SAS Institute Inc.

Assessing the Numerical Accuracy of JMP® 8.0.2

Statistical Discovery.™ From SAS.

Revised on 18 Mar 2010

Measuring Accuracy

There are many sources of error in statistical computation, including rounding error, truncation error, and the finite-precision inaccuracy involved in representing a number in binary form.

To measure these sources of error, we looked at the significant digits reported by JMP calculations. Significant digits are the first nonzero digit and all succeeding digits. For example, 3.14159 has six significant digits, while 0.00314 has three. A frequently used measure of the number of correct significant digits is the (common) logarithm of the relative error (LRE), calculated as

$$LRE = -\log(| q - c | / | c |)$$

where q is the reported value and c is the expected value. This quantity is not defined when $q = c$. In that situation, the LRE is given the value of the number of significant digits in c. Also, there are situations where the expected value is zero, which also results in an undefined LRE. In these cases, the LRE is defined as the logarithm of the absolute error

$$LRE = -\log(| q |)$$

The LRE is approximately analogous to the number of significant digits of accuracy of a reported value compared with its expected value.

It is worth noting that the LRE is valid only for values of q that are close to c. Therefore, any calculated value that differs from c by more than a factor of 2 is set to zero. Finally, any value of the LRE greater than the number of digits in c is set to the number of digits in c.

We use the Greek symbol lambda (λ), with a subscript, to represent the LRE. The subscript denotes the parameter to be estimated. For example, λ_{rsq} is the LRE between a reported r square value and its expected value.

Statistical Standards

The National Institute of Standards and Technology (NIST) provides the Statistical Reference Data Sets (StRD) to assist in the evaluation of the numerical accuracy of statistical software. More information about these data sets is available at www.itl.nist.gov/div898/strd/.

The StRD data sets are the subject of this paper.

The following sections report the results of the tests using JMP® 8.0.2 for the Microsoft® Windows® XP operating system. All tests used the same build: Version 8.0.2, Build Date: Oct 08 2009, 02:18:52, Release.

Univariate Results

The univariate tests consist of nine data sets, ranging from 3 to 5000 data points. Each data set has certified values to 15 decimal places for the mean (μ), standard deviation (σ), and first-order autocorrelation (ρ). Therefore, a λ of 15 indicates perfect agreement with NIST certified values. Results for μ and σ were calculated using the JMP Distribution platform. Values of ρ came from the Time Series platform. The results are presented in Table 1.

Table 1: Univariate Results

Data Set	Difficulty	λ_μ	λ_σ	λ_ρ
PiDigits	Low	15.0	14.9	13.0
Lottery	Low	15.0	15.0	15.0
Lew	Low	15.0	15.0	15.0
Mavro	Low	15.0	13.1	13.8
Michaelso	Average	15.0	13.8	13.4
NumAcc1	Average	15.0	15.0	15.0
NumAcc2	Average	14.0	14.6	13.7
NumAcc3	Average	15.0	9.5	11.2
NumAcc4	Average	15.0	8.3	9.0

Analysis of Variance Results

The Analysis of Variance (ANOVA) tests contain eleven data sets, with 5 to 2001 values for each level. As in the case of the previous white paper, only the LREs of *F*, R^2, and residual standard deviation are presented here. Again, 15 decimal places are provided in the certified values, so a score of 15 in the table corresponds to perfect agreement.

JMP provides two methods of calculating an ANOVA for two-variable cases. Its most direct method is through the Fit Y by X platform, designed specifically for bivariate data. The Fit Model platform, used for fitting general linear models, can also be used. Although much of the literature on numerical accuracy only reports results for one method per software application, we report methods for both platforms because JMP uses distinct calculation routines for the two platforms. Results using the Fit Y by X platform are reported in Table 2, while results using Fit Model appear in Table 3.

Table 2: Fit Y By X Results

Data Set	Difficulty	λ_F	λ_{rstd}	λ_{rsq}
SiRstv	Low	12.4	13.1	12.4
SmLs01	Low	14.0	14.5	14.2
SmLs02	Low	13.4	13.8	13.7
SmLs03	Low	12.4	12.9	12.6
AtmAgWt	Average	8.4	9.2	8.5
SmLs04	Average	8.2	8.9	8.5
SmLs05	Average	8.0	8.6	8.3
SmLs06	Average	6.2	6.8	6.5

Table 2: Fit Y By X Results *(Continued)*

Data Set	Difficulty	λ_F	λ_{rstd}	λ_{rsq}
SmLs07	High	2.4	3.1	2.7
SmLs08	High	1.9	2.5	2.2
SmLs09	High	0.3	0.9	0.5

Table 3: Fit Model Results

Data Set	Difficulty	λ_F	λ_{rstd}	λ_{rsq}
SiRstv	Low	13.1	13.4	13.2
SmLs01	Low	14.5	15.0	14.7
SmLs02	Average	13.8	14.1	14.0
SmLs03	Average	12.4	12.9	12.7
AtmAgWt	Average	10.1	11.2	10.3
SmLs04	Average	10.4	10.6	10.7
SmLs05	Average	NR[a]	NR[a]	NR[a]
SmLs06	Average	NR[a]	NR[a]	NR[a]
SmLs07	High	NR[a]	NR[a]	NR[a]
SmLs08	High	NR[a]	NR[a]	NR[a]
SmLs09	High	NR[a]	NR[a]	NR[a]

a. Values of F and R^2 are reported as missing in the Fit Model platform.

Linear Regression Results

The linear regression portion of the test consists of eleven data sets containing 3 to 82 data points and 1 to 11 parameters to be estimated. Each data set has certified values, to 15 digits, for each parameter's estimate, residual standard deviation R^2, and the entire analysis of variance table (which provides the residual sum of squares).

As is the case with the ANOVA results, JMP provides two methods of fitting linear regressions: The Fit Y by X platform and the Fit Model platform. Results for each are provided for LREs of the parameter estimate β, its standard deviation s, and the residual sum of squares rss.

JMP also does not report an R^2 when the intercept is missing. This is the case in both the Fit Y by X and Fit Model platforms. Therefore, the two NoInt data sets have "NR" as their LRE for R^2. Similarly, the Longley data set requires a multilinear fit, which is not available in the Fit Y by X platform. Therefore, results for Longley are listed as "NR" in the Fit Y by X table (Table 4), but with LREs in the Fit Model table (Table 5). Also, the Filip data requires a tenth degree polynomial fit that is not available in either the Fit Y by X or Fit Model platforms, and so the results for Filip are listed as "NR" in Table 4 and Table 5.

Table 4: Linear Regression Results Using Fit Y By X

Data Set	Difficulty	Parameter	λ_β	λ_s	λ_{rss}	λ_{rsq}
Norris	Low	β0	12.2	11.7	11.7	15.0
		β1	14.4	11.7		
Pontius	Low	β0	11.2	8.4	8.4	15.0
		β1	13.9	8.4		
		β2	12.1	8.4		
NoInt1	Average	β1	14.7	13.5	13.5	NR
NoInt2	Average	β1	15.0	14.6	14.7	NR
Filip	High	NR	NR	NR	NR	NR
Longley	High	NR	NR	NR	NR	NR
Wampler1	High	β0	8.4	15.0	15.0	15.0
		β1	8.0	15.0		
		β2	8.4	15.0		
		β3	9.2	15.0		
		β4	10.5	15.0		
		β5	12.2	15.0		
Wampler2	High	β0	12.8	15.0	15.0	15.0
		β1	11.8	15.0		
		β2	10.9	15.0		
		β3	10.6	15.0		
		β4	10.7	15.0		
		β5	11.4	15.0		
Wampler3	High	β0	8.4	11.3	11.7	15.0
		β1	8.0	10.9		
		β2	8.4	10.8		
		β3	9.2	10.8		
		β4	10.5	10.8		
		β5	12.2	10.8		
Wampler4	High	β0	8.4	11.5	14.8	15.0

Table 4: Linear Regression Results Using Fit Y By X *(Continued)*

Data Set	Difficulty	Parameter	λ_β	λ_s	λ_{rss}	λ_{rsq}
		β1	8.0	11.0		
		β2	8.4	10.9		
		β3	9.2	10.9		
		β4	10.5	10.9		
		β5	12.2	10.9		
Wampler5	High	β0	8.4	11.5	14.8	13.7
		β1	8.0	11.0		
		β2	8.4	10.9		
		β3	9.2	10.9		
		β4	10.5	10.9		
		β5	12.2	10.9		

Table 5: Linear Regression Results Using Fit Model

Data Set	Difficulty	Parameter	λ_β	λ_s	λ_{rstd}	λ_{rsq}
Norris	Low	β0	12.4	10.8	10.8	15.0
		β1	14.4	10.8		
Pontius	Low	β0	11.6	9.4	9.4	15.0
		β1	14.1	9.4		
		β2	12.5	9.4		
NoInt1	Average	β1	14.7	13.5	13.5	NR
NoInt2	Average	β1	15.0	14.6	14.7	NR
Filip	High	NR	NR	NR	NR	NR
Longley	High	β0	13.6	14.8	13.6	15.0
		β1	12.5	14.0		
		β2	12.9	13.6		
		β3	13.6	13.7		
		β4	14.0	13.7		
		β5	12.2	13.5		
		β6	13.6	14.6		
Wampler1	High	β0	6.7	15.0	15.0	15.0

Table 5: Linear Regression Results Using Fit Model *(Continued)*

Data Set	Difficulty	Parameter	λ_β	λ_s	λ_{rstd}	λ_{rsq}
		$\beta1$	6.4	15.0		
		$\beta2$	6.8	15.0		
		$\beta3$	7.7	15.0		
		$\beta4$	9.0	15.0		
		$\beta5$	10.7	15.0		
Wampler2	High	$\beta0$	12.2	15.0	15.0	15.0
		$\beta1$	11.0	15.0		
		$\beta2$	10.6	15.0		
		$\beta3$	10.5	15.0		
		$\beta4$	10.8	15.0		
		$\beta5$	11.6	15.0		
Wampler3	High	$\beta0$	6.7	10.8	10.9	15.0
		$\beta1$	6.4	10.5		
		$\beta2$	6.8	10.5		
		$\beta3$	7.7	10.4		
		$\beta4$	9.0	10.4		
		$\beta5$	10.7	10.4		
Wampler4	High	$\beta0$	6.7	11.3	14.5	15.0
		$\beta1$	6.4	10.8		
		$\beta2$	6.8	10.7		
		$\beta3$	7.7	10.6		
		$\beta4$	9.0	10.6		
		$\beta5$	10.7	10.6		
Wampler5	High	$\beta0$	6.8	11.3	14.8	13.7
		$\beta1$	6.5	10.8		
		$\beta2$	6.9	10.7		
		$\beta3$	7.8	10.6		
		$\beta4$	9.0	10.6		
		$\beta5$	10.7	10.6		

Nonlinear Regression Results

The Nonlinear regimen consists of twenty-seven data sets, with six to 250 data points and two to nine parameters. The certified values are presented to only eleven decimal places in this suite of tests, so an LRE of 11 implies perfect agreement with the standard.

Table 6: Nonlinear Results

Data Set	Difficulty	Parameter	λ_{β}	$\lambda_{\beta std}$	λ_{sse}	λ_{rstd}
Misra1a	Low	β1	9.7	9.3	10.5	10.6
		β2	9.6	10.7		
Chwirut2	Low	β1	10.3	10.5	11.0	10.9
		β2	10.6	11.0		
		β3	10.3	11.0		
Chwirut1	Low	β1	9.4	9.7	11.0	10.9
		β2	9.8	10.4		
		β3	9.6	11.0		
Lanczos3	Low	β1	10.2	8.4	10.6	11.0
		β2	10.5	8.4		
		β3	11.0	8.4		
		β4	11.0	8.4		
		β5	10.7	8.4		
		β6	11.0	8.4		
Gauss1	Low	β1	11.0	11.0	11.0	11.0
		β2	11.0	10.7		
		β3	11.0	11.0		
		β4	11.0	11.0		
		β5	10.7	11.0		
		β6	11.0	11.0		
		β7	10.8	11.0		
		β8	10.9	11.0		
Gauss2	Low	β1	11.0	10.8	10.6	10.7
		β2	11.0	10.8		
		β3	10.4	10.9		
		β4	10.4	10.2		

Table 6: Nonlinear Results *(Continued)*

Data Set	Difficulty	Parameter	λ_β	$\lambda_{\beta std}$	λ_{sse}	λ_{rstd}
		β5	10.4	10.9		
		β6	10.8	10.5		
		β7	11.0	9.9		
		β8	10.0	9.9		
DanWood	Low	β1	10.0	10.1	11.0	11.0
		β2	10.3	10.8		
Misra1b	Low	β1	11.0	10.8	11.0	11.0
		β2	11.0	11.0		
Kirby2	Average	β1	8.5	9.3	11.0	10.8
		β2	8.7	9.0		
		β3	8.9	8.9		
		β4	8.6	8.8		
		β5	9.2	9.1		
Hahn1	Average	β1	10.0	10.7	11.0	10.6
		β2	10.2	10.3		
		β3	10.3	11.0		
		β4	9.9	10.7		
		β5	11.0	11.0		
		β6	10.5	10.9		
		β7	10.1	10.6		
Nelson	Average	β1	10.9	10.6	10.9	11.0
		β2	10.9	11.0		
		β3	11.0	10.9		
MGH17	Average	β1	9.4	10.6	11.0	11.0
		β2	8.0	7.4		
		β3	7.8	7.4		
		β4	8.5	7.8		
		β5	8.4	8.1		
Lanczos1	Average	β1	11.0	3.2	2.9	3.2
		β2	10.6	3.2		

Table 6: Nonlinear Results *(Continued)*

Data Set	Difficulty	Parameter	λ_β	$\lambda_{\beta std}$	λ_{sse}	λ_{rstd}
		β3	11.0	3.2		
		β4	10.9	3.2		
		β5	10.6	3.2		
		β6	11.0	3.2		
Lanczos2	Average	β1	11.0	8.6	10.0	10.6
		β2	10.4	8.7		
		β3	11.0	8.6		
		β4	11.0	8.6		
		β5	10.7	8.6		
		β6	11.0	8.6		
Gauss3	Average	β1	11.0	11.0	11.0	10.8
		β2	11.0	10.5		
		β3	10.7	11.0		
		β4	10.6	11.0		
		β5	11.0	11.0		
		β6	11.0	10.7		
		β7	10.5	11.0		
		β8	10.7	11.0		
Misra1c	Average	β1	9.4	9.1	11.0	11.0
		β2	9.3	9.9		
Misra1d	Average	β1	9.5	9.2	11.0	11.0
		β2	9.4	10.1		
Roszman1	Average	β1	11.0	11.0	11.0	11.0
		β2	11.0	11.0		
		β3	10.9	11.0		
		β4	11.0	11.0		
Enso	Average	β1	10.6	10.1	11.0	11.0
		β2	9.8	11.0		
		β3	9.0	10.6		
		β4	9.1	8.5		

Table 6: Nonlinear Results *(Continued)*

Data Set	Difficulty	Parameter	λ_β	$\lambda_{\beta std}$	λ_{sse}	λ_{rstd}
		β5	8.6	8.3		
		β6	7.7	8.9		
		β7	8.8	8.3		
		β8	6.8	9.1		
		β9	8.5	7.9		
Mgh09	High	β1	9.1	8.4	11.0	11.0
		β2	7.8	7.9		
		β3	8.3	8.0		
		β4	8.0	8.0		
Thurber	High	β1	10.9	9.1	11.0	10.6
		β2	8.4	7.0		
		β3	8.2	7.0		
		β4	8.0	7.0		
		β5	8.4	7.1		
		β6	8.4	7.1		
		β7	7.7	7.0		
BoxBOD	High	β1	9.8	9.4	10.4	11.0
		β2	9.3	9.2		
Rat42	High	β1	9.7	9.0	11.0	10.4
		β2	9.8	9.3		
		β3	9.4	9.5		
Mgh10	High	β1	11.0	9.9	11.0	11.0
		β2	11.0	9.9		
		β3	10.9	9.9		
Eckerle4	High	β1	10.0	9.8	10.7	11.0
		β2	9.6	9.6		
		β3	11.0	9.6		
Rat43	High	β1	10.3	9.4	11.0	11.0
		β2	9.1	9.3		
		β3	9.2	9.1		

Table 6: Nonlinear Results *(Continued)*

Data Set	Difficulty	Parameter	λ_{β}	$\lambda_{\beta std}$	λ_{sse}	λ_{rstd}
		β4	9.0	9.2		
Bennett5	High	β1	11.0	7.0	11.0	10.6
		β2	11.0	7.0		
		β3	11.0	7.0		

Appendix 1: Replicating JMP 8.0.2 numerical accuracy results

To reproduce the results reported in tables 1-6:

1. Download the compressed archive of the NIST testing framework (available at www.jmp.com/qualitystatement).

2. Uncompress the archive to a directory of your choice.

3. Locate and execute the `RunNISTTests.jsl` script. The script creates the window shown in Figure 1. This window enables you to execute either all tests or selected tests. See Figure 2 for an example of the report that is generated when the tests are run.

The NIST testing framework can facilitate operational qualification or validation of JMP. In addition to displaying the LRE for each reported value, the testing framework also compares the difference between the reported and expected values to a relative error threshold (RET) value.

The RET represents the minimum computational accuracy that we deem acceptable for our software. If the difference between the standard and computed values is less than the RET, the test is passed. The resulting report shows the status of the test, the NIST standard value, the value actually computed, the LRE, and the RET. In this way, the tests serve as a tool to demonstrate that JMP is operating as expected.

Note: This framework is a specialized version of the framework described in the article titled "Unit Tests: Automated JSL Testing" that appeared in the Fall 2007 (issue #23) JMPer Cable® newsletter (available at www.jmp.com/about/newsletters/jmpercable/pdf/23_fall_2007.pdf). The framework consists of a set of JSL scripts and JMP data tables corresponding to each of the StRD data sets mentioned in the previous sections. See Appendix 2 for additional details.

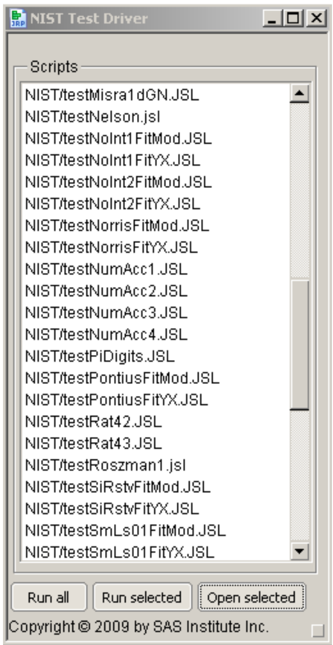

Figure 1: NIST Test Driver

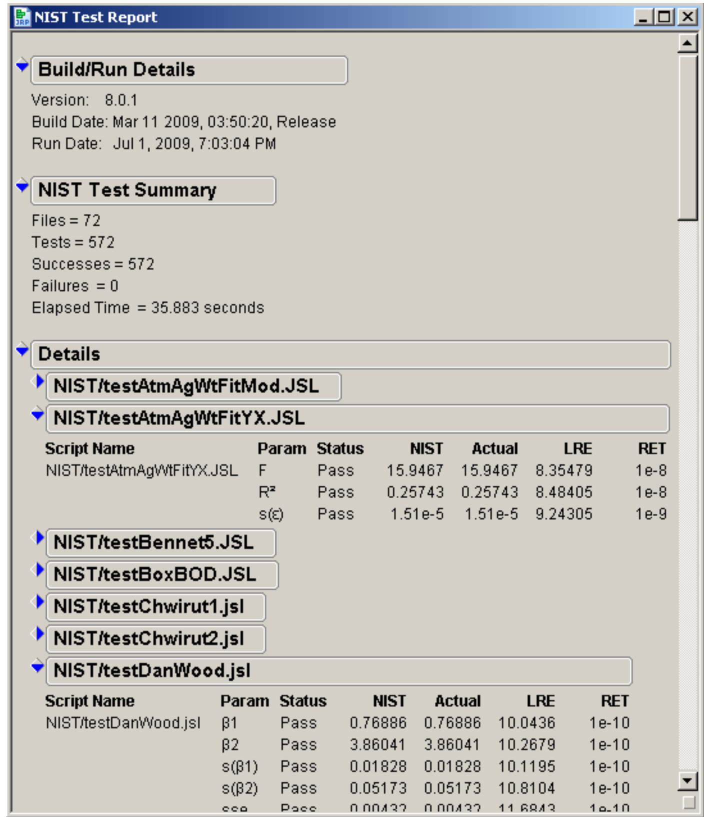

Figure 2: NIST Test Report

Appendix 2: NIST testing framework

The NIST framework consists of a set of JSL scripts and JMP data tables. It is intended to be host independent and should work for any version of JMP beginning with version 7. The architecture is as follows:

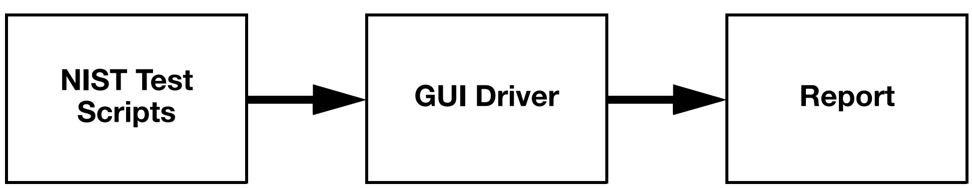

Figure 3: NIST testing framework

The GUI driver script (`RunNISTTests.jsl`) can reside anywhere in your file system. The NIST test scripts and associated data tables must reside in a subdirectory, named `tests/NIST/`, of the directory where the driver script is located. In addition, to be recognized by the driver, test scripts use a prefix of `test` (for example, `testMavro.jsl`).

NIST tests are specified as JSL scripts that access test data from JMP data tables. Individual test cases (for example, parameter estimates) are specified by way of a function named ut assert() that is defined by the GUI driver. The prototype of the function is

```
ut assert( expression, expected value )
```

where the `expression` argument specifies the actual result and the `expected value` argument the expected result. A test case is considered a success (or a pass) if the ratio of the difference between actual and expected, and expected, is less than an *epsilon* value defined in the script. Note that in the supplied test scripts, this *epsilon* value is stored in a global variable named `ut relative epsilon`. Test cases also pass if the expected and actual values are both missing. In addition to determining success or failure, the `ut assert()` function also computes the LRE.

Note that the framework is a JSL application and the scripts that constitute the framework are provided in unencrypted form. Users can therefore change the driver script, associated utility scripts, or the test scripts, if necessary. Users can also add additional test scripts to the framework. As long as a few simple conventions (described in the following section) are followed, the driver automatically detects these scripts and makes them available for execution.

Adding test scripts to the framework

1. Use the pattern below as a guide when writing your script.
2. Ensure that the script is stored in the `tests/NIST/`subdirectory.
3. Ensure that the script name has the `test` prefix (for example, `testMavro.jsl`).

NIST test script pattern

```
// Open datatable
dt=Open( < name > );

// Set relative epsilon
ut relative epsilon = < value >;

// Expected results
expected = < expected value >;

// Launch platform and define a reference to the report
obj = < platform launch expression >;
rpt = obj << report;

// Navigate display tree and get actual results
actual = rpt[ < subscript > ] << get as matrix;

// Invoke ut assert function to execute numerical accuracy tests
ut assert( expr(actual), expected );

close(dt, no save);
wait(0);   // give window a chance to close
```

References

Creighton, L. & Ding, J. (2000). “Assessing the numerical accuracy of JMP”. *SAS Whitepaper*.

McCullogh, B. D. (1998). “Assessing the reliability of statistical software: Part I”. *The American Statistician*, v52, n4 (November 1998):358-366.