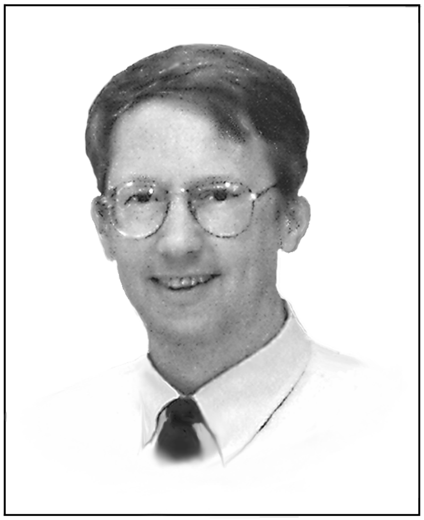
SAS INSTITUTE INC.
SPRING 1996
ISSUE 2

JMPer Cable®

NEWSLETTER FOR JMP® USERS

LOOKING AHEAD



John Sall, Senior Vice President
SAS Institute Inc.

JMP version 3.1.5 was recently released. Contact us or browse our web site <http://www.sas.com/jmp> if you don't have it already. Although we declared it a maintenance release, it does have a number of new features, including better Win95 support, contour profiling for response surfaces, and variability analysis. There will probably be one more release, later in 1996, for version 3.

We are excited to begin development for version 4, which promises to be the most revolutionary "jump" since the original product. Although it won't be out until 1997, we would like to share some of the charter for this new development.

As usual, development priorities are shaped by customer feedback. The most frequent feedback items from customers were:

- scriptability, especially under Windows
- interfaces, especially to Excel
- better graphics for presentation.

Some of these items have already been worked on for the Macintosh version. JMP is scriptable to some extent by Applescript. JMP 3.1.5 on the Mac has a translator to read Excel files directly. And we have taken some steps to improve presentation quality, especially on postscript printers connected to the Mac.

For version 4, we will be bringing these kinds of improvements to Windows and advancing them further. OLE Automation is a logical driving mechanism for JMP under Windows, using Microsoft Visual Basic for Applications (VBA) or other programming languages. This will require a major internal restructuring of JMP. We are also looking at embedability, but this seems to be of secondary concern to users, compared with scriptability.

In addition to these goals, we are restructuring to provide a better system of context so that users will have greater awareness of where results come from, and so that the system can keep better track of data, output, and command histories. We also intend to improve the user interface, embracing the right mouse button to display popup menus contextually. (This will be option-command click on the Mac.) A wider range of guidance mechanisms is also planned.

Version 4 will also have some new analytical platforms. Time series and mixed models are the highest priority from users. Enhancement of existing platforms will focus on supporting our industrial users. The Experimental Design dialogs will be redesigned.

JMP AND WINDOWS 95

by Richard C. Potter
SAS Institute

The year 1995 was a big year in the history of JMP. For it was in April of 1995 that the first Windows version of JMP, version 3.1.2, was released. This release of JMP ran under Windows 3.1 (with Win32s) and Windows NT 3.5. Customer response was tremendous.

Later in the year, Microsoft released its long-anticipated upgrade to Windows, known as Windows 95. Since that time, the developers in the JMP group have been busy modifying JMP for Windows to support Windows 95 features. We are pleased to announce that the latest version of JMP, version 3.1.5, is now fully Windows 95 certified. ➡

(continued on page 2)

IN THIS ISSUE

(Windows 95 continued)

In addition to supporting the new three-dimensional look and feel of Windows 95, some new Windows-only features were added to JMP. They are listed below. It should be noted that these new Windows-only features are also enabled when JMP is run under Windows NT; furthermore, the first two features are also enabled when JMP is run under Windows 3.1.

1) Context-sensitive popup menus appear when the user clicks the right mouse button in any spreadsheet, report, or calculator window. For example, try clicking in the row ID area of a spreadsheet with the right mouse button. You'll see the Rows menu appear as a popup. Try clicking in the column names area and see what other popup menu appears. There are four right mouse button context-sensitive menus active on the spreadsheet:

Click	To See
Row ID area	Rows Menu
Col ID area	Cols Menu
Table body	Edit Menu
Upper left table corner	Tables Menu

   You can also click with the right mouse button in the calculator and in any report window and a modified version of the Edit menu will be displayed.

2) JMP allows the user to open database files directly for those users who have installed Clear Access from Sterling Software. (1-800-522-4252)

3) The File menu contains a new command called Send Mail. Provided your network is running Microsoft Mail, you can use the Send Mail feature to send JMP data and graphics to other users. To send an entire data file, simply attach it to your mail message. To send some part of the data in a spreadsheet or to send a graphic in a report window, use the scissors tool to select what you want and then simply copy it to the clipboard. After you choose Send Mail in the File menu, you can click in the body of the mail message and then paste the contents of the clipboard into it.

4) Filenames specified according to the Universal Naming Convention (UNC) are supported. Thus, it is no longer necessary to first map a network server to a drive letter before accessing JMP data files on that network server. Instead, from within the JMP file open dialog, you can specify a path to the file you want to open by observing the UNC syntax:

   \\server_name\volume\regular_path

   For example, to open the file MyData.JMP in the directory NonLin in the volume Source on the server JMP001, simply specify

   \\JMP001\Source\NonLin\MyData.JMP

   in the file open dialog.

5) JMP supports the registry. In particular, JMP obeys the rules that govern the way in which an application should install and uninstall itself under Windows 95.

6) Last (but not least), the toolbar has been enhanced to provide tool tips, which are the little help notes that appear as you pass the cursor over a tool in the toolbar. These tool tips give a brief description of the functionality associated with each button in the toolbar.

Designed for Microsoft® Windows®95

MOVING DATA INTO JMP

by Jeff Perkinson
SAS Institute

Before you can perform any analysis in JMP, you need some data. There are a couple of ways to get data into JMP. You can type data into a JMP data table, or you can import data that already exists in another form.

Importing Text Data

JMP can import data from any delimited text file. (*What the heck does that mean?*) This means that JMP will read any unformatted text file* that has a unique character separating the columns (fields of information) from each other and another unique character separating the rows from each other. Some other names for delimited text files are CSV (comma-separated values) or tab-delimited files.

For use with JMP, the unique separation characters can be any character at all, but most frequently a tab or comma separates the columns and a carriage return (CR) or a carriage return/line feed (CR/LF) separates the rows.

Now, all of this discussion is fine and well, but what if your data isn't sitting conveniently in a delimited text file? Well, then you will have to convert it from its current format to a format that JMP can read.

With most applications this is done using the Save As command or the Export command from the application that contains the data.

When you are exporting data from another application, you can do a few things to make sure that it imports into JMP more easily. First, save variable/field/column names as the first row in the text file. Then when you import the data, you can use the Labels button on the Import dialog to tell JMP to use the first row in your text file as the column names in the new JMP data table.

Another thing that makes importing your file into JMP more seamless is to look at the first row of actual data in the text file (the row after the column names). Make sure this first row has a value for every column (no missing values). JMP uses the first row of data to determine the column type, numeric or character, of each column. If you have missing data in this first row, JMP can get confused and you might not end up with the data you expected.

Also, make sure that the values in

*Some people call text files ASCII files.

this first row of data correspond to the data type for the column. For example, if you have a column that contains alpha-numeric serial numbers such as "A123S5" or "543VX2," make sure that the first row doesn't contain an all numeric value like "78234." If it does, then JMP assumes the column is numeric. If alphabetic characters are found, they automatically convert to missing values (not what you want).

If needed, you can create a fake first row and use "AAA" (or any character string) for character columns and "999" (or any number) for numerics. This ensures that JMP correctly interprets the data type of each column.

Importing Data from Excel

One of the most frequently asked questions is "How can I import a Microsoft Excel spreadsheet?" There are three ways to do this:

1) You can copy and paste the data from Excel into JMP*. To copy the data from Excel into JMP, do the following:

- Select the cells that you want (including column names if you like) and choose Copy from the Excel Edit menu.
- Then go to JMP; choose New from the File menu to create a new untitled spreadsheet.
- Choose Paste from the Edit menu (hold down the Option/Alt key while choosing Paste if you copied column names with the data). JMP then creates the correct number of rows and columns and pastes the data into them.

2) You can save the spreadsheet as a text file with the Save As dialog as shown in **Figure A**, and import the text file into JMP. If you want to save the spreadsheet as a text file, make sure that it is organized appropriately for JMP. The data should start in the upper left-hand corner of the Excel spreadsheet (in cell A1). This first row can contain either column names or data. If it contains column names, then make sure that there is a name for each column. For the first row of data, make sure that there is a value for each column and that the value matches the data type you want for that column (see the previous section on importing text files).

After your data is organized correctly, you can choose Save As from the Excel File menu. You choose one of the text file options from the Save File as Type pop-up menu. (See the picture of the Excel Save As dialog in **Figure A**.)

There are three text file export options:

- Formatted Text (Space delimited)
- Text (Tab delimited)
- CSV (Comma delimited).

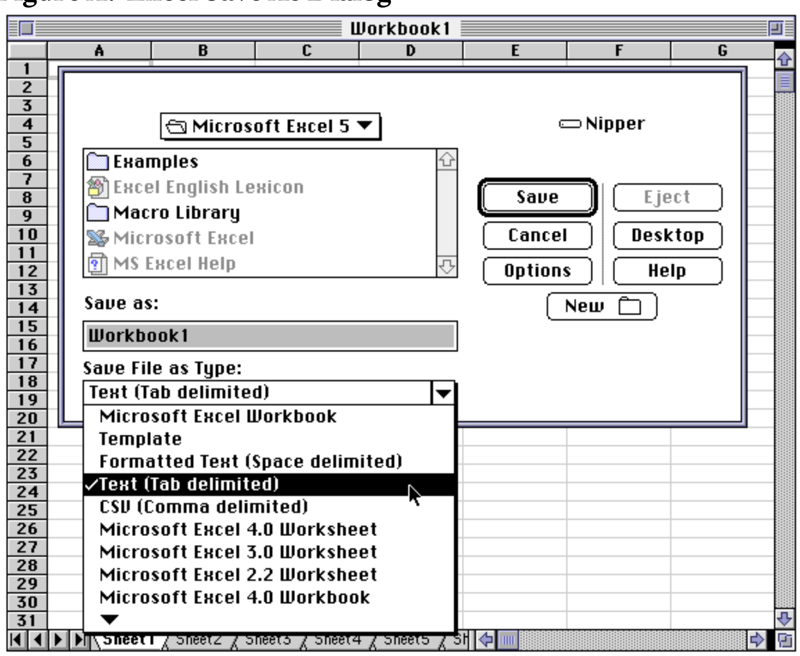
Usually the Text (Tab delimited) choice is best, but any of them will work. Whichever you choose when importing, make sure you check the appropriate End of Record delimiter (Space, Tab, or Comma) in the JMP Import dialog (see **Figure B**).

Figure B: JMP Import Dialog Delimiter Selection

3) On the Macintosh with JMP version 3.1.5, you can import an Excel spreadsheet directly through an XTND translator licensed from DataViz Inc. and distributed with JMP. Before importing your spreadsheet, make sure that it is organized correctly as discussed previously. Then choose Import→Text/Other from the File menu in JMP. In the Import dialog select Excel4.0 or Excel5.0 from the Show: pop-up menu (see **Figure C** on next page). This will show you only Excel4.0 or Excel5.0 files in the dialog file list. Select the Data Only or Labels radio button as appropriate for your text data. Then choose the file you want from the list and click the Import button.

In the future we hope to provide the ability to import from Excel files directly under Microsoft Windows.

Figure A: Excel Save As Dialog

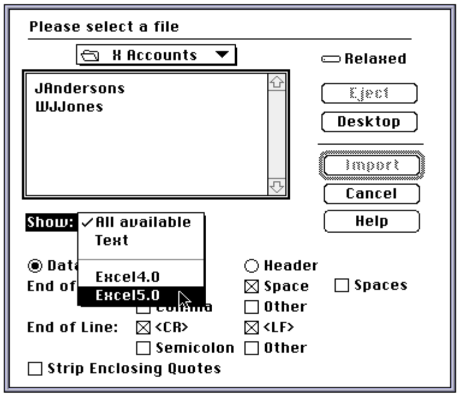


*Actually, you can copy and paste data from almost any application, not just Excel.

➡ (continued on page 4)

(Moving Data into JMP continued)

Figure C: JMP Import Dialog



Importing Data from SAS

If you are a SAS user, you probably have data in the SAS System that you would like to use in JMP. To get this data into JMP you will need to create a SAS Transport Library using the XPORT engine on the LIBNAME statement* A SAS Transport Library is a single file that can contain multiple SAS data files. The syntax for the LIBNAME statement varies according to the platform on which you are running the SAS System. The syntax for Microsoft Windows is:

```
LIBNAME tojmp XPORT 'c:\tojmp.xpt';
```

Note that there is a filename (not a directory name) in the quotes after the XPORT engine. This is a file that does not exist and will be created. It is different from the regular LIBNAME statement, which has a directory name in the quotes. After the LIBNAME statement you can use PROC COPY to copy data files from a standard SAS Data Library to the SAS Transport Library created by the XPORT engine. For example:

```
PROC COPY IN=standard OUT=tojmp;
SELECT somedata;
RUN;
```

This will copy the SAS data file "somedata" from the SAS Data Library "standard" to the SAS Transport Library "tojmp". You can then import the SAS Transport Library using the Import→SAS Transport command from the File menu in JMP.

*JMP will **not read** files created with PROC CPORT.

Tips and Techniques

Find First and Last Rows of a Group to Accumulate Totals

To do JMP table calculator operations based on the first and last rows of a group, you can create one or more new columns in the table to identify these rows. The table in **Figure A** shows the variable *group*. To flag the first row of each group create a new row state column (call it *First in group*) and assign its values with the formula:

$$\text{select}\ (group_i \neq group_{i-1})$$

This expression assigns the row state status as *selected* in the row state column whenever the value of *group* is not equal to the previous value. For SAS users, this is equivalent to the special FIRST.*variable* that SAS creates when a BY statement is in effect.

Figure A: Selection Flag for the First Row of Each Group

7 Rows / 2 Cols	group	First in group
1	A	■
2	B	■
3	B	•
4	C	■
5	C	•
6	C	•
7	D	■

Notice that this method of selection does not consider the order of the group values, or even whether all of the same group are listed together in the table. The selected rows are the same as those SAS selects when the *notsorted* option is in effect. If you want first group selection for each unique group, then the rows in a group must be consecutive by the grouping variable. If they are not, use the Sort command in the Tables menu to rearrange the rows.

You can create columns that contain first row flags for as many grouping columns as you like by using a compound Select formula like the one shown in **Figure B**.

Figure B: First Row Selection Flag for Nested Groups

$$\text{select}\ (group1_i \neq group1_{i-1})\ \text{or}\ (group2_i \neq group2_{i-1})$$

12 Rows / 3 Cols	group1	group2	first row
1	A	x	■
2	A	x	•
3	A	y	■
4	A	y	•
5	A	y	•
6	B	x	■
7	B	y	■
8	B	z	■
9	C	x	■
10	C	x	•

Finding the last row of a group is also simple. JMP lets you use the row index as a subscript to look ahead at the value in the next row. The following formula selects a row whenever the value of *group* is different from its next value.

$$\text{select}\ (group_i \neq group_{i+1})$$

You can use these kinds of formulas to find cumulative totals within a group. The table in **Figure C** shows clients and the dates they made payments. To accumulate payments and see a running total for each client, use the calculator to build the formula shown in **Figure C**.

Figure C: Accumulate Totals within Groups

$$\begin{cases} total_{i-1} + payment, & \text{If } client_i = client_{i-1} \\ payment, & \text{otherwise} \end{cases}$$

7 Rows / 4 Cols	client	date	payment	total
1	John	3/20/94	200.00	200.00
2	Betty	2/10/94	25.75	25.75
3	Betty	2/15/94	10.00	35.75
4	Chris	1/15/94	20.00	20.00
5	Chris	2/15/94	20.00	40.00
6	Chris	3/15/94	20.00	60.00
7	Chris	3/20/94	20.00	80.00

PERFORMING LOGISTIC REGRESSION ON SUMMARY DATA

by Duane Hayes
SAS Institute

Nominal logistic regression with a binary response variable is easily obtained with JMP Software when the data are individual observations. However, if your data are summarized as frequencies of events and trials, a few steps must be taken to rearrange the data correctly for a successful analysis. This article will discuss the steps needed to analyze events/trials data with JMP.

The INGOTS2 (ingots2.jmp) data table (**Figure A**) found in the JMP sample data can be used to demonstrate these steps. The INGOTS2 table contains the following columns:

- *Heat* and *Soak* are the covariates (or independent variables).
- *Ntotal* is the total number of ingots tested at each combination of the *Heat* and *Soak* levels.
- *Nready* contains the number of ingots that passed specifications for rolling at each *Heat* and *Soak* combination.

(The variables called *P* and *Loss* are not used in this example.)

The first thing to do is add another column to the data table to contain the number of ingots that did not pass specifications. Choose New Column from the Cols menu to create a new

Figure A: Partial Listing of the INGOTS2 Data Table

INGOTS2

6 cols / 19 Rows	Heat	Soak	Ntotal	Nready	P	Loss
1	7	1	0	10	0.50	6.9315
2	7	1.7	0	17	0.50	11.7835
3	7	2.2	0	7	0.50	4.8520
4	7	2.8	0	12	0.50	8.3178
5	7	4	0	9	0.50	6.2383
6	14	1	0	31	0.50	21.4876

column (call it *Nready:not*). Use the calculator to compute its values with the formula *Ntotal – Nready*.

Next, you need to create a single column containing frequency counts of both the ready and the not ready ingots. To do this, use the Stack Columns command from the Tables menu. Complete the Stack Columns dialog as shown in **Figure B**: choose *Nready:not* and *Nready* as the columns to stack, name the stacked column *Count*, and name the ID column *Response*. When you click OK, the new Untitled table shown to the right in **Figure B** is created. The newly created data table is now set up to run a logistic regression analysis.

Now, because JMP models the lower (alphanumeric) ordered value as the event, we know that we will be fitting the logit of *Nready*:

$$\text{Logit}(\Pr(Nready)) = \ln\left(\frac{\Pr(Nready)}{1 - \Pr(Nready)}\right)$$

At this point we can perform the analysis. Select Fit Model from the Analyze menu. Choose *Response* as Y, *Count* as the Frequency column, and *Heat* and *Soak* as the Effects in Model. Now, click Run Model to see the analysis shown in **Figure C**.

Figure C: Results of Logistic Regression

Response: Response

Iteration History

Converged by Gradient

Whole-Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	5.821410	2	11.64282	0.0030
Full	47.672807			
Reduced	53.494217			

RSquare (U)	0.1088
Observations (or Sum Wgts)	387

Lack of Fit

Source	DF	-LogLikelihood	ChiSquare
Lack of Fit	16	6.876314	13.75263
Pure Error	368	40.796493	Prob>ChiSq
Total Error	384	47.672807	0.6171

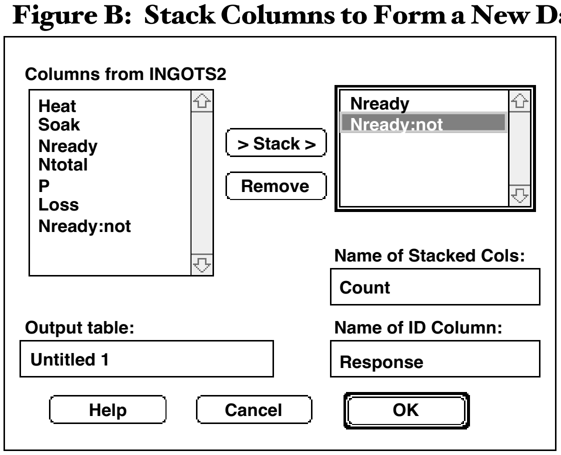
Parameter Estimates

Effect Test

Source	Nparm	DF	Wald ChiSquare	Prob>ChiSq
Heat	1	1	11.945205	0.0005
Soak	1	1	0.029379	0.8639

From the output, we can see that *Heat* is a significant factor (with a p value of 0.0005) in predicting the readiness of ingots. On the other hand, Soak , with a p value of 0.8639, does not appear to be significant at all. In this case, a subsequent model (not shown here) could be fit with *Heat* as the sole effect. If you want to fit a model with only one effect, you can use the Fit Y by X platform, which shows a cumulative logistic probability plot with the logistic regression report.

Figure B: Stack Columns to Form a New Data Table



Stacking columns interleaves the values of two or more columns into a single column. An ID column is automatically created that contains the original column names of the stacked values. In this example, the ID column (called Response) becomes the Y variable for the logistic regression analysis. The stacked column (called Count) is the X.

Untitled 1

7 Cols / 38 Rows	Heat	Soak	Ntotal	Count	Response
1	7	1	10	10	Nready:Not
2	7	1	10	0	Nready
3	7	1.7	17	17	Nready:Not
4	7	1.7	17	0	Nready
5	7	2.2	7	7	Nready:Not

ANALYZING REPEATED-MEASURES DATA

by Duane Hayes
SAS Institute

Often in an experiment, more than one measure is taken on the same subject or experimental unit. This means that a repeated measures analysis of the data is necessary to make valid inferences and to draw meaningful conclusions.

JMP offers two methods to analyze repeated measures: a univariate split-plot approach and a multivariate repeated-measures approach. These two types of analyses are compared in the following discussion. The example uses the DOGS (dogs.jmp) sample data table, which is the result of a study with repeated measures.

The left-hand table in **Figure A** shows selected columns from the DOGS table. This data table contains information on sixteen dogs assigned to groups defined by the independent variable *drug* with values 'morphine' and 'trimeth.' The blood concentration of histamine is recorded before drug injection and again at one and three minutes after injection. Also, there is an additional *subject* column, declared Nominal, which will be used to account for the within-subject variability in a univariate analysis of repeated-measures data.

Figure A: Arrangement of Data Tables for Repeated-Measures Analysis

Multivariate Repeated Measures

11 Cols / 16 Rows	subject	drug	LogHist0	LogHist1	LogHist3
1	1	morphine	-3.219	-1.609	-2.303
2	2	morphine	-3.912	-2.813	-3.912
3	3	morphine	-2.659	0.336	-0.734
4	4	morphine	-1.772	-0.562	-1.050
5	5	morphine	-2.303	-2.408	-2.040
6	6	morphine	-2.120	-2.207	-2.303
7	7	morphine	-2.659	-2.659	-2.813
8	8	morphine	-2.996	-2.659	-2.813
9	9	trimeth	-3.507	-0.478	-1.171

Univariate (Stacked) Measures

10 Cols / 48 Rows	Subject	drug	Loghist	Time
1	1	morphine	-3.219	LogHist0
2	1	morphine	-1.609	LogHist1
3	1	morphine	-2.303	LogHist3
4	2	morphine	-3.912	LogHist0
5	2	morphine	-2.813	LogHist1
6	2	morphine	-3.912	LogHist3
7	3	morphine	-2.659	LogHist0
8	3	morphine	0.336	LogHist1
9	3	morphine	-0.734	LogHist3

A Univariate Approach

A univariate model has a single response. Each source of variation (between-subjects and within-subjects) is included as an effect in the model.

The univariate analysis requires that the response measurements be in a single column. Using the Stack command in the Tables menu, you can stack the *LogHist0*, *LogHist1*, and *LogHist3* columns to create a new response column, as shown in the right-hand table of **Figure A**. The new response, *LogHist*, and also the new classification variable, *Time*, were created by the Stack command.

With the data table correctly set up, choose Fit Model from the Analyze Menu. Select *LogHist* as Y, and add the Effects in Model as shown in **Figure B**.

The between-subject effect, *drug* is the whole plot effect of a split-plot design. The *Subject* effect is nested within *drug*. This *Subject(drug)* effect is the appropriate error term for the between-subject effect. Therefore, it is specified as a random effect using the Random selection from the Effect Attributes pop-up menu. JMP uses a random effect as the error term to test appropriate terms in the Effects in Model list.

The within-subject effects, *Time* and *drug*Time* will be tested with the residual error term.

After running this model you can examine the significance of the model effects seen in the 'Tests wrt Random Effects' table shown in **Figure C**.

A Multivariate Approach

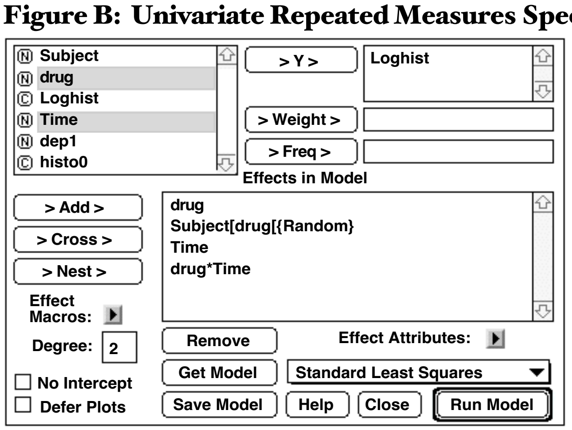
The original DOGS table with multiple response columns is in the form needed for a multivariate analysis that tests the same effects—nothing needs to be changed.

With this data table active, again choose Fit Model from the Analyze Menu. Specify *LogHist0*, *LogHist1* and *LogHist3* as Y variables. Assign *drug* as the model effect and run the model.

When there are multiple Y variables, JMP automatically performs a multivariate analysis. When you first run the model, the multivariate control panel appears. To test the effect of *drug* over time, select "Repeated Measures" as the response design from the popup menu on the control panel.

In the repeated-measures dialog that appears, use the default effect name Time but check 'Univariate Tests Also' to obtain univariate and adjusted univariate tests. This option includes a test of sphericity (not shown here), which checks whether the unadjusted ➡

Figure B: Univariate Repeated Measures Specification



To create the model shown to the left, which has a random effect for testing the between-subjects effect in the model, click the following variables and buttons:

- ***drug*** in the variable selection list, then **Add**
- ***subject*** in the variable selection list, then **Add**
- ***drug*** in the variable selection list, and
- ***Subject*** in the Effects in Model list, then **Nest**
- ***Subject[drug]*** in the Effects in Model list, then
  select **Random** from the Effect Attributes pop-up menu giving
  ***Subject[drug]*{Random}** in the Effects in Model list
- ***Time*** in the variable selection list, then **Add**
- ***drug*** in the variable selection list and
  Time in the variable selection list, then **Cross**

(Analyzing Repeated Measures continued)

univariate F tests are appropriate. If the sphericity chi-square test is not significant, you can use the unadjusted univariate F tests. However, if the sphericity test chi-square is significant, then the criterion is rejected and the multivariate F tests or the adjusted univariate F tests should be used. JMP gives both the Greenhouse-Geisser (G-G), and the Huynh-Feldt (H-F) adjusted F tests.

Comparing the Two Methods

A comparison of analysis results is shown in **Figure C**. By examining each analysis, you can see the relationship between the univariate and multivariate effects tests:

1 For between-subject effects (*drug*), the multivariate approach gives the same results as the univariate approach when there are no missing values. If there are missing values, a univariate analysis uses all nonmissing data values but the multivariate analysis excludes any subject with any missing values.

2 If there are no missing values, the within-subjects *Time* variable in the univariate model is the same as the unadjusted Time effect in a multivariate model having a "Repeated Measures" response design. However, these tests are appropriate only if the sphericity test criterion (mentioned previously) is met. Otherwise, the multivariate tests or the adjusted univariate tests should be used.

3 Often in a repeated-measures study the most important effect is the within-subject by between-subject interaction—the hypothesis of interest is whether the study treatment has an effect over time. In the univariate analysis, the *drug*Time* interaction appears insignificant.

4 In this example the more powerful multivariate approach shows the *drug*Time* interaction effect to have marginal significance.

In summary, if you have repeated-measures data, JMP can analyze the data as either a univariate split-plot model or a multivariate model. Each type of analysis has its advantages and disadvantages:

- The multivariate analysis is easy and intuitive to specify in JMP. Its tests are usually more powerful. From a computing standpoint, this method is most efficient. However, if a subject is missing a value, all information for that subject is lost to the analysis.
- The univariate analysis can use all the data—only a subject's missing measurement is lost to the analysis. However, the univariate analysis can be very computationally intensive, particularly if there are many subjects.

Also, the univariate tests for within-subject effects and interactions involving these effects require assumptions about the covariance matrix in order for the probabilities provided by the ordinary F tests to be correct. If these assumptions are not met (if the sphericity test is rejected), then probabilities for adjusted univariate F tests (given in the multivariate report) or the multivariate F tests should be used. Because of these assumptions, the univariate approach should only be considered when the Sphericity condition is met. For more information see the discussion on this test in the *JMP Statistics and Graphics Guide*, in the Multivariate Model Fitting chapter.

Cole and Grizzle, J.E. (1966), "Sixteen Dogs," Biometrics, 22:810

"JMP is one of the key elements in Read-Rite's strategy for product development. The time it takes us to debug a product using JMP has gone literally from a month to a day."

Tom Little
Director of Engineering Support
Read-Rite Corporation

Figure C: Comparison of Univariate and Multivariate Results

Univariate Analysis

Tests wrt Random Effects

Source	SS	MS Num	DF Num	F Ratio	Prob>F	
drug	3.10683	3.10683	1	1.3380	0.2667	1
Subject[drug]	32.5076	2.32197	14	3.6231	0.0018	
Time	12.6804	6.34018	2	9.8929	0.0006	2
drug*Time	1.82032	0.91016	2	1.4202	0.2585	3 4

Multivariate Analysis

drug effect (between-subjects)

drug

Test	Value	Exact F	DF Num	DF Den	Prob>F
Wilks' Lambda	0.912765	1.3380	1	14	0.2667
Pillai's Trace	0.087235	1.3380	1	14	0.2667
Hotelling-Lawley	0.0955723	1.3380	1	14	0.2667
Roy's Max Root	0.0955723	1.3380	1	14	0.2667

time effect (within-subjects)

Time

Test	Value	Exact F	DF Num	DF Den	Prob>F
Wilks' Lambda	0.560946	5.0876	2	13	0.0233
Pillai's Trace	0.439054	5.0876	2	13	0.0233
Hotelling-Lawley	0.7827029	5.0876	2	13	0.0233
Roy's Max Root	0.7827029	5.0876	2	13	0.0233
Univar unadj Epsilon=	1	9.8929	2	28	0.0006
Univar G-G Epsilon=	0.5487821	9.8929	1.0976	15.366	0.0056
Univar H-F Epsilon=	0.6030268	9.8929	1.2061	16.885	0.0042

drug by *time* effect (within-subject by between-subject interaction)

Time*Drug

Test	Value	Exact F	DF Num	DF Den	Prob>F	
Wilks' Lambda	0.6458622	3.5641	2	13	0.0583	
Pillai's Trace	0.3541378	3.5641	2	13	0.0583	4
Hotelling-Lawley	0.5483178	3.5641	2	13	0.0583	
Roy's Max Root	0.5483178	3.5641	2	13	0.0583	
Univar unadj Epsilon=	1	1.4202	2	28	0.2585	3
Univar G-G Epsilon=	0.5487821	1.4202	1.0976	15.366	0.2553	
Univar H-F Epsilon=	0.6030268	1.4202	1.2061	16.885	0.2569	

Calculator Corner

Computing Across? or Computing Down?

by
Michael Hecht
SAS Institute

Sometimes you need to compute a statistic for a set of numbers in a data table using the JMP calculator. The calculator has functions that compute statistics for a column of numbers, and other functions to find statistics across columns. Also, there is often more than one way to compute simple univariate statistics with calculator functions. Here we will give examples of operating both down columns and across columns.

Finding Means

When you click the Statistical category in the function browser you see the list of functions shown in **Figure A.** All these functions except Index and Of operate on a single argument that can be either a column of numbers or an expression involving multiple numeric columns. But by themselves they do not find statistics across columns. When you want to compute across columns, you combine these functions with the Of operator.

Figure A: Statistical Functions

Conditions	# of nonmissing
Random	Quantile
Probability	Mean
Statistical	Std Deviation
Dates	Sum (Σ)
Row State	Product (Π)
Parameters	Index
Variables	Of
Editing	

The table in **Figure B** shows examples of using the Mean function for both down and across computing:

- The *mean wt1* column is computed with the formula that is denoted

  $$\overline{wt1}$$

  giving the constant 107.8.
- The *mean of wt1, wt2* column has the formula

  $$\overline{\text{of}(wt1, wt2)}$$

  which gives the mean of the *wt1* and *wt2* values for each row.

Figure B: Compare Mean **with** Mean **and** Of **Functions**

4 Cols / 5 Rows	wt1	wt2	mean wt1	mean of wt1,wt2
1	95	100	107.8	97.5
2	123	120	107.8	121.5
3	77	77	107.8	77.0
4	145	150	107.8	147.5
5	99	95	107.8	97.0

You can explicitly construct the mean across columns with this formula:

$$\frac{wt1 + wt2}{\text{N of}(wt1, wt2)}$$

But beware, if any term in the numerator sum is missing, the computation gives a missing result. It is best to use

$$\overline{\text{of}(arg, arg, arg, \ldots)}$$

The Sum **(Σ) Function**

To calculate a sum, select Sum(Σ) from the Statistics function list and choose a variable or create an expression as its argument. The expression automatically appears with the index variable j. Sum(Σ) repeatedly evaluates its expression for j = 1, j = 2, through j = n and then adds the nonmissing results together to determine the final result.

Both the upper and lower indices of the Sum(Σ) function and its argument can be complex numeric expressions.

As with other statistical functions you can use Sum(Σ) to sum across columns. When the Of and Sum(Σ) are used together, the Sum(Σ) function sums the arguments of the Of function; the index variable n defaults to the number of arguments in the Of list. Thus the summation shown here

$$\sum_{j=1}^{n} \text{of}(wt1, wt2)$$

finds the sum of *wt1* and *wt2* for each row.

The advantage to using the Sum(Σ) function instead of just creating an addition expression is that it ignores missing values. The result of any arithmetic expression is missing if any of its terms are missing.

You can also find the mean of a numeric column by constructing its explicit formula. To do this you use the Sum (Σ) and # of nonmissing (shown as N in formulas) functions.

So, the two calculator formulas for computing the mean of column *wt1*, are

$$\overline{wt1} \quad \text{and} \quad \frac{\sum_{j=1}^{n} wt1_j}{\text{N } wt1}$$

The Quantile **Function**

Next, look at the versatile Quantile_n function. Like the other statistical functions, it works on a single expression and returns a constant determined by the index you specify. The index can be simple: quantile_0 finds the minimum in a column of numbers, $\text{quantile}_{.5}$ finds the median, and quantile_1 finds the maximum.

Quantile used with the Of function computes quantiles across columns.

Note: The Max function in the list of Numeric functions is the special case of Quantile_n Of (*arg*) where the quantile subscript is 1. Likewise, Min is the case when the subscript is zero.

Figure C compares the Max, Quantile_1 and Quantile_1 Of Functions.

Figure C: Max**,** Quantile**,** Quantile Of

wt1	wt2	Quant1(wt1)	Quant1 Of(wt1,wt2)	max
95	100	145	100	100
123	120	145	123	123
77	77	145	77	77
145	150	145	150	150
99	95	145	99	99

What makes the Quantile function so powerful is that you can construct as complex an index as needed. For example, the Quantile function shown here

$$\text{quantile}_{\left(\frac{i-1}{n-1}\right)} wt1$$

produces a column in which the argument (*wt1*) is listed in ascending sorted order. (If you change the numerator of the index to n – i, *wt1* is listed in descending sorted order.)

THE VALUE OF CUT DIAMONDS

by Ann Lehman
SAS Institute

When you do a *t* test or one-way analysis of variance using the Fit Y by X command, the platform first displays a vertical point plot for each X group. As you request more information using commands in the options menus beneath the plot, more and more information is added to the point plot, and supporting statistical tables appear:

- The Means, Anova/t-Test platform command draws means diamonds with overlap marks for each group of points, and generates the Summary of Fit, Analysis of Variance, and Means for Oneway Anova tables.
- The Means, Std Dev, Std Err platform command draws means dots, standard deviation bars, and standard error bars on each group of points, and generates the Means and Std Deviations table.

When both of the above commands are in effect, you have four pieces of information (not including the mean itself and the X-axis proportional sample size information). The diagrams in **Figure A** show you how each item is calculated:

1 The dotted lines are one standard deviation above and below the mean of each group. These values for each group are listed in the Means and Std Deviations table.

2 Standard error bars are placed at 1 standard error above and below the group mean. The standard error for each group is the Std Err Mean listed in the Means and Std Deviations table. It is calculated by dividing the group Std Dev with the square root of the group sample size (Number).

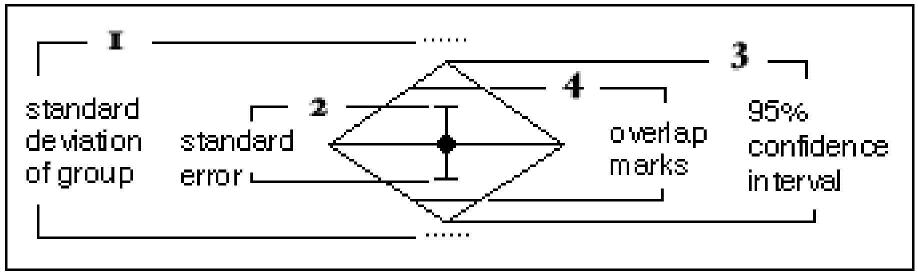
3 The vertical height of the means diamond shows the 95% confidence interval (CI) around the group mean. The CI computation assumes equal variances across groups, so is computed using the Std Errors shown in the Means for Oneway Anova table. These Std Errors are the Root Mean Square Error (shown in the Summary of Fit table) divided by the square root of the group sample size (Number). The Root Mean Square Error is the square root of the pooled estimate of error variance, which is the Mean Square for Error in the Analysis of Variance Table.

4 Overlap marks drawn in the means diamonds at the mean ± $\sqrt{2}/2$ • CI / 2. For groups with equal sample sizes, these marks show if the two group means are significantly different at the 95% confidence level.

Figure B diagrams means diamonds and comparison circles that represent a 95% statistical difference. The radius, R, is one-half the confidence interval. Tangent lines drawn at the intersection of the circles form a 90 degree angle, so the legs of the right triangle shown are equal. Set R to 1 and solve to find the overlap distance of $\sqrt{2}/2$.

See the *JMP Statistics and Graphics Guide* for more information about comparison circles.

Figure A: Means Diamond and Statistical Tables



Summary of Fit

RSquare	0.235587
RSquare Adj	0.145656
Root Mean Square Error	3.695709
Mean of Response	62.75
Observations (or Sum Wgts)	20

4 Overlap Marks are computed
Group Mean ± ($\sqrt{2}/2$) • CI / 2

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	2	71.55952	35.7798	2.6196
Error	17	232.19048	13.6583	
C Total	19	303.75000	15.9868	

3 Height of Means Diamond is the 95% confidence interval (CI)

CI = Group Mean ± $\left(t_{(Number-1, .95)}\right)$ • Std Error

$$\text{Std Error} = \frac{\text{Root Mean Square Error}}{\sqrt{\text{Number}}}$$

Means for Oneway Anova

Level	Number	Mean	Std Error
A	7	64.7143	1.3968
B	6	63.3333	1.5088
C	7	60.2857	1.3968

Std Error uses a pooled estimate of error variance

Means and Std Deviations

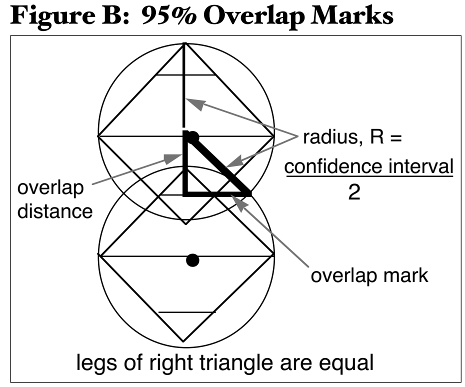
Level	Number	Mean	Std Dev	Std Err Mean
A	7	64.7143	3.49830	1.3222
B	6	63.3333	1.50555	0.6146
C	7	60.2857	4.95696	1.8736

2 Std Error Bars are at
Group Mean ± Std Err Mean

$$\text{Std Err Mean} = \frac{\text{Std Dev}}{\sqrt{\text{Number}}}$$

1 Std Dev Lines show at
Group Mean ± Group Std Dev

Figure B: 95% Overlap Marks



ENTERING MULTIPLE RESPONSE SURVEY DATA: A Shortcut

by Ann Lehman and Bill Gjertsen
SAS Institute

Survey data are typically somewhat messy. This is especially true when there can be multiple responses to a single survey question. This happens whenever you see an item that says something like “check all that apply to you.”

Often, you want to have a data table with a column for each possible response to the multiple response item. Then, for each respondent (each row in the table) a code is entered in each column for which there was a response. **Figure A** shows an example where a survey item has possible responses a, b, c, d, and e. The respondents can indicate any or all responses, which are then coded as 1 in the corresponding data table column.

Entering this kind of coded data can be tedious and is very vulnerable to human error. The following example shows a data entry shortcut that speeds up the entry and, hopefully, reduces the possibility of error.

First, you create a single character column in which you enter a string consisting of each respondent’s responses for an item. For example, the data table in **Figure B** shows the values “abd” for row 1, “bce” for row 2, and so on. Then, a calculator formula in each response column extracted the correct code (1, or blank) for each response code.

The calculator formula to parse the entry codes uses the infamous Munger function. Each response column has a formula like the one shown beneath the table in **Figure B**, that searches for its specific response. Whenever the response is found, the column is coded as 1. The example formula is for the “a” response (the *1a* column). Formulas for the other 4 columns use “b”, “c”, “d”, and “e” as find values in the Munger function. Note that the lowercase function allows the *code* entry to be either lowercase or uppercase.

In this example, a formula also checks the entry for invalid response values. When a response other than a, b, c, d, or e is found, the error is noted and shown in the column called *error* (see **Figure B**).

Figure C shows the formula that does the error checking. It uses the Assignment statement to create the error column, which works like this:

1) The first Munger function checks the entered code for a correct entry (“a” in this example). If an “a” is found, it is removed from the code string.
2) The modified entry string is assigned (⇐) to the temporary variable called ***a***.
3) The next Munger function checks the modified code string (now in ***a***) removes the next correct entry (“b” in this example), and assigns the result to the temporary variable ***b***.
4) The sequential checking process continues until all correct responses found have been removed from the entry string. This final result is assigned to the variable ***e***.

The code string in ***e*** should be blank after all correct responses have been removed. Any remaining characters are an entry error.

5) The results clause of the assignment statement checks the variable ***e*** to see if it is blank. If it is not, the error column shows the word “error” followed by the incorrect characters in parentheses.

Figure A: Survey Data Table

6 Cols / 8 Rows	ID (N)	1a (N)	1b (N)	1c (N)	1d (N)	1e (C)
1	AL	1	1	•	1	•
2	JS	•	1	1	•	1
3	MH	1	•	•	•	1
4	KN	1	1	1	•	•
5	DT	•	1	1	1	•
6	RP	1	1	1	1	1
7	RL	•	•	•	•	•
8	JP	1	•	•	•	•

Figure B: Assigning Response Codes with a Formula

8 Cols / 8 Rows	ID (N)	code (N)	1a (N)	1b (N)	1c (N)	1d (N)	1e (C)	error (N)
1	AL	abD	1	1	•	1	•	
2	JS	bce	•	1	1	•	1	
3	MH	aEq	1	•	•	•	1	error(q)
4	KN	abcfr	1	1	1	•	•	error(fr)
5	DT	bcd	•	1	1	1	•	
6	RP	abcde	1	1	1	1	1	

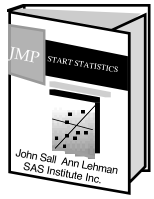
{ 1 , if munger (lowercase *code*, starting at 1, find “a”, replace with □) ≠ 0
{ □ , otherwise

Figure C: Formula to Check for Errors

*The **Variables** feature in the calculator is used to create temporary variables called **a**, **b**, **c**, **d**, and **e**. The five **Munger** functions remove all valid codes from the **Code** string. The **Assignment** statement assigns the results of the five **Munger** functions to the variables. If the last variable (**e**) is not blank, there is an incorrect code.*

a ⇐ munger (lowercase *code*, starting at 1, find “a”, replace with “ ”)
b ⇐ munger (***a***, starting at 1, find “b”, replace with “ ”)
c ⇐ munger (***b***, starting at 1, find “c”, replace with “ ”)
d ⇐ munger (***c***, starting at 1, find “d”, replace with “ ”)
e ⇐ munger (***d***, starting at 1, find “e”, replace with “ ”)

results { “error (” ll ***e*** ll “)” , if ***e*** ≠ “”
{ □ , otherwise



Book Review

JMP START STATISTICS

A Guide to Statistics and Data Analysis Using JMP and JMP IN Software

There has been a long-standing demand for a book that focuses on using JMP to learn about statistics. *JMP Start Statistics* is a new book available from SAS Institute that does just that. It is a friendly but comprehensive approach to statistics, and a guidebook for learning to use JMP software.

This new book is also the official reference document that accompanies JMP IN Version 3.5, the student edition of JMP. The JMP IN package (software and book) is available to students from Duxbury Press and also through many campus bookstores.

JMP Start Statistics is written by John Sall, the principle developer of JMP, and Ann Lehman. It has over 500 pages of statistical discussion and explanation (interspersed with occasional informal thought seeds). It also contains reference documentation for JMP Software.

Each topic is supported with hands-on examples of varying levels of complexity. JMP's interactive statistical platforms help clarify and simplify what are often perceived as difficult and tangled-up statistical concepts.

The interactive and graphical nature of the JMP examples promotes the idea that statistics is a discovery process:

- Graphics that accompany each analysis make it easier to understand results.
- Interactivity leads to further analysis, refines results, and can result in further discovery.

JMP Start Statistics begins with the basics—one variable, one sample to look at univariate distributions. Topics progress through simple regression, *t* tests and analysis of variance, analysis of categorical variables, multiple regression, correlations and multivariate relationships, and fitting general linear models. Also, specialty chapters discuss design of experiments and elementary time series.

For more information about *JMP Start Statistics*, contact SAS Institute at (919) 677-8000 x6802, or write to
SAS Institute Inc.
SAS Campus Drive
Bldg R-5134
Cary, NC 27513

For information about using JMP IN software in your classes, contact Duxbury Press at (800) 425-0563.

LEARN LEARN LEARN:
New Training for Engineering Users of JMP Software

Engineering Statistics Using JMP Software is a new, hands-on three-day course for both new and current JMP users. This course focuses on standard topics in interactive data analysis, as well as on reliability analysis and statistical quality control. All examples and exercises are based on applied engineering problems.

The course complements the two existing JMP software courses, Interactive Data Analysis Using JMP Software and Design and Analysis of Experiments Using JMP Software, by providing JMP training specific to engineers' needs.

This new engineering course begins with an introduction to interactive data analysis using JMP and covers the following topics:

- basic data table manipulation
- distributions of values
- analyzing single variables
- relationships between pairs of columns (variables)
- performing regression analysis using single predictors
- relationships between many variables
- assessing correlation
- identifying outliers
- performing regression analysis using multiple predictors
- introduction to reliability analysis
- understanding reliability concepts
- estimating the reliability function
- testing for group differences in survival times
- selecting an appropriate distribution for modeling reliability data
- estimating the distribution parameters
- Statistical Quality Control
- constructing control charts
- Shewhart control charts for both variable and attribute data
- individual measurement and moving range charts
- moving average charts using either exponential or uniform weighting
- constructing Pareto charts
- performing capability analysis.

This new course will be offered at SAS Institute regional training centers in

Boston	**March 20-22**
Austin	**April 17-19**
Philadelphia	**May 6-8**
Detroit	**June 24-26**

To register for the course, call a course registrar at a regional training center near you, or call Cary, NC at (919)-677-8000, then press 1-5005.
Ask about available discounts.

To schedule a course at your location, call your Professional Services account representative at (919)-677-8000, then press 1-7321.

For questions about course content or for information about the other JMP training courses, call (919)-677-8000, then press 1-7205.

JMP®

You are invited to visit and talk with us at conventions and tradeshows:

Pittcon	3/3-8	Chicago	Comdex/Windows World	6/3-6	Chicago
*SUGI	3/10-13	Chicago	*ICE 96	6/11-15	San Francisco
*ACS - (spring)	3/24-28	New Orleans	*ASA	8/4-8	Chicago
Experimental Biology 96	4/14-17	Washington	Mac World	8/7-10	Boston
Quality Expo	4/23-25	Chicago	*ACS - (fall)	8/25-29	Orlando
*ASQC	5/13-15	Chicago	Comdex/Windows World	11/18-22	Las Vegas
*ASBMB/ASIP/AAI	6/2-6	New Orleans			

***Complete Show Titles:**

SUGI – SAS User's Group International
ACS – American Chemical Society
ASQC – American Society for Quality Control
ASBMB/ASIP/AAI – joint conference: American Society of Biochemistry and Molecular Biology / American Society of Investigative Pathology / American Association of Immunologists
ICE – 96 International Congress of Endocrinology
ASA – American Statistical Association

EDITOR
Ann Lehman

CONTRIBUTORS
John Sall
Richard Potter
Duane Hayes
Jeff Perkinson
Michael Hecht
William Gjertsen
Ann Lehman

TYPOGRAPHY AND DESIGN
Ann Lehman
Jeff Perkinson
Mike Pezzoni

PRINTING
SAS Institute Copy Center

JMPer Cable is sent only to JMP users who are registered with SAS Institute. If you know of JMP users who are not registrered, pass them a copy of JMPer Cable and let them see what they are missing!

If you have questions or comments about JMPer Cable, or want to order more copies, write to

JMPer Cable
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

For More Information on JMP or to order, contact SAS Institute, JMP Sales
phone: 919-677-8000 x 5071
fax: 919-677-8224

You can also browse our web site at
<http://www.sas.com/jmp>

If you don't keep JMPer Cable for reference, please recycle it!

SAS®

SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

Address Correction Requested

Bulk Rate
U.S. Postage
PAID
SAS Institute Inc.