A Technical Publication for JMP® Users
Spring 2004 Issue 14

JMPer Cable

Inside This Newsletter

Breaking News

The JMP User Conference

We invite you to attend the first annual JMP User Conference August 30-31, 2004 at the SAS World Headquarters in Cary, North Carolina.

Join colleagues and experts in the JMP community for a conference that will provide new insight into the world of statistical discovery and exploration with JMP software. Come meet the developers who brought you JMP and share experiences and knowledge with other JMP users.

Enabling ODBC on JMP for Linux

Paul Nelson, JMP Development

One of the powerful features of JMP for Linux is its ability to connect to databases through ODBC (Open DataBase Connectivity). This is enabled by JMP's use of open source unixODBC.

The JMP installation CD contains a prebuilt copy of the unixODBC library, named libODBC.so, as well as the complete unixODBC source code. You can visit http://www.unixodbc.org to check for a newer unixODBC release than the one on the CD.

This article walks you through the three steps needed to enable ODBC on Linux. It assumes you are using the Bourne Again Shell (bash), which is the default shell on most Linux distributions. The syntax will vary if you are using tcsh, ksh, zsh, or some other shell. Also, you should be familiar with typical UNIX commands, permissions, working from the command line, and basic editing of .rc files using an editor such as emacs, vi, or kwrite. If you are not comfortable with the above, seek out your local Linux/UNIX guru or "friendly system administrator."

Step 1: Install unixODBC Libraries

When JMP launches, it searches for the presence of unixODBC shared libraries. If it finds the libraries, the **Open Database Table** button is shown on the JMP Starter (Figure 1).

If JMP cannot find the libraries, create a lib directory within your home directory for the unixODBC libraries. For the purposes of this article, we assume that you are installing without root privileges. If you are the system administrator (also called "root"), you may want to install the libraries in /usr/lib or, more typically, /usr/local/lib.

First, in a terminal window, type:

```
cd
mkdir lib
cd lib
```

Then, insert the JMP installation CD. Your system should auto-mount the CD to /mnt/cdrom. If the CD doesn't auto-mount, have the system administrator mount it for you.

Next, if you're using a current distribution, such as Red Hat 9, open the Linux kernel 2.4.20 (gcc 3.2.2) directory on the CD. If you're using an older or a different distribution, open the Linux kernel 2.4.18 (gcc 2.96) directory. Copy the odbc_lib.tgz file

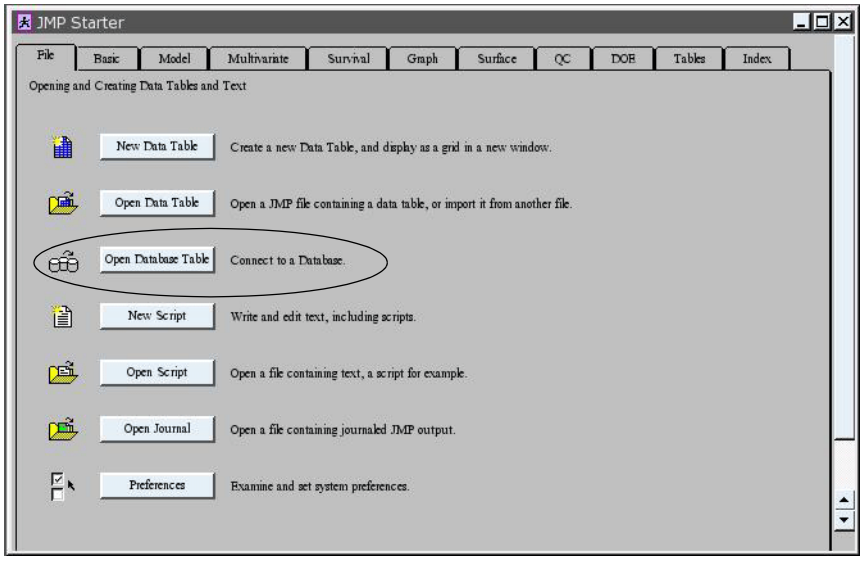

Figure 1: The Open Database Table button is present when JMP detects unixODBC shared libraries.

from the unixODBC directory on the JMP CD (/mnt/cdrom) to your current directory. If you are using Red Hat 9 and the JMP installation CD is mounted to /mnt/cdrom, you should type:

```
cp /mnt/cdrom/Linux\ kernel\
2.4.20\ \(gcc\ 3.2.2\)/unixODBC/
odbc_lib.tgz .
```

(Notice the \ to escape the spaces and the parentheses in the path.)

Now, to unpack the file, type:

```
tar xzvf odbc_lib.tgz
```

This results in the following output:

```
odbc_lib/
odbc_lib/
Configuring_ODBC_Libraries.txt
odbc_lib/libodbcinst.la
odbc_lib/libodbcinst.so.1.0.0
odbc_lib/libodbc.la
odbc_lib/libodbc.so.1.0.0
odbc_lib/makeODBCLinks
```

Now change directory into odbc_lib:

```
cd odbc_lib
```

Move everything up to the lib directory. Using the `mv` (move) command with the `-i` flag prompts you if the files already exist in the destination directory. If prompted, you do *not* want to overwrite the existing files. To execute the move, type:

```
mv -i * ..
```

Go up to where the files are now:

```
cd ..
```

Run the script to create symbolic links:

```
./makeODBCLinks
```

You can safely remove the now-empty odbc_lib directory:

```
rmdir odbc_lib
```

Now update your LD_LIBRARY_PATH to include these libraries. The example below puts them into a lib directory in your home directory ($HOME/lib).

In a terminal window, type:

```
   export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH
:$HOME/lib
```

For JMP to be able to find the libraries the next time you log in, the lines of code shown above need to be added to your .bash_profile file.

Users of tcsh would add a line to either the .login or .tcshrc file using the following syntax:

```
LD_LIBRARY_PATH=($LD_LIBRARY_
PATH    $HOME/lib)
```

You have now finished installing the ODBC libraries. However, the configuration to connect to a database is not yet complete. You must also install the database-specific ODBC driver and set up the Data Source Name (DSN) configuration in the .odbc.ini file.

Step 2: Configure the ODBC Driver

MySQL is a commonly-used database on Linux, so our example assumes you have a MySQL database installed locally or available on a network. If this is the case, you need to have the libmyodbc.so or libmyodbc3.so shared libraries and their dependencies. If you don't, we recommend adding them from your original Linux distribution and installing them at /usr/lib or /usr/local/lib or go to http://www.rpmfind.net for downloading. If the libmyodbc.so or libmyodbc3.so shared libraries and their dependencies don't work, visit http://dev.mysql.com/downloads, and download and install all the dynamic client libraries and programs for your version of the database.

If you are not using MySQL, visit http://www.unixodbc.org to find drivers for other databases.

Step 3: Set up the odbc.ini File

If you want to use unixODBC to connect to a database, you should have your own .odbc.ini file in your home directory. For example, my file is in /home/paul/.odbc.ini.

If you are installing for *all* users on the

system, configuration is typically done in either /etc/odbc.ini or /usr/local/etc/odbc.ini, depending on the installation location of libodbc.so.

As an example, two entries from my .odbc.ini file are shown below. The DRIVER line points to the MySQL ODBC driver library, i.e., the full path to either libmyodbc.so or libmyodbc3.so.

```
[bug_tracker]
DRIVER   = /usr/lib/libmyodbc.so
DSN      = bug_tracker
SERVER   = ourserver.xxxx.sas.com
USER     = my_username
PASSWORD = my_password
PORT     = 3306
DATABASE = bug_tracker

[jmp_requests]
Driver   = /usr/lib/libmyodbc.so
DSN      = jmp_requests
SERVER   = ourserver.xxxx.sas.com
USER     = username
PASSWORD =
PORT     = 3306
DATABASE = jmp_requests
#SOCKET  = /var/lib/mysql/mysql.sock
```

NOTE: Usernames and passwords refer to database username and passwords, not login usernames and passwords.

Notice that the # symbol in the last line above indicates that the line is commented out. If you run into difficulties connecting to the database, try adding the SOCKET line by removing the # symbol. The location of the mysql.sock file varies with Linux distributions and MySQL installations, so you may need to contact the database administrator for help.

Once you have created or edited your .odbc.ini file to reflect your database name, location, username, and passwords, JMP should be able to connect to the listed databases.

To see that the configuration is correct:

1. Launch JMP. Your JMP Starter window should contain the **Open Database Table** button.
2. Click the **Open Database Table** button. The window in Figure 2 appears.

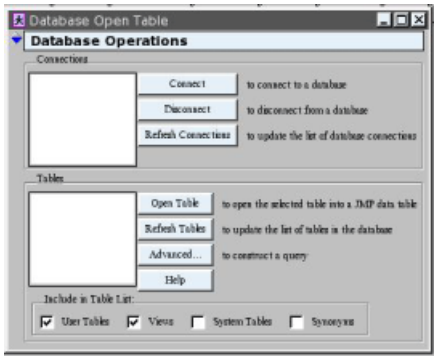

Figure 2: The Open Table window.

3. Click **Connect**.

Success is assured when you see the newly configured database entries in the Get Data Source Name window (Figure 3).

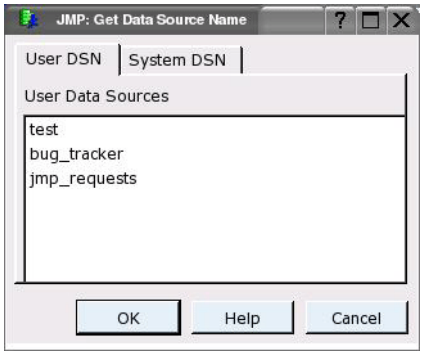

Figure 3: Your configured databases appear in this window.

System-wide data sources are in the SystemDSN tab (Figure 4), configured in the odbc.ini file, located in either /etc/ or /usr/local/etc.

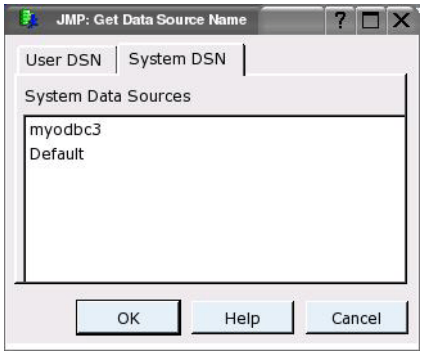

Figure 4: The SystemDSN tab lists system-wide data sources.

If you receive errors, make sure there are no typos in your .odbc.ini file and that you have permission to access the database(s) in question. Also, at a shell prompt, type `echo $LD_LIBRARY_PATH` and verify that your LD_LIBRARY_PATH includes the directory where the unixODBC libraries from the JMP CD were installed, such as /home/paul/lib. If it does not contain the directory where the libraries are installed, follow the instructions presented earlier in this article for setting the library path.

With such a wide variety of Linux systems and databases available, there is plenty of potential for difficulty in enabling unixODBC. For additional help that this article doesn't cover, check out the documentation at both http://www.unixodbc.org and http://www.mysql.com. Consulting these sources can help you soar past difficult configurations.

Measurement Systems Analysis: More than Just Part and Operator

Annie Dudley Zangi, JMP Development

Many quality experts, including Six Sigma Black Belts, saved companies over $250,000 by decreasing the measurement error generated by inaccurate gauges. Identifying and reducing the variance associated with measurement error enables them to focus on reducing the variance attributable to the actual parts.

JMP's Gage R&R platform helps them identify these errors with automatic "live" graphs and summary tables. JMP's graphs break down the performance by each part, instrument, and/or rater combination, where the user can click on specific observations, see the associated rows, and explore the problem further. JMP goes much further than the simple part-by-operator comparisons.

This article presents a case study of an actual manufacturing company. To protect the interests of the company, the company name and details of the project are altered.

Defining the Problem

In 2001, customer complaints at ACME surged because of excessive defective pistons. Customers were concerned about defects because they needed the pistons to generate the correct amount of pressure when placed in engines. They required the volume from the piston to be no less than 62.4 cubic inches and no greater than 62.6 cubic inches.

ACME spent hours scouring the data on the common suspects—height, diameter, and roundness of each piston—only to find that there wasn't much variation. They retrained operators and attempted to improve pistons' consistency in pressure generated by trying different settings at the assembly line stations. Despite these actions, they found the defect rate would not drop below 30%.

They defined the problem by examining the capability of their system. They bought several high precision engine simulators to test every piston coming off the line.

To repeat their steps:

1. Open the DieselPistonsMSA.jmp data table found at http://www.jmp.com/news/jmpercable.
2. Choose **Analyze > Distribution** with Volume as the **Y, Columns** variable and click **OK**.
3. Click the red triangle on the Volume title bar and choose **Capability Analysis**.
4. Enter the required specification limits in the dialog. In this case, customers required a lower limit of 62.4 cubic inches and an upper limit of 62.6 cubic inches.
5. Click **OK**.
6. Right-click the PPM column in lower table and select **Columns > Sigma Quality**.

In the lower right corner of the report (see Figure 5), a column called **Sigma Quality** shows how close to Six Sigma the process really is. With over 500,000 PPM (Parts Per Million) outside the spec limits, the sigma quality level at ACME was very low at 1.483—nowhere close to six.

ACME saw that nearly 30% of their products were failing customer-required specifications. The consequences of such a high defect rate were costly. In order for ACME to produce 1000 acceptable pistons, they had to actually produce 1450 pistons; nearly half again as many. Since production lines were already running

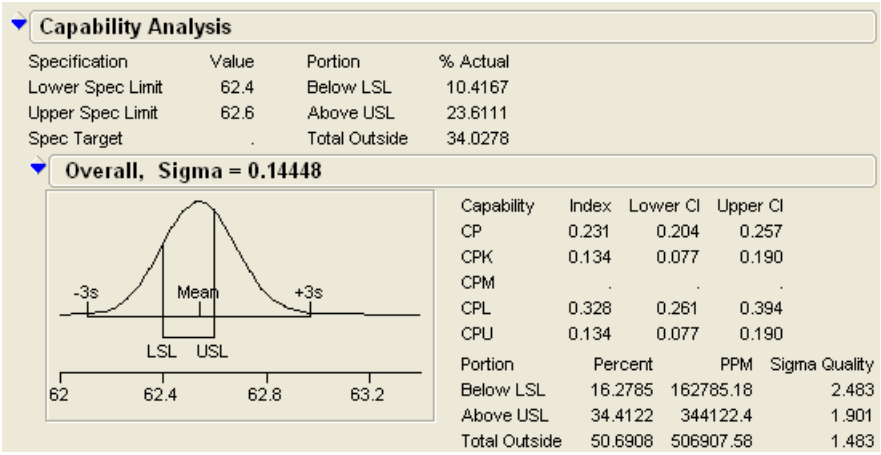

Figure 5: ACME's sigma quality level is too low.

at full capacity, they were missing deadlines and paying express shipping. Sometimes, they even had to charter a freight flight to deliver on time and avoid losing a customer. Having late orders was a serious problem: late deliveries meant customers would need to shut down the engine product line and plant, which would in turn shut down the downstream car plants' line. In addition to fees associated with late shipments, ACME found they also needed to hire someone dedicated to emptying the scrap pile on the factory floor. Summing all the factors associated with the 30% scrap (the defective pistons) shows a cost of over $300,000 per year.

Measuring the Problem

After ACME defined the problem, they measured it using a measurement systems analysis. Differing numbers arose from measuring the same part, which indicated that there was a problem with the measuring system. This problem needed correcting before the true, underlying problem could be addressed. They recognized that accurate measurements are fundamental to any study.

Each part was randomly tested several times by each operator on one of three test machines. They used the traditional model of examining the part by operator combination and confirmed the problems across operators and across parts.

To follow their steps in running a traditional measurement systems analysis on part and operator:

1. Choose **Graph >Variability/Gage Chart**.
2. Assign Volume as the **Y, Response** and the columns Operator and Part as **X, Grouping** variables and click **OK**.
3. Click the red triangle in the Variability Gage title bar and choose **Gage Studies > Gage RR**.
4. Since the acceptable interval is between 62.4 and 62.6, enter 0.2 for the tolerance interval and click **OK**.

In addition to the Variability chart showed in Figure 6, JMP gives the variance components for Gage R&R, shown in Figure 7. From this chart, ACME could see that there were other problems to fix before focusing on the parts because the Gage R&R is below 90%. The part-to-part variation only contributed 20% to the total variation in the process.

Adding Additional Information

Using just the operator and the part in their analysis, ACME couldn't understand the problem. However, because JMP allowed them to include information in addition to operator and part, they were able to analyze further. They added a third variable: machine number.

To see how they added the third variable:

1. Choose **Graph > Variability/Gage Chart**.
2. Choose Volume as the **Y, Response** and choose Test Machine as the first **X, Grouping**, then add Operator and Part.
3. Click **OK**.

The sorting order in the resulting variability chart is dependent on the order of the *x*'s. By adding the machine number as an additional *x* variable and ordering it first, the problem became clear: Machine 3 gave wildly different numbers for the same part being measured (Figure 8). This machine

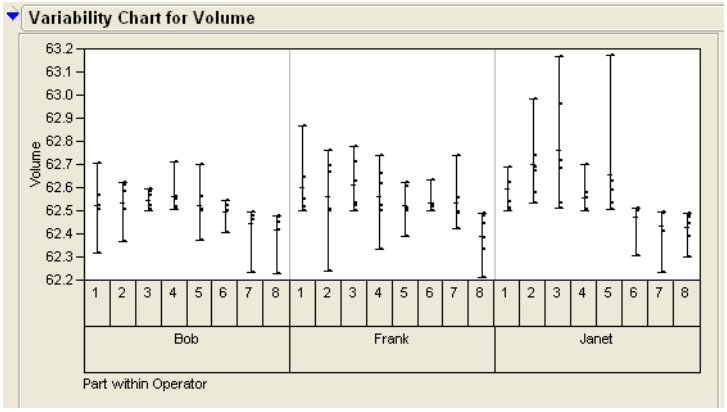

Figure 6: Using JMP's Variability/Gage chart, ACME used a traditional measurement systems analysis.

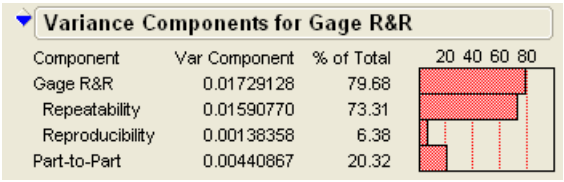

Figure 7: The Gage R&R is below 90% and part-to-part variation only 20%.

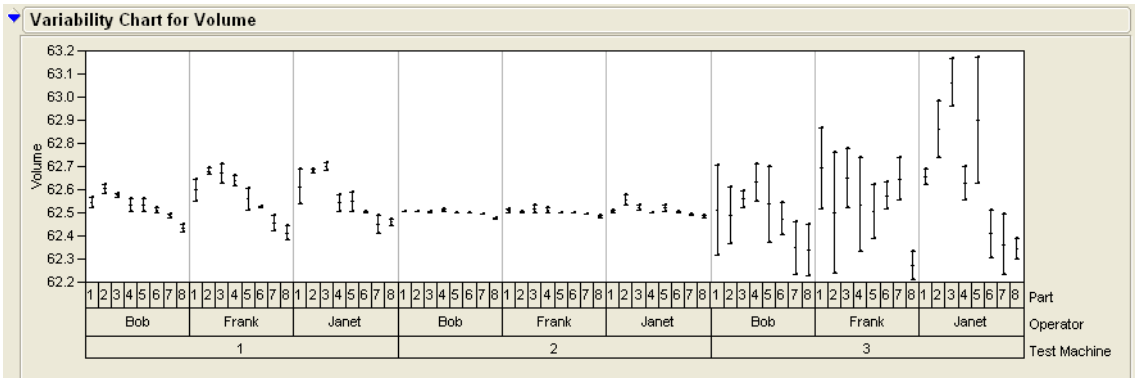

Figure 8: Machine 3 gave wildly different numbers, and Machine 1 shows a small problem.

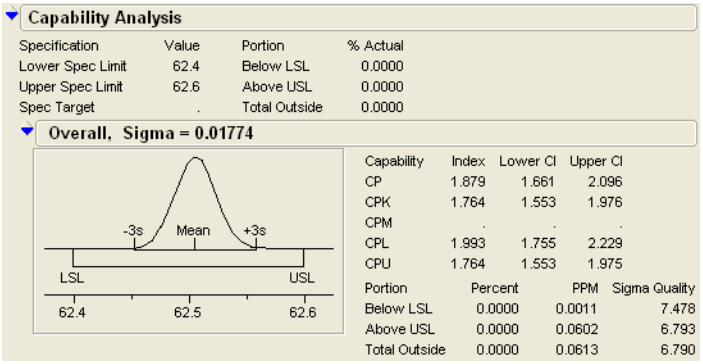

Figure 9: After fixing the machines, ACME's sigma quality level was 6.790.

was the cause of much of the variation. It was probably the cause of many of the parts being labeled as failures, when in reality, they were within the needed specification limits.

Lessons Learned

Although Machine 3 cost several thousand dollars for ACME to replace, the savings were immediate. Systems were no longer needed for managing scrap. Additionally, the variability chart alerted ACME that they needed to service Machine 1 to bring it back in alignment and ensure that it also provided accurate readings.

In summary, ACME's biggest problem wasn't a problem with parts at all. Rather, the problem was with the test machine measuring the parts. After replacing Machine 3 and servicing Machine 1, they achieved compliance with the specifications. All the training and tuning of the lines worked after all.

After implementing the changes and rerunning the analysis (using the DieselPistonsMSA2.jmp data table downloadable at http://www.jmp.com/news/jmpercable), fewer than ½ of 1 PPM was outside the specifications, giving them a great sigma quality level of 6.790 (see Figure 9).

Strut Your Demo

If you use JMP in your teaching, you've probably developed at least one demonstration that grabs the attention of your students. It conveys clear understanding about a profound idea or theory. It is fun to run and the students get excited about it.

If you have such a demo, we invite you to share it. We plan to publish a series of feature articles in the JMPer Cable newsletter about instructional demonstrations, and we would like you to take center stage. Email your favorite classroom demonstration to meredith.blackwelder@jmp.com.

Look for JMP at these Conferences

June 9-10, 2004	Quality Expo Detroit in Novi, MI
August 8-12, 2004	JSM (Joint Statistical Meetings) in Toronto, Canada
August 30-31, 2004	JMP User Conference in Cary, NC
September 26-28, 2004	Midwest SAS Users Group (MWSUG) in Chicago, IL

Demonstrating a Taylor Power Series Expansion with JMP

Mark Bailey, SAS Statistical Training & Technical Services

Bradley Jones, JMP Development

Sometimes, we know of a good theoretical model for our system. We understand the essential mechanism well enough to describe it with concise mathematical functions. These functions may serve as an appropriate surrogate (model) for the real system. They mimic important features while ignoring certain details. They enable us to predict the results of particular inputs to the system.

A good example of this kind of system is a chemical reaction. Knowledge about the reaction mechanism provides a complete description in the form of differential equations. For example, the reaction rate is observed to depend in certain ways on the instantaneous activity of the reactants and products and on their state of matter. From calculus, the solution to these equations is a rate equation based on the exponential function.

Often we do not have any theory to give us such a model. What do we do in such a case?

We may not know the true model, but we know the following from the calculus: any function can be approximated by way of a Taylor power series expansion about some arbitrary origin, *a*. (If *a* is zero, then this special case is known as the Maclaurin series.) The series is infinite, but it is usually truncated to a finite degree for practical reasons. The interval of the domain of the function for which this expansion is useful is all of the values for which the series converges.

Only the first few terms may be necessary to achieve an acceptable approximation. Let's see how this scheme works. We will start with the general exponential function.

After running demoExpTaylorSeries.jsl (downloadable from http://www.jmp.com/news/jmpercable), you will get the prompt shown in Figure 10. The initial parameter values are set so that the function of x is simply e^x and the origin is zero. Click **OK** to proceed.

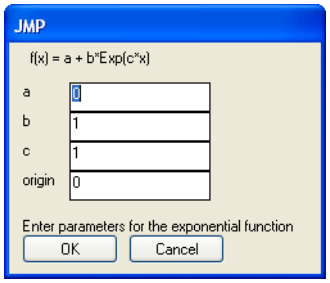

Figure 10: Dialog for exponential function expansion.

The graph of the function (Figure 11) appears on the left as a solid black line with visual registration of the origin by a dotted vertical black line. The values of the function parameters are determined by the controls on the right side of the window.

Let's investigate the approximation of this function by a truncated Taylor series. In fact, let's use a single term for a first order approximation. Click the **1st Order** button. The display changes to include the graph of the approximation (dashed red line) with the graph of the function, as shown in

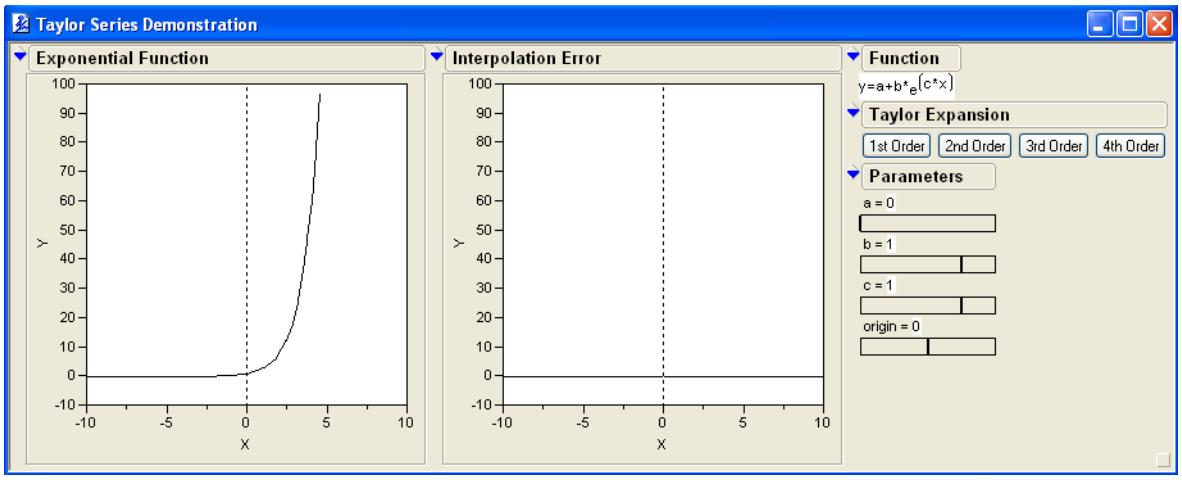

Figure 11: Demonstration window for exponential function.

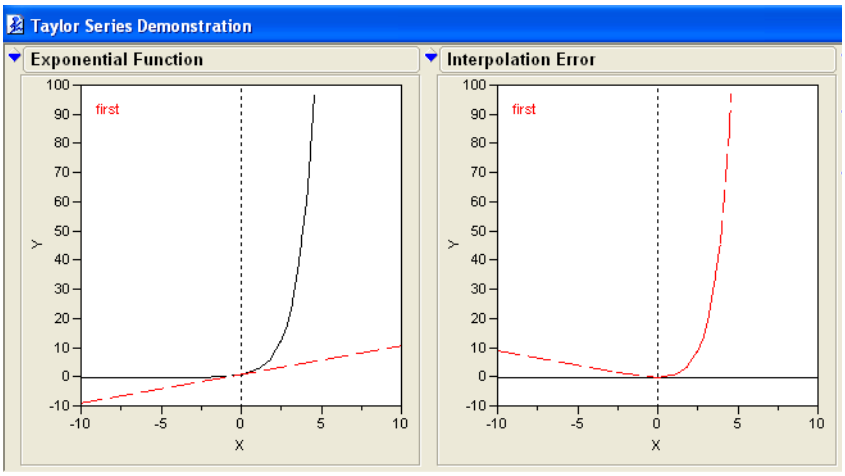

Figure 12: First order approximation.

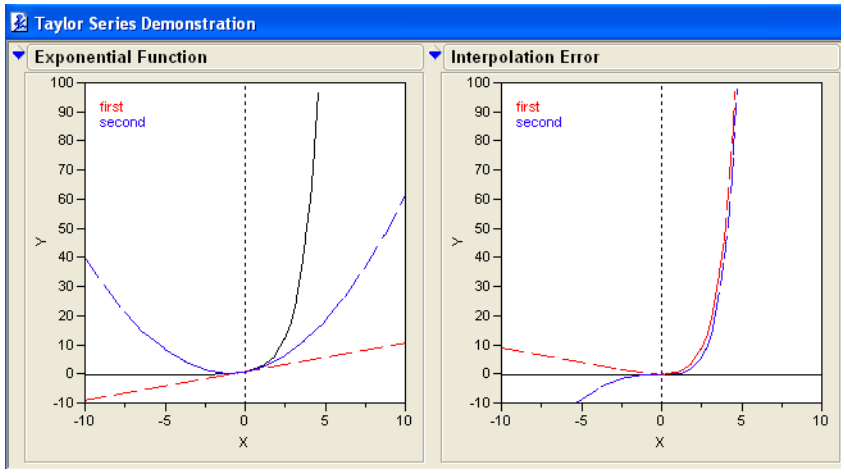

Figure 13: Second order approximation.

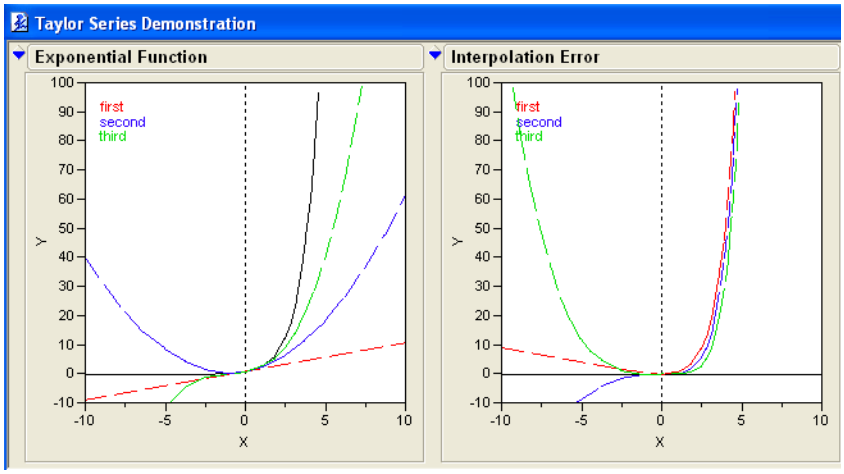

Figure 14: Third order approximation.

Figure 12. The plot on the right in Figure 12 shows the error for this approximation (i.e., the difference between the function and its approximation).

Notice that this approximation is very good at the origin but the error grows quickly even short distances away from the origin, especially where the function changes rapidly. We can do better. Click the **2nd Order** button. As shown in Figure 13, we can see that the additional term (dashed blue line) captures more of the features of the function and the error is reduced. The approximation is improved over a wider domain.

Hint: Use the magnifier tool to explore both plots and see the differences from a closer vantage point.

What you see in Figure 13 is a case where "more is better." The improvement from adding another term to the series is dramatic when there are only a couple of terms to begin with. Click the **3rd Order** button next. As shown in Figure 14, we can see that in spite of the eventual divergence, the new approximation (green dashed line) works better for a greater interval.

Of course, we can finally add the last term in this demonstration, but let's also consider some other scenarios. Click the **4th Order** button and notice the purple dashed line. What if the slope of the function was less steep ($c<1$) or the initial value was smaller ($b<1$)? Change the value of both b and c to 0.5 by typing in the text box next to the parameter name. Notice that the change in the function is smaller so the

approximation is better for a larger span or domain (Figure 15).

The fourth order approximation is clearly an improvement over all of the previous series. However, if we want to predict away from zero, the error may be great. Our current series is not very good, although our fourth order approximation is still close. The answer is to expand the series about a different point *a* by changing the origin from 0 to 5 (Figure 16).

Our function is changing more rapidly in Figure 16 than at the previous origin. Therefore, the useful range of this approximation is more limited, but the error is greatly reduced in the neighborhood of five. We cannot eliminate error for a large interval, even with many terms, but we do well in a reasonable neighborhood of a point of interest. This point might be where our system responds within design goals.

Periodic Functions

Another common class of functions is the set that regularly cycle, such as sine and cosine. Another demonstration script, demoTrigTaylorSeries.jsl (downloadable from http://www.jmp.com/news/jmpercable), shows what this looks like. After opening and running demoTrigTaylorSeries.jsl, you will get the prompt shown in Figure 17. The initial parameter values are set so that the function is simply sine *x* and the origin is zero. Click **OK** to proceed.

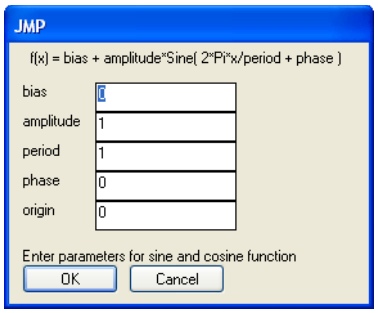

Figure 17: Dialog for trigonometric function expansion.

When the graphs appear, click the **1st Order**, **2nd Order**, **3rd Order**, and **4th Order** buttons. Then click the **Sin** button. Now change the parameters and origin to show the same principles as we saw above for the exponential function. Namely, the higher order series is a better approximation for a greater extent of the function domain and the approximation is best where the function is changing the least. Figure 18 shows the results. Notice that the bias and the phase are shown as dotted horizontal and vertical black lines, respectively, and the origin is now a solid vertical black line.

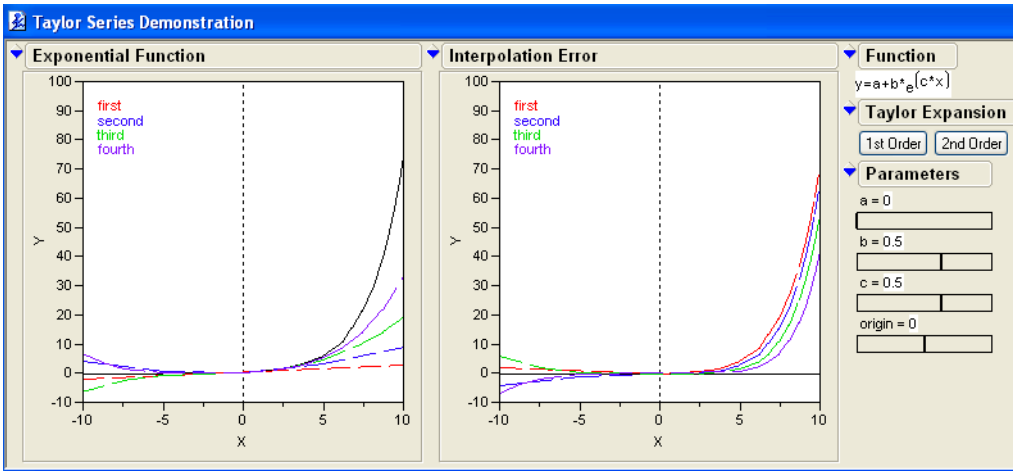

Figure 15: Fourth order approximation and shallower exponential.

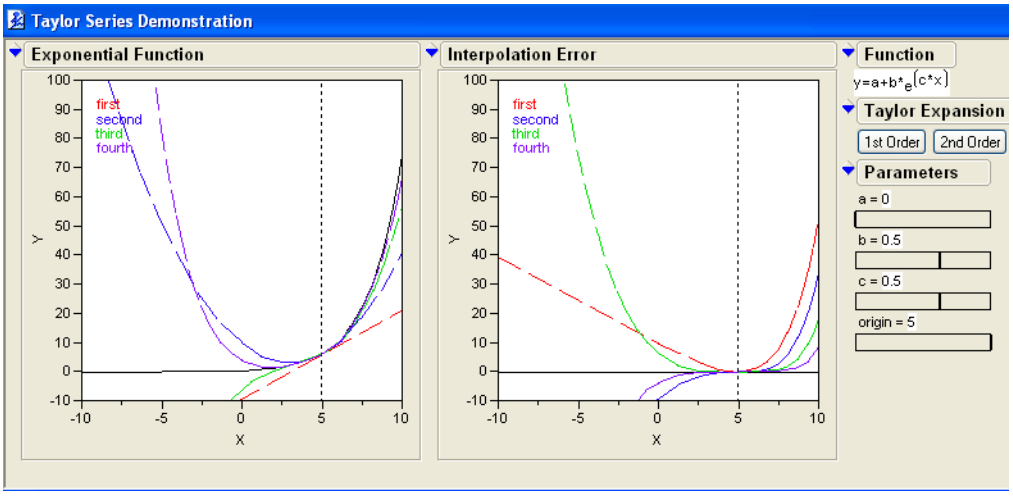

Figure 16: Expand about new origin of interest.

Design of Experiments

We often use second-order polynomials (a truncated power series) for the empirical interpolating function in the analysis of a designed experiment. This approximation is good near the center of the experiment. Polynomials are linear in their parameters (e.g. coefficients) so we can apply the vast statistical theory and methods behind linear models

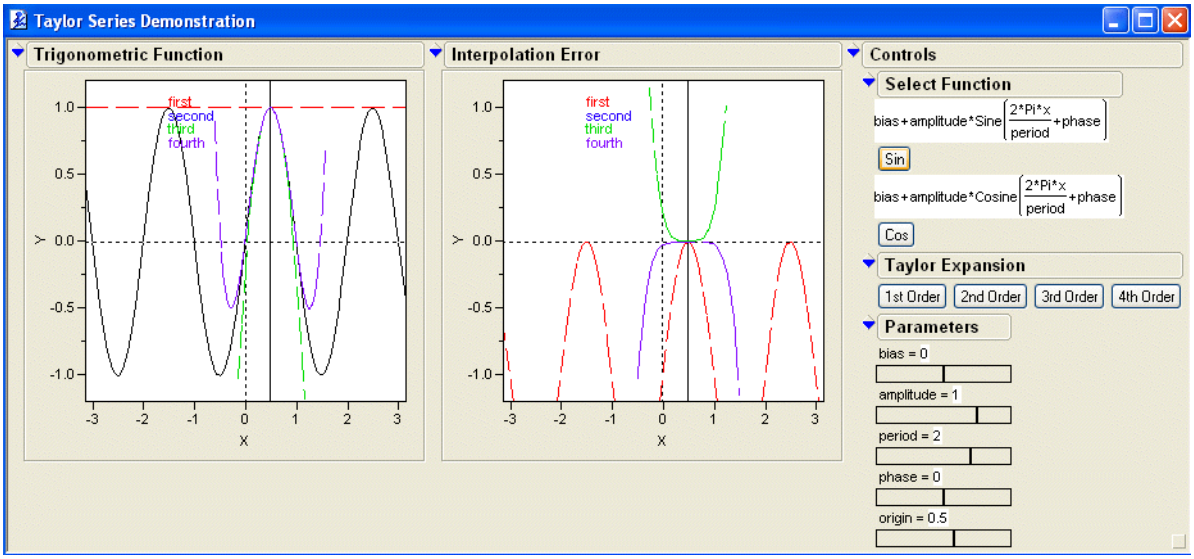

Figure 18: Approximating sine function.

to DOE problems. Experiments can be designed to be optimal for a given analysis (function). Polynomials are easily fit and evaluated using multiple (least squares) regression. They are easily solved for maxima.

We will address this application of the power series at length in a future article.

References

Hughes-Hallett, Gleason, McCallum, et al. *Calculus: Single Variable, Third Edition*, John Wiley & Sons, Inc., 2002.

SAS Institute Inc. *JMP 5.1 Scripting Guide*. Cary, NC: SAS Institute Inc., 2003.

JMP and iGrafix: Making Effective Business Process Decisions

iGrafx® Process™ 2003 for Six Sigma-plus-JMP® statistical analysis software combines business process mapping and simulation with statistical analysis using an easy-to-understand visual framework. This makes it possible to recreate any business process, run what-if simulations, and conduct in-depth statistical analysis of the results without the cost of a real world test. The result? Business process improvement options you can trust.

iGrafx provides process mapping and process simulation. JMP provides virtual experimentation and in-depth analysis of multifactor changes to the process.

The two software tools work together, enabling you to design an experiment in JMP, port the experiment settings back to iGrafx, and simulate the experiment prior to implementation. This virtual experimentation makes it easy to identify Six Sigma projects, develop and implement improvement plans, and measure results through in-depth statistical analysis. The benefits are faster identification of projects and solid statistical analysis to demonstrate the results.

iGrafx Process 2003 for Six Sigma-plus-JMP represents a risk-free environment in which to carry out multi-factor, multi-level Design of Experiments (DOE). Not only can you qualify the potential dividends, but process simulation and analytics can expose changes that, in the real process, have negative impacts. Put your office, factory, supply chain, and more on your desktop for unlimited, risk-free, zero-cost experimentation.

For a business case example, see http://www.jmp.com/partners/jmp_and_igrafx.shtml.

Meet the Trainer: An Interview

When considering or scheduling a training course, do you wonder what your instructor will be like? To give you the inside scoop on JMP training instructors, the JMPer Cable will feature interviews with them in upcoming issues. We'll give some background on their statistics knowledge and interests, along with a few fun facts you can bring up in class.

Introducing....Lori Rothenberg

Our first feature spotlights Lori Rothenberg, a full-time JMP instructor in the Statistical Training & Technical Services group.

Lori Rothenberg

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

Lori: I came to SAS with a solid background in quantitative methods, program evaluation, and the science of learning. My statistics and analyses experiences include structural equation modeling and methods for testing and assessment. While a professor, my academic responsibilities included many committees and teams for funding, recruitment, and consulting. I wanted to combine all these skills into one job, and I've been able to use all of this experience in my diversified role in JMP training.

JMPer Cable: Tell us about your history with and role in JMP training.

Lori: I started with the group five years ago and was promoted three years ago to lead the JMP training program. I actively direct course revision and curriculum development. I manage and support a growing team of contract instructors around the world.

JMPer Cable: In addition to managing the group, do you teach JMP classes?

Lori: Yes, I still find time to teach JMP classes! I taught the first class of the new two-day edition of the JMP scripting course in February, 2004. I strive to be acutely aware of the needs of both students and instructors, and I try to draw on this awareness when I'm back in the manager role.

JMPer Cable: Of which projects are you the proudest?

Lori: Piloting the first live Web course for SAS, *Getting Started with JMP*, which set the standard for subsequent Web courses. I also take pride in launching the JMP mentoring services program for users that need extraordinary help.

JMPer Cable: Recently, you spearheaded the start of the 2004 JMP User Conference and the launch of our Six Sigma training and service program. Tell us about that.

Lori: Both of these are very exciting projects. I have two fantastic teams working to give our JMP community more learning opportunities. JMP users will learn not only from our great trainers but also from one another. If your readers would like further information on either of these events, they can go to http://support.sas.com/jmpconf for the JMP User Conference and

(continued on page 12)

What Type of Training Do You Want?

- Have you been looking for training about a new method or analysis?
- Want to brush up on techniques that you haven't used in a while?
- Heard of new developments in a procedure that you would like to know more about?
- Is there something in JMP that you always wanted to know more about?
- Want to expand knowledge gained from a training course you took?

If you answered yes to any of these statements, then we would like to hear from you.

Please let Mark know by emailing him at mark.bailey@sas.com.

(continued from page 11)

http://support.sas.com/difss for our Design Institute for Six Sigma.

JMPer Cable: Lori, you are a person with many talents and lots of energy. One of your coworkers says he wonders if "Lori" is really the name of a small battalion of superheroes pulling off one miracle after another. He says it seems you can be in three places at once.

Lori: (laughs) Well, I am not quite that good. But I enjoy what I do because I work with so many talented colleagues and customers. The issues facing business people today are fascinating. I get a great satisfaction out of seeing students in JMP training classes learn and succeed.

Between managing instructors and teaching classes, things change fast in "Lori's World." We'll keep you posted on the latest developments. In the meantime, we hope you will see Lori in the classroom of a JMP training class.

About JMPer Cable
Issue 14 Spring 2004

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

Contributors
Mark Bailey
Annie Dudley Zangi
Bradley Jones
Paul Nelson
Lori Rothenberg

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

Copyright 2004, SAS Institute. All rights reserved. SAS, JMP, JMPer Cable, and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates registration. Other brand and product names are trademarks of their respective companies.