A Technical Publication for JMP® Users
Spring 2002 Issue 9

JMPer Cable

Inside This Newsletter

Breaking News

The next JMP Webinar, “Introduction to JMP Software,” will be on June 17 at 1 pm EST.

Join us to learn JMP’s ease of use and discover how using it can help you make discoveries that impact your bottom line.

For more information and to register online, visit www.jmpdiscovery.com.

To register over the phone, call 1-877-594-6567 and ask for JMP Sales.

Highlights of JMP version 5

John Sall, Executive Vice President, SAS Institute Inc.

JMP version 5 is on its way. The following article is a quick overview of version 5’s major additions and enhancements.

New Appearance

JMP version 5 puts a new look on the face of JMP. True to its support of Macintosh and Windows, it supports the appearance improvements from the latest versions of Windows XP and MacOS X.

Enhanced Data Exploration

Version 5 enhances the data exploration capabilities of JMP with features usually found only in data mining packages.

The Partition platform recursively partitions observations. It forms a tree that makes a split by one of the *x* variables that most predicts a *y* variable. In addition to this tree form, JMP shows results in an innovative partitioned graph (see Figure 1).

The Neural Net platform provides a way to fit flexible models using a layer of intermediate (hidden) variables formed by S-shaped functions of linear combinations of *x* variables. Figure 2 gives an example of the Neural Net platform. Neural Net supports both the Profiler and Contour Profiler to explore the response surface.

New Design of Experiments (DOE)

In version 5, Custom Design produces response surface designs that are optimal for the average variance of prediction (*I*-Optimal). For simple situations, these designs are similar to traditional classical response surface designs, but they can be adapted to special situations, including handling mixture factors. This is the default design optimization criterion for models with polynomial effects.

Furthermore, in version 5 you can ask for designs in which certain terms are not required to be estimable, but will be estimated if possible (Bayesian *D*-Optimal). This feature also enables you to obtain supersaturated designs.

Optimized for Six Sigma

In version 5, we’ve added new cause-and-effect diagrams, also called Ishikawa diagrams. Figure 3 shows an example of the Diagram platform.

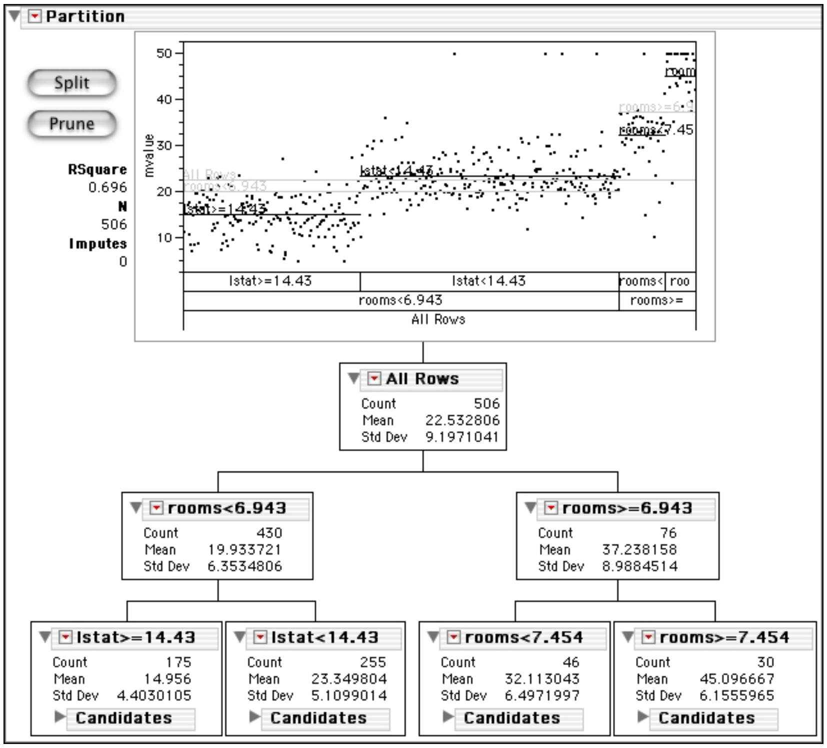


Figure 1: The Partition platform in version 5 shows results in a partitioned graph and in tree form.

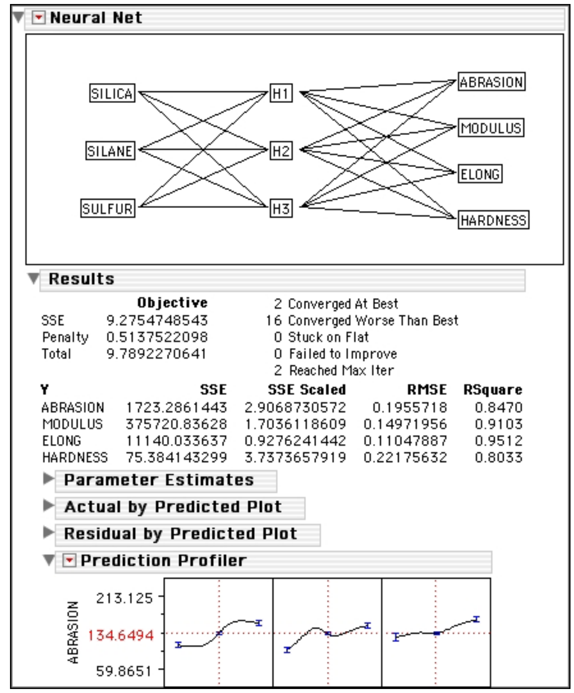


Figure 2: The Neural Net platform in version 5 supports both the profiler and contour profiler to explore the response surface.

Enhanced Reliability Analyses

Version 5 supports interval censoring in both the Univariate and Regression Survival platforms. Also, the Regression Survival platform supports more graphics and estimation features. The Recurrence platform has a new graph which compares different mean cumulant functions.

Improved Multivariate Platforms

Discriminant analysis, originally part of JMP's Fit Model platform, has been elevated to its own Discriminant platform. This new platform is capable of doing stepwise variable selection.

There is also a new PLS (partial least squares) platform that lets you analyze data that have more responses than effects (i.e., when ordinary least-squares analysis is impossible). PLS implements a profiler to explore the relationships captured.

Better Formulas

Version 5 implements new routines for random number generation. Probability functions have improved accuracy in the extreme tails and contains log-distribution functions.

More Robust Clustering

Hierarchical cluster analysis in version 5 supports color (heat) maps and two-way clustering (see Figure 4). K-Means clustering supports biplots and self-organizing maps.

Easier Access to Data

ODBC (Open DataBase Connectivity) is now fully implemented in both Windows and Macintosh operating systems. This implementation includes the ability to read and write Excel and SAS files. Any database driver installed on a local machine can be accessed through JMP.

In addition, the Windows edition of version 5 can now access files stored on a web server. This gives you the ability to import web pages and files as JMP data tables.

More Ways to Present Results

JMP now offers a complete set of drawing tools for lines, ovals, polygons, and other simple shapes. Combined with its layout capabilities, output can be formatted to almost any specification.

More Customizing

The Windows edition of version 5 introduces a drag-and-drop menu customization interface. This enables users to tailor the interface to their own specifications, including adding commands of their own designs. Version 5 can also access DLLs (Dynamic Link Libraries) stored on the client machine.

A Maturing JSL

Version 5 expands coverage of the platforms' scripting interface. Some platforms now provide scriptable access to internal matrices. A new Picture data type expands the reporting capabilities. The script editor is enhanced to allow for fence matching and immediate scrolling to lines containing errors.

Better Because of Your Feedback

All the new features are generated from ideas culled from users. This quick overview of version 5 includes major additions and enhancements, but user feedback has also brought about many changes too numerous to be noted here.

Please continue to send your feedback to jmp@sas.com as we begin planning for future versions.

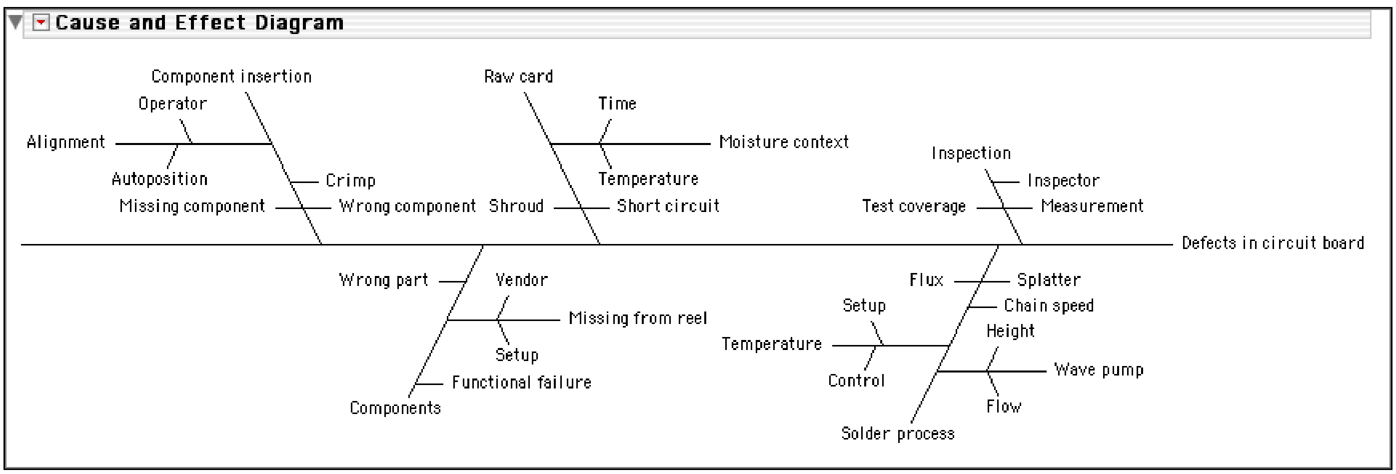


Figure 3: JMP version 5 supports Ishikawa, or cause-and-effect, diagrams.

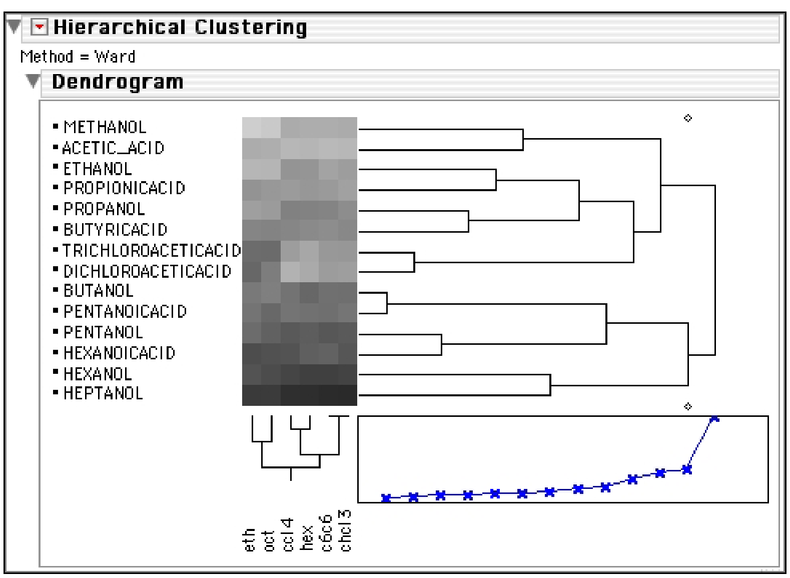


Figure 4: Hierarchical cluster analysis in version 5 includes color maps and two-way clustering.

Exploring Lack of Fit

Lee Creighton, JMP Development

Lack of fit is a topic covered in elementary linear models classes. Essentially, it is a statistical test that gives information on the form of the model under consideration. A significant lack-of-fit test suggests that there may be some systematic variation unaccounted for in the hypothesized model.

In JMP, a lack-of-fit test is presented automatically whenever there is an opportunity. This opportunity arises when there are exact replicate values of the independent variable in the model. These exact duplicates provide an estimate of pure error. Pure error is, in essence, the amount of error that cannot be accounted for by *any* model. Replication allows the estimation of this error, which then allows a test on whether there is error present aside from pure error.

One way to think about the construction of the lack-of-fit test is to examine three common types of linear models:

- single mean (one parameter)
- slope and intercept (two parameters)
- separate means for each x-value (many parameters)

The two-parameter model is the common regression model. Similarly, the many-parameter model is the one-way ANOVA model. The F-tests that appear in their ANOVA tables compare each with the single-mean model.

The single and many-parameter models are shown in Figure 5. They use the file named Big Class in the folder called Sample Data that was installed with JMP. This model fits the height variable to the age variable.

The ANOVA table (Figure 6) reveals the relevant sums of squares. The sum of squares from each mean to the data points is represented in the Error row of the table. The C. Total line shows the sum of squares from the single mean to the each data point. Since there are six groups, the hypothesized model adds five parameters to the single-mean model (therefore five degrees of freedom for Model). The F-ratio compares the explanatory power when increasing from one parameter to six parameters, and is in this case significant.

A similar explanation is valid for the two-parameter regression model. The graph in Figure 7 shows the single mean and regression model.

The ANOVA table (Figure 8) shows the relevant sums of squares. Notice that, as expected, the C. Total sum of squares (701.9) is equivalent in both ANOVA tables since it represents the same single-mean model. Also notice the decrease in the sum of squares as parameters are added to the model; 701.9 for the one-parameter model,

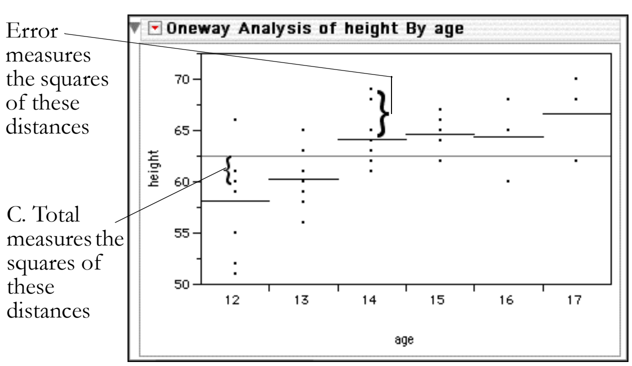


Figure 5: Single and many parameter fits.

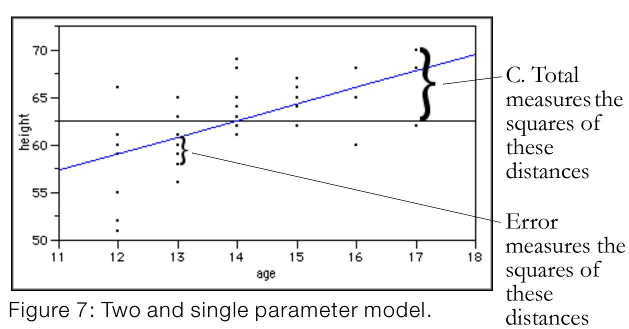


Figure 7: Two and single parameter model.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	5	312.88214	62.5764	5.4692	0.0008
Error	34	389.01786	11.4417		
C. Total	39	701.90000			

Figure 6: ANOVA table for many versus single means.

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	259.68935	259.689	22.3156
Error	38	442.21065	11.637	Prob > F
C. Total	39	701.90000		<.0001

Figure 8: ANOVA table for two versus single means.

442.21 for the two-parameter model, and 389.01 for the six-parameter model. The F-tests for each hypothesis can be thought of as in the illustration in Figure 9.

The lack-of-fit test is simply an F-test that compares the many-parameter model to the two-parameter model. The lack-of-fit table is produced automatically and is shown in Figure 10.

The numbers in the Sum of Squares column should look familiar. The degrees of freedom for the full model (labeled Lack of Fit) represent the increase in the number of parameters from the base model (two parameters) to the full model (six parameters). The remaining sum of squares is that which is due to replication and cannot be accounted for in any model. The base model (two-parameter) has 38 degrees of freedom, as seen in the regression ANOVA table in Figure 10. This example shows a lack-of-fit test that is not significant, indicating that additional parameters are unnecessary.

Therefore, the lack-of-fit test examines whether there is a systematic component of the separate-mean (six-parameter) model error that is not accounted for by the regression (two-parameter) model. See Figure 9.

Both the lack-of-fit and standard regression tests are available in JMP's Bivariate report, appearing when **Fit Line** is selected. The one-parameter model is also available in the Bivariate platform by selecting Fit Mean.

The standard ANOVA F-test appears in Oneway reports. Figures 5-13 were obtained by changing the modeling type of age from ordinal to

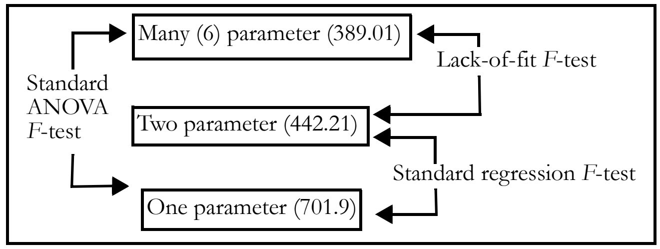


Figure 9: Illustration of the F-tests for each hypothesis.

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	4	53.19279	13.2982	1.1623
Pure Error	34	389.01786	11.4417	**Prob > F**
Total Error	38	442.21065		0.3446
				Max RSq
				0.4458

Figure 10: Lack-of-fit ANOVA table.

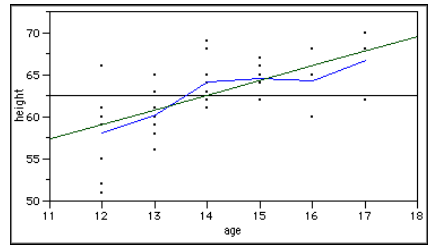


Figure 11: Regression with Fit Each Value.

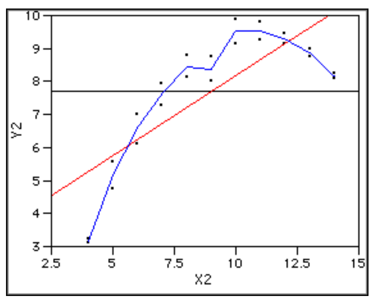


Figure 12: Significant lack of fit.

Separate Means ANOVA Table

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	80.559622	8.05596	43.8670
Error	11	2.020098	0.18365	**Prob > F**
C. Total	21	82.579720		<.0001

Regression ANOVA Table

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	51.135506	51.1355	32.5246
Error	20	31.444214	1.5722	**Prob > F**
C. Total	21	82.579720		<.0001

Lack-of-fit ANOVA Table

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	9	29.424115	3.26935	17.8025
Pure Error	11	2.020098	0.18365	**Prob > F**
Total Error	20	31.444214		<.0001
				Max RSq
				0.9755

Figure 13: Tables for significant lack of fit.

continuous, then re-running the analysis. It would be convenient, however, if this full model could be obtained without changing the modeling type.

This is where the **Fit Each Value** command comes in. It fits the full separate-means model within the Bivariate platform (see Figure 11).

The **Fit Each Value** command fits the same model as the full ANOVA. Instead of drawing lines at each mean, **Fit Each Value** connects the means, allowing lack of fit to be interpreted visually. In this case, the fit of each value is fairly close to the regression line, visually confirming the insignificant lack-of-fit F-test.

An example with a significant lack-of-fit is in Figure 12. It has eleven distinct values of the x variable.

Finally, note in Figure 13 the **Max RSq** appended to the lack-of-fit table. Since there are replications in the x-values, there is no way that the value of R^2 can equal 1 (unless there is zero variation in the replicated values). This is exactly the R^2 that is obtained when fitting the separate means model, 0.975 in this example.

Creating a Custom Report in JSL

Lee Creighton, JMP Development

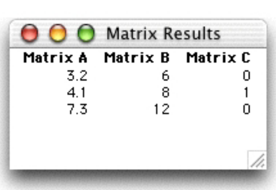
JMP Scripting Language (JSL) allows you to write scripts that automate repetitive and frequently-used tasks. In addition to writing scripts, you can write custom statistical analyses, extending JMP's abilities. The possibilities are as vast as the imagination of the user.

One thing that many custom statistical analyses have in common is their presentation of results. Without an efficient report, analysis results lose impact and usability.

When creating JSL output, display trees are built out of JMP's display boxes. These boxes allow presentation of text, matrices, and lists in ways that are similar to JMP's built-in reports.

As an example, the following script initializes several matrices and displays them in a new report window.

```
a=[3.2, 4.1, 7.3];
b=[6, 8, 12];
c=[0, 1, 0];
New Window("Matrix Results",
    HListBox(
         Number Col Box("Matrix
         A", A),
         Number Col Box("Matrix
         B", B),
         Number Col Box("Matrix
         C", C)
    )
)
```



Matrix A	Matrix B	Matrix C
3.2	6	0
4.1	8	1
7.3	12	0

This code illustrates some typical JSL concepts for display boxes. The window is created using the `New Window()` function, and it contains an `HListBox()` that arranges its child boxes horizontally. The three columns of the report are made from the three `Number Col Box()` calls.

Unknown Matrices

This code works well for cases where the dimensions of the matrices are known in advance to the JSL scripter. However, in many cases of custom scripting, a generic, reusable script is needed. How do you prepare a report full of display boxes when you don't know how many are needed?

The solution is JSL's `Append` command. If you don't know how many display boxes are needed for a particular report:

1. Create an empty display box in the script.
2. Send an `Append` message to this empty box for each needed display box.

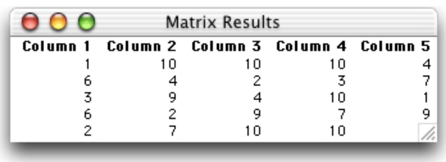
As an example, the following script constructs a matrix of random numbers, then displays them column by column in a new window.

```
//number of rows & columns
n=5;

//a holds the matrix
a=j(n, n, RandomInteger(10));

//construct an empty box
New Window("Matrix Results",
    hb=HListBox()
);

//This for loop adds a column
for each effect
for(i=1, i<=NCol(a), i++,
    hb << append(
         Number Col Box(
              "Column"||char(i),
              a[0,i]
         )
    )
);
```



Column 1	Column 2	Column 3	Column 4	Column 5
1	10	10	10	4
6	4	2	3	7
3	9	4	10	1
6	2	9	7	9
2	7	10	10	

Note the empty New Window contains nothing but a reference to an `HListBox`. The empty box is named `hb`, and the for loop simply sends messages to `hb`—in this case, `Append` messages. The building of the boxes has nothing to do with the number of columns of `a`. Simply change the first line of the script by setting `n` equal to the number of columns that a should contain. The report is then built dynamically to include each column, one by one.

Using this technique, you can now build reports on the fly, without knowing beforehand how many elements you need to display. It is essential when writing scripts that are intended to be used for general purposes.

When a Factor is Also a Response

Mark Bailey, Statistical Training and Technical Services

Suppose you want to find a given level of acidity for a vinegar within a narrow range of acceptable pH levels. You might even have more than one response to satisfy, each with its own goal and importance. The number and perhaps diverse nature of the responses may pose as a daunting task.

However, these requirements can be entered into JMP along with data from the experiment. It can then be incorporated into the search for the best factor settings.

Several JMP exploratory tools, such as the Prediction Profiler, use the model derived from an analysis to show all of the responses in parallel. The Prediction Profiler can also use the model specification information to find the best settings across all requirements. This article does not discuss the design or treatment structure of the study. Rather, it focuses on the analysis of data from an experiment.

Let's take our example a step further and suppose there is a factor that you want to minimize for reasons such as limited availability, high expense, difficulty in control, or risk to safety or the environment. How do you include this requirement in a search for the best operating conditions?

To get to this answer, consider the following problem adapted from an example by Box, Hunter, and Hunter (1978). A chemical process converts a single reactant into a single product. The reaction is catalyzed both for rapid conversion and for product specificity. One response is the yield of the reaction. High yields are desirable, but when yields are above 90% conversion, secondary reactions become significant and produce undesirable side products. A minimum of 80% yield is required. A second response is the cost of the reaction run. A goal of $375 per run may be a stretch, but if the run costs above $425, the process is not economically feasibility. The cost of a process run is twice as important as the yield.

Imagine that you want to find the best reaction conditions as defined by these four factors:

- Concentration—amount of reactant (10-12 pounds)
- Catalyst—amount of catalyst (10-15 percent by weight)
- Temperature—reaction temperature (220-240 degrees Fahrenheit)
- Pressure—the reaction pressure (50-80 pounds per square inch)

The catalyst for this reaction is a metal compound that is costly, difficult to obtain, and poses a problem for proper disposal. This problem is so severe that it is considered three times as important as yield. You must find the best conditions for reaction yield and costs but at the same time minimize the utilization of this material. How can this requirement be included in your search?

This additional requirement, like the first two goals, does not affect the design of the experiment. It is a special consideration for the analysis and prediction.

Suppose that you have data from a complete Central Composite Design for four continuous factors in 31 runs (Figure 14).

During the design of this experiment, the goals for Yield and Cost were saved as **Response Limits** column properties. Likewise, the Concentration, Temperature, Pressure, and Catalyst were saved with the **Coding** and **Design Role** properties.

You should add a new column that has the same values as the Catalyst column. You can create this new column (call it Utilization) using a formula that copies the values of Catalyst. Or, paste the values of Catalyst into the new column. In the new column's Column Info dialog, enter the column property (**Response Limits**), set the goal to **Minimize**, and

	Pattern	Concentration	Temperature	Pressure	Catalyst	Comment	Yield	Cost
1	----	10	220	50	10	FF	61.8	402
2	---+	10	220	50	15	FF	58.9	398
3	--+-	10	220	80	10	FF	69.9	391
4	--++	10	220	80	15	FF	61.8	441
5	-+--	10	240	50	10	FF	86.9	382
6	-+-+	10	240	50	15	FF	73.8	396
7	-++-	10	240	80	10	FF	88.8	403
8	-+++	10	240	80	15	FF	80.9	427

Figure 14: Partial listing of the chemical.

set **Importance** to 3 (Figure 15).

To begin the optimization:

1. Select **Analyze** > **Fit Model**.
2. Select the three response columns and give them the **Y** role.
3. To build all effects in one step, select the four factors. Then, from the **Macros** menu on the Fit Model dialog, select **Response Surface** (Figure 16).

Note that you can explore the variables using the Distribution or Spinning Plot platforms. You can also check of the model assumptions using row diagnostics.

---

4. The traces added to the right side of the profiles is a graphical representation of the goals that you entered.
5. Select **Maximize Desirability** from the menu on the Prediction Profiler's title bar.

The results are shown in Figure 17. Note that all three goals were achieved. This optimizer determines the best trade-off given the general or specific requirements for the entire set of responses. In this way, you have minimized the utilization of one of the factors while attaining the levels of yield and cost performance within your specifications.

An Alternative Scenario

There are cases where you might not want to use the approach described above. For example, suppose you want to fit a separate model for each response. Or, suppose you have no data but you have a model from previous experience or a theoretical model based on the factors that you want to use alongside the empirical models. The Prediction Profiler can be used outside the Fit Model platform to explore or optimize a response column with a formula.

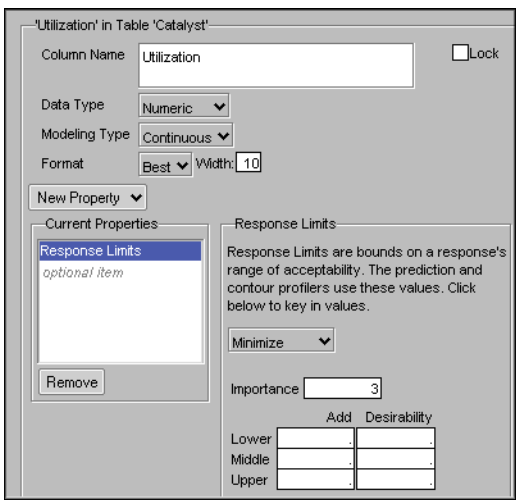


Figure 15: The Column Info dialog.

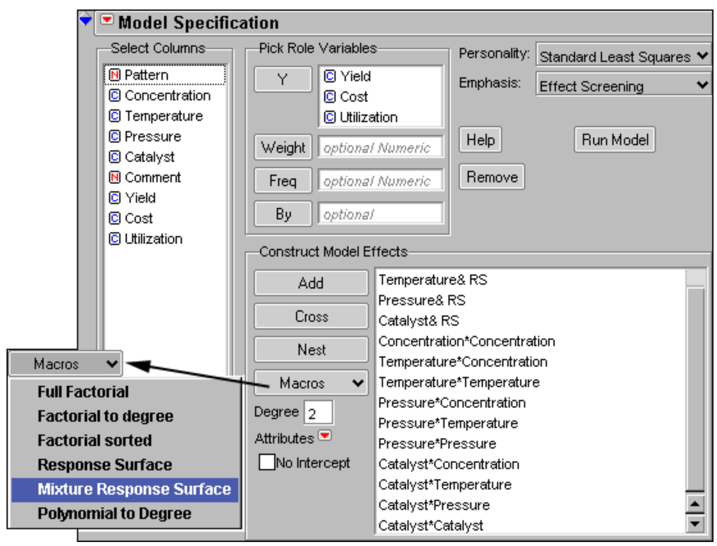


Figure 16: The Macros menu on the Fit Model dialog.

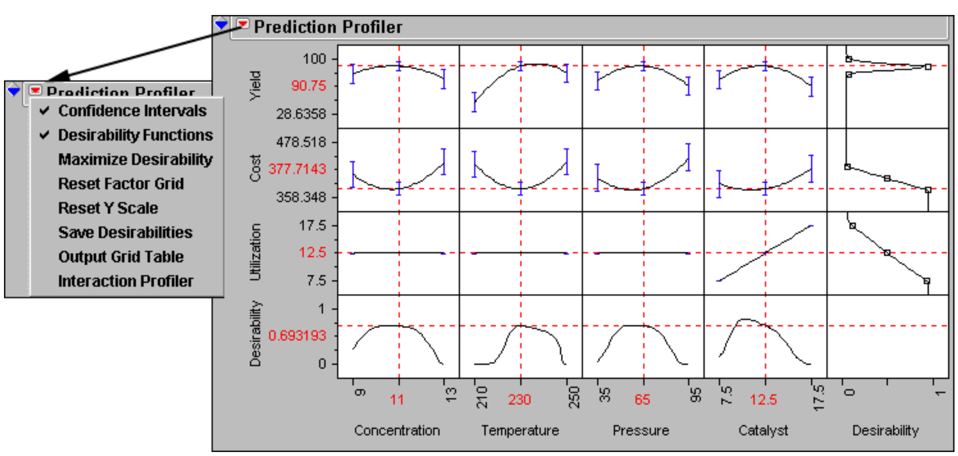


Figure 17: Successful results from the Prediction Profiler.

In our example, if Yield and Cost were analyzed separately, use the **Save Prediction Formula** command in the **Save Columns** submenu to save each model to the data table. Then select **Profiler** from the **Graph** menu and specify Utilization, Pred Formula Yield, and Pred Formula Cost as *y* variables. From this point, the Prediction Profiler works the same way as it did from within the fitting platform.

The Prediction Profiler can be used as an integral part of your data analysis and modeling, or separate from the initial fit.

<u>References</u>

"Statistics for Experimenters." George Box, William Hunter, and Stuart Hunter. John Wiley & Sons: New York. p. 324-329 (1978).

Look for JMP at these 2002 Trade Shows

June 12-13	Quality Expo Detroit in Novi, MI
June 16-18	PNWSUG in Portland, OR
July 15-19	Macworld East in New York, NY
July 21-25	IAAFSC in Quebec City, Canada
August 11-15	ASA in New York, NY
August 18-22	ACS National Meeting and Exposition II in Boston, MA
September 4-6	WUSS in San Diego, CA
September 22-24	SESUG in Savannah, GA

Tips and Techniques

Using JMP to Update Values

Ann Lehman, JMP Development

In days gone by, database maintenance systems updating transactions used basic batch-mode language commands to add, delete, or update rows in a table.

Today, JMP makes updating values easy with the **Join** command. By selecting **Tables** > **Join**, you can update values in a table with the values from a second table.

As an example, consider the table on the left in Figure 18, which was taken from the file named Big Class.jmp in the folder named Sample Data installed with JMP. Suppose you have an updated data table where age has changed for rows 5, 10, and 13. This is shown on the right in Figure 18.

	name	age	sex
5	LILLIE	12	F
6	TIM	12	M
7	JAMES	12	M
8	ROBERT	12	M
9	BARBARA	13	F
10	ALICE	13	F
11	SUSAN	13	F
12	JOHN	13	M
13	JOE	13	M

	name	age
1	JOE	14
2	LILLIE	13
3	ALICE	14
4	LOUISE	•
5	BILLIE	12

Figure 18: We will join the outdated Big Class.jmp table (shown on left) and the updated table (shown on right) into a new table (shown in Figure 21).

One way to update the values in Big Class.jmp is to create a new data table that contains the values in Big Class.jmp combined with those in the updated table using the **Join** command. To do this:

1. Choose **Tables** > **Join**.
2. Check **Update first table with data from second table** (see Figure 19).
3. Enter a name for the new table in the **Output Table** box (see Figure 19).

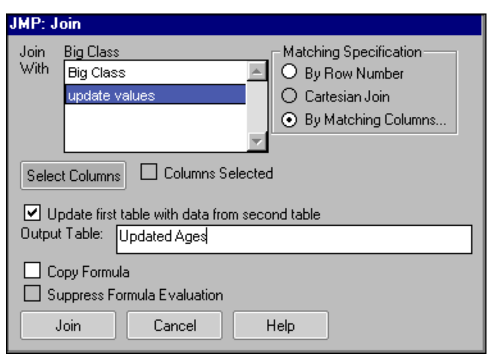


Figure 19: By checking the option "Update first table with data from second table," you create a new table.

4. Click the **By Matching Columns** button. The Match Columns dialog appears (see Figure 20).

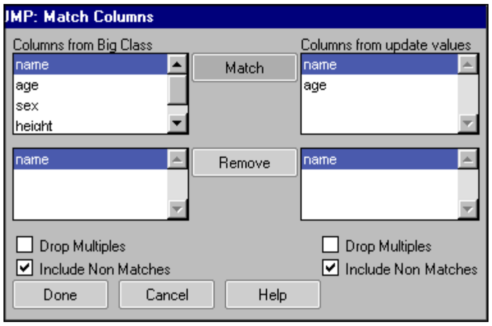


Figure 20: The Match Columns dialog lets you choose which columns to match.

5. Select columns in the two tables whose values match, then click the **Match** button. You can have as many matching columns as necessary to identify matching rows.
6. To include all original rows in the updated table, check both of the **Include Non Matches** boxes.
7. Click **Done**, then click **Join**.

	name	age	sex	height	weight
1	ALFRED	14	M	64	99
2	ALICE	14	F	61	107
3	AMY	15	F	64	112
4	BARBARA	13	F	60	112
5	BILLIE	12		•	•
6	CAROL	14	F	63	84
18	JEFFERY	14	M	69	113
19	JOE	14	M	63	105
20	JOHN	13	M	65	98
26	LEWIS	14	M	64	92
27	LILLIE	13	F	52	64
28	LINDA	17	F	62	116
29	LOUISE	12	F	61	123

Figure 21: The new table with updated values.

Your updated table (shown in Figure 21) should have the following properties:

- The new table is in alphabetic order by the matching variable. However, sorting is not a requirement of either the outdated or updated table.
- The missing age value for Louise in the updated table left the outdated value unchanged.
- The unmatched name, Billie, in the updated table caused a new row to be added in the new table.

JMP Information Briefing

You're invited to a free session on May 13, 2002 from 9-11:30 am or 1-3:30 pm at SAS Greenwood Village, Colorado. At this session, we'll discuss how JMP can help you integrate statistical thinking into process improvement and product design and development.

To register, contact Kathy Jablonski at kathy.jablonski@sas.com or (919) 531-4014.

DOE News

Latin Square Designs

Textbooks show the Latin Square experimental design as a two-way schematic of three factors. The level of the first factor is the same for each row while the level of the second factor is constant for each column. The level of the third factor (often a 'treatment') is shown as an element of the table and occurs only once in each row and only once in each column.

The designation of 'square' in the name Latin Square implies that the factors have the same number of levels, giving the dimension of the Latin Square. There are many variations of latin squares for any dimension.

	b1	b2	b3	b4
a1	c1	c3	c2	c4
a2	c3	c4	c1	c2
a3	c2	c1	c4	c3
a4	c4	c2	c3	c1

Figure 22: Example of a familiar four-by-four Latin Square.

	A	B	C
1	a1	b1	c1
2	a1	b2	c3
3	a1	b3	c2
4	a1	b4	c4
5	a2	b1	c3
6	a2	b2	c4
7	a2	b3	c1
8	a2	b4	c2
9	a3	b1	c2
10	a3	b2	c1
11	a3	b3	c4
12	a3	b4	c3
13	a4	b1	c4
14	a4	b2	c2
15	a4	b3	c3
16	a4	b4	c1

Figure 23: Example of a four-by-four Latin Square as a series of experimental runs in a JMP data table.

Creating a Latin Square with the JMP Custom Designer

Let's use the JMP Custom Designer to create a design similar to the one in Figure 23. Select **DOE** > **Customized Design** and complete the dialog by adding three four-level categorical variables with names and values as shown in Figure 24.

Note: You can produce the same type design by using two four-level categorical variables and one blocking variable with four runs per block, or one four-level categorical variable and two blocking variables with four runs per block.

When you click Continue in the DOE dialog, the default number of runs shows as 16, or 4^2, which is correct for the four-dimensional Latin Square defined by the number of levels of the categorical variables. See Figure 25 for details.

When you click Make Design, then click Make Table in the DOE dialog, JMP creates 16 runs that are a variation of the four-by-four design shown in Figure 22.

Rearranging the JMP Design Table

If you want to see a standard Latin Square, rearrange the design table produced by the DOE facility. Choose **Tables** > **Split** and complete the Split Columns dialog as is shown in Figure 26. You should see the following changes:

- The values of Split variable (C) become the cells of the new table.
- The values of the Col ID variable (B) label the new columns.
- The grouping variable (A) becomes the first column in the new table. Its values serve to identify the rows of the Latin Square.

To reverse this square arrangement and reconstruct the original table of runs, use **Table** > **Stack** and stack the b1, b2, b3, and b4 columns.

Using Higher Dimensions

The number of levels or runs per block defines the dimension of the Latin Square design. You can use the Custom Designer to produce designs with more than four levels, but note that it might require more 'starts' than occur by default.

The Custom Designer begins the design search process with a random starting design. It then iteratively improves this design until no further improvement is possible. However, the resulting design is not guaranteed to be globally optimal. So, the Custom Designer repeats this process several times using different random starting designs. The design finally chosen is the best of all these locally optimal designs.

The default number of random starts varies as a function of the specified factors and the model. However, you can set the number to random starts available for the Custom Designer by submitting the following JSL statement before

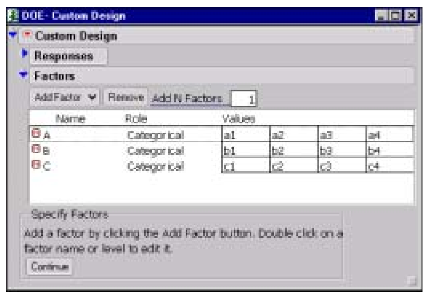


Figure 24: Add three four-level categorical variables in the custom design dialog.

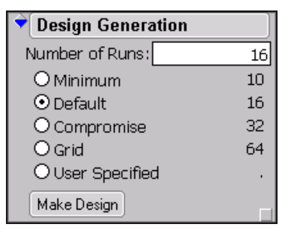


Figure 25: The default number of runs in this example is 16, or 4^2.

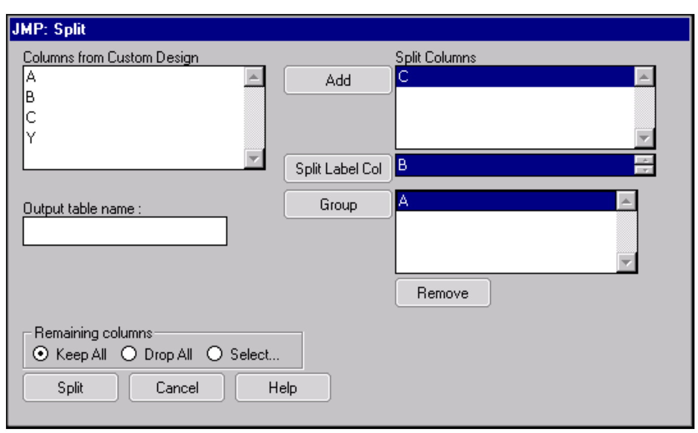


Figure 26: The split columns dialog rearranges the run table.

defining a design:

```
DOE Starts = nnn
```

This statement creates a global variable, DOE Starts, that defines the maximum number of random starts that the Custom Designer will use, if needed. This global variable value remains fixed until another JSL command changes it or you restart JMP.

As an example, submit this JSL statement:

```
DOE Starts = 400
```

Then use the Custom Designer to create a five-level Latin Square. As you increase the number of starts, the processing time to create the design increases. For larger Latin Square designs, the time is sometimes prohibitive and the result is not guaranteed to be a Latin Square even though it's always an optimal design.

It is simple to create any size Latin Square using the following JSL code. You only need to set the number of levels (N Levels).

```
dt = new table("Latin Square");
// Change N Levels to the desired
number of levels per factor.
N Levels = 5;
dt << Add Multiple Columns("C",N
Levels,First,Numeric);
dt << Add Rows(N Levels);
for (column index=1,column
index<=N Levels,column index++,
    c = Column(column index);
    for (row index=1,row index<=N
Levels,row  index++,
    value = row index + column
index - 1;
    if (value>N Levels,value -= N
Levels);
    c[row index] = value;)
```

About JMPer Cable

Issue 9 Spring 2002

JMPer Cable is mailed to JMP users who are registered users with SAS Institute. It is also available online at www.jmpdiscovery.com.

Contributors

John Sall

Lee Creighton

Ann Lehman

Mark Bailey

Editor and Designer

Meredith Blackwelder

Printing

SAS Institute Print Center

Questions or comments

jmp@sas.com

To order JMP software

1-877-594-6567

For more information on JMP

1-877-594-6567

www.jmpdiscovery.com

JMP
SAS Campus Drive
Cary, NC 27513 USA
Tel: (919) 677-8000