



### Inside This Newsletter

Using Confidence Intervals Other Than 95% p. 4

Using Cell Plots to Visualize Group Characteristics in Large Samples p. 5

Hierarchical Clustering of Large Data Tables p. 7

Many Formulas, Several Tables: A JSL Shortcut to Copying p. 9

Meet the Trainer: An Interview p. 11

### Breaking News

The JMP User Conference, June 7-8, 2005, in Cary, NC, will include a keynote address by J. Stuart Hunter, professor emeritus at Princeton University's School of Engineering and Applied Sciences.

Training classes are June 9-10. Additionally, we will hold a free pre-conference scripting panel discussion (June 6) and post-conference Executive Overview for Six Sigma (June 9).

When registering, mention the *JMP Per Cable* to receive the Early Bird \$200 discount at any time.

For more information, see <http://support.sas.com/training/jmp>.

## Partition Q & A

*John Sall, Executive Vice President, SAS Institute*

In version 5, JMP introduced the partition platform. Partitions, also called decision trees, meaningfully divide your data using the  $x$  values to make  $y$  values that are close to each other. This provides clues on how the  $x$ 's could be affecting the  $y$ 's.

The following FAQ gives an inside look at the platform and highlights valuable and interesting features.

*Why call it partition instead of decision tree?*

Though the partition platform produces a tree of conditions, we wanted people to be aware of the essential feature of the method. The pioneers of the Chi-squared Automatic Interaction Detector (CHAID) branch of the literature, Kass and Hawkins<sup>1</sup>, call it recursive partitioning, which we shortened to partition. The term partition gives more of the feel of a detective using clues to narrow a search for discovery, where as the term *decision tree* is more about making a black-box set of rules.

*How do you see partition's role in Six Sigma programs?*

Six Sigma projects are often fishing expeditions to find clues as to what affects productivity, cost, quality, etc. Partition is a natural for this, scanning

large numbers of candidate terms, and picking out the interesting ones. Of course, a clue in happenstance data is just a clue—it is not proof that there is a real effect. Designed experiments are still the gold standard for learning. But data is often free, and you could gain considerable insight by looking at it systematically. Most data are not looked at systematically because people don't have good, powerful tools to make the exploration easy. Partition makes it easy. Some Six Sigma practices call this kind of investigation of data a Multi-Vari study.

*Does the addition of the partition platform mean that JMP is now a data mining tool?*

The term data mining does not fit JMP very well because data mining has come to mean the automatic discovery of patterns in huge amounts of data. By design, JMP's partition is neither automatic nor suitable for huge amounts of data.

*Why isn't the partitioning process automatic?*

Partition is not automatic because JMP's primary charter is to be an interactive discovery tool. So splits are interactive; users make discoveries as



they split the data further. Other products produce splits until a stopping criterion is reached. With JMP, you click a button for each split and decide when to quit. You make discoveries by splitting data further. Most automated data mining tools are designed more for prediction—you don't care what variables are used for the splits as long as they predict well.

The big, automated tools, like SAS Enterprise Miner (EM), create rules that are used for predicting such things as credit defaults, direct mail response rates, buying patterns, patterns for detecting fraud, and so forth. The end product can be treated as a black box that just works.

Interactive tools, like JMP's partition,

are used to find out such things as factors associated with defects in a manufacturing line, or what combination of factors tends to lead to good production yields. The end product is the discovery that may lead to improving the process. This is detective work.

*Why won't partition handle huge amounts of data?*

First, JMP requires all the data to fit in memory, and JMP's great speed is mostly credited to having everything in memory. Second, JMP is interactive. If it takes minutes or hours to do each split, then you might want to use a more automated tool, like Enterprise Miner. Third, JMP is graphical. It shows you all the points, but millions

of points appear as a black cloud and take seconds to display. One of our largest examples has a quarter million rows and takes a few seconds to do each split because JMP sorts the data on each variable for each split. JMP fits the data in memory and completes millions of calculations. Although memory is getting cheaper and chips faster, there are still practical limits.

*Can partition produce a prediction formula?*

Yes. You can make a formula to predict a continuous response, or fit probabilities to categorical responses.

*What most distinguishes JMP's partition platform from similar procedures on other packages'?*

The graph, of course. The graph

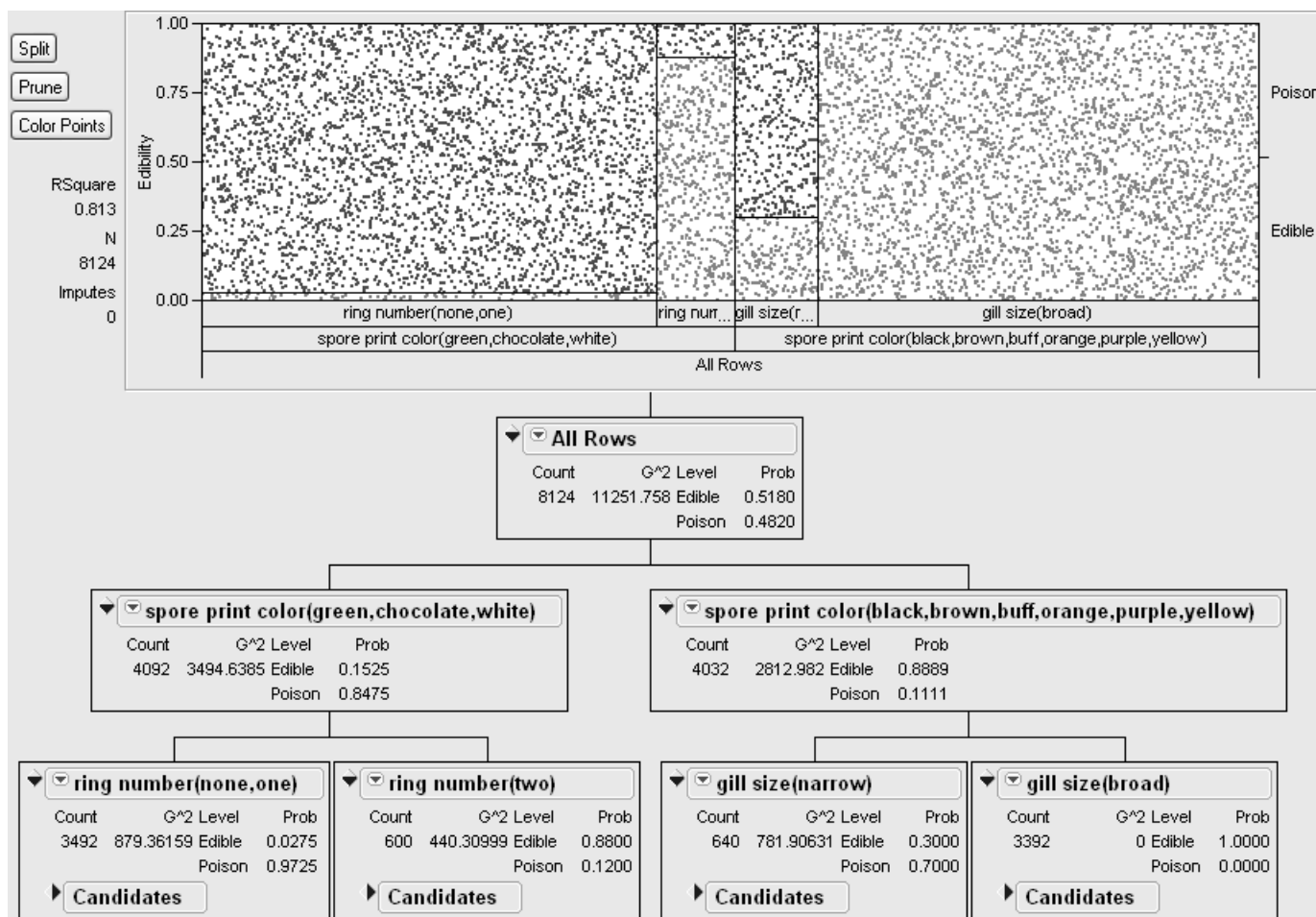


Figure 1: Partition showing the relationship of mushroom characteristics on the edibility of mushrooms (poisonous or not).

shown in Figure 1 illustrates the relationship of mushroom characteristics on edibility of mushrooms (poisonous or not). With one look you see both the current structure of the partition and its effect on the responses. In this example you can see that almost 85% of mushrooms with spore print color of green, etc. are poisonous. Within that single group, 97% of mushrooms with zero or one ring are poisonous<sup>2</sup>.

The directions of the splits are ordered so that higher values tend to sort right. The points are spaced out in random positions in the  $x$  direction within the cell. They are random in the  $y$  direction for categorical responses within the cell. Furthermore, you can color the points by the response, making it easy to visualize relationships.

As you split, the axis becomes more complicated. When the splits get small, there is not enough room for the labels, but you can look at the corresponding point in the tree for details. As you split, consider dragging the bottom right corner of the graph to make it bigger so that you can see more of the labels. No other product has this graph.

*What else is unique about partition?*

The splitting criteria. We have solved the problem of comparing terms that have varying number of category levels. No one else has addressed this problem as well. The statistic we use is standard: the  $F$  statistic for continuous responses (equivalent to Student's  $t$ ), or the likelihood ratio Chi-squared for categorical responses (equivalent to entropy). But these statistics need a  $p$ -value to order the candidates. If you

have two category levels to split, then the  $p$ -value is the standard unadjusted significance probability. However, as you have more levels, there are many more ways to split, and selection bias becomes a serious problem. This must be compensated for when you compare splitting terms. Most data mining packages use a simple Bonferonni adjustment. We did extensive Monte Carlo studies on the unadjusted and Bonferroni-adjusted  $p$ -values<sup>3</sup>. The studies show that, as the number of possible splits increases, the unadjusted  $p$ -value is much too liberal, and the Bonferonni adjustment is much too conservative. In other words, unadjusted  $p$ -values will over-choose the terms with many levels, and Bonferonni  $p$ -values will over-choose terms with few levels.

Chris Gotwalt, a member of the JMP development staff, derived the exact distribution for splits, but found the computing resources to do these were prohibitively expensive. He obtained the empirical Monte Carlo distributions of the statistics under the null hypothesis for a large variety of conditions and built a model to obtain  $p$ -values. The result is a very well-adjusted  $p$ -value, which is for comparative purposes only. It is not for hypothesis testing because it still has other selection biases from modeling itself. However, it is clean of the number-of-levels issue.

*Partition doesn't show these p-values in the report. Why not?*

Three reasons. First, the tradition in data mining is to report log-worth, which is  $-\log_{10}(p\text{-value})$ . Since the  $p$ -value shouldn't be used for stopping

rules or inferences, due to selection biases, we were not comfortable in labeling it as a  $p$ -value. Second, the log-worth scale is appropriate because bigger-is-better. Third, with large sample sizes, the  $p$ -values themselves are often so small that they are unrepresentable in IEEE machine form, *i.e.* a log-worth above 303 is a  $p$ -value smaller than  $10^{-303}$ , the limit of representability for the computer. By keeping things on the log-scale, the  $p$ -values are reasonable numbers that do not take a special effort to represent. It is not hard to translate; for example, a logworth of 2 is the  $p$ -value  $10^{-2}$  or 0.01, a logworth of 3 is the  $p$ -value  $10^{-3}$  or 0.001.

*Are any features missing that are common to other decision tree products?*

As indicated earlier, JMP has no stopping rules. Some tree products consider multi-way splits rather than just binary splits. Some offer different splitting criteria, such as the Gini coefficient. Some offer more complex missing value handling using surrogates. Some offer to work on random subsets of the data to make it faster. There is a rich literature of other features that have been tried. If you want to build large prediction models for applications such as credit scoring or fraud detection, then these features will be valuable, and you will want a richer tool such as SAS Enterprise Miner.

*Any performance trouble spots?*

If you have more than two categories for a response, and more than twenty categories for an X variable, then the number of possible splits that it checks is a large number. Fifty levels requires

more than a lifetime to calculate the candidate statistics, and a few more take more time than the current life of the universe. JMP warns you when this situation occurs and give you a chance to exit. In a future release, we will provide heuristic search methods for this case. In practice, this doesn't happen much. I haven't heard a single user request to ameliorate this situation, though I'm sure it does happen. Most categorical responses are two-levels, and when there are more, there are usually fewer than 20 levels in the  $x$  variables.

*What features were added in response to beta testing?*

Locking check boxes. The typical situation is that you pick a  $y$  and many many  $x$ 's to consider. Often, an  $x$ -variable that is the best candidate is one you don't want to consider. Either it has too many categories to be meaningful, is too obvious, can be a function of the response, or can be so expensive to change that it shouldn't be used in the results. Being able to lock columns allows you to ignore that variable gracefully and proceed, without having to restart.

*What should we expect for partition in version 6?*

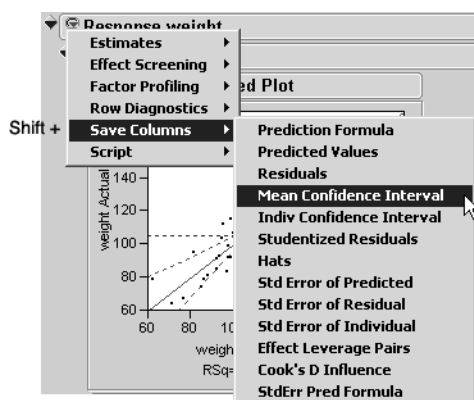
Partition in version 6 changes some of

the default options, and adds more graphics, including ROC and Lift curves.

### References

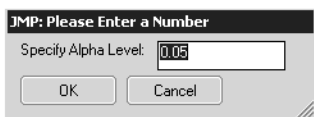
1. Hawkins, D.M. and Kass, G.V. (1982), "Automatic Interaction Detection," in Hawkins, D.M., ed., *Topics in Applied Multivariate Analysis*, 267-302, Cambridge Univ Press: Cambridge.
2. <http://www.ics.uci.edu/~mlearn/MLSummary.html>
3. <http://jmp.com/product/monte-carlocal.pdf>

## Using Confidence Intervals Other Than 95%

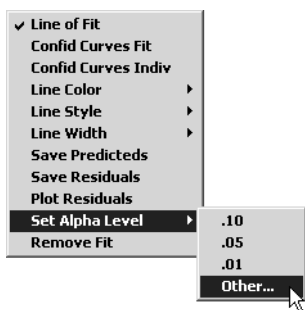


In the Fit Model platform, confidence interval commands use  $\alpha = 0.05$  by default.

To have JMP calculate confidence intervals with a different  $\alpha$ -level, press the Shift key while selecting **Save Columns > Mean Confidence Interval** or **Indiv Confidence Interval** under the red triangle menu.



Then enter the  $\alpha$ -level you would like to use.



In platforms other than Fit Model, click the red triangle icon and specify a different  $\alpha$ -level using the **Other...** command.

## Using Cell Plots to Visualize Group Characteristics in Large Samples

Marissa J. Langford, SAS Statistical Training and Technical Services

Microarrays, slides spotted with cDNA, are used to monitor gene expression levels on thousands of genes simultaneously. Researchers often use this method to compare gene expression levels under varying conditions. Essentially, researchers can identify which genes turn on and which genes turn off as conditions change. For example, the gene expression from a diseased tissue can be compared to the gene expression of a non-diseased tissue. Ultimately, researchers hope to turn this information into targeted drug treatments.

Because of the massive amounts of data often generated during a gene expression study, data analysis can be challenging. JMP provides several tools to help researchers visually assess differential expression levels between experimental groups. The cell plot is one of these tools.

### Case Study

Dr. Greg Gibson, an associate professor in the Department of Genetics at North Carolina State University, conducted a study to determine the effect of sex, age, and genotype on the gene expression levels of adult fruit flies. He used the SAS Microarray Solution to analyze the data collected in the experiment. He calculated standardized least squares means of expression intensity for each treatment group. We can use a subset

Significant Genes	Spot	F1 ORE	F6 ORE	F1 SAM	F6 SAM	M1 ORE	M6 ORE	M1 SAM	M6 SAM	
	1	4	-0.14846	-1.08534	0.771331	1.149675	-1.58351	-0.56041	0.709891	0.746832
	2	8	-0.34224	-0.94421	-1.04781	-1.19053	0.44374	0.824888	0.964353	1.291805
	3	22	-0.26122	0.685535	-1.23612	-1.70177	0.220057	0.503287	0.636624	1.1536
	4	28	0.195195	-0.3503	0.552027	1.867534	-0.36687	-1.22767	0.433358	-1.10327
	5	40	-0.55024	-0.95481	-1.10311	-0.97307	1.138251	0.542526	1.286404	0.614055
	6	44	0.089037	-1.12133	2.083569	-0.16201	-0.97527	-0.4963	0.281061	0.301235
	7	56	-0.05581	-0.36987	-1.22823	-1.57683	0.946229	0.628712	1.14703	0.508767
	8	57	-0.71801	1.043905	0.135786	0.836546	-0.03549	1.06695	-1.80637	-0.52332
	9	73	1.043452	0.411006	1.014545	1.063101	-1.36751	-0.94007	-0.81132	-0.4132
	10	75	-0.7042	-1.58312	0.329119	1.475593	-0.75496	-0.19315	0.513622	0.917093
	11	76	1.351557	0.536441	-0.84179	1.52425	-0.38798	-0.94295	-0.77088	-0.46864
	12	78	-0.83556	-0.86946	-1.05585	-0.95542	0.940132	0.755366	1.115301	0.90549
	13	82	0.46084	-0.05443	-1.73797	-1.17058	1.286204	0.795844	0.115336	0.304757
	14	114	-0.93548	-0.86901	-0.85894	-1.04798	1.082835	0.739261	1.066815	0.822505

Figure 2: Partial listing of the fruit fly gene expression data.

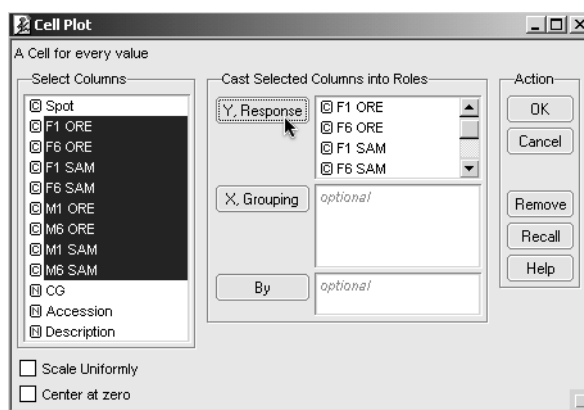


Figure 3: Starting a cell plot analysis.

of his data to illustrate creating and interpreting a cell plot.

Figure 2 shows a partial listing of the standardized least squares means for the fruit fly gene expression data. The variable names identify the group characteristics, male or female (F and M), age (1 and 6), and genotype (ORE and SAM).

To see the cell plot for this data:

1. Open the data table Significant Genes.jmp at <http://www.jmp.com/news/jmpcable>.

2. Choose **Graph > Cell Plot**.
3. Select the eight columns: F1 ORE through M6 SAM, as shown in Figure 3.
4. Click the **Y, Response** button, then click **OK**.
5. Click the red triangle on the Cell Plot title bar and select **Legend**. The legend maps the values of each variable to the color range used in the cell plot, as shown in Figure 4.

*(continued on page 6)*

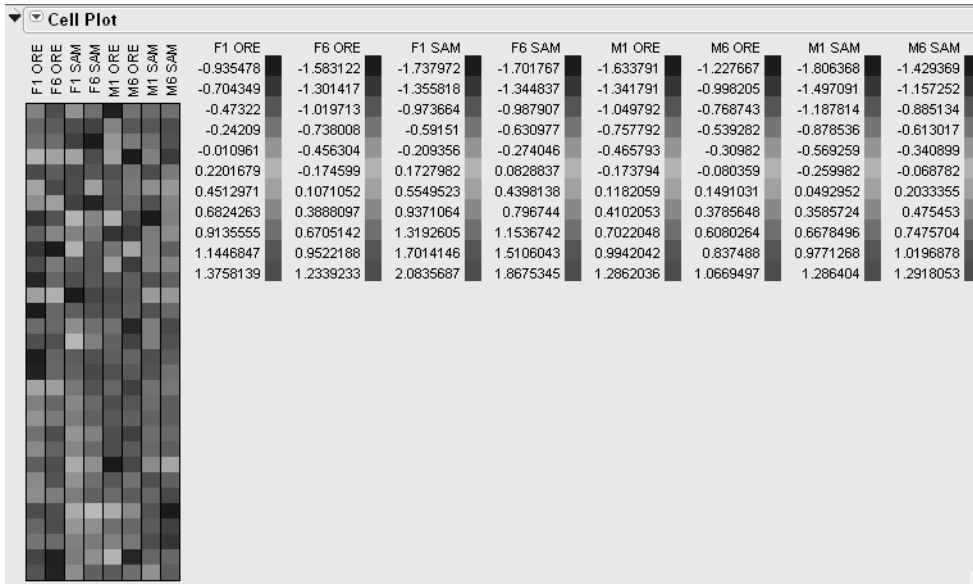


Figure 4: Cell plot for microarray analysis.

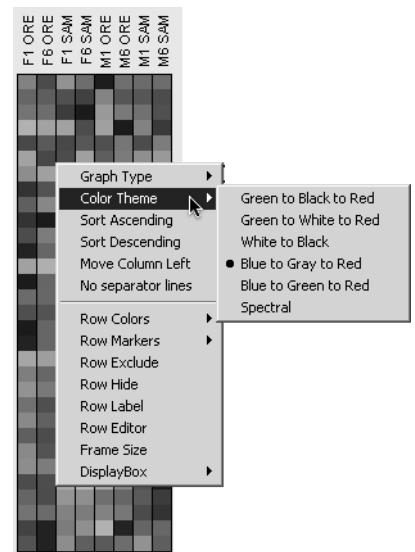


Figure 5: Options available for a cell plot.

## How to “Read” the Cell Plot

Each row in the cell plot corresponds to a row in the data table. Each individual square in a row represents the relative value of the standardized least squares mean in that column. The legend shows that the hues move from a dark blue for the lowest levels in each column to bright red for the highest value in each column.

The color-coding visually compares gene expression levels between groups. For example, the genes in the lower half of the graph have a high standardized least squares mean for

the females (varying shades of red), regardless of genotype and age, while the males have a low standardized least squares mean (varying shades of blue).

Right-click the plot to see the options shown in Figure 5. You can change the color theme, display different types of cell plots, or sort by the column you select to better illustrate patterns.

## Conclusion

The use of cell plots is not limited to Microarray analysis. They can be used any time a JMP user wants to visually compare column values across a row

in a data table, and can be especially useful when there are a large number of characteristics.

## References

*The Contributions of sex, genotype, and age to transcriptional variance in Drosophila melanogaster.* Wei Jin, Rebecca M. Riley, Russell D. Wolfinger, Kevin P. White, Gisele Passador-Gurgel, and Greg Gibson. Nature Genetics, volume 29, December 2001.

*Using SAS Microarray Solution,* Course Notes. SAS Institute, August 2003.

## This Book is Hot Off the Press! *JMP for Basic Univariate and Multivariate Statistics: A Step-by-Step Guide*

Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward J. Stepanski show you how to manage JMP data and perform statistical analyses commonly used in research in the social sciences fields. Topics include: screening data for errors and selecting subsets with the JMP Distribution platform, computing the coefficient alpha reliability index (Cronbach's alpha) for a multiple-item scale, performing bivariate analyses, performing a one-way analysis of variance (ANOVA), performing a multiple regression, and performing a one-way multivariate analysis of variance (MANOVA). Details can be found at <http://www.sas.com/apps/pubscat/bookdetails.jsp?catid=1&pc=59814>.

## Hierarchical Clustering of Large Data Tables

Mike Stockstill, JMP Technical Support

Sometimes hierarchical clustering might not be practical for a large data set due to the huge amount of memory required to store the distance matrix used in finding the clusters. However,  $k$ -means clustering is able to handle a very large data set easily. One solution to the problem of finding hierarchical clusters in a large data set involves three steps:

1. Use  $k$ -means to form preliminary clusters and save the results in a data table.
2. Use hierarchical clustering to group these preliminary results.
3. Join the hierarchical results with the original data to form a final solution.

### Preliminary K-Means Cluster

This example uses the iris data published by Fisher (1936). There are 150 rows to be clustered based on the values of four columns: sepal length, sepal width, petal length and petal width. This iris data table is small and used only for illustration.

The columns are measured on similar scales, so no standardization is used for the clustering. There are three known types of plants: Setosa, Virginica, and Versicolor. The goal is to see if the cluster method described above accurately identifies which type corresponds to each row. Figure 6 shows an excerpt of the data.

	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

Figure 6: Example of iris data from Iris.jmp.

Ten preliminary clusters will be chosen based on  $k$ -means clustering. It is a good idea to select a preliminary number of clusters much larger than you believe will be the number of final clusters (in this case three). Try different numbers of preliminary clusters to see if the end results are fairly consistent:

1. Open the data table Iris.jmp from the Sample Data folder that was installed when you installed JMP.
2. Choose **Analyze > Multivariate Methods > Cluster**.
3. Select Sepal length, Sepal width, Petal length, and Petal width and click **Y, Columns** to assign them as the analysis variables.
4. Choose **KMeans** under **Options**.
5. Type 10 in the **Number of Clusters** text box.
6. Un-check **Standardize Data**.

Your completed dialog should look like the one in Figure 7.

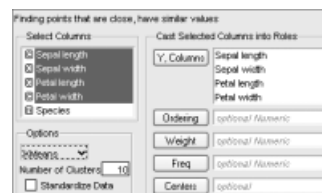


Figure 7: Dialog for K-means clustering.

7. Click **OK** on the completed window. The Iterative Clustering control panel appears (Figure 8).

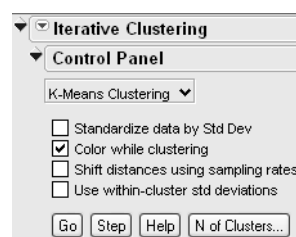


Figure 8: Iterative clustering control panel.

8. Click **Go**. The control panel shows the values updating during the iterative search for clusters.
9. When the iterative clustering finishes, click the red triangle on the Iterative Clustering title bar and select **Save Clusters**.

The data table now has two new columns. One called Cluster contains the preliminary cluster number of the cluster each observation is assigned. A second new column called Distance lists the multivariate distance of each observation from the center of its assigned cluster. The Distance column is not used in this example.

10. Change the name of the Cluster column to Preclus.

Next, save the cluster means found in the clustering results into a JMP table:

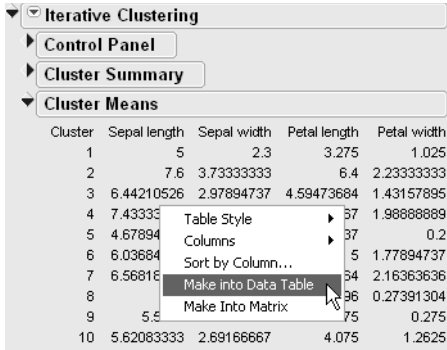


Figure 9: Save cluster means from k-means results to a JMP table.

	Preclus	Sepal length	Sepal width	Petal length	Petal width
1	1	5	2.3	3.275	1.025
2	2	7.6	3.73333333	6.4	2.23333333
3	3	6.44210526	2.97894737	4.59473684	1.43157895
4	4	7.43333333	2.92222222	6.26666667	1.98888889
5	5	4.67894737	3.08421053	1.37894737	0.2
6	6	6.03684211	2.70526316	5	1.77894737
7	7	6.56818182	3.08636364	5.53636364	2.16363636
8	8	5.1	3.51304348	1.52608696	0.27391304
9	9	5.5125	4	1.475	0.275
10	10	5.62083333	2.69166667	4.075	1.2625

Figure 10: The Means data table.

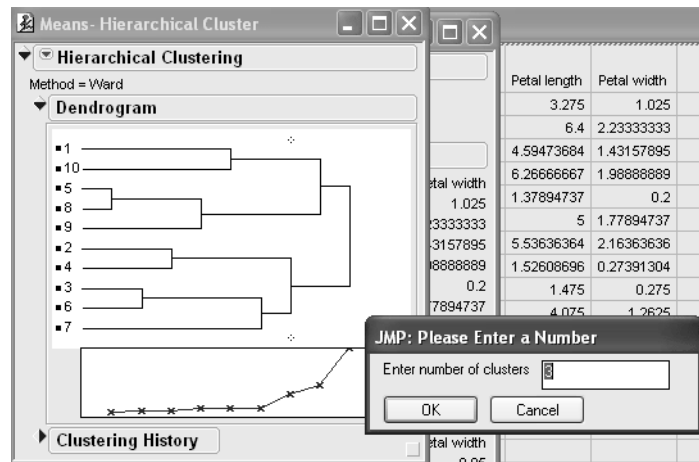


Figure 11: Changing the number of clusters.

Cluster	Sepal length	Sepal width	Petal length	Petal width	Species	Preclus	Distance
1	4.9	2.4	3.3	1.0	versicolor	1	0.1457738
2	5.0	2.0	3.5	1.0	versicolor	1	0.37583241
3	5.0	2.3	3.3	1.0	versicolor	1	0.03535534
4	5.1	2.5	3.0	1.1	versicolor	1	0.36228442
5	7.2	3.6	6.1	2.5	virginica	2	0.58214164
6	7.7	3.8	6.7	2.2	virginica	2	0.32489314
7	7.9	3.8	6.4	2.0	virginica	2	0.38586123
8	7.0	3.2	4.7	1.4	versicolor	3	0.61007243
9	6.4	3.2	4.5	1.5	versicolor	3	0.25356188

Figure 12: The Combined data table.

1. Scroll to the Cluster Means area of the  $k$ -means results.
  2. Right-click anywhere in the Cluster Means table to reveal the options, as shown in Figure 9.
  3. Choose **Make into Data Table**.
  4. Rename the table Means by choosing **Window > Set Title**. The table contains the mean values of the 10 preliminary clusters (Figure 10).
  5. Change the name of the Cluster column to Preclus.
- Now find hierarchical clusters for this new table.
1. Ensure that the Means data table is the active table.
  2. Choose **Analyze > Multivariate Methods > Cluster**.
  3. Select Preclus and click **Label**.
  4. Use the four sepal and petal variables as the analysis variables.
  5. This time, use the default Ward's hierarchical method.
  6. Un-check **Standardize Data** and click **OK**.
  7. Click the red triangle on the Hierarchical Clustering title bar and select **Number of Clusters**.
  8. Type 3 in the **Number of Clusters** text box, as shown in Figure 11.
  9. Again click the red triangle on the Hierarchical Clustering title bar and select **Save Clusters**. This saves the cluster number from this hierarchical clustering analysis to the Means data table.
- The next step is to join this hierarchical clustering of the preliminary  $k$ -means data table (the Means table) to the original Iris table.
1. Select **Tables > Join**.
  2. Highlight Iris in the With box and choose **By Matching Columns**.
  3. Select Preclus from both the Means and Iris tables.
  4. Click **Match**, then click **Done**.
  5. Click the **Select Columns** button.
  6. Select only Cluster from the Means column name list and click **Add**. If you select the Sepal and Petal variables, they overwrite those in the Iris table.
  7. From the Iris column name list, select everything and click **Add**. Then click **Done**.
  8. In the Output Table text box, type Combined to name the new table.
  9. Click **Join** to see the combined table (Figure 12).
- The Cluster column identifies the cluster assignment for each row. The column Preclus is no longer needed. Now the original table has been clustered.
- For this example, the true species

values are known, so it is interesting to see how well the cluster analysis placed each observation. From Combined:

1. Select **Analyze > Fit Y by X**.
2. Choose Species as *y* and Cluster as *x*.
3. Click **OK**.

When the Contingency Analysis window appears, scroll down to the Contingency Table (Figure 13).

		Species			
Count	setosa	versicolor	virginica		
Total %					
Col %					
Row %					
1	0	27	1	28	
	0.00	18.00	0.67	18.67	
	0.00	54.00	2.00		
	0.00	96.43	3.57		
2	0	23	49	72	
	0.00	15.33	32.67	48.00	
	0.00	46.00	98.00		
	0.00	31.94	68.06		
3	50	0	0	50	
	33.33	0.00	0.00	33.33	
	100.00	0.00	0.00		
	100.00	0.00	0.00		
	50	50	50	150	
	33.33	33.33	33.33		

Figure 13: Contingency analysis of Species by Cluster

A perfect clustering would show 50 in each of the three columns and each of the three rows indicating that all of same species were in the same cluster. This is the case for Setosa—all 50 cases are in cluster 3. For Virginica, 49 cases were in cluster 2, and only one was mis-classified in cluster 1. Versicolor showed 27 correctly classified, but 23 mis-classified as Virginica.

This Iris data table is small and used only for illustration. *K*-means works best with larger datasets (maybe 200 or more observations). With smaller tables, the results can be affected by the order of the observations.

This example offers a way to balance the desire for hierarchical clustering with the need to analyze large data sets.

## Scripting Session

### Many Formulas, Several Tables: A JSL Shortcut to Copying

Jeff Perkinson, JMP Systems Engineer

You can copy and paste formulas from any formula editor window to another, but that process becomes tiresome when you have many formulas and several different data tables. I find it handy to use a JSL (JMP Scripting Language) script to copy formulas from one data table and add them as new columns to another table.

The MovingFormulas.jsl script is printed below for reference. Download it from <http://www.jmp.com/news/jmpercable> for your use. To illustrate the script:

1. Open two data tables: one that contains formulas and one to which you want to copy the formulas.
2. Run the MovingFormulas.jsl script in Figure 14 (downloadable from <http://www.jmp.com/news/jmpercable>).

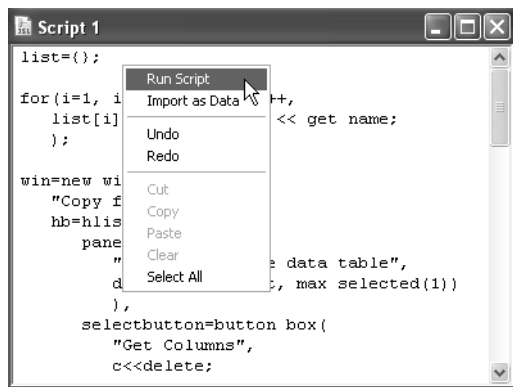


Figure 14: The MovingFormulas.jsl script.

3. The window shown in Figure 15 appears.

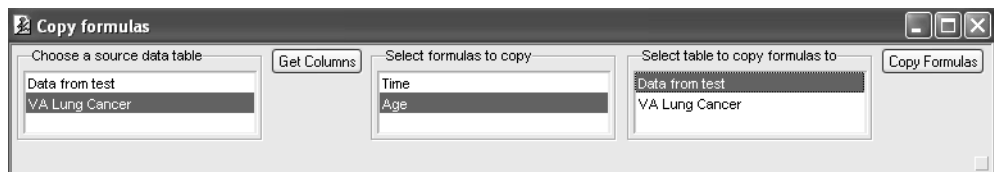


Figure 15: The script produces this window.

4. Highlight the data table containing the formulas you want to copy. Then, click **Get Columns**. A list of the columns in that data table that contain formulas appears in the middle box.
5. Highlight a destination table and click the **Copy Formulas** button.

(continued on page 10)

(continued from page 9)

New columns appear in the destination table containing formulas from the original table.

Below is the MovingFormulas.jsl script that creates a dialog allowing you to move formulas from one data table to another.

```
list={};

for(i=1, i<=n table(), i++,
  list[i]=data table(i) << get name;
);

win=new window(
  "Copy formulas",
  hb=hlist box(
    panel box(
      "Choose a source data table",
      dt=list box(list, max selected(1))
    ),
    selectbutton=button box(
      "Get Columns",
      c<<delete;
      chosen=data table((dt<<get selected)[1]);
      cols={};
      for(i=1, i<=ncol(chosen), i++,
        if(
          char(column(chosen,i)<<get formula) != "Empty()",
          cols[nitems(cols)+1]=column(chosen,i)
        )
      );
      pb << append(
        c=list box(cols)
      )
    ),
    pb=panel box(
      "Select formulas to copy",
      c = list box(" ")
    ),
    panel box(
      "Select table to copy formulas to",
      copyto=list box(list, max selected(1))
    ),
    copytobutton=button box(
      "Copy Formulas",
      collist=c<<get selected;
      dt2=data table((copyto<<get selected)[1]);
      for(i=1, i<=nitems(collist), i++,
        eval(
          substitute(
            expr(dt2<<new column(column(chosen,
              char(collist[i]))<<get name, formula(form))),
            expr(form), column(chosen,char(collist[i]))<<get formula
          )
        )
      );
      win<<close window;
    )
  );
);
```

## Look for JMP at these Conferences

June 7-8, 2005	JMP User Conference in Cary, NC
August 7-11, 2005	Joint Statistical Meetings/American Statistical Association (JSM/ASA) in Minneapolis, MN
September 11-14, 2005	NorthEast SAS Users Group (NESUG) in Portland, ME
September 21-23, 2005	Western Users of SAS Software (WUSS) in San Jose, CA

## Webinar: JMP for Six Sigma in Financial Services

Wednesday, June 15 and Thursday June 16, 2005, 1 pm EDT

This free presentation explores financial services data using interactive descriptive statistics and modeling to rapidly characterize an existing customer base. It shows how recursive partition is used as a time-saving discovery tool at the beginning of the Analyze phase in the DMAIC process to identify customers associated with high variation.

For details, see <http://www.jmp.com/news/webinars.shtml>.

## Meet the Trainer: An Interview

To give you the inside scoop on JMP training instructors, this column features interviews with JMP trainers. We'll learn about their statistics background and interests, along with a few fun facts you can bring up in class.

### Introducing....Paul Marovich

Our feature spotlights Paul Marovich, a full-time JMP instructor in the Statistical Training & Technical Services group.



Paul Marovich

**JMPer Cable:** Hi Paul. What is your background and how did you become interested in JMP training?

**Paul:** I got interested in statistics in a very round about way. I graduated with a degree in management, minor in math from Eckerd College in St. Petersburg, FL.

Right out of college I worked at a Florida dairy for a few years, where I learned to count cows (count the feet and divide by four), climb feed silos, count bales of hay, and manage apartments.

I then audited banks for almost five years. I left my banking job, much to my parent's chagrin, and went back to college pursuing a degree in accounting.

Luckily, I “found” statistics and SAS and earned my master's degree in 1982 from the University of Central Florida (UCF). After a brief stint at UCF consulting and teaching statistics, I worked for Lockheed Martin (LM), where I expanded my SAS knowledge with plenty of hands-on, self-taught experience. I worked at LM for eleven years and only used my statistics training three times.

I have worked for SAS for nine years, the last seven with Education. I first learned about JMP at a quality convention in Orlando.

**JMPer Cable:** What is your role within JMP training?

**Paul:** Teaching classes is my #1 priority. I have enjoyed revising a number of JMP and SAS statistics classes. It has been my pleasure to mentor a number of instructors to get certified in a stat course that I teach.

There is another part of my job that is most rewarding: helping former students/users solve their statistical problems with either JMP or SAS. I do enjoy helping people, so we both benefit.

A user in California was under a severe time restraint, and using her guidance, I devised a JMP solution to her

problem within the time allotted. She was thrilled and I was, too.

**JMPer Cable:** What do you like best about training?

**Paul:** I love sharing my love of statistics, JMP, and SAS. I enjoy sharing the simple things: descriptive statistics, the basics of hypothesis testing and rudimentary modeling. As powerful as some of the statistical methods are, 99.999% of analyses start with examining your data using descriptive statistics and graphs.

**JMPer Cable:** Where do you get the inspiration and enthusiasm for teaching these favorite areas/topics?

**Paul:** I wanted to pursue a degree in mathematics and teach; unfortunately, I experienced topology in college and was turned off. I wanted to be a teacher since my early teens, but economic motivations pushed me in different directions.

My enthusiasm comes from my opinion that statistics is fun. I am sure some people had a boring professor during their college career—I want to be the antithesis of that educator.

My inspiration has been an excellent, fair and motivating high school math teacher and my own brother, Mark, who teaches college mathematics. Mark has known to use self-deprecating humor to keep students interested.

*(continued on page 12)*

---

*(continued from page 11)*

**JMPer Cable:** Any other information you'd like to share?

**Paul:** I am the "mad recycler from the '60s." Both pairs of grandparents were farmers, so it's in my genes. When I arrive at my hotel when I am traveling, I ask if they recycle their newspapers; if not, I put them in my suitcase and take them home. I do the same with plastic bottles, glass containers and cans. Recycling is just a passion of mine.

I will also admit to reading *The Lord of the Rings* at least once a year since 1977.

My wife Pam and my 16-year old son Tyler left their home in Florida to move to Cary, NC. It was a huge sacrifice on their part and I am thankful. I was looking to find my bliss: learning statistics from incredible colleagues, teaching statistics and working for SAS.

#### **About JMPer Cable**

Issue 17 Spring 2005

JMPer Cable is mailed to JMP users who are registered users with SAS Institute. It is also available online at [www.jmp.com](http://www.jmp.com).

#### **Contributors**

Marissa Langford

Paul Marovich

Jeff Perkinson

John Sall

Mike Stockstill

#### **Editor and Designer**

Meredith Blackwelder

#### **Technical Editor**

Ann Lehman

#### **Printing**

SAS Institute Print Center

#### **Questions or comments**

[jmp@sas.com](mailto:jmp@sas.com)

#### **To order JMP software**

1-877-594-6567

#### **For more information on JMP**

1-877-594-6567

[www.jmp.com](http://www.jmp.com)

Copyright © 2005, SAS Institute Inc. All rights reserved. SAS, JMP, JMPer Cable, and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Six Sigma is a registered trademark of Motorola, Inc.