A Technical Publication for JMP® Users
Summer 2004 Issue 15

JMPer Cable

Inside This Newsletter

Breaking News

JMP User Conference
August 30-31, 2004
http://support.sas.com/jmpconf

Live Web course "JMP Software: Statistical Data Exploration"
September 13, 2004 1:00 pm - 4:30 EDT. http://support.sas.com/training/us/crs/lwjdex5.html

Free Webinar: "JMP: Preparing Graphics for Presentation"
August 16, 2004
http://www.jmp.com/news/webinars.shtml

Free Webinar: "JMP and SAS Performance Management for Six Sigma in Health Care"
September 15, 2004
http://www.jmp.com/news/webinars.shtml

Leptokurtosiphobia: Irrational Fear of Non-Normality

John Sall, Executive Vice President, SAS Institute Inc.
Bradley Jones, JMP Development

Every statistical procedure involves assumptions. We are taught to fear the violation of these assumptions. For example, the standard procedures for comparing means, the two-sample *t*-test, the paired *t*-test, and ANOVA all include the assumption that the residual error is normally distributed. If the error is normally distributed, and all the other assumptions hold, then all the test statistics have their claimed distributions, and confidence intervals and hypothesis tests can be trusted. We even have a rich source of analytic tools to probe our assumptions, such as the features in the distribution platform which test and graph against the normal distribution. What if the residuals are not normally distributed? How much should a practitioner worry about whether the residuals are normal? Few articles, and fewer books, go into any detail on how much to worry about the normality assumption.

Having normally distributed residuals is an issue we probably don't need to worry much about because:

- In large samples, it is easy to detect non-normality, but it doesn't matter.
- In small samples, non-normality may matter, but you can't detect it.

With large samples, the central limit theorem states that even if the data is not normal, mean-like statistics approach normal distributions as the sample size increases. With small samples, the test statistic is not necessarily normal, but we don't have a big enough sample to tell.

JMP's developers get frequent requests to implement a test for residual normality as a convenience feature for our fitting platforms. But this would only be useful if there were sample sizes for which non-normality could be detected and mattered. Let's investigate, using simulation.

For our simulation we compare the means of two groups using two different non-normal error distributions: a double exponential and a shift-contaminated normal. The double exponential is symmetric, but leptokurtotic, *i.e.*, much more peaked and having much heavier tails than the normal distribution. The shift-contaminated normal is a normal distribution skewed to the right by adding a shift of three standard deviations in a randomly-selected 5% of the samples.

The double exponential is simulated by this JMP formula:

```
if(randomUniform()>0.5,1,-
1)*log(randomUniform())
```

The shift-contaminated distribution uses this JMP formula:

```
10+5*(randomNormal()+
(randomUniform()<0.05)*3)
```

These are meant to be characteristic of strong, but not extreme, non-normality you might see in real data.

Figure 1 shows 10,000 points of the simulated double-exponential and Figure 2 shows the shift-contaminated distributions. The solid red diagonal line is a reference for how the data would look if it fit a normal distribution. Note that the data fall below the line for small quantiles and above the line for large quantiles. This is characteristic of leptokurtotic distributions. They have plenty of probability mass in the middle with long but sparsely populated tails.

The data in the normal quantile plot for the shift-contaminated distribution (Figure 2) fall above the solid red diagonal line for large quantiles. This is characteristic of right skewed distributions. Had the data been normally distributed, the top of the histogram would have ended around 30 instead of continuing a bit past 40.

A JSL script was run that generated 10,000 Monte Carlo samples with 10 observations for these two non-normal distributions. The script (MonteCarlo.jsl) can be found at http://www.jmp.com/news/jmpercable. Each sample was divided into two groups of equal size. By design, there is no difference between the means of the two groups. We also generated 10,000 samples for the standard normal distribution as a control.

For each sample, we ran a oneway analysis to test the difference of the means and a distribution to test the normality of the residuals. We repeated this entire procedure for samples of size 20, 30, 50, 70, and 100.

Test For Equal Means When There is no Difference in the Population Means

First, let us look at the plots of our 10,000 *p*-values for the non-normal distributions. Remember that our procedure is designed so that there is no true difference in the means for each sample. Because of random variation, the *p*-values for our 10,000 oneway analyses should be uniformly distributed between zero and one. The uniform distribution function is the diagonal line F(x) = x.

To produce the plots in Figure 3, we first sorted the 10,000 *p*-values. The

(continued on page 13)

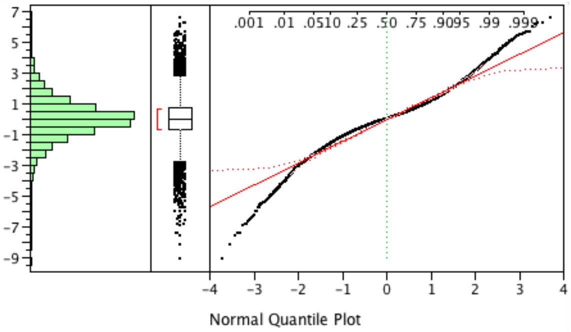


Figure 1: Distribution results for the double-exponential distribution.

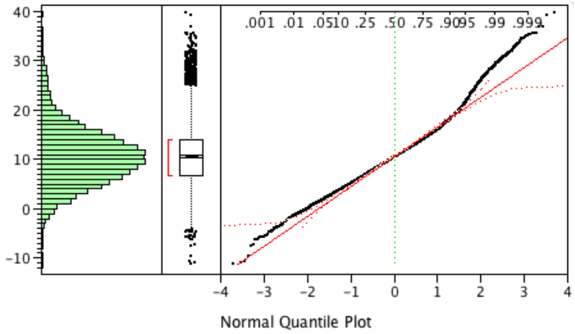


Figure 2: Distribution results for the shift-contaminated distribution.

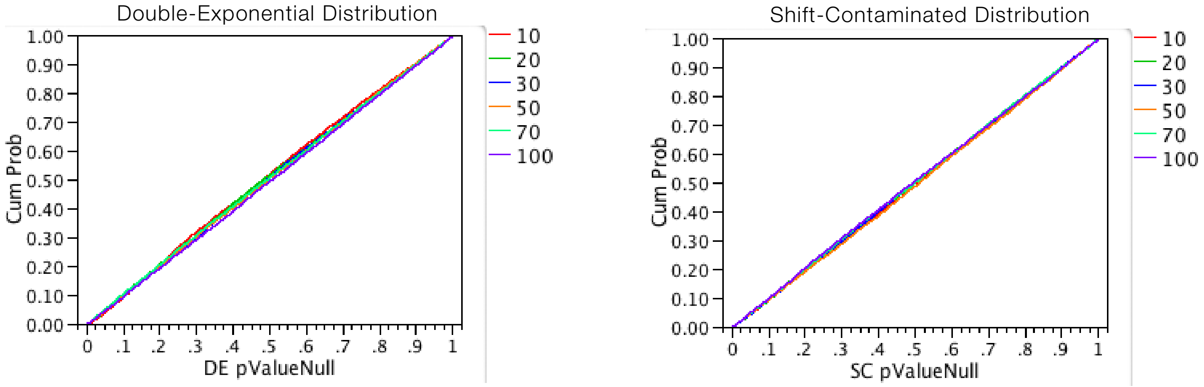


Figure 3: Probability plots of *p*-values for the double-exponential distribution (left) and shift-contaminated distribution (right).

Using Live Data Feeds with JMP

Meredith Blackwelder, JMP Development

The term live data feed describes the way an external data source sends information via a physical or a logical communication link to another device. You can connect JMP to a live data feed through the serial port of your Windows or Linux computer in order to read, in real-time, a stream of incoming data. The following example explains the practicality and effectiveness of connecting JMP to a data feed and offers a way to see this happen in JMP, even if your computer isn't hooked to a data feed.

An Example

On a factory floor, steel nuts are constructed so that their inside diameters are 8 mm plus or minus ½ mm. The quality control manager at the plant knows that if the diameter is too big or too small, the nuts will not fit properly onto his customers' bolts.

The manager has set up a caliper to measure the diameter of every tenth nut coming off the line. This caliper is connected to a computer running a JMP Scripting Language (JSL) script (see later in this article for a sample script). When a worker clicks a button on the caliper, it sends a measurement through a data feed into JMP (Figure 9). To avoid having a worker press a button for each measurement, the company could invest in a more high-tech instrument, such as a laser, which automatically measures and sends data.

Figure 9: Example caliper connected to computer running JMP.

The JSL script tells JMP to capture the data, store it in a data table, and simultaneously chart it on a control chart (Figure 10). By watching the data as it enters the data table and appears on the chart, the quality control manager can immediately see if the nut is "out of control"—that is, how far off its measurements are from the "ideal" measurements.

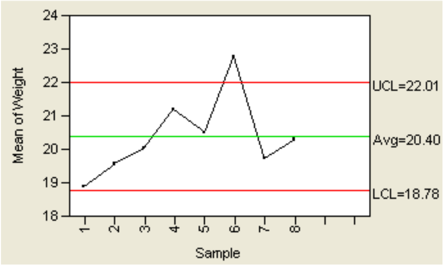


Figure 10: Example control chart showing data that is out of control.

Being able to see, in real-time, how consistently nuts are manufactured also provides the capability of observing the environmental effects during the nut-creation process, such as temperature and machine settings. As soon as workers see data points on the control chart that are out of control or inconsistent with the others, they can take note of these environmental effects as they are happening. This helps plant managers investigate the special cause of the problem, and makes correcting the problem easier.

Setting Up Your Own Data Feed

If you have a live data feed and aren't sure how to write JSL to direct the data to a data table, use the following script to set up the feed and gather data in JMP (you can also download this script, named DataFeed.jsl, at http://www.jmp.com/news/jmpercable.):

```
streamScript = expr(line = feed
<<Get Line;show(line);
            len = length(line);
show(len);
   if (length(line)>=1,
show("Hi"); show(line);
                    field = sub-
str(line,5,8); show(field);
        x = Num(field); show(x);
         if (!IsMissing(x), cur-
rent data table()<<add
   row({:Column1=x});
            show(x);
   )));
feed = open DataFeed();
 feed<<Set Script(streamScript);
feed<<Connect(Baud
Rate(4800),parity(even),Stop
bits(2), Data bits(7));
```

Connect the data feed to your computer, then open and run the script. Remember that:

- The data feed must come through a standard nine-pin serial port. Data cannot be read through a USB port.
- You need to know the exact baud rate, parity, stop bits, and data bits for the attached device. Once you obtain these numbers, enter them into the `open Datafeed` command in the script shown above.

Simulating a Data Feed

If you don't have access to a data feed and you want to see how it works, you can simulate JMP's ability to capture real-time data using the DemoLiveDataFeed.jsl script, which you can download at http://www.jmp.com/news/jmpercable (or, go to www.jmp.com, mouse over **Downloads**, and click **JMP Script Library**).

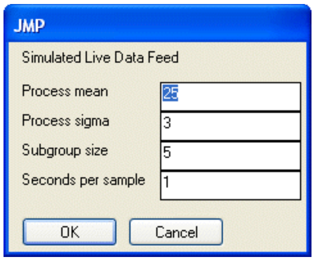
This script simulates opening a feed and watching new data appear. A source is simulated by using a random number generator to put values into the data stream. When data appears, JMP captures it and puts it in a new row of a data table. JMP then opens two control charts (an Xbar and r chart) on this data column, so as new data appear they are automatically charted. An example of all open windows is shown in Figure 11.

In addition to opening the data table and the demonstration window with control charts, the script opens two more windows: the Datafeed window and the Scheduler window. The Datafeed window shows you the current status of the data feed, and the Scheduler window shows the current status of any queued events.

Running the Demo Script

To see this demonstration for yourself, download, open, and run DemoLiveDataFeed.jsl. First, a dialog appears (Figure 12) with text fields for you to specify:

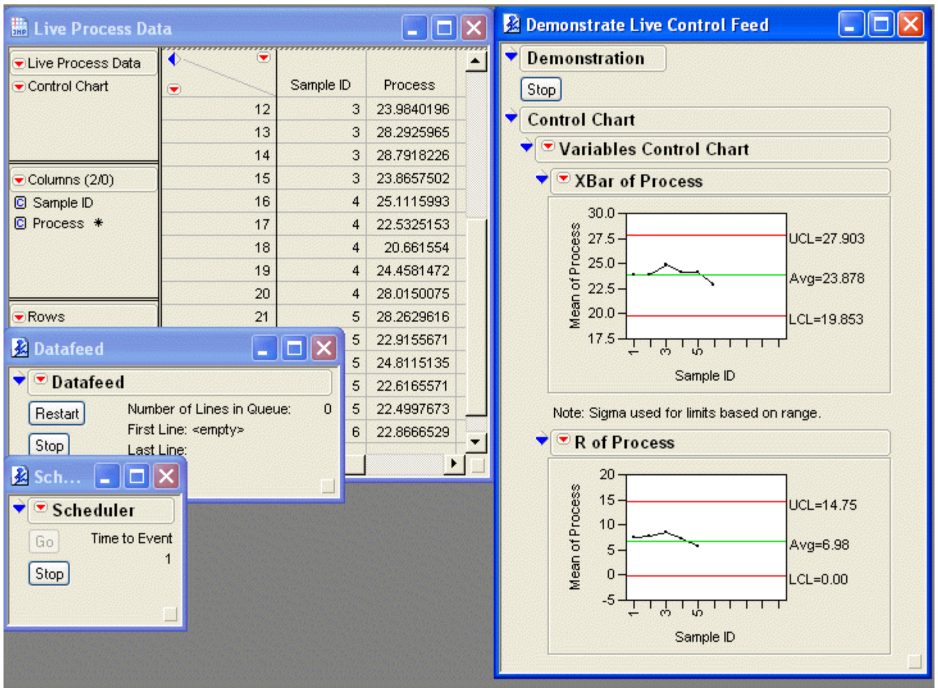
- Process mean—The average size you expect the part to be.
- Process sigma—The variance you expect the parts to have.
- Subgroup size—How many samples you want in each batch. This number should not be less than 2.
- Seconds per sample—How fast to run the samples. The default speed adds data values one per second. If you decrease this number, the values are plotted on the chart quicker. If you increase this number, the values are plotted slower.



Simulated Live Data Feed

Process mean	25
Process sigma	3
Subgroup size	5
Seconds per sample	1

Figure 12: The defaults in this dialog offer appropriate initial values for all parameters, so click **OK**.



	Sample ID	Process
12	3	23.9840196
13	3	28.2925965
14	3	28.7918226
15	3	23.8657502
16	4	25.1115993
17	4	22.5325153
18	4	20.661554
19	4	24.4581472
20	4	28.0150075
21	5	28.2629616
	5	22.9155671
	5	24.8115135
	5	22.6165571
	5	22.4997673
	6	22.8666529

Number of Lines in Queue: 0
First Line: <empty>
Last Line:

Time to Event
1

Note: Sigma used for limits based on range.

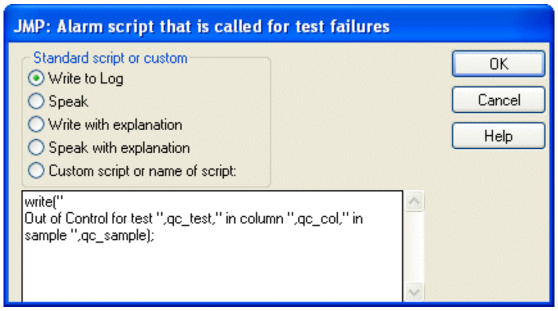
Figure 11: The demonstration script opens four windows.

After you click **OK**, data begins to enter JMP. The control charts update for each point entered. The mean and range update for each subgroup until all the values in the subgroup are present (this is apparent visually, since the most recent point seems to wobble a bit as new values are added), and then it starts on the next subgroup.

For this demonstration, refrain from clicking buttons in the Datafeed or Scheduler windows. To stop the script, click **Stop** in the demonstration window. When using the data feed in real life, buttons in the Datafeed and Scheduler windows give you the option to start and stop the feed and view the number of remaining samples.

In addition, JMP can trigger an alarm script if a data point is flagged as out-of-control. This alarm script can have the computer speak a warning, display an error message on the screen, send an email (useful especially with today's pagers and SMS-capable cell phones), or write messages to a log. The alarm signals the need for immediate action to fix the problem and minimize waste. For example, if you wanted to show management a report of which observations failed each test, turn the alarm feature on by following these steps:

1. Click the red triangle icon in the Variability Control Chart title bar and select **Alarm Script**.
2. Select **Write to Log** (see Figure 13), then click **OK**.



JMP: Alarm script that is called for test failures

Standard script or custom

● Write to Log
○ Speak
○ Write with explanation
○ Speak with explanation
○ Custom script or name of script:

```
write("
Out of Control for test ",qc_test," in column ",qc_col," in
sample ",qc_sample);
```

Figure 13: Select the action you want JMP to take to alert you of test failures.

3. Select **View > Log** from the main menu. The log will appear at the bottom of your screen.
4. After running your analyses, select the text in the log, copy it, and then paste it into a text document.

Viewing Control Charts and Distributions

The control charts that appear are live, and therefore fully functional. You can simulate any process variable that is a Gaussian (normal) distribution. Or, you can have them run special tests, assign alarm scripts, or execute any other commands in the red triangle menus.

For example, click the red triangle icon on the Xbar of Process title bar and select **Tests > Test 1**. You should see the violation of some tests, as shown in Figure 14. This chart shows what Deming (1982) refers to as special causes. According to Nelson (1984), "A special cause is fixed by removing the perturbing influence that caused the out-of-control signal."

Simulating an Out-of-Control Situation

Now let's take a look at a demo of a situation that is out of control. We'll use the script demoOCLiveDataFeed.jsl, which you can download at http://www.jmp.com/news/jmpercable. This script is almost identical to the previously-used one except the line

```
feed << Queue Line(Char( Random
Normal(m, s) ) );
```

is now

```
feed << Queue Line( Char( Random
Normal(m, s) - NRow() ) );
```

In other words, we have introduced a simulated drift in the process. The `Nrow()` function returns the number of rows in the current data table. This number is subtracted from the random normal data that we are generating. Naturally, as the number of rows increases, the larger this subtracted number will be. Subtracting larger numbers causes the results from the live data feed to decrease, thus causing the points on the control chart to drift farther and farther away from the center line (Figure 15).

If you have Test 2 turned on (click the red triangle icon on the Xbar of Process title bar and select **Tests > Test2**), you see points flagged as out of control, as shown in Figure 16. If this were a real process, something would be occurring to cause this drift in the mean—something that should be examined. According to Nelson (1984), the out-of-control points are

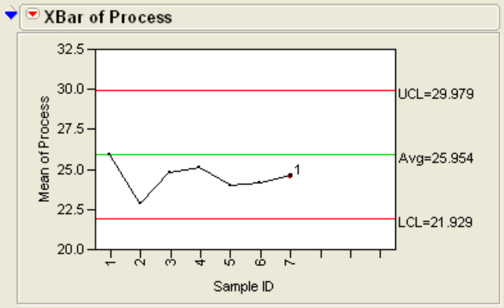


Figure 14: An XBar chart with Test 1 turned on.

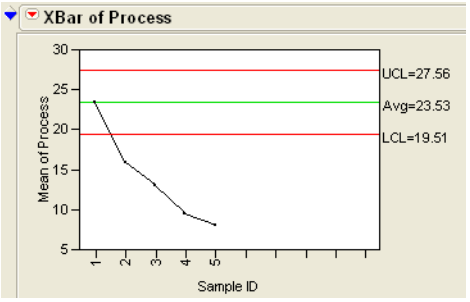


Figure 15: A control chart showing out of control data.

common causes: “A common cause is one that affects all the points on the chart, as when a centerline is too high. A common cause is fixed by changing the system.”

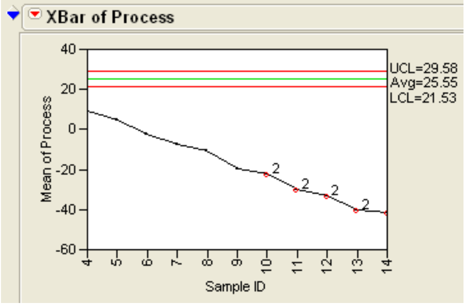


Figure 16: Test 2 flags points going out of control.

Conclusion

Use the scripts referenced in this article to help determine if a live data feed would be useful in your organization. Being able to see, in real-time, how consistently a product is being made gives you the advantage of observing the environmental effects during the creation process. JMP lets you take note of these environmental effects as they are happening. This real-time insight helps you investigate the special cause of the problem and makes correcting the problem easier.

References

Deming, W.E. (1982). *Quality, Productivity and Competitive Position*. Center for Advanced Engineering Study, Massachusetts Institute of Technology, Cambridge, MA.

Nelson, Lloyd S. (1984) Technical Aids: The Shewhart Control Chart—Tests for Special Causes. *Journal of Quality Technology*. Vol. 16, No. 4. pp 237-239.

Scripting Session

Combining Reports in JSL

Lee Creighton, JMP Development

A common question received by our technical support department is deceptively simple in its formation: “How do you combine reports from different platforms into a single window?”

In this article, I show you how to construct the report shown in Figure 17 from the Big Class.jmp data table. When you installed JMP, a folder named Sample Data was also installed. In that folder is Big Class.jmp. Note that the report shown in Figure 17 combines several analysis types, with standard report elements hidden, and that each graph is still “live”—you can click on the points or bars, and corresponding points highlight.

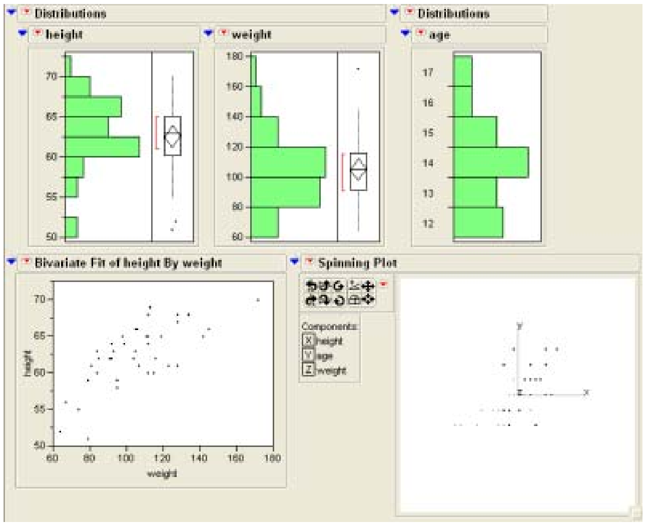


Figure 17: Combine many analysis reports into one.

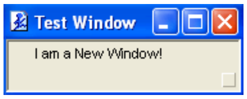
Remarkably, you only have to learn three JSL commands to generate this (and other) combined reports.

First, the `New Window` command. This command creates a new window, a container to hold the individual reports. The first of its two arguments is a string that names the window. The second argument—usually a long comma-separated list—is a list of the display boxes that populate the window.

A simple example of the `New Window` command might contain a single text box:

```
New Window("Test Window",
  Text Box( "I am a New Window!")
)
```

After running the script, the resulting window looks like that in Figure 18.


Test Window

I am a New Window!

Figure 18: Example using the New Window command.

Our combined reports example is slightly more complicated. Instead of holding a single display box, it holds results from two analysis platforms and a graph platform. In order to format these in an arrangement we want, we need to learn two more commands: `H List Box` and `V List Box`.

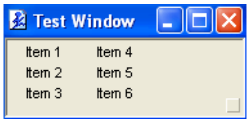
These two commands tell JMP how to arrange individual display boxes. `H List Box` arranges them horizontally, while `V List Box` arranges them vertically.

Here's a simple example to show how `H List Box` and `V List Box` work.

```
New Window("Test Window",
 H List Box(
  V List Box(
   Text Box("Item 1"),
   Text Box("Item 2"),
   Text Box("Item 3")
  ),

  V List Box(
   Text Box("Item 4"),
   Text Box("Item 5"),
   Text Box("Item 6")
  )
 )
)
```

After running the script, the resulting window looks like that in Figure 19.



Test Window

Item 1 Item 4
Item 2 Item 5
Item 3 Item 6

Figure 19: Example using the H List Box and V List Box commands.

Note that the outermost list box is horizontal, so the two inside boxes are glued together side to side. The inside boxes are vertical, so their items are glued together top to bottom. In essence, this script contains six text boxes. They are grouped vertically into groups of three, and the two groups are then combined horizontally.

With these commands, we are now ready to build a customized report window. Our example, though arbitrary, uses some non-standard reports. For example, the Distribution platforms only show histograms and box plots—no text reports. The spinning plot has a white background. The easiest way to find out how to customize the script to have these options in effect is to go through the process once using the menus, then selecting **Script > Save Script to Script Window** from the resulting report.

For example, select **Graph > Spinning Plot** and designate height, weight, and age as the three **Y, Columns** variables. When the report appears, select **White Background** from the red triangle menu beside the plot. Then, select **Script > Save Script to Script Window** from the platform drop-down menu. This shows the following JSL:

```
Spinning Plot(Y(:height,
:weight,:age), Biplot Rays(0),
White Background(1));
```

This is the JSL that we use in the customized report. Do the same for the Distribution platforms and the Fit Y by X platform, and you should arrive at script snippets similar to these.

```
Distribution(Y( :height,
:weight), Quantiles(0), Moments
(0))
Distribution(Y( :age), Frequen-
cies(0)))
Bivariate(Y( :height), X(
:weight))
Spinning Plot( Y(height, age,
weight), White Background(1))
```

We want to arrange them so that the three Distribution graphs are side by side (*i.e.* in an `H List Box`), and the Bivariate and Spinning plots are side by side (an `H List Box`). These two `H List Boxes` should then be stacked on top of each other, requiring a `V List Box`.

If we call the window "Combined Reports," we arrive at the following JSL.

Note particularly the placement of the commas. Inside each `H List Box`, the elements are separated by commas. In addition, the two `H List Boxes` are separated by commas within the `V List Box`.

```
New Window ("Combined Reports",
 V List Box(

  H List Box(
   Distribution(Y(:height,
     :weight), Quantiles(0),
     Moments (0)),
   Distribution(Y( :age),
     Frequencies(0))
  ),

  H List Box(
   Bivariate(Y(:height), X(
     :weight)),
   Spinning Plot( Y(height, age,
     weight), White
     Background(1))
  )
 )
)
```

Run the above script to get the plot shown at the beginning of this article in Figure 17.

Contrasts and Custom Tests II

Gunter Hartel, Research and Development, CSL Limited

Lee Creighton, JMP Development

In Issue 11 of *JMPer Cable*, the article “Contrasts and Custom Tests” described how to use JMP’s **Custom Test** command to get estimates and tests of linear contrasts in the Fit Model platform. In this article, we show how to use the **Custom Test** command to compute contrasts for effects that contain continuous variables. Contrasts for nominal or ordinal effects and comparisons among several groups can be accomplished through other commands in JMP. However, the **Custom Test** command is the only way to compute similar tests for effects that contain continuous variables.

Custom Tests on Slopes

As an example, consider the following (fictional) data from Stability Example.jmp—a stability trial whose data is shown in Figure 20 and can be downloaded from http://www.jmp.com/news/jmpercable. A product’s potency tends to decline over time, so different formulations are considered to see if the degradation can be slowed.

A total of 15 samples are taken from each formulation. Three samples are tested at the beginning of the trial (Month=0), and at 3, 6, 9, and 12 months. The main objective of the analysis is not to compare the potency of the formulations, but rather to compare the average degradation rate of the formulations.

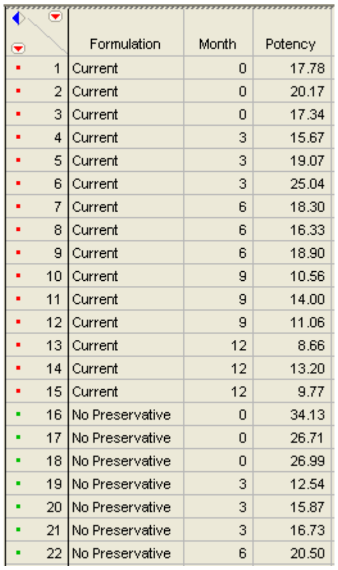
It is common in first-order degradation models to examine a regression of log(potency) on Formulation, Month, and Formulation*Month. The appropriate Fit Model dialog is shown in Figure 21.

We want to know whether the regression lines for each formulation are parallel. As shown in Figure 22, JMP’s regression plot (a scatterplot of the data with separate regression lines for each of the three formulations) shows that this is not the case.

The Effect Tests table (Figure 23) provided with the plot shows that all of the effects are significant.

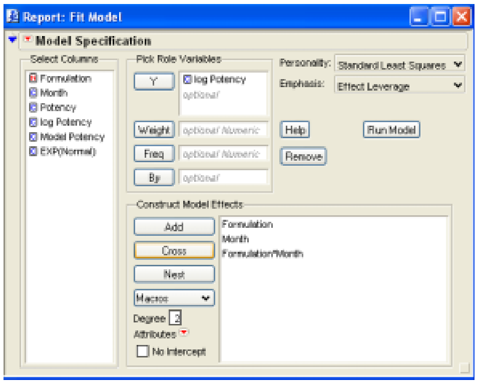
- The mean potency differs among formulations.
- The significant Month effect means the common slope is not zero.
- The interaction effect indicates that the degradation slopes differ by Formulation.

Selecting **Estimates > Expanded Estimates** from the red triangle menu on the Response log Potency title bar provides estimates and tests for the slopes and means of each formulation (Figure 24). Confidence limits, initially hidden, can be shown by right-clicking



	Formulation	Month	Potency
1	Current	0	17.78
2	Current	0	20.17
3	Current	0	17.34
4	Current	3	15.67
5	Current	3	19.07
6	Current	3	25.04
7	Current	6	18.30
8	Current	6	16.33
9	Current	6	18.90
10	Current	9	10.56
11	Current	9	14.00
12	Current	9	11.06
13	Current	12	8.66
14	Current	12	13.20
15	Current	12	9.77
16	No Preservative	0	34.13
17	No Preservative	0	26.71
18	No Preservative	0	26.99
19	No Preservative	3	12.54
20	No Preservative	3	15.87
21	No Preservative	3	16.73
22	No Preservative	6	20.50

Figure 20: Part of Stability Example.jmp.



Report: Fit Model

Model Specification

Select Columns: Formulation, Month, Potency, log Potency, Model Potency, EXP(Normal)

Pick Role Variables: Y: log Potency

Personality: Standard Least Squares

Emphasis: Effect Leverage

Construct Model Effects: Formulation, Month, Formulation*Month

Degree: 2

□ No Intercept

Figure 21: Starting a Fit Model analysis.

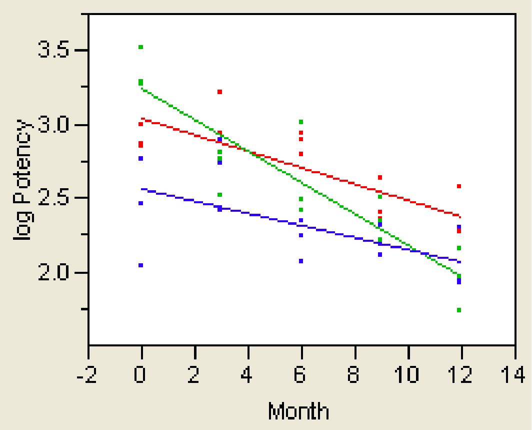
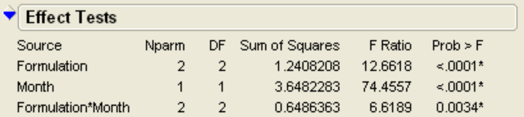


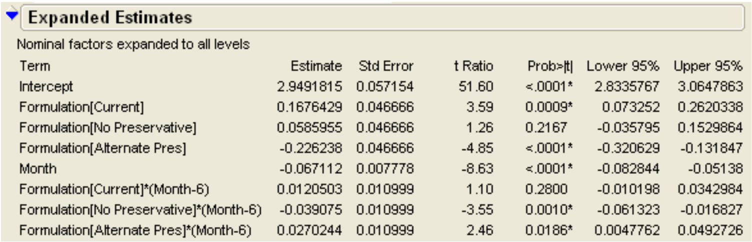
Figure 22: Regression plot.



Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Formulation	2	2	1.2408208	12.6618	<.0001*
Month	1	1	3.6482283	74.4557	<.0001*
Formulation*Month	2	2	0.6486363	6.6189	0.0034*

Figure 23: The Effects Test table.



Expanded Estimates

Nominal factors expanded to all levels

Term	Estimate	Std Error	t Ratio	Prob>|t|	Lower 95%	Upper 95%
Intercept	2.9491815	0.057154	51.60	<.0001*	2.8335767	3.0647863
Formulation[Current]	0.1676429	0.046666	3.59	0.0009*	0.073252	0.2620338
Formulation[No Preservative]	0.0585955	0.046666	1.26	0.2167	-0.035795	0.1529864
Formulation[Alternate Pres]	-0.226238	0.046666	-4.85	<.0001*	-0.320629	-0.131847
Month	-0.067112	0.007778	-8.63	<.0001*	-0.082844	-0.05138
Formulation[Current]*(Month-6)	0.0120503	0.010999	1.10	0.2800	-0.010198	0.0342984
Formulation[No Preservative]*(Month-6)	-0.039075	0.010999	-3.55	0.0010*	-0.061323	-0.016827
Formulation[Alternate Pres]*(Month-6)	0.0270244	0.010999	2.46	0.0186*	0.0047762	0.0492726

Figure 24: Estimates and tests for the slopes and means of each formulation.

the report and selecting **Columns >Lower 95%** and **Columns > Upper 95%**.

The purpose of the study is to compare the degradation slope of each new formulation with the slope of the Current formulation. If a difference is detected, we want to show that it is large enough to be of practical importance. That is, we want to test for non-inferiority of the new formulations versus the current.

Note that the Effect Detail sections for the slope (Month) and the Formulation by Month interaction have only one option: **Power Analysis** (Figure 25). So, selecting **Estimates > Custom Test** from the red triangle menu on the Response log Potency title bar is the only option for comparing the slopes.

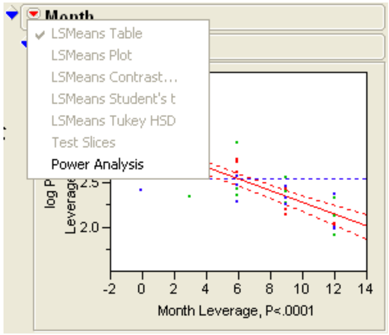


Figure 25: Power Analysis is the only available option.

An understanding the parameterization is important when constructing the correct test. Note that the Expanded Estimates sections reports the terms shown in Figure 24.

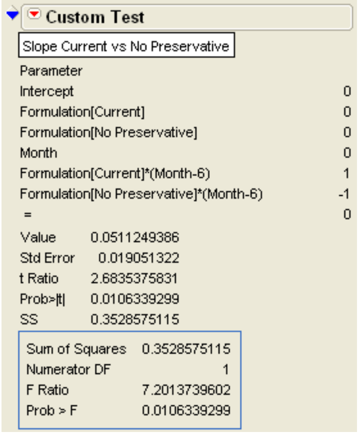
Intercept is the intercept of the common regression line, which is not pictured in the regression plot. However, since JMP uses polynomial centering, the offsets Formulation[Current] and Formulation[No Preservative] apply to the center of the regression line (in this example, six months), not the intercepts. Likewise, the effect test on Formulation compares the vertical distance between the regression lines at the same center (six months), not the difference of the intercepts (zero months). To calculate the intercept for, say, the Current formulation, we also need to subtract six months of slope offset:

```
intercept_Current= intercept +
Formulation[Current] -
6*{Formulation[Current]*
(month-6)}

= 2.9491815 + 0.1676429 -
6*0.0120503 = 3.0445226.
```

However, the slope for a particular formulation is simply the common slope plus the slope offset. For example, the slope for the Current formulation is -0.067112 + 0.0120503 = -0.0550617.

So, to compare the slope of the No Preservative formulation with the slope of the Current formulation (Figure 26), we want to test whether $d_1 = d_2$, where d_1 is the slope offset for the current formulation {Formulation[Current] *(month-6)} and d_2 is the slope offset for the No Preservative formulation.



Custom Test

Slope Current vs No Preservative

```
Parameter
Intercept                                  0
Formulation[Current]                       0
Formulation[No Preservative]               0
Month                                      0
Formulation[Current]*(Month-6)             1
Formulation[No Preservative]*(Month-6)    -1
 =                                         0
Value       0.0511249386
Std Error    0.019051322
t Ratio     2.6835375831
Prob>|t|    0.0106339299
SS          0.3528575115

 Sum of Squares   0.3528575115
 Numerator DF                1
 F Ratio          7.2013739602
 Prob > F         0.0106339299
```

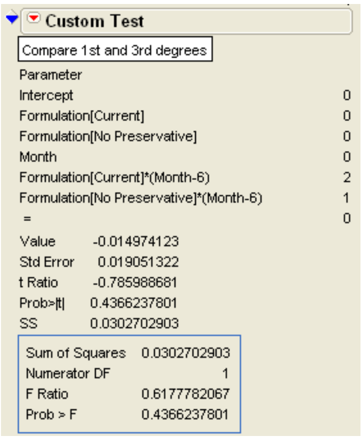
Figure 26: Comparing Current formulation to No Preservative formulation.

We know that $d_1 = d_2$ is equivalent to $d_1 – d_2 = 0$, and so the Custom Test dialog is filled out with a 1 next to Formulation[Current] * (month-6) and a –1 next to Formulation[No Preservative] * (month-6). Now click **Done**.

The estimated difference in slopes is a statistically significant 0.05112 (p=0.01). Using the standard error of 0.01905, we can calculate a 95% confidence interval for the difference (0.01259, 0.08966).

Since these estimates are all in terms of log(potency), the actual difference in slopes is exp(0.05112) = 1.0525, with 95% confidence interval (1.0127, 1.0938). The interpretation is that if Current and No Preservative formulations start with the same potency, then the Current formulation retains 1.0525 the potency of the No Preservative formulation per month of storage, or approximately a 5% per month advantage. The upper limit is 1.0938, or approximately a 9% per month advantage for the current formulation. Even though there is a statistically significant difference in stability slopes, the advantage of removal of the preservative may outweigh the potential potency drop of up to 9% per month.

To get a comparison of the Alternate Preservative formulation with the Current formulation, we need to consider the parameterization that JMP uses. As there are three formulations, there are three slopes and three degrees of freedom for estimating slopes. The parallelism of slopes model fits a common slope, and two slope offsets. The third offset is the negative of the sum of the other two, so that the sum of the three offsets is zero. That is, $d_1 + d_2 + d_3=0$ and $d_3= -(d_1 + d_2)$. To test the hypothesis that the third slope is equal to the first slope, we have d1 = d3, or $d_1 - d_3 = 0$. Substituting in the identity for d3 gives $d_1 + (d_1 + d_2) = 0$, or $2d_1 + d_2 = 0$. To complete the Custom Test dialog for this example, enter a 2 for the first Formulation*Month slope and a 1 for the second Formulation*Month slope (as shown in Figure 27). Now click **Done**. There is no statistical significance shown between these two formulations. The Alternate Preservative formulation is no better than the current one.



```
Custom Test
Compare 1st and 3rd degrees
Parameter
Intercept                                   0
Formulation[Current]                        0
Formulation[No Preservative]                0
Month                                       0
Formulation[Current]*(Month-6)              2
Formulation[No Preservative]*(Month-6)      1
 =                                          0
Value       -0.014974123
Std Error    0.019051322
t Ratio     -0.785988681
Prob>|t|    0.4366237801
SS          0.0302702903

 Sum of Squares   0.0302702903
 Numerator DF                1
 F Ratio          0.6177782067
 Prob > F         0.4366237801
```

Figure 27: Custom test comparing first and third slopes.

Although JMP contains many powerful commands for exploring estimates, there are some occasions when you must use your knowledge of JMP's parameterization to create custom tests. This article showed one such case by creating and computing contrasts between groups where the comparison involved a continuous covariate.

Look for JMP at these Conferences

August 30-31, 2004	JMP User Conference in Cary, NC
September 26-28, 2004	Midwest SAS Users Group (MWSUG) in Chicago, IL
October 7-8, 2004	SE INFORMS (Southeast Institute for Operations Research and the Management Sciences) 2004 Annual Meeting in Myrtle Beach, SC
October 12-16, 2004	WUSS (Western Users of SAS Software) in Pasadena, CA
October 24-27, 2004	INFORMS Annual Meeting in Denver, CO

JMP User Conference Registration Special

August 30-31, 2004 at SAS World Headquarters, Cary, NC

Summer madness has come early and it's twice as nice. As a special offer for the inaugural JMP User Conference, register and bring a friend for free. This offer applies to the conference or any conference/training package. (Only one discount per customer.) See http://support.sas.com/jmpconf for details.

In addition to the two-day user conference, stay for JMP training September 1-2, 2004. Both one-day and two-day courses are offered on design of experiments, analysis of attribute data, and mixture design of experiments.

Special Bonus: FREE Six Sigma Class

Conference attendees can attend a free half-day overview of Six Sigma training on either September 1 or 2, 2004. See http://support.sas.com/jmpconf for details.

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...Mark Bailey

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

Mark Bailey

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

Mark: I always wanted to be a scientist. I was one of those kids who got a chemistry set for Christmas when I was in grade school. Within a year I had a small laboratory set up in the basement. I dissected worms, studied water samples under the microscope, and made lots of smelly things. I settled on chemistry as the "central science" for my career. I received my masters and doctoral degrees in chemistry from the University of Rochester. My thesis was based on an eclectic collection of biophysical inorganic chemistry experiments to learn about the "cooperative" binding mechanism of hemoglobin. I created synthetic, intensely colored hemes, which replaced iron with a different transition metal. I then substituted them in place of the natural heme without disrupting the function of the protein. Then I tortured the molecule with many different analytical techniques to find out what changed and what it meant about hemoglobin function. I worked in industry for 15 years creating new medical devices for testing blood. I immediately began to study statistical science. By the end, I was teaching statistics and JMP to others. I proposed that I serve in this capacity to SAS, and John Sall, executive vice president of SAS, agreed that it was a good fit. I have been in the Statistical Training & Technical Service group for over six years.

JMPer Cable: What is your role within JMP training?

Mark: I develop training courses, which involves creating lessons, examples, and demonstrations. The finished work includes presentation slides, a book of course notes, and many JMP files. I customize training for on-site classes. I tailor the course to emphasize certain topics and include customer examples and data. I mentor customers who need intense informal training under contract. This work includes the design and analysis of experiments, scripting JMP for large automated and customized applications. I answer a lot of telephone calls and e-mail from former students and JMP users who have questions about data, statistics, design, or scripting. Over the past four years I have developed interactive demonstrations for teaching statistical topics using JMP. I presented my results as a Computer Technology Workshop at the 2003 Joint Statistical Meetings (JSM) and as a shortened version at the Fall Technical Conference (FTC) last year. I am presenting an entirely new collection of demonstrations again at both the JSM and the FTC this year. Along the same lines, I have become more active in supporting the high school Advanced Placement (AP) program and the statistics course in particular. I am particularly proud of bringing SAS Education closer to JMP development and technical support. Aside from minor revisions, all of our training is developed and tested with their full participation.

JMPer Cable: What have been some of your favorite projects?

Mark: Two training projects come to mind immediately. We created a brand new Design of Experiments (DOE) course for JMP 4, which introduced custom design. It took three attempts but we finally got it right. We felt that the training had to emphasize the amazing capabilities of the custom designs. My other favorite project was the first scripting course, written by Wayne Levin from JMP's Training department. I had accumulated several scripts created by Lee Creighton (JMP Development) and me, which served as a starting point for a complete scripting course.

JMPer Cable: What do you like best about training?

Mark: I bet that you are not shocked, given my background, when I say that I love design and analysis of experiments first and foremost. This area is very active now with new research and development into areas such as split-plot designs, uniform designs, and optimal design. I am also particularly fond of exploratory data analysis, which is a big strength of JMP. My favorite thing about training is to meet JMP users and learn about their business and technical problems. I can then show them how to use JMP in ways that they never thought of. They might not know about certain features or shortcuts. JMP has advanced so much that it is easy to overlook a capability that is just what you are looking for. I also like to learn new things. Training others forces me to learn new things all the time, too.

JMPer Cable: Where do you get your passion for teaching?

Mark: I think that my science teachers were always personal heroes for me. They taught me, more than anything, how to share the new knowledge and collaborate on projects. They confirmed that learning did not end when school was over. I have always enjoyed theory and practice. Whether it was computing molecular orbitals and the resulting spectroscopic characteristics or synthesizing air-sensitive organometallic complexes, it was intriguing. It was fun. Now it is learning about statistical science and theory and then adapting it for our lessons or applying it to a user's problem. I really enjoy the interplay and interaction between theory and practice.

JMPer Cable: As a long-time contributor to the JMPer Cable, where do you get the inspiration and enthusiasm for writing articles and scripts?

Mark: In the shower, of course! Seriously, I can answer that question by stealing a quote from the late master John Tukey, "The best part about being a statistician is that you get to play in everyone's backyard." I get to see a lot of different problems through my work. I encounter a problem and develop a solution in one area, then realize that other users might benefit from the news. I try to write a simple and straightforward story as a lesson so other users can easily understand the points and put them to work in their own projects. It might be a useful statistic, JMP command or function, an unfamiliar platform, or a script.

New JMP Software Mentoring Services Available

Take advantage of a new service offered through JMP Training. The JMP Software Mentoring Service provides you with the assistance of a JMP Technical Expert who will transfer knowledge and provide instruction to individuals or small groups in your organization. Let our Technical Experts personalize a solution for your JMP project, which may include JMP software proficiency, scripting efficiency, or other defined JMP related areas pertinent to your organization's business needs and success. Solutions for each group or individual will vary and may be characterized by informal coaching and knowledge transfer. To learn more about this new service, please call (919) 531-7312.

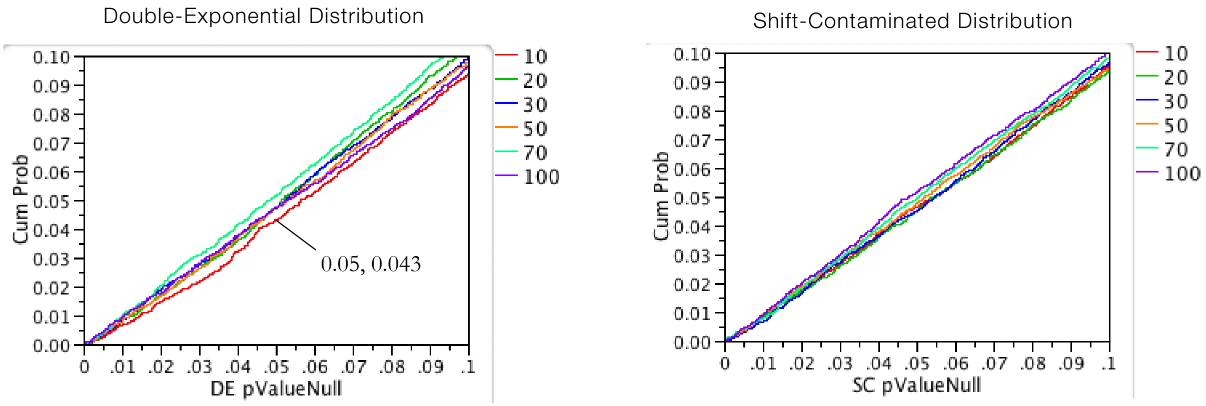


Figure 4. Expanded-scale view of p-value probability plots for the non-normal distributions.

(continued from page 2)

y-axis is the proportion of the 10,000 *p*-values less than or equal to a given significance probability. The *x*-axis shows the significance probabilities (*p*-values). Each sample size is shown in a different color. For a color version of this article, see http://www.jmp.com/news/jmpercable.

For both plots and all sample sizes, the distributions of the *p*-values look close to the diagonal line of the uniform cumulative distribution function. As we hoped, neither changing the distribution nor changing the sample size has much effect on the accuracy of significance probability of the ANOVA test for equal means.

Let's look closer at the low scale near 0.05 where we are judging significance. The plots in Figure 4 magnify the area.

Note that the red diagonal line corresponding to the test for equality of means for two groups with five observations strays slightly below the line y = x. What does this mean?

We know that if we choose a significance probability of 0.05, we will reject the hypothesis that our means are equal in 5% of our experiments due to the vagaries of chance even if the means are truly equal. That is, 5% of the time we will make a Type I error—we will reject the hypothesis that the means are equal when the hypothesis is true.

This 5% Type I error rate is built-in and what we expect if we choose a significance probability of 0.05. If this seems like too many errors, then you need to choose a smaller significance probability.

When the red diagonal line crosses the vertical dotted line x = 0.05, its *y*-value is 0.043. That means that instead of rejecting the hypothesis that our means are equal 5% of the time, we only rejected 4.3% of the time. We call this a conservative result because we are making fewer Type I errors than we expect. If you are going to miss, it is better to miss conservatively.

On the other hand, suppose a line had crossed the vertical dotted line at 0.05 at a CumProb value of 0.06. Then, instead of seeing the expected 5% Type I errors, we would actually see 6% Type I errors. We call such a test too liberal because the true error rate is larger than expected.

It is reassuring that in the only case where the lines deviate from the 45-degree line, it is on the conservative side. The shift-contaminated case also has the *n*=10 case shifted to the conservative side, but not so much as the double exponential.

Figure 5 shows the Monte Carlo simulation with a pure normal curve to

(continued on page 14)

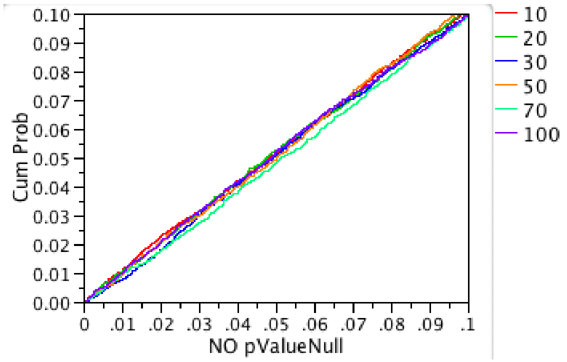


Figure 5. Expanded-scale view of p-value probability plot for the standard normal distribution.

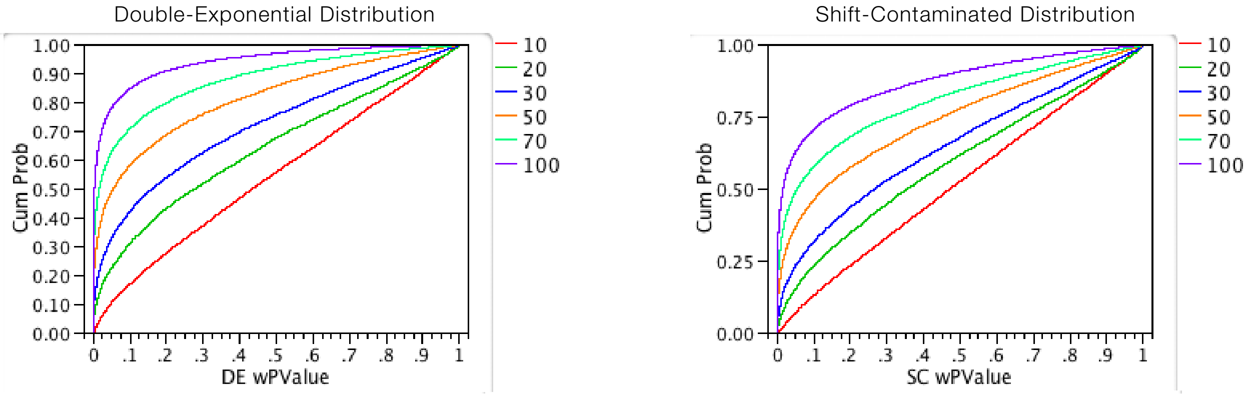


Figure 6: Power plots of Shapiro-Wilks tests for non-normal distributions.

(continued from page 13)

show the natural random variation in the Monte Carlo study.

Testing Non-Normality

The next question is whether we can detect that errors are non-normal. We start by computing the residuals from each oneway analysis we did before. Using the distribution platform we test these residuals for normality. Thus, for each distribution and sample size we do 10,000 Shapiro-Wilks tests for normality. The plots in Figure 6 show the cumulative percentage of normality test results as a function of the significance probability in the Shapiro-Wilks test.

What do these plots tell us? First we know that in both the double-exponential case and the shift-contaminated normal case, the distribution of the residuals is not normal. We want to know how likely it is that our test will detect this non-normality. In other words, we want to know how powerful our test is.

A perfectly powerful test would have a graph that looked like the Greek letter gamma, Γ.

By contrast, a test with no power would have a graph that looks like a diagonal line.

For double exponentially distributed residuals, we can see that the graph for the sample of 100 observations rises rapidly on the far left, as desired. Unfortunately, the resemblance of the graph to Γ deteriorates as the sample size drops. When the sample size is 10, the function looks much more like the diagonal line (no detection power).

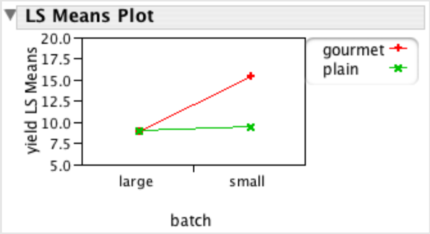
For the shift-contaminated normal distributed residuals, the detection power of the Shapiro-Wilks test is

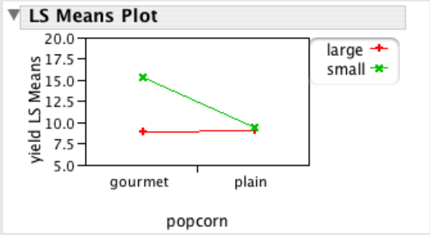
(continued on page 15)

Shortcut: Least Squares Means Plot

To transpose the factors in a two-factor interaction LSMeans plot, press Shift while selecting the LSMeans Plot command from the red triangle icon in the LSMeans title bar.

For example, the following pictures show the same interaction: the first image is the default plot, the second is the plot with the Shift key pressed.





worse for every sample size.

When the residuals really are normal, as plotted in Figure 7, we see a diagonal line for every sample size. This is what we want. We do not wish to detect non-existent non-normality. Here, the "power" of the test is the probability of a Type I error. It is impossible for a test to be less powerful than this.

Now let's expand the *p*-value area so we can examine the power at alpha=0.05, as shown in Figure 8. The bottom curve (sample size *n* = 10) in Figure 8 crosses the vertical grid line x=0.05 at the value 0.10. That means that if the Shapiro-Wilks significance probability is 0.05, you have a 10% chance of detecting the fact that the residuals are non-normal. We say that the power of the Shapiro-Wilks test is 10%. This is not impressive.

Another way of saying this is that the probability of a Type II error is 90%. That is, 90% of the time the Shapiro-Wilks test (given this error distribution and significance level) will incorrectly fail to reject the hypothesis of normal residuals.

By contrast, the top curve above (sample size *n* = 100) crosses the vertical grid line x=0.05 at the value 0.78. So, the power of this test is nearly 80%.

The power curves have a similar shape for the shift-contaminated normal residuals, but the resulting power is even less. For example, the top curve (sample size n = 100) crosses the 0.05 line at about 0.63 (power is 63%).

So we see that for small numbers of observations you can't detect non-normality. For large number of observations you don't care if the errors are non-normal because our simulated Type I error rate matched the expected rate for large sample sizes.

This study would tend to support not implementing a test for normality in the fitting platforms. Including this feature would only encourage people to worry about something that they

(continued on page 16)

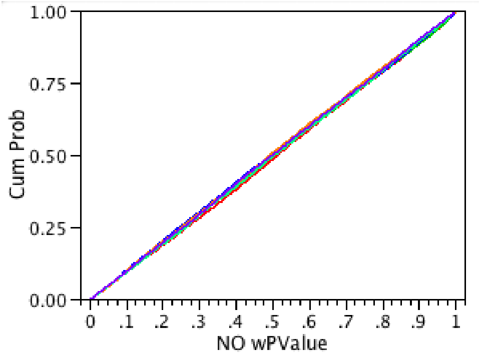
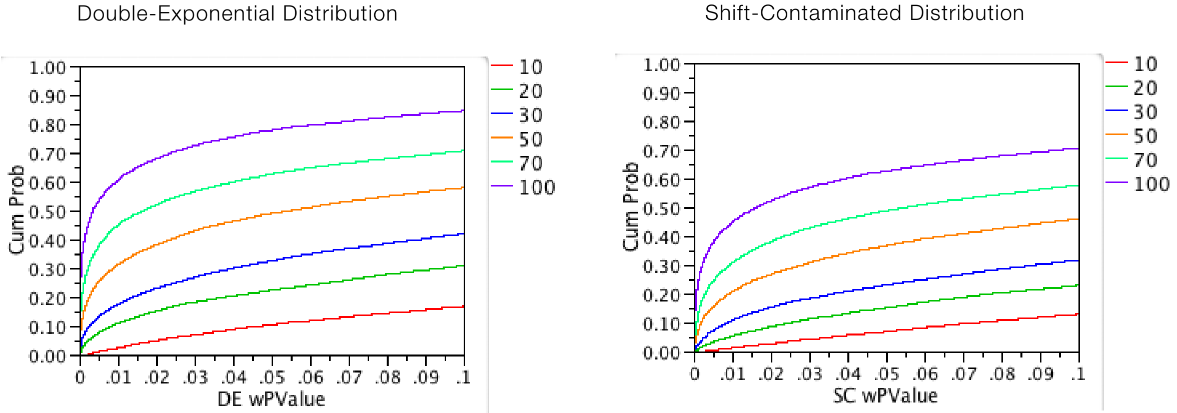


Figure 7: Power plot of Shapiro-Wilks test for the standard normal distribution.


Double-Exponential Distribution

Shift-Contaminated Distribution

Figure 8: Expanded-scale power plots for the non-normal distributions.

(continued from page 15)

either can't see or that doesn't matter.

Putting in graphical and diagnostic features for the residual distribution is a different matter. Instead of testing, you are looking for graphical anomalies, like outliers, or a pattern that might be a clue to some hidden structure of the mode. Outliers and patterns are often the main clue to an important discovery.

One of W. Edwards Deming's commandments was "Drive out fear," but as statisticians we usually do the opposite, instilling irrational fears in our clients and students. With this study, which you can easily duplicate, I hope you are now brave enough to fight "leptokurtosiphobia."

Isn't it nice to have one less thing to worry about?

About JMPer Cable
Issue 15 Summer 2004

JMPer Cable is mailed to JMP users who are registered users with SAS Institute. To register, go to http://www.jmp.com/product/regrequest_form.shtml. The newsletter is also available online at www.jmp.com.

Contributors
Mark Bailey
Meredith Blackwelder
Lee Creighton
Gunter Hartel
Bradley Jones
John Sall

Technical Editor
Ann Lehman

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.jmp.com