ELSEVIER

Computational Statistics & Data Analysis 31 (1999) 27–37

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

On the accuracy of statistical procedures in Microsoft Excel 97

B.D. McCullough *, Berry Wilson

Federal Communications Commission, 445 12th St. SW, Room 2C-134, Washington, DC 20554, USA

Received 1 June 1998; received in revised form 1 December 1998

Abstract

The reliability of statistical procedures in Excel are assessed in three areas: estimation (both linear and nonlinear); random number generation; and statistical distributions (e.g., for calculating *p*-values). Excel's performance in all three areas is found to be inadequate. Persons desiring to conduct statistical analyses of data are advised not to use Excel.

Keywords: DIEHARD; ELV; Numerical accuracy; Software testing; StRD

1. Introduction

Sawitzki (1994b) documented the failure of many statistical packages to pass entry-level tests of accuracy known as the Wilkinson Tests (Wilkinson, 1985). The need exists to know how statistical packages fare on more substantial tests of numerical accuracy (Sawitzki, 1994a). Recently McCullough (1998) proposed a collection of intermediate-level tests which assesses the numerical reliability of a package in three areas: estimation (both linear and nonlinear); random number generation; and statistical distributions (e.g., for calculating *p*-values). Estimation is assessed via the *Statistical Reference Datasets* (StRD), which recently was released by the American "National Institute of Standards and Technology" (NIST). The output of the random number generator (RNG) is assessed using statistical tests for randomness. The accuracy of statistical distributions is assessed by comparing the results of Excel to those from a specialized package such as Knüsel's (1989) ELV program.

* Corresponding author
E-mail address: bmccullo@fcc.gov (B.D.McCullough)

0167-9473/99/$ – see front matter
PII: S 0167-9473(99)00004-3

This methodology has been applied to statistical software (McCullough, 1999a) and econometric software (McCullough, 1999b) to uncover numerous flaws in each area. It is worth noting that vendors of these statistical and econometric packages participated fully in the application of this methodology to their products. These vendors verified all calculations, provided information on algorithms when such information was not included in the documentation, and otherwise assisted the process.

This methodology is applied to Microsoft Excel 97, which offers a variety of statistical procedures. Numerous texts are devoted to using Excel for statistics, decision and management science, and financial modeling. Since it is conceivable that more statistical calculations are performed using Excel than any other package, it is important that the statistical capabilities of Excel be assessed. Microsoft was invited to participate in this evaluation, but chose not to do so.

It is important to note that the purpose of benchmarking is not to count the number of accurate digits. Obviously, what constitutes an "acceptable" number of accurate digits varies from user to user and application to application. This difference between users notwithstanding, all users can reasonably expect that developers have correctly implemented reliable algorithms. Yet, software developers frequently do not disclose the algorithm used in a specific procedure, and rarely reveal the details of its implementation. Thus, the purpose of benchmarking is to assess the quality of the underlying algorithm. If benchmark results show that the implemented algorithm is faulty in some way, or that a known "bad" algorithm has been implemented, then the software can be judged inadequate in that regard. For all the tests considered here, double precision can achieve acceptable accuracy. Consequently, if a package "fails" a test in double precision but "passes" it in quadruple precision, its performance must be judged inadequate because the purpose of benchmarking is to assess the algorithm, not extended precision capabilities.

Another important consideration is that the tests be "reasonable" (Wilkinson, 1994). For any algorithm, a data set can be reverse-engineered to exploit a weakness in the algorithm. Therefore, the test problem should be amenable to solution by known reliable algorithms. If a good algorithm can compute the percentiles of the Normal distribution down to 1E-12, then failing to compute smaller percentiles is not evidence of bad software, but failing to compute larger percentiles is such evidence. Similarly, suppose a reliable algorithm will solve a particular problem to several digits of accuracy, and the package in question produces only a few accurate digits or even zero accurate digits. It can then be deduced that the package does not properly implement a reliable algorithm, and so can be judged inadequate.

By way of illustrating these points, the statistical distributions of Excel already have been assessed by Knüsel (1998), to which we refer the interested reader. He found numerous defects in the various algorithms used to compute several distributions, including the Normal, Chi-square, F and t, and summarized his results concisely: "So one has to warn statisticians against using Excel functions for scientific purposes". There exist well-known algorithms which do not suffer from these defects, and which many packages implement. The performance of Excel in this area can be judged unsatisfactory. Thus, the remainder of this paper shall focus on estimation and RNG testing. Our computer is a Pentium running Windows 95.

Finally, there is the matter of how a developer responds to known errors, especially errors which have been published. Sawitzki (1994b) used Wilkinson's (1985) entry level tests of numerical accuracy to uncover flaws in statistical procedures of Excel 4.0, in particular in the computation of the sample variance and in diagnosing singularity for linear regression. The StRD will expose these same flaws if they have not been corrected in subsequent releases of Excel.

2. StRD

The StRD (http://www.nist.gov/itl/div898/strd) was designed explicitly to assist researchers in benchmarking statistical software packages, and comprises four suites of numerical benchmarks for statistical software: univariate summary statistics, one-way analysis of variance, linear regression, and nonlinear regression. For each suite of tests there are several problems in each of three difficulty levels: low, average, and high, indicated in the tables by parenthetical (l), (a), and (h). Reliable algorithms implemented in double precision produce acceptable results for all four suites.

Using multiple precision computer arithmetic to 500 digits for linear procedures, the StRD can all but eliminate rounding error, thus providing solutions which may be considered exact. For nonlinear least-squares problems, NIST uses different algorithms with different implementations and quadruple precision to solve the test problems. Both of these algorithms can, in double precision, return 10 accurate digits for each of the nonlinear test problems. In addition, multiple profiles of the least squares surface are used to ensure that a global minimum has been attained. "Certified values" for the solutions are provided to 15 digits for linear procedures and 11 digits for nonlinear procedures. Computational details of the StRD, including problem selection and methods of solution, can be found in Rogers et al. (1998).

2.1. Univariate summary statistics

Consider estimating the sample mean ($\bar{x}$) and sample standard deviation (s) for the StRD dataset Michelso.[1] The StRD certified values calculated to 15 significant digits are given in Table 1, together with the results calculated by Excel. Inaccurate digits are underlined.

It can be seen that Excel returns about 15 accurate digits for the mean and about seven accurate digits for the standard deviation. The same information can be more concisely presented by use of the *log relative error* (LRE),

$$\lambda = \log_{10}(|x - c|)/|c|, \tag{1}$$

where x is the estimated quantity and c is the certified value, and λ is given the appropriate subscript. The LRE is a measure of the number of significant digits only if x is close to c. Before calculating the LRE, it should first be determined that x and c differ by a factor of less than two, else set the LRE to zero. Due to the computer's

[1] All the dataset names are UNIX-style, and are case sensitive.

Table 1
StRD results for univariate data set michelso

	$\bar{x}$	s
NIST	299.852400000000	0.0790105478190518
Excel	299.85240000000	0.0790105482336451

Table 2
StRD results for univariate summary statistics

Dataset	$\lambda_{\bar{x}}$	λ_s	Dataset	$\lambda_{\bar{x}}$	λ_s
Pidigits (l)	15	15	Numacc1 (l)	15	15
Lottery (l)	15	15	Numacc2 (a)	14.0	11.6
Lew (l)	15	15	Numacc3 (a)	15	1.2
Mavro (l)	15	9.4	Numacc4 (h)	14.0	0
Michelso (l)	15	8.3			

finite precision, it is possible for the LRE to exceed the number of significant digits in c, in which case set the LRE equal to the number of significant digits in c. Any LRE less than unity is set to zero. Finally, LREs of zero and the number of siginificant digits in c are displayed without a decimal point, to remind the reader that these are upper and lower bounds. Thus, in the above example, $\lambda_{\bar{x}} = 15$ and $\lambda_s = 8.3$. Table 2 presents such results for all nine of the univariate data sets.

In double precision, the usual formula

$$\hat{\sigma}^2 = \frac{\sum(x_i - \bar{x})^2}{n-1} \tag{2}$$

will return several accurate digits for Numacc3 and Numacc4, which Excel does not do, even though it is a double-precision program. While the user guide does not indicate what formula is used to calculate the variance, it is not unreasonable to think that Excel uses the "calculator formula" presented in many textbooks

$$\hat{\sigma}^2 = \frac{\sum x_i^2 - n\bar{x}^2}{n-1}. \tag{3}$$

A method designed for hand calculation with a few observations, each of which is small in magnitude, is not appropriate for use in a computer program which may handle a great many observations whose magnitude may be large. This formula, the least reliable of the five methods analyzed by Ling (1974), is frequently used as an example of "what not to do" in texts on statistical computing (e.g., Thisted, 1988, Section 3.2.2), because the algorithm is inherently unstable. On the basis that Excel implements an unreliable algorithm for computation of the sample variance, its performance on this suite of tests can be judged inadequate. Sawitzki (1994b) noted that Excel 4.0 had the same difficulty calculating the sample variance, so Microsoft did not fix this error.

Table 3
StRD results for ANOVA dataset SiResist

			StRD	
Source	df	ss	ms	F
Between	4	5.114E-02	1.2787E-02	1.18046237440225E + 00
Within	20	2.166E-01	1.0832E-02	–
			Excel	
Source	df	ss	ms	F
Between	4	5.114E-02	1.2787E-02	1.18046237<u>81126100</u>E + 00
Within	20	2.166E-01	1.0832E-02	–

Table 4
ANOVA results

Test	λ_F	Test	λ_F
SiResist (l)	8.5	Simon5 (a)	1.1
Simon1 (l)	14.3	Simon6 (a)	0[a]
Simon2 (l)	12.5	Simon7 (h)	0[b]
Simon3 (l)	12.6	Simon8 (h)	0[a]
AgWt (a)	1.8	Simon9 (h)	0[a]
Simon4 (a)	1.7		

[a]Produced negative within group sum-of-squares.
[b]Produced negative between group sum-of-squares.

2.2. Analysis of variance

Table 3 gives the one-way analysis of variance table for the data set SiResist; only the F-statistic is displayed to all fifteen digits. A convenient way to summarize this information is to report the LRE for the F-statistic. Calculation shows that $\lambda_F = 8.5$. If the F-statistic is not accurate to at least a few digits, then some gross error has occurred in the calculations of the sums of squares or elsewhere. Results for this suite of tests are presented in Table 4.

As can be seen, Excel delivers an acceptable performance only for the low-difficulty problems. A reliable algorithm can deliver eight or nine digits for the average difficulty problems, with performance degrading to only a few digits for the higher difficulty problems. From a computational perspective, it is worth noting that such problems may be better solved using symbolic methods. For example, using Hunka's (1997) ANOVA.NB module, *Mathematica 3.0* (Wolfram, 1996) can deliver 15 digits of accuracy for all the ANOVA problems. As far as numerical solution of such problems is concerned, these higher-difficulty ANOVA tests are an example that delivering only a few digits of accuracy is not necessarily evidence of bad software. However, delivering zero digits of accuracy for the average-difficulty problems is such evidence. Observing the negative sums of squares produced by Excel, it can be deduced that Excel uses an unstable algorithm. Thus, Excel's performance on this suite of tests can be judged inadequate.

Table 5
StRD results for Longley Dataset

Coefficient		Mean	Stand.dev.
β_0	NIST	−3482258.63459582	890420.383607373
	Excel	−3482258.6538903	890420.385773117
	LRE	8.3	8.6
β_1	NIST	15.0618722713733	84.9149257747669
	Excel	15.0618726770786	84.9149257825076
	LRE	7.6	10.0
β_2	NIST	−0.0358191792925910	0.0334910077722432
	Excel	−0.0358191798902255	0.0334910078242200
	LRE	7.8	8.8
β_3	NIST	−2.02022980381683	0.488399681651699
	Excel	−2.02022981272773	0.488399682456131
	LRE	8.4	8.8
β_4	NIST	−1.03322686717359	0.214274163161675
	Excel	−1.03322686974925	0.214274163322592
	LRE	8.6	9.1
β_5	NIST	−0.0511041056535807	0.226073200069370
	Excel	−0.0511041036005626	0.226073200148693
	LRE	7.4	9.5
β_6	NIST	1829.15146461355	455.478499142212
	Excel	1829.15147447748	455.478500251236
	LRE	8.3	8.6

Table 6
StRD results for linear regression

Dataset	$\lambda_{\hat{\beta}}$	$\lambda_{\hat{\sigma}}$	Dataset	$\lambda_{\hat{\beta}}$	$\lambda_{\hat{\sigma}}$
Norris(l)	12.1	13.8	Wampler1 (h)	7.0	7.2
Pontius (l)	11.2	14.3	Wampler2 (h)	9.7	11.8
Origin1 (a)	14.7	15	Wampler3 (h)	6.6	11.2
Origin2 (a)	15	15	Wampler4 (h)	6.6	11.2
Fillip (h)	0	0	Wampler5 (h)	6.6	11.2
Longley (h)	7.4	8.6			

2.3. Linear regression

Table 5 presents the NIST and Excel results for the famous Longley (1967) benchmark, together with LREs. Again, inaccurate digits are underlined. Even using LREs, this is too much information, and only one LRE for the coefficients and one LRE for the standard errors can be presented. Based on the "weakest link in the chain" principle, the respective minima are used, so $\lambda_{\hat{\beta}} = 7.4$ and $\lambda_{\hat{\sigma}} = 8.6$. Similar results for all the linear regression problems are presented in Table 6.

While Excel performs reasonably well on most of the data sets, its failure on the data set Filip indicates a serious problem. Filip is a tenth-degree polynomial, which of course is highly collinear and can stress the linear solver. Ill-conditioned

Table 7
StRD results for nonlinear problem Rat43

		Default		Precision=1E − 7	
Coefficient	NIST	Excel	LRE	Excel	LRE
β_1	699.64151270	676.0986499	1.5	699.6463758	5.2
β_2	5.2771253025	39.7190456	0	5.275510708	3.5
β_3	0.75962938329	4.559009025	0	0.759471816	3.6
β_4	1.2792483859	13.02379155	0	1.27875659	3.4

data matrices are highly susceptible to numerical error, and so it is important for a solver to recognize that a data set is ill-conditioned. This is a most important part of any linear regression routine. In fact, Press et al. (1992, p. 23) notes that "much of the sophistication of complicated 'linear equation-solving packages' is devoted to the detection" of near-singularity. This can be verified by consulting the IMSL or LAPACK documentation. If the data are too ill-conditioned for the solver to produce a reliable solution, the program should refuse to compute a solution, issuing a warning message such as, "near-singular data matrix". Excel, however, ignores the near-singularity and proceeds with the calculations, delivering coefficients which are accurate to zero digits. On the basis that Excel does not properly check for near-singularity of the data matrix, its performance on this suite can be judged inadequate. Sawitzki (1994b) noted a similar problem in Excel 4.0, so this problem also was not fixed.

2.4. Nonlinear regression

The nonlinear benchmarks provide for two sets of starting values, Start I and Start II, the former "far" from the solution and the latter "near" to the solution. Usually it is easier for a solver to achieve a more accurate solution from Start II than from Start I. Start II is used only in the case that the solver refuses to produce a solution from Start I.

Frequently nonlinear solvers have several options. Excel offers options for: method of derivative calculation, forward (default) or central numerical derivatives; convergence tolerance (default is 1.E-3); "scaling" (recentering) the variables; and method of solution (default is the GRG2 quasi-Newton method, with an option for an unspecified conjugate gradient method). Default options rarely represent the best a solver can do, as Table 7 indicates. Default solution of the data set Rat43 yields $\lambda_{\hat{\beta}} = 0$, while decreasing the tolerance to 1.E-7 yields $\lambda_{\hat{\beta}} = 3.4$. Changing the method of solution or method of derivative calculation did not improve the results.

Table 8 presents similar results for all 27 nonlinear data sets with four sets of options. Default solution is presented in column two, where it can be seen that zero-digit accuracy is produced 21 times. In each case, since a correct solution was not produced, the solver should have returned a "cannot find a solution" message. The third column presents results for decreasing the tolerance to 1E-7, which changed a zero-digit answer to a correct answer four times: Misra1a, Misra1c, Thurber and

Table 8
StRD results for nonlinear regression

	Nonlinear options			
Data set	A	B	C	D
Misra1a (l)	0	1.6	0	4.8
Chwirut2 (l)	4.3	4.3	4.6	4.6
Chwirut1 (l)	4.0	4.0	4.9	4.9
Lanczos3 (l)	0	0	0	0
Gauss1 (l)	0	0	0	0
Gauss2 (l)	0	0	0	0
DanWood (l)	4.7	4.7	5.5	5.5
Misra1b (l)	1.2	1.2	0	4.4
Kirby2 (a)	0	0	0	1.1
Hahn1 (a)	0	0	0	0
Nelson (a)	0	0	0	1.3
MGH17 (a)	0	0	0	0
Lanczos1 (a)	0	0	0	0
LANCZOS2 (a)	0	0	0	0
Gauss3 (a)	0	0	0	0
Misra1c (a)	0	2.1	4.6	4.6
Misra1d (a)	0	0	0	5.3
Roszman1 (a)	0	0	2.3	3.7
ENSO (a)	3.4	3.4	3.3	3.4
MGH09 (h)	0	0	0	0
Thurber (h)	0	1.7	0	1.8
BOXBOD (h)	0	0	0	0
Rat42 (h)	3.7	5.9	5.3	5.3
MGH10 (h)	0	0	0	0
Eckerle4 (h)	0	0	0	0
Rat43 (h)	0	3.4	0	0
Bennett5 (h)	0	0	0	0

A: Default estimation.
B: Convergence tolerance=1E-7.
C: Automatic scaling.
D: B+C.

Rat43. Invoking "automatic scaling" but leaving the convergence tolerance at default offers slight improvement over default estimation, as can be seen in column four: 20 problems had zero-digit accuracy. Finally, the fifth column presents the results for automatic scaling and tolerance set at 1E-7: fourteen zero-digit accuracy solutions. For any package it may be too much to expect that the solver can find a solution for each problem, but it is not too much to expect that the solver can figure out when it has not reached a solution. The zero digits of accuracy "solutions" in Table 8 do *not* represent local minima. Rather, the solver has labelled as a local minimum a point which is not a local minimum, and this it should not do. In such cases, reliable packages will report some sort of warning message rather than print out coefficients. By way of contrast, for the 27 nonlinear problems, the number of

Table 9
Results of Marsaglia's DIEHARD tests

Test		Test	
Birthday spacings test	p	Count the ones test (stream of bytes)	F
Overlapping 5-permutation test	p	Count the ones test (specific byte)	F
Binary rank test: 31×31 matrices	p	Parking lot test	p
Binary tank test: 32×32 matrices	p	Minimum distance test	p
Binary rank test: 6×8 matrices	p	3-D spheres test	F
Bitstream test	p	Squeeze test	F
OPSO test	F	Overlapping sums test	p
OQSO test	F	Runs test	p
DNA test	F	Craps test	p

"p"= Pass.
"F"=Fail.

zero-digit accuracy solutions for SAS, SPSS, and S-Plus are three, one, and none, respectively. The econometric package TSP also produces no zero-digit accuracy solutions.

3. Random number generator

Elementary details of random number generation are discussed in Gentle (1998). The RNG should produce output which passes tests for randomness. Testing the RNG is important, because many statistical procedures make use of random numbers. More than one test should be applied, since there are many possible departures from randomness. One such collection of tests is given by Knuth (1981), which was programmed by Dwyer and Williams (1996a, b). Marsaglia (1993) noted that these tests are not very stringent, and later proposed DIEHARD: *A battery of randomness tests* (Marsaglia, 1996). These tests are discussed in the *DIEHARD* documentation and in Gentle (1998, ch. 6).

Both collections of tests require millions of random numbers as input, and Excel will not generate such a large file. Therefore, another double precision package was used to generate numbers according to the algorithm for the Excel RNG, and these were submitted to both programs. The Excel RNG passed all the Knuth tests (results not presented), but failed several of the *DIEHARD* tests. Results are presented in Table 9.*DIEHARD* is predicated on a full 32-bit RNG, though many PC software programs use only 31-bit RNGs. A good 31-bit RNG will pass all but one of the DIEHARD tests. Since Excel does not use a good RNG, its performance on this suite of tests can be judged inadequate.

4. Conclusions and recommendations

We have applied the methodology outlined by McCullough (1998) to assess the reliability of Excel in three areas: estimation, random number generation, and sta-

tistical distributions. Excel has been found inadequate in all three areas. We also note that Microsoft did not correct flaws noted by Sawitzki (1994b). We advise that Excel not be used for statistical calculations. There is a large number of "add-on" statistical packages for Excel; these also need to be benchmarked. Persons wishing to conduct statistical analyses would do well to avail themselves of a package which performs well on benchmark tests.

Acknowledgements

The views expressed herein are those of the authors and do not necessarily reflect those of the Commission. In addition to the referees, we thank K.B. Williams for running the Knuth tests, and an associate editor for making many useful suggestions.

References

Dwyer, J., Williams, K.B., 1996a. Testing random number generators. C/C++ Users J. 39–48.

Dwyer, J., Williams, K.B., 1996b. Testing random number generators, Part 2. C/C++ Users J. 57–66.

Gentle, J., 1998. Random Number Generation and Monte Carlo Methods. Springer, New York.

Hunka, S., 1997. ANOVA.NB, www.mathsource.com.

Knüsel, L. 1989. Computergestützte Berechnung Statistischer Verteilungen. Oldenburg, München-Wien [English versions of the program and documentation at www.stat.uni-muenchen.de/~knuesel/elv].

Knüsel, L., 1998. On the accuracy of statistical distributions in Microsoft Excel 97. Comput. Statist. Data Anal. 26, 375–377.

Knuth, D., 1981. The Art of Computer Programming, vol. 2, Seminumerical Algorithms, 2nd ed., Addison-Wesley, Reading, MA.

Ling, R.F., 1974. Comparison of several algorithms for computing sample means and variances. J. Amer. Statist. Assoc. 69, 859–866.

Longley, J.W., 1967. An appraisal of computer programs for the electronic computer from the point of view of the user. J. Amer. Statist. Assoc. 62, 819–841.

Marsaglia, G., 1993. Monkey tests for random number generators. Comput. Math. Appl. 26, 1–10.

Marsaglia, G., 1996. DIEHARD: A battery of tests of randomness. http://stat.fsu.edu/pub/diehard.

McCullough, B.D., 1998. Asssessing the reliability of statistical software: Part I. The Amer. Statist. 52, 358–366.

McCullough, B.D., 1999a. Asssessing the reliability of statistical software: part II. The Amer. Statist., forthcoming.

McCullough, B.D., 1999b. The reliability of econometric software: Eviews, LIMDEP, SHAZAM and TSP. J. Appl. Econom., forthcoming.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.R., 1992. Numerical Recipes in Fortran, 2nd ed. Cambridge University Press, New York.

Rogers, J., Filliben, J., Gill, L., Guthrie, W., Lagergren, E., Vangel, M., 1998. StRD: statistical reference datasets for assessing the numerical accuracy of statistical software. NIST TN# 1396, National Institute of Standards and Technology, USA.

Sawitzki, G., 1994a. Testing numerical reliability of data analysis systems. Comput. Statist. Data Anal. 18, 269–286.

Sawitzki, G., 1994b. Report on the reliability of data analysis systems. Comput. Statist. Data Anal. (SSN) 18, 289–301.

Thisted, R.A. 1988. Elements of Statistical Computing. Chapman & Hall, London.
Wilkinson, L., 1985. Statistics Quiz. IL. SYSTAT, Evanston.
Wilkinson, L., 1994. Practical guidelines for testing statistical software. In: Dirschedl, P., Ostermann, R. (Eds.), Computational Statistics. Physica-Verlag, Berlin, pp. 111–124.
Wolfram, S. 1996, Mathematica 3.0 User's Guide. Cambridge University Press, New York.