# Version 16

# Predictive and Specialized Modeling

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

Marcel Proust

**16.1**

# Get the Most from JMP

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

Visit JMP.com to find the following:

- live and recorded webcasts about how to get started with JMP
- video demos and webcasts of new features and advanced techniques
- details on registering for JMP training
- schedules for seminars being held in your area
- success stories showing how others use JMP
- the JMP user community, resources for users including examples of add-ins and scripts, a forum, blogs, conference information, and so on

**https://www.jmp.com/getstarted**

# Contents

## Predictive and Specialized Modeling

# 4    Partition Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
### Use Decision Trees to Explore and Model Your Data

# 5    Bootstrap Forest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
### Fit a Model By Averaging Many Trees

## 17 Gaussian Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 309

### Fit Data Using Smoothing Models

## 18 Time Series Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 323

### Fit Time Series Models and Transfer Functions

## 21  Modeling Utilities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 403
### Exploring Data for Outliers, Missing Values, and Patterns

## 22  Response Screening . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 447
### Test Many Responses in Large-Scale Data

# Learn about JMP

## Documentation and Additional Resources

Learn about JMP documentation, such as book conventions, descriptions of each JMP document, the Help system, and where to find additional support.

# Contents

# Formatting Conventions in JMP Documentation

These 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 (or sans-serif online) font.

- Code appears in `Lucida Sans Typewriter` (or monospace online) font.

- Code output appears in *Lucida Sans Typewriter* italic (or monospace italic online) font and is indented farther than the preceding code.

- **Helvetica bold** formatting (or bold sans-serif online) indicates items that you select to complete a task:
  - buttons
  - check boxes
  - commands
  - list names that are selectable
  - menus
  - options
  - tab names
  - text boxes

- The following items appear in italics:
  - words or phrases that are important or have definitions specific to JMP
  - book titles
  - variables

- Features that are for JMP Pro only are noted with the JMP Pro icon **JMP PRO** . For an overview of JMP Pro features, visit https://www.jmp.com/software/pro.

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

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

# JMP Help

JMP Help in the Help menu enables you to search for information about JMP features, statistical methods, and the JMP Scripting Language (or *JSL*). You can open JMP Help in several ways:

- Search and view JMP Help on Windows by selecting **Help > JMP Help**.
- On Windows, press the F1 key to open the Help system in the default browser.
- 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.

**Note:** The JMP Help is available for users with Internet connections. Users without an Internet connection can search all books in a PDF file by selecting **Help > JMP Documentation Library**. See "JMP Documentation Library" for more information.

# JMP Documentation Library

The Help system content is also available in one PDF file called *JMP Documentation Library*. Select **Help > JMP Documentation Library** to open the file. You can also download the Documentation PDF Files add-in if you prefer searching individual PDF files of each document in the JMP library. Download the available add-ins from https://community.jmp.com.

The following table describes the purpose and content of each document 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. Also learn how to share your results. |
| *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. |

| Document Title | Document Purpose | Document Content |
|---|---|---|
| *Basic Analysis* | Perform basic analysis using this document. | Describes these Analyze menu platforms:<br><br>• Distribution<br>• Fit Y by X<br>• Tabulate<br>• Text Explorer<br><br>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. |
| *Essential Graphing* | Find the ideal graph for your data. | Describes these Graph menu platforms:<br><br>• Graph Builder<br>• Scatterplot 3D<br>• Contour Plot<br>• Bubble Plot<br>• Parallel Plot<br>• Cell Plot<br>• Scatterplot Matrix<br>• Ternary Plot<br>• Treemap<br>• Chart<br>• Overlay Plot<br><br>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. |

| Document Title | Document Purpose | Document Content |
| --- | --- | --- |
| *Design of Experiments Guide* | Learn how to design experiments and determine appropriate sample sizes. | Covers all topics in the DOE menu. |
| *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:<br><br>• Standard Least Squares<br>• Stepwise<br>• Generalized Regression<br>• Mixed Model<br>• MANOVA<br>• Loglinear Variance<br>• Nominal Logistic<br>• Ordinal Logistic<br>• 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: |

Describes these Analyze > Predictive Modeling menu platforms:

- Neural
- Partition
- Bootstrap Forest
- Boosted Tree
- K Nearest Neighbors
- Naive Bayes
- Support Vector Machines
- Model Comparison
- Model Screening
- Make Validation Column
- Formula Depot

Describes these Analyze > Specialized Modeling menu platforms:

- Fit Curve
- Nonlinear
- Functional Data Explorer
- Gaussian Process
- Time Series
- Matched Pairs

Describes these Analyze > Screening menu platforms:

- Modeling Utilities
- Response Screening
- Process Screening
- Predictor Screening
- Association Analysis
- Process History Explorer

| Document Title | Document Purpose | Document Content |
| --- | --- | --- |
| *Multivariate Methods* | Read about techniques for analyzing several variables simultaneously. | Describes these Analyze > Multivariate Methods menu platforms:<br><br>• Multivariate<br>• Principal Components<br>• Discriminant<br>• Partial Least Squares<br>• Multiple Correspondence Analysis<br>• Structural Equation Models<br>• Factor Analysis<br>• Multidimensional Scaling<br>• Item Analysis<br><br>Describes these Analyze > Clustering menu platforms:<br><br>• Hierarchical Cluster<br>• K Means Cluster<br>• Normal Mixtures<br>• Latent Class Analysis<br>• Cluster Variables |
| *Quality and Process Methods* | Read about tools for evaluating and improving processes. | Describes these Analyze > Quality and Process menu platforms:<br><br>• Control Chart Builder and individual control charts<br>• Measurement Systems Analysis<br>• Variability / Attribute Gauge Charts<br>• Process Capability<br>• Model Driven Multivariate Control Chart<br>• Legacy Control Charts<br>• Pareto Plot<br>• Diagram<br>• Manage Spec Limits<br>• OC Curves |

| 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:<br>• Life Distribution<br>• Fit Life by X<br>• Cumulative Damage<br>• Recurrence Analysis<br>• Degradation<br>• Destructive Degradation<br>• Reliability Forecast<br>• Reliability Growth<br>• Reliability Block Diagram<br>• Repairable Systems Simulation<br>• Survival<br>• Fit Parametric Survival<br>• 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:<br>• Categorical<br>• Choice<br>• MaxDiff<br>• Uplift<br>• Multiple Factor 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. |
| *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. |

# Additional Resources for Learning JMP

In addition to reading JMP help, you can also learn about JMP using the following resources:

- "JMP Tutorials"
- "Sample Data Tables"
- "Learn about Statistical and JSL Terms"
- "Learn JMP Tips and Tricks"
- "JMP Tooltips"
- "JMP User Community"
- "Free Online Statistical Thinking Course"
- "JMP New User Welcome Kit"
- "Statistics Knowledge Portal"
- "JMP Training"
- "JMP Books by Users"
- "The JMP Starter Window"

## JMP 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, 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\16\Samples\Data

On macOS: \Library\Application Support\JMP\16\Samples\Data

In JMP Pro, sample data is installed in the JMPPRO (rather than JMP) 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 https://jmp.com/tools.

## Learn about Statistical and JSL Terms

For help with statistical terms, select Help > Statistics Index. For help with JSL scripting and examples, select **Help > Scripting Index**.

**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 and get help on the commands.

## 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.

## JMP Tooltips

JMP provides descriptive tooltips (or *hover labels*) when you hover 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 macOS.

# 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.

# Free Online Statistical Thinking Course

Learn practical statistical skills in this free online course on topics such as exploratory data analysis, quality methods, and correlation and regression. The course consists of short videos, demonstrations, exercises, and more. Visit https://www.jmp.com/statisticalthinking.

# JMP New User Welcome Kit

The JMP New User Welcome Kit is designed to help you quickly get comfortable with the basics of JMP. You'll complete its thirty short demo videos and activities, build your confidence in using the software, and connect with the largest online community of JMP users in the world. Visit https://www.jmp.com/welcome.

# Statistics Knowledge Portal

The Statistics Knowledge Portal combines concise statistical explanations with illuminating examples and graphics to help visitors establish a firm foundation upon which to build statistical skills. Visit https://www.jmp.com/skp.

# JMP Training

SAS offers training on a variety of topics led by a seasoned team of JMP experts. Public courses, live web courses, and on-site courses are available. You might also choose the online e-learning subscription to learn at your convenience. Visit https://www.jmp.com/training.

## JMP Books by Users

Additional books about using JMP that are written by JMP users are available on the JMP website. Visit https://www.jmp.com/books.

## 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 macOS) **> 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 macOS, select **JMP > Preferences > Initial JMP Starter Window**.

# JMP 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 https://www.jmp.com/support, including the technical support phone number.

# 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 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 "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 "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 "Bootstrap Forest".

- **JMP PRO** The Boosted Tree platform produces an additive decision tree model that consists 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 "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 "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 "Naive Bayes".

- **JMP PRO** The Support Vector Machines platform classifies observations into groups that are defined by levels of a categorical response variable. The model classifies data by optimizing a hyperplane that separates the classes. See "Support Vector Machines".

- **JMP PRO** The Model Screening platform enables you to quickly run multiple predictive models and compare the results. Measures of fit are provided for each model along with overlaid diagnostic plots. See "Model Screening".

- **JMP PRO** The Model Comparison platform enables you to compare the predictive ability of different models. Measures of fit are provided for each model along with overlaid diagnostic plots. See "Model Comparison".

- **JMP PRO** The Make Validation Column platform lets you partition the data into two or three sets, using one of five different methods to create these partitions. See "Make Validation Column".

- **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 "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 "Fit Curve".

- The Nonlinear platform lets you fit custom nonlinear models, which include a model formula and parameters to be estimated. See "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 "Gaussian Process".

- **JMP PRO** The Functional Data Explorer platform enables you to convert functional data into a form that can be analyzed in another JMP platform. See "Functional Data Explorer".

- The Time Series platform lets you explore, analyze, and forecast univariate time series. See "Time Series Analysis".

- The Time Series Forecast platform lets you model and forecast multiple time series. The best fitting model is automatically selected from a set of up to 30 exponential smoothing models. See "Time Series Forecast".

- The Matched Pairs platform compares the means between two or more correlated variables and assesses the differences. See "Matched Pairs Analysis".

- 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 "Modeling Utilities".

- 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 "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 "Process Screening".

- The Predictor Screening platform enables you to screen a data set for significant predictors. See "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 "Association Analysis".

- The Process History Explorer platform enables you to identify problem components in complex process histories. See "Process History Explorer".

# Neural Networks

## Fit Nonlinear Models Using Nodes and Layers

**JMP PRO** *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 3.1** Example of a Neural Network
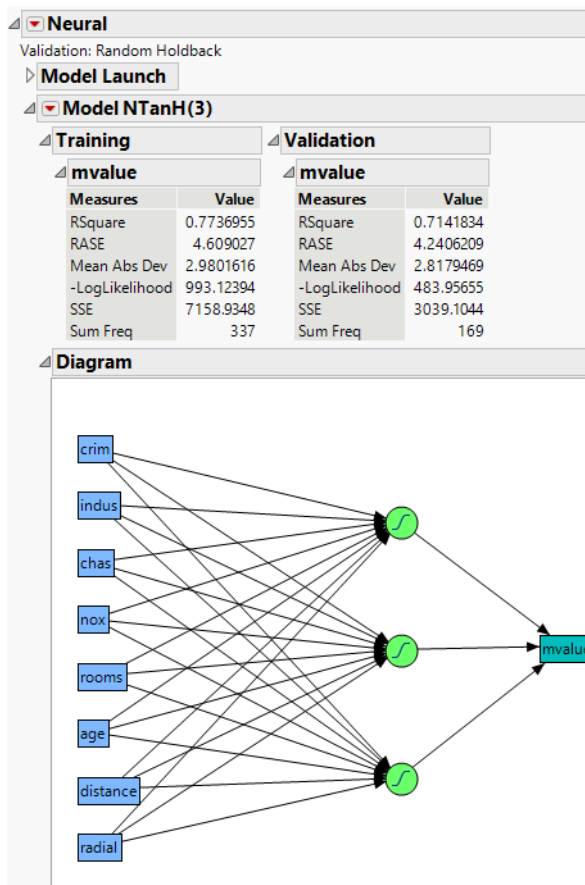
# Contents

# Overview of the Neural Platform

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, where each layer can contain as many hidden nodes as you want.

Figure 3.2 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 3.2**  Neural Network Diagram
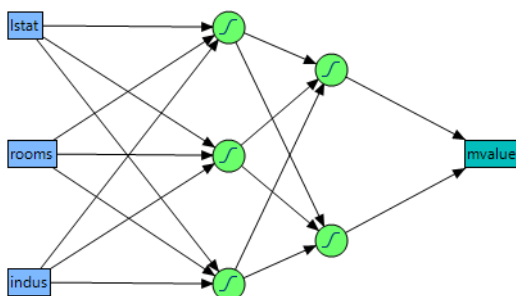


The functions applied at the nodes of the hidden layers are called activation functions. The activation function is a transformation of a linear combination of the X variables. For more information about the activation functions, see "Hidden Layer Structure". The function applied at the response is a 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. This is because 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.

## JMP PRO **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 3.3** The Neural Launch Window
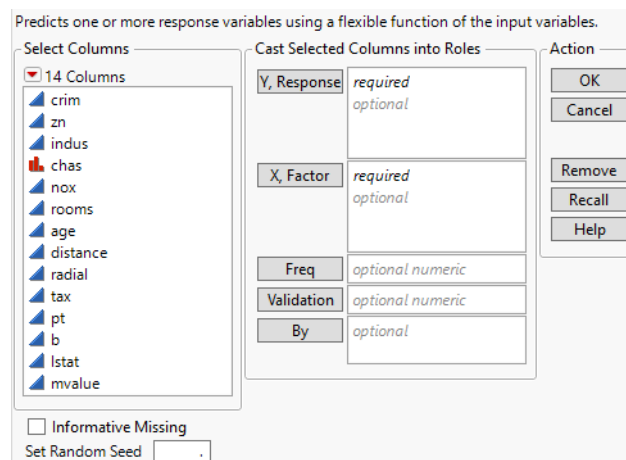


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**Y, Response**   The response variable or variables that you want to analyze. When multiple responses are specified, the models for the responses share all parameters in the hidden layers (those parameters not connected to the responses). The responses are conditionally independent given the predictor variables, but are marginally correlated through those variables to create one overall neural model.

**X, Factor**   The predictor variables.

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

JMP PRO **Validation**   A numeric column that defines the validation sets. See "Validation Methods for Neural". 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".

**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.

**JMP PRO** **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.

**Set Random Seed**   Sets the seed for the starting values used in the fitting procedure. Set Random Seed is useful if you want to reproduce an analysis. If you set a random seed and save the script, the seed is automatically saved in the script.

# 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 3.4**  The Model Launch Control Panel
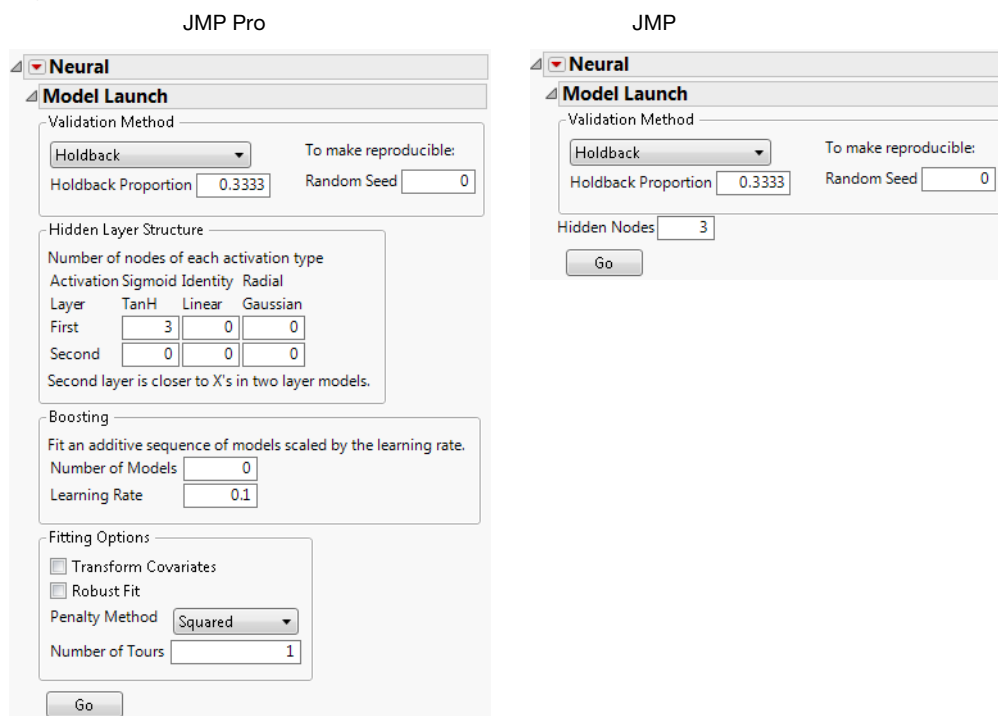
**Validation Method**   Select the method that you want to use for model validation. See
"Validation Methods for Neural".

   **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. See "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.

**JMP PRO** **Boosting**   Specify options for gradient boosting. See "Boosting".

**JMP PRO** **Fitting Options**   Specify options for variable transformation and model fitting. See
"Fitting Options".

**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 Methods for Neural

Neural utilizes a validation method in its fitting routine. Choices of validation method include
holdback, K-fold, or (in JMP Pro) the use of a validation column. To fit a model, the Neural
platform does the following:

• applies a penalty on the model parameters

• uses the validation set to tune the penalties on the parameters.

Select one of the following validation methods:

**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 *Using JMP*.

**Holdback**   Randomly divides the original data into the training and validation sets. You
specify the proportion of the original data to use as the validation set (holdback). The
random selection is based on stratified sampling across the model factors to attempt to
create training and validation sets that are more balanced than ones based on simple
random sampling.

**KFold**   Divides the data into K subsets, or folds. In turn, each of the K folds is used to validate the model fit on the rest of the data, fitting a total of K models. The final model is for the fold that resulted in the model with the best validation statistic.

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

**JMP PRO** **Validation Column**   Uses a numeric column that defines the validation sets. 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 the validation column has more than three unique values, then K-Fold validation is performed. The number of folds is determined by the number of values in the validation column.

The Neural platform uses the validation column to train and tune the model or to train, tune, and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

## 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.

**Caution:** You cannot use boosting with a two-layer neural network. If you specify any non-zero values in the second layer and also specify boosting, the second layer is ignored.

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

**JMP PRO** Boosting is the process of building a large additive neural network model by fitting a sequence of smaller models. Each of the smaller models is fit on the scaled residuals of the previous model. The models are combined to form the larger final model. The process uses validation to assess how many component models to fit, not exceeding the specified number of models.

Boosting is often faster than fitting a single large model. However, the base model should be a 1 to 2 node single-layer model. 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, where the response values are 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.

**Caution:** You cannot use boosting with a two-layer neural network. If you specify any non-zero values in the second layer and also specify boosting, the second layer is ignored.

## Fitting Options

**JMP® PRO** 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 "Neural Model Options".

**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".

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

## Penalty Method

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

**Table 3.1** 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 | $\sum \dfrac{\beta_i^2}{1 + \beta_i^2}$ | |
| 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. In addition, confusion statistics appear when the response is nominal or ordinal.

**Figure 3.5**  Example of a Neural Model Report



## Training and Validation Measures of Fit

Measures of fit appear for the training and validation sets (Figure 3.5).

**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**   (Appears only when the response is nominal or ordinal.) A measure of fit that compares the log-likelihoods from the fitted model and the constant probability model. Entropy RSquare ranges from 0 to 1, where values closer to 1 indicate a better fit. See "Entropy RSquare".

**RSquare**   Gives the RSquare for the model.

**RASE**   Gives the root average squared 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 *Fitting Linear Models*.

**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 (Figure 3.5). 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.

# Neural Platform Options

The Neural red triangle menu contains the following options:

See *Using JMP* 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.

# Neural Model Options

Each model report has a red triangle menu that contains options for producing additional output or saving results. The model report red triangle menu contains 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 more information about the options in the red triangle menu, see *Profilers*.

**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 more information about the options in the red triangle menu, see *Profilers*.

**Contour Profiler**    Launches the Contour Profiler. This is available only when the model contains more than one continuous factor. For more information about the options in the red triangle menu, see *Profilers*.

**Surface Profiler**    Launches the Surface Profiler. This is available only when the model contains more than one continuous factor. For more information about the options in the red triangle menu, see *Profilers*.

**ROC Curve**    Creates an ROC curve. Available only for nominal or ordinal responses. For more information about ROC Curves, see "ROC Curve".

**Lift Curve**    Creates a lift curve. Available only for nominal or ordinal responses. For more information about Lift Curves, see "Lift Curve".

**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 Interactive HTML 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 Interactive 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 "Formula Depot".

**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".

**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".

**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. Select **Help > Sample Data Library** and Boston Housing.jmp.
2. Launch the Neural platform by selecting **Analyze** > **Predictive Modeling** > **Neural**.
3. Assign mvalue to the **Y, Response** role.
4. Assign the other columns (crim through lstat) to the **X, Factor** role.
5. Click **OK**.
6. Enter 0.2 for the **Holdback Proportion**.
7. 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.

8. Enter 3 for the number of TanH nodes in the first layer.

9. Check the **Transform Covariates** option.

10. Click **Go**.

**Figure 3.6** Neural Report



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 0.913, signifying that the model is predicting well on data not used to train the model. As an additional assessment of model fit, click the red triangle next to Model NTanH(3) and select **Plot Actual by Predicted**.

**Figure 3.7** Actual by Predicted Plot
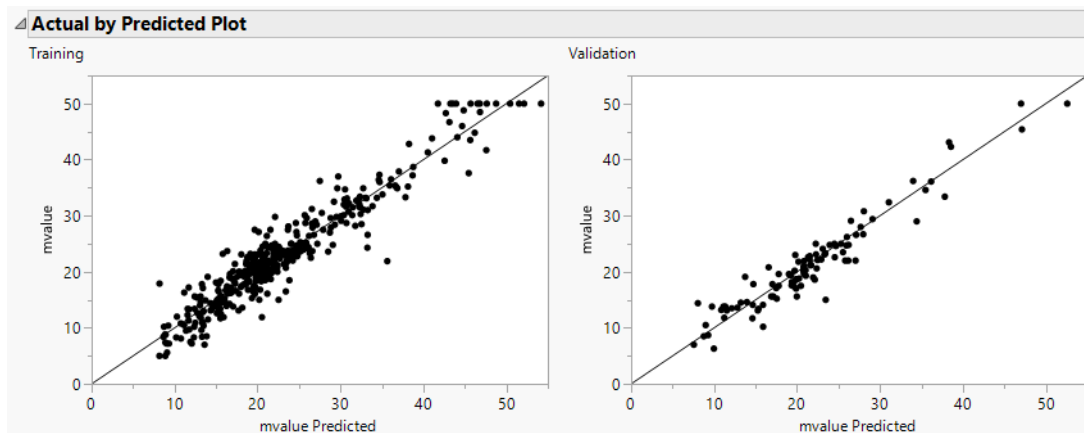


The points fall along the line, signifying that the predicted values are similar to the actual values.

To get a general understanding of how the X variables are impacting the predicted values, click the red triangle next to Model NTanH(3) and select **Profiler**.

**Figure 3.8** Profiler
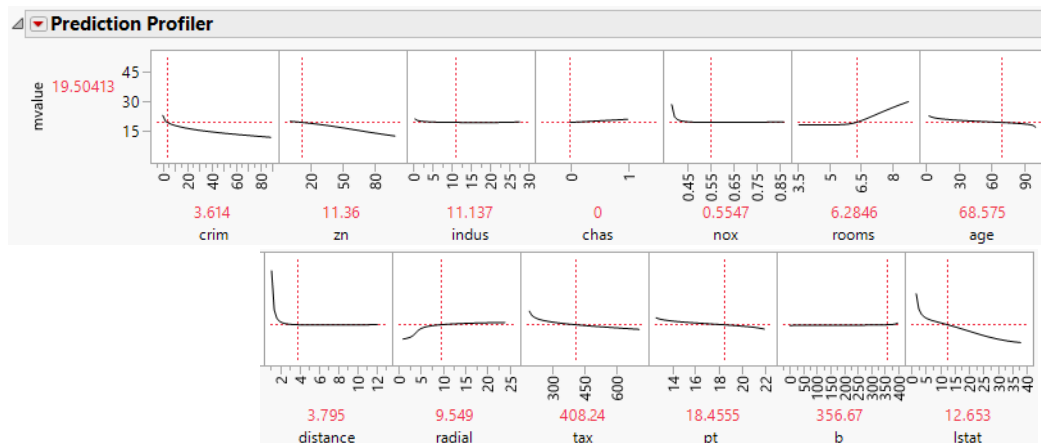


Some of the variables have profiles with positive slopes, and some negative. For example, the variable rooms has a positive slope. This indicates that the more rooms a home has, the higher the predicted median value. The variable pt is the pupil teacher ratio by town. This variable has a negative slope,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

# Chapter **4**

# 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 4.1** Example of a Decision Tree
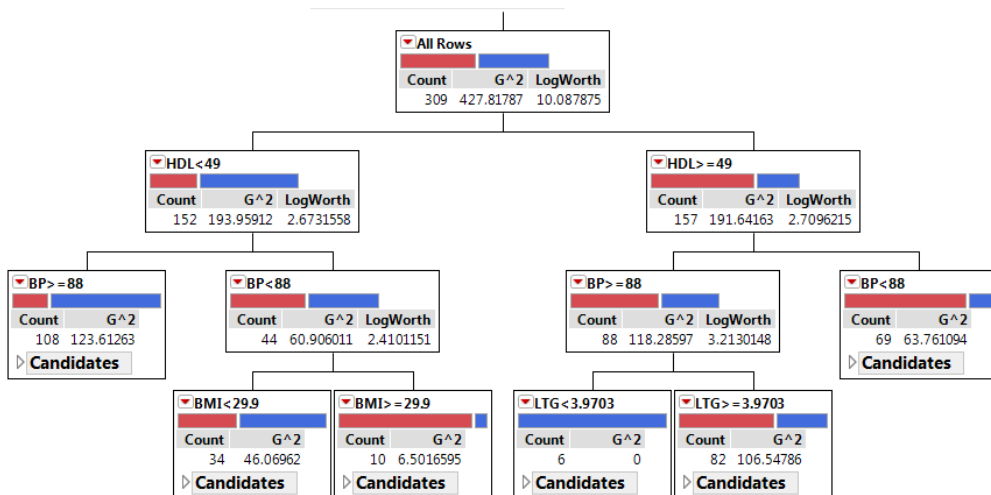
# Contents

# Overview of the Partition Platform

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 and the split is chosen to minimize the sum of squared errors. If the response is categorical, then the fitted value is a probability for the levels of the response and the split is chosen to minimize the residual log-likelihood chi-square.

For more information about split criteria, see "Statistical Details for the Partition Platform".

For more information about recursive partitioning, see Hawkins and Kass (1982) and Kass (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 4.2.

**Figure 4.2** Partition Report for Diabetes
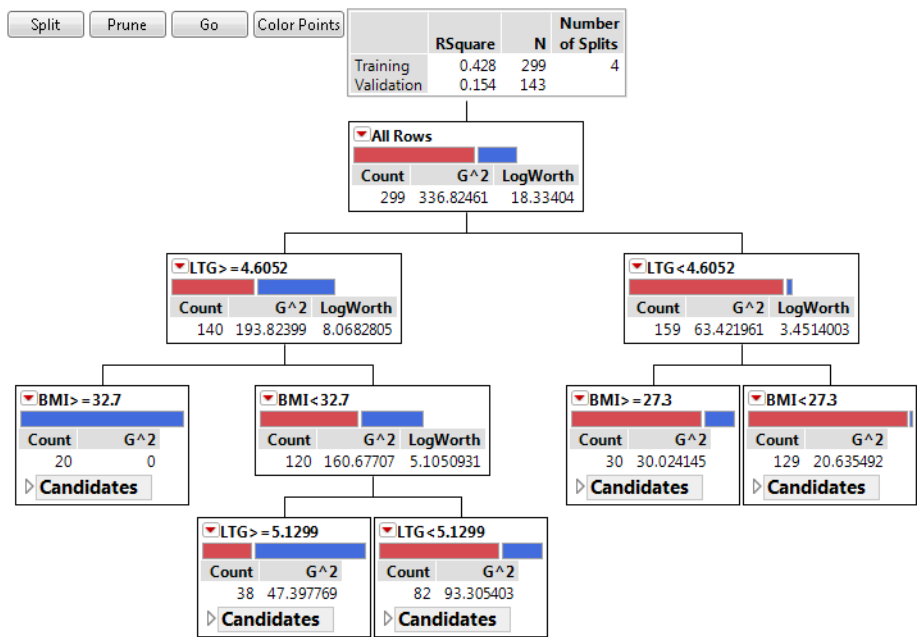


Automatic splitting resulted in four splits. The final RSquare for the Validation set is 0.154. The decision tree shows the four splits and the counts of observations in each split.

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

**Figure 4.3** Column Contributions Report



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 > Profiler.**

**Figure 4.4** Profiler for Partition Model



The profiler enables you to change the values of BMI and LTG to obtain the predicted Y Binary outcome. No other factors had a split in the partition model. Their profiles are flat lines.

10. 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 4.5**  Partition Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**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.

**JMP PRO** **Validation**    A numeric column that defines the validation sets. See "Validation in Partition". 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".

**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 information about these methods, see "Bootstrap Forest", "Boosted Tree", "K Nearest Neighbors", and "Naive Bayes".

**Validation Portion**   The portion of the data to be used as the validation set. See "Validation in Partition".

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

**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"
- "Report for Categorical Responses"
- "Report for 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 worst split.

**Go**   (Available when you are using validation.) Automatically adds splits to the decision tree until the validation statistic is optimized. See "Validation in Partition". 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 4.6** Partition Report for a Categorical Response
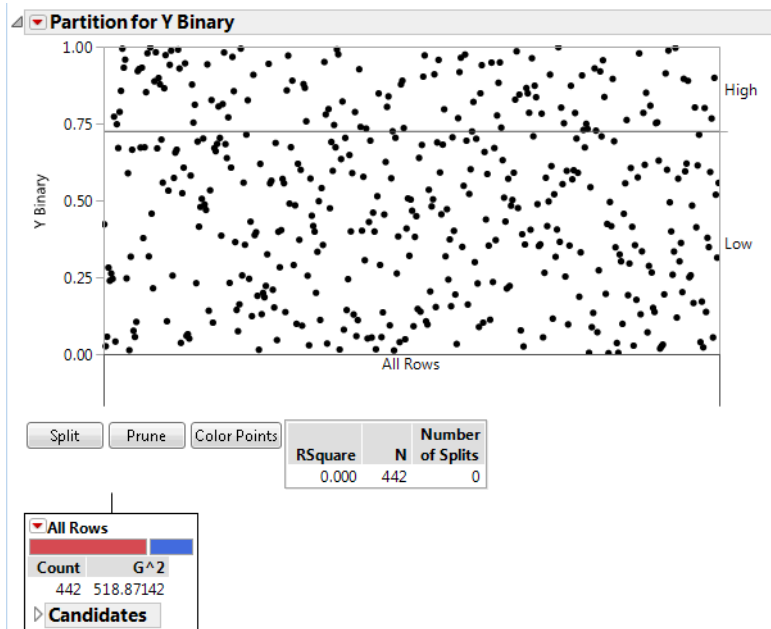


## Partition Plot

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 4.7** 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**   The number of observations.

**Number of Splits**   The 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 4.8** 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.

**G²**   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 for the Partition Platform".

**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 for the Partition Platform".

**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 for the Partition Platform".

# Report for Continuous Responses

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

**Figure 4.9**  Partition Report for a Continuous Response
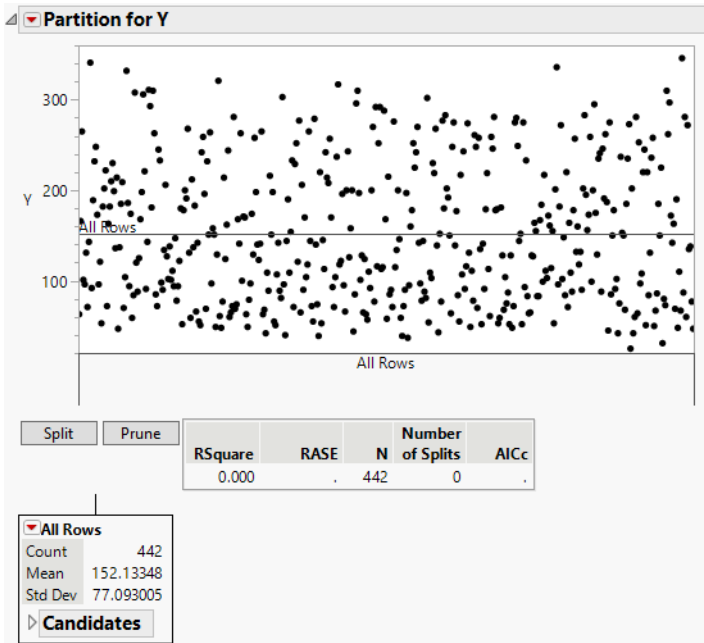


## Partition Plot

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, hover over the labels on the horizontal axis of the partition plot.

---

## Summary Report

**Figure 4.10**  Summary Report for a Continuous Response

| | RSquare | RASE | 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 $R^2$.

**RASE**    The root average square error. See *Fitting Linear Models*.

**N**    The number of observations.

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

**AICc**    The corrected Akaike's Information Criterion. See *Fitting Linear Models*.

## 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 4.11** Terminal Node Report for a Continuous Response
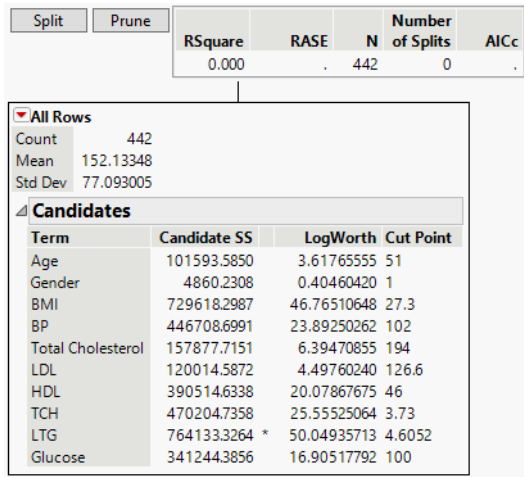


**Count**   The number of observations (rows) in the branch.

**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 $-\log_{10}(p\text{-value})$. The optimal split is the one that maximizes the LogWorth. See "Statistical Details for the Partition Platform".

> **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 for the Partition Platform".

# 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**     (Available only for categorical responses.) 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 for the Partition Platform".

   **Show Split Prob**     (Available only for categorical responses.) 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 for the Partition Platform".

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

   **Show Split Candidates**     Shows 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**   (Available only for continuous responses.) Shows a plot of actual values by predicted values. See "Actual by Predicted Plot".

**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, press Shift 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 cross validation RSquare is green.

**ROC Curve**   (Available only for categorical responses.) Receiver Operating Characteristic (ROC) curves display the efficiency of a model's fitted probabilities to sort the response levels. See "ROC Curve".

**Lift Curve**   (Available only for categorical responses.) Lift curves display the predictive ability of a partition model. See "Lift Curve".

**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".

**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 prediction formulas to a column or multiple columns in the data table. The formulas consist of nested conditional clauses that describe the tree structure. If the response is continuous, one column that contains a Predicting property is added. If the response is categorical, columns that contain a Response Probability property are added for each level of the response. In addition, a Most Likely column that contains the response level with the highest probability of occurrence for each observation is added.

**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.

**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 "Formula Depot".

**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 "Formula Depot". If you have checked Informative Missing, you can use this option by holding the Shift key as you click 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".

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

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

See *Using JMP* 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.

## Show Fit Details

**Figure 4.12** Fit Details for Categorical Response (Y Binary from Diabetes.jmp)

**Fit Details**

| Measure | Training | Validation | Definition |
|---|---|---|---|
| Entropy RSquare | 0.4276 | 0.1543 | $1-\text{Loglike(model)}/\text{Loglike(0)}$ |
| Generalized RSquare | 0.5656 | 0.2464 | $(1-(L(0)/L(\text{model}))^{(2/n)})/(1-L(0)^{(2/n)})$ |
| Mean -Log p | 0.3224 | 0.5312 | $\sum -\text{Log}(p[j])/n$ |
| RASE | 0.3201 | 0.4080 | $\sqrt{\sum (y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.2083 | 0.2917 | $\sum |y[j]-p[j]|/n$ |
| Misclassification Rate | 0.1371 | 0.2238 | $\sum (p[j]\neq pMax)/n$ |
| 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 |

| Actual | Predicted Rate | |
|---|---|---|
| Y Binary | Low | High |
| Low | 0.946 | 0.054 |
| High | 0.387 | 0.613 |

| Actual | Predicted Rate | |
|---|---|---|
| Y Binary | Low | High |
| Low | 0.866 | 0.134 |
| High | 0.413 | 0.587 |

**Entropy RSquare**   A measure of fit that compares the log-likelihoods from the fitted model and the constant probability model. Entropy RSquare ranges from 0 to 1, where values closer to 1 indicate a better fit. See "Entropy RSquare".

**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.

**RASE**    The root average 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 Report contains confusion matrices and confusion rates matrices. A confusion matrix is a two-way classification of actual and predicted responses. A confusion rates matrix is equal to the confusion matrix, with the numbers divided by the row totals.

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".

# 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 4.13** Specify Profit Matrix Window



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 more information about how values are calculated for the profit matrix, see *Using JMP*.

**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".

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 4.14**  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 | $(1-(L(0)/L(model))^{\wedge}(2/n))/(1-L(0)^{\wedge}(2/n))$ |
| Mean -Log p | 0.4134 | 0.6402 | $\sum -Log(p[j])/n$ |
| RASE | 0.3649 | 0.4694 | $\sqrt{\sum(y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.2854 | 0.3861 | $\sum |y[j]-p[j]|/n$ |
| Misclassification Rate | 0.1915 | 0.3810 | $\sum (p[j] \neq pMax)/n$ |
| N | 94 | 42 | n |

**Confusion Matrix**

Training

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

Validation

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

**Decision Matrix**

Training

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

Validation

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

Specified Profit Matrix

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

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

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

Misclassification Rate
0.2021

Misclassification Rate
0.2857

**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 the Informative Missing option is selected, categorical and continuous predictors are handled differently:

- 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 (Figure 4.15), 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 4.15** Impute Message in Summary Report

| RSquare | RASE | N | Number of Splits | AICc |
|---|---|---|---|---|
| 0.448 | 6.8246278 | 501 | 1 | 3352.2 |

## 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 (Figure 4.16).

**Figure 4.16** Actual by Predicted Plots for a Continuous Response
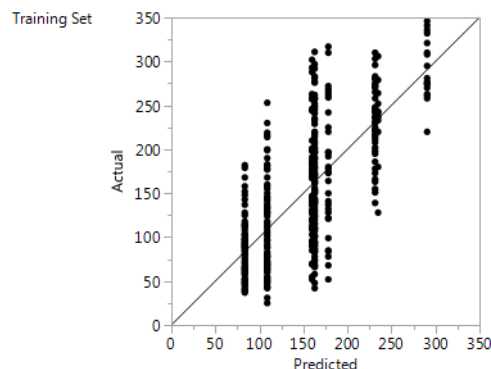
# ROC Curve

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 *Basic Analysis*.

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 4.17** ROC Curves for a Three Level Response
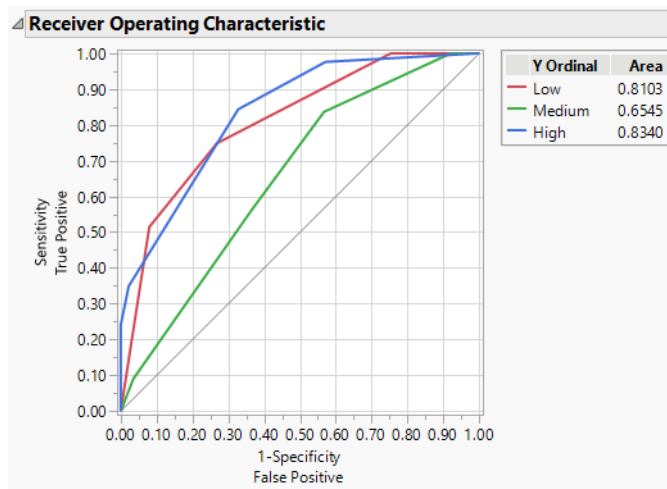
If the model perfectly rank-orders the response values, then the sorted data contains all of the positive values first, followed by all of the other values. In this situation, the curve moves all the way to the top before it moves at all to the right. If the model does not predict well, the curve follows 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. There is a point for each unique predicted probability value. Each predicted probability of a response level defines a portion of the observations that have a predicted probability greater than or equal to the unique predicted probability value. For a particular level of the response, the *lift* value is the ratio of the proportion of observed responses in that portion to the overall proportion of observed responses.

**Note:** For smaller models, it is possible that a large portion of the points have the same predicted probability value. If this probability is the highest predicted probability for the response level, the lift curve does not start at Portion = 0. For example, this is shown in the Low lift curve in Figure 4.18.

**Figure 4.18** Lift Curve
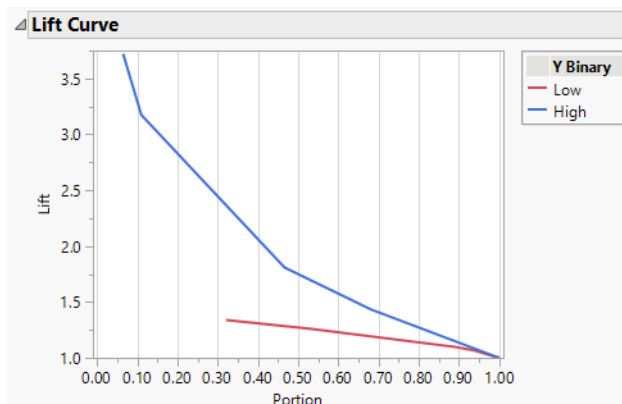
**Figure 4.19** Lift Table for Lift Curve

| Prob High | N >  Prob High | Portion | N High in Portion | Portion High | Lift = 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 4.19 provides a table of values to demonstrate the calculation of Lift and Portion used for the High lift curve shown in Figure 4.18. 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 observed High responses for an overall rate of 0.27.

- Prob High: The five unique predicted probability values from the partition model for the High response level.

- N > Prob High: The number of observations that have a predicted probability value equal to or greater than the value in Prob High.

- Portion: N > Prob High divided by 309, the total number of observations.

- N High in Portion: The number of observations in each portion that have an observed High response.

- Portion High: N High in Portion divided by N > Prob High.

- Lift: Portion High divided by 0.27, the overall rate of the observed High response.

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 in Partition

The use of validation with partition models is important given that partition models are easily overfit. 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. For more information about validation, see "Validation in JMP Modeling".

In Partition, 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 RSquare 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.

Select one of the following validation methods:

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

For more information about using row states and how to exclude rows, see *Using JMP*.

**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" for more information about the Validation Portion.

**JMP PRO** **Validation Column**   Uses a numeric column that defines the validation sets. This column should contain at most three distinct values:

– 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 the validation column has more than three levels, the rows that contain the smallest three values define the validation sets. All other rows are excluded from the analysis.

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".

**Tip:** To use K Fold or Nested K Fold crossvalidation, fit a partition model through the Model Screening platform. See "Model Screening".

# Additional Examples of Partitioning

- "Example of a Continuous Response"
- "Example of Informative Missing"
- "Example of Profit Matrix and Decision Matrix Report"

## 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 Portion**.

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

**Figure 4.20** Completed Launch Window with Validation Portion = 0.3
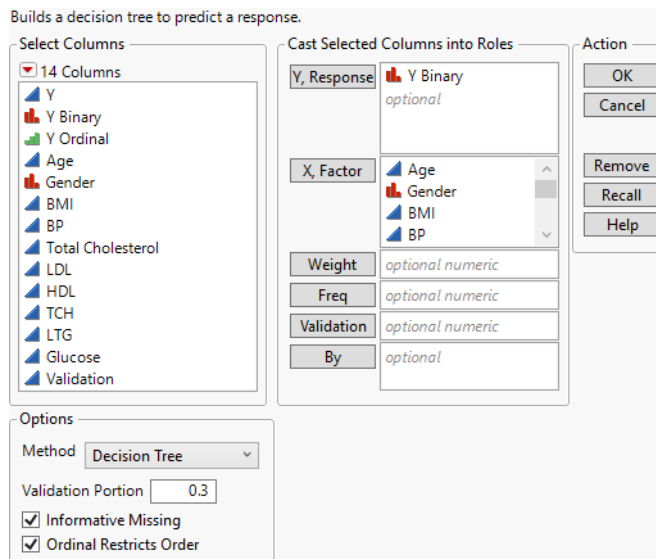


6. Click **OK**.

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

**Figure 4.21** Report after First Split with Decision Tree Hidden
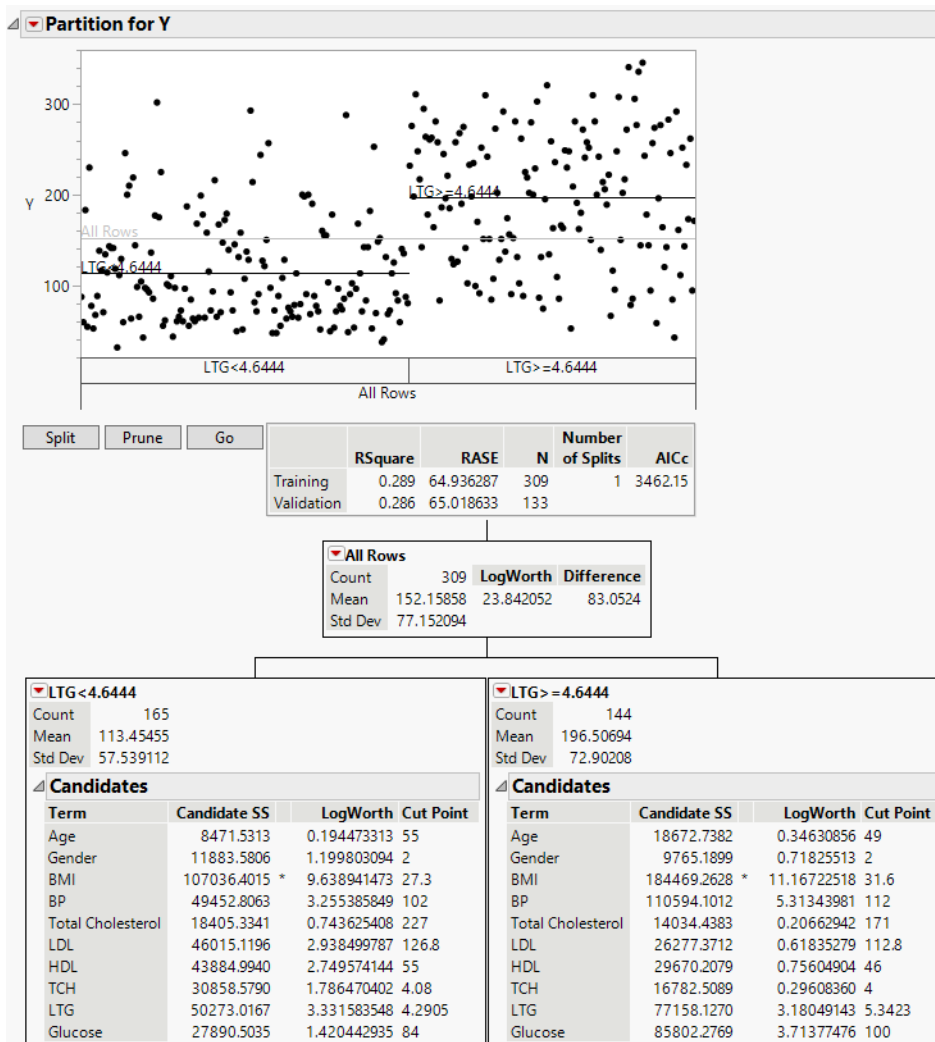


The original 309 values in the training data set are now split into two parts:

- 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 4.22** Report after Automatic Splitting with Validation
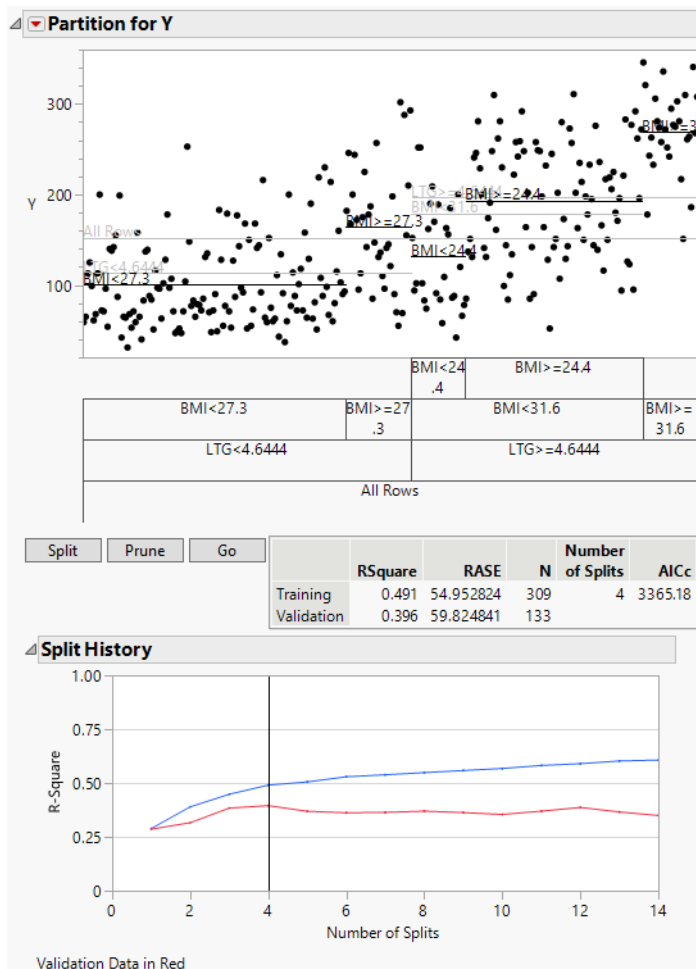


The solution found has four splits. The Split History plot shows that there is no further improvement in the validation data set after four splits. The RSquare value of 0.39 on the validation data does not support this model as a strong predictor of disease progression. The scatter across partitions 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. Press Shift 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 4.23**  ROC Curves for Models with (Left) and without (Right) Informative Missing
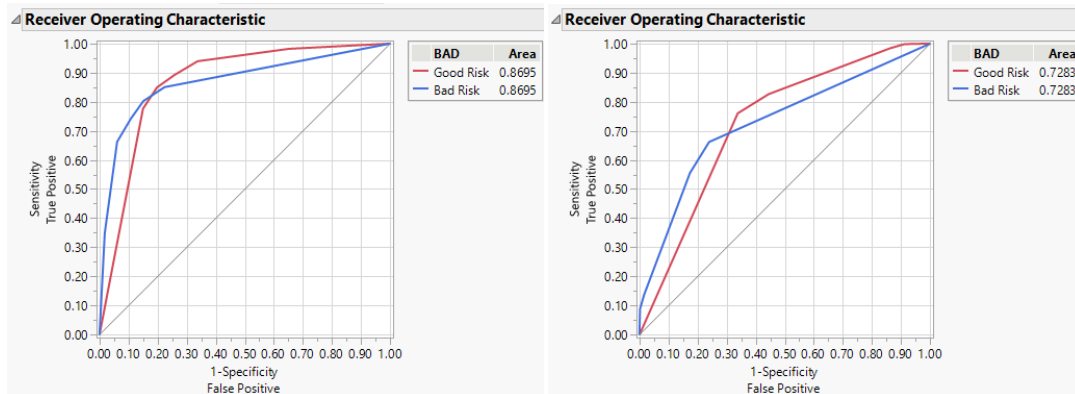


The area under the curve (AUC) for the model with informative missing (0.8695) is higher than the AUC for the model without informative missing (0.7283). Because there are only two levels for the response, the ROC curves for each model are reflections of one another and the AUCs are equal.

**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 contains the formulas from the model with informative missing. The second pair of formula columns contains the formulas from the model without informative missing.

3.  Click **OK**.

**Figure 4.24**  Measures of Fit from Model Comparison



| 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 | | 0.1232 | 0.1834 | 0.4381 | 0.3687 | 0.2737 | 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 this 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 4.25** Completed Launch Window with Validation Portion = 0.3



6. Click **OK**.

7. Press Shift 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 to the following values:

    – 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 4.26** Completed Profit Matrix



**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 4.27** 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 | $\sum -Log(p[j])/n$ |
| RASE | 0.3649 | 0.4694 | $\sqrt{\sum(y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.2854 | 0.3861 | $\sum |y[j]-p[j]|/n$ |
| Misclassification Rate | 0.1915 | 0.3810 | $\sum (p[j]\neq pMax)/n$ |
| N | 94 | 42 | n |

**Confusion Matrix**

Training

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

Validation

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

**Decision Matrix**

Training

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

Validation

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

Specified Profit Matrix

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

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

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

| Misclassification Rate |
|---|
| 0.2021 |

| Misclassification Rate |
|---|
| 0.2857 |

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 (Windows) or the Show Data Table icon on the tool bar menu (macOS).

– 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 for the Partition Platform

- "Responses and Factors"
- "Splitting Criterion"
- "Predicted Probabilities in Decision Tree and Bootstrap Forest"

## 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 $\Sigma -\log(p)$ for each observation, where *p* is the probability attributed to the response that occurred.

A candidate $G^2$ that has been chosen is:

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

Partition actually has two rates; one used for training that is the usual 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 defined as follows:

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

where the summation is across all response levels; $n_i$ is the number of observations at the node for the $i^{\text{th}}$ response level; and $\text{prior}_i$ is the prior probability for the $i^{\text{th}}$ response level, calculated as follows:

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

where $p_i$ is the $\text{prior}_i$ from the parent node, $P_i$ is the $\text{Prob}_i$ from the parent node, and $\lambda$ is a weighting factor currently set at 0.9.

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

# 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 5.1** Example of a Cumulative Validation Report

# Contents

# JMP PRO Overview of the Bootstrap Forest Platform

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 "Partition Models".

This is the fitting process for the training set:

1.  For each tree, select a bootstrap sample of observations.

2.  Fit the individual decision tree, using recursive partitioning.

    –   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).

# 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.

## 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. Enter 4 next to Number of Terms per Split.
8. Enter 30 next to Maximum Splits per Tree.
9. Select **Multiple Fits over Number of Terms** and enter 10 next to Max Number of Terms.
10. (Optional) Select **Suppress Multithreading** and enter 123 next to Random Seed.

    Because the bootstrap forest method involves random sampling, these actions ensure that your results are exactly the same as the results shown below.
11. Click **OK**.

**Figure 5.2** Overall Statistics Report

**Bootstrap Forest for BAD**

▷ **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 | $\sum$ -Log(p[j])/n |
| RASE | 0.2596 | 0.2886 | 0.2771 | $\sqrt{\sum(y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.1580 | 0.1779 | 0.1688 | $\sum$ |y[j]-p[j]|/n |
| Misclassification Rate | 0.0898 | 0.1133 | 0.0990 | $\sum$ (p[j]≠pMax)/n |
| N | 3576 | 1192 | 1192 | n |

◢ **Confusion Matrix**

| Training | | | Validation | | | Test | | |
|---|---|---|---|---|---|---|---|---|
| **Actual** | **Predicted Count** | | **Actual** | **Predicted Count** | | **Actual** | **Predicted Count** | |
| **BAD** | **Good Risk** | **Bad Risk** | **BAD** | **Good Risk** | **Bad Risk** | **BAD** | **Good Risk** | **Bad Risk** |
| Good Risk | 2819 | 86 | Good Risk | 884 | 33 | Good Risk | 921 | 28 |
| Bad Risk | 235 | 436 | Bad Risk | 102 | 173 | Bad Risk | 90 | 153 |

| **Actual** | **Predicted Rate** | | **Actual** | **Predicted Rate** | | **Actual** | **Predicted Rate** | |
|---|---|---|---|---|---|---|---|---|
| **BAD** | **Good Risk** | **Bad Risk** | **BAD** | **Good Risk** | **Bad Risk** | **BAD** | **Good Risk** | **Bad Risk** |
| Good Risk | 0.970 | 0.030 | Good Risk | 0.964 | 0.036 | Good Risk | 0.970 | 0.030 |
| Bad Risk | 0.350 | 0.650 | Bad Risk | 0.371 | 0.629 | Bad Risk | 0.370 | 0.630 |

Because the Multiple Fits over Number of Terms option was specified, models were created using 4, 5, 6, 8, and 10 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.

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

**Figure 5.3**  Column Contributions Report



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.
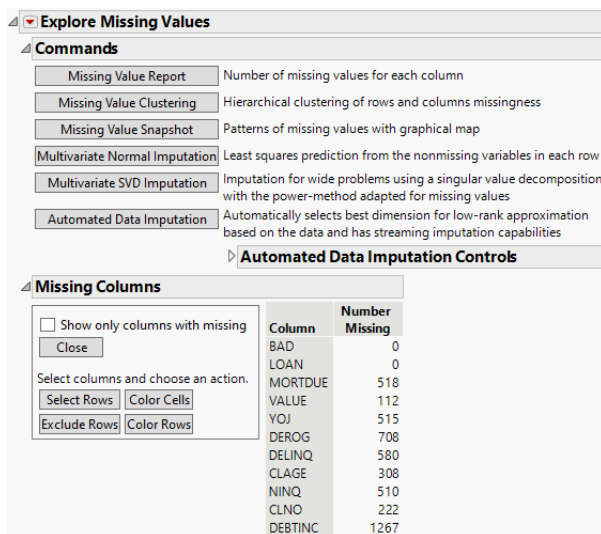
## Missing Values

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 5.4** Missing Values Report



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. See "Informative Missing".

# ![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 your results are exactly the same as the results shown below.

8. Click **OK**.

**Figure 5.5** Overall Statistics

| Bootstrap Forest for Percent body fat | | | | |
|---|---|---|---|---|
| **Specifications** | | | | |
| Target | Percent body fat | Training Rows: | 180 | |
| Validation Column: | Validation | Validation Rows: | 72 | |
| | | Test Rows: | 0 | |
| Number of Trees in the Forest: | 100 | Number of Terms: | 13 | |
| Number of Terms Sampled per Split: | 10 | Bootstrap Samples: | 180 | |
| | | Minimum Splits per Tree: | 10 | |
| | | Minimum Size Split: | 5 | |

| **Overall Statistics** | | |
|---|---|---|

| Individual Trees | RASE | |
|---|---|---|
| In Bag | 2.025251 | |
| Out of Bag | 5.995064 | |

| | RSquare | RASE | N |
|---|---|---|---|
| Training | 0.892 | 2.6941446 | 180 |
| Validation | 0.675 | 4.9577438 | 72 |

The Overall Statistics report shows that the Validation RSquare is 0.675.

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 5.6** Column Contributions

| Column Contributions | | | | |
|---|---|---|---|---|
| Term | Number of Splits | SS | | Portion |
| Abdomen circumference (cm) | 106 | 4349.56265 | | 0.6453 |
| Chest circumference (cm) | 35 | 700.64626 | | 0.1039 |
| Age (years) | 46 | 287.666673 | | 0.0427 |
| Knee circumference (cm) | 36 | 269.801386 | | 0.0400 |
| Height (inches) | 44 | 189.408593 | | 0.0281 |
| Wrist circumference (cm) | 48 | 171.226928 | | 0.0254 |
| Thigh circumference (cm) | 33 | 164.742015 | | 0.0244 |
| Weight (lbs) | 24 | 147.372437 | | 0.0219 |
| Hip circumference (cm) | 34 | 114.631003 | | 0.0170 |
| Ankle circumference (cm) | 29 | 101.095245 | | 0.0150 |
| Neck circumference (cm) | 32 | 94.1033567 | | 0.0140 |
| Biceps (extended) circumference (cm) | 27 | 83.2687719 | | 0.0124 |
| Forearm circumference (cm) | 28 | 66.9446242 | | 0.0099 |

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

# JMP PRO **Launch the Bootstrap Forest Platform**

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

# JMP PRO **Launch Window**

**Figure 5.7** Bootstrap Forest Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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 defines the validation sets. This column should contain at most three distinct values:.

– 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 the validation column has more than three levels, the rows that contain the smallest three values define the validation sets. All other rows are excluded from the analysis.

    The Bootstrap Forest platform uses the validation column to train and tune the model or to train, tune, and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

    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".

**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 information about these methods, see "Partition Models", "Boosted Tree", "K Nearest Neighbors", and "Naive Bayes".

**Validation Portion** The portion of the data to be used as the validation set.

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

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

# JMP PRO Specification Window

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

**Figure 5.8** Bootstrap Forest Specification Window



## JMP PRO 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.

## JMP PRO Forest Panel

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

**Number of Terms Sampled per Split**     The 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**     The proportion of observations to sample (with replacement) for growing each tree. A new random sample is generated for each tree.

**Minimum Splits Per Tree**     The minimum number of splits for each tree.

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

**Minimum Size Split**     The 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 the additional trees do 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.

## 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

## PRO 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, a window appears that informs you of approximately how long it will take JMP to fit the model. If the approximated time is too long, you have the option to view the current model by selecting **Accept Current Model**. The Bootstrap Forest report appears after the complete model is fit or after the current model is accepted.

**Figure 5.9** Accept Current Model Dialog



**Figure 5.10** Bootstrap Forest Report for a Categorical Response

**Bootstrap Forest for BAD**

▷ **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 | $\sum -Log(p[j])/n$ |
| RASE | 0.2596 | 0.2886 | 0.2771 | $\sqrt{\sum(y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.1580 | 0.1779 | 0.1688 | $\sum |y[j]-p[j]|/n$ |
| Misclassification Rate | 0.0898 | 0.1133 | 0.0990 | $\sum (p[j]\neq pMax)/n$ |
| N | 3576 | 1192 | 1192 | n |

**Confusion Matrix**

| Training Actual BAD | Predicted Count Good Risk | Bad Risk | | Validation Actual BAD | Predicted Count Good Risk | Bad Risk | | Test Actual BAD | Predicted Count Good Risk | Bad Risk |
|---|---|---|---|---|---|---|---|---|---|---|
| Good Risk | 2819 | 86 | | Good Risk | 884 | 33 | | Good Risk | 921 | 28 |
| Bad Risk | 235 | 436 | | Bad Risk | 102 | 173 | | Bad Risk | 90 | 153 |

| Actual BAD | Predicted Rate Good Risk | Bad Risk | | Actual BAD | Predicted Rate Good Risk | Bad Risk | | Actual BAD | Predicted Rate Good Risk | Bad Risk |
|---|---|---|---|---|---|---|---|---|---|---|
| Good Risk | 0.970 | 0.030 | | Good Risk | 0.964 | 0.036 | | Good Risk | 0.970 | 0.030 |
| Bad Risk | 0.350 | 0.650 | | Bad Risk | 0.371 | 0.629 | | Bad Risk | 0.370 | 0.630 |

**Figure 5.11** Bootstrap Forest Report for a Continuous Response
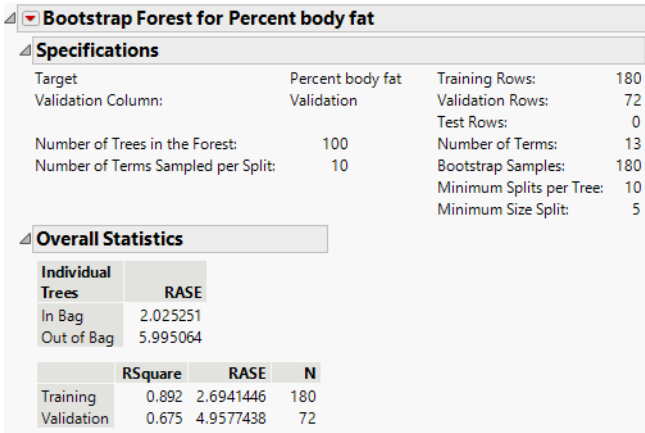


The following reports are provided, depending on whether the response is categorical or continuous:

- "Model Validation-Set Summaries"
- "Specifications"
- "Overall Statistics"
- "Cumulative Validation"
- "Per-Tree Summaries"

## 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 5.10 and "Multiple Fits Panel".

## 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.

### JMP PRO Categorical Response

#### Measures Report

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

**Note:** For Entropy RSquare and Generalized RSquare, values closer to 1 indicate a better fit. For Mean -Log p, RASE, Mean Abs Dev, and Misclassification Rate, smaller values indicate a better fit.

**Entropy RSquare**   A measure of fit that compares the log-likelihoods from the fitted model and the constant probability model. It ranges from 0 to 1. See "Entropy RSquare".

**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.

**RASE**   The root average squared prediction error. 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 Report

(Available only for categorical responses.) Shows classification statistics for the training set, and for the validation and test sets if they are specified. The Confusion Matrix Report contains confusion matrices and confusion rates matrices. A confusion matrix is a two-way classification of actual and predicted responses. A confusion rates matrix is equal to the confusion matrix, with the numbers divided by the row totals.

### 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".

## JMP PRO Continuous Response

### Individual Trees Report

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

### RSquare and RASE Report

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

## JMP PRO 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".

- RSquare (Entropy RSquare)
- Avg - Log p (Mean - Log p)
- Root Average Squared Prediction Error (RASE)

- 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**    The number of splits in the decision tree.

**Rank**    The 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.) The 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 red triangle menu contains 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 instances over all of the trees when the specified column is used to split the data.

- 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".

**ROC Curve**    (Available only for categorical responses.) See "ROC Curve".

**Lift Curve**    (Available only for categorical responses.) See "Lift Curve".

**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 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 "Formula Depot".

**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 "Formula Depot". If you have selected Informative Missing, you can use this option by holding the Shift key as you click 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".

**Profiler**   Shows a Prediction Profiler. See *Profilers*.

See *Using JMP* 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.

**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 predictions for that observation over all of the layers.

**Figure 6.1** Example of Boosted Tree Layers
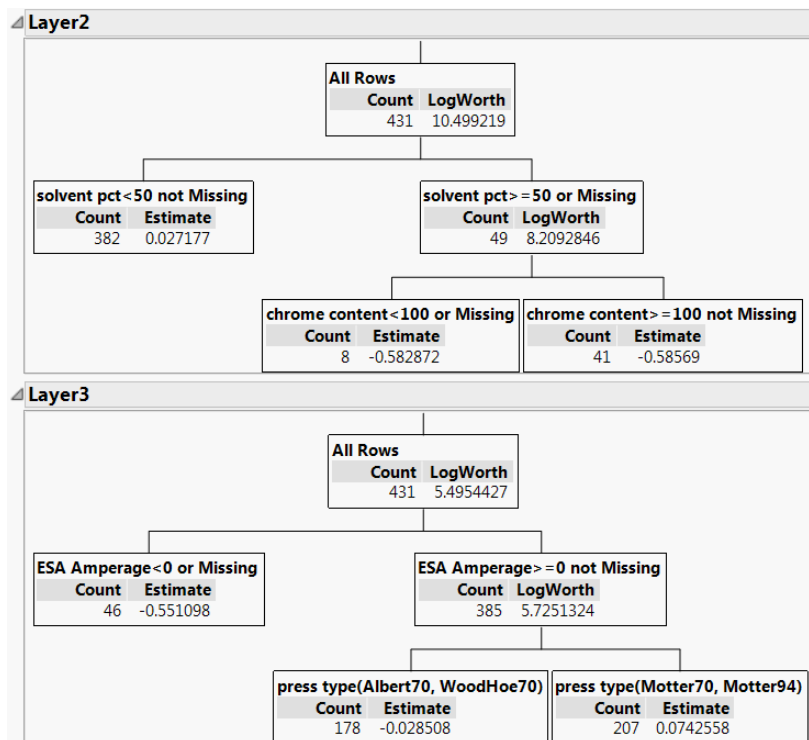
# Contents

# Overview of the Boosted Tree Platform

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 "Partition Models". 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.

This is the fitting process:

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 6.2** Overall Statistics for Nominal Response

◢ **Overall Statistics**

| Measure | Training | Validation | Definition |
|---|---|---|---|
| Entropy RSquare | 0.7103 | 0.3299 | 1-Loglike(model)/Loglike(0) |
| Generalized RSquare | 0.8340 | 0.4803 | $(1-(L(0)/L(model))^{\wedge}(2/n))/(1-L(0)^{\wedge}(2/n))$ |
| Mean -Log p | 0.1985 | 0.4383 | $\sum -Log(p[j])/n$ |
| RASE | 0.2196 | 0.3801 | $\sqrt{\sum(y[j]-p[j])^2/n}$ |
| Mean Abs Dev | 0.1650 | 0.2828 | $\sum |y[j]-p[j]|/n$ |
| Misclassification Rate | 0.0232 | 0.1852 | $\sum (p[j]\neq pMax)/n$ |
| N | 431 | 108 | n |

◢ **Confusion Matrix**

| | Training | | | | Validation | | |
|---|---|---|---|---|---|---|---|
| **Actual** | **Predicted Count** | | | **Actual** | **Predicted Count** | | |
| **Banding?** | band | noband | | **Banding?** | band | noband | |
| band | 179 | 9 | | band | 22 | 17 | |
| noband | 1 | 242 | | noband | 3 | 66 | |

| **Actual** | **Predicted Rate** | | | **Actual** | **Predicted Rate** | | |
|---|---|---|---|---|---|---|---|
| **Banding?** | band | noband | | **Banding?** | band | noband | |
| band | 0.952 | 0.048 | | band | 0.564 | 0.436 | |
| noband | 0.004 | 0.996 | | noband | 0.043 | 0.957 | |

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 0.1852, or about 19%.

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 6.3** Layer 1 of the Boosted Tree
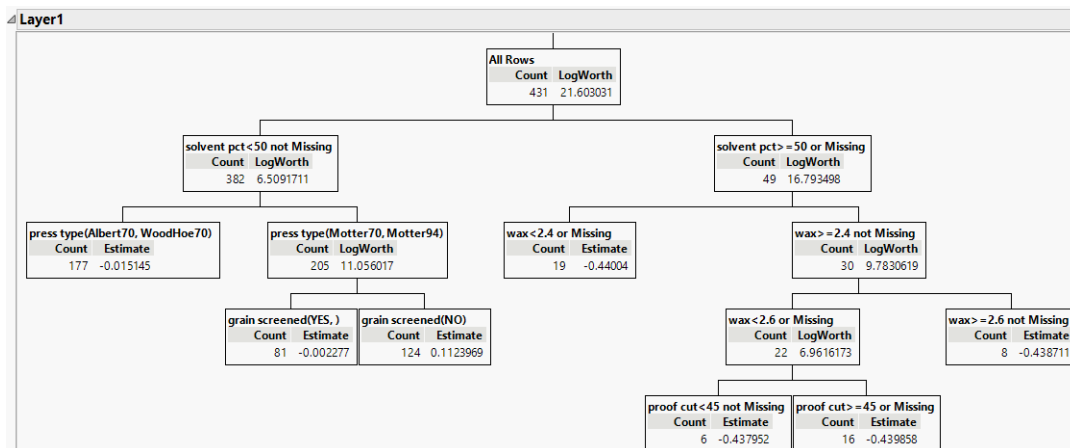


10. Click the red triangle next to Boosted Tree for Banding? and select **Save Columns > Save Prediction Formula**.

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 6.4** Overall Statistics for Continuous Response

| | RSquare | RASE | N |
|---|---|---|---|
| Training | 0.848 | 3.2012478 | 180 |
| Validation | 0.611 | 5.4301412 | 72 |

⊿ Overall Statistics

The Overall Statistics report provides the R-square and RASE for the boosted tree model. The R-square for the validation set is 0.611. The RASE for the validation set is about 5.43.

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 might not exactly match those in Figure 6.5.

**Figure 6.5** Summary Report for Variable Importance

⊿ Summary Report

| Column | Main Effect | Total Effect | .2 .4 .6 .8 |
|---|---|---|---|
| Abdomen circumference (cm) | 0.865 | 0.884 | |
| Age (years) | 0.065 | 0.084 | |
| Wrist circumference (cm) | 0.005 | 0.01 | |
| Hip circumference (cm) | 0.004 | 0.007 | |
| Biceps (extended) circumference (cm) | 0.003 | 0.005 | |
| Chest circumference (cm) | 0.002 | 0.004 | |
| Ankle circumference (cm) | 0.002 | 0.003 | |
| Height (inches) | 0.001 | 0.002 | |
| Weight (lbs) | 0.001 | 0.002 | |
| Thigh circumference (cm) | 0.001 | 0.002 | |
| Knee circumference (cm) | 0.001 | 0.001 | |
| Neck circumference (cm) | 9e-18 | 2e-17 | |
| Forearm circumference (cm) | 9e-18 | 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**.

**Figure 6.6** Boosted Tree Launch Window Using Body Fat.jmp



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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 defines the validation sets. This column should contain at most three distinct values:

- 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 the validation column has more than three levels, the rows that contain the smallest three values define the validation sets. All other rows are excluded from the analysis.

The Boosted Tree platform uses the validation column to train and tune the model or to train, tune, and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

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".

**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 information about these methods, see "Partition Models", "Bootstrap Forest", "K Nearest Neighbors", and "Naive Bayes".

**Validation Portion**   The portion of the data to be used as the validation set.

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

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

# JMP PRO Specification Window

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

**Figure 6.7** Boosted Tree Specification Window



## JMP PRO Boosting Panel

**Number of Layers**    The maximum number of layers to include in the final tree.

**Splits per Tree**    The 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".

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

## JMP PRO 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.

**JMP PRO** **Early Stopping**

**Early Stopping**    If selected, the boosting process stops fitting additional layers when the additional layers do 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.

# JMP PRO The Boosted Tree Report

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

**Figure 6.8**  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 6.9** Boosted Tree Report for a Categorical Response
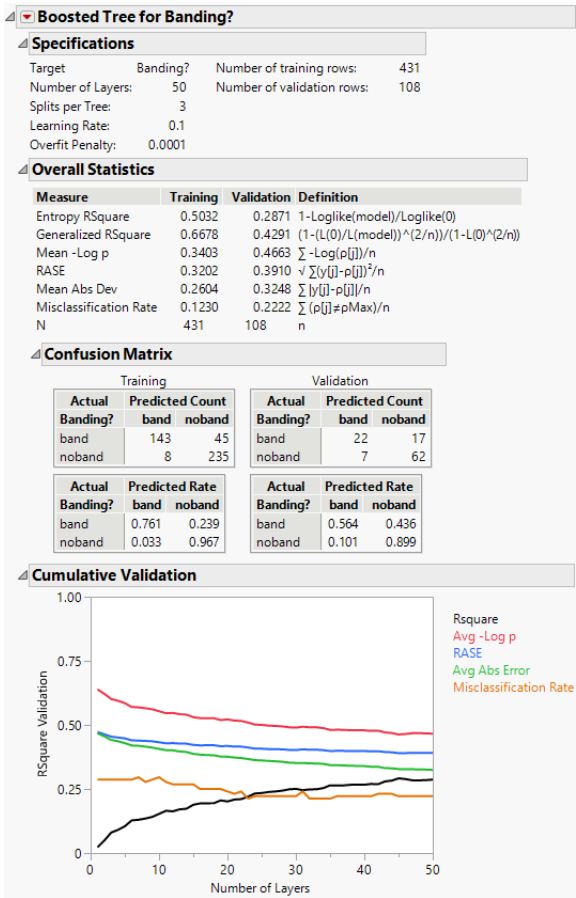


The following reports are provided, depending on whether the response is categorical or continuous:

- "Model Validation - Set Summaries"
- "Specifications"
- "Overall Statistics"
- "Cumulative Validation"

## JMP PRO 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 6.8 and "Multiple Fits Panel".

## 🅹🅼🅿 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 Splits and Learning Rate option in the Boosted Tree 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 they are specified.

---

**Note:** For Entropy R-Square and Generalized R-Square, values closer to 1 indicate a better fit. For Mean -Log p, RASE, Mean Abs Dev, and Misclassification Rate, smaller values indicate a better fit.

---

**Entropy RSquare**   A measure of fit that compares the log-likelihoods from the fitted model and the constant probability model. It ranges from 0 to 1. See "Entropy RSquare".

**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.

**RASE**   The root average squared prediction error. 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 Report

(Available only for categorical responses.) Shows classification statistics for the training set,
and for the validation and test sets if they are specified. The Confusion Matrix Report contains
confusion matrices and confusion rates matrices. A confusion matrix is a two-way
classification of actual and predicted responses. A confusion rates matrix is equal to the
confusion matrix, with the numbers divided by the row totals.

## 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".

# 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".

- R-Square (Entropy R-Square)
- Avg - Log p (Mean - Log p)
- Root Average Squared Prediction Error (RASE)
- 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.

# JMP PRO **Boosted Tree Platform Options**

The Boosted Tree red triangle menu contains 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 instances over all of the trees when the specified column is used to split the data.

– 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".

**Lift Curve**    (Available only for categorical responses.) See "Lift Curve".

**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 column property. If the response is categorical, the column contains a Response Probability column 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 column property. If the response is categorical, the column contains a Response Probability column property. If you have selected Informative Missing, you can save the Tolerant Prediction Formula by holding the Shift key as you click 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 "Formula Depot".

**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 "Formula Depot". If you have selected Informative Missing, you can use this option by holding the Shift key as you click 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".

**Profiler**   Shows a Prediction Profiler. See *Profilers*.

See *Using JMP* 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 statistical details about recursive decision trees, see "Statistical Details for the Partition Platform".

## 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.

*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 for classification of a categorical response as well as for prediction of a continuous response.

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, because the algorithm is sensitive to irrelevant predictors, the selection of predictors can impact your results.

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

**Figure 7.1** Example of the K Nearest Neighbors Platform
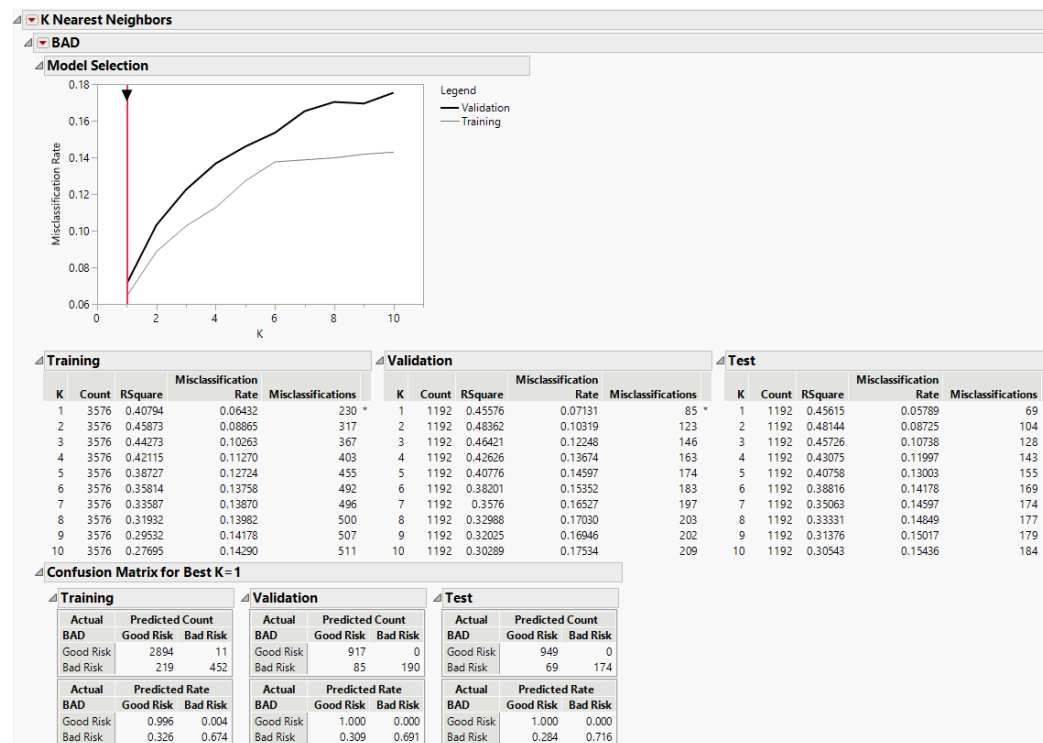
# Contents

# Overview of the K Nearest Neighbors Platform

The K Nearest Neighbors platform predicts a response value based on the responses of the *k* nearest neighbors. The *k* nearest neighbors to a given observation are determined by identifying the *k* smallest Euclidean distances between the predictor values for that observation and the predictor values for each of the other observations. The K Nearest Neighbors platform models both continuous and categorical responses.

A potential drawback of the *k* nearest neighbors method is that for large scale problems, the prediction formula is often complex and hard to interpret, limiting its usefulness. In addition, K Nearest Neighbors does not calculate 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).

## Continuous Responses

For a continuous response, the predicted value is the average of the responses for the *k* nearest neighbors. 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. See "Example of K Nearest Neighbors with Continuous Response".

## Categorical Responses

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. To obtain reproducible results, use the Set Random Seed option in the launch window or include the `Set Random Seed()` function in a JSL script.

In the categorical prediction models, 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.

# 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 are missing values for many 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 > K Nearest Neighbors**.
3. Select BAD and click **Y, Response**.
4. Select LOAN through CLNO and click **X, Factor**.

   Because one of the potential predictors, DEBTINC, has many missing values, you do not include it in your model. Missing values for continuous predictors are replaced by the average of the predictor. This procedure sometimes works well for values that are missing at random. Although the high missing rate of the DEBINC indicates that the missing might be informative, we do not investigate that in this example.

5. Select Validation and click **Validation**.
6. Click **OK**.

**Figure 7.2** K Nearest Neighbors Report



For each value of $K$, JMP constructs a model using only the training set observations. Each of these models is used to classify the validation set observations. The validation set results are used to select a best model. In this example, the 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 BAD red triangle and select **Publish Prediction Formula**.

8. Next to **Number of Neighbors, K**, leave the default value of 1.

9. Click **OK**.

The prediction equation is saved in the Formula Depot. You can compare the performance of alternative models published to the Formula Depot with that of the $K = 1$ nearest neighbor model using the Model Comparison option in the Formula Depot. See "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 Fat.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**.
7. Click the Percent body fat red triangle and select **Plot Actual by Predicted**.

**Figure 7.3** K Nearest Neighbors Report
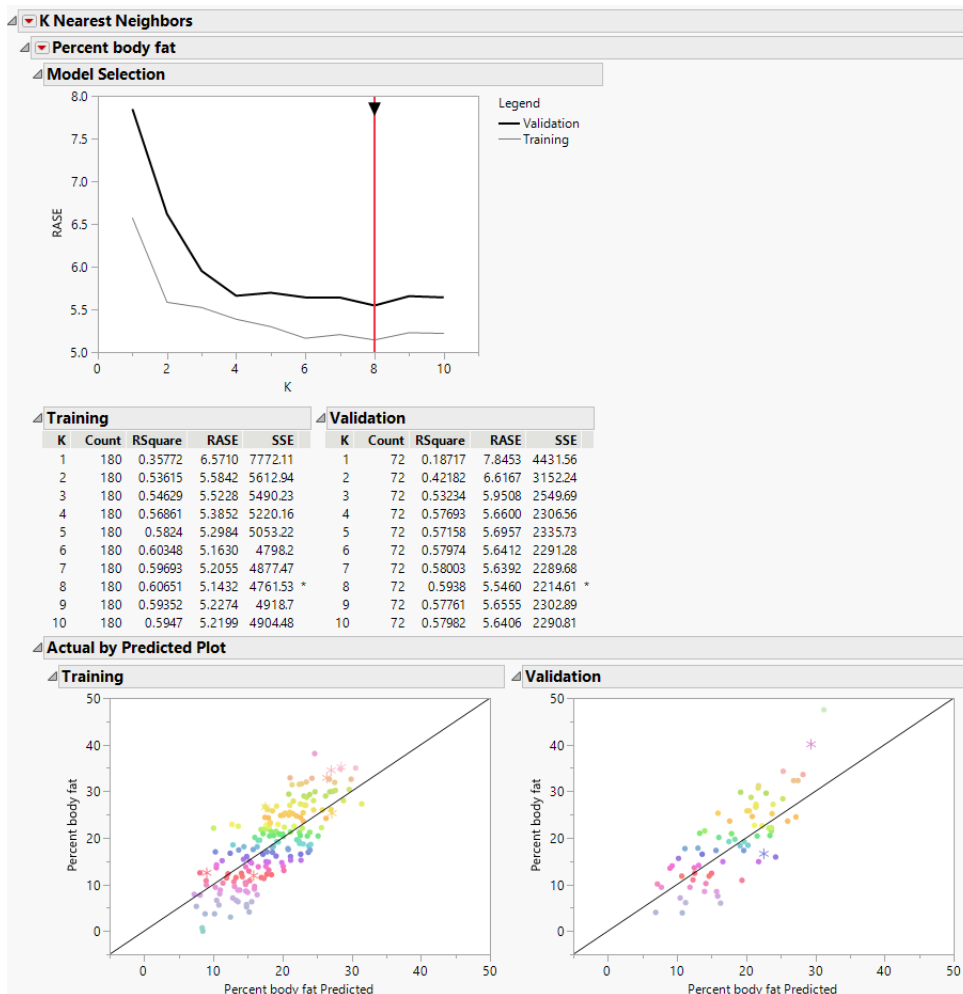


The *K* = 8 model had the lowest RASE for the validation set. Among *k* nearest neighbor models, the model based on 8 nearest neighbors seems to perform the best. The Actual by Predicted plot for the training set shows that the points fall along the line, signifying that the predicted values are similar to the actual values. Most of the points on the plot for the validation set fall along the line, with a few observations on the upper end that are a bit farther away.

# JMP PRO Launch the K Nearest Neighbors Platform

Launch the K Nearest Neighbors platform by selecting **Analyze > Predictive Modeling > K Nearest Neighbors**.

**Figure 7.4** K Nearest Neighbors Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The K Nearest Neighbors launch window provides the following options:

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

> **Note:** The K Nearest Neighbors platform can be used as a utility to determine the distances between neighboring observations, even without the presence of a response variable. If you do not specify a response variable, a blank report appears. However, the red triangle menu options Save Near Neighbor Rows and Save Near Neighbor Distances are available.

**X, Factor**   The predictor variables.

**Validation**   A numeric column that defines the validation sets. This column should contain at most three distinct values:

– 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 the validation column has more than three levels, the rows that contain the smallest three values define the validation sets. All other rows are excluded from the analysis.

The K Nearest Neighbors platform uses the validation column to train and tune the model or to train, tune, and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

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".

**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.

**Validation Portion**   The portion of the data to be used as the validation set.

**Number of Neighbors, K**   Maximum number of nearest neighbors to analyze. Models are fit for one nearest neighbor up to the value that you specify for *K*.

> **Note:** The maximum number of neighbors, *K*, must be no larger than one less than the number of rows in the training data table. If you specify a *K* that is larger than the maximum allowable *K*, a warning appears.

**Category Bias**   Specifies a tuning parameter that ensures that the fitted probabilities for categorical responses are always positive. By default, the Category Bias is 0.5.

**Set Random Seed**   Sets the seed for the randomization process used in tie-breaking for nominal and ordinal responses. If you specify a Validation Portion, this option also sets the seed for the rows used for validation. Set Random Seed is useful if you want to reproduce an analysis. If you set a random seed and save the script, the seed is automatically saved in the script.

# The K Nearest Neighbors Report

The K Nearest Neighbors report contains a separate report for each response variable. Each response variable report contains information about the fitted model for that response. This information includes a Model Selection report and summary information for each of the *k* models that were fit. The report shows tables for the training set and for the validation and test sets if you defined these using validation.

The Model Selection report displays a solution path plot across *K* based on the Misclassification Rate for categorical responses or the RASE for continuous responses. By default, the slider is placed on the value of *K* that corresponds to the best performing model. You can drag the slider to change the value of *K* in the report.

The statistics reported depend on the modeling type of the response. Each row in the summary tables 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 in the launch window.

# Continuous Responses

By default, in addition to the Model Selection graph, the report for a continuous response contains the Summary Table report.

### Summary Table

An asterisk marks the model for the value of *K* that has the smallest RASE. 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.

**RSquare**   The RSquare value for the model.

**RASE**   Root mean average squared prediction error for the model. The model with the smallest RASE is marked with an asterisk. If there are tied RASE values, the model with the smallest *K* is marked with the asterisk.

**SSE**   Sum of squared errors for the model.

## JMP PRO Categorical Responses

By default, in addition to the Model Selection graph, the report for a categorical response contains the Summary Table, Confusion Matrix, and Mosaic Plot reports.

### Summary Table

An asterisk marks the model for the value of *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.

**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**   Number of observations that are incorrectly predicted by the model.

### Confusion Matrix

By default, a Confusion Matrix Report is shown for the model with the smallest Misclassification Rate. If there are ties for the smallest misclassification rate, a report is shown for the model with the smallest *K*. The Confusion Matrix Report contains confusion matrices and confusion rates matrices. A confusion matrix is a two-way classification of actual and predicted responses. A confusion rates matrix is equal to the confusion matrix, with the numbers divided by the row totals. If you use validation, confusion matrices and confusion rates matrices for the validation and test sets appear. Use the Confusion Matrix Report and the misclassification rates to evaluate your model.

---

**Tip:** If you change the position of the slider in the solution path plot, an additional Confusion Matrix Report is displayed for the chosen value of *K*. Use the additional report to compare an alternative model to the default best model.

---

### Mosaic Plot

By default, a mosaic plot is shown for the model with the smallest Misclassification Rate. If there are ties for the smallest misclassification rate, a mosaic plot is shown for the model with the smallest *K*. A mosaic plot is a stacked bar chart where each segment is proportional to its group's frequency count. For more information about mosaic plots, see *Basic Analysis*. If you use validation, mosaic plots for the validation and test sets are shown.

**Tip:** If you change the position of the slider in the solution path plot, the mosaic plot updates to display the results for the chosen value of *K*.

# K Nearest Neighbors Platform Options

The K Nearest Neighbors red triangle menu contains the following options:

**Save Near Neighbor Rows**   Saves *K* columns to the data table. The columns are named RowNear <k>. For a given row, the $k^{\text{th}}$ column contains the row number of its $k^{\text{th}}$ 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 incorrect.

**Save Near Neighbor Distances**   Saves *K* columns to the data table. The columns are named Distance <k>. For a given row, the $k^{\text{th}}$ column contains the distance to that row's $k^{\text{th}}$ nearest neighbor.

See *Using JMP* 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.

## Model Fit Options

**Mosaic Plot**   (Available only for nominal or ordinal responses.) Shows or hides the mosaic plot. See "Mosaic Plot".

**Plot Actual by Predicted**   (Available only for continuous responses.) Plots the actual versus the predicted responses for the model with the smallest RASE. If there are ties for the smallest RASE, the plot is based on the model with the smallest *K*.

**Tip:** If you change the position of the slider on the solution path plot to a different *K*, the Actual by Predicted plot is updated to reflect the model for the chosen value of *K*.

**Plot Residual by Predicted**   (Available only for continuous responses.) Plots the residuals versus the predicted responses for the model with the smallest RASE. If there are ties for the smallest RASE, the plot is based on the model with the smallest *K*.

**Tip:** If you change the position of the slider on the solution path plot to a different *K*, the Residual by Predicted plot is updated to reflect the model for the chosen value of *K*.

**Save Predicteds**   Saves *K* predicted value columns to the data table. The columns are named Predicted <Response> <k>. The $k^{th}$ column contains predictions for the model based on the *k* nearest neighbors, where Response is the name of the response column.

**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. The values obtained from the Save Prediction Formula option use all of the rows in the training set, including the row for the predicted value. The values from the Save Predicteds option do not use the row for the predicted value in the training set, only all other rows.

**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 "Formula Depot".

## \text{JMP}_{\text{PRO}} **Naive Bayes**

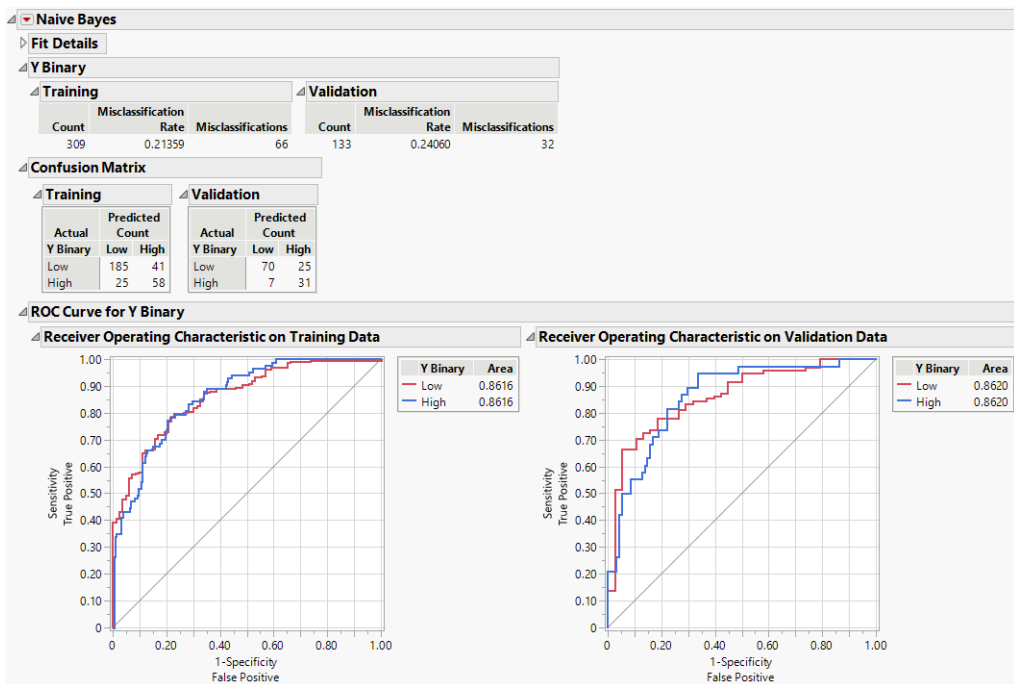### Classify Observations Using Probabilistic Assumptions

*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 8.1**  Example of Naive Bayes Analysis

# Contents

# JMP PRO **Overview of the Naive Bayes Platform**

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. (2009).

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 an observation in the Validation set is being classified and it has a categorical predictor value that was missing in the Training set, then the platform uses the non-missing features to predict. If an observation is missing all predictor values, the predicted response is the most frequent response. The prediction formulas handle missing values by having them contribute nothing to the observation's score.

For more information about the naive Bayes technique, see Hand et al. (2001), and Shmueli et al. (2010).

# Example of Naive Bayes

You want to construct a classification model to be used in predicting the disease progression for future patients as High or Low. 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.

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 8.2** Naive Bayes Report
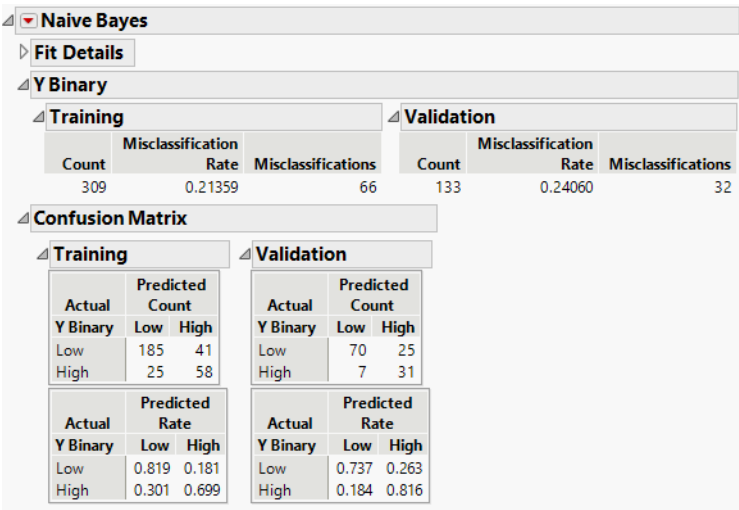


The Training Set has about a 21% misclassification rate and the Validation Set has about a 24% misclassification rate. The Confusion matrix suggests that, for both the Training and Validation sets, the larger source of misclassification comes from classifying patients with Low disease progression as 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 Naive Bayes red triangle and select **Profiler**.

**Figure 8.3**  Prediction Profiler for Disease Progression
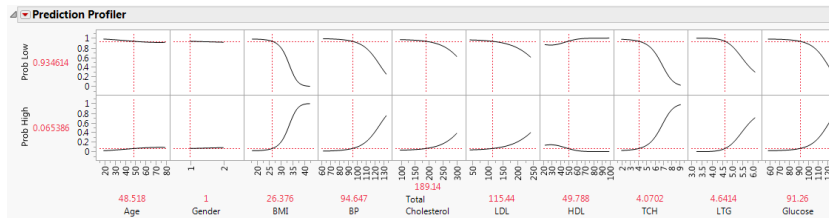


8.   Click the Prediction Profiler red triangle and select **Assess Variable Importance > Independent Uniform Inputs**.

**Figure 8.4**  Variable Importance



| Column | Main Effect | Total Effect | .2 .4 .6 .8 |
|---|---|---|---|
| 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 | |
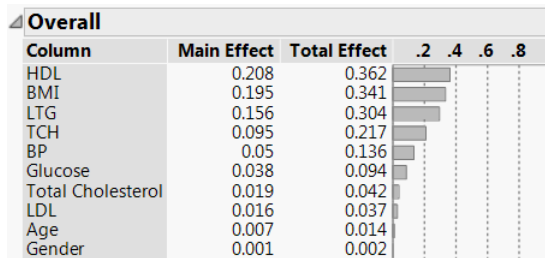
The Summary Report indicates that HDL, BMI, and LTG have the greatest impact on the estimated probabilities.

**Figure 8.5**  Marginal Model Plots Report
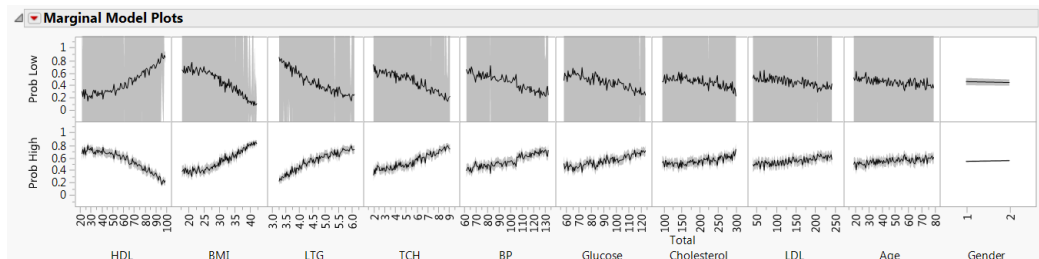


The second row of plots in the Marginal Model Plots report shows that higher values of HDL are associated with a lower probability of classifying a patient as High. Also, higher BMI and LTG values are associated with a higher probability of classifying a patient as High.

# Launch the Naive Bayes Platform

Launch the Naive Bayes platform by selecting **Analyze > Predictive Modeling > Naive Bayes**.

**Figure 8.6**  Naive Bayes Launch Window
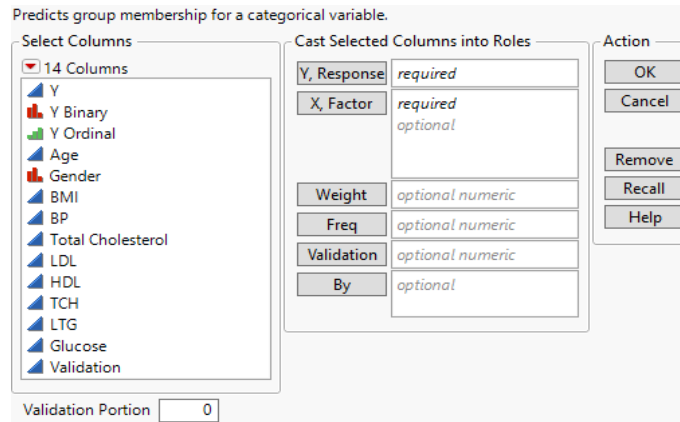


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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 defines the validation sets. This column should contain at most three distinct values:

– If the column has three unique values, then:

rows with the smallest value are used for the Training set.

rows with the middle value are used for the Validation set.

rows with the largest value are used for the Test set.

– If the column has two unique values, then only Training and Validation sets are used.

– If the validation column has more than three levels, the rows that contain the smallest three values define the validation sets. All other rows are excluded from the analysis.

The Naive Bayes platform uses the validation column to train and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

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".

**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.

---

**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. For more information about validation see "Validation in JMP Modeling".

---

## The Naive Bayes Report

After you click **OK** in the launch window, the Naive Bayes report appears. By default, the Naive Bayes report contains reports for fit details, the response column, the Confusion Matrix Report, and the ROC and Lift curves.

**Figure 8.7**  Naive Bayes Report

# Fit Details Report

The Fit Details report shows various measures of fit for the model for the Training set, and for the Validation and Test sets if they are specified. The Measure column lists the different fit statistics and the Definition column shows the formulas for the corresponding fit statistics. See "Measures of Fit for Categorical Responses". By default, the Fit Details report in the Naive Bayes report window is closed.

# 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**   The 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**   The number of observations in the corresponding set that are classified incorrectly.

# Confusion Matrix Report

The Confusion Matrix Report shows a confusion matrix and a confusion rates 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. A confusion rates matrix is equal to the confusion matrix, with the numbers divided by the row totals.

# ROC Curves

The report displays the Receiver Operating Characteristic (ROC) curve for the Training set, and for the Validation and Test sets if they are specified. The ROC curve measures the ability of the fitted probabilities to classify response levels correctly. The further the curve from the diagonal, the better the fit. An introduction to ROC curves is found in *Basic Analysis*.

See "ROC Curve" for more information about ROC curves.

# **Naive Bayes Platform Options**

The Naive Bayes red triangle menu contains the following options:

**ROC Curve**    Shows or hides the ROC plot. The ROC plot is shown by default. See "ROC Curves".

**Lift Curve**    Shows or hides the lift curve for the model. If you used validation, Lift curve is shown for each of the Training, Validation, and Test sets, if specified.

A lift curve shows how effectively response levels are classified as their fitted probabilities decrease. The fitted probabilities are plotted along the horizontal axis in descending order. The vertical coordinate for a fitted probability is the proportion of correct classifications for that probability or higher, divided by the overall correct classification rate. Use the lift curve to see whether you can correctly classify a large proportion of observations if you select only those with a fitted probability that exceeds a threshold value.

If the response has two levels, the Lift curve plot displays a lift curve for the first level of the response only. If the response has more than two levels, the Lift curve plot displays a sub-outline of the curves for each response level. See "Lift Curve" for more information about lift curves.

**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".

**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".

**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 "Formula Depot".

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

See *Using JMP* 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 8.8**  Naive Bayes Report for BAD

| BAD | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Training** | | | | **Validation** | | | | **Test** | |
| Count | Misclassification Rate | Misclassifications | | Count | Misclassification Rate | Misclassifications | | Count | Misclassification Rate | Misclassifications |
| 3576 | 0.18009 | 644 | | 1192 | 0.19128 | 228 | | 1192 | 0.18960 | 226 |

| Confusion Matrix | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Training** | | | **Validation** | | | **Test** | | |
| Actual BAD | Predicted Count Good Risk | Bad Risk | Actual BAD | Predicted Count Good Risk | Bad Risk | Actual BAD | Predicted Count Good Risk | Bad Risk |
| Good Risk | 2691 | 214 | Good Risk | 859 | 58 | Good Risk | 885 | 64 |
| Bad Risk | 430 | 241 | Bad Risk | 170 | 105 | Bad Risk | 162 | 81 |
| Actual BAD | Predicted Rate Good Risk | Bad Risk | Actual BAD | Predicted Rate Good Risk | Bad Risk | Actual BAD | Predicted Rate Good Risk | Bad Risk |
| Good Risk | 0.926 | 0.074 | Good Risk | 0.937 | 0.063 | Good Risk | 0.933 | 0.067 |
| Bad Risk | 0.641 | 0.359 | Bad Risk | 0.618 | 0.382 | Bad Risk | 0.667 | 0.333 |

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 Naive Bayes red triangle and select **Save Probability Formulas**.

    Three sets of columns are added to the data table.

    –   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 more information about the formula columns, see "Saved Probability Formulas".

# Statistical Details for the Naive Bayes Platform

- "Algorithm"
- "Saved Probability Formulas"

## 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 $C_1$, …, $C_K$. Denote the features, or predictors, by $X_1$, $X_2$, …, $X_p$.

The conditional probability that an observation with predictor values $x_1$, $x_2$, …, $x_p$ belongs in the class $C_k$ is computed as follows:

$$P(C_k|(x_1, \ldots, x_p)) = \frac{\left( P(C_k) \prod_{j=1}^{p} [P(x_j|C_k)] \right) / (\mathbf{R})}{\sum_{k=1}^{K} \left( P(C_k) \left( \prod_{j=1}^{p} [P(x_j|C_k)] \right) / \mathbf{R} \right)} ,$$

where $\mathbf{R}$ is a regularization constant. In the formula above, the conditional probability that an observation with $X_j = x_j$ belongs to the class $C_k$, $P(x_j|C_k)$, is determined as follows:

- If $X_j$ is categorical:

$$P(x_j|C_k) = \frac{\# \text{ of observations in } x_j|C_k}{\# \text{ of observations in } C_k}$$

- If $X_j$ is continuous:

$$P(x_j|C_k) = \frac{1}{s_{jk}} \phi((x_j - m_{jk})/s_{jk})$$

Here, $\phi$ is the standard normal density function, and $m$ and $s$ are the mean and standard deviation, respectively, of the predictor values within the class $C_k$.

The unconditional probability that an observation belongs in class $C_k$, $P(C_k)$, is computed as follows:

$$P(C_k) = \frac{\# \text{ of observations in } C_k + (0.5/K)}{\text{Total \# of observations} + 0.5}$$

An observation is classified into the class for which its conditional probability is the largest.

---

**Note:** In the formula for $P(C_k)$, 0.5 is the prior bias factor. This value is the default value. To change the prior default factor, go to File > Preferences > Platforms > Naive Bayes, select the Prior Bias check box and change the value.

## Saved Probability Formulas

This section describes the formulas saved using the Save Probability Formula option. The conditional probability that an observation with predictor values $x_1$, $x_2$, ..., $x_p$ belongs in the class $C_k$ differs slightly from that given by $P(C_k|(x_1,..., x_p))$, shown in the section "Algorithm". This is done for computational efficiency.

### Naive Score Formulas

The Naive Score formula for a given class $C_k$, $S(C_k)$, is a variation of the numerator in the expression for $P(C_k|(x_1,..., x_p))$ and is computed as follows:

$$S(C_k) = \exp[\ln\{P(C_k)\} + Continuous + Categorical + \ln(\mathbf{R})]$$

The Naive Score formula is a combination of scores from continuous and categorical predictors. Recall that $\mathbf{R}$ is a regularization constant. The continuous portion of the formula is computed as follows:

$$Continuous = \sum_{j=1}^{p_1} \left\{ \text{Normal Log Density}\left(\frac{x_j - m_{jk}}{s_{jk}}\right) + \ln(1/s_{jk}) \right\}$$

where

$j = 1,..., p_1$ continuous predictors.

The categorical portion of the formula is computed as follows:

$$Categorical = \sum_{r=1}^{p_2} \left\{ \sum_{l=1}^{L_r} 1_{rl}\ln(P(x_{rl}|C_k)) \right\}$$

where

$r = 1,...,p_2$ categorical variables

$l = 1,..., L_r$ levels of the $r^{th}$ categorical variable

$1_{rl}$ is an indicator variable that equals 1 when $x_{rl}$ is the $l^{th}$ level of the $r^{th}$ categorical predictor and 0 otherwise

## Naive Score Sum Formulas

The Naive Score Sum formula, $S$, sums the Naive Score formulas over all classes. This is a variation of the denominator in the expression for $P(C_k|(x_1,..., x_p))$.

$$S = \sum_{k=1}^{K} S(C_k)$$

## Naive Prob Formulas

The Naive Prob formula for a given class $C_k$ equals $P(C_k|(x_1,..., x_p))$. In the JMP formulas,

$$P(C_k|(x_1, ..., x_p)) = \frac{S(C_k)}{S}$$

## Naive Predicted Formula

The Naive Predicted Formula for an observation classifies that observation into the class for which $P(C_k|(x_1,..., x_p))$ is the largest. This is equivalent to classifying an observation into the class for which its Naive Score formula is the largest.

<div align="right">

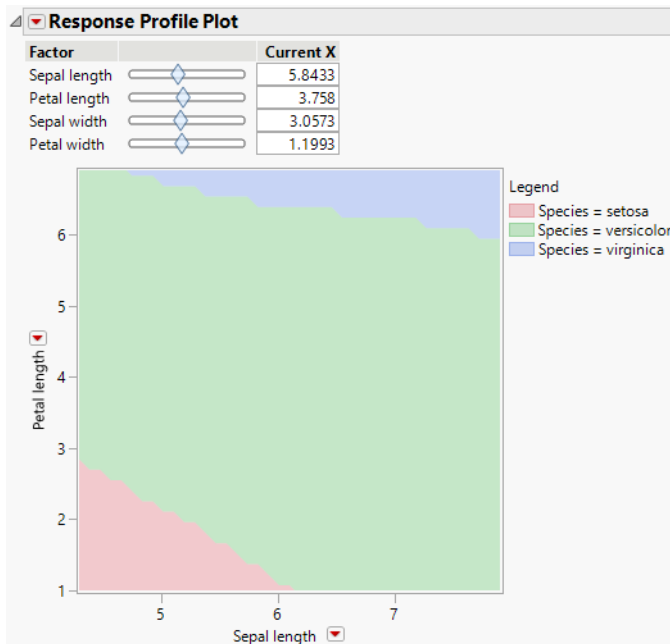**ᴶᴹᴾ Support Vector Machines**
**ᴾᴿᴼ**

**Predict Response Categories Using Support Vectors**

</div>

---

*The Support Vector Machines platform is available only in JMP Pro.*

In JMP Pro, the Support Vector Machines (SVM) platform provides a machine learning tool for both nonlinear regression and classification. The SVM algorithm is a supervised learning algorithm that uses the training data to build a model to predict or classify new observations.

If you have a categorical response, you can visualize the model classification in the Response Profile Plot. The SVM platform also provides misclassification rates, confusion matrices, and ROC and Lift curves to help you determine how well your model fits the data. If you have a continuous response, fit statistics and actual by predicted plots are used to determine how well the model fits the data.

**Figure 9.1** Response Profile Plot

## Contents

# JMP PRO **Overview of Support Vector Machines**

A support vector machine (SVM) model is a supervised learning algorithm that is used to predict or classify new observations. A model is fit on a set of training data where the responses are known. Then, the model is used to predict the responses of new observations.

When the response is categorical, SVM models classify data by optimizing a hyperplane that separates the classes. This can also be viewed as finding the hyperplane that maximizes the margin between the classes. In simple problems, this hyperplane is linear. However, more complicated data often cannot be separated linearly. For these scenarios, the SVM platform provides the option to use a radial basis function kernel to map the points to a nonlinear dimension that can make the classes easier to separate.

When the response is continuous, the models that are fit are known as support vector regression (SVR) models. In a typical regression problem, the goal is to fit a model that minimizes the error between a predicted response and the actual response. In an SVR problem, the goal is to fit a model such that the error between a predicted response and the actual response falls within a range of -ε to ε. This provides a more flexible fit. In JMP Pro, ε is equal to 0.1. The SVR algorithm doubles the data by creating two classes, Y + ε and Y - ε. Then the same algorithm that is used for the classification problem is also used for the prediction (SVR) problem.

The maximization in SVM algorithms is performed by solving a quadratic programming problem. In JMP Pro, the algorithm used by the SVM platform is based on the Sequential Minimal Optimization (SMO) algorithm introduced by John Platt in 1998. Typically, the SVM quadratic programming problem is very large. The SMO algorithm splits the overall quadratic programming problem into a series of smaller quadratic programming problems. Smaller quadratic programming problems are solved analytically instead of numerically, meaning they produce closed-form solutions. Therefore, the SMO algorithm is more efficient than solving the overall quadratic programming problem (Platt 1998).

# JMP PRO **Example of Support Vector Machines**

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 either Low or High. You want to construct a classification model to predict the disease progression for future patients as High or Low. You explore both kernel function options.

1. Select **Help > Sample Data Library** and open Diabetes.jmp.

2. Select **Analyze > Predictive Modeling > Support Vector Machines**.

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**.

7.  In the Model Launch control panel, check that the kernel function is a Radial Basis Function and select Tuning Design.

8.  Enter 10 next to Number of Runs.

9.  Click **Go**.

10. Click the gray triangle next to Model Launch to open the Model Launch control panel.

11. Change the kernel function to a Linear function and select Tuning Design.

12. Enter 10 next to Number of Runs.

13. Click **Go**.

**Figure 9.2**  Model Comparison Report

⊿ **Model Comparison**

| Show | Method | Kernel Function | Cost | Gamma | # SV | Training Misclassification Rate | Validation Misclassification Rate | Validation Generalized RSquare | Probability Threshold | Threshold Validation Misclassification Rate | Best |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | Model 1 | Radial Basis Function | 2.57061 | 0.43877 | 251 | 0.00324 | 0.25564 | 0.00838 | 0.39079 | 0.25564 | |
| ☐ | Model 2 | Radial Basis Function | 3.04271 | 0.31899 | 218 | 0.01294 | 0.28571 | -0.4206 | 0.23223 | 0.28571 | |
| ☐ | Model 3 | Radial Basis Function | 3.93906 | 0.46899 | 255 | 0.00324 | 0.28571 | -0.0081 | 0.53372 | 0.28571 | |
| ☐ | Model 4 | Radial Basis Function | 1.073 | 0.27968 | 208 | 0.0712 | 0.21805 | 0.22158 | 0.17372 | 0.21805 | |
| ☐ | Model 5 | Radial Basis Function | 0.41677 | 0.0281 | 156 | 0.15534 | 0.18797 | 0.50483 | 0.36015 | 0.18797 | |
| ☐ | Model 6 | Radial Basis Function | 0.23369 | 0.39898 | 246 | 0.26861 | 0.28571 | 0.22323 | 6e-8 | 0.28571 | |
| ☐ | Model 7 | Radial Basis Function | 4.8013 | 0.15144 | 167 | 0.05178 | 0.2782 | -0.4245 | 0.26219 | 0.2782 | |
| ☐ | Model 8 | Radial Basis Function | 3.51202 | 0.20295 | 181 | 0.04207 | 0.28571 | -0.454 | 0.22753 | 0.28571 | |
| ☐ | Model 9 | Radial Basis Function | 4.2693 | 0.06093 | 140 | 0.11974 | 0.19549 | 0.42841 | 0.32812 | 0.19549 | |
| ☐ | Model 10 | Radial Basis Function | 1.87641 | 0.0919 | 150 | 0.13269 | 0.21053 | 0.43499 | 0.30598 | 0.21053 | |
| ☐ | Model 11 | Linear | 4.42735 | . | 122 | 0.17476 | 0.17293 | 0.51943 | 0.47722 | 0.17293 | |
| ☐ | Model 12 | Linear | 4.99046 | . | 123 | 0.16828 | 0.17293 | 0.52037 | 0.47551 | 0.17293 | |
| ☐ | Model 13 | Linear | 3.8272 | . | 123 | 0.17476 | 0.17293 | 0.51923 | 0.47763 | 0.17293 | |
| ☐ | Model 14 | Linear | 2.74788 | . | 124 | 0.17152 | 0.17293 | 0.51724 | 0.48019 | 0.17293 | |
| ☐ | Model 15 | Linear | 3.22207 | . | 123 | 0.17476 | 0.17293 | 0.51851 | 0.47842 | 0.17293 | |
| ☐ | Model 16 | Linear | 1.29304 | . | 124 | 0.17152 | 0.17293 | 0.51518 | 0.4819 | 0.17293 | |
| ☐ | Model 17 | Linear | 1.74671 | . | 124 | 0.16828 | 0.17293 | 0.51557 | 0.48143 | 0.17293 | |
| ☐ | Model 18 | Linear | 0.83575 | . | 123 | 0.16505 | 0.15789 | 0.51498 | 0.48207 | 0.15789 | |
| ☐ | Model 19 | Linear | 2.22 | . | 125 | 0.17152 | 0.17293 | 0.51579 | 0.48255 | 0.17293 | |
| ☑ | Model 20 | Linear | 0.04975 | . | 138 | 0.15534 | 0.14286 | 0.52302 | 0.41946 | 0.14286 | Smallest ... |

The Model Comparison report shows that the best model in terms of misclassification rate and RSquare is Model 20. This model has a linear kernel function with cost parameter 0.04975. This is the model to further analyze.

**Figure 9.3**  Model Report for Best Fitting Model



The Model Summary report shows that the misclassification rates for the training and validation sets are very similar. This is a good indication that the model did not overfit the data. The confusion matrices provide more information about the types of observations that are misclassified by the model. In the confusion matrices, the upper left corner shows that the model correctly categorizes the Low responses most of the time (96% in training and 92.6% in validation). However, fewer of the High responses are correctly categorized (53% in training and 68.4% in validation). Therefore, most of the misclassifications are the High responses being misclassified as Low.

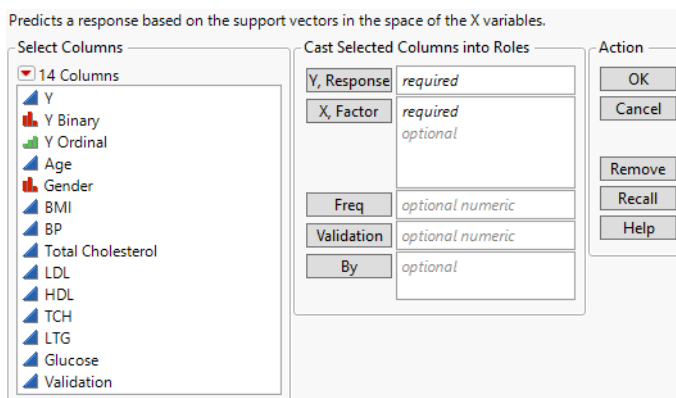# Launch the Support Vector Machines Platform

Launch the Support Vector Machines (SVM) platform by selecting **Analyze > Predictive Modeling > Support Vector Machines**.

Launching the SVM platform is a two-step process. First, enter your variables on the SVM launch window. Second, specify your options in the Model Launch control panel.

## JMP PRO **The Support Vector Machines Launch Window**

Use the SVM launch window to specify your response and model factors. If used, validation and by columns are also specified in the launch windows.

**Figure 9.4**  Support Vector Machines Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The Support Vector Machines launch window provides the following options:

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

**X, Factor**   The predictor variables.

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

**Validation**   A numeric column that defines the validation sets. See "Validation Method". 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".

**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.

## Model Launch Control Panel

When you click OK in the launch window, the Support Vector Machine report window appears, showing a Model Launch control panel for fitting models. Use the Model Launch control panel to specify the kernel function and associated parameter values, as well as the validation method.

**Figure 9.5** The Model Launch Control Panel
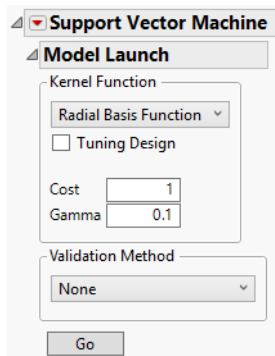


The Model Launch control panel contains the following options:

**Kernel Function**   Specifies the kernel function used in the model. Choose from the following kernel functions:

**Radial Basis Function**   Selects the radial basis function kernel to create a nonlinear hyperplane to separate the classes.

– The Cost parameter is the penalty associated with misclassifying an observation in the training set. A higher cost parameter implements an algorithm that is less likely to misclassify a point in the training set, whereas a lower cost parameter produces a wider margin. The Cost parameter must be greater than 0, and the default value is 1.

– The Gamma parameter is the parameter in the kernel function. This parameter determines the amount of curvature there is to the decision line; a higher Gamma value indicates more curvature. A nonlinear decision line provides a more flexible fit, but too much curvature can lead to overfitting. The Gamma parameter must be greater than 0, and the default value is 1/(# of predictors).

**Linear**   Selects the linear kernel function to create a linear hyperplane to separate the classes.

– The Cost parameter is the penalty associated with misclassifying an observation in the training set. A higher cost parameter implements an algorithm that is less likely to misclassify a point in the training set, whereas a lower cost parameter produces a wider margin. The Cost parameter must be greater than 0, and the default value is 1.

**Note:** If you specify parameter values that are out of range, the default values are used.

**Tip:** To find the best fitting model, fit a range of kernel functions and parameter values and use the Model Comparison report.

**Tuning Design**   Enables you to fit a range of parameter values for the specified kernel. The models with the largest RSquare and the smallest Misclassification Rate or RASE are identified in the Model Comparison report. After you select Tuning Design, you must specify minimum and maximum values for the parameters. Default values are provided based on the data and the minimum must be greater than zero. You must also specify a value for the Number of Runs. The SVM platform fits that many models over a grid of parameter values determined by the minimum and maximum values.

**Validation Method**   Specifies the model validation method. When you click the Go button for the first time, the first SVM model is fit using the specified validation method. This Validation Method is then used for all SVM models fit from within the SVM window. This ensures that all models in the report window are fit using the same validation method and validation set.

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

**KFold**   (Available only when $Y$ is continuous or nominal.) 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. If $Y$ is continuous, the model that has the best validation RASE statistic is chosen as the final model. If $Y$ is nominal, the model that has the best validation misclassification rate is chosen as the final model.

**Validation Column**   (Available only if you specified a Validation column in the launch window.) Uses the values in the specified Validation column to divide the data into parts. The column's values determine how the data are split, and what method is used for validation:

– If there are two values, the smaller value defines the training set and the larger value defines the validation set.

– If there are three values, these values define the training, validation, and test sets in order of increasing size.

– If the validation column has more than three levels, then Validation Column K Fold is used.

The SVM platforms uses the validation column to train and evaluate the model, unless a Tuning Design is used. If the Tuning Design option is selected, the SVM platform uses the validation column to train and tune the model or to train, tune, and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

---

**Note:** If the validation column does not lead to a valid partition of the data, the Holdback validation method is used instead.

---

**Validation Column K Fold**   (Available only when the Y, Response column has exactly two levels and a Validation column is specified in the launch window.) Uses the values in the specified Validation column to divide the data into *K* sets, where *K* is the number of unique values in the column. Then, K-Fold validation is performed.

**None**   No validation used.

**Go**   Fits the specified SVM model and shows the model report.

---

**Note:** If you have a large data table, a progress bar is shown for each model that is fit to the data. The total number of models fit is $k!/2(k-2)!$, where $k$ is the number of levels of the response variable. Each progress bar has an **Accept Current Estimates** button. Click this button if you want to stop the fitting algorithm early and accept the current estimates. Because prediction calculations are performed after you click this button, it may take some time for the report to appear.

---

## Missing Values

Rows that contain any missing predictor values are not included in the SVM modeling procedure. Therefore, any columns saved to the data table will contain missing values in those rows. If you want to include data with missing values in an SVM model, some form of preprocessing is required. See "Explore Missing Values Utility".

# <sub>JMP PRO</sub> The Support Vector Machine Report

When you click **Go** in the Model Launch control panel, a Model Comparison report appears above the control panel and a model report appears below the control panel. The model reports are numbered sequentially. Each time a new model is fit, it is added to the Model Comparison report and a new model report is added to the window. See "Model Comparison Report" and "Support Vector Machine Model Report". Click on the column headings to sort the models by any of the columns in the Model Comparison report. The first click sorts in ascending order; click the column heading a second time to sort in descending order.

## <sub>JMP PRO</sub> Model Comparison Report

The Model Comparison Report contains the following information:

**Show**   A check box that indicates whether the Support Vector Machine Model report for the corresponding fit should appear in the report window.

**Method**   The model number.

**Kernel Function**   The kernel function for the corresponding model.

**Cost**   The value of the cost parameter for the corresponding model.

**Gamma**   The value of the gamma parameter for the corresponding model. If the model uses a linear kernel, this value is missing.

**# SV**   The number of support vectors used in the corresponding model.

**Training Misclassification Rate**   (Appears only if the response is categorical.) The misclassification rate for the observations in the training set. This rate is based on the classification decision rule that is calculated by the platform.

**Validation Misclassification Rate**   (Appears only if the response is categorical and validation is used.) The misclassification rate for the observations in the validation set. This rate is based on the classification decision rule that is calculated by the platform.

**Test Misclassification Rate**   (Appears only if the response is categorical and a test set is specified using a validation column.) The misclassification rate for the observations in the test set. This rate is based on the classification decision rule that is calculated by the platform.

**Training RASE**   (Appears only if the response is continuous.) The square root of the mean squared prediction error (Root Average Square Error) for the observations in the training set. See "RASE".

**Validation RASE**   (Appears only if the response is continuous and validation is used.) The square root of the mean squared prediction error (Root Average Square Error) for the observations in the validation set. See "RASE".

**Test RASE**   (Appears only if the response is continuous and a test set is specified using a validation column.) The square root of the mean squared prediction error (Root Average Square Error) for the observations in the test set. See "RASE".

**Validation Generalized RSquare**   (Appears only if validation is used.) The Generalized RSquare value for observations in the validation set. See "Fit Details".

**Probability Threshold**   (Appears only if the response is binary and validation is used.) The cutoff probability determined by the classification decision rule that is calculated by the platform. An observation is classified into the target level when its predicted probability exceeds this value.

---

**Tip:** You can change this value by using the Set Probability Threshold option in the Confusion Matrix report. See "Confusion Matrix".

---

**Conditional Validation Misclassification Rate**   (Appears only if the response is binary and validation is used.) The misclassification rate for the observations in the validation set, conditioned on the probability threshold value.

**Best**   Indicates which model fit has the smallest misclassification rate. If validation is used without a test set, this is the model with the smallest validation misclassification rate. If validation is used with a test set, this is the model with the smallest test misclassification rate.

## Model Comparison Plot

Shows a contour plot of the model performance of the validation set over the grid of parameter values. This report is available only if Tuning Design is specified in the Model Launch Control Panel.

## Support Vector Machine Model Report

**Response Profile Plot**   (Available only when the response is categorical.) Gives a visual representation of the classification model. The points on the plot are the actual data observations and are on by default only when there are exactly two continuous variables. For the two variables plotted, the shaded contours represent a plane of the prediction space determined by fixed values of the remaining model factors. The predictions are based on the classification decision rule that is calculated by the platform. Controls for the fixed values are located above the plot. When you change the fixed values of the factors, using either the slider or number box, the prediction space for the plotted variables is

automatically updated. You can also change the axes of the plot to display any continuous factor using the red triangles on each axis.

The Response Profile Plot red triangle menu contains the following options:

**Grid Density**   Determines the fineness of the prediction grid underlying the shaded contours. A higher grid density provides a smoother decision line.

**Show Points**   Shows or hides the points on the plot. On by default when there are only two variables.

**Model Summary**   Gives the name of the response column, the validation method, and the type of kernel function used in the model fit. The Model Summary table also contains information about the model fit for the training, validation, and test sets. The number of observations and the number of support vectors are reported for each set. If the response is categorical, the misclassification rate is reported for each set. The misclassification rate is the proportion of observations misclassified by the model. This is calculated as the number of misclassifications divided by the total number of observations. If the response is continuous, the RASE and R Square values are reported for each set.

**Estimation Details**   Gives the values of the parameters used in the model.

**Fit Details**   (Available only if the response is categorical.) Gives the following statistics for the training set, and for the validation and test sets if they are specified:

**Entropy RSquare**   A measure of fit that compares the log-likelihoods from the fitted model and the constant probability model. Entropy RSquare ranges from 0 to 1, where values closer to 1 indicate a better fit. See "Entropy RSquare".

**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.

**RASE**   The square root of the mean squared prediction error (Root Average Square Error). RASE is computed as follows, where *Source* indicates the Training, Validation, or Test set.

$$\text{RASE}_{Source} = \sqrt{\frac{SSE_{Source}}{n}}$$

**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. 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.

**Note:** The misclassification rates in the Fit Details report might not match the misclassification rates in the Confusion Matrix report. When the response is binary, the rates in the Fit Details report use a probability cutoff of 0.5, but the rates in the Confusion Matrix report use the probability threshold value as the cutoff.

**N**   The number of observations.

**Confusion Matrix**   (Available only if the response is categorical.) A confusion matrix is shown for each set (training, validation, and test) specified in the model. A confusion matrix is a two-way classification of actual and predicted responses. Use the confusion matrices and the misclassification rates to evaluate your model.

The confusion matrices and the misclassification rates use the value in the probability threshold box as the cutoff value. By default, this value is based on the classification decision rule that is calculated by the platform. You can change the cutoff value by dragging the slider or entering a new value in the box next to Probability Threshold. If you change the probability threshold, the confusion matrices and misclassification rates automatically update. The probability threshold and conditional validation misclassification rate columns in the Model Comparison report are also updated.

**Actual by Predicted Plot**   (Available only if the response is continuous.) Plots the actual response versus the predicted response. The diagonal line is the Y = X line. The closer the points are to this line, the better the model fits the data. When validation is used, plots are shown for the training, validation, and, if specified, test sets.

# <img> **Support Vector Machines Platform Options**

See *Using JMP* 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.

## Support Vector Machine Model Report Options

Each Support Vector Machine Model report contains the following options:

**Response Profile Plot**   (Available only if the response is categorical.) Shows or hides the Response Profile Plot.

**Support Vector Coefficients**   Shows or hides the table of support vector coefficients. Select rows in the table to highlight the support vectors in the original data table and the Response Profile Plot.

**Confusion Matrix**   (Available only if the response is categorical.) Shows or hides the confusion matrix. See "Confusion Matrix".

**ROC Curve**   (Available only if the response is categorical.) Shows or hides the ROC Curve. If you use validation, ROC Curves for the validation and, if specified, test sets appear in addition to the curve for the training set. For more information, see "ROC Curve".

**Lift Curve**   (Available only if the response is categorical.) Shows or hides the Lift Curve. If you use validation, Lift Curves for the validation and, if specified, test sets appear in addition to the curve for the training set. For more information, see "Lift Curve".

**Plot Actual by Predicted**   (Available only if the response is continuous.) Shows or hides a plot of the actual values versus the predicted values.

**Plot Residual by Predicted**   (Available only if the response is continuous.) Show or hides a plot of the residuals versus the predicted values.

**Profiler**   Shows or hides the Prediction Profiler. Each response level is represented by a separate row in the Prediction Profiler. For more information about the options in the red triangle menu, see *Profilers*.

**Contour Profiler**   (Available only when the model contains more than one continuous factor.) Shows or hides the Contour Profiler. For more information about the options in the red triangle menu, see *Profilers*.

**Surface Profiler**   (Available only when the model contains at least one continuous factor.) Shows or hides a surface profiler. For more information about the options in the red triangle menu, see *Profilers*.

**Save Predicteds**   Saves the predicted classifications based on the threshold probability cutoff value to the data table in a column called Predicted <Y, Response>.

**Save Prediction Formula**   Saves the prediction formulas based on the threshold probability cutoff value to the data table. If the response is binary, a single prediction formula column is added to the data table. If the response has more than two levels, a prediction formula column is added to the data table for each pairwise model, in addition to a formula column for the final prediction.

**Publish Prediction Formula**   Creates probability formulas based on the threshold probability cutoff value 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 "Formula Depot".

**Save Probabilities**   Saves the probability of each response level as a separate column in the data table, as well as a column called Most Likely <Y, Response>.

**Save Probability Formula**   (Available only if the response is binary.) Saves the following formula columns to the data table:

**Decision Value**   A hidden column that contains the formula for the decision value.

**Probability**   Two probability formula columns, one for each level of the binary response.

**Most Likely**   The predicted classification based on which probability value is higher.

**Threshold Predicted**   The predicted classification based on the threshold probability cutoff value obtained from the confusion matrix. By default, this value is based on the classification decision rule that is calculated by the platform. If the threshold value is changed to 0.5, the Threshold Predicted column is not saved to the data table because it would be equivalent to the Most Likely column.

**Publish Probability Formula**   (Available only if the response is binary.) 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 "Formula Depot".

**Save Validation**   (Available only when the Holdback or KFold validation methods are used.) Creates a new column in the data table that identifies which rows were used in the training and validation sets.

**Remove Fit**   Removes the entire model report.

# Additional Example of the SVM Platform

## Example of Support Vector Regression for Continuous Response

Support vector regression (SVR) models use the SVM algorithm to predict continuous response data. This example uses the same medical data on diabetic patients that was used in "Example of Support Vector Machines". Instead of a binary measure of diabetes disease progression, you now have a continuous measure. Higher values correspond to more disease progression. Vary the cost parameter of the radial basis function to find the best fitting model.

1. Select **Help > Sample Data Library** and open Diabetes.jmp.

2. Select **Analyze > Predictive Modeling > Support Vector Machines**.

3. Select Y and click **Y, Response**.

4. Select Age through Glucose and click **X, Factor**.

5. Select Validation and click **Validation**.

6. Click **OK**.

7. In the Model Launch control panel, check that the kernel function is a Radial Basis Function with Cost parameter 1 and Gamma parameter 0.1.

8. Click **Go**.

9. Click the gray triangle next to Model Launch to open the Model Launch control panel.

10. Change the cost parameter to 0.1.

11. Click **Go**.

12. Click the gray triangle next to Model Launch to open the Model Launch control panel.

13. Change the cost parameter to 2.

14. Click **Go**.

**Figure 9.6** Model Comparison Report

| Show | Method | Kernel Function | Cost | Gamma | # SV | Training RASE | Validation RASE | Validation Generalized RSquare | Best |
|---|---|---|---|---|---|---|---|---|---|
| ✓ | Model 1 | Radial Basis Function | 1 | 0.1 | 266 | 43.8147 | 58.3285 | 0.42552 | |
| ✓ | Model 2 | Radial Basis Function | 0.1 | 0.1 | 280 | 52.5804 | 57.5025 | 0.44167 | Smallest RASE |
| ✓ | Model 3 | Radial Basis Function | 2 | 0.1 | 265 | 40.553 | 60.7801 | 0.37621 | |

Validation Column: Validation

The Model Comparison report contains the results of the models using different cost parameters. Recall that a higher cost parameter implements an algorithm that is less likely to misclassify a point, whereas a lower cost parameter implements a more flexible

algorithm. In this scenario, the more flexible model (cost parameter equal to 0.1) produces the best model fit.

15. In the Show column of the Model Comparison report, deselect Model 1 and Model 3.

**Figure 9.7**  Model Report for Continuous Response
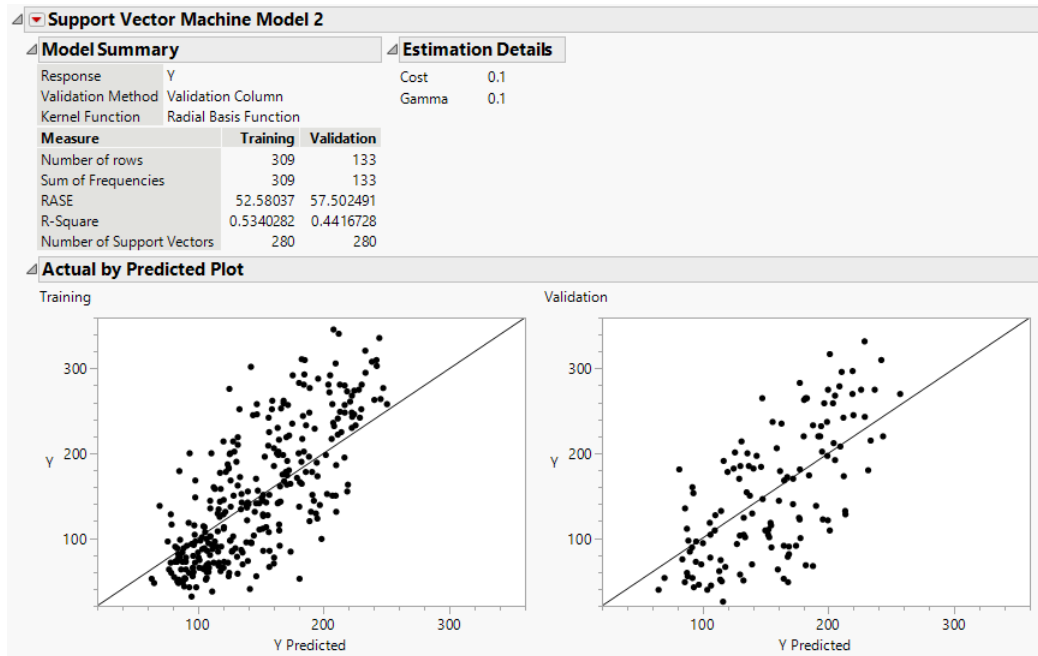


The model report for a continuous response contains the Model Summary, Estimation Details, and Actual by Predicted Plot. In the Model Summary report, the RASE is 52.58 for the training set and 57.50 for the validation set.
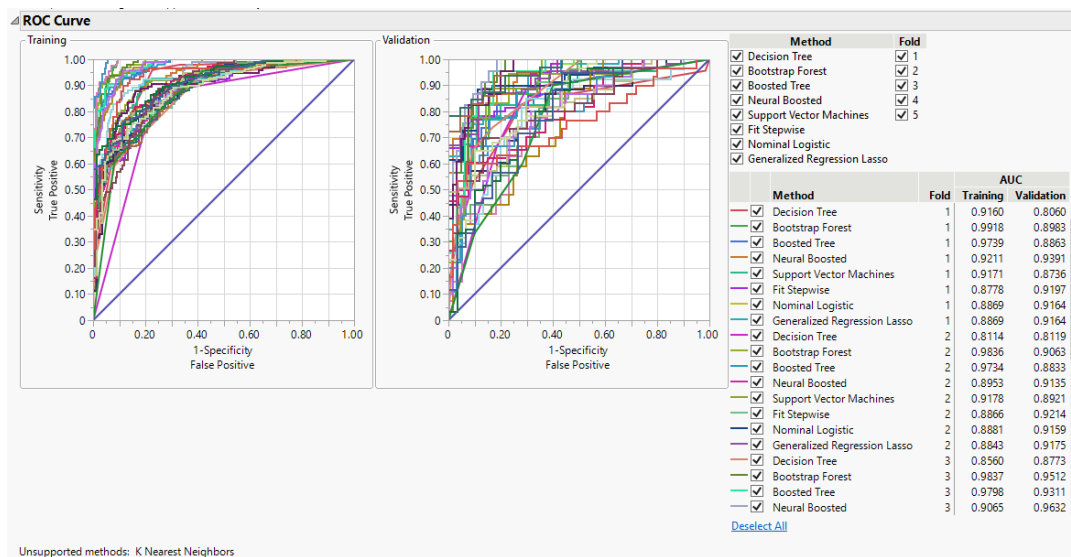
**JMP PRO Model Screening**

## Fit and Compare Multiple Prediction Models

*The Model Screening platform is available only in JMP Pro.*

The Model Screening platform in JMP Pro enables you to run multiple predictive modeling platforms from one launch window and assemble summaries from the different methods. This helps you pick the best performing model without launching each individual model platform. You can also easily launch an individual platform report for any of the models from the Model Screening report window for further refinement.

**Figure 10.1**  Model Screening Report

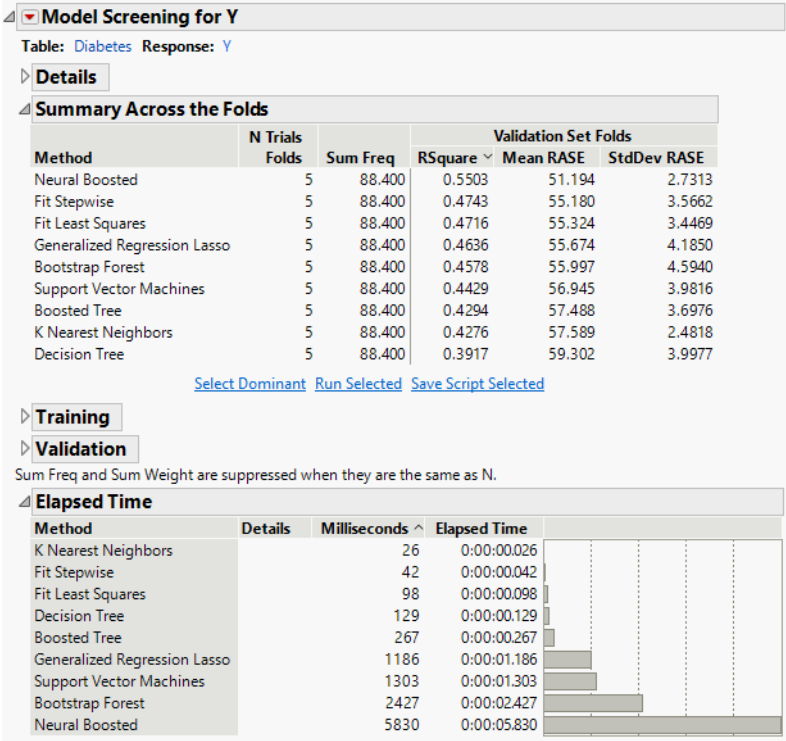# Contents

# Example of Model Screening for Continuous Response

You have baseline medical data for 442 diabetic patients. You also have a continuous measure of diabetes disease progression obtained one year after each patient's initial visit. Higher values correspond to more disease progression. You want to assess various models for their ability to predict diabetes disease progression.

1. Select **Help > Sample Data Library** and open Diabetes.jmp.
2. Select **Analyze > Predictive Modeling > Model Screening**.
3. Select Y and click **Y, Response**.
4. Select Age through Glucose and click **X, Factor**.
5. In the Folded Crossvalidation section, select the box next to **K Fold Crossvalidation**.
6. (Optional) In the Options section, type 42920 next to **Set Random Seed**.

   Use the random seed in order to match the example output.
7. Click **OK**.
8. Click the red triangle next to Model Screening for Y and select **Optional Reports > Elapsed Time**.

**Figure 10.2** Model Report for Continuous Response

| Model Screening for Y | | | | | | |
|---|---|---|---|---|---|---|
| Table: Diabetes  Response: Y | | | | | | |
| ▷ Details | | | | | | |
| ⊿ Summary Across the Folds | | | | | | |

| | N Trials | | Validation Set Folds | | | |
|---|---|---|---|---|---|---|
| **Method** | **Folds** | **Sum Freq** | **RSquare** ⌄ | **Mean RASE** | **StdDev RASE** | |
| Neural Boosted | 5 | 88.400 | 0.5503 | 51.194 | 2.7313 | |
| Fit Stepwise | 5 | 88.400 | 0.4743 | 55.180 | 3.5662 | |
| Fit Least Squares | 5 | 88.400 | 0.4716 | 55.324 | 3.4469 | |
| Generalized Regression Lasso | 5 | 88.400 | 0.4636 | 55.674 | 4.1850 | |
| Bootstrap Forest | 5 | 88.400 | 0.4578 | 55.997 | 4.5940 | |
| Support Vector Machines | 5 | 88.400 | 0.4429 | 56.945 | 3.9816 | |
| Boosted Tree | 5 | 88.400 | 0.4294 | 57.488 | 3.6976 | |
| K Nearest Neighbors | 5 | 88.400 | 0.4276 | 57.589 | 2.4818 | |
| Decision Tree | 5 | 88.400 | 0.3917 | 59.302 | 3.9977 | |

Select Dominant  Run Selected  Save Script Selected

| ▷ Training | | | | |
|---|---|---|---|---|
| ▷ Validation | | | | |

Sum Freq and Sum Weight are suppressed when they are the same as N.

| ⊿ Elapsed Time | | | | |
|---|---|---|---|---|
| **Method** | **Details** | **Milliseconds** ^ | **Elapsed Time** | |
| K Nearest Neighbors | | 26 | 0:00:00.026 | |
| Fit Stepwise | | 42 | 0:00:00.042 | |
| Fit Least Squares | | 98 | 0:00:00.098 | |
| Decision Tree | | 129 | 0:00:00.129 | |
| Boosted Tree | | 267 | 0:00:00.267 | |
| Generalized Regression Lasso | | 1186 | 0:00:01.186 | |
| Support Vector Machines | | 1303 | 0:00:01.303 | |
| Bootstrap Forest | | 2427 | 0:00:02.427 | |
| Neural Boosted | | 5830 | 0:00:05.830 | |

The best performing model based on the Validation RSquare averaged across the folds, is Neural Boosted. The average Validation RSquare for Neural Boosted is 0.5503. The Elapsed Time report shows that the Neural Boosted models also took the longest to fit. In this case, the elapsed time was only 5 seconds, but this information could be important for larger data sets or more complex models.

**Tip:** To see performance metrics for the individual models, click the gray disclosure icon next to Validation.

# Launch the Model Screening Platform

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

**Figure 10.3** The Model Screening Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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

**X, Factor**    The predictor variables.

**Weight**    (Not applicable to the K Nearest Neighbors, Support Vector Machines, or Neural modeling platforms.) A column whose numeric values assign a weight to each row in the analysis.

**Freq**   (Not applicable to the K Nearest Neighbors modeling platform.) A column whose numeric values assign a frequency to each row in the analysis.

**Validation**   (Not applicable if any of the Crossvalidation options are selected in the launch window.) A numeric column that defines the validation 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".

**Note:** If you specify a validation column with more than three levels, this column is used to perform K Fold crossvalidation.

**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.

**Methods**   Enables you to select the desired modeling platforms. By default, the modeling platforms that are fit are Decision Tree (Partition), Bootstrap Forest, Boosted Tree, K Nearest Neighbors, Neural, Support Vector Machines, Discriminant, Fit Least Squares, Fit Stepwise, Logistic Regression, and Generalized Regression. Naive Bayes, Partial Least Squares, and XGBoost are also available.

**Notes:**

- XGBoost is not supported by JMP and is available only if the XGBoost add-in is installed. For information about XGBoost, see https://community.jmp.com.

- Decision Tree (Partition), Discriminant, and Partial Least Squares all require some type of validation set in order to fit a model.

- If there are fewer than 20 observations in a validation set, a Decision Tree (Partition) model cannot be fit.

- The modeling platforms use default options and tuning parameters in model fitting. You can try to improve the fit past what the default yields by calling platforms directly and choosing different options.

## Options

Provides additional options.

**Remove Live Reports**   Does not include the individual model platform reports in the Model Screening report window.

**Tip:** Select this option to free up memory when you have a large problem with many methods and fits.

**Log Methods**   Writes out a progress message to the log each time a fitting platform is called.

**Time Limit Each**   Specifies a time limit, in seconds, for each fit. For platforms that support early stopping, the best estimates up to that point are provided. For platforms that do not support early stopping, no result is provided.

**Set Random Seed**   Sets a random seed that is used for any random components of the model fit routines. This enables you to rerun the platform and obtain the same model fits.

## Folded Crossvalidation

Provides options for various types of crossvalidation.

**K Fold Crossvalidation**   Divides the data randomly into $K$ parts or folds. Each model is fit to the data $K$ times, each time with a different fold held out as a crossvalidation set. A total of $K$ models are fit. The default value of $K$ is 5.

– $K$ specifies the number of folds for K Fold Crossvalidation. The default is 5 and $K$ must be greater than 1.

**Nested Crossvalidation**   Divides the data into nested folds for crossvalidation. First, the data are divided into $k = 1, ..., K$ equals parts, or *folds*. For each fold, the $k^{th}$ fold is used as a test set and the remaining data are divided further into $L$ equal parts. These $L$ subdivisions are called *inner folds*. Then, a model is fit to the data $L$ times with a different inner fold held out each time as a crossvalidation set. The $L$ models then use the $k^{th}$ fold as a common test data set. In all, a total of $K*L$ models are fit. The default value of $K$ is 4 and the default value of $L$ is 5.

For example, set K = 2 and L = 3. The data are initially divided into two folds. The first fold is held out as a test set and the second fold is divided into 3 inner folds. Three models are fit to the data, each time with a different inner fold held out as a crossvalidation set. Then, all three models are tested on the first fold.

The second fold is then held out as a test set and the first fold is divided into 3 inner folds. Three models are fit to the data, each time with a different inner fold held out as a crossvalidation set. Then, all three models are tested on the second fold.

– $K$ specifies the number of folds for Nested Crossvalidation. The default is 4 and $K$ must be greater than 1.

– $L$ specifies the number of inner folds for Nested Crossvalidation. The default is 5 and $L$ must be greater than 1.

> **Note:** If both K Fold Crossvalidation and Nested Crossvalidation are selected, Nested Crossvalidation is performed.

**Repeated K Fold**   Specifies the number of times the K Fold Crossvalidation or Nested Crossvalidation process is repeated.

### Modeling Options

Provides additional options for the modeling platforms.

**Add Two Way Interactions**   Adds all two way interaction effects to linear models.

**Add Quadratics**   Adds effects for the squares of continuous variables to linear models.

**Informative Missing**   Enables informative missing for all platforms.

**Additional Methods**   Calls several additional methods, such as Ridge, Elastic Net and Lasso, in the Generalized Regression platform. See *Fitting Linear Models*.

> **Caution:** This results in additional model fits.

When you click OK, the specified models are fit and a set of progress bars are shown. The upper progress bar reports the progress across all fits. The lower progress bar reports the progress for the current individual model fit. You can stop the lower progress bar to employ early stopping and the upper progress bar will continue to run.

## The Model Screening Report

The content of the Model Screening report depends on both the type of crossvalidation implemented and the modeling type of the response variable. At the top of the Model Screening report, the name of the data table and column name of the response variable are shown. If a validation column is specified, the name of that column is also shown.

By default, the Details report appears next. This report contains the individual model reports for all of the models for each fold and trial. These reports are the actual output from each platform and contain all reports and command features. For example, if you decide on a model that is best, you can go to the corresponding platform report in the Details report and save the desired prediction formula column. You can hide the Details report by selecting the Remove Live Reports option in the launch window.

If either the K Fold Crossvalidation option or the Nested Crossvalidation option is specified in the launch window, the Summary Across the Folds report appears. This report is a summary of the measures of fit averaged across the test folds.

Measures of fit are shown for each individual model fit, for the training, validation, and test sets if applicable. If crossvalidation is used by specifying a validation column in the launch window, only the individual model fit measures of fit are shown in the report.

## Summary Across the Folds Report

The Summary Across the Folds report contains a summary of the measures of fit across the folds, as well as across the trials if Repeated K Fold is specified. If the K Fold Crossvalidation option is specified, the measures of fit are summarized across the validation sets. If the Nested K Fold Crossvalidation option is specified, the measures of fit are summarized across the test sets. The report contains a table with the following columns:

**Method**    The name of the method used to fit the model.

**N Trials Folds**    The total number of models that were fit across all folds and trials, if applicable.

**Sum Freq**    The average number of observations in the validation or test sets. The test sets are used to estimate model performance.

**RSquare**    The mean RSquare across all validation or test set folds. This column contains the Entropy RSquare when the response is categorical.

**Mean RASE**    The mean RASE (Root Average Square Error) across all validation or test set folds.

**StdDev RASE**    The standard deviation of the RASE across all validation or test sets.

**Mean AUC**    (Available only for categorical responses.) The mean area under the ROC curve (AUC) across all validation or test sets.

**Mean MR**    (Available only for categorical responses.) The mean misclassification rate (MR) across all validation or test sets.

The following options are available below the table:

**Select Dominant**    Selects each model that is better than or equal to all of the other models in terms of a combination of model fitting criteria. This is also referred to as selecting the Pareto Frontier. For continuous responses, RSquare and Sum Freq are considered when determining the dominant model. For categorical responses, Entropy RSquare, Misclassification Rate, AUC, and Sum Freq are considered when determining the dominant model.

**Run Selected**    Runs the individual model for each selected row. If any type of folded crossvalidation is specified in the launch window, the model run uses the fold, inner fold, and trial combination that corresponds to the final model. The final model is the model that produces the highest weighted average RSquare. The weight average RSquare is the

average, weighted by number of observations, of the training RSquare, validation
RSquare, and when a test set is used, the test RSquare. A validation column is created in
the data table for any fold, inner fold, and trial combination that is required by clicking
Run Selected.

**Save Script Selected** Saves a model script to the script window for each selected row. If any
type of folded crossvalidation is specified in the launch window, the validation set
specified in the script uses the fold, inner fold, and trial combination that corresponds to
the final model. The final model is the model that produces the highest weighted average
RSquare. The weight average RSquare is the average, weighted by number of
observations, of the training RSquare, validation RSquare, and when a test set is used, the
test RSquare. A validation column is created in the data table for any fold, inner fold, and
trial combination that is requested by clicking Save Script Selected.

## Training, Validation, and Test Measures of Fit

There is a measures of fit report for each model data set that is specified. These could be
Training, Training and Validation, or Training, Validation, and Test sets. Each report contains a
table with the following columns:

**Method** The name of the method used to fit the model.

**N** The number of observations in the set.

**Sum Wgt** The sum of the weights.

**RSquare** (Available only for continuous responses.) The RSquare value of the fitted model.

**Entropy RSquare** (Available only for categorical responses.) A measure of fit that compares
the log-likelihoods from the fitted model and the constant probability model. Entropy
RSquare ranges from 0 to 1, where values closer to 1 indicate a better fit. See "Entropy
RSquare".

**Misclassification Rate** (Available only for categorical responses.) The proportion of
observations misclassified by the model. Smaller values indicate a better fit.

**Note:** In these tables, the misclassification rate is always calculated using a probability
threshold of 0.5.

**AUC** (Available only for categorical responses.) The area under the ROC curve. Values closer
to 1 indicate a better fit.

**RASE**   The square root of the mean squared prediction error (Root Average Square Error). RASE is computed as follows, where *Source* indicates the Training, Validation, or Test set.

$$\text{RASE}_{Source} = \sqrt{\frac{SSE_{Source}}{n}}$$

**Generalized RSquare**   (Available only for categorical responses.) 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).

**Fold**   (Available only if the K Fold Crossvalidation option or the Nested Crossvalidation option is specified in the launch window.) Identifies the fold that is held out for the model fit in that row.

**Inner Fold**   (Available only if the Nested Crossvalidation option is specified in the launch window.) Identifies the inner fold that is held out for the model fit in that row.

**Trial**   (Available only if the Repeated K Fold option is specified in the launch window.) Identifies the trial number for the model fit in that row.

The following options are available below each table:

**Select Dominant**   Selects each model that is better than or equal to all of the other models in terms of a combination of model fitting criteria. For continuous responses, RSquare and Sum Freq are considered when determining the dominant model. For categorical responses, Entropy RSquare, Misclassification Rate, AUC, and Sum Freq are considered when determining the dominant model.

**Run Selected**   Runs the individual models specified in each selected row.

**Save Script Selected**   Saves a model script to the script window for each selected row.

# Model Screening Platform Options

**Plot Actual by Predicted**   (Available only for continuous responses.) Shows or hides a report of plots that overlay the actual by predicted points from all of the model fits. There are separate plots for the Training, Validation, and Test sets. The report contains a Methods legend with check boxes that enable you to specify which fits appear on the plots based on the method, fold, inner fold, and trial combination. Click the Deselect All button at the bottom of the legend to remove all model fits from the plots.

**Notes:**

– If you select a subset of models from the Training, Validation, or Test reports and then select Plot Actual by Predicted, only the plots for the selected models are shown.

– The actual by predicted plots are not supported by all of the modeling platforms. A list of unsupported platforms is shown below the plots.

**Profiler** (Available only for continuous responses.) Shows or hides prediction profilers for each type of model fit. See *Profilers*. If any type of folded crossvalidation is specified in the launch window, the selected model in the prediction profiler uses the fold, inner fold, and trial combination that corresponds to the final model. The final model is the model that produces the highest weighted average RSquare. The weight average RSquare is the average, weighted by number of observations, of the training RSquare, validation RSquare, and when a test set is used, the test RSquare.

**Notes:**

– Confidence intervals are shown in prediction profilers for Standard Least Squares models and Generalized Regression models.

– Prediction profilers are not available for K Nearest Neighbors models.

**ROC Curve** (Available only for categorical responses.) Shows or hides a report of plots that overlay the ROC curves for all of the model fits. There are separate plots for the Training, Validation, and Test sets. The report contains two legends with check boxes that enable you to specify which fits appear on the plots. Click the Deselect All button at the bottom of the legend to remove all model fits from the plots. For more information about ROC Curves, see "ROC Curve".

**Notes:**

– If you select a subset of models from the Training, Validation, or Test reports and then select ROC Curves, only the plots for the selected models are shown.

– The ROC plots are not supported by all of the modeling platforms. A list of unsupported platforms is shown below the plots.

**Show Profit** (Available only for binary categorical responses that have a Profit Matrix column property.) Shows or hides the expected profit for each model.

**Decision Threshold** (Available only for binary categorical responses.) Shows or hides Decision Thresholds reports for the training, validation, and test sets, if specified. Each report contains a graph of the distribution of fitted probabilities for each model, confusion matrices for each model, and classification graphs to adjust thresholds and compare the model fits. See "Decision Thresholds Report".

**Note:** If you select a subset of models from the Training, Validation, or Test reports and then select Decision Threshold, only the information for the selected models is shown.

**Save Results Table**   Saves the information in the Validation report to a new data table titled Model Screening Statistics Validation Set. If there is a test set, such as when the Nested Crossvalidation option is specified, the information in the Test report is also saved to a new data table titled Model Screening Statistics Test Set.

**Save KFold Results Table**   (Available only if a Crossvalidation option is specified in the launch window.) Saves the information in the Summary Across the Folds report to a new data table titled KFold Results.

**Save Prediction Formulas**   (Available only if the live reports are included in the model report.) Saves the prediction formulas for the selected models to the data table. If you select a model in the Summary Across the Folds report and any type of fold crossvalidation is specified in the launch window, the prediction formula is for the fold, inner fold, and trial combination that corresponds to the final model. The final model is the model that produces the highest weighted average RSquare. The weight average RSquare is the average, weighted by number of observations, of the training RSquare, validation RSquare, and when a test set is used, the test RSquare.

**Notes:**

– A prediction formula is not saved for Stepwise models.

– A prediction formula is not saved for SVM models when the response is ordinal.

**Optional Reports**   Shows a submenu of additional report options.

**Elapsed Time**   Shows or hides a report that contains the total elapsed time that was spent fitting each type of model.

**Cardinality of Predictors**   Shows or hides a report of the number of levels and how many parameters are used in the linear model fit for each categorical predictor.

See *Using JMP* 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.

# Decision Thresholds Report

The Decision Thresholds report enables you to explore thresholds for binary classification models. There is a Decision Thresholds report for each model data set that is specified by the validation method. These could be Training, Training and Validation, or Training, Validation, and Test sets. Each Decision Thresholds report contains a graph of the distribution of fitted probabilities, a bar chart of classifications, and confusion matrices. All are organized by model fit, fold, trial, and class level. The report also contains a tabbed section on classification accuracy measures and an option to set the profit matrix. The report updates as you adjust the probability threshold.

### Distribution of Fitted Probabilities

The distribution of fitted probabilities, or model scores, enables you to see how each individual model fit differentiates between the two classes. A vertical line on the graph represents the probability threshold, which determines the classification of each observation. By default, the probability threshold is 0.5. You can change the probability threshold by dragging the vertical line or by clicking the Probability Threshold value and entering a new value. This changes the probability threshold value across the whole Decision Thresholds report. The value of the probability threshold must be between 0 and 1.

### Classification Counts

The bar chart of classifications shows the classification counts for each level of the response variable at the current threshold. Green bars represent correctly classified observations; red bars represent incorrectly classified observations.

### Confusion Matrices

The confusion matrices, also known as contingency tables, show the two-way classification of actual and predicted responses for each individual model fit. Confusion rates matrices are shown as well. The rates are equal to the values in the confusion matrices divided by the row totals.

### Classification Accuracy Measures

**False Classification by Threshold**   Shows a plot of the misclassification count by probability threshold and a plot of the misclassification rate by probability threshold. Each plot contains two curves for each individual model fit. The curves for the low response category are solid and the curves for the high response category are dashed. The curves intersect at the threshold that yields equal misclassification, counts or rates, of each response level. There is also a vertical line on each graph that represents the current

probability threshold value. You can change the probability threshold value by dragging the vertical line. This changes the probability threshold value across the whole report.

**False Classification by Portion**   Shows a plot of the misclassification count or rate by the portion of the rank ordered scores. Each plot contains two curves for each individual model fit. The curves for the low response category are solid and the curves for the high response category are dashed.

**True Classification by Threshold**   Shows a plot of the true count by the probability threshold and a plot of the true rate by the probability threshold. Each plot contains two curves for each individual model fit. The curves for the low response category are dashed and those for the high response level are solid. The curves intersect at the threshold that yields equal correct classifications, counts or rates, for each response. There is also a vertical line on each graph that represents the current probability threshold value. You can change the probability threshold value by dragging the vertical line. This changes the probability threshold value across the whole report.

**True Classification by Portion**   Shows a plot of the true count or rate by the portion of the rank ordered scores. Each plot contains two curves for each individual model fit. The curves for the low response category are dashed and the curves for the high response category are solid.

**Profit by Threshold**   (Available only if a profit matrix is specified.)Shows a plot of the average profit by the probability threshold. There is a curve for each individual model fit and a vertical line that represents the current probability threshold value. The specified profit matrix is also shown next to the plot.

**Metrics**   Shows a table of classification accuracy metrics for each model. A legend is provided to describe how the metric in each column is calculated.

## Set Profit Matrix

Enables you to assign costs to undesirable outcomes and profits to desirable outcomes. See "Specify Profit Matrix". If you change the probability threshold in the profit matrix window and click OK, the Decision Thresholds report is updated using that value as the probability threshold.

*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 11.1** Example of Comparing Models

# Contents

# 🔷 **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. Select **Analyze** > **Predictive Modeling** > **Make Validation Column**.
2. Do not select any columns in the launch window.

   This indicates that the platform will create a simple random validation column
3. Click **OK**.
4. In the box next to New Column Name, type Create Validation.
5. In the box next to Random Seed, enter 1234.
6. Click **Go**.

   A new Validation column is created. 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 crim through lstat and click **Add**.
4. Select Create Validation and click **Validation**.
5. Select **Stepwise** in the Personality list.
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.

**Figure 11.2** Fit Model Report

| Term | Estimate | Std Error | t Ratio | Prob>|t| |
|------|----------|-----------|---------|---------|
| Intercept | 38.500819 | 5.676983 | 6.78 | <.0001* |
| crim | -0.128236 | 0.03545 | -3.62 | 0.0003* |
| zn | 0.0434609 | 0.015555 | 2.79 | 0.0055* |
| chas[0] | -1.215423 | 0.490413 | -2.48 | 0.0136* |
| nox | -15.38954 | 4.180964 | -3.68 | 0.0003* |
| rooms | 3.8274522 | 0.445164 | 8.60 | <.0001* |
| distance | -1.501159 | 0.218407 | -6.87 | <.0001* |
| radial | 0.3364509 | 0.073112 | 4.60 | <.0001* |
| tax | -0.012471 | 0.003826 | -3.26 | 0.0012* |
| pt | -0.994528 | 0.146259 | -6.80 | <.0001* |
| b | 0.0088057 | 0.003206 | 2.75 | 0.0063* |
| lstat | -0.59558 | 0.054166 | -11.00 | <.0001* |

**Response mvalue**

Validation: Create Validation

▷ **Effect Summary**

◢ **Summary of Fit**

| | |
|---|---|
| RSquare | 0.753984 |
| RSquare Adj | 0.74663 |
| Root Mean Square Error | 4.787344 |
| Mean of Response | 23.02868 |
| Observations (or Sum Wgts) | 380 |

◢ **Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|--------|-----|----------------|-------------|---------|
| Model | 11 | 25848.508 | 2349.86 | 102.5306 |
| Error | 368 | 8434.069 | 22.92 | Prob > F |
| C. Total | 379 | 34282.577 | | <.0001* |

◢ **Parameter Estimates**

10. To save the prediction formula to a column, click the Response red triangle and select **Save Columns > Prediction Formula**.

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

1. Select **Analyze > Predictive Modeling > Bootstrap Forest**.

2. Select mvalue and click **Y, Response**.

3. Select crim through lstat and click **X, Factor**.

4. Select Create Validation and click **Validation**.

5. Click **OK**.

6. Select the **Early Stopping** check box.

7. Select the **Multiple Fits over number of terms** check box.

8. Enter 617 in the box next to **Random Seed**.

9. Click **OK**.

**Figure 11.3** Bootstrap Forest Model

| Bootstrap Forest for mvalue | | | |
|---|---|---|---|
| **Specifications** | | | |
| Target | mvalue | Training Rows: | 380 |
| Validation Column: | Create Validation | Validation Rows: | 126 |
| | | Test Rows: | 0 |
| Number of Trees in the Forest: | 100 | Number of Terms: | 13 |
| Number of Terms Sampled per Split: | 10 | Bootstrap Samples: | 380 |
| | | Minimum Splits per Tree: | 10 |
| | | Minimum Size Split: | 5 |

**Overall Statistics**

| Individual Trees | RASE |
|---|---|
| In Bag | 1.997080 |
| Out of Bag | 5.217378 |

| | RSquare | RASE | N |
|---|---|---|---|
| Training | 0.938 | 2.3691145 | 380 |
| Validation | 0.842 | 3.1747064 | 126 |

10. To save the prediction formula to a column, click the Bootstrap Forest red triangle and select **Save Columns > Save Prediction Formula**.

## Compare the Models

1. Select **Analyze > Predictive Modeling > Model Comparison**.
2. Select the two prediction formula columns and click **Y, Predictors**.
3. Select Create Validation and click **Group**.

   **Tip:** If a **Group** column is not specified, JMP automatically recognizes when the same validation column has been used for all predictors and prompts you to add it as a grouping variable.

4. Click **OK**.

**Figure 11.4** Model Comparison Report

| Model Comparison | | | | | | | |
|---|---|---|---|---|---|---|---|
| ▷ Predictors | | | | | | | |
| **Measures of Fit for mvalue** | | | | | | | |

| Create Validation | Predictor | Creator | .2 .4 .6 .8 | RSquare | RASE | AAE | Freq |
|---|---|---|---|---|---|---|---|
| Training | Pred Formula mvalue 2 | Fit Least Squares | | 0.7540 | 4.7111 | 3.3770 | 380 |
| Training | mvalue Predictor 2 | Bootstrap Forest | | 0.9378 | 2.3691 | 1.4382 | 380 |
| Validation | Pred Formula mvalue 2 | Fit Least Squares | | 0.6550 | 4.6972 | 3.3155 | 126 |
| Validation | mvalue Predictor 2 | Bootstrap Forest | | 0.8424 | 3.1747 | 1.8776 | 126 |

The rows in the training set were used to build the models, so the RSquare statistics for Create Validation = Training 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 Create Validation = Validation. In this case, the bootstrap forest model predicts better than the regression model.

5. Click the Model Comparison red triangle and select Profiler.

**Figure 11.5** Prediction Profiler for All Models
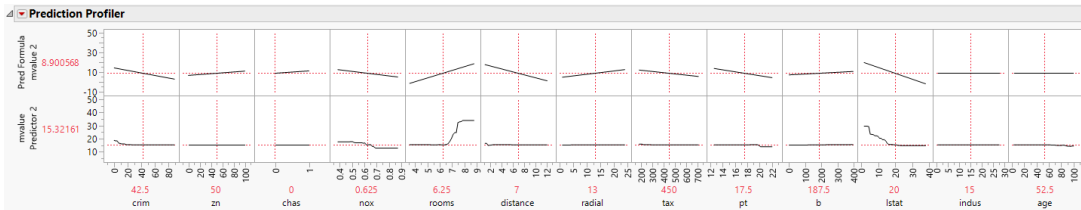


The prediction profiler enables you to compare the impact of each factor in the different models. The profiler is especially interesting when comparing different types of models such as here where you have a regression model and a partition model.

**Related Information**

- *Fitting Linear Models*
- "Partition Models"

# Launch the Model Comparison Platform

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

**Figure 11.6** The Model Comparison Launch Window
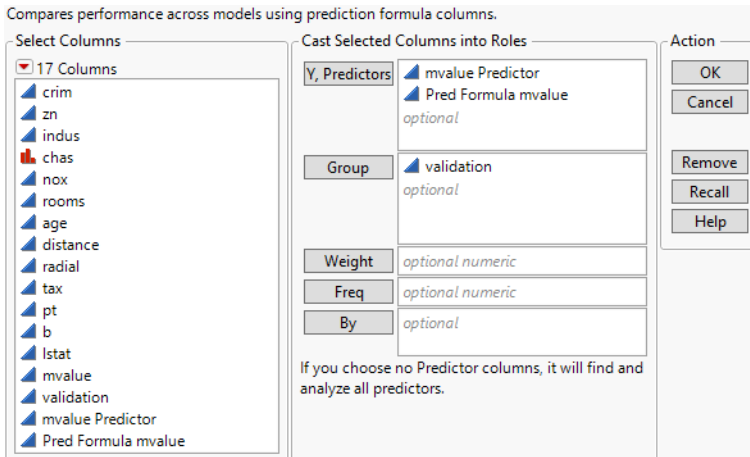
For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**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 evaluated separately. If a **Group** column is not specified, JMP automatically recognizes when the same validation column has been used for all predictors and prompts you to add it as a grouping variable.

The other role buttons are common among JMP platforms. See *Using JMP*.

## The Model Comparison Report

Figure 11.7 shows an example of the initial Model Comparison report for a continuous response.

**Figure 11.7**  Initial Model Comparison Report



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

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

## Measures of Fit for Continuous Responses

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

**RASE**   The 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.

**RASE**   The root average squared prediction error. 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" provides more information about measures of fit for categorical responses.

# Model Comparison Platform Options

## 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, where 1 is 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 more information about these matrices, see "Additional Examples of Partitioning".

**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.

**Decision Threshold**   (Available only for binary categorical responses.) Shows or hides Decision Thresholds reports for the training, validation, and test sets, if specified. Each report contains a graph of the distribution of fitted probabilities for each model, confusion matrices for each model, and classification graphs to compare the model fits. See "Decision Thresholds Report".

**Related Information**

- "ROC Curve"
- "Lift Curve"

# 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. To save the prediction formulas to columns, click the Nominal Logistic red triangle and select **Save Probability Formula**.

### 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. To save the prediction formulas to columns, click the Partition red triangle and select **Save Columns > Save Prediction Formula**.

### 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**.

**Figure 11.8**  Initial Model Comparison Report

| Creator | .2 .4 .6 .8 | Entropy RSquare | Generalized RSquare | Mean -Log p | RASE | 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 |

⊿ ▾ Model Comparison
▷ Predictors
⊿ Measures of Fit for Type

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. Click the Model Comparison red triangle and select **ROC Curve**.

   ROC curves appear for each Type, one of which is shown in Figure 11.9.

**Figure 11.9** ROC Curve for Medium
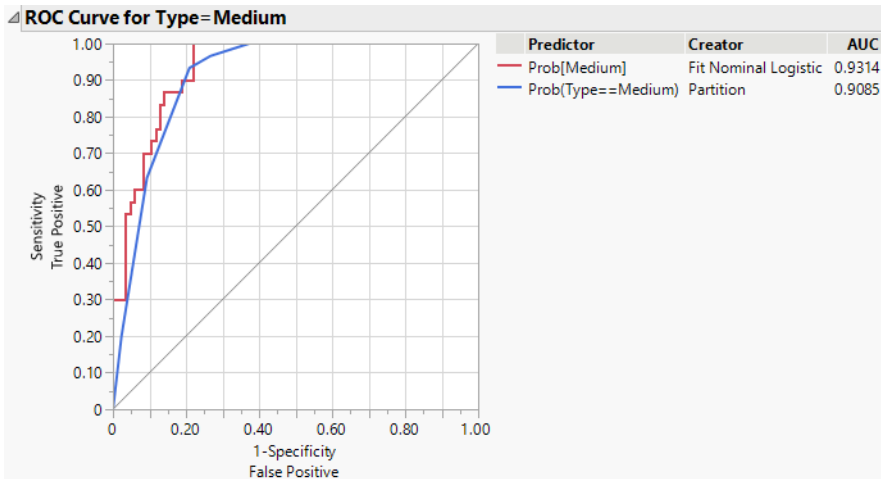


Examining all the ROC curves, you see that the two models are similar in their predictive ability.

5.  Click the Model Comparison red triangle and select **AUC Comparison**.

    AUC Comparison reports appear for each Type, one of which is shown in Figure 11.10.

**Figure 11.10** AUC Comparison for Medium



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.

### **Make Validation Column**

**Divide Data into Training, Validation, and Test Sets**

*The Make Validation Column platform is available only in JMP Pro.*

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. You can partition the data into two or three sets. The Make Validation Column platform provides five different methods to create these partitions.

# Contents

# Overview of the Make Validation Column Platform

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 many JMP platforms. See "Validation in JMP Modeling".

The Make Validation Column platform enables you to create training, validation, and test sets using a variety of methods. You can specify stratification, grouping, or cutpoint columns to determine the method used to create the validation column.

# Example of Make Validation Column

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 creating a validation column to use during future analyses.

1. Select **Help > Sample Data Library** and open Lipid Data.jmp.
2. Select **Analyze > Distribution**.
3. Select Gender and click **Y, Columns**.
4. Click **OK**.

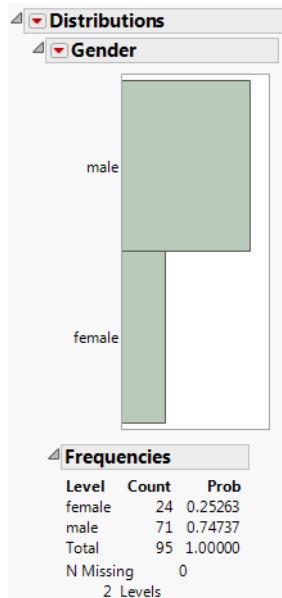**Figure 12.1** Distribution of Gender in Lipid Data.jmp



Figure 12.1 illustrates the distribution of Gender in the data set. Notice that there is not an equal proportion of males and females represented. Because there are fewer females within the data, you want to be sure to balance the genders across the validation and training sets.

5. Select **Analyze > Predictive Modeling > Make Validation Column**.

6. Select Gender and click **Stratification Columns**.

7. Click **OK**.

The Make Validation Column report appears with a description of the validation method you selected. There are also options to change the rates, column types, or set a seed.

8. (Optional) Type 1234 next to **Random Seed** in the Options section of the report.

9. Click **Go**.

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.

10. Select **Analyze > Fit Y by X**.

11. Assign Validation to **Y, Response**, and Gender to **X, Factor**.

12. Click **OK**.

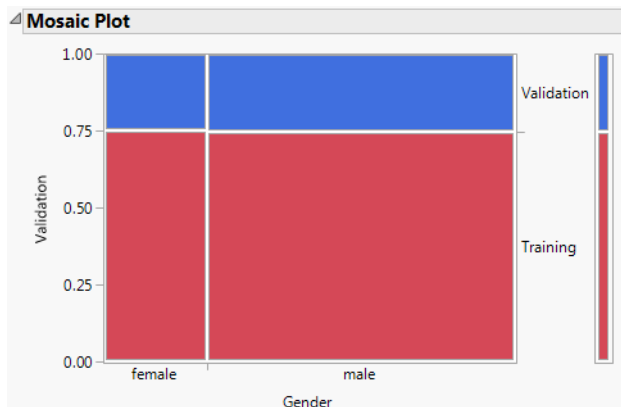**Figure 12.2** Distribution of Gender across Validation and Training Sets



Figure 12.2 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 Platform

Launch the Make Validation Column platform by selecting **Analyze > Predictive Modeling > Make Validation Column**.

**Figure 12.3** Make Validation Column Launch Window
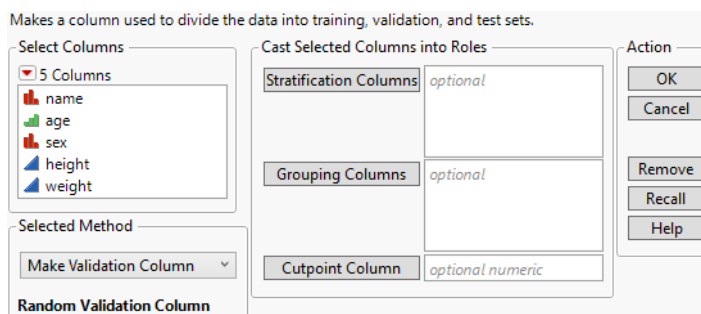


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The Make Validation Column launch window provides the following options:

**Stratification Columns**   Assigns one or more stratification columns.

**Grouping Columns**   Assigns one or more grouping columns.

**Cutpoint Column**    Assigns a numeric cutpoint column.

**Cutpoint Batch ID**    When a cutpoint column is assigned, you can also assign a column for cutpoint batch IDs. This enables you to determine cutpoint values within each level of the Cutpoint Batch ID column.

## Selected Method

Provides two methods for validation.

**Make Validation Column**    creates a validation column based on the specified stratification, grouping, and cutpoint columns. The validation column method determined by the specified stratification, grouping, and cutpoint columns is described below the box. After a method is selected and you click OK, you specify the allocations for each set in the Make Validation Column report. See "Specify Rates or Relative Rates" and "Set Cutpoints". There are five methods for constructing the holdback sets.

**Random Validation Column**    The default method if there are no column assignments in the launch window. This method partitions the data into sets based on the allocations entered in the Make Validation Column report.

**Stratified Validation Column**    The selected method if one or more stratification columns are assigned. This method partitions the data into balanced sets based on the levels of the specified stratification columns. As in the Random Validation Column method, rows are randomly assigned to the holdback sets based on the allocations entered in the Make Validation Column report. However, this is done at each level or combination of levels of the stratifying columns. Use this method when you want a balanced representation of the levels of a column in each of the training, validation, and test sets.

**Grouped Validation Column**    The selected method if one or more grouping columns are specified. This method 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 set. Because of this, the sizes of the resulting sets vary slightly from the sizes that you specified. Use this option when splitting levels across holdback sets is not desirable.

**Stratify by Group Validation Column**    The selected method if both stratification and grouping columns are specified. This method partitions the data to balance the levels across the stratification column while requiring that the specified groups stay together in the same holdback sets. As in Grouped Validation Column, groups can be created as levels of a specified column or combination of levels of two or more columns. The sizes of the resulting sets vary slightly from the sizes you specified.

**Cutpoint Validation Column**   The selected method if a cutpoint column is specified. This method partitions the data into sets based on the time series cutpoints. Use this option when you want to assign your data to holdback sets based on time periods. 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 based on options in the Set Cutpoints report.

**Make Autovalidation Table**   Creates a new data table that contains a duplication of the rows in the original data table concatenated to the rows in the original data table. The new data table, which can be used for crossvalidation, has four additional columns:

**Valid Set**   Assigns a value of 0 to the original data and a value of 1 to the duplicated data. The values in this column designate the training and validation sets. Use this column in the Validation role in the launch window of an analysis.

**Valid ID**   Assigns the row number of the original observation. This allows matching of training and validation set rows for each original observation.

**Valid Weight**   Assigns the autovalidation weight, to be used in the Freq role in the launch window of an analysis. For each value of Valid ID, the same uniform random number is generated for the training observation and the validation observation. For the training set, Valid Weight is calculated as follows:

Valid Weight = -log(1 - Valid Uniform)

For the validation set, Valid Weight is calculated as follows:

Valid Weight = -log(Valid Uniform)

The Valid Weight column is constructed so that the training data weights are negatively correlated with the validation data weights. This ensures that the difference in the fit of the validation data yields an effective crossvalidation of the fitting method.

**Null Factor**   Assigns the same normal random number for each value of Valid ID.

**Tip:** Use Make Autovalidation Table for small data tables, where using a subset as the training data could cause estimation problems.

# JMP PRO　Make Validation Column Report

When you click OK in the launch window, the Make Validation Column report window appears. The report contains a description of the validation method you selected and relevant options for rates, cutpoints, column type, and setting a random seed.

## JMP PRO　Specify Rates or Relative Rates

This report enables you to specify the allocations for the training, validation, and test sets. In the boxes 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.

Depending on the number of rows in the data table and the selected validation method, the actual sizes of the resulting sets might vary slightly from the sizes that you specified. The Adjusted Rates, Row Counts, and Group Count columns provide the actual sizes of the sets.

## JMP PRO　Set Cutpoints

This report is shown only when a cutpoint column is specified in the launch window. The cutpoints are determined by one of the following four methods:

**Proportions**　Determines the cutpoints based on the proportions of rows specified for each set. In the boxes next to Training Set, Validation Set, and Test Set, enter values that represent the proportions that you would like to include in each of these sets. Depending on the number of rows and the proportions, the actual sizes of the resulting sets might vary slightly from the sizes you specified.

**Number of Rows**　Determines the cutpoints based on the number of rows specified for each set. In the boxes next to Training Set, Validation Set, and Test Set, enter values that represent the number of rows that you would like to include in each of these sets. This option enables you to specify the sets exactly.

**Fixed Time or Date**　Determines the cutpoints based on fixed data points in the specified cutpoint column. If you select this option, the minimum and maximum values for the cutpoint column are shown for reference. In the boxes next to Training Set, Validation Set, and Test Set, enter the value that represents the minimum value that you would like to include in each of these sets.

**Elapsed Time**　Determines the cutpoints based on the amount of time that has elapsed since the first timestamp in the cutpoint column. If you select this option, the total elapsed time from the first value to the last value is shown for reference. In the boxes next to Training

Set, Validation Set, and Test Set, enter the values that represent the amounts of elapsed time that you would like to include in each of these sets.

## Options

You can also specify the following additional options:

**New Column Name**    Enables you to specify the name of the validation column.

**Validation Column Type**    (Not available for the Cutpoint Validation Column method.) Enables you to specify the validation column type as fixed or formula.

- The fixed column type produces a column with fixed values.
- The formula column type produces a formula column as the validation column.

**Random Seed**    (Not available for the Cutpoint Validation Column method or if a formula column type is specified.) Enables you to set a random seed to reproduce the same validation column in the future.

## Go

Click **Go** in the Make Validation Column report to add the validation column to the data table. The validation column has a Notes property that gives the stratifying, grouping, or cutpoint variables.

## Make Validation Column Platform Options

The Make Validation Column Utility has the following red triangle menu options:

See *Using JMP* 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.

# Additional Example of the Make Validation Column Platform

This example uses weekly weather data collected over one year from 16 weather stations across the United States. Run the Weather Station Locations script in the data table to view a map of the locations. Not every weather station has a weekly temperature measurement for every week of the year. You are interested in creating a validation column for this data table based on the dates of data collection. For each weather station, you want to use the first 60% of observations for training, the next 25% of observations for validation, and the final 15% of observations for testing. This example shows the importance of using a Batch ID column in this type of scenario.

### Create Validation Column with Cutpoints

1. Select **Help > Sample Data Library** and open Functional Data/Weekly Weather Data.jmp.
2. Select **Analyze > Predictive Modeling > Make Validation Column**.
3. Select Week of Year and click **Cutpoint Column**.
4. Click **OK**.
5. In the list next to Determine cutpoints using, select **Proportions**.
6. In the boxes next to Training Set, Validation Set, and Test Set, enter 0.60, 0.25, and 0.15, respectively.
7. In the box next to New Column Name, type Cutpoint Validation.
8. Click **Go**.

   A validation column called Cutpoint Validation is added to the data table.
9. Select **Analyze > Tabulate**.
10. Click ID and drag it to the Drop zone for rows.
11. Click Cutpoint Validation and drag it on top of N.
12. Click Row% and drag it on top of the cells.

**Figure 12.4**  Cutpoint Validation Column Proportions

| | Cutpoint Validation | | |
| | Training | Validation | Test |
| ID | Row % | Row % | Row % |
| --- | --- | --- | --- |
| Bristol, TN | 62.79% | 20.93% | 16.28% |
| Corpus Christi, TX | 70.27% | 18.92% | 10.81% |
| Denver, CO | 58.14% | 23.26% | 18.60% |
| Goodwell, OK | 64.71% | 17.65% | 17.65% |
| Grand Canyon National Park, AZ | 53.66% | 26.83% | 19.51% |
| Greenville, ME | 60.47% | 20.93% | 18.60% |
| Harrisburg, PA | 60.53% | 26.32% | 13.16% |
| Helena, MT | 60.00% | 25.00% | 15.00% |
| Lynchburg, VA | 71.79% | 17.95% | 10.26% |
| Madera, CA | 67.50% | 22.50% | 10.00% |
| Miami Beach, FL | 63.16% | 26.32% | 10.53% |
| Minneapolis/St. Paul, MN | 57.50% | 27.50% | 15.00% |
| N. Myrtle Beach, SC | 69.05% | 16.67% | 14.29% |
| Peason Ridge Range, LA | 57.89% | 28.95% | 13.16% |
| Philip, SD | 59.52% | 23.81% | 16.67% |
| Terra Haute, IN | 47.22% | 33.33% | 19.44% |

Figure 12.4 shows that not all of the weather stations have the correct proportions for the training, validation, and test sets. Use a Batch ID column to obtain the correct percentages.

## Create Validation Column with Cutpoints and Batch ID

1. Select **Analyze > Predictive Modeling > Make Validation Column**.
2. Select Week of Year and click **Cutpoint Column**.
3. Select ID and click **Cutpoint Batch ID**.
4. Click **OK**.
5. In the list next to Determine cutpoints using, select **Proportions**.
6. In the boxes next to Training Set, Validation Set, and Test Set, enter 0.60, 0.25, and 0.15, respectively.
7. In the box next to New Column Name, type Cutpoint Batch Validation.
8. Click **Go**.

   A validation column called Cutpoint Batch Validation is added to the data table.
9. Select **Analyze > Tabulate**.
10. Click ID and drag it to the Drop zone for rows.
11. Click Cutpoint Batch Validation and drag it on top of N.
12. Click Row% and drag it on top of the cells.

**Figure 12.5** Cutpoint Validation Column with Batch ID Proportions

| ID | Cutpoint Batch Validation | | |
| | Training Row % | Validation Row % | Test Row % |
| --- | --- | --- | --- |
| Bristol, TN | 60.47% | 25.58% | 13.95% |
| Corpus Christi, TX | 59.46% | 24.32% | 16.22% |
| Denver, CO | 60.47% | 25.58% | 13.95% |
| Goodwell, OK | 58.82% | 26.47% | 14.71% |
| Grand Canyon National Park, AZ | 60.98% | 24.39% | 14.63% |
| Greenville, ME | 60.47% | 25.58% | 13.95% |
| Harrisburg, PA | 60.53% | 26.32% | 13.16% |
| Helena, MT | 60.00% | 25.00% | 15.00% |
| Lynchburg, VA | 58.97% | 25.64% | 15.38% |
| Madera, CA | 60.00% | 25.00% | 15.00% |
| Miami Beach, FL | 60.53% | 26.32% | 13.16% |
| Minneapolis/St. Paul, MN | 60.00% | 25.00% | 15.00% |
| N. Myrtle Beach, SC | 59.52% | 26.19% | 14.29% |
| Peason Ridge Range, LA | 60.53% | 26.32% | 13.16% |
| Philip, SD | 59.52% | 26.19% | 14.29% |
| Terra Haute, IN | 61.11% | 25.00% | 13.89% |

Figure 12.5 shows that using a Cutpoint Batch ID column ensures that each weather station has proportions for the training, validation, and test sets that are much closer to the specified values.

**PRO** **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 13.1** Example of the Formula Depot
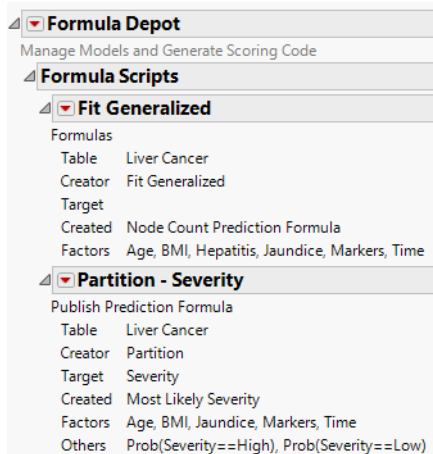
## Contents

# JMP PRO **Overview of the Formula Depot Platform**

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

- save intermediate cleaning formulas within a prediction formula

- compare models

- profile models

- add prediction formulas to a data table for scoring of new data within JMP

- generate scoring code (C, Python, JavaScript, SAS, or SQL) for deploying models outside of JMP

- The Formula Depot includes intermediate formulas in the generated scoring code. A model constructed from inputs with column formulas includes the corresponding formulas in the scoring code. For example, if your model includes a recoded variable the recode formula is included in the scoring code. This enables you to apply your prediction formula to raw data that has not been processed.

**Note:** Informative missing is not a supported intermediate formula.

# JMP PRO **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 Poisson 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 in the Formula Depot.

    **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

5.  To generate scoring code for use outside of JMP select a code type from the Fit Generalized
    - Node Count red triangle. A script window appears containing the code. See "Generating
    Scoring Code from the Formula Depot Platform".

6.  To save the Formula Depot, select **File > Save**.

    **Note:** The Formula Depot is saved as a JMP Report File (*.jrp).

**Figure 13.2**  Formula Depot with a Generalized Model
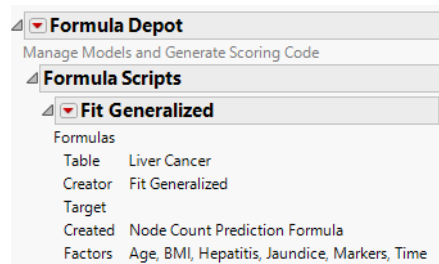


## Launch the Formula Depot Platform

Launch the Formula Depot by selecting **Analyze > Predictive Modeling > Formula Depot**.

**Figure 13.3**  Empty Formula Depot from Launch
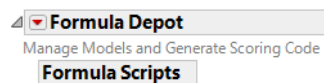


Alternatively, if there is not an open Formula Depot then a Formula Depot opens when you
select a Publish command.

## Platforms That Publish Prediction Formulas to the Formula Depot

The platforms that publish prediction formulas and generate 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 more information about scoring code see "Generating Scoring Code from the Formula Depot Platform".

In addition, you can publish a formula directly from the formula editor. Select the formula, right-click, and select **Publish expression to formula depot**.

## Formula Depot Platform Options

**Add Formula From Column**   Enables you to add existing prediction formulas from data table columns to the current formula depot. If your table has only one formula, that formula is added to the depot. If you have multiple column formulas, a dialog box enables you to select the formulas of interest. If your formula columns are selected in your data table, they will be pre-selected in the dialog box.

**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 Functions**   Enables you to select models from the current formula depot
to be copied as functions for use in JSL. Selected models are copied onto the clipboard. The
functions act on scalar variables rather than columns. One Namespace is generated for
each selected model. Each Namespace has one function, called predict, that takes the
model inputs as parameters and returns the model predictor.

**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. Formulas with intermediate column formulas will add the
intermediate columns to your JMP data table.

---

**Note:** If you have more than one data table open, a Choose Table window appears that
enables you to specify which data table to use for Run Scripts.

---

**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".

**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, first select the table of interest and then the models of interest. Hold the shift
key when selecting the Model Comparison option to launch the Model Comparison in a
new window. See "Model Comparison".

---

**Note:** If you have more than one data table open, a Choose Table window appears that
enables you to specify which data table to use for the Model Comparison.

---

**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. Hold the shift key when selecting the profiler
option to launch the profiler in a new window. See *Profilers*.

**Note:** If you have more than one data table open, a Choose Table window appears that enables you to specify which data table to use for the Profiler.

**Remove Profiler**    Removes all Profilers from the current formula depot.

See *Using JMP* 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.

## 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. Error messages often indicate an unsupported function call that requires additional programming in the targeted language. Target language specific libraries might be required for the implementation of more complex functions. These libraries are discussed in the language specific sections below. 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.

---

**Tip:** To find the location of installation folder on your machine, use the JSL command `Get Path Variable("$ALL_HOME/Scoring");`.

---

**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. When deploying your models, copy jmp_score.py to the same location as your executable scoring code.

When calling the generated code to score your data, it is important that the input data is passed in a data structure compatible with the code generated by JMP. The Python scoring code expects the input and output arguments to be of a dictionary-like type. Values can be accessed and written by key, such as DataFrame row objects from the Spark, Dask and Pandas frameworks, or the standard Python "dict" class.

Use of linear algebra operators or advanced operators such as Vec Quadratic and Design Norm require the NumPy Python library.

**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. The code is added to a SAS Window and includes a variable name mapping section in the comment block that precedes the code. Any temporary variable names associated with intermediate columns are deleted at the end of the DATA Step by the drop command.

---

**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.

---

**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.
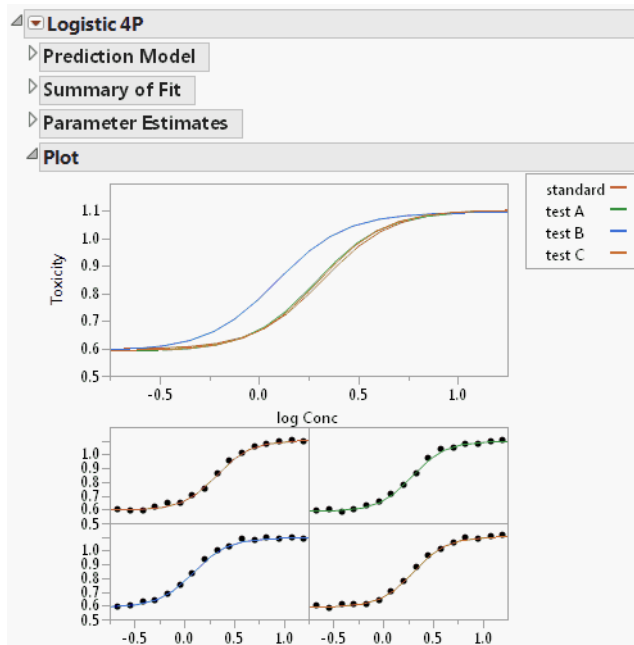
# Fit Curve

### 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 platform 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 platform, which can also fit any nonlinear model. See "Nonlinear Regression".

**Figure 14.1** Example of Nonlinear Fit in the Fit Curve Platform

# Contents

# Overview of 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 *Fitting Linear Models* shows a significant linear relationship between oxygen uptake and time spent running. For more information about Fit Model, see *Fitting Linear Models*. For more information about Fit Y by X, see *Basic Analysis*.

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:   $Y = \beta_0 + \beta_1 e^{\beta_2 x}$

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. See "Example of Setting Parameter Limits". In JMP Pro, you can also specify a set of supplementary variables and fit a generalized regression model within the Fit Curve platform to determine how these variables affect the response.

# Example Using the Fit Curve Platform

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 non-surviving to 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 higher 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. The Plot report contains an overlaid plot of the fitted model of each formulation.

**Figure 14.2**  Initial Fit Curve Report
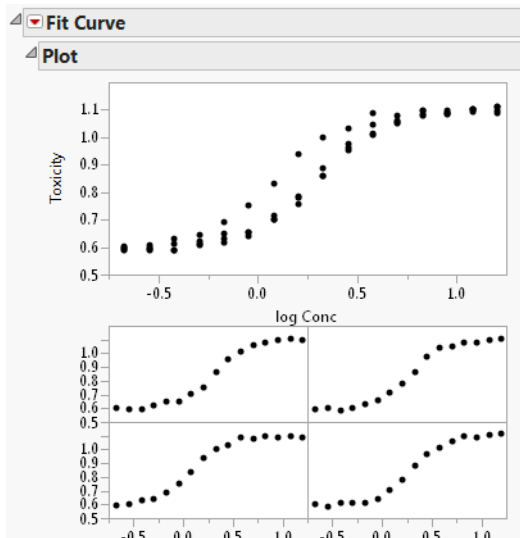
7. To see a legend identifying each drug formulation, right-click one of the graphs and select
   **Row Legend**. Select **Formulation** for the column and click **OK**.

**Figure 14.3** Fit Curve Report with Plot Legend
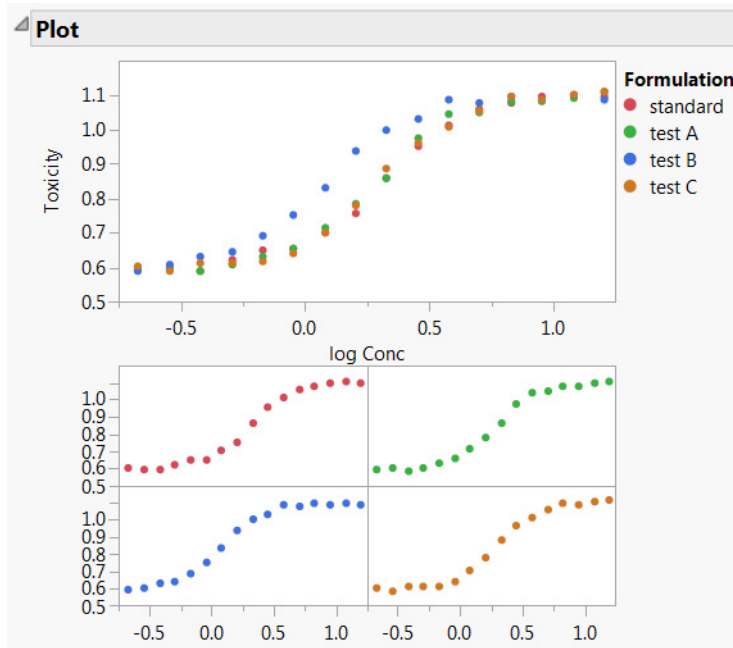


The curves appear S-shaped, so a sigmoid curve would be an appropriate fit. Table 14.1
shows formulas and graphical depictions of the different types of models that the Fit
Curve platform offers.

8. Click the Fit Curve red triangle and select **Sigmoid Curves > Logistic Curves > Fit Logistic
   4P**.

**Figure 14.4** Logistic 4P Report
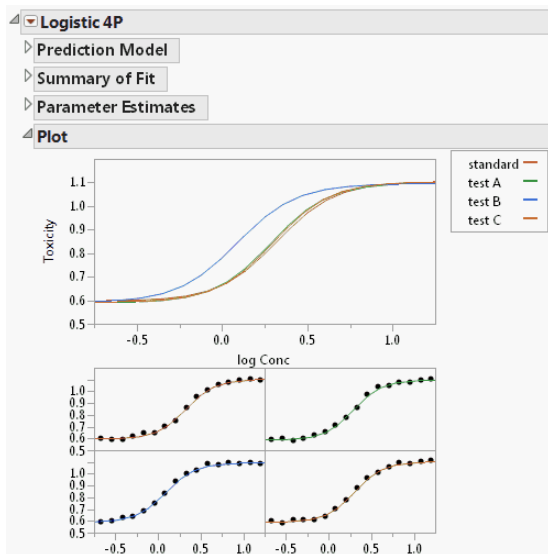


The Logistic 4P report appears. There is also a separate plot for each drug formulation. The plot of the fitted curves suggests that formulation B might be different, because the test B curve starts to rise sooner than the others. Inflection point parameters cause this rise.

9.  Click the Logistic 4P red triangle and select **Compare Parameter Estimates**.

**Figure 14.5** Parameter Comparison Report
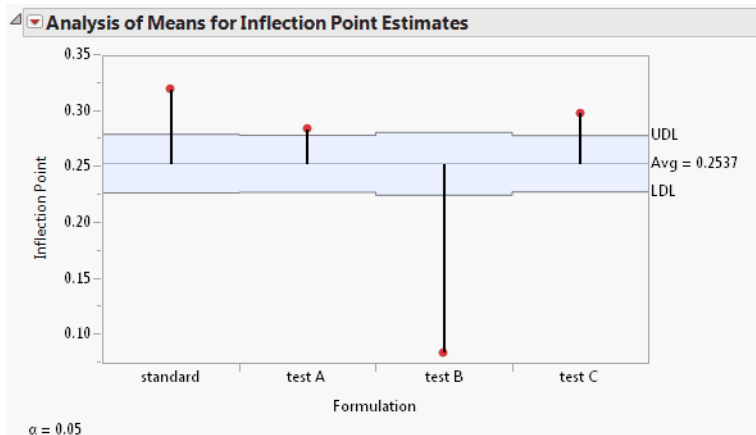


Notice that the Inflection Point parameter for the test B formulation is significantly lower than the average inflection point. This agrees with the plots shown in Figure 14.4. Drug formulation B has a higher toxicity ratio than the other formulations.

# Launch the Fit Curve Platform

To launch the Fit Curve platform, select **Analyze > Specialized Modeling > Fit Curve**. The launch window is shown in Figure 14.6.

**Figure 14.6** Fit Curve Platform Launch Window
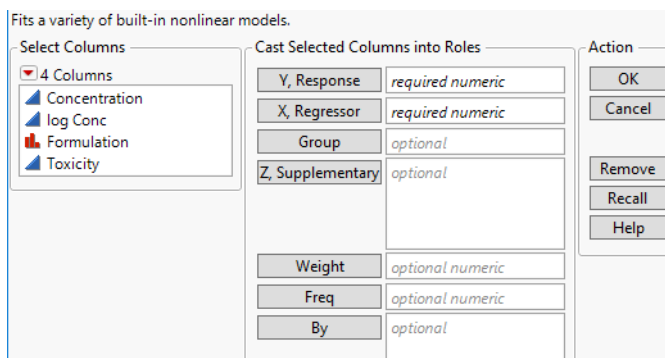


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The Fit Curve platform launch window has the following features:

**Y, Response**   Specify the *Y* variable.

**X, Regressor**   Specify 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.

**Z, Supplementary**   Specify one or more supplementary variables. Supplementary variables are not used in any of the calculations in the Fit Curve platform and including them does not affect the results. Supplementary variables are variables you might want to use in future analyses of the results from Fit Curve. When you specify supplementary variables, they are included in the table that is created by the Make Parameter Table option. In JMP Pro, if you specify one or more supplementary variables and a grouping variable, a Curve DOE Analysis option is made available for the fitted models. See "Curve DOE Analysis".

**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. If you specify a Group variable, the report includes overlaid and individual plots for each group of the fitted model.

**Figure 14.7** Fit Curve Reports: No Grouping Variable (left) and with Group Variable (right)
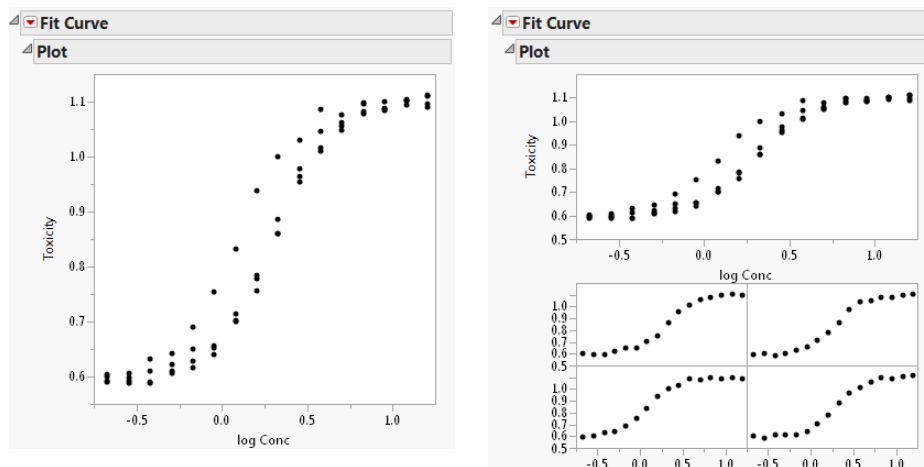


After fitting a model, the fitted model appears on the plot (when no grouping variable is specified on the platform launch window). A Model Comparison report appears above the plots and a Model Report appears below the plot. Each time a new model is fit, it appears in the Model Comparison report and a new model report appears in the report window. See "Model Comparison Report" and "Model Fit Report".

**Note:** Any rows that are excluded in the data table are also hidden in the Fit Curve plot.

# Fit Curve Options

Select any of the following built-in models from the Fit Curve red triangle menu. See "Model Formulas".

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

**Sigmoid Curves** Fits Logistic, Probit, Gompertz, and Weibull 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. The Weibull Growth is available only when both the response values and regressor values are

non-negative. 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, Mechanistic Growth, and Cell 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 (or decrease), but the rate of growth (or decay) 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 *Using JMP* 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.

## Model Comparison Report

To create the report shown in Figure 14.8, select **Sigmoid Curves > Logistic Curves > Fit Logistic 4P** and **Sigmoid Curves > Fit Gompertz 4P** from the Fit Curve red triangle menu.

**Figure 14.8** Model Comparison Report

| 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 be computed only 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 *Fitting Linear Models*.

**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 best fit. In our example, the Logistic 4P model is clearly the better fit. The AICc weights are calculated using only nonmissing AICc values:

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 *Fitting Linear Models*.

**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 "Model Comparison".

# Model Fit Report

A report is created for each fitted model. The red triangle menu for each model report contains 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, Chi-squared statistics, *p*-values, and confidence intervals. The correlations and covariances of the estimates are given also.

**Tip:** You can sort the Parameter Estimates report table by clicking on the column that you want to sort by.

**Plot**   Gives plots of the data with the fitted model (Figure 14.7). The plots are shown only when you select a Grouping variable on the platform launch window.

# Model Fit Options

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. See "Test Parallelism".

**Area Under Curve**   Gives the area under the fitted curve. This option is available for the following models: One Compartment, Two Compartment, Gaussian Peak, and Lorentzian Peak. This option is also available for Bi-Exponential 4P Models, but only when all parameters are positive. 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.

**Time to Peak Response**   Displays the estimate of the regressor, *X*, at the peak of the fitted curve. The standard error of the estimate is also shown. This option is available for Cell Growth 4P and One Compartment models.

**Peak Response**   Displays the estimate of the response, *Y*, at the peak of the fitted curve. The standard error of the estimate is also shown. This option is available for Cell Growth 4P and One Compartment models.

**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. See "Compare Parameter Estimates".

**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. See "Equivalence Test".

**JMP PRO** **Curve DOE Analysis**   (Available only if a grouping variable and at least one supplementary variable are specified in the launch window.) Launches a Generalized Regression report within the Fit Curve platform. A generalized regression model is fit to each parameter of the nonlinear model using the supplementary variables as model effects. By default, a two degree factorial model is fit and the Estimation Method is Forward Selection. These models are then combined to create a profiler of the response as a function of the regressor variable and the supplementary variables. You can then use the CDOE Profiler to explore how the supplementary variables affect the response.

The Curve DOE Analysis report contains the following red triangle menu options:

**Generalized Regression for Model Parameters**   Shows or hides the Generalized Regression reports for each model parameter. For more information on Generalized Regression model reports, see *Fitting Linear Models*.

**Diagnostic Plots**   Shows or hides actual by predicted and residual plots for the response variable.

**CDOE Profiler**   Shows or hides the CDOE Profiler, which enables you to explore how the response changes based on the supplementary variables. For more information about the CDOE Profiler red triangle menu options, see *Profilers*.

**Save Prediction Formula**   Saves the prediction formula for the response to a new column in the data table.

**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 *Profilers*.

**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 custom 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 *Fitting Linear Models*.

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

## 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 14.9). 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 14.9** Parallelism Test
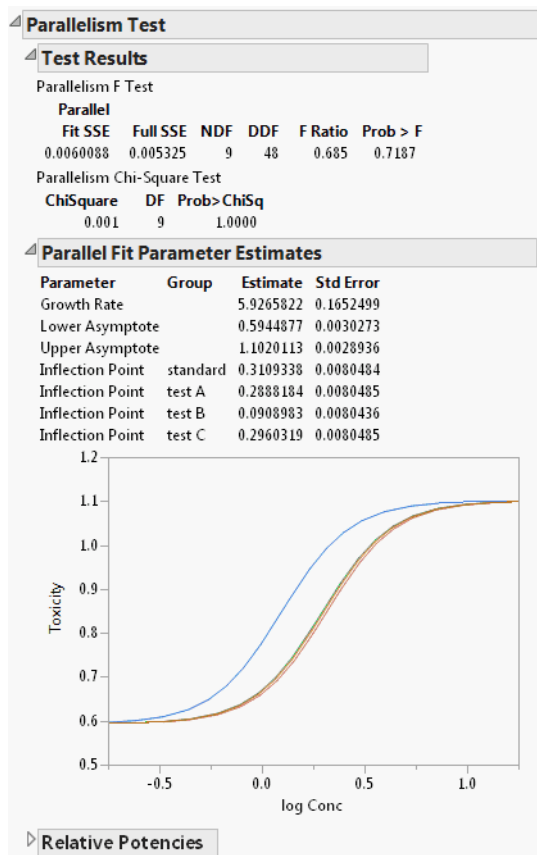


The report gives the following results:

**Test Results**  Gives the results of an *F* Test and a Chi-Square Test for parallelism. The tests compare the error sums-of-squares for a full model (Full SSE) and a reduced model (Fit SSE). The full model gives each group different parameters. The reduced model forces the groups to share every parameter except for the inflection point. A small *p*-value indicates that the group models are significantly different from one another. Therefore, the reduced model with shared parameters is insufficient and the full model should be used. In Figure 14.9, the *p*-value is greater than 0.05, indicating that there is not enough evidence to conclude that differences exist between the curves. In this case, the reduced model with shared parameters is appropriate.

The fit statistics are calculated as follows:

**F Ratio**  [(Fit SSE - Full SSE) / NDF] / [Full SSE / DDF]

**ChiSquare**  Fit SSE - Full SSE

**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 potency and relative potency for each level of the grouping variable. The potency is $10^\wedge(EC_{50})$, 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 potency is $10^\wedge$(inflection point parameter). The relative potency is the potency of one level of the grouping variable divided by the potency of another level. For example, in Figure 14.10, the relative potency of test A to standard is $Potency_{test A}/Potency_{standard}$.

**Figure 14.10**  Relative Potencies by Group



**Relative Potencies**

**Relative Potency versus standard**

| Group | Potency | Relative 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**

| Group | Potency | Relative 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**

| Group | Potency | Relative 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**

| Group | Potency | Relative 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 14.10, 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 14.11 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 14.11** Parameter Comparison for Growth Rate Estimates
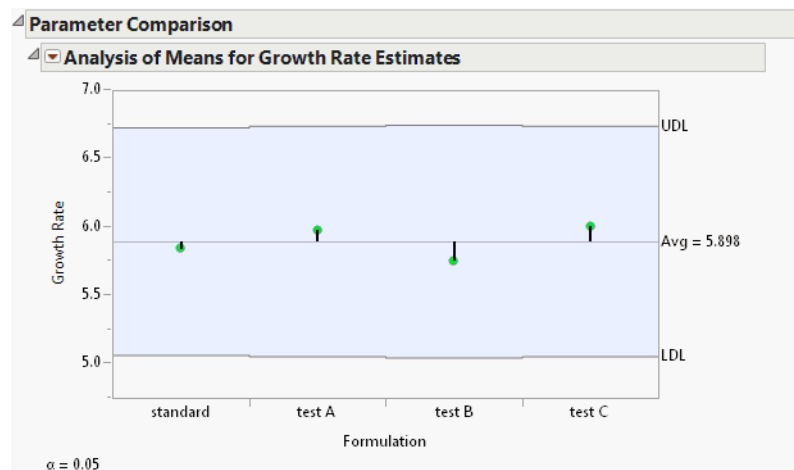


The Analysis of Means red triangle menu contains the following options:

**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 *Basic Analysis*.

# Equivalence Test

The Equivalence Test report gives an analysis for testing the equivalence of models across levels of the grouping variable (Figure 14.12). 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 (Figure 14.12), 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 14.12** Equivalence Test
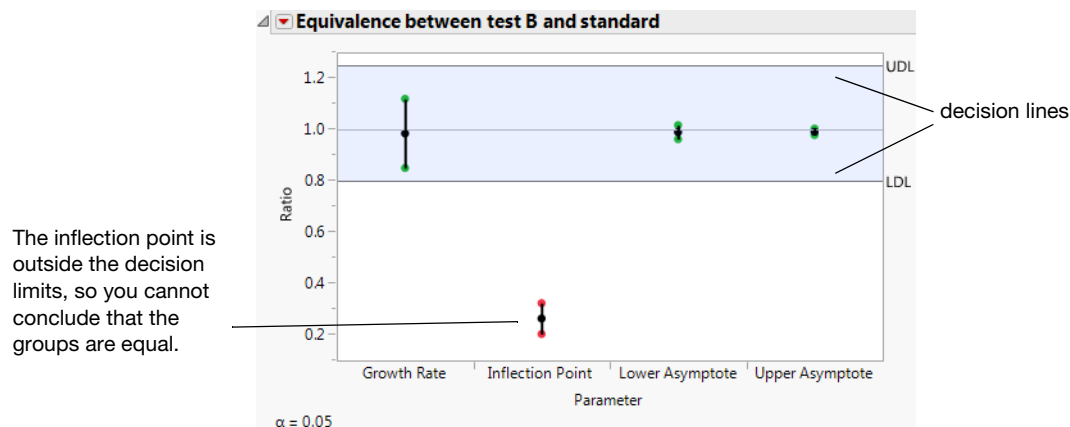


The Equivalence red triangle menu contains the following options:

**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**.

# Statistical Details for the Fit Curve Platform

## Model Formulas

Table 14.1 provides the formulas for the models on the Fit Curve red triangle menu.

**Table 14.1**  Fit Curve Model Formulas
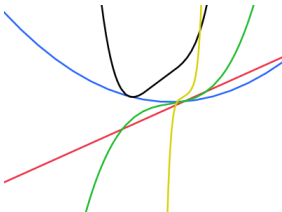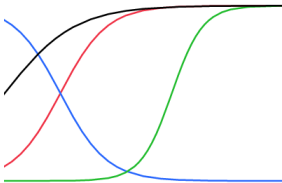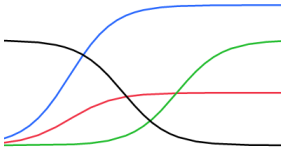
| Model | Formula |
|---|---|
| Polynomials  | $$\beta_0 + \sum_{i=1}^{k} \beta_i x^i$$ where $k$ is the order of the polynomial. These models can also be fit using the Fit Model and Fit Y by X platforms. |
| Logistic 2P  | $$\frac{1}{1 + \text{Exp}(-a(x-b))}$$ $a$ = Growth Rate $b$ = Inflection Point 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 |

**Table 14.1** Fit Curve Model Formulas *(Continued)*
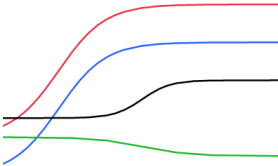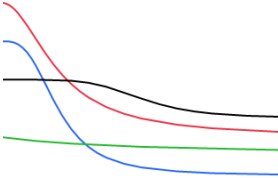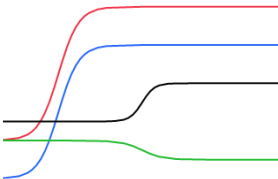
| Model | Formula |
|---|---|
| Logistic 4P | $$c + \frac{d - c}{1 + \text{Exp}(-a(x - b))}$$ $a$ = Growth Rate <br> $b$ = Inflection Point <br> $c$ = Lower Asymptote <br> $d$ = Upper Asymptote |
| Logistic 4P Rodbard | $$c + \frac{d - c}{1 + (x/b)^{a}}$$ $a$ = Growth Rate <br> $b$ = Inflection Point <br> $c$ = Lower Asymptote <br> $d$ = Upper Asymptote <br> Available only when the regressor values are positive. |
| Logistic 4P Hill | $$c + \frac{d - c}{1 + 10^{(-a(x - b))}}$$ $a$ = Growth Rate <br> $b$ = Inflection Point <br> $c$ = Lower Asymptote <br> $d$ = Upper Asymptote |

**Table 14.1** Fit Curve Model Formulas  *(Continued)*
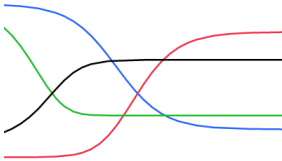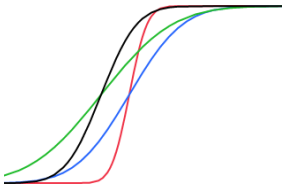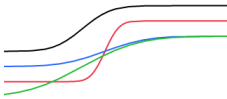
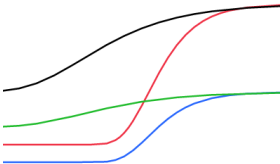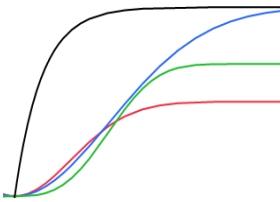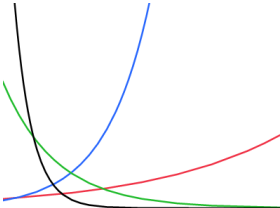| Model | Formula |
| --- | --- |
| Logistic 5P | $$c + \frac{d - c}{(1 + \text{Exp}(-a(x - b)))^f}$$ $a$ = Growth Rate<br>$b$ = Inflection Point<br>$c$ = Asymptote 1<br>$d$ = Asymptote 2<br>$f$ = Power |
| Probit 2P | $$\Phi\left(\frac{x - b}{a}\right)$$ $a$ = Growth Rate<br>$b$ = Inflection Point<br>$\Phi$ = Normal Distribution CDF<br>Available only when all response values are between zero and one. |
| Probit 4P | $$c + (d - c) \cdot \Phi\left(\frac{x - b}{a}\right)$$ $a$ = Growth Rate<br>$b$ = Inflection Point<br>$c$ = Asymptote 1<br>$d$ = Asymptote 2<br>$\Phi$ = Normal Distribution CDF |

**Table 14.1** Fit Curve Model Formulas  *(Continued)*
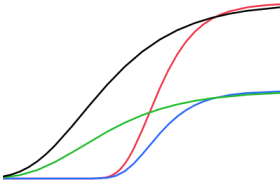
| Model | Formula |
| --- | --- |
| Gompertz 3P | $a\text{Exp}(-\text{Exp}(-b(x-c)))$ |

$a$ = Asymptote

$b$ = Growth Rate

$c$ = Inflection Point

| Model | Formula |
| --- | --- |
| Gompertz 4P | $a + (b-a)\text{Exp}(-\text{Exp}(-c(x-d)))$ |

$a$ = Lower Asymptote

$b$ = Upper Asymptote

$c$ = Growth Rate

$d$ = Inflection Point

| Model | Formula |
| --- | --- |
| Weibull Growth | $a(1 - \text{Exp}(-(x/c)^{b}))$ |

$a$ = Upper Asymptote

$b$ = Growth Rate

$c$ = Inflection Point

Available only when both the response values and regressor values are non-negative.

| Model | Formula |
| --- | --- |
| Exponential 2P | $a\text{Exp}(bx)$ |

$a$ = Scale

$b$ = Growth Rate

**Table 14.1** Fit Curve Model Formulas *(Continued)*
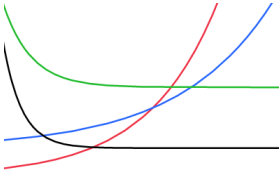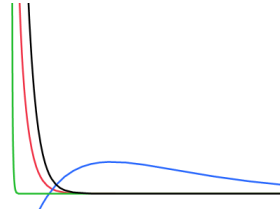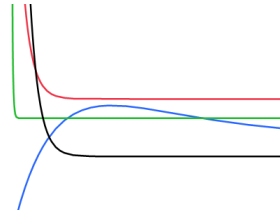
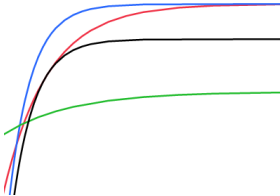| Model | Formula |
|---|---|
| Exponential 3P | $a + b\,\text{Exp}(cx)$ |
| | $a$ = Asymptote |
| | $b$ = Scale |
| | $c$ = Growth Rate |
| Biexponential 4P | $a\,\text{Exp}(-bx) + c\,\text{Exp}(-dx)$ |
| | $a$ = Scale 1 |
| | $b$ = Decay Rate 1 |
| | $c$ = Scale 2 |
| | $d$ = Decay Rate 2 |
| | Available 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 14.1**  Fit Curve Model Formulas  *(Continued)*
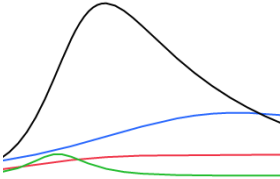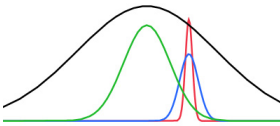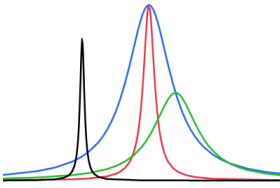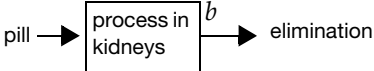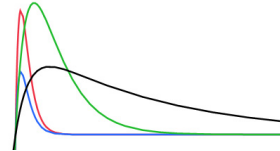
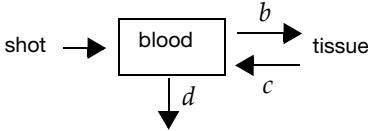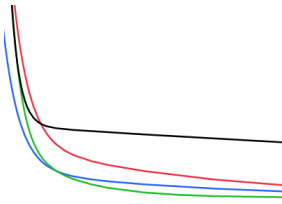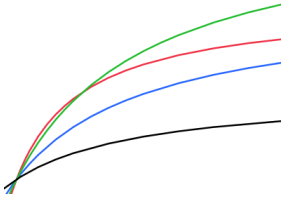| Model | Formula |
|---|---|
| Cell Growth 4P | $$\frac{ab}{(a-b)\mathrm{Exp}(-cx)+b\mathrm{Exp}(dx)}$$ |

$a$ = Peak value if mortality rate, $d$, is zero

$b$ = Response at time zero

$c$ = Cell Division Rate

$d$ = Cell Mortality Rate

Available only when the response values are positive and the regressor values are non-negative.

| 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 | $$\frac{abc}{c-b}(\mathrm{Exp}(-bx)-\mathrm{Exp}(-cx))$$ |
|---|---|

pill → | process in kidneys | $b$ → elimination

$a$ = Area Under Curve

$b$ = Elimination Rate

$c$ = Absorption Rate

Available only when the response values and the regressor values are all positive.

**Table 14.1** 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. |

# 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 15.1** Example of a Custom Nonlinear Fit
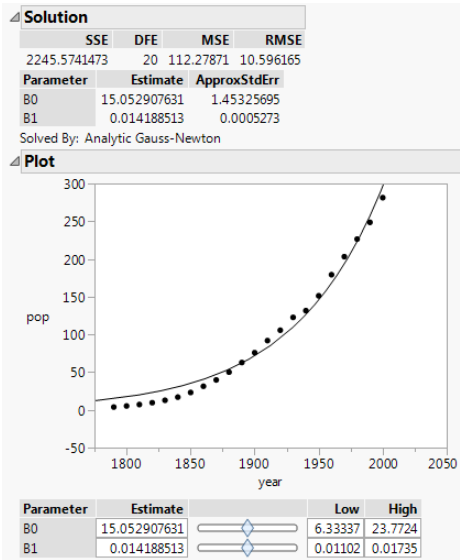


The Nonlinear platform also provides predefined models, such as polynomial, logistic, Gompertz, exponential, peak, and pharmacokinetic models. See "Fit Curve".

---

**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 *Fitting Linear Models* and *Basic Analysis*.

---

# Contents

# Example of Fitting a Custom Model

This example fits a custom model by creating a formula column for the model and then fitting the model using the Nonlinear platform. This method requires a few more steps than fitting a built-in model, but it allows any nonlinear model to be fit. You can also provide a custom loss function, and specify several other options for the fitting process.

**Tip:** If the custom model is one that you use often for several data tables, you can add the model to the Model Library instead. See "Customize the Nonlinear Model Library".

The data are 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.

### Create a Formula Column

To fit a custom model, you must first create a model column with initial parameter estimates.

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. Select **Parameters** from the list below the list of columns.
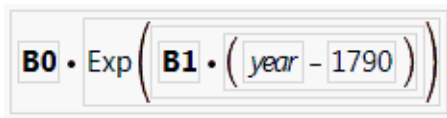
**Figure 15.2** Select Parameters
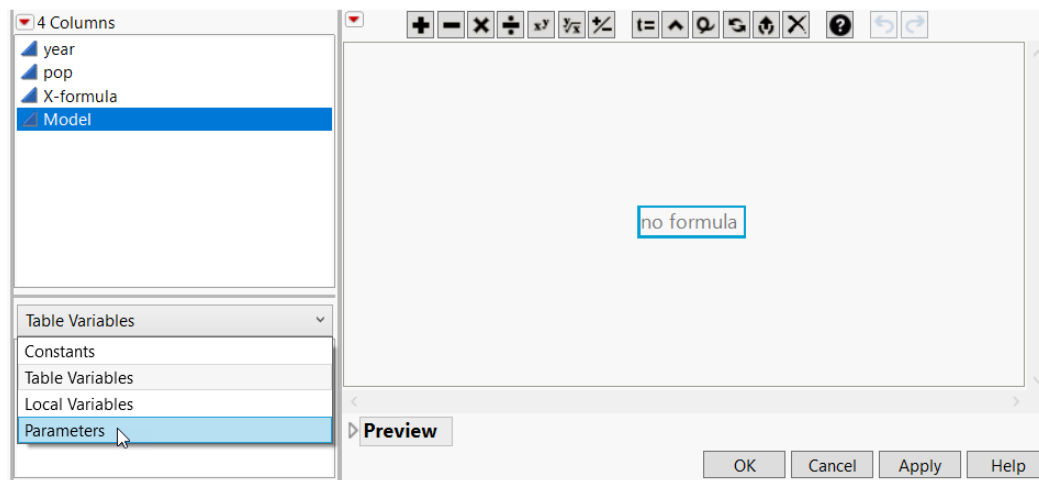


5. Select **New Parameter**.
6. Type B0 for Name.

7. Type 3.9 for Value. This is the initial estimate of the parameter.

8. Click **OK**.

9. Select **New Parameter**.

10. Type B1 for Name and enter 0.022 for Value.

11. Click **OK**.

12. Enter the model formula using the Formula Editor functions, the column year, and the parameters.

**Figure 15.3**  Completed Model Formula



---

**Tip:** Click the gray triangle next to Transcendental to find the Exp command.

---

13. Click **OK**.

## Fit a Nonlinear Model

1. Select **Analyze > Specialized Modeling > Nonlinear**.

2. Assign Model to the **X, Predictor Formula** role.

3. Assign pop to the **Y, Response** role.

4. Click **OK**.

5. Click **Go** on the Control Panel to fit the model.

**Figure 15.4** Plot and Solution Report
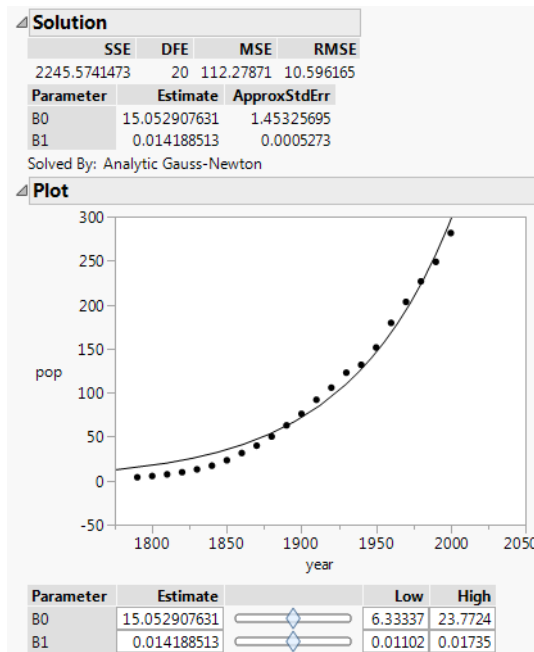


The final parameter estimates are shown in the **Solution** report, along with other fit statistics. The fitted model is shown on the plot.

## 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**.

**Figure 15.5** Nonlinear Platform Launch Window
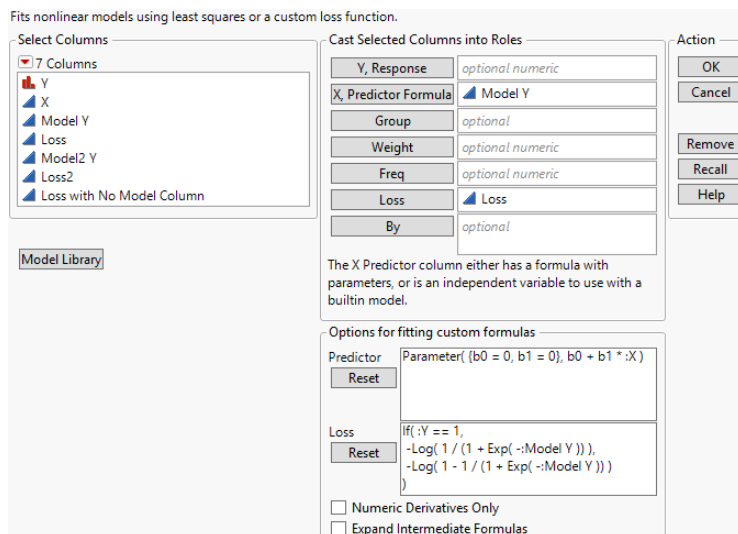


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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.

> **Note:** If you select a column that does not contain a model formula with parameters and you also do not specify a custom formula, the Fit Curve platform is launched instead.

> **Note:** If you select a formula column that contains a parameter list item that does not have an assignment, JMP automatically sets that parameter to zero and displays a message in the log.

**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.

**Note:** If the formula does not have separate parameters for each group level, the Group variable is ignored by JMP.

**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".

**Options for fitting custom formulas**    If you specify a column in the X, Predictor Formula role that does not contain a model formula with parameters, you can use this option to create a custom formula. In the Predictor field, specify parameters, parameter values, and a custom formula that uses the selected *X* variable. Then, click the **Reset** button and launch the platform.

**Numeric Derivatives Only**    Uses 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 15.6.

**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 15.6**  Initial Nonlinear Fit Report
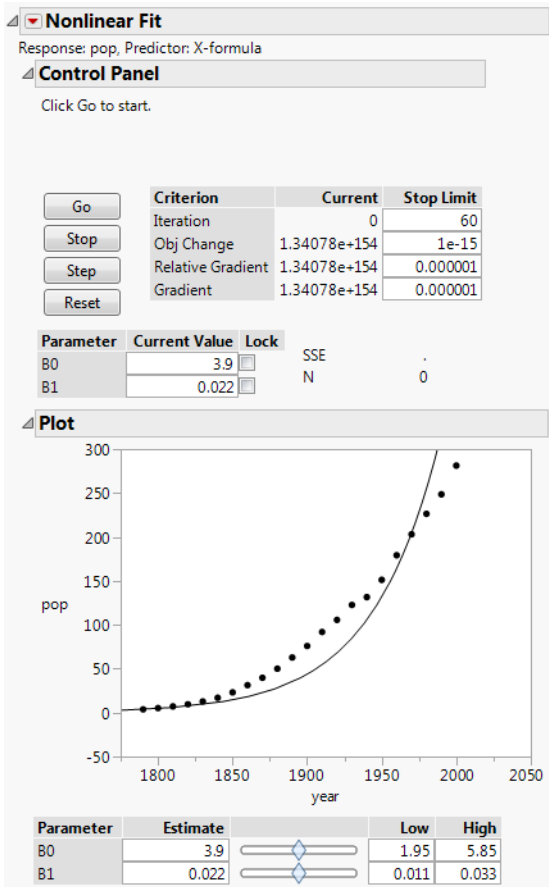


After you click **Go** to fit a model, the report includes the following additional items, shown in Figure 15.7.

**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 more information about the **Goal SSE for CL**, see "Profile Likelihood Confidence Limits".

**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.

> **RSquare**   (Available only if the model has an intercept and the sum of the residuals is close to zero.) Shows the RSquare value.

> **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.

> > **Note:** If the number of observations is less than or equal to the number of parameters in the fitted model, ApproxStdErr is not reported.

> **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 information about the confidence intervals, see "Profile Likelihood Confidence Limits".

> **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.

**Covariance of Estimates**   Displays the covariances between the parameter estimates.

**Figure 15.7** Fitted Model Report
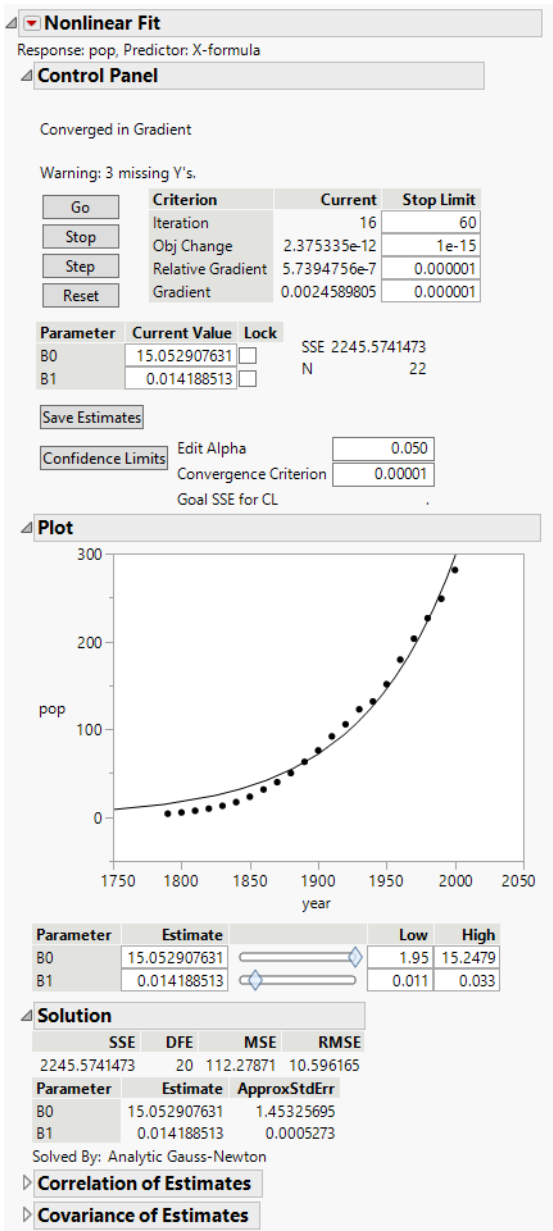
# Nonlinear Platform Options

The Nonlinear Fit red triangle menu contains 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", 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 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.

**Show Prediction Expression**   Shows the prediction model or the loss function at the top of the report.

**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.

**Save Estimates to Table**   Saves the parameter estimates to a new data table.

# 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 15.8** Nonlinear Model Library Dialog
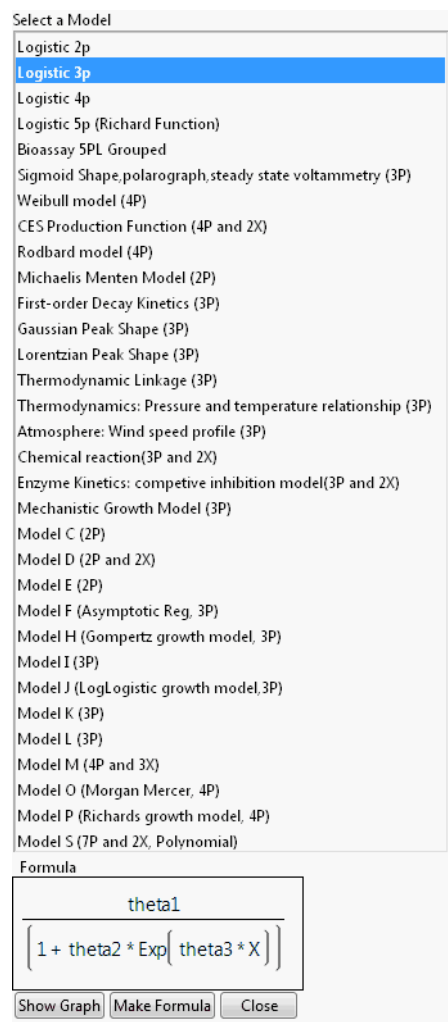


Click **Show Graph** to show a 2-D theoretical curve for one-parameter models and a 3-D surface plot for two-parameter models. No graph is available for models with more than two explanatory (*X*) variables. On the graph window, change the default initial values of parameters using the slider, or clicking and entering values in directly.

**Figure 15.9** Example Graph in Model Library
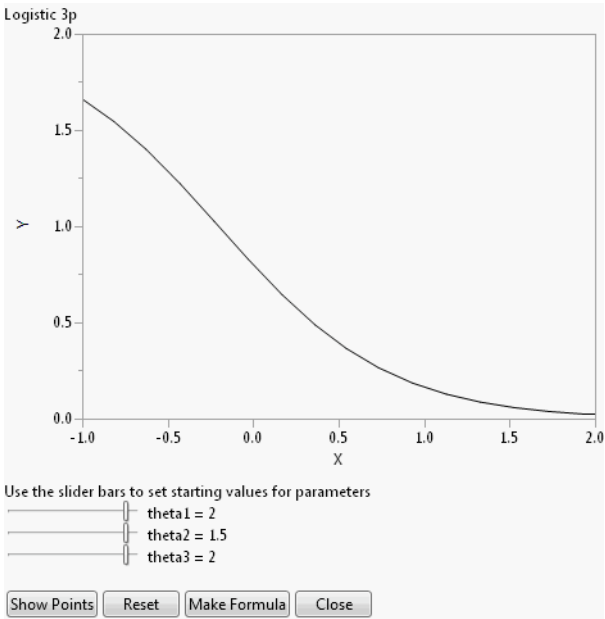


The **Reset** button sets the initial values of parameters back to their default values.

Click **Show Points** to overlay the actual data points to the plot. The dialog in Figure 15.10 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 15.10** Select Roles



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 15.11.
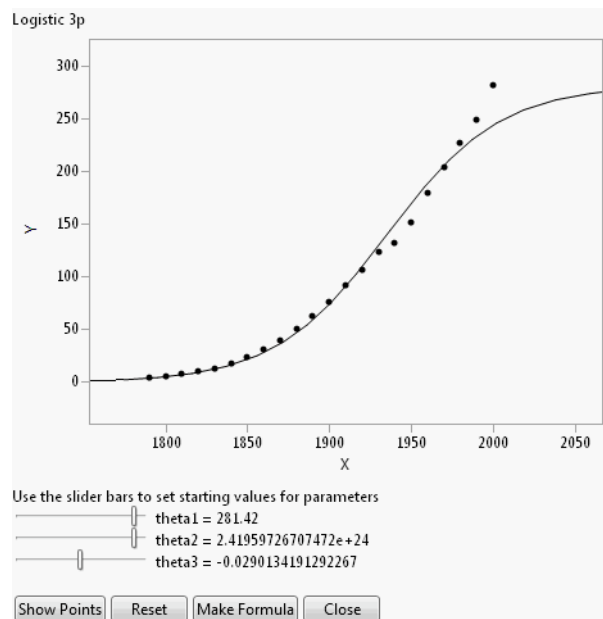
**Figure 15.11** Show Points



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

**Note:** If you click **Make Formula** before using the **Show Graph** or **Show Points** buttons, you are asked to provide the *X* and *Y* roles, and an optional Group role (Figure 15.10). 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 (macOS). You can customize the nonlinear library script by modifying this script.

**Note:** You might not be able to directly edit the file. If this is the case, open the script, make the desired edits, and save to a different location on your computer. Use the same filename. Then, delete the old NonlinLib.jsl script in the Resources/Builtins folder and move the new NonlinLib.jsl file into the Resources/Builtins folder.

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

- "Example of Maximum Likelihood: Logistic Regression"
- "Example of a Probit Model with Binomial Errors: Numerical Derivatives"
- "Example of a Poisson Loss Function"
- "Example of Setting Parameter Limits"

# 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 15.12** Nonlinear Launch Window
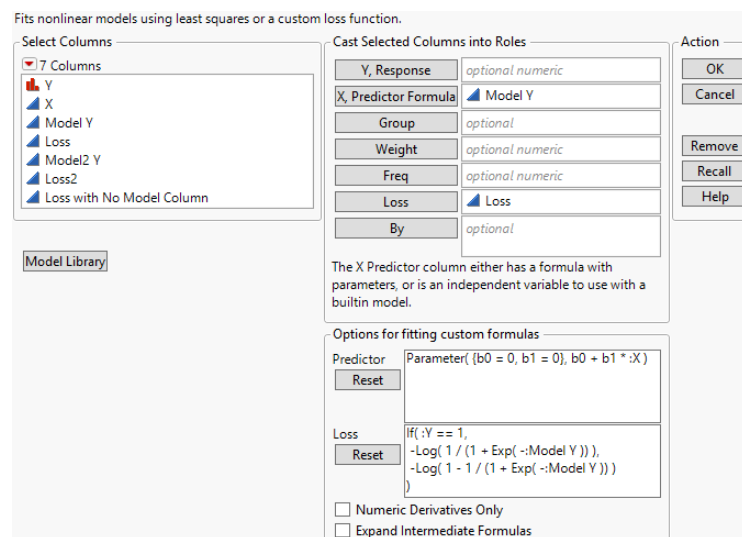


5. Click **OK**.

   The Nonlinear Fit Control Panel appears.

**Figure 15.13** Nonlinear Fit Control Panel
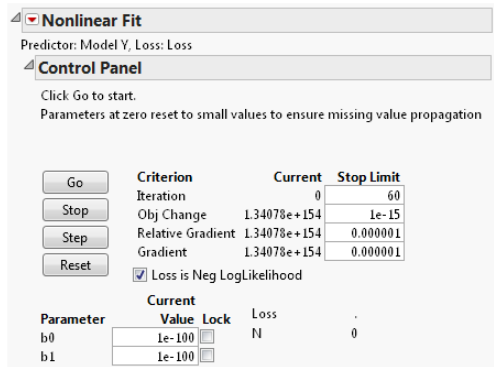


6.  Click **Go**.

    The parameter estimates are shown in the Solution report.

**Figure 15.14**  Solution Report
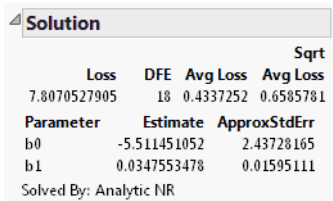


The Loss value in the Solution report is the negative log-likelihood evaluated at the parameter estimates.

# 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 **Help > Sample Data Library** and open Ingots2.jmp.
2.  Select **Analyze > Specialized Modeling > Nonlinear**.
3.  Assign P to the **X, Predictor Formula** role,
4.  Assign Loss to the **Loss** role.
5.  Select the **Numeric Derivatives Only** option.
6.  Click **OK**.
7.  Click **Go**.

The platform used the Numerical SR1 method to obtain the parameter estimates shown in Figure 15.15.

**Figure 15.15** 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.

$$P(Y = n) = \frac{e^{-\mu}\mu^{n}}{n!}, \ n = 0, 1, 2, \dots$$

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 **Help > Sample Data Library** and open Ship Damage.jmp.
2.  Select **Analyze > Specialized Modeling > Nonlinear**.
3.  Assign model to the **X, Predictor Formula** role.
4.  Assign Poisson to the **Loss** role.
5.  Click **OK**.
6.  Set the **Current Value** (initial value) for b0 to 1, and the other parameters to 0.

**Figure 15.16**  Enter New Parameters
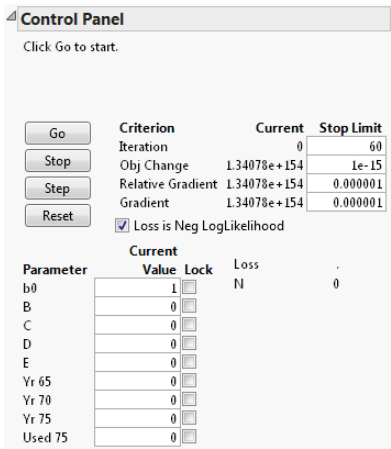


7.  Click **Go**.
8.  Click the **Confidence Limits** button.

The Solution report appears. The results include the parameter estimates and confidence intervals, and other summary statistics.

**Figure 15.17**  Solution Table for the Poisson Loss Example

| | 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 Platform" to fit the model. This example shows how to save the prediction formula from Fit Curve and then set parameter limits in Nonlinear.

1.  Click the Logistic 4P red triangle and 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. 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 15.18**  Nonlinear Fit Control Panel
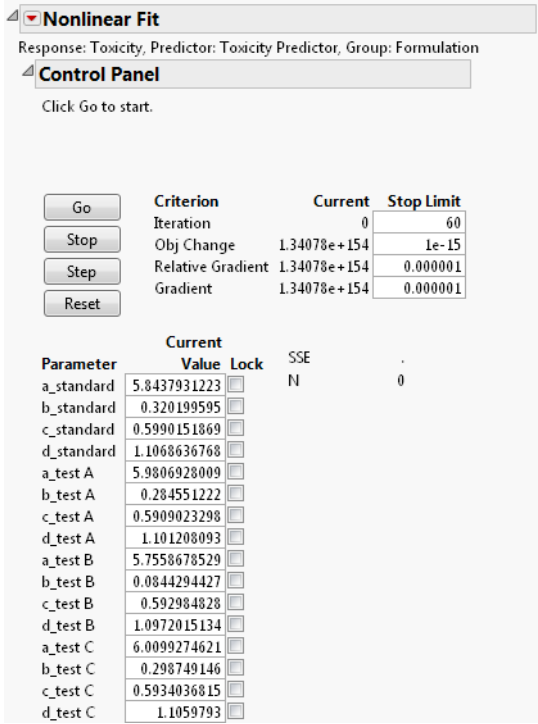


**Tip:** You can lock parameters if you know the values from prior information.

7.  Click the Nonlinear Fit red triangle and 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 15.19. You know from prior experience that the maximum toxicity of the drug is at least 1.1.

**Figure 15.19** Setting Parameter Bounds



9.  Click **Go**.

    The final parameter estimates are shown in the **Solution** report, along with other fit
    statistics. The fitted model is shown on the plot.

**Figure 15.20** Nonlinear Fit Plot and Parameter Estimates

| Parameter | Estimate | | Low | High |
|---|---|---|---|---|
| a_standard | 5.843793154 | | 2.9219 | 8.76569 |
| b_standard | 0.3201995954 | | 0.1601 | 0.4803 |
| c_standard | 0.5990151873 | | 0.29951 | 0.89852 |
| d_standard | 1.1068636765 | | 0.55343 | 1.6603 |
| a_test A | 5.9806928191 | | 2.99035 | 8.97104 |
| b_test A | 0.2845512223 | | 0.14228 | 0.42683 |
| c_test A | 0.5909023301 | | 0.29545 | 0.88635 |
| d_test A | 1.1012080929 | | 0.5506 | 1.65181 |
| a_test B | 5.6527018259 | | 2.87793 | 8.6338 |
| b_test B | 0.0860027315 | | 0.04221 | 0.12664 |
| c_test B | 0.5917244683 | | 0.29649 | 0.88948 |
| d_test B | 1.1 | | 0.5486 | 1.6458 |
| a_test C | 6.0099274821 | | 3.00496 | 9.01489 |
| b_test C | 0.2987491449 | | 0.14937 | 0.44812 |
| c_test C | 0.5934036814 | | 0.2967 | 0.89011 |
| d_test C | 1.1059792995 | | 0.55299 | 1.65897 |

Options below the plot allow for adjusting parameter limits and estimates.

# Statistical Details for the Nonlinear Platform

- "Profile Likelihood Confidence Limits"
- "How Custom Loss Functions Work"
- "Notes Concerning Derivatives"
- "Notes on Effective Nonlinear Modeling"

# 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 the *SAS/ETS User's Guide* (SAS Institute Inc. 2020).

Figure 15.21 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 15.21**  Diagram of Confidence Limits for Parameters
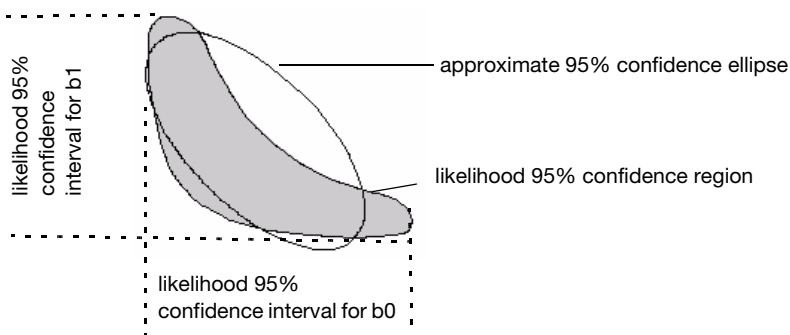


Likelihood confidence intervals are more trustworthy than confidence intervals calculated from approximate standard errors. If a particular limit cannot be found, computations begin for the next limit. When you have difficulty obtaining convergence, try the following:

- 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 the following:

$$a_1 + b_1 x + c_1 x^2$$

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.

$$a_1 = a_2 - b_2\bar{x} + c_2\bar{x}^2$$
$$b_1 = b_2 - 2c_2\bar{x}$$
$$c_1 = c_2$$

If the number of iterations still goes to the maximum, increase the maximum number of iterations 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.

**JMP PRO Functional Data Explorer**

**Perform Functional Data Analysis**

*The Functional Data Explorer platform is available only in JMP Pro.*

The Functional Data Explorer (FDE) platform is designed for data that are functions, signals, or series. It can be used as an exploratory data analysis tool or as a dimension-reduction technique. In the case of dimension reduction, the FDE platform converts functional data into a form that can be analyzed in another JMP platform.

Data preprocessing tools are available in the FDE platform, including several types of transformations for output data and alignments for input data. A functional model is created by fitting a B-spline, P-spline, or Fourier basis model to the data. When a model is fit, functional principal components analysis (functional PCA) is automatically performed on the functional model. Results from the functional PCA, such as the functional principal component scores, are saved to a separate data table for feature extraction or dimension reduction.

**Figure 16.1** B-Spline Model Selection in FDE
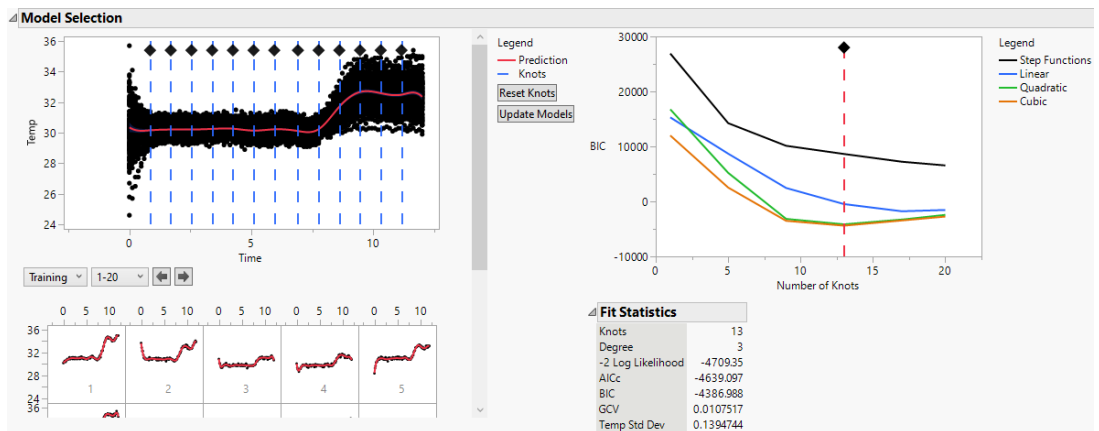
# Contents

# JMP PRO  Overview of the Functional Data Explorer Platform

Functional data can be defined as data that are recorded over a continuous domain, where a set of measurements form a curve or image. Often, the domain is time and the sets of measurements are defined by an ID variable. A functional data observation for ID level $i$ at a specific point $t$ on the domain is written as $f_i(t)$. The Functional Data Explorer platform enables you to explore and analyze functional data.

The form of functional data can be dense or sparse. Dense functional data occur when observations are on the same equally spaced grid of points for all levels of the ID variable. Sparse functional data occur when ID levels have different numbers of observations that are unequally spaced across the domain. The Functional Data Explorer platform can handle both forms of functional data.

Although functional data can be expressed in many ways, it can generally be classified into the following two cases:

- The response of interest, $f(t)$, has a functional form.

- There are one or more covariates, $f(t)$'s, that have functional forms. These are sometimes referred to as functional or signal processes.

The Functional Data Explorer platform is useful as an exploratory tool for any type of functional data. However, the strength of the platform is taking many functional processes (that might be associated with a scalar response) and extracting key features to use in further modeling. This can be done by first fitting a functional model to the data using a B-spline, P-spline, or Fourier basis model. Then, a functional principal components analysis (functional PCA) is performed on the functional model. Results from the functional PCA, such as the functional principal component (FPC) scores, are saved and used for feature extraction and analysis in another modeling platform, such as the Generalized Regression personality of the Fit Model platform. Alternatively, you can specify a set of supplementary variables and fit a generalized regression model within the FDE platform to determine how these variables affect the response.

For more information about functional data analysis, see Ramsay and Silverman (2005).

# JMP PRO  Example of Functional Data Explorer

This example analyzes weekly weather data collected from 16 weather stations across the United States. Run the Weather Station Locations script in the data table to view a map of the locations. Daily temperatures are summarized as weekly averages. Not every weather station has a weekly temperature measurement for every week of the year. This is an example of sparse functional data.

1. Select **Help > Sample Data Library** and open Functional Data/Weekly Weather Data.jmp.

2. Select **Analyze > Specialized Modeling > Functional Data Explorer**.

3. Select TMAX and click **Y, Output**.

4. Select Week of Year and click **X, Input**.

5. Select ID and click **ID, Function**.

6. Click **OK**.

**Figure 16.2** Initial Functional Data Explorer Report
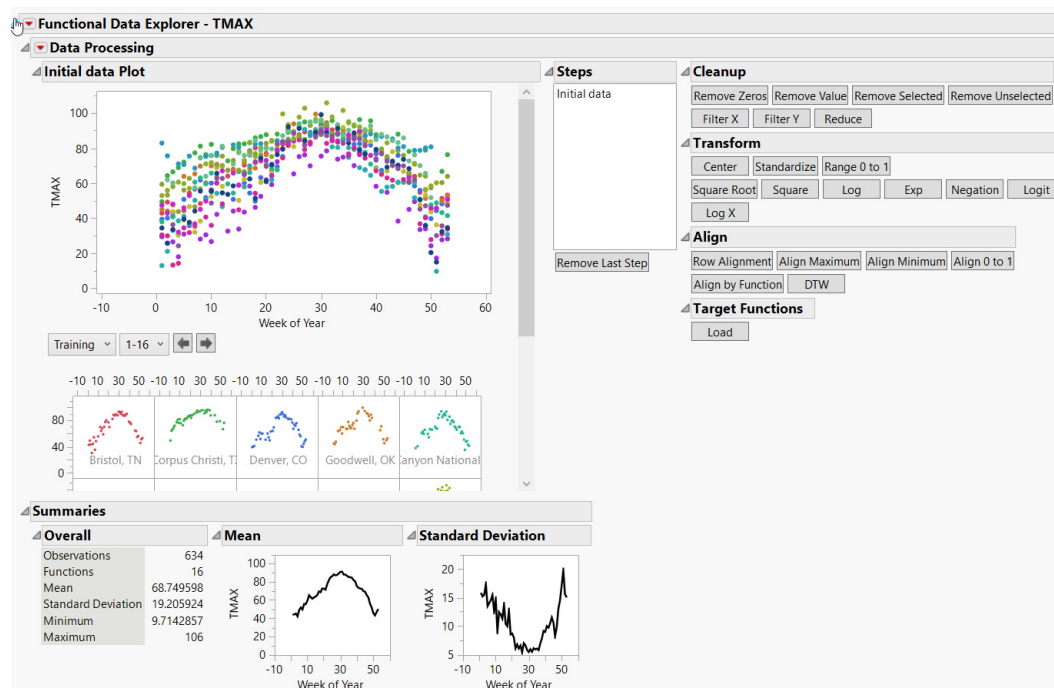


The initial Functional Data Explorer report contains plots of the raw data, summary statistics, and summary plots for the functional mean and functional standard deviation of the data. There are also buttons for data processing options. Data processing options are

also accessible from the Data Processing red triangle menu. Prior to modeling, it is often a good idea to standardize your output data.

7.  Click the **Standardize** button under the Transform menu.

    The data plots and summary statistics are updated based on the specified transformation. Standardized is added to the Steps list.

8.  Click the Functional Data Explorer red triangle and select **Models > Fourier Basis**.

**Figure 16.3**  Fourier Basis Model Report
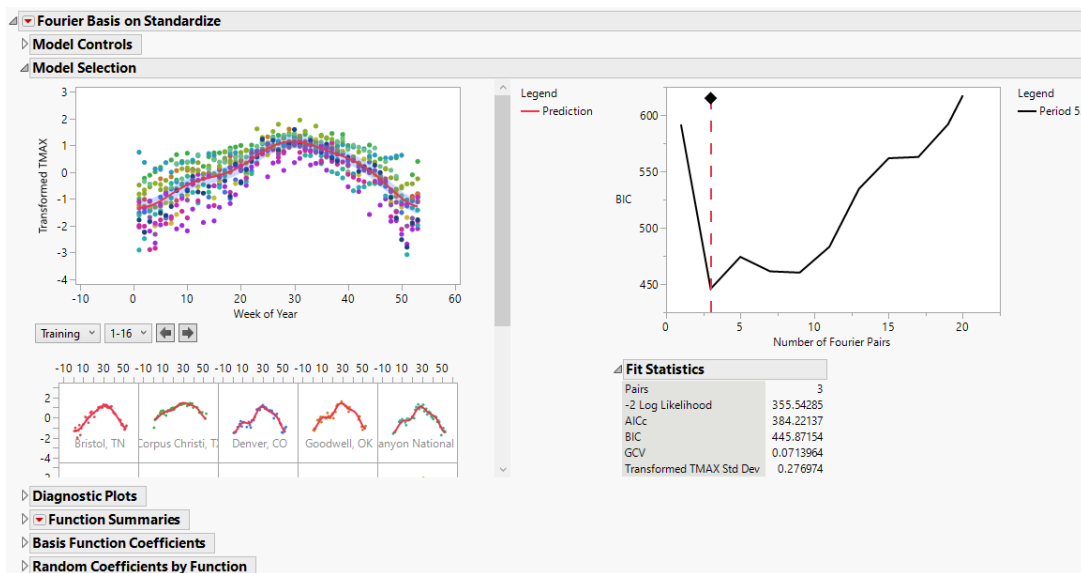


The Fourier Basis report includes several reports that contain information about the selected model. In the Model Selection report, the displayed model is the best fitting model according to the BIC fit criterion. For the weather data, the Fourier Basis model that is chosen has a period of 53 and three basis function pairs. Fit statistics and coefficients are also available for the model. Scroll down to view the Functional PCA report.

**Figure 16.4** Functional PCA Report
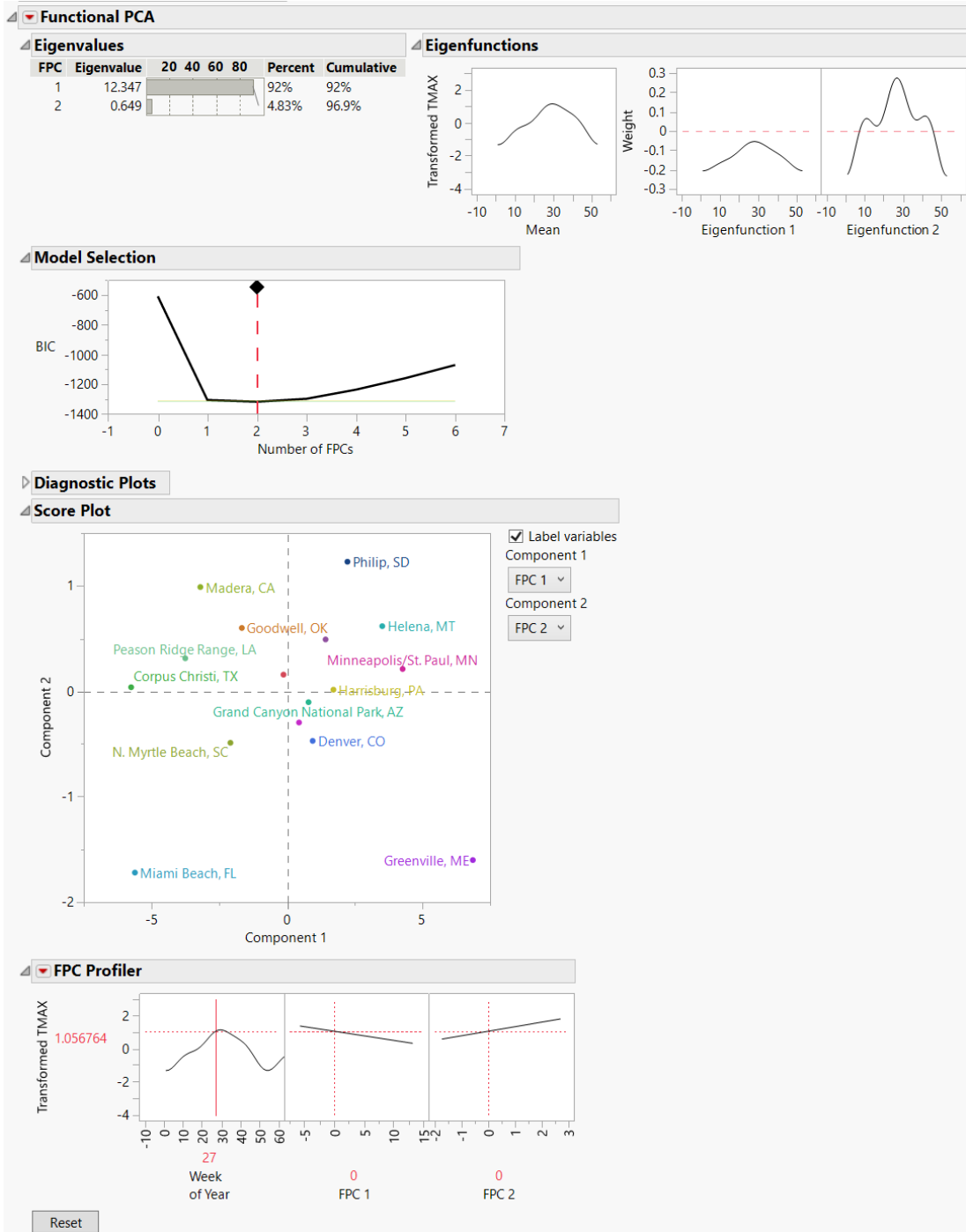
The Functional PCA report shows that the first two eigenvalues explain nearly 97% of the variation in the data. In the Model Selection graph, click and drag the red dashed line to 3 FPCs to see that the first three eigenvalues explain 99% of the variation in the data. However, the first eigenvalue alone explains 92%. You can use the Score Plot to detect individual functions that are outliers from the other functions. In the Score Plot, most of the locations are clustered together except for the Miami Beach, FL and Greenville, ME locations. Scroll up to the individual function plots. The function for the Miami Beach location is flatter, indicating less temperature variability than the rest of the locations. The function for the Greenville location has a lower maximum, indicating consistently colder temperatures than the rest of the locations.

**Tip:** Deselect the Label variables option in the Score Plot report to better identify outliers.

# Launch the Functional Data Explorer Platform

Launch the Functional Data Explorer platform by selecting **Analyze > Specialized Modeling > Functional Data Explorer**.

**Figure 16.5**  Functional Data Explorer Launch Window
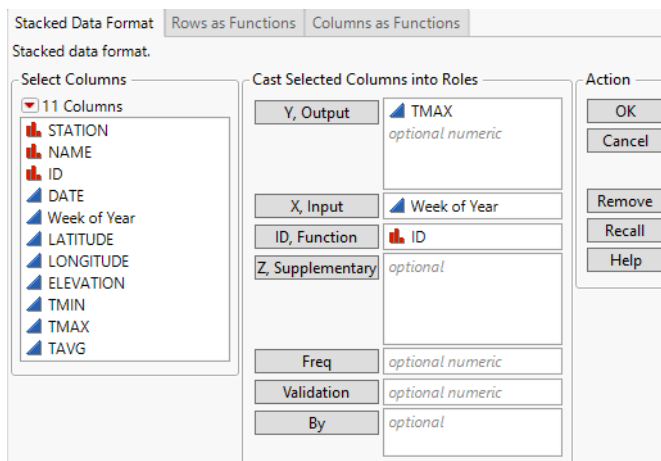


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The Functional Data Explorer launch window includes tabs for different types of data formats. Select the tab based on your data format.

**Stacked Data Format**   Select for data tables where each row corresponds to a single observation. There are separate columns for the output, input, and ID variables.

**Note:** The Stacked Data Format is the only data format that enables you to specify multiple functional processes. If you assign more than one column to **Y, Output** in the Stacked Data Format tab, each Y variable is analyzed separately. A Fit Group report contains the individual reports for the Y variables.

**Rows as Functions**   Select for data tables where each row corresponds to the full output function for one level of the ID variable. Each column is a level of the input variable.

**Caution:** The Rows as Functions format assumes that your observations are equally spaced in the input domain unless the FDE X column property is used. The FDE X column property enables this data format to use input variables specified in the column names.

**Columns as Functions**   Select for data tables where each column corresponds to the entire output function for one level of the ID variable. Each row corresponds to a level of the input variable.

## Launch Window Options

**Y, Output**   Assigns the functional process, $f(t)$. There must be at least two observed output values for each level of the ID variable.

**Note:** Functions with fewer than two observed output values are removed from the analysis.

**X, Input**   (Available for Stacked Data Format and Columns as Functions.) Assigns the input variable $t$. If no variable is specified for X, Input, the row number is used. Using the row number assumes that the observations are equally spaced in the input domain.

**ID, Function**   (Available for Stacked Data Format and Rows as Functions.) Assigns the ID variable to each function. For Stacked Data, if no ID variable is assigned all observations are assumed to come from one function.

**Z, Supplementary**   (Available for Stacked Data Format and Rows as Functions.) Assigns one or more supplementary variables. Supplementary variables are not used in any of the calculations in the Functional Data Explorer platform and including them does not affect the results. Supplementary variables are variables you might want to use in future analyses of the results from Functional Data Explorer. When you specify supplementary variables, they are included in a Supplementary column group in the tables that are created by the Save Data and Save Summaries options. These columns retain any column properties that were specified in the original data table. A Functional DOE Analysis option is also made available for the fitted models. See "Functional DOE Analysis".

**Freq**   (Available only for Stacked Data Format.) Assigns a column whose numeric values represent a frequency for each row in the analysis. The effect of a frequency column is to expand the data table, so that any row with integer frequency $k$ is expanded to $k$ identical rows.

**Validation**   Assigns an optional numeric column that defines the validation sets. This column should contain only two distinct values. The smaller value defines the training set and the larger value defines the validation set. If there are more than two values, the smallest value defines the training set and all other values define the validation set. The FDE platform uses the validation column to train and evaluate the model. For more information about validation, see "Validation in JMP Modeling".

**Note:** The Validation option enables you to hold out complete functions, not a sample of observations from each function. Therefore, all observations that have the same ID value must be classified as either test or validation. You cannot have observations with the same ID value in both sets. For more information about this type of validation column, see "Grouped Validation Column".

**Caution:** If you click the Validation button with no columns selected in the Select Columns list, and add a validation column to your data table it will not be a grouped validation column. Instead, use the Make Validation Column Platform, see "Grouped Validation Column".

**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.

# The Functional Data Explorer Report

The initial Functional Data Explorer report contains a Data Processing report and a Summaries report.

**Figure 16.6** Initial Functional Data Explorer Report
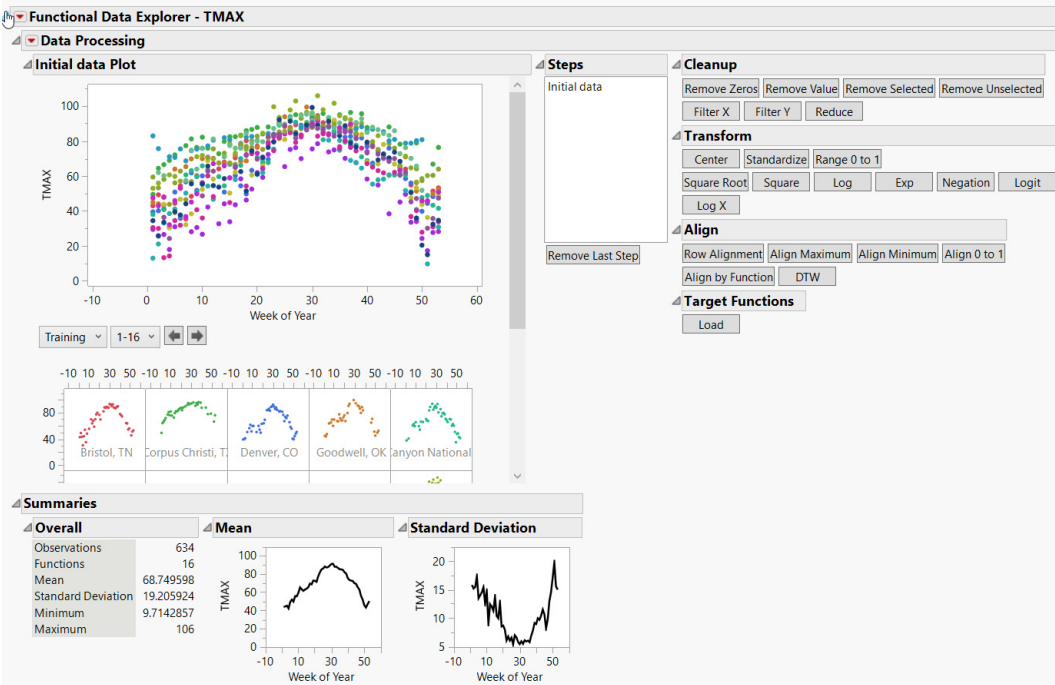


The Data Processing Report includes an initial data plot of all observations, as well as a grid of individual data plots that correspond to levels of the ID variable. The maximum number of individual plots shown in the grid is twenty. There are drop-down menus and arrows above the grid to interactively select which individual plots are shown. If you specified a validation set, the first drop-down menu enables you to select if the individual plots shown are from the training set or the validation set. If there are more than twenty functions, the second drop-down menu and the arrows enable you to view different groups of plots. All graphs plot the functional response data over the values of the input variable. The Data Processing Report also contains data processing buttons.

The Summaries report contains a table of overall summary statistics, including the number of observations, number of functions, and the overall mean, standard deviation, minimum, and maximum values. There are also plots of the functional mean and functional standard deviation. The functional summary statistics displayed in the plots are computed at each unique value of the input variable.

The Data Processing and Summaries reports are useful for preprocessing your data prior to fitting a model or performing direct functional principal components analysis. There are data cleaning, transformation, and alignment buttons available in the Data Processing Report that correspond to the options in the Data Processing red triangle menu. See "Data Processing Report Options". Each time you perform a preprocessing step, the graphs and summary results are automatically updated. You can perform multiple preprocessing steps in sequence. Each step is added to the Steps list of the Data Processing report. To remove the most recent step, click the **Remove Last Step** button below the Steps outline.

## Model Reports

Use the options in the Models submenu in the Functional Data Explorer red triangle menu to fit models to your data. See "Models" for the available models. Each time you fit a different type of model to the data, a model report appears. Each model report contains the following reports:

- "Model Controls"
- "Model Selection"
- "Diagnostic Plots"
- "Function Summaries"
- "Basis Function Coefficients"
- "Random Coefficients by Function"
- "Functional PCA"

## Model Controls

The Model Controls report enables you to define parameters of models to compare in the Model Selection report. The appearance of the Model Controls report depends on the type of model that is fit.

### B-Spline and P-Spline Model Controls

When a B-Spline or P-Spline model is fit, you can specify the following parameters:

**Number of Knots**    Add, remove, or specify a range for the number of knots in each spline. The knots must be non-zero integers.

**Note:** The maximum number of knots allowed for B-Spline models is one less than the maximum number of observations per function or the number of unique inputs. The maximum number of knots allowed for P-Spline models is two less than the number of unique inputs. If you specify a number larger than the maximum, a warning message appears.

**Spline Degree**    Add or remove spline degree fits from the Model Selection report.

### Fourier Basis Model Controls

When a Fourier Basis model is fit, you can specify the following parameters:

**Number of Fourier Pairs**    Add, remove, or specify a range for the number of Fourier pairs to compare.

**Period**    Change the period of the function.

After you specify the model controls, click **Go** to view the updated models in the Model Selection report.

**Tip:** To specify the Model Controls prior to fitting a model, press Shift, click the Functional Data Explorer red triangle, and select the desired model. See "Models".

## Model Selection

The Model Selection report contains an overall prediction plot, a grid of individual prediction plots, a solution path plot, and a table of fit statistics. The grid of individual prediction plots has the same layout and controls as the grid of individual plots in the Data Processing report. At most, there are twenty plots shown at a time. There are drop-down menus and arrows that enable you to view different groups of individual prediction plots. The solution path plot shows a model selection criterion plotted over values of a model parameter. The Bayesian Information Criterion (BIC) is the default fitting criterion. See "Model Report Options". For B-Spline and P-Spline models, there is a separate solution path for each spline degree plotted across the defined number of knots. For Fourier Basis models, the solution path is plotted across the number of Fourier pairs for a defined period. Use the Model Selection option to change the solution path plot parameters.

The current solution is designated by the dotted vertical line in the solution path plot. By default, the slider is placed at the number of knots or Fourier pairs that corresponds to the model that has the smallest model selection criterion value. You can drag the slider at the top of the dotted vertical line to change the number of knots or Fourier pairs in the current model. Dragging the slider automatically updates the prediction plots in the Model Selection report, as well as the information in all other reports.

The Fit Statistics table contains a description of the current solution model. It also displays the -2 Log Likelihood, the values for the AICc, BIC, and GCV model fitting criterion, and a value for the response standard deviation, denoted as <Y, Output> Std Dev. The response standard deviation is defined as the residual sigma from the fitted model. When a P-Spline model is selected, the penalty parameter λ (Lambda) is also displayed.

The prediction plots show the raw data and prediction curves that correspond to the current model. If there is a validation set, the predicted curves are not shown for functions that are in the validation set. For spline models, the default model selected is the degree of spline with the best fit. Click a specific spline in the solution path plot or the legend to change the current model selection. The curve in the overall prediction plot is a prediction of the mean curve. The curves in the individual prediction plots are prediction curves for each specific function. For B-Spline models, the overall prediction plot also displays the location of the knots. You can change the location of the knots by dragging the blue slider bars to different locations. To update the model reports according to the new knot locations, click the **Update Models** button. To reset the knots to their default locations, click the **Reset Knots** button.

## Diagnostic Plots

The Diagnostic Plots report contains the Actual by Predicted plot and the Residual by Predicted plot. These plots help assess how well the current model fits the data. The Diagnostic Plots report is closed by default.

## Function Summaries

Displays summaries from the Functional PCA for each level of the ID variable. The functional principal components associated with eigenvalues that explain more than 1% variation in the data are displayed by default. The mean, standard deviation, median, minimum, maximum, integrated difference, root integrated square error (RISE), and root integrated function square (RIFS) are also shown. The integrated difference and RISE summary values are used to determine how much the ID specific function differs from the overall mean function. The RIFS summary value is used for optimal curve fitting. See "Function Summaries Details". The Function Summaries red triangle menu contains the following options:

**Customize Function Summaries**   Displays a window that enables you to select the number of FPCs and the summary statistics that are shown in the Function Summaries report. If the number of FPCs to be shown is specified, the Functional PCA report is also updated. There is also a checkbox, Save Graphs, that determines whether a graph for each function is included in the data table produced by the Save Summaries option.

---

**Tip:** If you have multiple functional processes, you can customize all Function Summaries reports to show the same summary values by clicking Ctrl and selecting Customize Function Summaries.

---

**Save Summaries**   Saves the summary statistics specified in the Function Summaries report to a new data table. The name of the new data table describes the model fit. This data table contains formula columns for the eigenfunctions, mean function, prediction function, and conditional prediction function. There is also a column that contains the image of a graph of the raw data and the specified model fit for each function. In the data table, there is a profiler script that launches the prediction profilers for the prediction and conditional prediction formulas. These formulas are functions of the input variable, the ID variable, and the eigenfunctions.

## Basis Function Coefficients

Displays the estimated basis function coefficients and their standard deviations. These are common across all levels of the ID variable and are fixed estimates in the mixed model framework. To view standard errors and confidence intervals for the coefficients, right-click in the table and select **Columns**.

## Random Coefficients by Function

Displays the estimated random coefficients for each basis function and functional process combination. These are unique to each level of the ID variable and are random effects estimates in the mixed model framework.

## Functional PCA

Functional principal components analysis (functional PCA) is performed on the fitted functional model. The Functional PCA report lists the eigenvalues that correspond to each functional principal component (FPC) in order from largest to smallest. The percent of variation accounted for by each FPC and the cumulative percent is listed and shown in a bar chart. There is a graph of the mean function as well as a graph for each component. The component graphs show the values of the eigenfunction.

You can perform model selection in the Functional PCA report to refine the selected number of functional principal components. There is a solution path plot that shows the Bayesian Information Criterion (BIC) plotted versus the number of FPCs. The current number of FPCs is designated by the dotted vertical line in the solution path plot. It is possible that models with different numbers of FPCs might have similar fits. Therefore, the solution path plot provides zones, which are intervals of values of the BIC statistic. There is a green zone and a yellow zone. The green zone contains values in the interval of the minimum BIC to the minimum BIC plus four and the yellow zone contains values in the interval of the minimum BIC plus four to the minimum BIC plus 10. By default, the model with the smallest number of FPCs within the green zone is selected. You can drag the slider at the top of the vertical line to change the number of FPCs. Dragging the slider automatically updates the other information in the Functional PCA report.

**Note:** The zones may appear small on your plot. Zoom in on the y-axis to better visualize the zones.

When Direct Functional PCA is performed, there is also an overall prediction plot and a grid of individual prediction plots. The grid of individual prediction plots has the same layout and controls as the grid of individual plots in the Data Processing report. At most, there are twenty plots shown at a time and there are drop-down menus and arrows that enable you to view different groups of individual prediction plots. Updating the number of FPCs automatically updates the prediction plots as well.

The prediction plots show the raw data and prediction curves that correspond to the current model. If there is a validation set, the predicted curves are not shown for functions that are in the validation set. The curve in the overall prediction plot is a prediction of the mean curve, given the specified number of FPCs. The curves in the individual prediction plots are prediction curves for each specific function, given the specified number of FPCs.

**Note:** The Functional PCA report is not shown if only a single function is modeled. Otherwise, if JMP is unable to perform Functional PCA, an error message appears in the Functional PCA report.

The following options are available in the Functional PCA red triangle menu:

**Diagnostic Plots**   Shows or hides the Actual by Predicted and the Residual by Predicted plots. Use these plots help assess how well the model fits the data, given the selected number of functional principal components.

**Score Plot**   Shows or hides a score plot of the FPC scores. Use the lists under Select Component to specify which FPCs are plotted on each axis of the Score Plot. If there is only one FPC, the FPC scores are plotted on the line $y = x$ and the lists to change the components are not shown. Score plots are useful for detecting outliers. In the case of FPC scores, the Score Plot is useful for detecting levels of the ID variable that have outlier functions. If you select a point in the score plot, the FPC Profiler is set to the scores for that function.

**Tip:** Hover over a point in the score plot to view a prediction plot of the fitted curve for that level of the ID variable.

**FPC Profiler**   Shows or hides a profiler of the FPC scores. The FPC Profiler includes a column for the input variable and a column for each FPC score. For each target function that is specified, there are two additional profilers. One measures the difference from the target function, and the other measures the integrated error from the target function. For more information about FPC Profiler red triangle menu options, see *Profilers*.

**Tip:** Use the Reset button to reset all of the FPC scores to 0 in the profiler.

**Customize Number of FPC's** Specifies the number of FPC scores to show in the Functional PCA. Specifying the number of FPC scores in this option also updates the Function Summaries report. To view the mean model, set the number of FPC's to 0.

# Functional Data Explorer Platform Options

The Functional Data Explorer red triangle menu contains the following options:

**Summaries** A submenu of the following options for functional summary statistics:

**Plot Mean** Shows or hides a plot of the functional mean in the Summaries report. On by default.

**Plot Standard Deviation** Shows or hides a plot of the functional standard deviation in the Summaries report. On by default.

**Plot Median** Shows or hides a plot of the functional median in the Summaries report.

**Models** A submenu of the following model options:

**B-Splines** Fits a basis spline (B-Spline) model to the data. Use the B-Spline model for non-periodic data.

**P-Splines** Fits a penalized basis spline (P-Spline) model to the data.

**Fourier Basis** Fits a Fourier Basis model to the data. Use the Fourier Basis model for periodic data. A periodic model assumes that the function finishes where it starts. See "Fourier Basis Model".

**Note:** If there are fewer than three unique input values, neither a Fourier basis model or a P-Spline model can be fit to the data, and a warning message appears.

**Model Controls** Shows a submenu that enables you to open the Model Controls panel prior to fitting a model. See "Model Controls".

**Direct Functional PCA** (Not available if there is only a single function.) Performs functional principal components analysis directly on the data, without fitting a basis function model first. This reduces computation time, particularly for large data sets. The implementation of Direct Functional PCA is as follows:

1. Align the input data to be between 0 and 1 and interpolate the observations to a common grid of input values.

2. Perform functional principal components analysis on the data.

3. Smooth the first eigenfunction using a P-Spline model with a knot at each grid point.

4. Remove the first smoothed eigenfunction from the data and repeat step 2 to step 4 until a large amount of the variation in the data is explained.

Once you perform a Direct Functional PCA, a Functional PCA report is shown. See "Functional PCA".

**Save Data**   Saves the processed data to a new data table. The processed data are saved in the stacked data format.

See *Using JMP* 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.

## **Functional Data Explorer Group Options**

Functional Data Explorer Group options are available only when there are multiple functional processes specified in the Stacked Data Format tab in the launch window.

**Summaries**   A submenu of options to show or hide plots of functional summary statistics for all functional processes. See "Summaries".

**Data Processing**   A submenu of data processing options that can be applied to all functional processes. See "Data Processing Report Options".

**Models**   A submenu of model options that can be fit to all functional processes. See "Models".

**Save Data**   Saves the processed data for all functional processes to a new data table. The processed data are saved in the stacked data format.

**Save Summaries**   Saves the summary statistics specified in each Function Summaries report to a new data table. The name of the new data table describes the model fit. This data table contains formula columns for the eigenfunctions, mean function, prediction function, and

conditional prediction function for each of the Y variables. In the data table, there is a profiler script for each Y variable that launches the prediction profilers for the prediction and conditional prediction formulas. These formulas are functions of the input variable, the ID variable, and the eigenfunctions.

**Arrange in Rows**   Specifies how many reports are displayed across the window.

See *Using JMP* 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.

## Data Processing Report Options

The Data Processing red triangle menu contains the following options:

**Cleanup**   A submenu of the following data cleanup options:

**Remove Zeros**   Removes observations with zero values. If there are no zeros in the data, an alert appears, indicating that no zero values were found.

**Remove Value**   Displays a specifications window that enables you to specify a value to remove from the data.

**Remove Selected**   Removes observations that correspond to rows that are selected in the data table.

**Remove Unselected**   Removes observations that correspond to rows that are not selected in the data table.

**Caution:** Remove Selected and Remove Unselected remove the row numbers. When Auto Recalc is enabled, you must add or delete rows before using these options.

**Filter X**   Removes X values that fall outside of a specified interval. When you select the Filter X option, you must specify Below and Above values. The X values that fall outside of the specified interval are not used for the analysis.

**Filter Y**   Removes Y values that fall outside of a specified interval. When you select the Filter Y option, you must specify Below and Above values. The Y values that fall outside of the specified interval are not used for the analysis.

**Reduce**   Reduces the data over the X values using one of the following techniques:

– Use the Grid tab to interpolate observations to a common grid of values. You can specify the grid size. By default, the grid size is half the number of unique input values and therefore reduces the number of total observations. If you are not interested in reducing the number of total observations, but simply want your observations to be on the same grid, specify the grid size to be the number of unique input values.

– Use the Bin tab to create a specified number of bins that are evenly spaced over the unique X values. For each function (or level of the ID, Function variable), the observations within a bin are averaged to produce a Y value for the corresponding bin level.

– Use the Thin tab to remove every N observation over the X values, where N is determined by the specified thinning rate. This is done for each function (or level of the ID, Function variable). By default, the thinning rate is 2, which removes half of the observations in each function.

**Note:** The Remove options exclude the specified observations from the analysis and modeling reports, but the observations remain unchanged in the data table.

**Transform**   A submenu of the following options to transform the data:

**Center**   Centers the output.

**Standardize**   Standardizes the output by centering and scaling the data to have mean 0 and variance 1.

**Range 0 to 1**   Scales the output to lie within the range of 0 and 1.

**Square Root**   Transforms the data by computing the square root of the output. The output values must be nonnegative.

**Square**   Transforms the data by computing the square of the output.

**Log**   Transforms the data by computing the natural logarithm of the output.

**Exp**   Transforms the data by computing the exponential function of the output.

**Negation**   Transforms the data by negating the output.

**Logit**   Transforms the data by computing the logit function of the output. The output values must be between 0 and 1.

**Log X**   Transforms the data by computing the natural logarithm of the input.

**Align** A submenu of the following options to align the input data:

**Row Alignment** Replaces the input values with the row number.

**Align Maximum** Aligns the functions using the observed maximum output value for each ID level. The input value associated with the observed maximum output value is set to zero for each ID level and the other input values are shifted up or down based on the difference between the observed maximum and zero.

**Align Minimum** Aligns the functions using the observed minimum output value for each ID level. The input value associated with the observed minimum output value is set to zero for each ID level and the other input values are shifted up or down based on the difference between the observed minimum and zero.

**Align 0 to 1** Aligns the output functions such that the range of the input values is 0 to 1.

---

**Tip:** Align 0 to 1 is particularly useful when you fit a P-Spline model.

---

**Align by Function** Aligns the output functions such that each function starts at the overall minimum of the input values and ends at the overall maximum of the input values.

**Dynamic Time Warping** (Available only when there is more than one function.) Aligns the output functions using dynamic time warping (DTW). DTW is a function alignment technique that finds an optimal warping to align two or more functions together. When you select the DTW option, a Select Reference Function window appears. Use this to select the reference function. The reference function is the function that the remaining functions are aligned to.

Once you select a reference function and click OK, a warping function plot is shown along with a list for the remaining query functions. On the warping function plot, the reference function is on the y-axis and the selected query function is on the x-axis. Deviations from the red diagonal line ($y = x$) indicate that the inputs of the query function have been warped for better alignment.

**Target Functions** (Available only when there is more than one function.) A submenu that enables you to load target functions.

**Load Targets** Shows a window that enables you to specify a target function. A target function is used for curve matching, where it is desirable for all of the functions to look like the target function. You can also specify two target functions to compare the remaining curves to the "best" and "worse" case functions.

If you specify one or more target functions, the data from the functions are not used in model fitting. For each specified target function, two rows are added to the FPC Profiler. See "FPC Profiler".

---

**Note:** Target functions must be loaded before any other preprocessing steps are performed.

---

## Dynamic Time Warping Options

**Plot Warping Functions**   Shows or hides the warping function plot. On by default.

**Save Distance Matrix**   Saves the distance matrix to a separate data table. The distance matrix can be useful for clustering the functions. The distance matrix data table contains a hierarchical clustering script.

**Save Warping Functions**   Saves the warping functions to a separate data table. Each row of the data table contains the DTW adjusted input variable, the original input variable, and the ID variable.

## Model Report Options

**Model Selection**   Displays a submenu of choices to use as the model selection criteria. The choices are AICc, BIC, and GCV. See *Fitting Linear Models*.

**Plot Basis**   Shows or hides a plot of all the basis functions on one graph.

**Diagnostic Plots**   Shows or hides the Diagnostics Plots report. See "Diagnostic Plots"

**Function Summaries**   Shows or hides the Function Summaries report. See "Function Summaries".

**Basis Function Coefficients**   Shows or hides the Basis Function Coefficients report. See "Basis Function Coefficients".

**Random Coefficients by Function**   Shows or hides the Random Coefficients by Function report. See "Random Coefficients by Function".

**Functional PCA**   Shows or hides the Functional PCA report. See "Functional PCA".

**Functional DOE Analysis**   (Available only if at least one supplementary variable is specified in the launch window.) Launches a Generalized Regression report within the FDE platform. A generalized regression model is fit to each of the FPC score functions using the supplementary variables as model effects. By default, a two degree factorial model is fit and the Estimation Method is Best Subset. Note that categorical quadratic effects are not included. If the number of terms in the model is greater than 21 or the number of functions is greater than 1000, the Estimation Method automatically switches to Pruned Forward. Alternatively, you can specify a model script in the original data table that defines the desired model fit. Modeling the FPC scores using the supplementary variables enables you to use the model fit to determine how the response changes based on the supplementary

variables. Use the FDOE Profiler to explore how the supplementary variables affect the response.

The Functional DOE Analysis report contains the following red triangle menu options:

**Generalized Regression for FPC Scores**   Shows or hides the Generalized Regression reports for each FPC score. For more information on Generalized Regression model reports, see *Fitting Linear Models*.

**Diagnostic Plots**   Shows or hides actual by predicted and residual plots for the response variable.

**FDOE Profiler**   Shows or hides the FDOE Profiler, which enables you to explore how the response changes based on the supplementary variables. For more information about the FPC Profiler red triangle menu options, see *Profilers*.

**Save Columns**   Shows a list with the options Save Prediction Formula and Save Residual Formula. These options save the corresponding formula to a new column in the data table.

**Save Data**   Saves the modeled data to a new data table. The modeled data are saved in the stacked data format.

**Remove Fit**   Removes the model report for the specified fit.

# Additional Examples of the Functional Data Explorer Platform

- "Example for Multiple Functional Processes"
- "Example of Functional DOE"

# Example for Multiple Functional Processes

This example uses the Fermentation Process.jmp and Fermentation Process Batch Yield Results.jmp sample data tables to analyze enzyme production. Yield is the amount of an enzyme produced by genetically modified yeast. There are 100 process measurements per batch that were taken at equally spaced times over a 12-hour period.

Use the Functional Data Explorer platform to fit models to the data and save functional principal components to a new data table. The functional principal components are then analyzed in the Generalized Regression personality of the Fit Model platform.

## Fit Functional Models

1. Select **Help > Sample Data Library** and open Functional Data/Fermentation Process.jmp.

2. Select **Analyze > Specialized Modeling > Functional Data Explorer**.

3. In the Stacked Data Format tab, select Ethanol through pH and click **Y, Output**.

4. Select Time and click **X, Input**.

5. Select BatchID and click **ID, Function**.

6. Click **OK**.

7. Click the Functional Data Explorer Group red triangle and select **Data Processing > Align > Align 0 to 1**. This aligns the input variable to be between 0 and 1 in each Functional Data Explorer report.

8. Click the Functional Data Explorer Group red triangle and select **Models > B-Splines**. This fits a B-spline model to each of the functional processes.

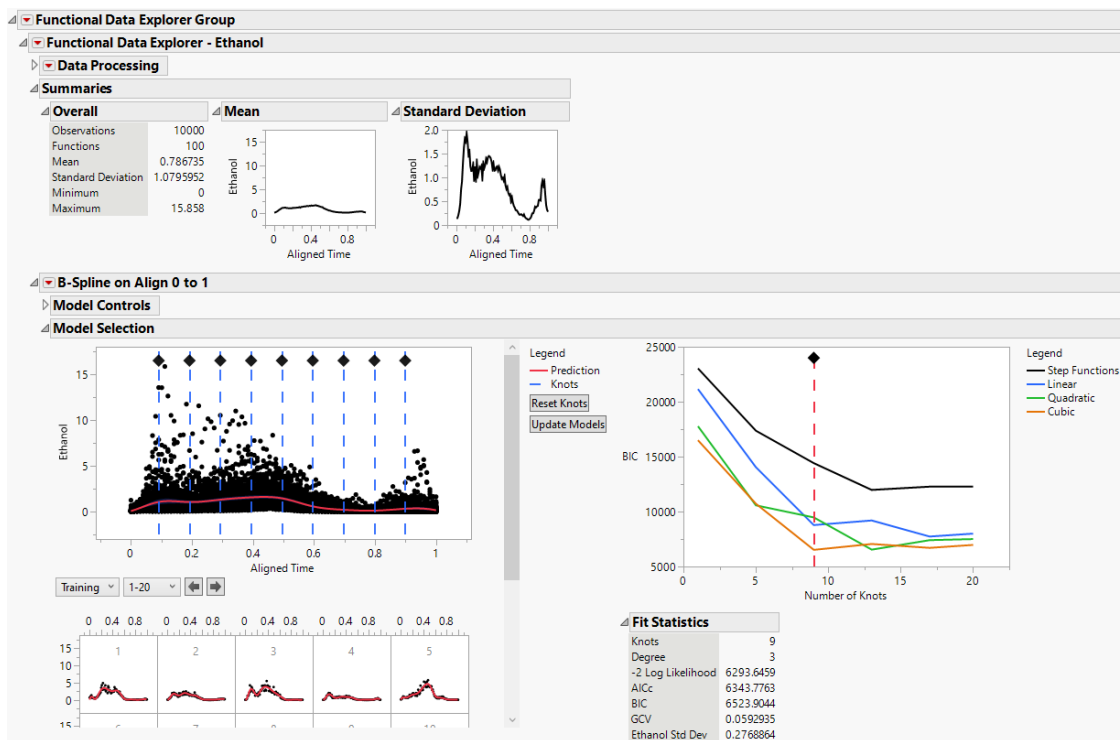**Figure 16.7**  Functional Data Explorer Report for Ethanol

**Figure 16.8** Model Summary Report for Ethanol



Figure 16.7 and Figure 16.8 show the model reports for one of the functional process variables, Ethanol. Scroll through the full report to view the models fit for each of the process variables. Next, use the FPCs in the Function Summaries report in an analysis.

### Save FPCs and Link Yield Results

1. Press Ctrl, click any Function Summaries red triangle, and select **Customize Function Summaries**.

2. In the box next to **Enter number of FPCs to show**, type 3.

3. Click the **Deselect All Summaries** box.

4. Click **OK**.

5. Click the Functional Data Explorer Group red triangle and select **Save Summaries**.

6. Select **Help > Sample Data Library** and open Functional Data/Fermentation Process Batch Yield Results.jmp.

7. In the Functional Data Explorer Model Summaries.jmp data table, right-click BatchID and deselect **Link ID**.

8. In the Functional Data Explorer Model Summaries.jmp data table, right-click BatchID and select **Link Reference** > Fermentation Process Batch Yield Results.jmp.
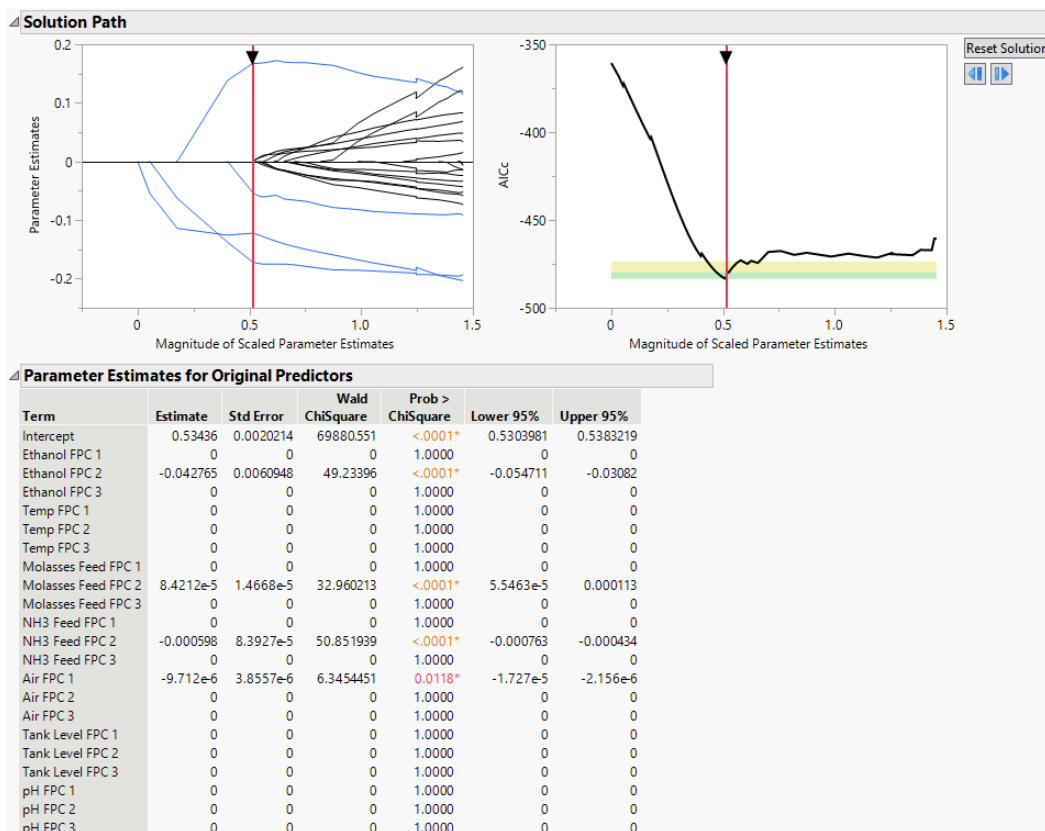
   This virtually joins the yield data table and the summaries data table.

### Fit a Generalized Regression Model

Use the Generalized Regression personality of the Fit Model platform to determine how Yield is affected by the functional process variables.

1. In the Functional Data Explorer Model Summaries.jmp data table, select **Analyze > Fit Model**.
2. Click the triangle next to referenced by BatchID to Fermentation Process Batch Yield Results.
3. Select Yield[BatchID] and click **Y**.
4. Select the remaining columns, except Time and BatchID, and click **Add**.
5. Change the Personality to Generalized Regression.
6. Click **Run**.
7. Select the **Adaptive** box.
8. Click **Go**.

**Figure 16.9** Generalized Regression Report for Batch Yield



#### Parameter Estimates for Original Predictors

| Term | Estimate | Std Error | Wald ChiSquare | Prob > ChiSquare | Lower 95% | Upper 95% |
|---|---|---|---|---|---|---|
| Intercept | 0.53436 | 0.0020214 | 69880.551 | <.0001* | 0.5303981 | 0.5383219 |
| Ethanol FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Ethanol FPC 2 | -0.042765 | 0.0060948 | 49.23396 | <.0001* | -0.054711 | -0.03082 |
| Ethanol FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Temp FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Temp FPC 2 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Temp FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Molasses Feed FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Molasses Feed FPC 2 | 8.4212e-5 | 1.4668e-5 | 32.960213 | <.0001* | 5.5463e-5 | 0.000113 |
| Molasses Feed FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| NH3 Feed FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| NH3 Feed FPC 2 | -0.000598 | 8.3927e-5 | 50.851939 | <.0001* | -0.000763 | -0.000434 |
| NH3 Feed FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Air FPC 1 | -9.712e-6 | 3.8557e-6 | 6.3454451 | 0.0118* | -1.727e-5 | -2.156e-6 |
| Air FPC 2 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Air FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Tank Level FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Tank Level FPC 2 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| Tank Level FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| pH FPC 1 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| pH FPC 2 | 0 | 0 | 0 | 1.0000 | 0 | 0 |
| pH FPC 3 | 0 | 0 | 0 | 1.0000 | 0 | 0 |

The Generalized Regression report shows that Yield is significantly affected by certain components of Ethanol, Molasses Feed, NH3 Feed, and Air. The RSquare for the model is 0.73225. By using FDE to perform dimension reduction on the functional processes first, you greatly reduce the number of variables, while still retaining the ability to build a reasonable prediction models.

# Example of Functional DOE

This example uses the Formulation For Homogeneity DOE.jmp data table to explore how combinations of ingredients effect the homogeneity grade of a compound across different temperatures. The three ingredients that form the compound are the primary ingredient labeled Active, Water, and Solvent. There are 32 combinations of ingredients; these combinations are denoted by the values in the Formulation column. For each formulation, the Homogeneity Grade is measured from 20 to 70 degrees Celsius at 5 degree intervals. By plotting these measurements across the different temperatures, you can form a curve for each formulation. Use the Functional Data Explorer platform to determine how the curves change depending on the values of Active, Water, and Solvent.

1.  Select **Help > Sample Data Library** and open Functional Data/Formulation For Homogeneity DOE.jmp.

2.  Select **Analyze > Specialized Modeling > Functional Data Explorer**.

3.  In the Stacked Data Format tab, select Homogeneity Grade and click **Y, Output**.

4.  Select T and click **X, Input**.

5.  Select Formulation and click **ID, Function**.

6.  Select Solvent, Active, and Water and click **Z, Supplementary**.

    These variables are not used in the initial functional data analysis. However, by specifying them here as supplementary variables, you can use them later when fitting models.

7.  Click **OK**.

8.  Click the Functional Data Explorer red triangle and select **Models > B-Splines**.

    The best fitting model is a linear spline model with one knot.

**Figure 16.10** Functional PCA for Homogeneity Grade
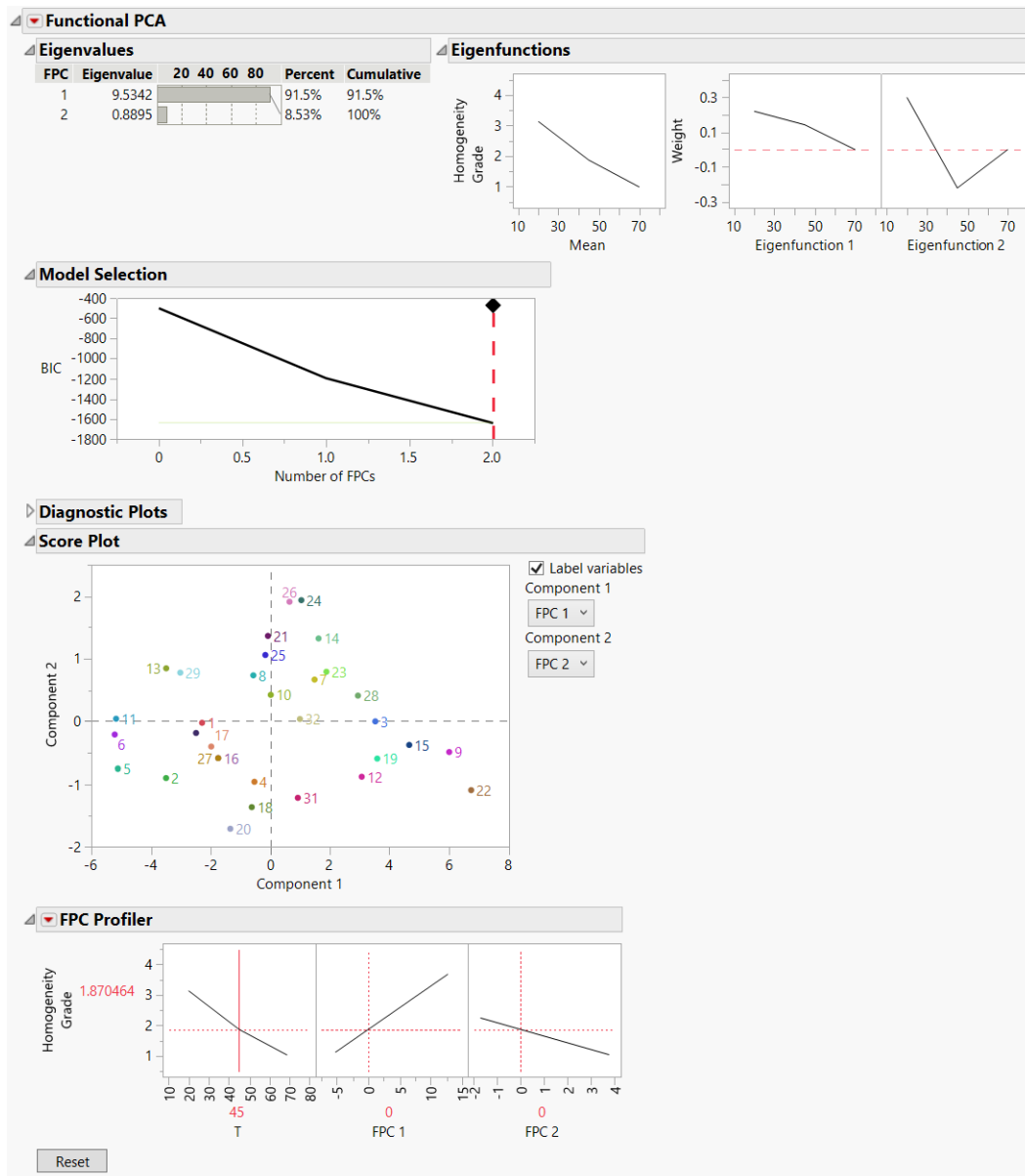


The Functional PCA report shows that two functional principal components are needed to explain the variability in the data. The FPC Profiler shows how Homogeneity Grade is affected by the FPCs across the different temperatures. However, there is no way to tell how the actual ingredients affect the Homogeneity Grade.

9.  Click the red triangle next to B-Spline on Initial data and select **Functional DOE Analysis**.

The Functional DOE Analysis option fits a generalized regression model to each principal component score using the three supplementary variables as predictors. Because the data table contains a Model script, the models that are fit are based on the model defined in the script.

**Note:** If your data table does not contain a Model script, a two degree factorial model is fit.

**Figure 16.11** FDOE Profiler for Homogeneity Grade
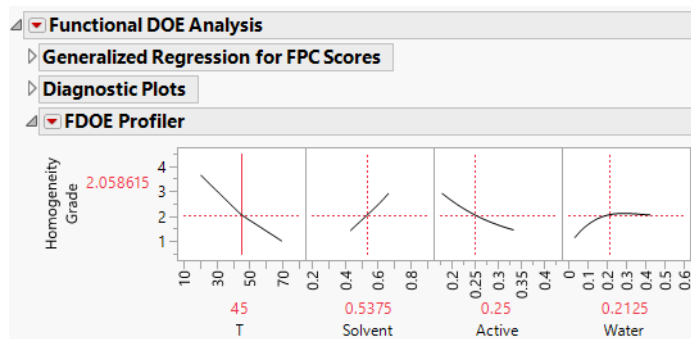


The FDOE Profiler enables you to explore how Homogeneity Grade changes based on the predictors you are actually interested in. In general, Homogeneity Grade seems to decrease as the amount of Active increases.

**Note:** In this example, the values of Active, Solvent, and Water are constrained to sum to 1. This constraint is reflected in the profiler.

# Statistical Details for the Functional Data Explorer Platform

- "Functional Model Fits"
- "Function Summaries Details"

# Functional Model Fits

All of the model fits in the Functional Data Explorer platform rely on basis function expansion. Basis functions are a set of independent functions. Any function, $f(t)$, can be approximated by taking a linear combination of $K$ basis functions, denoted as $\phi_k$. In general, a function is approximated as follows:

$$\hat{f}(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

where the $c_k$ parameters are the basis coefficients. The amount of smoothing is determined by the number of basis functions, $K$. For more information about basis function expansion, see Ramsay and Silverman (2005).

## Fourier Basis Model

The $k = 1,..., K$ functions that define the Fourier basis are defined such that $\phi_0 = 1$, $\phi_{2k-1} = \sin(r\omega t)$, and $\phi_{2k} = \cos(r\omega t)$. Then, the approximated function is defined as follows:

$$\hat{f}_i(t) = c_{i0} + c_{i1}\sin\omega t + c_{i2}\cos\omega t + c_{i3}\sin 2\omega t + c_{i4}\cos 2\omega t + \dots$$

This approximates a periodic function with period $A$ defined as $A = 2\pi/\omega$. The coefficients are a combination of fixed basis function coefficients and random coefficients by function. Each $c$ is defined as follows:

$$c_{ik} = \beta_k + \alpha_{ik}$$

where $\beta_k$ is the fixed coefficient for basis function $k$ and $\alpha_{ik}$ is the random coefficient for basis function $k$ for a specific functional process $i$. For the Fourier Basis model, the estimates for the $\beta_k$ and $\alpha_{ik}$ parameters are found in the Basis Function Coefficients table and Random Coefficients by Function table, respectively.

Fourier basis models have an intercept term and an equal number of sine and cosine terms, which are referred to as Fourier pairs. Therefore, $K$ is always odd. For example, if $K = 7$, there is an intercept term and three Fourier pairs.

# Function Summaries Details

The Function Summaries report includes the following summary statistics: integrated difference, root integrated square error (RISE), and root integrated function squared (RIFS). These summary statistics are defined in this section. For all equations in this section, it is assumed that the input values have been aligned between 0 and 1. The following notation is used for all functions:

$\hat{f}_i(x)$   is the estimated function for curve $i$ at input value $x$

$\hat{m}(x)$   is the estimated mean function at input value $x$

### Integrated Difference

The integrated difference is the average difference between points on the ID specific curve and the overall mean curve. It can be used to determine if the ID specific curve falls above or below the mean curve, on average. The integrated difference for curve $i$ is defined as follows:

$$ID_i = \int_0^1 (\hat{f}_i(x) - \hat{m}(x))\, dx$$

### RISE

Root integrated square error (RISE) is the average distance between the ID specific curve and the overall mean curve. It can be used to determine which ID specific curves are most similar or most different from the overall mean curve. The RISE summary value for curve $i$ is defined as follows:

$$RISE_i = \sqrt{\int_0^1 (\hat{f}_i(x) - \hat{m}(x))^2\, dx}$$

### RIFS

Root integrated function squared (RIFS) is a summary value for analyzing optimal curve scenarios. If you pre-subtract off an optimal curve, you can use the smallest values of RIFS to determine which functions are closest to the optimal curve. The RIFS summary value for curve $i$ is defined as follows:

$$\text{RIFS}_i = \sqrt{\int_0^1 \hat{f}_i(x)^2 \, dx}$$
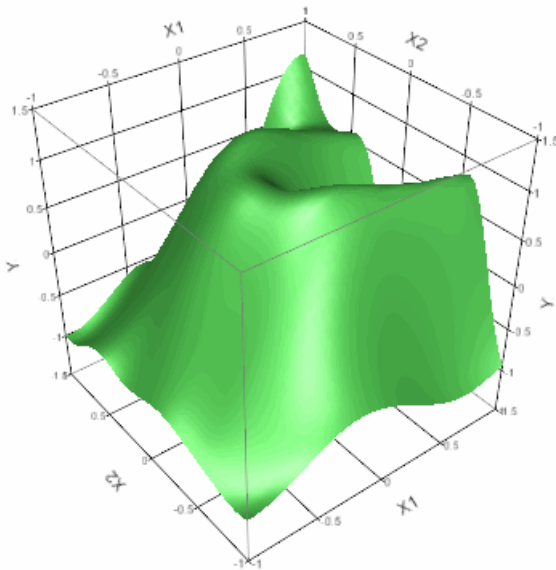
# Chapter **17**

## 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 17.1** Gaussian Process Prediction Surface Example

# Contents

# 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 17.2** Gaussian Process Report
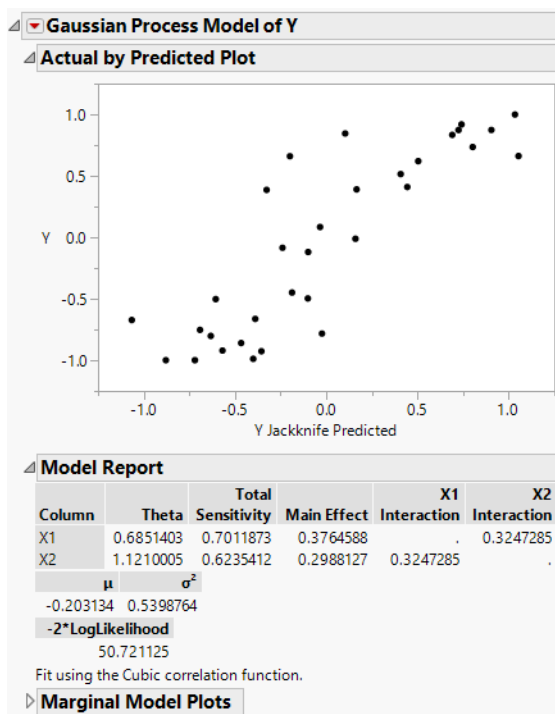


**Note:** The estimated parameters can be different due to different starting points in the minimization routine, the choice of correlation type, and the inclusion of a nugget parameter.

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 17.3**  3D Surface Plot of the Actual and Predicted Ys
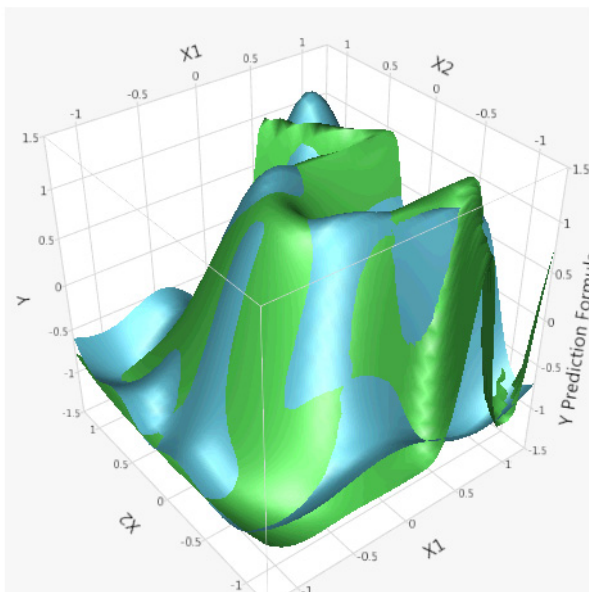


The two surfaces are similar. The impact of X1 and X2 on the response Y can be visualized. You can rotate the plot to view it from different angles. Marginal plots are another tool to use to understand the impact of the factors on the response.

# Launch the Gaussian Process Platform

Launch the Gaussian Process platform by selecting **Analyze > Specialized Modeling > Gaussian Process**.

**Figure 17.4** Gaussian Process Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

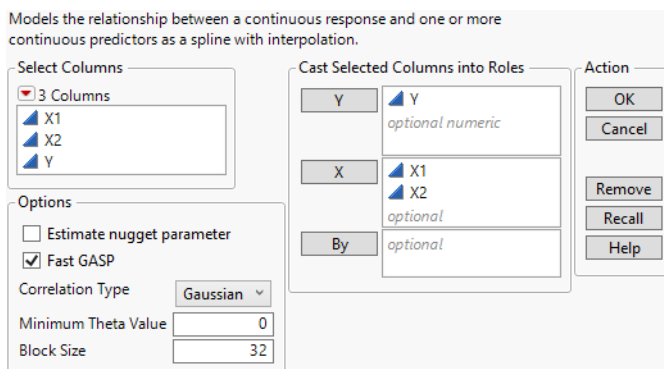**Y**   Assigns the continuous columns to analyze.

**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.

**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 more 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**    The 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

- "Actual by Predicted Plot"
- "Model Report"
- "Marginal Model Plots"

## 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".

**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".

**Mu and Sigma$^2$**   The mean and variance model parameters.

**Nugget**   The 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**   The estimated value of twice the negative log-likelihood function. See *Fitting Linear Models*.

## Marginal Model Plots

A marginal plot appears for each factor in the model. For each plot, all factors except one are integrated out using a distribution that is uniform over the ranges of the other factors in the data. The marginal prediction of the response for the remaining factor of interest is drawn in blue. The data points for the observed response values are included to show how well the marginal model fits the data. For models with a small number of factors, these plots can be used as diagnostic plots.

# Gaussian Process Platform Options

The Gaussian Process red triangle menu contains the following options:

**Profiler**   Opens the standard Profiler. See *Profilers*.

**Contour Profiler**   Opens the Contour Profiler. See *Profilers*.

**Surface Profiler**   Opens the Surface Profiler. See *Profilers*.

**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 "Formula Depot".

**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 "Formula Depot".

**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 *Using JMP* 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"
- "Example of Gaussian Process Model with Categorical Predictors"

# 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.
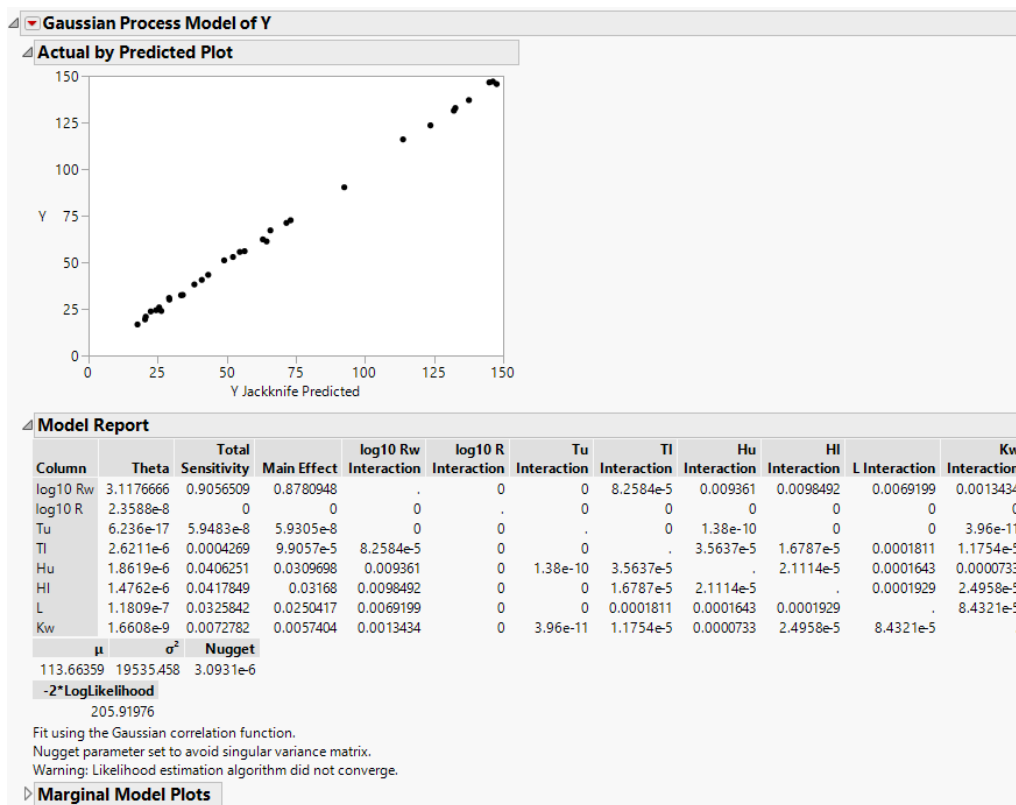
1. Select **Help > Sample Data Library** and open Design Experiment/Borehole Latin Hypercube.jmp.
2. Select **Analyze > Specialized Modeling > Gaussian Process**.
3. Select log10 Rw through Kw and click **X**.
4. Select Y and click **Y**.
5. **JMP PRO** In JMP Pro, to run the analysis faster, leave the Fast GASP checked.
6. Click **OK**.

**Figure 17.5** Borehole Latin Hypercube 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 | 3.1176666 | 0.9056509 | 0.8780948 | . | 0 | 0 | 8.2584e-5 | 0.009361 | 0.0098492 | 0.0069199 | 0.0013434 |
| log10 R | 2.3588e-8 | 0 | 0 | 0 | . | 0 | 0 | 0 | 0 | 0 | 0 |
| Tu | 6.236e-17 | 5.9483e-8 | 5.9305e-8 | 0 | 0 | . | 0 | 1.38e-10 | 0 | 0 | 3.96e-11 |
| Tl | 2.6211e-6 | 0.0004269 | 9.9057e-5 | 8.2584e-5 | 0 | 0 | . | 3.5637e-5 | 1.6787e-5 | 0.0001811 | 1.1754e-5 |
| Hu | 1.8619e-6 | 0.0406251 | 0.0309698 | 0.009361 | 0 | 1.38e-10 | 3.5637e-5 | . | 2.1114e-5 | 0.0001643 | 0.0000733 |
| Hl | 1.4762e-6 | 0.0417849 | 0.03168 | 0.0098492 | 0 | 0 | 1.6787e-5 | 2.1114e-5 | . | 0.0001929 | 2.4958e-5 |
| L | 1.1809e-7 | 0.0325842 | 0.0250417 | 0.0069199 | 0 | 0 | 0.0001811 | 0.0001643 | 0.0001929 | . | 8.4321e-5 |
| Kw | 1.6608e-9 | 0.0072782 | 0.0057404 | 0.0013434 | 0 | 3.96e-11 | 1.1754e-5 | 0.0000733 | 2.4958e-5 | 8.4321e-5 | . |

| μ | σ² | Nugget |
|---|---|---|
| 113.66359 | 19535.458 | 3.0931e-6 |

**-2*LogLikelihood**
205.91976

Fit using the Gaussian correlation function.
Nugget parameter set to avoid singular variance matrix.
Warning: Likelihood estimation algorithm did not converge.

▷ **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. Use the profiler to visualize the sensitivities.

7.  Click the Gaussian Process Model of Y red triangle and select **Profiler**.

    The desirability functions automatically appear because the response, Y, has a Response Limits column property.

8.  Click the Prediction Profiler red triangle and select **Optimize and Desirability > Maximize Desirability**.

**Figure 17.6** Gaussian Process Model Profiler
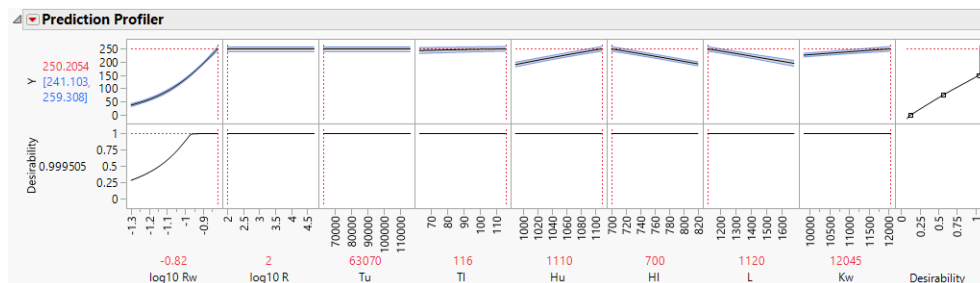


The desirability function is set to maximize the response Y because Y has goal set to Maximize in the Response Limits column property. Maximizing the desirability functions identifies the values of the factors that maximize the response. The shaded bands represent the 95% confidence intervals.

**Note:** Your estimates can differ from those shown in Figure 17.5, 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".

**Figure 17.7** Algorithm Data Report
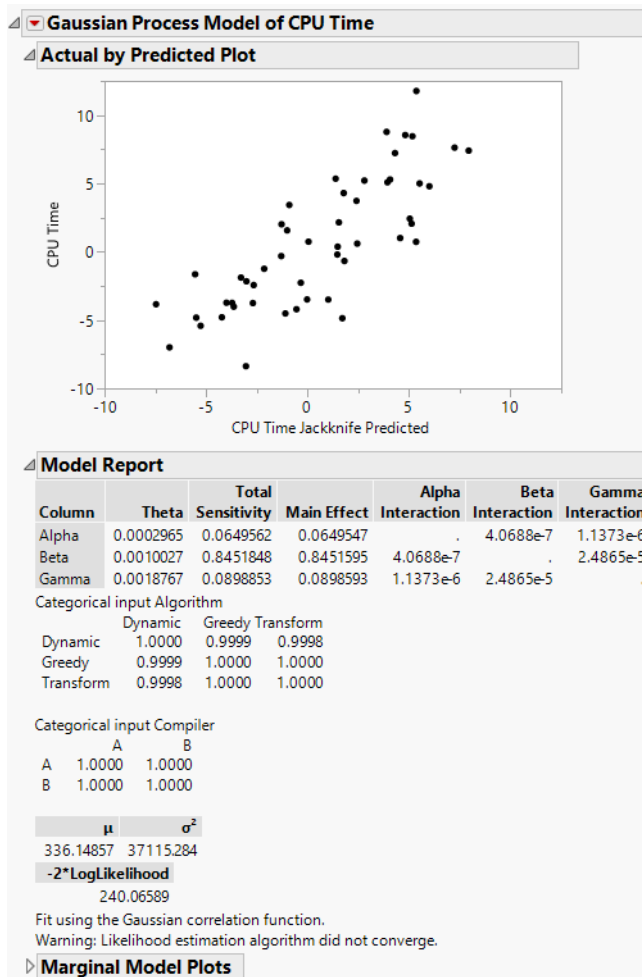


The actual by predicted plot shows a strong correlation between the actual and predicted CPU times. This in an indication that the Gaussian process prediction model is a good approximation of the true function. In the Model Report, the Beta predictor has the highest total sensitivity. This indicates 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"
- "Models with Categorical Predictors"
- "Variance Formula Parameterization"
- "Model Fit Details"

## 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 $\mathbf{R}$ 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 $k^{\text{th}}$ predictor

$x_{ik}$ = the value of the $k^{\text{th}}$ predictor for subject $i$

$x_{jk}$ = the value of the $k^{\text{th}}$ 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 $\mathbf{R}$ matrix consists of the following elements:

$$r_{ij} = \prod_{k} \rho(d; \theta_k)$$

where

$$d = x_{ik} - x_{jk}$$

$$\rho(d;\theta) = \begin{cases} 1 - 6(d\theta)^2 + 6(|d|\theta)^3, & |d| \leq \dfrac{1}{2\theta} \\[2mm] 2(1 - |d|\theta)^3, & \dfrac{1}{2\theta} < |d| \leq \dfrac{1}{\theta} \\[2mm] 0, & \dfrac{1}{\theta} < |d| \end{cases}$$

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

$\theta_k$ = theta parameter for the $k^{th}$ continuous predictor

$x_{ik}$ = the value of the $k^{th}$ continuous predictor for subject $i$

$x_{jk}$ = the value of the $k^{th}$ continuous 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 $\tau$ parameter for each combination of levels of a categorical variable, where $\tau_{ij}$ corresponds 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. See Qian et al. (2012).

# 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. These are the parameters:

- $\mu$ is the Gaussian Process mean,

- $\sigma^2$ is the Gaussian Process variance,

- Theta corresponds to the values of $\theta_k$ in the definition of **R**.

- The off-diagonals of the categorical input correlation matrices correspond to the values of $\tau_{pij}$ 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.
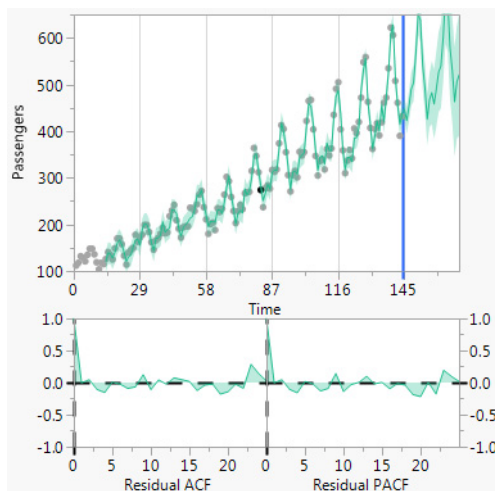
---

# 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 and State Space Smoothing models that have the ability to incorporate seasonality and long term trends all in one model. You can also perform a Box-Cox transformation and analyze and model the transformed series.

There are several methods to assess the forecasting performance of models. The Forecast on Holdback feature partitions the time series into a training portion to build models and a and holdback portion to assess forecasting performance.

**Figure 18.1** Forecast Plot

# Contents

# Overview of the Time Series Platform

A time series is a set $y_1, y_2, \dots, y_N$ 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. You can also view and apply a Box-Cox transformation to your data.

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 (ARIMA) models and State Space Smoothing models. Both of these types of models are the most statistically complex, but also provide the most flexibility. Advanced exponential smoothing, ARIMA, and State Space Smoothing 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 18.2** Time Series Analysis Report for Raleigh Temps.jmp
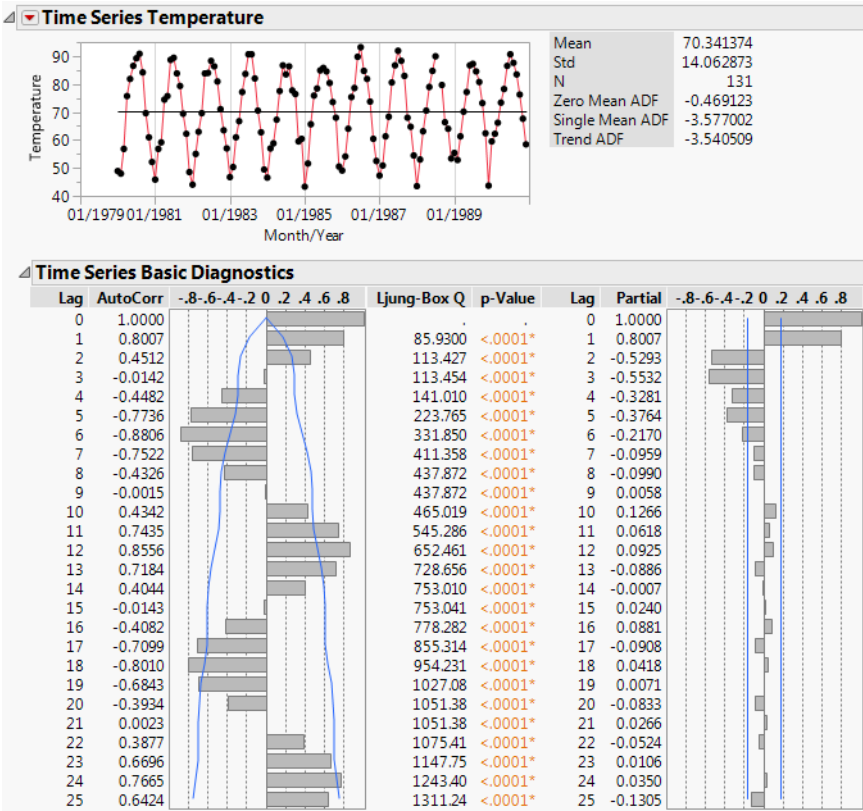
The Time Series graph shows that the series is cyclic. This cyclic component is also apparent in the autocorrelation chart. Points that are 1 lag apart are positively correlated, with an AutoCorr value of 0.8007. As points become farther apart, they become negatively correlated, then positively correlated 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 18.3** Model Comparison Table for Raleigh Temps.jmp
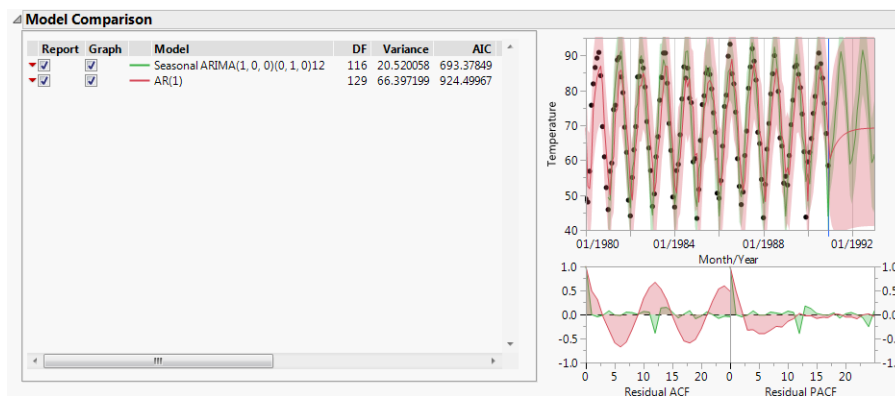


The Model Comparison table is sorted by the AIC statistic, in decreasing order. This means that the best fitting model appears at the top of the report. The AIC value for the seasonal ARIMA model (693.4) is much smaller than the value for the regular ARIMA model (924.5). The graph shows that while the 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 18.4.

**Figure 18.4** The Time Series Launch Window
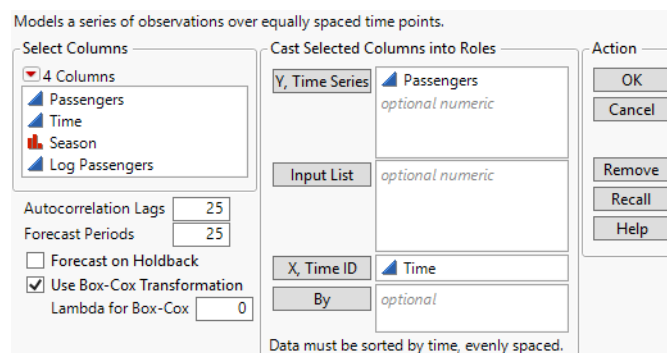


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

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 checks only whether the time stamps are increasing. 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 observations that are forecast using each model
fitted to the data. The default number of forecasts is 25.

– If you select the Forecast on Holdback option, the Forecast Periods value specifies the
number of observations at the end of the series that are not included in the model
fitting. These values are then forecast using the fitted model.

– If you do not select the Forecast on Holdback option, the Forecast Periods value
specifies the number of future observations that are forecast after the end of the series.

**Forecast on Holdback**   Determines whether forecasts are made on future observations or on
the holdback observations. If the Forecast on Holdback option is selected, the forecasts are
made on the holdback set that is determined by the number specified in the Forecast
Periods option.

**Use Box-Cox Transformation**   Transforms the original data using a Box-Cox transformation
with the lambda that is specified in the Lambda for Box-Cox option. When this option is
selected, all analyses in the Time Series report are performed on the transformed data.
However, all forecasts are transformed back and reported on the original scale.

**Lambda for Box-Cox**   (Available only when Use Box-Cox Transformation is selected.)
Specifies the lambda parameter used for the Box-Cox transformation of the original
data.

**Note:** In order to preserve the order and spacing of a time series, the Time Series platform
treats excluded rows in the data table as missing values.

# 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"
- "Time Series Basic Diagnostics Chart"

## 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. If a holdback set is specified, there is a vertical reference line on the graph that separates the training data from the holdback data.

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:

**Lambda for Box-Cox**   (Appears only if a Box-Cox Transformation is specified.) The value of lambda used in the Box-Cox transformation.

**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 as the default. 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, which is computed as follows:

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

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 ($\pm 2$ standard errors), which is computed as follows:

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

**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 $\pm\, 2$ standard errors for approximate 95% prediction limits, where the standard error is computed as follows:

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

## 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

- "Time Series Diagnostics"
- "Differencing and Decomposition"
- "ARIMA and Seasonal ARIMA Models"
- "Smoothing Models"
- "State Space Smoothing Models"
- "Transfer Function Models"
- "Smoothing Model Specification Windows"

## 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".

**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".

**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".

**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".

**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" and "Statistical Details for Spectral Density".

# Differencing and Decomposition

**Difference**   Shows the Differencing Specification window (Figure 18.5). 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:

$$w_t = (1 - \mathbf{B})^d (1 - \mathbf{B}^s)^D y_t$$

where $t$ is the time index and $\mathbf{B}$ 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 18.5**  Differencing Specification Window



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. See "Additional Examples of the Time Series Platform".

**Decomposition**   (Not available when either Forecast on Holdback or Use Box-Cox
   Transformation are selected in the launch window.) 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" and "Linear Trend Report".

   **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" and "Cycle Report".

   **X11**   Removes trend and seasonal effects using the X-11 method developed by the US
      Bureau of the Census (Shiskau et. al. 1967). For more information about the X-11
      method, see "Statistical Details for X-11 Decomposition". 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".

      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. For an example of an appropriate time column for an X11
      analysis, see "Create Appropriate Time ID Column".

**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.

**Show Box-Cox Transformation Plot**    (Not available if you specified Use Box-Cox Transformation in the launch window.) Shows or hides the Box-Cox Transformation plot. The Box-Cox transformation is a power transformation based on the Lambda parameter ($\lambda$) and uses the following formula:

$$Y_\lambda = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(y) & \text{if } \lambda = 0 \end{cases}$$

The Box-Cox Transformation Plot shows a graph of the transformed time series and options for setting the Lambda parameter ($\lambda$). The Lambda box shows the current value of the parameter. You can also use the Lambda slider to change the value of the parameter, which then automatically updates the plot. The Low and High boxes set the range of the slider.

***

**Tip:** Use the Box-Cox Transformation plot to see the transformed time series for different values of Lambda. Once you decide on a value, relaunch the analysis and select **Use Box-Cox Transformation** and enter the desired Lambda next to Lambda for Box-Cox. This runs the Time Series platform on the transformed series.

***

**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 × 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.

***

**Note:** For the cross correlation plot, the standard errors are calculated under the null hypothesis as 1/sqrt($n$ - $k$), where $n$ is the number of nonmissing data values and $k$ is the number of autocorrelation lags.

***

**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. This technique involves fitting an ARIMA model to the input series such that the residuals are equivalent to white noise. The same model is then used to estimate the output series. You can use the cross correlation between the filtered input and filtered output series to

determine an appropriate lag for 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".

---

**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 18.6**  ARIMA Specification Window
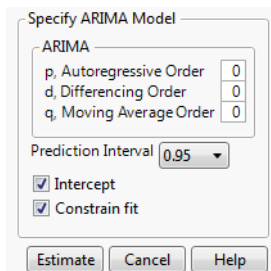


$p$, **Autoregressive Order**   The order $p$ of the polynomial $\varphi(\mathbf{B})$ operator.

$d$, **Differencing Order**   The order $d$ of the differencing operator.

$q$, **Moving Average Order**   The order $q$ of the differencing operator $\theta(\mathbf{B})$.

**Prediction Interval**   Enables you to set the prediction level between 0 and 1 for the forecast prediction intervals.

**Intercept**   Determines whether the intercept term $\mu$ 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".

**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".

**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".

## Smoothing Models

Shows a submenu of smoothing models. Once you select a smoothing model, a specification window appears. See "Smoothing Model Windows". For each model that is specified, a Smoothing Model Report appears in the report window. See "ARIMA and Seasonal ARIMA Model Report". Smoothing models represent the evolution of a time series by the model:

$$y_t = \mu_t + \beta_t t + s(t) + a_t$$

where

$\mu_t$ is the time-varying mean term

$\beta_t$ is the time-varying slope term

$s(t)$ is one of the s time-varying seasonal terms

$a_t$ are the random shocks

For more information about the general smoothing model equation, see "Statistical Details for Exponential Smoothing Models". The following smoothing models are available:

**Simple Moving Average**   (Not available when either Forecast on Holdback or Use Box-Cox Transformation are selected in the launch window.) 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 the smoothing window options are 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. See "Simple Smoothing Average Specification Window".

**Simple Exponential Smoothing**     A model with a level component. See "Simple Exponential Smoothing".

**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".

**Linear Exponential Smoothing**     A model with a level component and a trend component. See "Linear (Holt) Exponential Smoothing".

**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".

**Seasonal Exponential Smoothing**     A model with a level component and a seasonal component. See "Seasonal Exponential Smoothing".

**Winters Method**     A model with a level component, a trend component, and a seasonal component. See "Winters Method (Additive)".

---

**Note:** Each smoothing model, except for the Simple Moving Average model, is a traditional exponential smoothing model and 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.

---

## State Space Smoothing Models

Shows the Specify State Space Smoothing Models window, which enables you to fit a variety of state space smoothing models as defined by Hyndman et al. (2008). A state space smoothing model is defined based on its error, trend component, and seasonal component.

- The errors can be additive (A) or multiplicative (M).
- The trend component can be none (N), additive (A), additive damped ($A_d$), multiplicative (M), or multiplicative damped ($M_d$).
- The seasonal component can be none (N), additive (A), or multiplicative (M).

A specific model can be represented by its ETS (Error, Trend, Seasonal). Use the check boxes in the Specify State Space Smoothing Models window to select the error, trend, and seasonality for the desired models. Click **Select Recommended** to select the check boxes that correspond to the models recommended by the platform. The window opens with the recommended models selected. Click **Select All** to select all check boxes or **Deselect All** to deselect all check boxes. The window also contains the following options:

**Period**    Specifies seasonality values to be considered in the model fitting process.

**Constrain Parameters**    Constrains the parameters in such a way that the further an observation is from the present, the less effect it has on the present state of the model. In State Space Smoothing models, a forecast at time *t*, given all previous observations, is the same as the weighted sum of all observations up to time *t*. The weights are a function of the parameters. Therefore, constraining the parameters ensures that the weights for past observations go to zero and that the further an observation is from the present, the faster the weight goes to zero.

When you click **OK**, the specified set of models is fit. Summary values for each state space smoothing model are added to the Model Comparison table. Individual fit reports are added to the report window.

## 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 "Prewhitening". For more information about transfer functions, see "Statistical Details for Transfer Functions".

**Note:** Currently, the Transfer Function option has limited support of missing values. Because the Time Series platform also treats excluded rows as missing values, transfer function models cannot be fit on a data table with excluded rows.

**Figure 18.7** Transfer Function Model Specification Window



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 the model has an intercept. If unchecked, it is assumed that $\mu$ 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 18.8**  ARIMA Model Group Specification Window



Once you specify the models and click **Estimate**, a Model Report for each specified model
    is added to the report window. See "Reports".

**Combine and Save Forecasts from Models**    Creates a new data table with the combined
    results from all model fits in the report.

**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 *Using JMP* 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.

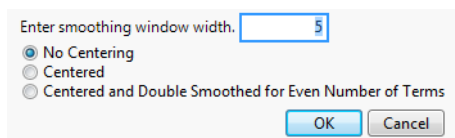**Save Script**   Contains options that enable you to save a script that reproduces the report to several destinations.

**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 18.9** Simple Smoothing Average Specification Window



**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 18.10**  Smoothing Model Specification Window



**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 following constraint options are available:

**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 18.11**  Custom Smoothing Weights



The example shown in Figure 18.11 has the Level weight ($\alpha$) fixed at a value of 0.3 and the Trend weight ($\gamma$) bounded by 0.1 and 0.8. In this case, the value of the Trend weight is allowed to move within the range 0.1 to 0.8 while the Level weight is held 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"
- "Decomposition Reports"
- "Model Comparison Report"
- "ARIMA and Seasonal ARIMA Model Report"
- "State Space Smoothing Report"
- "Transfer Function Report"
- "Spectral Density Report"

## 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".

**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.

**Remove Fit**   Removes the Difference report from the report window.

# Decomposition Reports

This section provides details about the reports obtained from the three decomposition options:

- "Linear Trend Report"
- "Cycle Report"
- "X11 Report"

## Linear Trend Report

Contains the values of $\beta_0$ and $\beta_1$ from the linear regression model that is fit to the data:

$$\text{Trend}_t = \beta_0 + \beta_1 \, time$$

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:

- $C$ is the (optional) Constant
- $A$ is the Amplitude of the cosine wave

- *U* is the number of Units per Cycle
- *P* is the Phase of the cosine wave
- *t* is one less than the row number of a given observation

The decycled series is equal to $D_t = O_t - \text{Cycle}_t$, 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 18.1.

**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.

**Remove Fit**   Removes the X11 report from the report window.

**Table 18.1** 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 |

**Table 18.1** Descriptions of the Categories of X11 Output Tables  *(Continued)*

| Letter Prefix | Category Description |
| --- | --- |
| D | final estimates of seasonal, trend cycle, and irregular components |
| E | analytical tables |
| F | summary measures |

For more information about the contents of the X11 output tables, see Shisken et. al. (1967) or *SAS/ETS 15.2 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. By default, the models are sorted by the AIC statistic, in increasing order. To sort the model by a different statistic, right-click any column in the Model Comparison Table and select Sort by Column. Alternatively, you can also click on the column header to sort. For definitions of the fit statistics, see "Model Summary Table". 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:

$$\text{AICWeight} = \exp[-0.5(\text{AIC} - \text{BestAIC})] / \sum_{k=1}^{K} (\exp[-0.5(\text{AIC}_k - \text{BestAIC})])$$

$K$ is the total number of models, $\text{AIC}_k$ is the AIC value for model $k$, and BestAIC is the AIC value for the model with the minimum AIC value.

If you specify a holdback set, the fit statistics that are computed on the training data are not shown in the Model Comparison table. Instead, the following metrics to determine the forecasting performance of the individual models on the holdback set are computed: Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE). By default, the models are sorted by the RMSE statistic, in increasing order. To sort the model by a different statistic, right-click any column in the Model Comparison Table and select Sort by Column. Alternatively, you can also click on the column header to sort. For definitions of the fit statistics, see "Model Summary Table".

**Caution:** If you do not specify a holdback set and fit a combination of state space smoothing models and ARIMA models, a caution message is shown at the top of the Model Comparison table. In this case, the state space smoothing models are not comparable to the ARIMA models because likelihood values and likelihood-based information criteria, such as the AIC statistic, are not comparable between the two different model classes. However, if you specify a holdback set, the models can be compared because the RMSE statistic is used.

**Figure 18.12** 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** check box. 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** check box.

**Figure 18.13** Model Plots
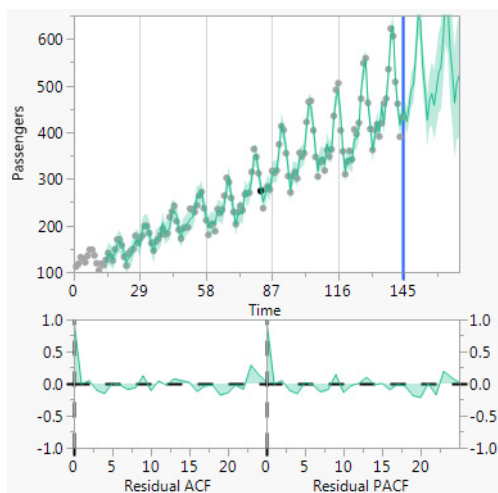


## Model Comparison Report Options

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**   (Not available if a holdback set is specified.) Provides one simulation of the model to the upper end of the time axis. The simulation is shown on the Model Comparison time series plot.

**Simulate More**   (Not available if a holdback set is specified.) Provides the specified number of simulations of the model to the upper end of the time axis. The simulations are shown on the Model Comparison time series plot.

**Remove Model Simulation**   (Not available if a holdback set is specified.) Removes the simulations for the model.

**Remove All Simulation**   (Not available if a holdback set is specified.) Removes the simulations for all models.

**Generate Simulation**   (Not available if a holdback set is specified.) 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**   (Not available if a holdback set is specified.) Specifies the seed for generating the simulated trajectories.

**Remove Fit**   Removes the corresponding model from the report window.

**Remove All Models**   Removes all models that are listed in the Model Comparison table from the report window.

**Remove Unselected**   Removes all models that do not have the Report check box selected from the report window.

**Remove Selected**   Removes all models that have the Report check box selected from the report window.

**Hide All Reports**   Hides all of the models that are listed in the Model Comparison table from the report window. This is equivalent to deselecting all of the Report check boxes.

**Hide All Graphs**   Hides all of the models that are listed in the Model Comparison table from the Model Plots. This is equivalent to deselecting all of the Graph check boxes.

---

**Note:** The Time Series platform treats excluded rows as missing values. If the last several observations in the data table are excluded or missing, the simulation options in the Model Comparison Report will not work properly. The simulation procedure uses the last several observations of the time series. If the last several observations are missing, the simulation produces all missing values into the future.

---

# ARIMA and Seasonal ARIMA 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. When the Report check box 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**    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 Innovations**    The sum of the squared innovations.

**Sum of Squared Residuals**    The sum of the squared residuals.

**Variance Estimate**    The sum of squared innovations divided by the number of degrees of freedom ($n - k$). This is the sample estimate of the variance of the random shocks $a_t$, described in the section "ARIMA Model".

**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 'A' Information Criterion**    The AIC value, computed as $-2\text{loglikelihood} + 2k$. Smaller AIC values indicate better fit.

**Schwarz's Bayesian Criterion**    The SBC value, computed as $-2\text{loglikelihood} + k*\ln(n)$. Smaller SBC values indicate better fit. Schwarz's Bayesian Criterion is equivalent to the Bayesian Information Criterion (BIC).

**RSquare**    The R-Square value, computed as follows:

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

where

$$\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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

$\hat{y}_i$   are the one-step-ahead forecasts

$\bar{y}$   is the mean

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 $R^2$ value, computed as follows:

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

**MAPE**   The Mean Absolute Percentage Error value, 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 value, computed as follows:

$$\frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|$$

**–2LogLikelihood**   Twice the negative log-likelihood function evaluated at the best-fit parameter estimates. Smaller values are better fits. See *Fitting Linear Models*.

**Stable**   Indicates whether the autoregressive operator is stable. That is, whether all the roots of $\phi(z) = 0$ lie outside the unit circle.

**Invertible**   Indicates whether the moving average operator is invertible. That is, whether all the roots of $\theta(z) = 0$ lie outside the unit circle.

---

**Note:** The $\phi$ and $\theta$ operators are defined in the section "ARIMA Model".

---

**Parameter Estimates Table**   Shows 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 a *t*-ratio greater (in absolute value) than the computed value, given a true hypothesis.

**Constant Estimate**   (Shown only 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**   A plot that shows both the observed and predicted values for the time series. The plot is divided by a vertical line into two regions.

If you specified a holdback set, a vertical line separates the training data from the holdback data. To the left of the vertical line, the one-step-ahead forecasts are overlaid with the

training data points. To the right of the vertical line, the values forecast by the model are overlaid with the holdback data points. The prediction intervals for the forecast are also shown. You can control the number of forecast values using the Forecast Periods option in the platform launch window.

If you did not specify a holdback set, the vertical line separates the observed data from the future forecasts. 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, the future values forecast by the model and the prediction intervals for the forecast are shown. You can control the number of future forecast values using the Forecast Periods option in the platform launch window or by selecting Number of Forecast Periods from the Time Series red triangle menu.

**Residuals**   A graph that shows 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.

**Forecasting Errors on Holdback Series**   (Available only if a holdback set is specified.) A plot of the forecasting errors on the observations in the holdback set. The autocorrelation and partial autocorrelation reports for these errors are also shown.

**Iteration History**   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.

## ARIMA and Seasonal ARIMA Model Report Options

Each ARIMA and Seasonal ARIMA 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**   (Not available if a holdback set is specified.) 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". However, they are applied to the series of residuals.

**Remove Fit**   Removes the individual model fit from the report.

## State Space Smoothing Report

Each State Space Smoothing model is added to the Model Comparison table. If the Report check box for a state space smoothing model in the Model Comparison table is selected, a State Space Smoothing report appears. Each State Space Smoothing report has a unique outline name that identifies the model. The following reports are included:

**Model Type**   Describes the best fitting model for the series in terms of error, trend, and seasonality.

**Model Summary**   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.

**–2LogLikelihood**   Twice the negative log-likelihood function evaluated at the best-fit parameter estimates. Smaller values are better fits. See *Fitting Linear Models*.

**AIC**   The Akaike's Information Criterion (AIC) value, computed as $-2$loglikelihood $+ 2k$. Smaller AIC values indicate better fit.

**BIC**   The Schwarz's Bayesian Criterion (SBC) value, computed as $-2$loglikelihood $+ k*\ln(n)$. Smaller SBC values indicate better fit. Schwarz's Bayesian Criterion is equivalent to the Bayesian Information Criterion (BIC).

**Nparm**   The number of parameters in the calculation of the information criteria. This number is one more than the number of free parameters in the likelihood function.

**Note:** Nparm is not the number of parameters in the parameter estimates table.

**Sigma**   The estimate of the one-step-ahead forecast standard deviation.

**RSquare**   The R-Square value, computed as follows:

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

where

$$\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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

$\hat{y}_i$   are the one-step-ahead forecasts

$\bar{y}$   is the mean

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 $R^2$ value, computed as follows:

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

**MAPE**   The Mean Absolute Percentage Error value, computed as follows:

$$\frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

where

$\hat{y}_i$   are the one-step-ahead forecasts

**MAE**   The Mean Absolute Error value, computed as follows:

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

where

$\hat{y}_i$   are the one-step-ahead forecasts

**Parameter Estimates**   A table that shows the estimate, standard error, and 95% confidence interval for each parameter in the model.

**Forecast**   A plot that shows both the observed and predicted values for the time series. See "Forecast".

**One-Step-Ahead Forecasting Errors**   A plot of the one-step-ahead forecasting errors over time. For each time point $t$, the error is calculated as the observed value of the time series at $t$ minus the one-step-ahead predicted value at $t$. If a holdback set is specified, the errors are computed only for the training data. The autocorrelation and partial autocorrelation reports for these errors are also shown.

**Forecasting Errors on Holdback Series**   (Available only if a holdback set is specified.) A plot of the forecasting errors on the observations in the holdback set. The autocorrelation and partial autocorrelation reports for these errors are also shown.

**Component States**   A report that contains a plot for each state of the model and the corresponding formula for the model. Each plot includes the state estimates of the series and the state forecasts. If a holdback set is specified, the state estimates are for the training portion and the state forecasts are for the holdback portion. If a holdback set is not specified, the state estimates are for the portion up to the last observation of the time series and the state forecasts are for after the last observation of the time series. A model can have up to three states: Level, Trend, and Seasonal.

The Component States red triangle menu has a Save States option. Select this option to save the component states at each time point to a separate data table.

**One-Step-Ahead Relative Forecasting Errors for Multiplicative Error Model**   (Available only for multiplicative error models.) A plot of the one-step-ahead relative forecasting errors over time. For each time point $t$, the error is calculated as the one-step-ahead-forecasting error at $t$ divided by the one-step-ahead predicted value at $t$.

## State Space Smoothing Report Options

Each state space smoothing 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.

**Remove Fit**    Removes the individual model fit from the report.

# Transfer Function Report

Each transfer function model is added to the Model Comparison table. If the Report check box for a transfer function model in the Model Comparison table is selected, a Transfer Function Model report is produced. If a holdback set is specified, a model plot is also available in the Model Comparison report.

Each Transfer Function Model report contains the following reports:

- Model Summary
- Parameter Estimates
- Residuals
- Forecasting Errors on Holdback Series
- Interactive Forecasting
- Iteration History

The information in the Model Summary, Parameter Estimates, Residuals, Forecasting Errors on Holdback Series, and Iteration History reports is the same as in the ARIMA and Seasonal ARIMA model report. For more information about these reports, see "ARIMA and Seasonal ARIMA Model Report". 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 prediction interval specified in the Transfer Function Specification window. 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 18.14** Interactive Forecasting Graph
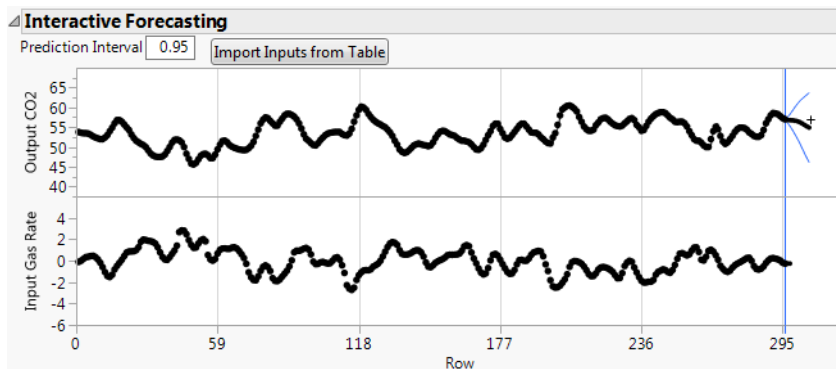


## Spectral Density Report

**Figure 18.15** Spectral Density Plots and White Noise Test Report
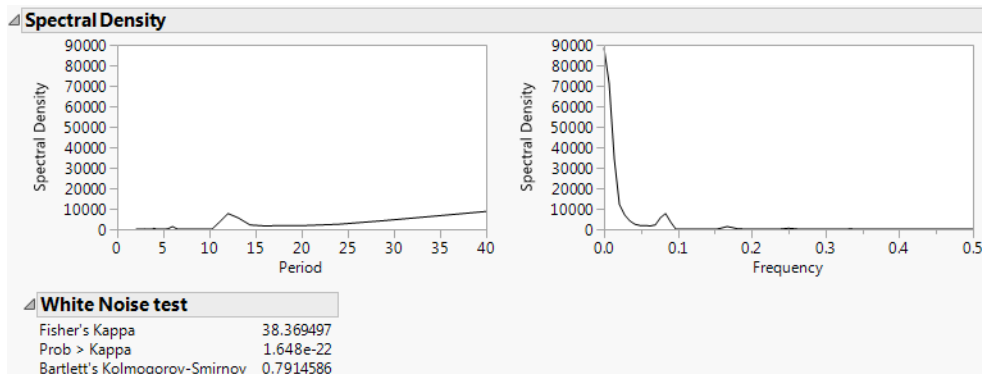


The White Noise Test report contains the following statistics:

**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

$\kappa$ is the observed value of Kappa

The null hypothesis is rejected if this probability is less than the significance level $\alpha$.

**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 Examples of the Time Series Platform

## Example of Creating Time ID Column

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 **Best > Date > yyyyQq** next to Format.

4.  Select **Column Properties > Formula**.
5.  Click the gray triangle next to Date Time and click **Informat**.
6.  Click the gray triangle next to Character.
7.  Select Year, click **Char,** and then click **Concat**.
8.  Type "Q" in the box (including quotation marks) and press Enter.
9.  Click **Concat**.
10. Select Quarter and click **Char**.
11. Click the <formatName> box, type yyyyQq, and press Enter.
12. Click **OK**.

The completed New Column window should appear as in Figure 18.16.

**Figure 18.16**  New Column



13. Click **OK**.

---

**Note:** This time column also works for X11 analyses.

---

## 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 18.17** Time Series Report for SeriesP.jmp
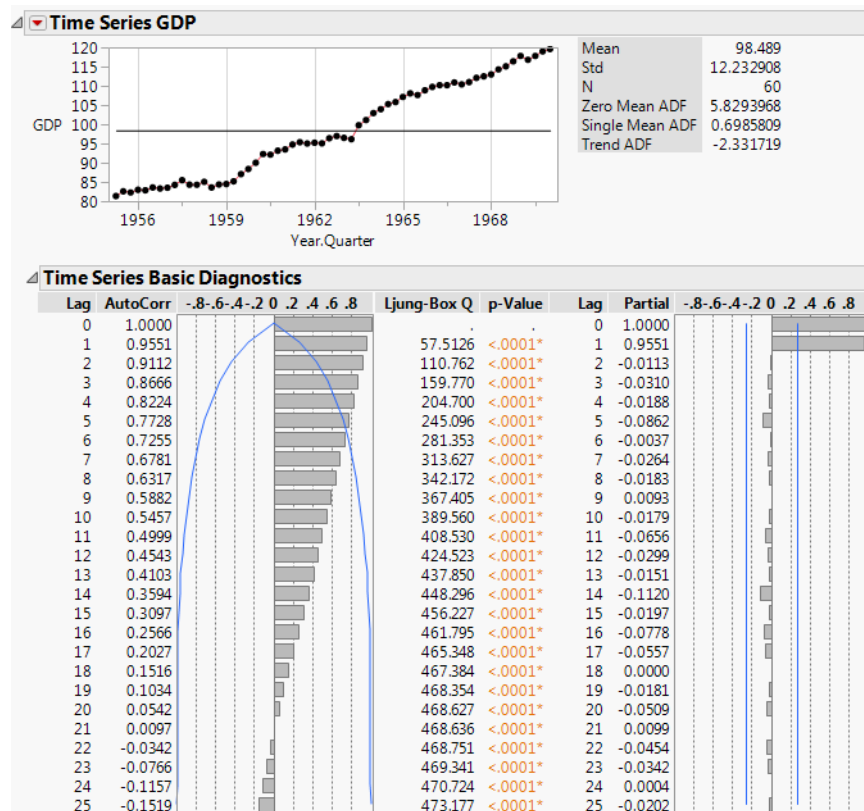


The series shows an increasing trend over time that is fairly linear. In addition, the autocorrelation chart shows that there is strong correlation between points that are close together. The AutoCorr values for points with lags of 1, 2, and 3 are 0.9551, 0.9112, 0.8666, respectively.

5.  Click the Time Series GDP red triangle and select **Difference**.

6.  Select 1 for the Nonseasonal Differencing Order and click **Estimate**.

**Figure 18.18**  Difference Report for SeriesP.jmp
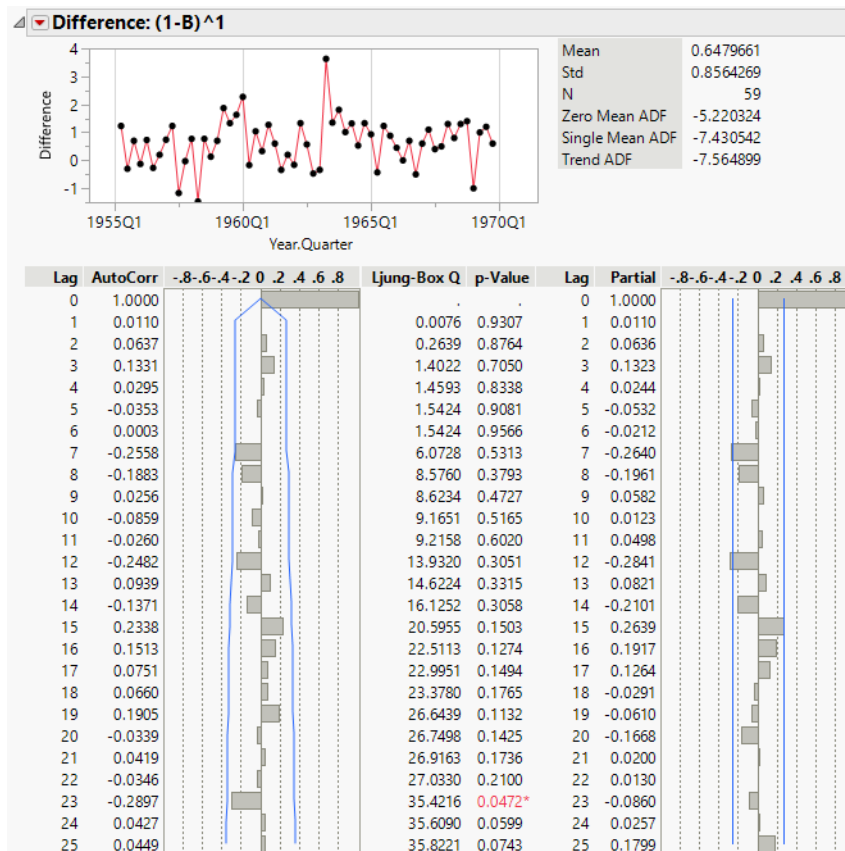


The Difference report helps determine an appropriate model to be fit to the original time series. The plot of differences shows that the differenced series no longer has the trend that was observed in the original data. This indicates that lag-1 differencing is an appropriate choice. Also, even after 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 four total models.

**Figure 18.19** ARIMA Model Group Specification



11. Click **Estimate**.

**Figure 18.20** Model Comparison Table



| 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 18.21** Model Report for ARIMA(0,1,0)
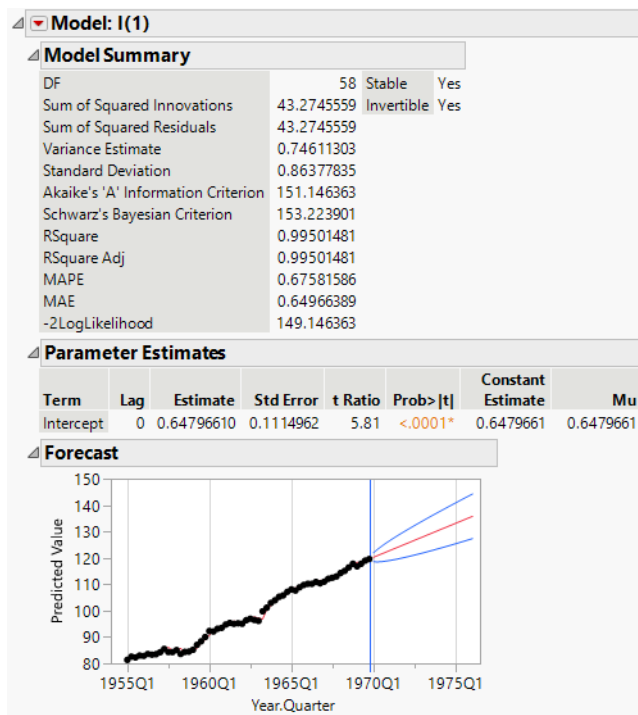


The model report for I(1) shows the forecast graph. The blue lines indicate the prediction intervals. GDP is predicted to continue increasing at a linear rate.

## Example Using Box-Cox Transformation

In this example, you use a Box-Cox transformation to analyze the Passengers column in the Seriesg.jmp data table. The Lambda value is set to zero, which makes the Box-Cox transformation an approximate log transformation. The script in the Seriesg.jmp data table runs a time series analysis on the Log Passengers column. By using the Box-Cox transformation for the analysis, the results are easier to interpret since all forecasts are transformed back and reported on the original scale.

1. Select **Help > Sample Data Library** and open Time Series/Seriesg.jmp.

2. Select **Analyze > Specialized Modeling > Time Series**.

3. Select Passengers and click **Y, Time Series**.

4. Select Time and click **X, Time ID**.
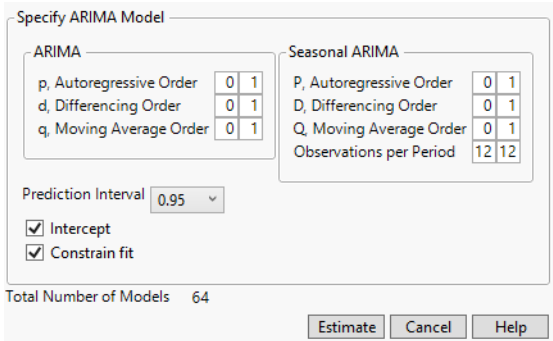
5. Select **Use Box-Cox Transformation**.

   Use the default value of 0 for Lambda.

6.  Click **OK**.

7.  Click the red triangle next to Time Series Box-Cox Transformed Passengers and select
    **Seasonal ARIMA**.

8.  In the ARIMA section of the Seasonal ARIMA Specification window, set the following
    parameters.

    –   Set $p$, Autoregressive Order to 0.

    –   Set $d$, Differencing Order to 1

    –   Set $q$, Moving Average Order to 1

9.  In the Seasonal ARIMA section of the Seasonal ARIMA Specification window, set the
    following parameters.

    –   Set $P$, Autoregressive Order to 0

    –   Set $D$, Differencing Order to 1

    –   Set $Q$, Moving Average Order to 1

10. Deselect Intercept to run a model without an intercept.

    Your window should now match the window shown in Figure 18.22.

**Figure 18.22**  Seasonal ARIMA Specifications



11. Click **Estimate**.

**Figure 18.23** Seasonal ARIMA Model for Box-Cox Transformed Passengers Data
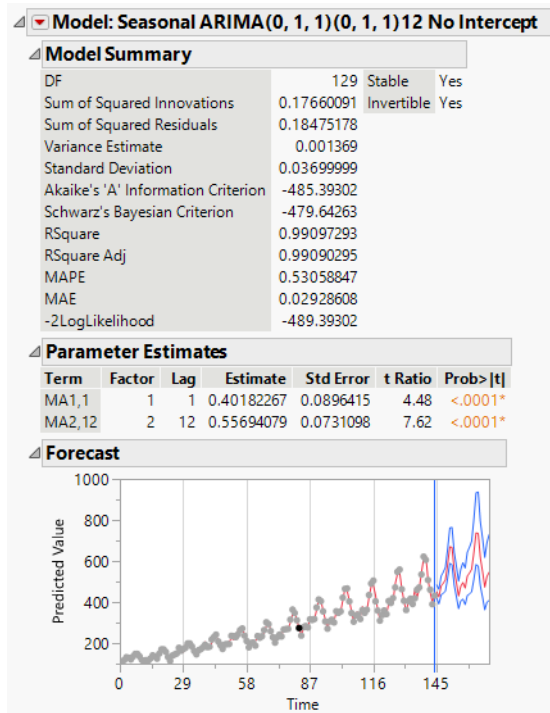


The parameter estimates are equivalent to those obtained from running the same model on the Log Passengers column. However, the forecasts and prediction intervals in Figure 18.23 are on the original scale.

## Example Using a Holdback Set

This example uses a holdback set to evaluate predictions made on monthly data on lead production. Since a holdback set is used, both State Space Smoothing and ARIMA models can be fit and compared to find the best fitting model for the data.

1. Select **Help > Sample Data Library** and open Time Series/Lead Production.jmp.

2. Select **Analyze > Specialized Modeling > Time Series**.

3. Select LEADPROD and click **Y, Time Series**.

4. Select DATE and click **X, Time ID**.

5. Select **Forecast on Holdback**.

   Selecting this options means that forecasts are made on the holdback set instead of on future observations.

6. Enter 12 next to Forecast Periods.

This assigns the last 12 observations in the data to the holdback set. Since this is monthly data, this is equivalent to the last year.

7. Click **OK**.

8. Click the Time Series LEADPROD red triangle and select **State Space Smoothing**.

9. Type 12 next to Period.

10. Select Constrain Parameters.

11. Click **OK**.

12. Click the Time Series LEADPROD red triangle and select **ARIMA Model Group**.

13. Set the range for all parameters from 0 to 1. The Specify ARIMA Model window looks like the one in Figure 18.24.

**Figure 18.24** ARIMA Model Group Specifications



14. Click **Estimate**.

**Figure 18.25** Model Comparison



| | | | | Holdback Evaluation | | | |
|---|---|---|---|---|---|---|---|
| Report | Graph | Model Class | Model | RMSE ^ | MSE | MAPE | MAE |
| | | ARIMA | Seasonal ARIMA(0, 1, 1)(1, 0, 0)12 | 3219.4131 | 10364621 | 8.900164 | 2416.3319 |
| | | ARIMA | Seasonal ARIMA(0, 1, 1)(0, 0, 1)12 | 3316.8746 | 11001657 | 9.148390 | 2456.0873 |
| | | State Space Smoothing | AAA12 Constrained | 3350.9303 | 11228734 | 8.623519 | 2245.5657 |
| | | ARIMA | Seasonal ARIMA(1, 1, 0)(1, 0, 0)12 | 3397.0855 | 11540190 | 9.082427 | 2392.5823 |
| | | ARIMA | Seasonal ARIMA(0, 1, 1)(1, 1, 0)12 | 3448.6372 | 11893099 | 9.829459 | 2649.8142 |
| | | ARIMA | Seasonal ARIMA(1, 1, 0)(1, 1, 0)12 | 3484.3764 | 12140879 | 9.708056 | 2604.6290 |
| | | ARIMA | Seasonal ARIMA(1, 1, 0)(0, 0, 1)12 | 3491.2648 | 12188930 | 9.302681 | 2435.7441 |
| | | State Space Smoothing | MAM12 Constrained | 3512.1513 | 12335206 | 10.552196 | 2813.4655 |
| | | ARIMA | Seasonal ARIMA(0, 1, 0)(1, 0, 0)12 | 3525.0858 | 12426230 | 9.374006 | 2440.6892 |
| | | ARIMA | Seasonal ARIMA(0, 1, 0)(1, 1, 0)12 | 3551.3973 | 12612423 | 9.367947 | 2480.3210 |
| | | ARIMA | Seasonal ARIMA(0, 1, 1)(1, 0, 1)12 | 3579.0242 | 12809414 | 10.520274 | 2739.9652 |
| | | ARIMA | Seasonal ARIMA(0, 1, 0)(0, 1, 0)12 | 3597.3098 | 12940638 | 11.622868 | 3219.0476 |
| | | ARIMA | Seasonal ARIMA(0, 1, 0)(0, 0, 1)12 | 3631.8176 | 13190099 | 9.792052 | 2544.4269 |
| | | ARIMA | Seasonal ARIMA(1, 1, 0)(0, 1, 0)12 | 3661.8395 | 13409068 | 12.030672 | 3366.0935 |
| | | ARIMA | Seasonal ARIMA(0, 1, 1)(0, 1, 0)12 | 3673.6369 | 13495608 | 12.063523 | 3379.4644 |
| | | ARIMA | Seasonal ARIMA(1, 1, 0)(1, 0, 1)12 | 3700.6621 | 13694900 | 11.077791 | 2890.6426 |
| | | ARIMA | Seasonal ARIMA(0, 1, 0)(1, 0, 1)12 | 3723.7664 | 13866437 | 11.222827 | 2935.8075 |

The fit statistics are comparable between the State Space Smoothing models and the ARIMA models because they are evaluated on the holdback set.

15. Select the Graph check box next to the first three models in the Model Comparison report.

**Figure 18.26** Model Comparison Graph
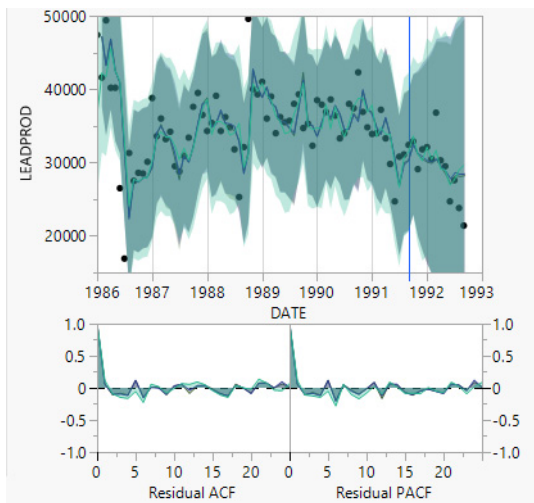


The evaluation statistics and the forecasting plots show that there is no substantial differences in forecasting performance among the top models. Based on the report, either an ARIMA model or a State Space Smoothing model is appropriate for this data.

# Statistical Details for the Time Series Platform

- "Statistical Details for Spectral Density"
- "Statistical Details for X-11 Decomposition"
- "Statistical Details for Exponential Smoothing Models"
- "Statistical Details for ARIMA Models"
- "Statistical Details for Transfer Functions"

# 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 $I(f_i) = (N/2)(a_i^2 + b_i^2)$, which represents the intensity at frequency $f_i$.

The periodogram is then smoothed and scaled by $1/(4\pi)$ 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 15.2 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 Exponential Smoothing Models

Smoothing models are defined as follows:

$$y_t = \mu_t + \beta_t t + s(t) + a_t$$

where

$\mu_t$ is the time-varying mean term

$\beta_t$ is the time-varying slope term

$s(t)$ is one of the s time-varying seasonal terms

$a_t$ are the random shocks

Models without a trend have $\beta_t = 0$ and nonseasonal models have $s(t) = 0$. The estimators for these time-varying terms are defined as follows:

$L_t$ is a smoothed level that estimates $\mu_t$

$T_t$ is a smoothed trend that estimates $\beta_t$

$S_{t-j}$ for $j = 0, 1,..., s - 1$ 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*:

$\alpha$ is the level smoothing weight

$\gamma$ is the trend smoothing weight

$\varphi$ is the trend damping weight

$\delta$ 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. For more information on smoothing weights, see *SAS/ETS 15.2 User's Guide* (the Forecasting Process Details chapter).

## Simple Exponential Smoothing

The model for simple exponential smoothing is $y_t = \mu_t + \alpha_t$.

The smoothing equation, $L_t = \alpha y_t + (1 - \alpha)L_{t-1}$, is defined in terms of a single smoothing weight $\alpha$. This model is equivalent to an ARIMA(0, 1, 1) model where the following is true:

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

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 $\alpha$, 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:

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

The moving average form of the model is 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 $\alpha$ and $\gamma$, 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:

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

The moving average form of the model is 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 $\alpha$, $\gamma$, and $\varphi$, 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:

$$(1 - \varphi B)(1 - B) y_t = (1 - \theta_1 B - \theta_2 B^2) a_t$$

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 $\alpha$ and $\delta$ 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:

$$y_t = a_t + \sum_{j=1}^{\infty} \psi_j a_{t-j} \quad \text{where} \quad \psi = \begin{cases} \alpha \text{ for } j \bmod s \neq 0 \\ \alpha + \delta(1-\alpha) \text{ for } j \bmod s = 0 \end{cases}$$

## Winters Method (Additive)

The model for the additive version of the Winters method is $y_t = \mu_t + \beta_t t + s(t) + a_t$.

The smoothing equations in terms of weights $\alpha$, $\gamma$, and $\delta$ are 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:

$$(1 - B)(1 - B^2)y_t = \left(1 - \sum_{i=1}^{s+1} \theta_i B^i\right) a_t$$

The moving average form of the model is 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 defined as follows:

$$\phi(B)(w_t - \mu) = \theta(B)a_t$$

where

$t$ is the time index

**B** is the backshift operator defined as B$y_t = y_{t-1}$

$w_t = (1 - B)^d y_t$ is the response series after differencing

$\mu$ is the intercept or mean term

$\phi(B)$ and $\theta(B)$ are the autoregressive operator and the moving average operator, respectively, and are defined as follows:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$$

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:

$$\phi(B)w_t = \delta + \theta(B)a_t$$

The constant estimate $\delta$ is given by the relation:

$$\delta = \phi(B)\mu = \mu - \phi_1\mu - \phi_2\mu - \ldots - \phi_p\mu$$

### 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:

$$w_t = (1-B)^d(1-B^s)^D y_t$$

$$\varphi(B) = (1 - \varphi_{1,1}B - \varphi_{1,2}B^2 - \ldots - \varphi_{1,p}B^p)(1 - \varphi_{2,s}B^s - \varphi_{2,2s}B^{2s} - \ldots - \varphi_{2,Ps}B^{Ps})$$

$$\theta(B) = (1 - \theta_{1,1}B - \theta_{1,2}B^2 - \ldots - \theta_{1,q}B^q)(1 - \theta_{2,s}B^s - \theta_{2,2s}B^{2s} - \ldots - \theta_{2,Qs}B^{Qs})$$

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

$X_{1, t-d1}$ indicates the series $X_1$ is indexed by $t$ with a $d1$-step lag

$\mu$ represents the mean level of the model

$\varphi(B)$ and $\theta(B)$ represent autoregressive and moving average polynomials from an ARIMA model

$\omega_k(B)$ and $\delta_k(B)$ represent numerator and denominator factors (or polynomials) for individual transfer functions, with $k$ representing an index for the 1 to $m$ individual inputs.

Each polynomial in the above model can contain two parts, either nonseasonal, seasonal, or a product of the two as in seasonal ARIMA. When specifying a model, leave the default 0 for any part that you do not want to include in the model.
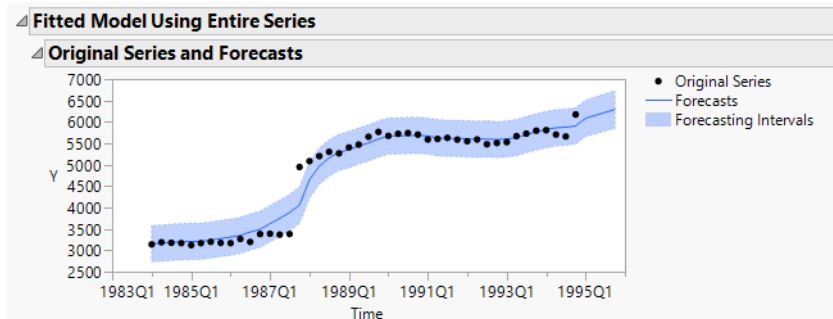
# Time Series Forecast

## Forecast Multiple Time Series

The Time Series Forecast platform enables you to forecast multiple time series, where the number of series could be very large. The platform provides compact summaries of the multiple time series and provides options for efficiently fitting up to 30 different forecasting models in the exponential smoothing family. The best fitting model is automatically selected.

The Time Series Forecast platform is designed to select a best fitting model automatically from a set of exponential smoothing models. This is in contrast to the Time Series platform which requires manual selection of a best fitting model and uses ARIMA models.

**Figure 19.1** Original Series and Forecasts

# Contents

# Example of Time Series Forecast

1. Select **Help > Sample Data Library** and open Time Series/M3C Quarterly.jmp.

2. Select **Analyze > Specialized Modeling > Time Series Forecast**.

3. Select Y and click **Y**.

4. Select Series and click **Grouping**.

5. Select Time and click **Time**.

6. Click **OK**.

   The initial Time Series Forecast report appears. It contains summaries of the Time and Y variables. There is also a Modeling Specifications report that provides options for fitting forecast models.

**Figure 19.2** Initial Report for Time Series Forecast



7. In the Model Specifications report, click the Complete Specifications tab.

8. Click the menu below Model Selection Strategy and select Forecasting Performance.

9. Click **Run**.

   This fits a set of recommended models to the data. The best model for each level of the grouping variable is chosen and reported in the Model Reports. See "Model Reports".

**Figure 19.3** Model Reports
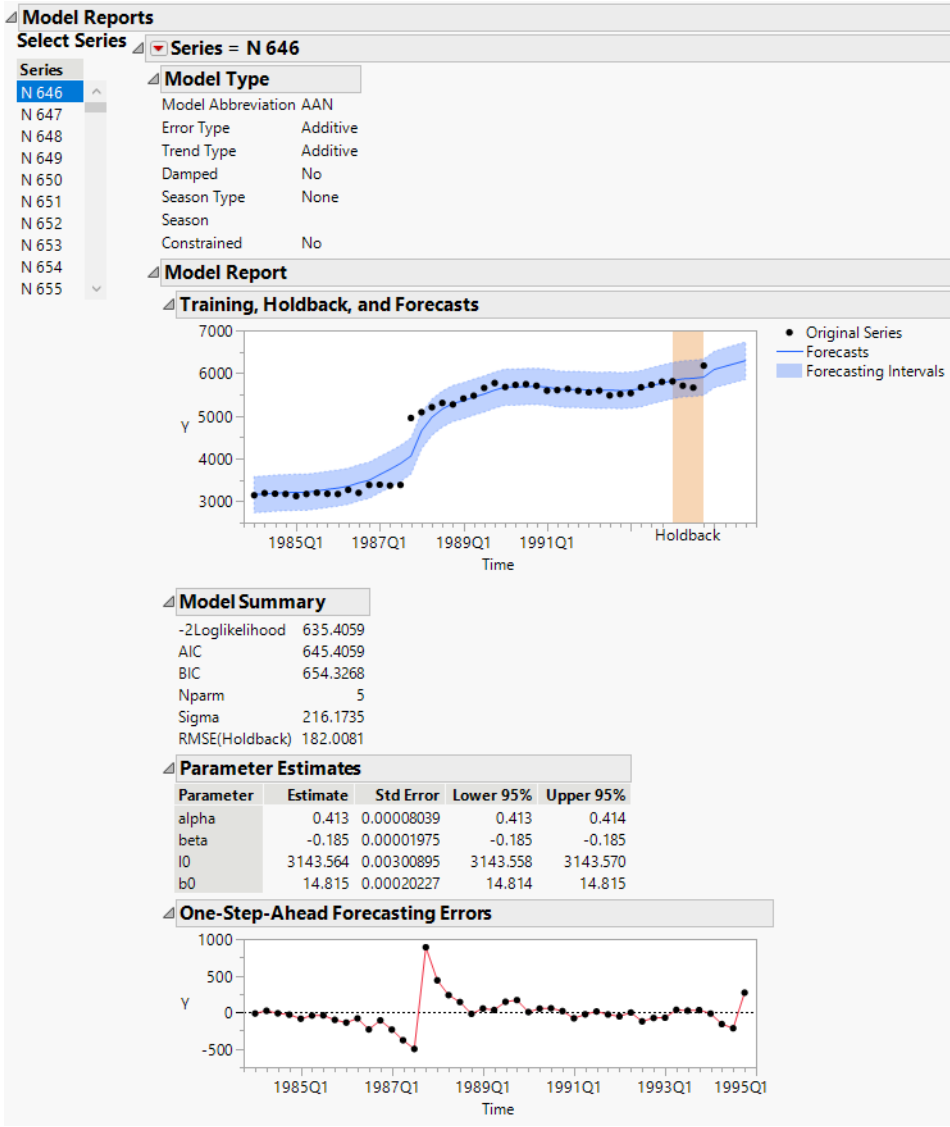


The model report for one series, Series N646, is shown. This corresponds to the series name that is selected in the Select Series list. You can use the down arrow key to scan through the model reports for each series. Alternatively, you can click a different name in the list to display the model report for the selected series.

# Launch the Time Series Forecast Platform

Launch the Time Series Forecast platform by selection **Analyze > Specialized Modeling > Time Series Forecast**.

**Figure 19.4** The Time Series Forecast Launch Window
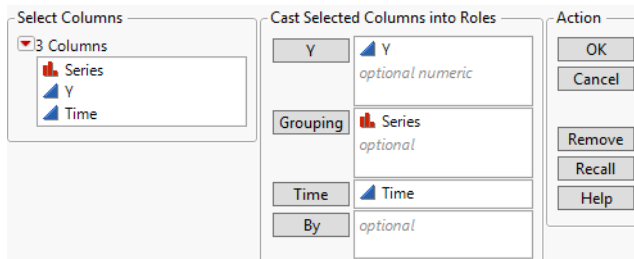


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

The Time Series Forecast platform launch window contains the following options:

**Y**   Assigns one or more columns as time series variables. Multiple time series can be organized as individual columns in a table or they can be stacked in a single column with one or more grouping variables.

**Grouping**   Assigns one or more columns as grouping variables. Grouping variables are used to identify individual time series when they are in stacked format.

**Time**   Assigns a column for the time variable. If the individual time series are organized as individual columns, the timestamps in the Time column must not contain any duplicates.

**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.

## Data Format

Multiple time series can be organized as individual columns in a table or they can be stacked in a single column with one or more grouping columns to identify the individual series. You can also have a combination of multiple Y columns and multiple grouping columns.

The platform makes the following assumptions about the data:

- All of the time series are recorded at the same time resolution.

- All of the time series are collected up to the same final observed time point. This final observed time point is defined as the latest time point that occurs in any of the individual time series. A series without data up to the final observed time point is assumed to have missing values. For example, in Figure 19.5, the final observed time point is 5, represented by the solid black line. It is assumed that Series 2 has missing observations at time points 4 and 5 and that Series 4 has a missing observation at time point 5.

- All of the forecasts start at the same time point. This starting time point is the next time point following the final observed time point. For example, in Figure 19.5, the final observed time point is 5, represented by the solid black line. Therefore, all of the forecasts start at time point 6, represented by the dashed black line.

**Figure 19.5** Demonstration of Assumptions
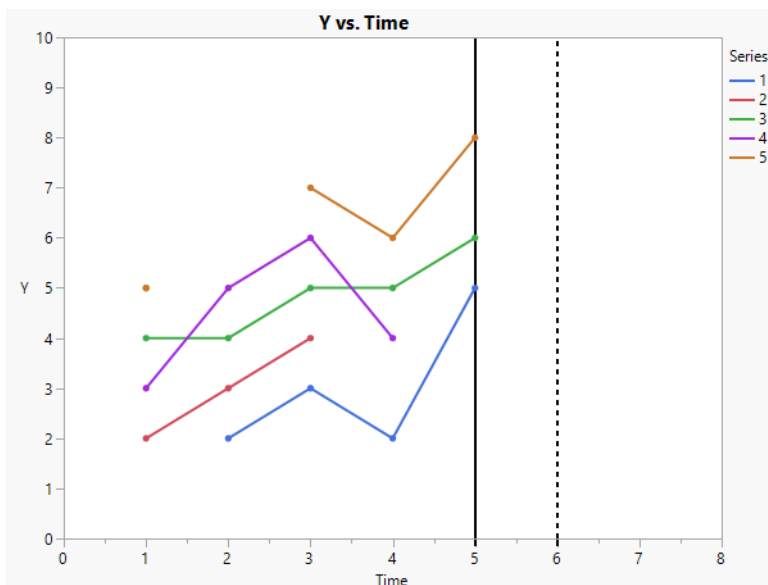
# The Time Series Forecast Report

The Time Series Forecast report contains summary tables based on the variables specified in the launch window. A summary table of the time series by group is always provided. If a Time column is specified, the report also contains a summary of the time pattern and a summary table of the time by group. The report also contains a Modeling Specifications report that enables you to fit models to each time series.

- "Analysis of Time Pattern"
- "Group Summary of Time"
- "Group Summary of Y"
- "Modeling Specifications"

## Analysis of Time Pattern

If a time column is specified, the Time Series Forecast platform analyzes the entries of the column to determine a pattern. The platform is designed to recognize a variety of time patterns, including daily, business daily, monthly, bi-weekly, quarterly, and so on. The pattern is reported as the Date Resolution. A Suggested Seasonality is also provided based on the time pattern.

## Group Summary of Time

If a time column is specified, the report contains summaries of the individual time series. The summaries contain the following information for each time series:

**\<Grouping Column Name\>**   (Available only if a Grouping variable is specified.) The level of the grouping variable that defines the individual series when data is provided in the stacked format.

**N Rows**   The number of observations in the series.

**Missing Value**   Indicates whether any of the time values are missing values.

**Unique Timestamps**   Indicates whether the time stamps are unique.

**Start**   The starting time stamp.

**End**   The ending time stamp.

**Missing Record**   Indicates whether there are observations missing from the data table, based on the starting and ending time stamps and the data resolution/pattern.

# Group Summary of Y

The report contains a summary table for each Y variable. The summaries contain the following information for each time series:

**Variable**　(Available only if there are multiple Y columns and at least one grouping column.) The column name of the time series variables.

**Time Series**　(Available only if there are multiple series.) The name of the individual time series. When the data are provided in the stacked format, this is the level of the grouping variable. When the data are provided in the wide format, with multiple columns, this is the column name of the individual series.

> **Note:** When the data are provided in the stacked format, the column heading in the table is the name of the Grouping column.

**Min**　The minimum value of Y for the individual series.

**Max**　The maximum value of Y for the individual series.

**Length**　The duration of the series. This is determined based on the starting and ending time stamps and the result of the Analysis of Time Pattern.

**N Obs**　The number of observations. If there are no missing records, this number is the same as the Length.

**N Missing**　The number of missing Y values.

**All Positive**　Indicates whether all of the Y values are positive.

**Comment**　Indicates if there is a problem with the series, such as non-unique time stamps.

# Modeling Specifications

The Modeling Specifications report enables you to fit models to the individual time series. A variety of models are considered. Hyndman et al. (2008) defines state space smoothing models based on their error, trend component, and seasonal component:

- The errors can be additive (A) or multiplicative (M).
- The trend component can be none (N), additive (A), additive damped ($A_d$), multiplicative (M), or multiplicative damped ($M_d$).
- The seasonal component can be none (N), additive (A), or multiplicative (M).

A specific model can be represented by its ETS (Error, Trend, Seasonal). These are the models used in Time Series Forecasting.

There are two tabs in the Modeling Specifications report. Use the Recommended Specifications tab to fit a set of state space smoothing models that is chosen by the platform. Use the Complete Specifications tab to select the specific state space smoothing models that you want to fit. For each individual time series, the best fitting model from the given set is then used for forecasting.

### Select Models

Available only in the Complete Specifications tab, shown in "Complete Specifications Tab". The Select Models report enables you to specify the state space smoothing models to fit to the individual time series. Use the check boxes to select the error, trend, and seasonality for the models. Click **Select Recommended** to select the check boxes that correspond to the models recommended by the platform. Click **Select All** to select all check boxes or **Deselect All** to deselect all check boxes. Click **Constrain Parameters** to constrain the parameters in such a way that the further an observation is from the present, the less effect it has on the present state of the model. In State Space Smoothing models, a forecast at time $t$, given all previous observations, is the same as the weighted sum of all observations up to time $t$. The weights are a function of the parameters. Therefore, constraining the parameters ensures that the weights for past observations go to zero and that the further an observation is from the present, the faster the weight goes to zero.

**Figure 19.6** Complete Specifications Tab

### Forecast Settings

Enables you to specify the following optional settings:

**NAhead**   Specifies the number of steps ahead to forecast. The number of steps must be nonnegative.

**Period**   Specifies seasonality values to be considered in the model fitting process. By default, the period is set to the suggested seasonality from the Analysis of Time Pattern report. If you want to consider models with different seasonality, specify the additional periods here and separate them with a comma. The period must be greater than zero.

### Model Selection Strategy

Enables you to specify how the best model is chosen.

**Information Criteria**   Determines the best model based on the specified information criteria. The available information criteria are AIC and BIC.

---

**Tip:** Use BIC if the time series is long.

---

**Forecasting Performance**   Determines the best model based on a performance selection algorithm that involves a holdback set. First, the time series is partitioned into a training set and a holdback set. The value of NHoldback specifies the number of observations in the holdback set. Then, the algorithm fits all recommended or user-specified models on the data in the training set. Forecasts are made on the holdback set using the individual fitted models. These forecasts are compared to the actual holdback observations and the models are evaluated using the specified metric. The model with the best metric is selected for the final model. Last, the selected final model is fit to the entire time series (training set and holdback set) and the refitted model is used to forecast beyond the last observation of the time series.

**Metric**   Specifies the metric to evaluate the forecasts made by the individual model fits. The available metrics are Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE).

**NHoldback**   Specifies the number of observations used in the holdback set.

### Other Options

Enables you to specify the following options:

**Preserve Model Selection Criterion**   Saves the model selection criterion for the set of models that is fit to the data. After model fitting, this information is shown in the Information Criterion of All Models for All Series report.

**Forecast Interval Level**　The prediction interval level for the forecasts. This changes the width of the shaded area in the forecasting plots.

**Imputation for Applicable Models**　Specifies the imputation method used during the model fitting process.

---

**Note:** These options change only the data that are used for model fitting. The raw data are not changed.

---

**None**　Does not impute missing values.

**Last Value**　Imputes missing values by using the last value that is available before a sequence of missing observations.

**Middle Value**　Imputes missing values by averaging the last value that is available before a sequence of missing observations and the first value that is available after a sequence of missing observations.

**Linear Interpolation**　Imputes missing values by creating a linear interpolation between the last value that is available before a sequence of missing observations and the first value that is available after a sequence of missing observations.

When you click **Run**, the specified set of models is fit. A progress bar appears that reports the number of active threads, the number of finished and total tasks, and the percentage of finished tasks. Note that a task is defined as the fitting of all specified models for one series and response variable combination. When the fitting process is complete, the Model Reports report is shown. See "Model Reports". If you make any changes to the options in the Model Specifications report and click **Run** again, the Model Reports report is replaced with a new report.

# Time Series Forecast Platform Options

The Time Series Forecast red triangle menu contains the following options:

**Report Options**　(Effective only after fitting a model.) Contains the following report options:

**Set Forecast Interval Level**　Enables you to change the forecast interval level. The individual model reports update automatically.

**Set NAhead**　Enables you to change the number of steps ahead to forecast. The individual model reports update automatically.

**Set NHoldout**　Enables you to change the duration of the holdout set. The individual model reports update automatically.

**Save Results**   (Effective only after fitting a model.) Saves the forecasts to a separate data table named Forecasts. The Forecasts data table has the same layout and format as the original data table. If a holdout set is specified by NHoldout, an additional data table that saves the actual values from the holdout data and the forecasts is produced. You can use this table to calculate different types of error estimates.

The following additional save options are shown in the Request Outputs window:

**Save Forecast Intervals**   Saves the lower and upper prediction intervals as columns in the Forecasts data table.

**Save One-Step-Ahead Predictions**   Stacks the one-step-ahead predictions and the forecasts and saves them to a separate data table.

**Save Original Series**   Stacks the original series and the forecasts and saves them to a separate data table.

**Save Forecast Results to Original Data Table**   Saves the results to new columns in the original data table.

See *Using JMP* 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.

# Model Reports

The Model Reports report contains a Select Series list and one or more individual series modeling reports. The individual reports that are displayed are determined by the Select Series list. Click on the name of a series in this list to show the report for the selected series. If you select more than one series name by using either Ctrl or Shift, an individual model fit report for each of the selected series is shown. If you deselect one or more series from the Select Series list, the reports for the corresponding series are removed.

The Select Series list is linked to the data table. When you select a series in the Select Series list, the observations for the series and response variable combination are also selected in the data table. The Select Series list is also linked to the Group Summary of Y table. If a Time variable is specified, the Select Series list is also linked to the Group Summary of Time table. When you select a series in the Select Series list, the corresponding entries are also selected in the tables.

Each individual model report contains the following information:

**Model Type**   Describes the best fitting model for the series in terms of error, trend, and seasonality.

**Model Report**   A group of reports based on the model that is fit to the series.

   **Series and Forecasts**   (Available only when Information Criteria is specified as the model selection strategy.) A plot of the original series and forecasts. During the time period of the original series, the one-step-ahead predictions are shown. Beyond the time period of the original series, the forecasts are shown. The shaded region represents the forecasting interval.

   **Note:** All of the forecasts start at the time point that immediately follows the last time point across all individual series. See "Data Format".

   **Training, Holdback, and Forecasts**   (Available only when Forecast Performance is specified as the model selection strategy.) A plot of the training series, holdback series, and the forecasted series. A shaded vertical region represents the holdback series. Since this is the model refit on the entire time series, the one-step-ahead predictions are shown during the time period of the entire original series. During the time period past the last observation, the forecasts are shown. The shaded region represents the forecasting interval.

   **Model Summary**   Provides the -2Loglikelihood, the AIC, the number of parameters, and the sigma value. The sigma value is the estimate of the one-step-ahead forecast standard deviation. If Forecast Performance is specified as the model selection strategy, the root mean square error (RMSE) for the holdback set is also shown. See *Fitting Linear Models*.

**Parameter Estimates**   Provides the parameter estimates for the model. The parameters correspond to the parameters in the best fitting model, which is described in the Model Type report.

**One-Step-Ahead Forecasting Errors**   A plot of the one-step-ahead forecasting errors over time. For each time point $t$, the error is calculated as $e(t) = y(t) - \mu(t)$.

**One-Step-Ahead Relative Forecasting Errors for Multiplicative Error Model**   (Available only for multiplicative error models.) A plot of the one-step-ahead relative forecasting errors over time. For each time point $t$, the error is calculated as $e(t) = [y(t) - \mu(t)]/\mu(t)$.

# 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 20.1** Example of Matched Pairs Analysis
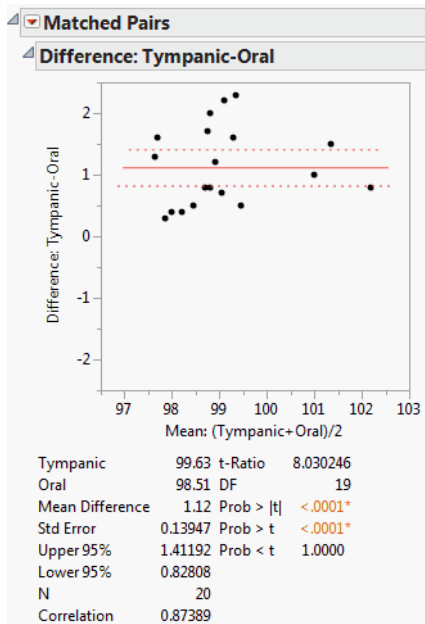
# Contents

# 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 20.2**  The Matched Pairs Report Window
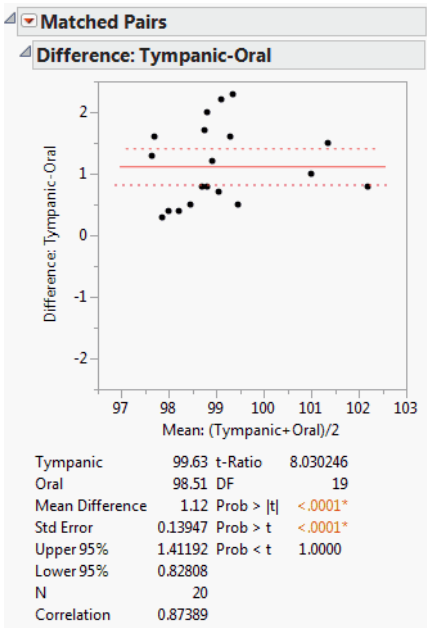


The results show that, on average, the tympanic thermometer measures 1.12 degrees higher than the oral thermometer. The small *p*-value (Prob > |t|) indicates that this difference is statistically significant, and not due to chance.

Note that this matched pairs analysis does not indicate which thermometer is correct (if either), but indicates only 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 20.3** The Matched Pairs Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**Y, Paired Response**    Provide the two response columns. For information about analyzing more than two responses, see "Multiple Y Columns".

**X, Grouping**    Provide a grouping variable to compare the differences across groups. See "Across Groups".

**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.

After you click **OK**, the Matched Pairs report window appears. See "The Matched Pairs Report".

## 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" to produce the report window shown in Figure 20.4.

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". If you specified an **X, Grouping** variable, the report also includes the Across Groups report. See "Across Groups".

**Figure 20.4**  Example of Matched Pairs Report
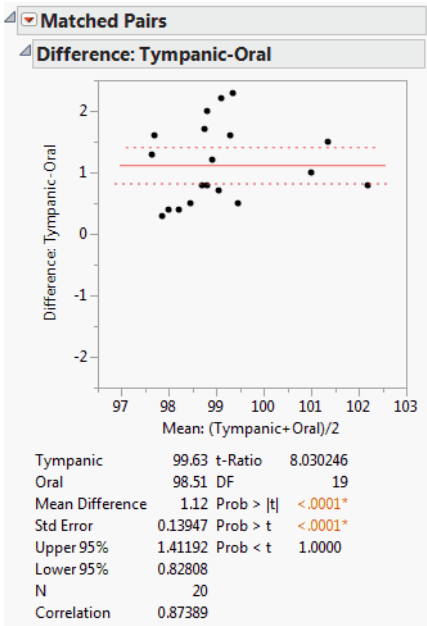
**Note:** The red triangle menu provides additional options that can add reports to the initial report window. See "Matched Pairs Platform Options".

## 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 more information about a reference frame, see "Matched Pairs Platform Options".

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"

# 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".

**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. See *Basic Analysis*. See also Lehman (2006), Conover (1999, p. 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 *Using JMP* 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 20.5 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 20.5** Examples of Matched Pairs across Groups and Fit Model MANOVA with Repeated Measures
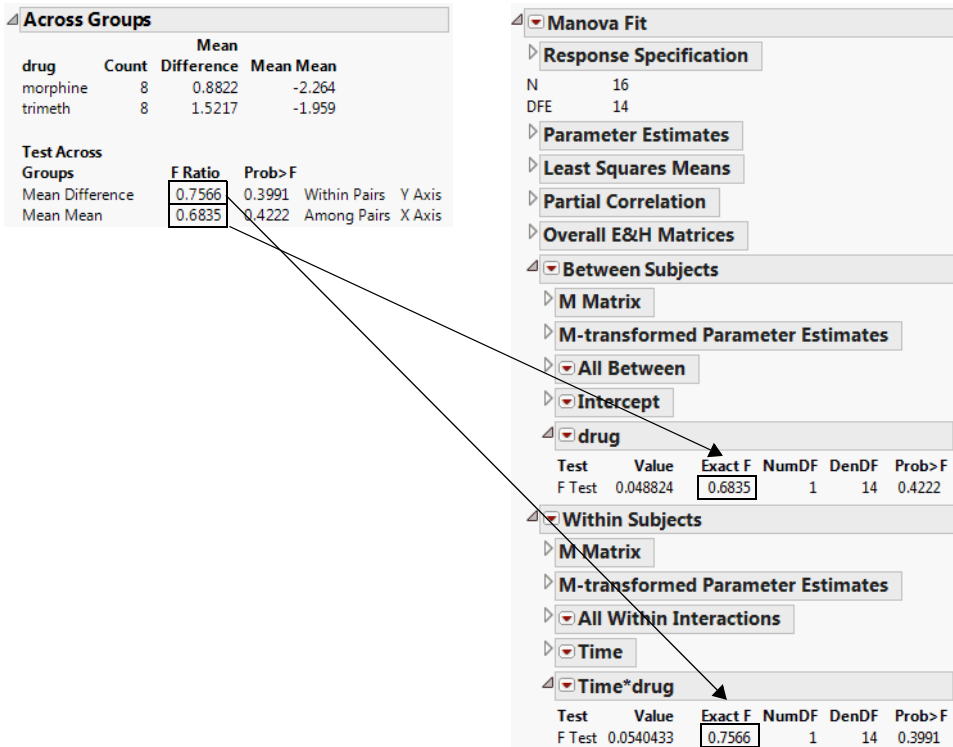


The F Ratio for the Mean Difference in the Across Groups report corresponds to the F Ratio for Time*drug under the Within Subjects report. The F Ratio for the Mean Mean in the Across Groups report corresponds to the F Ratio for drug under Between Subjects in the Manova Fit report.

# Statistical Details for the Matched Pairs Platform

- "Graphics for Matched Pairs"
- "Correlation of Responses"

## Graphics for Matched Pairs

The primary graph in the platform is a Tukey mean-difference (Cleveland 1994, p. 130). This graph plots the difference of the two responses on the vertical axis against the mean of the two responses on the horizontal 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 20.6**  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 $y_1 > y_2$

$\dfrac{y_2 + y_1}{2}$ rescaled to mean of responses
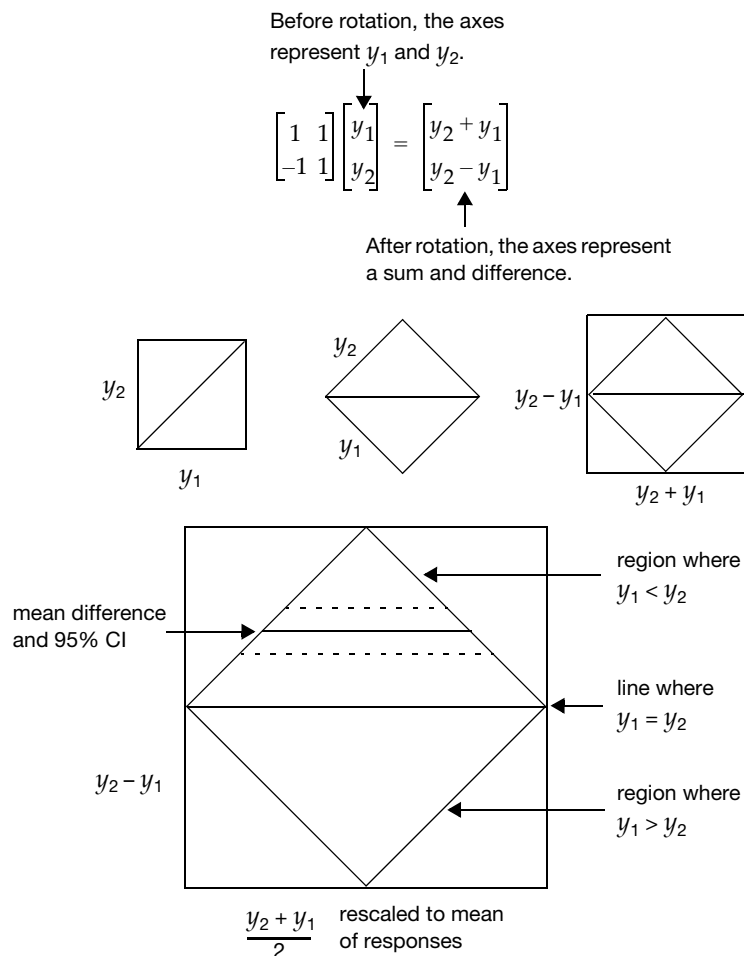
# Correlation of Responses

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 20.7 shows how the positive correlation of the two responses becomes the small variance on the difference (the vertical 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 vertical axis.

**Figure 20.7** Examples of Positive Correlation Before and After Rotation
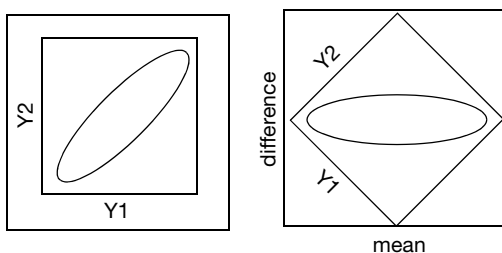
# Modeling Utilities

### Exploring Data for Outliers, Missing Values, and Patterns

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.
- Explore and find unusual features or patterns in your data.

**Figure 21.1** Multivariate k-Nearest Neighbor Outlier Example
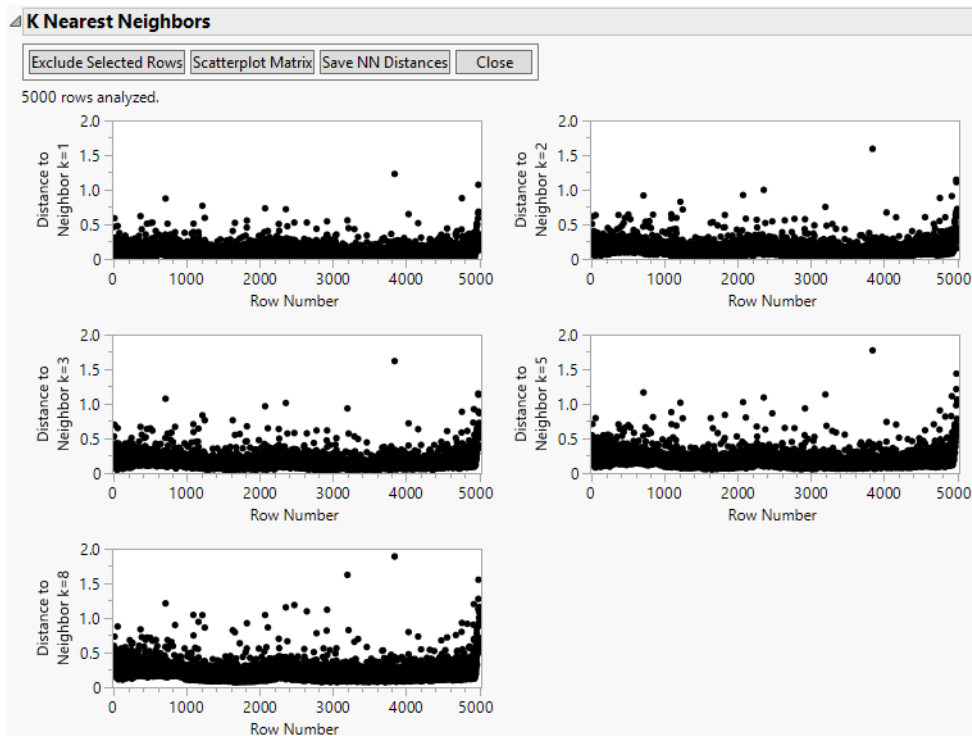
# Contents

# Explore Outliers Utility

The Explore Outliers tool provides four different options to identify, explore, and manage outliers. 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, the inclusion of error or missing value codes in the data set, or simply an unusual value. The presence of outliers can distort estimates and bias results 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, it is a good practice to locate them. There are many ways to visually inspect for outliers. For example, box plots, histograms, and scatter plots can easily display these extreme values. See *Discovering JMP*.

The Explore Outliers tool provides the following options:

**Univariate**   There are two options for exploring outliers in your univariate 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".

  **Robust Fit Outliers**   Finds robust estimates of the center and spread of each column and identifies outliers as those data points that are far from those values. See "Robust Fit Outliers".

**Multivariate**   There are two options for exploring outliers in your multivariate data.

  **Robust PCA Outliers**   Decomposes data into a low-rank matrix and residuals and uses the residuals to detect outliers. See "Robust PCA Outliers".

  **K Nearest Neighbor Outliers**   Identifies outliers as values that are far from their k-nearest neighbors. See "K Nearest Neighbor Outliers".

## Example of the Explore Outliers Utility

Use the Explore Outliers utility to identify outliers that can then be examined using the Distribution platform. 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.

1.  Select **Help > Sample Data Library** and open the Probe.jmp sample data table.

2.  Select **Analyze > Screening > Explore Outliers**.

3.  Click the triangle next to Responses(387/0) to show all of the columns in the group.

4.  Select columns VDP_M1 through VDP_SICR and click **Y, Columns**. There should be 14 columns selected.

**Figure 21.2**  Explore Outliers Launch Window
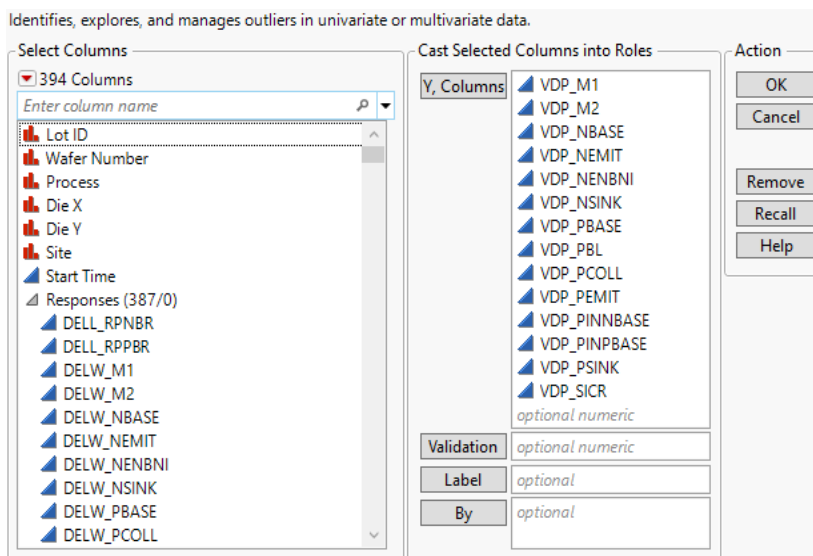


5.  Click **OK**.

6.  Click **Quantile Range Outliers**.

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

7.  In the Quantile Range Outliers report, select **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.

8.  In the Nines report, select each column.

9.  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.

10. Click **OK**.

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

12. Select **Restrict search to integers**.

In 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.

13. 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 21.3** Distribution of Columns with Outliers Selected
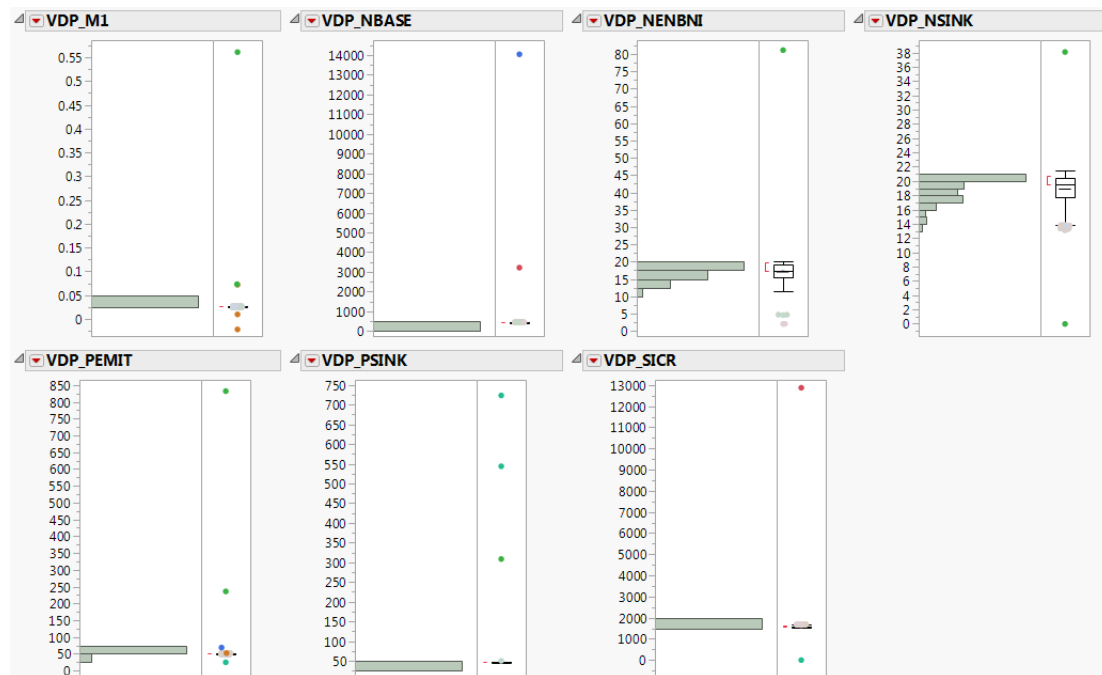


In columns VDP_M1 and VDP_PEMIT, notice that some of the selected outliers are somewhat close to the majority of data. For the rest of the columns, the selected outliers appear distant from the majority of data. Investigate the data points and exclude them from your analyses.

## 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.

6. 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 Distributions red triangle and select **Redo > Redo Analysis**.

**Figure 21.4** Distributions of Columns with Outliers Excluded
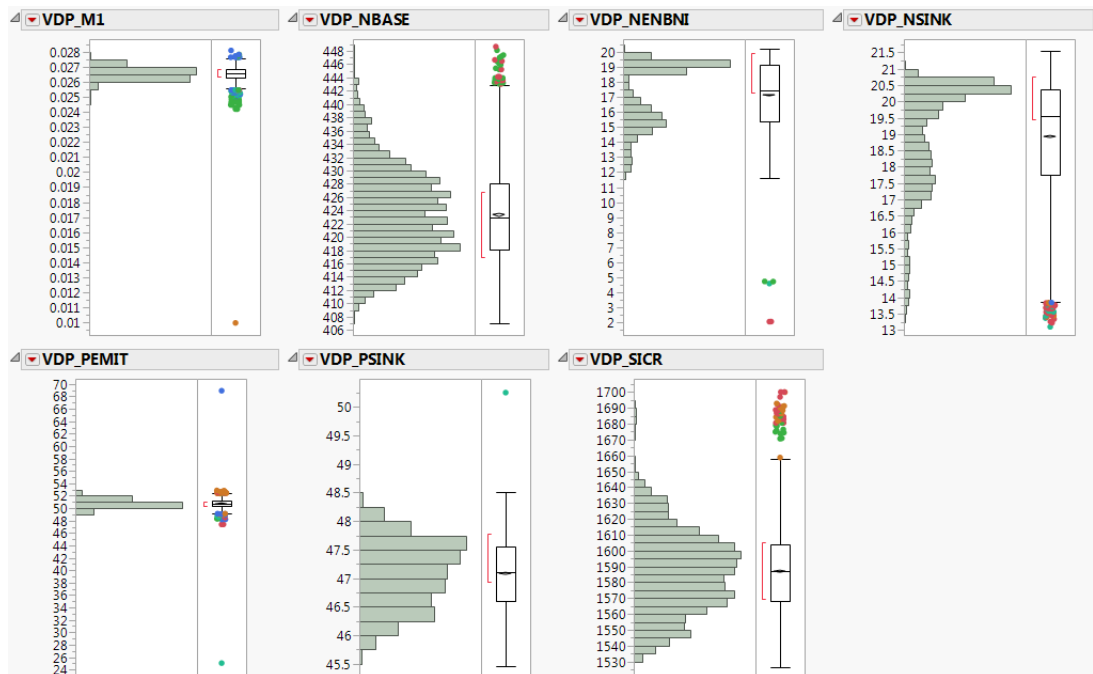


The displays of the distributions of the data are now more informative without the outliers.

# Launch the Explore Outliers Utility

To launch Explore Outliers, select  **Analyze > Screening > Explore Outliers**.

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

**Figure 21.5**  Explore Outliers Utility Launch Window
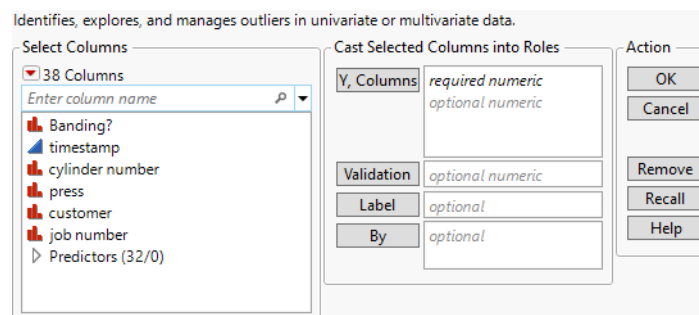


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**Y, Columns**    Specifies the columns that you want to analyze.

**Validation**    Specifies a validation column that is used for Robust PCA Outliers.

**Label**    Specifies a column that replaces row numbers in multivariate analysis reports with labels.

**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.

After you click **OK**, the Explore Outliers report appears. You are presented with the following four outlier analysis commands:

- "Quantile Range Outliers"
- "Robust Fit Outliers"
- "Robust PCA Outliers"
- "K Nearest Neighbor Outliers"

**Tip:** To run an outlier analysis across all levels of a **By** variable, press Ctrl and click the desired outlier analysis command button.

# Quantile Range Outliers

The Quantile Range Outliers method of outlier detection uses the quantile distribution of the values in a column to locate 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 information about how quantiles are computed, see *Basic Analysis*.

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 or 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 identified and how you want to manage them.

**Figure 21.6** Quantile Range Outliers Window
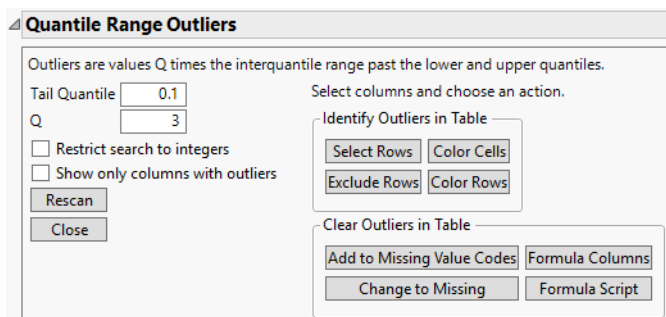


An outlier is considered any value more than $Q$ times the interquantile range from the lower and upper quantiles. You can adjust the value of $Q$ and the size of the interquantile range.

**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 determines the outlier threshold. Values that fall beyond *Q* times the interquantile range past the Tail Quantile or 1 - Tail Quantile values are identified as outliers. 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.

---

**Tip:** If no columns are selected in the report and you click one of the following buttons, a JMP Alert appears that enables you to select all of the columns.

---

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

**Exclude Rows**   Turns on the exclude row state for outliers in the selected columns in the Quantile Range Outliers Report. 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 a series of nines. Click **Rescan** to update the Quantile Range Outliers report.

---

**Note:** Add to Missing Value Codes is not available with Quantile Range Outliers if a By variable is specified in the launch window.

---

**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.

---

**Note:** If the selected outlier has been added to the missing value codes, the outlier is not changed to a missing value.

---

**Formula Columns**   Creates a new formula column for each column that is specified in the launch window. Each new column contains the original column's value if the value is within the outlier limits and is set to missing otherwise. The new columns are prefixed or suffixed by a user specified name to distinguish them from the original columns. By default, the suffix is set to "Culled".

**Formula Script**   Creates a script that is added to the data table. When the script is run, it creates a new formula column for each column that is specified in the launch window. Each new column contains the original column's value if the value is within the outlier limits and is set to missing otherwise. The new columns are prefixed or suffixed by a user specified name to distinguish them from the original columns. By default, the suffix is set to "Culled".

**Rescan**   Rescans the data after outlier actions have been taken.

**Note:** Press Ctrl and click **Rescan** to rescan across all command groups.

**Close**   Closes the Quantile Range Outliers panel.

**Note:** Press Ctrl and click **Close** to close all command windows.

## 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.

**Note:** Add Highest Nines to Missing Value Codes is not available with Quantile Range Outliers if a By variable is specified in the launch window.

**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, you will automatically be prompted to 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 thresholds for identifying outliers.

**Figure 21.7**  Robust Fit Outliers Window
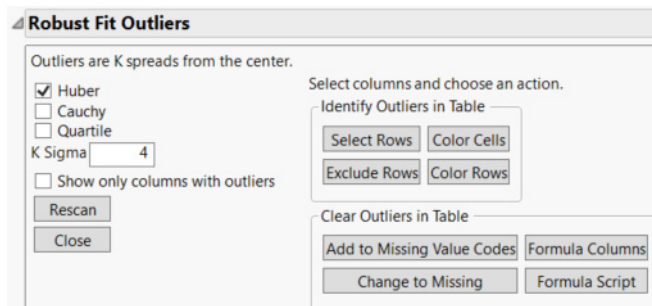
## Robust Fit Outliers Options

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 the data are normally distributed.

**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.

---

**Tip:** If no columns are selected in the report and you click one of the following buttons, a JMP Alert appears that enables you to select all of the columns.

---

**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.

**Note:** Add to Missing Value Codes is not available with Robust Fit Outliers if a By variable is specified in the launch window.

**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.

**Formula Columns**  Creates a new formula column for each column that is specified in the launch window. Each new column contains the original column's value if the value is within the outlier limits and is set to missing otherwise. The new columns are prefixed or suffixed by a user specified name to distinguish them from the original columns. By default, the suffix is set to "Culled".

**Formula Script**  Creates a script that is added to the data table. When the script is run, it creates a new formula column for each column that is specified in the launch window. Each new column contains the original column's value if the value is within the outlier limits and is set to missing otherwise. The new columns are prefixed or suffixed by a user specified name to distinguish them from the original columns. By default, the suffix is set to "Culled".

**Rescan**  Rescans the data after outlier actions have been taken.

**Note:** Press Ctrl and click **Rescan** to rescan across all command groups.

**Close**  Closes the Robust Fit Outliers panel.

**Note:** Press Ctrl and click **Close** to close all command windows.

## Robust PCA Outliers

You can use the Robust PCA Outliers utility to quickly identify outlier cells in correlated multivariate data. This method is useful because many other multivariate approaches identify only the outlier rows. Before the method is applied to the data, the columns are first centered (optional) and scaled. The scaling factor is defined as follows:

max [Q(.75) - Q(.50), Q(.50) - Q(.25)] / [normalQuantile(0.75)]

where

Q($p$) is the $p^{\text{th}}$ quantile

**Note:** If Q(75) or Q(25) are equal to the median, then more extreme quantiles are used until there is a non-zero range.

After the data are centered and scaled, the Robust PCA Outliers utility performs a sequence of singular value decompositions and thresholding steps to decompose the data matrix. The data are decomposed into a low-rank matrix and a sparse matrix of residuals. The thresholding is done so that the residuals are either very large or outliers or very close to zero for non-outliers. The algorithm determines a matrix rank appropriate to capture the systematic variation without the outliers or small noise. Outliers that are not in the low-rank space are detected based on their residuals. See Candes et al (2009) and Lin et al (2013). If there are missing values, they are initially replaced with zeros after the centering and scaling steps. Then, after each singular value decomposition (SVD) iteration, the missing values are updated by their predicted values from the SVD.

## Robust PCA Outliers Report

When you select Robust PCA Outliers from the list of commands, you must specify a value for Lambda and select if the data should be centered. If you Shift+Click the Robust PCA Outliers button, the following options are also available:

**Lambda**   Specifies a value that determines the sparsity of the matrix of residuals. For larger values of Lambda, the matrix of residuals is more sparse. For a data table with $n$ training rows and $p$ columns, the default value of Lambda is defined as follows:

$$\lambda = \frac{2}{\sqrt{\max(n, p)}}$$

**Max Iterations**   Specifies the maximum number of SVD iterations.

**Converge Criterion**   Determines when to stop the algorithm.

**Outlier Threshold**   Specifies the outlier threshold that determines which outliers are shown in the Cell Large Residuals table. An observation is shown if the scaled residuals is larger than the following:

   $\min[0.99 \times \max\{\text{abs}(\text{residuals})\}, \text{Outlier Threshold}]$

By default, the Outlier Threshold is 2.

**Center**   Determines if the data are centered before the Robust PCA Outlier algorithm is performed.

**Note:** If the number of rows is less than or equal to 10, the data are not centered.

**Scale**   Determines if the data are scaled before the Robust PCA Outlier algorithm is performed.

**Note:** If the number of rows is less than or equal to 10, the data are not scaled.

The Robust PCA Outliers report contains a table with information about the method. This table includes the rank of the low-rank matrix, the number of SVD iterations, the convergence criterion, the value of Lambda, and the number of imputed missing values. The report also contains the following tables and reports:

**Cell Large Residuals**   A table that shows the largest outlier cells. The number of cells shown is determined by the Outlier Threshold. The table contains the column name and row number of the cell, the residual value, and the scaled residual value.

> **Tip:** To color specific outlier cells in the data table, select rows in the Cell Large Residuals table and click **Colorize**.

**Row Root Mean Square**   A table that shows the root mean square value for each row in the data table. The root mean square is calculated using the scaled residuals.

> **Tip:** If you select a row in the Row Root Mean Square table, the corresponding row is selected in the data table.

**Column Root Mean Square**   A table that shows the root mean square value for each column specified in the launch window. The root mean square is calculated using the scaled residuals.

> **Tip:** If you select a row in the Column Root Mean Square table and click **Select Columns**, the corresponding column is selected in the data table.

**Snapshot**   A graphical representation of the outlier cells in the data table. The outlier cells are colored in red.

**Residuals**   The matrix of residuals from the matrix decomposition. A cell is colored if the absolute value of the scaled residual is greater than the following:

$$\min[0.99 \times \max\{\mathrm{abs(residuals)}\}, \text{Outlier Threshold}]$$

**Low Rank Approximation**   The matrix of scaled residuals from the matrix decomposition.

**Singular Values**   The vector of singular values from the SVD.

### Robust PCA Outliers Options

There are buttons at the bottom of the Robust PCA Outliers report that provide options to save different parts of the report.

**Close**   Closes the Robust PCA Outliers report.

**Save Large Outliers**   Saves the information in the Cell Large Residuals table to a new data table.

**Save Cleaned**   Opens a window that provides several techniques to clean the outliers based on thresholds and save new columns to the data table.

**Trim**   Trims outlier cells if the corresponding absolute scaled residual is greater than the specified threshold. By default, the threshold is 10. Select Color ■ to color the outlier cells red. The trimmed cells are set to the value of the unscaled threshold.

**Impute**   Sets outlier cells to the value of the low rank approximation if the corresponding absolute scaled residual is greater than the specified threshold. By default, the threshold is 100. Select Color ■ to color these cells green.

**Make Missing**   Sets outlier cells to missing if the corresponding absolute scaled residual is greater than the specified threshold. By default, the threshold is 1000. Select Color ■ to color these cells blue.

**Color imputed from missing** ■   If selected, colors cells that originally had missing values and were imputed.

**Save Residuals**   Saves the residuals to new columns in the original data table.

**Save Scaled Residuals**   Saves the scaled residuals to new columns in the original data table.

**Save Low Rank Approx**   Saves the low-rank approximation to new columns in the original data table.

# K Nearest Neighbor Outliers

Use K Nearest Neighbor Outliers to identify an outlier based on distance to its nearest neighbor. For each value of $k$, the K Nearest Neighbor Outliers utility displays a plot of the Euclidean distance from each point to its $k$th nearest neighbor. You specify the largest value of $k$, denoted as $K$. Plots are provided for $k = 1,2,3,5,...,K$, using the Fibonacci sequence to avoid displaying too many plots.

Before the nearest neighbors are calculated, the columns are centered and scaled. The scaling factor is as follows:

max [Q(.75) - Q(.50), Q(.50) - Q(.25)] / [normalQuantile(0.75)]

where

$Q(p)$ is the $p^{\text{th}}$ quantile

**Note:** If Q(75) or Q(25) are equal to the median, then more extreme quantiles are used until there is a non-zero range.

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 studying only 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 have small distances to their nearest neighbors. You might 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 can appear to be outliers. You might overlook the fact that the points form a cluster, interpreting the individual cluster members as outliers instead.

## K Nearest Neighbor Outliers Report

When you select K Nearest Neighbor Outliers from the list of commands, you must specify the value of *K* to use as an upper bound for the farthest neighbor to be considered. You must also specify whether missing values should be imputed. Notice that *K* is set to 8 and Impute Missing is selected by default.

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. It is of the form Distance to Neighbor k = <a>, where a is an integer denoting the $a^{\text{th}}$ closest neighbor. Each plot shows the distance from the point in the $i^{\text{th}}$ row to its $a^{\text{th}}$ 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.

**Save NN Distances**    Saves the distances from each row to its $n^{\text{th}}$ nearest neighbor as new columns in the data table.

**Close**    Closes the K Nearest Neighbors report.

### Largest Outliers

The report also includes a Largest Outliers table. This table contains the 20 observations with the largest distances from their $K^{th}$ nearest neighbor. The table has the following columns:

**Row**    The row number of the observation.

**Distance**    The distance from the observation in the specified row and its $K^{th}$ nearest neighbor. The table is sorted by this column in descending order.

**Nearest Neighbors**    Lists the row numbers for the $k$ nearest neighbors. The first row number is the closest nearest neighbor. The last row number is the $K^{th}$ nearest neighbor and the distance between this observation and the specified row is found in the Distance column.

**Col<n>**    Specifies the column name for the corresponding RMS value.

**RMS<n>**    Calculates the root mean squared differences across the $k$ nearest neighbors for each column. The five largest RMS values are displayed in order, where RMS1 is the maximum RMS value. The $p^{th}$ RMS value is calculated as follows:

$$\text{RMS}_p = \sqrt{\frac{\sum\limits_{k=1}^{K}(D_{p,i} - D_{p,i_k})^2}{K}}$$

where

$D_p$ is the $p^{th}$ column

$D_{p,i}$ is the value of the $p^{th}$ column for row $i$

$D_{p,ik}$ is the value of the $p^{th}$ column for the $k^{th}$ nearest neighbor of row $i$

**Note:** The number of Col and RMS columns shown in the Largest Outliers table is the minimum of the number of columns specified in the launch and the number five.

## Explore Outliers Utility Options

See *Using JMP* 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 Explore Outliers Utility

## Robust PCA 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.  Shift+Click **Robust PCA Outliers**.
6.  Enter 10 next to Outlier Threshold.

    Use the default values for the other options.
7.  Click **OK**.

**Figure 21.8** Robust PCA Outliers Report
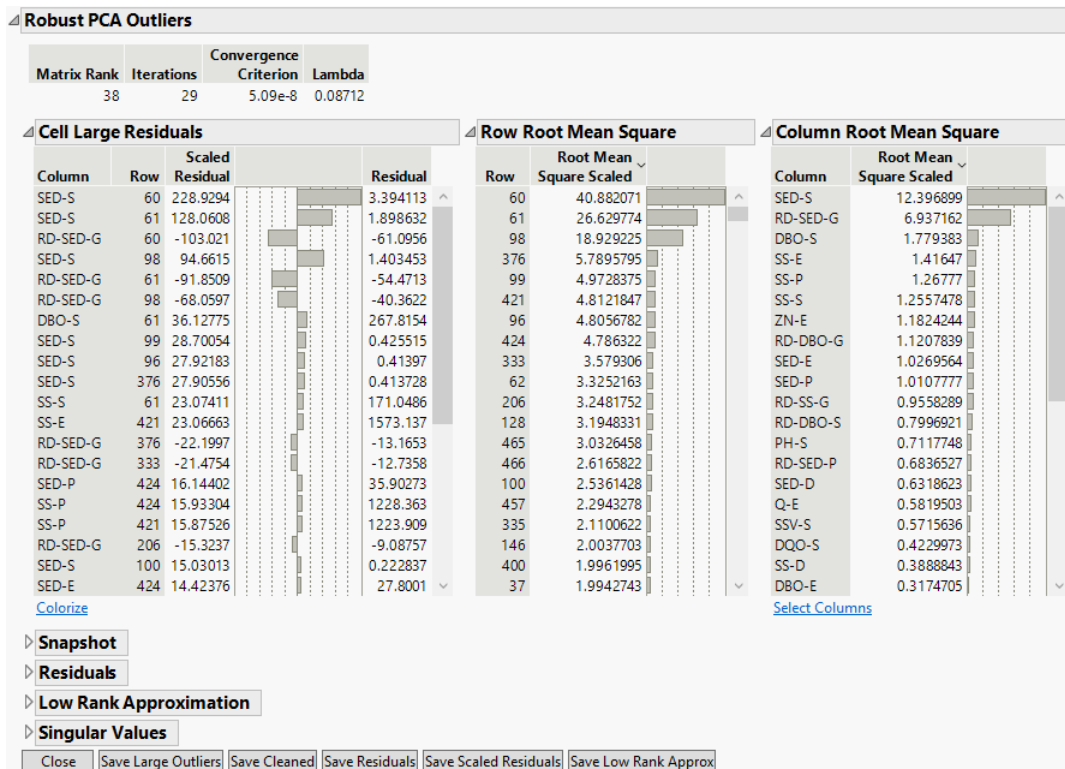


The Cell Large Residuals table shows the cells that have a scaled residual greater than 10 or less than -10. There are several cells with large scaled residuals from the SED-S column. If you look at the Column Root Mean Square table, it shows that the SED-S column does in fact have the largest scaled Root Mean Square value. The row with the largest scaled Root Mean Square value is row 60.

8.  Click **Save Cleaned** at the bottom of the report.

    This option enables you to adjust cells with large outliers by trimming the values, imputing new values, or setting the cells to missing. The Save Cleaned Columns report provides these options.

9.  Select Impute. Trim is already selected by default.

10. Click **OK**.

    There are 38 new Cleaned columns added to the Water Treatment.jmp data table. Since you know row 60 had the largest scaled Root Mean Square value, scroll to this row to view some of the cells that were trimmed and imputed. You can now use the cleaned columns in any type of analysis.

**Figure 21.9** Cleaned Columns from Robust PCA Outliers



# 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 imputation methods assume that data are *missing at random*, which means that any differences between missing and non-missing data cannot be explained by 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. See *Fitting Linear Models*.

**Caution:** Be careful when analyzing data after imputing missing values, as the results have the potential to be biased.

# 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 conduct missing value imputation only for columns that have a continuous modeling type, you 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** check box.

**Figure 21.10**  Missing Value Report



The Missing Columns report shown in Figure 21.10 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, imputed values might not result in a meaningful analysis. For such data, you could explore a model using the Informative Missing option for Column J in a platform that supports the Informative Missing option.

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**.

   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 21.11** Imputation Report
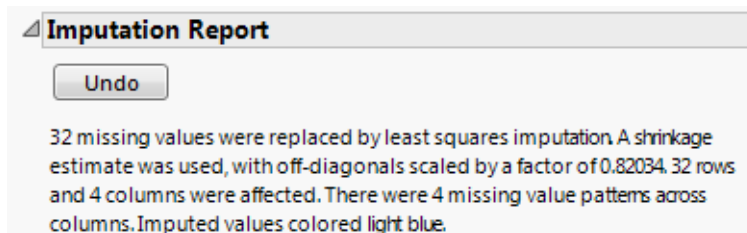


The Imputation Report indicates how many missing values were imputed and the specific imputation details. No missing data remain in the four columns that had missing values.

**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. You can also specify a **By** variable.

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

# The Missing Value Report

**Figure 21.12** Missing Value Report for Continuous Variables in Arrhythmia.jmp.
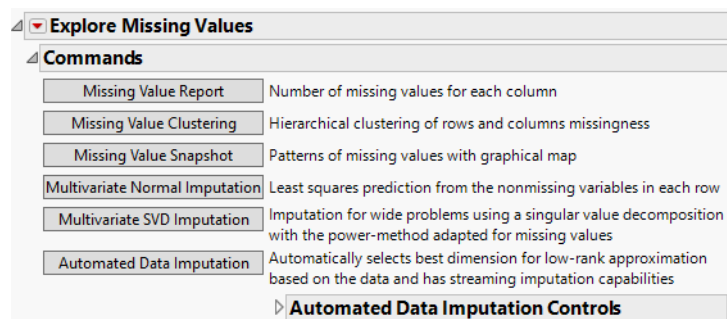


After you click OK in the launch window, the report opens to show a Commands outline and a Missing Columns report. The commands are the following:

- "Missing Value Report"
- "Missing Value Clustering"
- "Missing Value Snapshot"
- "Multivariate Normal Imputation" (Not available if you entered a Numeric column with a Nominal or Ordinal modeling type in the launch window.)
- "Multivariate SVD Imputation" (Not available if you entered a Numeric column with a Nominal or Ordinal modeling type in the launch window.)
- **JMP PRO** "Automated Data Imputation" (Not available if you entered a Numeric column with a Nominal or Ordinal modeling type in the launch window.)

**Tip:** To run a missing value command across all levels of a **By** variable, press Ctrl and click the desired command button.

## 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 whether 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. Hover over a cell to see the list of values represented. Click in the plot to select missing data pattern rows. Vertical bars are displayed 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 can 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 nonmissing values for each variable. The off-diagonal entries for any two variables are computed using all observations that are nonmissing 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 Schäfer 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 shows the 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 about 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.

### JMP PRO Automated Data Imputation

The Automated Data Imputation (ADI) utility imputes missing values using a low-rank matrix approximation method, also known as matrix completion. Once trained, the ADI model is capable of performing missing data imputations for streaming data through scoring formulas. Streaming data are added rows of observations that become available over time and were not used for tuning or validating the imputed model. This utility is flexible, robust, and automated to select the best dimension for the low-rank approximation. These features enable ADI to work well for many different types of data sets.

A low-rank approximation of a matrix is of the form $\mathbf{X} = \mathbf{UDV}'$ and can be viewed as an extension of singular value decomposition (SVD). ADI uses the Soft-Impute method as the imputation model and is designed such that the data determines the rank of the low-rank approximation.

The ADI algorithm performs the following steps:

1. The data are partitioned into training and validation sets.
2. Each set is centered and scaled using the observed values from the training set.
3. For each partitioned data set, additional missing values are added within each column and are referred to as induced missing (IM) values.
4. The imputation model is fit on the training data set along a solution path of tuning parameters. The IM values are used to determine the best value for the tuning parameter.
5. Additional rank reduction is performed using the training data set by de-biasing the results from the chosen imputation model in step 4.
6. Final rank reduction is performed to calibrate the model for streaming data and to prevent overfitting. This is done by fitting the imputation model on the validation set, using the rank determined in step 5 as an upper bound.

### Automated Data Imputation Controls

The ADI utility contains options for saving the imputed values and advanced controls.

**Figure 21.13** ADI Controls



**Options for Saving Imputed Values**   The following three options for saving the imputed values for the ADI method are available:

**Create New Data Table**   Creates a new data table that has the same dimensions as the original data table. In the new data table, the columns selected in the launch window contain the imputed values.

**Save Scoring Formula to Current Data Table**   Saves a column group, named Imputed_, to the current data table that contains the imputed columns specified in the launch window. A hidden column, ADI Impute Column, is also added to the current data table that contains the imputed vectors and the scoring formula used in the data imputation. The column formulas automatically update if any additional rows are added to the data table, enabling missing data imputation for streaming data. This is the default option.

**Impute Values in Place**   Imputes the missing values in the current data table. The imputed values are displayed in light blue.

**Advanced Controls**   Contains the following advanced controls, which default to recommended settings based on the data:

**Dimension Upper Bound**   Determines the maximum rank allowed in the low-rank approximation. This is determined by the dimension of the matrix formed by the chosen columns.

**Maximum Iterations**   Determines the number of values that are iterated over to determine the tuning parameter for the imputation model. The default is 10.

**Proportion of Observations to Induce as Missing**    Determines the proportion of IM values that are added to the training and validation sets. The default proportion for each set is 0.2.

**Proportion of Rows to Use for Validation**    Determines the proportion of rows to use in the training and validation sets. The default proportion for the validation set is 0.3.

**Set Random Seed**    Determines the random seed for ADI. Use this option to obtain reproducible results.

## Explore Missing Values Utility Options

**ADI Loading Matrix**    (Available only for the ADI utility.) Shows or hides a report that contains the columns that correspond to the factor loading for each component.

See *Using JMP* 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.

# Explore Patterns Utility

Data can be messy, so it is sometimes hard to determine the validity of your data by visual inspection alone. The Explore Patterns utility is a tool for detecting unusual or unexpected patterns in your data. Although some patterns occur by chance or some innocent mechanism, they might indicate data that has been falsified or tampered with. This utility is most useful when the values are precise so that matches and patterns are less likely to happen by coincidence. The Explore Patterns utility is also designed to handle large data tables efficiently.

The Explore Patterns utility looks for duplicate values and duplicate sequences of values. These types of patterns can indicate data tampering strategies such as pasting the same value many times, pasting the same value across a range of cells, or copying a range of data and pasting it somewhere else. The utility also analyzes properties of the formatted values to determine whether the data are actual measurements or might have been generated by a random number generator. There are also options to examine distribution properties and detect linear relationships between columns across groups of rows.

**Note:** The Explore Patterns utility ignores excluded rows, but the results are not necessarily equivalent to if you had deleted the row. For runs of rows, such as the Longest Runs and Longest Duplicated Sequences options, an excluded row stops the run.

## Example of the Explore Patterns Utility

The Nicardipine Lab Patterns.jmp sample data table is adapted from a study of the drug Nicardipine. The data table contains 27 columns of numeric laboratory results. Use the Explore Patterns Utility to determine whether there are any unexpected patterns in the laboratory results.

1. Select **Help > Sample Data Library** and open the Nicardipine Lab Patterns.jmp sample data table.

2. Select **Analyze > Screening > Explore Patterns**.

3. Select the Laboratory Results column group and click **Y, Columns**.

4. Click **OK**.

5. Click the gray disclosure icon next to Univariate Summary to open the Univariate Summary report.

**Figure 21.14** Explore Patterns Univariate Summary Report

◢ **Univariate Summary**

Longest Runs

| Column | Starting Row | Count | Value | Rarity |
|---|---|---|---|---|
| Creatinine | 2265 | 7 | 0.03536 | 18.9 |
| Hematocrit | 3048 | 4 | 28.999999 | 13.1 |
| Prothrombin Time | 2931 | 6 | 12 | 10.1 |
| BUN | 2267 | 5 | 2.142 | 8.5 |
| LDH | 3456 | 3 | 602 | 7.1 |
| Activated PTT | 1460 | 3 | 30.299999 | 6.2 |
| Platelet | 521 | 3 | 150 | 5.8 |
| Platelet | 1092 | 3 | 336 | 5.1 |
| Creatine Kinase | 3456 | 3 | 63 | 4.9 |
| Activated PTT | 3435 | 3 | 29.5 | 4.6 |
| ALT | 3457 | 3 | 66 | 4.4 |
| CO2 | 1390 | 4 | 82.612 | 4.0 |

Longest Duplicate Sequences

| Column | Starting Row I | Starting Row J | Count | Rarity |
|---|---|---|---|---|
| Activated PTT | 2816 | 3034 | 4 | 18.2 |
| Prothrombin Time | 2942 | 2966 | 6 | 14.9 |
| Hematocrit | 3051 | 3057 | 3 | 14.6 |
| Leukocytes | 547 | 1275 | 3 | 11.1 |
| Creatinine | 1464 | 1681 | 8 | 11.0 |
| Leukocytes | 975 | 1485 | 3 | 10.8 |
| ALT | 840 | 2425 | 3 | 10.5 |
| Sodium | 2339 | 2976 | 6 | 9.8 |
| Leukocytes | 456 | 2805 | 3 | 9.8 |
| Activated PTT | 1050 | 1055 | 4 | 9.4 |
| Potassium | 1046 | 1067 | 6 | 9.4 |
| Hemoglobin | 1247 | 3273 | 3 | 9.2 |

The Univariate Summary report summarizes the longest runs and longest duplicate sequences found across all of the columns. Each table is sorted by Rarity, a calculation of how likely the run or sequence is to occur by chance. The column with the highest rarity score for longest runs is Creatinine, whereas the column with the highest rarity score for longest duplicate sequences is Activated PTT. You will further examine these columns in the Univariate Patterns Report.

6. In the Univariate Patterns report, the report for the first column, Activated PTT, is shown initially. Press Ctrl and select Creatinine from the Select Columns list.

**Figure 21.15** Explore Patterns Univariate Patterns Report
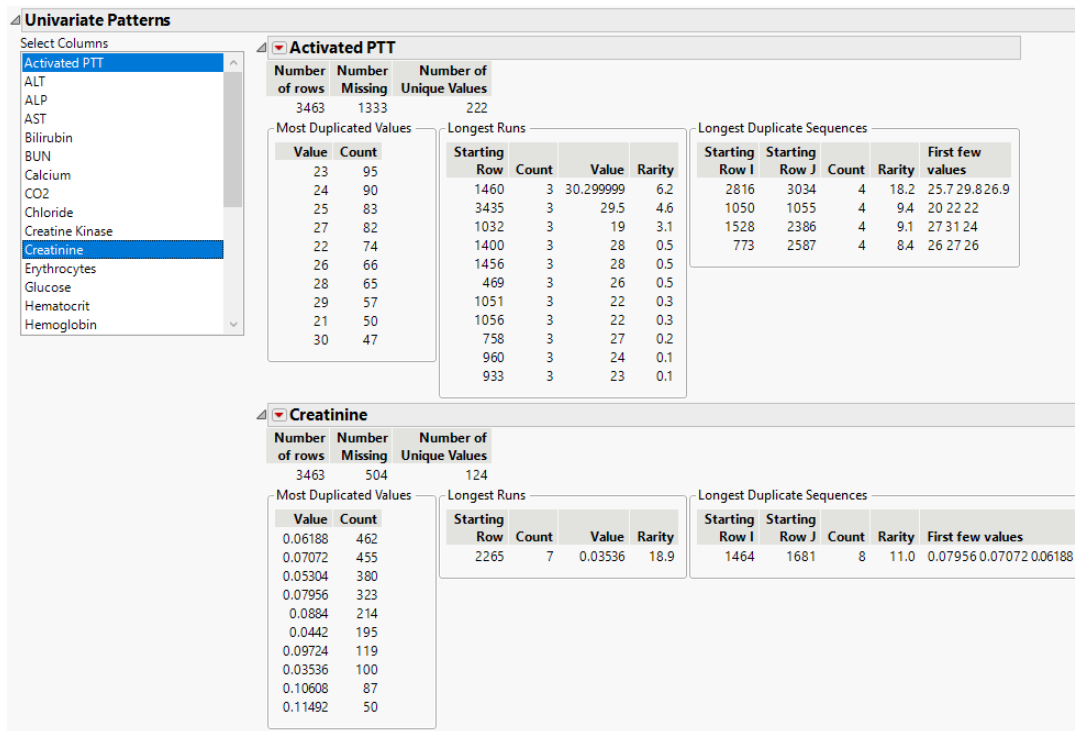


As you select column names from the Select Columns list, the corresponding column reports are shown. The Activated PTT report shows that this column contains several three count runs and four count sequences. The Creatinine report shows that although it is the column with the highest rarity score for Longest Runs, there is only one run in the column. This column also has one eight count sequence.

# Launch the Explore Patterns Utility

Launch the Explore Patterns utility by selecting **Analyze > Screening > Explore Patterns**.

**Figure 21.16**  Explore Patterns Utility Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**Y, Columns**    Assigns the columns to analyze. The variables must be numeric and continuous.

**By**    Assigns one or more 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.

**Minimum Run Size**    Specifies the minimum number of values in a row for a sequence to be reported as a run.

**Minimum Longest Duplicate Size**    Specifies the minimum length of a sequence for it to be reported as a duplicate sequence within a column.

**Minimum Cross Column Duplicate Run Size**    Specifies the minimum length of a sequence for it to be reported as a duplicate sequence across columns.

**Minimum Rows for Linear Relationship**    Specifies the minimum number of rows used for detecting linear relationships between columns across groups of rows.

**Note:** The minimum number of rows must be greater than or equal to three.

**Include Missing**   Specifies that missing values be included in pattern detection.

# Explore Patterns Report

The initial Explore Patterns report contains a Control Panel, Univariate Summary report, and Univariate Patterns report. Many of the tables in these reports contain rows that can be selected. If a row is selected and you right-click the first column, a menu appears that enables you to select or colorize the related rows and columns in the data table.

## Control Panel

The control panel contains options that enable you to select which reports appear in the report window. These are the same options that are in the Explore Patterns red triangle menu. See "Explore Patterns Utility Options". There is also a section of the control panel to specify report settings. These are the same settings that are found in the launch window. See "Launch the Explore Patterns Utility". If you change the settings in the control panel, the reports are automatically updated.

## Univariate Summary

The Univariate Summary report combines results from several patterns reports across columns. By default, there are summary tables for Longest Runs and Longest Duplicate Sequences. Longest Runs and Longest Duplicate Sequences with rarity $\geq 2$ are shown in the tables.

If the Fraction Length option is selected a Fraction Length summary table is also shown. This table shows the proportion of values in each column that have a fraction length greater than or equal to 15. Columns that do not contain any values that have fraction lengths greater than or equal to 15 are not shown in the Fraction Length summary table.

If at least one of the Y columns has a Spec Limit column property and the Distribution wrt Spec Limits option is selected, a Spec Limits Distrib summary table is also shown. This table shows the columns that have a statistically significant number of observations outside of the specification limits.

For more information about the Univariate Summary report contents, see "Explore Patterns Utility Options".

### Univariate Patterns

The Univariate Patterns report contains a Select Columns list and one or more column-specific patterns reports. The column-specific reports that are displayed are determined by the Select Columns list. To examine each column's report, select the first column and then successively use the down arrow key to advance to the next column. Alternatively, you can click a column name in this list to display the patterns report for the selected column. If you select more than one column name by using either Ctrl or Shift, a patterns report for each of the selected columns is displayed. If you deselect one or more columns from the Select Columns list, the reports for the corresponding columns are removed.

Each individual patterns report displays a summary table that contains the number of rows, number of missing observations, and number of unique values for the given column. By default, there are also tables for Most Duplicated Values, Longest Runs, and Longest Duplicate Sequences. If the Formatted Widths, Fraction Lengths, or Leading and Trailing Digits options are selected from either the control panel or the Explore Patterns red triangle menu, a corresponding table is added to each individual patterns report.

If you right-click on a row in any of the tables in the Univariate Patterns report, a menu of right-click options appears.

**Select Rows and Column**    Selects the column and rows in the original data table that correspond to the selected row in a specific table in the Univariate Patterns report. For example, if you right-click the second row of the Leading Digits table for a specific column, the rows in that column that have a leading digit of 2 are selected in the original data table.

**Colorize Cells**    Colors the cells in the original data table that correspond to the selected row in a specific table in the Univariate Patterns report. The rows of the colored cells are also selected.

There are red triangle options for the column-specific patterns report. See "Univariate Patterns Report Options".

## Explore Patterns Utility Options

The control panel and Explore Patterns red triangle menu contain options for summary and pattern tables.

**Most Duplicated Values**    Shows or hides a table of the values that appear most frequently in the data. Shows up to ten numbers for each column.

**Longest Runs**    Shows or hides a table of the longest run of the same value. The table contains the following information:

**Starting Row**    The row number that corresponds to the start of the run.

**Count**    The length of the run.

**Value**    The value of the entries in the run.

**Rarity**    A measurement of how rare it would be for the run to occur by chance. A high
rarity indicates that it is not likely that the run occurred by chance. See "Rarity in
Longest Runs".

**Longest Duplicated Sequences**    Shows or hides a table of the longest sequence of values that
appears more than once in the same column. Sequences of repeated values are also
considered. If a sequence is duplicated more than once, each instance appears in the table.
The table contains the following information:

**Starting Row I**    The row number that corresponds to the start of the first sequence.

**Starting Row J**    The row number that corresponds to the start of the duplicate sequence.

**Count**    The length of the sequence.

**Rarity**    A measurement of how rare it would be for the duplicate sequence to occur by
chance. A high rarity indicates that it is not likely that the duplicate sequence occurred
by chance. See "Rarity in Longest Sequences".

**First few values**    The first few values of the duplicate sequence. Shows up to three values
for each sequence.

**Formatted Widths**    Shows or hides a table of overall widths and decimal widths of the
formatted values. The table contains the following information:

**Width**    The width size.

**Overall Count**    The number of observations in the column whose overall width equals the
specified width.

**Decimal Count**    The number of observations in the column whose decimal width equals
the specified width.

**Fraction Length**    Shows or hides a table of the continued fraction lengths for the values. A
continued fraction is a representation of a number as a sequence of continually divided
terms. The fraction length can indicate how rational a number is. Lengths of 15 or more
indicate that the number is an irrational number, such as from a root function, a function
involving an irrational number, or from a random number generator (Benford, 1938).

**Note:** In Explore Patterns, the maximum fraction length is 15. Numbers with fraction
lengths of 15 or more are all reported as 15.

**Length**    The length of the continued fraction.

**Count**    The number of observations in the column whose continued fraction length
  equals the specified continued fraction length.

**Leading and Trailing Digits**    Shows or hides a table of counts for each leading and trailing
  digit, 1 through 9. For the leading digits, there are also columns for the expected rate and
  the minimum and maximum values that contain the corresponding leading digit. The
  expected rate is based on Benford's Law, which states that the frequency that a number has
  a first digit that is equal to $a$ is $F_a = \log_{10}(a+1/a)$. This law applies to data that have a large
  range of numbers that all have four or more digits.

**Spec Limit Matches**    (Available when at least one of the columns has a Spec Limit column
  property.) Shows or hides a table of values that are exactly the lower specification limit,
  upper specification limit, or target value. If there are no exact matches, the report is not
  shown.

**Distribution wrt Spec Limits**    (Available only when at least one of the columns has a Spec
  Limit column property.) Shows or hides a table of observed versus expected observations
  that are outside of the specification limits. The Chi-Square and PValue columns appear
  only if the number of observed observations is significantly higher than the number of
  expected observations.

**Duplicates Across Columns**    Shows or hides the Duplicates Across Columns report. The
  report contains a table of sequences of values that appear in the same rows across more
  than one column. There is a note at the top of the report that states the minimum length of
  a sequence for it to be considered a duplicate. This number is specified in the launch
  window or control panel as Minimum Cross Column Duplicate Run Size. There is also a
  Colorize option at the top of the report. See "Colorize Duplicates Across Columns".

**Colorize Duplicates Across Columns**    (Available only after Duplicates Across Columns has
  been selected.) Colors cells in the data table to correspond to the duplicate matches found
  in the Duplicates Across Columns report. The rows that contain the duplicates are also
  selected.

**Linear Relationships**    Shows or hides the Linear relationships between variables report. The
  report contains a table of exact linear relationships across columns over a specified
  number of sequential rows. An exact linear relationship is defined as a linear relationship
  that has an $R^2$ value greater than or equal to 0.999999. There is a note at the top of the
  report that states the minimum number of rows considered. This number is specified in
  the launch window or control panel as Minimum Rows for Linear Relationships. There is
  also a Colorize option at the top of the report. See "Colorize Linear Relationships".

**Colorize Linear Relationships**    (Available only after Linear Relationships has been selected.)
  Colors cells in the data table to correspond to the linear relationships found in the Linear
  relationships between variables table. The rows that contain the linear relationships are
  also selected.

**Save Columns**    Contains the following save options:

   **Save Table of Runs**    Saves the longest runs for all columns to a new data table. The data
      table contains the same columns as the Longest Runs table in the Univariate Patterns
      report and is sorted by Starting Row.

   **Save Table of Duplicate Sequences**    Saves the longest duplicate sequences for all
      columns to a new data table. The data table contains the same columns as the Longest
      Duplicate Sequences table in the Univariate Patterns report and is sorted by Starting
      Row I.

   **Save Duplicates Across Columns**    Saves the Duplicates Across Columns table to a new
      data table.

   **Save Linear Relationships**    Saves the Linear relationships between values table to a new
      data table.

**Ordering Columns**    Contains options for ordering. When you change the ordering, the
   column list in the Univariate Patterns report is reordered. The top column is automatically
   selected and the report for only that column is displayed.

   **Order by Runs Rarity**    Orders the column list by the rarity values of the longest runs.
      Higher rarity values are at the top.

   **Order by Sequence Rarity**    Orders the column list by the rarity values of the long
      duplicate sequences. Higher rarity values are at the top.

   **Order by Column Name**    Orders the column list in alphabetical order by column name.

   **Original Order**    Orders the column list in the original order in which the columns were
      specified in the launch window.

**Clear Cell Colors**    Clears the cell colors for columns selected in the data table or for all
   columns if no columns are selected in the data table.

See *Using JMP* 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.

### Univariate Patterns Report Options

Each Univariate Patterns red triangle menu contains the following options:

**Select Most Duplicated**    Selects the observations in the data table that correspond to the values in the Most Duplicated Values table.

**Select Longest Runs**    Selects the observations in the data table that correspond to the values in the Longest Runs table.

**Select Longest Duplicate Sequences**    Selects the observations in the data table that correspond to the values in the Longest Duplicate Sequences table.

**Select by Formatted Width**    Selects the observations in the data table that correspond to the values in the Formatted Widths table that have the user-specified overall width.

**Select by Decimal Width**    Selects the observations in the data table that correspond to the values in the Formatted Widths that have the user-specified decimal width.

**Select by Fraction Length**    Selects the observations in the data table that correspond to the values in the Fraction Length table that have the user-specified fraction length.

**Select by Leading Digit**    Selects the observations in the data table that correspond to the values in the Leading and Trailing Digits table that have the user-specified leading digit.

## Additional Example of the Explore Patterns Utility

The Water Treatment.jmp data table contains data from an urban waste water treatment plant. The columns are daily measurements of different sensors in the plant. Use the Explore Patterns Utility to determine whether there are any unexpected patterns in the sensor measurements.

1. Select **Help > Sample Data Library** and open the Water Treatment.jmp sample data table.
2. Select **Analyze > Screening > Explore Patterns**.
3. Select the Sensor Measurements column group and click **Y, Columns**.
4. Click **OK**.
5. In the Control Panel, under Across Columns, click the box next to Duplicates Across Columns.

**Figure 21.17** Duplicates Across Columns Report



The Duplicates Across Columns report shows a long list of duplicate columns. However, the minimum run size is 2 and there are many small runs in the list. You want to focus on the larger runs of at least 10.

6. In the Control Panel, under Settings, enter 10 in the box next to Minimum Cross Column Duplicate Run Size.

---

**Tip:** Click anywhere outside of the box to automatically update the report.

---

**Figure 21.18** Duplicates Across Columns Report With Runs at Least 10



The updated Duplicates Across Columns report shows that there are two duplicate sets with runs of at least 10. Both sets include the PH-P column.

7. Click **Colorize** in the Duplicates Across Columns Report.

**Figure 21.19**  Colorized Water Treatments.jmp Data Table

| | PH-E | DBO-E | DQO-E | SS-E | SSV-E | SED-E | COND-E | PH-P | DBO-P | SS-P | SSV-P | SED-P | COND-P | PH-D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 206 | 7.7 | 124.0 | 234.0 | 120.0 | 61.7 | 1.8 | 1550.0 | 7.9 | 123.0 | 220.0 | 59.0 | 4.0 | 1651.0 | 7.8 |
| 207 | 7.6 | 172.0 | 394.0 | 226.0 | 51.3 | 5.5 | 1324.0 | 7.6 | 197.0 | 302.0 | 47.0 | 4.0 | 1306.0 | 7.7 |
| 208 | 7.6 | 268.0 | 324.0 | 226.0 | 53.1 | 4.0 | 1390.0 | 7.6 | 265.0 | 236.0 | 53.4 | 4.5 | 1389.0 | 7.6 |
| 209 | 7.6 | 176.0 | 200.0 | 176.0 | 50.0 | 3.0 | 859.0 | 7.6 | 149.0 | 172.0 | 51.2 | 2.0 | 860.0 | 7.6 |
| 210 | 7.6 | 146.0 | 243.0 | 164.0 | 57.3 | 3.2 | 1095.0 | 7.6 | 193.0 | 208.0 | 55.8 | 3.0 | 1089.0 | 7.6 |
| 211 | 7.8 | 139.0 | 380.0 | 256.0 | 53.1 | 5.5 | 2040.0 | 7.8 | 201.0 | 340.0 | 47.6 | 8.5 | 2000.0 | 7.9 |
| 212 | 7.9 | 205.0 | 529.0 | 264.0 | 61.4 | 6.5 | 2750.0 | 7.9 | 220.0 | 324.0 | 60.5 | 6.0 | 2850.0 | 7.8 |
| 213 | 7.8 | 117.0 | 366.0 | 500.0 | 27.2 | 4.2 | 3230.0 | 7.8 | 135.0 | 620.0 | 32.3 | 7.0 | 3170.0 | 7.7 |
| 214 | 7.6 | 148.0 | 400.0 | 564.0 | 31.6 | 7.0 | 1554.0 | 7.6 | 235.0 | 798.0 | 28.6 | 11.0 | 1515.0 | 7.6 |
| 215 | 7.8 | 158.0 | 256.0 | 194.0 | 47.4 | 3.0 | 1831.0 | 7.8 | 218.0 | 240.0 | 46.7 | 4.0 | 1860.0 | 7.8 |
| 216 | 7.7 | 123.0 | 246.0 | 186.0 | 39.8 | 2.5 | 1288.0 | 7.7 | 241.0 | 300.0 | 25.3 | 4.5 | 1328.0 | 7.7 |
| 217 | 7.7 | 133.0 | 410.0 | 204.0 | 62.7 | 4.5 | 1490.0 | 7.7 | 161.0 | 256.0 | 57.8 | 4.5 | 1421.0 | 7.7 |
| 218 | 7.5 | 237.0 | 388.0 | 178.0 | 57.3 | 4.0 | 1620.0 | 7.6 | 300.0 | 240.0 | 53.3 | 4.5 | 1571.0 | 7.7 |
| 219 | 7.7 | 203.0 | 357.0 | 222.0 | 58.6 | 3.0 | 1828.0 | 7.6 | 215.0 | 208.0 | 73.1 | 5.0 | 1817.0 | 7.6 |
| 220 | 7.8 | 199.0 | 380.0 | 244.0 | 55.7 | 5.0 | 1453.0 | 7.8 | 110.0 | 196.0 | 60.2 | 3.5 | 1430.0 | 7.8 |
| 221 | 7.8 | 214.0 | 941.0 | 304.0 | 53.3 | 6.5 | 1454.0 | 7.8 | 396.0 | 320.0 | 57.5 | 7.0 | 1528.0 | 7.8 |
| 222 | 7.7 | 208.0 | 298.0 | 260.0 | 37.7 | 4.0 | 1236.0 | 7.8 | 245.0 | 284.0 | 46.5 | 5.0 | 1285.0 | 7.8 |
| 223 | 7.9 | 206.0 | 305.0 | 188.0 | 56.4 | 4.0 | 1200.0 | 7.9 | 220.9 | 272.0 | 50.0 | 6.0 | 1228.0 | 7.9 |
| 224 | 7.7 | 247.0 | 361.0 | 286.0 | 51.0 | 4.5 | 1180.0 | 7.7 | 146.0 | 212.0 | 56.4 | 2.5 | 1251.0 | 7.7 |
| 225 | 7.5 | 438.0 | 681.0 | 370.0 | 56.8 | 6.0 | 1396.0 | 7.6 | 212.0 | 348.0 | 64.4 | 6.0 | 1446.0 | 7.6 |
| 226 | 7.8 | 255.0 | 282.0 | 166.0 | 67.5 | 2.0 | 950.0 | 7.8 | 284.0 | 172.0 | 65.1 | 2.0 | 965.0 | 7.8 |
| 227 | 7.7 | 283.0 | 310.0 | 170.0 | 61.2 | 2.5 | 1065.0 | 7.8 | 235.0 | 270.0 | 51.1 | 5.5 | 1100.0 | 7.8 |
| 228 | 7.9 | 313.0 | 341.0 | 512.0 | 33.6 | 6.0 | 915.0 | 7.9 | 205.0 | 520.0 | 33.8 | 7.5 | 976.0 | 7.9 |
| 229 | 7.7 | 300.0 | 610.0 | 452.0 | 45.1 | 8.0 | 1313.0 | 7.7 | 375.0 | 556.0 | 43.9 | 10.0 | 1312.0 | 7.7 |
| 230 | 7.8 | 177.0 | 412.0 | 196.0 | 69.4 | 4.5 | 2190.0 | 7.8 | 330.0 | 476.0 | 52.1 | 7.5 | 2330.0 | 7.8 |

The Colorize feature colors the cells in the original data table that correspond to the duplicate sets in the report.

# Statistical Details for the Explore Patterns Utility

## Rarity in Longest Runs

To calculate the rarity for longest runs, first define the following variables:

$n$ = the number of rows in the column

$k$ = the number of times a specific value occurs in the column

$p = k/n$ = the probability of observing the specific value in the column

$m$ = the length of the run

$N$ = the number of unique runs

Then, the rarity for longest runs is calculated as follows:

$$Rarity = -\log_2(1 - (1 - p^{m-1})^N)$$

### Rarity in Longest Sequences

To calculate the rarity for longest sequences, first define the following variables:

$p$ = the probability of observing the specific sequence one time in the column

$k$ = the number of times the starting value of the sequence occurs in the column

Then, the rarity for longest sequences is calculated as follows:

$$Rarity = -\log_2(1 - (1 - p)^k)$$

# 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 22.1** Example of a Response Screening Plot
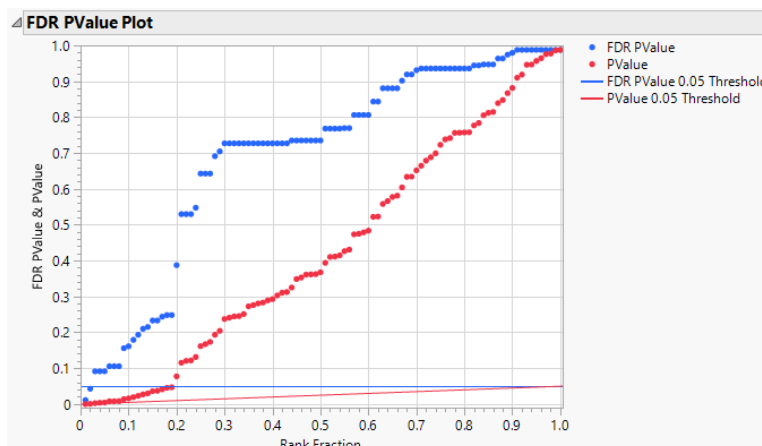
# Contents

# Overview of the Response Screening Platform

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 22.1. 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. 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 22.1** 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 the *JSL Syntax Reference*.

# 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**.

3.  Select the Responses column group and click **Y, Response**.

4.  Select Process and click **X**.

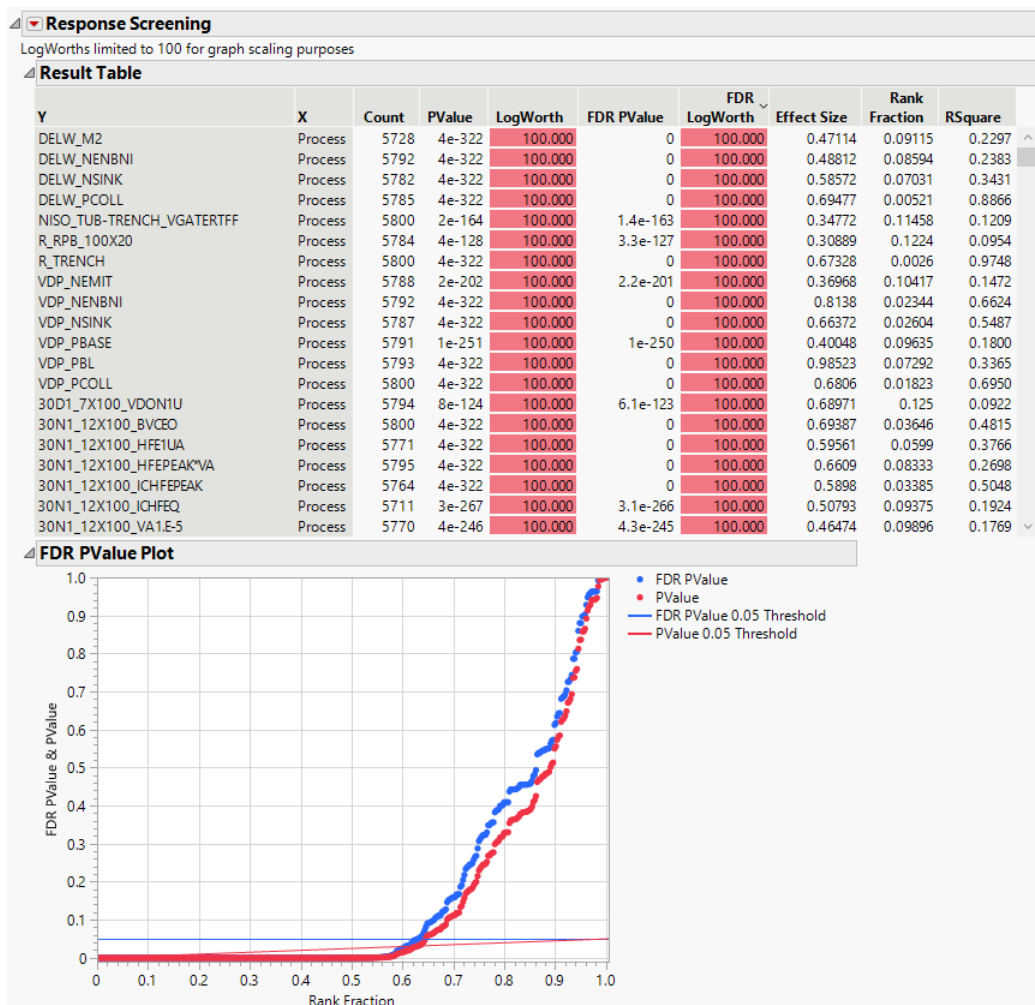5.  Enter 100 in the **MaxLogWorth** box.

    A logworth ($-\log_{10}(p\text{-value})$) of 100 or larger corresponds to an extremely small *p*-value. Setting a value for the MaxLogWorth helps control the scale of plots as it limits the reported logworth value.

6.  Click **OK**.

    The report shows a table of *p*-value results and the FDR PValue Plot, and also contains two other plot reports.

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. In this way, the plot enables you to read FDR significance from either set of *p*-values.

**Figure 22.2** Response Screening Report for 387 Tests against Process

**Response Screening**

LogWorths limited to 100 for graph scaling purposes

**Result Table**

| Y | X | Count | PValue | LogWorth | FDR PValue | FDR LogWorth | Effect Size | Rank Fraction | RSquare |
|---|---|---|---|---|---|---|---|---|---|
| DELW_M2 | Process | 5728 | 4e-322 | 100.000 | 0 | 100.000 | 0.47114 | 0.09115 | 0.2297 |
| DELW_NENBNI | Process | 5792 | 4e-322 | 100.000 | 0 | 100.000 | 0.48812 | 0.08594 | 0.2383 |
| DELW_NSINK | Process | 5782 | 4e-322 | 100.000 | 0 | 100.000 | 0.58572 | 0.07031 | 0.3431 |
| DELW_PCOLL | Process | 5785 | 4e-322 | 100.000 | 0 | 100.000 | 0.69477 | 0.00521 | 0.8866 |
| NISO_TUB-TRENCH_VGATERTFF | Process | 5800 | 2e-164 | 100.000 | 1.4e-163 | 100.000 | 0.34772 | 0.11458 | 0.1209 |
| R_RPB_100X20 | Process | 5784 | 4e-128 | 100.000 | 3.3e-127 | 100.000 | 0.30889 | 0.1224 | 0.0954 |
| R_TRENCH | Process | 5800 | 4e-322 | 100.000 | 0 | 100.000 | 0.67328 | 0.0026 | 0.9748 |
| VDP_NEMIT | Process | 5788 | 2e-202 | 100.000 | 2.2e-201 | 100.000 | 0.36968 | 0.10417 | 0.1472 |
| VDP_NENBNI | Process | 5792 | 4e-322 | 100.000 | 0 | 100.000 | 0.8138 | 0.02344 | 0.6624 |
| VDP_NSINK | Process | 5787 | 4e-322 | 100.000 | 0 | 100.000 | 0.66372 | 0.02604 | 0.5487 |
| VDP_PBASE | Process | 5791 | 1e-251 | 100.000 | 1e-250 | 100.000 | 0.40048 | 0.09635 | 0.1800 |
| VDP_PBL | Process | 5793 | 4e-322 | 100.000 | 0 | 100.000 | 0.98523 | 0.07292 | 0.3365 |
| VDP_PCOLL | Process | 5800 | 4e-322 | 100.000 | 0 | 100.000 | 0.6806 | 0.01823 | 0.6950 |
| 30D1_7X100_VDON1U | Process | 5794 | 8e-124 | 100.000 | 6.1e-123 | 100.000 | 0.68971 | 0.125 | 0.0922 |
| 30N1_12X100_BVCEO | Process | 5800 | 4e-322 | 100.000 | 0 | 100.000 | 0.69387 | 0.03646 | 0.4815 |
| 30N1_12X100_HFE1UA | Process | 5771 | 4e-322 | 100.000 | 0 | 100.000 | 0.59561 | 0.0599 | 0.3766 |
| 30N1_12X100_HFEPEAK^VA | Process | 5795 | 4e-322 | 100.000 | 0 | 100.000 | 0.6609 | 0.08333 | 0.2698 |
| 30N1_12X100_ICHFEPEAK | Process | 5764 | 4e-322 | 100.000 | 0 | 100.000 | 0.5898 | 0.03385 | 0.5048 |
| 30N1_12X100_ICHFEQ | Process | 5711 | 3e-267 | 100.000 | 3.1e-266 | 100.000 | 0.50793 | 0.09375 | 0.1924 |
| 30N1_12X100_VA1.E-5 | Process | 5770 | 4e-246 | 100.000 | 4.3e-245 | 100.000 | 0.46474 | 0.09896 | 0.1769 |

**FDR PValue Plot**

The FDR PValue Plot shows that more than 60% of the tests are significant. A handful of tests are significant using the unadjusted *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 Result table, where the names of the characteristics are given in the first column.

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

The remaining columns give information about the test of Y against X. Here the test is a Oneway Analysis of Variance. In addition to other information, the table gives the test's *p*-value, LogWorth, FDR (False Discovery Rate) *p*-value, and FDR LogWorth. 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" for more information about the report and PValues table.

# Launch the Response Screening Platform

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

**Figure 22.3** Response Screening Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

## 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**    Analyzes the rows assigned to each level of the specified column separately. All results are presented 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. See *Fitting Linear Models*.

**Freq**    Identifies a column whose values assign a frequency to each row. These values enable you to account for pre-summarized data. See *Fitting Linear Models*.

**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

**PValues Table on Launch**    Creates a data table for the *p*-values and individual model fit statistics. This data table is linked to the Result Table in the Response Screening report.

**Robust**    (Available only for continuous responses.) Fits regression and ANOVA models using the Huber M-estimation method (Huber and Ronchetti 2009). This method estimates parameters by minimizing the Huber loss function. The Huber loss function penalizes outliers, which reduces the influence of the outliers on tests. If there are no outliers, the estimates are close to the least squares estimates. When you select the Robust option, several Robust measures and model fit statistics columns are added to the Result Table. See *Basic Analysis* for more information about Huber M-estimation. For an example, see "Example of Robust Fit".

---

**Caution:** The Robust option can increase processing time.

---

**Cauchy**    (Available only for continuous responses.) Estimates parameters using maximum likelihood and a Cauchy link function. This estimation method assumes that the errors have a Cauchy distribution, which 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. When you select the Cauchy option, several Cauchy measures and model fit statistics columns are added to the Result Table.

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

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

**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.

**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".

**MaxLogWorth**   Controls 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" for an example.

**Advanced Options**   Contains the following options:

**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.

If X and Y are both binary, the Pearson product-moment calculation gives Spearman's rho and 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.

**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.

**Unthreaded**   Suppresses multithreading used for computational speed.

# The Response Screening Report

The Response Screening report consists of a results table and several plots. These plots focus on False Discovery Rate (FDR) statistics. See "The False Discovery Rate".

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 FDR LogWorth plot is presented to help assess the impact of using the robust fit.

# Result Table

The Response Screening Result table contains a row for each pair of Y and X variables. The columns of the table contain measures and model fit statistics that are specific to the selected fit and Y and X modeling types.

**Group**    (Appears only if there is a grouping variable.) The level of the grouping column.

**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 more information about Fit Y by X statistics, see *Basic Analysis*.

**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 (-$\log_{10}$(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 more information about the FDR correction, see Benjamini and Hochberg (1995). For more information about the false discovery rate, see "The False Discovery Rate".

**FDR LogWorth**    The quantity -$\log_{10}$(FDR PValue). This is the statistic to use 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 from the hypothesis test divided by a robust estimate of the response standard deviation. If the interquartile range (IQR) is nonzero and IQR > range/20, the standard deviation estimate is IQR/1.3489795. Otherwise 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. The Rank Fraction is used in plotting the PValues and FDR PValues in rank order of decreasing significance.

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

**Kappa**   (Appears only when Y and X are both categorical and have the same number of levels.) A measure of agreement between Y and X.

**Corr**   (Appears only when Y and X are both categorical.) The Pearson product-moment correlation in terms of the indices defined by the value ordering.

The following columns are added to the Result table when the Robust option is selected in the launch window. The Robust option applies only when Y is continuous, so Robust column cells are empty when Y is categorical.

**Robust PValue**   The *p*-value for the significance test corresponding to the pair of Y and X variables using a robust fit.

**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.

The following columns are added to the Result table when the Cauchy option is selected in the launch window. The Cauchy option applies only when Y is continuous, so Cauchy column cells are empty when Y is categorical.

**Cauchy PValue**   The *p*-value for the significance test corresponding to the pair of Y and X variables using a Cauchy fit.

**Cauchy LogWorth**   The quantity $-\log_{10}$(Cauchy PValue).

**Cauchy FDR PValue**   The False Discovery Rate calculated for the Cauchy 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.

**Cauchy FDR LogWorth**   The quantity $-\log_{10}$(Cauchy FDR PValue).

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

**Cauchy ChiSquare**   The chi-square value associated with the test from the Cauchy fit.

**Cauchy Sigma**   The estimate of the error standard deviation from the Cauchy fit.

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

## 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 vertical 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 22.4 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 22.4** FDR PValue Plot
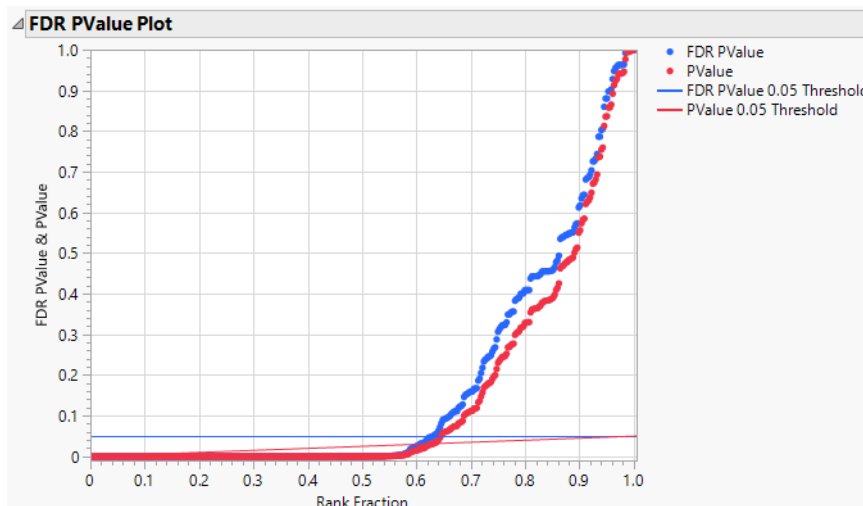


## FDR LogWorth by Effect Size

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 ($-\log_{10}(p\text{-value})$) scale, highly significant *p*-values have large LogWorths and nonsignificant *p*-values have low LogWorths. A LogWorth of zero corresponds to a nonsignificant *p*-value of 1. Any LogWorth above 2 corresponds to a *p*-value below 0.01.

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

Figure 22.5 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 22.5** FDR LogWorth by Effect Size
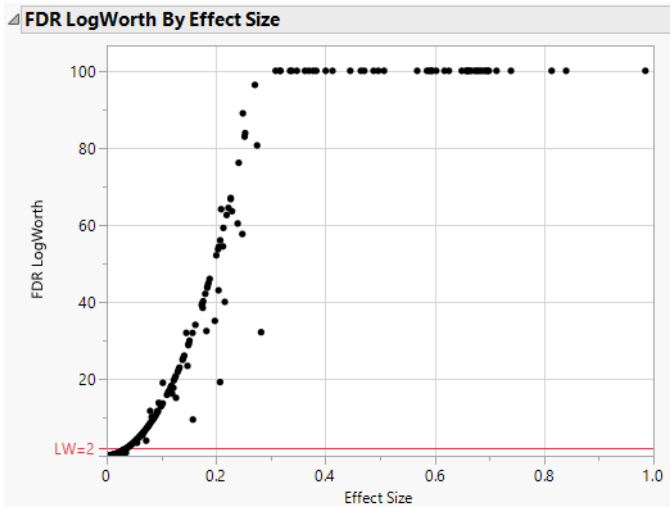


## FDR LogWorth by RSquare

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

If the PValues Tables on Launch option is selected in the launch window, a PValues data table is created when the Response Screening platform is launched. This data table contains the same information as the Result table that is in the Response Screening report. The PValues data table also contains the additional, model specific columns:

**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.

**Intercept**    Appears when Y and X are both continuous. The intercept of the regression model relating the corresponding pair of Y and X variables.

**Slope**   Appears when Y and X are both continuous. The slope of the regression model relating the corresponding pair of Y and X variables.

**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.

**Robust Elapsed Time**   (Appears only when Robust is specified in the launch window.) Computing 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.

**Results Table**   Shows or hides the table of results.

**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 points in the plots, rows in the Result Table, or if opened, rows in the PValue data table.

**Select Columns**   Selects the columns in the original data table that correspond to rows that you select in the Result Table, to points that you select in plots in the Response Screening report window, or if opened, rows in the PValues table. 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.

**Select Where**   Opens the Select Where window. You can select specific responses in the Result table that correspond to a particular condition by using the Comparison menu and Value text box. For example, you can select all responses such that Effect Size > 0.80. After you click OK, the responses are selected in the Result table.

> **Tip:** You can also access the Select Where window by right-clicking anywhere in the Result table.

**Save PValues**    Creates the PValues data table. See "The PValues 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". For an example, see "Example of Tests of Practical Significance and Equivalence".

**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 *Using JMP* 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 22.6**  Means Data Table



The Means data table includes the following columns:

**Y**   The continuous response variables.

**X**   The categorical variables.

**Level**   The level of the categorical X variable.

**Count**   The count of values in the corresponding Level.

**Mean**   The mean of the Y variable for the specified Level.

**StdDev**    The standard deviation of the Y variable for the specified Level.

**Robust Mean**    The robust M-estimate of the mean. Appears when you select the Robust option on the launch window.

# Compare Means Data Table

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

Equivalence tests enable you to determine whether two levels have essentially the same effect, from a practical perspective, on the response. In other words, an equivalence test tests whether the difference is smaller than the practical difference.

The Compare Means data table provides results for both tests of practical difference and tests of practical equivalence. Each row compares a response across two levels of a categorical factor. Results of the pairwise comparisons are color-coded to facilitate interpretation. See "Practical Difference" for a description of how the practical difference is specified. See "Example of Tests of Practical Significance and Equivalence" for an example.

**Figure 22.7** Compare Means Data Table



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

**Y**    The continuous response variables.

**X**    The categorical variables.

**Leveli**    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"
- "The Fit Response Screening Report"
- "Fit Response Screening Options"
- "PValues Data Table"
- "Y Fits Data Table"

# Launch Response Screening in Fit Model

Select **Analyze > Fit Model**. Enter your Ys and model effects. Select **Response Screening** from the Personality list.

**Figure 22.8**  Response Screening from the Fit Model Window
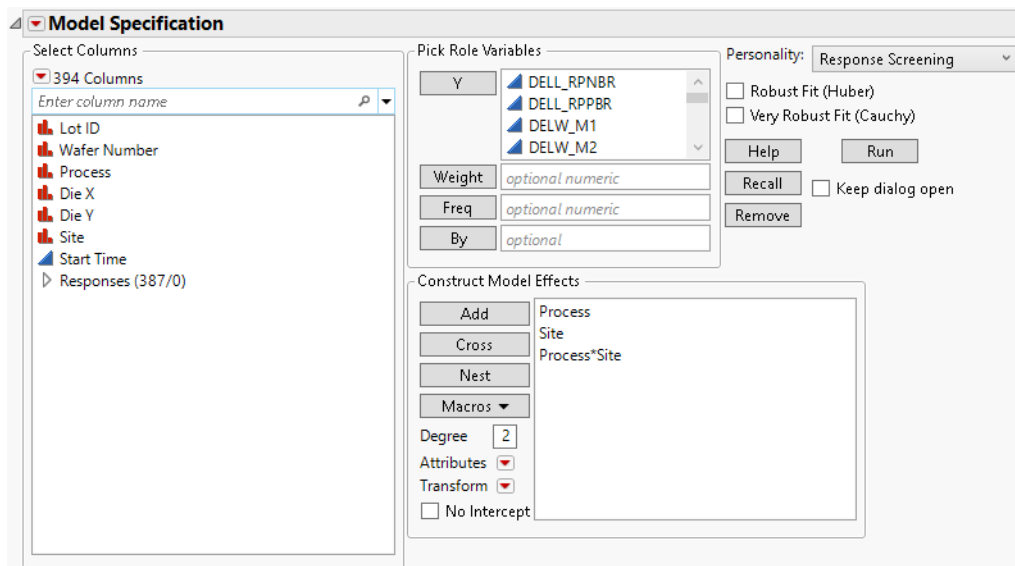


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

There are two options for robust estimation:

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

**Very Robust Fit (Cauchy)**   Specifies very robust (Cauchy) estimation to down weight outliers for continuous responses. 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.

**Tip:** If both robust options are selected, the platform uses only Cauchy estimation.

The **Informative Missing** option provides a coding system for missing values. 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 22.9** Informative Missing Option
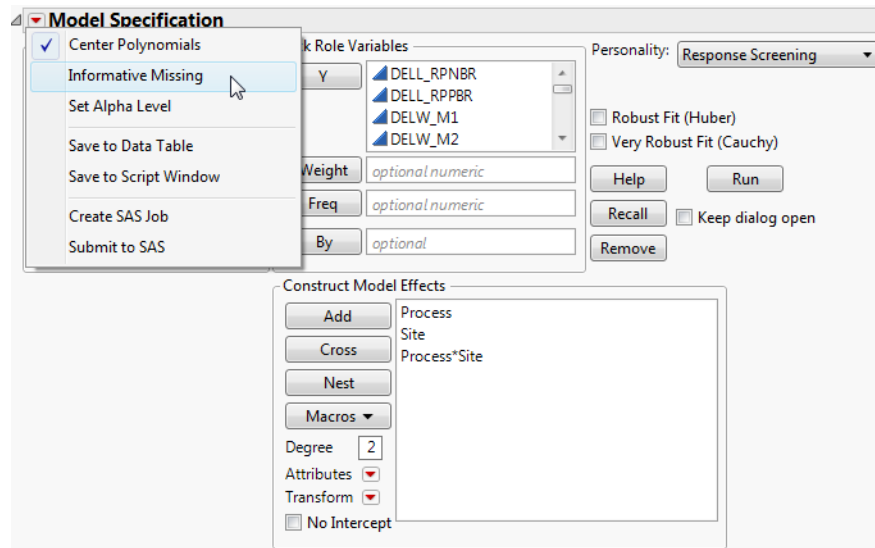


For more information about the Fit Model window, see *Fitting Linear Models*.

# 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".

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".

# Fit Response Screening Options

The Fit Response Screening red triangle menu contains options to save the calculated data.

**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 *Using JMP* 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 *Fitting Linear Models* for more information about Effect Tests.

**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. For more information about the FDR correction, see Benjamini and Hochberg (1995). For more information about the false discovery rate, see "The False Discovery Rate" or Westfall et al. (2011).

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

**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".)

**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

## 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. Click the Response Screening red triangle and select **Save Compare Means**.

   Figure 22.10 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 22.10**  Compare Means Table, Partial View

| | 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 | 0.1486469486 | 1 | 1.57079e-153 | Equivalent |
| 2 | DELL_RPPBR | New | Old | Process | -0.024751371 | 0.0349025251 | 0.4782555718 | 1.1939335307 | 1 | 2.40983e-225 | Equivalent |
| 3 | DELW_M1 | New | Old | Process | 0.0275288913 | 0.0205582035 | 0.1806001982 | 0.7030518329 | 1 | 1.40119e-217 | Equivalent |
| 4 | DELW_M2 | New | Old | Process | 0.0739317751 | 0.0017890895 | 0 | 0.0704822866 | 0.0269479498 | 0.9730520502 | Different |
| 5 | DELW_NBASE | New | Old | Process | -1.280753006 | 1.4836374126 | 0.388034756 | 50.689291592 | 1 | 6.62236e-223 | Equivalent |
| 6 | DELW_NEMIT | New | Old | Process | -0.040133074 | 0.002521398 | 7.306034e-56 | 0.0848381437 | 1 | 7.74847e-69 | Equivalent |
| 7 | DELW_NENBNI | New | Old | Process | -0.93842129 | 0.0220490008 | 0 | 0.864024229 | 0.0003726256 | 0.9996273744 | Different |
| 8 | DELW_NSINK | New | Old | Process | 1.3973855188 | 0.0254310074 | 0 | 1.0723100972 | 3.228438e-37 | 1 | Different |
| 9 | DELW_PBASE | New | Old | Process | -0.043649097 | 0.0037311377 | 2.908704e-31 | 0.1296380428 | 1 | 8.21675e-113 | Equivalent |
| 10 | DELW_PCOLL | New | Old | Process | 1.8169399617 | 0.0085444762 | 0 | 1.17539613 | 0 | 1 | Different |
| 11 | DELW_PEMIT | New | Old | Process | -0.022697392 | 0.0534170363 | 0.6709186367 | 1.8265075627 | 1 | 1.2499e-228 | Equivalent |
| 12 | DELW_PSINK | New | Old | Process | 0.911093371 | 0.0678503246 | 1.666427e-40 | 2.3558597461 | 1 | 3.147655e-97 | Equivalent |
| 13 | DELW_RPNBR | New | Old | Process | -0.101421251 | 0.0100452789 | 9.008231e-24 | 0.3464883523 | 1 | 1.6441e-125 | Equivalent |
| 14 | DELW_RPPBR | New | Old | Process | 0.3015474189 | 0.0280757905 | 1.169725e-26 | 0.9698077792 | 1 | 7.48088e-120 | Equivalent |
| 15 | DELW_SICR | New | Old | Process | 0.0275370732 | 0.0176173979 | 0.1180928699 | 0.6024110643 | 1 | 7.77189e-215 | Equivalent |
| 16 | M1_COMB_VGAT... | New | Old | Process | -0.532453055 | 0.6037284633 | 0.3778451214 | 20.662273741 | 1 | 1.83138e-223 | Equivalent |
| 17 | M1_TRENCH_VG... | New | Old | Process | -0.61657228 | 0.1695447206 | 0.0002786205 | 5.8087989555 | 1 | 2.39165e-191 | Equivalent |
| 18 | M2/M1_CAP_VG... | New | Old | Process | -0.047230999 | 0.0978790012 | 0.6294372314 | 3.3496974301 | 1 | 2.54456e-228 | Equivalent |
| 19 | M2_COMB_BB_V... | New | Old | Process | 0.0001788376 | 0.1270241159 | 0.9988767052 | 4.3470387459 | 1 | 3.07472e-234 | Equivalent |
| 20 | M2_COMB_VGAT... | New | Old | Process | 0.2971147541 | 0.2860872543 | 0.2990592705 | 9.7914324315 | 1 | 1.42412e-221 | Equivalent |
| 21 | NISO_TUB-TREN... | New | Old | Process | 7.0118092399 | 0.2482751793 | 1.58784e-164 | 9.0621071628 | 1 | 9.077234e-17 | Equivalent |
| 22 | NISO_TUB-TUB_V... | New | Old | Process | -0.074789866 | 0.1081535234 | 0.4892691678 | 3.7013990674 | 1 | 8.97914e-226 | Equivalent |
| 23 | PS_RPNBR | New | Old | Process | -64.69026527 | 15.2943887 | 0.0000237641 | 530.27422027 | 1 | 3.12381e-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 22.11 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 22.11**  Distribution of Practical Significance Results
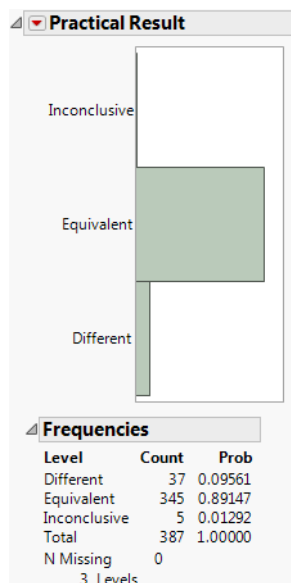


The 37 responses can be selected for further study by clicking on the corresponding bar in the plot.

# Example of the MaxLogWorth Option

Use the Response Screening MaxLogWorth option to control the LogWorth scale from being distorted by very large values. 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. In some data sets the *p*-values are so small that the LogWorth scale is distorted by very large 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 might 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 difficult to see, because of the huge Robust FDR LogWorth value of about 58,000. To ensure that your graphs show sufficient detail, you can set a maximum value of the LogWorth.

**Figure 22.12** Robust FDR LogWorth vs. Effect Size, MaxLogWorth Not Set
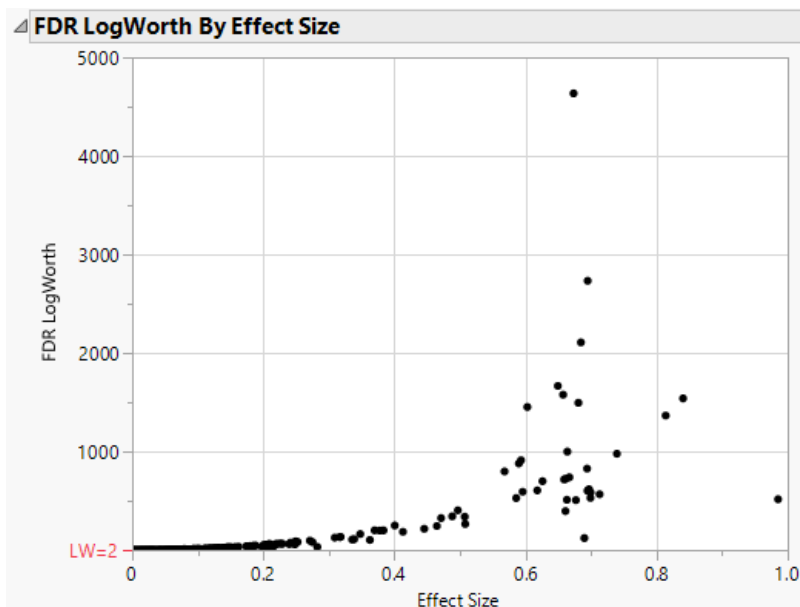
8. Repeat step 1 through step 5.

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

10. Click **OK**.

   The analysis might 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 22.13**  Robust FDR LogWorth vs. Effect Size, MaxLogWorth = 1000
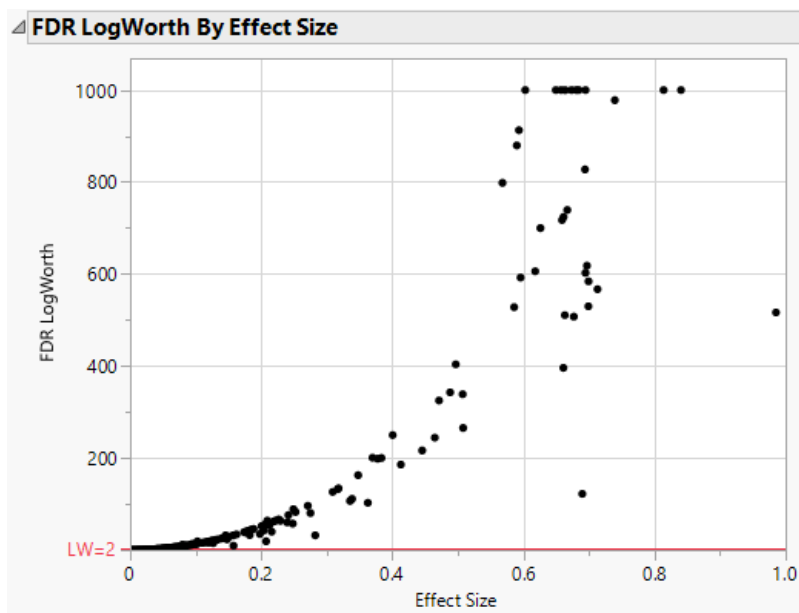


## Example of Robust Fit

1. Select **Help > Sample Data Library** and open Drosophila Aging.jmp.

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**.

**Figure 22.14**  Robust FDR PValue Plot for Drosophila Data
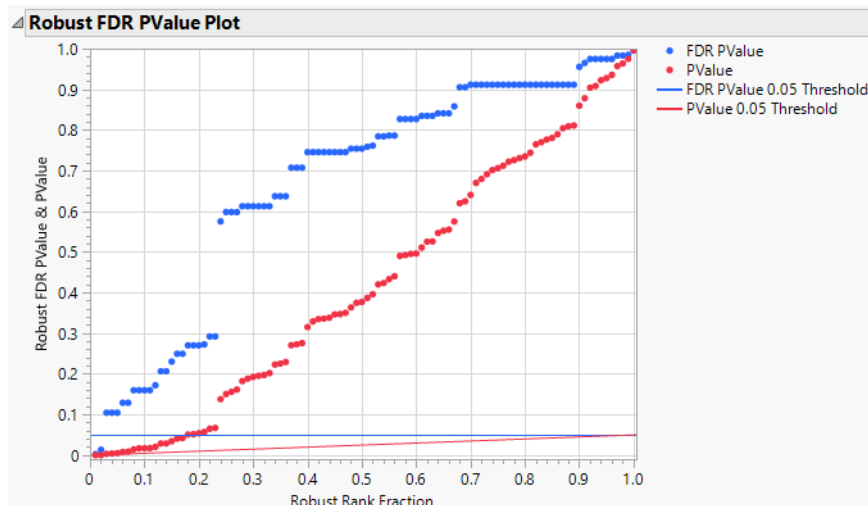


Note that a number of tests are significant using the unadjusted robust *p*-values, as indicated by the red points that are less than 0.05. However, only two tests are significant according to the robust FDR *p*-values. These two points are more easily identified in a plot that shows FDR LogWorths.

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

8.   Drag a rectangle around the point with a Robust FDR LogWorth value that exceeds 1.5.

9.   In the graph, right click and select **Row Label**.

**Figure 22.15**  Robust LogWorth by Effect Size for Drosophila Data
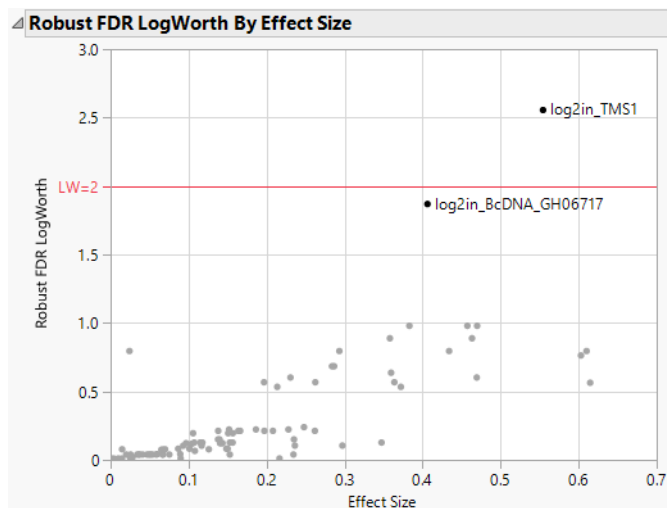
Points above the red line at 2 have significance levels below 0.01 (-log$_{10}$(0.01) = 2). A horizontal line at about 1.3 corresponds to a 0.05 significance level.

10. Click the Robust LogWorth by LogWorth disclosure icon.

**Figure 22.16** Robust LogWorth by LogWorth for Drosophila Data
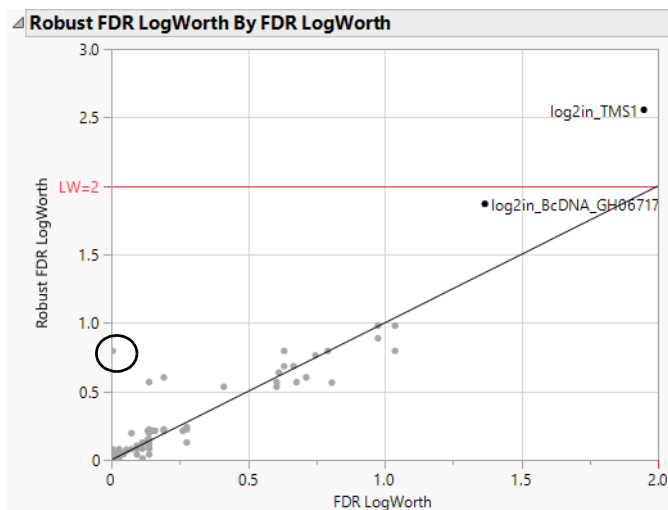


If the robust test for a response were identical to the usual test, its corresponding point would fall on the diagonal line in Figure 22.16. The circled point in the plot does not fall near the line, because it has a Robust LogWorth value that exceeds its LogWorth value.

11. Drag a rectangle around this point in the plot.

12. Find the row for this point in the Result table.

    Note that the response, log2in_CG8237 has PValue 0.9568 and Robust PValue 0.0176.

13. Click the Response Screening red triangle and select **Fit Selected Items**.

    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 22.17). 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 22.17**  Oneway Analysis for log2in_CG8237
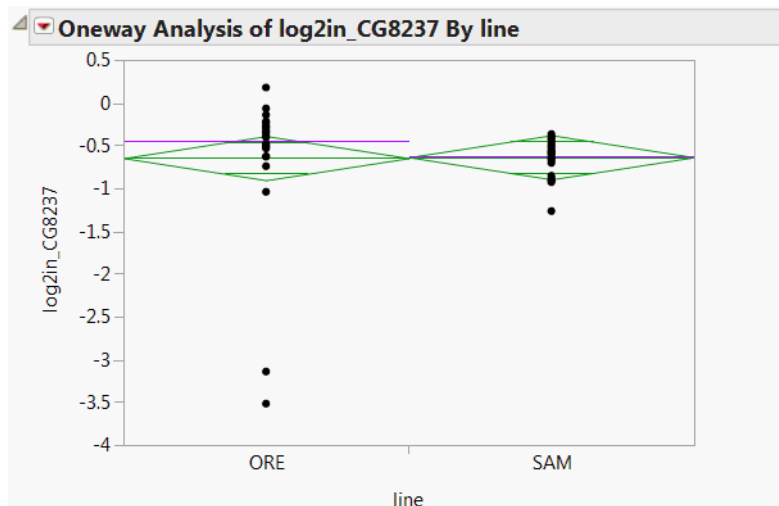


## Response Screening Personality

The Response Screening personality in Fit Model enables you to study tests of multiple responses against linear model effects.

1. Select **Help > Sample Data Library** and open Drosophila Aging.jmp.

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 22.18 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 22.18** FDR LogWorth vs Rank Fraction Plot with line*age Tests Selected
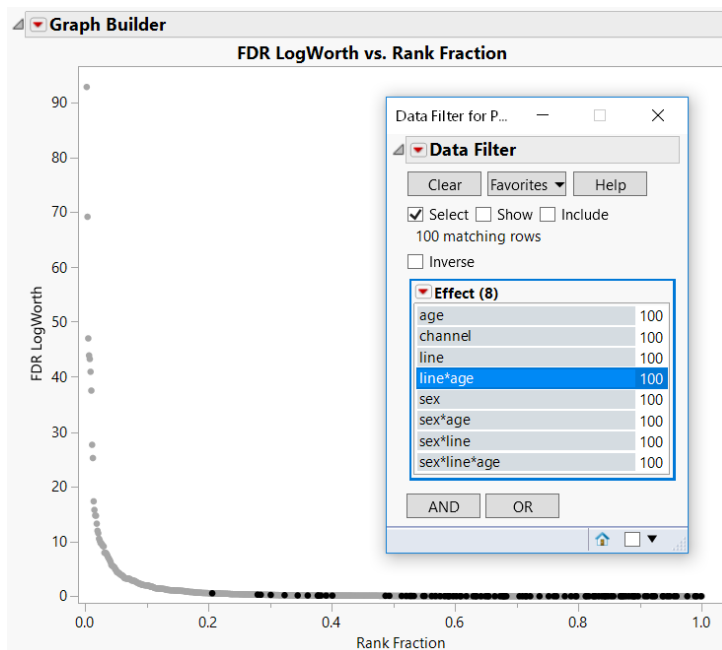


# Statistical Details for the Response Screening Platform

## 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 $\alpha$:

1. Conduct the $m$ hypothesis tests of interest to obtain $p$-values $p_1, p_2,..., p_m$.

2. Rank the $p$-values from smallest to largest. Denote these by $p_{(1)} \leq p_{(2)} \leq ... \leq p_{(m)}$.

3. Find the largest $p$-value for which $p_{(i)} \leq (i/m)\alpha$. Suppose this first $p$-value is the $k^{\text{th}}$ largest, $p_{(k)}$.

4. Reject the $k$ hypotheses associated with $p$-values less than or equal to $p_{(k)}$.

This procedure ensures that the false discovery rate does not exceed $\alpha$.

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

$$
p_{(i), FDR} = \begin{cases} p_{(m)} & \text{for } i = m \\ min\left[ p_{(i+1), FDR}, \left(\frac{m}{i}\right) p_{(i)} \right] & \text{for } i = m - 1, \ldots, 1 \end{cases}
$$

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

# 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 23.1** Example of a Process Performance Graph
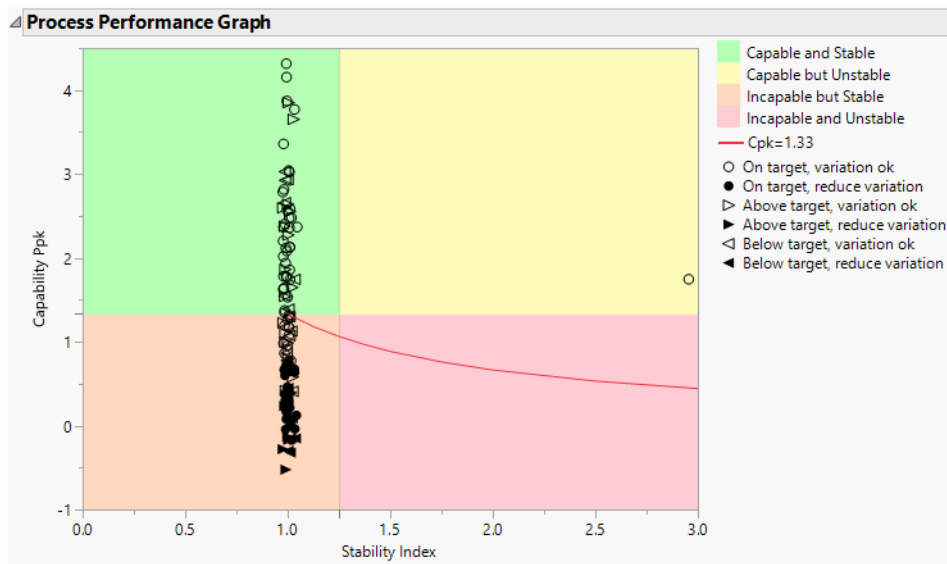
# Contents

# Overview of the Process Screening Platform

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 do the following:

- Specify a constant subgroup size or use a variable containing subgroup identifiers for control chart calculations using subgroups.

- Use grouping variables. For each combination of values of the group variables, an analysis is provided for each process variable.

- 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 specific control chart tests, 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.

**Tip:** For information about adding specification limits, see *Using JMP*.

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 identifying unstable processes. 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 23.2** Partial View of Initial Report



The Process Screening window appears, showing a table of results for each process. The table is sorted by Stability Index. This is a measure of the stability of a process, where a stable process has a stability index near 1. Higher values of the stability index indicate a less stable process. (The sorting is indicated by the caret beside Stability Index in the report.) You want to take a closer look at processes with a stability index value greater than or equal to 1.03.

5. In the report window, select processes PMS1 through IVP9.

Each of these first 11 processes has a value of 1.03 or larger in the Stability Index column.

6. Right-click the selected processes and select **Quick Graph for Selected Items**.

**Figure 23.3** Quick Graphs for Highest Alarm Rate Processes
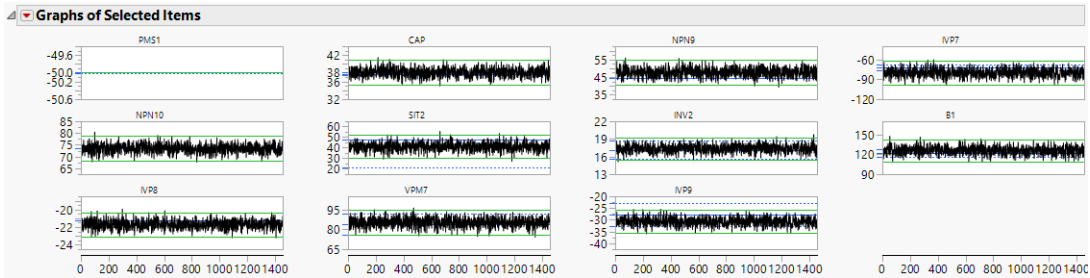


You decide to take a closer look at IVP8 (row 3, column 1 in the Graphs of Selected Items).

7.   Select the second process in the Summary table, which corresponds to IVP8.

8.   Right-click the selected process and select **Control Charts for Selected Items**.

**Figure 23.4**  Control Chart Builder Report for IVP8
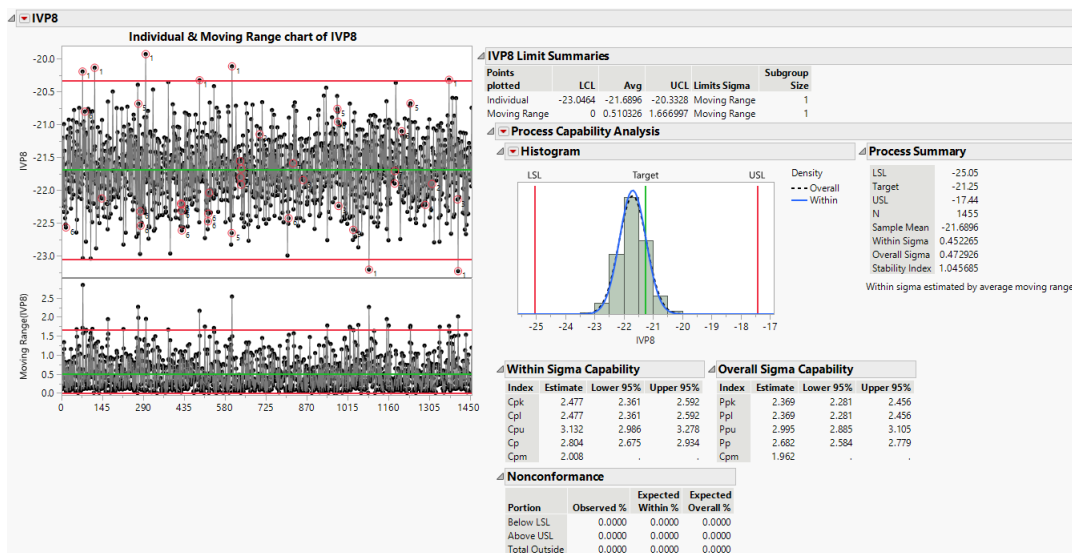


A Control Chart Builder report appears. Because IVP8 has a Spec Limits column property, the report also includes a capability analysis.

# Launch the Process Screening Platform

Launch the Process Screening platform by selecting **Analyze > Screening > Process Screening**.

**Figure 23.5**  Process Screening Launch Window



For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

## Launch Window Roles

**Process Variables**    The columns of process data containing the measurements to be analyzed. The columns must have a Numeric data type. If a process variable has a Control Limits column property, the Process Screening platform uses those limits to calculate the Specified Sigma.

**Note:** The platform does not support control limits that are specified in a Control Limits column property for dispersion (R) charts.

**Grouping**   Columns assigned as grouping variables. Each process variable is analyzed at each combination of levels of the grouping variables. The results are presented in a single report.

**Subgroup**   Assigns one or more subgroup variables. If more than one subgroup variable is assigned, the subgroup levels are defined as a combination of the multiple columns.

**Note:** The Subgroup role is ignored for Indiv and MR charts.

**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 time stamp is 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 five control chart types: Indiv and MR (Individual Measurement and Moving Range); XBar and R; XBar and S; XBar, MR, and R; or XBar, MR, and S. The XBar, MR, and R control chart and the XBar, MR, and S control chart are also referred to as three way control charts. For more information about statistical details, see *Quality and Process Methods*.

**Note:** If the subgroup size for a process is 1, the chart automatically switches to an Indiv and MR chart.

**Subgroup Sample Size**   Specifies 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.

**KSigma**   Specifies the sigma multiplier. KSigma is the value that is multiplied by sigma in the calculation of control limits. By default, KSigma is 3.

**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 Choose limits table window appears. Once you choose a limits tables and click OK, a Limits Specification window appears. Assign columns in your limits table to appropriate roles and click OK. See "Limits Table".

---

**Note:** If you do not select this option, limits are obtained from the Control Limit and Specification Limit column properties for the Process Variables. If you do not select this option and the Process Variables do not have control limit column properties, the control limits are calculated from the data.

---

**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 for the Process Screening Platform". 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 or the Process Capability for Selected Items red triangle menu options do not match the Process Screening results.

---

**Sort by Subgroup**    Sorts the process data by the subgroup variable, or combination of nested subgroup variables, before calculations are performed.

**Advanced Options**    Contains the following options:

**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".

**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".

**Shift Lambda**    Enables you to change the exponentially weighted moving average (EWMA) weight used in the Shift Graph. See "Shift Magnitudes and Positions".

**Drift Beta**　The weight used in the exponentially weighted moving average (EWMA) for drift detection. Drift Beta is set to 0.05 by default.

**Minimum Process Length**　The minimum number of data values that a process must have in order to be included in the analysis. By default, this value is set to 3.

## 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 do not need to specify variables for all of these roles. All of these roles are optional.

**Figure 23.6** Limits Specifications Window



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 use the Derived Sigma options.

**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. This is labeled as Specified Sigma in the Process Screening report.

**Derived Sigma**   Calculates sigma based on the given control limits and subgroup size. This is labeled as Specified Sigma in the Process Screening report.

---

**Note:** If there is a specified sigma, the within-sigma, stability index, and stability ratio values do not match between the Process Screening and Control Chart Builder platforms. The Process Screening platform uses the specified sigma to calculate the control limits, but still calculates the within-sigma using the data. The Control Chart Builder platform uses the specified sigma as the historical sigma and in control charts, the historical sigma is used as the within-sigma.

---

The derived sigma is calculated as follows:

$\quad$ Sigma = $d*(UCL - LCL)/6$

where $d$ is the square root of the subgroup size.

**LCL**   A column containing the lower control limits for each process.

**UCL**   A column containing the upper control limits for each process.

**Subgroup Size**   A column containing the subgroup size for each process.

**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.

**Importance**   A column containing an importance value for each process. The Importance column provides a mechanism to sort processes in the order that you prefer.

# The Process Screening Report

The Process Screening report opens with a Summary table that contains results about process stability. There is a label above the summary table that shows the type of control chart on which the results are based. The sample size or subgroup variable is also shown above the summary table for every chart type except for Indiv and MR. The summary table also contains capability results if you have provided specification limits. The processes and groups are initially sorted in decreasing order by Stability Index. If Importance values are specified by the user, the processes and groups are sorted in decreasing order by Stability Index within Importance. The columns for Stability Index, Ppk, Cpk, Cp, and Target Index are colored as green, yellow, and red to indicate adequate, marginal, and poor stability or capability, respectively. This color coding scheme matches the Process Performance graph color coding scheme.

**Tip:** To sort the report by a column, click 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 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. If you use the Medians instead of Means option, the center line is given by the median of the observations.

- Control limits are placed at *K* sigma units from the center line. Use the KSigma option in the launch window to specify *K*. By default, KSigma is 3.

- 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".

**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.

**Tip:** To access options that operate on selected items, right-click the **Column** 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.

**Importance**   (Appears only when there are importance values specified by the user, as inputs from a limits table.) The user-specified importance value for the processes.

**Variability**   Contains the following columns:

**Stability Index**   A measure of stability of the process. A stable process has a stability index near one. Higher values indicate less stability. The stability index is defined as follows:

(Overall Sigma / Within Sigma)

If a three way control chart is selected in the launch, the stability index is defined as follows:

(Overall Sigma / Between-and-Within Sigma)

**Stability Ratio**   A measure of stability of the process. A stable process has a stability ratio near one. Higher values indicate less stability. The stability ratio is defined as follows:

(Overall Sigma / Within Sigma)$^2$

If a three way control chart is selected in the launch, the stability ratio is defined as follows:

(Overall Sigma / Between-and-Within Sigma)$^2$

**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 *Quality and Process Methods* for statistical details. If you select Medians instead of Means, Within Sigma is computed as described in "Use Medians instead of Means".

**Overall Sigma**   The usual estimate of standard deviation based on all observations.

**Between Sigma**   (Appears only when three way control chart is selected in the launch window.) An estimate of the standard deviation based on the variation between subgroups. See *Quality and Process Methods* for statistical details.

**Between-and-Within Sigma**   (Appears only when three way control chart is selected in the launch window.) An estimate of the standard deviation based on the variation between subgroups and the variation within subgroups. The Between-and-Within Sigma estimate is defined as follows:

$$\text{Between-and-Within Sigma} = \sqrt{\text{Within Sigma}^2 + \text{Between Sigma}^2}$$

**Specified Sigma**   The standard deviation specified by a sigma control limit in the Limits Specification dialog or an estimate of the standard deviation derived from the control limits and subgroup size. See "Derived Sigma". The control limits and subgroup size can be specified using a limits table or a Control Limits column property.

**Summary**   Contains the following columns:

**Centerline**   (Appears if you do one of the following: select Use Medians instead of Means in the launch window, import a Center value using a limits table, or import control limits to derive sigma.) 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.

– If you import control limits, the value displayed is calculated as (UCL + LCL)/2.

**Mean**   The average of all observations.

**Count**   The number of observations.

**N Subgroups**   The number of subgroups.

**Control Chart Alarms**   Contains information about the subgroups that result in alarms for a variety of tests, including each of the 8 Western Electric rules. 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 non-missing 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 14th 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 15th 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.

**Moving Range Limit Exceeded**   The number of subgroups that exceed the moving range limit on the three way control 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:

**Ppk**   Capability index based on Overall Sigma and assuming a normal distribution. See *Quality and Process Methods* for statistical details. On by default.

**Cpk**   Capability index based on Within Sigma or Between-and-Within Sigma and assuming a normal distribution. See *Quality and Process Methods* for statistical details. On by default.

**Cp**   The potential capability if target and drift issues are resolved. See *Quality and Process Methods* for details.

**Target Index**   The number of short-term standard deviations that the process average differs from the target value. This measures the ability of the process to hit the target value. The Target Index is calculated as 3(Cp - Cpk). A target index is considered poor if above 1, marginal if between 0.5 and 1, and adequate if less than 0.5. See White et al. (2018).

**Out of Spec Count**   The number of observations that fall outside the specification limits. On by default.

**Out of Spec Rate**   The proportion of observations that fall outside the specification limits. On by default.

**Expected Out of Spec Rate**   The expected proportion of observations that fall outside of the specification limits. The Expected Out of Spec Rate assumes a stable and normally distributed process and uses overall sigma.

**Latest Out of Spec**   The number of observations, counting from the last observation to the most recent observation that falls outside the specification limits. On by default.

**(Mean-Tgt)/SpecRange**   The spec centered mean. This is the same as the Mean Shift Standardized to Spec in Process Capability. See *Quality and Process Methods* for statistical details.

**StdDev/SpecRange**   The spec scaled standard deviation. This is the same as the Std Deviation Standardized to Spec in Process Capability. See *Quality and Process Methods* for statistical details.

**LSL**   The lower specification limits.

**USL**   The upper specification limits.

**Target**   The target value.

**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 EWMA smoothing approach for the individual observations. This is the algorithm:

– 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 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.

**Drift Magnitudes and Positions**   (Shown only if you have selected the Drift Summaries option from the Process Screening red triangle menu.) Drift detection is performed to detect smaller, more gradual changes in processes. The algorithm is identical to the one used in shift detection, except that drift detection uses a Holt Double-Exponential Smoother instead of an EWMA. This is the algorithm:

– Outliers are removed through the same process that is used in shift detection. See "Shift Magnitudes and Positions".

– The drift detection algorithm fits a Holt Double-Exponential Smoothing model for the subgroup means in forward time order and fits another Holt Double-Exponential Smoothing model for the subgroup means in reverse time order. The two smoothing model fits each have two smoothing constants: $\alpha$ for the level and $\beta$ for the slope. The $\beta$ smoothing constant is set at 0.05 and the $\alpha$ smoothing constant is estimated to minimize the error.

– The means of the positive drift values, negative drift values, and absolute drift values are reported as Mean Up Drift, Mean Down Drift, and Mean Abs Drift, respectively.

**Mean Up Drift**   The sum of the positive drift values divided by the count.

**Mean Down Drift**   The sum of the negative drift values divided by the count.

**Mean Abs Drift**   The sum of the absolute value of all drift values divided by the count.

**Note:** Process Screening does not utilize the Distribution or Process Capability Distribution column properties and always performs capability analyses assuming a normal distribution. If the distribution type for one of these column properties is non-normal, a Warnings outline is shown below the Summary table.

# Process Screening Platform Options

The Process Screening red triangle menu contains options to customize the display and to save calculated statistics. The options that operate on selected items are also accessible by right-clicking the **Column** column in the Summary table.

**Summary**   Shows or hides the Summary table. See "The Process Screening Report".

**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.

**Select Where**   Opens the Select Where window. You can select specific processes in the Summary table that correspond to a particular condition by using the Comparison menu and Value text box. For example, you can select all processes such that Ppk < 1.33. After you click OK, the processes are selected in the Summary table.

**Tip:** You can also access the Select Where window by right-clicking anywhere in the Summary table.

**Show Summary Columns**   Enables you to choose some of the summary columns that are shown 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. Each quick graph also has lines for the spec limits. There is a solid blue line for the target value and dotted blue lines for the LSL and USL values. To remove the quick graphs, click the Graphs of Selected Items red triangle and select Remove.

**Tip:** To change the layout of the quick graphs, click the Graphs of Selected Items red triangle and select Number of Plots Across.

**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.

**Notes:**

– Only the tests selected in Process Screening are sent to Control Chart Builder. Tests that have been turned off are not sent.

– If you have multiple subgroup variables where the inner subgroup variable is not sorted within the outer subgroup variable, a new subgroup variable column is created and used when Control Charts for Selected Items is launched. This does not apply if Sort by Subgroup is selected 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

**Notes:**

– If you specify sigma using a sigma control limit or the Derived Sigma option in the Limits Specifications window, this value is not used in Process Capability. This is because the Process Screening and Process Capability platforms use these control limits differently.

– If you have multiple subgroup variables where the inner subgroup variable is not sorted within the outer subgroup variable, a new subgroup variable column is created and used when Process Capability for Selected Items is launched. This does not apply if Sort by Subgroup is selected in the Process Screening launch window.

**Color Selected Items**   Applies a color of your choosing to the values in the selected rows of the Summary table.

**Remove Selected Items**   Removes the rows selected in the Summary table and reruns the analysis without those processes.

**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 Alt 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".

**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".

**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".

**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). You can also change the Shift Threshold by using the slider bar on the Shift Graph. Hover over the diamond to see the value. 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".

**Show Shifts in Quick Graphs**  (Available only when a Quick Graph has been added to the report window.) Shows the location of the shifts in the Quick Graphs using green and red vertical lines.

**Drift Summaries**  Adds columns for Mean Up Drift, Mean Down Drift, and Mean Abs Drift to the Summary Table. See "Drift Magnitudes and Positions".

**Drift Graph Selected**  Displays a drift graph for each process that you select in the Summary Table. Drift graphs enable you to detect smaller, more gradual changes in the selected processes. The values plotted are the slope estimates from a Holt Double-Exponential Smoothing model. See "Drift Magnitudes and Positions". Each drift graph displays the within-sigma value and the estimates for the two smoothing parameters in the Holt Double-Exponential Smoothing model.

**Process Performance Graph**    (Available only when specification limits are defined for at least one process variable.) Shows a four-quadrant graph that assesses the performance of processes in terms of stability and capability. See "Process Performance Graph".

**Process Performance Graph Boundaries**    (Available only once the Process Performance Graph is in the report window.) Opens a window where you can set values for the Process Performance Graph's stability index and Ppk capability boundaries.

**Tip:** You can set preferences for your desired boundaries for Stability Index and Ppk Capability in File > Preferences > Platforms > Process Screening.

**Goal Plot**    (Available only when specification limits are defined for at least one process variable.) The Goal Plot shows, for each variable, the spec-normalized mean shift on the X axis, and the spec-normalized standard deviation on the Y axis. If you define importance values for the processes, the goal plot points are sized by importance. It is useful for getting a quick, summary view of how the variables are conforming to specification limits. See *Quality and Process Methods* for more information about goal plots and their features.

**Tip:** Hover over a point in the Goal Plot to view the Quick Graph for that process. Click the Quick Graph to add it to the report window.

The Goal Plot red triangle menu contains the following options:

**Show Within Sigma Points**    Shows or hides the points calculated using the within sigma estimate.

**Show Within or Between-and-Within Sigma Points**    (Available only when three way control chart is selected in the launch window.) Shows or hides the points calculated using the between-and-within sigma estimate.

**Show Overall Sigma Points**    Shows or hides the points calculated using the overall sigma estimate.

**Shade Levels**    Shows or hides the Ppk level shading. This is turned on by default. The shaded areas depend on the relationship between $p$ and Ppk, with $p$ representing the value shown in the box beneath Ppk. By default, $p = 1.33$.

– Points in the red area have Ppk $< p$.
– Points in the yellow area have $p <$ Ppk $< 2p$.
– Points in the green area have $2p <$ Ppk.

**Label Within Sigma Points**    Shows or hides labels for points calculated using the within sigma estimate.

**Label Within or Between-and-Within Sigma Points**   (Available only when a three way control chart is selected in the launch window.) Shows or hides labels for points calculated using the between-and-within sigma estimate.

**Label Overall Sigma Points**   Shows or hides labels for points calculated using the overall sigma estimate.

**Defect Rate Contour**   Shows or hides a contour representing a specified defect rate.

**Show Capability**   (Available only when specification limits are defined for at least one process variable.) Displays a submenu that enables you to show or hide the Capabilities in the Summary table.

---

**Note:** Spec Centered Mean and Spec Scaled Std Dev are displayed in the Summary table as (Mean-Tgt)/SpecRange and StdDev/SpecRange. The Spec Limits option displays the LSL, USL, and Target columns in the Summary table.

---

The following option is also available in the submenu:

**Color Out of Spec Values**   Colors the cells in the data table that correspond to values that are out of spec. The cell is colored blue if the value is above the USL and red if the value is below the LSL.

---

**Tip:** To remove colors in specific cells, select all cells of interest. Right-click in one of the cells and select Clear Color.

---

**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. As in the summary table, the columns for Stability Index, Ppk, Cpk, Cp, and Target Index are colored as green, yellow, and red to indicate adequate, marginal, and poor stability or capability, respectively.

**Save Summary Table with Graphs**   Creates the Process Summary data table with an additional column titled Graph. The Graph column contains a quick graph for each process in the Process Summary table.

**Save Details Table**   Saves detailed information about control chart calculations to a Process Details data table. For each combination of process and grouping variables, the table contains a row for each subgroup showing:

–   The value of the subgroup variable

–   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.

– The Drift

**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 *Using JMP* 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.

## Process Performance Graph

The Process Performance Graph is a four-quadrant plot of capability versus stability. Each process for which specification limits are provided is represented by a marker. The marker type depends on if the process violates guidelines for the Target Index and Cp values. A left-pointing triangle indicates the process is below target, a circle indicates the process is on target, and a right-pointing triangle indicates the process is above target. An open marker indicates that the variation of the process is adequate and a filled marker indicates that the variation of the process could be reduced. A complete description of the marker combinations is provided in Table 23.1. Additionally, if you define importance values for the processes, the markers on the Process Performance Graph are sized by importance.

**Table 23.1**  Process Performance Graph Markers

| Graph Marker | Target Index | Cp |
| --- | --- | --- |
| ○ | On Target | Adequate Variation |
| ● | On Target | Reduce Variation |
| ▷ | Above Target | Adequate Variation |
| ▶ | Above Target | Reduce Variation |
| ◁ | Below Target | Adequate Variation |
| ◀ | Below Target | Reduce Variation |

On the graph, the horizontal coordinates represent the stability index of the process and the vertical coordinates represent the capability of the process, given as Ppk. The graph is divided into four quadrants based on the following default boundaries:

– A stability index that exceeds 1.25 indicates that the process is unstable.

– A Ppk that is smaller than 1.33 indicates that the process is not capable.

Additionally, there is a red line on the graph at 1.33 divided by Stability Index that indicates where the Cpk value is 1.33. This line categorizes the processes in the unstable and incapable quadrants in terms of how they might be fixed. Processes below the red line are fixable by special causes; processes above the red line are fixable by reducing variability. Selecting points in the graph selects the corresponding processes in the Summary table.

**Tip:** Hover over a point in the Process Performance Graph to view the Quick Graph for that process. Click the Quick Graph to add it to the report window.

# Additional Example of Process Screening

This 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".

    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". 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 shows shifts of only 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 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 23.7** Shift Graph
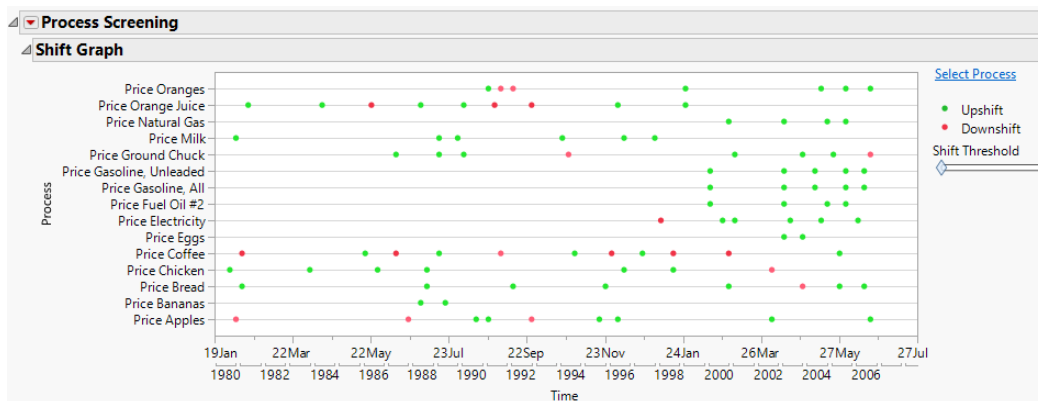


Most series show primarily upshifts. Price Coffee, however, has several alternating downshifts and upshifts. To better understand this series, obtain a control chart.

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. Right-click the selected process in the Summary table and select **Control Charts for Selected Items**.

**Figure 23.8**  Control Chart for Coffee
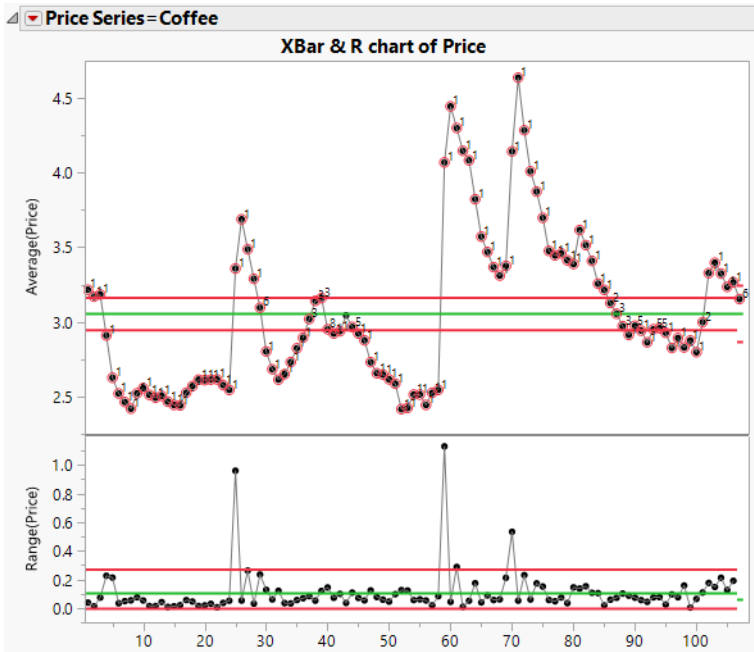


The control chart shows the upshifts and downshifts that are identified in the Shift Graph.

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 23.8, 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 for the Process Screening Platform

## 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 $d2\_Median*\sigma$, where $d2\_Median$ is the value corresponding to $n$.

- The theoretical median of the standard deviations is approximately $c4\_Median*\sigma$, where $c4\_Median$ is the value corresponding to $n$.

**Table 23.2** 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 |

**Table 23.2** Scaling Constants for Median Range and Median Standard Deviation  *(Continued)*

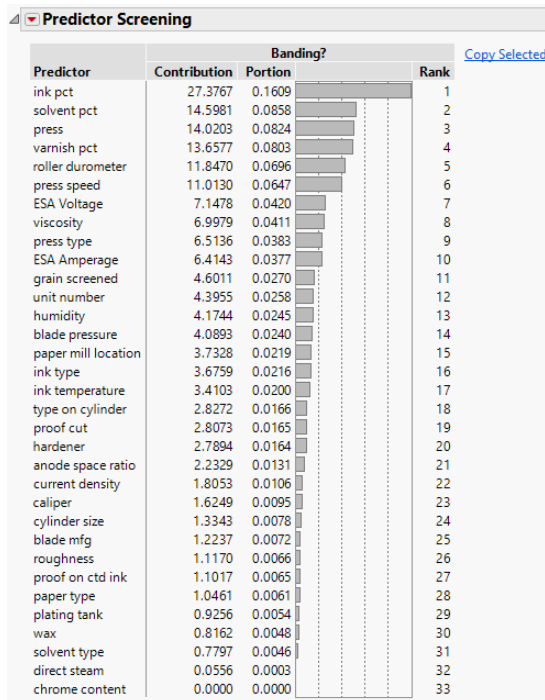| n  | d2_Median | c4_Median |
|----|-----------|-----------|
| 23 | 3.811     | 0.984     |
| 24 | 3.846     | 0.985     |
| 25 | 3.883     | 0.986     |

# 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".

**Figure 24.1** Example of a Predictor Screening Report



| Predictor | Contribution | Portion | Banding? | Rank |
|---|---|---|---|---|
| ink pct | 27.3767 | 0.1609 | | 1 |
| solvent pct | 14.5981 | 0.0858 | | 2 |
| press | 14.0203 | 0.0824 | | 3 |
| varnish pct | 13.6577 | 0.0803 | | 4 |
| roller durometer | 11.8470 | 0.0696 | | 5 |
| press speed | 11.0130 | 0.0647 | | 6 |
| ESA Voltage | 7.1478 | 0.0420 | | 7 |
| viscosity | 6.9979 | 0.0411 | | 8 |
| press type | 6.5136 | 0.0383 | | 9 |
| ESA Amperage | 6.4143 | 0.0377 | | 10 |
| grain screened | 4.6011 | 0.0270 | | 11 |
| unit number | 4.3955 | 0.0258 | | 12 |
| humidity | 4.1744 | 0.0245 | | 13 |
| blade pressure | 4.0893 | 0.0240 | | 14 |
| paper mill location | 3.7328 | 0.0219 | | 15 |
| ink type | 3.6759 | 0.0216 | | 16 |
| ink temperature | 3.4103 | 0.0200 | | 17 |
| type on cylinder | 2.8272 | 0.0166 | | 18 |
| proof cut | 2.8073 | 0.0165 | | 19 |
| hardener | 2.7894 | 0.0164 | | 20 |
| anode space ratio | 2.2329 | 0.0131 | | 21 |
| current density | 1.8053 | 0.0106 | | 22 |
| caliper | 1.6249 | 0.0095 | | 23 |
| cylinder size | 1.3343 | 0.0078 | | 24 |
| blade mfg | 1.2237 | 0.0072 | | 25 |
| roughness | 1.1170 | 0.0066 | | 26 |
| proof on ctd ink | 1.1017 | 0.0065 | | 27 |
| paper type | 1.0461 | 0.0061 | | 28 |
| plating tank | 0.9256 | 0.0054 | | 29 |
| wax | 0.8162 | 0.0048 | | 30 |
| solvent type | 0.7797 | 0.0046 | | 31 |
| direct steam | 0.0556 | 0.0003 | | 32 |
| chrome content | 0.0000 | 0.0000 | | 33 |

## Contents

# Overview of the Predictor Screening Platform

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.
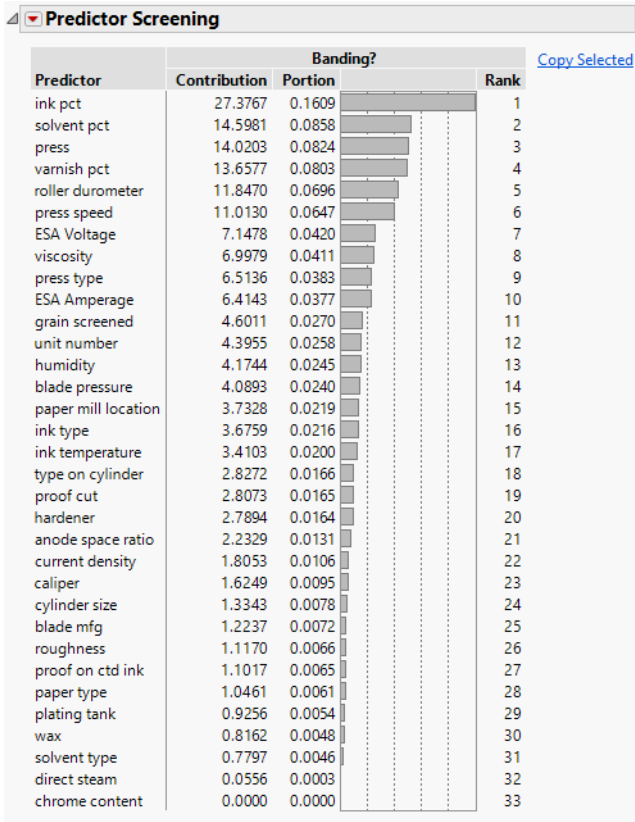
The predictor screening platform uses a bootstrap forest to identify potential predictors of your response. For each response, a bootstrap forest model using 100 decision trees is built. The column contributions to the bootstrap forest model for each predictor are ranked. Because the bootstrap forest method involves a random component, column contributions can differ when you rerun the report. For more information about decision trees, see "Partition Models".

# 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 24.2** Ranked Column Contributions



| Predictor | Contribution | Portion | Banding? | Rank | Copy Selected |
|---|---|---|---|---|---|
| ink pct | 27.3767 | 0.1609 | | 1 | |
| solvent pct | 14.5981 | 0.0858 | | 2 | |
| press | 14.0203 | 0.0824 | | 3 | |
| varnish pct | 13.6577 | 0.0803 | | 4 | |
| roller durometer | 11.8470 | 0.0696 | | 5 | |
| press speed | 11.0130 | 0.0647 | | 6 | |
| ESA Voltage | 7.1478 | 0.0420 | | 7 | |
| viscosity | 6.9979 | 0.0411 | | 8 | |
| press type | 6.5136 | 0.0383 | | 9 | |
| ESA Amperage | 6.4143 | 0.0377 | | 10 | |
| grain screened | 4.6011 | 0.0270 | | 11 | |
| unit number | 4.3955 | 0.0258 | | 12 | |
| humidity | 4.1744 | 0.0245 | | 13 | |
| blade pressure | 4.0893 | 0.0240 | | 14 | |
| paper mill location | 3.7328 | 0.0219 | | 15 | |
| ink type | 3.6759 | 0.0216 | | 16 | |
| ink temperature | 3.4103 | 0.0200 | | 17 | |
| type on cylinder | 2.8272 | 0.0166 | | 18 | |
| proof cut | 2.8073 | 0.0165 | | 19 | |
| hardener | 2.7894 | 0.0164 | | 20 | |
| anode space ratio | 2.2329 | 0.0131 | | 21 | |
| current density | 1.8053 | 0.0106 | | 22 | |
| caliper | 1.6249 | 0.0095 | | 23 | |
| cylinder size | 1.3343 | 0.0078 | | 24 | |
| blade mfg | 1.2237 | 0.0072 | | 25 | |
| roughness | 1.1170 | 0.0066 | | 26 | |
| proof on ctd ink | 1.1017 | 0.0065 | | 27 | |
| paper type | 1.0461 | 0.0061 | | 28 | |
| plating tank | 0.9256 | 0.0054 | | 29 | |
| wax | 0.8162 | 0.0048 | | 30 | |
| solvent type | 0.7797 | 0.0046 | | 31 | |
| direct steam | 0.0556 | 0.0003 | | 32 | |
| chrome content | 0.0000 | 0.0000 | | 33 | |

**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 24.2. See "Bootstrap Forest".

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.

**Tip:** Click on a column heading to change how the columns are sorted in the table.

# Launch the Predictor Screening Platform

Launch the Predictor Screening platform by selecting **Analyze > Screening > Predictor Screening**.

**Figure 24.3** Predictor Screening Launch Window
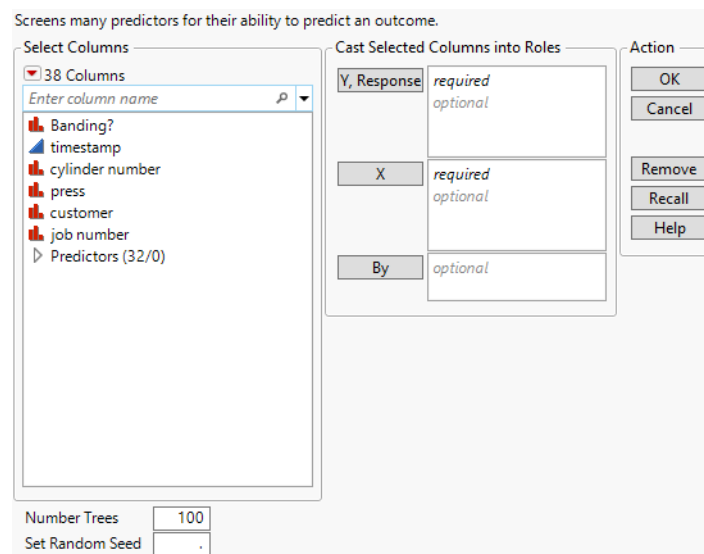


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**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.

**Number of Trees**   The number of decision trees in the bootstrap forest model.

**Set Random Seed**   Sets the seed for the starting values used in the fitting procedures. Set Random Seed is useful if you want to reproduce an analysis. If you set a random seed and save the script, the seed is automatically saved in the script.

# The Predictor Screening Report

The report (Figure 24.2) 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. The Copy Selected button enables you to copy the selected columns to the clipboard so that you can paste them into a model as well. In this way, the Predictor Screening enhances the modeling process.

# Predictor Screening Platform Options

See *Using JMP* 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.
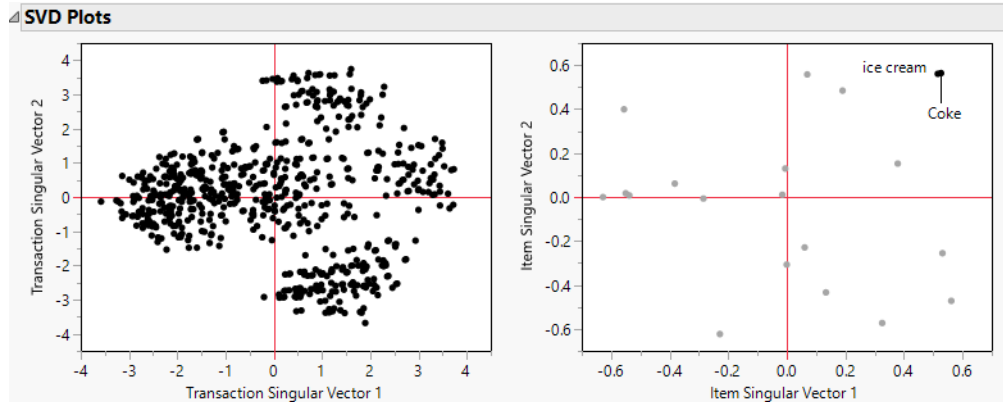
JMP
PRO *The Association Analysis platform is available only in JMP Pro.*

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 25.1** Example of Singular Value Decomposition Plots

# Contents

# Overview of the Association Analysis Platform

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. Note that confidence in association analysis is not related to the concept of confidence intervals.

- *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".

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**.

   By default, the Rules report is sorted in decreasing order by Confidence. However, association rules with extremely high confidence also tend to have a higher number of items in the condition set. Since you want to view association rules with smaller condition sets, sort the report by Confidence, but in increasing order.

6. Right-click in the Rules report and select **Sort By Column**.

   The Select Columns window appears.

7. Select Confidence and then check the Ascending option.
8. Click **OK**.

**Figure 25.2**  Association Analysis Report

| Rule | | Confidence | Lift |
|---|---|---|---|
| **Condition** | **Consequent** | | |
| apples | sardines | 40% | 1.357 |
| Coke | turkey | 40% | 1.422 |
| sardines | chicken, Heineken | 40% | 2.002 |
| herring, olives | ham, turkey | 40% | 3.006 |
| ice cream | turkey | 40% | 1.424 |
| artichoke | baguette, Heineken | 40% | 1.547 |
| ham | corned beef, herring | 40% | 1.648 |
| ham | corned beef, olives | 40% | 1.703 |
| peppers | sardines | 41% | 1.371 |
| sardines | peppers | 41% | 1.371 |
| Coke | Heineken, sardines | 41% | 2.23 |
| sardines | Coke, Heineken | 41% | 2.332 |
| avocado, Heineken | crackers, ham | 41% | 2.707 |
| soda | baguette, herring | 41% | 1.631 |
| corned beef | ham | 41% | 1.343 |
| Heineken, soda | bourbon, crackers, olives | 41% | 3.152 |
| artichoke, Heineken | avocado, baguette | 41% | 1.921 |
| Heineken, herring | avocado, baguette | 41% | 1.924 |

9. Scroll down the report to where the Confidence values are 58%.

There is an entry in the Rules report table that 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.

10. Click the disclosure icon next to Frequent Item Sets.

**Figure 25.3** Frequent Item Sets Report



The Frequent Item Sets report shows that 36% of customers purchased avocados. The Rules report in Figure 25.2 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.

11. Right-click in the Rules report table and select **Sort By Column**.

The Select Columns window appears.

12. 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 5.642 and 83% confidence. You want to verify that both the condition set, {chicken, ice cream}, and the consequent item set, {Coke, sardines}, have adequate support.

13. Right-click in the Frequent Item Sets report and select **Sort By Column**.

The Select Columns window appears.

14. Select Item Set and then check the Ascending option.

15. Click **OK**.

The Frequent Item Sets table is sorted alphabetically by item set. Scroll through the list to see that the condition item set, {chicken, ice cream}, has 14% support. The consequent item set, {Coke, sardines}, has 15% 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 25.4** Association Analysis Launch Window
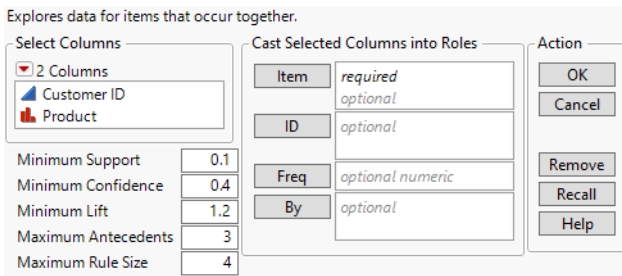


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

**Item** The categorical column or columns that contain the item data to be analyzed. If the column has the Multiple Response modeling type, each of the multiple responses in each row is treated as an item. If multiple columns are specified, each column in each row is treated as an item.

**ID** The column that identifies the transaction to which an item belongs. The ID role is required, unless the Item role is filled by a column with the Multiple Response modeling type or the Item role is filled by multiple columns.

**Note:** If an ID role is specified, all rows with the same ID are combined into one transaction.

**Freq** The column that identifies a frequency for the transactions in the Item column. For example, if a transaction has a frequency of 3, that is treated as three identical transactions that contain the items in the Item column.

**Note:** The Freq role is ignored if the Item role is filled by a column that does not have the Multiple Response modeling type.

**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. The default value is 4.

---

**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".

---

## Data Format

There are three data formats that are accepted in the Association Analysis platform:

- You can specify a single item response in each row in the Item role and identify which items are included in each transaction using the ID role. In this format, the Item and ID roles are required and the Freq role is ignored.

- You can specify a Multiple Response modeling type column for the Item role. In this format, the Item role is required, the ID and Freq roles are optional. If an ID role is specified, all rows with the same ID are combined into one transaction. If a Freq role is specified, the frequency applies to the transaction, not the specific items in the transaction.

- You can specify multiple item response columns in the Item role. In this format, the ID and Freq roles are optional. If an ID role is specified, all rows with the same ID are combined into one transaction. If a Freq role is specified, the frequency applies to the transaction, not the specific items in the transaction.

# The Association Analysis Report

By default, the Association Analysis report contains the following reports:

- "Frequent Item Sets"
- "Rules"

**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 decreasing order of confidences. 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.

# JMP PRO **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".

**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".

**SVD**   Shows or hides a report of a partial singular value decomposition (SVD) of the incidence matrix for the items. This decomposition reduces the incidence matrix into a user-specified number of dimensions for analysis. The SVD report also contains options to visualize the singular vectors, cluster transactions and items, save out singular vectors, and perform a rotated SVD analysis. See "Singular Value Decomposition".

See *Using JMP* 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.

# Singular Value Decomposition

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 SVD analysis is equivalent to performing principal components analysis (PCA) on a correlation matrix.

The *transaction item matrix* is a matrix for which each row corresponds to a transaction and 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*.

The partial singular value decomposition approximates the column-standardized 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 *nTran* as the number of transactions (rows) in the transaction item matrix, *nItem* 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(*nTran*, *nItem*). It follows that **U** is an *nTran* by *nVec* matrix that contains the left singular vectors of the transaction item matrix. **S** is a diagonal matrix of dimension *nVec*. The diagonal entries in **S** are the singular values in the transaction item matrix. **V′** is an *nVec* by *nItem* matrix. The rows in **V′** (or columns in **V**) are the right singular vectors.

The right 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 *nTran* 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. The SVD implementation takes advantage of the sparsity of the transaction item matrix.

## SVD Report

The SVD option produces two SVD plots and a table of the singular values from the singular value decomposition.

## SVD Plots

The first plot contains a point for each transaction. For a given transaction, the point that is plotted is defined by the transaction's values in the first two singular vectors (the first two columns of the **U** matrix) multiplied by the diagonal singular values matrix (**S**). This plot is equivalent to the Score Plot in the Principal Components platform. Each point in this plot represents a transaction. In the Transaction SVD plot, points that are visibly grouped together indicate transactions with a similar composition.

The second plot contains a point for each item. For a given item, the point that is plotted is defined by the item's values in the first two singular vectors (the first two rows of the **V**′ matrix) multiplied by the diagonal singular values matrix (**S**). This plot is equivalent to the Loadings Plot in the Principal Components platform. Each point in this plot represents an item. In the Item SVD plot, items that are visibly grouped together indicate items that have similar functions or topic areas.

## 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 Singular Values table also contains a column of corresponding eigenvalues for the equivalent principal components analysis. Like in the Principal Components platform, there are columns for the percent and cumulative percent of variation explained by each eigenvalue (or singular value). 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.

## <sub>JMP PRO</sub> SVD Report Options

The SVD red triangle menu contains the following options:

**SVD Scatterplot Matrix**   Shows or hides a scatterplot matrix of the item and transaction singular value decomposition vectors. You must select the size of the scatterplot matrix when you select this option. This scatterplot matrix enables you to visualize more than the first two dimensions of the singular value decomposition.

**Topic Analysis, Rotated SVD**   Performs a varimax rotated partial singular value decomposition of the item transaction matrix to produce groups of items called topics. You can select this option multiple times to find different numbers of topics. See "Topic Analysis".

**Cluster Items**   Shows or hides a hierarchical clustering analysis of the items in the data. The following options are available to the right of the dendrogram:

> **Set Clusters**   Specifies the number of clusters that are colored in the dendrogram and used in the Save Clusters option. Points in the Item SVD plot are also colored by cluster.

> **Save Clusters**   Saves a data table that contains the items and their cluster assignments. You must set the number of clusters before clicking Save Clusters.

> For more information about clustering and dendrograms, see *Multivariate Methods*.

**Cluster Transactions**   Shows or hides a hierarchical clustering analysis of the transactions in the data. The following options are available to the right of the dendrogram:

> **Set Clusters**   Specifies the number of clusters that are colored in the dendrogram and used in the Save Clusters option. Points in the Transaction SVD plot are also colored by cluster.

> **Save Clusters**   Saves a data table that contains the transactions and their cluster assignments. You must set the number of clusters before clicking Save Clusters.

> For more information about clustering and dendrograms, see *Multivariate Methods*.

**Save Transaction Singular Vectors**   Saves a user-specified number of singular vectors from the transaction singular value decomposition as columns to a new data table where each row corresponds to a transaction. The first two saved columns represent the points plotted in the transaction SVD plot. See "Singular Value Decomposition".

**Save Item Singular Vectors**   Saves a user-specified number of singular vectors from the items singular value decomposition as columns to a new data table where each row corresponds to an item. The first two saved columns represent the points plotted in the item SVD plot. See "Singular Value Decomposition".

# JMP® PRO  **Topic Analysis**

The Topic Analysis, Rotated SVD option performs a varimax rotation on the partial singular value decomposition (SVD) of the transaction item matrix. See "Singular Value Decomposition". You must specify a number of rotated singular vectors, which corresponds to the number topics that you want to retain from the transaction item matrix. After you specify a number of topics, the Topic Analysis report appears.

Topic analysis is equivalent to a rotated principal component analysis (rotated PCA). The varimax rotation takes a set of singular vectors and rotates them to make them point more directly in the coordinate directions (toward the items). This rotation makes the vectors help explain the composition of the transactions as each rotated vector orients toward a set of items. Negative values indicate a repulsion force. The items with negative values occur in a topic less frequently compared to the items with positive values.

# JMP® PRO  **Topic Analysis Report**

The Topic Analysis 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 Top Loadings by Topic report shows a table 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 Analysis report also contains the following reports:

**Topic Loadings**   Contains a matrix of the loadings across topics for each item. This matrix is equivalent to the factor loading matrix in a rotated PCA.

**Word Clouds by Topic**   Contains a matrix of word clouds, one for each topic.

**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.

**Topic Scores Plots**   Shows the topic scores for all transactions in one-dimensional scatterplots. Negative values indicate transactions that are negatively associated with a topic. Use these plots to explore the distribution of transactions within each topic.

**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.

**Rotation Matrix**   Contains the rotation matrix for the varimax rotation.

## Topic Analysis Report Options

The Topic Analysis red triangle menu contains the following options:

**Topic Scatterplot Matrix**   Shows or hides a scatterplot matrix of the rotated singular value decomposition vectors.

**Display Options**   Contains options to show or hide content that appears in the Topic Analysis report. See "Topic Analysis Report".

**Rename Topics**   Shows a dialog that enables you to add descriptive names for one or more of the topics.

**Save Transaction Topic Vectors**   Saves a user-specified number of singular vectors from the rotated singular value decomposition as columns to the data table.

**Save Item Topic Vectors**   Saves the topic vectors as columns to a new Item Topic Scores data table.

**Remove**   Removes the Topic Analysis report from the SVD report.

---

# 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 Association Analysis red triangle and select **SVD**.
7. Click **OK**.

**Figure 25.5** SVD Plots
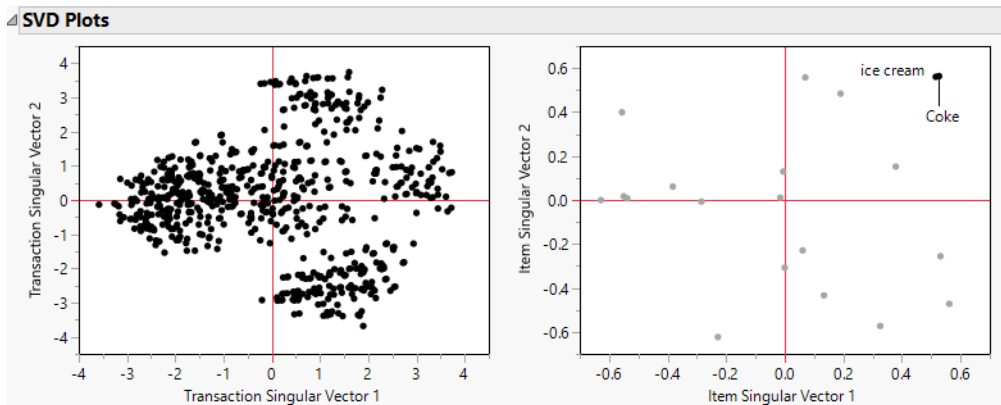


The transaction SVD plot suggests that there might be two or three groups of transactions. In the upper right corner of the item SVD plot, notice that the points that represent Coke and ice cream overlap. The proximity of these two items indicates that there is a strong affinity between them.

8. Click the SVD red triangle and select **Topic Analysis, Rotated SVD**.

9. Type 3 next to Number of topics (rotated singular vectors) and click **OK**.

   The Topic Items and Topic Scores reports appear.

**Figure 25.6** Topic Items Report



| 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 do not contain avocados, but do contain olives.

**Figure 25.7**  Topic Scores
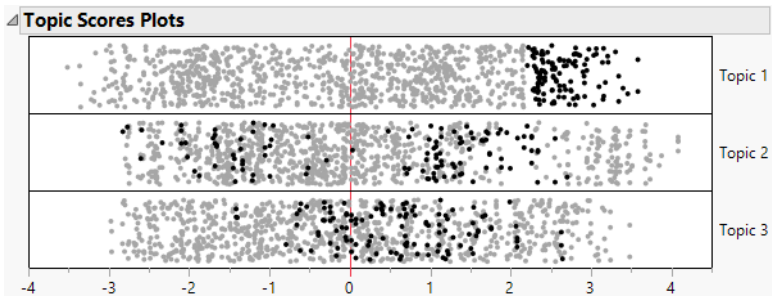


The topic scores that are assigned to each of the 1001 transactions are plotted in the Topic Scores report. Select groups of points for a topic to see how those transactions relate to other topics. For example, transactions with very high values on Topic 1 tend to have low values on Topics 2 and 3.

10. Open the Singular Values report.

**Figure 25.8**  Singular Values Table

| 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 25.8, 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

- "Frequent Item Set Generation"
- "Association Analysis Performance Measures"

## 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 the Association Analysis platform. Denote an association rule with condition item set $X$ and consequent item set $Y$ by $X \Rightarrow Y$. Hahsler (2015) contains a collection of performance measures used in association analysis, including support, confidence, and lift.

## Support

Support is the proportion of transactions in which an item set occurs. Support can also be viewed as the probability that a transaction contains an item set.

The support $S$ of a condition item set $X$ is defined as follows:

$$supp(X) = \frac{N_X}{N} = Pr(X)$$

where:

$N_X$ is the number of transactions that contain the item set $X$

$N$ is the total number of transactions.

Support for an association rule is defined as follows:

$$supp(X \Rightarrow Y) = supp(X \cup Y) = Pr(X \cap Y)$$

In this case, support is equivalent to the probability that a transaction contains both item sets $X$ and $Y$.

For both the item set and association rule cases, support ranges from 0% to 100%.

## Confidence

Confidence is the proportion of transactions that contain the consequent item set, given that the transaction contains the condition item set. Confidence can also be viewed as the conditional probability that a transaction contains the consequent item set, given that the transaction contains the condition item set.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} = \frac{Pr(X \cap Y)}{Pr(X)} = Pr(Y|X)$$

Confidence ranges from 0% to 100%. 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.

**Note:** Confidence in association analysis is not related to the concept of confidence intervals.

## Lift

Lift measures dependency between $X$ and $Y$.

$$lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)} = \frac{Pr(X \cap Y)}{Pr(X)Pr(Y)}$$

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.

Lift ranges from 0 to $\infty$. 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.

**Note:** The lift for association rule $X \Rightarrow Y$ is equal to the lift for association rule $Y \Rightarrow X$.

# Process History Explorer

## Identify Problem Components in Complex Process Histories

**Note:** The Process History Explorer platform is new and experimental. It is subject to change in future releases.

In complex manufacturing situations, such as in semiconductor factories, a defective tool or route combination might lead to poor yield or reduced quality. However, there might be hundreds of steps that a unit passes through that involve many different tools. It can be difficult to determine the exact combination of factors that leads to the poor yield.

The Process History Explorer platform organizes all of the production tracking data and enables you to identity factors that seem to be associated with poor yield. You can then perform stepwise regression. You can analyze the time that a unit spends waiting between steps or tools in the process and see how that waiting time is related to yield. You can also analyze the effect on yield of transitions from one tool to another.

**Figure 26.1** Process History Explorer Report

## Contents

# Example of Process History Explorer

This example uses two data tables. The first data table contains step-by-step history of 25 wafers in 50 lots that went through a complex manufacturing process. The second data table contains the yield for each wafer in each lot.

1. Select **Help > Sample Data Library** and open Quality Control > Lot Wafer History.jmp and Quality Control > Lot Wafer Yield.jmp.

2. From the Lot Wafer History.jmp data table, select **Analyze > Screening > Process History Explorer**.

3. Select Lot and Wafer and click **ID**.

4. Select Tool and Route and click **X, Process**.

5. Select Layer and Operation and click **Step**.

6. Select TimeIn and TimeOut and click **Timestamp**.

7. Click **OK**.

   A window appears that enables you to select the corresponding yield table.

8. Select Lot Wafer Yield and click **OK**.

   A window appears that enables you to select the yield column.

9. Select Yield and click **OK**.

10. Click the Process History Explorer red triangle and select **Levels with Lowest Yield**.

**Figure 26.2** Process History Explorer Report

| Term | Level | Count | Mean ^ | Std Error | Std Dev |
|------|-------|-------|--------|-----------|---------|
| Tool | PlanarTool008 | 1000 | 0.75299 | 0.00188 | 0.05952 |
| Tool | DepTool003 | 1375 | 0.75577 | 0.00168 | 0.06245 |
| Tool | EtchTool019 | 750 | 0.76192 | 0.0021 | 0.05749 |
| Tool | EtchTool012 | 500 | 0.76203 | 0.0026 | 0.05807 |
| Tool | EtchTool034 | 850 | 0.76279 | 0.00199 | 0.0581 |
| Tool | PlanarTool021 | 1050 | 0.7637 | 0.00186 | 0.06028 |
| Tool | PlanarTool030 | 1025 | 0.76539 | 0.00184 | 0.05894 |
| Tool | EtchTool016 | 900 | 0.76566 | 0.00191 | 0.05734 |
| Tool | EtchTool009 | 925 | 0.76598 | 0.00197 | 0.05994 |
| Tool | Dep2Tool018 | 1525 | 0.76766 | 0.00156 | 0.06088 |
| Tool | Dep2Tool014 | 1825 | 0.76813 | 0.00138 | 0.05893 |
| Tool | DepTool019 | 1650 | 0.76839 | 0.00143 | 0.05794 |
| Tool | EtchTool020 | 1000 | 0.76909 | 0.00177 | 0.05586 |
| Tool | Dep2Tool015 | 1475 | 0.76948 | 0.0015 | 0.05775 |
| Tool | PhotoTool010 | 3525 | 0.76996 | 0.00098 | 0.05822 |
| Tool | Dep2Tool013 | 1825 | 0.7705 | 0.00141 | 0.06025 |
| Tool | DepTool012 | 1750 | 0.77063 | 0.00145 | 0.06051 |
| Tool | Dep2Tool006 | 1400 | 0.77063 | 0.00158 | 0.05905 |
| Tool | PlanarTool028 | 900 | 0.77103 | 0.00202 | 0.06071 |
| Tool | Dep2Tool011 | 1600 | 0.77124 | 0.00145 | 0.05797 |
| Tool | EtchTool010 | 850 | 0.77125 | 0.00213 | 0.06208 |

**Process History Explorer**
1250 Units, using ID columns: Lot, Wafer
125 Steps, using columns: Layer, Operation
156250 Operations, using time columns: TimeIn, TimeOut
195 levels in X columns: Tool, Route
1 response: Yield
**Response Yield**
**Levels with Lowest Yields**

The report contains information about the process, including the number of units, steps, operations, and levels in the X columns. The Levels with Lowest Yields table contains counts and yield information for the 195 levels in the X columns. Notice that the PlanarTool008 tool had the lowest yield and 1,000 units went through that tool.

# Launch the Process History Explorer Platform

Launch the Process History Explorer platform by selecting **Analyze > Screening > Process History Explorer**.

**Figure 26.3** Process History Explorer Launch
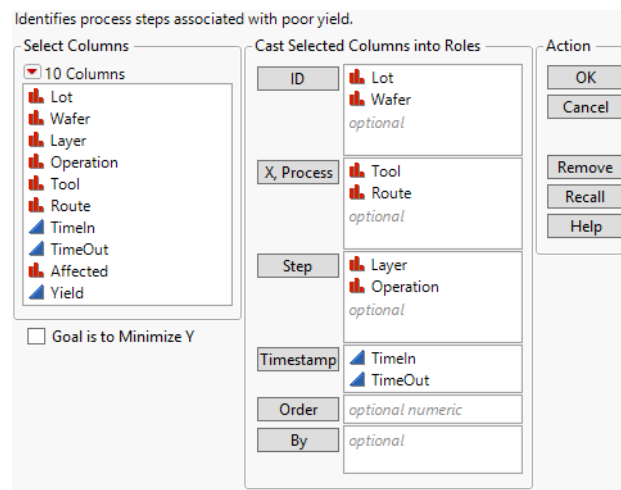


For more information about the options in the Select Columns red triangle menu, see *Using JMP*.

### Launch Window Roles

**ID**   Specifies one or more columns that identify a unit in the process.

**Note:** The ID columns must have corresponding columns in the yield table.

**X, Process**   Specifies one or more columns that identify the components or specific details that make up the steps of the process.

**Step**   Optional columns that specify the steps and sub-steps that a unit goes through. For example, the layer and the type of operation within the layer.

**Timestamp**   Specifies the time of the operation. The time can be specified as a single time point or as a pair of time points that indicate the times that the operation started and ended.

**Order**   Optional column that specifies the order that a batch of units went through the components specified in the X, Process role.

**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.

### Launch Window Options

**Goal is to Minimize Y**    Specifies that the goal of the analysis is to minimize the response. By default, the analysis seeks to maximize the response.

**Note:** The response is specified in a second data table, not in the data table used to launch the platform.

## Data Format

The data for Process History Explorer must be specified in two data tables. The first table is used to launch the platform and contains the steps and details of the process. The second table contains the response, or yield, for each level of the ID columns.

The sample data tables Quality Control > Lot Wafer History and Quality Control > Lot Wafer Yield provide an example of how the data are specified in two data tables.

## The Process History Explorer Report

The initial Process History Explorer Report provides a summary of the information in the data table(s).

- The number of Units, determined by the ID columns. If there is more than one ID column, this is the number of unique combinations.

- The number of Steps, determined by the Step columns. If there is more than one Step column, this is the number of unique combinations.

- The number of Operations, determined by the Timestamp columns. This is the same as the number of rows in your data table.

- The number of levels, determined by the X, Process columns. If there is more than one X, Process column, this is the sum of the number of levels in each column.

- The number of responses, determined by the Yield columns.

Select options from the Process History Explorer red triangle menu to continue your analysis. See "Process History Explorer Platform Options".

# Process History Explorer Platform Options

The Process History Explorer red triangle menu contains the following options:

**Levels with Lowest Yield**    Displays a table of summary statistics for the yield for each level of each X, Process variable. The summary statistics provided for each level are the number of operations, and the mean, standard error, and standard deviation of the yield. By default, the table is sorted in ascending order by mean yield.

**Levels with Lowest Yield with Time Filter**    Displays a table of summary statistics for the time intervals of the top 50 levels with the lowest yields. The routine searches for a time interval for each of the top 50 levels in the Levels with Lowest Yield table that minimizes the mean yield within the interval. The interval must contain at least 25% of the operations in that level. The summary statistics provided for each level are the number of operations in the identified interval, the mean yield in the identified interval, the timestamps for the first and last operations in the level (First Time and Last Time), and the timestamps for the first and last operations in the identified interval (RunStart Time and RunEnd Time).

**Stepwise Regression**    Displays a Stepwise Regression window that enables you to specify options for a stepwise regression analysis. You can use stepwise regression to find problems in the process, where a problem is something that causes your yield to decrease. At each step in the regression, a term enters according to which term has the largest negative yield. This helps you identify process steps that most impact the yield. Stepwise Regression provides a table that shows the order in which terms entered the model at each step of the regression.

**Note:** If the Goal is to Minimize Y option is selected in the launch window, Stepwise Regression adjusts to detect terms that cause the yield to increase.

**Goal**    Specifies the criterion used in the stepwise regression model. The Biggest individual difference option identifies the terms that individually cause the yield to change the most. The Biggest total difference option identifies the terms that cause the biggest total difference, which is the difference multiplied by the number of units affected. The Most predictive option produces a traditional stepwise regression model.

**Time Filtering**    Enables you to use the starting time of a component in the stepwise regression model. If you choose the None option, the times are averaged over the entire time period. If you choose the Starting Time option, the component is allowed to fail at a particular starting time. The starting time is the time that maximizes the negative yield affect. If you choose the Starting Time option, the Stepwise Regression report table contains a Start Time column.

**X Transform**　Enables you to use the regular count data or the Log(Count + 1) data for the X, Process variables in the stepwise regression. The Log(Count + 1) option transforms the variables in the regression such that high frequency counts have less of a linear influence in the regression. If you choose the None option, high frequency counts have a linear influence in the regression.

**Number of Regression Steps**　The number of steps in the Stepwise Regression analysis. The default is ten.

**Figure 26.4** Stepwise Regression Window



**Waiting Time Analysis**　(Available only when two Timestamp variables are specified.) Each unique ID can go through only one level of a step at a time. In addition, each step can process only one ID at a time. Waiting Time Analysis provides a table of waiting time summary statistics for each ID. In this table, waiting time is the amount of time between when an ID is done with one step and when it starts another step. Additional Waiting Time Analysis tables are also provided. These tables contain waiting time summary statistics for each ID within the levels of each Step variable. In these tables, waiting time is the amount of time between when an ID is done with a particular level of a Step variable and when it goes through that same level again. The tables provide the following statistics:

**Waiting Sum**　The sum of the waiting times for the given combination of ID and Step variable levels.

**Waiting Count**　The number of waiting times for the given combination of ID and Step variable levels.

**Waiting Mean**　The mean of the waiting times for the given combination of ID and Step variable levels.

**Waiting Max**   The maximum waiting time for the given combination of ID and Step
variable levels.

**Transition Analysis**   Displays a report that helps you identify transitions from one
component to another that are problematic for the resulting yield. This report can help you
identify alignment and registration issues in the process. For example, some levels of the
process column might be out of alignment with respect to other levels of the process
column. If there is a transition problem, you would expect that items that repeat a specific
level to have better yields than items that go through different levels of the process
column.

When you select this option, you must choose one or more X, Process columns and one or
more levels in each of those columns to include in the analysis. For each column that you
specify for the analysis, a report appears that contains four tables. The first shows yields
and counts for transitions from a specific level of the column to another specific level of the
column. The second shows yields and counts for transitions away from specific levels of
the column. The third shows yields and counts for transitions to specific levels of the
column. The fourth shows yields and counts for transitions where the unit repeats a
specific level of the column.

**Order Interaction**   (Available only if an Order variable is specified in the launch window.)
Displays a report of a one-way analysis of the yield on the Order column. The report
contains a table for each X, Process column. In each table, the levels of the process column
are listed in descending order of LogWorth of the corresponding analysis of variance
model. The tables also contain the $p$-values and a heat map for each level of the process
column.

**Save Count Table**   Creates a new data table that contains counts of how many units went
through each level of the X, Process variables. The count table has a row for each unique
unit, determined by the ID values.

**Save Log Count Table**   Creates a new data table that is the same format as the Count table,
except with entries of Log(Count + 1).

**Save Waiting Time**   (Available only when two Timestamp variables are specified.) Saves each
table in the Waiting Time Analysis report to a separate data table.

# Appendix **A**

## Statistical Details

### Predictive and Specialized Modeling

This appendix discusses measures of fit and the types of cross validation available in each platform.

# Contents

# Measures of Fit

## Entropy RSquare

The Entropy RSquare is computed for the training set and for the validation and test sets if validation is used. For the training set, Entropy RSquare is computed as follows:

- The specified model is fit using the training set.
- Predicted probabilities for each level of the response are obtained, based on the model.
- Using these predicted probabilities, the likelihood is computed for observations in the training set. Call this *Likelihood_Full$_{Training}$*.
- The reduced model (no predictors) is fit using the training set.
- The predicted probabilities for the levels of the response from the reduced model are used to compute the likelihood for observations in the training set. Call this quantity *Likelihood_Reduced$_{Training}$*.
- The Entropy RSquare for the training set is:

$$\text{Entropy RSquare}_{Training} = 1 - \frac{\log(Likelihood\_Full_{Training})}{\log(Likelihood\_Reduced_{Training})}$$

The Entropy RSquare for the validation and test sets are computed in a manner analogous to the Entropy RSquare for the training set.

# Validation in JMP Modeling

Data can be partitioned into sets before modeling to avoid overfitting and to select a good predictive model. This process uses part of the original data to estimate parameters and uses the rest of the data to tune or evaluate the parameters, or do both. In JMP Pro, you can partition the data into two or three sets in the following ways:

**Train and Evaluate**   Partitions the data into two sets, called Training and Validation. The training set is used to estimate the model parameters. The validation set is used to independently evaluate the performance of the fitted model.

**Train and Tune**   Partitions the data into two sets, called Training and Validation. The training set is used to estimate the model parameters. The validation set is used in the model fitting algorithm to tune the model parameters and ultimately choose a model with good predictive ability. There is no independent model evaluation done in this case.

**Train, Tune, and Evaluate**    Partitions the data into three sets, called Training, Validation, and Test. The training set is used to estimate the model parameters. The validation set is used in the model fitting algorithm to tune the model parameters and ultimately choose a model with good predictive ability. The test set is then used to independently evaluate the performance of the fitted model.

## Validation Column Role

The Validation Column role is available only in JMP Pro. For JMP, see "Excluded Rows as Validation Holdback".

One way to create data partitions is to use the Validation Column role. The Validation Column role uses the column's values to divide the data into parts. The column is assigned using the Validation role in the platform's launch window. For information about how to create a validation column, see "Make Validation Column".

**Caution:** The use of a validation column is platform specific. Different platforms use the levels of the validation column differently. See notes in Table A.1.

**Table A.1**   Validation Column by Platform

| Platform | Train & Evaluate | Train & Tune | Train, Tune, & Evaluate | Notes |
|---|---|---|---|---|
| *Fit Model* | | | | |
| Fit Least Squares | Yes | No | No | If there are more than three levels, the validation column is ignored. |
| Stepwise Regression | No | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |
| Logistic Regression | Yes | No | No | If there are more than three levels, the validation column is ignored. |
| Generalized Regression | No | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |
| Partial Least Squares | No | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |
| *Predictive Models* | | | | |
| Neural | No | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |

| Platform | Train & Evaluate | Train & Tune | Train, Tune, & Evaluate | Notes |
|---|---|---|---|---|
| Partition | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| Bootstrap Forest | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| Boosted Tree | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| K Nearest Neighbors | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| Naive Bayes | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| Support Vector Machines | Yes | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |
| *Specialized Models* | | | | |
| Functional Data Explorer | Yes | No | No | Must be created as a Grouped Random validation column.<br><br>If there are more than two levels, the smallest value defines the training set and all other values define the validation set. |
| *Multivariate Models* | | | | |
| Discriminant | Yes | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |
| Partial Least Squares | No | Yes | Yes | If there are more than three levels, K-Fold Cross-Validation is used. |

| Platform | Train & Evaluate | Train & Tune | Train, Tune, & Evaluate | Notes |
|---|---|---|---|---|
| Uplift | No | Yes | Yes | If there are more than three levels, the platform only uses rows with the three smallest values. |

# K-Fold and Holdback Validation

The Validation Column role provides a framework for partitioning data into cross validation sets. In addition, some JMP platforms also support K-Fold and various types of Holdback validation.

**K-Fold Cross-Validation**   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 that produces the best validation statistic is chosen as the final model, and the fold that is not used in the building of that model provides the test set performance statistics.

**Note:** For some platforms, you must specify K-Fold Cross-Validation in the model control panel. For other platforms, you must specify K-Fold Cross-Validation in the platform launch window. For still other platforms, you must specify K-Fold Cross-Validation through a validation column that contains more than three levels.

**Random Validation Holdback**   (Available as a launch option for specific platforms.) Randomly divides the original data into the training and validation sets. A test set can also be included. You can specify the proportions of the original data to use in each set.

**Leave-One-Out Validation Holdback**   (Available as an option for specific platforms.) Repeatedly fits the model leaving out one observation at a time. Leave-one-out validation is also known as the jackknife procedure.

**Excluded Rows as Validation Holdback**   Uses the excluded rows in the data table as a validation holdback set. For JMP Pro, this option is available by selecting in the platform preferences.

**Note:** For platforms that support using excluded rows as a validation holdback set, the excluded rows are used only when there is no validation column or validation proportion specified in the launch window.

**Table A.2**  K-Fold and Holdback Validation by Platform

| Platform | Excluded Rows as Validation Holdback | Random Validation Holdback | Leave-One-Out Holdback | K-Fold Cross-Validation |
|---|---|---|---|---|
| *Fit Model* | | | | |
| Fit Least Squares | No | No | No | No |
| Stepwise Regression | No | No | No | Yes (for continuous response models only) |
| Logistic Regression | No | No | No | No |
| JMP PRO Generalized Regression | No | Yes | Yes | Yes (though the model controls) |
| JMP PRO Partial Least Squares | No | Yes | Yes | Yes (through the model controls) |
| *Predictive Model* | | | | |
| Neural | Yes | Yes | No | Yes (through model launch or validation column) |
| Partition | Yes | Yes | No | Yes (select the option in the platform preferences) |
| JMP PRO Bootstrap Forest | Yes | Yes | No | No |
| JMP PRO Boosted Tree | Yes | Yes | No | No |
| JMP PRO K Nearest Neighbors | Yes | Yes | No | No |
| JMP PRO Naive Bayes | Yes | Yes | No | No |
| JMP PRO Support Vector Machines | No | Yes | No | Yes (through model launch) |
| *Specialized Models* | | | | |

| Platform | Excluded Rows as Validation Holdback | Random Validation Holdback | Leave-One-Out Holdback | K-Fold Cross-Validation |
|---|---|---|---|---|
| **JMP PRO** Functional Data Explorer | No | No | No | No |
| *Multivariate Models* | | | | |
| Discriminant | Optional | No | No | No |
| Partial Least Squares | No | Yes | Yes | Yes (through model launch or validation column) |
| **JMP PRO** Uplift | No | Yes | | No |

# Appendix B

# References

Agrawal, R., and Srikant, R. (1994). "Fast Algorithms for Mining Association Rules." In *Proceedings of the 20th VLDB Conference*. Santiago, Chile: IBM Almaden Research Center. Accessed July 5, 2016. https://web.stanford.edu/class/cs345d-01/rl/ar-mining.pdf.

Baglama, J., and Reichel, L. (2005). "Augmented implicitly restarted Lanczos bidiagonalization methods." *SIAM Journal on Scientific Computing*, 27:19–42.

Bates, D. M., and Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*. New York: John Wiley & Sons.

Benford, F. (1938). "The law of anomalous numbers." *Proceedings of the American philosophical society*, 551–572.

Benjamini, Y., and Hochberg, Y. (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. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*. 3rd ed. Englewood Cliffs, NJ: Prentice-Hall.

Candes, E. J., Li, X., Ma, Y., and Wright, J. (2009). "Robust Principal Component Analysis?" *Journal of the ACM*, 58:1–37.

Cleveland, W. S. (1994). *Visualizing Data*, Summit, NJ: Hobart Press.

Conover, W. J. (1999). *Practical Nonparametric Statistics*. 3rd ed. New York: John Wiley & Sons.

Cureton, E. E. (1967). "The Normal Approximation to the Signed-Rank Sampling Distribution when Zero Differences are Present." *Journal of the American Statistical Association* 62: 319, 1068–1069.

Hahsler, M. (2015). "A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules." Accessed September 17, 2018. http://michael.hahsler.net/research/association_rules/measures.html.

Hand, D. J., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. Cambridge, MA: MIT Press.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer-Verlag.

Hawkins D. M., and Kass G. V. (1982). "Automatic Interaction Detection." In *Topics in Applied Multivariate Analysis*, edited by D. M. Hawkins, 267–300. Cambridge: Cambridge University Press.

Huber, P. J., and Ronchetti, E. M. (2009). *Robust Statistics*. 2nd ed. New York: John Wiley & Sons.

Hyndman, R. J., Koehler, A. B., Ord, J. K., and Snyder, R. D. (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Berlin: Springer-Verlag.

Jolliffe, I. T. (2002). *Principal Component Analysis*. New York: Springer-Verlag.

Kass, G. V. (1980). "An Exploratory Technique for Investigating Large Quantities of Categorical Data." *Journal of the Royal Statistical Society, Series C* 29:119–127

Lehman, E. L. (2006). *Nonparametrics: Statistical Methods Based on Ranks.* 2nd ed. New York: Springer.

Lin, Z., Chen, M., and Ma, Y. (2013). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055.

Mason, R. L., and Young, J. C. (2002). *Multivariate Statistical Process Control with Industrial Applications*. Philadelphia: SIAM.

McCullagh, P., and Nelder, J. A. (1989). *Generalized Linear Models.* 2nd ed. London: Chapman & Hall.

Nagelkerke, N. J. D. (1991). "A Note on a General Definition of the Coefficient of Determination." *Biometrika* 78:691–692.

Nelder, J. A., and Wedderburn, R. W. M. (1972). "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*. Ph.D. diss., Department of Statistics, North Carolina State University. Accessed June 30, 2016. https://repository.lib.ncsu.edu/ir/bitstream/1840.16/10572/1/etd.pdf.

Platt, J. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines*. Technical Report MST-TR-98-14, Microsoft Research. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.pdf.

Qian, P. Z., Huaiqing, W., and Wu, C. F. (2012). "Gaussian process models for computer experiments with qualitative and quantitative factors." *Technometrics* 50:383–396.

Ramsay, J. O., and Silverman, B. W. (2005). *Functional Data Analysis*. 2nd ed. New York: Springer.

Ratkowsky, D. A. (1990). *Handbook of Nonlinear Regression Models*. New York: Marcel Dekker.

Sall, J. (2002). "Monte Carlo Calibration of Distributions of Partition Statistics." SAS Institute Inc.,Cary, NC. Accessed July 29, 2015. https://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.

SAS Institute Inc. (2020). *SAS/ETS 15.2 User's Guide*. Cary, NC: SAS Institute Inc. https://go.documentation.sas.com/api/docsets/etsug/15.2/content/etsug.pdf.

Schäfer, J., and Strimmer, K. (2005). "A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics." *Statistical Applications in Genetics and Molecular Biology* 4, Article 32.

Schuirmann, D. J. (1987). "A Comparison of the Two One-sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability." *Journal of Pharmacokinetics and Biopharmaceutics* 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, US Department of Commerce, Bureau of the Census.

Shmueli, G., Patel, N. R., and Bruce, P. C. (2010). *Data Mining For Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner*. 2nd ed. Hoboken, NJ: John Wiley & Sons.

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, NJ: John Wiley & Sons.

Westfall, P. H., Tobias, R. D., and Wolfinger, R. D. (2011). *Multiple Comparisons and Multiple Tests Using SAS*. 2nd ed. Cary, NC: SAS Institute Inc.

White, K., Szarka, J., and Jensen, W. (2018). "A Recommended Set of Indices for Evaluating Process Health." Presentation at the 2018 Fall Technical Conference, sponsored by the American Statistical Association and the American Society for Quality.

# Appendix **C**

# **Technology License Notices**

- Scintilla is Copyright © 1998–2017 by Neil Hodgson <neilh@scintilla.org>.

  All Rights Reserved.

  Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

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

- Progressë® Telerik® UI for WPF: Copyright © 2008-2019 Progress Software Corporation. All rights reserved. Usage of the included Progress® Telerik® UI for WPF outside of JMP is not permitted.

- ZLIB Compression Library is Copyright © 1995-2005, Jean-Loup Gailly and Mark Adler.

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

- Packages is Copyright © 2009–2010, Stéphane Sudre (s.sudre.free.fr). All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  Neither the name of the WhiteBox nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- iODBC software ia Copyright © 1995–2006, OpenLink Software Inc and Ke Jin (www.iodbc.org). All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  – Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  – Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  – Neither the name of OpenLink Software Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

- This program, "bzip2", the associated library "libbzip2", and all documentation, are Copyright © 1996–2019 Julian R Seward. All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

  3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

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

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

Julian Seward, jseward@acm.org

bzip2/libbzip2 version 1.0.8 of 13 July 2019

- R software is Copyright © 1999–2012, R Foundation for Statistical Computing.

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

- libopc is Copyright © 2011, Florian Reuter. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  – Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  – Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and / or other materials provided with the distribution.

  – Neither the name of 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

2. 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.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

- Snowball is Copyright © 2001, Dr Martin Porter, Copyright © 2002, Richard Boulton.

  All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  2. 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.

  3. Neither the name of the copyright holder 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 HOLDER 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.

- Pako is Copyright © 2014–2017 by Vitaly Puzrin and Andrei Tuputcyn.

  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 THE AUTHORS OR COPYRIGHT HOLDERS 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.

- HDF5 (Hierarchical Data Format 5) Software Library and Utilities are Copyright 2006–2015 by The HDF Group. NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 1998-2006 by the Board of Trustees of the University of Illinois. All rights reserved. DISCLAIMER: THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

- agl-aglfn technology is Copyright © 2002, 2010, 2015 by Adobe Systems Incorporated. All Rights Reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  – Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  – Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  – Neither the name of Adobe Systems Incorporated 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 HOLDER 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.

- dmlc/xgboost is Copyright © 2019 SAS Institute.

  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- libzip is Copyright © 1999–2019 Dieter Baron and Thomas Klausner.

  This file is part of libzip, a library to manipulate ZIP archives. The authors can be contacted at <libzip@nih.at>.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  2. 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.

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

  THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``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 AUTHORS 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.

- OpenNLP 1.5.3, the pre-trained model (version 1.5 of en-parser-chunking.bin), and dmlc/xgboost Version .90 are licensed under the Apache License 2.0 are Copyright © January 2004 by Apache.org.

  You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- – You must give any other recipients of the Work or Derivative Works a copy of this License; and

- – You must cause any modified files to carry prominent notices stating that You changed the files; and

- – You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- – If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

- – You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

- LLVM is Copyright © 2003–2019 by the University of Illinois at Urbana-Champaign.

  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- clang is Copyright © 2007–2019 by the University of Illinois at Urbana-Champaign.

  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS", WITHOUT WARRANTIES OR CONDITIONS OF

ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- lld is Copyright © 2011–2019 by the University of Illinois at Urbana-Champaign.

  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS", WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- libcurl is Copyright © 1996–2021, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS file. All rights reserved.

  Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

  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 OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS 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 a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.