

EXHIBIT J



On the accuracy of statistical procedures in Microsoft Excel 2007

B.D. McCullough^{a,*}, David A. Heiser^b

^a Department of Decision Sciences, LeBow College of Business, Drexel University, Philadelphia, PA 19104, United States

^b Carmichael, CA, United States

ARTICLE INFO

Article history:

Available online 12 March 2008

ABSTRACT

Excel 2007, like its predecessors, fails a standard set of intermediate-level accuracy tests in three areas: statistical distributions, random number generation, and estimation. Additional errors in specific Excel procedures are discussed. Microsoft's continuing inability to correctly fix errors is discussed. No statistical procedure in Excel should be used until Microsoft documents that the procedure is correct; it is not safe to assume that Microsoft Excel's statistical procedures give the correct answer. Persons who wish to conduct statistical analyses should use some other package.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

McCullough (1998) proposed an intermediate-level set of tests for assessing statistical software. This methodology has been applied to statistical and econometric software (McCullough, 1999a,b; Kitchen et al., 2003; Keeling and Pavur, 2007; Yalta, 2007). The methodology assesses the software in three areas: statistical distributions, random number generation, and the NIST StRD ("Statistical Reference Datasets", www.itl.nist.gov/div898/strd). Typically, when errors are found in a software package, the developer fixes the errors, and the matter is done; there is no further need to re-evaluate the software on this set of tests. Not so with Microsoft and its product Excel. Microsoft occasionally fixes errors, more often ignores them, and sometimes fixes them incorrectly. Consequently, every time there is a new version of Excel, the tests must be repeated.

Indeed, every time a new version of Excel is released, we receive emails asking, "Is it safe to use Excel?" There appears to be a sentiment that if only Excel would pass the intermediate tests, then it would be safe to use. Nothing could be farther from the truth. Microsoft has not even fixed all the flaws identified by Sawitzki (1994) over 15 years ago (his paper took a couple years to be published). There are scores of statistical functions in Excel, and we have benchmarked only a fraction of them. In the fraction that we have examined, we have found enough unfixed/incorrectly-fixed errors to cast grave doubt on the above-mentioned sentiment. To wit, we write this paper in part to warn members of the statistical community that even should there come a day when Microsoft can fix enough of the errors in Excel that Excel can pass these intermediate-level tests, it will not necessarily be safe to use.

In this paper we repeat the intermediate level tests. We also examine some other statistical procedures in which Excel has long been deficient. This is by no means intended to be a comprehensive list but, rather, merely a sampling of errors that we have encountered and that Microsoft has not bothered to fix. Given Microsoft's track record, we recommend that no statistical procedure be used unless Microsoft demonstrates that the procedure in question has been correctly programmed, e.g., by the use of test datasets from textbooks, or by comparison with another software package.

In Section 2 we discuss the intermediate level tests and examine a major flaw in the Excel Solver. In Section 3 we show that Excel's procedure for performing Exponential Smoothing is grievously flawed; we wonder how such obvious errors could have been made. Section 4 similarly analyzes Excel's flawed LOGEST function. Section 5 discusses Excel's "Normal

* Corresponding author. Tel.: +1 215 895 2134.

E-mail addresses: bdmccullough@drexel.edu (B.D. McCullough), dah_box1@innercite.com (D.A. Heiser).

Probability Plot” ATP function, not from a graphical perspective but from a statistical perspective. In the statistical literature, the usual Normal Probability Plot compares regression residuals to a normal distribution as a way to check the normality of the residuals. Excel’s version compares the dependent variable to a uniform distribution, though we are unaware of any reason that a user might wish to check the dependent variable for uniformity. In Section 6 we recount some miscellaneous errors that are by no means exhaustive, but merely illustrative of the difficulties that await the unsuspecting Excel user. Section 7 presents the conclusions.

2. Intermediate level tests

In this section, we update previous assessments of Excel (McCullough and Wilson, 1999, 2002, 2005), and elaborate on Excel Solver’s inability to identify that it has stopped at a point that is not a solution.

Excel’s statistical distributions have always been inadequate. Over the years, Microsoft has fixed some distributions, fixed others incorrectly, and failed to fix others. For further details, see the accompanying article by Yalta (2008). The performance of Excel 2007 on this suite of tests can be judged inadequate.

The random number generator has always been inadequate. With Excel 2003, Microsoft attempted to implement the Wichmann–Hill generator and failed to implement it correctly. The “fixed” version appears in Excel 2007 but this “fix” was done incorrectly. Microsoft has twice failed to implement correctly the dozen lines of code that constitute the Wichmann–Hill generator; this is something that any undergraduate computer science major should be able to do. The Excel random number generator does not fulfill the basic requirements for a random number generator to be used for scientific purposes: (1) it is not known to pass standard randomness tests, e.g., L’Ecuyer and Simard’s (2007) CRUSH tests (these supersede Marsaglia’s (1996) DIEHARD tests—see Altman et al. (2004) for a comparison); (2) it is not known to produce numbers that are approximately independent in a moderate number of dimensions; (3) it has an unknown period length; and (4) it is not reproducible. For further discussion of these points, see the accompanying article by McCullough (2008); the performance of Excel 2007 in this area is inadequate.

The NIST StRD has five suites of test problems: univariate summary statistics, one-way ANOVA, linear regression, nonlinear least squares, and Markov-chain Monte Carlo (this last suite is not applicable here). Each of these suites contains several test problems with known solutions. The performance of Excel 97, Excel 2000 and Excel XP in all four areas was unacceptable. In Excel 2003, Microsoft made improvements to the univariate, ANOVA and linear regression procedures and performed satisfactorily in those areas. However, with default settings Solver returns zero digits of accuracy for over half of the 27 nonlinear test problems and, even tuned for better accuracy, still returns zero digits of accuracy for 11 of the 27 problems (Berger, 2007). Consequently, Excel 2003 failed the StRD portion of the tests.

Applying the StRD to Excel 2007 indicates that Microsoft made no changes in this area for Excel 2007. The performance of Excel 2007 in this area is inadequate. It has been noted that Excel Solver has a marked tendency to stop at a point that is not a solution and declare that it has found a solution. See also Mondragon and Borchers (2005). Many books recommend Solver for solving nonlinear statistical problems; see, *inter alia*, Barreto and Howland (2006), de Levie (2003), and Gottfreid (2002). While these books tend to be used in applied settings (*i.e.*, not for publication in scholarly journals), Solver is used in academic journal articles: for solving a GARCH problem in finance (Jones et al., 1998); for factor analysis (Miles, 2005); for modeling kinetic chemical reactions (Denton, 2000); in chromatography (Nikitas and Pappa-Louisi, 2000). It is not difficult to find uses of Solver for statistical purposes. Here we briefly consider the “Misra 1a” StRD nonlinear problem that is discussed at length in McCullough (2004a):

$$y = \beta_1(1 - \exp(-\beta_2x)) + \epsilon \quad (1)$$

for $n = 14$ observations and starting values of 500 for β_1 and 0.0001 for β_2 . The correct solution is $\hat{\beta}_1 = 238.94212918$, $\hat{\beta}_2 = 0.00055015643181$ and sum of squared residuals = 0.12455138894.

The Solver default algorithm is a “Newton” method, with an option for a “Conjugate” method. The default convergence tolerance (“ct”) is 0.001. Running from default, Solver stops in eight iterations, yielding a “solution” $\hat{\beta}_1 = 454.12$ and $\hat{\beta}_2 = 0.0002676$; Solver specifically states: “Solver has converged to the current solution. All constraints are satisfied.” Such a message might lead a researcher to conclude that the software had found a solution to the nonlinear least squares problem Misra 1a, but such a conclusion would be incorrect. If the researcher could examine the gradient (which should componentwise be zero at a point that minimizes the sum of squared residuals), he would find that the gradient was nowhere zero, not even close to zero. However, Solver does not allow the user to inspect the gradient.

If the researcher simply changed the algorithm to “Conjugate” then Solver would stop in seven iterations, and report “Solver has converged to the current solution. All constraints *and optimality conditions* are satisfied.” [*italics added*] This seemingly more impressive declaration of having found a solution would be even more misleading than the prior declaration. This time Solver reports $\hat{\beta}_1 = 500$ and $\hat{\beta}_2 = 0.0002422$. Apparently the first “solution” did not satisfy some “optimality conditions” but this second solution does, despite the fact that Solver has not budged from the starting point for β_1 , and despite the fact that the sum of squared residuals has *increased* from 16.725 to 19.516. Of course, the gradient at this point is not even close to zero.

Nonlinear solvers may stop for a variety of reasons other than having reached a solution (see McCullough and Renfro (2000, Section 7) for a discussion), and a reliable solver can distinguish whether a stopping point is or is not a solution.

As mentioned above, Solver does find the correct solution for 16 of the 27 nonlinear problems. However, the only reason we are able to know that Solver has actually found the correct solution is because we know what the correct solutions are in advance. If we had to rely on Solver's convergence messages, we would have no confidence that it had stopped at the correct solution. Therefore, let us be explicit: do not use the Excel Solver to solve nonlinear least squares problems. Additionally, in the absence of evidence that Solver can reliably solve a particular class of nonlinear problems, we would hesitate to use it for any such problem.

3. Exponential smoothing

Exponential Smoothing is a classical statistical procedure. Here we follow the standard derivation given by [Montgomery et al. \(1990, p. 82\)](#). For a series X_t , model it as

$$X_t = b + \epsilon_t \quad (2)$$

where b is the expected demand in any period and ϵ_t is an error term. In period t , we have the estimate of b made in the previous period, \hat{b}_{t-1} as well as the current value of the series, X_t . The forecast error is given by $e_t = X_t - \hat{b}_{t-1}$. Using this information to create an updated forecast of the current period, \hat{b}_t , we can modify the old forecast by some proportion α of the forecast error to yield

$$\hat{b}_t = \hat{b}_{t-1} + \alpha(X_t - \hat{b}_{t-1}). \quad (3)$$

Setting $\hat{b}_t = S_t$ produces the more usual equation

$$S_t = S_{t-1} + \alpha(X_t - S_{t-1}) \quad (4)$$

or

$$S_t = \alpha X_t + (1 - \alpha)S_{t-1}, \quad 0 \leq \alpha \leq 1 \quad (5)$$

where α is commonly referred to a "smoothing constant". Obviously the initial value of S_t must be set in some fashion, and there are many such ways discussed in the literature. For our purposes it suffices to note that simply starting with $t = 1$ and setting $X_1 = S_1$ and then calculating S_2, S_3, \dots is perfectly acceptable ([Newbold and Bos, 1990, p. 161](#)).

Microsoft's implementation of Exponential Smoothing is found in the Analysis ToolPak, and is worth quoting extensively from the documentation; see [Fig. 1](#).

All this seems well and good. Let us now use Excel to perform Exponential Smoothing on the thirty observations of sales data given by [Newbold and Bos \(1990, Table 6.1\)](#), presented in our [Table 1](#). Newbold and Bos also provide the smoothed values for $\alpha = 0.3$, and these values were the same ones produced by MINITAB, RATS, and every other statistical package we tried; these also are given in [Table 1](#).

Upon selecting the "Exponential Smoothing" option of the ATP, Excel asks for an "input range", a "damping factor" and an "output range". It seems reasonable to suppose that the "damping factor" is the "smoothing constant" referred to in the Excel help files, so we entered 0.3. The first thing to notice about the Excel output is that it does not give a smoothed value for the first observation, and gives the smoothed value of 86 as 103. If the procedure is initialized in the usual fashion, setting $X_1 = S_1$, then we would expect 103 to be the smoothed value for the first observation. While this is definitely incorrect, we might try to make sense of this by deducing that Microsoft's mistake was to shift all the smoothed values by one place. However, the smoothed value after 103 is not the same as that given by Newbold and Bos (and the other statistical packages). A little reverse engineering reveals that the relationship between the "damping factor" that the program asks for, and the "smoothing constant" referred to in the documentation is

$$\text{damping factor} = 1 - \text{smoothing constant}. \quad (6)$$

Indeed, when we use a damping factor of 0.7, we get the correct numbers, but they are still in the incorrect positions. Employing equation (6) is blatantly misleading, and even if one is warned about this, the Excel output is still wrong: it gives incorrect smoothed values for all the observations.

The Exponential Smoothing analysis tool predicts a value based on the forecast for the prior period, adjusted for the error in that prior forecast. The tool uses the smoothing constant α , the magnitude of which determines how strongly forecasts respond to errors in the prior forecast. **Note** Values of 0.2 to 0.3 are reasonable smoothing constants. These values indicate that the current forecast should be adjusted 20 to 30 percent for error in the prior forecast. Larger constants yield a faster response but can produce erratic projections. Smaller constants can result in long lags for forecast values.

Fig. 1. Help file for exponential smoothing.

Table 1
Exponential smoothing results

Obs.	Sales	Smoothed values		
		Newbold/Bos $\alpha = 0.3$	Excel Damping = 0.3	Excel Damping = 0.7
1	103	103	#N/A	#N/A
2	86	97.90	103	103
3	84	93.73	91.10	97.90
4	84	90.81	86.13	93.73
5	92	91.17	84.64	90.81
...				
26	116	108.30	105.13	105.00
27	100	105.81	112.74	108.30
28	128	112.47	103.82	105.81
29	109	111.43	120.75	112.47
30	122	114.60	112.52	111.43

We searched several textbooks, and could not find any exponential smoothing example that even remotely resembles what Microsoft has programmed into Excel for this function. Indeed, we cannot imagine that there is a textbook which gives Excel's output for this particular input. We are left to wonder, How does Microsoft check the accuracy of the procedures it programs into Excel? Does anyone use the program before writing the help files? What else might account for the “damping factor” / “smoothing constant” confusion?

4. LOGEST

Excel's LOGEST and GROWTH functions are bad; here we only discuss the LOGEST function, which has been analyzed separately by Cook et al. (1999) and Hesse (2006) and who reached the same conclusion. To see why the LOGEST function is bad, it is useful to start with the Excel Help file, given in Fig. 2.

LOGEST: In regression analysis, calculates an exponential curve that fits your data and returns an array of values that describes the curve. Because this function returns an array of values, it must be entered as an array formula.

The equation for the curve is:

$$y = b * m^x \quad \text{or}$$

$$y = (b * (m1^{x1}) * (m2^{x2}) * \dots) \quad (\text{if there are multiple x-values})$$

where the dependent y-value is a function of the independent x-values. The m-values are bases corresponding to each exponent x-value, and b is a constant value. Note that y, x, and m can be vectors. The array that LOGEST returns is mn,mn-1,...,m1,b.

Fig. 2. Excel help file for LOGEST.

As is obvious, the equation $y = b * m^x$ is a nonlinear equation and needs to be solved nonlinearly. We know that the Excel Solver is not reliable and its results would have to be checked by using a reliable package, so let us solve the problem in a reliable package to begin with. We choose “R”, and use the example data from the Excel Help files.

```
units <- c(33100,47300,69000,102000,150000,220000)
month <- c(11,12,13,14,15,16)
summary( nls(units~b*m^month,start=list(b=500,m=1.5))
```

```
Formula: units ~ b * m^month
```

```
Parameters:
```

```
Estimate Std. Error t value Pr(>|t|)
b 473.71781 10.86801 43.59 1.66e-06 ***
m 1.46784 0.00221 664.09 3.08e-11 ***
```

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 503.2 on 4 degrees of freedom
```

However, the additional statistics LOGEST returns are based on the following linear model:

$$\ln y = x_1 \ln m_1 + \dots + x_n \ln m_n + \ln b$$

You should keep this in mind when you evaluate the additional statistics, especially the *sei* and *seb* values, which should be compared to $\ln m_i$ and $\ln b$, not to m_i and b . For more information, consult an advanced statistics manual.

Fig. 3. Excel help file on LOGEST standard errors.

In sharp contrast, the Excel LOGEST output is

	estimate	std error
b	495.3048	0.035834
m	1.463276	0.002633

The coefficients are markedly different. We can indulge a presumption that when “*R*” and Excel disagree, Excel is wrong. The sum of squared residuals from the estimated models yields 2,758,373 for Excel and 1,012,875 for *R*; our presumption appears to be justified. Consequently, it appears that Excel coefficients are accurate to one digit for *b* and three digits for *m*. Abstracting from the issue of how to calculate standard errors for the coefficients in a nonlinear model, and simply assuming that *R* is correct and Excel is wrong, we see that the Excel standard errors are not accurate even to one digit! Why do these differences occur?

The reason for the differences in the coefficients and sum of squared errors is that Excel uses a crude linear approximation. LOGEST actually runs a linear regression on

$$\log(\text{units}) = c + n \text{ month} \quad (7)$$

to obtain $c = 6.2052$ (0.035834) and $n = 0.3807$ (0.002633) [standard errors in parentheses] and then uses exponentiation to obtain $b = \exp(c) = \exp(6.2052) = 495.3048$ and $m = \exp(n) = \exp(0.3807) = 1.463276$. However, this crude approximation does *not* minimize the sum of squared residuals. The Excel approach minimizes

$$\sum_i (\log_i y_i - (x_i \log m + \log b))^2$$

which is clearly not the same thing as minimizing

$$\sum_i (y_i - bm_i^x)^2$$

though, from reading the Excel Help files, Microsoft clearly believes that the two methods are equivalent.

As to the differences in the standard errors, the relevant information for *m* in the Microsoft Help File is given in Fig. 3.

Of course, if the relevant persons at Microsoft had themselves consulted an “advanced statistics manual” they would have discovered their error. Nonetheless, Excel is presenting the standard errors from the regression given by Eq. (7) with the exponentiated coefficients from that regression. To present standard errors that are not to be interpreted on the same basis as the coefficients is highly unusual, even misleading, given the customary statistical practice of dividing a coefficient by its standard error to compute a *t*-statistic. This information about the presented standard errors is hidden at the bottom of the Help file and a user who is accustomed to standard statistical output might be forgiven for calculating *t*-statistics as the ratio of the presented coefficient to the presented standard error. Further, Microsoft makes the point that it exponentiates the coefficients to obtain the desired estimates; a novice might well reason by analogy that he should exponentiate the presented standard errors. Of course, exponentiating the standard errors is not the right thing to do, but let us do it anyhow: for *b* we have $\exp(0.035834) = 1.036484$ and for *m* we have $\exp(0.002633) = 1.002636$. Why does Microsoft even bother to present these useless and misleading standard errors? We are unable to find or imagine any reason for such a practice.

The GROWTH function suffers from the same methodological flaw: using a crude and incorrect approximation instead of the correct approach (Cook et al., 1999).

5. Normal probability plot

We discuss this here because from our perspective, this is not a graphical issue but a statistical issue: what exactly are the statistics that underlie the Normal Probability Plot? After running a regression, it is often useful to check the residuals for normality. A common way to do this is to graph a Normal Probability Plot of the residuals. The Normal Probability Plot puts the expected value of the order statistic of the standard normal distribution on the abscissa with the corresponding data value on the ordinate. If the data are normally distributed, then the plot should be a straight line. Departures from a straight line indicate departures from normality.

In the ATP Regression procedure there is a box for “Normal Probability Plots”. The entire Help File for this function, given in Fig. 4, is especially useless.

Normal Probability Plots

Select to generate a chart that plots normal probability.

Fig. 4. Excel help file for normal probability plot.

In a departure from customary practice, the user is not told how the plot is generated, nor even of what variable is plotted. But it would not be outlandish for a trusting user to suppose that he would see the usual normal probability plot of the residuals. Such trust would be betrayed.

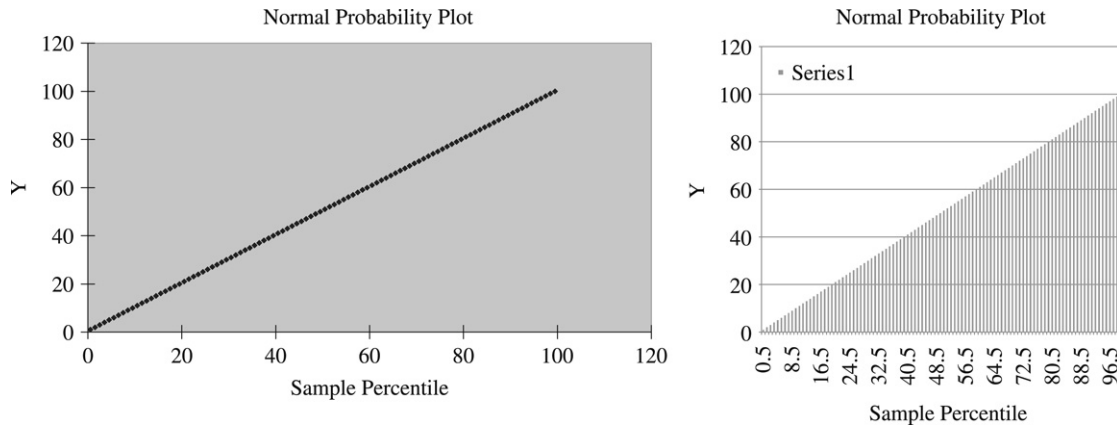


Fig. 5. Excel 2003 (left) and Excel 2007 (right) “normal probability plots” for sequence 1–100.

To investigate this feature, we ran a regression whose dependent variable is the integers 1 through 100 and whose independent variable is 100 random uniforms. The Normal Probability Plot from this regression has “Sample Percentile” on the abscissa and “Y” on the ordinate. Note that the confusing chart Excel 2003 (see Fig. 5(left)) is even more confusing in the “improved” Excel 2007 (see Fig. 5(right)): why the vertical bars for something that should be represented by a line? The trusting user might think that perhaps this is a Normal Probability Plot of the dependent variable, but he would be wrong. It is somewhat difficult to see given the poor aspect ratio, but the plot is a straight line. The Normal Probability plot is only supposed to yield a straight line if the input variable is normally distributed, and in this instance the input variable is the sequence 1 through 100. Some insight into this mystery can be gained from the “Probability Output” table that corresponds to Fig. 5 is given in Table 2.

Table 2
“Probability output” from Excel ATP regression

Percentile	Y
0.5	1
1.5	2
2.5	3
3.5	4
...	...
96.5	97
97.5	98
98.5	99
99.5	100

Obviously the data in the “Percentile” column are not the expected values of the order statistics of the standard normal, and neither are they the percentiles corresponding to the expected values of the order statistics of the standard normal. What, then, are they?

A bit of math shows that the “Percentile” column is governed by the formula

$$\text{Percentile} = \frac{100i - 50}{n} \quad \forall i = 1, 2, \dots n.$$

In truth, the plot should be labeled “Uniform Probability Plot” because Microsoft is comparing the dependent variable to a uniform distribution. In sum, as far as the statistical theory of the Excel “Normal Probability Plot” is concerned, Excel computes the plot for the wrong variable, the dependent variable instead of the residuals, and has managed to confuse the

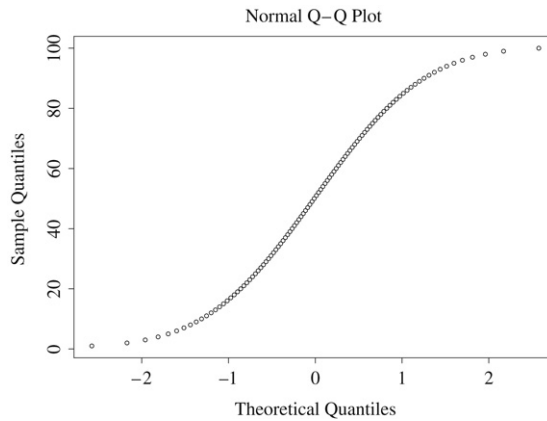


Fig. 6. Correct “normal probability plot” for sequence 1–100.

uniform distribution for the normal distribution. For purposes of comparison, the correct normal probability plot for the sequence 1 through 100 is given by R , and is presented in Fig. 6. The nonnormality of the data is evident, whereas Excel incorrectly indicated that the uniform data were normal.

6. Miscellaneous

Here we present several other errors in Excel of which we are aware. Again, we stress that this is not a comprehensive list. We simply wish to warn the reader that errors in Excel are not limited to those we have discussed above, and that a particular procedure is not necessarily safe to use just because no errors have been reported in newsgroups or elsewhere. We note that Microsoft does not make available to Excel users a list of the errors of which Microsoft is aware. The most comprehensive such resource of which we are aware is Heiser’s (2006a) website, “Microsoft Excel 2000, 2003 and 2007 Faults, Problems, Workarounds and Fixes,” (www.daheiser.info/excel/frontpage.html).

- *Bad random normal generator in ATP.* Use the pull-down menus for “Tools”, “Data Analysis”, and “Random Number Generation”; enter “10” for number of variables, “2000” for number of random variables, leave mean set to zero and standard deviation set to unity, enter “123” for “Random Seed” and click ok. Inspect the resulting worksheet. Cells 253:H, 1212:G, and 1245:D will show “−9.53674” and cell 1845:H will show “5.364418”. Values this extreme should not be seen in a properly functioning standard normal random number generator. McCullough and Wilson (2002) noted this problem years ago in Excel 2000/XP, and Microsoft has neither fixed it nor noted in the help files that this function is bad.
- Heiser (2006b) notes several problems with two-sample test procedures in Excel 2003, including misleading documentation and inaccurate routines, all of which are present in Excel 2007. Below is a partial list:
 - *Inaccurate t -test results in the presence of missing values.* Does not correctly calculate t test values when there are missing data cells in the range. Microsoft identified this fault (KBA 829252) in previous Excel versions, but did not fix it.
 - *Inaccurate p -values from a t -test.* Excel uses the Welch test method, which calculates a non-integer degrees-of-freedom value. The Excel algorithm uses the TTEST function, which truncates the degrees-of-freedom input to an integer. This gives an incorrect p value for the test.
 - *Incorrect labeling of t -test and z -test tables.* For the t -test, Excel describes the one tail test as “ $P(T \leq t)$ one-tail” and incorrectly describes the two tail test as “ $P(T \leq t)$ two-tail”.
- *Bad regression results from ATP “Trendline” function for ill-conditioned data.* Excel graphics have a feature whereby right-clicking on a data point enables the user to fit one of six trendlines to the data: Linear, Logarithmic, Polynomial, Power, Exponential, and Moving Average, as an option the fitted equation and its R^2 can be placed on the figure. Pottel (2003), using an ill-conditioned dataset from Simonoff (2002), shows that the Linear trendline calculates a different equation than either the ATP Regression procedure or the LINESST command. For the Simonoff data, Trendline calculates

$$\hat{y} = 1E + 09 - 0.125x \quad R^2 = -0.6667$$

(the negative sign in the R^2 is not a typo) while the other two methods yield

$$\hat{y} = 9903477999 + 0.096522x \quad R^2 = 0.000283.$$

- *Bad Exponential and Power results from ATP “Trendline” function even for datasets that are not ill-conditioned.* There are five Trendlines that ATP can place on a graph. Hesse (2006) shows that the Exponential Trendline and Power Trendline are incorrect. Does any user of Excel wish to bet that the remaining three Trendlines are correctly implemented?
- *Bad linear regression algorithm.* Until version Excel 2003, Excel could not pass the StRD linear regression problem “Filip”, which is a 10th order polynomial that is nearly singular. Excel would give completely incorrect coefficients. Microsoft improved the linear regression algorithm in Excel 2003, and both Excel 2003 and Excel 2007 correctly solve the Filip

problem. None of the StRD problems tests for perfectly collinear data. A reliable linear regression routine should be able to detect perfectly collinear data (for which a traditional regression solution does not exist) and refuse to report a solution. Such a test is Test IV-C from [Wilkinson's \(1985\) Tests](#) (see also [McCullough \(2004b\)](#)). Excel 4.0 failed this test ([Sawitzki, 1994](#)), and so does Excel 2007. Excel reports an answer, which is necessarily incorrect. If Excel warned the user, “rank deficiency encountered” then it would be okay. On the basis that the linear regression routine cannot recognize a singular regression, we can judge the linear regression routine in Excel to be unacceptable.¹

- *Bad linear regression output.* When Excel detects a singular design matrix, it removes a column from the model and sets that coefficient and standard error to zero, as per KB 828533. In this situation, the ATP fails to subtract one from the numerator degrees of freedom and, as a consequence, the “ANOVA OUTPUT” (for testing the significance of the regression) is incorrect: the regression sum of squares is correct, but its mean-square and consequently the F -statistic are incorrect, too, as is the p -value for the F -statistic.
- *Non-standard operator precedence.* In FORTRAN and most computer languages with which we are familiar, if $X = -1$ then X^2 evaluates to $+1$ and $-X^2$ evaluates to -1 . In Excel, $-X^2$ evaluates to $+1$. See [Berger \(2007\)](#) for further discussion. While Microsoft is free to use whatever rules it choose for operator precedence, to the extent that they are nonstandard, users should be clearly warned.
- *Bad default options for graphics.* Representing data graphically is more than just flashy presentation. In fact, it is a well-defined discipline called “statistical graphics” that has its own organization (American Statistical Association Section on Statistical Graphics), its own journal (*Journal of Computational and Statistical Graphics*), and several books. Canned graphical procedures in a software package should make it easy for users to produce graphics that embody the best practices of the discipline of statistical graphics. Specifically, the software, and not the user, should be “knowledgeable” about how to display data well. Microsoft has chosen a contrary route; only users who already are intimately familiar with the discipline of statistical graphics can produce good graphs in Excel, and then only after the tedious exercise of figuring out the appropriate way to override the Excel defaults. See the article by [Su \(2008\)](#) in this issue.

7. Conclusions

The statistical literature has regularly identified flaws in Excel's statistical procedures at least since [Sawitzki \(1994\)](#), and Microsoft has repeatedly proved itself incapable of providing reliable statistical functionality. It is little wonder that introductory texts on statistics warn students not to use Excel when the results matter (e.g., [Keller \(2001\)](#) and [Levine et al. \(2002\)](#)). There is nothing inherently difficult about programming a spreadsheet that precludes reliable statistical programming. The open-source spreadsheet package Gnumeric was originally such a good Excel clone that it even reproduced many Excel errors. When informed that these were errors, the handful of part-time programmers who maintain Gnumeric managed to fix them correctly in a matter of months ([McCullough, 2004c](#)). What is particularly pernicious, at least so far as consumers of statistics are concerned, is that Microsoft seems to take the approach that specialized domain knowledge is not necessary to the production of statistical code, *i.e.*, that the entire field of statistical computing has no merit. As but one case to illustrate this point, in KB 829215, Microsoft discusses the enhancements made to the ATP ANOVA function in Excel 2003. The KB describes pseudocode for the computations in previous versions of Excel; this pseudocode embodies an algorithm that would be implemented by no one who has read an introductory text on statistical computing, e.g., [Kennedy and Gentle \(1980\)](#) or [Thisted \(1988\)](#). Microsoft then remarks,

Again, for model 2 and model 3, sums of squares are calculated and a quantity is subtracted from the sum of squares as in the calculator formula. Unfortunately, basic statistics texts frequently suggest approaches for ANOVA such as the one that is shown earlier in this article.

The obvious implication is that Microsoft programmed its statistical functions by using statistics texts, rather than statistical computing texts. Of course, this approach to programming – eschewing specialized domain knowledge – explains much of Microsoft's inability to program functions correctly, or fix them correctly.

The improvements in Excel 2003 gave hope that Microsoft might finally be committed to providing reliable statistical functionality in Excel. That Microsoft once again decided to allow errors to remain unfixed, or fixed incorrectly, in the Excel 2007 release dashes those hopes. Regardless of the reasons for Microsoft's inability to program Excel correctly, Microsoft has had years to fix these errors, and has proven itself unable to do so. It is clear that when Microsoft claims to have “fixed” a procedure that is known to be in error, Microsoft's claim cannot be safely believed. It is also clear that there are procedures in Excel that are in error, and Microsoft either does not know of the error or does not choose to warn users that the procedure is in error. Since it is generally impossible for the average user to distinguish between the accurate and inaccurate Excel functions, the only safe course for Excel users is to rely on no statistical procedure in Excel unless Microsoft provides tangible evidence of its accuracy consisting of:

¹ How exactly Excel should inform the user of a singular regression is not clear. Should the program generate a specific error message instead of producing any output, or should some of the cells in the output return the “#NUM” message? Either approach would bring a program to a crashing halt, but perhaps this would be a good thing. Usually, singular data imply that whoever entered the data made some sort of mistake and, if the input data are bad, the program probably should not run to completion.

1. test data with known input and output;
2. description of the algorithm sufficient to permit a third party to use the test data to reproduce Excel's output; and
3. a bona fide reference for the algorithm.

If Microsoft does not perform these actions for each statistical procedure in Excel, then there are only two safe alternatives for the user who is concerned about the accuracy of his statistical results: the user can perform all these actions himself, or simply not use Excel.

Acknowledgements

Thanks for comments to the referee, and also to Patrick Burns, Jerry Lewis, John Nash, Yu-Sung Su, and A. Talha Yalta.

References

- Altman, Micah, Gill, Jeff, McDonald, Michael P., 2004. Numerical Issues in Statistical Computing for the Social Scientists. John Wiley and Sons, New York.
- Barreto, Humberto, Howland, Frank M., 2006. Introductory Econometrics: Using Monte Carlo Simulation with Microsoft Excel. Cambridge University Press, New York.
- Berger, Roger L., 2007. Nonstandard operator precedence in Excel. *Computational Statistics & Data Analysis* 51 (6), 2788–2791.
- Cook, H.R., Cox, M.G., Dainton, M.P., Harris, P.M., 1999. Testing spreadsheets and other packages used in metrology: Testing the intrinsic functions of Excel. Report to the National Measurement System Policy Unit, Department of Trade and Industry, from the UK Software Support for Metrology Programme. NPL Report CISE 27/99.
- de Levie, Robert, 2003. Advanced Excel for Scientific Analysis. Oxford University Press, New York.
- Denton, Philip, 2000. Analysis of first-order kinetics using Microsoft Excel solver. *Journal of Chemical Education* 77 (11), 1524–1525.
- Gottfried, Byron S., 2002. Spreadsheet Tools for Engineers Using Excel. McGraw-Hill, New York.
- Heiser, David A., 2006a. Microsoft Excel 2000 and 2003 faults, problems, workarounds and fixes. *Computational Statistics & Data Analysis* 51 (2), 1442–1443.
- Heiser, David A., 2006b. Statistical tests, tests of significance, and tests of a hypothesis using Excel. *Journal of Applied Statistical Methods* 5 (2), 155–171.
- Hesse, Rick, 2006. Incorrect nonlinear trend curves in Excel. *Foresight: International Journal of Applied Forecasting* 3, 39–43.
- Jones, Charles M., Lamont, Owen, Lumsdaine, Robin, 1998. Macroeconomic news and bond market volatility. *Journal of Financial Economics* 47, 315–337.
- Keeling, Kellie, Pavur, Robert, 2007. A comparative study of the reliability of nine statistical software packages. *Computational Statistics & Data Analysis* 51 (8), 3811–3831.
- Keller, G., 2001. Applied Statistics with Microsoft Excel. Duxbury, Belmont, CA.
- Kennedy, William, Gentle, James, 1980. Statistical Computing. Marcel Dekker, New York.
- Kitchen, A., Drachenberg, R., Symanzik, J., 2003. Assessing the reliability of web-based statistical software. *Computational Statistics* 18 (1), 107–122.
- L'Ecuyer, Pierre, Simard, Richard, 2007. TESTU01: A C library for empirical testing of random number generators. *ACM TOMS* 33 (4), Article 22.
- Levine, D., Stephan, D., Krehbiel, T., Berenson, M., 2002. Statistics for Managers Using Microsoft Excel, 3e. Prentice-Hall, Englewood Cliffs, NJ.
- Marsaglia, George, 1996. DIEHARD: A Battery of Tests of Randomness. <http://stat.fsu.edu/pub/diehard>.
- McCullough, B.D., 1998. Assessing the reliability of statistical software: Part I. *The American Statistician* 52, 358–366.
- McCullough, B.D., 1999a. Assessing the reliability of statistical software: Part II. *The American Statistician* 53 (2), 149–159.
- McCullough, B.D., 1999b. Econometric software reliability: E-Views, LIMDEP, SHAZAM, and TSP. *Journal of Applied Econometrics* 14 (2), 191–202.
- McCullough, B.D., 2004a. Some details of nonlinear estimation, chapter eight. In: Altman, Micah, Gill, Jeff, McDonald, Michael P. (Eds.), *Numerical Methods in Statistical Computing for the Social Sciences*. Wiley, New York.
- McCullough, B.D., 2004b. Wilkinson's tests and econometric software. *Journal of Economic and Social Measurement* 29 (1–3), 261–270.
- McCullough, B.D., 2004c. Fixing statistical errors in spreadsheet software: The cases of gnumeric and Excel. *CSDA Statistical Software Newsletter*, www.csdassn.org/software_reports.cfm.
- McCullough, B.D., 2008. Microsoft Excel's 'Not The Wichmann–Hill' random number generators. *Computational Statistics & Data Analysis*.
- McCullough, B.D., Renfro, Charles G., 2000. Some numerical aspects of nonlinear estimation. *Journal of Economic and Social Measurement* 26 (1), 63–77.
- McCullough, B.D., Wilson, Berry, 1999. On the accuracy of statistical procedures in Microsoft Excel 97. *Computational Statistics and Data Analysis* 31, 27–37.
- McCullough, B.D., Wilson, Berry, 2002. On the accuracy of statistical procedures in Microsoft Excel 2000 and XP. *Computational Statistics and Data Analysis* 40 (4), 27–37.
- McCullough, B.D., Wilson, Berry, 2005. On the accuracy of statistical procedures in Microsoft Excel 2003. *Computational Statistics and Data Analysis* 49 (4), 1244–1252.
- Miles, J.N.V., 2005. Confirmatory factor analysis using Microsoft Excel. *Behavior and Research Methods* 37 (4), 672–676.
- Mondragon, P.F., Borchers, B., 2005. A comparison of nonlinear regression codes. *Journal of Modern Applied Statistical Methods* 4 (1), 343–351.
- Montgomery, Douglas C., Johnson, Lynwood A., Gardiner, John S., 1990. Forecasting and Time Series Analysis, 2nd ed. McGraw-Hill, New York.
- Newbold, Paul, Bos, Theodore, 1990. Introductory Business Forecasting. Southwestern, Cincinnati, OH.
- Nikitas, P., Pappa-Louisi, A., 2000. Non-linear least-squares fitting with Microsoft Excel solver and related routines in HPLC modelling of retention. *Chromatographia* 52 (7–8), 477–486.
- Pottel, Hans, 2003. Statistical flaws in Excel. Mimeo, Innogenetics NV, Zwijnaarde, Belgium.
- Sawitzki, Gunther, 1994. Report on the numerical reliability of data analysis systems. *Computational Statistics and Data Analysis* 18 (2), 289–301.
- Simonoff, Jeffrey, 2002. Statistical analysis using Microsoft Excel 2000. Mimeo, Stern School of Business, NYU.
- Su, Yu-Sung, 2008. It's easy to produce chartjunk using Microsoft Excel 2007 but hard to make good graphs. *Computational Statistics and Data Analysis*.
- Thisted, R.A., 1988. Elements of Statistical Computing: Numerical Computation. Chapman and Hall, New York.
- Wilkinson, Leland, 1985. Statistics Quiz. Systat, Inc., Evanston, IL.
- Yalta, A.Talha, 2007. The numerical reliability of GAUSS 8.0. *The American Statistician* 61 (3), 262–268.
- Yalta, A.Talha, 2008. On the accuracy of statistical distributions in Microsoft Excel 2007. *Computational Statistics & Data Analysis*.