Paper 008-2011

Using a JMP® Add-In with SAS® to Evaluate the Measurement Process: Going Beyond the Traditional Gauge R&R

José G. Ramírez, Ph.D., W.L. Gore and Associates Inc., Landenberg, PA
Jon T. Weisz, SAS Institute Inc., Cary, NC

ABSTRACT

In this paper we demonstrate how the JMP Scripting Language (JSL) was used to create an application to convert a SAS/IntrNet® web app, designed for performing EMP (Evaluating the Measurement Process) studies, to run from within JMP. The original SAS code was not modified, but the html form code was converted to JSL and some of the SAS graphical output was rewritten to use native interactive JMP graphics. The JMP app takes advantage of the many SAS® analytical procedures (PROC SHEWHART, PROC MIXED, PROC ANOM), and the powerful visualization tools that the JMP environment provides. This gives the user an easy and efficient way of analyzing EMP studies.

INTRODUCTION

SAS has several technologies that are used by customers to create custom applications. Two of these technologies are:

- SAS/AF® SCL language used to create interactive client applications,
- SAS/IntrNet® software used to create web based applications and reporting systems by customers.

Both of these technologies leverage the server for data and computing services while using the client primarily as a graphical user interface (GUI).

SAS' JMP product (pronounced jump) is an interactive standalone platform for data analysis and exploration. The product also includes a SAS language called the JMP Scripting Language (JSL). JSL has some similar capabilities to SCL. Both languages have the capability to create GUI elements like dialogs, graphics and textual output. JSL allows automation of the JMP interface (scripting) and offers a full programing language to manipulate various data elements, text and images. JSL also includes full support for submitting SAS code to local or remote servers. The SAS support encompasses data set passing, JSL variable to SAS macro variable passing and rich handling of SAS ODS results.

This paper demonstrates one application of JSL as a SAS GUI language. The example is the conversion of a SAS/IntrNet® application from a web application to a rich client application using JMP. The SAS code is not modified, but the html form code is converted to JSL and some of the SAS graphical output is rewritten to use native interactive JMP graphics.

We use an example, Gauge 109, to show the results of the Gauge R&R analysis using JMP, and compare them with the results of using the JMP Add-In tool for conducting EMP analysis.

PROBLEM MOTIVATION

SAS/IntrNet® applications tend to be focused on reporting problems. An HTML form is created to gather user input, submitted to SAS code on a server, then results are fed back to the browser using ODS. These applications are fairly simple to create and SAS/IntrNet® offers rich support for scalability. These applications are limited in the client side interactivity and intimacy with the data. The data stay on the server and the reports sent back to the user offer limited capabilities for customization or modification.

Swapping out the client technology from a browser to JMP offers high levels of interactivity for SAS generated reports and user intimacy with the data for further analysis. These applications also offer the benefit of lower user support because users can easily modify and reformat reports to meet their needs without asking for programing modifications of the SAS code on the server.

This paper will show the swapping out of browser for JMP clients using a.SAS/IntrNet® based application developed to analyze data from EMP studies. The application and statistical details are covered in other sections of this paper.

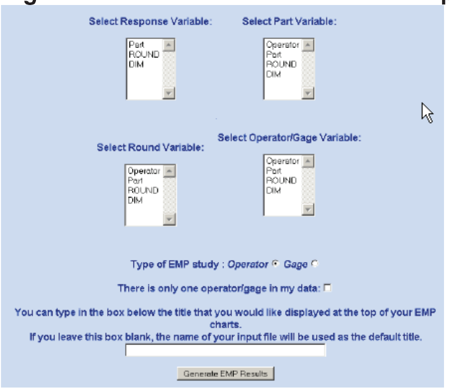
CREATING THE USER INTERFACE IN JSL

The browser based GUI set up screen for the EMP tool application asks for user input for four variables:

1. Response Variable,
2. Part Variable,
3. Round/Trial Variable,
4. Operator/Gage Variable.

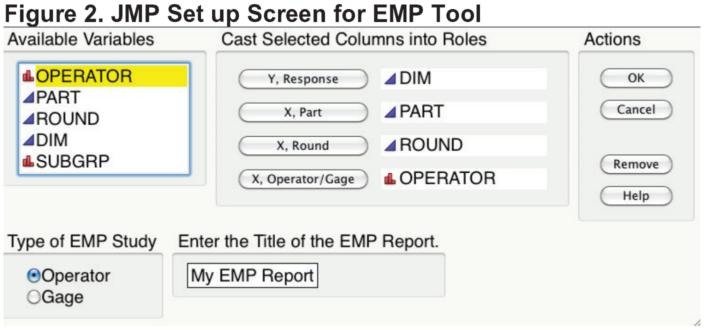
A few other questions are asked and the opportunity to provide a title is offered with a text box field. A screen shot of the browser based GUI is in figure 1 below.

Figure 1. EMP SAS/IntrNet® Tool Set up Screen



Here is the same dialog developed using JSL.

Figure 2. JMP Set up Screen for EMP Tool



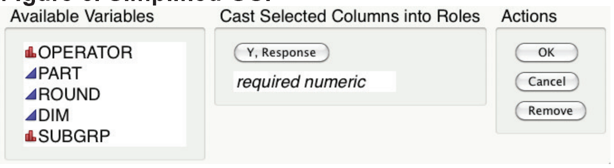
The following JSL was used to create the “Available Variables” portion of the dialog in figure 2.

```
//Build column source selection
sourceCol = Panel Box("Available Variables", sourceColList = Col List Box( All ) );
```

The “Panel Box” code is used to create the outlined box around the available variables and add a title. The “Col List Box” function creates the list box for the variables available in the current data set being used in JMP. The “ALL” argument indicated that all variable types could be listed (options are <ALL>, <Character> or <Numeric>).

This type of code is used to create all the elements of the dialog in figure 3. Figure 3 is a simplified version of Figure 2 to help illustrate the concepts in this paper.

Figure 3. Simplified GUI



Completed code for these elements is below:

```
//Build column source selection
sourceCol = Panel Box( "Available Variables", sourceColList = Col List Box( All ) );
//Build response variable target
responseVar = Panel Box( "Select Response Variable", Col List Box( numeric ) );
//Build response var and select button
colListY = Col List Box( numeric );
colListYB = Button Box( "Y, Response",
   colListY << Append( sourceColList << GetSelected ) );
```

Note that the function Col List Box creates an object in memory. This object can receive messages that can be used to modify the data in the Col List Box object. The message "Append" is used to add the user's selection to the Col List Box "colListY". The message "GetSelected" is used to retrieve the selected data in the Col List Box "sourceCol".

Now we need to place these dialog objects into a neat layout as shown in figure 2. We need a top-level container and two rows of objects with panel box objects used again to hold and provide titles.

The "New Window" function is used to create a top level object to display and hold the other graphical elements of the dialog. "H List Box" acts as horizontal box containing display objects, it organized display object horizontally. Importantly below you will see a button box object named "OK". This is the button that is used to initiate the collection JSL variables from the dialog and submit the SAS code.

```
//Build GUI window
custdial = New Window( "EMP Setup",
        H List Box(
                H List Box( sourceCol, Panel Box( "Cast Selected Columns into Roles", colListYB,
colListY ) ),
                Panel Box( "Actions",
                        Button Box( "OK", OKScript ),
                        Button Box( "Cancel",
                                custdial << CloseWindow;
                                Throw();
                        ),
                        Button Box( "Remove", colListY << RemoveSelected ),
                )
        )
);
```

SUBMITTING SAS CODE

SAS code is submitted to a SAS connection using the SAS Submit message. To establish a connection to local PC SAS, the following code can be used.

```
sasconn = SAS Connect();
```

It is also possible to connect to a metadata server or remote workspace server using options with the SAS Connect function. SAS code can now be submitted using the following code that is evaluated when the "go" button is clicked. Note that using the DeclareMacros option with a JSL variable of the same name creates the macro variable &stringY. Using the getItems message grabs the value of the selected item in the dialog and populates the JSL variable StringY. The SAS Name function is then used to ensure it is SAS safe.

```
//GO Script
OKScript = Expr(
      custdial << CloseWindow;
      StringY = Char( SAS Name( colListY << getItems ) );

      sasconn << SAS Submit(
            "
           proc means data=work.data n mean max min range std fw=8;
            var &stringY;
          run;

              ",
            DeclareMacros( stringY, ),
            ODS( 1 ),
            ODSGraphics( 1 ),
            OpenODSResults( 0 ),
            noOutPutWindow
      );
```

In this case options are used to pass JSL variables to SAS macro variables using the DeclareMacros option. ODS options are also set and an option is set to not bring up the SAS Output Window when the submission is complete.

Now let's examine the SAS Log from the run:

```
%let stringY=%NRBQUOTE(DIM);

proc means data=work.data n mean max min range std fw=8;
 var &stringY;
run;
```

Notice that JMP generated the %let code to pass the JSL variable to SAS as a macro variable. Similar methods where used in submitting the more complex EMP SAS code to the SAS server for processing of the users set up.

MANIPULATING SAS RESULTS

To complete the creation of a custom JSL viewer or our results, we need to get the PROC MEANS output and mix those with a JMP native graphic of the histogram in this case. This is done by creating a top level container to hold our results with the New Window() function. This function takes a title argument and then can hold any number of display boxes (a JMP display box object). The use of the V List Box is the same as for the launch dialog, it aligns the display boxes in a vertical layout.

The key new feature introduced in this section is the use of indexing into a display tree. The output in Figure 4 is just the PROC MEANS output displayed in a JMP report format. The display tree is shown below the report

Figure 4. PROC MEANS Output and Display Tree

The SAS System

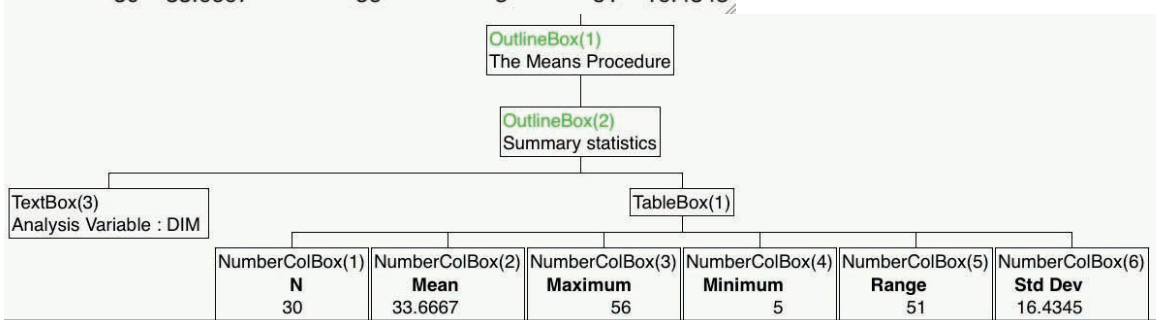
The MEANS Procedure

The Means Procedure

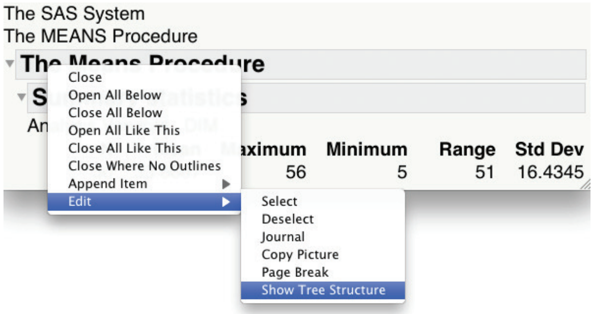
Summary statistics

Analysis Variable : DIM

N	Mean	Maximum	Minimum	Range	Std Dev
30	33.6667	56	5	51	16.4345



The display tree is displayed by right clicking the mouse in the title of an outline box in a report then selecting Edit as shown below.



We need the display tree because we need to index into the tree and select object to include in our custom viewer.

GETTING REFERENCES TO SPECIFIC SAS OUTPUT IN SAS RESULTS

The object we want in this simple example is the outlinebox named “The Means Procedure”. The get a reference to that object, we just index the reference to the JMP report that we named MeansResults. The index method uses square brackets and the name as an argument to the object type. The code snipped below shows how to get a reference to this outline box.

```
Reference = MeansResults[Outline Box( "The Means Procedure" )]
```

ASSEMBLING RESULTS VIEWER IN JMP

Now let’s put together our custom viewer. The viewer is to include the PROC MEANS output as a JMP report, and to include JSL code to create a histogram. The histogram code is generated by choosing “Copy Script” in the Script sub menu of the distribution platform in JMP. This code generation is available in any JMP report.
The following code forms our completed custom results viewer. The viewer for the EMP tool is similar, just handles more content and has lots of indexes into the more complex SAS display tree of results.

```
//Create Results Viewer
        ResultsViewer = New Window( "My Results",
                V List Box(
                        Outline Box( "Means Results", MeansResults[Outline Box( "The Means
Procedure" )] ),
                        dt = Distribution(
                                Continuous Distribution(
                                        Column( Column( stringY ) ),
                                        Quantiles( 0 ),
                                        Moments( 0 ),
                                        Vertical( 0 )
                                )
                        )
                )
        );
```

Figure 5 shows the results of our custom results views. The histogram is live to the data, so selecting bars on the histogram highlights those rows in the data set. The means results are also a live report, so users can format number of significant digits, select columns, create a data table from the report, etc.

Figure 5. Custom Results Viewer

Means Results

The Means Procedure

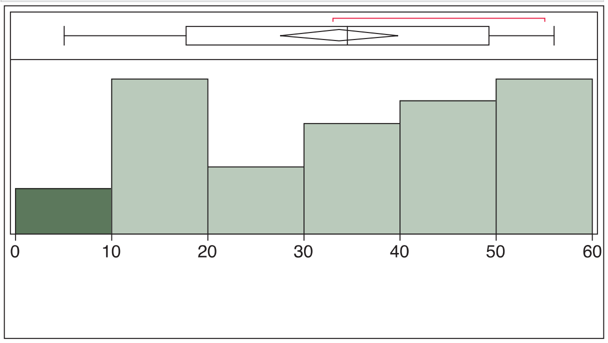
Summary statistics

Analysis Variable : DIM

N	Mean	Maximum	Minimum	Range	Std Dev
30	33.6667	56	5	51	16.4345

Distributions

DIM



GAUGE 109, AN EXAMPLE OF A GAUGE R&R AND EMP ANALYSES

The Gauge 109 data comes from Chapter 6 of EMP III Using Imperfect Data (Wheeler (2006)). Five parts were selected over a period of five days, and were measured by 3 operators (A, B, C) in 2 rounds for a total of 30 measurements. The measurements represent the dimension of a part to the nearest micron. We would like to know if the three operators measure in a similar way, if the have similar test-retest errors, if the gage has a good enough resolution, etc.

1. TRADITIONAL GAUGE R&R ANALYSIS USING JMP V 9.0.1

Gauge R&R studies are traditionally used (AIAG's Measurement System Analysis) to assess the ability of a measurement system to give consistent readings when done several times (repeatability) and when using different operators (reproducibility). A Gauge R&R study helps classify a measurement system into good, marginal, or acceptable. An EMP study goes beyond a Gauge R&R by providing visual tools for learning and improving the measurement process, to get it to operate to its fullest potential.

Gauge R&R studies or the Automotive Industry Action Group (AGIG) Measurement System Analysis (MSA) are traditionally used to assess the ability of a measurement system to give consistent readings when done several times (repeatability) and when using different operators (reproducibility). In its basic form a study is set up using different operators, which measure different parts two or three times. The data is analyzed by calculating

- Repeatability – Equipment Variation (EV)
- Reproducibility – Appraiser Variation (AV)
- Part Variation (PV)
- Repeatability & Reproducibility (R&R)

The calculations are based upon predicting 6 sigma (AGIG MSA 3rd edition), or 99.73% of the area under the normal distribution. Guidelines are given for the acceptance of a measurement system based of the value of R&R:

- Under 10% error — The measurement system is acceptable
- 10% to 30% error — The measurement system may be acceptable
- Over 30% error — The measurement system needs improvement

Gauge R&R studies are, for the most part, aimed at classifying the system and suggesting “improvement” (R&R > 30%), without clear indication of what to improve. An EMP (Evaluating the Measurement Process) study, on the other hand, offers

A Gauge R&R can be performed in JMP using the Variability/Gauge Chart platform. Figure 4 shows the required inputs for the analysis, while Figure 5 shows the results.

Figure 4. JMP Variability/Gauge Chart Platform

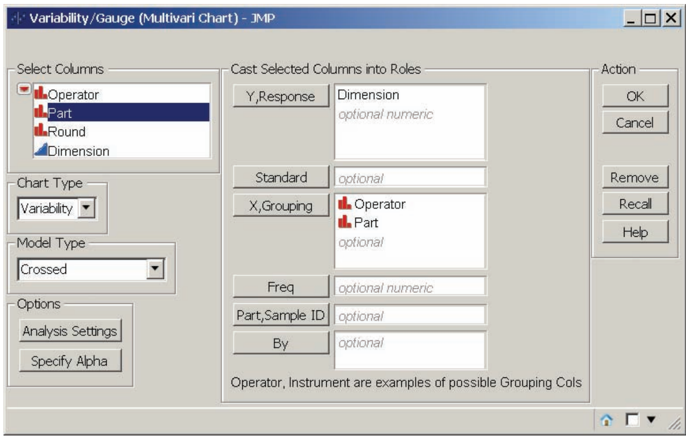
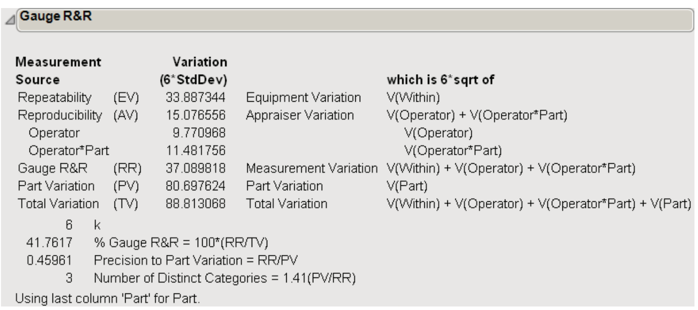


Figure 5. JMP Gauge R&R Results



Gauge R&R

Measurement Source		Variation (6*StdDev)		which is 6*sqrt of
Repeatability	(EV)	33.887344	Equipment Variation	V(Within)
Reproducibility	(AV)	15.076556	Appraiser Variation	V(Operator) + V(Operator*Part)
Operator		9.770968		V(Operator)
Operator*Part		11.481756		V(Operator*Part)
Gauge R&R	(RR)	37.089818	Measurement Variation	V(Within) + V(Operator) + V(Operator*Part)
Part Variation	(PV)	80.697624	Part Variation	V(Part)
Total Variation	(TV)	88.813068	Total Variation	V(Within) + V(Operator) + V(Operator*Part) + V(Part)

6	k
41.7617	% Gauge R&R = 100*(RR/TV)
0.45961	Precision to Part Variation = RR/PV
3	Number of Distinct Categories = 1.41(PV/RR)

Using last column 'Part' for Part.

The results indicate that Gauge 109 is in need of improvement because the %Gauge R&R = 42% > 30%.

2. EMP ANALYSIS USING JMP –SAS JSL APPLICATION

An EMP study improves the traditional Gauge R&R by offering a different classification of the measurement system, as well as clues that can be used to improve the measurement system, if needed. The output of an EMP study is primarily graphs, making it easier for the analyst to interpret the results.

Figure 6. EMP Primary Graph

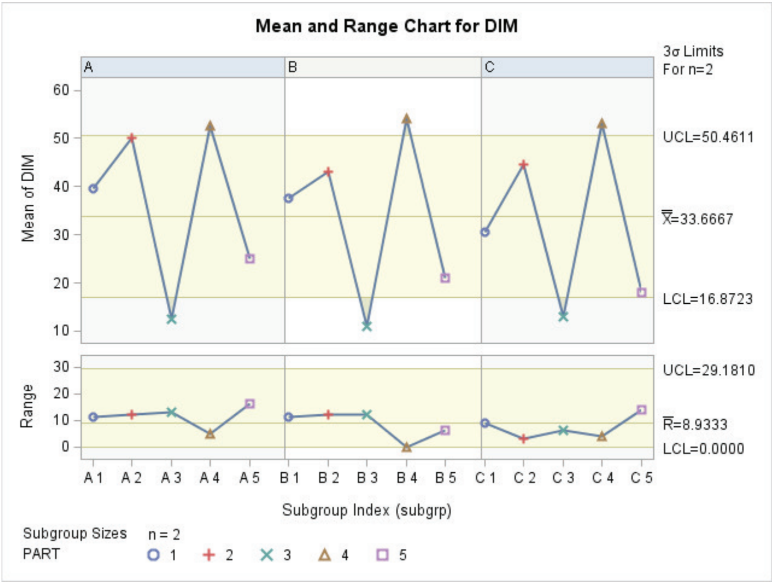
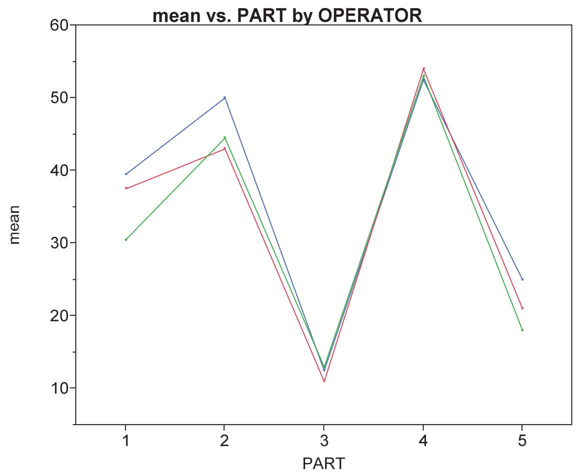


Figure 6 shows the primary EMP chart that gives a global view of the results. The points in the chart are color-coded to reflect the different parts used in the study, and the chart has been broken down by Operator. A quick glance shows good parallelism between the traces for Operators A, B and C, indicating agreement. The Range charts gives a visual comparison of the test-retest error for the three operators, with an average range of about 9 units.

A parallelism plot, Figure 7, helps to compare the operator profiles. We can see that for Parts 1 and 2 Operator A measures higher than Operators B and C.

Figure 7. EMP Parallelism Plot



The numerical results of the EMP study are given in Tables 1 and 2. The reported test-retest error is 7.91 units, with a probable error of 5.34 units; i.e., the measurements coming from the gauge 109 system are good to ±5.34 units. The Reproducibility impact, Operator effect, is only 0.15% of the total variation, with an intraclass correlation coefficient of 0.881, indicating a First Class system. Wheeler (2008) states that “First Class Monitors will have only slight attenuation for signals coming from the production process (less than 10%) while signals from the measurement system are greatly attenuated (more than 55%).” Therefore, contrary to what the traditional Gauge R&R study suggests (the measurement system was condemned because the %Gauge R&R = 42% > 30%) the Gage 109 measurement system is pretty good.

Table 1. EMP Results

EMP Results for DIM

Repeatability (Test-Retest)	Effective Degrees of Freedom	Probable Error	IntraClass Correlation* (No Bias)	IntraClass Correlation* (with Bias)	Reproducibility Impact	System Classification	Potential Classification
7.9170	13.38	5.3399	0.882	0.881	0.15%	First Class	First Class

The test-retest error calculation is one of the many dimensions of an EMP study. Users are also interested in the effective (actual) resolution of the measurement system. In terms of resolution, an EMP study compares the numerical resolution of the gauge with the resolution of the measurement system. Table 2 shows the results indicating that, as compared to the Probable Error, the gauge is recording too many digits. In other words, the stated resolution of the gauge is worst than the calculated test-retest error.

Table 2. EMP Effective Resolution of the Measurement System

Effective Resolution for DIM

Probable Error	Lower Bound Measurement Increment	Measurement Increment	Smallest Effective Measurement Increment	Largest Effective Measurement Increment	Action
5.3399	0.534	1	1.1748	11.7478	May drop a digit

The EMP application also provides the Gauge R&R percent breakdown as shown in Table 3. The results indicate that most of the variation is coming from the product (88%), while the %R&R component is about 12%.

Table 3. Gauge R&R Breakdown

R&R Results forDIM

Component	Std. Dev.	% of Total Variation
(EV+AV) Gage R & R	6.1691	11.94%
(EV) Repeatability	6.1241	11.77%
(AV) Reproducibility	0.7436	0.17%
(PV) Product Variation	16.7505	88.06%
(IV) A x P Variation	0.0000	0.00%
(TV) Total Variation	17.8504	100.00%

CONCLUSION

Converting an existing SAS web application, or other client technology, to a JMP client is fairly straight forward if the client code mostly collects user data, and the SAS code on the server side does most of the report and analytic work.

Rendering SAS output as JMP reports has the advantage that users can customize the reports and have access to the underlying data for further analysis.

JMP 9 introduced add-ins so that applications can be packaged into a single binary that includes any source code and menu customizations. This allows applications to be distributed and installed. The JMP menus can be extended to include any number of custom applications to meet the needs of a specific organization.

REFERENCES

Automotive Industry Action Group. (2002) Measurement Systems Analysis Manual,

Wheeler, Donald J. (2006) EMP III Using Imperfect Data. Knoxville: SPC Press.

Wheeler, Donald J. (2008) “An Honest Gauge R&R Study”.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Name: José G. Ramírez
Enterprise: W.L. Gore and Associates, Inc.
Address: 380 Starr Road
City, State ZIP: Landenberg, PA, 19350
Work Phone: 610 268-1805
Fax:
E-mail: jramirez@wlgore.com
Web: www.gore.com

Name: Jon Weisz
Enterprise: SAS Institute Inc
Address: SAS Campus Dr.
City, State ZIP: Cary, NC 27513
Work Phone: 919 531 3576
Fax: 919 677 4444
E-mail: jon.weisz@jmp.com
Web: www.jmp.com