent of positive observations with a predicted response greater than the cut point.
- The *specificity* is the proportion of true negatives or the proportion of negative observations with a predicted response less than the cut point.

The ROC curve plots sensitivity against 1 - specificity. A partition model with *n* splits has *n*+1 predicted values. The ROC curve for the partition model has *n*+1 line segments.

If your response has more than two levels, the Partition report contains a separate ROC curve for each response level versus the other levels. Each curve is the representation of a level as the positive response level. If there are only two levels, one curve is the reflection of the other.

Figure 5.16 ROC Curves for a Three Level Response



Receiver Operating Characteristic

	Y Ordinal	Area
—	High	0.8340
—	Medium	0.6545
—	Low	0.8103

If the model perfectly rank-orders the response values, then the sorted data contains all of the positive values first, followed by all of the other values. In this situation, the curve moves all the way to the top before it moves at all to the right. If the model does not predict well, the curve follows the diagonal line from the bottom left to top right of the plot.

In practice, the ROC curve lies above the diagonal. The area under the curve is the indicator of the goodness of fit for the model. A value of 1 indicates a perfect fit and a value near 0.5 indicates that the model cannot discriminate among groups.

When your response has more than two levels, the ROC curve plot enables you to see which response categories have the largest area under the curve.

Lift Curve

The Lift Curve option provides another plot to display the predictive ability of a partition model. The lift curve plots the lift versus the portion of the observations. Each predicted response level defines a portion of the observations that are greater than or equal to that predicted response. The *lift* value is the ratio of the proportion of positive responses in that portion to the overall proportion of positive responses.

Figure 5.17 Lift Curve

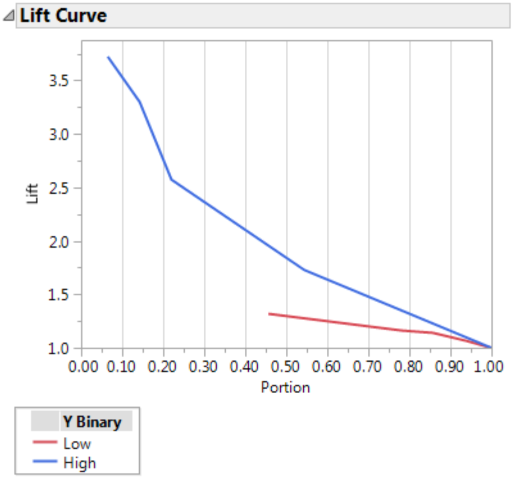
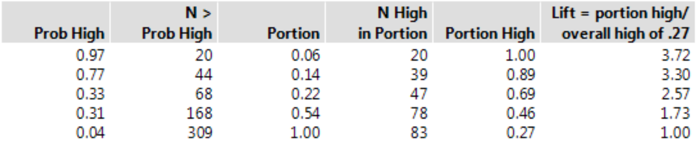


Figure 5.18 Lift Table for Lift Curve



	N >		N High		Lift = portion high/
Prob High	Prob High	Portion	in Portion	Portion High	overall high of .27
0.97	20	0.06	20	1.00	3.72
0.77	44	0.14	39	0.89	3.30
0.33	68	0.22	47	0.69	2.57
0.31	168	0.54	78	0.46	1.73
0.04	309	1.00	83	0.27	1.00

Figure 5.18 provides a table of values to demonstrate the calculation of Lift and Portion used for the High lift curve shown in Figure 5.17. A partition model with five splits was built to predict the response, Y Binary. Y Binary has two levels: Low and High. The lift curve is based on 309 observations. There are 83 High responses for an overall rate of 0.27.

- Prob High: The five predicted values from the partition model for the High response level.
- N > Prob High: The number of observations that have a predicted value equal to or greater than the value in Prob High.
- Portion: N > Prob High divided by 309.
- N High in Portion: The number of High responses in the portion.
- Portion High: N > Prob High divided by Portion High.
- Lift: Portion High divided by 0.27.

Lift measures how many High responses fall in each portion as compared to the expected number of High responses for that portion. For the first 6% of the data set the lift is 3.72. Using the model to select the 6% of the observations with the highest predicted values results in 3.72 more High responses than if that 6% were selected at random.

Node Options

This section describes the options on the red triangle menu for each node.

Split Best Finds and executes the best split at or below this node.

Split Here Splits at the selected node on the best column to split by.

Split Specific Lets you specify where a split takes place. This is useful in showing what the criterion is as a function of the cut point, as well as in determining custom cut points. When specifying a splitting column, you can choose the following options for how the split is performed:

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 appears in the node title.

Validation

If you build a tree with enough splits, partitioning can overfit data. When this happens, the model predicts the data used to build the model 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 is used to estimate 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 "Launch the Partition Platform" on page 74.

When a validation method is used, the **Go** button appears. The **Go** button provides for repeated splitting without having to repeatedly click the **Split** button. When you click the **Go** button, splitting occurs until the validation R-Square is better than what the next 10 splits would obtain. This rule can result in 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.

The training, validation, and test sets are created by subsetting the original data into parts. Select one of the following methods to subset a data set:

Excluded Rows Uses row states to subset the data. Rows that are unexcluded are used as the training set, and excluded rows are used as the validation set.

For more information about using row states and how to exclude rows, see the Enter and Edit Data chapter in the *Using JMP* book.

Holdback Randomly divides the original data into the training and validation data sets. The Validation Portion on the platform launch window is used to specify the proportion of the original data to use as the validation data set (holdback). See "Launch the Partition Platform" on page 74 for details about the Validation Portion.

KFold Crossvalidation Randomly 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 final model is selected based on the cross validation RSquare, where a stopping rule is imposed to avoid overfitting the model. This method is useful for small data sets, because it makes efficient use of limited amounts of data. See "K-Fold Crossvalidation" on page 94.

Validation Column Uses a column's values to divide the data into subsets. A validation column must contain at most three numeric values. The column is assigned using the Validation role on the Partition launch window. See "Launch the Partition Platform" on page 74.

The column's values determine how the data is split:

- If the validation column has two levels, the smaller value defines the training set and the larger value defines the validation set.
- If the validation column has three levels, the values, in order of increasing size, define the training, validation, and test sets.

If you click the Validation button with no columns selected in the Select Columns list, you can add a validation column to your data table. For more information about the Make Validation Column utility, see "Make Validation Column Utility".

K-Fold Crossvalidation

In K-Fold cross validation, the entire set of observations is partitioned into *K* subsets, called *folds*. Each fold is treated as a holdback sample with the remaining observations as a training set.

Unconstrained optimization of the crossvalidation RSquare value tends to overfit models. To address this tendency, the KFold crossvalidation stopping rule terminates stepping when improvement in the crossvalidation RSquare is minimal. Specifically, the stopping rule selects a model for which none of the next ten models have a crossvalidation RSquare showing an improvement of more than 0.005 units.

When you select the K Fold Crossvalidation option, a Crossvalidation report appears. The results in this report update as you split the decision tree. Or, if you click Go, the outline shows the results for the final model.

Crossvalidation Report

The Crossvalidation report shows the following:

k-fold Number of folds.

-2LogLike or SSE Gives twice the negative log-likelihood (-2LogLikelihood) values when the response is categorical. Gives sum of squared errors (SSE) when the response is continuous. The first row gives results averaged over the folds. The second row gives results for the single model fit to all observations.

RSquare The first row gives the RSquare value averaged over the folds. The second row gives the RSquare value for the single model fit to all observations.

Additional Examples of Partitioning

The following examples illustrate a continuous response, missing data in the predictors, and the use of the profit matrix.

Example of a Continuous Response

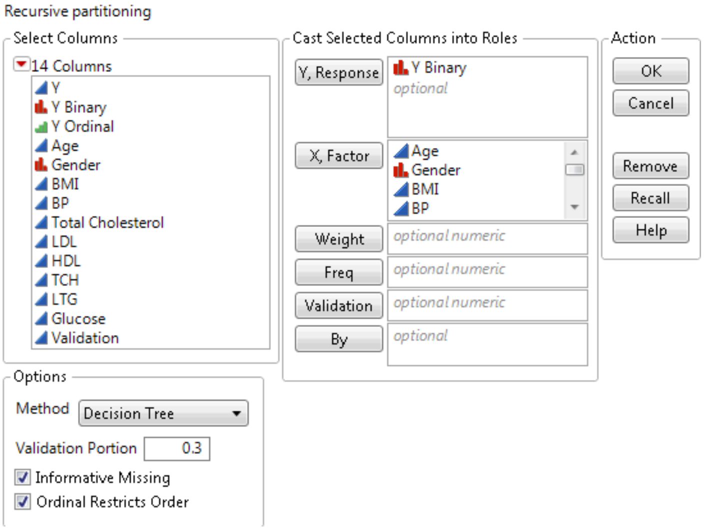
In this example, you use the Partition platform to construct a decision tree that predicts the one-year disease progression measured on a quantitative scale for patients with diabetes.

1. Select **Help > Sample Data Library** and Diabetes.jmp.
2. Select **Analyze** > **Predictive Modeling** > **Partition**.
3. Select Y and click **Y, Response**.
4. Select Age through Glucose and click **X, Factor**.
5. Select a validation procedure based on your JMP installation:
   – For JMP Pro, select Validation and click **Validation**.
   – For JMP, enter 0.3 as the **Validation Proportion**.

   The completed launch window for JMP users is shown in Figure 5.19.

Note: Results using the validation proportion can differ from those shown here due to the random selection of validation rows.

Figure 5.19 Completed Launch Window with Validation Portion = 0.3



Y, Response: Y Binary

X, Factor: Age, Gender, BMI, BP

Method: Decision Tree

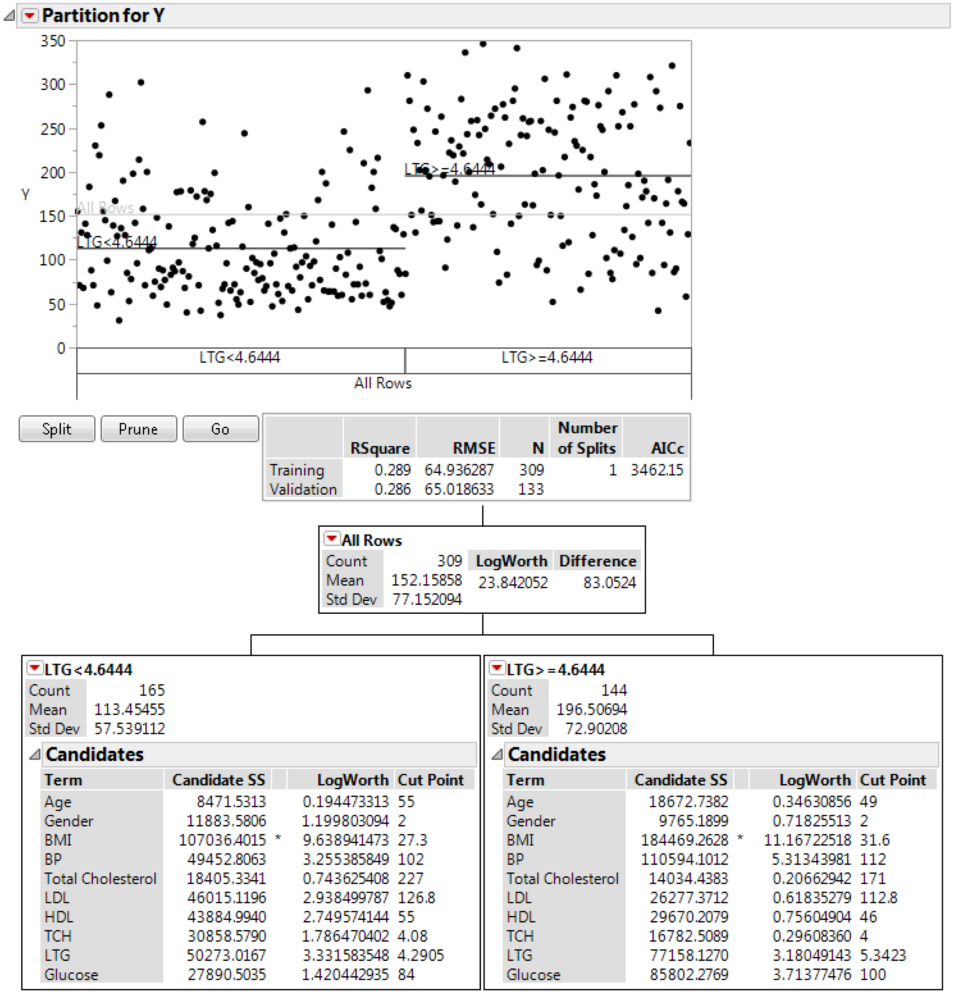
Validation Portion: 0.3

☑ Informative Missing

☑ Ordinal Restricts Order

6. Click **OK**.
7. On the platform report window, click **Split** once to perform a split.

Figure 5.20 Report after First Split with Decision Tree Hidden



	RSquare	RMSE	N	Number of Splits	AICc
Training	0.289	64.936287	309	1	3462.15
Validation	0.286	65.018633	133		

All Rows

		LogWorth	Difference
Count	309	23.842052	83.0524
Mean	152.15858		
Std Dev	77.152094		

LTG<4.6444

Count	165
Mean	113.45455
Std Dev	57.539112

Candidates

Term	Candidate SS		LogWorth	Cut Point
Age	8471.5313		0.194473313	55
Gender	11883.5806		1.199803094	2
BMI	107036.4015	*	9.638941473	27.3
BP	49452.8063		3.255385849	102
Total Cholesterol	18405.3341		0.743625408	227
LDL	46015.1196		2.938499787	126.8
HDL	43884.9940		2.749574144	55
TCH	30858.5790		1.786470402	4.08
LTG	50273.0167		3.331583548	4.2905
Glucose	27890.5035		1.420442935	84

LTG>=4.6444

Count	144
Mean	196.50694
Std Dev	72.90208

Candidates

Term	Candidate SS		LogWorth	Cut Point
Age	18672.7382		0.34630856	49
Gender	9765.1899		0.71825513	2
BMI	184469.2628	*	11.16722518	31.6
BP	110594.1012		5.31343981	112
Total Cholesterol	14034.4383		0.20662942	171
LDL	26277.3712		0.61835279	112.8
HDL	29670.2079		0.75604904	46
TCH	16782.5089		0.29608360	4
LTG	77158.1270		3.18049143	5.3423
Glucose	85802.2769		3.71377476	100

The original 309 values in the training data set are now split into two parts:

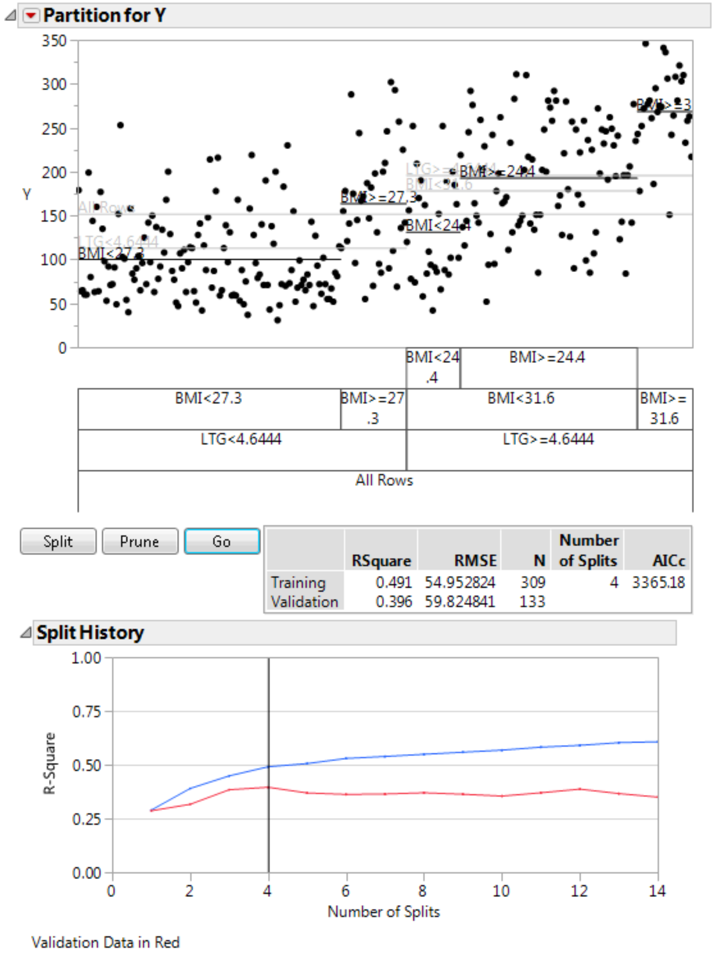
– The left leaf, corresponding to LTG < 4.6444, has 165 observations.

– The right leaf, corresponding to LTG >= 4.6444 has 144 observations.

For both the right and left leaf the next split would be on BMI. The Candidate SS for BMI on the right leaf is higher than the Candidate SS for BMI on the left leaf. Thus, the next split is on the right leaf.

8. Click **Go** to use automatic splitting.

Figure 5.21 Report after Automatic Splitting with Validation



	RSquare	RMSE	N	Number of Splits	AICc
Training	0.491	54.952824	309	4	3365.18
Validation	0.396	59.824841	133		

The solution found has four splits. The Split History plot shows that there is no further improvement in the validation data set after four splits. The RSquare value of 0.39 on the validation data does not support this model as a strong predictor of disease progression. The scatter across partitions in the partition plot further indicate that this model does not separate the Y levels well.

Example of Informative Missing

In this example, you construct a decision tree model to predict if a customer is a credit risk. Since your data set contains missing values, you also explore the effectiveness of the Informative Missing option.

Launch the Partition Platform

1. Select **Help > Sample Data Library** and open Equity.jmp.
2. Select **Analyze** > **Predictive Modeling** > **Partition**.
3. Select BAD and click **Y, Response**.
4. Select LOAN through DEBTINC and click **X, Factor**.
5. Click **OK**.

Create the Decision Tree and ROC Curve with Informative Missing

1. Hold down the Shift key and click **Split**.
2. Enter 5 for the number of splits and click **OK.**
3. Click the red triangle next to Partition for BAD and select **ROC Curve**.
4. Click the red triangle next to Partition for BAD and select **Save Columns > Save Prediction Formula**.

   The columns Prob(BAD==Good Risk) and Prob(BAD==Bad Risk) contain the formulas that Informative Missing utility uses to classify the credit risk of future loan applicants. You are interested in how this model performs in comparison to a model that does not use informative missing.

Create the Decision Tree and ROC Curve without Informative Missing

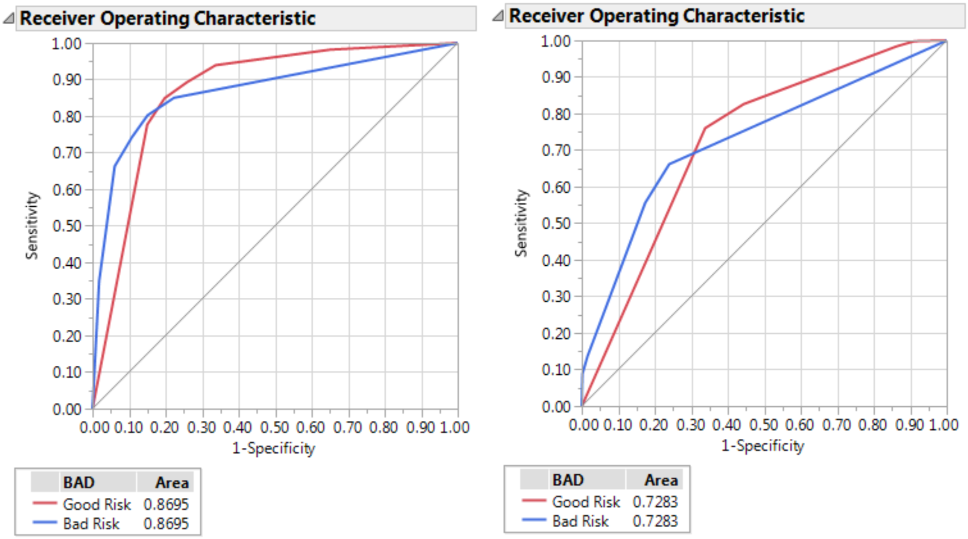
1. Click the red triangle next to Partition for BAD and select **Redo > Relaunch Analysis**
2. De-select **Informative Missing**.
3. Click OK and repeat the steps in "Create the Decision Tree and ROC Curve with Informative Missing".

   The columns Prob(BAD==Good Risk) 2 and Prob(BAD==Bad Risk) 2 contain the formulas that do not use the informative missing utility.

Compare the ROC Curves

Visually compare the ROC curves from the two models. The model at left is with Informative Missing, and the model at right is without Informative Missing.

Figure 5.22 ROC Curves for Models with (Left) and without (Right) Informative Missing



BAD	Area
— Good Risk	0.8695
— Bad Risk	0.8695

BAD	Area
— Good Risk	0.7283
— Bad Risk	0.7283

The area under the curve (AUC) for the model with informative missing (0.8695) is higher than the AUC for the model without informative missing (0.7283). Because there are only two levels for the response, the ROC curves for each model are reflections of one another and the AUCs are equal.

Note: Your AUC can differ from that shown for the model without informative missing. When informative missing is not used, the assignment of missing rows to sides of a split is random. Rerunning the analysis can result in slight differences in results.

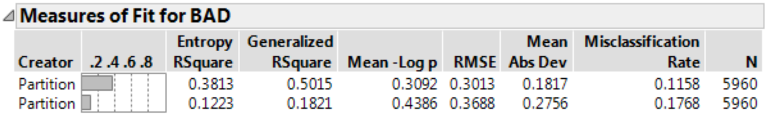
Use the Model Comparison Platform

Next, compare the models using the Model Comparison platform to compare the two sets of formulas that you created in step 4 and step 3.

1. Select **Analyze > Predictive Modeling > Model Comparison**.
2. Select Prob(BAD==Good Risk), Prob(BAD==Bad Risk), Prob(BAD==Good Risk) 2, and Prob(BAD==Bad Risk) 2 and click **Y, Predictors**.

   The first pair of formula columns contain the formulas from the model with informative missing. The second pair of formula columns contain the formulas from the model without informative missing.
3. Click **OK**.

Figure 5.23 Measures of Fit from Model Comparison



Measures of Fit for BAD

Creator	.2 .4 .6 .8	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Partition		0.3813	0.5015	0.3092	0.3013	0.1817	0.1158	5960
Partition		0.1223	0.1821	0.4386	0.3688	0.2756	0.1768	5960

The Measures of Fit report shows that the first model, which was fit with informative missing, performs better than the second model, which was not fit with informative missing. The first model has higher RSquare values as well as a lower RMSE value and a lower Misclassification Rate. These findings align the ROC curves comparison.

Note: Again, your results can differ due to the random differences when Informative Missing is not used.

Example of Profit Matrix and Decision Matrix Report

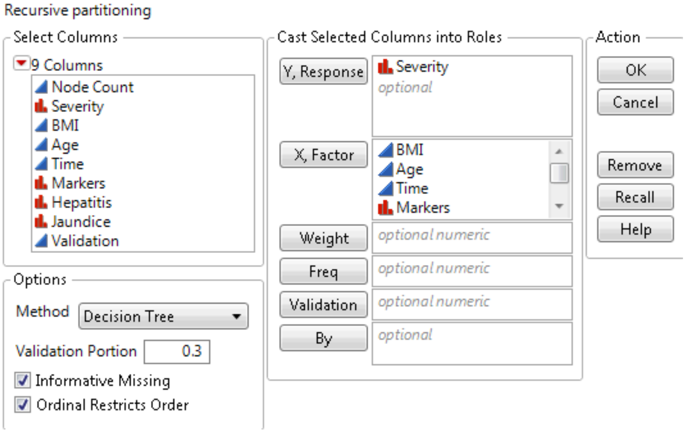
For this example, consider a study of patients who have liver cancer. Based on various measurements and markers, you want to classify patients according to their disease severity (high or low). There are two errors that one can make in classification of patients: classifying a subject who has high severity into the low group, or classifying a patient with low severity into the high group. Clinically, the misclassification of a high patient as low is a costly error, as that patient might not receive the aggressive treatment needed. Classifying a patient with low severity into the high severity group is a less costly error. That patient might receive the more aggressive treatment than needed, but this is not a major concern.

In the following example, you define a profit matrix in the context of a liver cancer study and obtain a Decision Matrix report. The Decision Matrix report helps you assess your classification rates relative to the costs in your profit matrix.

1. Select **Help > Sample Data Library** and open Liver Cancer.jmp.
2. Select **Analyze > Predictive Modeling > Partition**.
3. Select Severity and click **Y, Response**.
4. Select BMI through Jaundice and click **X, Factor**.
5. Select a validation procedure based on your JMP installation:
   - For JMP Pro, select Validation and click **Validation**.
   - For JMP, enter 0.3 as the **Validation Proportion**.

Note: Results using the validation proportion can differ from those shown here, due to the random selection of validation rows.

Figure 5.24 Completed Launch Window with Validation Portion = 0.3



Recursive partitioning

Select Columns

9 Columns: Node Count, Severity, BMI, Age, Time, Markers, Hepatitis, Jaundice, Validation

Cast Selected Columns into Roles

- Y, Response: Severity
- X, Factor: BMI, Age, Time, Markers
- Weight: optional numeric
- Freq: optional numeric
- Validation: optional numeric
- By: optional

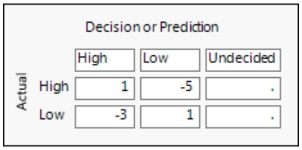
Options

- Method: Decision Tree
- Validation Portion: 0.3
- ☑ Informative Missing
- ☑ Ordinal Restricts Order

6. Click **OK**.
7. Hold down the Shift key and click **Split.**
8. Enter 10 for the number of splits and click **OK**.

   Check that the Number of Splits is 10 in the panel beneath the plot.
9. Click the red triangle next to Partition for Severity and select **Specify Profit Matrix**.
10. Change the entries as follows:
    - Enter 1 in the High, High box.
    - Enter -5 in the High, Low box.
    - Enter -3 in the Low, High box.
    - Enter 1 in the Low, Low box.

Figure 5.25 Completed Profit Matrix



Decision or Prediction

Actual	High	Low	Undecided
High	1	-5	.
Low	-3	1	.

Tip: You can save this profit matrix as a column property for use in later analyses. Select the check box “Save to column as property” at the bottom of the profit matrix window.

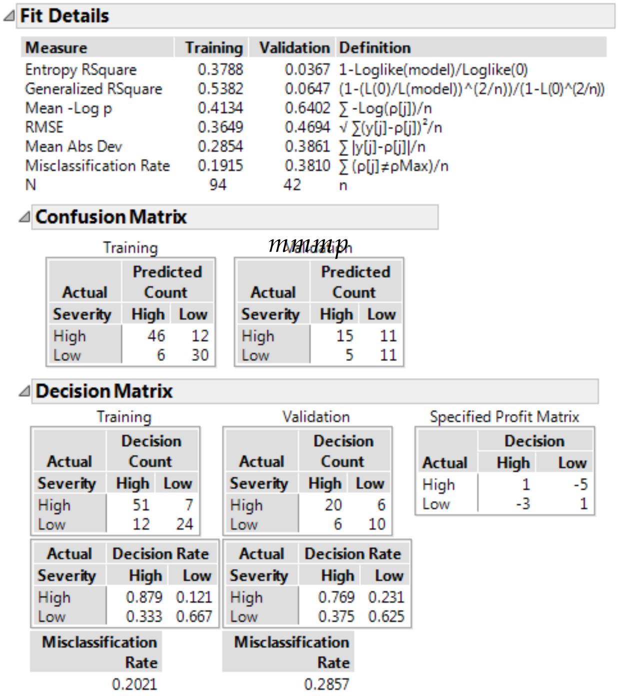
Note the following:

- Each value of 1 reflects your profit when you make a correct decision.
- The -3 value indicates that if you classify a Low severity patient as High severity, your loss is 3 times as much as the profit of a correct decision.

– The -5 value indicates that if you classify a High severity patient as Low severity, your loss is 5 times as much as the profit of a correct decision.

11. Click **OK**.
12. Click the red triangle next to Partition for Severity and select **Show Fit Details**.

Figure 5.26 Confusion Matrix and Decision Matrix Reports



Fit Details

Measure	Training	Validation	Definition
Entropy RSquare	0.3788	0.0367	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5382	0.0647	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4134	0.6402	∑ -Log(ρ[j])/n
RMSE	0.3649	0.4694	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2854	0.3861	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1915	0.3810	∑ (ρ[j]≠ρMax)/n
N	94	42	n

Confusion Matrix

Training

Actual	Predicted Count	
Severity	High	Low
High	46	12
Low	6	30

mmmp
Validation

Actual	Predicted Count	
Severity	High	Low
High	15	11
Low	5	11

Decision Matrix

Training

Actual	Decision Count	
Severity	High	Low
High	51	7
Low	12	24

Actual	Decision Rate	
Severity	High	Low
High	0.879	0.121
Low	0.333	0.667

Misclassification Rate
0.2021

Validation

Actual	Decision Count	
Severity	High	Low
High	20	6
Low	6	10

Actual	Decision Rate	
Severity	High	Low
High	0.769	0.231
Low	0.375	0.625

Misclassification Rate
0.2857

Specified Profit Matrix

	Decision	
Actual	High	Low
High	1	-5
Low	-3	1

The Confusion Matrix and Decision Matrix reports follow the list of Measures in the Fit Details report. Notice that the Confusion Matrix report and the confusion matrices in the Decision Matrix report show different counts. This is because the weighting in the profit matrix results in different decisions than do the predicted probabilities without weighting.

The Confusion Matrix for the validation set shows classifications based on predicted probabilities alone. Based on these, 11 High severity patients would be classified as Low severity and 5 Low severity patients would be classified as High severity.

The Decision Matrix report incorporates the profit matrix weights. Using those weights, only 6 High severity patients are classified as Low severity. However, this comes at the expense of misclassifying 6 Low severity patients into the High severity group (1 additional patient).

13. Click the red triangle next to Partition for Severity and select **Save Columns > Save Prediction Formula**.

Eight columns are added to the data table.

Tip: To quickly return to the data table, click the View Associated Data icon in the bottom right corner of the report window.

- The first three columns involve only the predicted probabilities. The confusion matrix counts are based on the Most Likely Severity column, which classifies a patient into the level with the highest predicted probability. These probabilities are given in the Prob(Severity == High) and Prob(Severity == Low) columns.
- The last five columns involve the profit matrix weighting. The column called Most Profitable Prediction for Severity contains the decision based on the profit matrix. The decision for a patient is the level that results in the largest profit. The profits are given in the Profit for High and Profit for Low columns.

Statistical Details

This section provides quantitative details and additional information.

Responses and Factors

The response can be either continuous or categorical (nominal or ordinal):

- If the response is categorical, then it is fitting the probabilities estimated for the response levels, minimizing the residual log-likelihood chi-square [2*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 the factor is continuous, then the partition is done according to a splitting “cut” value for the factor.
- If the factor is categorical, then it divides the *X* categories into two groups of levels and considers all possible groupings into two levels.

Splitting Criterion

Node splitting is based on the LogWorth statistic, which is reported in Candidate reports for nodes. LogWorth is calculated as follows:

$$-\log_{10}(p\text{-value})$$

where the adjusted *p*-value is calculated in a complex manner that takes into account the number of different ways splits can occur. This calculation is very fair compared to the unadjusted *p*-value, which favors *X*s with many levels, and the Bonferroni *p*-value, which favors *X*s with small numbers of levels. Details about the method are discussed in Sall (2002).

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) appears in the report. This is actually twice the [natural log] entropy or twice the change in the entropy. Entropy is Σ -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 ratio 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-Square.

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 The proportion of observations at the node for each response level.

Prob The predicted probability for that node of the tree. The method for calculating Prob for the ith response level at a given node is as follows:

$$\mathrm{Prob_i} = \frac{\mathrm{n_i} + \mathrm{prior_i}}{\sum(\mathrm{n_i} + \mathrm{prior_i})}$$

where the summation is across all response levels; $\mathrm{n_i}$ is the number of observations at the node for the ith response level; and $\mathrm{prior_i}$ is the prior probability for the ith response level, calculated as:

$\text{prior}_i = \lambda p_i + (1\text{-}\lambda)P_i$

where p_i is the prior_i from the parent node, P_i is the Prob_i from the parent node, and λ is a weighting factor currently set at 0.9.

The method for calculating Prob assures that the predicted probabilities are always nonzero.

Chapter 6

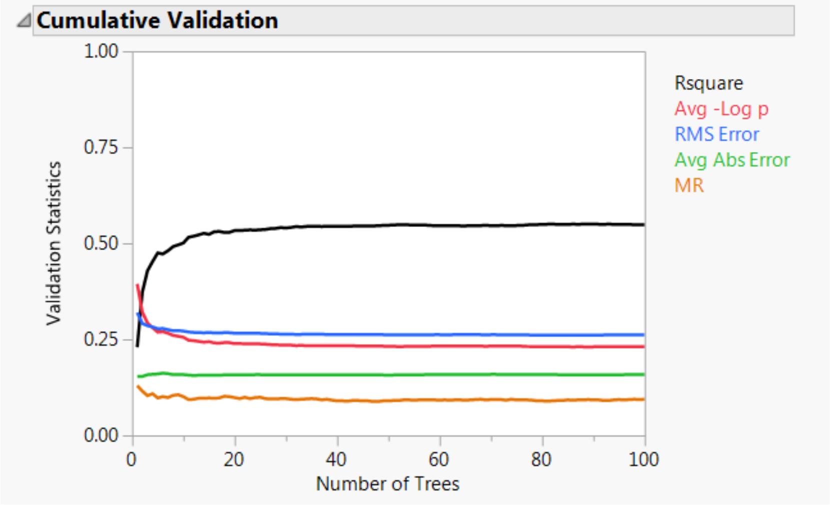
JMP PRO Bootstrap Forest

Fit a Model By Averaging Many Trees

The Bootstrap Forest platform is available only in JMP Pro.

The Bootstrap Forest platform fits an ensemble model by averaging many decision trees each of which is fit to a bootstrap sample of the training data. Each split in each tree considers a random subset of the predictors. In this way, many weak models are combined to produce a more powerful model. The final prediction for an observation is the average of the predicted values for that observation over all the decision trees.

Figure 6.1 Example of a Cumulative Validation Report



Bootstrap Forest Platform Overview

The Bootstrap Forest platform predicts a response value by averaging the predicted response values across many decision trees. Each tree is grown on a *bootstrap sample* of the training data. A bootstrap sample is a random sample of observations, drawn with replacement. In addition, the predictors are sampled at each split in the decision tree. The decision tree is fit using the recursive partitioning methodology described in the "Partition Models" chapter.

The fitting process for the training set proceeds as follows:

1. For each tree, select a bootstrap sample of observations.
2. Fit the individual decision tree, using recursive partitioning, as follows:
   - Select a random set of predictors for each split.
   - Continue splitting until a stopping rule that is specified in the Bootstrap Forest Specification window is met.
3. Repeat step 1 and step 2 until the number of trees specified in the Bootstrap Forest Specification window is reached or until Early Stopping occurs.

For an individual tree, the bootstrap sample of observations that is used to fit the tree is drawn with replacement. You can specify the proportion of observations to be sampled. If you specify that 100% of the observations are to be sampled, because they are drawn with replacement, the expected proportion of unused observations is 1/e, or approximately 36.8%. For each individual tree, these unused observations are called the *out-of-bag* observations. The observations used in fitting the tree are called *in-bag* observations. For a continuous response, the Bootstrap Forest platform provides measures for the error rate for out-of-bag observations, called *out-of-bag error*.

For a continuous response, the predicted value for an observation is the average of its predicted values over the collection of individual trees. For a categorical response, the predicted probability for an observation is the average of its predicted probabilities over the collection of individual trees. The observation is classified into the level for which its predicted probability is the highest.

For more information about bootstrap forests, see Hastie et al. (2009).

JMP PRO Example of Bootstrap Forest with a Categorical Response

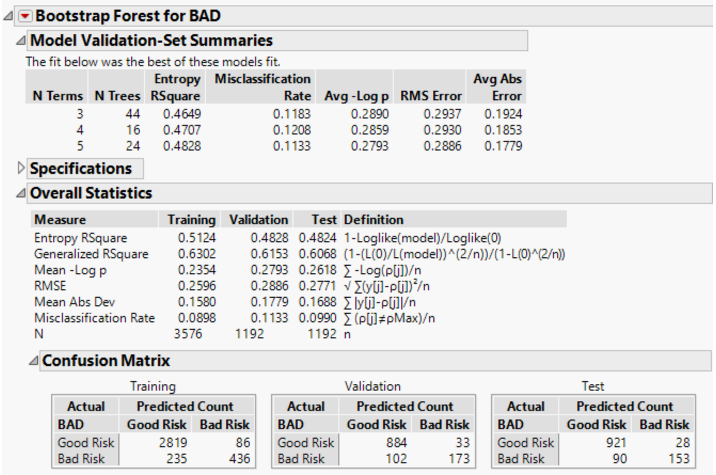
In this example, you construct a bootstrap forest model to predict whether a customer is a bad credit risk. But you are aware that your data set contains missing values, so you also explore the degree to which values are missing.

JMP PRO Bootstrap Forest Model

1. Select **Help > Sample Data Library** and open Equity.jmp.
2. Select **Analyze** > **Predictive Modeling** > **Bootstrap Forest**.
3. Select BAD and click **Y, Response**.
4. Select LOAN through DEBTINC and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.
7. Next to Maximum Splits per Tree, enter 30.
8. Select **Multiple Fits over Number of Terms** and enter 5 next to Max Number of Terms.
9. (Optional) Select **Suppress Multithreading** and enter 123 next to Random Seed.

   Because the bootstrap forest method involves random sampling, these actions ensure that you will obtain the exact results shown below.
10. Click **OK**.

Figure 6.2 Overall Statistics Report



Bootstrap Forest for BAD

Model Validation-Set Summaries

The fit below was the best of these models fit.

N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	RMS Error	Avg Abs Error
3	44	0.4649	0.1183	0.2890	0.2937	0.1924
4	16	0.4707	0.1208	0.2859	0.2930	0.1853
5	24	0.4828	0.1133	0.2793	0.2886	0.1779

Specifications

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.5124	0.4828	0.4824	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6302	0.6153	0.6068	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.2354	0.2793	0.2618	∑ -Log(p[j])/n
RMSE	0.2596	0.2886	0.2771	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0.1580	0.1779	0.1688	∑ |y[j]-p[j]|/n
Misclassification Rate	0.0898	0.1133	0.0990	∑ (p[j]≠pMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2819	86
Bad Risk	235	436

Validation

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	884	33
Bad Risk	102	173

Test

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	921	28
Bad Risk	90	153

Because the Multiple Fits over Number of Terms option was specified, models were created using 3, 4, and 5 as the number of predictors in each split. The Model

Validation-Set Summaries report shows that the model whose Validation set has the highest Entropy RSquare is the five-term model. This is also the model with the smallest misclassification rate. This model is determined to be the best model, and the results in the Overall report are for this model.

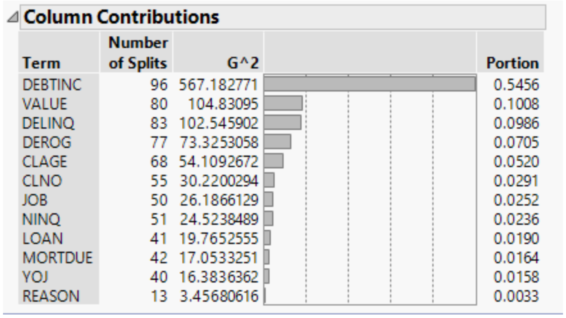
The Overall report shows that the misclassification rates for the Validation and Test sets are about 11.3% and 9.9%, respectively. The confusion matrices suggest that the largest source of misclassification is the classification of bad risk customers as good risks.

The results for the Test set give you an indication of how well your model extends to independent observations. The Validation set was used in selecting the Bootstrap Forest model. For this reason, the results for the Validation set give a biased indication of how the model generalizes to independent data.

You are interested in determining which predictors contributed the most to your model.

11. Click the red triangle next to Bootstrap Forest for BAD and select **Column Contributions**.

Figure 6.3 Column Contributions Report



Column Contributions

Term	Number of Splits	G^2	Portion
DEBTINC	96	567.182771	0.5456
VALUE	80	104.83095	0.1008
DELINQ	83	102.545902	0.0986
DEROG	77	73.3253058	0.0705
CLAGE	68	54.1092672	0.0520
CLNO	55	30.2200294	0.0291
JOB	50	26.1866129	0.0252
NINQ	51	24.5238489	0.0236
LOAN	41	19.7652555	0.0190
MORTDUE	42	17.0533251	0.0164
YOJ	40	16.3836362	0.0158
REASON	13	3.45680616	0.0033

The Column Contributions report suggests that the strongest predictor of a customer's credit risk is DEBTINC, which is the debt to income ratio. The next highest contributors to the model are DELINQ, the number of delinquent credit lines, and VALUE, the assessed value of the customer.

Missing Values

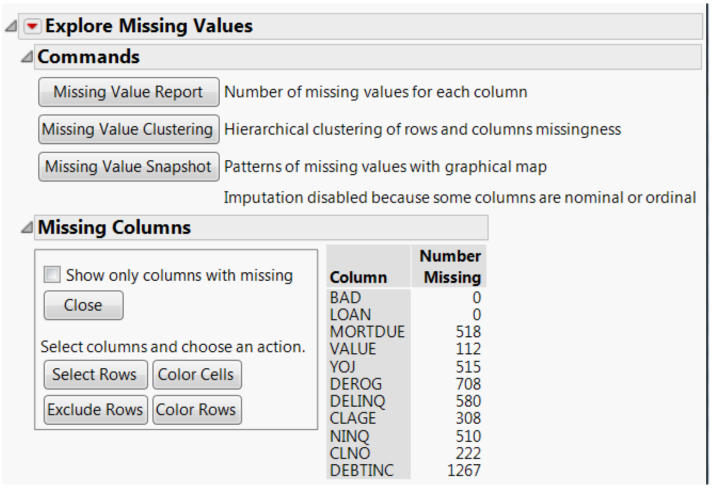
Next, you explore the extent to which predictor values are missing.

1. Select **Analyze > Screening > Explore Missing Values**.
2. Select Bad through DEBTINC and click **Y, Columns**.
3. Click **OK** in the Alert that appears.

   The columns REASON and JOB are not added to the Y, Columns list because they have a Character data type. You can see how many values are missing for these two columns using Distribution (not illustrated in this example).

4. Click **OK**.

Figure 6.4 Missing Values Report



Number of missing values for each column

Hierarchical clustering of rows and columns missingness

Patterns of missing values with graphical map

Imputation disabled because some columns are nominal or ordinal

□ Show only columns with missing

Select columns and choose an action.

Column	Number Missing
BAD	0
LOAN	0
MORTDUE	518
VALUE	112
YOJ	515
DEROG	708
DELINQ	580
CLAGE	308
NINQ	510
CLNO	222
DEBTINC	1267

The DEBTINC column contains 1267 missing values, which amounts to about 21% of the observations. Most other columns involved in the Bootstrap Forest analysis also contain missing values. The Informative Missing option in the launch window ensures that the missing values are treated in a way that acknowledges any information that they carry. For details, see “Informative Missing” on page 88 in the “Partition Models” chapter.

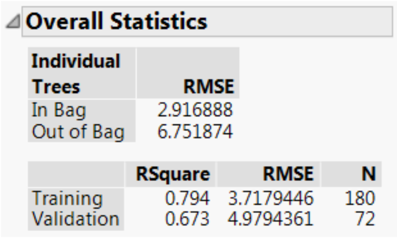
JMP PRO Example of Bootstrap Forest with a Continuous Response

In this example, you construct a bootstrap forest model to predict the percent body fat for male subjects.

1. Select **Help > Sample Data Library** and open Body Fat.jmp.
2. Select **Analyze > Predictive Modeling > Bootstrap Forest**.
3. Select Percent body fat and click **Y, Response**.
4. Select Age (years) through Wrist circumference (cm) and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.
7. (Optional) Select **Suppress Multithreading** and enter 123 next to Random Seed.

   Because the bootstrap forest method involves random sampling, these actions ensure that you will obtain the exact results shown below.
8. Click **OK**.

Figure 6.5 Overall Statistics



Overall Statistics

Individual Trees	RMSE
In Bag	2.916888
Out of Bag	6.751874

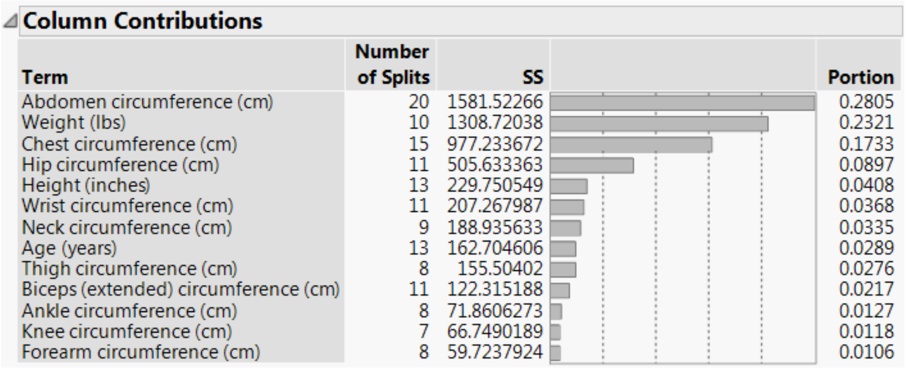
	RSquare	RMSE	N
Training	0.794	3.7179446	180
Validation	0.673	4.9794361	72

The Overall Statistics report shows that the Validation RSquare is 0.673.

You are interested in obtaining a model-independent indication of the most important predictors.

9. Click the red triangle next to Bootstrap Forest for Percent body fat and select **Column Contributions**.

Figure 6.6 Column Contributions



Column Contributions

Term	Number of Splits	SS	Portion
Abdomen circumference (cm)	20	1581.52266	0.2805
Weight (lbs)	10	1308.72038	0.2321
Chest circumference (cm)	15	977.233672	0.1733
Hip circumference (cm)	11	505.633363	0.0897
Height (inches)	13	229.750549	0.0408
Wrist circumference (cm)	11	207.267987	0.0368
Neck circumference (cm)	9	188.935633	0.0335
Age (years)	13	162.704606	0.0289
Thigh circumference (cm)	8	155.50402	0.0276
Biceps (extended) circumference (cm)	11	122.315188	0.0217
Ankle circumference (cm)	8	71.8606273	0.0127
Knee circumference (cm)	7	66.7490189	0.0118
Forearm circumference (cm)	8	59.7237924	0.0106

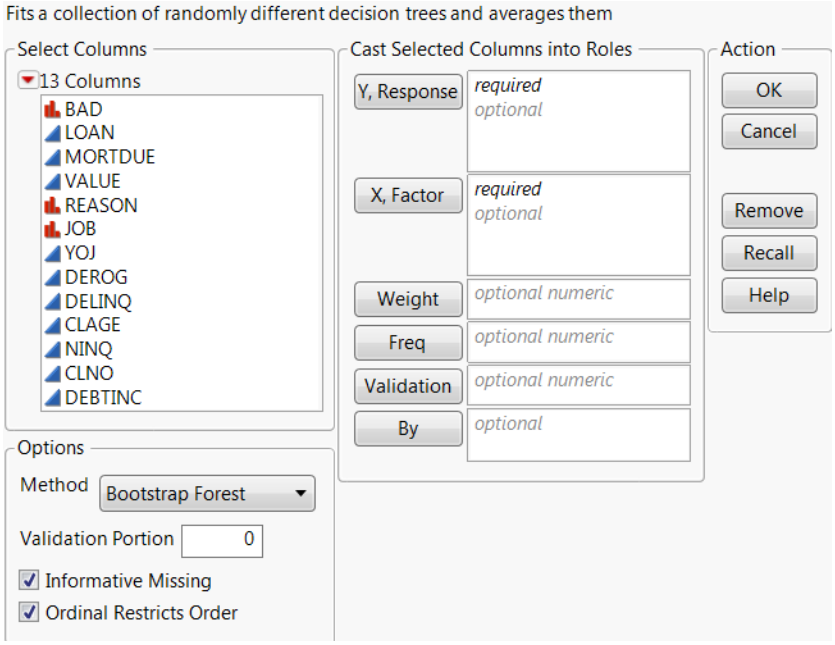
The Column Contributions report suggests that Abdomen circumference (cm), Weight (cm), and Chest circumference (cm) are the strongest predictors for Percent body fat.

Launch the Bootstrap Forest Platform

Launch the Bootstrap Forest platform by selecting **Analyze > Predictive Modeling > Bootstrap Forest**.

Launch Window

Figure 6.7 Bootstrap Forest Launch Window



Fits a collection of randomly different decision trees and averages them

Select Columns

13 Columns

BAD
LOAN
MORTDUE
VALUE
REASON
JOB
YOJ
DEROG
DELINQ
CLAGE
NINQ
CLNO
DEBTINC

Options

Method: Bootstrap Forest

Validation Portion: 0

☑ Informative Missing

☑ Ordinal Restricts Order

The Bootstrap Forest platform launch provides the following options:

Y, Response The response variable or variables that you want to analyze.

X, Factor The predictor variables.

Weight A column whose numeric values assign a weight to each row in the analysis.

Freq A column whose numeric values assign a frequency to each row in the analysis.

Validation A numeric column that contains at most three distinct values. See “Validation” on page 93 in the “Partition Models” chapter.

By A column or columns whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate reports. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

Method Enables you to select the partition method (Decision Tree, Bootstrap Forest, Boosted Tree, K Nearest Neighbors, or Naive Bayes). These alternative methods, except for Decision Tree, are available in JMP Pro.

For more details on these methods, see Chapter 5, “Partition Models”, Chapter 7, “Boosted Tree”, Chapter 8, “K Nearest Neighbors”, and Chapter 9, “Naive Bayes”.

Validation Portion The portion of the data to be used as the validation set. See “Validation” on page 93 in the “Partition Models” chapter.

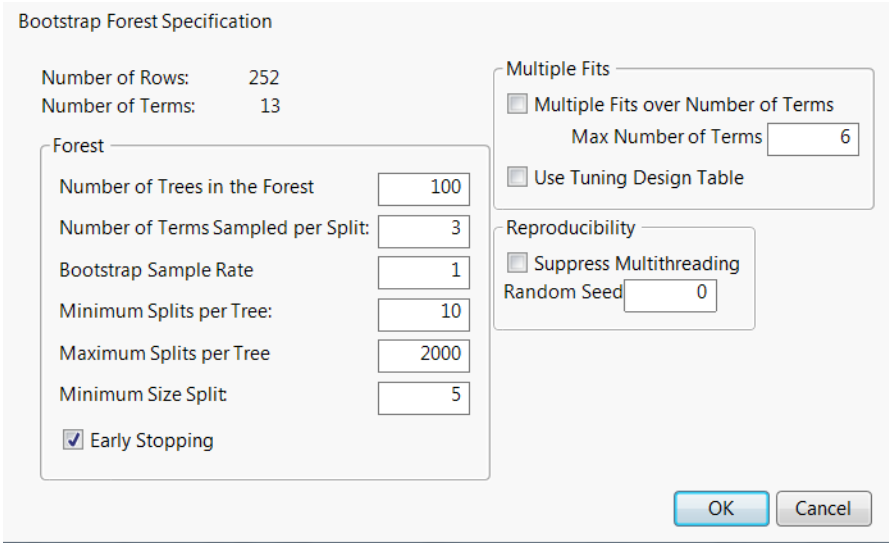
Informative Missing If selected, enables missing value categorization for categorical predictors and informative treatment of missing values for continuous predictors. See “Informative Missing” on page 88 in the “Partition Models” chapter.

Ordinal Restricts Order If selected, restricts consideration of splits to those that preserve the ordering.

Specification Window

After you select **OK** in the launch window, the Bootstrap Forest Specification window appears.

Figure 6.8 Bootstrap Forest Specification Window



Bootstrap Forest Specification

Number of Rows: 252
Number of Terms: 13

Forest

Number of Trees in the Forest 100
Number of Terms Sampled per Split: 3
Bootstrap Sample Rate 1
Minimum Splits per Tree: 10
Maximum Splits per Tree 2000
Minimum Size Split 5
☑ Early Stopping

Multiple Fits

□ Multiple Fits over Number of Terms
Max Number of Terms 6
□ Use Tuning Design Table

Reproducibility

□ Suppress Multithreading
Random Seed 0

Specification Panel

Number of Rows The number of rows in the data table.

Number of Terms The number of columns that are specified as predictors.

Forest Panel

Number of Trees in the Forest Number of trees to grow and then average.

Number of Terms Sampled per Split Number of predictors to consider as splitting candidates at each split. For each split, a new random sample of predictors is taken as the candidate set.

Bootstrap Sample Rate Proportion of observations to sample (with replacement) for growing each tree. A new random sample is generated for each tree.

Minimum Splits Per Tree Minimum number of splits for each tree.

Maximum Splits Per Tree Maximum number of splits for each tree.

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

Early Stopping (Available only if validation is used.) If selected, the process stops growing additional trees if adding more trees does not improve the validation statistic. The validation statistic is the validation set's Entropy RSquare value for a categorical response and its RSquare value for a continuous response. If not selected, the process continues until the specified number of trees is reached.

JMP PRO Multiple Fits Panel

Multiple Fits over Number of Terms If selected, creates a bootstrap forest for several values of number of terms sampled per split. The model for which results are displayed is the model whose Validation Set's Entropy RSquare value (for a categorical response) or RSquare (for a continuous response) is the largest.

The lower bound is the Number of Terms Sampled per Split specification. The upper bound is specified by the following option:

Max Number of Terms The maximum number of terms to consider for a split.

Use Tuning Table Design Opens a window where you can select a data table containing values for the Forest panel tuning parameters, called a *tuning design table*. A tuning design table has a column for each option that you want to specify and has one or multiple rows that each represent a single Bootstrap Forest model design. If an option is not specified in the tuning design table, the default value is used.

For each row in the table, JMP creates a Bootstrap Forest model using the tuning parameters specified. If more than one model is specified in the tuning design table, the Model Validation-Set Summaries report lists the RSquare value for each model. The Bootstrap Forest report shows the fit statistics for the model with the largest RSquare value.

You can create a tuning design table using the Design of Experiments facilities. A bootstrap forest tuning design table can contain the following case-insensitive columns in any order:

- Number Trees
- Number Terms

- Portion Bootstrap
- Minimum Splits per Tree
- Maximum Splits per Tree
- Minimum Size Split

Reproducibility Panel

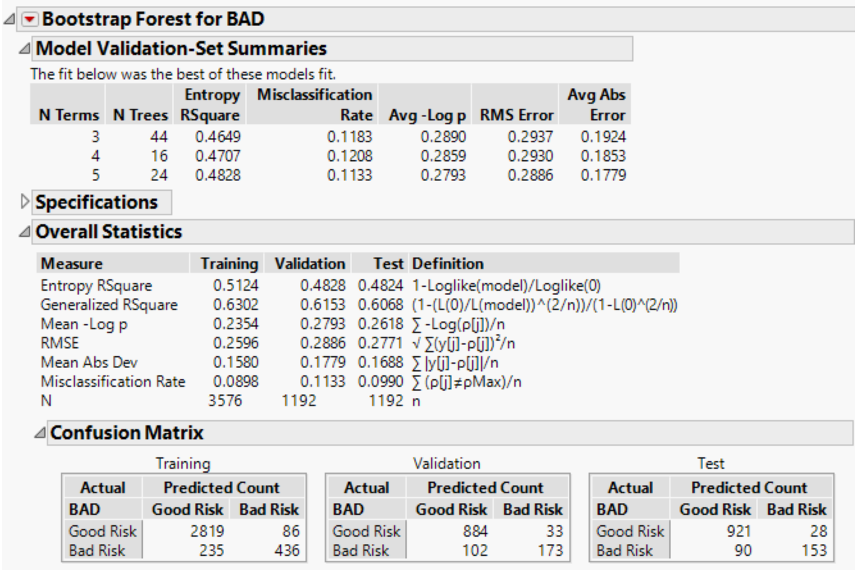
Suppress Multithreading If selected, all calculations are performed on a single thread.

Random Seed Specify a nonzero numeric random seed to reproduce the results for future launches of the platform. By default, the Random Seed is set to zero, which does not produce reproducible results. When you save the analysis to a script, the random seed that you enter is saved to the script.

The Bootstrap Forest Report

After you click **OK** in the Bootstrap Forest Specification window, the Bootstrap Forest report appears.

Figure 6.9 Bootstrap Forest Report for a Categorical Response



Bootstrap Forest for BAD

Model Validation-Set Summaries

The fit below was the best of these models fit.

N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	RMS Error	Avg Abs Error
3	44	0.4649	0.1183	0.2890	0.2937	0.1924
4	16	0.4707	0.1208	0.2859	0.2930	0.1853
5	24	0.4828	0.1133	0.2793	0.2886	0.1779

Specifications

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.5124	0.4828	0.4824	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6302	0.6153	0.6068	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.2354	0.2793	0.2618	∑ -Log(ρ[j])/n
RMSE	0.2596	0.2886	0.2771	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.1580	0.1779	0.1688	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.0898	0.1133	0.0990	∑ (ρ[j]≠ρMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2819	86
Bad Risk	235	436

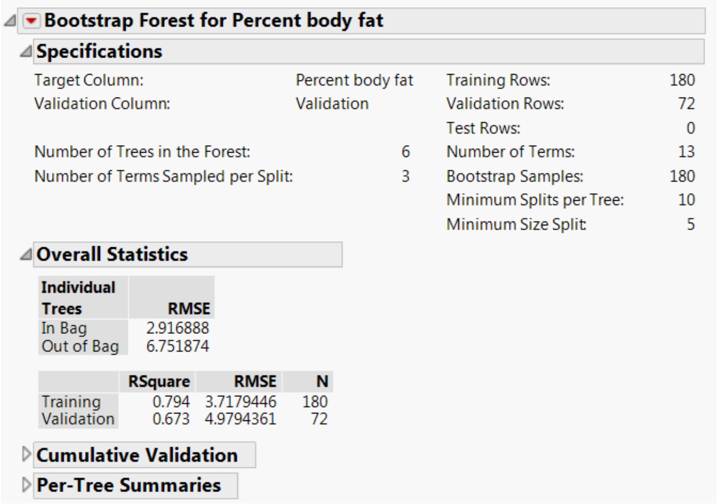
Validation

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	884	33
Bad Risk	102	173

Test

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	921	28
Bad Risk	90	153

Figure 6.10 Bootstrap Forest Report for a Continuous Response



Bootstrap Forest for Percent body fat

Specifications

Target Column:	Percent body fat	Training Rows:	180
Validation Column:	Validation	Validation Rows:	72
		Test Rows:	0
Number of Trees in the Forest:	6	Number of Terms:	13
Number of Terms Sampled per Split:	3	Bootstrap Samples:	180
		Minimum Splits per Tree:	10
		Minimum Size Split:	5

Overall Statistics

Individual Trees	RMSE
In Bag	2.916888
Out of Bag	6.751874

	RSquare	RMSE	N
Training	0.794	3.7179446	180
Validation	0.673	4.9794361	72

The following reports are provided, depending on whether the response is categorical or continuous:

- "Model Validation-Set Summaries" on page 117
- "Specifications" on page 117
- "Overall Statistics" on page 117
- "Cumulative Validation" on page 119
- "Per-Tree Summaries" on page 120

Model Validation-Set Summaries

(Available when you select the Multiple Fits over Number of Terms option in Bootstrap Forest Specification window.) Provides fit statistics for all the models fit. See Figure 6.9 and "Multiple Fits Panel" on page 115.

Specifications

Shows the settings used in fitting the model.

Overall Statistics

Provides fit statistics for the training set, and for the validation and test sets if they are specified. The specific form of the report depends on the modeling type of the response.

Suppose that multiple models are fit using the Multiple Fits over Multiple Terms option in the Bootstrap Forest Specification window. Then the model for which results are displayed in the Overall Statistics and Cumulative Validation reports is the model for which the validation set's Entropy RSquare value (for a categorical response) or RSquare (for a continuous response) is the largest.

Categorical Response

Measures Report

Gives the following statistics for the training set, and for the validation and test sets if there are specified.

Note: 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.

Entropy RSquare One minus the ratio of the negative log-likelihoods from the fitted model and the constant probability model. It ranges from 0 to 1.

Generalized RSquare A measure that can be applied to general regression models. It is based on the likelihood function L and is scaled to have a maximum value of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model. The Generalized RSquare measure simplifies to the traditional RSquare for continuous normal responses in the standard least squares setting. Generalized RSquare is also known as the Nagelkerke or Craig and Uhler R2, which is a normalized version of Cox and Snell's pseudo R2.

Mean -Log P The average of negative 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. 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. 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.

Confusion Matrix

(Available only for categorical responses.) Shows classification statistics for the training set, and for the validation and test sets if they are specified.

Decision Matrix

(Available only for categorical responses and if the response has a Profit Matrix column property or if you specify costs using the Specify Profit Matrix option.) Gives Decision Count and Decision Rate matrices for the training set, and for the validation and test sets if they are specified. See "Additional Examples of Partitioning" on page 95 in the "Partition Models" chapter.

Continuous Response

Individual Trees Report

Gives RMSE values, which are averaged over all trees, for In Bag and Out of Bag observations. Training set observations that are used to construct a tree are called *in-bag* observations. Training observations that are not used to construct a tree are called *out-of-bag* (OOB) observations.

For each tree, the Out of Bag RMSE is computed as the square root of the sum of squared errors divided by the number of OOB observations. The squared Out of Bag RMSE for each tree is given in the Per-Tree Summaries report as OOB SSE/N.

RSquare and RMSE Report

Gives Rsquare, root mean square error, and the number of observations for the training set, and for the validation and test sets, if they are defined.

Cumulative Validation

(Available only if validation is used.) Shows a plot of the fit statistics for the Validation set versus the number of trees.

For a continuous response, the single fit statistic is R-Square. For a categorical response, the fit statistics are listed below and are described in "Measures Report" on page 118.

- RSquare (Entropy RSquare)
- Avg - Log p (Mean - Log p)
- RMS Error (RMSE)
- Avg Abs Error (Mean Abs Dev)
- MR (Misclassification Rate)

The Cumulative Details report below the Cumulative Validation plot gives the values used in the plot.

Per-Tree Summaries

The Per-Tree Summaries report involves the concepts of in-bag and out-of-bag observations. For an individual tree, the bootstrap sample of observations used in fitting the tree is drawn with replacement. Even if you specify that 100% of the observations are to be sampled, because they are drawn with replacement, the expected proportion of unused observations is 1/*e*. For each individual tree, the unused observations are called the *out-of-bag* observations. The observations used in fitting the tree are called *in-bag* observations.

The Per-Tree Summaries report shows the following summary statistics for each tree:

Splits Number of splits in the decision tree.

Rank Rank of the tree's OOB Loss in ascending order. The tree with the smallest OOB loss has Rank 1.

OOB Loss A measure of the total predictive inaccuracy of the tree when applied to the Out Of Bag rows. Lower values indicate a higher predictive accuracy.

OOB Loss/N The OOB Loss divided by the number of OOB rows, OOB N.

RSquare (Available only for continuous responses.) The RSquare value for the tree.

IB SSE (Available only for continuous responses.) Sum of squared errors for the In Bag rows.

IB SSE/N (Available only for continuous responses.) Sum of squared errors for the In Bag rows divided by the number of In Bag observations. The number of In Bag observations is equal to the number of observations in the training set multiplied by the bootstrap sampling rate that you specify in the Bootstrap Forest Specification window.

OOB N (Available only for continuous responses.) Number of Out Of Bag rows.

OOB SSE (Available only for continuous responses.) Sum of squared errors when the tree is applied to the Out Of Bag rows.

OOB SSE/N (Available only for continuous responses.) The OOB SSE divided by the number of OOB rows, OOB N.

Bootstrap Forest Platform Options

The Bootstrap Forest report red triangle menu has the following options:

Plot Actual by Predicted (Available only for continuous responses.) Provides a plot of actual versus predicted values.

Column Contributions Displays a report that shows each input column's contribution to the fit. The report also shows:

– The total number of splits defined by a column.

- The total G^2 (for a categorical response) or SS, sum of squares (for a continuous response), attributed to the column.
- A bar chart of G^2 or SS.
- The proportion of G^2 or SS attributed to the column.

Show Trees Provides various options for displaying trees in the Tree Views report. The report gives a picture of the tree that is fit at each layer of the boosting process. For a description of the Prob column shown by the Show names categories estimates option, see "Predicted Probabilities in Decision Tree and Bootstrap Forest" on page 105 in the "Partition Models" chapter.

ROC Curve (Available only for categorical responses.) See "ROC Curve" on page 90 in the "Partition Models" chapter.

Lift Curve (Available only for categorical responses.) See "Lift Curve" on page 91 in the "Partition Models" chapter.

Save Columns Contains 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. 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 (The Save Prediction Formula option should be used instead of this option. Use this option only when Publish Prediction Formula is not available.) Saves a formula that predicts even when there are missing values and when Informative Missing has not been selected. 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 selected Informative Missing, you can save the Tolerant Prediction Formula by holding the Shift key as you click on the report's red triangle.

Save Residuals (Available only for continuous responses.) Saves the residuals to the data table.

Save Cumulative Details (Available only if validation is used.) Creates a data table containing the fit statistics for each tree.

Publish Prediction Formula Creates a prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169.

Publish Tolerant Prediction Formula (The Publish Prediction Formula option should be used instead of this option. Use this option only when Publish Prediction Formula is not available.) Creates a tolerant prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169. If you have selected Informative Missing, you can use this option by holding the Shift key as you click on the report's red triangle.

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

Specify Profit Matrix (Available only for categorical responses.) Enables you to specify profit or costs associated with correct or incorrect classification decisions. See "Show Fit Details" on page 85 in the "Partition Models" chapter.

Profiler Shows a Prediction Profiler. For more information, see the Profiler chapter in the *Profilers* book.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Chapter 7

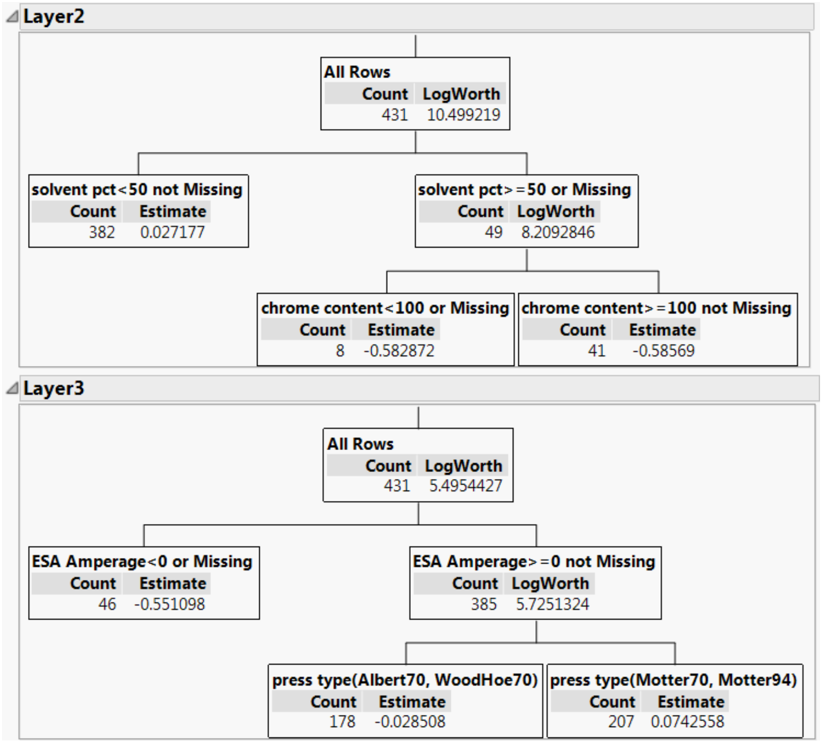
Boosted Tree

Fit Many Layers of Trees, Each Based on the Previous Layer

The Boosted Tree platform is available only in JMP Pro.

Boosting is the process of building a large, additive decision tree by fitting a sequence of smaller decision trees, called *layers*. The tree at each layer consists of a small number of splits. The tree is fit based on the residuals of the previous layers, which allows each layer to correct the fit for bad fitting data from the previous layers. The final prediction for an observation is the sum of the predicted residuals for that observation over all the layers.

Figure 7.1 Example of Boosted Tree Layers



Layer2

All Rows	Count	LogWorth
	431	10.499219
solvent pct<50 not Missing	Count	Estimate
	382	0.027177
solvent pct>=50 or Missing	Count	LogWorth
	49	8.2092846
chrome content<100 or Missing	Count	Estimate
	8	-0.582872
chrome content>=100 not Missing	Count	Estimate
	41	-0.58569

Layer3

All Rows	Count	LogWorth
	431	5.4954427
ESA Amperage<0 or Missing	Count	Estimate
	46	-0.551098
ESA Amperage>=0 not Missing	Count	LogWorth
	385	5.7251324
press type(Albert70, WoodHoe70)	Count	Estimate
	178	-0.028508
press type(Motter70, Motter94)	Count	Estimate
	207	0.0742558

Boosted Tree Platform Overview

The Boosted Tree platform produces an additive decision tree model that is based on many smaller decision trees that are constructed in *layers*. The tree in each layer consists of a small number of splits, typically five or fewer. Each layer is fit using the recursive fitting methodology described in the "Partition Models" chapter. The only difference is that fitting stops at a specified number of splits. For a given tree, the predicted value for an observation in a leaf is the mean of all observations in that leaf.

The fitting process proceeds as follows:

1. Fit an initial layer.
2. Compute residuals. These are obtained by subtracting the predicted mean for observations within a leaf from their actual value.
3. Fit a layer to the residuals.
4. Construct the additive tree. For a given observation, sum its predicted values over the layers.
5. Repeat step 2 to step 4 until the specified number of layers is reached, or, if validation is used, until fitting an additional layer no longer improves the validation statistic.

The final prediction is the sum of the predictions for an observation over all the layers.

By fitting successive layers on residuals from previous layers, each layer can improve the fit.

For categorical responses, only those with two response levels are supported. For a categorical response, the residuals fit at each layer are offsets of linear logits. The final prediction is a logistic transformation of the sum of the linear logits over all the layers.

For more information about boosted trees, see Hastie et al. (2009).

Example of Boosted Tree with a Categorical Response

In this example, you construct a boosted tree model to predict which printing jobs are affected by a defect called *banding*.

1. Select **Help > Sample Data** and open Bands Data.jmp.
2. Select **Analyze > Predictive Modeling > Boosted Tree**.
3. Select Banding? and click **Y, Response**.
4. Select the Predictors column group and click **X, Factor**.
5. Enter 0.2 for **Validation Portion**.
6. Click **OK**.

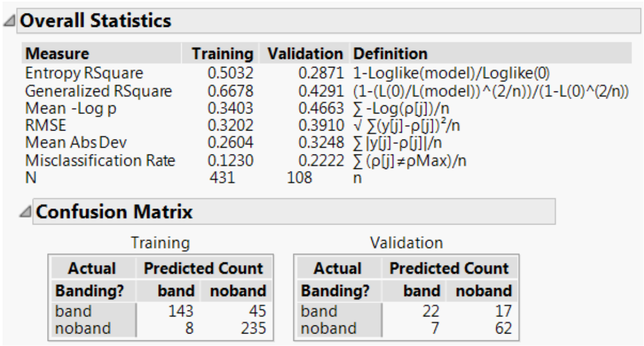
The Boosted Tree Specification window appears.

7. (Optional) In the Reproducibility panel, select **Suppress Multithreading** and enter 123 for Random Seed.

   Because the boosted tree fit involves a random component, these actions ensure that you obtain the exact results shown below.

8. Click **OK**.

Figure 7.2 Overall Statistics for Nominal Response



Overall Statistics

Measure	Training	Validation	Definition
Entropy RSquare	0.5032	0.2871	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6678	0.4291	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3403	0.4663	∑ -Log(ρ[j])/n
RMSE	0.3202	0.3910	√ ∑(y[j]-ρ[j])^2/n
Mean Abs Dev	0.2604	0.3248	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1230	0.2222	∑ (ρ[j]≠ρMax)/n
N	431	108	n

Confusion Matrix

Training

Actual	Predicted Count	
Banding?	band	noband
band	143	45
noband	8	235

Validation

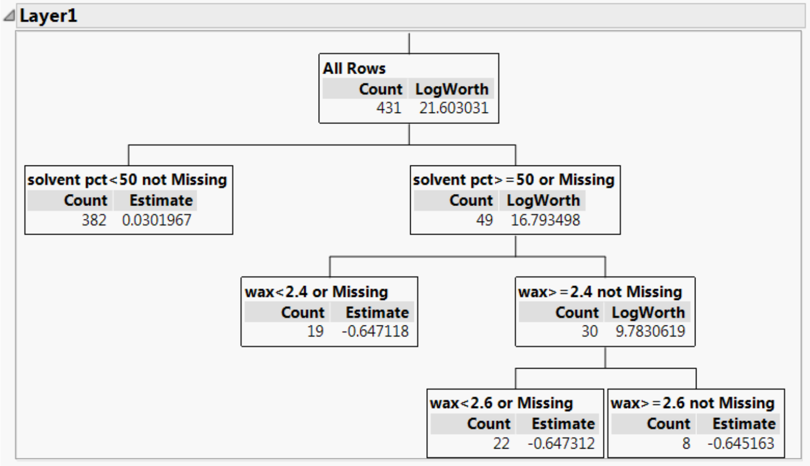
Actual	Predicted Count	
Banding?	band	noband
band	22	17
noband	7	62

Because the response, Banding?, is categorical, the Boosted Tree analysis provides a Misclassification Rate under Measure and a Confusion Matrix report. The Misclassification Rate for the validation set is 0.2222, or about 22%.

9. Click the red triangle next to Boosted Tree for Banding? and select **Show Trees > Show names categories estimates**.

   A Tree Views report appears, with outlines for the layers. You can examine the layers to see the trees that are fit and the predicted values.

Figure 7.3 Layer 1 of the Boosted Tree



Layer1

All Rows

Count	LogWorth
431	21.603031

solvent pct<50 not Missing

Count	Estimate
382	0.0301967

solvent pct>=50 or Missing

Count	LogWorth
49	16.793498

wax<2.4 or Missing

Count	Estimate
19	-0.647118

wax>=2.4 not Missing

Count	LogWorth
30	9.7830619

wax<2.6 or Missing

Count	Estimate
22	-0.647312

wax>=2.6 not Missing

Count	Estimate
8	-0.645163

10. Click the red triangle next to Boosted Tree for Banding? and select **Save Columns > Save Prediction Formula**.

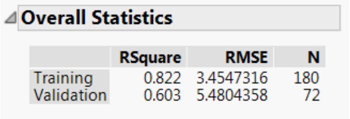
    Columns called Prob(Banding?==noband), Prob(Banding?==band), and Most Likely Banding? are added to the data table. Examine the Prob(Banding?==noband) column to see how model predictions are calculated from the layers.

Example of Boosted Tree with a Continuous Response

In this example, you construct a boosted tree model to predict the percent body fat given a combination of nominal and continuous factors.

1. Select **Help > Sample Data** and open the Body Fat.jmp sample data table.
2. Select **Analyze > Predictive Modeling > Boosted Tree**.
3. Select Percent body fat and click **Y, Response**.
4. Select Age (years) through Wrist circumference (cm) and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.
7. Click **OK**.

Figure 7.4 Overall Statistics for Continuous Response



Overall Statistics

	RSquare	RMSE	N
Training	0.822	3.4547316	180
Validation	0.603	5.4804358	72

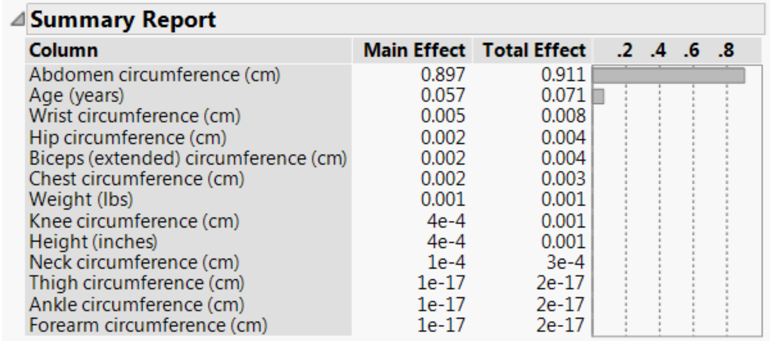
The Overall Statistics report provides the R-square and RMSE for the boosted tree model. The R-square for the validation set is 0.603. The RMSE for the validation set is about 5.48.

You are interested in obtaining a model-independent indication of the important predictors for Percent body fat.

8. Click the red triangle next to Boosted Tree for Percent body fat and select **Profiler**.
9. Click the red triangle next to Prediction Profiler and select **Assess Variable Importance > Independent Uniform Inputs**.

Note: Because Assess Variable Importance uses randomization, your results will not exactly match those in Figure 7.5.

Figure 7.5 Summary Report for Variable Importance



Summary Report

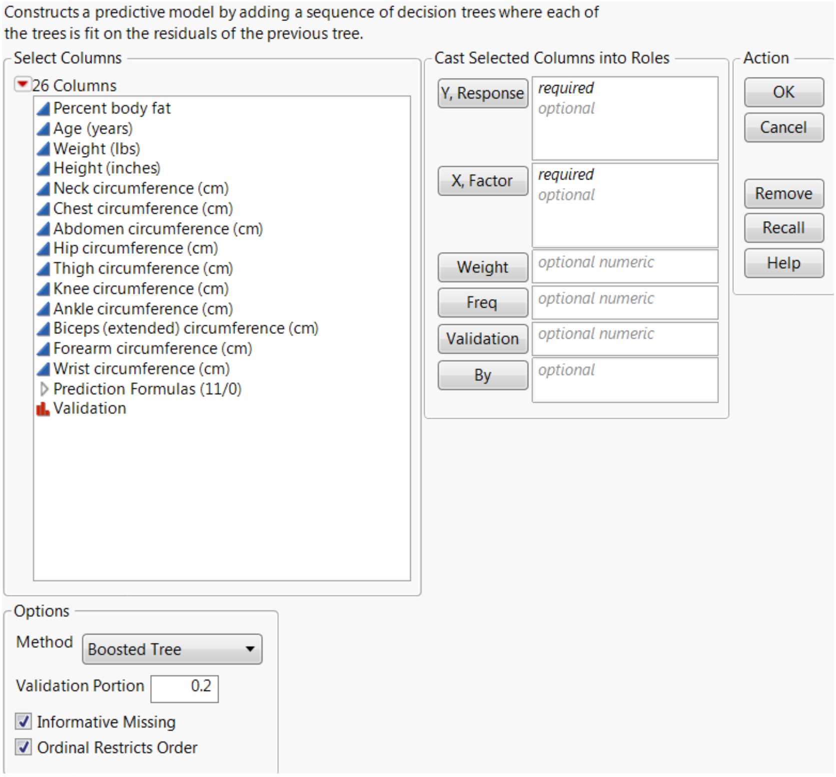
Column	Main Effect	Total Effect	.2 .4 .6 .8
Abdomen circumference (cm)	0.897	0.911	
Age (years)	0.057	0.071	
Wrist circumference (cm)	0.005	0.008	
Hip circumference (cm)	0.002	0.004	
Biceps (extended) circumference (cm)	0.002	0.004	
Chest circumference (cm)	0.002	0.003	
Weight (lbs)	0.001	0.001	
Knee circumference (cm)	4e-4	0.001	
Height (inches)	4e-4	0.001	
Neck circumference (cm)	1e-4	3e-4	
Thigh circumference (cm)	1e-17	2e-17	
Ankle circumference (cm)	1e-17	2e-17	
Forearm circumference (cm)	1e-17	2e-17	

The Summary Report shows that Abdomen circumference (cm) is the most important predictor of Percent body fat.

Launch the Boosted Tree Platform

Launch the Boosted Tree platform by selecting **Analyze > Predictive Modeling > Boosted Tree**.

Boosted Tree Launch Window Using Body Fat.jmp



Constructs a predictive model by adding a sequence of decision trees where each of the trees is fit on the residuals of the previous tree.

Method: Boosted Tree

Validation Portion: 0.2

☑ Informative Missing

☑ Ordinal Restricts Order

The Boosted Tree platform launch window has the following options:

Y, Response The response variable or variables that you want to analyze.

X, Factor The predictor variables.

Weight A column whose numeric values assign a weight to each row in the analysis.

Freq A column whose numeric values assign a frequency to each row in the analysis.

Validation A numeric column that contains at most three distinct values. See "Validation" on page 93 in the "Partition Models" chapter.

By A column or columns whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate reports. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

Method Enables you to select the partition method (Decision Tree, Bootstrap Forest, Boosted Tree, K Nearest Neighbors, or Naive Bayes). These alternative methods, except for Decision Tree, are available in JMP Pro.

For more details about these methods, see Chapter 5, "Partition Models", Chapter 6, "Bootstrap Forest", Chapter 8, "K Nearest Neighbors", and Chapter 9, "Naive Bayes".

Validation Portion The portion of the data to be used as the validation set. See "Validation" on page 93 in the "Partition Models" chapter.

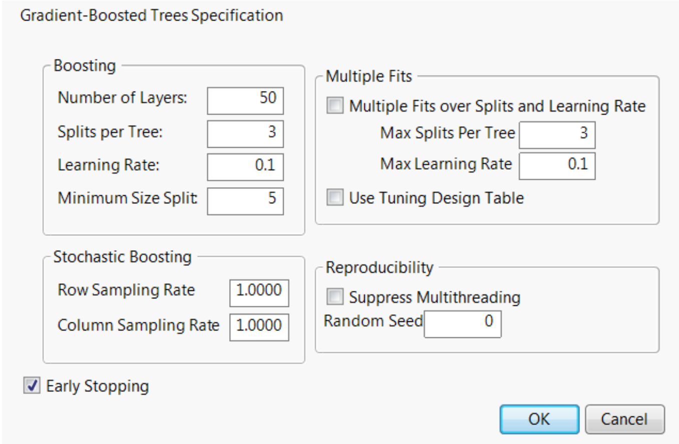
Informative Missing If selected, enables missing value categorization for categorical predictors and informative treatment of missing values for continuous predictors. See "Informative Missing" on page 88 in the "Partition Models" chapter.

Ordinal Restricts Order If selected, restricts consideration of splits to those that preserve the ordering.

Specification Window

After you select **OK** in the launch window, the Gradient-Boosted Trees Specification window appears.

Figure 7.6 Boosted Tree Specification Window



Gradient-Boosted Trees Specification

Boosting
Number of Layers: 50
Splits per Tree: 3
Learning Rate: 0.1
Minimum Size Split 5

Multiple Fits
☐ Multiple Fits over Splits and Learning Rate
Max Splits Per Tree 3
Max Learning Rate 0.1
☐ Use Tuning Design Table

Stochastic Boosting
Row Sampling Rate 1.0000
Column Sampling Rate 1.0000

Reproducibility
☐ Suppress Multithreading
Random Seed 0

☑ Early Stopping

Boosting Panel

Number of Layers Maximum number of layers to include in the final tree.

Splits per Tree Number of splits for each layer.

Learning Rate 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. The learning rate is a small fraction typically between 0.01 and 0.1 that slows the convergence of the model. This preserves opportunities for later layers to use different splits than the earlier layers.

Overfit Penalty (Available only for categorical responses.) A biasing parameter that helps protect against fitting probabilities equal to zero.See "Overfit Penalty" on page 138.

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

Multiple Fits Panel

Multiple Fits over Splits and Learning Rate If selected, creates a boosted tree for every combination of Splits per Tree (in integer increments) and Learning Rate (in 0.1 increments).

The lower bounds for the combinations are specified by the Splits per Tree and Learning Rate options. The upper bounds for the combinations are specified by the following options:

Max Splits per Tree Upper bound for Splits per Tree.

Max Learning Rate Lower bound for Learning Rate.

Use Tuning Design Table Opens a window where you can select a data table containing values for some tuning parameters, called a *tuning design table*. A tuning design table has a column for each option that you want to specify and has one or multiple rows that each represent a single Boosted Tree model design. If an option is not specified in the tuning design table, the default value is used.

For each row in the table, JMP creates a Boosted Tree model using the tuning parameters specified. If more than one model is specified in the tuning design table, the Model Validation-Set Summaries report lists the R-Square value for each model. The Boosted Tree report shows the fit statistics for the model with the largest R-Square value.

You can create a tuning design table using the Design of Experiments facilities. A boosted tree tuning design table can contain the following case-insensitive columns in any order:

– Number of Layers

– Splits per Tree

– Learning Rate

– Minimum Size Split

– Row Sampling Rate

– Column Sampling Rate

Stochastic Boosting Panel

Row Sampling Rate Proportion of training rows to sample for each layer.

Note: When the response is categorical, the training rows are sampled using stratified random sampling.

Column Sampling Rate Proportion of predictor columns to sample for each layer.

Reproducibility Panel

Suppress Multithreading If selected, all calculations are performed on a single thread.

Random Seed Specify a nonzero numeric random seed to reproduce the results for future launches of the platform. By default, the Random Seed is set to zero, which does not produce reproducible results. When you save the analysis to a script, the random seed that you enter is saved to the script.

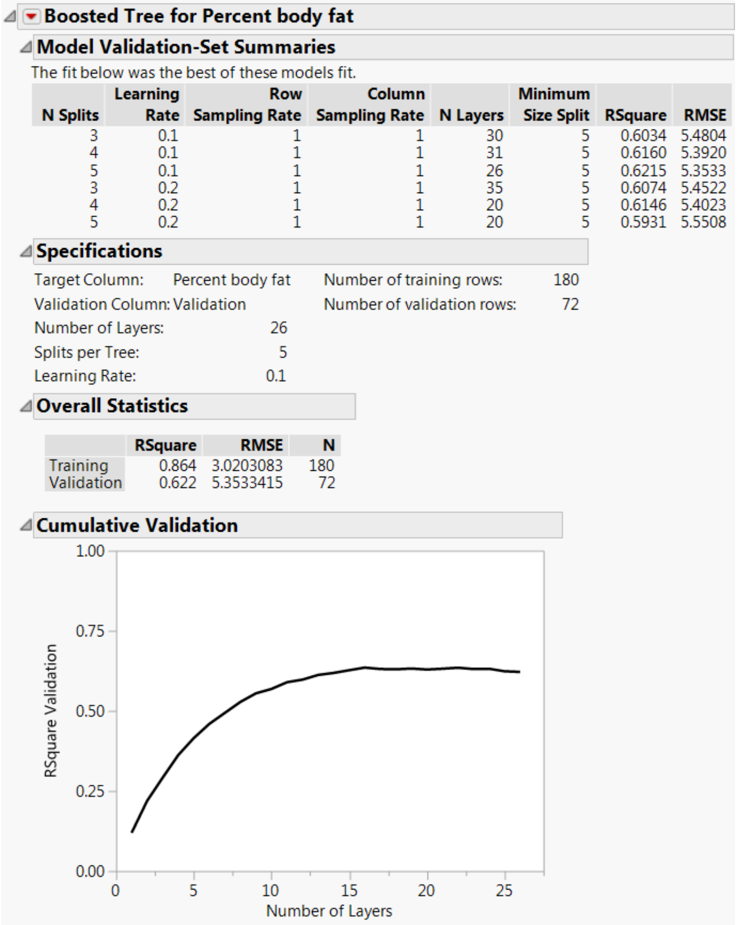
Early Stopping

Early Stopping If selected, the boosting process stops fitting additional layers when adding more layers does not improve the validation statistic. If not selected, the boosting process continues until the specified number of layers is reached. This option appears only if validation is used.

The Boosted Tree Report

After you click OK in the Gradient-Boosted Trees Specification window, the Boosted Tree report opens.

Figure 7.7 Boosted Tree Report for a Continuous Response



Boosted Tree for Percent body fat

Model Validation-Set Summaries

The fit below was the best of these models fit.

N Splits	Learning Rate	Row Sampling Rate	Column Sampling Rate	N Layers	Minimum Size Split	RSquare	RMSE
3	0.1	1	1	30	5	0.6034	5.4804
4	0.1	1	1	31	5	0.6160	5.3920
5	0.1	1	1	26	5	0.6215	5.3533
3	0.2	1	1	35	5	0.6074	5.4522
4	0.2	1	1	20	5	0.6146	5.4023
5	0.2	1	1	20	5	0.5931	5.5508

Specifications

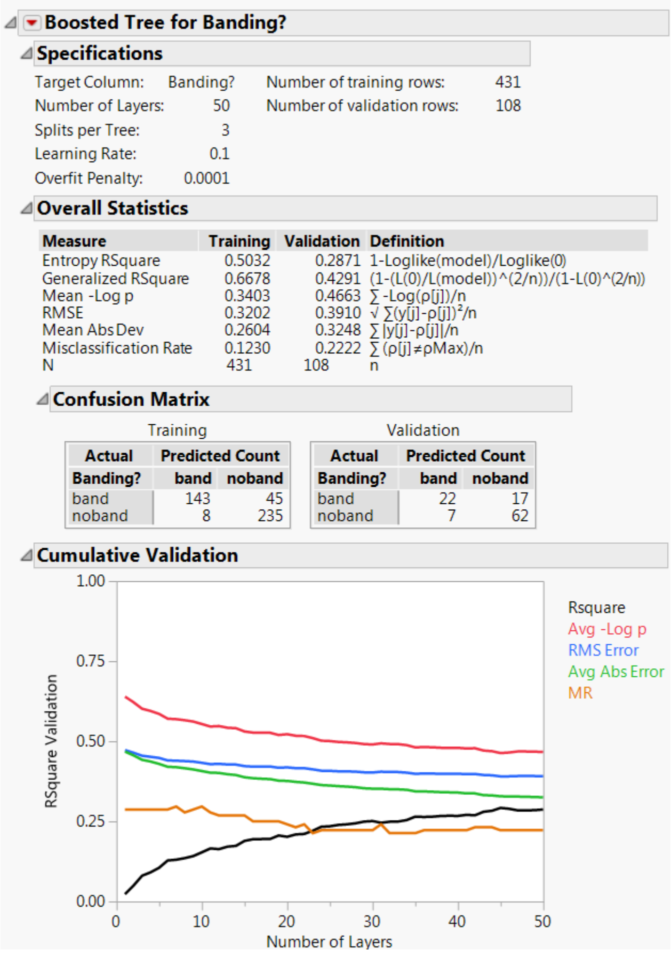
Target Column:	Percent body fat	Number of training rows:	180
Validation Column:	Validation	Number of validation rows:	72
Number of Layers:	26		
Splits per Tree:	5		
Learning Rate:	0.1		

Overall Statistics

	RSquare	RMSE	N
Training	0.864	3.0203083	180
Validation	0.622	5.3533415	72

Cumulative Validation

Figure 7.8 Boosted Tree Report for a Categorical Response



Boosted Tree for Banding?

Specifications

Target Column:	Banding?	Number of training rows:	431
Number of Layers:	50	Number of validation rows:	108
Splits per Tree:	3		
Learning Rate:	0.1		
Overfit Penalty:	0.0001		

Overall Statistics

Measure	Training	Validation	Definition
Entropy RSquare	0.5032	0.2871	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6678	0.4291	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3403	0.4663	∑ -Log(ρ[j])/n
RMSE	0.3202	0.3910	√ ∑(y[j]-ρ[j])^2/n
Mean Abs Dev	0.2604	0.3248	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1230	0.2222	∑ (ρ[j]≠ρMax)/n
N	431	108	n

Confusion Matrix

Training

Actual	Predicted Count	
Banding?	band	noband
band	143	45
noband	8	235

Validation

Actual	Predicted Count	
Banding?	band	noband
band	22	17
noband	7	62

Cumulative Validation

The following reports are provided, depending on whether the response is categorical or continuous:

- “Model Validation - Set Summaries” on page 134
- “Specifications” on page 134
- “Overall Statistics” on page 134
- “Cumulative Validation” on page 135

Model Validation - Set Summaries

Shows fit statistics for all the models fit if you selected the Multiple Fits over Splits and Learning Rate option in the Specification window. See Figure 7.7 and "Multiple Fits Panel" on page 130.

Specifications

Shows the settings used in fitting the model.

Overall Statistics

Shows fit statistics for the training set, and for the validation and test sets if they are specified.

Suppose that you fit multiple models using the Multiple Fits over Multiple Terms option in the Bootstrap Forest Specification window. Then the model for which results are displayed in the Overall Statistics and Cumulative Validation reports is the model for which the validation set's Entropy R-square value (for a categorical response) or R-square (for a continuous response) is the largest.

Measures Report

(Available only for categorical responses.) Gives the following statistics for the training set, and for the validation and test sets if there are specified.

Note: For Entropy R-Square and Generalized R-Square, 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.

Entropy RSquare One minus the ratio of the negative log-likelihoods from the fitted model and the constant probability model. Entropy R-Square ranges from 0 to 1.

Generalized RSquare A measure that can be applied to general regression models. It is based on the likelihood function *L* and is scaled to have a maximum value of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model. The Generalized R-Square measure simplifies to the traditional R-Square for continuous normal responses in the standard least squares setting. Generalized R-Square is also known as the Nagelkerke or Craig and Uhler R^2, which is a normalized version of Cox and Snell's pseudo R^2.

Mean -Log P The average of negative 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. 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. 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.

Confusion Matrix

(Available only for categorical responses.) Shows classification statistics for the training set, and for the validation and test sets if they are specified.

Decision Matrix

(Available only for categorical responses and if the response has a Profit Matrix column property or if you specify costs using the Specify Profit Matrix option.) Gives Decision Count and Decision Rate matrices for the training set, and for the validation and test sets if they are specified. See "Additional Examples of Partitioning" on page 95 in the "Partition Models" chapter.

Cumulative Validation

(Available only if validation is used.) Shows a plot of the fit statistics for the Validation set versus the number of layers.

For a continuous response, the single fit statistic is R-Square. For a categorical response, the fit statistics are listed below and are described in "Measures Report" on page 134.

- R-Square (Entropy R-Square)
- Avg - Log p (Mean - Log p)
- RMS Error (RMSE)
- Avg Abs Error (Mean Abs Dev)
- MR (Misclassification Rate)

The Cumulative Details report below the Cumulative Validation plot gives the values used in the plot.

Boosted Tree Platform Options

The Boosted Tree report red-triangle menu has the following options:

Show Trees Provides options for displaying trees in the Tree Views report. The report gives a picture of the tree that is fit at each layer of the boosting process.

Plot Actual by Predicted (Available only for continuous responses.) Provides a plot of actual versus predicted values.

Column Contributions Displays a report showing each input column's contribution to the fit. The report also shows:

– The total number of splits defined by a column.

– The total G^2 (for a categorical response) or SS, sum of squares (for a continuous response) attributed to the column.

– A bar chart of G^2 or SS.

– The proportion of G^2 or SS attributed to the column.

ROC Curve (Available only for categorical responses.) See "ROC Curve" on page 90 in the "Partition Models" chapter.

Lift Curve (Available only for categorical responses.) See "Lift Curve" on page 91 in the "Partition Models" chapter.

Save Columns Contains 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. 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 (The Save Prediction Formula option should be used instead of this option. Use this option only when Save Prediction Formula is not available.) Saves a formula that predicts even when there are missing values and when Informative Missing has not been selected. 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 selected Informative Missing, you can save the Tolerant Prediction Formula by holding the Shift key as you click on the report's red triangle.

Save Residuals (Available only for continuous responses.) Saves the residuals to the data table.

Save Offset Estimates (Available only for categorical responses.) Saves the sums of the linear components. These are the logits of the fitted probabilities.

Save Tree Details Creates a data table containing split details and estimates for each layer.

Save Cumulative Details (Available only if validation is used.) Creates a data table containing the fit statistics for each layer.

Publish Prediction Formula Creates a prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169.

Publish Tolerant Prediction Formula (The Publish Prediction Formula option should be used instead of this option. Use this option only when Publish Prediction Formula is not available.) Creates a tolerant prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the "Formula Depot" chapter on page 169. If you have selected Informative Missing, you can use this option by holding the Shift key as you click on the report's red triangle.

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

Specify Profit Matrix (Available only for categorical responses.) Enables you to specify profit or costs associated with correct or incorrect classification decisions. See "Show Fit Details" on page 85 in the "Partition Models" chapter.

Profiler Shows a Prediction Profiler. For more information, see the Profiler chapter in the *Profilers* book.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Statistical Details for the Boosted Tree Platform

This section describes details specific to the Boosted Tree Platform. For details about recursive decision trees, see "Statistical Details" on page 104 in the "Partition Models" chapter.

Overfit Penalty

When the response is categorical, a parametric penalty is imposed. For each layer, the estimates minimize the negative log-likelihood plus the penalty value multiplied by the sum of squares of the estimates for each observation. This penalty encourages each new layer not to overfit the training data.

Chapter 8

JMP PRO K Nearest Neighbors

Predict Response Values Using Nearby Observations

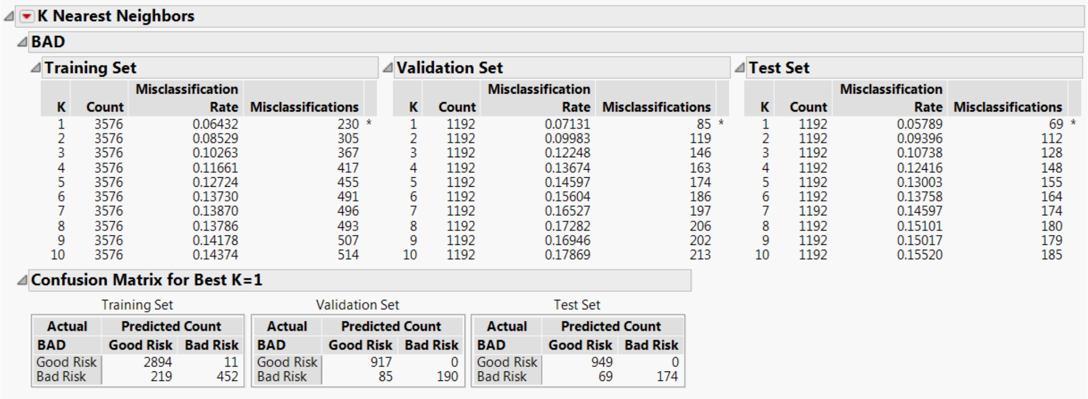
The K Nearest Neighbors platform is available only in JMP Pro.

The K Nearest Neighbors platform predicts a response value for a given observation using the responses of the observations in that observation's local neighborhood. It can be used with a categorical response for classification and with a continuous response for prediction.

K Nearest Neighbors is a nonparametric method that is based on the distance to neighboring observations. Because of this fact, K Nearest Neighbors is able to classify observations using irregular predictor value boundaries. However, the algorithm is sensitive to irrelevant predictors, so selection of predictors can be beneficial before implementing K Nearest Neighbors.

K Nearest Neighbors has been used successfully in many applications, such as classifying satellite imagery and EKG patterns.

Figure 8.1 Example of the K Nearest Neighbors Platform



K Nearest Neighbors

BAD

Training Set

K	Count	Misclassification Rate	Misclassifications	
1	3576	0.06432	230	*
2	3576	0.08529	305	
3	3576	0.10263	367	
4	3576	0.11661	417	
5	3576	0.12724	455	
6	3576	0.13730	491	
7	3576	0.13870	496	
8	3576	0.13786	493	
9	3576	0.14178	507	
10	3576	0.14374	514	

Validation Set

K	Count	Misclassification Rate	Misclassifications	
1	1192	0.07131	85	*
2	1192	0.09983	119	
3	1192	0.12248	146	
4	1192	0.13674	163	
5	1192	0.14597	174	
6	1192	0.15604	186	
7	1192	0.16527	197	
8	1192	0.17282	206	
9	1192	0.16946	202	
10	1192	0.17869	213	

Test Set

K	Count	Misclassification Rate	Misclassifications	
1	1192	0.05789	69	*
2	1192	0.09396	112	
3	1192	0.10738	128	
4	1192	0.12416	148	
5	1192	0.13003	155	
6	1192	0.13758	164	
7	1192	0.14597	174	
8	1192	0.15101	180	
9	1192	0.15017	179	
10	1192	0.15520	185	

Confusion Matrix for Best K=1

Training Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2894	11
Bad Risk	219	452

Validation Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	917	0
Bad Risk	85	190

Test Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	949	0
Bad Risk	69	174

K Nearest Neighbors Platform Overview

The K Nearest Neighbors platform predicts a response value based on the responses of the *k* nearest rows. The *k* nearest rows to a given row are determined by identifying the *k* smallest Euclidean distances between the predictor values for that row and the predictor values for each of the other rows. For a continuous response, the predicted value is the average of the responses for the *k* nearest rows. For a categorical response, the predicted value is the most frequent response level for the *k* nearest neighbors. If two or more levels are tied as the most frequent levels, the predicted response is assigned by selecting one of these levels at random.

Note: Because ties for most frequent levels in the case of a categorical response are broken at random, results from independent runs of the platform might differ. In a script, add the JSL keyword `Nonrandom` to the function for a K Nearest Neighbor model to obtain reproducible results.

Each continuous predictor is scaled by its standard deviation. With this scaling, a single predictor with a large range does not excessively influence the distance calculation. Missing values for a continuous predictor are replaced by the mean of that predictor.

Each categorical predictor is expressed in terms of indicator variables, with one indicator variable representing each level. A row with a missing value for a categorical predictor is represented by values of zero on all indicator variables for that predictor.

Note the following potential drawbacks of the *k* nearest neighbors method:

- K Nearest Neighbors does not make a prediction formula that is practical for large problems.
- K Nearest Neighbors does not produce fitted probabilities for categorical responses.

For more information about the *k* nearest neighbors method, see Hastie et al. (2009), Hand et al. (2001), and Shmueli et al. (2017).

Example of K Nearest Neighbors with Categorical Response

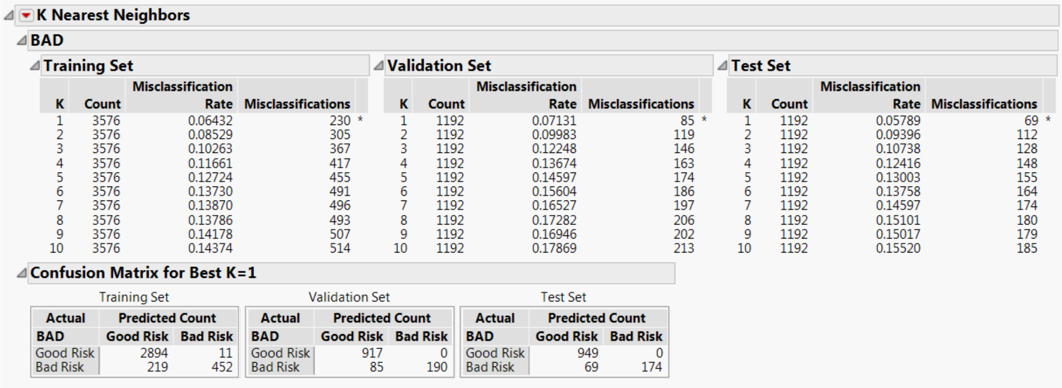
You have historical financial data for 5,960 customers who applied for home equity loans. Each customer was classified as being a Good Risk or Bad Risk. There is missing data on most of the predictors. You want to construct a model to use in classifying the credit risk of future customers.

1. Select **Help > Sample Data** and open Equity.jmp.
2. Select **Analyze > Predictive Modeling > K Nearest Neighbors**.
3. Select BAD and click **Y, Response**.

Because one of the potential predictors, DEBTINC, has many missing values that might be informative, you do not include it in your model.

4. Select LOAN through CLNO and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.

Figure 8.2 K Nearest Neighbors Report



K Nearest Neighbors

BAD

Training Set

K	Count	Misclassification Rate	Misclassifications
1	3576	0.06432	230 *
2	3576	0.08529	305
3	3576	0.10263	367
4	3576	0.11661	417
5	3576	0.12724	455
6	3576	0.13730	491
7	3576	0.13870	496
8	3576	0.13786	493
9	3576	0.14178	507
10	3576	0.14374	514

Validation Set

K	Count	Misclassification Rate	Misclassifications
1	1192	0.07131	85 *
2	1192	0.09983	119
3	1192	0.12248	146
4	1192	0.13674	163
5	1192	0.14597	174
6	1192	0.15604	186
7	1192	0.16527	197
8	1192	0.17282	206
9	1192	0.16946	202
10	1192	0.17869	213

Test Set

K	Count	Misclassification Rate	Misclassifications
1	1192	0.05789	69 *
2	1192	0.09396	112
3	1192	0.10738	128
4	1192	0.12416	148
5	1192	0.13003	155
6	1192	0.13758	164
7	1192	0.14597	174
8	1192	0.15101	180
9	1192	0.15017	179
10	1192	0.15520	185

Confusion Matrix for Best K=1

Training Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2894	11
Bad Risk	219	452

Validation Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	917	0
Bad Risk	85	190

Test Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	949	0
Bad Risk	69	174

For each value of K, K Nearest Neighbors constructs a model using only the Training Set observations. Each of these models is used to classify the Validation Set observations. In this example, based on the Validation Set results, a model based on the single nearest neighbor (K = 1) has the smallest misclassification rate. The Test Set verifies that the single nearest neighbor model is the best performer for independent data.

7. Click the K Nearest Neighbors red triangle and select **Publish Prediction Formula**.
8. Next to **Number of Neighbors, K**, type 1.

   This action saves the prediction equation in Formula Depot. Now you can fit other models using other techniques and compare their performance with that of the K = 1 nearest neighbor model using the Model Comparison platform within Formula Depot.

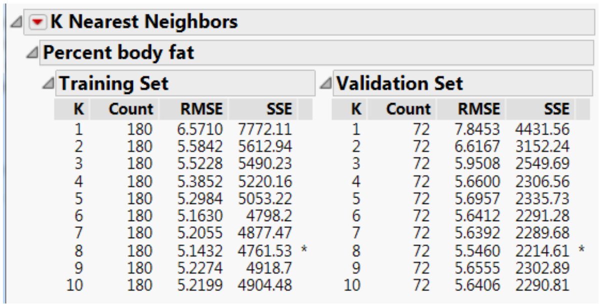
Example of K Nearest Neighbors with Continuous Response

In this example, you want to predict the percent body fat for males using 13 predictors. The Body.jmp sample data table contains percent body fat estimates that are based on underwater weighing and on various body circumference measurements.

1. Select **Help > Sample Data Library** and open Body Fat.jmp.
2. Select **Analyze > Predictive Modeling > K Nearest Neighbors**.
3. Select Percent body fat and click **Y, Response**.

4. Select Age (years) through Wrist circumference (cm) and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.

Figure 8.3 K Nearest Neighbors Report



K Nearest Neighbors

Percent body fat

Training Set

K	Count	RMSE	SSE	
1	180	6.5710	7772.11	
2	180	5.5842	5612.94	
3	180	5.5228	5490.23	
4	180	5.3852	5220.16	
5	180	5.2984	5053.22	
6	180	5.1630	4798.2	
7	180	5.2055	4877.47	
8	180	5.1432	4761.53	*
9	180	5.2274	4918.7	
10	180	5.2199	4904.48	

Validation Set

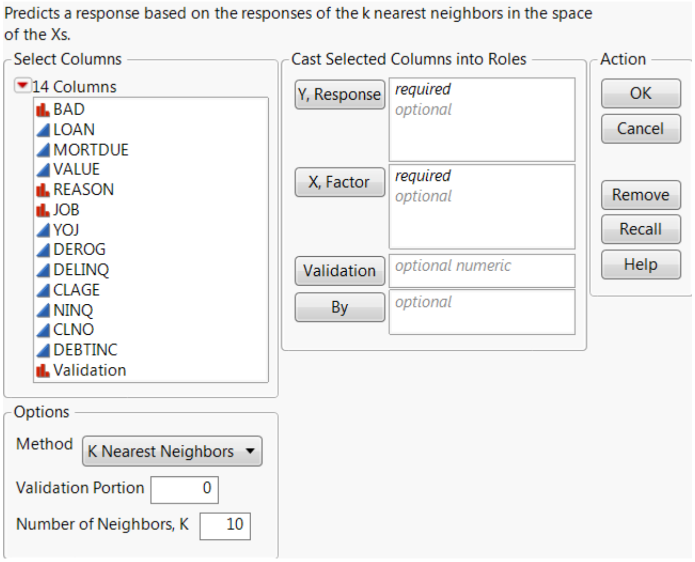
K	Count	RMSE	SSE	
1	72	7.8453	4431.56	
2	72	6.6167	3152.24	
3	72	5.9508	2549.69	
4	72	5.6600	2306.56	
5	72	5.6957	2335.73	
6	72	5.6412	2291.28	
7	72	5.6392	2289.68	
8	72	5.5460	2214.61	*
9	72	5.6555	2302.89	
10	72	5.6406	2290.81	

The K = 8 model had the lowest RMSE for the Validation set. Among *k* nearest neighbor models, the model based on 8 nearest neighbors seems to perform the best.

Launch the K Nearest Neighbors Platform

Launch the K Nearest Neighbors platform by selecting **Analyze > Predictive Modeling > K Nearest Neighbors**.

Figure 8.4 K Nearest Neighbors Launch Window



Predicts a response based on the responses of the k nearest neighbors in the space of the Xs.

Select Columns

14 Columns
BAD
LOAN
MORTDUE
VALUE
REASON
JOB
YOJ
DEROG
DELINQ
CLAGE
NINQ
CLNO
DEBTINC
Validation

Cast Selected Columns into Roles

Y, Response: *required* *optional*
X, Factor: *required* *optional*
Validation: *optional numeric*
By: *optional*

Action

OK
Cancel
Remove
Recall
Help

Options

Method: K Nearest Neighbors
Validation Portion: 0
Number of Neighbors, K: 10

The K Nearest Neighbors launch window provides the following options:

Y, Response The response variable or variables that you want to analyze.

X, Factor The predictor variables.

Validation A numeric column that contains at most three distinct values. See "Validation" on page 93 in the "Partition Models" chapter.

By A column or columns whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate reports. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

Method Enables you to select the partition method (Decision Tree, Bootstrap Forest, Boosted Tree, K Nearest Neighbors, or Naive Bayes). These alternative methods, except for Decision Tree, are available in JMP Pro.

For more details on these methods, see Chapter 5, "Partition Models", Chapter 6, "Bootstrap Forest", Chapter 7, "Boosted Tree", and Chapter 9, "Naive Bayes".

Validation Portion The portion of the data to be used as the validation set. See "Validation" on page 93 in the "Partition Models" chapter.

Number of Neighbors, K Maximum number of nearest neighbors to analyze. Models are fit for one nearest neighbor up to the Number of Neighbors, K that you specify.

The K Nearest Neighbors Report

For each response, the K Nearest Neighbors report is entitled with the name of the response variable and lists information about the models that are fit. The report for the response provides summary information for each of the *K* models that are fit. The report shows tables for the training set and for the validation and test sets if you defined these using validation.

The statistics reported depend on the modeling type of the response. Each row corresponds to a model defined by *K* nearest neighbors, where *K* ranges from one to the value that you specified as Number of Neighbors, K.

Continuous Responses

An asterisk marks the model for the value *K* that has the smallest RMSE. The report for a continuous response contains the following columns:

K Number of nearest neighbors used in the model. *K* ranges from 1 to the Number of Neighbors, K that you specified in the launch window.

Count Number of observations used to fit the model.

RMSE Root mean square error for the model. The model with the smallest RMSE is marked with an asterisk. If there are tied RMSE values, the model with the smallest *K* is marked with the asterisk.

SSE Sum of squared errors for the model.

Categorical Responses

Summary Table

An asterisk marks the model for the value *K* that has the smallest misclassification rate. The report for a categorical response contains the following columns:

K Number of nearest neighbors used in the model. *K* ranges from 1 to the Number of Neighbors, K that you specified in the launch window.

Count Number of observations used to fit the model.

Misclassification Rate Proportion of observations misclassified by the model. This is calculated as Misclassifications divided by Count. The model with the smallest misclassification rate is marked with an asterisk. If there are tied misclassification rates, the model with the smallest *K* is marked with the asterisk.

Misclassifications Gives the number of observations that are incorrectly predicted by the model.

Confusion Matrix

A confusion matrix is shown for the model with the smallest Misclassification Rate (or the model with the smallest *K* if there are ties for the smallest misclassification rate). If you use validation, confusion matrices for the validation and test sets appear. A confusion matrix is a two-way classification of actual and predicted responses. Use the confusion matrices and the misclassification rates to guide your selection of a model.

K Nearest Neighbors Platform Options

The K Nearest Neighbors red triangle menu contains the following options:

Save Predicteds Saves *K* predicted value columns to the data table. The columns are named Predicted <Y, Response> <k>. The k^{th} column contains predictions for the model based on the *k* nearest neighbors, where Y, Response is the name of the response column.

Save Near Neighbor Rows Saves *K* columns to the data table. The columns are named RowNear <k>. For a given row, the k^{th} column contains the row number of its k^{th} nearest neighbor.

Caution: The row numbers in the columns RowNear <k> do not update when you reorder the rows in your data table. If you reorder the rows, the values in those columns are misleading.

Save Prediction Formula Saves a column that contains a prediction formula for a specific *k* nearest neighbor model. Enter a value for *k* when prompted. The prediction formula contains all the training data, so this option might not be practical for large data tables.

Caution: The values obtained from Save Prediction Formula and Save Predicteds do not necessarily match.

Publish Prediction Formula Creates a prediction formula for the specified *k* nearest neighbor model and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the Formula Depot chapter in the *Predictive and Specialized Modeling* book.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Chapter 9

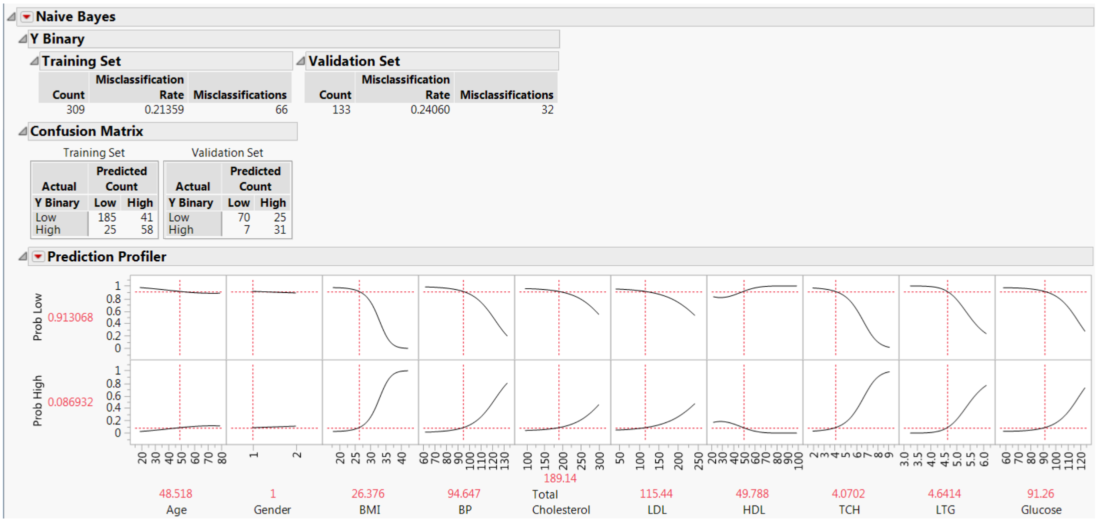
Naive Bayes

Classify Observations Using Probabilistic Assumptions

The Naive Bayes platform is available only in JMP Pro.

The Naive Bayes platform fits a model to predict the value of a categorical variable. Naive Bayes is a fast and computationally simple method. It is especially suitable for situations where there are a large number of predictors.

Figure 9.1 Example of Naive Bayes Analysis



Naive Bayes

Y Binary

Training Set

Count	Misclassification Rate	Misclassifications
309	0.21359	66

Validation Set

Count	Misclassification Rate	Misclassifications
133	0.24060	32

Confusion Matrix

Training Set

Actual	Predicted Count	
Y Binary	Low	High
Low	185	41
High	25	58

Validation Set

Actual	Predicted Count	
Y Binary	Low	High
Low	70	25
High	7	31

Prediction Profiler

Naive Bayes Platform Overview

The Naive Bayes platform classifies observations into *classes* that are defined by the levels of a categorical response variable. The variables (or factors) that are used for classification are often called *features* in the data mining literature.

For each class, the naive Bayes algorithm computes the conditional probability of each feature value occurring. If a feature is continuous, its conditional marginal density is estimated. The naive Bayes technique assumes that, within a class, the features are independent. (This is the reason that the technique is referred to as "naive".) Classification is based on the idea that an observation whose feature values have high conditional probabilities within a certain class has a high probability of belonging to that class. See Hastie et al. (2001).

Because the algorithm estimates only one-dimensional densities or distributions, the algorithm is extremely fast. This makes it suitable for large data sets, and in particular, data sets with large numbers of features. All nonmissing feature values for an observation are used in calculating the conditional probabilities.

Each observation is assigned a *naive score* for each class. An observation's naive score for a given class is the proportion of training observations that belong to that class multiplied by the product of the observation's conditional probabilities. The *naive probability* that an observation belongs to a class is its naive score for that class divided by the sum of its naive scores across all classes. The observation is assigned to the class for which it has the highest naive probability.

Caution: Because the conditional probabilities of class membership are assumed independent, the naive Bayes estimated probabilities are inefficient.

Naive Bayes requires a large number of training observations to ensure representation for all predictor values and classes. If a new observation is being classified and it has a categorical predictor value that was missing in the Training set, then the platform will use the non-missing features to predict. However, if you save a prediction formula, that formula does not handle missing values.

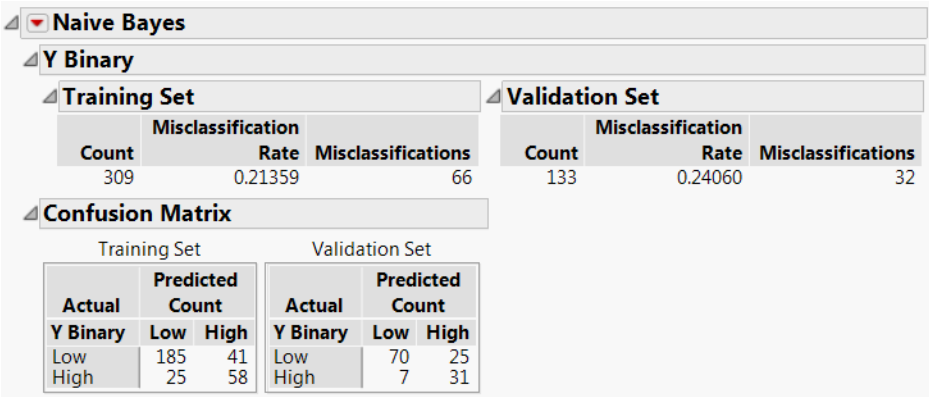
For more information about the naive Bayes technique, see Hand et al. (2016), and Shmueli et al. (2010)

Example of Naive Bayes

You have baseline medical data for 442 diabetic patients. You also have a binary measure of diabetes disease progression obtained one year after each patient's initial visit. This measure quantifies disease progression as being either Low or High. You want to construct a classification model to be used in predicting the disease progression for future patients as High or Low.

1. Select **Help > Sample Data Library** and open Diabetes.jmp.
2. Select **Analyze > Predictive Modeling > Naive Bayes**.
3. Select Y Binary and click **Y, Response**.
4. Select Age through Glucose and click **X, Factor**.
5. Select Validation and click **Validation**.
6. Click **OK**.

Figure 9.2 Naive Bayes Report



Naive Bayes

Y Binary

Training Set

Count	Misclassification Rate	Misclassifications
309	0.21359	66

Validation Set

Count	Misclassification Rate	Misclassifications
133	0.24060	32

Confusion Matrix

Training Set

Actual	Predicted Count	
Y Binary	Low	High
Low	185	41
High	25	58

Validation Set

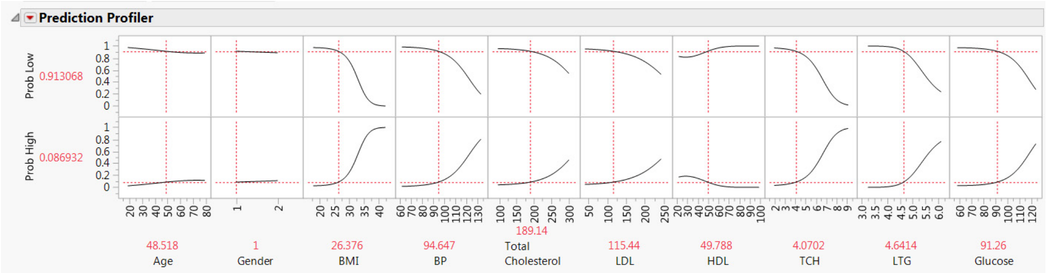
Actual	Predicted Count	
Y Binary	Low	High
Low	70	25
High	7	31

The Training Set has about a 21% misclassification rate and the Validation Set has about a 24% misclassification rate. The Confusion matrix suggests that, for both the Training and Validation sets, the larger source of misclassification comes from classifying patients with Low disease progression as having High disease progression. The Validation set results indicate how your model extends to independent observations.

You are interested in which individual predictors have the greatest impact on the naive Bayes classification.

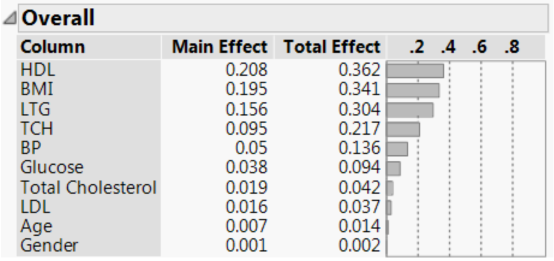
7. Click the red triangle next to Naive Bayes and select **Profiler**.

Figure 9.3 Prediction Profiler for Disease Progression



8. Click the red triangle next to Prediction Profiler and select **Assess Variable Importance > Independent Uniform Inputs**.

Figure 9.4 Variable Importance

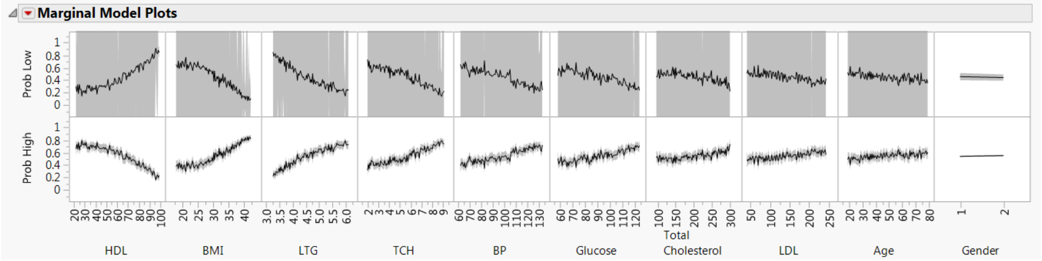


Overall

Column	Main Effect	Total Effect
HDL	0.208	0.362
BMI	0.195	0.341
LTG	0.156	0.304
TCH	0.095	0.217
BP	0.05	0.136
Glucose	0.038	0.094
Total Cholesterol	0.019	0.042
LDL	0.016	0.037
Age	0.007	0.014
Gender	0.001	0.002

The Summary Report indicates that HDL, BMI, and LTG have the greatest impact on the estimated probabilities.

Figure 9.5 Marginal Model Plots Report

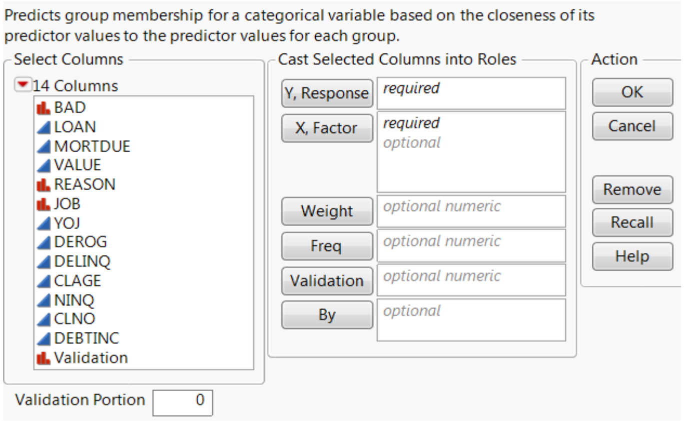


The second row of plots in the Marginal Model Plots report shows that higher values of HDL are associated with a lower probability of classifying a patient as High. Also, higher BMI and LTG values are associated with a higher probability of classifying a patient as High.

JMP PRO Launch the Naive Bayes Platform

Launch the Naive Bayes platform by selecting **Analyze > Predictive Modeling > Naive Bayes**.

Figure 9.6 Naive Bayes Launch Window



Predicts group membership for a categorical variable based on the closeness of its predictor values to the predictor values for each group.

Validation Portion 0

The Naive Bayes launch window provides the following options:

Y, Response The categorical response column whose values are the classes of interest.

X, Factor Categorical or continuous predictor columns.

Weight A column whose numeric values assign a weight to each row in the analysis.

Freq A column whose numeric values assign a frequency to each row in the analysis.

Validation A numeric column that contains at most three distinct values. See "Validation" on page 93 in the "Partition Models" chapter.

Note: If neither a Validation column or a Validation Portion is specified in the launch window and if there are excluded rows, these rows are treated as a Validation set.

By A column or columns whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate reports. If more than one By variable is assigned, a separate analysis is produced for each possible combination of the levels of the By variables.

Validation Portion The portion of the data to be used as the Validation set. See "Validation" on page 93 in the "Partition Models" chapter.

The Naive Bayes Report

After you click **OK** in the launch window, the Naive Bayes report appears. By default, the Naive Bayes report contains a report for the response column and a Confusion Matrix report.

Response Column Report

The response column report shows performance statistics for the naive Bayes classification in a summary table for the Training set, and the Validation and Test sets if they are specified. The summary tables contain the following columns:

Count Number of observations in the set corresponding to the table (Training, Validation, or Test set).

Misclassification Rate Proportion of observations in the corresponding set that are misclassified by the model. This is calculated as Misclassifications divided by Count.

Misclassifications Number of observations in the corresponding set that are classified incorrectly.

Confusion Matrix Report

The Confusion Matrix report shows a confusion matrix for the Training set, and for the Validation and Test sets if they are specified. A confusion matrix is a two-way classification of actual and predicted responses.

Naive Bayes Platform Options

The Naive Bayes red triangle menu contains the following options:

Save Predicteds Saves the predicted classifications to the data table in a column called Naive Predicted <Y, Response>.

Save Prediction Formula Saves a column called Naive Predicted Formula <Y, Response> to the data table. This column contains the prediction formula for the classifications.

Save Probability Formula Saves columns to the data table that contain formulas used for classifying each observation. Three groups of columns are saved:

Naive Score <Class>, Sum For each column that represents a class, this column gives a score formula that measures strength of membership in the given class. In the Naive Score Sum column., these scores are summed across classes. See "Saved Probability Formulas" on page 155.

Naive Prob <Class> For each class, this column gives a formula for the conditional probability that an observation is in that class. See “Saved Probability Formulas” on page 155.

Naive Predicted Formula <Y, Response> Gives the formula for the predicted class.

Publish Probability Formula Creates probability formulas and saves them as formula column scripts in the Formula Depot platform. If a Formula Depot report is not open, this option opens the Formula Depot window. See the Formula Depot chapter in the *Predictive and Specialized Modeling* book.

Profiler Shows or hides an interactive profiler report. Changes in the factor values are reflected in the estimated classification probabilities. See the Profiler chapter in the *Profilers* book for more information.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Additional Example of Naive Bayes

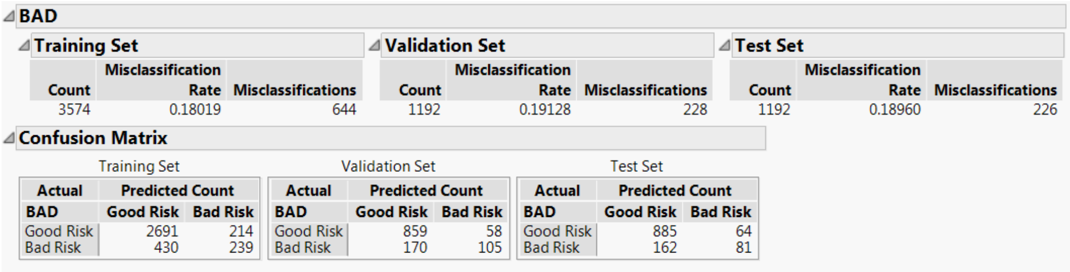
You have historical financial data for 5,960 customers who applied for home equity loans. Each customer was classified as being a Good Risk or Bad Risk. There is missing data on most of the predictors. You want to construct a model to use in classifying the credit risk of future customers.

1. Select **Help > Sample Data Library** and open Equity.jmp.
2. Select **Analyze > Predictive Modeling > Naive Bayes**.
3. Select BAD and click **Y, Response**.

   One of the potential predictors, DEBTINC, has many missing values that might be informative. However, naive Bayes is not prepared to handle large number of missing values well, so you do not include DEBTINC in your model.
4. Select LOAN through CLNO and click **X, Factor**.

5. Select Validation and click **Validation**.
6. Click **OK**.

Figure 9.7 Naive Bayes Report for BAD



BAD

Training Set

Count	Misclassification Rate	Misclassifications
3574	0.18019	644

Validation Set

Count	Misclassification Rate	Misclassifications
1192	0.19128	228

Test Set

Count	Misclassification Rate	Misclassifications
1192	0.18960	226

Confusion Matrix

Training Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2691	214
Bad Risk	430	239

Validation Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	859	58
Bad Risk	170	105

Test Set

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	885	64
Bad Risk	162	81

The Training, Validation, and Test sets show misclassification rates between 18% and 19%. The confusion matrices for all of the sets suggest that the largest source of misclassification is the classification of Bad Risk customers as Good Risk customers.

You are interested in the probabilities that customers with certain financial background values are classified as High Risk.

7. Click the red triangle next to Naive Bayes and select **Save Probability Formulas**.

Three sets of columns are added to the data table. Notice that observations with any missing predictor values have missing values in the new columns.

- The three Naive Score columns contain naive score formulas for Good Risk, Bad Risk, and the sum of both.
- The two Naive Prob columns contain probability formulas for Good Risk and Bad Risk.
- The Naive Predicted Formula Bad column contains a formula that assigns an observation to the class for which the observation has the highest naive probability.

Use these formulas to score new customers. For details about the formula columns, see "Saved Probability Formulas" on page 155.

Statistical Details for the Naive Bayes Platform

Algorithm

The naive Bayes method classifies an observation into the class for which its probability of membership, given the values of its features, is highest. The method assumes that the features are conditionally independent within each class.

Denote the possible classifications by $C_1, \ldots, C_k$. Denote the features, or predictors, by $X_1, X_2, \ldots, X_p$.

The conditional probability that an observation with $X_j = x_j$ belongs to the class C_r is given as follows:

- If X_j is categorical: $P(C_r | x_j)$
- If X_j is continuous:

$$P(C_r|x_j) = \frac{1}{s}\phi((x_j - m)/s)$$

Here, ϕ is the standard normal density function, and m and s are the mean and standard deviation, respectively, of the predictor values within the class C_r.

The conditional probability of that an observation with predictor values $x_1, x_2, \ldots, x_p$ belongs in the class C_r is computed as follows:

$$P(C_r|(x_1, \ldots, x_p)) = \left(P(C_r)\prod_{j=1}^{p}[P(x_j|C_r)]\right) \Big/ \left(\sum_{i=1}^{k} P(C_i)\left(\prod_{j=1}^{p}[P(x_j|C_i)]\right)\right)$$

An observation is classified into the class for which its conditional probability is the largest.

Saved Probability Formulas

This section describes the formulas saved using the Save Probability Formula option. The conditional probability that an observation with predictor values $x_1, x_2, \ldots, x_p$ belongs in the class C_r is given by $P(C_r|(x_1, \ldots, x_p))$ as shown in the section "Algorithm" on page 154.

Naive Score Formulas

The Naive Score formula for a given class C_r is the numerator in the expression for $P(C_r|(x_1, \ldots, x_p))$.

The Naive Score Sum formula sums the conditional probabilities $P(C_r|(x_1, \ldots, x_p))$ over all classes. This is the denominator in the expression for $P(C_r|(x_1, \ldots, x_p))$.

Naive Prob Formulas

The Naive Prob formula for a given class C_r equals $P(C_r|(x_1, \ldots, x_p))$.

Naive Predicted Formula

The Naive Predicted Formula for an observation classifies that observation into the class for which $P(C_r|(x_1, \ldots, x_p))$ is the largest. This is equivalent to classifying an observation into the class for which its Naive Score formula is the largest.

Chapter 10

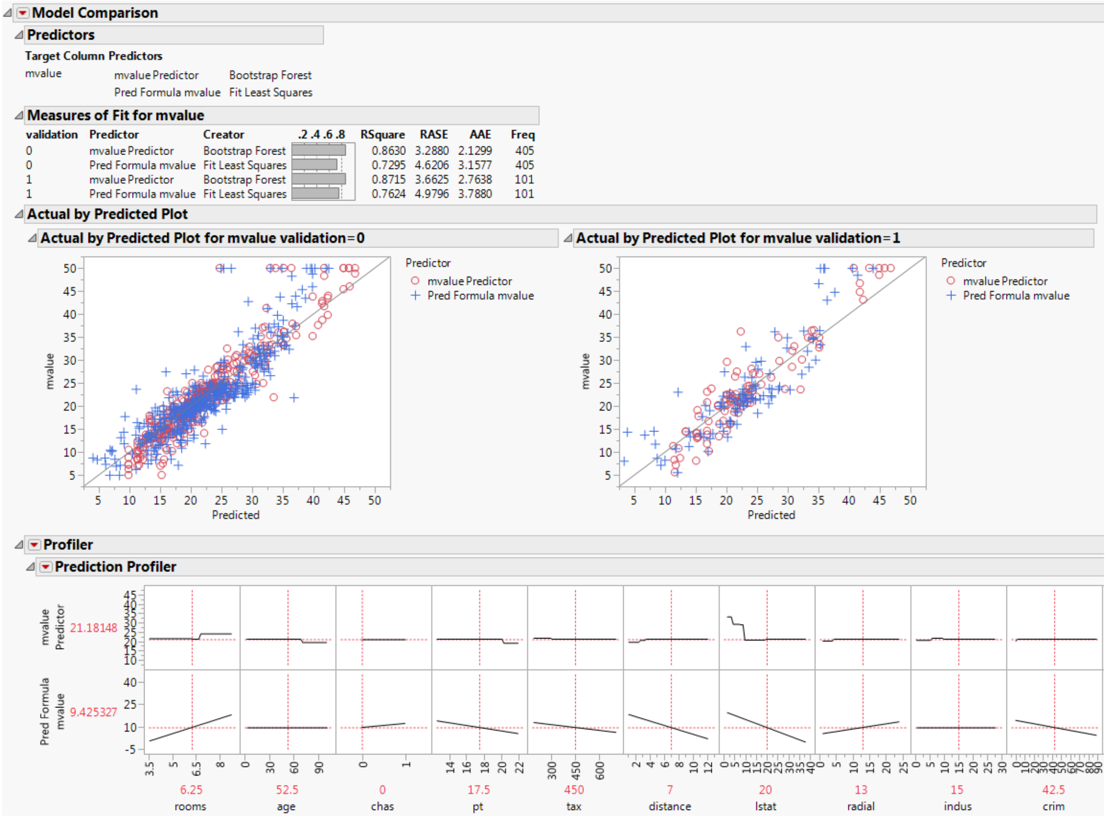
Model Comparison

Compare the Predictive Ability of Fitted Models

The Model Comparison platform is available only in JMP Pro.

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 10.1 Example of Comparing Models



Model Comparison

Predictors

Target Column	Predictors	
mvalue	mvalue Predictor	Bootstrap Forest
	Pred Formula mvalue	Fit Least Squares

Measures of Fit for mvalue

validation	Predictor	Creator	.2 .4 .6 .8	RSquare	RASE	AAE	Freq
0	mvalue Predictor	Bootstrap Forest		0.8630	3.2880	2.1299	405
0	Pred Formula mvalue	Fit Least Squares		0.7295	4.6206	3.1577	405
1	mvalue Predictor	Bootstrap Forest		0.8715	3.6625	2.7638	101
1	Pred Formula mvalue	Fit Least Squares		0.7624	4.9796	3.7880	101

Actual by Predicted Plot

Actual by Predicted Plot for mvalue validation=0

Actual by Predicted Plot for mvalue validation=1

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 selecting **Help > Sample Data Library** and opening Boston Housing.jmp.

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**.

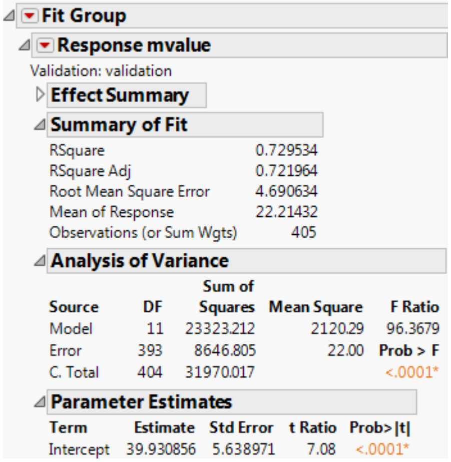
   The rows assigned a 0 are the training set. The rows assigned a 1 are the validation set.

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. Select **P-value Threshold** from the Stopping Rule list.
8. Click the **Go** button.
9. Click the **Run Model** button.

   The Fit Group report appears, a portion of which is shown in Figure 10.2.
10. Save the prediction formula to a column by selecting **Save Columns > Prediction Formula** on the Response red triangle menu.

Figure 10.2 Fit Model Report



Fit Group

Response mvalue

Validation: validation

Effect Summary

Summary of Fit

RSquare	0.729534
RSquare Adj	0.721964
Root Mean Square Error	4.690634
Mean of Response	22.21432
Observations (or Sum Wgts)	405

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	11	23323.212	2120.29	96.3679
Error	393	8646.805	22.00	Prob > F
C. Total	404	31970.017		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	39.930856	5.638971	7.08	<.0001*

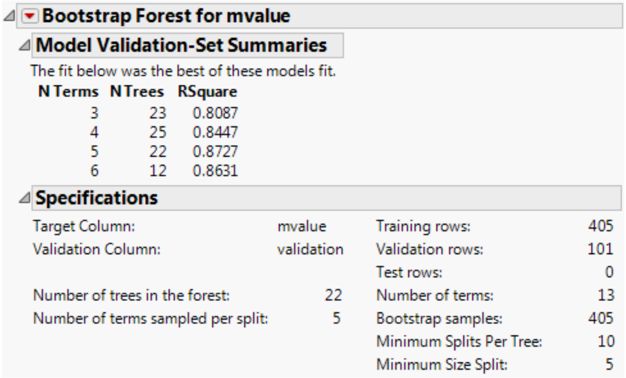
Create the Bootstrap Forest Model and Save the Prediction Formula to a Column

1. Select **Analyze > Predictive 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 10.3.

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

Figure 10.3 Bootstrap Forest Model



Bootstrap Forest for mvalue

Model Validation-Set Summaries

The fit below was the best of these models fit.

N Terms	N Trees	RSquare
3	23	0.8087
4	25	0.8447
5	22	0.8727
6	12	0.8631

Specifications

Target Column:	mvalue	Training rows:	405
Validation Column:	validation	Validation rows:	101
		Test rows:	0
Number of trees in the forest:	22	Number of terms:	13
Number of terms sampled per split:	5	Bootstrap samples:	405
		Minimum Splits Per Tree:	10
		Minimum Size Split:	5

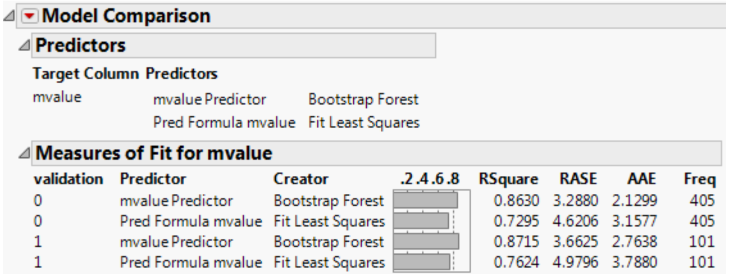
Compare the Models

1. Select **Analyze > Predictive 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 10.4).

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

Figure 10.4 Model Comparison Report



Model Comparison

Predictors

Target Column	Predictors	
mvalue	mvalue Predictor	Bootstrap Forest
	Pred Formula mvalue	Fit Least Squares

Measures of Fit for mvalue

validation	Predictor	Creator	.2.4.6.8	RSquare	RASE	AAE	Freq
0	mvalue Predictor	Bootstrap Forest		0.8630	3.2880	2.1299	405
0	Pred Formula mvalue	Fit Least Squares		0.7295	4.6206	3.1577	405
1	mvalue Predictor	Bootstrap Forest		0.8715	3.6625	2.7638	101
1	Pred Formula mvalue	Fit Least Squares		0.7624	4.9796	3.7880	101

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.

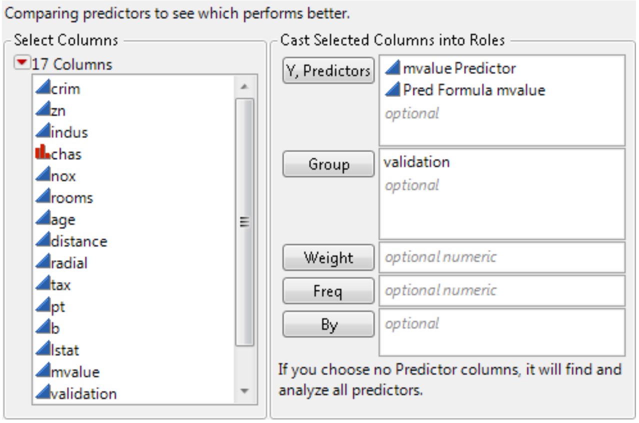
Related Information

- The Model Specification chapter in the *Fitting Linear Models* book
- "Partition Models" chapter on page 71

Launch the Model Comparison Platform

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

Figure 10.5 The Model Comparison Launch Window



Comparing predictors to see which performs better.

Select Columns

17 Columns: crim, zn, indus, chas, nox, rooms, age, distance, radial, tax, pt, b, lstat, mvalue, validation

Cast Selected Columns into Roles

Y, Predictors: mvalue Predictor, Pred Formula mvalue, optional

Group: validation, optional

Weight: optional numeric

Freq: optional numeric

By: optional

If you choose no Predictor columns, it will find and analyze all predictors.

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. Prediction formula columns created by JMP platforms have either the Predicting or Response Probability column property. If you specify a column that does not contain one of these properties, the platform prompts you to specify which column is being predicted by the specified Y column.

For a categorical response with *k* levels, most model fitting platforms save *k* columns to the data table, each predicting the probability for a level. All *k* columns need to be specified as **Y, Predictors**. For platforms that do not save *k* columns of probabilities, the column containing the predicted response level can be specified as a **Y, Predictors** column.

If you do not specify any **Y, Predictors** columns, JMP uses the prediction formula columns in the data table that have either the Predicting or Response Probability column property.

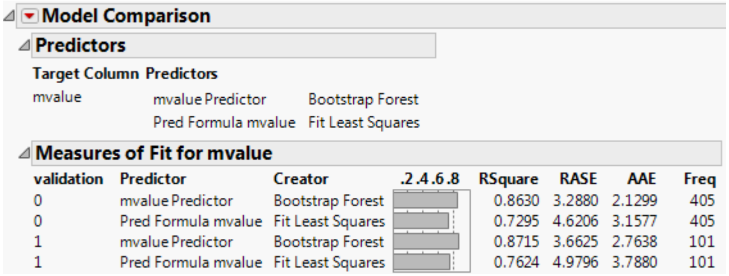
Group The column that separates the data into groups, which are fit separately.

The other role buttons are common among JMP platforms. See the Get Started chapter in the *Using JMP* book for details.

The Model Comparison Report

Figure 10.6 shows an example of the initial Model Comparison report for a continuous response.

Figure 10.6 Initial Model Comparison Report



Model Comparison

Predictors

Target Column	Predictors	
mvalue	mvalue Predictor	Bootstrap Forest
	Pred Formula mvalue	Fit Least Squares

Measures of Fit for mvalue

validation	Predictor	Creator	.2.4.6.8	RSquare	RASE	AAE	Freq
0	mvalue Predictor	Bootstrap Forest		0.8630	3.2880	2.1299	405
0	Pred Formula mvalue	Fit Least Squares		0.7295	4.6206	3.1577	405
1	mvalue Predictor	Bootstrap Forest		0.8715	3.6625	2.7638	101
1	Pred Formula mvalue	Fit Least Squares		0.7624	4.9796	3.7880	101

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 square root of the mean squared prediction error. This is computed as follows:

- Square and sum the prediction errors (differences between the actual responses and the predicted responses) to obtain the *SSE*.
- Denote the number of observations by *n*.
- RASE is:

$$\text{RASE} = \sqrt{\frac{SSE}{n}}$$

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 negative log-likelihoods from the fitted model and the constant probability model. It ranges from 0 to 1.

Generalized RSquare A measure that can be applied to general regression models. It is based on the likelihood function L and is scaled to have a maximum value of 1. The value is 1 for a perfect model, and 0 for a model no better than a constant model. The Generalized RSquare measure simplifies to the traditional RSquare for continuous normal responses in the standard least squares setting. Generalized RSquare is also known as the Nagelkerke or Craig and Uhler R^2, which is a normalized version of Cox and Snell's pseudo R^2. See Nagelkerke (1991).

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 65 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
- 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. A cumulative gains curve is a plot of the proportion of a response level that is identified by the model against the proportion of all responses. A cumulative gains curve for a perfect model would reach 1.0 at the overall proportion of the response level. The curves for the different models are overlaid.

Confusion Matrix Shows confusion matrices for each model. A confusion matrix is a two-way classification of actual and predicted responses. Count and rate confusion matrices are shown. Separate confusion matrices are produced for each level of the Group variable.

If the response has a Profit Matrix column property, then Actual by Decision Count and Actual by Decision Rate matrices are shown to the right of the confusion matrices. For details about these matrices, see “Additional Examples of Partitioning” on page 95 in the “Partition Models” chapter.

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 90 in the “Partition Models” chapter
- “Lift Curve” on page 91 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 selecting **Help > Sample Data Library** and opening Car Physical Data.jmp.

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 > Predictive Modeling > Partition**.
2. Select Type and click **Y, Response**.
3. Select the Country, Weight, Turning Cycle, Displacement, and Horsepower columns and click **X, Factor**.
4. Make sure that **Decision Tree** is selected in the Method list.
5. Click **OK**.

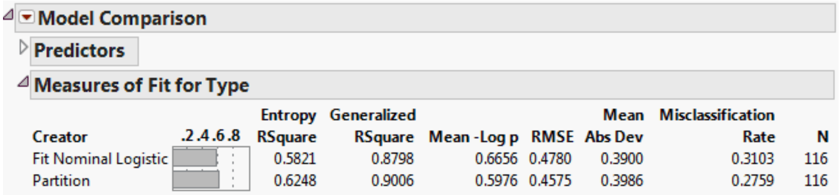
   The Partition report appears.
6. Click **Split** 10 times.
7. Save the prediction formulas to columns by selecting **Save Columns > Save Prediction Formula** from the Partition red triangle menu.

Compare the Models

1. Select **Analyze > Predictive Modeling > Model Comparison**.
2. Select all columns that begin with Prob and click **Y, Predictors**.
3. Click **OK**.

   The Model Comparison report appears (Figure 10.7).

Figure 10.7 Initial Model Comparison Report



Model Comparison

Predictors

Measures of Fit for Type

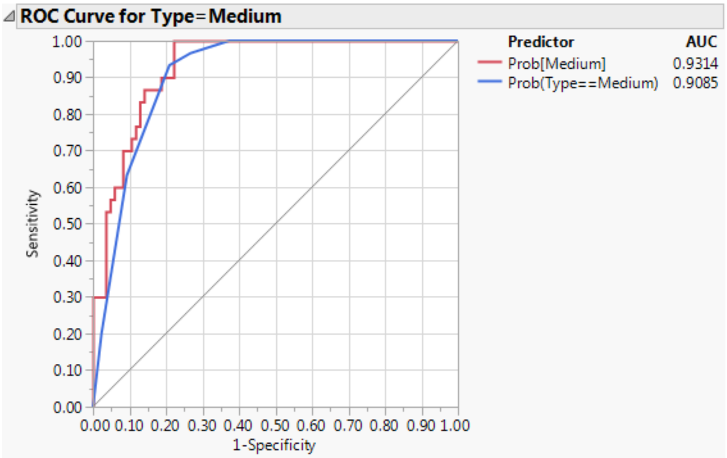
Creator	.2 .4 .6 .8	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Fit Nominal Logistic		0.5821	0.8798	0.6656	0.4780	0.3900	0.3103	116
Partition		0.6248	0.9006	0.5976	0.4575	0.3986	0.2759	116

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 10.8.

Figure 10.8 ROC Curve for Medium



ROC Curve for Type=Medium

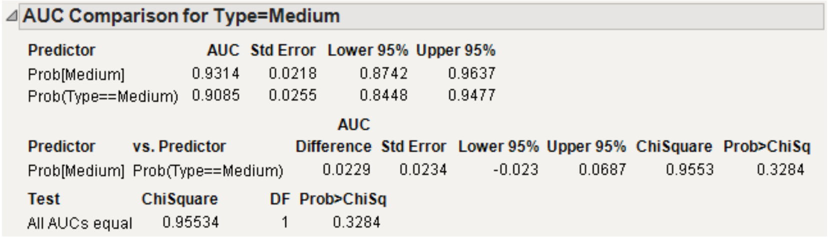
Predictor	AUC
Prob[Medium]	0.9314
Prob(Type==Medium)	0.9085

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 10.9.

Figure 10.9 AUC Comparison for Medium



AUC Comparison for Type=Medium

Predictor	AUC	Std Error	Lower 95%	Upper 95%
Prob[Medium]	0.9314	0.0218	0.8742	0.9637
Prob(Type==Medium)	0.9085	0.0255	0.8448	0.9477

		AUC					
Predictor	vs. Predictor	Difference	Std Error	Lower 95%	Upper 95%	ChiSquare	Prob>ChiSq
Prob[Medium]	Prob(Type==Medium)	0.0229	0.0234	-0.023	0.0687	0.9553	0.3284

Test	ChiSquare	DF	Prob>ChiSq
All AUCs equal	0.95534	1	0.3284

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.

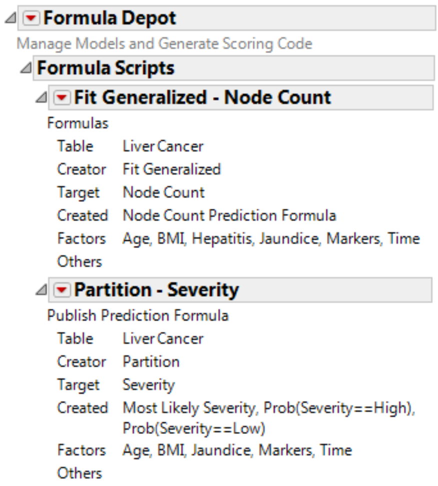
Formula Depot

Manage Models and Generate Scoring Code

The Formula Depot Platform is available only in JMP Pro.

The Formula Depot is a repository to organize, compare, and profile models. Scoring code for deployment within or outside of JMP can be generated for models published to the Formula Depot. For model exploration work, you can use the Formula Depot to store candidate models outside of your JMP data table. The model profiler and model compare platforms are accessible from the Formula Depot. A model that is selected for further use can be saved to your JMP table or saved to a JMP table with new data for scoring. For use in an environment outside of JMP, you can generate scoring code in C, Python, JavaScript, SAS, or SQL.

Figure 11.1 Example of the Formula Depot



Formula Depot
Manage Models and Generate Scoring Code
Formula Scripts
Fit Generalized - Node Count
Formulas

Table	Liver Cancer
Creator	Fit Generalized
Target	Node Count
Created	Node Count Prediction Formula
Factors	Age, BMI, Hepatitis, Jaundice, Markers, Time
Others	

Partition - Severity
Publish Prediction Formula

Table	Liver Cancer
Creator	Partition
Target	Severity
Created	Most Likely Severity, Prob(Severity==High), Prob(Severity==Low)
Factors	Age, BMI, Jaundice, Markers, Time
Others	

Formula Depot Platform Overview

The Formula Depot is a repository to organize, compare, profile, and score models for deployment. Models are prediction formulas. Prediction formulas are saved to the Formula Depot as column scripts. You can add prediction formulas to JMP tables to score data. You can also use the Formula Depot to generate scoring code for prediction formulas to facilitate the deployment of models in environments outside of JMP.

The Formula Depot enables you to perform the following tasks:

- save prediction formulas outside of data tables
- save prediction formulas from multiple data tables in a common location
- compare models
- profile models
- add prediction formulas to a data table for scoring of new data within JMP
- generate scoring code (C, Python, Java Script, SAS, or SQL) for deploying models outside of JMP

Example of Formula Depot

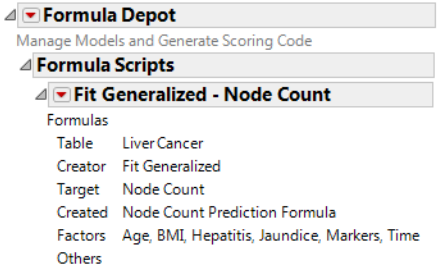
The Liver Cancer.jmp sample data table contains data on the severity of liver cancer in patients when they entered a study. The file also contains a number of table scripts for models. This example uses these scripts to generate models to demonstrate the Formula Depot.

1. Select **Help > Sample Data Library** and open Liver Cancer.jmp.
2. Click the green triangle next to the Lasso Poisson, Validation Column script.
3. Click the red triangle next to Adaptive Lasso with Validation Column and select **Save Columns > Publish Prediction Formula.**

   This option opens a Formula Depot that contains the prediction formula for the Fit Generalized model.
4. To add the prediction formula to the data table, select **Run Script** from the Fit Generalized - Node Count red triangle menu in the Formula Depot.
5. To generate C scoring code for use outside of JMP select **Generate C Code** from the Fit Generalized - Node Count red triangle menu. A script window appears containing the C code. See "Generating Scoring Code from the Formula Depot Platform" on page 173.
6. To save the Formula Depot, select **File > Save**.

Note: The result of step 3 and step 4 can be obtained using the **Save Columns > Save Prediction Formula** at step 3**.** However, then the prediction formula is not part of the Formula Depot.

Figure 11.2 Formula Depot with a Generalized Model



Formula Depot

Manage Models and Generate Scoring Code

Formula Scripts

Fit Generalized - Node Count

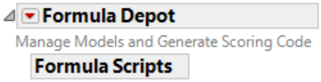
Formulas

Table	Liver Cancer
Creator	Fit Generalized
Target	Node Count
Created	Node Count Prediction Formula
Factors	Age, BMI, Hepatitis, Jaundice, Markers, Time
Others	

Launch the Formula Depot Platform

Launch the Formula Depot by selecting **Analyze > Predictive Modeling > Formula Depot**.

Figure 11.3 Empty Formula Depot from Launch



Formula Depot

Manage Models and Generate Scoring Code

Formula Scripts

Alternatively, if there is not an open Formula Depot then a Formula Depot opens when you select a Publish command.

Platforms That Publish Prediction Formulas to the Formula Depot

The platforms that publish prediction formulas and generate complete scoring code include:

- Discriminant
- Least Squares Regression
- Logistic Regression
- Partition
- Uplift
- K Nearest Neighbors
- Naive Bayes
- Neural

- Latent Class Analysis
- Principal Components
- Generalized Regression
- PLS
- Gaussian Process

In platforms that do not publish prediction formulas to the Formula Depot, you can save the prediction formula to the data table. From the data table add it to the Formula Depot by selecting **Add Formula from Column**. However, the scoring code might not be fully functional for such models.

For details about scoring code see “Generating Scoring Code from the Formula Depot Platform” on page 173.

Formula Depot Platform Options

Add Formula From Column Enables you to add an existing prediction formula column to the current formula depot.

Show Scripts Opens a new Formula Window (or appends to an open Formula Window) that contains scripts for all of the formulas that are in the current formula depot.

Copy Scripts Copies all scripts from the current formula depot to the clipboard.

Copy Formulas as Transforms Enables you to select models from the current formula depot to be copied. Selected models are copied onto the clipboard within a `Transform Columns()` statement.

Run Scripts Enables you to save models from the current formula depot to new columns in your JMP data table.

Generate C Code, Generate Python Code, Generate JavaScript Code, Generate SAS Code, and Generate SQL Code Enables you to select models from the current formula depot for code generation. A new script window appears that contains scoring code for the selected models in C, Python, JavaScript, SAS DS2, or SQL, respectively. You can use this code to facilitate the deployment of the model in the environment or framework of your choice. See “Generating Scoring Code from the Formula Depot Platform” on page 173.

Model Comparison Enables you to select models from the current formula depot to be compared using the model comparison utility. If models from multiple tables are stored in the depot you first select the table of interest and then the models of interest. See the Model Comparison chapter in the *Predictive and Specialized Modeling* book.

Remove Model Comparison Removes all Model Comparison reports from the current formula depot.

Profiler Enables you to select models from the current formula depot to be profiled using the profiler. If models from multiple tables are stored in the depot you first select the table of interest and then the models of interest. See the Profiler chapter in the *Profilers* book.

Remove Profiler Removes all Profilers from the current formula depot.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Formula Depot Model Options

Each prediction formula that is saved in the Formula Depot has an individual options menu. In addition to Show, Copy, Run, and Code Generation options that correspond to the main menu options, the individual menus include the following options:

Rename New Column Enables you to rename the model. This new name is applied as the column name if you run the script to add the prediction formula to a JMP data table. In addition, this name is included in generated code.

Remove removes the model from the Formula Depot. This command cannot be undone.

Generating Scoring Code from the Formula Depot Platform

Scoring code generation is intended to facilitate using models built in JMP in a production environment or other framework of your choice. Many platforms publish prediction formulas to the Formula Depot, however; not all prediction formulas generate complete code. Code can be complete, it can be a code fragment, or the code can include unsupported functions that require additional programming for implementation.

For example, you can perform the following tasks:

- Deploy your model to SAS Model Manager using the generated SAS code.
- Augment your ETL process with in-database scoring using the generated SQL code.
- Create a node for a data transformation pipeline with an application built with the generated C code.

- Create a Jupyter notebook to show live scoring results using the generated Python code.
- Enable customers to score their own data with a web application that includes the generated JavaScript.

For the C, Python, and JavaScript languages, you must include supporting code such as .h files and utility libraries when deploying or compiling the generated code. These files are available in your JMP installation folder inside the Scoring folder

C Code The C scoring code that is generated must be compiled into a library and then linked into an application. You might use either a static or a dynamic link approach. The files jmp_lib.h, jmp_parms.h, and jmp_score.h needed for compiling and linking can be found in the Scoring/C folder in your JMP installation folder.

Python Code The file jmp_score.py that is needed to run your Python scoring application can be found in the Scoring/Python folder in your JMP installation folder.

JavaScript Code The file jmp_score.js that is needed to run your JavaScript scoring application can be found in Scoring/JavaScript folder.

SAS Code The generated code fragment that once wrapped by PROC DS2 statements can be used in SAS applications including the SAS In-Database Code Accelerator.

Tip: For models with an ifmax call, such as logistic or neural models for a categorical response, move temporary variable declarations before the **method run()** statement. Rename variable names to conform to SAS naming conventions.

SQL Code The SQL code fragment that once wrapped in a select statement can be used in SQL queries against most major database servers.

Note: Code that contains “placeholder” or “ERROR” indicates an unsupported function call.

Chapter 12

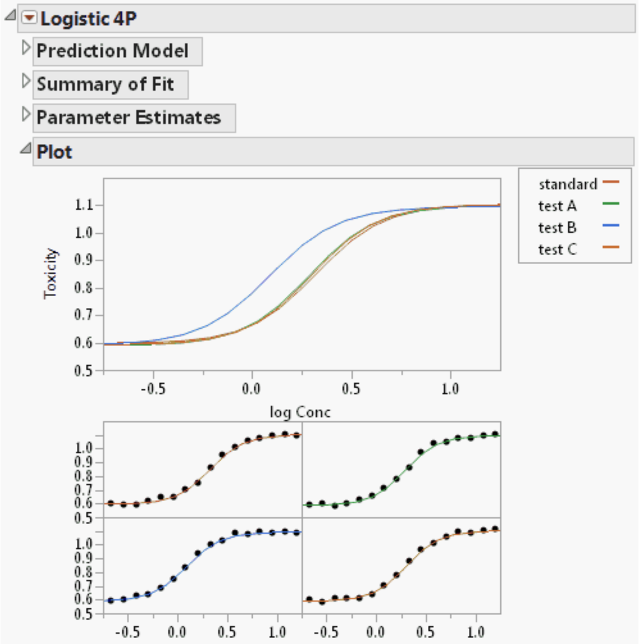
Fit Curve

Fit Built-In Nonlinear Models to Your Data

In many situations, especially in the physical and biological sciences, well-known nonlinear equations describe the relationship between variables. For example, pharmacological bioassay experiments can demonstrate how the strength of the response to a drug changes as a function of drug concentration. Sigmoid curves often accurately model response strength as a function of drug concentration. Another example is exponential growth curves, which can model the size of a population over time.

The Fit Curve personality does not require you to specify starting values for parameter estimates or create model formulas. To specify your own starting values and create model formulas, use the more powerful custom Nonlinear personality, which can also fit any nonlinear model. For details, see the "Nonlinear Regression" chapter on page 197.

Figure 12.1 Example of Nonlinear Fit in the Fit Curve Personality



Introduction to the Fit Curve Platform

Some models are *linear* in the parameters (for example, a quadratic or other polynomial); others 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. An example in the Model Specification chapter in the *Fitting Linear Models* book shows a significant linear relationship between oxygen uptake and time spent running. For more information about Fit Model, see the Model Specification chapter in the *Fitting Linear Models* book. For more information about Fit Y by X, see the Introduction to Fit Y by X chapter in the *Basic Analysis* book.

The Fit Curve platform enables you to fit models that are *nonlinear* in the parameters. The initial example in this chapter shows the analysis of a nonlinear relationship: drug toxicity as a function of concentration. The effect of concentration on toxicity changes from low to high doses, so this relationship is nonlinear.

The following are examples of equations for linear and nonlinear functions.

- Linear function: $Y=\beta_0+\beta_1 e^X$
- Nonlinear function: $Y=\beta_0+\beta_1 e^{\beta 2x}$

The Fit Curve platform provides predefined models, such as polynomial, logistic, probit, 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.

Fit Curve also enables you to build a model to create the prediction formula. Then you set upper and lower parameter limits in Nonlinear. For details, see "Example of Setting Parameter Limits" on page 216 in the "Nonlinear Regression" chapter.

Example Using the Fit Curve Personality

This example shows how to build a model for toxicity as a function of the concentration of a drug. You have a standard formulation of the drug and want to compare it to three new formulations.

You are interested in a toxicity ratio of surviving to non-surviving cells at a specific concentration of each drug. From prior research, you know the toxicity ratios for 16 different concentrations of each drug formulation. A lower ratio indicates more toxicity, which could be detrimental to development of the drug. Log concentration was calculated to decrease the range of concentration values and make it easier to detect differences in the curves.

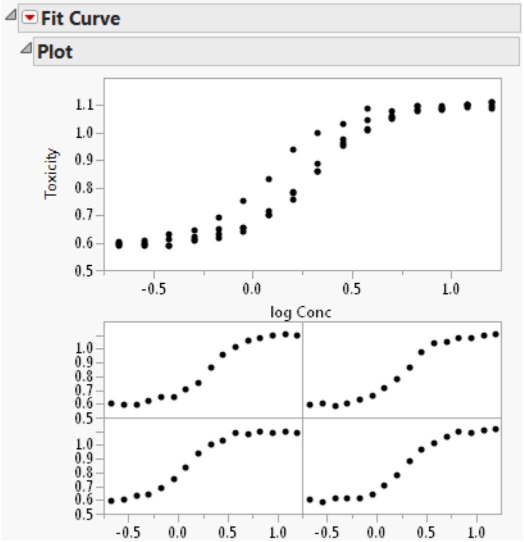
Follow these steps to build the model:

1. Select **Help > Sample Data Library** and open Nonlinear Examples/Bioassay.jmp.

2. Select **Analyze > Specialized Modeling > Fit Curve**.
3. Assign Toxicity to the **Y, Response** role.
4. Assign log Conc to the **X, Regressor** role.
5. Assign Formulation to the **Group** role.
6. Click **OK**.

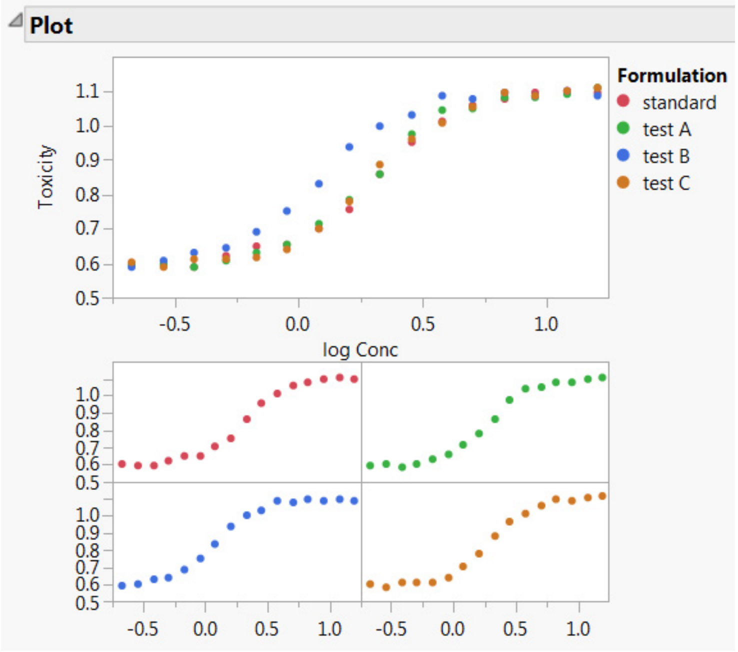
   The Fit Curve Report appears as shown in Figure 12.2. The Plot report contains an overlaid plot of the fitted model of each formulation.

Figure 12.2 Initial Fit Curve Report



7. To see a legend identifying each drug formulation, right-click one of the graphs and select **Row Legend**. Select **Formulation** for the column and click **OK.** The plot shown in Figure 12.3 appears.

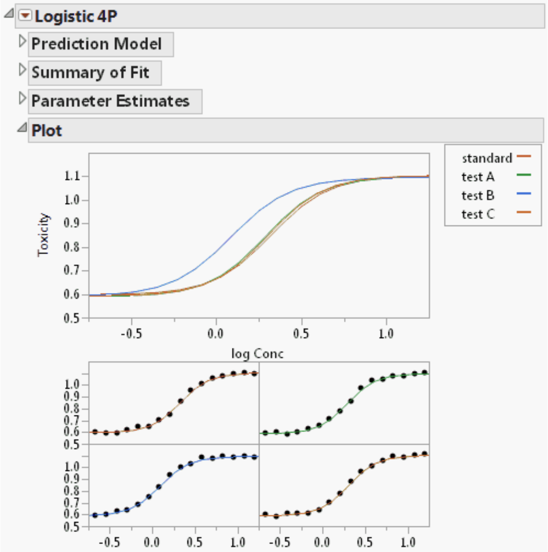
Figure 12.3 Fit Curve Report with Plot Legend



The curves appear S-shaped, so a sigmoid curve would be an appropriate fit. Table 12.1 shows formulas and graphical depictions of the different types of models that the Fit Curve personality offers.

8. Select **Sigmoid Curves > Logistic Curves > Fit Logistic 4P** from the Fit Curve red triangle menu.

Figure 12.4 Logistic 4P Report

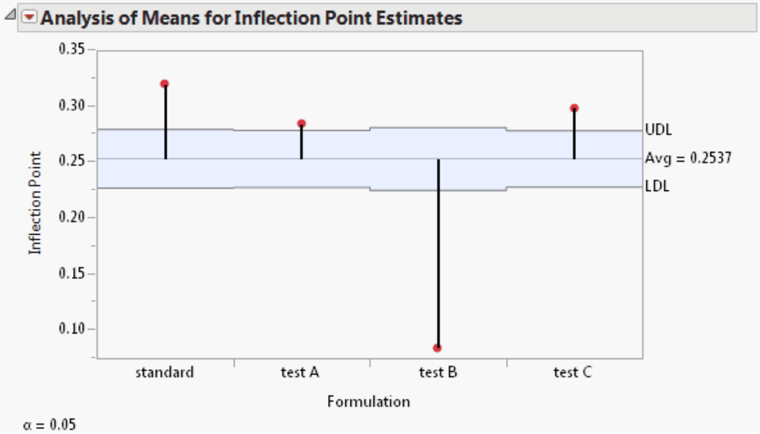


The Logistic 4P report appears (Figure 12.4). There is also a separate plot for each drug formulation. The plot of the fitted curves suggests that formulation B might be different, because the test B curve starts to rise sooner than the others. Inflection point parameters cause this rise.

9. Select **Compare Parameter Estimates** from the Logistic 4P red triangle menu.

   A portion of the Parameter Comparison report is shown in Figure 12.5.

Figure 12.5 Parameter Comparison Report

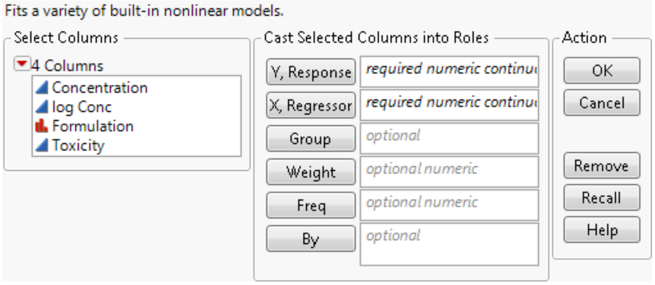


Notice that the Inflection Point parameter for the test B formulation is significantly lower than the average inflection point. This agrees with the plots shown in Figure 12.4. Drug formulation B has a lower toxicity ratio than the other formulations.

Launch the Fit Curve Platform

To launch the Fit Curve platform, select **Analyze > Specialized Modeling > Fit Curve**. The launch window is shown in Figure 12.6.

Figure 12.6 Fit Curve Platform Launch Window



Fits a variety of built-in nonlinear models.

The Fit Curve platform launch window has the following features:

Y, Response Select the *Y* variable.

X, Regressor Select the *X* variable.

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 that contains the weights of the observations.

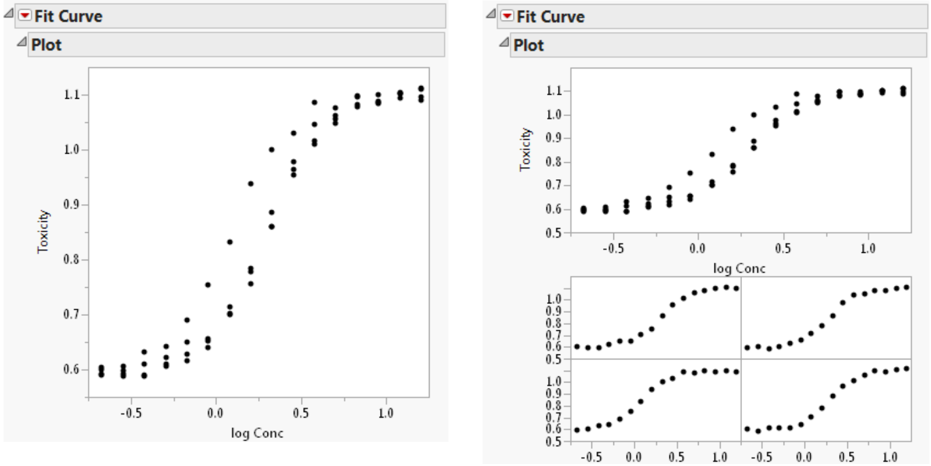
Freq Specify a variable that contains the frequencies of the observations.

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

The Fit Curve Report

The Fit Curve report initially contains only a plot of Y versus X (Figure 12.7). If you specify a Group variable, the report includes overlaid and individual plots for each group of the fitted model (shown in Figure 12.7 on the right).

Figure 12.7 Fit Curve Reports: No Grouping Variable (left) and with Group Variable (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, Probit, and Gompertz models. These models are S-shaped and have both upper and lower asymptotes. The Logistic 2P, 3P, and 4P and Probit 2P 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. Examples of Sigmoid curves include learning curves and modeling tumor growth, both of which increase initially and then taper off.

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. Examples of exponential growth and decay functions are virus spread and drug half-life, respectively.

Peak Models Fits Gaussian Peak and Lorentzian Peak models. These models increase up to a peak and 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 distribution, a continuous probability distribution. These models can be used for some chemical concentration assays and artificial neural networks.

Pharmacokinetic Models Fits the One Compartment Oral Dose model, the Two Compartment IV Bolus Dose model, and the Biexponential 4P model. This option is used to model the concentration of drugs in the body.

Fit Michaelis-Menten Fits the Michaelis-Menten biochemical kinetics model, which relates the rate of enzymatic reactions to substrate concentration.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

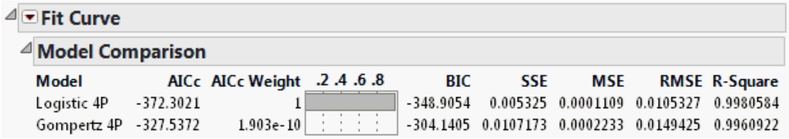
Initial Fit Curve Reports

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

To create the report shown in Figure 12.8, select **Sigmoid Curves > Logistic Curves > Fit Logistic 4P** and **Sigmoid Curves > Fit Gompertz 4P** from the Fit Curve red triangle menu.

Figure 12.8 Model Comparison Report



Fit Curve

Model Comparison

Model	AICc	AICc Weight	.2 .4 .6 .8	BIC	SSE	MSE	RMSE	R-Square
Logistic 4P	-372.3021	1	▇▇▇▇▇	-348.9054	0.005325	0.0001109	0.0105327	0.9980584
Gompertz 4P	-327.5372	1.903e-10		-304.1405	0.0107173	0.0002233	0.0149425	0.9960922

The Model Comparison report shows fit statistics used for comparing multiple models. The statistics are AICc, AICc Weight, BIC, SSE, MSE, RMSE, and R-Square, and are defined below.

AICc Gives a measure of the goodness of fit of an estimated statistical model that can be used to compare two or more models. AICc is a modification of the AIC adjusted for small samples. AICc can only be computed when the number of data points is at least two greater than the number of parameters. The model with the lowest AICc value is the best, which is the Logistic 4P in our example. See the Statistical Details appendix in the *Fitting Linear Models* book.

AICc Weight Gives normalized AICc values that sum to one. The AICc weight can be interpreted as the probability that a particular model is the true model given that one of the fitted models is the truth. Therefore, the model with the AICc weight closest to one is the better fit. In our example, the Logistic 4P model is clearly the better fit. The AICc weights are calculated using only nonmissing AICc values, as follows:

AICcWeight = exp[-0.5(AICc-min(AICc))] / sum(exp[-0.5(AICc-min(AICc))])

where min(AICc) is the smallest AICc value among the fitted models. The AICc Weight column is then sorted in decreasing order.

BIC Gives a measure based on the likelihood function of model fit that is helpful when comparing different models. The model with the lower BIC value is the better fit. See the Statistical Details appendix in the *Fitting Linear Models* book.

SSE The sum of the squared differences between each observation and its predicted value.

MSE Gives the average of the squares of the errors of each value.

RMSE The square root of the MSE that estimates the standard deviation of the random error.

R-Square Estimates the proportion of variation in the response that can be attributed to the model rather than to random error. The model with the R-Square value closest to one is the better fit.

The Model Comparison platform provides additional options, such as plotting residual and actual values. See the “Model Comparison” chapter on page 157 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.

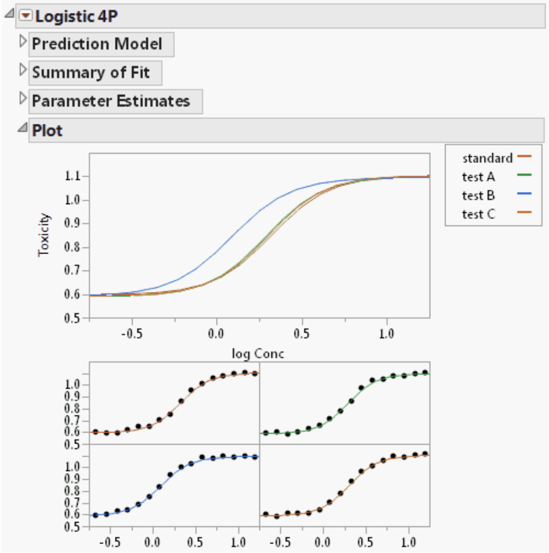
Prediction Model Gives the algebraic form of the prediction formula and the parameters.

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 12.9. The plots are shown only when you select a Grouping variable on the platform launch window.

Figure 12.9 Initial Model Reports for Logistic 4P Model



Each model report contains a red triangle menu with some or all of the following options:

Test Parallelism Helps determine whether the curves are similar in shape when they are shifted along the x-axis. In certain situations, it is important to establish parallelism before making further comparisons between groups. This option is available only when a Group variable is specified on the platform launch window. This option is available for the Sigmoid models (Logistic and Gompertz), as well as the Linear Regression model, with the exception of higher-order polynomials. For details, see “Test Parallelism” on page 191.

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 193.

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 194.

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.

Plot Actual by Predicted Plots actual Y values on the vertical axis and predicted Y values on the horizontal axis.

Plot Residual by Predicted Plots the residuals on the vertical axis and the predicted Y values on the horizontal axis.

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 a variety of 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 in parametric form. This is helpful if you want to use the fitted model in the full 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 the Standard Least Squares chapter in the *Fitting Linear Models* book.

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 12.1 provides the formulas for the models on the Fit Curve red triangle menu.

Table 12.1 Fit Curve Model Formulas

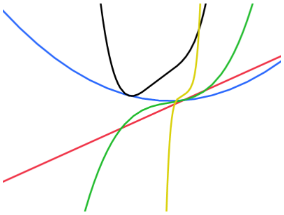
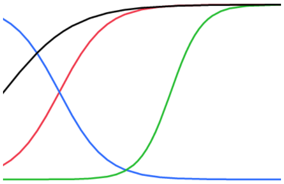
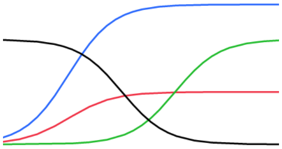
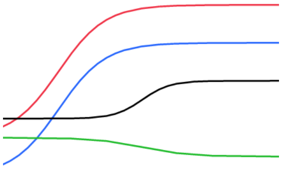
Model	Formula
Polynomials 	$\beta_0 + \sum_{i=1}^{k} \beta_i x^i$ where *k* is the order of the polynomial. These models can also be fit using the Fit Model and Fit Y by X platforms.
Logistic 2P 	$\frac{1}{1 + \text{Exp}(-a(x-b))}$ a = Growth Rate b = Inflection Point Available only when all response values are between zero and one.
Logistic 3P 	$\frac{c}{1 + \text{Exp}(-a(x-b))}$ a = Growth Rate b = Inflection Point c = Asymptote
Logistic 4P 	$c + \frac{d-c}{1 + \text{Exp}(-a(x-b))}$ a = Growth Rate b = Inflection Point c = Lower Asymptote d = Upper Asymptote

Table 12.1 Fit Curve Model Formulas *(Continued)*

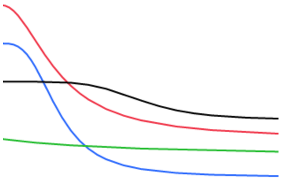
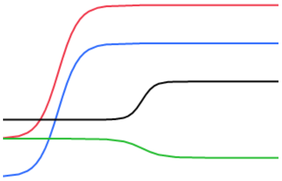
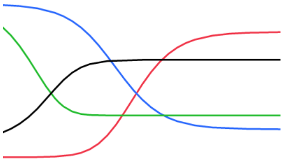
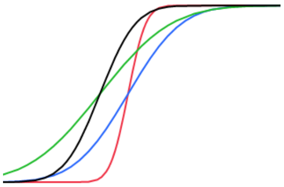
Model	Formula
Logistic 4P Rodbard 	$c + \frac{d-c}{1+(x/b)^{a}}$ a = Growth Rate b = Inflection Point c = Lower Asymptote d = Upper Asymptote Available only when the regressor values are positive.
Logistic 4P Hill 	$c + \frac{d-c}{1+10^{(-a(x-b))}}$ a = Growth Rate b = Inflection Point c = Lower Asymptote d = Upper Asymptote
Logistic 5P 	$c + \frac{d-c}{(1+\mathrm{Exp}(-a(x-b)))^{f}}$ a = Growth Rate b = Inflection Point c = Asymptote 1 d = Asymptote 2 f = Power
Probit 2P 	$\Phi\left(\frac{x-b}{a}\right)$ a = Growth Rate b = Inflection Point Φ = Normal Distribution CDF Available only when all response values are between zero and one.

Table 12.1 Fit Curve Model Formulas *(Continued)*

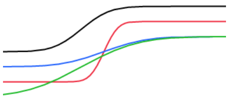
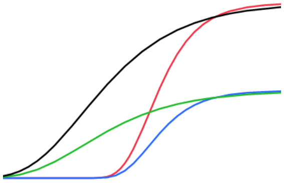
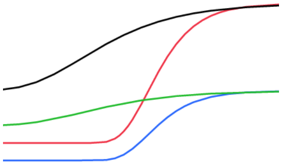
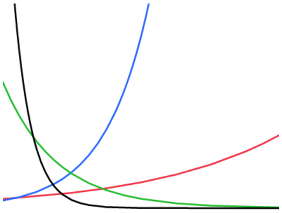
Model	Formula
Probit 4P 	$c + (d - c) \cdot \Phi\left(\frac{x - b}{a}\right)$ a = Growth Rate b = Inflection Point c = Asymptote 1 d = Asymptote 2 Φ = Normal Distribution CDF
Gompertz 3P 	$a\text{Exp}(-\text{Exp}(-b(x - c)))$ a = Asymptote b = Growth Rate c = Inflection Point
Gompertz 4P 	$a + (b - a)\text{Exp}(-\text{Exp}(-c(x - d)))$ a = Lower Asymptote b = Upper Asymptote c = Growth Rate d = Inflection Point
Exponential 2P 	$a\text{Exp}(bx)$ a = Scale b = Growth Rate

Table 12.1 Fit Curve Model Formulas *(Continued)*

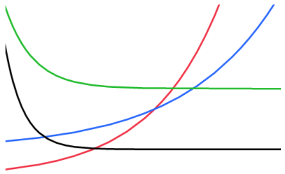
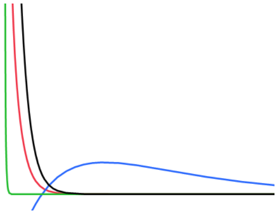
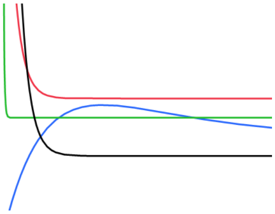
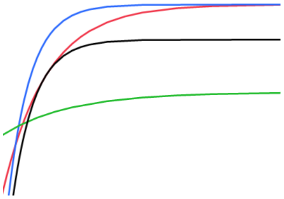
Model	Formula
Exponential 3P 	$a + b\text{Exp}(cx)$ a = Asymptote b = Scale c = Growth Rate
Biexponential 4P 	$a\text{Exp}(-bx) + c\text{Exp}(-dx)$ a = Scale 1 b = Decay Rate 1 c = Scale 2 d = Decay Rate 2 Available only when the response values are positive.
Biexponential 5P 	$a + b\text{Exp}(-cx) + d\text{Exp}(-fx)$ a = Asymptote b = Scale 1 c = Decay Rate 1 d = Scale 2 f = Decay Rate 2
Mechanistic Growth 	$a(1 - b\text{Exp}(-cx))$ a = Asymptote b = Scale c = Growth Rate

Table 12.1 Fit Curve Model Formulas *(Continued)*

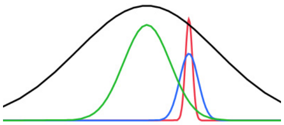
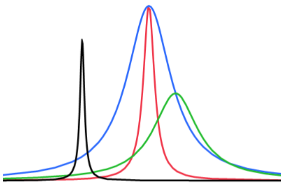
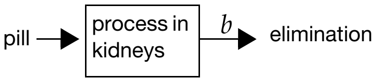
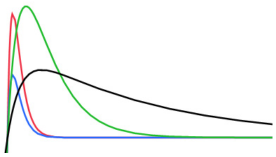
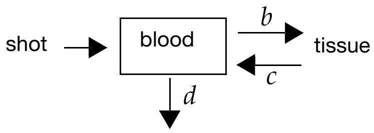
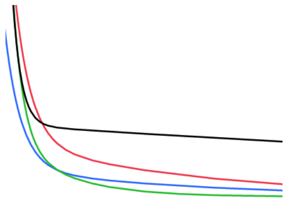
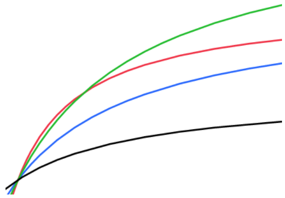
Model	Formula
Gaussian Peak 	$a\mathrm{Exp}\left(-\frac{1}{2}\left(\frac{x-b}{c}\right)^2\right)$ a = Peak Value b = Critical Point c = Growth Rate
Lorentzian Peak 	$\frac{ab^2}{(x-c)^2+b^2}$ a = Peak Value b = Growth Rate c = Critical Point
One Compartment Oral Dose  	$\frac{abc}{c-b}(\mathrm{Exp}(-bx)-\mathrm{Exp}(-cx))$ a = Area Under Curve b = Elimination Rate c = Absorption Rate Available only when the response values and the regressor values are all positive.

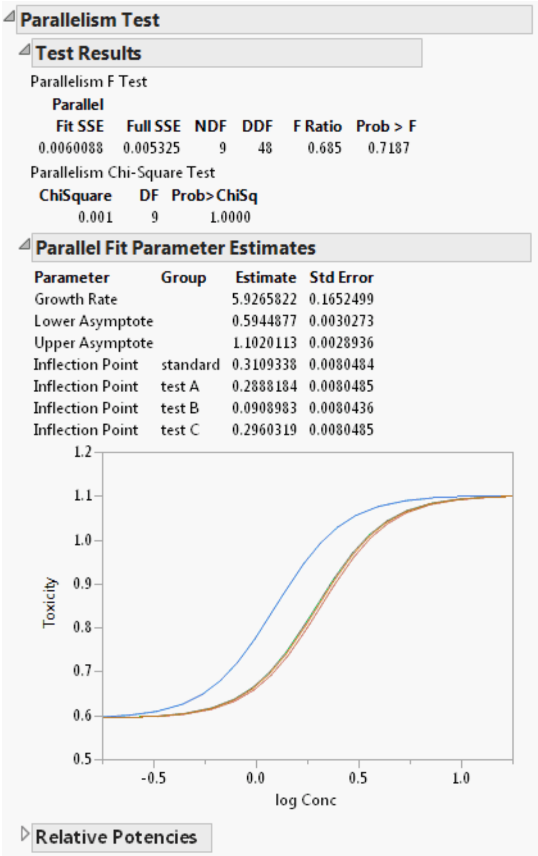
Table 12.1 Fit Curve Model Formulas *(Continued)*

Model	Formula
Two Compartment IV Bolus Dose  	$\frac{a}{\alpha-\beta}((\alpha-b)\mathrm{Exp}(-\alpha x)-(\beta-b)\mathrm{Exp}(-\beta x))$ $\alpha=\frac{1}{2}(b+c+d+\sqrt{(b+c+d)^2-4bd})$ $\beta=\frac{1}{2}(b+c+d-\sqrt{(b+c+d)^2-4bd})$ a = Initial Concentration b = Transfer Rate In c = Transfer Rate Out d = Elimination Rate Available only when the response values and the regressor values are all positive.
Michaelis-Menten 	$\frac{ax}{b+x}$ a = Max Reaction Rate b = Inverse Affinity Available only when the response values and the regressor values are all positive.

Test Parallelism

The Test Parallelism option provides an analysis for testing if the fitted models between groups have the same shape, but are shifted along the X-axis (Figure 12.10). In the Bioassay example, the curve for drug formulation B is shifted to the left of the other three curves. However, you do not know whether the curves still have the same shape (are parallel), or if formulation B is different. The Parallelism Test tells us if the shapes for the different drug formulations have similar shapes and are shifted along the horizontal axis. Select **Test Parallelism** from the fitted model’s red triangle menu to add the report.

Figure 12.10 Parallelism Test



Parallelism Test

Test Results

Parallelism F Test

Parallel Fit SSE	Full SSE	NDF	DDF	F Ratio	Prob > F
0.0060088	0.005325	9	48	0.685	0.7187

Parallelism Chi-Square Test

ChiSquare	DF	Prob>ChiSq
0.001	9	1.0000

Parallel Fit Parameter Estimates

Parameter	Group	Estimate	Std Error
Growth Rate		5.9265822	0.1652499
Lower Asymptote		0.5944877	0.0030273
Upper Asymptote		1.1020113	0.0028936
Inflection Point	standard	0.3109338	0.0080484
Inflection Point	test A	0.2888184	0.0080485
Inflection Point	test B	0.0908983	0.0080436
Inflection Point	test C	0.2960319	0.0080485

Relative Potencies

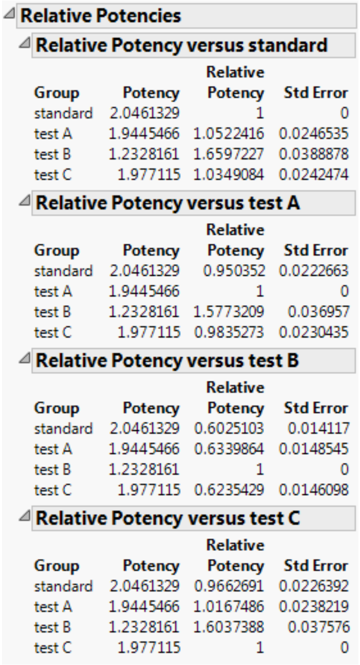
The report gives the following results:

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. In this example, the p-value is greater than 0.05, indicating that there is not enough evidence to conclude that differences exist between the curves.

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. The inflection point for drug formulation B is much lower than that of the other three drug formulations.

Relative Potencies Gives the relative potency for each level of the grouping variable. The relative potency is 10^(EC_{50}), where EC_{50} is the concentration at which the response half way between baseline and maximum is obtained. For the Logistic 2P, 3P, and 4P, the relative potency is 10^(inflection point parameter).

Figure 12.11 Relative Potencies by group



Relative Potencies

Relative Potency versus standard

		Relative	
Group	Potency	Potency	Std Error
standard	2.0461329	1	0
test A	1.9445466	1.0522416	0.0246535
test B	1.2328161	1.6597227	0.0388878
test C	1.977115	1.0349084	0.0242474

Relative Potency versus test A

		Relative	
Group	Potency	Potency	Std Error
standard	2.0461329	0.950352	0.0222663
test A	1.9445466	1	0
test B	1.2328161	1.5773209	0.036957
test C	1.977115	0.9835273	0.0230435

Relative Potency versus test B

		Relative	
Group	Potency	Potency	Std Error
standard	2.0461329	0.6025103	0.014117
test A	1.9445466	0.6339864	0.0148545
test B	1.2328161	1	0
test C	1.977115	0.6235429	0.0146098

Relative Potency versus test C

		Relative	
Group	Potency	Potency	Std Error
standard	2.0461329	0.9662691	0.0226392
test A	1.9445466	1.0167486	0.0238219
test B	1.2328161	1.6037388	0.037576
test C	1.977115	1	0

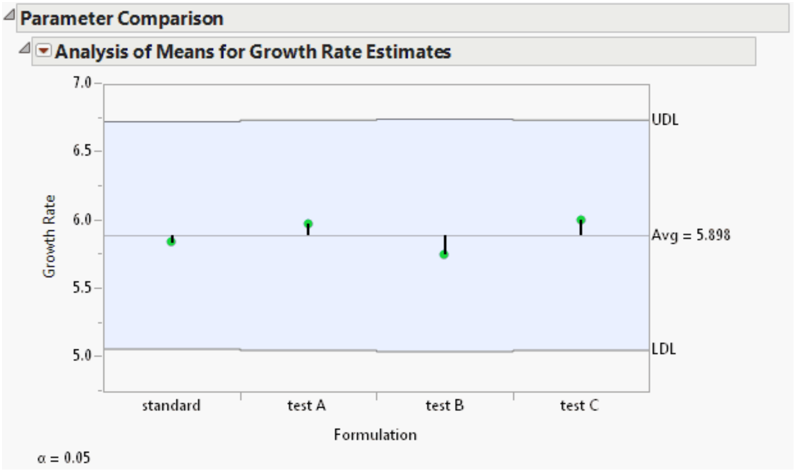
In the **Relative Potency versus standard** panel from Figure 12.11, note that the relative potencies for drug formulations A and C are nearly one. This indicates that their potencies are similar to that of the standard formulation. The potency for drug formulation B is lower than that of the standard. This means that drug formulation B increases in toxicity as a function of concentration faster than the standard.

In the parallelism test, the curves are parallel, which enables you to calculate relative potencies. Based on the relative potencies, you conclude that formulation B is more potent than the other drug formulations. Taken with the prior findings, drug formulation B appears to be more toxic.

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 12.12 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 12.12 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 about the Analysis of Means report, see the Oneway chapter in the *Basic Analysis* book.

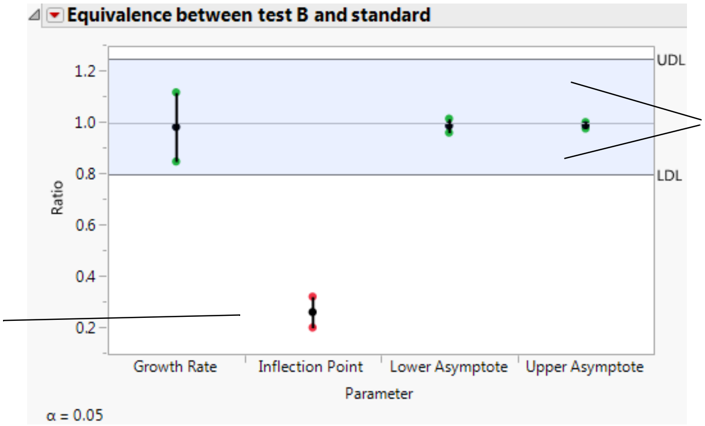
Equivalence Test

The Equivalence Test report gives an analysis for testing the equivalence of models across levels of the grouping variable (Figure 12.13). 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 12.13), then you cannot conclude that the groups are equal. The inflection point for drug formulation B is lower than the standard, which agrees with the previous findings.

Figure 12.13 Equivalence Test



decision lines

The inflection point is outside the decision limits, so you cannot conclude that the groups are equal.

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**.

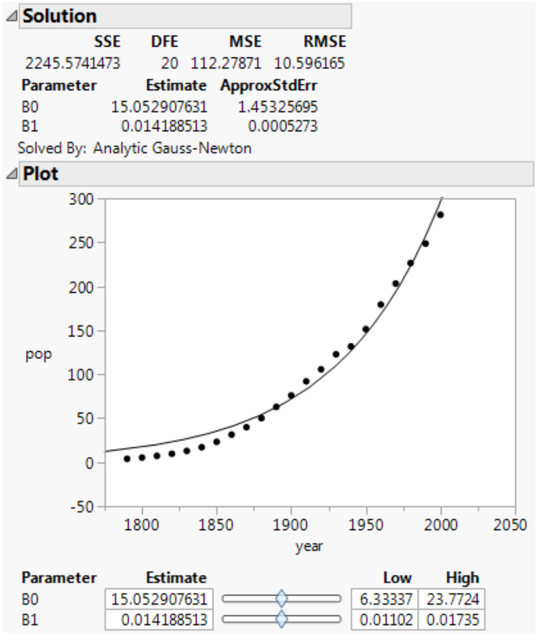
Chapter 13

Nonlinear Regression

Fit Custom Nonlinear Models to Your Data

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.

Figure 13.1 Example of a Custom Nonlinear Fit



Solution

SSE	DFE	MSE	RMSE
2245.5741473	20	112.27871	10.596165

Parameter	Estimate	ApproxStdErr
B0	15.052907631	1.45325695
B1	0.014188513	0.0005273

Solved By: Analytic Gauss-Newton

Plot

Parameter	Estimate	Low	High
B0	15.052907631	6.33337	23.7724
B1	0.014188513	0.01102	0.01735

The Nonlinear platform also provides predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models. See the "Fit Curve" chapter on page 175 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 Model Specification chapter in the *Fitting Linear Models* book and the Introduction to Fit Y by X chapter in the *Basic Analysis* book.

Example of Fitting a Custom Model

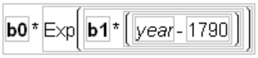
To fit a custom model, you must first create a model column with initial parameter estimates. This method does require a few more steps than fitting a built-in model, but it does allow any nonlinear model to be fit. Also, you can provide a custom loss function, and specify several other options for the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

1. Select **Help > Sample Data Library** and open Nonlinear Examples/US Population.jmp.
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**.
9. Select **New Parameter**.
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 13.2 shows the completed model.

Figure 13.2 Completed Model Formula

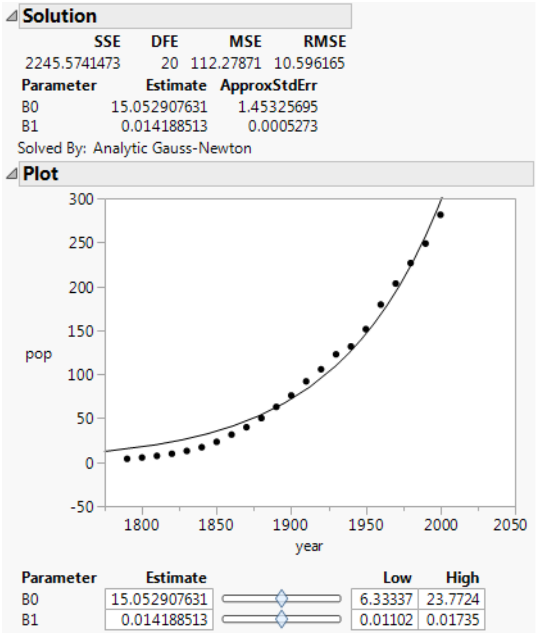


b0 * Exp(**b1** * (*year* - 1790))

13. Click **OK**.
14. Select **Analyze > Specialized Modeling > Nonlinear**.
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 13.3.

Figure 13.3 Plot and Solution Report



Solution

SSE	DFE	MSE	RMSE
2245.5741473	20	112.27871	10.596165

Parameter	Estimate	ApproxStdErr
B0	15.052907631	1.45325695
B1	0.014188513	0.0005273

Solved By: Analytic Gauss-Newton

Plot

Parameter	Estimate	Low	High
B0	15.052907631	6.33337	23.7724
B1	0.014188513	0.01102	0.01735

The final parameter estimates are shown in the **Solution** report, along with other fit statistics. The fitted model is shown on the plot.

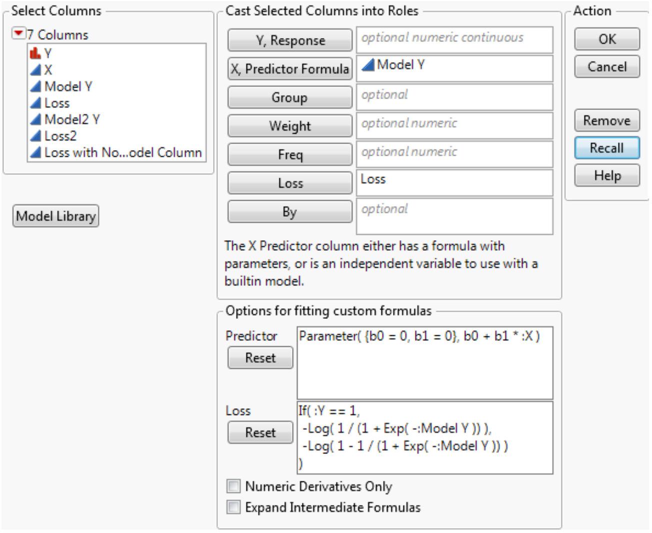
Parameters for Models with a Grouping Variable

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 > Specialized Modeling > Nonlinear**. The launch window is shown in Figure 13.4.

Figure 13.4 Nonlinear Platform Launch Window



Y, Response	optional numeric continuous
X, Predictor Formula	Model Y
Group	optional
Weight	optional numeric
Freq	optional numeric
Loss	Loss
By	optional

The X Predictor column either has a formula with parameters, or is an independent variable to use with a builtin model.

Predictor

```
Parameter( {b0 = 0, b1 = 0}, b0 + b1 * :X )
```

Loss

```
If( :Y == 1,
 -Log( 1 / (1 + Exp( -:Model Y )) ),
 -Log( 1 - 1 / (1 + Exp( -:Model Y )) )
)
```

□ Numeric Derivatives Only

□ Expand Intermediate Formulas

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 208.

Numeric Derivatives Only Uses only numeric derivatives only. This option is useful when you have a model for which it is too messy to take analytic derivatives. It can also be valuable in obtaining convergence in tough cases. This option is used only when a formula column is provided in the X, Predictor Formula role.

Expand Intermediate Formulas Tells JMP that if an ingredient column to the model 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. 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 13.5.

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, resets the iteration values, and calculates the SSE at these new values.

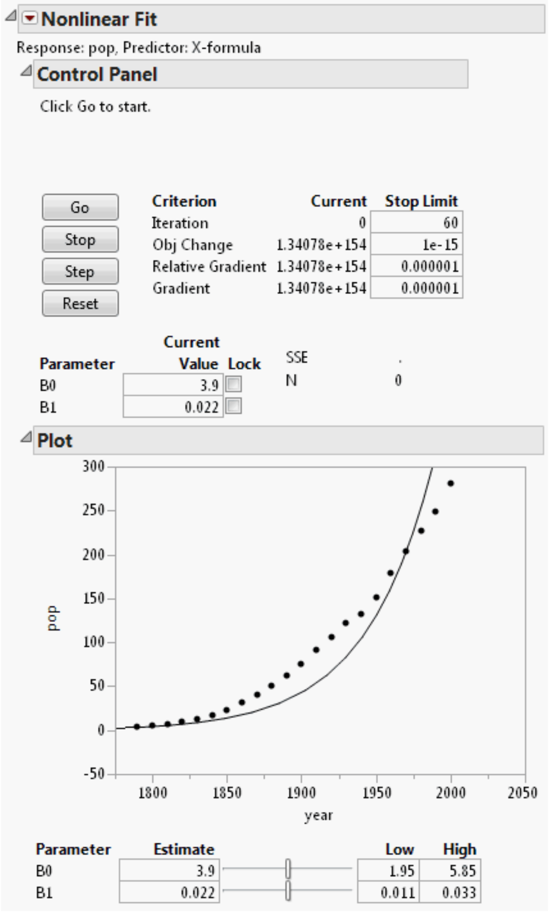
Criterion Shows iteration measures from the fitting process.

Current Shows the current value of each Criterion.

Stop Limit Sets limits on the measures listed under 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.

Figure 13.5 Initial Nonlinear Fit Report



Response: pop, Predictor: X-formula

Click Go to start.

Criterion	Current	Stop Limit
Iteration	0	60
Obj Change	1.34078e+154	1e-15
Relative Gradient	1.34078e+154	0.000001
Gradient	1.34078e+154	0.000001

Parameter	Current Value	Lock
B0	3.9	□
B1	0.022	□

SSE	.
N	0

Parameter	Estimate	Low	High
B0	3.9	1.95	5.85
B1	0.022	0.011	0.033

After you click **Go** to fit a model, the report includes the following additional items, shown in Figure 13.6.

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 219.

Solution Shows the parameters estimates and other statistics.

SSE Shows the residual sum of squared errors. 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 Shows 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.

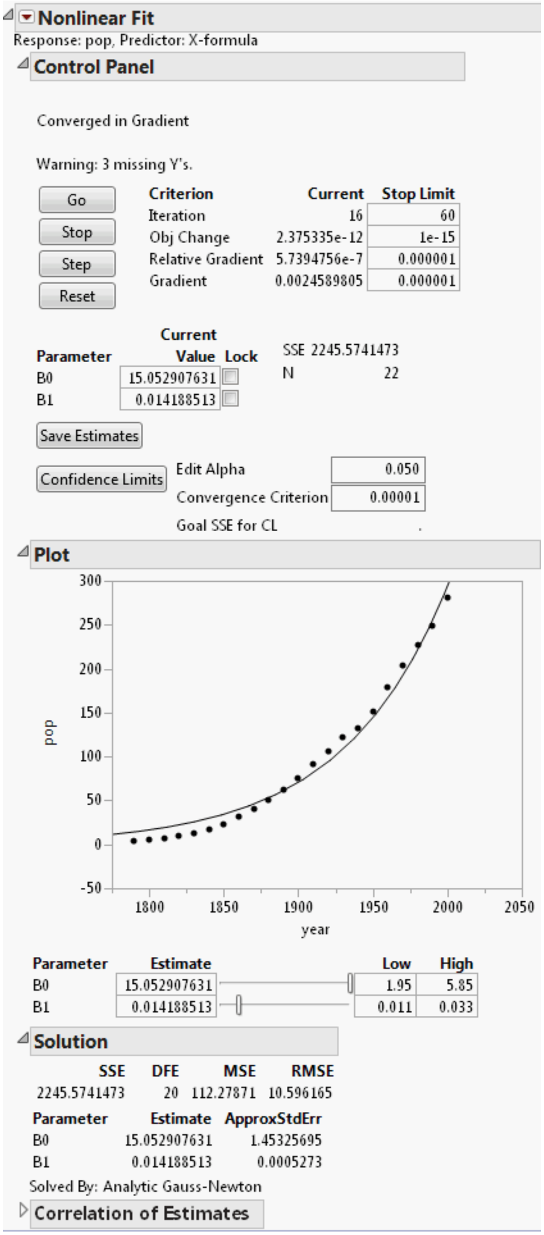
ApproxStdErr Lists the approximate standard error, which is computed analogously to linear regression. It is formed by the product of the RMSE and the square root of the diagonals of the derivative cross-products matrix inverse.

Lower CL and Upper CL Shows 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 219.

Excluded Data Shows 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.

Figure 13.6 Fitted Model Report



Nonlinear Fit

Response: pop, Predictor: X-formula

Control Panel

Converged in Gradient

Warning: 3 missing Y's.

Criterion	Current	Stop Limit
Iteration	16	60
Obj Change	2.375335e-12	1e-15
Relative Gradient	5.7394756e-7	0.000001
Gradient	0.0024589805	0.000001

Parameter	Current Value	Lock
B0	15.052907631	□
B1	0.014188513	□

SSE 2245.5741473
N 22

Edit Alpha 0.050
Convergence Criterion 0.00001
Goal SSE for CL .

Plot

Parameter	Estimate	Low	High
B0	15.052907631	1.95	5.85
B1	0.014188513	0.011	0.033

Solution

SSE	DFE	MSE	RMSE
2245.5741473	20	112.27871	10.596165

Parameter	Estimate	ApproxStdErr
B0	15.052907631	1.45325695
B1	0.014188513	0.0005273

Solved By: Analytic Gauss-Newton

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 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 Specifies whether Gauss-Newton (for regular least squares) or Newton-Raphson (for models with loss functions) is the optimization method.

QuasiNewton SR1 Specifies QuasiNewton SR1 as the optimization method.

QuasiNewton BFGS Specifies 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 221, for technical information about derivatives.

Unthreaded Runs the iterations in the main computational thread. In most cases, JMP does the computations in a separate computational thread. This improves the responsiveness of JMP while doing other things during the nonlinear calculations. However, there are some isolated cases (models that have side effects that call display routines, for example) that should be run in the main thread, so this option should be turned on.

Profilers Provides various profilers for viewing response surfaces.

Profiler Shows 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 Shows 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 Shows the Prediction Profiler and profiles the SSE or loss as a function of the parameters.

Parameter Contour Profiler Shows 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(VecQuadratic(matrix1,vector1))`. `matrix1` is the covariance matrix associated with the parameter estimates, and `vector1` is a composition of the partial derivatives of the model with respect to each parameter.

Save Std Error of Individual 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(VecQuadratic(matrix1,vector1)+mse)`. `matrix1` is the covariance matrix associated with the parameter estimates, `vector1` is a composition of the partial derivatives of the model with respect to each parameter, and `mse` is the estimate of error variance.

Save Residual Formula 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 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 works only 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.

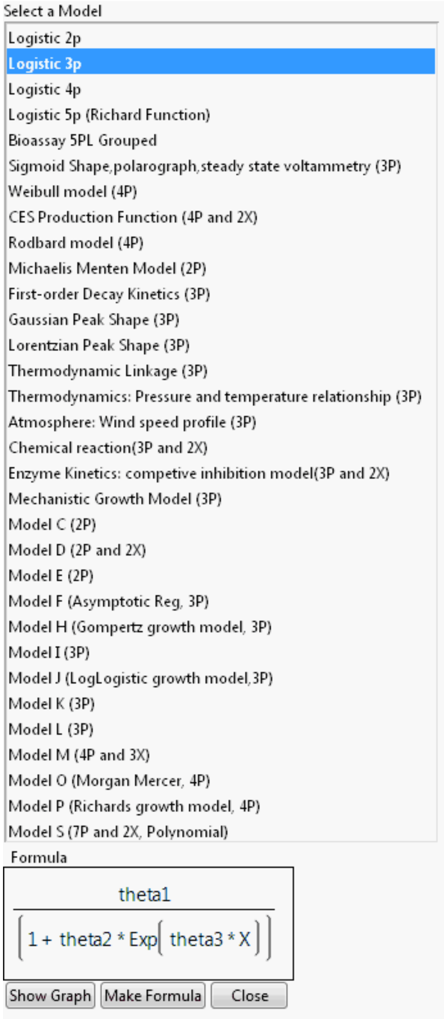
Note: The standard errors, confidence intervals, and hypothesis tests are correct only if least squares estimation is done, or if maximum likelihood estimation is used with a proper negative log-likelihood.

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 13.7).

Figure 13.7 Nonlinear Model Library Dialog



Select a Model

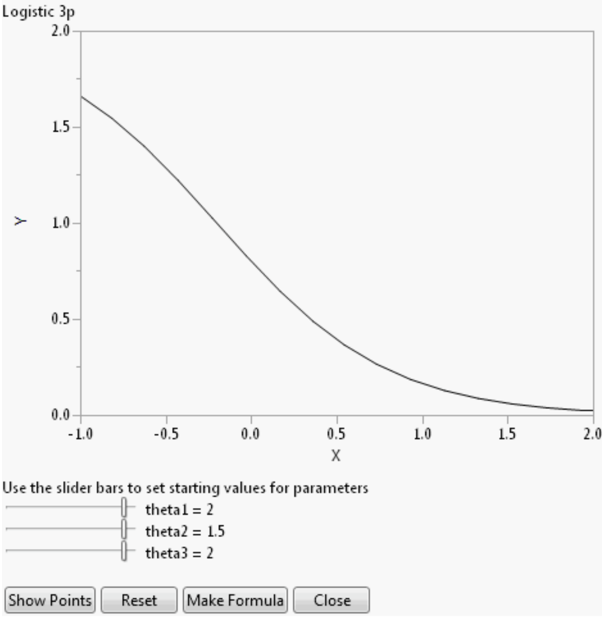
- Logistic 2p
- Logistic 3p
- Logistic 4p
- Logistic 5p (Richard Function)
- Bioassay 5PL Grouped
- Sigmoid Shape,polarograph,steady state voltammetry (3P)
- Weibull model (4P)
- CES Production Function (4P and 2X)
- Rodbard model (4P)
- Michaelis Menten Model (2P)
- First-order Decay Kinetics (3P)
- Gaussian Peak Shape (3P)
- Lorentzian Peak Shape (3P)
- Thermodynamic Linkage (3P)
- Thermodynamics: Pressure and temperature relationship (3P)
- Atmosphere: Wind speed profile (3P)
- Chemical reaction(3P and 2X)
- Enzyme Kinetics: competive inhibition model(3P and 2X)
- Mechanistic Growth Model (3P)
- Model C (2P)
- Model D (2P and 2X)
- Model E (2P)
- Model F (Asymptotic Reg, 3P)
- Model H (Gompertz growth model, 3P)
- Model I (3P)
- Model J (LogLogistic growth model,3P)
- Model K (3P)
- Model L (3P)
- Model M (4P and 3X)
- Model O (Morgan Mercer, 4P)
- Model P (Richards growth model, 4P)
- Model S (7P and 2X, Polynomial)

Formula

$$\frac{\text{theta1}}{\left(1+\text{theta2}*\text{Exp}\left(\text{theta3}*X\right)\right)}$$

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 entering values in directly. See Figure 13.8.

Figure 13.8 Example Graph in Model Library



Use the slider bars to set starting values for parameters

theta1 = 2

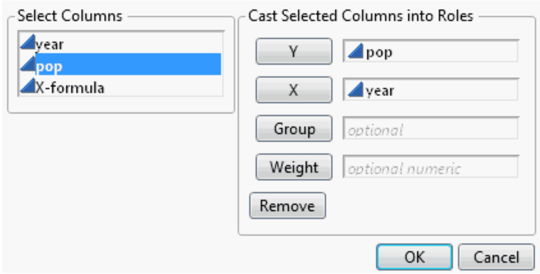
theta2 = 1.5

theta3 = 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 13.9 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 13.9 Select Roles



Y pop

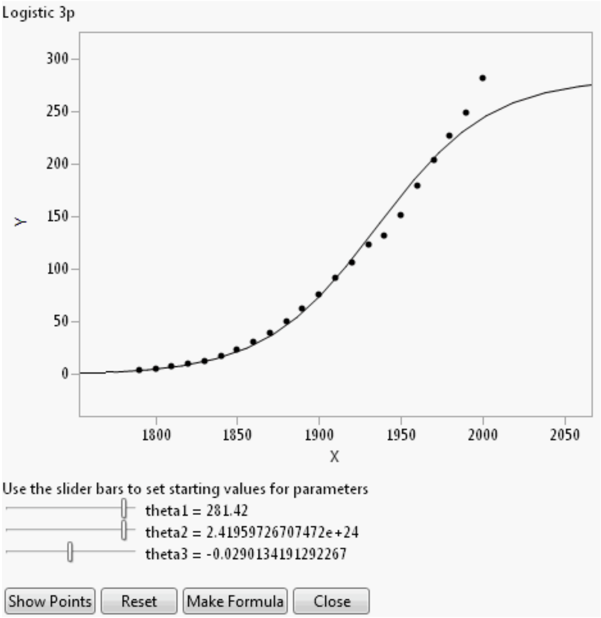
X year

Group optional

Weight optional numeric

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 13.10.

Figure 13.10 Show Points



Use the slider bars to set starting values for parameters

theta1 = 281.42

theta2 = 2.41959726707472e+24

theta3 = -0.0290134191292267

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 13.9. After that, you are brought back to the plot so that 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/Builtins 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 `Listofmodellist#`. These three lines are actually a list themselves, which consists of the following three parts.

- Model name, a quoted string
- Model formula, an expression

- Model scale

For example, suppose you want to add a model called “Simple Exponential Growth” that has the form

$$y = b_1 e^{kx}$$

Add the following lines to the NonlinLib.jsl script

```
{//Simple Exponential Growth
     "Simple Exponential Growth",
     Expr(Parameter({b1=2, k=0.5}, b1*exp(k * :X))),
     lowx = -1; highx = 2; lowy = 0;  highy = 2},
```

Some things to note:

- The first line is simply an open bracket (starting the list) and an optional comment. The second line is the string that is displayed in the model library window.
- The values of `lowx`, `highx`, `logy`, 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

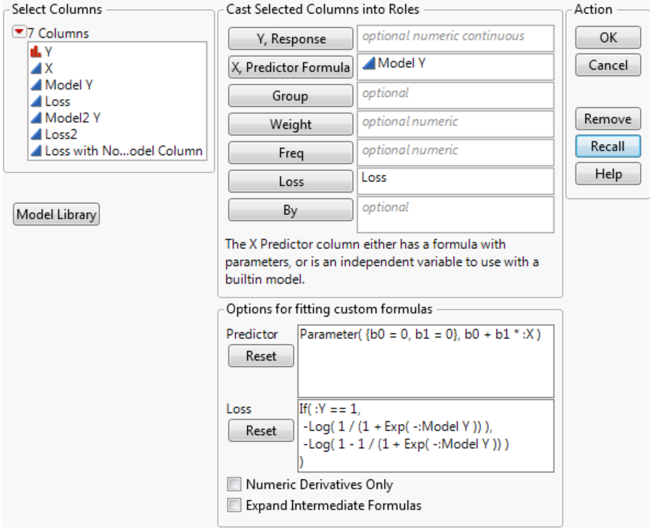
This example shows how to use the Nonlinear platform to minimize a loss function. The loss function is the negative of a log-likelihood function, thus producing maximum likelihood estimates.

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 contains ones for events and zeros for non-events. 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. Select **Help > Sample Data Library** and open Nonlinear Examples/Logistic w Loss.jmp.
2. Select **Analyze > Specialized Modeling > Nonlinear**.
3. Assign Model Y to the **X, Predictor Formula** role.
4. Assign Loss to the **Loss** role.

Figure 13.11 Nonlinear Launch Window



X, Predictor Formula: Model Y

Loss: Loss

The X Predictor column either has a formula with parameters, or is an independent variable to use with a builtin model.

Predictor: Parameter({b0 = 0, b1 = 0}, b0 + b1 * :X)

Loss:

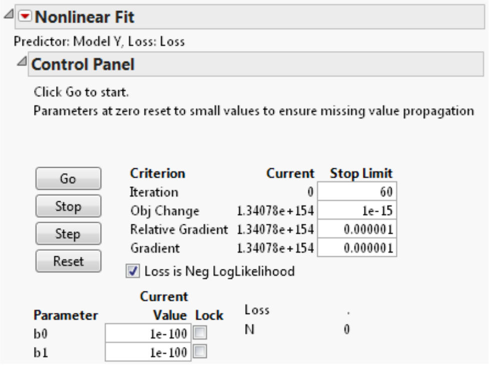
```
If( :Y == 1,
 -Log( 1 / (1 + Exp( -:Model Y )) ),
 -Log( 1 - 1 / (1 + Exp( -:Model Y )) )
)
```

□ Numeric Derivatives Only
□ Expand Intermediate Formulas

5. Click **OK**.

   The Nonlinear Fit Control Panel appears.

Figure 13.12 Nonlinear Fit Control Panel



Predictor: Model Y, Loss: Loss

Click Go to start.
Parameters at zero reset to small values to ensure missing value propagation

Criterion	Current	Stop Limit
Iteration	0	60
Obj Change	1.34078e+154	1e-15
Relative Gradient	1.34078e+154	0.000001
Gradient	1.34078e+154	0.000001

☑ Loss is Neg LogLikelihood

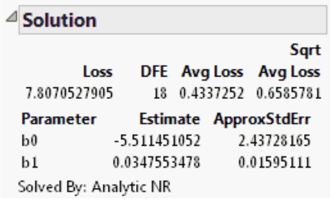
Parameter	Current Value	Lock
b0	1e-100	□
b1	1e-100	□

Loss	.
N	0

6. Click **Go**.

The parameter estimates are shown in the Solution report. See Figure 13.13.

Figure 13.13 Solution Report



Solution

Loss	DFE	Avg Loss	Sqrt Avg Loss
7.8070527905	18	0.4337252	0.6585781

Parameter	Estimate	ApproxStdErr
b0	-5.511451052	2.43728165
b1	0.0347553478	0.01595111

Solved By: Analytic NR

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

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:

```
Normal Distribution(b0+b1*Heat+b2*Soak)
```

The argument to the `Normal Distribution` function is a linear model of the treatments.

To specify binomial errors, the loss function, Loss, has the formula

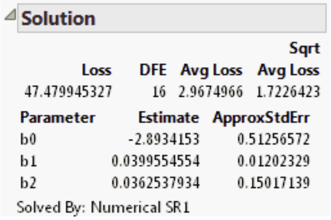
```
-(Nready*Log(p) + (Ntotal - Nready)*Log(1 - p))
```

Follow these steps to fit the model:

1. Select **Analyze > Specialized 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 13.14.

Figure 13.14 Solution for the Ingots2 Data



Solution

Loss	DFE	Avg Loss	Sqrt Avg Loss
47.479945327	16	2.9674966	1.7226423

Parameter	Estimate	ApproxStdErr
b0	-2.8934153	0.51256572
b1	0.0399554554	0.01202329
b2	0.0362537934	0.15017139

Solved By: Numerical SR1

Example of a Poisson Loss Function

A Poisson distribution is often used to model count data.

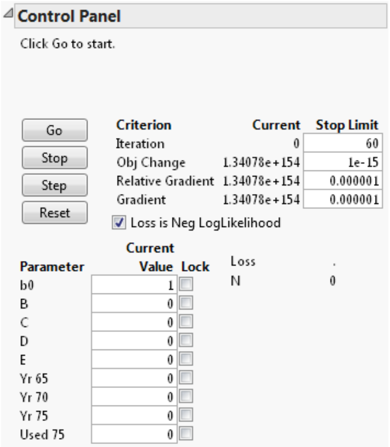
$$P(Y = n) = \frac{e^{-\mu}\mu^{n}}{n!}, n = 0, 1, 2, \ldots$$

where μ 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 negative log-likelihood) is in the Poisson column. To fit the model, follow the steps below:

1. Select **Analyze > Specialized 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 13.15).

Figure 13.15 Enter New Parameters



Control Panel

Click Go to start.

Criterion	Current	Stop Limit
Iteration	0	60
Obj Change	1.34078e+154	1e-15
Relative Gradient	1.34078e+154	0.000001
Gradient	1.34078e+154	0.000001

☑ Loss is Neg LogLikelihood

Parameter	Current Value	Lock
b0	1	□
B	0	□
C	0	□
D	0	□
E	0	□
Yr 65	0	□
Yr 70	0	□
Yr 75	0	□
Used 75	0	□

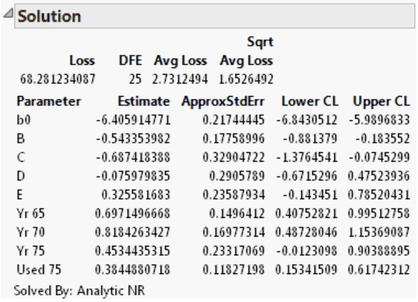
Loss	.
N	0

6. Click **Go**.

7. Click the **Confidence Limits** button.

The Solution report is shown in Figure 13.16. The results include the parameter estimates and confidence intervals, and other summary statistics.

Figure 13.16 Solution Table for the Poisson Loss Example



Solution

Loss	DFE	Avg Loss	Sqrt Avg Loss
68.281234087	25	2.7312494	1.6526492

Parameter	Estimate	ApproxStdErr	Lower CL	Upper CL
b0	-6.405914771	0.21744445	-6.8430512	-5.9896833
B	-0.543353982	0.17758996	-0.881379	-0.183552
C	-0.687418388	0.32904722	-1.3764541	-0.0745299
D	-0.075979835	0.2905789	-0.6715296	0.47523936
E	0.325581683	0.23587934	-0.143451	0.78520431
Yr 65	0.6971496668	0.1496412	0.40752821	0.99512758
Yr 70	0.8184263427	0.16977314	0.48728046	1.15369087
Yr 75	0.4534435315	0.23317069	-0.0123098	0.90388895
Used 75	0.3844880718	0.11827198	0.15341509	0.61742312

Solved By: Analytic NR

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.

Example of Setting Parameter Limits

The Fit Curve personality enables you to fit a model and then use the prediction equation in the full personality of the Nonlinear platform. This method requires more steps and user input but allows any nonlinear model to be fit.

Complete “Example Using the Fit Curve Personality” on page 176 in the “Fit Curve” chapter to fit the model. This example shows how to save the prediction formula from Fit Curve and then set parameter limits in Nonlinear.

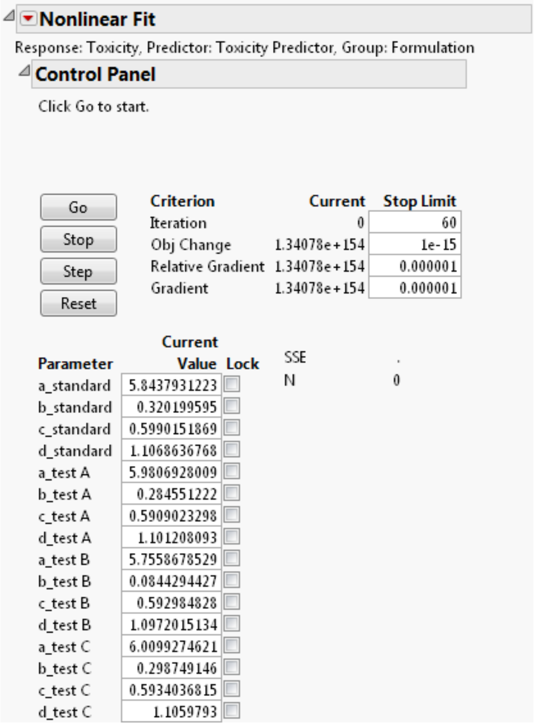
1. From the Logistic 4P red triangle menu, select **Save Formulas > Save Parametric Prediction Formula**.

   A new column named Toxicity Predictor appears in the data table.

2. Select **Analyze > Specialized Modeling > Nonlinear.**
3. Assign Toxicity to the **Y, Response** role.
4. Assign Toxicity Predictor to the **X, Predictor Formula** role.
5. Assign Formulation to the **Group** role.
6. Click **OK**.

   The Nonlinear Fit window appears (Figure 13.17). In the Control Panel, parameter values and locking options are shown. The letters listed before each parameter correspond to variables from the Prediction Model in the Fit Curve function.

Figure 13.17 Nonlinear Fit Control Panel



Response: Toxicity, Predictor: Toxicity Predictor, Group: Formulation

Click Go to start.

Criterion	Current	Stop Limit
Iteration	0	60
Obj Change	1.34078e+154	1e-15
Relative Gradient	1.34078e+154	0.000001
Gradient	1.34078e+154	0.000001

Parameter	Current Value	Lock
a_standard	5.8437931223	□
b_standard	0.320199595	□
c_standard	0.5990151869	□
d_standard	1.1068636768	□
a_test A	5.9806928009	□
b_test A	0.284551222	□
c_test A	0.5909023298	□
d_test A	1.101208093	□
a_test B	5.7558678529	□
b_test B	0.0844294427	□
c_test B	0.592984828	□
d_test B	1.0972015134	□
a_test C	6.0099274621	□
b_test C	0.298749146	□
c_test C	0.5934036815	□
d_test C	1.1059793	□

SSE	.
N	0

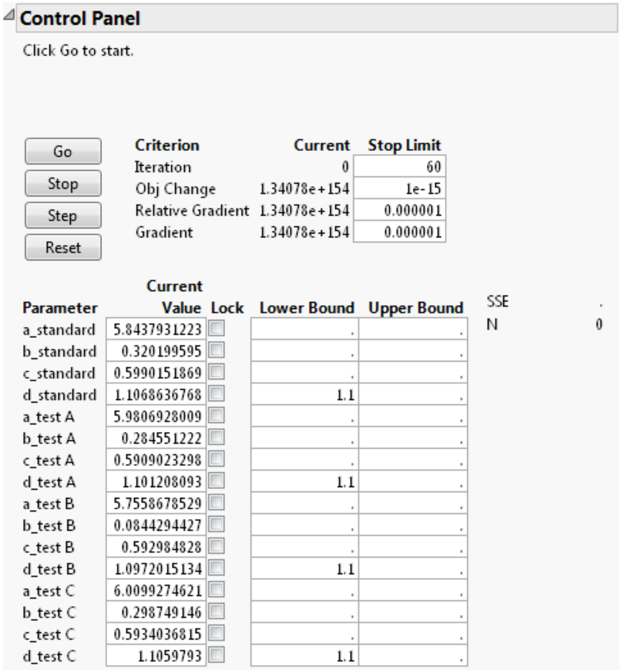
Tip: You can lock parameters if you know the values from prior information.

7. Select the red triangle menu next to **Nonlinear Fit** and then select **Parameter Bounds**.

   Options for setting the lower and upper parameters appear next to the parameters.

8. Set the lower bounds for the parameters as shown in Figure 13.18. You know from prior experience that the maximum toxicity of the drug is at least 1.1.

Figure 13.18 Setting Parameter Bounds



Click Go to start.

Criterion	Current	Stop Limit
Iteration	0	60
Obj Change	1.34078e+154	1e-15
Relative Gradient	1.34078e+154	0.000001
Gradient	1.34078e+154	0.000001

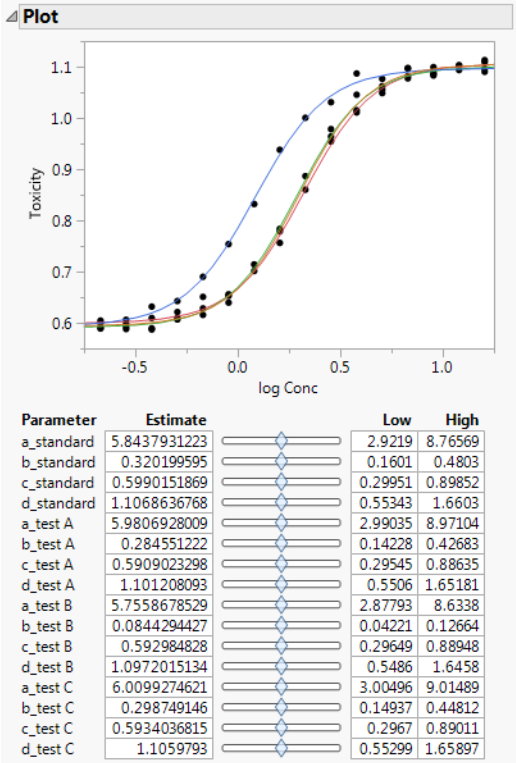
Parameter	Current Value	Lock	Lower Bound	Upper Bound
a_standard	5.8437931223	□	.	.
b_standard	0.320199595	□	.	.
c_standard	0.5990151869	□	.	.
d_standard	1.1068636768	□	1.1	.
a_test A	5.9806928009	□	.	.
b_test A	0.284551222	□	.	.
c_test A	0.5909023298	□	.	.
d_test A	1.101208093	□	1.1	.
a_test B	5.7558678529	□	.	.
b_test B	0.0844294427	□	.	.
c_test B	0.592984828	□	.	.
d_test B	1.0972015134	□	1.1	.
a_test C	6.0099274621	□	.	.
b_test C	0.298749146	□	.	.
c_test C	0.5934036815	□	.	.
d_test C	1.1059793	□	1.1	.

SSE	.
N	0

9. Click **Go**.

   The final parameter estimates are shown in the **Solution** report, along with other fit statistics (Figure 13.19). The fitted model is shown on the plot.

Figure 13.19 Nonlinear Fit Plot and Parameter Estimates



Parameter	Estimate	Low	High
a_standard	5.8437931223	2.9219	8.76569
b_standard	0.320199595	0.1601	0.4803
c_standard	0.5990151869	0.29951	0.89852
d_standard	1.1068636768	0.55343	1.6603
a_test A	5.9806928009	2.99035	8.97104
b_test A	0.284551222	0.14228	0.42683
c_test A	0.5909023298	0.29545	0.88635
d_test A	1.101208093	0.5506	1.65181
a_test B	5.7558678529	2.87793	8.6338
b_test B	0.0844294427	0.04221	0.12664
c_test B	0.592984828	0.29649	0.88948
d_test B	1.0972015134	0.5486	1.6458
a_test C	6.0099274621	3.00496	9.01489
b_test C	0.298749146	0.14937	0.44812
c_test C	0.5934036815	0.2967	0.89011
d_test C	1.1059793	0.55299	1.65897

Options below the plot allow for adjusting parameter limits and estimates (Figure 13.19).

Statistical Details

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

Profile Likelihood Confidence Limits

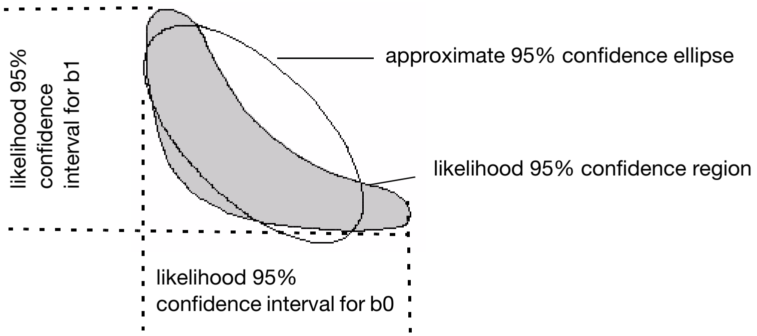
The upper and lower confidence limits for the parameters are based on a search for the value of each parameter after minimizing with respect to the other parameters. The search looks for values that produce an SSE greater by a certain amount than the solution's minimum SSE. The goal of this difference 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 13.20 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.

Figure 13.20 Diagram of Confidence Limits for Parameters


approximate 95% confidence ellipse
likelihood 95% confidence interval for b1
likelihood 95% confidence region
likelihood 95% confidence interval for b0

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
- 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 defined as follows:

$$L = \sum_{i=1}^{n} \rho(f(\beta))$$

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:

$$L = \sum_{i=1}^{n} \rho(f(\beta))$$

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

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

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

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

The second term is probably small if ρ 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.

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 for the Nonlinear column looks something like this:

```
Parameter({b0=0.5, b1=0.5}, b0*(1-Exp(-b1*X)))
```

The Parameter block in the formula is hidden if you use the formula editor. 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 `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–(Exp(-b1*X))`. The derivative with respect to **b1** is `T#3*b0`, which is `–(–1*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 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 `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.

$$\begin{aligned} a_1 &= a_2 - b_2\bar{x} + c_2\bar{x} \\ b_1 &= b_2 - 2c_2\bar{x} \\ c_1 &= c_2 \end{aligned}$$

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.

Chapter **14**

Gaussian Process

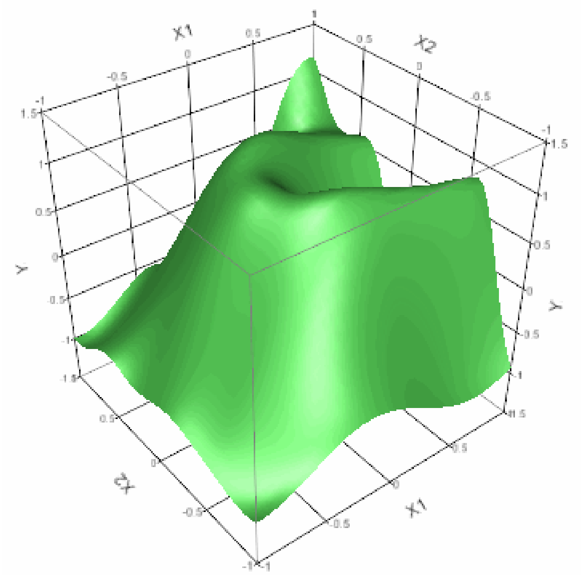
Fit Data Using Smoothing Models

Use the Gaussian Process platform to model the relationship between a continuous response and one or more predictors. These types of models are common in 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 results in the same output value.

The Gaussian Process platform fits a spatial correlation model to the data. The correlation of the response between two observations decreases as the values of the independent variables become more distant.

One purpose for using this platform is to obtain a prediction formula that can be used for further analysis and optimization.

Figure 14.1 Gaussian Process Prediction Surface Example

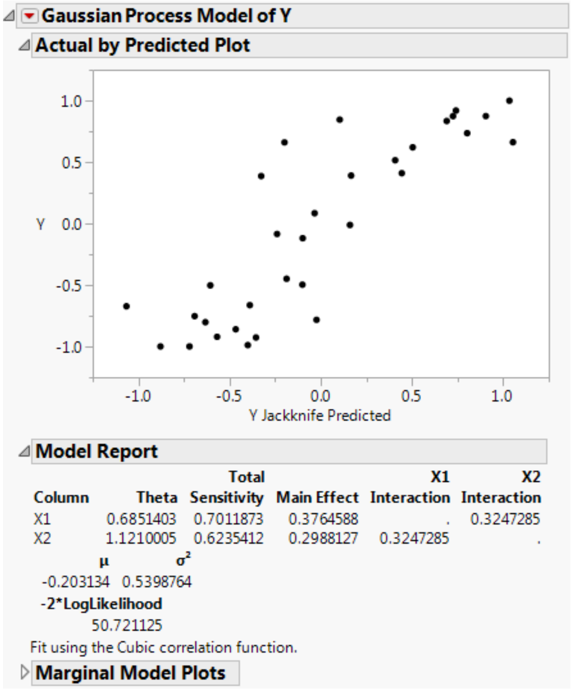


Example of Gaussian Process

This example uses data from a space filling design in two variables with a deterministic equation for Y (the response). You can use the Gaussian Process platform to find the explanatory power of X1 and X2 on Y. You can view the equation for Y in the column formula.

1. Select **Help > Sample Data Library** and open 2D Gaussian Process Example.jmp.
2. Select **Analyze > Specialized Modeling > Gaussian Process.**
3. Select X1 and X2 and click **X**.
4. Select Y and click **Y**
5. Select **Correlation Type > Cubic.**
6. JMP PRO Deselect **Fast GASP**.
7. Click **OK**.

Figure 14.2 Gaussian Process Report



Column	Theta	Total Sensitivity	Main Effect	X1 Interaction	X2 Interaction
X1	0.6851403	0.7011873	0.3764588	.	0.3247285
X2	1.1210005	0.6235412	0.2988127	0.3247285	.

μ	σ^2
-0.203134	0.5398764

-2*LogLikelihood
50.721125

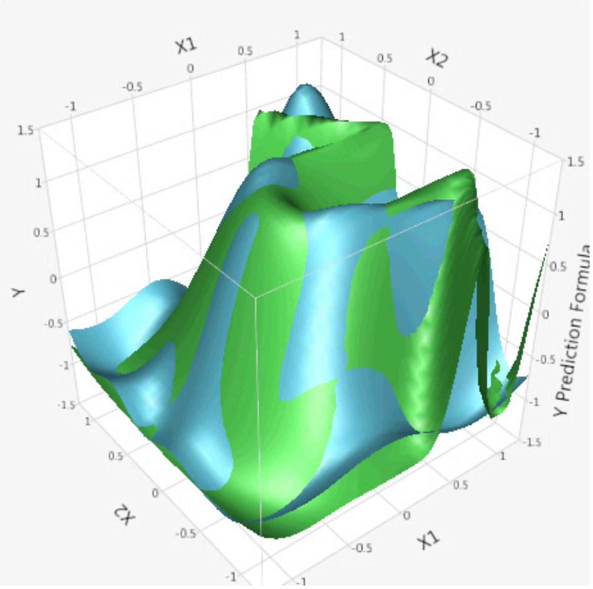
Fit using the Cubic correlation function.

Note: The estimated parameters can be different due to different starting points in the minimization routine, the choice of correlation type, and the inclusion of a nugget parameter.

Now, visualize the fitted surface compared to the original surface.

8. Click the red triangle next to Gaussian Process Model of Y and select **Save Prediction Formula**.
9. Select **Graph > Surface Plot**.
10. Select X1 through Y Prediction Formula and click **Columns**.
11. Click **OK**.
12. In the Surface column, select **Both sides** for the Y Prediction Formula.

Figure 14.3 3D Surface Plot of the Actual and Predicted Ys

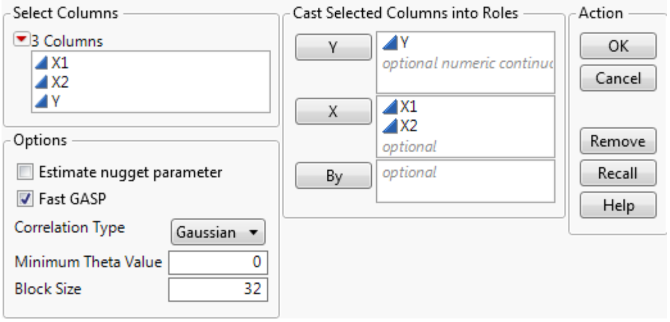


The two surfaces are similar. The impact of X1 and X2 on the response Y can be visualized. You can rotate the plot to view it from different angles. Marginal plots are another tool to use to understand the impact of the factors on the response.

Launch the Gaussian Process Platform

Launch the Gaussian Process platform by selecting **Analyze > Specialized Modeling > Gaussian Process**.

Figure 14.4 Gaussian Process Launch Window



Select Columns

3 Columns

X1
X2
Y

Cast Selected Columns into Roles

Y: Y, optional numeric continu

X: X1, X2, optional

By: optional

Options

☐ Estimate nugget parameter

☑ Fast GASP

Correlation Type: Gaussian

Minimum Theta Value: 0

Block Size: 32

Y Assigns the continuous columns to analyze.

X Assigns the columns to use as explanatory variables. Categorical variables are allowed in JMP Pro when the Fast GASP option is specified.

Estimate Nugget Parameter introduces a ridge parameter into the estimation procedure. A ridge parameter is useful if there is noise or randomness in the response, and you want the prediction model to smooth over the noise instead of perfectly interpolating.

JMP PRO **Fast GASP** Option to use the Fast GASP algorithm. Fast GASP breaks the Gaussian process model into small pieces (called blocks) to speed computation time. Blocks allow for the use of multiple CPUs and parallel processing.

Note: When there are more than 2,500 observations, the Fast GASP algorithm is required.

For additional information about Fast GASP, see Parker (2015).

Correlation Type Choose the correlation structure for 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 nonzero, no matter the distance between the points.

Cubic allows the correlation between two points to be zero for points that are far enough apart. This method is a generalization of a cubic spline.

JMP PRO The Fast GASP algorithm does not support the cubic correlation function.

Minimum Theta Value Sets the minimum theta value to use in the fitted model. The default is 0. The theta values are analogous to a slope parameter in regular regression models. Small theta values indicate that a variable has little influence on the predicted values.

JMP PRO **Block Size** Number of observations in each computational block used by the Fast GASP algorithm. There must be at least 25 observations per block and a maximum of the number of rows in the data set up to a maximum of 2,500.

The Gaussian Process Report

The initial Gaussian Process report shows the actual by predicted plot and a model report. The marginal plots for each factor are initially hidden.

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 diagonal (Y = X) of the plot.

The jackknife values are not true jackknife values in that the model is not re-fit with the associated row for each Y excluded. Rather, the row is excluded from the prediction model for each associated Y but the correlation parameters retain the contribution of the row in them. For Gaussian processes that perfectly interpolate the data this jackknife procedure provides predictions that are not equal to the input.

Model Report

The Model Report shows a functional ANOVA table for the model parameter estimates. Specifically, it is an analysis of variance table where the variation is computed using a function-driven method.

Theta Gaussian Process model parameter estimates. See "Statistical Details for the Gaussian Process Platform" on page 234.

Total Sensitivity Sum of the main effect and all interaction terms for each factor. It is a measure of the amount of influence a factor and all its two-way interactions have on the response variable.

Total variation is the integrated variability over the entire experimental space.

Main Effect The functional main effect of each factor is the integrated total variation due to that factor alone. The main effect is the ratio of the functional effect and the total variation for each factor in the model.

Interactions Functional interaction effects are computed in a similar way to main effects.

JMP PRO **Categorical Input** When the model includes categorical factors, a correlation matrix for each categorical factor is provided. The off-diagonal entries correspond to Gaussian Process model parameter estimates. See "Models with Categorical Predictors" on page 235.

Mu and Sigma^2 Mean and variance model parameters.

Nugget Estimated nugget value. A nugget value is reported if you selected **estimate nugget parameter** in the Gaussian Process launch window. A nugget value is also reported if JMP has added a nugget parameter in order to avoid a singular covariance matrix.

-2LogLikelihood Estimated value of the minimized -2log likelihood function.

Marginal Model Plots

A marginal plot appears for each factor in the model. It shows the response across the levels of a factor where all other factors are set to their average value.

Gaussian Process Platform Options

Use the options in the Gaussian Process red triangle menu to customize the report according to your individual needs.

Profiler Opens the standard Profiler. For details, see the Profiler chapter in the *Profilers* book.

Contour Profiler Opens the Contour Profiler. For details, see the Contour Profiler chapter in the *Profilers* book.

Surface Profiler Opens the Surface Profiler. For details, see the Surface Plot chapter in the *Profilers* book.

Save Prediction Formula Creates a new prediction formula column in the active data table.

Save Variance Formula Creates a new variance formula column in the active data table.

JMP PRO **Publish Prediction Formula** Creates a prediction formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the “Formula Depot” chapter on page 169.

JMP PRO **Publish Variance Formula** Creates a variance formula and saves it as a formula column script in the Formula Depot platform. If a Formula Depot report is not open, this option creates a Formula Depot report. See the “Formula Depot” chapter on page 169.

Save Jackknife Predicted Values Saves the jackknife predicted values to the active data table. These are the *x*-axis values for the Actual by Predicted Plot.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

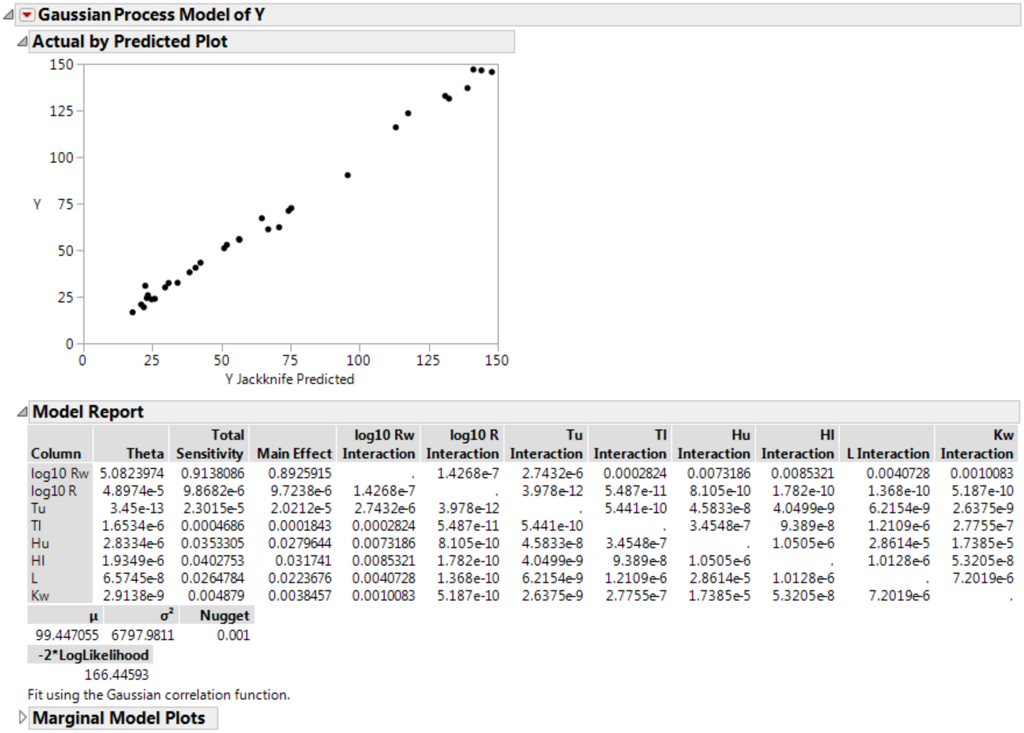
Additional Examples of the Gaussian Process Platform

Example of a Gaussian Process Model

This example uses data that demonstrates the flow of water through a Borehole that is drilled from the ground surface through two aquifers. Given a specified engineering model the Gaussian process lets us understand the impact of factors included in the model on the response, Y.

1. Select **Help > Sample Data Library** and open Design Experiment/Borehole Latin Hypercube.jmp.
2. Select **Analyze > Specialized Modeling > Gaussian Process**.
3. Select log10 Rw through Kw and click **X**.
4. Select Y and click **Y**.
5. JMP PRO In JMP Pro, to run the analysis faster, leave the Fast GASP checked.
6. Click **OK**.

Figure 14.5 Borehole Latin Hypercube Report



Gaussian Process Model of Y

Actual by Predicted Plot

Model Report

Column	Theta	Total Sensitivity	Main Effect	log10 Rw Interaction	log10 R Interaction	Tu Interaction	Tl Interaction	Hu Interaction	Hl Interaction	L Interaction	Kw Interaction
log10 Rw	5.0823974	0.9138086	0.8925915	.	1.4268e-7	2.7432e-6	0.0002824	0.0073186	0.0085321	0.0040728	0.0010083
log10 R	4.8974e-5	9.8682e-6	9.7238e-6	1.4268e-7	.	3.978e-12	5.487e-11	8.105e-10	1.782e-10	1.368e-10	5.187e-10
Tu	3.45e-13	2.3015e-5	2.0212e-5	2.7432e-6	3.978e-12	.	5.441e-10	4.5833e-8	4.0499e-9	6.2154e-9	2.6375e-9
Tl	1.6534e-6	0.0004686	0.0001843	0.0002824	5.487e-11	5.441e-10	.	3.4548e-7	9.389e-8	1.2109e-6	2.7755e-7
Hu	2.8334e-6	0.0353305	0.0279644	0.0073186	8.105e-10	4.5833e-8	3.4548e-7	.	1.0505e-6	2.8614e-5	1.7385e-5
Hl	1.9349e-6	0.0402753	0.031741	0.0085321	1.782e-10	4.0499e-9	9.389e-8	1.0505e-6	.	1.0128e-6	5.3205e-8
L	6.5745e-8	0.0264784	0.0223676	0.0040728	1.368e-10	6.2154e-9	1.2109e-6	2.8614e-5	1.0128e-6	.	7.2019e-6
Kw	2.9138e-9	0.004879	0.0038457	0.0010083	5.187e-10	2.6375e-9	2.7755e-7	1.7385e-5	5.3205e-8	7.2019e-6	.

μ	σ^2	Nugget
99.447055	6797.9811	0.001

-2*LogLikelihood
166.44593

Fit using the Gaussian correlation function.

Marginal Model Plots

The data on the actual by predicted plot fall along the Y = X line, indicating that the Gaussian process prediction model is a good approximation of the true function. In the Model Report, you see that the first factor, log10 Rw, has the highest total sensitivity. The estimated total sensitivity for log10 Rw explains more than 90% of the variation in the response. Factors with small theta values have little (or no) impact on the prediction formula.

Note: Your estimates can differ from those shown in Figure 14.5, which were found using the Fast GASP algorithm.

Example of Gaussian Process Model with Categorical Predictors

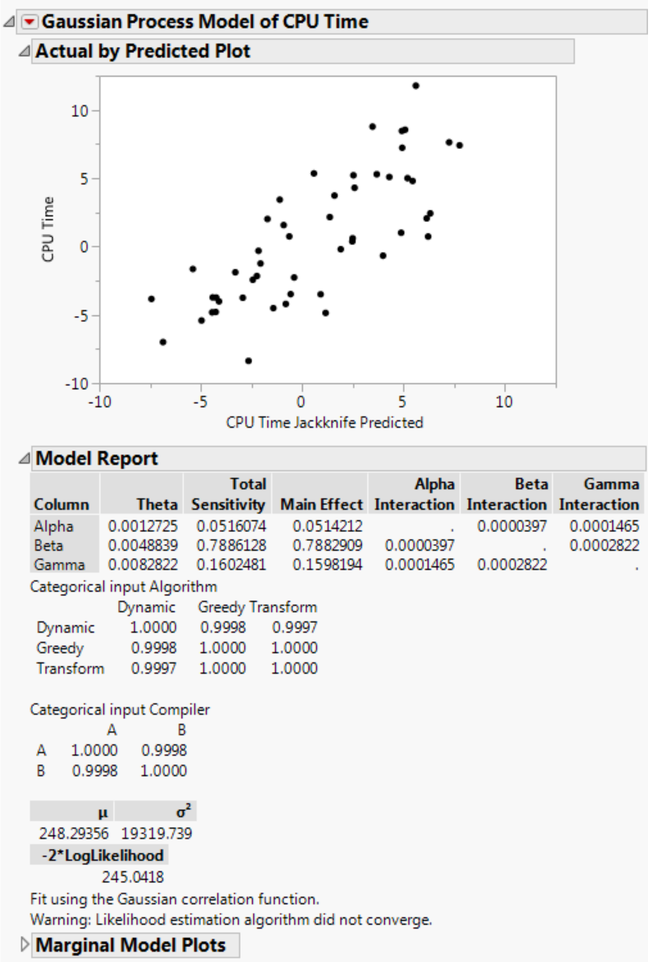
This example uses the Algorithm Data.jmp sample data table. These data are simulated CPU times from a 50 run space filling designed experiment. The Algorithm Factors.jmp sample data table provides the factors and settings for the design. The design has three continuous and two categorical factors. The goal is to predict CPU Time using a Gaussian Process model that contains both continuous and categorical factors.

1. Select **Help > Sample Data Library** and open Design Experiment/Algorithm Data.jmp.

2. Select **Analyze > Specialized Modeling > Gaussian Process**.
3. Select Alpha through Compiler and click **X**.
4. Select CPU Time and click **Y**.
5. JMP PRO To run the analysis, leave the Fast GASP checked. Click **OK**.

Note: The Fast GASP option must be used for models that contain categorical factors. See "Models with Categorical Predictors" on page 235 for details.

Figure 14.6 Algorithm Data Report



Column	Theta	Total Sensitivity	Main Effect	Alpha Interaction	Beta Interaction	Gamma Interaction
Alpha	0.0012725	0.0516074	0.0514212	.	0.0000397	0.0001465
Beta	0.0048839	0.7886128	0.7882909	0.0000397	.	0.0002822
Gamma	0.0082822	0.1602481	0.1598194	0.0001465	0.0002822	.

Categorical input Algorithm

	Dynamic	Greedy	Transform
Dynamic	1.0000	0.9998	0.9997
Greedy	0.9998	1.0000	1.0000
Transform	0.9997	1.0000	1.0000

Categorical input Compiler

	A	B
A	1.0000	0.9998
B	0.9998	1.0000

μ	σ^2
248.29356	19319.739

-2*LogLikelihood
245.0418

Fit using the Gaussian correlation function.
Warning: Likelihood estimation algorithm did not converge.

The actual by predicted plot shows a strong correlation between the actual and predicted CPU times. This in an indication that the Gaussian process prediction model is a good approximation of the true function. In the Model Report, the Beta predictor has the highest total sensitivity. This indicates that of the continuous predictors, Beta explains the most variation in the response. There is a separate Categorical Input matrix for each of the categorical predictors, Algorithm and Compiler. These matrices are correlation matrices and show the correlation between levels for each categorical predictor. The off-diagonals of the matrices are the τ parameters.

Statistical Details for the Gaussian Process Platform

Models with Continuous Predictors

If the Gaussian Process model contains only continuous predictors, the Gaussian Process platform implements two possible correlation structures, the Gaussian and the Cubic.

The Gaussian correlation structure uses the product exponential correlation function with a power of 2 as the estimated model. This model assumes that Y is normally distributed with mean μ and covariance matrix $\sigma^2\mathbf{R}$. The elements of the **R** matrix are defined as follows:

$$r_{ij} = \exp\left(-\sum_{k=1}^{K} \theta_k (x_{ik} - x_{jk})^2\right)$$

where

$K = $ # of continuous predictors

$\theta_k = $ theta parameter for the k^{th} predictor

$x_{ik} = $ the value of the k^{th} predictor for subject i

$x_{jk} = $ the value of the k^{th} predictor for subject j

The Cubic correlation structure also assumes that Y is normally distributed with mean μ and covariance matrix $\sigma^2\mathbf{R}$. The **R** matrix consists of the following elements:

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

where

$d = x_{ik} - x_{jk}$

$$\rho(d;\theta) = \begin{cases} 1 - 6(d\theta)^2 + 6(|d|\theta)^3, & |d| \le \frac{1}{2\theta} \\ 2(1 - |d|\theta)^3, & \frac{1}{2\theta} < |d| \le \frac{1}{\theta} \\ 0, & \frac{1}{\theta} < |d| \end{cases}$$

For more information, see Santer (2003). The theta parameter used in the Cubic correlation structure is the reciprocal of the parameter often used in the literature. The reciprocal is used so that when theta has no effect on the model, then rho has a value of zero, rather than infinity.

Models with Categorical Predictors

If the Gaussian Process model includes categorical predictors, the Gaussian correlation structure is used for the correlation structure. The elements of the **R** matrix are defined as follows:

$$r_{ij} = \left(\prod_{p=1}^{P} \tau_{p_{ij}}\right) \exp\left(-\sum_{k=1}^{K} \theta_k (x_{ik} - x_{jk})^2\right)$$

where

K = # of continuous predictors

P = # of categorical predictors

θ_k = theta parameter for the k^{th} continuous predictor

x_{ik} = the value of the k^{th} continous predictor for subject i

x_{jk} = the value of the k^{th} continous predictor for subject j

$\tau_{p_{ij}}$ = the correlation between the observed level of predictor p for subject i and the observed level of predictor p for subject j

There is a τ parameter for each combination of levels of a categorical variable, with τ_{ij} corresponding to the unique combination formed by the observed levels of subject i and subject j. Thus, the covariance element, r_{ij}, depends on the combination of levels of the categorical predictors obtained from the i^{th} and j^{th} observations. For more information, see Qian et al. (2008).

Variance Formula Parameterization

The saved variance formula uses the previously defined parameterization of **R**, except when the model includes categorical predictors. When the Gaussian Process model includes categorical predictors, the saved variance formula uses the following parameterization of **R**:

$$r_{ij} = \exp\left(-\sum_{k=1}^{K} \theta_k (x_{ik} - x_{jk})^2 - \sum_{p=1}^{P} \phi_{p_{ij}} \right)$$

where $\phi_{p_{ij}} = -\ln(\tau_{p_{ij}})$ and all other variables are as previously defined.

Model Fit Details

The model parameters are fit via maximum likelihood. The fitted parameters are provided in the platform report. The parameters are as follows:

- μ is the Gaussian Process mean,
- σ^2 is the Gaussian Process variance,
- Theta corresponds to the values of θ_k in the definition of **R**.
- The off-diagonals of the categorical input correlation matrices correspond to the values of τ_{pij} in the definition of **R**.

Note: If your report contains the note **Nugget parameters set to avoid singular variance matrix**, JMP has added a ridge parameter to the variance matrix so that it is invertible.

Chapter 15

Time Series Analysis

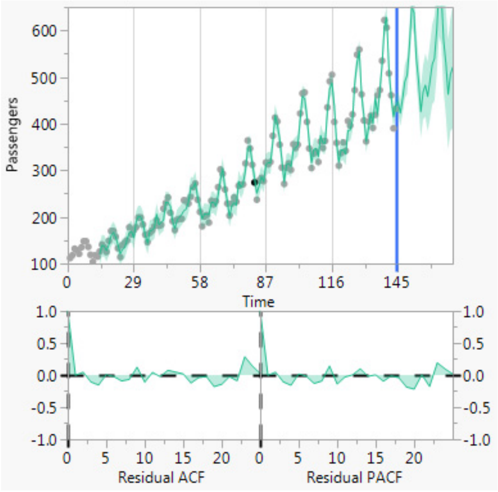
Fit Time Series Models and Transfer Functions

The Time Series platform enables you to explore, analyze, and forecast univariate time series. A time series is a set of observations taken over a series of equally spaced time periods. Observations that are close together in time are typically correlated. Time series methodology takes advantage of this dependence between observations to better predict what the series will look like in the future.

Characteristics that are common in time series data include seasonality, trend, and autocorrelation. The Time Series platform provides options to handle these characteristics. Graphs such as variograms, autocorrelation plots, partial autocorrelation plots, and spectral density plots can be used to identify the type of model appropriate for describing and predicting (forecasting) the time series. There are also several decomposition methods in the platform that enable you to remove seasonal or general trends in the data to simplify the analysis. Alternatively, the platform can fit more sophisticated ARIMA models that have the ability to incorporate seasonality and long term trends all in one model.

The Time Series platform can also fit transfer function models when supplied with an input series.

Figure 15.1 Forecast Plot



Time Series Platform Overview

A time series is a set $y_1, y_2, \ldots, y_N$ of observations that are observed over a series of equally spaced time periods. Some examples of time series data include quarterly sales reports, monthly average temperatures, and counts of sunspots. The Time Series platform enables you to explore patterns and trends found in these types of data. You can then use these patterns and trends to forecast, or predict, into the future.

Characteristics that are common in time series data include seasonality, trend, and autocorrelation. *Seasonality* refers to patterns that occur over a known period of time. For example, data that are collected monthly might look similar in summer months across all years of data collection. *Trend* refers to long term movements of a series, such as gradual increases or decreases of values across time. *Autocorrelation* is the degree to which each point in a series is correlated with earlier values in the series.

There are many different models and forecasting methods available in the Time Series platform. However, not all methods can handle trend or seasonality. In order to choose an appropriate model, it is essential to determine which characteristics are present in the series. The Time Series platform provides graphs such as variograms, autocorrelation plots, partial autocorrelation plots, and spectral density plots that can be used to identify the type of model appropriate for describing and forecasting the evolution of the time series. There are also several differencing and decomposition methods in the platform that enable you to remove seasonal or general trends in the data to explore and simplify the analysis.

Alternatively, the platform can fit more sophisticated models that can incorporate seasonality and long term trends. One such model in the platform that has this ability is Winter's Additive Method, which is an advanced exponential smoothing model. In addition, the platform can fit AutoRegressive Integrated Moving Average, or ARIMA, models. These models are the most statistically complex, but also provide the most flexibility. Both advanced exponential smoothing and ARIMA models are harder to interpret, but they are excellent tools for forecasting.

The Time Series platform can also fit transfer function models when supplied with an input series.

Example of the Time Series Platform

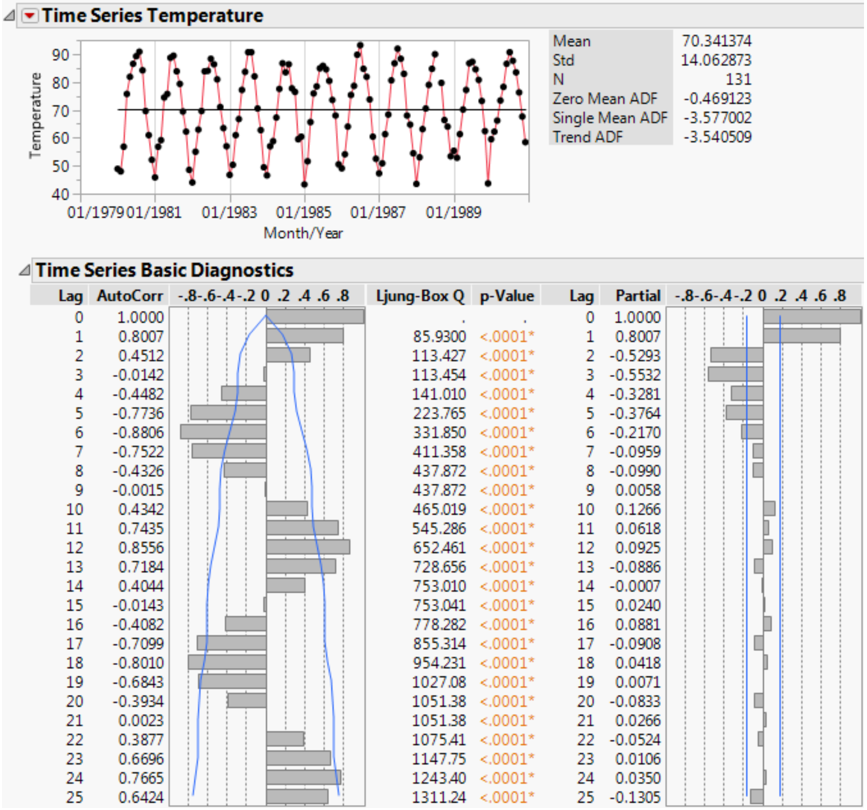
This example uses the Raleigh Temps.jmp sample data table, which contains maximum monthly temperatures measured in degrees Fahrenheit from 1980 to 1990. Use the Time Series platform to examine the series and predict the maximum monthly temperatures for the next two years.

1. Select **Help > Sample Data Library** and open Time Series/Raleigh Temps.jmp.
2. Select **Analyze > Specialized Modeling > Time Series**.

3. Select Temperature and click **Y, Time Series**.
4. Select Month/Year and click **X, Time ID**.
5. In the box next to **Forecast Periods**, type 24.

   This is the number of future periods that are forecast by the models fit to the data. You want to predict the monthly temperature for the next two years, which is 24 months.
6. Click **OK**.

Figure 15.2 Time Series Analysis Report for Raleigh Temps.jmp



Time Series Temperature

Mean	70.341374
Std	14.062873
N	131
Zero Mean ADF	-0.469123
Single Mean ADF	-3.577002
Trend ADF	-3.540509

Time Series Basic Diagnostics

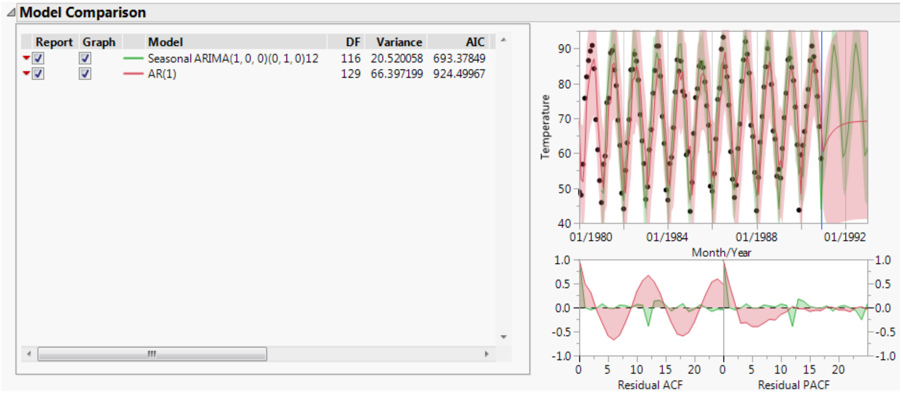
Lag	AutoCorr	-.8-.6-.4-.2 0 .2 .4 .6 .8	Ljung-Box Q	p-Value	Lag	Partial	-.8-.6-.4-.2 0 .2 .4 .6 .8
0	1.0000		.	.	0	1.0000	
1	0.8007		85.9300	<.0001*	1	0.8007	
2	0.4512		113.427	<.0001*	2	-0.5293	
3	-0.0142		113.454	<.0001*	3	-0.5532	
4	-0.4482		141.010	<.0001*	4	-0.3281	
5	-0.7736		223.765	<.0001*	5	-0.3764	
6	-0.8806		331.850	<.0001*	6	-0.2170	
7	-0.7522		411.358	<.0001*	7	-0.0959	
8	-0.4326		437.872	<.0001*	8	-0.0990	
9	-0.0015		437.872	<.0001*	9	0.0058	
10	0.4342		465.019	<.0001*	10	0.1266	
11	0.7435		545.286	<.0001*	11	0.0618	
12	0.8556		652.461	<.0001*	12	0.0925	
13	0.7184		728.656	<.0001*	13	-0.0886	
14	0.4044		753.010	<.0001*	14	-0.0007	
15	-0.0143		753.041	<.0001*	15	0.0240	
16	-0.4082		778.282	<.0001*	16	0.0881	
17	-0.7099		855.314	<.0001*	17	-0.0908	
18	-0.8010		954.231	<.0001*	18	0.0418	
19	-0.6843		1027.08	<.0001*	19	0.0071	
20	-0.3934		1051.38	<.0001*	20	-0.0833	
21	0.0023		1051.38	<.0001*	21	0.0266	
22	0.3877		1075.41	<.0001*	22	-0.0524	
23	0.6696		1147.75	<.0001*	23	0.0106	
24	0.7665		1243.40	<.0001*	24	0.0350	
25	0.6424		1311.24	<.0001*	25	-0.1305	

The Time Series graph shows that the series is cyclic. This cyclic component is also apparent in the autocorrelation chart. Points that are 1 lag apart are positively correlated, with an AutoCorr value of 0.8666. As points become farther apart, they become negatively correlated, then positively correlated again, and then the pattern repeats. The Time Series graph and the autocorrelation chart provide evidence of seasonality in the time series.

7. Click the Time Series red triangle and select **ARIMA**.
8. Set *p*, the autoregressive order, to 1 because the series showed evidence of autocorrelation.
9. Click **Estimate**.

10. Click the Time Series red triangle and select **Seasonal ARIMA**.
11. In the ARIMA box, set *p*, the autoregressive order, to 1 because the series showed evidence of autocorrelation.
12. In the Seasonal ARIMA box, set *D*, the seasonal differencing order, to 1 because the series showed evidence of seasonality.
13. Click **Estimate**.
14. In the Model Comparison table, check the box under **Graph** for both models.

Figure 15.3 Model Comparison Table for Raleigh Temps.jmp



Model Comparison

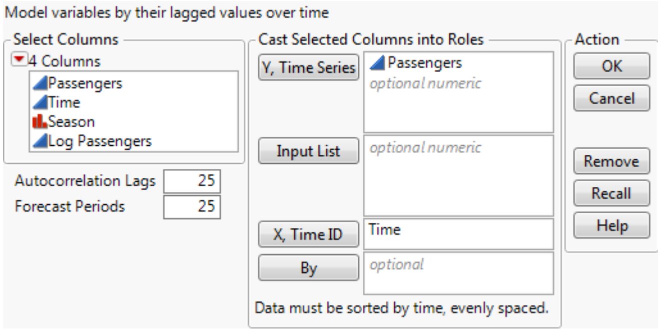
Report	Graph	Model	DF	Variance	AIC
☑	☑	Seasonal ARIMA(1, 0, 0)(0, 1, 0)12	116	20.520058	693.37849
☑	☑	AR(1)	129	66.397199	924.49967

The Model Comparison table is sorted by the AIC statistic, in decreasing order. This means that the best fitting model appears at the top of the report. The AIC value for the seasonal ARIMA model (689.4) is much smaller than the value for the regular ARIMA model (920.5). The graph shows that while the ARIMA model predicts the observed points relatively well, the residuals are larger than those from the Seasonal ARIMA model. Also, the Seasonal ARIMA model has more realistic predictions for future observations with narrower prediction intervals. These results make sense since the series showed evidence of a seasonal component.

Launch the Time Series Platform

Launch the Time Series platform by selecting **Analyze > Specialized Modeling > Time Series**. The Time Series launch window for the Seriesg.jmp sample data table is shown in Figure 15.4.

Figure 15.4 The Time Series Launch Window



Model variables by their lagged values over time

Select Columns: 4 Columns — Passengers, Time, Season, Log Passengers

Autocorrelation Lags: 25

Forecast Periods: 25

Cast Selected Columns into Roles

Y, Time Series: Passengers

Input List: optional numeric

X, Time ID: Time

By: optional

Data must be sorted by time, evenly spaced.

The Time Series platform launch window contains the following options:

Y, Time Series Assigns one or more columns as time series variables. Displayed on the *y*-axis.

Input List Assigns one or more columns as input series variables. Displayed in the Input Time Series Panel and used in transfer function models. The input series variable must be numeric, either as a time series or an indicator.

X, Time ID Assigns one variable for labeling the time axis (*x*-axis). If no variable is specified for Time ID, the row number is used instead.

Note: If you use an **X, Time ID** variable, you can specify the time frequency by using the Time Frequency column property. You can choose Annual, Quarterly, Monthly, Weekly, Daily, Hourly, By Minute, and By Second. This helps JMP determine the spacing of the data when plotting the forecast values. If no frequency is specified, the data is treated as equally spaced numeric data.

Caution: It is assumed that the observations of the variable assigned to **X, Time ID** are equally spaced. However, the Time Series platform only checks whether the time stamps are increasing, and it. The platform does not check if the observations are equally spaced.

By Assigns a column that creates a report consisting of separate analyses for each level of the variable. If more than one By variable is assigned, a separate analysis is produced for each possible combination of the levels of the By variables.

Note: If you use a By variable, you might have to change the number of autocorrelation lags depending on how many observations there are for each level of the By variable. The number of lags must be greater than one but less than the number of observations per level.

Autocorrelation Lags Specifies the number of lags to use in computing the autocorrelations and partial correlations. This is the maximum number of periods between points used in the computation of these correlations. It must be greater than one but less than the number of rows. The default number of lags is 25.

Tip: A commonly used rule for the maximum number of lags is *n*/4, where *n* is the number of observations.

Forecast Periods Specifies the number of future periods that are forecast using each model fitted to the data. The default number of forecast periods is 25.

The Time Series Analysis Report

The initial Time Series report displays the time series graph, summary statistics and tests for the time series variable, and a basic diagnostics report. If a column is specified for Input List in the launch window, the Transfer Function Analysis and Input Time Series Panel reports are shown. Both the Transfer Function Analysis report and the Input Time Series Panel report contain the same initial information as the Time Series report.

Time Series Graph

The Time Series graph plots each times series by the time ID. If no time ID is specified, the row number is used instead. The platform also performs several tests for stationarity using *Augmented Dickey-Fuller* (ADF) tests. The following tests and summary statistics are displayed next to the time series graph:

Mean The sample mean.

SD The sample standard deviation.

N The series length.

Zero Mean ADF A test against a random walk with zero mean, which is defined as follows:

$$x_t = \phi x_{t-1} + e_t$$

Single Mean ADF A test against a random walk with a non-zero mean, which is defined as follows:

$$x_t - \mu = \phi(x_{t-1} - \mu) + e_t$$

Trend ADF A test against a random walk with a non-zero mean and a linear trend, which is defined as follows:

$$x_t - \mu - \beta t = \phi[x_{t-1} - \mu - \beta(t-1)] + e_t$$

Time Series Basic Diagnostics Chart

The information that is shown in the Time Series Basic Diagnostics chart depends on the Time Series Report red triangle menu options. The red triangle menu options that show or hide information from the diagnostics chart are Autocorrelation, Partial Autocorrelation, Variogram, and AR Coefficients. By default, Autocorrelation and Partial Autocorrelation are shown.

Autocorrelation Chart

The Autocorrelation option shows or hides the following columns in the Time Series Basic Diagnostics chart:

Lag The number of periods between points.

Note: 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**.

AutoCorr The autocorrelation for the *k*th lag, computed as follows:

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

and $\bar{y}$ is the mean of the *N* non-missing points in the time series. By definition, the first autocorrelation (lag 0) always has length 1.

The bars graphically depict the autocorrelations. The blue curves represent twice the large-lag standard error (± 2 standard errors), computed as follows:

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

Ljung-Box *Q* 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. *Q* is the test statistic.

p-Value The p-value from the Ljung-Box test.

Partial Autocorrelation Chart

The Partial Autocorrelation option shows or hides the following columns in the Time Series Basic Diagnostics chart:

Lag The number of periods between points.

Partial The partial autocorrelation for the *k*th lag.

The bars graphically depict the partial autocorrelations. The blue lines represent ± 2 standard errors for approximate 95% prediction limits, where the standard error is computed as follows:

$$\mathrm{SE}_k = \frac{1}{\sqrt{n}} \text{ for all } k$$

Variogram Chart

The Variogram option shows or hides the following columns in the Time Series Basic Diagnostics chart:

Lag The number of periods between points.

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 follows:

$$V_k = \frac{1 - r_{k+1}}{1 - r_1}$$

where r_k is the autocorrelation at lag *k*.

AR Coefficients Chart

The AR Coefficients option shows or hides the following columns in the Time Series Basic Diagnostics chart:

Lag The number of periods between points.

AR Coef The coefficients approximate those that you would obtain from fitting a high-order, purely autoregressive model.

Time Series Platform Options

The Time Series, Transfer Function Analysis, and Input Series red triangle menus contain the same set of options.

Time Series Diagnostics

Graph Shows a submenu of options to control the time series plot appearance.

Time Series Graph Shows or hides the time series graph.

Show Points Shows or hides the points in the time series graph.

Connecting Lines Shows or hides the lines connecting the points in the time series graph.

Mean Line Shows or hides a horizontal line in the time series graph that depicts the mean of the time series.

Autocorrelation Shows or hides the Autocorrelation plot in the Time Series Basic Diagnostics Chart. The autocorrelation graph describes the correlation between all pairs of points in the time series for a given separation in time (lag). See "Autocorrelation Chart" on page 243.

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

Partial Autocorrelation Shows or hides the Partial Autocorrelation plot in the Time Series Basic Diagnostics Chart. The partial autocorrelation graph describes the partial correlation between all the pairs of points in the time series for a given separation in time (lag). See "Partial Autocorrelation Chart" on page 243.

Tip: The 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.

Variogram Shows or hides the graph of the variogram in the Time Series Basic Diagnostics Chart. See "Variogram Chart" on page 244.

AR Coefficients Shows or hides the graph of the least squares estimates of the autoregressive (AR) coefficients in the Time Series Basic Diagnostics Chart. See "AR Coefficients Chart" on page 244.

Spectral Density Shows or hides graphs of the spectral density as a function of period and frequency. The spectral density option also displays the White Noise test report, which gives results from two tests on the data. See "Spectral Density Report" on page 264 and "Statistical Details for Spectral Density" on page 269.

Differencing and Decomposition

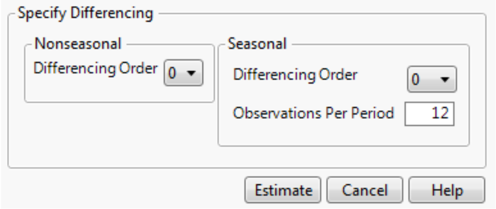
Difference Shows the Differencing Specification window, shown in Figure 15.5. The window enables you to specify the differencing operation that you want to apply to the time series. Differencing the values in a time series can transform a nonstationary series into a stationary series. The differenced series is given by the following equation:

$$w_t = (1 - \mathrm{B})^d (1 - \mathrm{B}^s)^D y_t$$

where t is the time index and **B** is the backshift operator defined by $\mathbf{B}y_t = y_{t-1}$.

Note: Many time series do not exhibit a fixed mean, such as time series with trend or seasonality. Such nonstationary series are not suitable for description by time series models that assume a stationary time series such as ARMA models. Removing the trend and/or seasonality creates a differenced series that is stationary and enables you to describe the series using the models that assume stationarity.

Figure 15.5 Differencing Specification Window



Nonseasonal Differencing Order: 0

Seasonal Differencing Order: 0

Observations Per Period: 12

The Differencing Specification window enables you to specify the Nonseasonal Differencing Order, *d*, the Seasonal Differencing Order, *D*, and the number of Observations per Period, *s*. Selecting zero for the value of the differencing order is equivalent to no differencing of that kind. Each time you specify a differencing operation and click **Estimate**, a new Difference Report is displayed in the report window. For more information, see "Additional Example of the Time Series Platform" on page 265.

Decomposition Shows a submenu of decomposition methods. Decomposition of time series data isolates and removes linear trends and seasonal cycles from a time series. This can help with better model estimation. Three Decomposition options are provided.

Remove Linear Trend Estimates the linear trend of the time series using a linear regression model and removes the linear trend from the data. A Time Series report for the detrended series is added to the report window, along with the linear trend information. See "The Time Series Analysis Report" on page 242 and "Linear Trend Report" on page 255.

Remove Cycle Estimates the cyclic component of a time series using a single cosine wave and then removes the cyclic component from the data. When you select the Remove Cycle option, the Define Cycle dialog appears. This dialog window enables you to specify the number of units per cycle and indicate whether a constant should be subtracted from the data. A Time Series report for the decycled series is added to the report window, along with the cycle information. See "The Time Series Analysis Report" on page 242 and "Cycle Report" on page 255.

X11 Removes trend and seasonal effects using the X-11 method developed by the US Bureau of the Census (Shiskin et. al., 1967). For details about the X-11 method, see "Statistical Details for X-11 Decomposition" on page 270. When selected, the Select

Decomposition Type dialog appears. This dialog window enables you to specify a multiplicative or additive X-11 adjustment. Once you click **OK**, an X11 report is added to the report window. See "X11 Report" on page 256.

The X11 option is available only for monthly or quarterly data. The X, Time ID column must contain numeric values equally spaced by month or quarter without any gaps or missing values. JMP returns an error if you request X11 for a time column that does not satisfy these requirements.

Note: When you select the **Remove Linear Trend** or the **Remove Cycle** options, JMP adds a column to the data table that contains the detrended or decycled data. If this column is already present in the data table when you select the option, JMP overwrites the existing column.

Tip: Typically, you would begin decomposition by removing any linear trend, and then removing long cycles, such as a 12-month cycle. Then you could start removing short cycles, such as 6-month cycles.

Show Lag Plot Shows or hides a plot with observations at time *t* on the *y*-axis and observations at time *t* +/- *p* on the *x*-axis. The +/- *p* is known as the lag. This plot is useful in determining how an observation at time *t* is related to another observation at time *t* +/- *p*. If there is not an identifiable structure to the plot, the observations are not related. However, if there is a structure to the plot, this indicates that there is some relationship between observations across time. Identifying the structure helps when building a time series model.

Cross Correlation (Available only in the Transfer Function Analysis red triangle menu.) Shows or hides a cross-correlation plot to the report. The length of the plot is twice that of an autocorrelation plot, or $2 \times \text{ACF length} + 1$. The plot includes plots of the output series versus all input series, in both numerical and graphical forms. The blue lines indicate two standard errors.

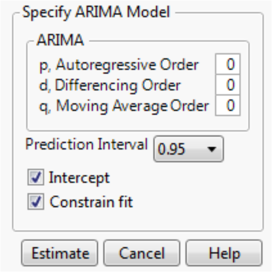
Prewhitening (Available only in the Input Series red triangle menu.) Shows the Prewhitening Specification window that enables you to set the prewhitening order. Prewhitening is a technique used to help identify the transfer function model. For information about prewhitening, see Box et al. (1994).

ARIMA and Seasonal ARIMA Models

ARIMA Shows the ARIMA Specification window, which enables you to specify the ARIMA model that you want to fit. An 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 model performs a maximum likelihood fit of the specified ARIMA model to the time series. See "ARIMA Model" on page 274.

Note: 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 the model would simply be a moving average model, denoted as MA(q).

Figure 15.6 ARIMA Specification Window



Specify ARIMA Model

ARIMA

p, Autoregressive Order 0
d, Differencing Order 0
q, Moving Average Order 0

Prediction Interval 0.95

☑ Intercept
☑ Constrain fit

p, **Autoregressive Order** The order p of the polynomial $\varphi(B)$ operator.

d, **Differencing Order** The order d of the differencing operator.

q, **Moving Average Order** The order q of the differencing operator $\theta(B)$.

Prediction Interval Enables you to set the prediction level between 0 and 1 for the forecast prediction intervals.

Intercept Determines whether the intercept term μ is a part of the model.

Constrain fit If checked, the fitting procedure constrains the autoregressive parameters to always remain within the stable region and the moving average parameters within the invertible region.

Tip: Deselect the Constrain fit option if the fitter is having difficulty finding the true optimum or if you want to speed up the fit. You can use the Model Summary table to see whether the resulting fitted model is stable and invertible.

Once you specify the model and click **Estimate**, a Model Report is added to the report window. See “Reports” on page 254.

Seasonal ARIMA Shows the Seasonal ARIMA Specification window, which enables you to specify the Seasonal ARIMA model that you want to fit. This window has the same elements as the ARIMA specification window, but it also contains the seasonal element specifications. The additional Observations per Period option enables you to specify the number of observations per period, denoted as s. For more information about the Seasonal ARIMA model, see “Seasonal ARIMA Model” on page 275.

Note: Seasonal ARIMA models are denoted as Seasonal ARIMA(p,d,q)(P,D,Q)s.

Once you specify the model and click **Estimate**, a Model Report is added to the report window. See "Reports" on page 254.

Smoothing Models

Shows a submenu of smoothing models. Once you select a smoothing model, a specification window appears. See "Smoothing Model Windows" on page 253. For each model that is specified, a Smoothing Model Report appears in the report window. See "Model Report" on page 258. Smoothing models represent the evolution of a time series by the model:

$$y_t = \mu_t + \beta_t t + s(t) + a_t$$

where

μ_t is the time-varying mean term

β_t is the time-varying slope term

$s(t)$ is one of the s time-varying seasonal terms

a_t are the random shocks

For more information about the general smoothing model equation, see "Statistical Details for Smoothing Models" on page 270. The following smoothing models are available:

Simple Moving Average A model that estimates values by using an average of several adjacent points, defined by the smoothing window. The Simple Smoothing Average Specification window enables you to specify aspects of the smoothing window. Once specified, a Simple Moving Average report is shown. By default, this report produces plotted values that are equal to the average of consecutive observations in a time window. Multiple Simple Moving Average models can be added and shown on the same plot. For details, see "Simple Smoothing Average Specification Window" on page 252.

Simple Exponential Smoothing A model with a level component. See "Simple Exponential Smoothing" on page 271.

Double Exponential Smoothing A model with a level component and a trend component. This is a special case of Linear Exponential Smoothing. See "Double (Brown) Exponential Smoothing" on page 271.

Linear Exponential Smoothing A model with a level component and a trend component. See "Linear (Holt) Exponential Smoothing" on page 272.

Damped-Trend Linear Exponential Smoothing A model with a level component and a damped trend component. This model is appropriate for a series that exhibits a trend more complicated than a linear trend. See "Damped-Trend Linear Exponential Smoothing" on page 272.

Seasonal Exponential Smoothing A model with a level component and a seasonal component. See "Seasonal Exponential Smoothing" on page 273.

Winters Method A model with a level component, a trend component, and a seasonal component. See "Winters Method (Additive)" on page 273.

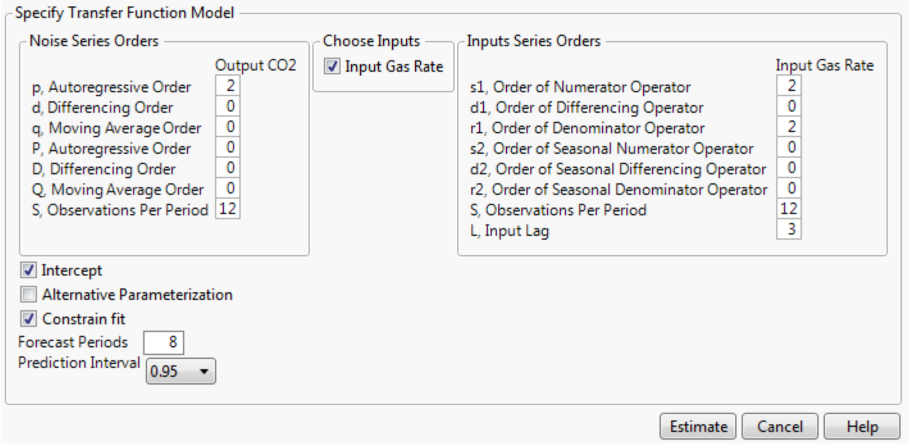
Note: Each smoothing model has an ARIMA model equivalent. You might not be able to specify the equivalent ARIMA model using the ARIMA option because some smoothing models intrinsically constrain the ARIMA model parameters in ways that the ARIMA option does not allow.

Transfer Function Models

Transfer Function (Available only in the Transfer Function Analysis red triangle menu.) Shows the Transfer Function Model Specification window. Building a transfer function model is similar to building an ARIMA model; it is an iterative process of exploring, fitting, and comparing models. Before building a model and during the data exploration process, it is sometimes useful to prewhiten the data. See "Statistical Details for Transfer Functions" on page 275.

Note: Currently, the transfer function model platform has limited capability of supporting missing values.

Figure 15.7 Transfer Function Model Specification Window



Specify Transfer Function Model

Noise Series Orders

	Output CO2
p, Autoregressive Order	2
d, Differencing Order	0
q, Moving Average Order	0
P, Autoregressive Order	0
D, Differencing Order	0
Q, Moving Average Order	0
S, Observations Per Period	12

Choose Inputs

☑ Input Gas Rate

Inputs Series Orders

	Input Gas Rate
s1, Order of Numerator Operator	2
d1, Order of Differencing Operator	0
r1, Order of Denominator Operator	2
s2, Order of Seasonal Numerator Operator	0
d2, Order of Seasonal Differencing Operator	0
r2, Order of Seasonal Denominator Operator	0
S, Observations Per Period	12
L, Input Lag	3

☑ Intercept

□ Alternative Parameterization

☑ Constrain fit

Forecast Periods 8

Prediction Interval 0.95

The Transfer Function Model Specification window contains the following sections:

Noise Series Orders Contains specifications for the noise series. Lowercase letters are coefficients for non-seasonal polynomials, and uppercase letters are coefficients for seasonal polynomials.

Choose Inputs Enables you select the input series for the model.

Input Series Orders Contains specifications for the input series. The first three orders relate to non-seasonal polynomials. The next four orders relate to seasonal polynomials. The final option is for an input lag.

There are three additional options that control model fitting:

Intercept Specifies whether μ is zero.

Alternative Parameterization Specifies whether the general regression coefficient is factored out of the numerator polynomials.

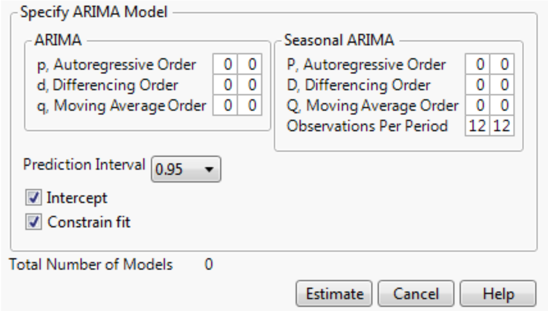
Constrain Fit Toggles the constraining of the AR and MA coefficients.

Forecast Periods Specifies the number of forecasting periods that are used for forecasting. If there are rows at the end of the data table that contain missing values for the Y variable and nonmissing values for the input variables, these rows are used in the initial forecasting settings. The values for the input variables are treated as future values of the input variables.

Prediction Interval Specifies the confidence level for the prediction interval.

ARIMA Model Group Shows the ARIMA Model Group window, which enables you to fit a range of ARIMA or Seasonal ARIMA models by specifying the range of orders. As you enter ranges into the window, the Total Number of Models updates accordingly.

Figure 15.8 ARIMA Model Group Specification Window



p, Autoregressive Order 0 0
d, Differencing Order 0 0
q, Moving Average Order 0 0

P, Autoregressive Order 0 0
D, Differencing Order 0 0
Q, Moving Average Order 0 0
Observations Per Period 12 12

Prediction Interval 0.95

☑ Intercept
☑ Constrain fit

Total Number of Models 0

Once you specify the models and click **Estimate**, a Model Report for each specified model is added to the report window. See "Reports" on page 254.

Save Spectral Density Creates a new data table containing the spectral density and periodogram where the *(i+1)*th row corresponds to the frequency $f_i = i / N$ (that is, the *i*th harmonic of $1 / N$). The new data table has the following columns:

Period The period of the *i*th harmonic, $1 / f_i$.

Frequency The frequency of the harmonic, f_i.

Angular Frequency The angular frequency of the harmonic, $2\pi f_i$.

Sine The Fourier sine coefficients, a_i.

Cosine The Fourier cosine coefficients, b_i.

Periodogram The periodogram, $I(f_i)$.

Spectral Density The spectral density, a smoothed version of the periodogram.

Number of Forecast Periods Shows a window that enables you to set the number of future periods that are forecast for the fitted models. The initial value is set in the Time Series launch window. All existing and future forecast results will use the new number of periods once it is changed.

Maximum Iterations Shows a window that enables you to reset the maximum number of iterations for future optimizations used in fitting ARIMA models.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

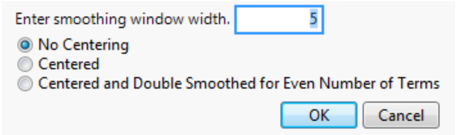
Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Smoothing Model Specification Windows

Simple Smoothing Average Specification Window

The Simple Smoothing Average Specification window appears when you select Simple Moving Average as the smoothing model. Let w be the smoothing window width in a simple moving average (SMA) model. Let $f_t=(y_t+y_{t-1}+y_{t-2}+...y_{t-(w-2)}+y_{t-(w-1)})/w$ be the average of w consecutive observations for some time point t.

Figure 15.9 Simple Smoothing Average Specification Window



Enter smoothing window width. 5

● No Centering

○ Centered

○ Centered and Double Smoothed for Even Number of Terms

Enter smoothing window width The smoothing window width, *w*, that defines the number of consecutive points to average. The larger the window width, the more the series is smoothed.

No Centering The smoothing window is constructed from the points leading up to and including the time point, *t*, the point at which the series is being estimated. In other words, f_t is the plotted value for time *t*.

Centered The smoothing window is centered around the time point at which the series is being estimated.

- For odd *w*, f_t is the plotted value for time *t*-(*w*-1)/2.
- For even *w*, f_t is the plotted value for time *t*-(*w*-1)/2. When saved to a data table, f_t is at *t*-(*w*-2)/2.

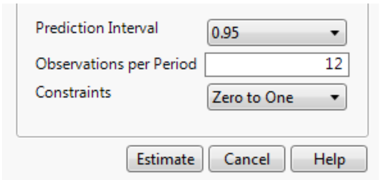
Centered and Double Smoothed for Even Number of Terms For even *w*, the smoothing window cannot be centered around the time point at which the series is being estimated. This option creates two smoothing windows that are almost centered, and averages them together. The smoothing estimates are calculated as follows:

$$f_{t-\frac{w}{2}} = \frac{y_t + 2\sum_{i=1}^{w-1} y_{t-i} + y_{t-w}}{2w}$$

Smoothing Model Windows

The Smoothing Model specification windows appear when you select one of the smoothing model options other than Simple Moving Average. The title of the window and the available options depend on the smoothing model option that you select.

Figure 15.10 Smoothing Model Specification Window



Prediction Interval 0.95

Observations per Period 12

Constraints Zero to One

Prediction Interval Enables you to set the prediction level for the forecast prediction intervals.

Observations per Period (Available only for seasonal smoothing models.) Enables you to set the number of observations per period in a seasonal smoothing model.

Constraints Enables you to specify what type of constraint you want to enforce on the smoothing weights during the fit. The constraint options are as follows:

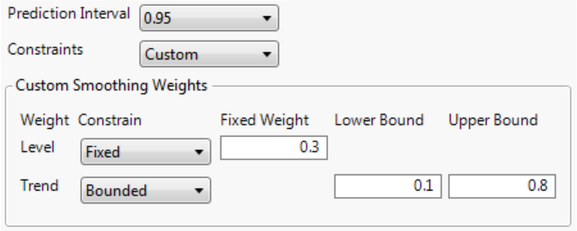
Zero To One Constrains the values of the smoothing weights to the range zero to one.

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 enable 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.

Figure 15.11 Custom Smoothing Weights



Prediction Interval: 0.95
Constraints: Custom

Custom Smoothing Weights

Weight	Constrain	Fixed Weight	Lower Bound	Upper Bound
Level	Fixed	0.3		
Trend	Bounded		0.1	0.8

The example shown in Figure 15.11 has the Level weight (α) fixed at a value of 0.3 and the Trend weight (γ) 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 constant 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.

Reports

Difference Report

The Difference Report contains graphs of the autocorrelations and partial autocorrelations of the differenced series. These graphs can be used to determine whether the differenced series is stationary.

The ARIMA and Seasonal ARIMA models that are available in the Time Series platform accommodate a differencing operation. In a two step process, these models first difference the time series according to the differencing operation, and then fit the differenced series. The Difference option is a useful preprocessing tool for determining the order of differencing to specify for the ARIMA model.

The Difference red triangle menu contains the following options:

Graph Shows a submenu of options to control the appearance of the differenced series plot. See "Time Series Platform Options" on page 244.

Autocorrelation Shows or hides the autocorrelation of the differenced series.

Partial Autocorrelation Shows or hides the partial autocorrelations of the differenced series.

Variogram Shows or hides the variogram of the differenced series.

Save Saves a new column that contains the values in the differenced series to the original data table. Some of the leading elements are lost in the differencing process. They are represented as missing values in the saved Difference column.

Decomposition Reports

This section provides details about the reports obtained from the three decomposition options:

- "Linear Trend Report" on page 255
- "Cycle Report" on page 255
- "X11 Report" on page 256

Linear Trend Report

Contains the values of β_0 and β_1 from the linear regression model that is fit to the data:

$$\text{Trend}_t = \beta_0 + \beta_1 \textit{time}$$

The detrended series is equal to $D_t = O_t - \text{Trend}_t$, where O_t is the original time series.

Cycle Report

Contains the values of the cyclical component that is fit to the data:

$$\text{Cycle}_t = C + A \cdot \cos\left(2 \cdot \pi \cdot \left(\left(\frac{1}{U}\right) \cdot t + P\right)\right)$$

The parameter values are defined as follows:

C The (optional) Constant

A The Amplitude of the cosine wave

U The number of Units per Cycle

P The Phase of the cosine wave

t One less than the row number of a given observation

The decycled series is equal to $D_t = O_t$ - Cycle$_t$, where O_t is the original time series.

X11 Report

Depending on your selection of Decomposition Type, the X11 option adds an X11 - Multiplicative report or an X11 - Additive report. The reports contain the same four plots:

Original and Adjusted Overlays the X11-adjusted time series on the original time series, O_t. The X11-adjusted values are O_t/S_t for the multiplicative adjustment and $O_t - S_t$ for the additive adjustment.

D10 - Final Seasonal Factors Plots the seasonal factor components, S_t, over time.

D12 - Final Trend Cycle Plots the trend cycle components, C_t, over time.

D13 - Final Irregular Series Plots the irregular components, I_t, over time.

X11 Report Options

The X11 reports have the following red triangle options:

Show Tables Shows or hides the X11 summary tables, as described in Shiskin et. al. (1967). The tables are grouped into five categories (labeled B through F), described in Table 15.1.

Save Columns Saves four columns to the data table: the seasonally adjusted time series, the seasonal components (S_t), the trend cycle components (C_t), and the irregular series components (I_t).

Save All Columns Saves columns to the data table for all of the tables produced in the report by the **Show Tables** option.

Table 15.1 Descriptions of the Categories of X11 Output Tables

Letter Prefix	Category Description
B	preliminary estimates of seasonal, trend cycle, and irregular components
C	intermediate estimates of seasonal, trend cycle, and irregular components
D	final estimates of seasonal, trend cycle, and irregular components
E	analytical tables

Table 15.1 Descriptions of the Categories of X11 Output Tables *(Continued)*

Letter Prefix	Category Description
F	summary measures

For details about the contents of the X11 output tables, see Shisken et. al. (1967) or *SAS/ETS 13.1 User's Guide* (search for "The output from PROC X11").

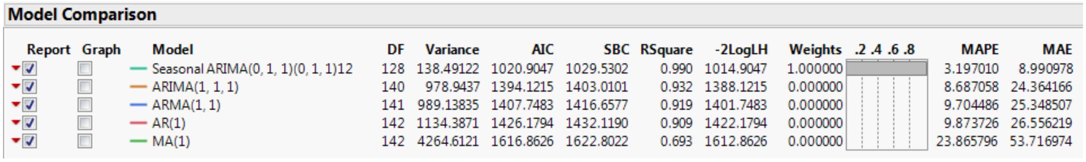
Model Comparison Report

Once a model is fit, the Model Comparison Report is displayed in the report window. This report contains the Model Comparison table and plots for the models. Each time a new model is fit, a new row is added to the Model Comparison table, with a unique color-coding. The Model Comparison table summarizes the fit statistics for each model and is used to compare several models fitted to the same time series. The models are sorted by the AIC statistic, in decreasing order. For definitions of the fit statistics, see "Model Summary Table" on page 259. The only fit statistic that is unique to the Model Comparison Table is Weights. This fit statistic is the normalized AIC Weight. The AIC Weight for a model is calculated as follows:

$$\mathrm{AICWeight} = \exp[-0.5(\mathrm{AIC} - \mathrm{BestAIC})] / \sum_{k=1}^{K} (\exp[-0.5(\mathrm{AIC}_k - \mathrm{BestAIC})])$$

K is the total number of models, AIC_k is the AIC value for model k, and BestAIC is the AIC value for the model with the minimum AIC value.

Figure 15.12 Model Comparison Table

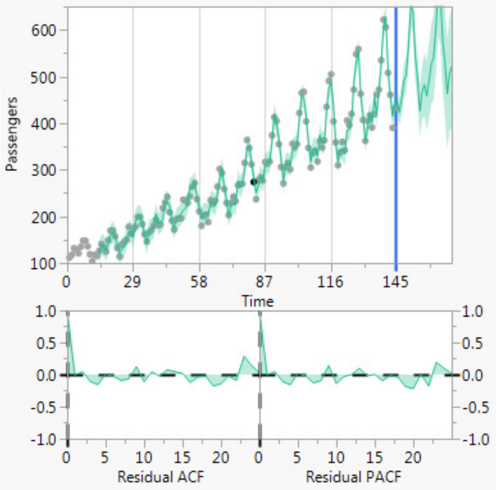


Model Comparison

Report	Graph	Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	Weights	.2 .4 .6 .8	MAPE	MAE
☑	□	Seasonal ARIMA(0, 1, 1)(0, 1, 1)12	128	138.49122	1020.9047	1029.5302	0.990	1014.9047	1.000000		3.197010	8.990978
☑	□	ARIMA(1, 1, 1)	140	978.9437	1394.1215	1403.0101	0.932	1388.1215	0.000000		8.687058	24.364166
☑	□	ARMA(1, 1)	141	989.13835	1407.7483	1416.6577	0.919	1401.7483	0.000000		9.704486	25.348507
☑	□	AR(1)	142	1134.3871	1426.1794	1432.1190	0.909	1422.1794	0.000000		9.873726	26.556219
☑	□	MA(1)	142	4264.6121	1616.8626	1622.8022	0.693	1612.8626	0.000000		23.865796	53.716974

You can select which full model reports are shown in the report window using the **Report** checkbox. Two model plots appear to the right of the Model Comparison table. The top plot is a time series plot of the data, forecasts, and prediction limits. Below that are plots of the autocorrelation and partial autocorrelation functions. You can select which models are displayed on the model plots using the **Graph** checkbox.

Figure 15.13 Model Plots



Model Comparison Report Options

Each model in the Model Comparison report has the following red triangle menu options:

Fit New Opens a specification window for 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 in the Time Series 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 in the Time Series red triangle menu.

Remove Model Simulation Removes the simulations for the model.

Remove All Simulation Removes the simulations for all models.

Generate Simulation Generates simulations for the model, and stores the results in a data table. You can specify the random seed, number of simulations, and the number of forecast periods.

Set Seed Specifies the seed for generating the simulated forecast values.

Model Report

The time series modeling options are used to fit theoretical models to the series and use the fitted model to predict (forecast) future values of the series. These options also produce statistics and residuals that enable you to determine the adequacy of the model that you have chosen to use. You can select the modeling options multiple times. Each time you select a model, that model is added to the Model Comparison table. a report of the results of the fit and a forecast is added to the Time Series report window. When the Report checkbox next to a model in the Model Comparison table is selected, a report is produced for that model. The report specifies the model in its title.

The following reports are shown by default:

- Model Summary Table
- Parameter Estimates Table
- Forecast Plot
- Residuals
- Iteration History

Model Summary Table

The Model Summary table contains fit statistics for the model. In the formulas below, n is the length of the series and k is the number of fitted parameters in the model.

DF The number of degrees of freedom in the fit, $n - k$.

Sum of Squared Errors The sum of the squared residuals.

Variance Estimate The unconditional sum of squares (SSE) divided by the number of degrees of freedom ($n - k$). The variance estimate is computed as SSE / ($n - k$). This is the sample estimate of the variance of the random shocks a_t, described in the section "ARIMA Model" on page 274.

Standard Deviation The square root of the variance estimate. This is a sample estimate of the standard deviation of the random shocks a_t.

Akaike's Information Criterion [AIC] Smaller AIC values indicate better fit. AIC is computed as follows:

$$\text{AIC} = -2\text{loglikelihood} + 2k$$

Schwarz's Bayesian Criterion [SBC or BIC] Smaller SBC values indicate better fit. Schwarz's Bayesian Criterion is equivalent to the Bayesian Information Criterion (BIC). SBC is computed as follows:

$$\text{SBC} = -2\text{loglikelihood} + k\ln(n)$$

RSquare RSquare is computed as follows:

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

where

$$\mathrm{SST} = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2$$

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

$\hat{y}_i$ are the one-step-ahead forecasts

$\bar{y}_i$ is the mean y_i

If the model does not fit the series well, the model error sum of squares, SSE, might be larger than the total sum of squares, SST. As a result, R^2 can be negative.

RSquare Adj The adjusted R^2 is computed as follows:

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

MAPE The Mean Absolute Percentage Error is computed as follows:

$$\frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

MAE The Mean Absolute Error is computed as follows:

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

–2LogLikelihood Twice the negative log-likelihood function evaluated at the best-fit parameter estimates. Smaller values are better fits. See the Statistical Details appendix in the *Fitting Linear Models* book.

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 ϕ and θ operators are defined in the section "ARIMA Model" on page 274.

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, which are described in the sections on specific time series models. Each Parameter Estimates table contains the following columns:

Term The name of the parameter, which are described in the sections 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 (Shown only for multiplicative Seasonal ARIMA models.) The factor of the model that contains the parameter. In the multiplicative seasonal models, Factor 1 is nonseasonal and Factor 2 is seasonal.

Lag (Shown only for ARIMA and Seasonal ARIMA models.) The degree of the lag or backshift operator that is applied to the term to which the parameter is multiplied.

Estimate The parameter estimates of the time series model.

Std Error The estimates of the standard errors of the parameter estimates. These estimates are used to calculate tests and prediction intervals.

t Ratio The test statistics for the hypotheses that each parameter is zero. The test statistic for a parameter 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 for judging significance because it approximates the 0.05 significance level.

Prob>|t| The observed *p*-value calculated for each parameter. The *p*-value is the probability of getting, by chance alone, a *t*-ratio greater (in absolute value) than the computed value, given a true hypothesis.

Constant Estimate Shown for models that contain an intercept or mean term. The definition of the constant estimate is given under ARIMA model.

Mu (Shown only for ARIMA and Seasonal ARIMA models.) The estimate for the intercept value of an ARIMA or seasonal ARIMA model.

Forecast Plot

Each model has its own Forecast plot. The Forecast plot shows both the observed and predicted values for the time series. The plot is divided by a vertical line into two regions. To the left of the vertical line, the one-step-ahead forecasts are overlaid with the observed data points. To the right of the vertical line are the future values forecast by the model and the prediction intervals for the forecast.

You can control the number of future forecast values by changing the setting of the Forecast Periods option in the platform launch window or by selecting Number of Forecast Periods from the Time Series red triangle menu.

Residuals

The graphs in the Residuals report show the values of the residuals based on the fitted model. These values are the observed values of the time series minus the one-step-ahead predicted values. The autocorrelation and partial autocorrelation reports for these residuals are also shown. These reports can be used to determine whether the fitted model is adequate to describe the data. If the fitted model is adequate, the points in the residual plot should be normally distributed about zero 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 report is shown for each model, and it contains the value of the objective function at each iteration. This can be useful for diagnosing problems with the fitting procedure. Attempting to fit a model that is poorly suited to the data can result in a failure to converge on an optimum value for the likelihood. The Iteration History table contains the following quantities:

Iter The iteration number.

Iteration History The objective function value for each step.

Step The type of iteration step.

Obj-Criterion The norm of the gradient of the objective function.

Model Report Options

Each model report has a red triangle menu that contains the following options:

Show Points Shows or hides the data points in the forecast graph.

Show Prediction Interval Shows or hides the prediction intervals in the forecast graph.

Save Columns Creates a new data table that contains columns that represent 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 SAS code to SAS that duplicates the model analysis. If you are not connected to a SAS server, this option guides you through the connection process.

Residual Statistics Controls the displays of residual statistics are shown for the model. These displays are described in the section “Time Series Platform Options” on page 244. However, they are applied to the series of residuals.

Transfer Function Report

Each transfer function model is added to the Model Comparison table. If the **Report** checkbox for a transfer function model in the Model Comparison table is selected, a Transfer Function

Model report is produced. Each Transfer Function Model report contains the following reports:

- Model Summary
- Parameter Estimates
- Residuals
- Interactive Forecasting
- Iteration History

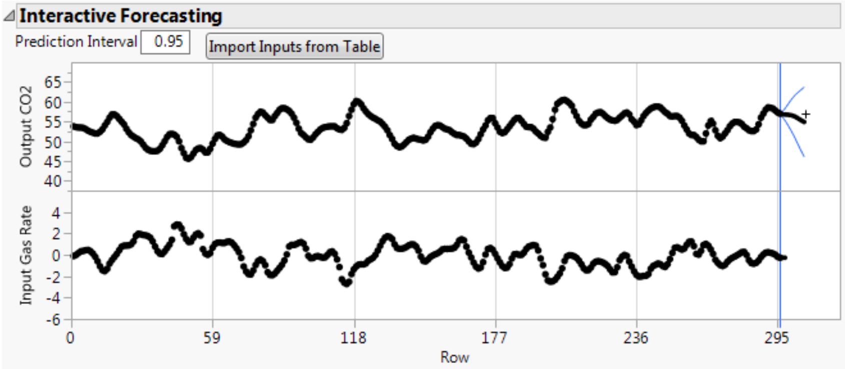
The information in the Model Summary, Parameter Estimates, Residuals, and Iteration History reports is the same as in the Time Series report. For more information about these reports, see "Model Report" on page 258. The Parameter Estimates table is followed by the formula of the model, where **B** is the backshift operator.

Interactive Forecasting

The Interactive Forecasting report provides a forecasting graph based on a specified prediction interval. The prediction interval around the prediction is shown in blue. Change the confidence level for this prediction interval by entering a number in the Prediction Interval box above the graph.

You can drag the plus sign in the graph to change the number of forecast periods in the graph. In the forecast periods, you can change the input values using the Import Inputs from Table button or by dragging points in the input graph to different values. The results of your changes are reflected in the forecast periods of the output graph.

Figure 15.14 Interactive Forecasting Graph

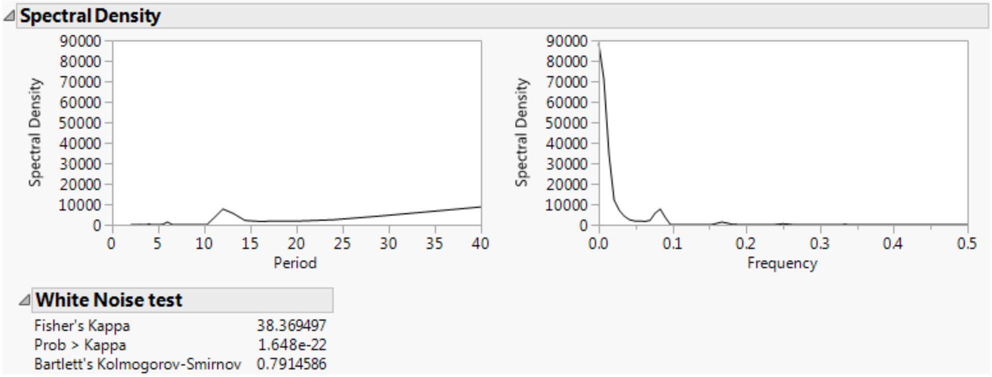


Interactive Forecasting

Prediction Interval 0.95

Spectral Density Report

Figure 15.15 Spectral Density Plots and White Noise Test Report



Fisher's Kappa	38.369497
Prob > Kappa	1.648e-22
Bartlett's Kolmogorov-Smirnov	0.7914586

The White Noise Test report contains the following statistics:

Fisher's Kappa 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 a periodic component. Kappa is the ratio of the maximum value of the periodogram, $I(f_i)$, and its average value.

Prob > Kappa The probability of observing a value larger than Kappa if the null hypothesis is true, given by the following equation:

$$Pr(k > \kappa) = 1 - \sum_{j=0}^{q} (-1)^j \binom{q}{j} \left[\max\left(1 - \frac{jk}{q}, 0\right)\right]^{q-1}$$

where

$q = N / 2$ if N is even, $q = (N - 1) / 2$ if N is odd

κ is the observed value of Kappa

The null hypothesis is rejected if this probability is less than the significance level α.

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 one typically rejects 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.

Additional Example of the Time Series Platform

This example uses the SeriesP.jmp sample data table to show how to perform a time series analysis. You first create a new column that is appropriate for the Time ID.

Create Appropriate Time ID Column

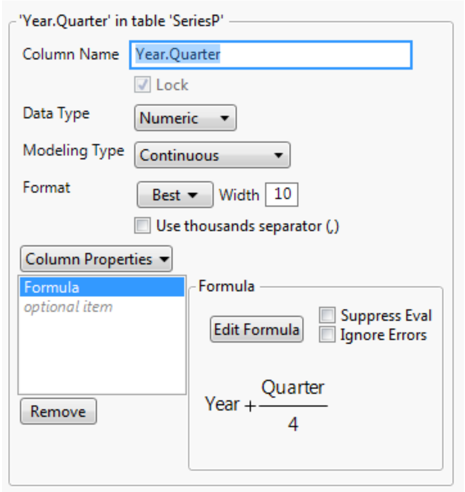
1. Select **Help > Sample Data Library** and open Time Series/SeriesP.jmp.

   The SeriesP.jmp data table contains a Year column and a Quarter column to identify the time period during which the responses were observed. However, the Time Series platform requires one column with unique, equally spaced time points to label the x-axis. If no Time ID is specified, then the row number is used to identify the time periods. To avoid this and make the report easier to interpret, you construct a Time ID column from Year and Quarter.

2. Select **Cols > New Columns**. In the Column Name box, type Year.Quarter.
3. Select **Column Properties > Formula.**
4. Select Year and then click the plus sign.
5. Select Quarter and then click the division sign. Type in 4 and press **Enter**.
6. Click **OK**.

   The completed New Column dialog box should appear as in Figure 15.16.

Figure 15.16 New Column



'Year.Quarter' in table 'SeriesP'

Column Name: Year.Quarter

☑ Lock

Data Type: Numeric

Modeling Type: Continuous

Format: Best Width 10

☐ Use thousands separator (,)

Formula

☐ Suppress Eval

☐ Ignore Errors

$$\text{Year} + \frac{\text{Quarter}}{4}$$

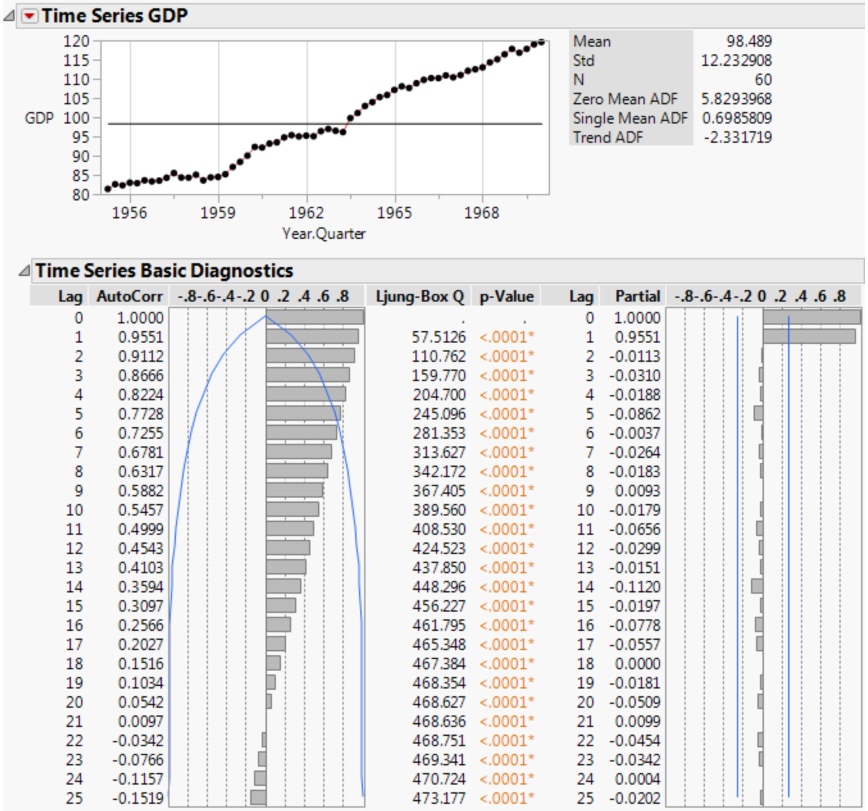
7. Click **OK**.

Time Series Analysis

Now that the data table contains an appropriate Time ID column, proceed with the analysis.

1. Select **Analyze > Specialized Modeling > Time Series**.
2. Select GDP and click **Y, Time Series**.
3. Select Year.Quarter and click **X, Time ID**.
4. Click **OK**.

Figure 15.17 Time Series Report for SeriesP.jmp



Time Series GDP

Mean	98.489
Std	12.232908
N	60
Zero Mean ADF	5.8293968
Single Mean ADF	0.6985809
Trend ADF	-2.331719

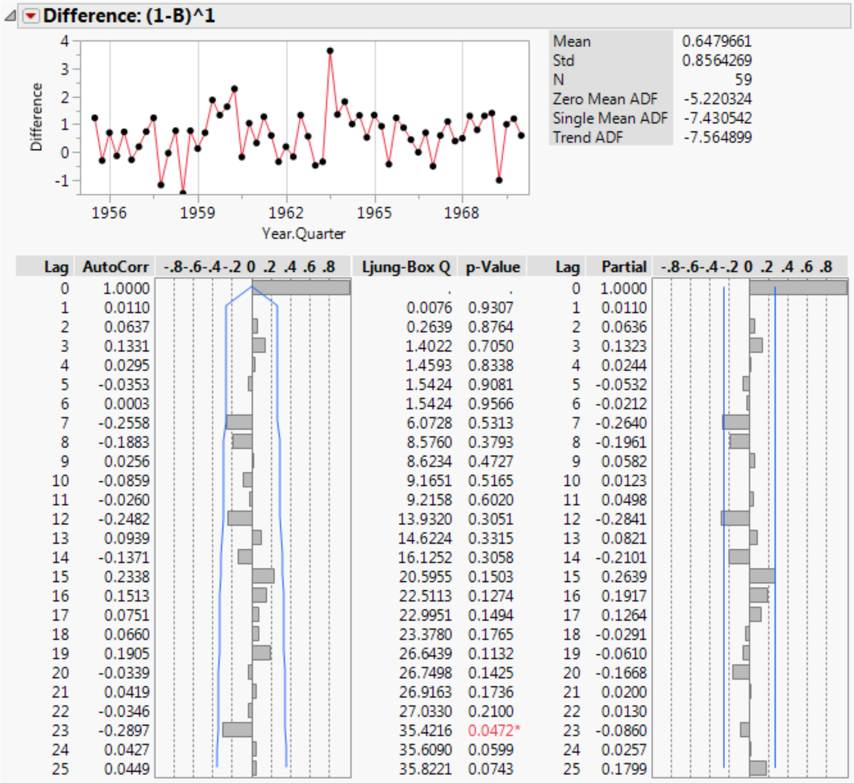
Time Series Basic Diagnostics

Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	0.9551	57.5126	<.0001*	1	0.9551
2	0.9112	110.762	<.0001*	2	-0.0113
3	0.8666	159.770	<.0001*	3	-0.0310
4	0.8224	204.700	<.0001*	4	-0.0188
5	0.7728	245.096	<.0001*	5	-0.0862
6	0.7255	281.353	<.0001*	6	-0.0037
7	0.6781	313.627	<.0001*	7	-0.0264
8	0.6317	342.172	<.0001*	8	-0.0183
9	0.5882	367.405	<.0001*	9	0.0093
10	0.5457	389.560	<.0001*	10	-0.0179
11	0.4999	408.530	<.0001*	11	-0.0656
12	0.4543	424.523	<.0001*	12	-0.0299
13	0.4103	437.850	<.0001*	13	-0.0151
14	0.3594	448.296	<.0001*	14	-0.1120
15	0.3097	456.227	<.0001*	15	-0.0197
16	0.2566	461.795	<.0001*	16	-0.0778
17	0.2027	465.348	<.0001*	17	-0.0557
18	0.1516	467.384	<.0001*	18	0.0000
19	0.1034	468.354	<.0001*	19	-0.0181
20	0.0542	468.627	<.0001*	20	-0.0509
21	0.0097	468.636	<.0001*	21	0.0099
22	-0.0342	468.751	<.0001*	22	-0.0454
23	-0.0766	469.341	<.0001*	23	-0.0342
24	-0.1157	470.724	<.0001*	24	0.0004
25	-0.1519	473.177	<.0001*	25	-0.0202

The series shows an increasing trend over time that is fairly linear. In addition, the autocorrelation chart shows that there is strong correlation between points that are close together. The AutoCorr values for points with lags of 1, 2, and 3 are 0.9551, 0.9112, 0.8666, respectively.

5. Click the Time Series GDP red triangle and select **Difference**.
6. Select 1 for the Nonseasonal Differencing Order and click **Estimate**.

Figure 15.18 Difference Report for SeriesP.jmp



Difference: (1-B)^1

Mean	0.6479661
Std	0.8564269
N	59
Zero Mean ADF	-5.220324
Single Mean ADF	-7.430542
Trend ADF	-7.564899

Lag	AutoCorr	-.8-.6-.4-.2 0 .2 .4 .6 .8	Ljung-Box Q	p-Value	Lag	Partial	-.8-.6-.4-.2 0 .2 .4 .6 .8
0	1.0000		.	.	0	1.0000	
1	0.0110		0.0076	0.9307	1	0.0110	
2	0.0637		0.2639	0.8764	2	0.0636	
3	0.1331		1.4022	0.7050	3	0.1323	
4	0.0295		1.4593	0.8338	4	0.0244	
5	-0.0353		1.5424	0.9081	5	-0.0532	
6	0.0003		1.5424	0.9566	6	-0.0212	
7	-0.2558		6.0728	0.5313	7	-0.2640	
8	-0.1883		8.5760	0.3793	8	-0.1961	
9	0.0256		8.6234	0.4727	9	0.0582	
10	-0.0859		9.1651	0.5165	10	0.0123	
11	-0.0260		9.2158	0.6020	11	0.0498	
12	-0.2482		13.9320	0.3051	12	-0.2841	
13	0.0939		14.6224	0.3315	13	0.0821	
14	-0.1371		16.1252	0.3058	14	-0.2101	
15	0.2338		20.5955	0.1503	15	0.2639	
16	0.1513		22.5113	0.1274	16	0.1917	
17	0.0751		22.9951	0.1494	17	0.1264	
18	0.0660		23.3780	0.1765	18	-0.0291	
19	0.1905		26.6439	0.1132	19	-0.0610	
20	-0.0339		26.7498	0.1425	20	-0.1668	
21	0.0419		26.9163	0.1736	21	0.0200	
22	-0.0346		27.0330	0.2100	22	0.0130	
23	-0.2897		35.4216	0.0472*	23	-0.0860	
24	0.0427		35.6090	0.0599	24	0.0257	
25	0.0449		35.8221	0.0743	25	0.1799	

The Difference report helps determine an appropriate model to be fit to the original time series. The plot of differences shows that the differenced series no longer has the trend observed in the original data. This indicates that lag-1 differencing is an appropriate choice. Also, even after removing the trend, the series shows no sign of seasonality. For these reasons, models to fit the original series should be able to handle linear trends, but do not necessarily need to handle seasonality. Linear exponential smoothing and ARIMA models would be appropriate.

7. Click the Time Series GDP red triangle and select **Smoothing Model > Linear Exponential Smoothing**.

8. Click **Estimate**.

9. Click the Time Series GDP red triangle and select **ARIMA Model Group**. This enables you to fit multiple ARIMA models for a range of values of (*p*,*d*,*q*)(*P*,*D*,*Q*).

10. In the ARIMA box, set the following ranges:

    – Fix *d*, the differencing order, at 1 by setting the range from 1 to 1 because the differencing report showed lag-1 differencing was appropriate.

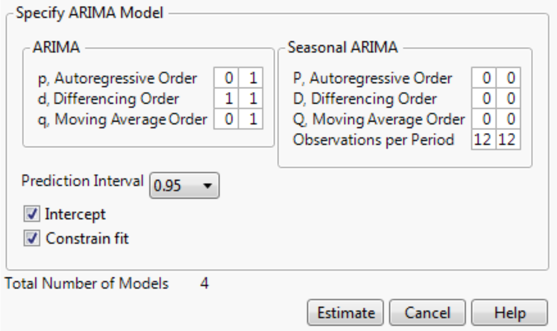
- Set *p*, the autoregressive order, to range from 0 to 1 because the original series showed evidence of autocorrelation.
- Set *q*, the moving average order, to range from 0 to 1.

Note: In most cases, it is sufficient to keep *p* and *q* small.

- Leave *P*, *D*, and *Q* set at 0, since the series showed no evidence of seasonality.

These settings lead to the fitting of 4 total models.

Figure 15.19 ARIMA Model Group Specification



Specify ARIMA Model

ARIMA

p, Autoregressive Order	0	1
d, Differencing Order	1	1
q, Moving Average Order	0	1

Seasonal ARIMA

P, Autoregressive Order	0	0
D, Differencing Order	0	0
Q, Moving Average Order	0	0
Observations per Period	12	12

Prediction Interval 0.95

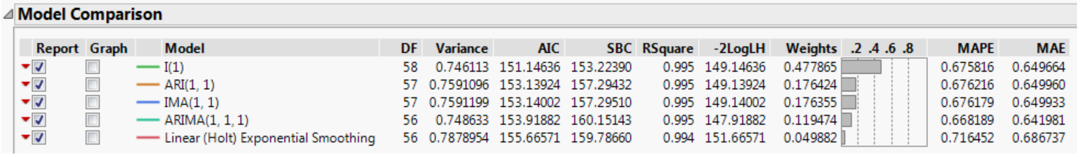
☑ Intercept

☑ Constrain fit

Total Number of Models 4

11. Click **Estimate**.

Figure 15.20 Model Comparison Table

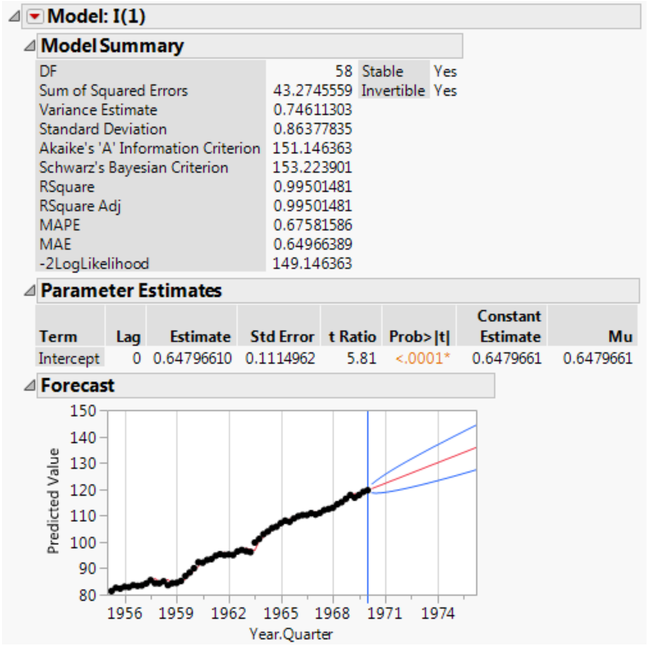


Model Comparison

Report	Graph	Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	Weights	.2 .4 .6 .8	MAPE	MAE
☑	□	I(1)	58	0.746113	151.14636	153.22390	0.995	149.14636	0.477865		0.675816	0.649664
☑	□	ARI(1, 1)	57	0.7591096	153.13924	157.29432	0.995	149.13924	0.176424		0.676216	0.649960
☑	□	IMA(1, 1)	57	0.7591199	153.14002	157.29510	0.995	149.14002	0.176355		0.676179	0.649933
☑	□	ARIMA(1, 1, 1)	56	0.748633	153.91882	160.15143	0.995	147.91882	0.119474		0.668189	0.641981
☑	□	Linear (Holt) Exponential Smoothing	56	0.7878954	155.66571	159.78660	0.994	151.66571	0.049882		0.716452	0.686737

The Model Comparison Table is sorted such that the best fitting model, according to the AIC criterion, is at the top of the list. In this case, the ARIMA(0,1,0) model (denoted I(1) in the report) best fits the original time series. It should also be noted that although the I(1) model is "best," all of the models have extremely similar values for the fit statistics. They could all be considered appropriate.

Figure 15.21 Model Report for ARIMA(0,1,0)



Model: I(1)

Model Summary

DF	58	Stable	Yes
Sum of Squared Errors	43.2745559	Invertible	Yes
Variance Estimate	0.74611303		
Standard Deviation	0.86377835		
Akaike's 'A' Information Criterion	151.146363		
Schwarz's Bayesian Criterion	153.223901		
RSquare	0.99501481		
RSquare Adj	0.99501481		
MAPE	0.67581586		
MAE	0.64966389		
-2LogLikelihood	149.146363		

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob>|t|	Constant Estimate	Mu
Intercept	0	0.64796610	0.1114962	5.81	<.0001*	0.6479661	0.6479661

Forecast

The model report for I(1) shows the forecast graph. The blue lines indicate the prediction intervals. GDP is predicted to continue increasing at a linear rate.

Statistical Details for the Time Series Platform

This section contains details for the following parts of the Time Series platform:

- "Statistical Details for Spectral Density" on page 269
- "Statistical Details for X-11 Decomposition" on page 270
- "Statistical Details for Smoothing Models" on page 270
- "Statistical Details for ARIMA Models" on page 274
- "Statistical Details for Transfer Functions" on page 275

Statistical Details for Spectral Density

The least squares estimates of the coefficients of the Fourier series are computed as follows:

$$a_t = \frac{2}{N}\sum_{i=1}^{N} y_t \cos(2\pi f_i t)$$

$$b_t = \frac{2}{N}\sum_{i=1}^{N} y_t \sin(2\pi f_i t)$$

Then the $f_i = i/N$ are combined to form the periodogram $I(f_i) = \frac{N}{2}(a_i^2 + b_i^2)$, which represents the intensity at frequency f_i.

The periodogram is then smoothed and scaled by 1/(4π) to form the spectral density.

Statistical Details for X-11 Decomposition

This method adjusts the original time series using either a multiplicative or an additive decomposition. The model is fit using an iterative process to estimate the three X-11 components: trend cycle, seasonal, and irregular. The trend cycle component contains both the long-term trend and the long-term cyclical effects. The irregular component contains the effects of variation unexplained by the trend and seasonal components. For a historical overview of the development of the X-11 method, see *SAS/ETS 13.1 User's Guide* (search for "Historical Development of X-11").

The multiplicative adjustment fits the following model:

$$O_t = C_t \cdot S_t \cdot I_t$$

where

O_t is the original time series

C_t is the trend cycle component

S_t is the seasonal component

I_t is the irregular component

The adjusted multiplicative trend is O_t/S_t.

The additive adjustment fits the following model:

$$O_t = C_t + S_t + I_t$$

The adjusted additive trend is $O_t - S_t$.

Statistical Details for Smoothing Models

Smoothing models are defined as follows:

$$y_t = \mu_t + \beta_t t + s(t) + a_t$$

where

μ_t is the time-varying mean term

β_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 defined as follows:

L_t is a smoothed level that estimates μ_t

T_t is a smoothed trend that estimates β_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 describe the evolution of these estimators. The smoothing equations are written in terms of model parameters called *smoothing weights*:

α is the level smoothing weight
γ is the trend smoothing weight
φ is the trend damping weight
δ is 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.

Simple Exponential Smoothing

The model for simple exponential smoothing is $y_t = \mu_t + \alpha_t$.

The smoothing equation, $L_t = \alpha y_t + (1 - \alpha)L_{t\text{-}1}$, is defined in terms of a single smoothing weight α. This model is equivalent to an ARIMA(0, 1, 1) model where the following is true:

$$(1 - \mathrm{B})y_t = (1 - \theta\mathrm{B})\alpha_t \text{ where } \theta = 1 - \alpha$$

The moving average form of the model is defined as follows:

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

Double (Brown) Exponential Smoothing

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

The smoothing equations, defined in terms of a single smoothing weight α, are defined as follows:

$$L_t = \alpha y_t + (1-\alpha)L_{t-1}$$

$$T_t = \alpha(L_t - L_{t-1}) + (1-\alpha)T_{t-1}$$

This model is equivalent to an ARIMA(0, 1, 1)(0, 1, 1)$_1$ model where the following is true:

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

The moving average form of the model is defined as follows:

$$y_t = a_t + \sum_{j=1}^{\infty} (2\alpha + (j-1)\alpha^2)a_{t-j}$$

Linear (Holt) Exponential Smoothing

The model for linear exponential smoothing is $y_t = \mu_t + \beta_t t + a_t$.

The smoothing equations, in terms of smoothing weights α and γ, are defined as follows:

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1}$$

This model is equivalent to an ARIMA(0, 2, 2) model where the following is true:

$$(1-\mathrm{B})^2 y_t = (1-\theta\mathrm{B}-\theta_2\mathrm{B}^2)a_t \text{ with } \theta = 2-\alpha-\alpha\gamma \text{ and } \theta_2 = \alpha-1$$

The moving average form of the model is defined as follows:

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

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 α, γ, and φ, are defined as follows:

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + \varphi T_{t-1})$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)\varphi T_{t-1}$$

This model is equivalent to an ARIMA(1, 1, 2) model where the following is true:

$$(1-\varphi\mathrm{B})(1-\mathrm{B})y_t = (1-\theta_1\mathrm{B}-\theta_2\mathrm{B}^2)a_t$$

where

$$\theta_1 = 1 + \varphi - \alpha - \alpha\gamma\varphi$$

$$\theta_2 = (\alpha - 1)\varphi$$

The moving average form of the model is defined as follows:

$$y_t = \alpha_t + \sum_{j=1}^{\infty}\left(\frac{\alpha + \alpha\gamma\varphi(\varphi^j - 1)}{\varphi - 1}\right)\alpha_{t-j}$$

Seasonal Exponential Smoothing

The model for seasonal exponential smoothing is $y_t = \mu_t + s(t) + a_t$.

The smoothing equations in terms of smoothing weights α and δ are defined as follows:

$$L_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)L_{t-1}$$

$$S_t = \delta(y_t - L_{t-s}) + (1 - \delta)S_{t-s}$$

This model is equivalent to a seasonal ARIMA(0, 1, s+1)(0, 1, 0)$_s$ model:

$$(1 - B)(1 - B^s)y_t = (1 - \theta_1 B - \theta_2 B^2 - \theta_3 B^{s+1})a_t$$

where

$$\theta_1 = 1 - \alpha$$

$$\theta_2 = (1 - \delta)(1 - \alpha)$$

$$\theta_3 = (1 - \alpha)(\delta - 1)$$

The moving average form of the model is defined as follows:

$$y_t = a_t + \sum_{j=1}^{\infty}\psi_j a_{t-j} \text{ where } \psi = \begin{cases} \alpha \text{ for } j \bmod s \neq 0 \\ \alpha + \delta(1 - \alpha) \text{ for} j \bmod s = 0 \end{cases}$$

Winters Method (Additive)

The model for the additive version of the Winters method is $y_t = \mu_t + \beta_t t + s(t) + a_t$.

The smoothing equations in terms of weights α, γ, and δ are defined as follows:

$$L_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$$

$$S_t = \delta(y_t - L_t) + (1 - \delta)S_{t-s}$$

This model is equivalent to a seasonal ARIMA(0, 1, *s*+1)(0, 1, 0)s model defined as follows:

$$(1-\mathrm{B})(1-\mathrm{B}^2)y_t = \left(1-\sum_{i=1}^{s+1}\theta_i\mathrm{B}^i\right)a_t$$

The moving average form of the model is defined as follows:

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

where

$$\psi = \begin{cases}\alpha + j\alpha\gamma\ , & j\bmod s \neq 0\\ \alpha + j\alpha\gamma + \delta(1-\alpha)\ , & j\bmod s = 0\end{cases}$$

Statistical Details for ARIMA Models

ARIMA Model

For a response series $\{y_i\}$, the general form for the ARIMA model is as follows:

$$\phi(\mathrm{B})(w_t - \mu) = \theta(\mathrm{B})a_t$$

where

t is the time index

B is the backshift operator defined as $\mathrm{B}y_t = y_{t-1}$

$w_t = (1-\mathrm{B})^d y_t$ is the response series after differencing

μ is the intercept or mean term

$\phi(\mathrm{B})$ and $\theta(\mathrm{B})$ are the autoregressive operator and the moving average operator, respectively, and are written as follows:

$$\phi(\mathrm{B}) = 1-\phi_1\mathrm{B}-\phi_2\mathrm{B}^2-\ldots-\phi_p\mathrm{B}^p$$

$$\theta(\mathrm{B}) = 1-\theta_1\mathrm{B}-\theta_2\mathrm{B}^2-\ldots-\theta_q\mathrm{B}^q$$

where

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 follows:

$$\phi(\mathrm{B})w_t = \delta + \theta(\mathrm{B})a_t$$

The constant estimate δ is given by the relation:

$$\delta = \phi(\mathrm{B})\mu = \mu - \phi_1\mu - \phi_2\mu - \ldots - \phi_p\mu$$

Seasonal ARIMA Model

In the case of **Seasonal ARIMA** modeling, the differencing, autoregressive, and moving average operators are the product of seasonal and nonseasonal polynomials:

$$w_t = (1-\mathrm{B})^d(1-\mathrm{B}^s)^D y_t$$

$$\varphi(\mathrm{B}) = (1 - \varphi_{1,1}\mathrm{B} - \varphi_{1,2}\mathrm{B}^2 - \ldots - \varphi_{1,p}\mathrm{B}^p)(1 - \varphi_{2,s}\mathrm{B}^s - \varphi_{2,2s}\mathrm{B}^{2s} - \ldots - \varphi_{2,Ps}\mathrm{B}^{Ps})$$

$$\theta(\mathrm{B}) = (1 - \theta_{1,1}\mathrm{B} - \theta_{1,2}\mathrm{B}^2 - \ldots - \theta_{1,q}\mathrm{B}^q)(1 - \theta_{2,s}\mathrm{B}^s - \theta_{2,2s}\mathrm{B}^{2s} - \ldots - \theta_{2,Qs}\mathrm{B}^{Qs})$$

where s is the number of observations per period. 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.

Statistical Details for Transfer Functions

A typical transfer function model with m inputs can be represented as follows:

$$Y_t - \mu = \frac{\omega_1(B)}{\delta_1(B)}X_{1,\,t-d1} + \ldots + \frac{\omega_m(B)}{\delta_m(B)}X_{m,\,m-dm} + \frac{\theta(B)}{\varphi(B)}e_t$$

where

Y_t denotes the output series

X_1 to X_m denote m input series

e_t represents the noise series

$X_{1,\,t\text{–}d1}$ indicates the series X_1 is indexed by t with a d1-step lag

μ represents the mean level of the model

φ(B) and θ(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 to include in the model.

Chapter **16**

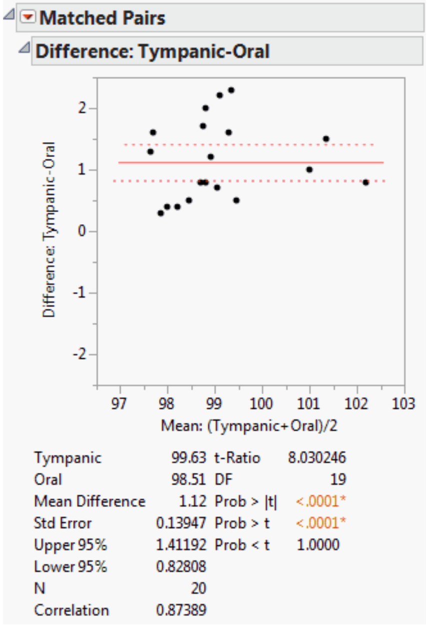
Matched Pairs Analysis

Compare Measurements on the Same Subject

The Matched Pairs platform compares the means between two or more correlated variables and assesses the differences. For example, you might compare a blood pressure measurement taken on the same subject before a treatment and again after the treatment. A statistical method called the *paired t-test* takes the correlated responses into account.

The platform produces a graph of the paired differences by the paired means, and the paired *t*-test results for all three alternative hypotheses. Additional features provide for more than two matched responses and for a grouping column to test across samples, in a simple version of repeated measures analysis.

Figure 16.1 Example of Matched Pairs Analysis



```
Tympanic            99.63  t-Ratio    8.030246
Oral                98.51  DF               19
Mean Difference      1.12  Prob > |t|  <.0001*
Std Error         0.13947  Prob > t    <.0001*
Upper 95%         1.41192  Prob < t     1.0000
Lower 95%         0.82808
N                      20
Correlation       0.87389
```

Overview of the Matched Pairs Platform

The Matched Pairs platform compares row-by-row differences between two response columns using a paired *t*-test. Often, the two columns represent measurements on the same subject before and after some treatment. Alternatively, the measurements could represent data taken on the same subject with two different instruments.

If you have paired data arranged in two data table columns, then you are ready to use the Matched Pairs platform. However, if all of your measurements are in a single column, then perform one of the following tasks:

- Use the **Split** option in the **Tables** menu to split the column of measurements into two columns. Then you can use the Matched Pairs platform.
- For two response columns, create a third column that calculates the difference between the two responses. Then test that the mean of the difference column is zero with the Distribution platform.
- For the two responses stored in a single column, you can do a two-way analysis of variance. One factor (the ID variable) identifies the two responses and the other factor identifies the subject. Use the Fit Y by X Oneway platform with a blocking variable (the subject column), or use the Fit Model platform to do a two-way ANOVA. The test on the ID factor is equivalent to the paired *t*-test.

Note: If the data are paired, do not do a regular independent *t*-test. Do not stack the data into one column and use the Fit Y by X One-way ANOVA on the ID without specifying a block variable. To do this has the effect of ignoring the correlation between the responses. This causes the test to overestimate the effect if responses are negatively correlated, or to underestimate the effect if responses are positively correlated.

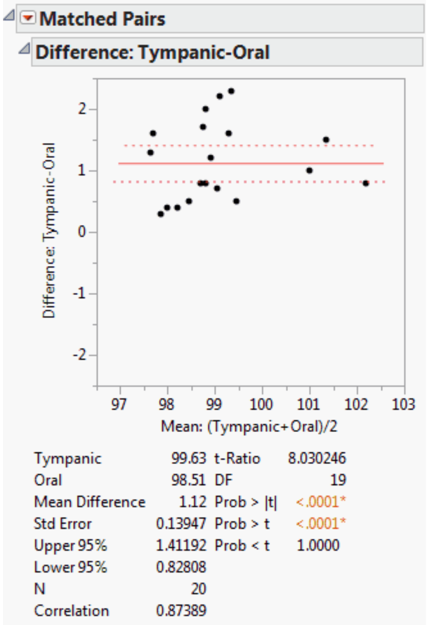
Example of Comparing Matched Pairs

This example uses the Therm.jmp sample data table. The data contains temperature measurements on 20 people. Temperature is measured using two types of thermometers: oral and tympanic (ear). You want to determine whether the two types of thermometers produce equal temperature readings. Note that the differences in temperature between the different people are not important. The matched pairs analysis is testing the differences between the thermometers.

1. Select **Help > Sample Data Library** and open Therm.jmp.
2. Select **Analyze > Specialized Modeling > Matched Pairs**.
3. Select Oral and Tympanic and click **Y, Paired Response**.
4. Click **OK**.

The report window appears.

Figure 16.2 The Matched Pairs Report Window



```
Tympanic          99.63  t-Ratio     8.030246
Oral              98.51  DF                19
Mean Difference    1.12  Prob > |t|   <.0001*
Std Error       0.13947  Prob > t     <.0001*
Upper 95%       1.41192  Prob < t     1.0000
Lower 95%       0.82808
N                    20
Correlation     0.87389
```

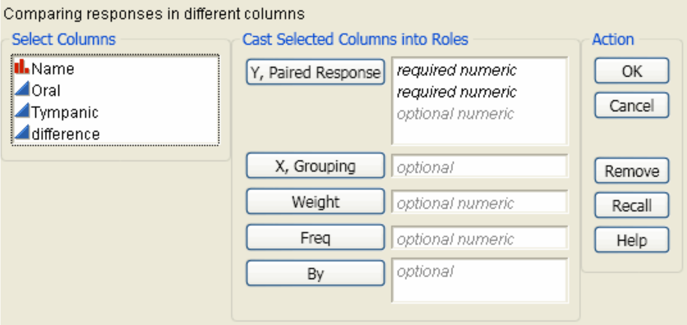
The results show that, on average, the tympanic thermometer measures 1.12 degrees higher than the oral thermometer. The small p-value (Prob > |t|) indicates that this difference is statistically significant, and not due to chance.

Note that this matched pairs analysis does not indicate which thermometer is correct (if either), but only indicates that there is a difference between the thermometers.

Launch the Matched Pairs Platform

Launch the Matched Pairs platform by selecting **Analyze > Specialized Modeling > Matched Pairs**.

Figure 16.3 The Matched Pairs Launch Window



Y, Paired Response Provide the two response columns. For information about analyzing more than two responses, see "Multiple Y Columns" on page 280.

X, Grouping Provide a grouping variable to compare the differences across groups. For more information, see "Across Groups" on page 282.

Weight Identifies one column whose numeric values assign a weight to each row in the analysis.

Freq Identifies one column whose numeric values assign a frequency to each row in the analysis.

By Performs a separate matched pairs analysis for each level of the By variable.

For more information about the launch window, see the Get Started chapter in the *Using JMP* book.

After you click **OK**, the Matched Pairs report window appears. See "The Matched Pairs Report" on page 281.

Multiple Y Columns

You can have more than two responses. If the number of responses is odd, all possible pairs are analyzed. The following table shows an example for three responses.

Y1 by Y2	Y1 by Y3
	Y2 by Y3

If the number of responses is even, the Matched Pairs platform asks whether you want to do all possible pairs. If you do not do all possible pairs, adjacent responses are analyzed as a pair. The following table shows the arrangement of analyses for four responses.

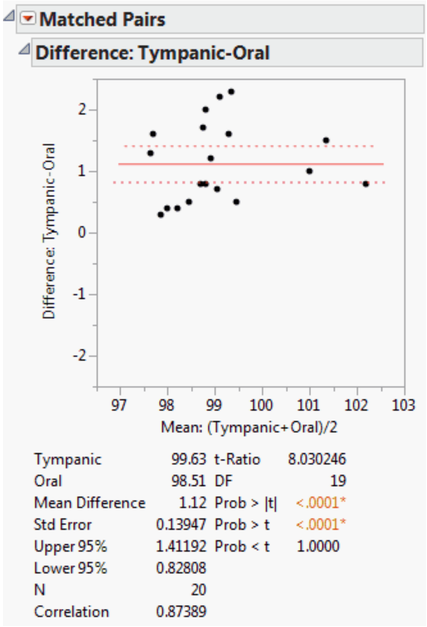
Y1 by Y2	Y3 by Y4

The Matched Pairs Report

Follow the instructions in “Example of Comparing Matched Pairs” on page 278 to produce the report window shown in Figure 16.4.

The Matched Pairs report shows a Tukey mean-difference plot, summary statistics, and the results of the paired t-test. See “Difference Plot and Report” on page 282. If you specified an **X, Grouping** variable, the report also includes the Across Groups report. See “Across Groups” on page 282.

Figure 16.4 Example of Matched Pairs Report



```
Tympanic           99.63  t-Ratio      8.030246
Oral               98.51  DF                 19
Mean Difference     1.12  Prob > |t|    <.0001*
Std Error        0.13947  Prob > t      <.0001*
Upper 95%        1.41192  Prob < t       1.0000
Lower 95%        0.82808
N                     20
Correlation      0.87389
```

Note: The red triangle menu provides additional options that can add reports to the initial report window. See “Matched Pairs Platform Options” on page 282.

Difference Plot and Report

The Difference plot shows differences by means. In the Difference plot, note the following:

- The mean difference is shown as the horizontal line, with the 95% confidence interval above and below shown as dotted lines. If the confidence region includes zero, then the means are not significantly different at the 0.05 level. In this example the difference is significant.
- If you add a reference frame, the mean of pairs is shown by the vertical line. For details about a reference frame, see "Matched Pairs Platform Options" on page 282.

The Difference report shows the mean of each response, the difference of the means, and a confidence interval for the difference. The Difference report also shows the results of the paired t-test.

Across Groups

Note: The Across Groups report appears only if you have specified an **X, Grouping** variable.

The Across Groups analysis corresponds to a simple repeated measures analysis. (You can get the same test results using the **Manova** personality of the Fit Model platform.)

Mean Difference Shows the mean of the difference across rows in each group between the two paired columns. In other words, this is the within-subject by across-subject interaction, or split-plot by whole-plot interaction.

Mean Mean Shows the mean of the mean across rows in each group across the two paired columns. In other words, this is the across-subject or whole-plot effect.

Test Across Groups Two *F*-tests determine whether the across-groups values are different:

- **Mean Difference** tests that the change across the pair of responses is different in different groups.
- **Mean Mean** tests that the average response for a subject is different in different groups

Related Information

- "Example Comparing Matched Pairs across Groups" on page 283

Matched Pairs Platform Options

The Matched Pairs red triangle menu contains the following options:

Plot Dif by Mean Shows or hides the plot of the paired differences by paired means. For a detailed description of this plot, see "Difference Plot and Report" on page 282.

Plot Dif by Row Shows or hides the plot of paired differences by row number.

Reference Frame Shows or hides the reference frame on the Plot Dif by Mean plot. A rectangle showing where a plot of Y2 by Y1 would be located inside the plot, tilted and possibly squished. A vertical red line is shown representing the mean of means. The reference frame is shown initially when the range of the differences is greater than half the range of the data.

Wilcoxon Signed Rank Shows or hides the Wilcoxon signed rank test. The Wilcoxon signed rank test is applied to the paired differences. It is a nonparametric test that compares the sizes of the positive differences to the sizes of the negative differences. The test uses the Pratt method to address zero differences. The test also assumes that the distribution of differences is symmetric. For details, see the Distributions chapter in the *Basic Analysis* book. See also Lehman (2006), Conover (1999, page 350), and Cureton (1967).

Sign Test Shows or hides the sign test. This is a nonparametric version of the paired t-test that uses only the sign (positive or negative) of the difference for the test.

Set α Level Changes the alpha level used in the analyses. Affects the confidence intervals in the report and on the plot.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Example Comparing Matched Pairs across Groups

This example uses the Dogs.jmp sample data table. This example shows you how to produce both a Matched Pairs Across Groups report and the corresponding MANOVA report using Fit Model.

1. Select **Help > Sample Data Library** and open Dogs.jmp.
2. Select **Analyze > Specialized Modeling > Matched Pairs**.
3. Select LogHist0 and LogHist1 and click **Y, Paired Response**.

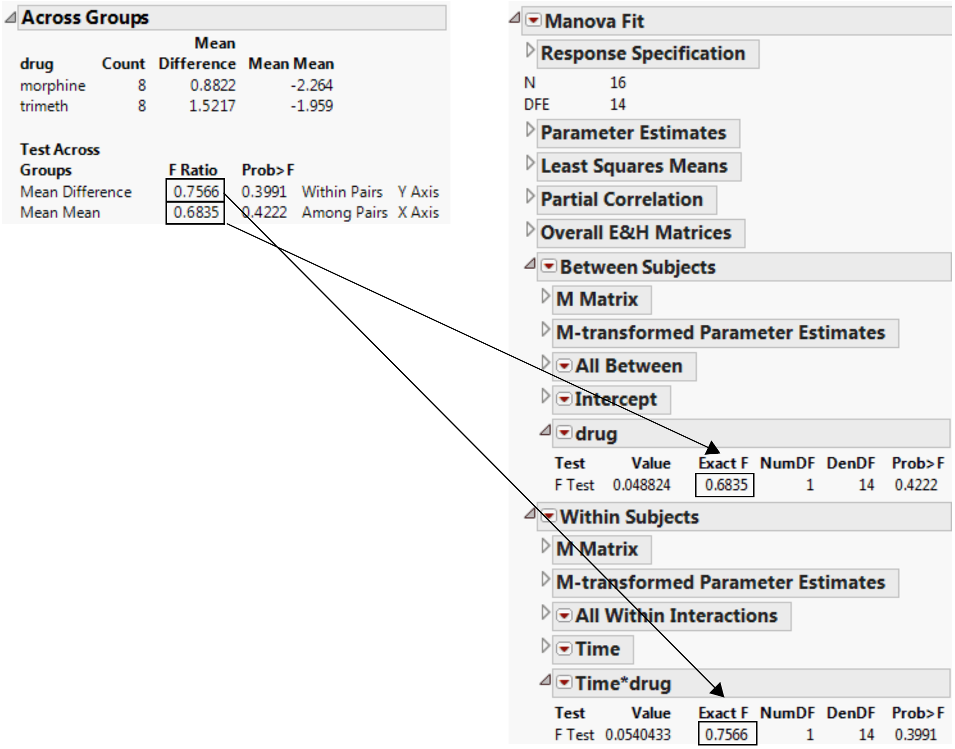
4. Select drug and click **X, Grouping**.
5. Click **OK**.

The report on the left in Figure 16.5 appears.

Now produce the Fit Model report using the same data table.

1. Select **Analyze > Fit Model**.
2. Select LogHist0 and LogHist1 and click **Y**.
3. Select drug and click **Add**.
4. Select the **Manova** personality.
5. Click **Run Model**.
6. In the Response Specification report, select **Repeated Measures** from the **Choose Response** menu.
7. Click **OK**.

Figure 16.5 Examples of Matched Pairs Across Groups and Fit Model MANOVA with Repeated Measures



Across Groups

drug	Count	Mean Difference	Mean Mean
morphine	8	0.8822	-2.264
trimeth	8	1.5217	-1.959

Test Across Groups	F Ratio	Prob>F		
Mean Difference	0.7566	0.3991	Within Pairs	Y Axis
Mean Mean	0.6835	0.4222	Among Pairs	X Axis

Manova Fit

Response Specification

N	16
DFE	14

Parameter Estimates

Least Squares Means

Partial Correlation

Overall E&H Matrices

Between Subjects

M Matrix

M-transformed Parameter Estimates

All Between

Intercept

drug

Test	Value	Exact F	NumDF	DenDF	Prob>F
F Test	0.048824	0.6835	1	14	0.4222

Within Subjects

M Matrix

M-transformed Parameter Estimates

All Within Interactions

Time

Time*drug

Test	Value	Exact F	NumDF	DenDF	Prob>F
F Test	0.0540433	0.7566	1	14	0.3991

The F Ratio for the Mean Difference in the Across Groups report corresponds to the F Ratio for Time*drug under the Within Subjects report. The F Ratio for the Mean Mean in the Across Groups report corresponds to the F Ratio for drug under Between Subjects in the Manova Fit report.

Statistical Details for the Matched Pairs Platform

This section contains statistical details for Matched Pairs analyses.

Graphics for Matched Pairs

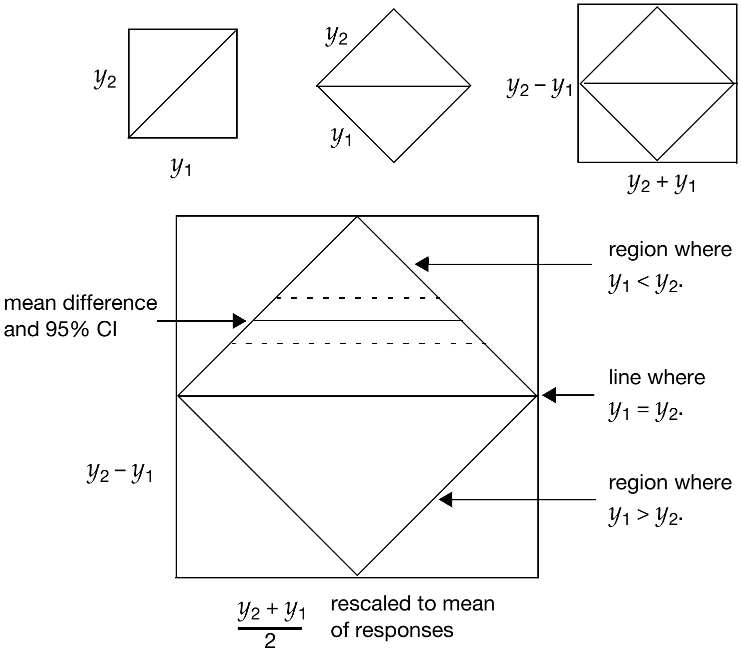
The primary graph in the platform is a Tukey mean-difference (Cleveland 1994, p. 130), which plots the difference of the two responses on the *y*-axis against the mean of the two responses on the *x*-axis. This graph is the same as a scatterplot of the two original variables, but turned 45 degrees. A 45 degree rotation and rescaling turns the original coordinates into a difference and a mean.

Figure 16.6 Example of Transforming to Difference by Mean, Rotated by 45 Degrees

Before rotation, the axes represent y_1 and y_2.

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} y_2 + y_1 \\ y_2 - y_1 \end{bmatrix}$$

After rotation, the axes represent a sum and difference.

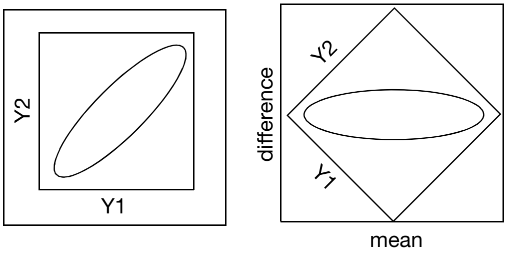

y_2
y_1
y_2
y_1
$y_2 - y_1$
$y_2 + y_1$
region where $y_1 < y_2$.
mean difference and 95% CI
line where $y_1 = y_2$.
$y_2 - y_1$
region where $y_1 > y_2$.
$\frac{y_2 + y_1}{2}$
rescaled to mean of responses

Correlation of Responses

In most cases where the pair of measurements is taken from the same individual at different times, they are positively correlated. However, if they represent competing responses, the correlation can be negative.

Figure 16.7 shows how the positive correlation of the two responses becomes the small variance on the difference (the y-axis). If the correlation is negative, the ellipse is oriented in the other direction and the variance of the rotated graph is large on the y-axis.

Figure 16.7 Examples of Positive Correlation Before and After Rotation



Chapter 17

Response Screening

Test Many Responses in Large-Scale Data

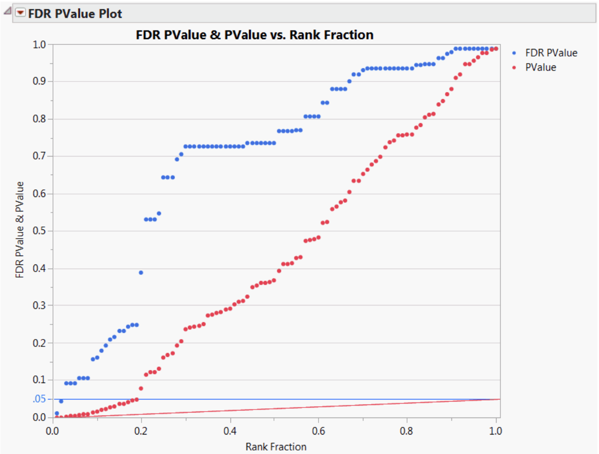
The analysis of large-scale data sets, where hundreds or thousands of measurements are taken on a part or an organism, requires innovative approaches. But testing many responses for the effects of factors can be challenging, if not misleading, without appropriate methodology.

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 17.1 Example of a Response Screening Plot



Response Screening Platform Overview

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 17.1. 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; Westfall et al., 2011).

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 17.1 Analyses Performed by Response Screening

Response	Factor	Fit Y by X Analysis	Description
Continuous	Categorical	Oneway	Analysis of Variance
Continuous	Continuous	Bivariate	Simple Linear Regression
Categorical	Categorical	Contingency	Chi-Square
Categorical	Continuous	Logistic	Simple Logistic Regression

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 Summarize YByX in 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. Select **Help > Sample Data Library** and open Probe.jmp.
2. Select **Analyze > Screening > 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. Enter 100 in the **MaxLogWorth** box.

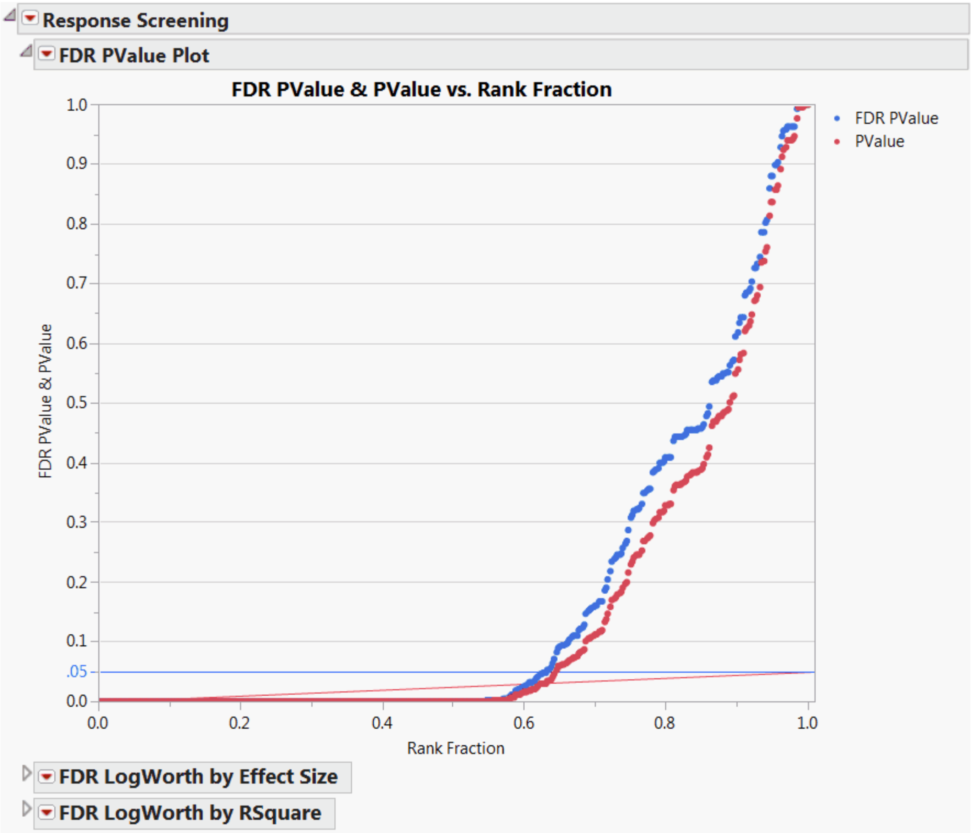
   A log worth of 100 or larger corresponds to an extremely small *p*-value. Setting a value for the MaxLogWorth helps control the scale of plots.
6. Click **OK**.

The Response Screening report appears, along with a data table of supporting information. The report (Figure 17.2) 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 17.2 Response Screening Report for 387 Tests against Process

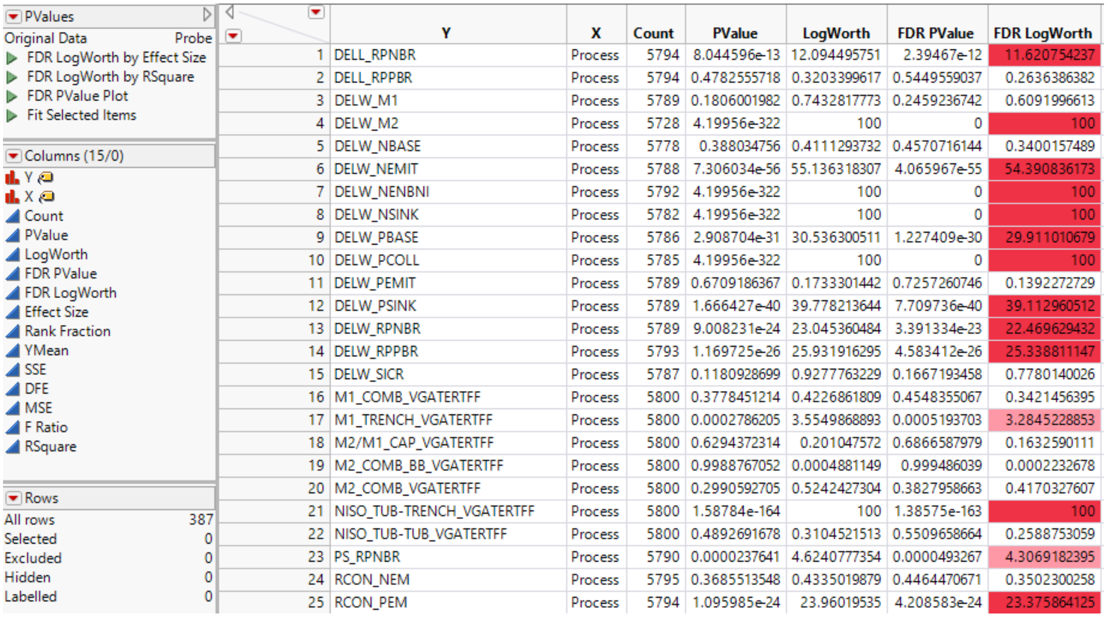


The FDR PValue Plot shows that more than 60% of the tests are significant. A handful of tests are significant using the usual *p*-value, but not significant using the FDR *p*-value. These tests correspond to the red points that are above the red line, but below the blue line.

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 17.3) 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.

Figure 17.3 PValues Data Table, Partial View



PValues
Original Data Probe
▶ FDR LogWorth by Effect Size
▶ FDR LogWorth by RSquare
▶ FDR PValue Plot
▶ Fit Selected Items

Columns (15/0)
Y
X
Count
PValue
LogWorth
FDR PValue
FDR LogWorth
Effect Size
Rank Fraction
YMean
SSE
DFE
MSE
F Ratio
RSquare

Rows
All rows 387
Selected 0
Excluded 0
Hidden 0
Labelled 0

	Y	X	Count	PValue	LogWorth	FDR PValue	FDR LogWorth
1	DELL_RPNBR	Process	5794	8.044596e-13	12.094495751	2.39467e-12	11.620754237
2	DELL_RPPBR	Process	5794	0.4782555718	0.3203399617	0.5449559037	0.2636386382
3	DELW_M1	Process	5789	0.1806001982	0.7432817773	0.2459236742	0.6091996613
4	DELW_M2	Process	5728	4.19956e-322	100	0	100
5	DELW_NBASE	Process	5778	0.388034756	0.4111293732	0.4570716144	0.3400157489
6	DELW_NEMIT	Process	5788	7.306034e-56	55.136318307	4.065967e-55	54.390836173
7	DELW_NENBNI	Process	5792	4.19956e-322	100	0	100
8	DELW_NSINK	Process	5782	4.19956e-322	100	0	100
9	DELW_PBASE	Process	5786	2.908704e-31	30.536300511	1.227409e-30	29.911010679
10	DELW_PCOLL	Process	5785	4.19956e-322	100	0	100
11	DELW_PEMIT	Process	5789	0.6709186367	0.1733301442	0.7257260746	0.1392272729
12	DELW_PSINK	Process	5789	1.666427e-40	39.778213644	7.709736e-40	39.112960512
13	DELW_RPNBR	Process	5789	9.008231e-24	23.045360484	3.391334e-23	22.469629432
14	DELW_RPPBR	Process	5793	1.169725e-26	25.931916295	4.583412e-26	25.33881147
15	DELW_SICR	Process	5787	0.1180928699	0.9277763229	0.1667193458	0.7780140026
16	M1_COMB_VGATERTFF	Process	5800	0.3778451214	0.4226861809	0.4548355067	0.3421456395
17	M1_TRENCH_VGATERTFF	Process	5800	0.0002786205	3.5549868893	0.0005193703	3.2845228853
18	M2/M1_CAP_VGATERTFF	Process	5800	0.6294372314	0.201047572	0.6866587979	0.1632590111
19	M2_COMB_BB_VGATERTFF	Process	5800	0.9988767052	0.0004881149	0.999486039	0.0002232678
20	M2_COMB_VGATERTFF	Process	5800	0.2990592705	0.5242427304	0.3827958663	0.4170327607
21	NISO_TUB-TRENCH_VGATERTFF	Process	5800	1.58784e-164	100	1.38575e-163	100
22	NISO_TUB-TUB_VGATERTFF	Process	5800	0.4892691678	0.3104521513	0.5509658664	0.2588753059
23	PS_RPNBR	Process	5790	0.0000237641	4.6240777354	0.0000493267	4.3069182395
24	RCON_NEM	Process	5795	0.3685513548	0.4335019879	0.4464470671	0.3502300258
25	RCON_PEM	Process	5794	1.095985e-24	23.96019535	4.208583e-24	23.375864125

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. Use this table to sort by the various statistics, select rows, or plot quantities of interest.

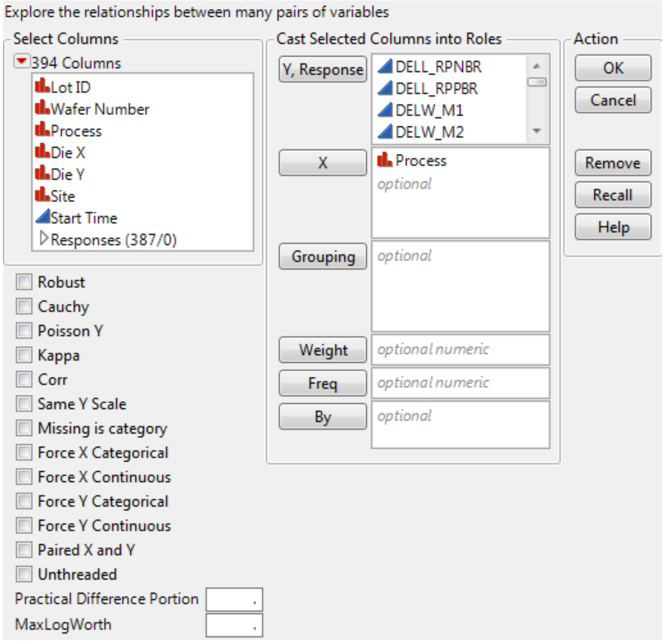
Notice that LogWorth and FDR LogWorth values that correspond to *p*-values of 1e-100 or less are reported as 100, because you set MaxLogWorth to 100 in the launch window. Also, cells corresponding to FDR LogWorth values greater than two are colored with an intensity gradient.

See "The Response Screening Report" on page 296 for details about the report and PValues table.

Launch the Response Screening Platform

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

Figure 17.4 Response Screening Launch Window



Explore the relationships between many pairs of variables

Select Columns

394 Columns

- Lot ID
- Wafer Number
- Process
- Die X
- Die Y
- Site
- Start Time
- Responses (387/0)

Cast Selected Columns into Roles

Y, Response: DELL_RPNBR, DELL_RPPBR, DELW_M1, DELW_M2

X: Process

Grouping: optional

Weight: optional numeric

Freq: optional numeric

By: optional

☐ Robust
☐ Cauchy
☐ Poisson Y
☐ Kappa
☐ Corr
☐ Same Y Scale
☐ Missing is category
☐ Force X Categorical
☐ Force X Continuous
☐ Force Y Categorical
☐ Force Y Continuous
☐ Paired X and Y
☐ Unthreaded

Practical Difference Portion: .

MaxLogWorth: .

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 Model Specification chapter in the *Fitting Linear Models* book.

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 Model Specification chapter in the *Fitting Linear Models* book.

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.

Cauchy Assumes that the errors have a Cauchy distribution. A Cauchy distribution has fatter tails than the normal distribution, resulting in a reduced emphasis on outliers. This option can be useful if you have a large proportion of outliers in your data. However, if your data are close to normal with only a few outliers, this option can lead to incorrect inferences. The Cauchy option estimates parameters using maximum likelihood and a Cauchy link function.

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 The Corr option computes the Pearson product-moment correlation in terms of the indices defined by the value ordering.

The calculation of the Pearson product-moment correlation gives Spearman’s rho in the following instances:

- X and Y are both ordinal
- X and Y are nominal where their value ordering corresponds to the order relation

If X and Y are both binary, the Pearson calculation gives Kendall's Tau-b. Otherwise, a value of Corr that is large in magnitude indicates an association; a Corr value that is small in magnitude does not preclude an 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 $\hat{\sigma} = (IQR)/(1.3489795)$.

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 304.

MaxLogWorth Use to control the scale of plots involving LogWorth values (-$\log_{10}$ 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 313 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

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 320.

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

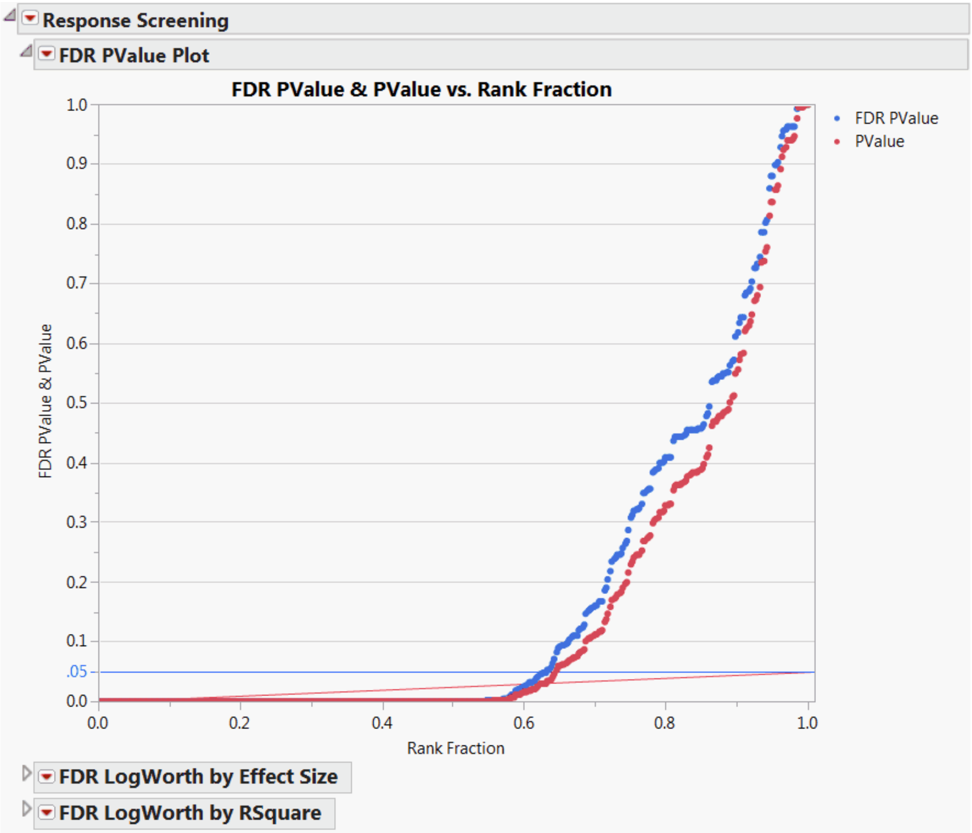
The FDR PValue Plot report shows a plot of FDR PValues and PValues against the Rank Fraction. The Rank Fraction ranks the PValues in order of decreasing significance. FDR PValues are plotted in blue and PValues in red.

A blue horizontal line shows the 0.05 significance level. Note that you can change this level by double-clicking the *y*-axis, removing the current reference line, and adding a new reference line.

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 17.5 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.

Figure 17.5 FDR PValue Plot



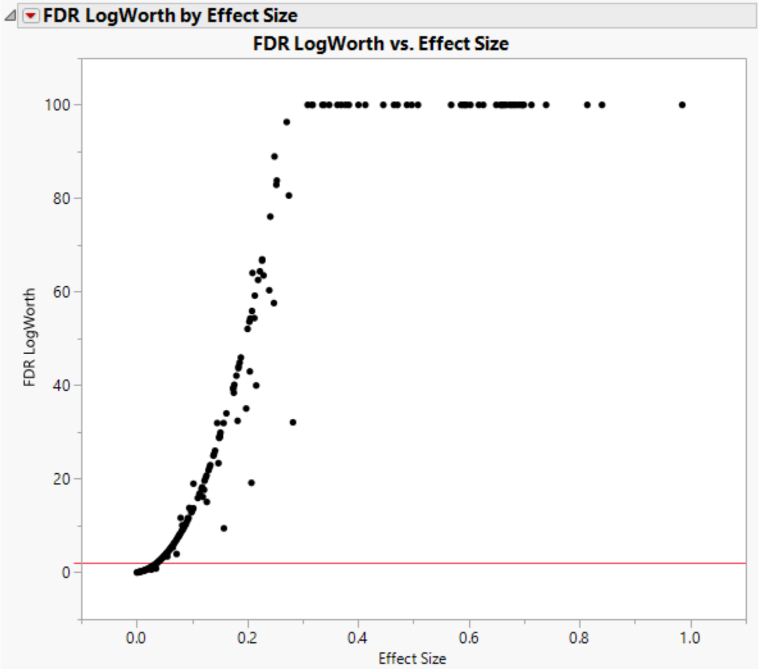
FDR LogWorth by Effect Size

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*-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 17.6 shows the FDR LogWorth by Effect size plot for the Probe.jmp sample data table with MaxLogWorth set to 100. Most FDR LogWorth values exceed 2, which indicates that most effects are significant at the 0.01 level. The FDR LogWorth values of 100 correspond to extremely small *p*-values.

Figure 17.6 FDR LogWorth by Effect Size



FDR LogWorth by RSquare

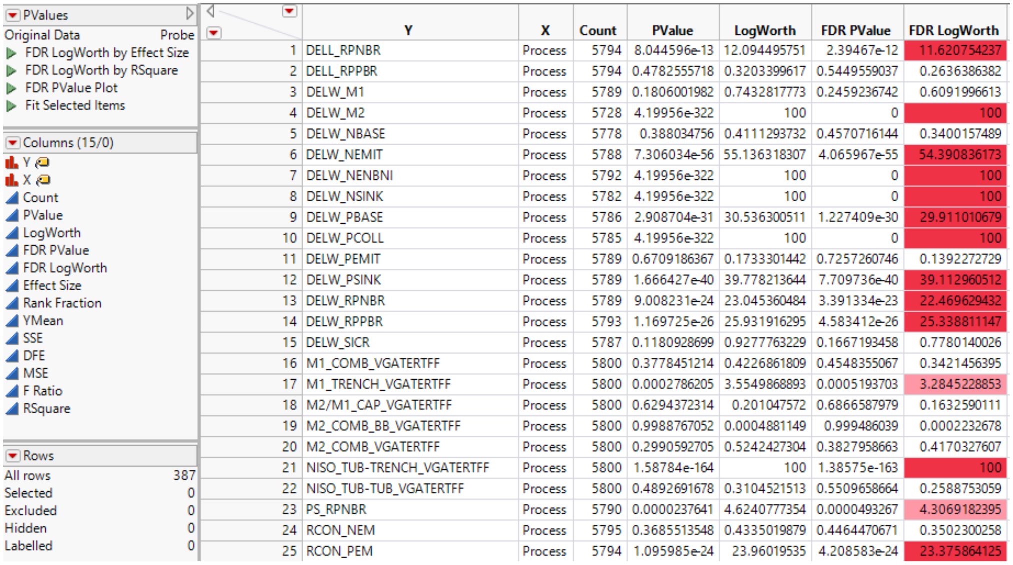
The FDR LogWorth by RSquare plot shows the FDR LogWorth on the vertical axis and RSquare values on the horizontal axis. Larger LogWorth values tend to be associated with larger RSquare values, but this relationship also depends on the number of observations.

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. The PValues data table also contains a table variable called Original Data that gives the name of the data table that was used for the analysis. If you specified a By variable, JMP creates a PValues table for each level of the By variable, and the Original Data variable gives the By variable and its level.

Figure 17.7 shows the PValues data table created in "Example of Response Screening" on page 291.

Figure 17.7 PValues Data Table, Partial View



	Y	X	Count	PValue	LogWorth	FDR PValue	FDR LogWorth
1	DELL_RPNBR	Process	5794	8.044596e-13	12.094495751	2.39467e-12	11.620754237
2	DELL_RPPBR	Process	5794	0.4782555718	0.3203399617	0.5449559037	0.2636386382
3	DELW_M1	Process	5789	0.1806001982	0.7432817773	0.2459236742	0.6091996613
4	DELW_M2	Process	5728	4.19956e-322	100	0	100
5	DELW_NBASE	Process	5778	0.388034756	0.4111293732	0.4570716144	0.3400157489
6	DELW_NEMIT	Process	5788	7.306034e-56	55.136318307	4.065967e-55	54.390836173
7	DELW_NENBNI	Process	5792	4.19956e-322	100	0	100
8	DELW_NSINK	Process	5782	4.19956e-322	100	0	100
9	DELW_PBASE	Process	5786	2.908704e-31	30.536300511	1.227409e-30	29.911010679
10	DELW_PCOLL	Process	5785	4.19956e-322	100	0	100
11	DELW_PEMIT	Process	5789	0.6709186367	0.1733301442	0.7257260746	0.1392272729
12	DELW_PSINK	Process	5789	1.666427e-40	39.778213644	7.709736e-40	39.112960512
13	DELW_RPNBR	Process	5789	9.008231e-24	23.045360484	3.391334e-23	22.469629432
14	DELW_RPPBR	Process	5793	1.169725e-26	25.931916295	4.583412e-26	25.338811147
15	DELW_SICR	Process	5787	0.1180928699	0.9277763229	0.1667193458	0.7780140026
16	M1_COMB_VGATERTFF	Process	5800	0.3778451214	0.4226861809	0.4548355067	0.3421456395
17	M1_TRENCH_VGATERTFF	Process	5800	0.0002786205	3.5549868893	0.0005193703	3.2845228853
18	M2/M1_CAP_VGATERTFF	Process	5800	0.6294372314	0.201047572	0.6866587979	0.1632590111
19	M2_COMB_BB_VGATERTFF	Process	5800	0.9988767052	0.0004881149	0.999486039	0.0002232678
20	M2_COMB_VGATERTFF	Process	5800	0.2990592705	0.5242427304	0.3827958663	0.4170327607
21	NISO_TUB-TRENCH_VGATERTFF	Process	5800	1.58784e-164	100	1.38575e-163	100
22	NISO_TUB-TUB_VGATERTFF	Process	5800	0.4892691678	0.3104521513	0.5509658664	0.2588753059
23	PS_RPNBR	Process	5790	0.0000237641	4.6240777354	0.0000493267	4.3069182395
24	RCON_NEM	Process	5795	0.3685513548	0.4335019879	0.4464470671	0.3502300258
25	RCON_PEM	Process	5794	1.095985e-24	23.96019535	4.208583e-24	23.375864125

PValues Data Table Columns

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. For details about Fit Y by X statistics, see the Introduction to Fit Y by X chapter of the *Basic Analysis* book.

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 320.

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. Cells corresponding to FDR LogWorth values greater than two (*p*-values less than 0.01) are colored with an intensity gradient.

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 Bivariate chapter in the *Basic Analysis* book for additional details about Huber M-estimation. For an example, see "Example of Robust Fit" on page 315.

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 -$\log_{10}$(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 -$\log_{10}$(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.

Select Columns Selects the columns in the original data table that correspond to rows that you select in the PValues table or to points that you select in plots in the Response Screening report window. Select the rows or points first, then select Select Columns. The corresponding columns in the data table are selected. You can select columns corresponding to additional rows in the PValues table or points in plots by first selecting them and then selecting Select Columns again. To select columns corresponding to different rows or points, first clear the current column selection in the original data table.

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 304. For an example, see “Example of Tests of Practical Significance and Equivalence” on page 311.

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.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

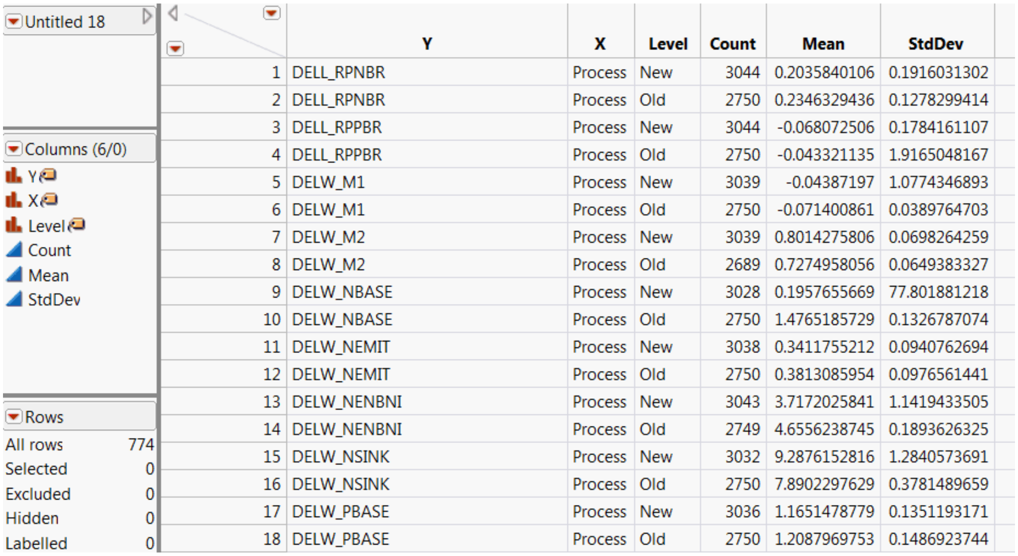
Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

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 387x2 = 774 rows (Figure 17.8).

Figure 17.8 Means Data Table



	Y	X	Level	Count	Mean	StdDev
1	DELL_RPNBR	Process	New	3044	0.2035840106	0.1916031302
2	DELL_RPNBR	Process	Old	2750	0.2346329436	0.1278299414
3	DELL_RPPBR	Process	New	3044	-0.068072506	0.1784161107
4	DELL_RPPBR	Process	Old	2750	-0.043321135	1.9165048167
5	DELW_M1	Process	New	3039	-0.04387197	1.0774346893
6	DELW_M1	Process	Old	2750	-0.071400861	0.0389764703
7	DELW_M2	Process	New	3039	0.8014275806	0.0698264259
8	DELW_M2	Process	Old	2689	0.7274958056	0.0649383327
9	DELW_NBASE	Process	New	3028	0.1957655669	77.801881218
10	DELW_NBASE	Process	Old	2750	1.4765185729	0.1326787074
11	DELW_NEMIT	Process	New	3038	0.3411755212	0.0940762694
12	DELW_NEMIT	Process	Old	2750	0.3813085954	0.0976561441
13	DELW_NENBNI	Process	New	3043	3.7172025841	1.1419433505
14	DELW_NENBNI	Process	Old	2749	4.6556238745	0.1893626325
15	DELW_NSINK	Process	New	3032	9.2876152816	1.2840573691
16	DELW_NSINK	Process	Old	2750	7.8902297629	0.3781489659
17	DELW_PBASE	Process	New	3036	1.1651478779	0.1351193171
18	DELW_PBASE	Process	Old	2750	1.2087969753	0.1486923744

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.

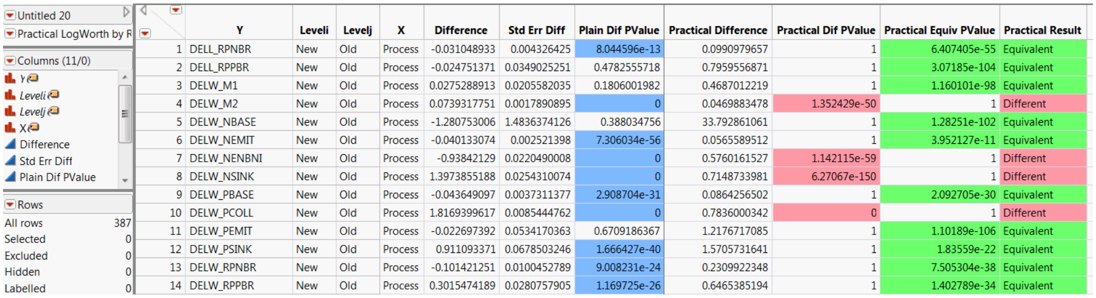
Compare Means Data Table

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 305 for a description of how the practical difference is specified. See “Example of Tests of Practical Significance and Equivalence” on page 311 for an example.

Figure 17.9 Compare Means Data Table



Untitled 20
Practical LogWorth by R
Columns (11/0)
Y
Leveli
Levelj
X
Difference
Std Err Diff
Plain Dif PValue
Rows
All rows 387
Selected 0
Excluded 0
Hidden 0
Labelled 0

	Y	Leveli	Levelj	X	Difference	Std Err Diff	Plain Dif PValue	Practical Difference	Practical Dif PValue	Practical Equiv PValue	Practical Result
1	DELL_RPNBR	New	Old	Process	-0.031048933	0.004326425	8.044596e-13	0.0990979657	1	6.407405e-55	Equivalent
2	DELL_RPPBR	New	Old	Process	-0.024751371	0.0349025251	0.4782555718	0.7959556871	1	3.07185e-104	Equivalent
3	DELW_M1	New	Old	Process	0.0275288913	0.0205582035	0.1806001982	0.4687012219	1	1.160101e-98	Equivalent
4	DELW_M2	New	Old	Process	0.0739317751	0.0017890895	0	0.0469883478	1.352429e-50	1	Different
5	DELW_NBASE	New	Old	Process	-1.280753006	1.4836374126	0.388034756	33.792861061	1	1.28251e-102	Equivalent
6	DELW_NEMIT	New	Old	Process	-0.040133074	0.002521398	7.306034e-56	0.0565589512	1	3.952127e-11	Equivalent
7	DELW_NENBNI	New	Old	Process	-0.93842129	0.0220490008	0	0.5760161527	1.142115e-59	1	Different
8	DELW_NSINK	New	Old	Process	1.3973855188	0.0254310074	0	0.7148733981	6.27067e-150	1	Different
9	DELW_PBASE	New	Old	Process	-0.043649097	0.0037311377	2.908704e-31	0.0864256502	1	2.092705e-30	Equivalent
10	DELW_PCOLL	New	Old	Process	1.8169399617	0.0085444762	0	0.7836000342	0	1	Different
11	DELW_PEMIT	New	Old	Process	-0.022697392	0.0534170363	0.6709186367	1.2176717085	1	1.10189e-106	Equivalent
12	DELW_PSINK	New	Old	Process	0.911093371	0.0678503246	1.666427e-40	1.5705731641	1	1.83559e-22	Equivalent
13	DELW_RPNBR	New	Old	Process	-0.101421251	0.0100452789	9.008231e-24	0.2309922348	1	7.505304e-38	Equivalent
14	DELW_RPPBR	New	Old	Process	0.3015474189	0.0280757905	1.169725e-26	0.6465385194	1	1.402789e-34	Equivalent

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 The 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 $6\hat{\sigma}$ 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.

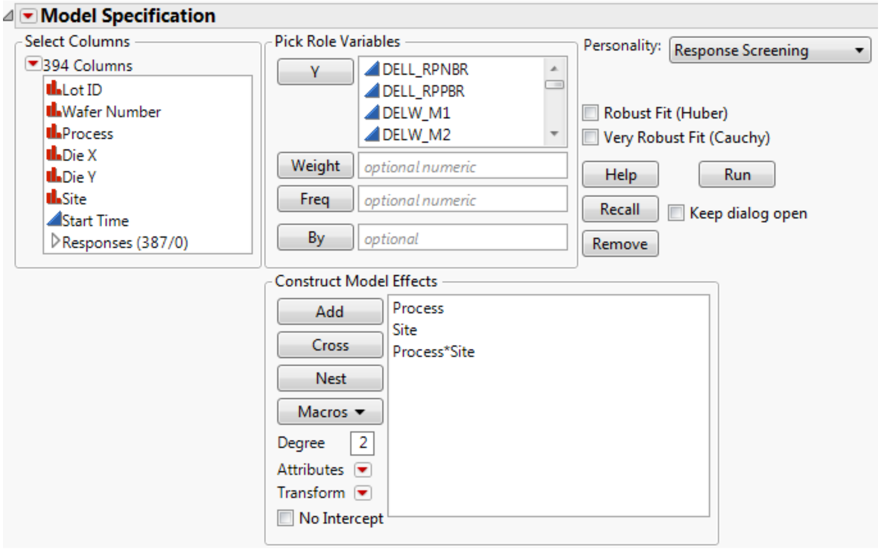
The Response Screening Personality in Fit Model

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 17.10).

Figure 17.10 Response Screening from the Fit Model Window



Model Specification

Select Columns: 394 Columns: Lot ID, Wafer Number, Process, Die X, Die Y, Site, Start Time, Responses (387/0)

Y: DELL_RPNBR, DELL_RPPBR, DELW_M1, DELW_M2

Weight: optional numeric

Freq: optional numeric

By: optional

Personality: Response Screening

□ Robust Fit (Huber)

□ Very Robust Fit (Cauchy)

□ Keep dialog open

Construct Model Effects: Process, Site, Process*Site

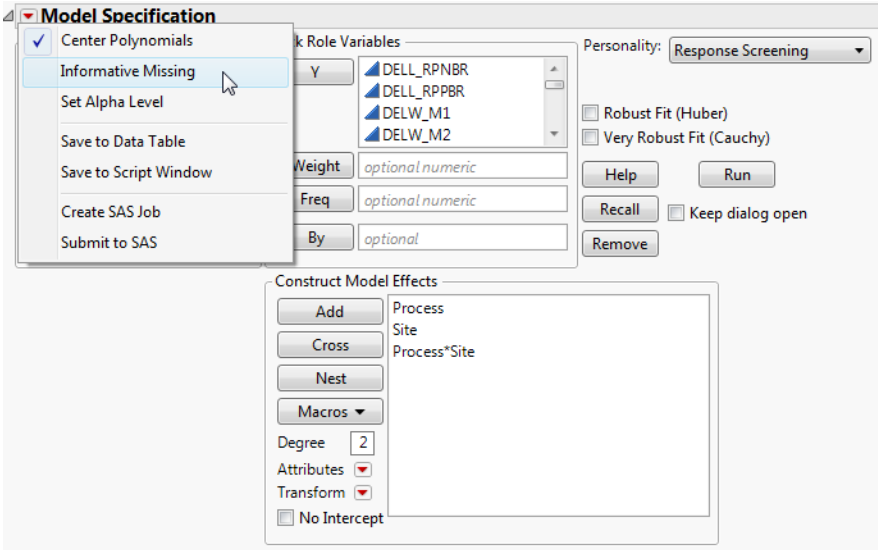
Degree: 2

□ No Intercept

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 down weight 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 17.11). 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 17.11 Informative Missing Option



For details about the Fit Model window, see the Model Specification chapter in the *Fitting Linear Models* book.

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 296.

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 319.

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. This data table also contains a table variable called Original Data that gives the name of the data table that was used for the analysis. If you specified a

By variable, JMP creates an estimates table for each level of the By variable, and the Original Data variable gives the By variable and its level.

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

Save Least Squares Means Opens a data table where each row corresponds to a response and a combination of effect settings. The row contains the least squares mean and standard error for that combination of settings.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

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 Fit 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 320 or Westfall et al. (2011).

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.

The PValues data table also contains a table variable called Original Data that gives the name of the data table that was used for the analysis. If you specified a By variable, JMP creates a PValues table for each level of the By variable, and the Original Data variable gives the By variable and its level.

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 Fit 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 296.)

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.

<Effect> PValue These columns contain *p*-values for tests of each model effect. These columns are arranged in a group called PValue in the columns panel.

<Effect> LogWorth These columns contain LogWorths for the *p*-values for tests of each model effect. These columns are arranged in a group called LogWorth in the columns panel.

<Effect> FDR LogWorth These columns contain FDR LogWorths for tests of each model effect. These columns are arranged in a group called FDR LogWorth in the columns panel.

The Y Fits data table also contains a table variable called Original Data that gives the name of the data table that was used for the analysis. If you specified a By variable, JMP creates a Y Fits table for each level of the By variable, and the Original Data variable gives the By variable and its level.

Additional Examples of Response Screening

The following examples illustrate various aspects of Response Screening.

Example of Tests of Practical Significance and Equivalence

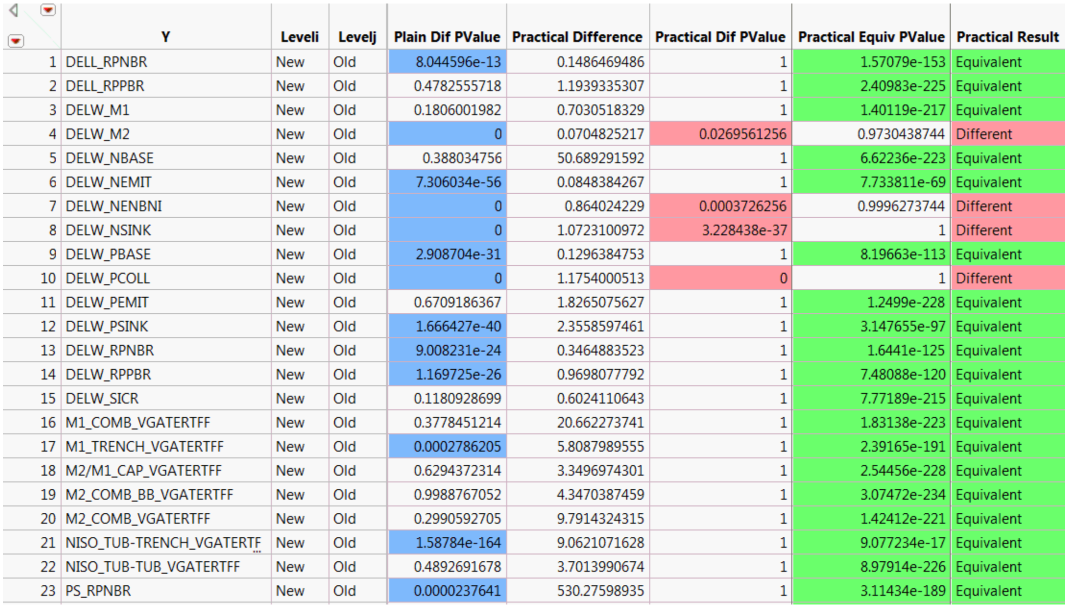
This example tests for practical differences using the Probe.jmp sample data table.

1. Select **Help > Sample Data Library** and open Probe.jmp.
2. Select **Analyze > Screening > 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 17.12 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.

Figure 17.12 Compare Means Table, Partial View



	Y	Leveli	Levelj	Plain Dif PValue	Practical Difference	Practical Dif PValue	Practical Equiv PValue	Practical Result
1	DELL_RPNBR	New	Old	8.044596e-13	0.1486469486	1	1.57079e-153	Equivalent
2	DELL_RPPBR	New	Old	0.4782555718	1.1939335307	1	2.40983e-225	Equivalent
3	DELW_M1	New	Old	0.1806001982	0.7030518329	1	1.40119e-217	Equivalent
4	DELW_M2	New	Old	0	0.0704825217	0.0269561256	0.9730438744	Different
5	DELW_NBASE	New	Old	0.388034756	50.689291592	1	6.62236e-223	Equivalent
6	DELW_NEMIT	New	Old	7.306034e-56	0.0848384267	1	7.733811e-69	Equivalent
7	DELW_NENBNI	New	Old	0	0.864024229	0.0003726256	0.9996273744	Different
8	DELW_NSINK	New	Old	0	1.0723100972	3.228438e-37	1	Different
9	DELW_PBASE	New	Old	2.908704e-31	0.1296384753	1	8.19663e-113	Equivalent
10	DELW_PCOLL	New	Old	0	1.1754000513	0	1	Different
11	DELW_PEMIT	New	Old	0.6709186367	1.8265075627	1	1.2499e-228	Equivalent
12	DELW_PSINK	New	Old	1.666427e-40	2.3558597461	1	3.147655e-97	Equivalent
13	DELW_RPNBR	New	Old	9.008231e-24	0.3464883523	1	1.6441e-125	Equivalent
14	DELW_RPPBR	New	Old	1.169725e-26	0.9698077792	1	7.48088e-120	Equivalent
15	DELW_SICR	New	Old	0.1180928699	0.6024110643	1	7.77189e-215	Equivalent
16	M1_COMB_VGATERTFF	New	Old	0.3778451214	20.662273741	1	1.83138e-223	Equivalent
17	M1_TRENCH_VGATERTFF	New	Old	0.0002786205	5.8087989555	1	2.39165e-191	Equivalent
18	M2/M1_CAP_VGATERTFF	New	Old	0.6294372314	3.3496974301	1	2.54456e-228	Equivalent
19	M2_COMB_BB_VGATERTFF	New	Old	0.9988767052	4.3470387459	1	3.07472e-234	Equivalent
20	M2_COMB_VGATERTFF	New	Old	0.2990592705	9.7914324315	1	1.42412e-221	Equivalent
21	NISO_TUB-TRENCH_VGATERTF	New	Old	1.58784e-164	9.0621071628	1	9.077234e-17	Equivalent
22	NISO_TUB-TUB_VGATERTFF	New	Old	0.4892691678	3.7013990674	1	8.97914e-226	Equivalent
23	PS_RPNBR	New	Old	0.0000237641	530.27598935	1	3.11434e-189	Equivalent

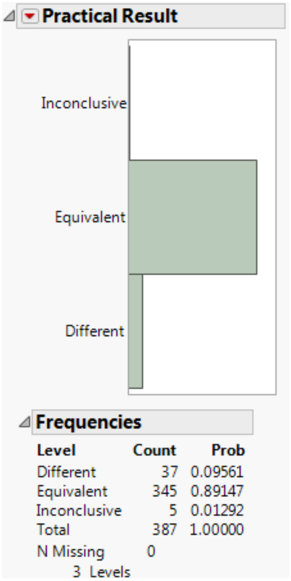
Because specification limits are not saved as column properties in Probe.jmp, JMP calculates a value of the practical difference for each response. The practical difference of 0.15 that you specified is multiplied by an estimate of the 6σ range of the response. This value is used in testing for practical 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**.
9. Select Practical Result and click **Y, Columns**.
10. Click **OK**.

Figure 17.13 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.

Figure 17.13 Distribution of Practical Significance Results



Frequencies

Level	Count	Prob
Different	37	0.09561
Equivalent	345	0.89147
Inconclusive	5	0.01292
Total	387	1.00000
N Missing	0	

3 Levels

The 37 responses can be selected for further study by clicking on the corresponding bar in the plot.

Example of the MaxLogWorth Option

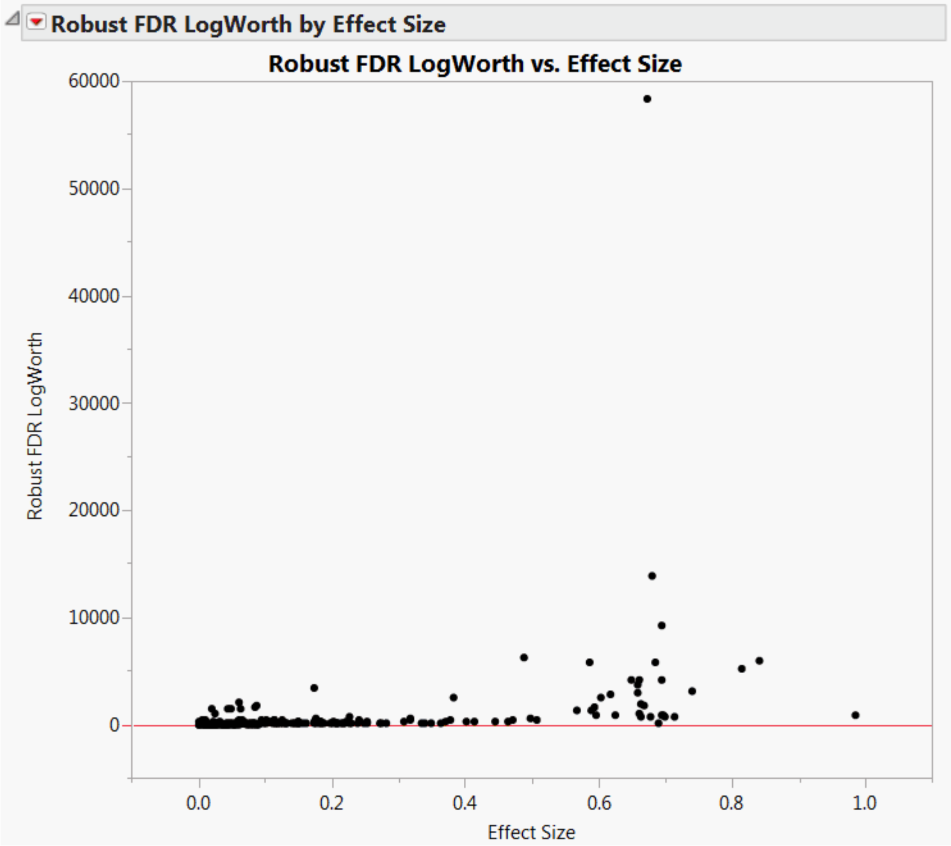
When data sets have a large number of observations, *p*-values can be very small. LogWorth values provide a useful way to study *p*-values graphically in these cases. But sometimes *p*-values are so small that the LogWorth scale is distorted by huge values.

1. Select **Help > Sample Data Library** and open Probe.jmp.
2. Select **Analyze > Screening > 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 17.14). To ensure that your graphs show sufficient detail, you can set a maximum value of the LogWorth.

Figure 17.14 Robust FDR LogWorth vs. Effect Size, MaxLogWorth Not Set

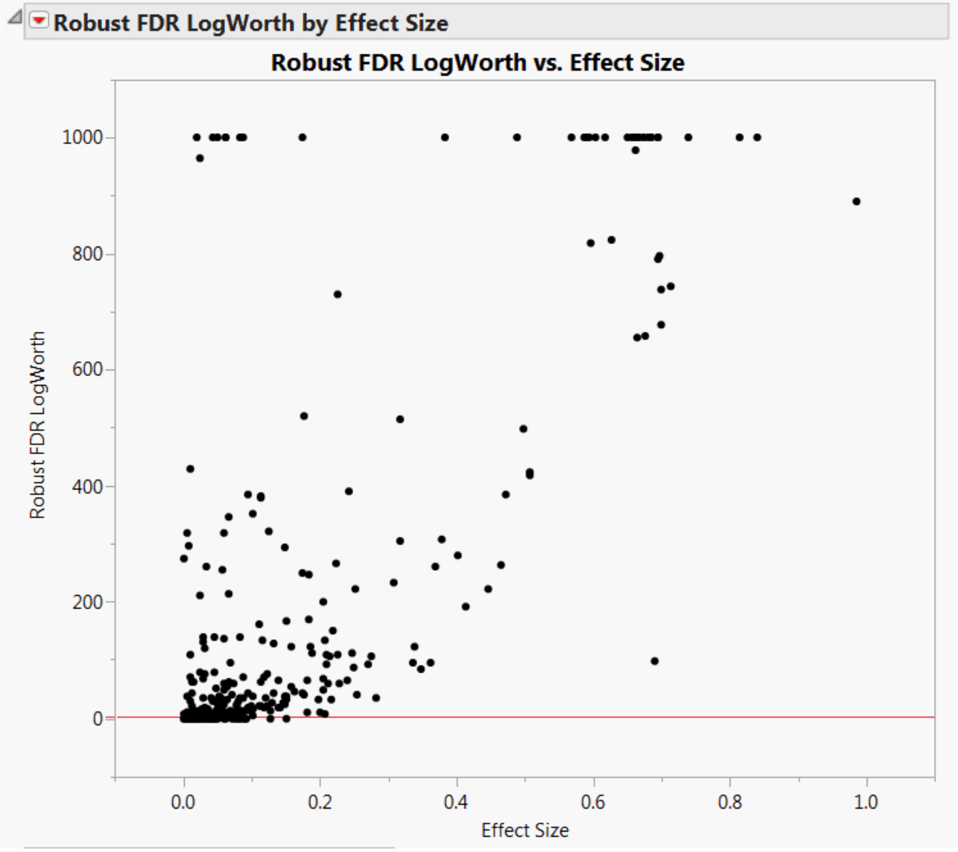


8. Repeat step 1 through step 5.
9. Type 1000 in the MaxLogWorth box at the bottom of the launch window.
10. Click **OK**.

    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 17.15).

Figure 17.15 Robust FDR LogWorth vs. Effect Size, MaxLogWorth = 1000

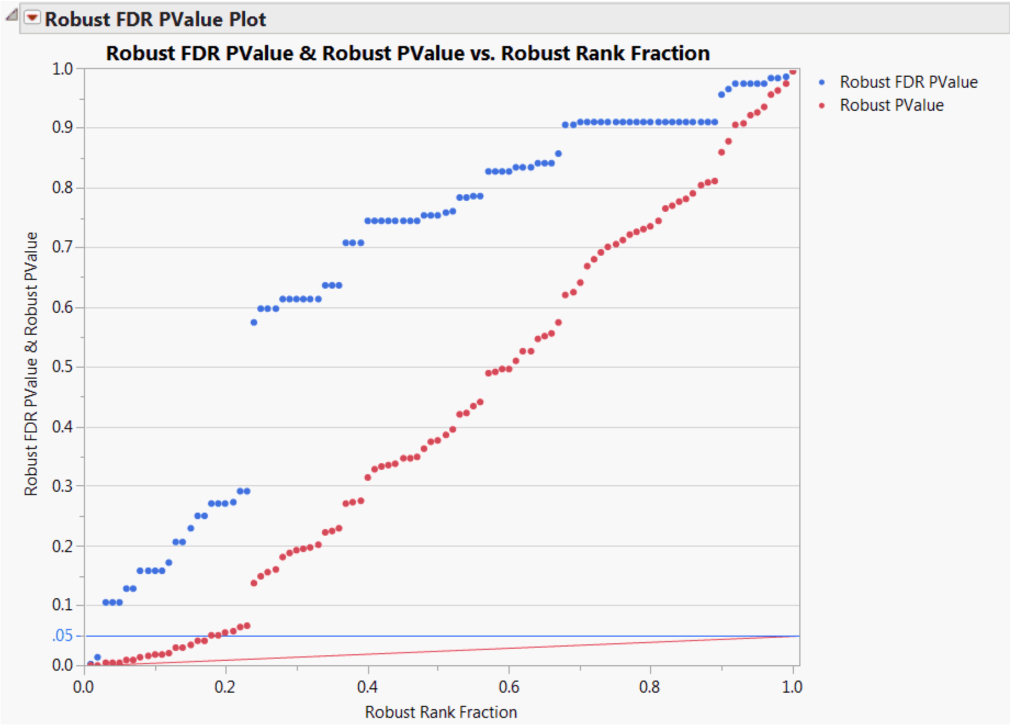


Example of Robust Fit

1. Open the Drosophila Aging.jmp table.
2. Select **Analyze > Screening > 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 17.16. 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.

Figure 17.16 Robust FDR PValue Plot for Drosophila Data

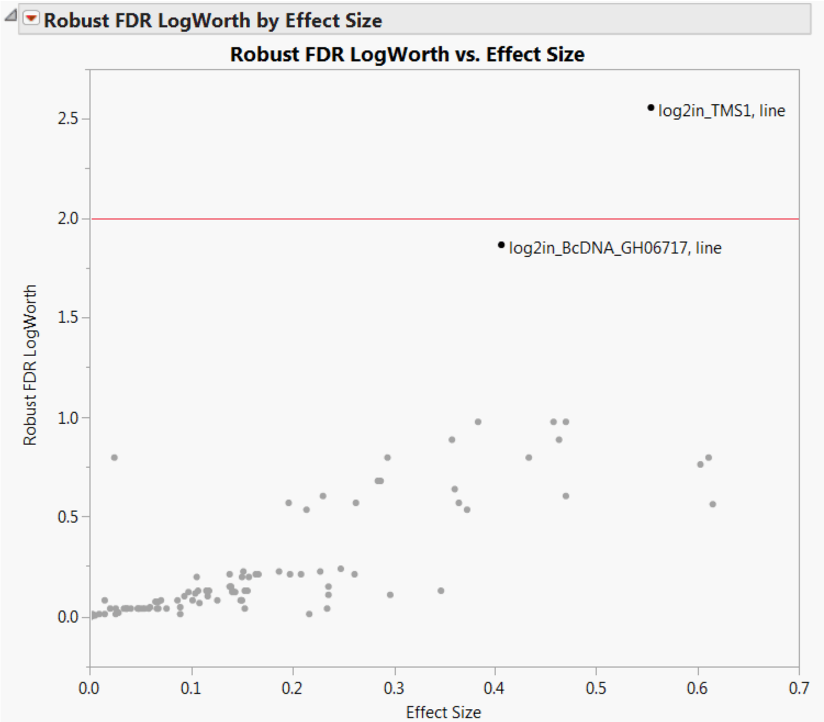


These two points are more easily identified in a plot that shows FDR LogWorths.

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 17.17 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.

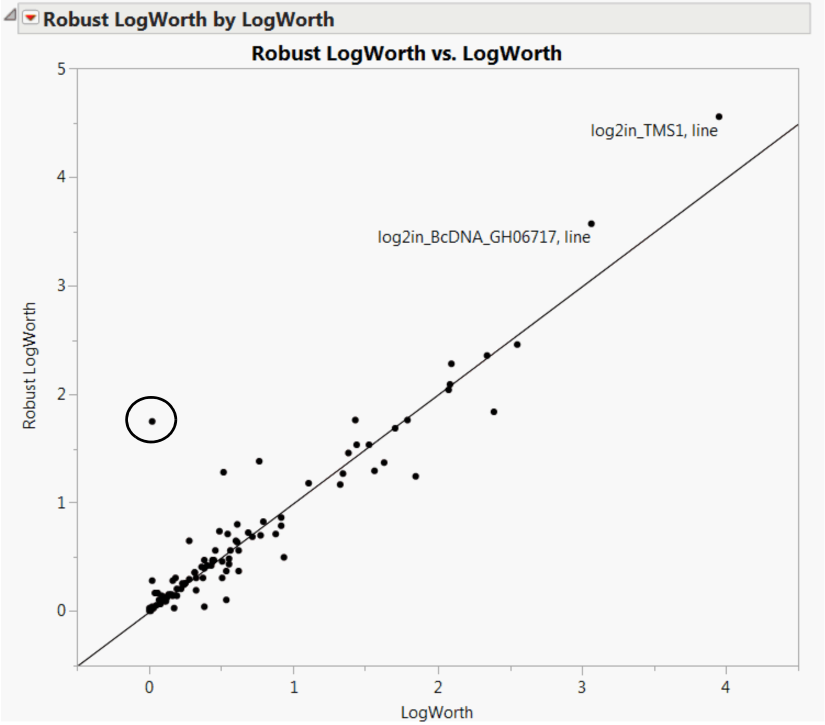
Figure 17.17 Robust LogWorth by Effect Size for Drosophila Data



10. Click the Robust LogWorth by LogWorth disclosure icon.

    The plot shown in Figure 17.18. If the robust test for a response were identical to the usual test, its corresponding point would fall on the diagonal line in Figure 17.18. The circled point in the plot does not fall near the line, because it has a Robust LogWorth value that exceeds its LogWorth value.

Figure 17.18 Robust LogWorth by LogWorth for Drosophila Data

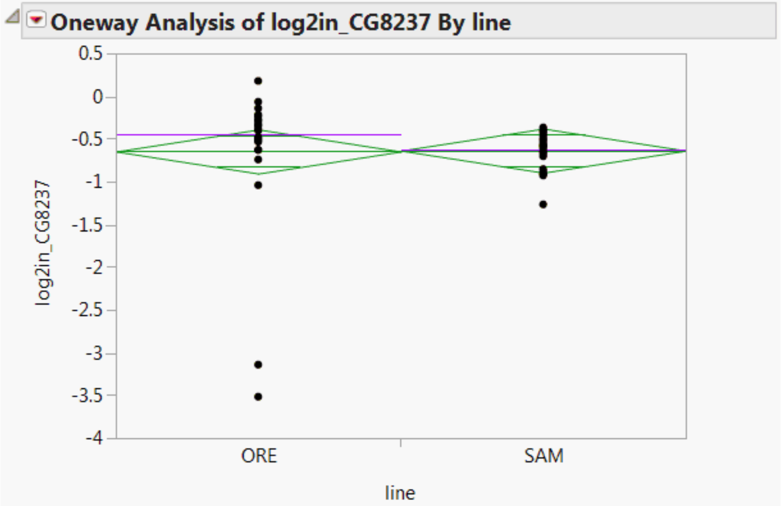


11. Drag a rectangle around this point in the plot.
12. Find the row for this point in the PValues data table.

    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 17.19). 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 17.19 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.

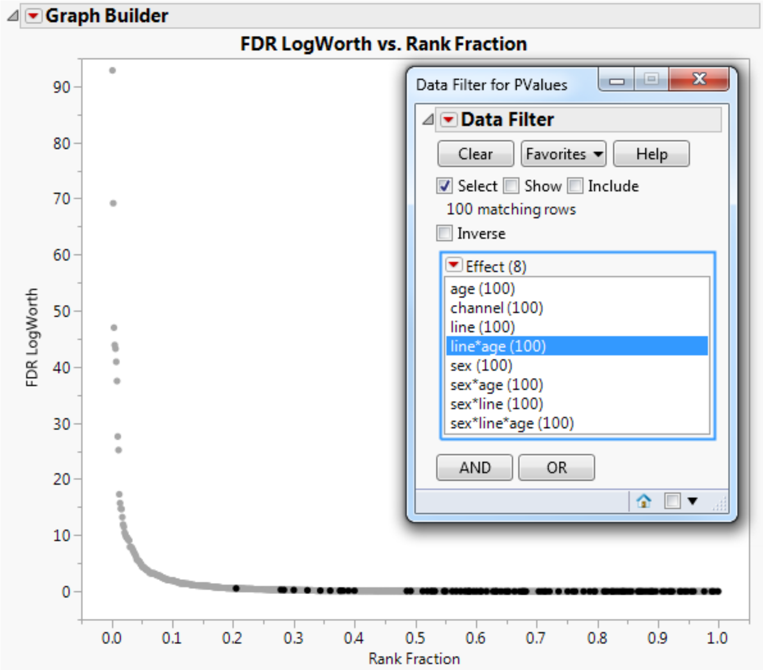
1. Open the Drosophila Aging.jmp table.
2. Select **Analyze > Fit Model**.
3. Select all the continuous columns and click **Y**.
4. Select channel and click **Add**.
5. Select sex, line, and age and select **Macros > Full Factorial**.
6. Select **Response Screening** from the Personality list.
7. Click **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 17.20 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 17.20 FDR LogWorth vs Rank Fraction Plot with line*age Tests Selected



☑ Select □ Show □ Include

100 matching rows

□ Inverse

Effect (8)

- age (100)
- channel (100)
- line (100)
- line*age (100)
- sex (100)
- sex*age (100)
- sex*line (100)
- sex*line*age (100)

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. See also Westfall et al. (2011). This method assumes that the *p*-values are independent and uniformly distributed.

JMP uses the following procedure to control the false discovery rate at level α:

1. Conduct the *m* 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 $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}$.
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 α.

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

$$p_{(i),\,FDR} = \begin{cases} p_{(m)} & \text{for } i = m \\ min\left[p_{(i+1),\,FDR}, \left(\frac{m}{i}\right)p_{(i)}\right] & \text{for } i = m-1, \ldots, 1 \end{cases}$$

If a hypothesis has an FDR-adjusted p-value that falls below α, then it is rejected by the procedure.

Chapter 18

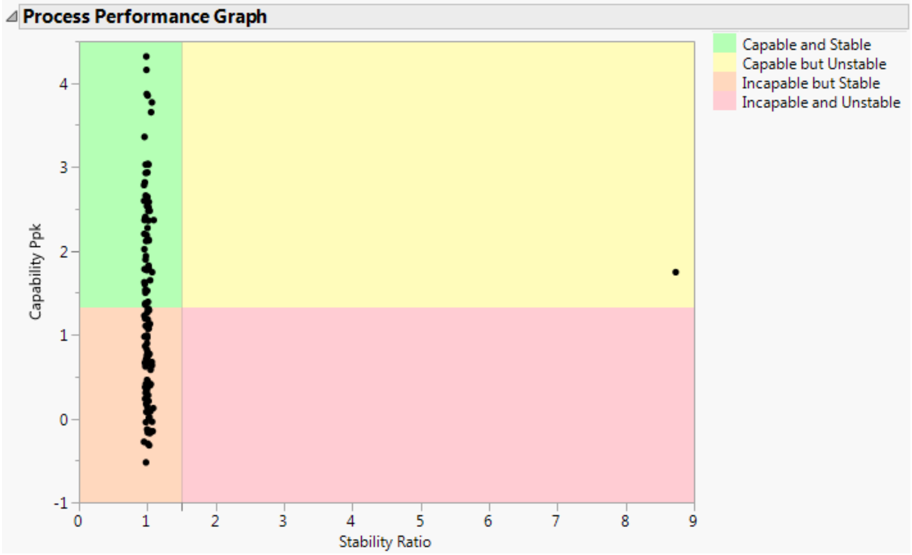
Process Screening

Screen Many Processes for Stability and Capability

Use the Process Screening platform for exploring a large number of processes across time. The platform calculates process stability and process capability metrics. The platform creates control charts and detects large process shifts. The platform is intended to expedite the evaluation of a very large number of processes by enabling you to quickly focus on the processes that are unstable, not capable of meeting specification limits, or subject to shifts in the mean.

Based on your initial results, you can choose to explore specific processes graphically or in greater analytical depth. You can easily access the Control Chart Builder and Process Capability platforms. You can save detailed results for all of your processes or for specific processes.

Figure 18.1 Example of a Process Performance Graph



Process Screening Platform Overview

The Process Screening platform facilitates the task of assessing data from a large number of processes for stability and capability. The results are largely based on control chart calculations to determine when a process is out of control. The Process Screening platform enables you to:

- Specify a constant subgroup size or use a variable containing subgroup identifiers for control chart calculations using subgroups.
- Use variables that identify different processes as grouping variables. An analysis is provided for each process variable for each combination of values of the grouping variables.
- Use medians to make your centerline and sigma calculations robust to outliers.
- Obtain information about the location of large shifts in the mean of your processes.

You can customize the Summary report to show the control chart tests that you want, including tests for changes in process mean and spread. The report also provides capability information when you supply specification limits. The Process Performance Graph gives you a visual representation of the performance of your processes in terms of stability and capability. The Shift Graph shows locations of upshifts and downshifts.

Process Screening makes it easy to select specific processes for further analysis. The platform provides small run charts for these processes - the size of the plots makes it easy for you to view a substantial number at a time. You can also link to Control Chart Builder and the Process Capability platform for analyses of select processes.

You can save data tables containing results in various forms, either for your entire set of processes or only for select processes.

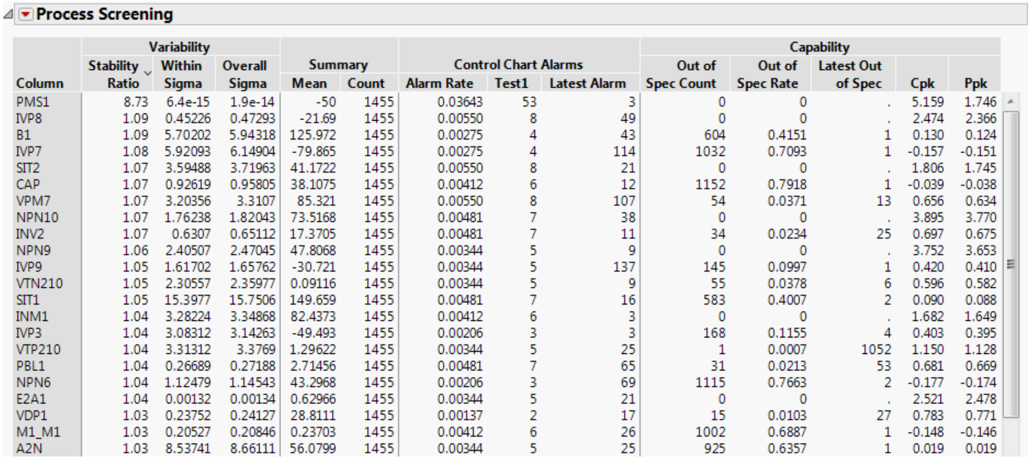
Example of Process Screening

The Semiconductor Capability.jmp sample data table contains 128 columns of process measurements. Each column contains 1,455 measurements. You are interested in which of the processes are unstable. Also, each column contains a Spec Limits column property. If a process is stable, it is appropriate to calculate its process capability. You proceed to assess both stability and capability for this data table.

1. Select **Help > Sample Data Library** and open Semiconductor Capability.jmp.
2. Select **Analyze > Screening > Process Screening**.
3. Select the Processes column group and click **Process Variables**.

   Notice that the Control Chart Type is set to Indiv and MR.
4. Click **OK**.

Figure 18.2 Partial View of Initial Report



Process Screening

	Variability								Capability				
	Stability Ratio	Within Sigma	Overall Sigma	Summary		Control Chart Alarms			Out of Spec Count	Out of Spec Rate	Latest Out of Spec		
Column				Mean	Count	Alarm Rate	Test1	Latest Alarm				Cpk	Ppk
PMS1	8.73	6.4e-15	1.9e-14	-50	1455	0.03643	53	3	0	0	.	5.159	1.746
IVP8	1.09	0.45226	0.47293	-21.69	1455	0.00550	8	49	0	0	.	2.474	2.366
B1	1.09	5.70202	5.94318	125.972	1455	0.00275	4	43	604	0.4151	1	0.130	0.124
IVP7	1.08	5.92093	6.14904	-79.865	1455	0.00275	4	114	1032	0.7093	1	-0.157	-0.151
SIT2	1.07	3.59488	3.71963	41.1722	1455	0.00550	8	21	0	0	.	1.806	1.745
CAP	1.07	0.92619	0.95805	38.1075	1455	0.00412	6	12	1152	0.7918	1	-0.039	-0.038
VPM7	1.07	3.20356	3.3107	85.321	1455	0.00550	8	107	54	0.0371	13	0.656	0.634
NPN10	1.07	1.76238	1.82043	73.5168	1455	0.00481	7	38	0	0	.	3.895	3.770
INV2	1.07	0.6307	0.65112	17.3705	1455	0.00481	7	11	34	0.0234	25	0.697	0.675
NPN9	1.06	2.40507	2.47045	47.8068	1455	0.00344	5	9	0	0	.	3.752	3.653
IVP9	1.05	1.61702	1.65762	-30.721	1455	0.00344	5	137	145	0.0997	1	0.420	0.410
VTN210	1.05	2.30557	2.35977	0.09116	1455	0.00344	5	9	55	0.0378	6	0.596	0.582
SIT1	1.05	15.3977	15.7506	149.659	1455	0.00481	7	16	583	0.4007	2	0.090	0.088
INM1	1.04	3.28224	3.34868	82.4373	1455	0.00412	6	3	0	0	.	1.682	1.649
IVP3	1.04	3.08312	3.14263	-49.493	1455	0.00206	3	3	168	0.1155	4	0.403	0.395
VTP210	1.04	3.31312	3.3769	1.29622	1455	0.00344	5	25	1	0.0007	1052	1.150	1.128
PBL1	1.04	0.26689	0.27188	2.71456	1455	0.00481	7	65	31	0.0213	53	0.681	0.669
NPN6	1.04	1.12479	1.14543	43.2968	1455	0.00206	3	69	1115	0.7663	2	-0.177	-0.174
E2A1	1.04	0.00132	0.00134	0.62966	1455	0.00344	5	21	0	0	.	2.521	2.478
VDP1	1.03	0.23752	0.24127	28.8111	1455	0.00137	2	17	15	0.0103	27	0.783	0.771
M1_M1	1.03	0.20527	0.20846	0.23703	1455	0.00412	6	26	1002	0.6887	1	-0.148	-0.146
A2N	1.03	8.53741	8.66111	56.0799	1455	0.00344	5	25	925	0.6357	1	0.019	0.019

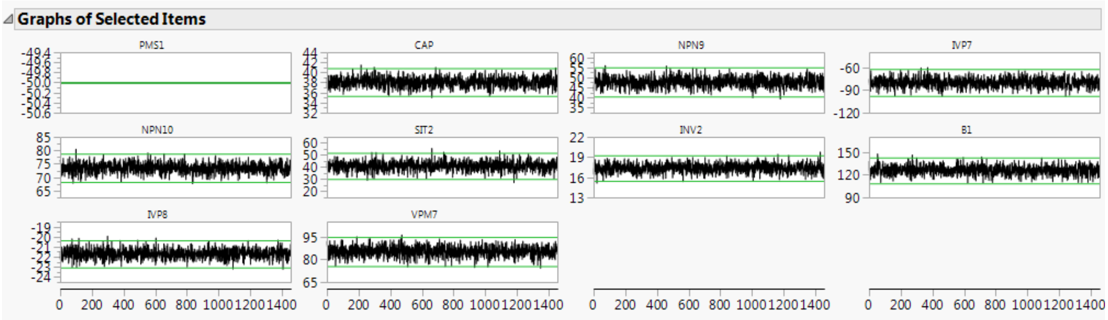
The Process Screening window appears, showing a table of results for each process. The table is sorted by Stability Ratio. This is a measure of the stability of a process, where a stable process has a stability ratio near 1. Higher values of the stability ratio indicate a less stable process. (The sorting is indicated by the caret beside Stability Ratio in the report.) You want to take a closer look at processes with a stability ratio greater than 1.05.

5. In the report window, select processes PMS1 through NPN9.

   Each of these first 10 processes has a value of 1.05 or larger in the Stability Ratio column.

6. Click the red triangle next to Process Screening and select **Quick Graph for Selected Items**.

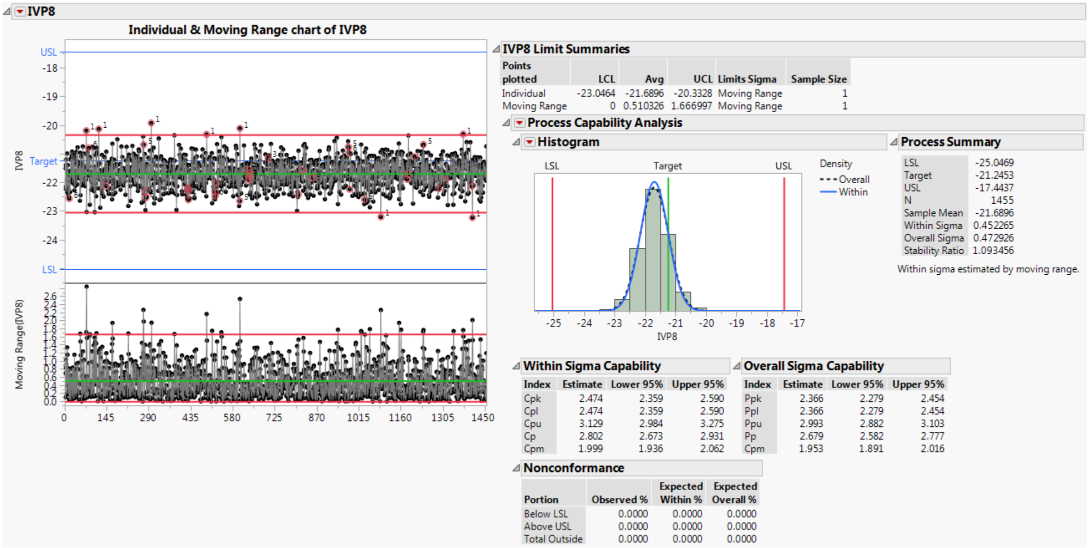
Figure 18.3 Quick Graphs for Highest Alarm Rate Processes



You decide to take a closer look at IVP8 (row 3, column 1 in the Graphs of Selected Items).

7. Select the second process in the Summary table, which corresponds to IVP8.
8. Click the red triangle next to Process Screening and select **Control Charts for Selected Items**.

Figure 18.4 Control Chart Builder Report for PNP9



IVP8 Limit Summaries

Points plotted	LCL	Avg	UCL	Limits Sigma	Sample Size
Individual	-23.0464	-21.6896	-20.3328	Moving Range	1
Moving Range	0	0.510326	1.666997	Moving Range	1

Process Summary

LSL	-25.0469
Target	-21.2453
USL	-17.4437
N	1455
Sample Mean	-21.6896
Within Sigma	0.452265
Overall Sigma	0.472926
Stability Ratio	1.093456

Within sigma estimated by moving range.

Within Sigma Capability

Index	Estimate	Lower 95%	Upper 95%
Cpk	2.474	2.359	2.590
Cpl	2.474	2.359	2.590
Cpu	3.129	2.984	3.275
Cp	2.802	2.673	2.931
Cpm	1.999	1.936	2.062

Overall Sigma Capability

Index	Estimate	Lower 95%	Upper 95%
Ppk	2.366	2.279	2.454
Ppl	2.366	2.279	2.454
Ppu	2.993	2.882	3.103
Pp	2.679	2.582	2.777
Cpm	1.953	1.891	2.016

Nonconformance

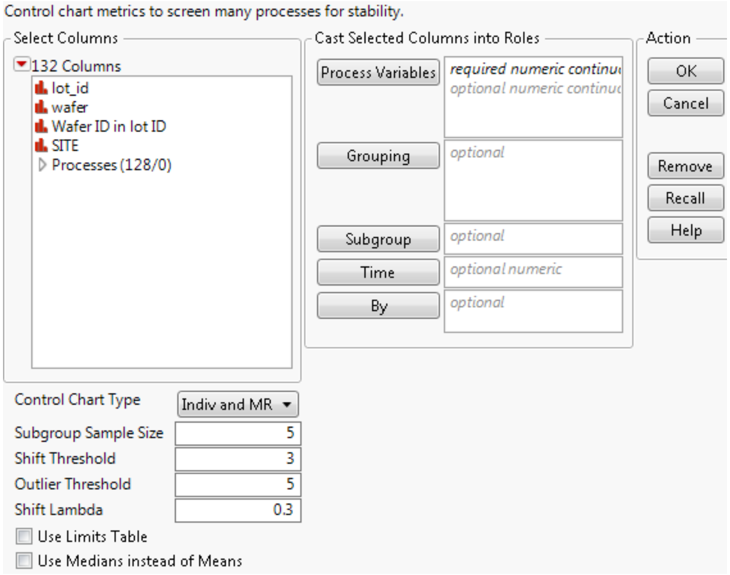
Portion	Observed %	Expected Within %	Expected Overall %
Below LSL	0.0000	0.0000	0.0000
Above USL	0.0000	0.0000	0.0000
Total Outside	0.0000	0.0000	0.0000

A Control Chart Builder report appears. Because IVP8 has a Spec Limits column property, the report also includes a capability analysis.

Launch the Process Screening Platform

Launch the Process Screening platform by selecting **Analyze > Screening > Process Screening**.

Figure 18.5 Response Screening Launch Window



Control chart metrics to screen many processes for stability.

Control Chart Type: Indiv and MR
Subgroup Sample Size: 5
Shift Threshold: 3
Outlier Threshold: 5
Shift Lambda: 0.3
□ Use Limits Table
□ Use Medians instead of Means

Launch Window Roles

Process Variables The columns of process data containing the measurements to be analyzed. The columns must have a Numeric data type.

Grouping For each column entered as a Process Variable and each combination of levels of the Grouping columns, the platform analyzes the corresponding processes separately. The results are presented in a single report.

Subgroup A column whose values assign a subgroup identifier to each row. The process data are sorted by the Subgroup variable before calculations are performed.

Time A numeric column whose values are used for the time order for the data. Use the Time role for data that are time-stamped. The timestamp will be used for the time axis in quick graphs and shift graphs. The process data are sorted by the Time variable before calculations are performed.

By A column whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate tables and reports. If more than one By variable is assigned, a separate analysis is produced for each possible combination of the levels of the By variables.

Launch Window Options

Control Chart Type Select one of three control chart types: Indiv and MR (Individual Measurement and Moving Range), XBar and R, or XBar and S. For more information about statistical details, see the Control Chart Builder chapter in the *Quality and Process Methods* book.

Subgroup Sample Size Specify a constant sample size for subgroups. The minimum subgroup size is 2. A subgroup size of 5 is the default. The Subgroup Sample Size specification is ignored for Indiv and MR charts or when a subgroup variable is specified.

Shift Threshold Specify a value that controls the sensitivity of the Shift Graph. Shift Threshold is set to three by default. After outlier removal, the Shift Graph shows a plot of the time occurrence of all process shifts that exceed the number of within-sigma units specified by the Shift Threshold. See "Shift Graph" on page 335.

Outlier Threshold Specify a value that controls the sensitivity of outlier removal for detection of large recent shifts and for the Shift Graph. Outlier Threshold is set to five by default. If the number of within sigma units from an observation to both of its neighboring observations exceeds the specified Outlier Threshold, that observation is replaced with a value that is one within-sigma unit away from its closest neighboring observation. See "Shift Magnitudes and Positions" on page 333.

Shift Lambda Enables you to change the exponentially-weighted moving average (EWMA) weight used in the Shift Graph. See "Shift Magnitudes and Positions" on page 333.

Use Limits Table Enables you to import historical control limits and specification limits from a data table. When you select this option and click OK in the launch window, a Limits Specification window appears. Assign columns in your limits table to appropriate roles and click OK. For details. See "Limits Table" on page 329.

Use Medians instead of Means Estimates the center line using the median of the observations. Sigma is estimated using scaling factors obtained using Monte Carlo simulation. The table of factors is given in "Statistical Details" on page 339. The calculation depends on the type of chart selected:

- For XBar and R chart or Indiv and MR chart calculations, sigma is estimated using the scaled median of the ranges.
- For XBar and S chart calculations, sigma is estimated using the scaled median of the standard deviations.

– For unequal subgroup sizes, the scaling factor corresponds to the average subgroup size rounded to the nearest integer.

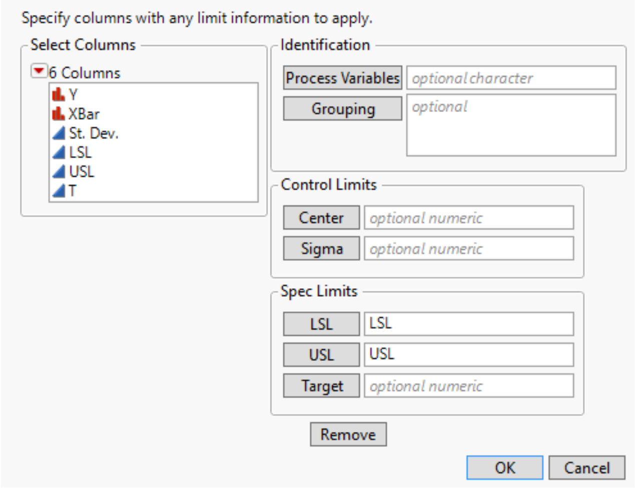
When one or more outliers influence the location of the center line, many subgroups can appear out of control. Using the median alleviates this problem.

Note: When Use Medians instead of Means is selected, the results obtained from the Control Charts for Selected Items red triangle menu option will not match the Process Screening results. The Control Chart Builder has no method of using the median instead of the mean.

Limits Table

A Limits Table contains a row for each process defined by the Process Variables and Grouping variables in your table of process data. When you use a Limits Table, the Limits Specifications window enables you to specify variables with the roles listed below. You need not specify variables for all of these roles. All of these roles are optional.

Figure 18.6 Limits Specifications Window



Specify columns with any limit information to apply.

LSL: LSL
USL: USL

Columns in the Limits Table that have appropriate names or names that match the role buttons are auto-filled. For example, any column called “Process”, “Column”, or “Parameter” is auto-filled into the Process Variables list.

If you have control limits but do not have columns for Center or Sigma, then you can construct Center and Sigma columns in the Limits Table using formulas. For example:

Center = $(UCL + LCL)/2$

Sigma = $d^*(UCL - LCL)/6$

where *d* is the square root of the subgroup size.

Process Variables A column that contains values corresponding to the column names in your table of process data.

Grouping One or more columns that contain the values of the grouping variables for your table of process data.

Center A column containing values for the center line for each process. This is usually the historical process mean.

Sigma A column containing values for the within standard deviation for each process. This is usually the historical standard deviation.

LSL A column containing lower specification limits for each process.

USL A column containing upper specification limits for each process.

Target A column containing a target value for each process.

The Process Screening Report

The Process Screening report opens with a Summary table that contains results about process stability. It also contains capability results if you have provided specification limits. The processes and groups are initially sorted in decreasing order by Stability Ratio.

Tip: To sort the report by a column, click on the column name. A caret appears to the right of the column name. The direction of the caret indicates whether the sorting is descending or ascending. To change the order of the sorting, click on the column name again.

The control chart calculations in the Summary table include Nelson tests and a Range Limit Exceeded test. These tests assume the following about the control chart limits:

- The center line for the XBar or X control charts is given by the mean of all measurements or by the median of the observations if you use the Medians instead of Means option.
- Control limits are placed at three sigma units from the center line.
- Sigma is estimated using the conventions that correspond to the control chart type that you specified or, if you use Medians instead of Means, as described in “Use Medians instead of Means” on page 328.

Tip: The eight Nelson tests in the Process Screening platform follow the test settings in the Control Chart Builder platform preferences. You can customize the tests at File > Preferences > Platforms > Control Chart Builder.

The Summary table can contain the following information:

Column The columns that you entered as Process. There is a row for each distinct combination of Process and Grouping columns. This column is suppressed if there is only one process column.

Grouping Columns There is a report column for each column in the data table that you entered as Grouping. The levels of the Grouping columns are listed so that there is a unique row in the report table for each distinct combination of Process name and Grouping columns values.

Variability Contains the following columns:

Stability Ratio A measure of stability of the process. The stability ratio is defined as follows:

(Overall Sigma/Within Sigma)^2

A stable process has stability ratio near one. Higher values indicate less stability.

Within Sigma An estimate of the standard deviation based on within subgroup variation. The estimate is based on the control chart type that you specified, and is a short-term measure of variation. See the Process Capability chapter in the *Quality and Process Methods* book for statistical details. If you select Medians instead of Means, Within Sigma is computed as described in "Use Medians instead of Means" on page 328.

Overall Sigma The usual estimate of standard deviation based on all observations.

Summary Contains the following columns:

Centerline (Appears only if you select Use Medians instead of Means in the launch window or if you import a Center value using a limits table.) The value listed under Centerline is used in control chart calculations as the center line.

- If you select Use Medians instead of Means in the launch window, the overall median of the observations is displayed.
- If you import a Center value from a limits table, that value is displayed.

Mean The average of all observations.

Count The number of observations.

Subgroups The number of subgroups.

Control Chart Alarms Contains information on the subgroups that result in alarms for a variety of tests, including each of the 8 Western Electric rules. In the descriptions below, for an Indiv and MR chart, the single measurement is considered a subgroup of size 1. The standard deviation estimate is the Within Sigma value. By default, only the Alarm Rate, Test 1, and Latest Alarm columns are shown in the Summary table.

Alarm Rate The number of subgroups that resulted in alarms for any of the tests selected under the Choose Test option (Any Alarm) divided by the number of subgroups (Subgroups).

Any Alarm (Appears only when more than one Test column is shown.) The number of subgroups that trigger alarms for any of the tests selected under the Choose Test option. These are the eight Nelson tests and the test for Range Limit Exceeded.

Tip: The eight Nelson tests in the Process Screening platform follow the test settings in the Control Chart Builder platform preferences. You can customize the tests at File > Preferences > Platforms > Control Chart Builder.

Test 1 One point is more than three standard deviations from the center line. The subgroup associated with that point triggers the alarm.

Test 2 Nine or more consecutive points are on the same side of the center line. The subgroup associated with the ninth point triggers the alarm.

Test 3 Six or more consecutive points are continually increasing or decreasing. The subgroup associated with the sixth point triggers the alarm.

Test 4 Fourteen consecutive points alternate in direction: increasing and then decreasing or decreasing and then increasing. The subgroup associated with the fourteenth point triggers the alarm.

Test 5 Two out of three consecutive points on the same side of the center line are more than two standard deviations from the center line. The subgroup associated with the second point that exceeds two standard deviations triggers the alarm.

Test 6 Four out of five consecutive points on the same side of the center line are more than one standard deviation from the center line. The subgroup associated with the fourth point that exceeds one standard deviation triggers the alarm.

Test 7 Fifteen consecutive points, on either side of the center line, are all within one standard deviation of the center line. The subgroup associated with the fifteenth point triggers the alarm.

Test 8 Eight consecutive points, on either side of the center line, all fall beyond one standard deviation of the center line. The subgroup associated with the eighth point triggers the alarm.

Range Limit Exceeded The number of subgroups that exceed the upper control limit on the R, S, or MR chart calculation.

Latest Alarm The position of the subgroup, counting from the last subgroup, that signaled the most recent alarm for any of the Nelson or Range Limit Exceeded tests.

Capability (Appears only when there are Spec Limits specified for some processes.) Contains the following options:

Out of Spec Count The number of observations that fall outside the specification limits.

Out of Spec Rate The proportion of observations that fall outside the specification limits.

Latest Out of Spec The number of observations, counting from the last observation, to the most recent observation that falls outside the specification limits.

Cpk Capability index based on Within Sigma and assuming a normal distribution. See the Process Capability chapter in the *Quality and Process Methods* book for statistical details.

Ppk Capability index based on Overall Sigma and assuming a normal distribution. See the Process Capability chapter in the *Quality and Process Methods* book for statistical details.

Shift Magnitudes and Positions (Shown only if you have selected a Shift Detection option from the Process Screening red triangle menu.) Shift detection is performed to identify shifts that exceed one within-sigma unit. The algorithm uses outlier-correction and an exponentially-weighted moving average (EWMA) smoothing approach for the individual observations.

- Outliers are removed so that single outliers do not indicate shifts. The value specified as Outlier Threshold (five by default) on the launch window controls the sensitivity of outlier removal. If the number of within-sigma units from an observation to both of its neighboring observations exceeds the specified Outlier Threshold, that observation is replaced with a value that is one within-sigma unit away from its closest neighboring observation.
- An exponentially-weighted moving average (EWMA) fit is constructed for the subgroup means in forward time order and another EWMA fit is constructed for the subgroup means in reverse time order. The EWMA fits have lambda equal to 0.3.
- The largest positive and negative differences between successive EWMA values that exceed one within-sigma unit are identified.
- The absolute values of these differences, divided by the within estimate of sigma, are the values reported as Largest Upshift and Largest Downshift.
- The locations of the first subgroups involved in these largest shifts define the Upshift Position and Downshift Position.

Largest Upshift The magnitude of the largest upward shift that exceeds one within-sigma unit, reported in within-sigma units.

Upshift Position or Upshift <Time Variable> The position of the subgroup having the largest Upshift. If you specify a Time variable, the column in the Summary table is named Upshift <Time Variable> and the position of the shift is given in terms of the Time variable.

Largest Downshift The magnitude of the largest downward shift that exceeds one within-sigma unit, reported in within-sigma units.

Downshift Position or Downshift <Time Variable> The position of the subgroup having the largest Downshift. If you specify a Time variable, the column in the Summary table is named Downshift <Time Variable> and the position of the shift is given in terms of the Time variable.

Process Screening Platform Options

The Process Screening red triangle menu contains options to customize the display and to save calculated statistics.

Summary Shows or hides the Summary table. See "The Process Screening Report" on page 330.

Find and Select Enter search strings for columns that you entered as Process Variables or Grouping in the launch window. A panel appears for each column. The corresponding processes are selected in the Summary table.

Quick Graph for Selected Items Plots small graphs of the processes that you select in the Summary table in a Graphs of Selected Items report. The report makes it possible to view and compare many processes at once. The plots are ordered according to their order of entry on the launch window.

Control Charts for Selected Items Opens a Control Chart Builder report window for the processes that you selected in the Summary table. The control chart corresponds to your selections in the Process Screening launch window.

Process Capability for Selected Items Opens a Process Capability report window showing Individual Detail Reports for the processes that you select in the Summary table. If you select a process for which specification limits are not specified, a Spec Limits window appears. In this window, you can specify specification limits by selecting a data table or entering values directly.

The Process Capability analysis assumes normal distributions and uses within sigma values that correspond to your Control Chart Type selection in the Process Screening launch window:

- Moving range for Indiv and MR
- Average of ranges for XBar and R
- Average of unbiased standard deviations for XBar and S

Color Selected Items Applies a color of your choosing to the values in the selected rows of the Summary table.

Show Tests Shows or hides test results for Nelson tests that are selected under the Choose Tests option in the Process Screening report's summary table.

Choose Tests Enables you to choose the tests that you want to include in the calculation of Alarm Rate and Any Alarm.

Tip: To select multiple tests, press the Alt key and click the Process Screening red triangle to open a menu of all platform options.

Shift Detection Provides options for detecting shifts after outlier removal. See "Shift Magnitudes and Positions" on page 333.

Largest Upshift Adds columns for Largest Upshift and Upshift Position to the Summary table. The largest upward shift in the series that exceeds one within-sigma unit is identified. See "Shift Magnitudes and Positions" on page 333.

Largest Downshift Adds columns for Largest Downshift and Downshift Position to the Summary table. The largest downward shift in the series that exceeds one within-sigma unit is identified. See "Shift Magnitudes and Positions" on page 333.

Shift Graph (Available only when some processes have shifts that exceed the Shift Threshold.) Shows a plot of the time occurrence of all process shifts that exceed the number of within-sigma units specified by the Shift Threshold (three by default). Green markers indicate upshifts and red markers indicate downshifts. The markers are located at the local peaks of the shifts.

To identify the processes that correspond to one or more shift occurrences, select the points and click Select Process. The corresponding processes are selected in the Summary table. Processes that have no shifts exceeding the Shift Threshold number of within-sigma units are not plotted.

Note: The Shift Graph does not show the positions of Largest Upshift and Largest Downshift values that appear in the Summary table if the shifts are less than the specified Shift Threshold number of within-sigma units in magnitude. See "Additional Example of Process Screening" on page 336.

Show Shifts in Quick Graphs (Available only when a Shift Graph has been added to the report window.) Shows the location of the shifts in the Quick Graphs using green and red vertical lines.

Process Performance Graph Shows a four-quadrant graph that assesses the performance of processes in terms of stability and capability. Each process for which specification limits are provided is represented by a point. Its horizontal coordinate equals the stability ratio of the process and its vertical coordinate equals the capability of the process, given as Ppk. The graph is divided into four quadrants based on the following default boundaries:

– A stability ratio that exceeds 1.5 indicates that the process is unstable.

– A Ppk that is smaller than 1.33 indicates that the process is not capable.

Selecting points in the graph selects the corresponding processes in the Summary table.

Process Performance Graph Boundaries Opens a window where you can set your desired values for the Process Performance Graph's stability ratio and Ppk capability boundaries.

Tip: You can set preferences for your desired boundaries for Stability Ratio and Ppk Capability in File > Preferences.

Save Summary Table Saves all of the information that can appear in the Summary table to a Process Summary data table. The Process Summary data table also contains specification limit details, if these are specified for at least one process.

Save Details Table Saves detailed information about control chart calculations to a Process Details data table. For each combination of a process and grouping variables, the table contains a row for each subgroup showing:

- The values of the subgroup sample statistics.
- The control limits.
- The subgroup size.
- A list of indicators for which, if any, alarms were triggered. Alarms for the Nelson tests are indicated with the numbers for the tests. An alarm for the Range Limit Exceeded test is indicated with an R.

Save Selected Details For the selected rows in the Summary table, saves a Process Details table with the information that is saved when you select Save Details Table.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Additional Example of Process Screening

The following example illustrates the use of a Grouping column and the construction of a Shift Graph using Consumer Prices.jmp.

The consumer price index data table contains monthly data on 17 products. The time periods vary by product. The data are arranged so that all 17 products are listed in a single column called Series. To separate the products, you must treat Series as a Grouping column.

1. Select **Help > Sample Data Library** and open Consumer Prices.jmp.
2. Select **Analyze > Screening > Process Screening**.
3. Select Price and click **Process Variables.**
4. Select Series and click **Grouping**.

   This ensures that each level of Series is treated as a separate process.
5. Select Date and click **Time**.
6. Set the **Control Chart Type** to XBar and R.
7. Set the **Subgroup Sample Size** to 3.

   Because the data are given monthly, subgroups of size three represent quarters.
8. Click **OK**.
9. Press Alt and click the red triangle next to Process Screening.

   This opens a window showing the available red triangle options. You can select multiple options at once in this window.
10. Check the following: **Largest Upshift**, **Largest Downshift**, and **Shift Graph**.
11. Click **OK**.

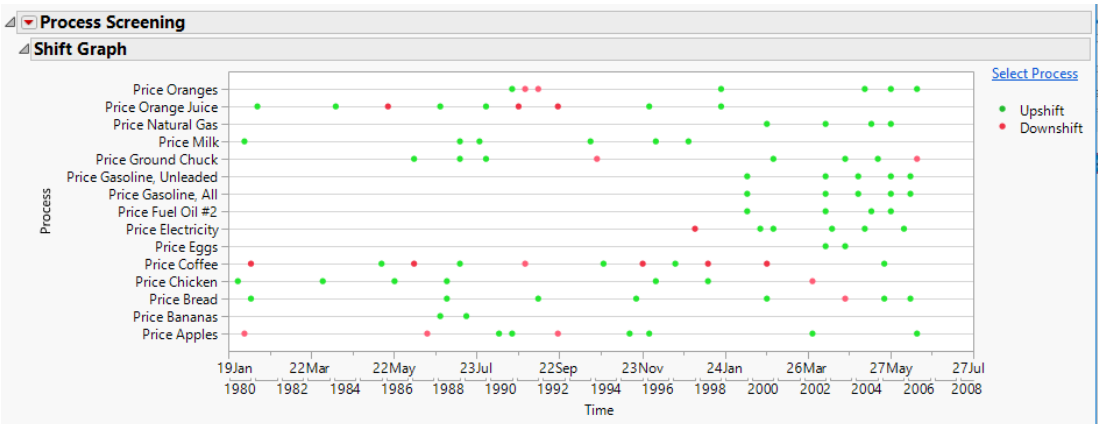
    The columns Largest Upshift, Upshift Date, Largest Downshift, and Downshift Date are added to the Summary table. The shifts are the largest shifts exceeding *one* within-sigma unit. The position of each shift is given in terms of the Time variable, Date. See“Shift Magnitudes and Positions” on page 333.

    A Shift Graph also appears. The Shift Graph shows all shifts that exceed the number of *Shift Threshold* within-sigma units, which is set to three by default. See “Shift Graph” on page 335. Green points correspond to upshifts and red points correspond to downshifts.

    Notice that Gasoline, All has values for both Largest Upshift and Largest Downshift in the Summary table. The Largest Downshift value, 1.8296, is less than three. Because the Shift Graph only shows shifts of three or more within-sigma units, the Largest Downshift value for Gasoline, All is not plotted on the Shift Graph.

    Also notice that Tomatoes is not included on the Shift Graph. For Tomatoes, no shifts of three or more within-sigma units were found.
12. Double click on the horizontal axis of the Shift Graph to open the X Axis Settings window.
13. In the Tick/Bin Increment panel, set **# Minor Ticks** to 1.
14. Set **Label Row Nesting** to 2.
15. Click **OK**.

Figure 18.7 Shift Graph



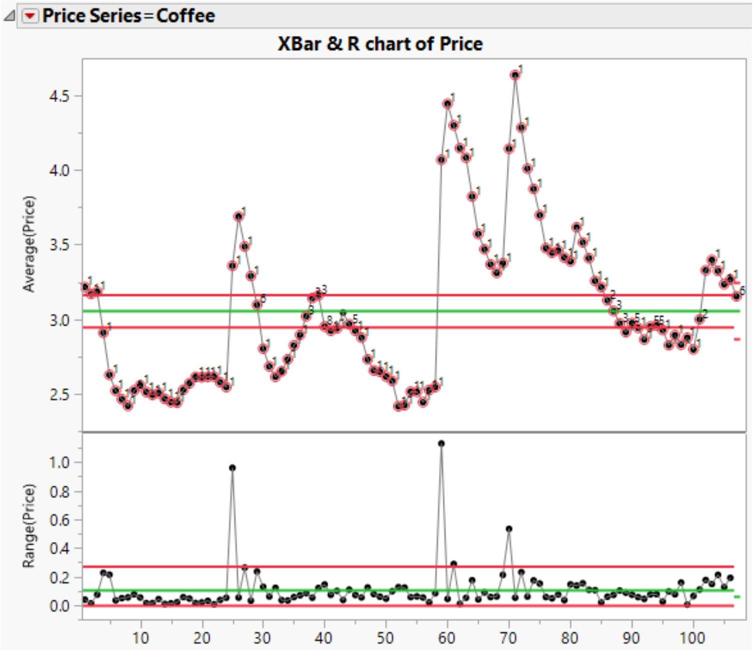
Most series show primarily upshifts. Price Coffee, however, has several alternating downshifts and upshifts. To better understand this series, obtain a control chart.

16. Select any point to the right of Price Coffee in the Shift Graph and click **Select Process**.

    This action selects the row of the Summary table corresponding to Coffee.

17. Click the Process Screening red triangle and select **Control Charts for Selected Items**.

Figure 18.8 Control Chart for Coffee



The control chart shows the upshifts and downshifts that are identified in the Shift Graph.

The Summary table indicates that the largest upshift (25.399 within-sigma units) occurs for the subgroup that includes September 1994. In the control chart in Figure 18.8, this is the subgroup in position 59. The Summary table also indicates that the largest downshift (9.1674 within-sigma units) occurs for the subgroup that includes March 1981. This is the subgroup in position 5 in the control chart.

Because shifts are calculated using EWMA-smoothed series and an outlier-correction algorithm, the shift positions might not precisely correspond to the subgroups that seem to start the shifts on a Shewhart control chart.

Statistical Details

Scaling Factors for Using Medians to Estimate Sigma

When you select Use Medians instead of Means, sigma is estimated using a scaled median range or median standard deviation. The table below gives the scaling factors, which were obtained using Monte Carlo simulation.

For subgroups of size *n* drawn from a normal distribution, the following are true:

- The theoretical median of the ranges is approximately *d2_Median**σ, where *d2_Median* is the value corresponding to *n*.
- The theoretical median of the standard deviations is approximately *c4_Median**σ, where *c4_Median* is the value corresponding to *n*.

Table 18.1 Scaling Constants for Median Range and Median Standard Deviation

n	d2_Median	c4_Median
2	0.953	0.675
3	1.588	0.833
4	1.978	0.888
5	2.257	0.917
6	2.471	0.933
7	2.646	0.944
8	2.792	0.952
9	2.915	0.959
10	3.024	0.963
11	3.118	0.967
12	3.208	0.969
13	3.286	0.972
14	3.357	0.975
15	3.422	0.976
16	3.483	0.978
17	3.539	0.979
18	3.590	0.980
19	3.640	0.981
20	3.685	0.982
21	3.731	0.983
22	3.770	0.984
23	3.811	0.984
24	3.846	0.985
25	3.883	0.986

Chapter 19

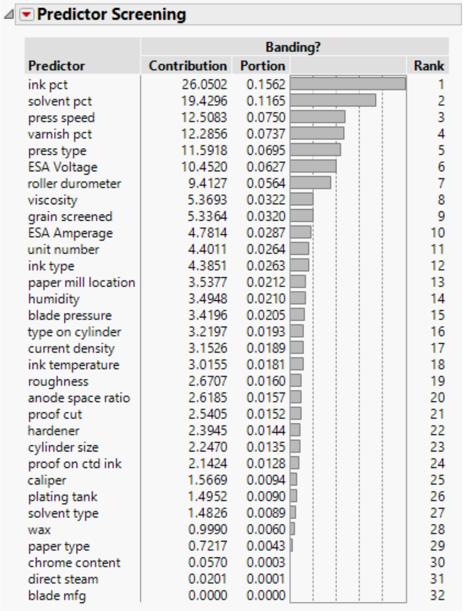
Predictor Screening

Screen Many Predictors for Significant Effects

The analysis of large data sets, where hundreds to thousands of measurements on a part, process, or sample are taken requires innovative approaches. The Predictor Screening platform provides a method of screening many predictors for their ability to predict an outcome. For example, predictor screening can be used to help identify biomarkers from thousands tested in samples from patients with and without a condition to predict the condition.

Predictor screening differs from response screening. Response screening tests factors one at a time as a predictor of the response. Predictor screening uses bootstrap forest partitioning to evaluate the contribution of predictors on the response. The partition models are built on multiple predictors. Predictor screening can identify predictors that might be weak alone but strong when used in combination with other predictors. See "Response Screening" chapter on page 289 for details.

Figure 19.1 Example of a Predictor Screening Report



Predictor Screening

	Banding?		
Predictor	Contribution	Portion	Rank
ink pct	26.0502	0.1562	1
solvent pct	19.4296	0.1165	2
press speed	12.5083	0.0750	3
varnish pct	12.2856	0.0737	4
press type	11.5918	0.0695	5
ESA Voltage	10.4520	0.0627	6
roller durometer	9.4127	0.0564	7
viscosity	5.3693	0.0322	8
grain screened	5.3364	0.0320	9
ESA Amperage	4.7814	0.0287	10
unit number	4.4011	0.0264	11
ink type	4.3851	0.0263	12
paper mill location	3.5377	0.0212	13
humidity	3.4948	0.0210	14
blade pressure	3.4196	0.0205	15
type on cylinder	3.2197	0.0193	16
current density	3.1526	0.0189	17
ink temperature	3.0155	0.0181	18
roughness	2.6707	0.0160	19
anode space ratio	2.6185	0.0157	20
proof cut	2.5405	0.0152	21
hardener	2.3945	0.0144	22
cylinder size	2.2470	0.0135	23
proof on ctd ink	2.1424	0.0128	24
caliper	1.5669	0.0094	25
plating tank	1.4952	0.0090	26
solvent type	1.4826	0.0089	27
wax	0.9990	0.0060	28
paper type	0.7217	0.0043	29
chrome content	0.0570	0.0003	30
direct steam	0.0201	0.0001	31
blade mfg	0.0000	0.0000	32

Predictor Screening Platform Overview

Predictor screening is useful for the identification of significant predictors from a large number of candidates. Suppose you had hundreds of Xs and needed to determine which of those were most significant as predictors of an outcome.

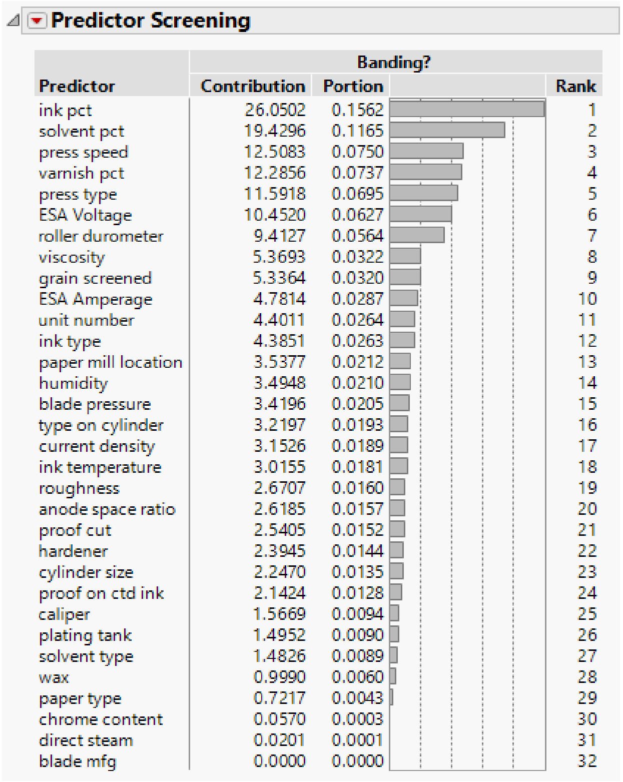
The predictor screening platform uses a bootstrap forest to identify potential predictors of your response. For each response, a bootstrap forest model using 100 decision trees is built. The column contributions to the bootstrap forest model for each predictor are ranked. Because the bootstrap forest method involves a random component, column contributions can differ when you rerun the report. For more information about decision trees, see the Partition Models chapter in the *Predictive and Specialized Modeling* book

Example of Predictor Screening

The Bands Data.jmp data table contains measurements from machinery in the rotogravure printing business. The data set contains 539 records and 38 variables. The response Y is the column Banding? and its values are "BAND" and "NOBAND". You are interested in understanding what properties are most likely to contribute to the response.

1. Select **Help > Sample Data Library** and open Bands Data.jmp.
2. Select **Analyze > Screening > Predictor Screening**.
3. Select Banding? as **Y, Response.**
4. Select the grouped columns grain screened to chrome content and click **X.**
5. Click **OK**.

Figure 19.2 Ranked Column Contributions



Predictor Screening

	Banding?			
Predictor	**Contribution**	**Portion**		**Rank**
ink pct	26.0502	0.1562		1
solvent pct	19.4296	0.1165		2
press speed	12.5083	0.0750		3
varnish pct	12.2856	0.0737		4
press type	11.5918	0.0695		5
ESA Voltage	10.4520	0.0627		6
roller durometer	9.4127	0.0564		7
viscosity	5.3693	0.0322		8
grain screened	5.3364	0.0320		9
ESA Amperage	4.7814	0.0287		10
unit number	4.4011	0.0264		11
ink type	4.3851	0.0263		12
paper mill location	3.5377	0.0212		13
humidity	3.4948	0.0210		14
blade pressure	3.4196	0.0205		15
type on cylinder	3.2197	0.0193		16
current density	3.1526	0.0189		17
ink temperature	3.0155	0.0181		18
roughness	2.6707	0.0160		19
anode space ratio	2.6185	0.0157		20
proof cut	2.5405	0.0152		21
hardener	2.3945	0.0144		22
cylinder size	2.2470	0.0135		23
proof on ctd ink	2.1424	0.0128		24
caliper	1.5669	0.0094		25
plating tank	1.4952	0.0090		26
solvent type	1.4826	0.0089		27
wax	0.9990	0.0060		28
paper type	0.7217	0.0043		29
chrome content	0.0570	0.0003		30
direct steam	0.0201	0.0001		31
blade mfg	0.0000	0.0000		32

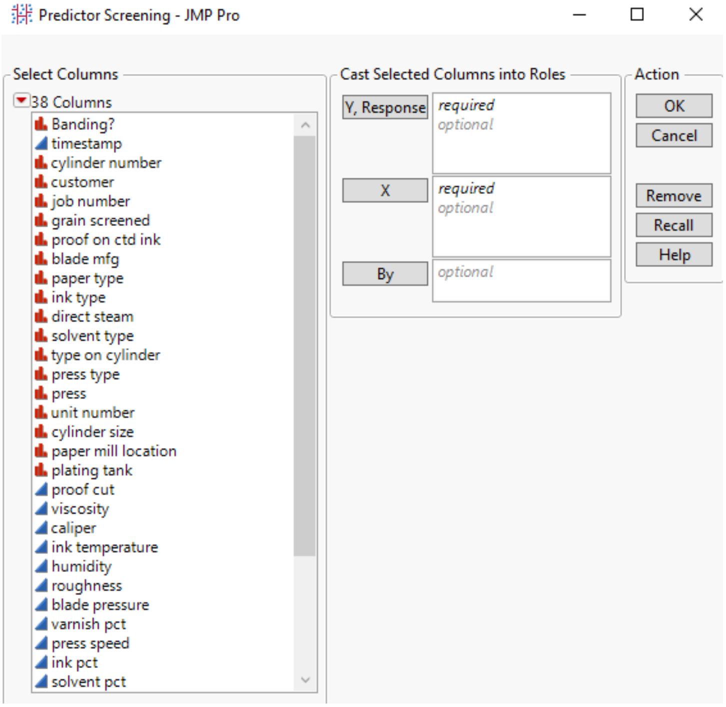
Note: Because this analysis is based on the Bootstrap Forest method that has a random selection component, your results can differ slightly from those in Figure 19.2. See "Bootstrap Forest" on page 107.

The columns are sorted and ranked in order of contribution in the bootstrap forest model. Predictors with the highest contributions are strong candidate predictors for the response of interest.

Launch the Predictor Screening Platform

Launch the Predictor Screening platform by selecting **Analyze > Screening > Predictor Screening**.

Figure 19.3 Predictor Screening Launch Window



Y, Response The response columns.

X Predictor columns.

By A column or columns whose levels define separate analyses. For each level of the specified column, the corresponding rows are analyzed using the other variables that you have specified. The results are presented in separate reports. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

The Predictor Screening Report

The report (Figure 19.2) shows the list of predictors with their respective contributions and rank. Predictors with the highest contributions are likely to be important in predicting Y.

The Contribution column shows the contribution of each predictor to the bootstrap forest model. The Portion column in the report shows the percent contribution of each variable.

You can select the important predictors in the Predictor Screening report. Selecting the important predictors selects the corresponding columns in the data table, enabling you to easily enter these columns into the launch windows for modeling platforms. In this fashion, the Predictor Screening enhances the modeling process

Predictor Screening Platform Options

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

Chapter 20

Association Analysis

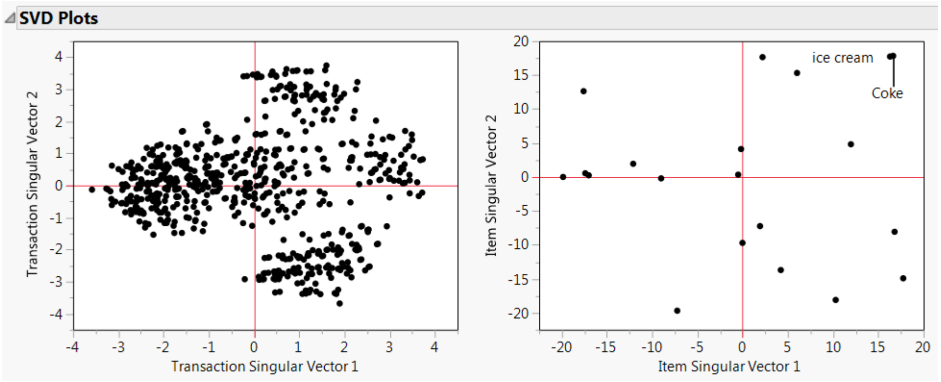
Perform Market Basket Analysis

The Association Analysis platform is available only in JMP Pro.

Association analysis enables you to identify items that have an affinity for each other. It is frequently used to analyze transactional data (also called *market baskets*) to identify items that often appear together in transactions. For example, grocery stores and online merchants use association analysis to strategically organize and recommend products that tend to be purchased together.

Association analysis is also used for identifying dependent or associated events. For example, you can identify car parts that seem to fail around the same time. In this application, car inspections are treated as the market baskets and you analyze the associations among groups of faulty parts found in each inspection.

Figure 20.1 Example of Singular Value Decomposition Plots



Association Analysis Platform Overview

The Association Analysis platform identifies connections among groups of items in an independent event or *transaction*. In association analysis, an *item* is the basic object of interest. For example, an item could be a product, a web page, or a service. An *item set* is a list of one or more items.

The relationship between two item sets is defined by an *association rule*. An association rule consists of a *condition* item set and a *consequent* item set. *Antecedents* are the individual items in the condition item set. Association analysis identifies association rules, which predict that a consequent item set will be in a transaction, given that the condition item set is already in the transaction. Some association rules are stronger, and therefore more useful, than others. The following three performance measures describe the strength of an association rule:

- *Support* is the proportion of transactions in which an item set appears. A high value for support indicates that the item set occurs frequently.
- *Confidence* is the proportion of transactions that contain the consequent item set, given that the condition item set is in the transaction. Confidence measures the strength of implication, or the predictive power, of an association rule.
- *Lift* is the ratio of an association rule's confidence to its expected confidence, assuming that the condition and consequent item sets appear in transactions independently. Lift measures how much the consequent item set depends on the presence of the condition item set. The minimum value for lift is 0.
  - A lift ratio less than 1 indicates that the condition and consequent repel each other, because they occur together less frequently than one would expect by chance alone.
  - A lift ratio close to 1 indicates that the consequent occurs at the same rate in transactions that contain the condition as one would expect from chance alone.
  - A lift ratio greater than 1 indicates that the consequent item set has an affinity for the condition item set. The consequent item set occurs more often with the condition item set than one would expect by chance alone.

For more information about these performance measures, see "Association Analysis Performance Measures" on page 360.

The Association Analysis platform also enables you to perform singular value decomposition. Singular value decomposition (SVD) groups similar transactions and also groups similar items using a matrix reducing methodology that is different from what is used in association analysis. Use the SVD methodology to gain insights that complement what you learn from association analysis.

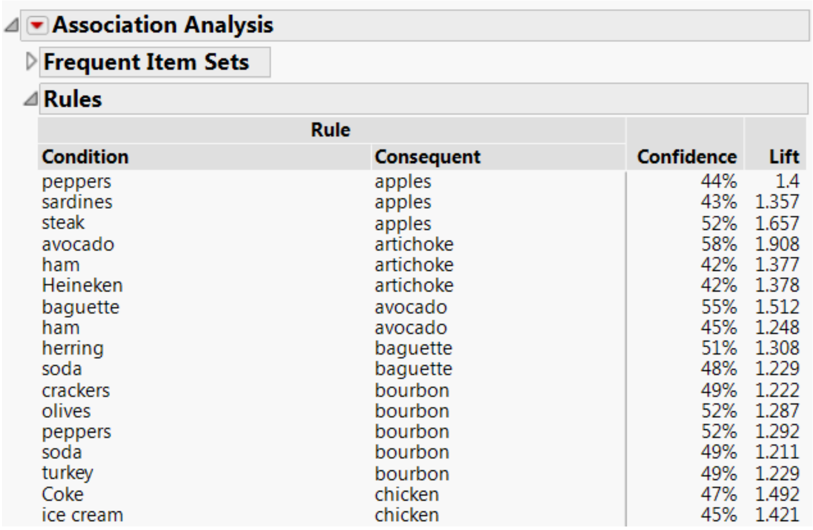
For more information about association analysis, see Hastie et al. (2009) and Shmueli et al. (2010). For more information about singular value decomposition, see Jolliffe (2002).

Example of the Association Analysis Platform

This example uses the Grocery Purchases.jmp sample data table, which contains transactional data reported by a grocery store. The data table lists the items purchased by 1001 customers, each assigned a unique customer ID. You want to explore the associations among items in order to identify patterns in consumer behavior.

1. Select **Help > Sample Data Library** and open Grocery Purchases.jmp.
2. Select **Analyze > Screening > Association Analysis**.
3. Select Product and click **Item**.
4. Select Customer ID and click **ID**.
5. Click **OK**.

Figure 20.2 Association Analysis Report

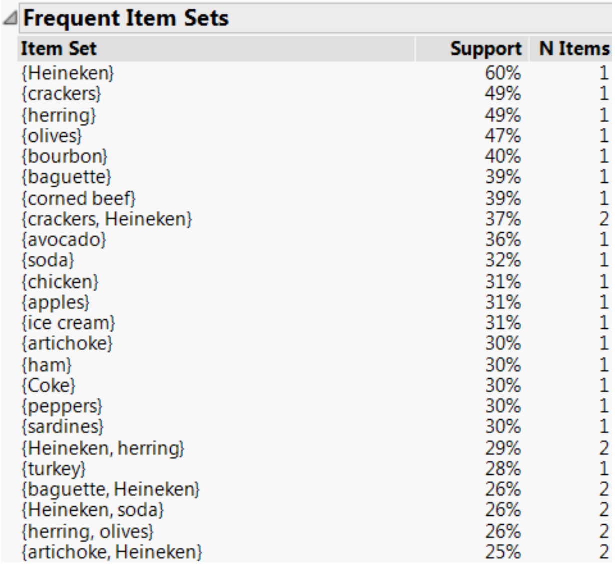


Rule			
Condition	Consequent	Confidence	Lift
peppers	apples	44%	1.4
sardines	apples	43%	1.357
steak	apples	52%	1.657
avocado	artichoke	58%	1.908
ham	artichoke	42%	1.377
Heineken	artichoke	42%	1.378
baguette	avocado	55%	1.512
ham	avocado	45%	1.248
herring	baguette	51%	1.308
soda	baguette	48%	1.229
crackers	bourbon	49%	1.222
olives	bourbon	52%	1.287
peppers	bourbon	52%	1.292
soda	bourbon	49%	1.211
turkey	bourbon	49%	1.229
Coke	chicken	47%	1.492
ice cream	chicken	45%	1.421

The fourth entry in the Rules report table indicates that 58% of customers who bought an avocado also bought an artichoke. The value of Lift is 1.908, indicating that there is a likely dependency. You want to verify that avocados and artichokes occur in a significant portion of transactions.

6. Click the disclosure icon next to Frequent Item Sets.

Figure 20.3 Frequent Item Sets Report



Frequent Item Sets

Item Set	Support	N Items
{Heineken}	60%	1
{crackers}	49%	1
{herring}	49%	1
{olives}	47%	1
{bourbon}	40%	1
{baguette}	39%	1
{corned beef}	39%	1
{crackers, Heineken}	37%	2
{avocado}	36%	1
{soda}	32%	1
{chicken}	31%	1
{apples}	31%	1
{ice cream}	31%	1
{artichoke}	30%	1
{ham}	30%	1
{Coke}	30%	1
{peppers}	30%	1
{sardines}	30%	1
{Heineken, herring}	29%	2
{turkey}	28%	1
{baguette, Heineken}	26%	2
{Heineken, soda}	26%	2
{herring, olives}	26%	2
{artichoke, Heineken}	25%	2

The Frequent Item Sets report shows that 36% of customers purchased avocados. The Rules report in Figure 20.2 shows that 58% of these customers also bought artichokes. Because of the large proportion of customers who follow this behavior, the grocery store management might use this information to strategically locate avocados and artichokes.

You also decide to look at the association rules with the highest lift.

7. Right-click in the Rules report table and select **Sort By Column**.

   The Select Columns window appears.

8. Select Lift and click **OK**.

   The Rules table is sorted by decreasing values of lift. Notice that the second association rule has a lift of 6.912 and 97% confidence. You want to verify that both the condition set, {Coke, Heineken, sardines}, and the consequent item set, {chicken, ice cream}, have adequate support.

9. Right-click in the Frequent Item Sets report and select **Sort By Column**.

   The Select Columns window appears.

10. Select Item Set and the check the ascending order option.

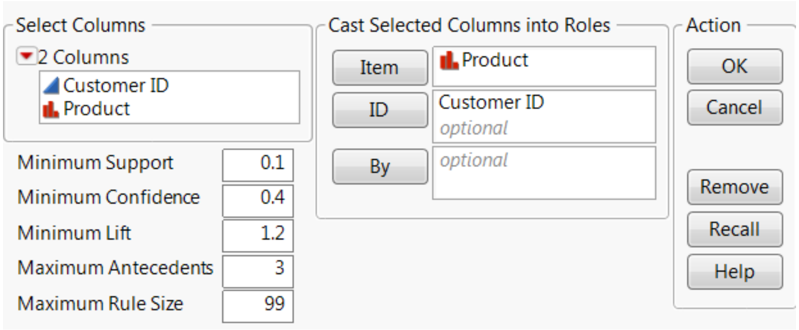
11. Click **OK**.

    The Frequent Item Sets table is sorted alphabetically by item set. Scroll through the list to see that the condition item set, {Coke, Heineken, sardines}, has 12% support and that the consequent item set, {chicken, ice cream}, has 14% support. This association rule has high lift, but represents fewer transactions than the first association rule that you examined.

Launch the Association Analysis Platform

Launch the Association Analysis platform by selecting **Analyze > Screening > Association Analysis**.

Figure 20.4 Association Analysis Launch Window



Select Columns
2 Columns
Customer ID
Product

Cast Selected Columns into Roles
Item: Product
ID: Customer ID, optional
By: optional

Minimum Support 0.1
Minimum Confidence 0.4
Minimum Lift 1.2
Maximum Antecedents 3
Maximum Rule Size 99

Item The categorical column that contains the item data to be analyzed.

ID The column that identifies the transaction that an item belongs to.

By Produces a separate report for each level of the By variable. If more than one By variable is assigned, a separate report is produced for each possible combination of the levels of the By variables.

Minimum Support Specifies a minimum value for the proportion of occurrences of an item set. This value must be between 0 and 1. Only item sets with support equal to or exceeding this value are considered in the analysis.

Minimum Confidence Specifies a minimum value for the proportion of occurrences that a consequent item set occurs within transactions that contain the conditional item set. This value must be between 0 and 1. Only association rules with confidence equal to or exceeding this value appear in the report.

Minimum Lift Specifies a minimum dependency ratio. Lift values must be 0 or greater. Only association rules with lift equal to or exceeding this value appear in the report.

Maximum Antecedents Specifies the maximum number of items in the condition item set. Association rules with more than this number of items in the condition set are not considered in the analysis.

Maximum Rule Size Specifies the maximum number of items that appear in the union of the condition and consequent item sets. Association rules with more than this combined number of items are not considered in the analysis.

Note: You can use the minimum support, maximum antecedent, and maximum rule size options in the launch window to reduce computational time for large data sets. For more information about these measures, see "Statistical Details for the Association Analysis Platform" on page 360.

The Association Analysis Report

By default, the Association Analysis report contains the following reports:

- "Frequent Item Sets" on page 352
- "Rules" on page 352

Tip: To order the contents of a table in a report by any of its columns, right-click in the table and select **Sort by Column**.

Frequent Item Sets

The Frequent Item Sets report lists item sets in decreasing order of support. The listed item sets meet the Minimum Support value that you specified in the launch window. Each item set is considered as a conditional and as a consequent item set to form association rules. The table contains the following columns:

Item Set The item sets that are considered as conditional or consequent sets for the association rules.

Support The proportion of transactions in which all of the items in the Item Set occur.

N Items The number of items in the Item Set.

Rules

The Rules report shows a table of association rules that are sorted in increasing order of number of items in the condition item set. The rules are further sorted alphabetically by the items contained in the union of the condition and consequent item sets. Only association rules that meet the Minimum Support, Minimum Confidence, Minimum Lift, Maximum Antecedents, and Maximum Rule Size requirements that you specified in the launch window appear in this report.

The Rules report table contains the following columns:

Rule The association rules formed by combining Condition and Consequent item sets.

Condition The item set that is thought to influence the presence of a Consequent item set within transactions.

Consequent The item set whose presence is thought to be influenced by the presence of a Condition item set.

Confidence The proportion of transactions that contain the Consequent item set, given that the condition item set is in the transaction. Confidence measures the strength of implication, or the predictive power, of an association rule.

Lift •The ratio of an association rule's confidence to its expected confidence, assuming that the condition and consequent item sets appear in transactions independently. Lift measures how much the Consequent item set depends on the presence of the Condition item set. The minimum value for lift is 0.

- A lift ratio less than 1 indicates that the Condition and Consequent item sets repel each other, because they occur together less frequently than one would expect by chance alone.
- A lift ratio close to 1 indicates that the Consequent item set occurs at the same rate in transactions that contain the Condition item set as one would expect from chance alone.
- A lift ratio greater than 1 indicates that the Consequent item set has an affinity for the Condition item set. The Consequent item set occurs more often with the Condition item set than one would expect by chance alone.

Association Analysis Platform Options

The Association Analysis red triangle menu contains the following options:

Transaction Listing Shows or hides a table listing each Transaction ID value and the items included in that transaction. The table is sorted by the Transaction ID column.

Frequent Item Sets Shows or hides a list of item sets whose support exceeds the Minimum Support value specified in the launch window. See "Frequent Item Sets" on page 352 for more information.

Rules Shows or hides a table of association rules that meet the Minimum Support, Minimum Confidence, Minimum Lift, Maximum Antecedents, and Maximum Rule Size requirements specified in the launch window. See "Rules" on page 352 for more information.

SVD Shows or hides scatterplots of the first two singular vectors for transactions and for items, calculated by singular value decomposition on the incidence matrix for the items. The report also contains a table of singular values sorted in descending order. The Percent and Cum Percent columns show the additional and cumulative variability in the data explained by the corresponding singular value. The bar chart shows the Percent variation explained by each singular value. For more information, see "SVD" on page 354.

Rotated SVD (Available only if SVD is selected.) Shows or hides the Topic Items and Topic Scores reports. This option performs a varimax rotated singular value decomposition of the transaction item matrix to produce groups of similar transactions called topics. See "Rotated SVD" on page 356.

Save Transaction SVD Creates a data table that contains a number of singular vectors that you specify for each transaction. These are the left singular values in the transaction item matrix. See "Singular Value Decomposition" on page 355.

Save Item SVD Creates a data table that contains a number of singular vectors that you specify for each item. These are the right singular values in the transaction item matrix. See "Singular Value Decomposition" on page 355.

See the JMP Reports chapter in the *Using JMP* book for more information about the following options:

Local Data Filter Shows or hides the local data filter that enables you to filter the data used in a specific report.

Redo Contains options that enable you to repeat or relaunch the analysis. In platforms that support the feature, the Automatic Recalc option immediately reflects the changes that you make to the data table in the corresponding report window.

Save Script Contains options that enable you to save a script that reproduces the report to several destinations.

Save By-Group Script Contains options that enable you to save a script that reproduces the platform report for all levels of a By variable to several destinations. Available only when a By variable is specified in the launch window.

SVD

Singular value decomposition (*SVD*) complements association analysis by providing another method to identify items that have an affinity for each other. Singular value decomposition of the transaction item matrix reduces the matrix to a manageable number of dimensions, thereby enabling you to group similar transactions and similar items. The partial singular value decomposition in the Association Analysis platform is equivalent to performing principal components analysis (PCA).

Transaction Item Matrix

The *transaction item matrix* is a matrix for which each row corresponds to a transaction each column corresponds to an item. The entries of the matrix are zeros and ones. If an item occurs in a transaction, the corresponding row and column entry is one. Otherwise, the row and column entry is zero. Because the transaction item matrix usually contains more values of zero than one, it is called a *sparse matrix*.

Singular Value Decomposition

The partial singular value decomposition approximates the transaction item matrix using three matrices: **U**, **S**, and **V‘**. The relationship between these matrices is defined as follows:

Transaction Item Matrix ≈ **U** * **S** * **V‘**

Define *nTransactions* as the number of transactions (rows) in the transaction item matrix, and *nItems* as the number of items (columns) in the transaction item matrix, and *nVec* as the specified number of singular vectors. Note that *nVec* must be less than or equal to min(*nTransactions, nItems*). It follows that **U** is an *nTransactions* by *nVec* matrix. **S** is a diagonal matrix of dimension *nVec*. The diagonal entries in **S** are the singular values of the transaction item matrix. **V‘** is an *nVec* by *nTransactions* matrix. The rows in **V‘** are the singular vectors.

The singular vectors capture connections among different items with similar functions or topic areas. If three items tend to appear in the same transactions, the SVD is likely to produce a singular vector in **V‘** with large values for those three items. The **U** singular vectors represent the transactions projected into this new item space.

The SVD also captures indirect connections. If two items never appear together in the same transaction, but they generally appear in transactions with another third item, the SVD is able to capture some of that connection. If two transactions have no items in common but contain items that are connected in the dimension-reduced space, they map to similar vectors in the SVD plots.

The SVD transforms transaction data into a fixed-dimensional vector space, making it amenable to clustering, classification, and regression techniques. The Save options enable you to export this vector space to be analyzed in other JMP platforms.

The transaction item matrix is centered, scaled, and divided by *nTransactions* minus 1 before the singular value decomposition is carried out. This analysis is equivalent to a PCA of the correlation matrix of the transaction item matrix.

SVD Report

SVD Plots

The SVD Plots report shows scatterplots of the first two singular vectors for both the transaction and the item data.

Tip: To see the transaction or item that a point represents, place your cursor over the point. To add the label to the plot, select the point, right-click in the plot, and select **Row Label**.

The *Transaction SVD plot* contains a point for each transaction. For a given transaction, the point that is plotted is defined by the transaction's values on the first two singular vectors in **U**. In the Transaction SVD plot, points that are visibly grouped together indicate transactions with a similar composition. This plot is equivalent to the Score Plot in the Principal Components platform.

The *Item SVD plot* contains a point for each item. For a given item, the point that is plotted is defined by the item's values on the first two singular vectors in **V**. In the Item SVD plot, items that are visibly grouped together indicate items that have similar functions or topic areas. This plot is equivalent to the Loadings Plot in the Principal Components platform.

See "Additional Example: SVD Analysis" on page 358.

Caution: The first two singular vectors might not adequately capture the structure of your data. The "Singular Values" report shows how much variability is explained by the singular vectors.

Singular Values

Below the transaction and item SVD plots, a table of the singular values appears. These are the diagonal entries of the **S** matrix in the singular value decomposition of the transaction item matrix. The k^{th} row in the Singular Values table shows the additional and cumulative percent of variability explained by using the k^{th} singular value or singular vector column. Like in the Principal Components platform, you can use the Cum Percent column to decide what percent of variance from the transaction item matrix you want to preserve, and then use the corresponding number of singular vectors.

Rotated SVD

(Available only when SVD is selected from the red triangle menu next to Association Analysis.) The Rotated SVD option performs a varimax rotation on the singular value decomposition (SVD) of the transaction item matrix. See "Transaction Item Matrix" on page 354. You must specify a number of rotated singular vectors, which corresponds to the number of *topics* that you want to retain from the transaction item matrix. After you specify a number of topics, the Topic Terms and Topic Scores Plots reports appear. Topic analysis is equivalent to a rotated principal components analysis (rotated PCA).

Topics are groups of transactions that are grouped based on a primary item indicator, as well as secondary item indicators. For each topic, every item has a weight that influences a transaction's membership in the topic. The cumulative sum of the item weights for all of the items that are present in a transaction is called the *topic score*. Topic scores reflect the strength of a transaction's membership for a topic.

The varimax rotation rotates the singular vectors to more closely align them with the coordinate axes. This rotation helps facilitate interpretation by resulting in high loadings on a few axes and small loadings on the others. The loadings are given in the Rotated V Matrix and Rotated U Matrix reports.

See "Additional Example: SVD Analysis" on page 358.

Topic Items Report

(Available only when Rotated SVD is selected from the red triangle menu next to Association Analysis.) The Topic Items report shows the items that have the largest loadings in each topic after rotation. There are additional reports that show the components of the rotated singular value decomposition.

The report shows a table of items for each topic. The items in each table are the ones that have the largest loadings in absolute value for each topic. Each table is sorted in descending order by the absolute value of the loading. These tables can be used to determine conceptual themes that correspond to each topic.

The Topic Items report also contains the following matrix reports:

Variance Explained by Each Topic Contains a table of the variance explained by each topic. The table also contains columns for the percent and cumulative percent of the variation explained by each topic.

Term Topic Loadings Contains a matrix of the loadings across topics for each item. This matrix is equivalent to the transpose of the factor loading matrix in a rotated PCA.

Document Topic Scores Contains a matrix of transaction scores for each topic. Transactions with higher scores in a topic are more likely to be associated with that topic.

Rotation Matrix Contains the rotation matrix for the varimax rotation.

See "Additional Example: SVD Analysis" on page 358.

Topic Scores Plots Report

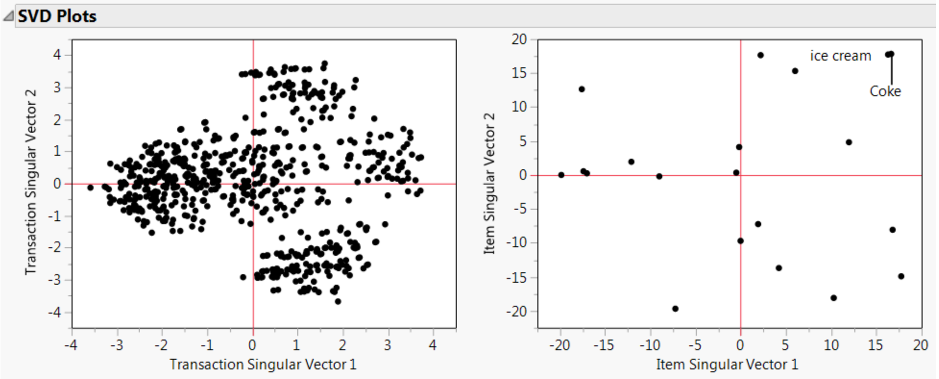
(Available only when Rotated SVD is selected from the red triangle menu next to Association Analysis.) The Topic Scores Plots report is a visual representation of the matrix in the Document Topic Scores report. Each panel in the plot corresponds to one of the topics, or one of the rows of the Document Topic Scores matrix. Within each panel, each point corresponds to one of the transactions in the data table, or one of the columns of the transaction item matrix. See "Additional Example: SVD Analysis" on page 358.

Additional Example: SVD Analysis

In this example, you use singular value decomposition of the transaction item matrix to gain further insight into the Grocery Purchases.jmp sample data.

1. Select **Help > Sample Data Library** and open Grocery Purchases.jmp.
2. Select **Analyze > Screening > Association Analysis**.
3. Select Product and click **Item**.
4. Select Customer ID and click **ID**.
5. Click **OK**.
6. Click the red triangle next to Association Analysis and select **SVD**.

Figure 20.5 SVD Plots

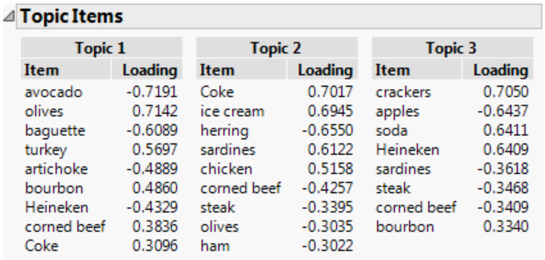


The transaction SVD plot suggests that there might be two or three groups of transactions. In the upper right corner of the item SVD plot, notice that the points that represent Coke and ice cream overlap. The proximity of these two items indicates that there is a strong affinity between them.

7. Click the red triangle next to Association Analysis and select **Rotated SVD**.
8. Enter 3 next to Number of Topics (rotated singular vectors) and click **OK**.

The Topic Items and Topic Scores Plots reports appear.

Figure 20.6 Topic Items Report

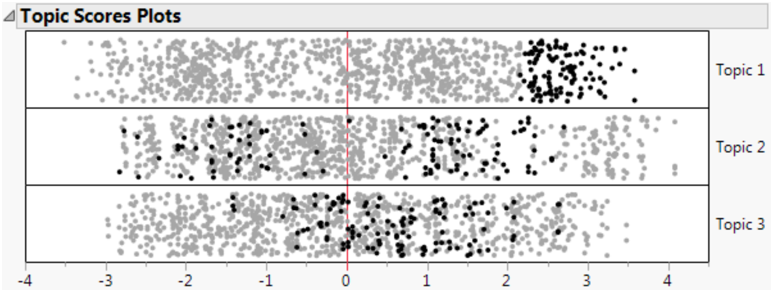


Topic Items

Topic 1		Topic 2		Topic 3	
Item	**Loading**	**Item**	**Loading**	**Item**	**Loading**
avocado	-0.7191	Coke	0.7017	crackers	0.7050
olives	0.7142	ice cream	0.6945	apples	-0.6437
baguette	-0.6089	herring	-0.6550	soda	0.6411
turkey	0.5697	sardines	0.6122	Heineken	0.6409
artichoke	-0.4889	chicken	0.5158	sardines	-0.3618
bourbon	0.4860	corned beef	-0.4257	steak	-0.3468
Heineken	-0.4329	steak	-0.3395	corned beef	-0.3409
corned beef	0.3836	olives	-0.3035	bourbon	0.3340
Coke	0.3096	ham	-0.3022		

Three groups, or topics, are created and shown in the Topic Items report. The first items listed in the Topic Item tables represent the primary items for that group. For example, Topic 1 is a group that is identified primarily by transactions that contain olives, but do not contain avocados.

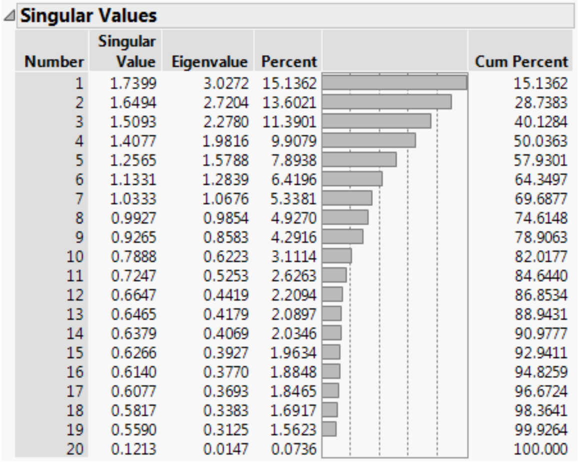
Figure 20.7 Topic Scores Plots



The topic scores that are assigned to each of the 1001 transactions are plotted in the Topic Scores report. Select groups of points for a topic to see how those transactions relate to other topics. For example, transactions with very high values on Topic 1 tend to have lower values on Topics 2 and 3.

9. Open the Singular Values report.

Figure 20.8 Singular Values Table



Singular Values

Number	Singular Value	Eigenvalue	Percent		Cum Percent
1	1.7399	3.0272	15.1362		15.1362
2	1.6494	2.7204	13.6021		28.7383
3	1.5093	2.2780	11.3901		40.1284
4	1.4077	1.9816	9.9079		50.0363
5	1.2565	1.5788	7.8938		57.9301
6	1.1331	1.2839	6.4196		64.3497
7	1.0333	1.0676	5.3381		69.6877
8	0.9927	0.9854	4.9270		74.6148
9	0.9265	0.8583	4.2916		78.9063
10	0.7888	0.6223	3.1114		82.0177
11	0.7247	0.5253	2.6263		84.6440
12	0.6647	0.4419	2.2094		86.8534
13	0.6465	0.4179	2.0897		88.9431
14	0.6379	0.4069	2.0346		90.9777
15	0.6266	0.3927	1.9634		92.9411
16	0.6140	0.3770	1.8848		94.8259
17	0.6077	0.3693	1.8465		96.6724
18	0.5817	0.3383	1.6917		98.3641
19	0.5590	0.3125	1.5623		99.9264
20	0.1213	0.0147	0.0736		100.000

As seen in Figure 20.8, the first two singular values explain only about 30% of the variability in the grocery store data. Additional dimensions might be required to explain a sufficient amount of variability.

Statistical Details for the Association Analysis Platform

This section contains statistical details for the Association Analysis platform.

Frequent Item Set Generation

The Association Analysis platform uses the *Apriori algorithm* to reduce computational time when generating frequent item sets. The Apriori algorithm leverages the fact that an item set's support is never larger than the support of its subsets. The platform generates larger item sets from combinations of smaller item sets that meet the minimum support level. In addition, the platform does not generate item sets that exceed either the specified maximum number of antecedents or the maximum rule size. These options are useful when working with large data sets, because the total possible number of rules increases exponentially with the number of items. For more information about the Apriori algorithm, see Agrawal and Srikant (1994).

Association Analysis Performance Measures

This section defines the performance measures used in Association Analysis. Denote the condition item set by X and the consequent item set by Y. Denote an association rule with condition set X and consequent set Y by $X \Rightarrow Y$.

Support

Support is the proportion of transactions in which an item set occurs.

$$Support(X) = \frac{\text{Number of Transactions Containing X}}{\text{Total Number of Transactions}}$$

Confidence

Confidence is the proportion of transactions that contain the consequent item set, given that the transaction contains the condition item set.

$$Confidence(X \Rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}$$

An association rule with a confidence of 0% has a consequent item set that does not appear in any transaction with the condition item set. A confidence of 100% indicates that every transaction that contains the condition item set also contains the consequent item set.

Lift

Lift measures dependency between X and Y.

$$Lift(X \Rightarrow Y) = \frac{Support(X \cup Y)}{Support(X) \times Support(Y)}$$

The numerator for lift is the proportion of transactions where X and Y occur jointly. The denominator is an estimate of the expected joint occurrence of X and Y, assuming that they occur independently.

A lift value of 1 indicates that X and Y jointly occur in transactions with the frequency that would be expected by chance alone. Increasing lift values suggest that Y occurs more often than expected when X is present.

Appendix A

References

Agrawal, R. and Srikant, R. (1994), "Fast Algorithms for Mining Association Rules," *Proceedings of the 20th VLDB Conference*. Santiago, Chile: IBM Almaden Research Center. Retrieved July 5, 2016 from http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf.

Bates, D.M. and Watts, D.G. (1988), *Nonlinear Regression Analysis & its Applications*. New York, John Wiley and Sons.

Benjamini, Yoav and Hochberg, Yosef (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society, Series B*, 57, 289–300.

Box, G., et al. (1994). *Time series analysis: forecasting and control*. New York, John Wiley and Sons.

Dwass, M. (1955), "A Note on Simultaneous Confidence Intervals," *Annals of Mathematical Statistics* 26: 146–147.

Farebrother, R.W. (1981), "Mechanical Representations of the L1 and L2 Estimation Problems," *Statistical Data Analysis*, Second Edition, Amsterdam, North Holland: edited by Y. Dodge.

Fieller, E.C. (1954), "Some Problems in Interval Estimation," *Journal of the Royal Statistical Society, Series B*, 16, 175-185.

Goodnight, J.H. (1978), "Tests of Hypotheses in Fixed Effects Linear Models," *SAS Technical Report R–101*, Cary: SAS Institute Inc, also in Communications in Statistics (1980), A9 167–180.

Goodnight, J.H. and W.R. Harvey (1978), "Least Square Means in the Fixed Effect General Linear Model," *SAS Technical Report R–103*, Cary NC: SAS Institute Inc.

Hand, D, Mannila, H, and Smyth, P. (2001), *Principles of Data Mining*, MIT Press.

Hastie, T., Tibshirani, R., and Friedman, J.H.(2009), *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, New York: Springer Science and Business Media.

Hawkins D.M., Kass G.V. (1982), "Automatic Interaction Detection," in: Hawkins D.M. ed. *Topics in Applied Multivariate Analysis*. Cambridge University Press, Cambridge

Hocking, R.R. (1985), *The Analysis of Linear Models*, Monterey: Brooks–Cole.

Huber, Peter J. and Ronchetti, Elvezio M. (2009), *Robust Statistics*, Second Edition, New York: John Wiley and Sons.

Jolliffe, I.T. (2002), *Principal Component Analysis*, Second Edition, New York, Springer-Verlag New York, Inc.

Kass GV (1980), “An Exploratory Technique for Investigating Large Quantities of Categorical Data,” *Applied Statistics*, 29:119–127

McCullagh, P. and Nelder, J.A. (1983), *Generalized Linear Models*, London: Chapman and Hall Ltd.

Nagelkerke, N.J.D. (1991), “A Note on a General Definition of the Coefficient of Determination,” *Biometrika*, 78:3 691-692.

Nelder, J.A. and Wedderburn, R.W.M. (1983), “Generalized Linear Models,” *Journal of the Royal Statistical Society*, Series A, 135, 370–384.

Parker, R. J. (2015), *Efficient Computational Methods for Large Spatial Data* [PhD Dissertation], North Carolina State University, Raleigh, NC. Retrieved June 30, 2016 from http://repository.lib.ncsu.edu/ir/bitstream/1840.16/10572/1/etd.pdf

Qian, P.Z., Huaiquing, W., and Wu, C.F. (2012). “Gaussian process models for computer experiments with qualitative and quantitative factors.” *Technometrics*, 50:3 383-396.

Ratkowsky, D.A. (1990), *Handbook of Nonlinear Regression Models*, New York, Marcel Dekker, Inc.

Sall, J. (2002), “Monte Carlo Calibration of Distributions of Partition Statistics,” SAS Institute. Retrieved July 29, 2015 from http://www.jmp.com/content/dam/jmp/documents/en/white-papers/montecarlocal.pdf.

Santer, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, New York, Springer-Verlag New York, Inc.

SAS Institute Inc. (2013), *SAS/ETS User’s Guide*, Version 13.1, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (2010), *SAS/ETS User’s Guide*, Version 9.22, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (2004), *SAS/STAT User’s Guide*, Version 9.1, Cary, NC: SAS Institute Inc.

Schuirmann, D. J. (1987), “A Comparison of the Two One-sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability,” *J. Pharmacokin. Biopharm.*, 15, 657–680.

Shiskin, J., Young, A.H., and Musgrave, J.C. (1967), “The X-11 Variant of the Census Method II Seasonal Adjustment Program,” *Technical Report 15*, U.S. Department of Commerce, Bureau of the Census.

Shmueli, G., Patel, N.R., Bruce, P.C (2010), *Data Mining For Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner*, Second Edition, Hoboken, New Jersey: John Wiley & Sons, Inc.

Shmueli, G., Bruce, P.C., Stephens M.L., and Patel, N.R., (2017), *Data Mining For Business Intelligence: Concepts, Techniques, and Applications with JMP Pro*, Hoboken, New Jersey: John Wiley & Sons, Inc.

Westfall, P.H., Tobias, R.D., and Wolfinger, R.D. (2011), *Multiple Comparisons and Multiple Tests Using SAS*, Second Edition, SAS Institute.

Wright, S.P. and R.G. O’Brien (1988), “Power Analysis in an Enhanced GLM Procedure: What it Might Look Like,” *SUGI 1988, Proceedings of the Thirteenth Annual Conference*, 1097–1102, Cary NC: SAS Institute Inc.

Index

Specialized and Predictive Modeling