Choosing Reliable Statistical Software

Micah Altman, *Harvard University*
Michael P. McDonald, *University of Illinois, Springfield*

There's a credibility gap: We don't know how much of the computer's answers to believe. Novice computer users solve this problem by implicitly trusting in the computer as an infallible authority; they tend to believe that all digits of a printed answer are significant. Disillusioned computer users have just the opposite approach; they are constantly afraid that their answers are almost meaningless.

—Donald Knuth 1997, 229.

Should we trust the results of our statistical computations? Closely following the development of the mainframe computer, Longley (1967) criticized the accuracy of the first regression programs. Approximately every 10 years thereafter, similar comments echoed for each new generation of statistical software. In a recent criticism, McCullough and Vinod (1999) argue that commonly used statistical packages may give "horrendously inaccurate" results, which have gone largely unnoticed (635–37). Moreover, they argue that in consequence of these inaccuracies, past inferences are in question, and future work must document and archive statistical software alongside statistical models (660–62). When political scientists discuss accuracy in computer-intensive quantitative analysis, however, we are relatively sanguine. Numerical accuracy is almost never discussed in articles or even in textbooks geared toward the most sophisticated and computationally intensive techniques (e.g., King 1989; Mooney 1997). Notable exceptions are a forthcoming *APSR* controversy that depends on the meaning and evaluation of numerical accuracy in ecological inference (King 2001; Tam Cho and Gaines 2001) and a study of numerical accuracy issues in replication (Altman and McDonald 2001).

In our ongoing research project, we apply techniques from the field of statistical computation to improve the accuracy of inferences from complex statistical procedures, such as Gary King's (1997) solution to the ecological inference problem. We find significant inaccuracies in the estimations of these complex models. Our working essay and an extensive list of resources are available at our website <http://data.fas.harvard.edu/micah_altman/numal/>.

In this essay, we evaluate the accuracy of the common, relatively simple, statistical estimations that political scientists make every day, and the statistical packages we use to make them such as Gauss, Stata, SST, and Excel. Here we summarize the results of our investigation, along with others' investigations into SAS, SPSS, S-Plus, E-Views, SHAZAM, LIMDEP, and TSP (McCullough 1998, 1999a, 1999b; McCullough and Vinod 1999). We find that inaccuracy may be a problem in even simple estimations, but one may avoid inaccuracy by choosing the right statistical package for the task at hand. Our hope is that this article leads political scientists to be more informed consumers of statistical software.

Micah Altman *is director of the Virtual Data Center project, which aims to promote the sharing of research data by building open-source software tools. He is also associate director of the Harvard-MIT Data Center, and a research associate in the Center for Basic Research in the Social Sciences at Harvard University. He has published (or in press) over ten articles, datasets, and reviews, on the topic of redistricting, numerical accuracy, and digital libraries.*

Michael P. McDonald *is assistant professor of political studies at University of Illinois, Springfield. His methodological work is on the mechanics of estimation. His substantive work is in American Politics, covering voting behavior, Congress, and the intersection of the two, in both current and historical perspectives.*

Issues in Implementation

The world of bits and bytes is different than the world of pencil and paper. Surprisingly, computers cannot always exactly represent even a simple written number. The translation of decimal numbers into the binary language of computers often introduces small errors. Calculations that occur behind the scenes of routine statistical procedures may run afoul of these translation errors, too, compounding small errors and producing results that are completely wrong.

Floating-Point Arithmetic

To understand the limits of computers, we need to review how computers do math. Nearly all statistical software programs use floating-point arithmetic that represents numbers as fixed-length sequences of ones and zeros, or bits (b), with a single bit indicating the sign. Binary arithmetic on integers operates according to the "normal" rules of arithmetic, as long as the integers are not too large ($>2^{b-1}-1$), which can cause an undetected *overflow* error. For example, usually $b=32$, so the number 2,147,483,648 ($1+2^{32-1}-1$) would overflow and be treated as –1.

Decimal, or floating-point numbers, are represented by two bit sequences, with one sequence representing a mantissa (m), and another representing an exponent (e): $\pm m*10^{e}$. (An additional bit indicates the sign.) This representation is also susceptible to overflow, and to *underflow*—when a number is smaller than the smallest value capable of being represented by the computer (known as precision). An example is .1, which surprisingly has no exact binary representation using this technique. The infinitely repeating floating-point transformation of .1 must be represented somehow, leading to *rounding* or *truncation* errors. Rounding, truncation, overflow, and underflow cause the associative laws of arithmetic to fail, producing sometimes surprising numerically inaccurate results (Higham 1996).[1]

Higham (1996) observes that researchers are often overconfident of the accuracy of their calculations because they are unaware of the consequence of floating-point inaccuracy, such as:

- Even a simple computation may be subject to numerical inaccuracy.
- Even one error propagated through a calculation may cause significantly inaccurate results.
- Increasing the precision of a calculation does not monotonically increase the accuracy of the final result.

The moral is that a careful analysis of the accuracy of each numerical *algorithm*, in its entirety, is necessary for one to have the correct degree of confidence in the accuracy of one's results.

Inaccuracy by Accident and Design

All software contains bugs and numerical statistical software is no exception. A few examples, culled from bug re-

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

TABLE 1
Reported Standard Deviations for Columns of Data in Excel.

Number of Significant Digits	2	7	8	9	10	15
	1	1000001	10000001	100000001	1000000001	100000000000001
	2	1000002	10000002	100000002	1000000002	100000000000002
	1	1000001	10000001	100000001	1000000001	100000000000001
Values	2	1000002	10000002	100000002	1000000002	100000000000002
	1	1000001	10000001	100000001	1000000001	100000000000001
	2	1000002	10000002	100000002	1000000002	100000000000002
	1	1000001	10000001	100000001	1000000001	100000000000001
	2	1000002	10000002	100000002	1000000002	100000000000002
	1	1000001	10000001	100000001	1000000001	100000000000001
	2	1000002	10000002	100000002	1000000002	100000000000002
Reported Result	0.50	0.50	0.51	0.00	12.80	1186328.32
Correct Result	0.50	0.50	0.50	0.50	0.50	0.50

ports, follow: [2] Gauss (ML 4.24) reported *t*-statistics for maximum likelihood estimations that were half the correct values. SAS (7.0) produced incorrect results for regressions using variables with long names, and performed exponentiation incorrectly (among other errors). SPSS (8.01) incorrectly calculated *t*-tests, and incorrectly dropped cases from *crosstabs*. These bugs were identified and fixed long ago, but that doesn't mean that current software is bug-free. Users should periodically update their software with any patches provided by the manufacturer.

Not all inaccuracies are by accident. A Microsoft technical note states that some functions in Excel (v. 5.0-98) are inaccurate by design. The standard deviation, kurtosis, binomial distributions, linear and logistic regressions functions produce incorrect results when intermediate calculations—calculations that are hidden from the user to construct a final calculation—yield large values. The calculation of the standard deviation by Microsoft Excel is an excellent example of a software design choice that illustrates how unreliable accuracy may result from differences in implementation.

In typical statistics texts, the standard deviation of a population is defined as:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{n}}$$

The Textbook Definition for the Standard Deviation

The textbook formula is not the fastest way to calculate the standard deviation since it requires one pass through data to compute the mean and a second pass to compute the difference terms. If a data set is large, a faster and mathematically equivalent formula that computes the standard error in one pass is given by:

$$\sqrt{\frac{n\sum x^2-(\sum x)^2}{n^2}}$$

The One-Pass Formula for the Standard Deviation

Microsoft Excel uses the one-pass formula. Unfortunately, the one-pass formula is prone to severe rounding errors when the $n\Sigma x^2-(\Sigma x)^2$term requires subtracting two large numbers. As a consequence, Excel reports the standard deviation incorrectly when the number of significant digits in a column of numbers is large. Table 1 illustrates this. Each column of ten numbers in Table 1 has a standard deviation of precisely one half, yet the number reported by Excel ranges from zero to over a million.

Note that the inaccuracy of the reported standard deviation is a direct result of the algorithm choice, and not a limitation in the precision of the underlying floating-point arithmetic Excel uses.[3] Excel's fundamental numerical operations are as accurate as those of most other packages that perform the standard deviation calculation correctly.[4] In fact, by implementing the textbook equation within Excel, using the "*average*" function, we were able to obtain the correct standard deviations for all the cases shown in Table 1. Excel's designers would perhaps argue that they made the correct choice in choosing a more time-efficient calculation over one that is more accurate in some circumstances. Given the types of uses to which Excel is normally put, the choice to use a faster algorithm might have some justification. However, there are other one-pass algorithms for the standard deviation that are nearly as fast and much more accurate than the one that Excel uses (see Higham 1996).

More importantly, software packages should not report incorrect results *without warning the user*. In this case, Excel reports severely inaccurate results for a small column of numbers, all of which are similarly scaled, and each of which is well within the documented precision and magnitude used by the statistical package. Since numeric inaccuracies can occur in intermediate calculations that the program obscures from the user and since such inaccuracies may be undocumented, users may have no way of knowing when results are inaccurate. We greatly prefer software that warns of internal errors, even if the package implements algorithms that produce otherwise inaccurate results.

The Accuracy of Statistical Packages

Our examination of Excel's standard deviation shows that there are multiple ways of tackling the same programming problem, and that programmers may not choose the best solution. Indeed, there

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

may be no universally agreed upon "best" solution, as different solutions may be a compromise between speed and accuracy. The situation is compounded for extremely complex algorithms, which usually comprise several simpler algorithms strung together. Since software companies make money selling their proprietary software, they often shroud their programming in secrecy. Users are forced to treat a program as a black box, hoping that the programmers chose accurate algorithms and did not inadvertently make mistakes.

One way to determine if statistical software is inaccurate is to compare its output with benchmarks, test problems with known answers. In a recent series of articles, Bruce McCullough (1998, 1999a, 1999b) implements benchmarks to determine the reliability of econometrics software. We summarize what we know of the accuracy of these packages and extend McCullough's analysis of statistical software programs to those we identify as commonly used by the political science community, but which McCullough missed in his surveys.

Statistical Software Used by Political Scientists

In order to define the scope of our analysis, we first ask which statistical software packages are most commonly used by political scientists. Nathaniel Beck (1995), in *The Political Methodologist*, reports answers he and 10 colleagues gave to this question. Twenty-three stand-alone statistical software programs were mentioned, with nine mentioned more than twice. For each program mentioned by more than two respondents, we searched for political research articles that reported which program calculated statistical analysis. In addition to these, we looked for use of Microsoft Excel, because its increasing set of features and enormous market share indicate its popularity. Our search covered approximately the last 10 years of the *American Political Science Review, American Journal of Political Science, Journal of Politics,* and *Political Science Quarterly*.[5]

Our survey, reported in Table 2, shows a wide distribution of statistical packages employed in published works. SAS, SPSS, and SHAZAM dominate, but there is still use of all other programs. As of the time of the writing of this essay, the packages that have not been comprehensively tested are: Excel (Office 97), Gauss (v3.2.43), SST (v2.0), and Stata (v6.0).[6] We fill this gap and report a summary for all statistical software packages in order to provide political scientists with a complete guide to the reliability of their statistical software.

TABLE 2
Survey of Use of Software Packages in Political Science, 1990–2000.

Statistical Package	Frequency in Beck (1995)	Frequency in Journals
Excel	0	2
Gauss	11	12
Limdep	5	10
RATS	4	10
SAS	11	42
SHAZAM	3	24
S-Plus	9	2
SPSS	5	30
SST	6	9
Stata	7	15

Benchmarking Statistical Software

Benchmarks solve a problem using a statistical software package, and then compare the generated answer to the known truth. Bruce McCullough has elsewhere benchmarked a number of statistical software packages used in econometrics, drawing upon both a series of benchmarks proposed by the National Institute for Standards and Technology (NIST) for univariate summary statistics, ANOVA, regression, and nonlinear regression (Rogers et al. 2000)[7]; and a series of tests of statistical distributions proposed by Knüsel (1995, 1998).[8] In order to leverage previous work, we verify these same benchmarks on the untested programs we identify above.

For each category of the NIST benchmarks, 12 problems probe various aspects of implementation. The benchmarks are not exhaustive, as many commonly used functions lack a benchmark, most notably maximum likelihood estimation (although we note that nonlinear regression can use similar maximization search algorithms). Each problem consists of data taken from published research or generated using a known function, a model to be estimated, and a set of "certified values" for the parameters. Certified values are either calculated analytically (for those generated from a known function) or are solved using 500 digits of precision on a supercomputer. The benchmark problems are assigned three levels of difficulty, generally corresponding to the level of difficulty of the problem.

To benchmark statistical distributions, we, and McCullough, use Knüsel's ELV program (1989) and Brown et al.'s (1998) high-precision C libraries to calculate true values generated at regular and random intervals within the legal ranges for the parameters of a particular distribution and its inverse. We then compare these true values to values generated by each statistical package, and record the minimum number of significant digits of agreement in the central, tails, and extreme tails of the Normal, T, F, Chi, and Gamma distributions. In addition to McCullough's benchmarking, Knüsel (1995, 1998) has tested Gauss and Excel, leaving benchmarks for SST and Stata for us to perform.

We present a summary of our benchmark analysis in Table 3. We have archived the detailed results of our benchmark analysis, which are quite extensive, as an ICPSR Publication-Related Archive #1243. We summarize the performance of the packages on a "Reliable/Some Errors" scale. A rating of "Reliable" indicates that the package passes *all* of the benchmarks described above, or provided clear error messages whenever inaccurate results were presented in the benchmarks. A Reliable package will fail to produce an answer or give a warning if it encounters problems, and these instances are noted with a "No Solution." Although No Solution indicates a limitation of the implementation, an implementation that notifies the user of its shortcomings is greatly preferable to one that does not, since the user receives warning and may correct potential problems. A rating of "Some Errors" denotes that the implementation provided incorrect estimates (estimates that are not even correct to one significant digit, for at least one parameter within at least one benchmark test in a particular category) without any accompanying warnings.[9]

We complete the survey of the reliability of statistical software packages used by political scientists in Table 3, with a summary of other benchmark studies, using our rating scale (Knüsel 1995, 1998; McCullough 1998, 1999a, 1999b; McCullough and Vinod 1999). Our notes as to why a package earns a Some Errors rating are not as complete as our own analysis simply because we must rely on other authors' published material, which does not go into lengthy detail.

Before we summarize the benchmark results, we note that software companies continually update their programs and that any published investigation of their operations will necessarily be dated. Some software companies are aware of the benchmarks and have updated their software to pass the benchmark tests. New versions of software may take a step forward by implementing new algorithms with improved accuracy, although a new implementation may also take two

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

TABLE 3
Our Benchmark Analyses of Excel, Gauss, Stata, and SST.

	Univariate Statistics	ANOVA	Linear Regression	Nonlinear Regression	Distributions
Excel (Office 97)	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels. Uses flawed standard deviation algorithm.	**Reliable** Low difficulty level. **Some Errors** Average difficulty level. Returned correct first digit only. High difficulty levels. Incorrect solutions to all problems.	**Reliable** Low and high difficulty levels. **No Solution** One 10^{th} order polynomial. **Some Errors** Average difficulty level. Incorrect R^2 for equations without a constant.	**Some Errors** All difficulty levels. Returned a credible solution 16 out of 27 times. Failed only one low difficulty level.	**Reliable** Central distributions. **Some Innacuracies** Tails of distributions. **Some Errors** Extreme tails of normal distribution.
Gauss	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels. Uses flawed standard deviation algorithm.	**Not Supported**	**Reliable** Low difficulty levels. **Some Errors** Average difficulty level. Incorrect R^2 for equations without a constant. High difficulty level. Low number of correct digits and returned a complex number as a standard error.	**Reliable** All difficulty levels. Returned a credible solution 26 out of 27 times. Accuracy, even in low difficulty problems was inconsistent, varying from 15 digits to less than 2.	**Reliable** Accurate in all u tested. **Not Supported** Most inverse distributions not supported.
Stata (6.0)	**Reliable**	**Reliable** Low difficulty level. **Some Errors** Average difficulty level. Low number of digits for 2 problems. **No solution** All other average and high difficulty levels.	**Reliable** All difficulty levels. **No Solution** One 10^{th} polynomial.	**Reliable** All difficulty levels. Returned a credible solution 26 out of 27 times. Consistently reported low number (4) of correct digits for standard errors.	**Reliable** Central distributions. **Some Inaccuracies** Tails of distributions. **Some Errors** Extreme tail of F distribution.
SST (2.0) (Limited Precision of Output)	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels. Uses flawed standard deviation algorithm.	**Reliable** Low difficulty level. **Some Errors** Average difficulty level. Returned only two correct digits on two of four problems. **No Solution** High difficulty levels.	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels. Incorrect solution for 10^{th} order polynomial. Low number of correct digits for one other problem.	**Some Errors** All difficulty levels. Returned a credible solution 17 out of 27 times. Even for credible solutions, rarely reproduced more than 2 correct digits.	**Reliable** **Some Inaccuracies** Central distributions. **Some Errors** Tails of F distributions.

steps backwards by introducing new bugs. Past performance is not necessarily an indicator of present performance, though researchers who used previous versions of software with demonstrable inaccuracy certainly should be critical of their results.[10]

The good news is that statistical software typically provides the correct answers, to at least the fourth significant digit, for univariate statistics, regression problems, low-difficulty analysis of variance problems, and low-difficulty nonlinear regression problems. There are exceptions: SHAZAM reports a negative standard deviation for one problem, a sure sign of a flawed algorithm; Excel, Gauss, and SST appear to use the one-pass standard deviation algorithm from our example; Excel and Gauss experience trouble computing the correct R^2 for regressions lacking a constant; and many programs encounter difficulty solving a tenth-order polynomial regression, a model we suspect is infrequently used in political science. SST stood out as it returns only two correct digits on many regression and nonlinear regression coefficients.

For researchers who perform more complex ANOVA and nonlinear regression, the benchmark results are sobering. Many programs are unable to return accurate ANOVA results for problems of even average difficulty. In addition, statistical packages exhibit considerable variation in the accuracy of nonlinear regression. E-Views, Excel, SHAZAM, and SST solve only half of the problems and report (without warning) many inaccurate answers. Gauss, LIMDEP, S-Plus, Stata, and TSP perform well, although the accuracy of the solutions is sometimes inconsistent. Moreover, obtaining accurate results often requires extensive experimentation with optimization options. One should heed McCullough's advice on this point, "Default nonlinear

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

TABLE 4
Summary of McCullough (1998, 1999a, 1999b), McCullough and Vinod (1999), and Kneüsel (1995, 1998) Benchmarks

	Univariate Statistics	ANOVA	Linear Regression	Nonlinear Regression	Distributions
SAS (v6.12)	**Reliable** All difficulty levels.	**Reliable** Low difficulty levels. **No Solution** average and high difficulty levels.	**Reliable** All difficulty levels. **No Solution** One 10th order polynomial.	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels. Produced 0 significant digits for three problems.	**Reliable** **Some Inaccuracies** Inaccurate with unusual degrees of freedom for some distributions.
SPSS (v7.5)	**Reliable** All difficulty levels.	**Reliable** Low difficulty levels. **No Solution** average and high difficulty levels.	**Reliable** All difficulty levels. **No Solution** One 10th order polynomial.	**Reliable** All difficulty levels. Produced 0 significant digits for one average difficulty level problem.	**Reliable** Accurate in all distributions tested. **Not Supported** Most inverse distributions not supported.
S-Plus (v4.0)	**Reliable** All difficulty levels.	**Reliable** Low and average difficulty levels. **No Solution** High difficulty levels.	**Reliable** All difficulty levels.	**Reliable** All difficulty levels.	**Reliable** Accurate in all distributions tested
Eviews (v3.0)	**Reliable** All difficulty levels.	Not Tested	**Reliable** All difficulty levels. **No Solution** One 10th order polynomial.	**Some Errors** All difficulty levels. Returned a credible solution 11 out of 24 times.	Not Tested
LIMDEP (v7.0 for Win95)	**Reliable** All difficulty levels.	Not Tested	**Reliable** All difficulty levels.	**Reliable** All difficulty levels. Returned a credible solution 24 out of 25 times.	**Reliable**
SHAZAM (v8.0)	**Reliable** Low and average difficulty levels. **Some Errors** High difficulty levels.	Not Tested	**Reliable** All difficulty levels.	**Some Errors** All difficulty levels. Returned a credible solution 15 out of 23 times.	**Reliable** Some Inaccuracies Central distributions. **Some Errors** Tails of F and Chi-square distributions.
TSP (v4.4)	**Reliable** All difficulty levels.	Not Tested	**Reliable** All difficulty levels. **No Solution** One 10th order polynomial.	**Reliable** All difficulty levels. Returned a credible solution 26 out of 26 times.	Not Tested

options should not be relied upon" (McCullough 1999a, 196).

Choosing a Statistical Package

Political scientists often overlook numerical inaccuracy and, as we have shown, this may have serious consequences for inference. Fortunately, there are steps for researchers to follow to avoid problems with numerical accuracy. *In much research, it may only be necessary to choose a statistically accurate software package, and then to apply common sense and the standard diagnostics. In other research, especially that involving complex maximum likelihood estimation or simulation, it may be necessary to repeat certain portions with different techniques, and to track down all of the implications of particular methods.* We start with the steps that we believe to be most generally applicable, and finish with steps geared toward numerically intensive models.

Trust, but verify

Most packages correctly solve routine statistical analyses, such as univariate statistics, linear regression, and many simple nonlinear regression models. Moreover, when we, as a discipline, scrutinize numerical accuracy, statistical software companies (and individual authors) pay attention. In response to an earlier conference-paper version of this article, Stata Corporation fixed the problems that we identified through our tests, and now publishes the results of the NIST tests for new versions of Stata. The R statistical language has also adopted the NIST test suite.[11]

Choose regularly updated software that uses well-documented algorithms and buttresses itself with standard tests of accuracy and reliability.[12] For example, statistical packages such as R and

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

Stata, adhere to high standards by providing very thorough documentation, regular benchmarks, and regular bug-fixes, and have demonstrated a commitment to accuracy. Other packages such as S-Plus, TSP, and Limdep have not yet made accuracy benchmarks part of their public documentation, but also fare very well in terms of accuracy. Avoid software packages that have not been tested adequately for accuracy, or that do not perform well.

Check for Substantive and Statistical Plausibility

Look at your results to make sure that they make sense and fit the data. Also, look for evidence of misspecification, outliers, or other signs of measurement or coding errors. These steps have two benefits. First, by checking for substantive and statistical plausibility, one is less likely to be misled if numerical accuracy affects the model. Second, if estimation is susceptible to numeric inaccuracy, then cleaning data and considering respecification of the model may eliminate the numeric inaccuracy, and also improve the overall analysis.

There has been a recent call for reevaluation of the practice of placing a "star" next to the significant value of reported results that pass a significance test (Gill 1999). Although this may serve as a quick summary method, the numeric inaccuracies inherent in the underlying calculations suggest that estimation beyond the third significant digit is essentially white noise. Results that are close to the critical value may, in fact, pass or fail simply because of numeric inaccuracies.

Test Complex or Problematic Models for Accuracy

In problematic cases, in which a model gives implausible results or in which complex models or simulations are used, especially for the first time, researchers should explicitly test for numerical accuracy. Ideally, when implementing a new model, one should create a comprehensive set of tests whose answers are known through such methods as Monte Carlo simulation, by solving the problem in closed-form, or by using multiple-precision arithmetic in programs such as Maple or Mathematica.

The vast majority of complex estimations in political science have not been examined for numerical accuracy. In our ongoing research, we find that correct ecological inference and other complex maximum likelihood problems require attention to numerical research. Political scientists who are building complex nonlinear models may wish to consult texts in statistical computation (e.g. Gill, Murray, and Wright 1981), as well as our working paper and online bibliography of resources, <http://data.fas.harvard.edu/micah_altman/numal/>.

Notes

1. In addition, these inaccuracies cannot be simply subsumed into the error term of most statistical estimates, because, among other reasons, they are not distributed at random.

2. These errors are reported in (respectively): The release notes (README.ml) for Gauss Maximum Likelihood module version 4.34, Stata Technical Bulletin #48, available from <www.stata.com/support/updates/ado/whatsnew.html>; Microsoft knowledge base article, "XL: Problems With Statistical Functions and Large Numbers," available from <http://support.microsoft.com>; and SAS technical notes, V7-SYS.COMP-0043 and V7-LAB-0201, available from the technical support section of <www.sas.com>; and SPSS release notes for version 8.02, available from <www.spss.com/tect/patches/SP-061199>.

3. Table 1 is an extension of the NIST univariate statistics benchmarks, particularly the numerical accuracy tests that Simon and Lesage (1988) derived. In this case, the textbook formula is more accurate than the algorithm used by Excel, a fact known since 1983. However, we are not claiming that the textbook formula is the most robust method of calculating the standard deviation; numerical stability could be improved in this formula in a number of ways, such as by sorting the differences before summation. Nor are we claiming that textbook formulas are always numerically robust. Quite the opposite is true (see Higham 1996, 94).

4. Although no other package that we are aware of documents this algorithm as a design feature, some apparently use it behind the scenes. We entered this table into our favorite statistics programs: STATA 6.0, SST 2.0, SPSS 9.0, SAS 7, Gauss 3.2.43 (descriptive statistics module), and JMP 3.2. Most of these packages calculated the table above (or its equivalent, using the sample standard deviation) correctly. SPSS, however, was very slightly inaccurate in one column, and Gauss and JMP produced errors similar in magnitude to Excel!

5. We performed an electronic full-text search to find all articles that mentioned any of these packages in the following journal volumes: *American Journal of Political Science* (33–41), *American Political Science Review* (83–89), *Journal of Politics* (51–57), and *Political Science Quarterly* (103–108). We screened each article individually, to reject false hits and references to software packages that were not actually used in support of the original research of that article. A small number of authors reported using two or more packages in one research article, which we counted as a "vote" for each package mentioned.

6. We omit RATS because it is a highly specialized time-series package, and is outside of the scope of most of the tests discussed in this paper. Subsequent to the completion of our analysis, McCullough and Wilson (1999) and Vinod (2000) analyzed Excel and Gauss, respectively. Their results do not differ substantively from our own.

7. Replication datasets and certified values are available from <www.itl.nist.gov>.

8. Not reported here, we, and McCullough (1998, 1999a, 1999b), test the reliability of the built-in pseudorandom number generators. Many of the tests are described previously in Marsaglia (1984). Researchers who perform simulation should be wary of their random number generators, as we found many were known to be poor. The results of our tests and a C library containing the current "best" generator can be found in the replication archive.

9. Since statistical distributions are often used in situations where it is important to be accurate to more than one significant digit, we also denote when there are some inaccuracies in the second through fifth digits of the distribution.

10. We (Altman and McDonald 2001) replicate two studies that use SHAZAM (Alvarez and Brehm 1995; Nagler 1994) and one that uses E-Views (Franzese 1999) since these programs fail a number of benchmarks. Replicating these analyses on multiple software packages with reliable benchmarks reveals some discrepancies, but in no case were the strong results of these studies overturned.

11. Since this paper was originally written, many of the statistical packages have been significantly updated. We are hopeful that numeric accuracy is improved with new versions, but we recommend that prudent researchers retest these new versions nonetheless.

12. Since most companies do not notify their customers of bug fixes, unless the bug results in spectacular failures, users should consider periodically checking with the company, perhaps at the start of each new project.

References

Altman, Micah, and Michael P. McDonald. 2001. "Replication with Attention to Numerical Accuracy." *Political Analysis*. Forthcoming.

Alvarez, R. Michael, and John Brehm. 1995. "American Ambivalence Towards Abortion Policy: Development of a Hetreoskedastic Probit Model of Competing Values." *American Journal of Political Science* 39:1055–89.

Beck, Nathaniel. 1995. "What Software Do Political Scientists Use?" *The Political Methodologist* 7(1): 23–26.

Brown, Barry W., David Gutierrez, James Lovato, Marty Spears, and John Venier. 1998. DSTATTAB v 1.1. (computer program). Houston: University of Texas.

Franzese, Robert J. 1999. "Partially Independent Central Banks, Politically Responsive Governments, and Inflation." *American Journal of Political Science* 43:681–706.

Gill, Jefferson. 1999. "The Insignificance of Null Hypothesis Significance Testing." *Political Research Quarterly* 52:647–74.

Gill, Phillip E., Walter Murray, and Margaret H. Wright. 1981. *Practical Optimization*. San Diego: Academic Press, Inc.

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press

Higham, Nicholas J. 1996. *Accuracy and Stability of Numerical Algorithms*. Philadelphia: SIAM.
King, Gary. 1989. *Unifying Political Methodology.* Cambridge: Cambridge University Press.
——. 1997. *A Solution to the Ecological Inference Problem*. Princeton: Princeton University Press.
——. 2001. "Numerical Stability vs. Specification Uncertainty: A Concluding Comment on Cho and Gaines, and Burden and Kimball." *American Political Science Review.* Forthcoming.
Knüsel, Leo. 1989. Computergestützte Berechnung statistischer Verteilungen. Oldenbourg, München-Wien. <www.stat.uni-muenchen.de/~knuesel/elv/elv.htm>.
——. 1995. "On the Accuracy of Statistical Distributions in GAUSS." *Computational Statistics and Data Analysis* 20:699–702.
——. 1998. "On the Accuracy of the Statistical Distributions in Microsoft Excel." *Computational Statistical and Data Analysis* 26:375–77.
Knuth, Donald E. 1997. *The Art of Computer Programming, 3rd Edition*. Reading, MA: Addison-Wesley.
Longley, James W. 1967. "An Appraisal of Computer Programs for the Electronic Computer from the Point of View of the User." *Journal of the American Statistical Association* 62:819–41.
Marsaglia, George. 1984. "A Current View of Random Number Generators." Presented at Computer Science and Statistics: 16th Symposium on the Interface, New York.
McCullough, Bruce D. 1998. "Assessing the Reliability of Statistical Software: Part I." *The American Statistician* 52:358–66.
——. 1999a. "Econometric Software Reliability: Eviews, LIMDEP, SHAZAM, and TSP." *Journal of Applied Econometrics* 14:191–202.
——. 1999b. "Assessing the Reliability of Statistical Software: Part II." *The American Statistician* 53: 149–59.
——, and H. D. Vinod. 1999. "The Numerical Reliability of Econometric Software." *Journal of Economic Literature* 37:633–65.
——, and B. Wilson. 1999. "On the Accuracy of Statistical Procedures in Microsoft Excel 97." *Computational Statistics and Data Analysis* 31:27–37.
Mooney, Christopher Z. 1997. *Monte Carlo Simulation, Quantitative Applications in the Social Sciences #113.* Thousand Oaks, CA: Sage.
Nagler, Jonathan. 1994. "Scobit: An Alternative Estimator to Logit and Probit." *American Journal of Political Science* 38:230–55.
Rogers, Janet, James Filliben, Lisa Gill, William Guthrie, Eric Lagergren, and Mark Vangel. 2000. "StRD: Statistical Reference Datasets for Testing the Numerical Accuracy of Statistical Software." NIST # 1396. Washington, DC: National Institute of Standards and Technology.
Simon, Stephen D., and James P. Lesage. 1988. "Benchmarking Numerical Accuracy of Statistical Algorithms." *Computational Statistics and Data Analysis* 7:197–209.
Tam Cho, Wendy, and Brian Gaines. 2001. "Reassessing the Study of Split-Ticket Voting." *American Political Science Review*. Forthcoming.
Vinod, H.D. 2000. "Review of GAUSS for Windows Including Its Numerical Accuracy." *Journal of Applied Econometrics* 15:211–20.

https://doi.org/10.1017/S1049096501001093 Published online by Cambridge University Press