A Technical Publication for JMP® Users
Issue 27 Winter 2011

JMPer Cable

Inside This Newsletter

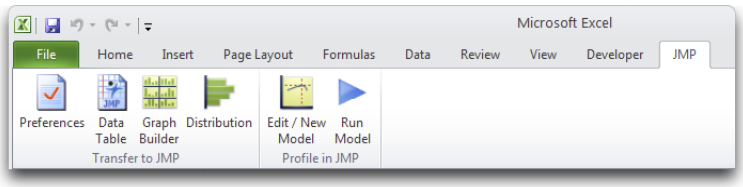
JMP and Excel	1
Is it Uniform?	4
Reliability Roars Back to JMP Training	6
File Type Associations in JMP	7
A New 'Little JMP Book'	8
Using JMP 9 Scripting to Process Image Data	9
Splitting Columns with a Group Variable	12
JMP Add-Ins	14
New in JMP 9: Normal Mixtures Fit in Distribution Platform	15

JMP and Excel

Paul Nelson, JMP Division of SAS Institute

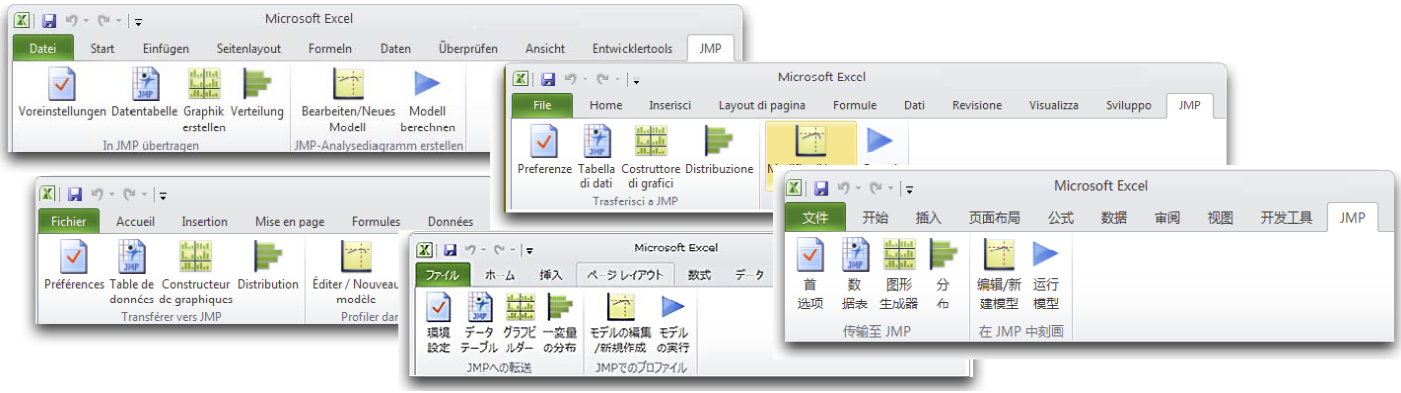
Heads up Excel users! JMP 9 offers a bonanza to Excel users everywhere. A new Excel add-in from JMP easily moves data from Excel to JMP and can launch basic graphing platforms automatically. Your valuable Excel formula compositions can now be visualized interactively in the JMP profiler, letting you drag to instantly see the effects when you change formula factor values.

The Excel add-in places a JMP tab on the Excel ribbon. The JMP tab creates buttons arranged in two panels. The first panel, labeled Transfer to JMP, gives you the ability to transfer data from Excel to JMP. Preferences let you use information from one or more of the Excel rows as column names in the new JMP table. The Transfer to JMP section of the ribbon also has buttons that create the JMP table and then automatically access either the Graph Builder or the Distribution launch window. Here is the English version of the Excel JMP tab as it displays in Excel 2010.



The second panel, labeled Profile in JMP, takes the Excel add-in's ability much further. Many Excel users store formulas or models in an Excel spreadsheet. The first button, Edit/New Model, provides a dialog for you to convey the structure of an Excel formula to JMP. The second button, Run Model, uses the formula to display the data in JMP's interactive profiler.

Both Excel and JMP are longstanding global applications and this new Excel add-in is designed for users in all five JMP localizations. When you install JMP, the Excel add-in places the translated JMP button panels onto the version of Excel in use—English, Japanese, Simplified Chinese, German, French, or Italian. The next sections of this article explain how to use the new Excel add-in.



JMP and Excel (continued)

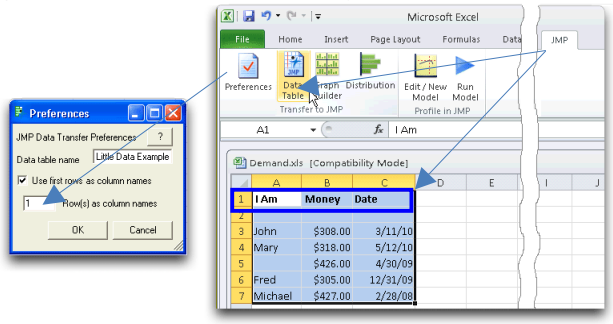
On Windows, the JMP 9 installer automatically installs JMP's new Excel add-in into Excel. This new Excel add-in gives JMP and Excel users a simple but powerful Excel-to-JMP transfer capability, access to Graph Builder and Distribution platforms from Excel, and the ability to enable JMP's profiler to display calculation models residing in Excel workbooks.

Excel-to-JMP Data Transfer

There has been a longstanding issue of how to easily get data from Excel into JMP. To create a JMP table from Excel,

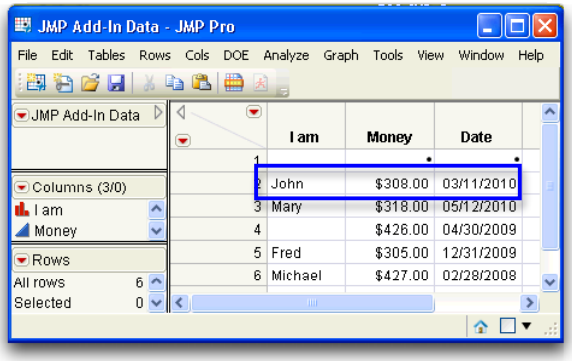
- Click the JMP tab showing on the Excel tab bar to see the JMP add-in options.
- Drag to select the Excel data you want.
- Click on the Data Table icon in the Transfer To JMP section of the JMP add-in menu.
- Optionally, before you transfer the data to JMP, use Preferences (see *Figure 1*) to name the table and specify the number of rows in the Excel table to use as column names in the new JMP table.

Figure 1 Select Excel Data and Use Preferences



Note in *Figure 2* that the new JMP column names correspond to the first row in the Excel spreadsheet. There are three columns, one character column and two numeric columns correctly formatted as currency and date. There are missing values in the JMP table corresponding to empty cells in the Excel spreadsheet.

Figure 2 JMP Table Created by Excel Ad-In

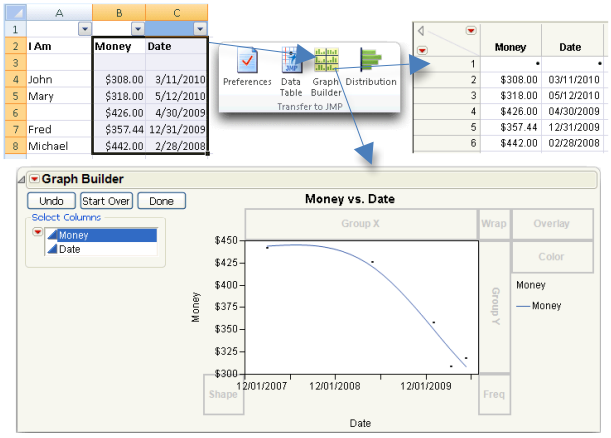


Excel-to-JMP Transfer and Plot Data

The Transfer to JMP section on the Excel ribbon also shows icons for the Distribution and Graph Builder platforms in JMP. If you click these icons, the Excel add-in creates the JMP table containing the highlighted rows and columns in the Excel spreadsheet and opens either the Distribution or the Graph Builder launch window.

Figure 3 illustrates what happens when you select a subset in the Excel spreadsheet (Money and Date) and click on the Graph Builder icon. The add-in creates the JMP table shown on the right and opens the Graph Builder window. This simple example shows a plot of Money by Date.

Figure 3 Transfer Data To JMP and Create a Graph or Plot

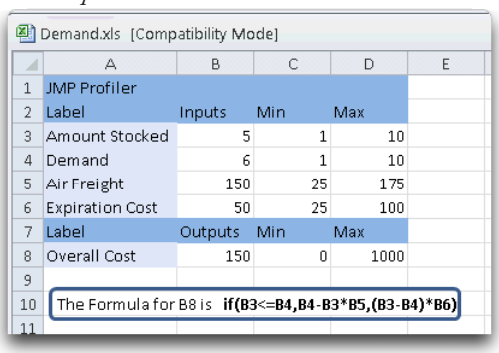


The JMP Profiler and Excel Formulas

Excel users often store formulas and models with data in their spreadsheets, making it convenient to do things like 'what-if' analyses and generate simple charts. Now, the new Excel add-in lets you use the JMP profiler to display these models.

Figure 4 shows an example spreadsheet with an Excel formula that lets you see the effect of changing Amount Stocked Demand, Air Freight, and Expiration Cost on Overall Cost. You can find this table, called Demand.xls, in the Sample Import Data folder installed with JMP.

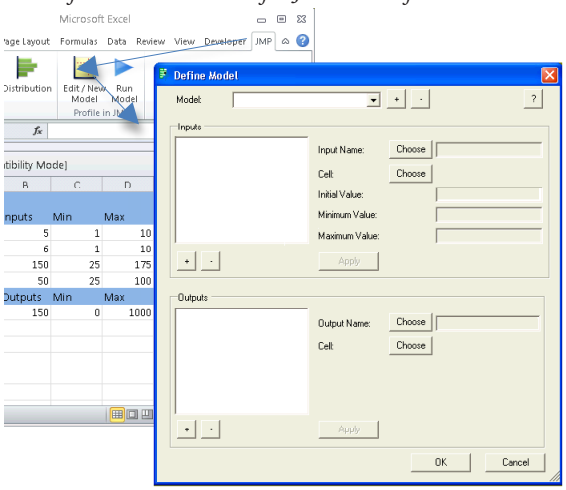
Figure 4 Excel Spreadsheet with Formula



JMP and Excel (continued)

The JMP profiler gives you a dialog for conveying the model to JMP. To construct a model, click the Edit / New Model button in the Profile in JMP group found on the Excel ribbon buttons. These commands open the Define Model window shown in *Figure 5*.

Figure 5 Define an Excel Model for JMP to Profile



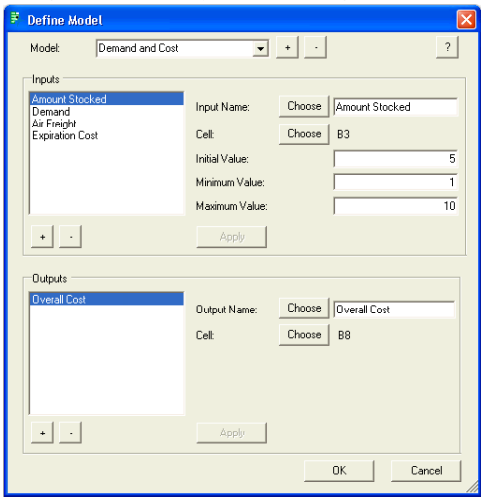
You want to complete the model window with descriptions of inputs and outputs, and the cells in the Excel spreadsheet that give them values. Here are the steps to configure a model from the example data.

1. Press + next to the Model text box. Then enter any model name you want. The example uses the name Demand and Cost. (See the completed model in *Figure 6*).
2. Press + beneath the Inputs box to add an Input.
3. Enter the Input Name or use the Choose button to select a cell from the worksheet.

Note: If you select Choose, the Excel spreadsheet becomes the active window so you can choose a cell. The model-building dialog does not show again until you have chosen a cell.

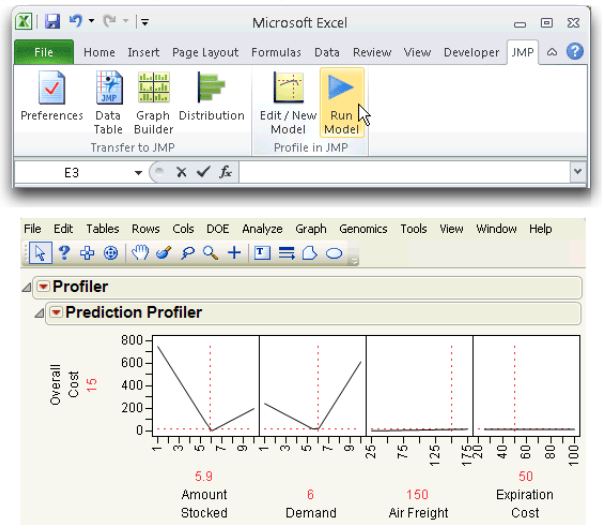
4. Select the Cell that is an input to the calculation formula(s). This momentarily hides the dialog so a cell can be selected with the mouse. Initial Value is then set to the selected cell value.
5. By default, Minimum Value and Maximum Value are computed as ± 5% of the initial value previously set by Cell selection. Change minimum, maximum and initial values if you need values other than the defaults.
6. Select Apply after the last input, or repeat the previous steps until finished with selecting inputs.
7. Press + under the Outputs list to add an output.
8. Set the Output Name.
9. Select the cell that is the result of the formula's computation.
10. For more inputs repeat from Step 7; otherwise press Apply.
11. Press OK to close the dialog and save the information to the Excel workbook.

Figure 6 Completed Model Definition Dialog



To run the completed model, select Run Model from the Excel ribbon. A dialog opens with a list of all models defined for the workbook. Select the model you want and click the Profile in JMP button on the dialog. JMP runs a hidden copy of Excel in the background, modifying the input cells and reading the output cells, to drive the live profiler, as shown in *Figure 7*.

Figure 7 JMP Profiler Showing Excel Formula



Technical Notes

The JMP installation program automatically installs Excel Profiler Add-in if it detects Office 2003, Office 2007, or Office 2010.

The installation installs the appropriate Primary Interop Assemblies for Office, .Net 4 (required by JMP 9) and the two COM Add-in components of the JMP profiler.

On Office 2003 systems without the ribbon menu, a JMP menu is installed in the toolbar.

Is It Uniform?

José G. Ramírez, Ph.D., W.L. Gore and Associates, Inc.

If you have taken a basic course in probability and statistics you were probably introduced to the Uniform distribution. Don't blame yourself if it isn't foremost in your memory; the Uniform distribution doesn't have much appeal because it describes a phenomenon with constant probability. It seems hard to find applications in engineering and science for which the probability of occurrence is constant for all the values in a given interval. Pseudo-random numbers between 0 and 1 come to mind since any number in this interval should have the same probability of occurring. The Uniform distribution has sometimes been used as a model for the distribution of traffic along a straight road.

In general, probability distributions are mathematical models, or equations, used to describe and quantify the degree of uncertainty that we observe in our data. Under a continuous uniform probability model, the likelihood of observing a value x in the interval $[\theta, \theta+\sigma]$ is constant and equal to, $1/\sigma$ or 1 divided by length of the interval. The mathematical equation describing a continuous uniform (rectangular) probability density function is

$$f(x) = \begin{cases} \frac{1}{\sigma} & for \;\; \theta \le x \le \theta+\sigma \;\&\; \sigma > 0 \\ 0 & for \;\; x < \theta \;\; or \;\; x > \theta+\sigma \end{cases}$$

One of the readers of our blog, StatInsights,

http://statinsights.blogspot.com

asked how to fit a Uniform distribution in JMP. Uniform is not one of the choices in the Continuous Fit contextual menu of the Distribution platform. However, Beta is one of the choices.

Uniform(θ, θ + σ) is Beta(1, 1, θ, σ)

Fortunately, there is a convenient relationship between a Beta distribution, which is one of the JMP choices, and the Uniform distribution. The general form of the Beta probability density function is

$$f(x) = \frac{1}{B(\alpha,\beta)\sigma^{\alpha+\beta-1}}(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}$$
$$for \;\; \theta \le x \le \theta+\sigma; \;\; 0 < \alpha, \beta, \sigma$$

where $B(\alpha, \beta)$ is the Beta function. This pdf generalizes the standard 2-parameter Beta distribution, *Beta*(α, β), from the interval [0,1] to an arbitrary bounded interval $[\theta, \theta+\sigma]$. When the shape parameter values are $\alpha = 1$ and $\beta = 1$, then the beta function $B(1, 1) = 1$ and the pdf has constant probability equal to $1/\sigma$. In other words, if X is distributed as uniform in the interval $(\theta, \theta + \sigma)$, then X is distributed as Beta(1, 1) with threshold θ and scale σ.

Let's explore this relationship. The histogram in *Figure 1* shows 100 simulated observations from a Uniform distribution in the interval (0,1). These observations were generated using the Random Uniform generator in the Formula Editor.

As expected, this distribution has a somewhat rectangular shape. To fit a Uniform distribution to this data, select Continuous Fit > Beta from the contextual menu (red triangle) on the histogram Uniform(0, 1) title bar.

Figure 1 Histogram and Continuous Fit > Beta Command

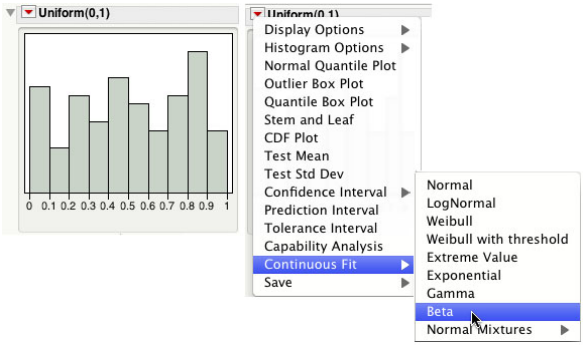
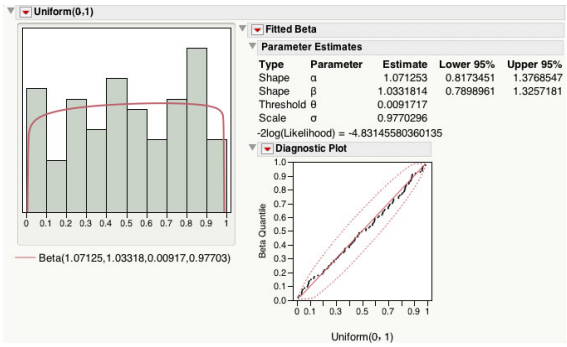


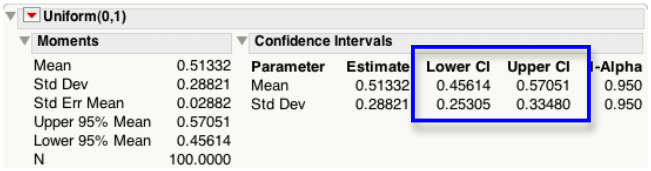
Figure 2 shows the histogram with the superimposed Beta fit, which has a rectangular shape. The fitted Beta Parameters Estimates show α=1.07 and β=1.03 (close to 1—but how close?). The 95% confidence intervals for α and β both include 1, indicating that there is not enough evidence to say that they are different from 1. In practical terms, that means α and β are assumed to be 1 so the Uniform distribution does a good job at describing this data. The threshold parameter, θ, is 0.009 and scale parameter, σ, is 0.977 supporting a Uniform(0, 1) (Note that θ and σ are not maximum likelihood estimates. JMP sets θ to the minimum data value, and σ to the range (=maximum-minimum) of the data). Finally the Diagnostic Plot shows the points hovering closely around the line, and within the 95% confidence bands.

Figure 2 Beta Fit to Uniform(0, 1) Data



For a Uniform(θ, θ+σ), the mean is equal to θ+σ/2 and the standard deviation is equal to σ/√12 . The Moments table in *Figure 3* shows the mean to be close to 0.5 and the standard deviation to be close to 1/√12 = 0.2886. Both the 95% confidence intervals for the mean and standard deviation contain 0.5 and 0.2886, respectively.

Figure 3. Moments and Confidence Intervals for Uniform(0, 1) Data



What to Look for When Fitting the Beta(1, 1, θ, σ)

To use the beta distribution fit to see if the Uniform distribution is a good approximation for data, follow these steps.

1. Fit a Beta distribution in the Distribution platform. Select Continuous Fit > Beta from the red triangle menu on the Histogram title bar.
2. Look at the Parameter Estimates report and verify that the estimates for the shape parameters α and β are close to 1.
3. Verify that the 95% confidence intervals for α and β include 1.
4. Select Diagnostic Plot from the red triangle menu on the Fitted Beta title bar. The points on the diagnostic plot should fall close to the straight line.
5. The Uniform(θ, θ+σ) parameters are given by the threshold θ and scale σ.
6. Select Confidence Intervals > .95 from the red triangle menu on the Histogram to see 95% confidence intervals for the mean and standard deviation (see *Figure 3*).
7. Check that θ+σ/2 (mean) and σ/√12 (Std. Dev.) are within the intervals.

Example: The Brisbane Baby Boom Data

December 18, 1997 was a record-breaking day for Brisbane in Queensland, Australia, Forty four babies were born in a 24-hour period at the Mater Mothers' Hospital.

Figure 4 shows the histogram of the number of minutes since midnight for each birth occurring that day. It is reasonable to believe that in a 24-hour period a birth can occur at any minute, and the histogram seems to support that. Will a Uniform distribution fit the data well?

Figure 4 Number of Minutes Since Midnight for Each Birth

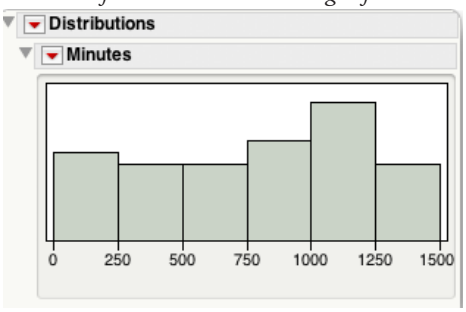


Figure 5 shows the results for the Beta fit. The Parameter Estimates, α=1.357 and β=1.154, are in the vicinity of one, and their 95% confidence intervals contain 1. The Diagnostic Plot looks reasonably straight. The Threshold = 5 and the Scale =1430, suggests that a Uniform(5, 1435) distribution describes the data. In other words, the probability of being born on December 18,1997, in any given minute in a 24-hour interval, at the Mater Mothers' Hospital in Brisbane, Australia is 1/1430 = 0.0007 or 0.07%.

Figure 5 Beta Fit for Brisbane's Babies Births

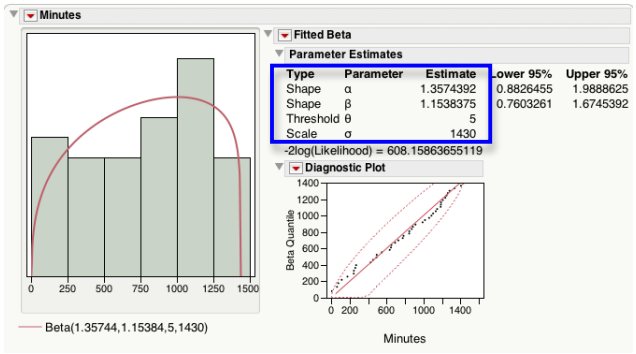
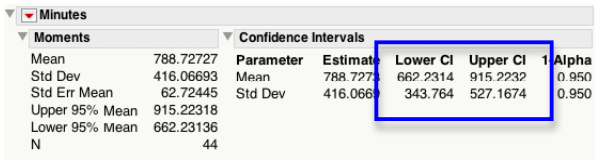


Figure 6 shows the moments and 95% confidence intervals for the mean and standard deviation. For a Uniform(5, 1435) the mean should be close to 5 + 1430/2 = 720, and standard deviation close to 1450/√12 = 412.8054. The 95% confidence interval for the mean does contain 720, and the one for the standard deviation contains 412.8054.

Figure 6. Moments and Confidence Intervals for Beta Fit



Next time you are wondering if your data can be described by a Uniform distribution, think 'beta' and use the Distribution platform with Continuous Fit > Beta fit.

Reliability Roars Back to JMP Training

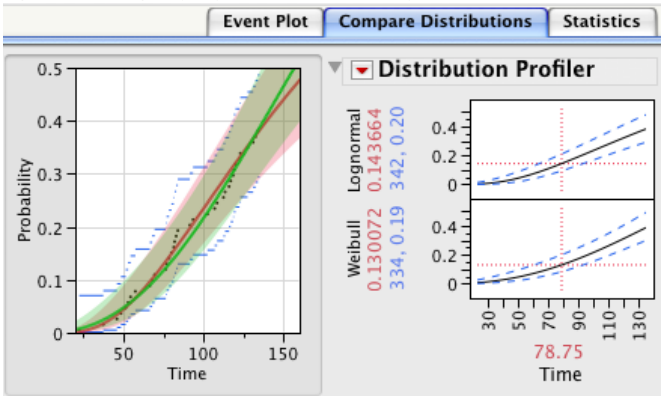
Mark Bailey, Education Division of SAS Institute Inc.

JMP Training offers a one-day course for the analysis of life data in the context of reliability engineering. The original edition of this course first appeared in 1996 among a handful of introductory courses for JMP users. It focused on common analyses and highlighted the analytical capabilities of JMP 3. The past fifteen years has brought much advancement in the theory, modeling, and estimation methods for reliability data. JMP 8 started to incorporate these advancements by introducing two new reliability platforms, Life Distribution and Fit Life by X. With the completion of this development cycle in JMP 9, it is time to write a new reliability course that incorporates all of these advances. So, what are you in store for?

The Life Distribution Platform

Life Distribution is a highly interactive platform designed to explore a wide variety of distribution models. It features an Event Plot and a dynamic probability plot, to compare candidate models. It initially presents non-parametric failure probability estimates with confidence intervals on linear scales. You can select one or more alternative models, dynamically change the axes to assess goodness of fit, or systematically fit all possible models, and then review a variety of statistics and diagnostics for each model. Each fit is given its own set of profilers to explore distributions, quantiles, hazard, and density. If you identify different failure modes when you launch Life Distribution, then it determines which modes are critical and estimates the improvement in the reliability. *Figure 1* illustrates the kinds of Life Distribution results you learn to create and interpret

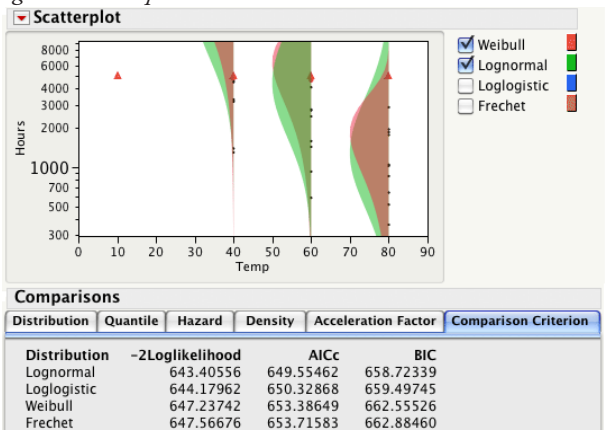
Figure 1 Example of Life Distribution Platform Results



The Life by X Platform

Fit Life by X is another highly interactive model that can be used for a variety of situations such as accelerated life tests or or accelerated degradation tests. A covariate (X) is usually a stress factor, such as temperature or current, which affects the reliability distribution. You can choose from a built-in set of common acceleration relationships, such as Arrhenius for temperature, and common models. You visualize the effect in a scatter plot that overlays density curves for any X and a quantile contour for any probability (*Figure 2*). The nested model tests provide diagnostic plots and statistics to help you decide if you need a separate location and scale for each stress level. Of course, you can predict the reliability at nominal conditions

Figure 2 Scatterplot with Distributions and Statistics



These methods are the same ones that are used in survival analysis and, more generally, in any situation where the response is the *time to an event*. As such, these methods have broader applications than originally foreseen, in areas such as data mining and Lean Six Sigma projects.

Our new course is currently under development. It will present these advancements and those soon to premiere in JMP 9, including a new platform dedicated to the analysis of degradation and another for DOE for reliability tests! It will initially be offered as an exclusive training event at the JMP Discovery Summit 2011 in September. This half-day workshop will be expanded after the conference into a full day of training, which will be regularly offered to public classes, held in our SAS training centers and Live Web sessions, and to private classes, at customer sites.

Please let us know about any topics or examples that you would like to see in our training for reliability analysis. Please write directly to me at

mark.bailey@sas.com

with your suggestions.

File Type Associations in JMP

Win LeDinh, SAS Institute, Technical Support

A common question that JMP Technical Support receives deals with file type associations on Windows operating systems. For example, when double clicking on an Excel file, JMP is no longer the default application that opens the file. Or, if multiple versions of JMP are currently installed and JMP 9 is installed last, the file type associations are no longer associated to an earlier version that a user may want to use. This article discusses how to correct these problems.

First, here is some background information on how JMP manages file type associations. JMP files have the following file types, which are described in the *Using JMP* book.

- .jmp (a data table)
- .jrp (a report), .jrn (a journal)
- .jmpprj (a project)
- .jsl (a script)
- .jmpcust (customization set)

These file types are automatically recognized and opened up by JMP when you double click on them. This is because when you install JMP, the installer updates the Windows Registry to associate the installed JMP version with all of these JMP file types. This is accomplished by merging a Windows registry file named JMP.reg located under

<default installed dir>\Resources\RegFile\

folder. For JMP 9 that is:

C:\Program Files\SAS\JMP\9\Resources\RegFile\JMP.reg

In addition to JMP files, JMP can also read and recognize other file types such as .xls/.xlsx (Excel), .txt (text), .csv (comma separated values text), and .sas7bdat (SAS data set). However, this list of files is opened using the current default file type associations set up by the Windows operating system. Even though you may end up using JMP to work with these file types, the association to JMP is not done automatically for you.

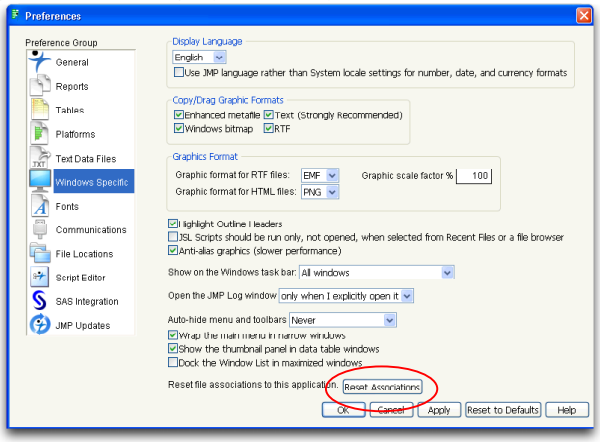
Here are two kinds of JMP file type association problems you might want to fix:

– Re-associate JMP files that are no longer associated with JMP.

– Associate JMP files with a specific version when multiple versions of JMP are installed.

The solution is to use the JMP Preferences Reset Associations command. First, run the specific version of JMP that you wish to associate with your JMP files. Access the JMP menu File > Preferences and click on the Windows Specific category (see *Figure 1*). Click the Reset Associations button, then click OK to close the preferences window, and exit JMP. Double click on a JMP file again to verify that the correct JMP version opens your JMP file.

Figure 1 Windows Specific JMP Preferences Panel



Clicking on the Reset Associations button from within a specific version of JMP tells the Windows operating system to use that same version of JMP to open a JMP file in response to a double click on it. Behind the scenes, what is really happening is that the JMP.reg file is read into the Windows Registry. That is, JMP rebuilds and fixes the JMP file type associations from the JMP.reg file.

> **Note:** Because of this role that the JMP.reg file plays, it is very important that the JMP.reg file is left untouched and unmodified by any user or process.

The Reset Associations button does not fix all file type associations. For example, you might want to have Excel or text files automatically opened in JMP. To do this, you first need to examine the current Windows file type associations Then, either create new ones or fix existing ones. To do so, follow these steps.

1) In Windows Explorer (such as when opening 'My Computer' or 'My Document'), select Tools > Folder Options.

2) In Windows Explorer (such as when opening 'My Computer' or 'My Document'), select Tools > Folder Options.

File Type Associations in JMP (continued)

3) Click on the File Type tab.
4) Look for a JMP filetype like 'JMP' or an Excel filetype like 'XLS' under the Extensions column, and click on the entry.
5) Near the bottom where it says Opens with: , click on Change. Otherwise click on the New button if a filetype does not exist
6) In the Open With window, select JMP Application if shown. If not shown, click on Browse and navigate to the specific JMP version you wish to use, such as C:\Program Files\SAS\JMP\8\jmp.exe
7) Click on OK to close all the windows
8) Reboot your computer
9) Double click on a file to see if the correct JMP version opens it.

If the Reset Associations button and the Windows Folder Options solution did not fix the problem then the last thing to check is your Windows registry. Bring up the Windows registry editor by clicking on Start > Run, then type 'regedit' in the Open text field. Look for the following registry entry and make sure the value matches what is shown here:

HKEY_CLASS_ROOT\JMP.Document\shell\open\command
= C:\Program Files\SAS\JMP\8\jmp.exe /dde

Navigate the registry folder structure by clicking on the '+' sign to expand each nested folder, until you reach the last value. In this case, once you reach the open folder, click on command to see the default value. If it does not match the path to your JMP version, right click on the Default name and select Modify. Copy and paste in the value data, and click OK to close that dialog. Then exit the Windows Registry by selecting File > Exit. Now your double clicking on a file should be correctly associated with JMP.

Think Ahead—

—to September 13-16 and the 2011 JMP Discovery Summit. This year's conference is being held at the Denver Marriott City Center in Denver, CO. Last year's Discovery Summit attendees said they especially enjoyed

- seeing new release features in action,
- watching problem-solving showcases presented by noted analysts,
- seeing how other companies apply analytics and visualization to specific business problems,

and much more.

So, put September 13-16 on your calendar and plan to be in Denver for the 2011 JMP Discover Summit.

A New 'Little JMP Book'

…Introducing *Discovering JMP*, the newest book in the JMP user guide series. At only 150 pages and $9.95, it is the perfect way for a new user to become familiar with JMP software.

Discovering JMP doesn't try to cover every command and option available in any of the ever-growing number of platforms and features in JMP. Instead, it targets selected critical need-to-know features that will help every new user get get started. The book includes an overview and basic instructions on

- working with data—getting data into JMP, calculating values, filtering, subsetting, combining and restructuring tables.
- visualizing data—histograms and bar charts for single variables, scatterplots, overlay plots, box plots, bubble plots for multiple variables, building custom plots.
- analyzing data and graphing analyses—distributions of continuous and categorical values, simple regression, multiple regression, comparing proportions, comparing averages for single and multiple variables .
- saving and sharing work—creating a journal, creating a JMP project, saving and running scripts,
- special features in JMP—updating analyses and graphs with automatic recalculation when values change, customizing JMP with preferences, creating and submitting SAS code.

Discovering JMP was released with JMP 9 in October 2010, and you can access it using Help > Books on the main menu.

Best in Show for *Discovering JMP*

The Society for Technical Communication honored the JMP book *Discovering JMP* as both "Distinguished" and "Best in Show" in its 2010-2011 competition in the category of User Support Materials.

Discovering JMP was translated in five languages for the release of JMP 9.0—so users in Italy, France, Germany, China and Japan now have a reference in their own languages to help them get started with JMP.

Using JMP 9 Scripting to Process Image Data

John Sall, JMP Division of SAS Institute

I spent some time earlier this year playing with large *hyperspectral* image files. While the human eye just sees visible light in three specific wavelengths, hyperspectral sensors see with many more wavelengths, including infrared and ultraviolet.

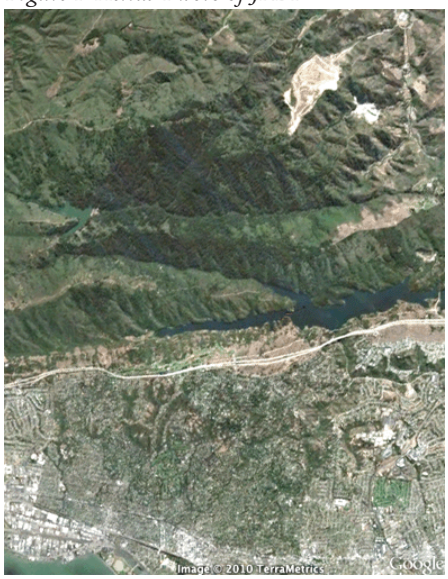
Some applications of hyperspectral analysis include

- airborne or satellite ground cover analysis
- product inspection for quality control
- military intelligence.

I used sample data found on the web from JPL, the Jasper Ridge AVIRIS data—one of their free samples at the time. The project included images of Jasper Ridge Biological Preserve (JRBP). An aircraft takes pictures with a special camera that works in 224 colors, from infrared to ultraviolet, instead of the normal 3-colors that human-eye-oriented images are taken.

Figure 1 is a standard aerial view of the Jasper Ridge area that shows general topological features. Spectral images can find and emphasize specific features.

Figure 1 Aerial Photo of JRBP



However, the spectral images are composed of very large amounts of data that are challenging to process and interpret. So I wrote a principal components analysis to help bring out the prominent features.

The following is a brief tour of a JSL script that uses new JMP 9 features to read the Jasper Ridge image data, process the data, and present images using the first four principle components.

The script starts with

```
NamesDefaultToHere(1);
```

This new JMP 9 statement makes the program variable names local to the script so that you don't pollute the global name space. This reduces the chance that your script will interact adversely with other scripts.

Next, the script prompts the user to select a directory.

```
inDir = PickDirectory("Navigate to directory
   with AVIRIS rfl files", defaultDir,
   showEdit(1));
```

Then the script makes a list of files with suffix `.rfl`

```
filelist = filesInDirectory(inDir);
ni = nItems(filelist);
rflFiles = {};
for(i=1,i<=ni,i++,if(EndsWith(fileList[i],".
   rfl"),
   InsertInto(rflFiles,fileList[i])));
```

Then loop through the `.rfl` files.

```
for(iFile=1, iFile<=nRFL, iFile++,
   file = rflFiles[iFile];
```

The `.rfl` files are binary, so the file needs to be read as a *binary large object* or 'BLOB', as follows:

```
fBlob = loadTextFile(inDir||file,blob);
sz = length(fBlob);
```

Each image has 224 integer binary values for a given pixel, and there are 614 pixels in a scan line, and usually 512 scan lines for an image file. The data needs to be in a matrix that is 224 columns (wavelengths) and 314,368 rows (pixels). The new `Blob To Matrix` function converts the integer-binary data to a matrix of values, using these dimensions.

```
nchannel = 224; // colors/wavelengths
nsample = 614; // pixels across each line
nLines = 512; // lines in the image
rawData = Blob To Matrix( fBlob, "int", 2,
   "big", nchannel );
fBlob = 0; // blob is no longer needed
nLines = (sz/2)/(nsample*nchannel); // if
   not 512
n = nsample*nLines; // data table rows
```

The `BlobToMatrix` arguments are

- the object (fBlob)
- the binary data type ("int")
- the length in bytes (2)
- whether the binary integers are 'big' endian or 'little' endian ("big")
- the number of columns to make the matrix (nchannel).

The next task is to standardize the data, although not all similar application do so.

```
stdv = vstd(rawData);
stdv += stdv == 0; // don't divide by zero
stdData = (rawData - J(n,1,1)*vmean(
    rawData)):/(J(n,1,1)*stdv); //standardize
    rawData = 0;
```

Note that the prevention of zero divide, which leads to missing values, is done with `stdv += stdv==0;`

Note: The new `Correlation` function calculates the correlations using multithreading and takes advantage of symmetry, so it is faster than using matrix operations. Speed is important in this application because it is a 224 by 224 correlation matrix from 314,368 rows; the correlation operation takes much longer than anything else in the script.

```
corr = Correlation(stdData);
```

Now I want the first four principal components. For that, I obtain the eigenvectors of the correlation matrix.

```
{m,e} = eigen(corr); show(m[1::10,0]);
pc1v=shape(stdData*e[0,1],nLines,nsample)`;
pc2v=shape(stdData*e[0,2],nLines,nsample)`;
pc3v=shape(stdData*e[0,3],nLines,nsample)`;
pc4v=shape(stdData*e[0,4],nLines,nsample)`;
```

The multiplication gives the principal component scores. The `shape` command forms a 512 by 614 matrix, which is more appropriate for constructing images.

Next, rescale to a scale of zero to one and free memory of unneeded large matrices.

```
pc1s =
(.5+pc1v/(2*max(pc1v))):*(pc1v>0)+(.5-
pc1v/(2*min(pc1v))):*(pc1v<0);
pc2s =
(.5+pc2v/(2*max(pc2v))):*(pc2v>0)+(.5-
pc2v/(2*min(pc2v))):*(pc2v<0);
pc3s =
(.5+pc3v/(2*max(pc3v))):*(pc3v>0)+(.5-
pc3v/(2*min(pc3v))):*(pc3v<0);
pc4s =
(.5+pc4v/(2*max(pc4v))):*(pc4v>0)+(.5-
pc4v/(2*min(pc4v))):*(pc4v<0);
stdData = u = v = 0; // free memory
```

I make images using `HeatColor` to transform the 0 to 1 scale into blue-to-gray-to-red scale, then create pictures using `NewImage`.

```
imagepc1 = NewImage(HeatColor(pc1s));
imagepc2 = NewImage(HeatColor(pc2s));
imagepc3 = NewImage(HeatColor(pc3s));
imagepc4 = NewImage(HeatColor(pc4s));
```

I want to show the eigenvectors so that you can see what each picture is portraying, i.e. which frequencies each component is using to score with.

```
imageEig1 = NewImage(DirectProduct(
    HeatColor(e[0,1]`/2+.5),J(12,2,1)));
imageEig2 = NewImage(DirectProduct(
    HeatColor(e[0,2]`/2+.5),J(12,2,1)));
imageEig3 = NewImage(DirectProduct(
    HeatColor(e[0,3]`/2+.5),J(12,2,1)));
imageEig4 = NewImage(DirectProduct(
    HeatColor(e[0,4]`/2+.5),J(12,2,1)));
```

The `DirectProduct` with a J-matrix repeats the values into multiple rows giving a 224 by 12 presentation of the eigenvectors to show for each image.

Now I present my results, alternating each picture with the eigenvector scores as another image.

```
NewWindow(rflFiles[iFile]||" PCA",
    LineupBox(ncol(2),spacing(3),
        imageEig1,imageEig2,
        imagepc1,imagepc2,
        imageEig3,imageEig4,
        imagepc3,imagepc4));
```

The PC1 image in *Figure 2* shows the first principal component. It has moderate coefficients across the whole range of frequencies and thus measures the general intensity level. These heat-map colored images are blue for small values and red for large values. The lake is blue because it is dark in the original.

The second principal component (PC2 in *Figure 2*) loads higher values on the mid-low end of the spectrum. It shows the red end compared to the green and blue end of the spectrum. Here you see buildings much more clearly—they didn't show up in the general intensity, but the color was different.

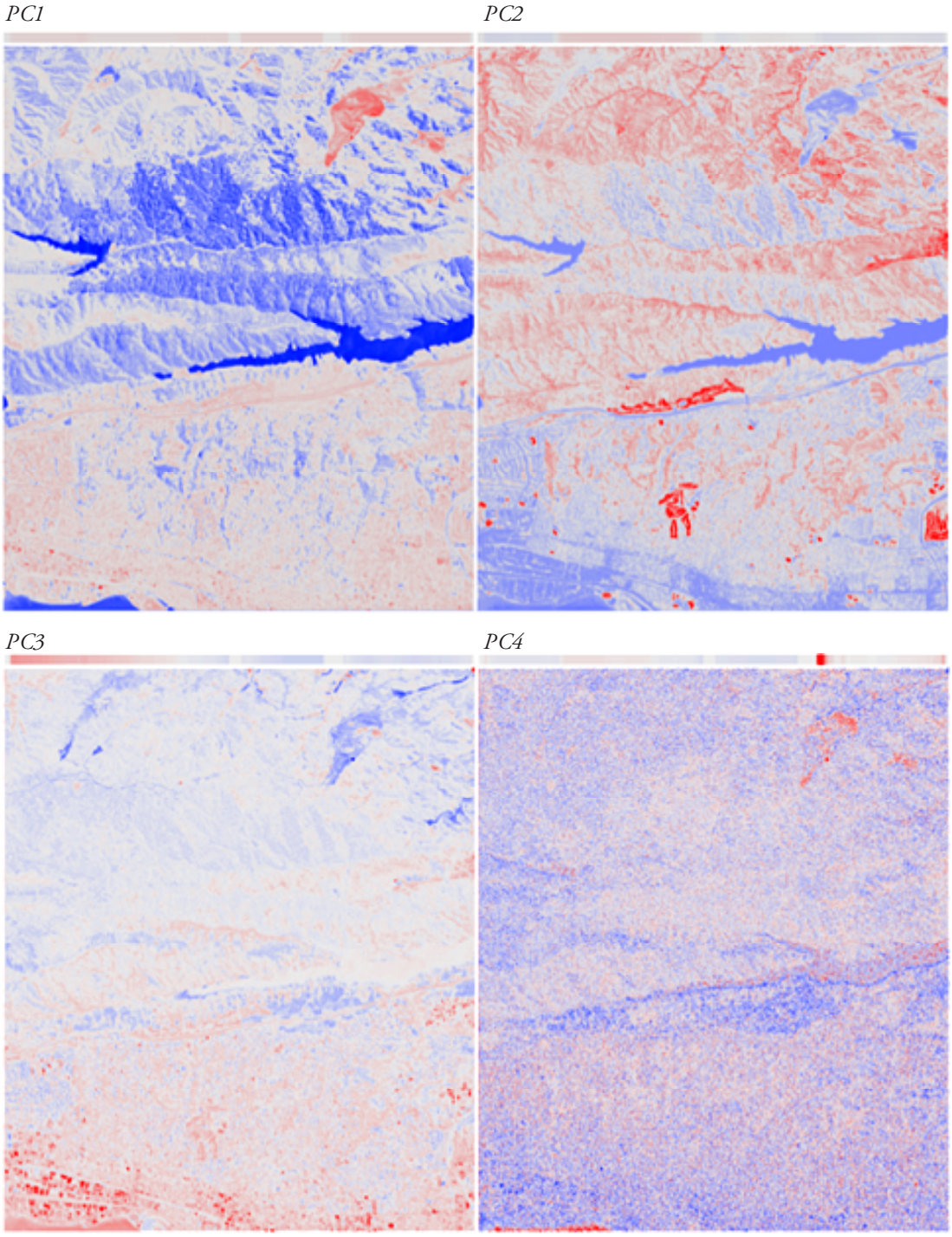
The third principal component (PC3 in *Figure 2*) has higher loadings on the very low end of the scale, the infrared. This is a heat image showing what is reflecting heat. There are buildings that seem to have hot roofs, and a warm surrounding area.

The fourth principal component (PC4 in *Figure 2*) is narrowly loaded at the high end, the blue end, and shows very unevenly-pixelated features that pop out all over the image. Features at the extreme bottom left are most prominent.

References

Google Maps xxxx
NASA Jet Propulsion Laboratory, California Institute of Technology,
AVIRIS aviris.jpl.nasa.gov/html/aviris.freedata
Wikipedia en.wikipedia.org/wiki/Hyperspectral_imaging

Figure 2 Images From First Four Principle Components



Splitting Columns with a Group Variable

Ann Lehman, JMP Division of SAS Institute

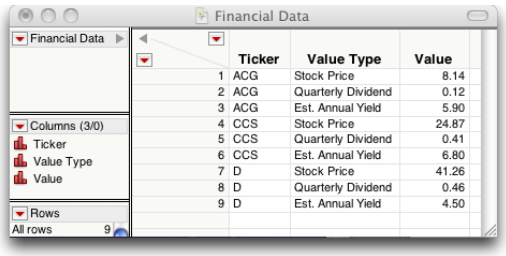
Unless you can find someone else to do data house cleaning for you, cleaning data and reorganizing the structure of information is often a necessary preliminary to doing analyses. Data rarely reach you neat, clean, and ready to go. The Tables menu in JMP has commands that can do data rearranging similar to that in the SAS system. The key is learning how to specify what you want in the JMP Tables menu launch windows.

One example of restructuring data is to use Tables > Split to split a single column into multiple columns. The Split launch window can cause confusion if you don't understand the mechanics of JMP's split operation. This article uses a simple table and shows what happens when you split columns in various situations. Like most Tables menu commands, Split always creates a new JMP table.

A Simple Split Example

Consider the table in *Figure* 1 (call it the source table). The data of interest is the Value column and the data is organized such that each company, identified by Ticker, has information in three rows. Further, each row provides a different type of information as shown in the Value Type column. Notice that this example table in is well organized. Each company has three rows and the values are in order by Ticker and Value Type within Ticker.

Figure 1 JMP Table with Stacked Data



	Ticker	Value Type	Value
1	ACG	Stock Price	8.14
2	ACG	Quarterly Dividend	0.12
3	ACG	Est. Annual Yield	5.90
4	CCS	Stock Price	24.87
5	CCS	Quarterly Dividend	0.41
6	CCS	Est. Annual Yield	6.80
7	D	Stock Price	41.26
8	D	Quarterly Dividend	0.46
9	D	Est. Annual Yield	4.50

Data tables like this one can be summarized or analyzed using the Summary command in the Tables menu, or by specifying a By variable in a platform launch. But it is often simpler to rearrange the table so that each row contains all the data for one company (or unit), showing one data column for each type of value. That is what the Split command does.

To do a simple split operation on this table, select Split from the Tables menu and complete the Split launch window as shown in *Figure 2*.

- Select Value Type in the select columns list on the launch window and click Split By.

  The values of the Split By variable become the column names in the new table.

- Select Value in the select columns list on the launch window and click Split Columns.

  The values of the Split variable are listed in multiple columns as defined by the Split By values.

- In this example, click the Keep All radio button because you want to see the ticker symbol in the data table to identify the rows.

Figure 2 Split Command Launch Window

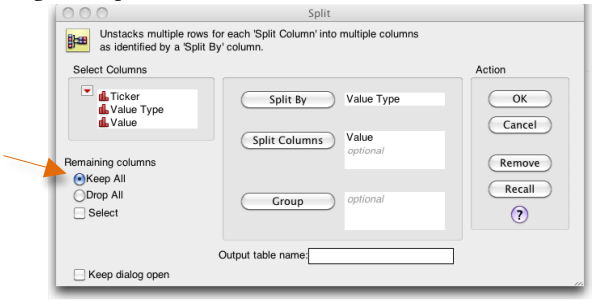
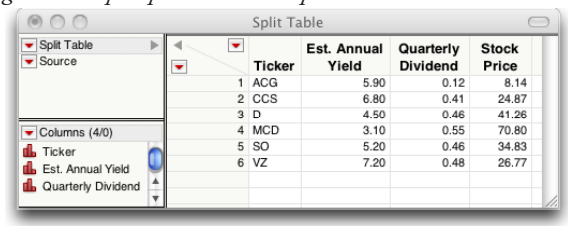


Figure 3 shows the table that results from the split operation. When the data are in this form it is easy to see how to summarize the numeric variables individually, or use Graph menu commands to show the relationship between them.

Figure 3 Simple Split Table Example



	Ticker	Est. Annual Yield	Quarterly Dividend	Stock Price
1	ACG	5.90	0.12	8.14
2	CCS	6.80	0.41	24.87
3	D	4.50	0.46	41.26
4	MCD	3.10	0.55	70.80
5	SO	5.20	0.46	34.83
6	VZ	7.20	0.48	26.77

When to Use a Group Variable

As noted, the financial data table is in order by Ticker and Value Type within Ticker, and all companies have three rows (a row for each type of data). However, there are situations that can cause a split operation to produce an incorrect table unless a Group variable is specified on the launch window. To see how splitting a table actually works, and how an incorrect table can result, examine the following subset (*Figure 4*) of the original example table in *Figure 1*.

Figure 4 Subset of Financial Data

	Ticker	Value Type	Value
1	ACG	Stock Price	8.14
2	ACG	Quarterly Dividen	0.12
3	ACG	Est. Annual Yield	5.90
4	CCS	Quarterly Dividen	0.41
5	D	Quarterly Dividen	0.46
6	D	Est. Annual Yield	4.50

If you split the Value column for this table, you want the new table to have one row for each company.

- Company ACG has values for all three value types.
- Company CCS has a value for Quarterly Dividend and missing values for Stock Price and Est. Yearly Yield.
- Company D has values for Quarterly Dividend and Est. Annual Yield but a missing value for Stock Price.

Splitting without a Group variable produces the incorrect table in *Figure 5* and illustrates the mechanics of the Split command.

Figure 5 An Incorrect Split

	Ticker	Est. Annual Yield	Quarterly Dividend	Stock Price
1	ACG	5.90	0.12	8.14
2	D	4.50	0.41	•
3	D	•	0.46	•

Here is what happens. Split first creates the column(s) you specified to keep (Ticker).

	Ticker	Est. Annual Yield	Quarterly Dividend	Stock Price

Then, new columns are created for each value of the Split By variable (Value Type), and arranged in alphabetical order (Est. Annual Yield, Quarterly Dividend, and Stock Price).

The first row in the new table forms correctly because the AGS company has a row in the original table for each new column in the split table, and the rows are grouped together.

Let's step through the construction of the second and third rows in the split results (see *Figure 6*).

1) The Split process scans the source table for the next value of Est. Annual Yield and finds 4.50 for the company with ticker symbol D (see *Figure 6*). That value is placed in the second row. Note that company CCS had no value for Est. Annual Yield.
2) The next value for Quarterly Dividend is 0.41, which is placed the second row even though it is for company CCS.
3) No further values are found for Stock Price so the second row has a missing value for that column.

The problem is clear. After the ticker value D was determined for the second row, it was not checked again even though the next value for quarterly dividend was for the CCS company.

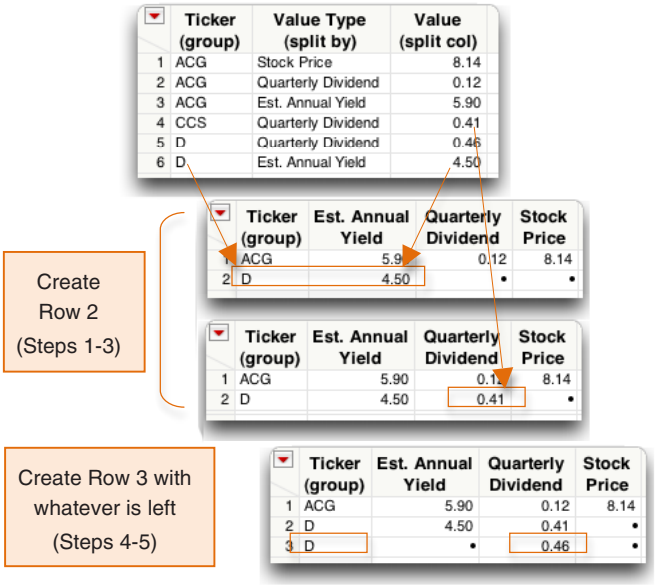
The third row unfolds similarly.

4) The split process finds no further values for Est. Annual Yield and records a missing value for that column in the third row.
5) The last value for Quarterly Dividend is found for the ticker symbol D, so those two values are recorded in the third row

To summarize, the split process uses the order of the new columns in the split table to scan for values in the source table, and places values as they are found. If no grouping value is specified, this is the logical way for the new rows to be formed. However, if there is a grouping of values in the source table, then incorrect results can occur if

- a group has missing members
- groups of values are out of order.

Figure 6 The Mechanics of Splitting a Column



The Ticker variable in the Financial data table and the example subset identifies groups of values. When there is a grouping variable, be sure to specify it in the launch window. The Group variable tells the split process to place values in a row only when the values of their grouping variable are the same.

Compare the correct result in *Figure 7* with the incorrect result shown previously in *Figure 5*. The Group variable (Ticker) causes the split process to check each the value of the group variable before writing a value to the new table.

Figure 7 A Group Variable Gives Correct Results

	Ticker (group)	Est. Annual Yield	Quarterly Dividend	Stock Price
1	ACG	5.90	0.12	8.14
2	CCS	•	0.41	•
3	D	4.50	0.46	•

In this small data table example, the mechanics and results are easy to see. But in tables with thousands (or millions) of rows, an incorrect split might occur without warning. A safe approach is to always use a Group variable when there is one in the data table that groups the values you want to split.

Food for thought—be aware that the Split command recognizes both multiple Split Columns and multiple Split By columns.

JMP Add-ins

Melanie Drake, JMP Division, SAS Institute

Often someone writes a useful JSL script to perform a task and wants to send that script to co-workers so they can use the script, too. However, many JMP users are not scripters and might be uncomfortable with opening and running scripts.

JMP 9 introduces the add-in, which is a single executable file that installs itself into JMP. You can take any script you've written and turn it into an add-in without doing anything special to the script itself.

Steps to write an add-in

Here are the basic steps to turning a script into an add-in.

1. Create a folder and put your script, along with any support files that are required (for example, data tables or icons).

   Tip: An add-in can contain multiple JSL files, icons, images, data tables, journals, text files, html files, PDF files, and so on. If you can find a way to use something in JMP, you can include it as part of the add-in.

2. In this new folder, create a file named `addin.def` that contains these three lines:

   ```
   id="com.yourcompany.youraddin"
   name="friendly name"
   autoload=1
   ```

   Tip: The `id` must be a unique identifier for your add-in, different from any other ID anyone else's add-in might use. We recommend using reverse DNS. The `name` is a display name. It does not need to be unique, and it can provide a friendly description of what your add-in does. Setting `autoload` to 1 (true) ensures that once your add-in is installed, it loads each time JMP starts.

3. Register your add-in. This is the same as installing it. You can run this script to register your add-in while you're developing it:

   ```
   registerAddin(
   "unique ID" // for example,
   com.yourcompany.youraddin
   "full folder pathname" // your add-in
   folder
   );
   ```

   Tip: Once the add-in is registered with JMP, you can reference its folder using the `$ADDIN_HOME(uniqueID)` path variable. For example:

   ```
   include("$ADDIN_HOME(com.yourcompany.
   youraddin)\addin_script.jsl");
   ```

4. Create a custom menu entry to run your script. When JMP users install your add-in, they can run it from the menu instead of having to run a script.

   Tip: When you create the menu entry, select the Use add-in Home Folder check box, and then select your add-in from that menu. Doing so saves the custom menu (addin.jmpcust) into your add-in folder.

5. Navigate into your add-in folder, select all the files in it, and create a ZIP archive.

   Tip: Don't select the folder itself to create the ZIP archive. Doing so creates an add-in that cannot be recognized by JMP. Instead, be sure to select the files within the folder to create the ZIP archive.

6. Change the extension of the archive from `.zip` to `.jmpaddin`. You can also change the filename, if you decide your original folder name is not what you want to use.

Your add-in is now ready to go. Anyone who uses JMP 9 or later can install it by double-clicking the file, by dragging and dropping it into JMP, or by opening it from the File > Open menu.

To see the contents of a JMP add-in folder, copy the file called Simple Calculator.jmpaddin from the Sample Scripts folder installed with JMP 9. Change the .jmpaddin extension to .zip. Unzip the folder to see the contents of the add-in, shown here. Note that the add-in has the script file as well as other support files.

Once you install this add-in, an Add-Ins menu shows on the JMP main menu that lists lists this item.

Example Add-ins

The JMP staff has written a variety of interesting add-ins that are free to download.

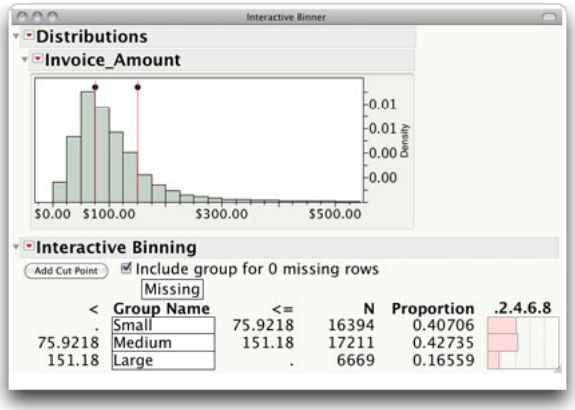
Here is a summary of featured add-ins. Some JMP users might recognize long-awaited functionality.

- **Add-in Builder**
  This JMP add-in helps you build a JMP add-in installation file by collecting information about the add-in and using an existing JSL file. You simply complete

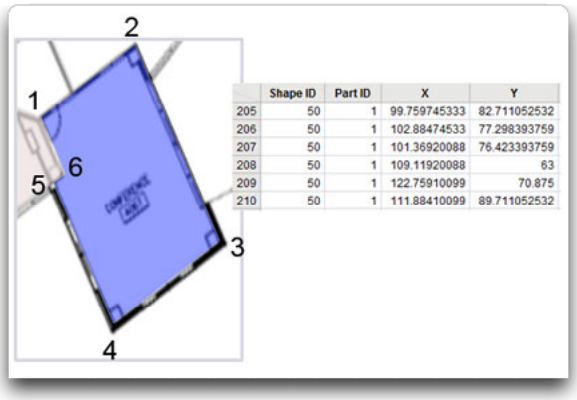
JMP Add-Ins (continued)

a dialog that finds the files you specify and then wraps them into an add-in for you.

- **Interactive Binning**
  Create discrete groups from continuous data using this add-in. A histogram shows the distribution of values for your variable. Add a cut point to the histogram and drag it to set the cut point value. After adding all your cut points, you save your groups to a new column in your data table.



- **Custom Map Creator**
  With the Custom Map Creator, you can easily create custom shapes used by Graph Builder. This add-in creates two tables to define the shapes: an XY table and a Name table.



- **Stratified Split**
  This add-in creates or updates a column that splits the rows in a table according to specified proportions within each level of a selected column. This kind of column splitting can be useful when building predictive models and the number of rows for each level of the selected column is not roughly equal.

Check out the full list of new add-ins or submit one of your own to the JMP File Exchange.

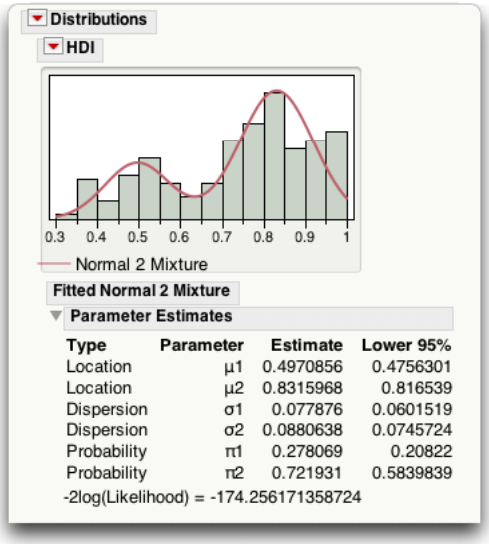
New in JMP 9: Normal Mixtures Fit in the Distribution Platform

The Normal Mixtures option on the Distribution platform fits a mixture of normal distributions to a single variable. This flexible distribution is especially useful for fitting multi-modal data.

Fit a mixture of two or three normal distributions by selecting the Normal 2 Mixture or Normal 3 Mixture options. Alternatively, you can fit a mixture of *k* normal distributions by selecting the Other option. A separate mean, standard deviation, proportion of the whole, along with upper and lower 95% confidence intervals is estimated for each normal distribution group.

The example below shows the distribution of HDI (Human Development Index) for the world populations from in the World Demographics data table found in the Sample Data library installed with JMP 9. This is clearly a bi-modal distribtion. Using the Normal Mixtures > Normal 2 Mixture fitting option gives the curve shown in *Figure 1*, with its table of parameter estimates for the two normal distributions.

Figure 1 Normal Mixtures Fit of Two Normal Distributions



For this variable, the normal Mixtures fit indicates two populations, which offers a clue that begs for further exploration.

Note: The World Demographics sample data table is new in JMP 9. It contains the names of 249 countries with the most current demographic, financial, and health data available. The country names are compatible with JMP's build-in map data so variables can be easily displayed geographically on the Graph Builder platform. Exploring the world is fun and informative using the World Demographics sample data table.

About JMPer Cable
A Technical Publication for JMP® Users
Issue 27 Winter 2011

Editor:
Ann Lehman

Winter 2011 Issue Contributors:
Mark Bailey
Melanie Drake
Win LeDinh
Ann Lehman
Paul Nelson
José G. Ramírez
John Sall

Subscribe Today

If you enjoy reading JMPer Cable, add yourself to our subscription list. You can receive the printed newsletter in the mail or read the PDF version online. We will soon be moving to subscription-only delivery, which means you can't count on us sending it to you unless you subscribe. Don't miss a single issue. Subscribe today.

You can also find JMPer Cable current and back issues, along with their data tables and scripts, online. It's all at

www.jmp.com/jmpercable

For additional user resources, go to

www.jmp.com/community

To order additional licenses of JMP, to learn about the most recent release, or to inquire about content of this JMPer Cable issue, call 1-877-594-6567