

XV. TESTING FOR ACCURACY AND RELIABILITY OF STATISTICAL DISTRIBUTIONS	2
INTRODUCTION	2
MEASURES OF COMPUTER ACCURACY OF A PROBABILITY VALUE	2
PROBABILITIES AS FLOATING POINT NUMBERS	2
PROBABILITY DISTRIBUTION VALUES IN TAIL AREAS	3
P VALUES GREATER THAN 0.9	3
P VALUES LESS THAN 0.01	3
THE PROBABILITY DISPLAY OBJECT	4
DISPLAY TO N ACCURATE DIGITS	7
ACCURACY NEEDED FOR A DECISION:	8
METHODS OF LIMITING THE P VALUE TO ONLY THE NUMBER OF ACCURATE DIGITS	9
1. ACCURACY ON A FIXED POINT BASIS:	9
2. ACCURACY ON A FLOATING POINT BASIS	9
3. ACCURACY ON A TRUNCATED FLOATING POINT BASIS	10
LIMITATIONS ON DEFINING ACCURACY BY NUMBER OF DIGITS	11
THE ESSENTIAL P VALUE ACCURACY PROBLEM	11
ACCURACY OF INVERSE STATISTIC VALUES	12
A DECISION BASED ON A STATISTIC	12
GENERATION OF RANDOM VARIATES:	12
GENERAL METHODS RELATED TO TESTING	13
SELECTION OF INPUT PARAMETER VALUES	13
RANDOM VALUES IN PARAMETER SPACE C	13
SHOTGUN METHOD	13
ACCURATE REFERENCE P VALUES	14
STANDARD TABLES OF VALUES:	14
SPECIAL REFERENCE FUNCTIONS AND SUBROUTINES IN VBA WITH PRECISE VALUES:	14
TEST MEASURES	17
ACCURATE DIGITS METHOD	19
LIMITATIONS ON THE ACCURACIES OF COMPLEMENTATION	20
INTERPRETATION OF TEST RESULTS	21

RELIABILITY OF PROBABILITY FUNCTION OUTPUTS	22
FALSE ZEROS.....	22
NON-NUMERIC ERROR MESSAGE RETURNS.....	23
GROSS ERRORS	24
NON-ENDING INTERNAL LOOPS OR LOGIC TRAPS	24
SYSTEM HALTS, CRASHES AND UNRESPONSIVE BEHAVIOR.....	24

XV. TESTING FOR ACCURACY AND RELIABILITY OF STATISTICAL DISTRIBUTIONS

INTRODUCTION

This section describes the testing program for testing the accuracies of the statistical distributions. The outputs are probability values, and as such have different measures of accuracies and different testing methods than that for a statistic.

Microsoft made no change to any of these function for the 2007 version. Therefore everything discussed and found on the 2003 version is applicable to the 2007 version.

In general the Excel function outputs (as floating-point {IF} objects) were compared to reference values ({IF} objects) from a unique, high accuracy set of VBA functions. Parameter space values were obtained by different methods, including random number (input parameter values) techniques.

MEASURES OF COMPUTER ACCURACY OF A PROBABILITY VALUE

PROBABILITIES AS FLOATING POINT NUMBERS

Probability values are numbers between 0 and 1. There are two situations here that often cannot be resolved.

The first is the mathematical view where p is a mathematical concept {IR}, not limited in any way by its representation by real numbers. Zero or one represent improbable occurrences, because probability theory arrives at this conclusion. The symbol {IR} represents the traditional, statistical concept of p as a purely mathematical object, or field.

The second comes from the fact that a value of p is an {IF} calculation. {IF} values of zero or one are now limits of a calculation, and occur frequently.

An {IF} value can be, zero (0), one (1), an error code (#NUM!) or a NZ number. A NZ number is any return from the probability function that is NOT a zero, is NOT a 1, is NOT #NUM! or is NOT any error code. It represents numbers between 0 and 1, but not at the ends. A zero (0) or a one (1) represent limits on the ability of the computer and algorithm to return a good {IR} approximation.

PROBABILITY DISTRIBUTION VALUES IN TAIL AREAS

P VALUES GREATER THAN 0.9

Probability values greater than 0.9 represent a specific, unresolved accuracy problem. Here, the leading digits are all nines, and are not treated as “non-digits” such as leading “zeros” are. There is no shifting in the floating point scheme, so the significant information in the small tail is lost. The information in the right tail cannot be recovered by complementation. {IR(q = 1-p)} and {IR(p)} are defined as being complements, but as Lewis (2004) pointed out {IF(p ≠ (1-(1-p)))} for all values of p.

Given a cumulative probability distribution that only returns p values for the left tail, then a returned p value of 0.9999999 (for example) would have a {IR} complement of 0.0000001. The following table shows what actually happens in {IF}, given a precision of 15 digits.

Table 15-1: Computer Arithmetic {IF} on Upper Tail P Values

Equation Statement	{IF} Value of Statement	{IF} Value of Complement
Let {p} ==	1.000000000000000E-07	9.999999900000000E-01
{q} == {1} - {p}	9.999999000000000E-01	9.9999999473644E-08
{p'} == {1} - {q}	9.99999999473644E-08	9.999999000000000E-01
{q'} == {1} - {p'}	9.999999000000000E-01	9.9999999473644E-08

The {IF} complements are not the same as the {IR} complements.

Knüsel (2003) says that a basic criteria for any software package that provides statistical distributions is that it contain two entirely different functions for each distribution, one to give p values and the other to give q values. Nether function should depend on complementation for its values. Each function should have an entirely different algorithm. Both the IAS and ELV reference sets (described below) have this feature.

The Excel routines do not include separate p and q tail functions. The difficulty is that in Excel, a typical user, doing a simple test is likely to be confused about which function to use, the p or q function. Most introductory statistics books never provide clear information of which tail goes with which test. It is buried in the tables provided in the text and by historic conventions which get “lost.”

P VALUES LESS THAN 0.01

Related to this, is how does the user interpret small tails? There are four views here.

1. This view is strictly a mathematical view on the distribution as a mathematical function (an {IF} object). The basis is the double floating point representation of a number between 0 and 1.
2. Small tails as floating point numbers are important. The range needed is from 0.5 down to rmin, or 2.23E-308. The issue of false zeros is important, since the user needs to be sure that any reported zero value represents a p value less than rmin..

3. Small tails are not of concern, since tail areas less than $1E-15$ (depending on the user's conventions) are always taken as being zero. False zeros are not of concern, since they are always below this threshold. This is the principle view of business and engineering, where p values below some threshold such as $1E-15$ cannot be related to any tangible effects.

4. Given the approximate nature of statistical distributions as measures of reality, p values of less than "epsilon" (where the value of epsilon depends on the viewer¹), should just be taken as being less than the numerical value of epsilon. The issue is about the occurrence of extreme values in nature which occur, but not often enough or the time scales are too long, to establish valid p values. This issue is current with Nassim Taleb's two books, "Fooled by Randomness" and "The Black Swan".

B.D. McCullough (2007) has pointed out in our discussions on this, that there are some applications in business analysis where the ratios of tails (as small p values) is part of the analysis. On this basis, then it is important that small tail p values (<0.1) be accurately expressed as floating point numbers.

THE PROBABILITY DISPLAY OBJECT

The Excel probability functions return {IF} objects as double floating point numbers. Given a single {IF} probability number, there are four ways in to view the {IF} object in terms of its display.

The **fixed point** (xp) view in which the value consists of decimal numbers (including zeros) in which there are n digits to the right of a set (true) decimal point. The precision here is n decimal digits. The true decimal point is a clear demarcation between whole numbers on the left and fractions on the right. Zeros are inserted to the right of the numbers to fill spaces when the format requires more than n decimals to the right of the decimal point. Numbers may be rounded down to less than n decimal digits, eliminating the showing of right hand numbers (and zeros). Leading zeros to the right of the decimal point are counted as significant figures when the {IF} object is less than one. The display setting sequence is Format > Cells > Number > Number > Decimal Places. LAE is an appropriate measure of accuracy.

The **floating point** (fp) view in which the value consists of m significant numbers left of a relative decimal point, one significant digit left of the decimal point and a trailing exponent designator to identify the number of preceding zeros. The precision again is set by the user, with a maximum of 15 decimal digits. The relative decimal point is an indicator to be used in conjunction with the trailing exponent designator to locate the numbers in relation to the true decimal point. Numbers may be rounded to display less than 14 digits after the relative decimal point. Any leading zeros are not considered significant figures. The setting sequence is Format > Cells >

¹ Knusel views epsilon as 0.0001, Mosteller views epsilon as 0.01. Fisher varied from 0.01 to 0.0001. This is why it is difficult to establish an accuracy baseline.

Number > Scientific > Decimal Places. LRE is an appropriate measure of accuracy.

The **truncated floating point** (tfp) view, which is a mixture of the fixed point and floating point views. It is a floating point number, truncated by a fixed point reference. For double precision numbers, it usually is a truncation to 15 decimal numbers (including zeros) to the right of a true decimal point. . The display setting sequence is Format > Cells > Number > Number > Decimal Places, where “decimal places” has a variable value. LRE is an appropriate measure of accuracy only down to 1E-15 or up to 0.9(13 9’s)9.

For example if a p value is known to be accurate to the leading four non-zero digits, then as the p value decreases toward zero, the accurate leading four digits move to the right (as a group) as preceding zeros (or nines) are added, until the fourth digit reaches a fixed point reference of 15 digits. The “decimal places” value here varies from 4 to 15 depending on the magnitude of the value. From this point downward, the number of accurate digits decreases from 4 to 3, to 2, to 1, then to 0, when all 15 digits (in a fixed point reference) are either all zeros or all nines. When all 15 digits are either all zeros or all nines, the p value is taken as being either a completely accurate (to 15 digits) zero or one.

The accuracy of many Excel p functions can be expressed as a truncated floating point number. This is the view that Microsoft has taken on many of the probability distributions.

The **default display** (dd) view. This is the display when the cell is given the default attributes, as shown in column 3 of table 3-1. This is the display when Excel is started with a new worksheet in the default mode (no background macro, no reformatting macros, etc.). The precision here is specified by the display.

Number: Category: General
Alignment:
 Horizontal: General
 Vertical: Bottom
 Text Direction: Context
 Orientation: Horizontal
 Text Indent: 0
Font:
 Arial, Regular, Size 10
 Normal Font: Box checked
Border: None
Pattern: None
Column Width: 8.43

The default display combines the fixed point and floating point objects described above. Table 12-2 shows the result. The default display is usually very quickly lost when column widths are changed.

Table 15-2: Excel Default Number Display

Input {IF}Object	Default Display
1	1
0.123456789	0.123457
0.0123456789	0.012346
0.00123456789	0.001235
0.000123456789	0.000123
0.0000123456789	1.23E-05

A probability value of 0.000001139 could represent the tail area of a probability distribution. A major concern is, to what accuracy does this number represent. The number remains the same whether it is displayed as 0.000001139 or 1.139E-06, but the precision is clearly different. Under a fixed-point view, there are 9 significant digits. Under a floating point view there are 4 significant digits. Under a truncated floating point view there are 15 significant digits. When the display object is used to indicate accuracy, the accuracy is 15, 9, 6 or 4 digits, depending on how it is displayed.

Under Kahan’s view (Kahan 2004), accuracy is relative to the precision displayed. Consequently there is an inherent ambiguity about any claims about the accuracy of a p value.

Microsoft is ambivalent on this issue of the accuracy shown by the display:

For some statistical distribution functions, Microsoft takes a 15 decimal digit fixed point position (when the function returns are really truncated floating point objects). None of Microsoft’s statistical distribution functions are fully accurate to 15 decimal digits for every allowable input value. As the function p value decreases from one, 15 nines first appear, and then the nines become the other digits until 0.1 is reached. Then leading zeros after the decimal point appear, and the number to the right of the zeros become fewer, until all 15 digits are zero. Everything smaller than 1E-15, is returned as zero. The standard normal distribution is in this category.

For other distributions such as the F distribution, the number is a floating-point number, that varies from 1 to 2.2250738585072E-308. Anything below this is returned as a zero. In the tool-Pak add-in, the p values are shown as floating point numbers.

Because of the difference in view points, statements about the accuracy of a software product and the users interpretation of the developers literature, leads to confecting allegations about the accuracy of “printed numbers”.

DISPLAY TO N ACCURATE DIGITS

Following McCullough (1998) in his seminal paper, he states, “what accuracy is acceptable varies from user to user. An accuracy² of 6 significant digits for a display is acceptable, and for p values, then 2 or 3 digits may be acceptable.” Other authors have different standards.

Both Knüsel (1999, 2003) and McCullough (1998) view accuracy not as an inherent {IF} binary object property, but as a software-controlled {DISP} object reflecting the accuracy of the {IF} object. This view requires that the software be able to interpret the {IF} object as to the number of accurate digits and determine the n decimal digits in the {DISP} object in order to be considered as “error free”. Knüsel (1989) solved this problem by declaring that every p value from his ELV program is accurate to 6 decimal digits (his DISP object), regardless of input parameters, and setting the FORTRAN format lines to reflect this.

Knüsel (1998) says, "In the view of the author a user of Excel may expect that the computed results are correct with all given digits; if Excel displays a result³ with nine or ten digits and in fact only one or two of those digits are correct, then such a result is unacceptable."

McCullough's (2004) position is based on the default display. If this display is not accurate, then this is a fault of Excel since it misleads the user into believing a higher level of accuracy in a p value than it inherently has. Excel will allow one to see all 15 digits of a non-zero number from a calculation even though the calculation was from a very crude or inaccurate algorithm.

Knüsel's (1999) view, was to report tail p values down to a small limit. He uses the term “rmin” for this lower limit. Returning tail areas as a floating point number also improves the confidence in the user that the function is giving correct values. Reporting that $p < 1E-16$ does not convey the confidence in the calculation that $p = 3.5684E-289$ does.

In a later publication, Knüsel (2003) indicated a fixed point view for all p values, considering p values less than 0.00005 as zero and p values larger than 0.99995 as one. By selecting: Format → Cells → Number → Number → Decimal Places → 4 → OK, the {DISP} object will meet his standard.

In McCullough (2003), when Excel returns a zero value instead of a small tail value, he considers this as an Excel fault. However, the returned zeros for the probability values comes from Microsoft's Truncated Floating Point view, which in this case are correctly shown as zeros. In this particular case, the returned values would be correct under Knüsel's 2003 view.

² Note that the word “accuracy” used by McCullough and by Knüsel mixes Kahan's “precision” and “accuracy” concepts, so that it is difficult to “infer” just exactly what they mean when they use the work “accuracy” with respect to the display “precision”.

³ As discussed earlier, the display is controlled or set by the user, not controlled by the Excel function. Other commercial software programs, give outputs with a fixed precision.

ACCURACY NEEDED FOR A DECISION:

The number of significant digits evolved from the history of statistics and from the general usage of p values.

The tables of the normal distribution, that Fisher used (Fisher 1973) dates to about 1925. The normal table he used has 2 significant digits for p values, but 6 decimal places for z values (p has a precision of 2 and z has a precision of 6). Around a z value of 2, a change in z from 1.959964 to 1.959963 changes the p value by $-6E-08$, implying that the equivalent accuracy of the p values were viewed as 8 significant decimal places rounded to 2 or 3 decimal places for use in tables. The mechanical calculators used at the time to do the calculations of the series, automatically entered any trailing zeros if no digits were entered.

Another acceptable accuracy view is in the use of published statistical distribution tables. For example, the standard normal table with z values to a precision of 3 digits and p values with a precision of 4 digits is frequently found. (see almost any introductory statistics book). Given a z value of 2.12, the corresponding table value is 0.9830, which is accurate to 4 figures. It is entirely acceptable. There is no need to report a p value of 0.982996977352367. No one, it seems, criticizes these tables and the algorithms that produced them for accuracy. If they were generated using fitted polynomials (derived from the tables and good to only 4+ figures) would still be considered an acceptable algorithm.

Although 2 to 3 significant digits may only be needed to make a decision, there are three considerations here:

One, is that the computed value be accurate to k digits, where k is larger than 2. The purpose is to see that the effect of rounding does not result in a false decision.

Two, is that the value be reported to the limit of the computing capability. This is needed when software packages or routines are compared, (benchmarking). The approach is to compare printed/displayed numbers from the package under consideration with reference to a standard value. The “seller/developer” of the package is concerned that the number is “close” to the standard.

Three, in current practices with fitting models to data, or interpreting the results of data in terms of a hypothesis, the magical p value used is 0.05. For a paper to be published, a p value of 0.501 would be considered by the editors as not meeting the 0.05 criteria, consequently there exists, a p region around 0.05 (or 0.95) where many digits have to be displayed. Accuracy of the p value does not appear to be a consideration, just the magnitude (See Abelson 1995).

In testing software, the “accuracy” relates to a “correctness” of the “algorithm” and “instruction set” under the assumption that any “errors” are strictly due to “rounding” errors. The problem then becomes one of trying to determine faults in the algorithm. This is important where one is testing “black boxes” where the code is not visible and cannot be examined. (Disclosing the algorithm is not sufficient, since the coding (and the coding in called subroutines or functions) may introduce errors, visible only under extended testing. This was the problem Microsoft had with the 2003 RAND function.)

METHODS OF LIMITING THE P VALUE TO ONLY THE NUMBER OF ACCURATE DIGITS

To be able to manage the display (the precision) of only accurate digits in Excel, the method used depends on whether the accuracy statement is on a fixed point, floating point or truncated floating point number.

1. ACCURACY ON A FIXED POINT BASIS:

Cell Format: Use of the cell format to restrict the number of digits displayed, in the number format.

Round Function: Use of the Round Function: If the user knows what the accuracy is in terms of the number of digits, he can run the outputs of the p functions through the ROUND function shell. The distribution function can be nested within the ROUND function parenthesis, such as “=ROUND(CHIDIST(A4,B4),3)”. This permanently changes the 15 digits, whenever the calculation is done. In this example, the reduction is down to 3 significant digits (to the right of the true decimal point).

Use of the SelectionNumberFormat function. For example:

```
Function NewNormalProb(z As Double) As Double
    Dim p1 As Double
    Dim s1 as string
    p1 = WorksheetFunction.NormSDist(z)
    s1 = Format(p1, "0.000000")
    NewNormalProb = CDBL(s1)
End Function
```

Returns a number to the cell, which is then displayed according to the cell display format. In this case only 6 digits would be shown, values less than 0.000001 would be zero, and values above 0.999999 would be 1

By converting the formatted string to double, allows the cell contents to be used in subsequent calculations

2. ACCURACY ON A FLOATING POINT BASIS

Cell Format: Use of the cell format to restrict the number of digits displayed, in the scientific format.

Round Function: The ROUND function may be used. In this case the rounding number MUST be the sum of all the preceding zeros plus the number of digits to be rounded. Not practical for small p values.

Use of the SelectionNumberFormat function. For example:

```
Function NewNormalProb(z As Double) As Double
    Dim p1 As Double
    Dim s1 as String
    p1 = WorksheetFunction.NormSDist(z)
    s1 = SelectionNumberFormat(p1,
    "0.000000E+000")
    NewNormalProb = CDBL(s1)
```

End Function

Returns a number to the cell, which is then displayed according to the cell display format. In this case only 6 digits would be shown, small values can be displayed in floating point down to zero, but values >0.999999 would be returned as 1.

z	s1
-1.2	1.150697E-001
-2.5	6.209665E-003
-3.6	1.591086E-004
-4.1	2.065751E-005
-5.8	3.315746E-009

3. ACCURACY ON A TRUNCATED FLOATING POINT BASIS

User Function: Requires a special function, to round and convert least digits to zero, as the right trailing digits are truncated at the 15th digit position. No matter how the number is viewed (fixed or floating point) the number appears as Knüsel (2003) and McCullough (2003) want it to appear.

To round internally within the function, would require some internal structure that would identify the level of accuracy, based on logic on the input parameter values. For a one-parameter distribution such as the normal, this would be easy. For a three-parameter distribution like the beta, the estimating of an accuracy value may not be possible. It would be a disservice to set everything at 3 digits when actually it was better than 3.

Once the accuracy of the calculated p value can be estimated from some internal function logic, then the output (as a string variable) can be set. Note K describes how this can be done.

The other commercial statistical software packages may limit the reported number to only 5 digits. This creates problems when benchmark tests are done. This appeared to be the problem with Altman's (2002) analysis of JMP, according to the company (Sall 2002). This also limits the assessment of the accuracy of Knüsel's (1989) ELV package, because he set in FORTRAN formats to F9.6 on all the outputs.

Megastat, an Excell Add-in, reports p values in the form of 0.0000 and tail values below 0.01 in the scientific format of 4 digits. Megastat uses the SelectionNumberFormat method using internal logic to switch the display from fixed point to floating point. The output here is a string, rather than a number.

Some of the critics of Excel on this issue, want all p values less than 0.0001 to be reported as " <0.0001 " rather than "0.0000" to show that a zero probability was not obtained. This can easily be done inside a function, but the output has to always be a string, since " <0.0001 " is a string, not a number.

Note that all these conversions destroy any use of the function in simulations where the output must always be a floating point number.

LIMITATIONS ON DEFINING ACCURACY BY NUMBER OF DIGITS

Statistical distributions involve series summations, as {IF} interpretations of {IR} series. Changes in input parameter values in {IF} will result in different values of a sum at the point of {IF}-convergence. Numerical differentiation of the function will show a large “noise”, representing the limitation in {IF} to approximate {IR}-convergent values. (Like any measurement, it is variable.). It is this “noise” that creates “a randomness” in the returned p value.

In applications, extreme values of parameter space are likely to be entered as input. In most cases, these values and in different combinations will cause a well-behaved algorithm to return {IF}-convergent output values well below an accuracy of n digits, without the user being aware that this occurred. The proper way to deal with this is to make a statement like “the value from the function is accurate to n digits, P% of the time, as long as (a statement about the limits of parameter space) is met”. This statement fits reality, is in accord with accepted statistical concepts and requires a defined boundary on input parameter space.

Statistical functions do not behave uniformly within the {IF} object. They tend to have lower accuracy when the number of series terms is large (central region) and higher accuracies when the terms rapidly converge (small tail areas). The functions will also show odd behavior when different algorithms are combined to achieve complete coverage over a specified parameter space. The switching as to which algorithm is used may be by some complicated (unknown) test on input parameter values. As a result, a generalization as to n accurate digits over the entire parameter space requires the acceptance of a least value (i.e. 2 digits for a general 13 digit accuracy), which is of no help to a user.

THE ESSENTIAL P VALUE ACCURACY PROBLEM

First of all, relying on a default {DISP} p value, is a very poor position to base p value accuracies on, given that the user can easily change the display formats, and easily lose any relationship to a “standard default” display.

Second, Microsoft’s inconsistency between fixed point, floating point and truncated floating point representations of p values, prevents any generalization as to the accuracy of output p values (and their inverses).

Thirdly, the literature references on software p value outputs represent different views:

McCullough’s (1998) standard for p values is 6 significant digits (floating point). However McCullough (2004) will accept 3 accurate digits for statistical decisions, since that is an accepted practice.

Knüsel (1989) in his ELV package only reports 6 digits (floating point) signifying his view on the accuracy of the ELV values. In his papers on the testing of different statistical programs, he reports p values from 5 to 7 (floating point).

Knüsel (2003) proposed a standard of 4 digits either in fixed point or in floating point, depending on the software package capabilities. A p value less than

0.00005 is to be returned as zero. A p value greater than 0.99995 is to be returned as 1.

Altman's (2002) standard is 5 for the central region of 0.01 to 0.99 and 2 outside (fixed point). Altman does not consider the rounding issue, McCullough (1998) does.

In McCabe and Moore (2003), two significant p digits (fixed point) is fine, when a decision is to be made. The values in statistic book tables are all 2 to 3 digit p values with assumed trailing zeros.

There is no agreed on common position from the "experts". There is no agreed on way to handle small tails.

Fourthly, the total freedom by the user to change the {DISP} object to any thing he wants, removes any restrictions built in to the software. The only barrier to this is to build into the functions the SelectionNumberFormat method shown in note K.

Fifthly, there is the nines or complement problem when dealing with p values larger than 0.5. The standard interpretation that leading zeros are insignificant digits and the values can be handled as floating point numbers, but leading 9's are significant, creates its own set of accuracy problems. Microsoft does not provide complementary p value functions, resulting in the computation of complements as one minus the p value, which is inherently inaccurate.

Sixth, by controlling the {DISP} object, exact matches to reference values may be attained, yet the un-displayed digits may be totally in error. This illustrates how the {DISP} object can drive conclusions on the accuracy of the {IF} object. A true test of accuracy is one on how close the {IF} object comes to the true {IR} object, independent of how it is displayed.

ACCURACY OF INVERSE STATISTIC VALUES

A DECISION BASED ON A STATISTIC

In this case, an input p value referred to as an "alpha" value is used as the input to the inverse of a probability distribution to obtain a test statistic. The alpha value is commonly given by two or three significant digits, with the assumption that trailing zeros are automatically appended to precisely give 15 significant digits. This is an implied fixed-point view.

The traditional practice has been to obtain a test statistic value with as many significant figures as given in standard tables. The number of digits usually varies anywhere from 3 up to 6, depending on which tables are used. For computed inverse distributions then, they should be able to return a test statistic that has at least 5-6 accurate figures.

Knüsel, McCullough and others do not address the accuracy of a test statistic.

GENERATION OF RANDOM VARIATES:

This is the area of simulation, Monte Carlo methods, and choosing random subsets of population values for sampling, for selections and likely outcomes from inverse discrete functions.

For simulation using Excel, a general requirement would be that 6 or more digits of numerical values always be returned (no #NUM!s), and that they come from a “smooth”, robust function. The differences between the p value input and the inverse function output value with reference to a true output value can be large. The fast approximations currently used in simulations are acceptable. The range of p values that the function needs to respond correctly to is from 0.999999999 to 0.000000001. With current random number generators there is an extremely small probability that the random number generator will output a value outside of this range. A crude approximation can be used to give a return if a random number outside of this range occurs.

GENERAL METHODS RELATED TO TESTING

All testing was done using Excel spreadsheets.

SELECTION OF INPUT PARAMETER VALUES

RANDOM VALUES IN PARAMETER SPACE C

Random number generation of function input parameters was extensively used. This is a systematic approach to testing, and avoids the arbitrary shotgun approach used in the literature. For the results, one can also estimate probabilities of occurrence of zeros, false zeros, “NUM! returns, etc.

The random number generator used was Marsaglia’s Multiply With Carry, MWC256. The function was converted from C++ to VBA and put into module 8 as MWC256(x). The value x had no effect on the random number generated, it only served to trigger off the generation of a new number when the calculate operation detected a new input value. This resulted in the choice of constantly changed values when x was RAND() or a quasi stable value when a number or cell reference was put in. With a zero as input, the cell random value would remain the same under “F9” or automatic recalculations, but would change when cells were copied down.

MWC256 was Marsaglia’s latest (2003), and was reported by him to pass all of the random number generator tests. The choice here was to avoid any problems known to be in both the Excel 2000 and Excel 2003 versions of RAND.

For the discrete inverse functions, the sequence, input parameters > direct function > p value > inverse function > output parameter, should result with the same (integer) value output as used in the input.

For the continuous inverse functions, the sequence p value > inverse function > parameters > direct function > p value was sometimes used, and testing of the difference in p values reported. It penalizes the pair of functions, making specific accuracy statements on the separate functions difficult.

SHOTGUN METHOD

This is the method used in the literature. It is a trial method, banging away at different parameter value sets until a fault from the function being tested occurs. Parameter space is not defined or systematically explored. McCullough reported that the decision to use this method was based on the fact that any systematic method would take too long. It was also forced by the publication policy of only allowing a maximum of 5 pages.

The reported tables (of a short list of values) give no clues, as how to interpret 3 wrong values with 3 correct values, in relation to “how does the user avoid wrong values?” Sometimes, just a narrative is reported such as in Knüsel (1998a), “The computations seem to be correct even for very small probabilities (relative error smaller than 1E-6 for probabilities as small as 1E-101 and even smaller”. On the following pages of this reference, he just gives some tables with parameter values that give incorrect output values.

The shotgun method is in many cases the only way to test statistical software. The software may only allow keyboard entry of distribution function values into a selected distribution. This is a very slow, tedious method. In essence, the developer has very effectively blocked adequate testing (finding defects) of his software.

Excel fortunately does not have this block, and 5000 input parameter sets and an analysis of the returns can be in seconds.

A shotgun approach may also be the only way to generate some test cases for a paper that covers the entire Excel suite of statistical distributions in only the 5 pages required for publication.

ACCURATE REFERENCE P VALUES

STANDARD TABLES OF VALUES:

For testing accuracy, there are no commonly available standardized values other than the usual printed tables such as that in Abramowitz (1964). Most of these tables do not have enough accurate significant figures to provide a standard. They do however provide values for testing the coarse fit of distribution functions, define which tail is used as the standard p value, and resolve issues on one or two tail p values and conventions.

SPECIAL REFERENCE FUNCTIONS AND SUBROUTINES IN VBA WITH PRECISE VALUES:

Most of the references dealing with distribution accuracies in Excel use some external program or subroutine to calculate a value that represents a more accurate value. Knüsel (1989) used his ELV program sets. Cook (CISE 27/99) used the IMSL package.

Knüsel (1989) made his ELV program available for use as a reference function. However it works only in a DOS environment, and can't be used to generate values on an Excel spreadsheet. McCullough (2004a) reports it is very awkward and slow to get values from ELV. Also the 6 figure limit blocks it from adequately testing the more accurate Excel functions such as BINOMDIST. Smith (2007) provides a list of errors and faults with the ELV program and the values it gives. Comparisons were limited here to sources that gave ELV values.

I took a different approach, initially writing special subroutines and functions in Excel's VBA package to calculate precise values. These were carefully programmed and tested to ensure that the resulting values were more accurate than Excel's own internal functions. These routines use the infinite series in Abramowitz (1964), and were modified to give accurate values to 12 or more significant figures (except for some regions of the Beta distribution).

Subsequent to this, Ian Smith (Smith 2003) released a set of VBA functions (IAS) in 2003 that had some unusual algorithms. Lewis (2004) tested Smith's (Smith 2003) reference functions using externally generated value sets (Use of Maple 7 on a Unix system, programmed with the basic cumulative distribution equations and generating floating point values with 50 significant figures.). His findings were that Smith's routines had LRE accuracies exceeding 13.5, over the range down to 1E-307. Lewis did not put his test results into a published form. Smith's routines have some unusual empirical approaches to attaining accurate values for both the p and q tails.

The version used for all distribution testing was the 1.0.24 version (2002-2003). The current version is 3.2.6 (as of February 2007). Ian Smith responded to a question about the issue of differences between the two versions as follows.

“Off the top of my head, I would say the most significant changes were adding the non-central distributions and re-parameter-ising the hypergeometric distribution to make it possible to add accurate functions for the Beta-Binomial and the Beta-NegativeBinomial.

I also improved the binApprox function which provides an asymptotic approximation for the Beta distribution for large shape parameters. I could only calculate the first 7 terms in the series manually and one of these was fudged to make the function work more accurately in the range required. Since then I have used programs with multi-precision integers to calculate the first 16 terms of the series.

There have been other changes, quite a few of which Jerry W. Lewis asked for, but I could not give a complete and accurate list quickly.

I keep a spreadsheet with over a thousand calculations which I use to check modifications haven't made the program obviously worse. I also have a program which calculates the answers more accurately using the full 80-bit precision available on my PC. This makes sure the expected answers in the checking spreadsheet are trustworthy.

I also use routines which test convergence by running the inverse routines over a wide range of parameters. The main purpose of these routines is to see if they finish.” (Smith 2007).

It is by conjecture that all of Smith's functions (in version 1.0.24) produced accurate values as a basis to test Excel provided distribution functions. I have not found any reports, comments or messages on the nets, news groups or lists regarding inaccuracies in his algorithms.

I did some preliminary testing that indicates that Smith's set was more accurate than my reference set. Smith's routines also gave accurate benchmark outputs. Also from Knüsel's published test papers, his ELV values agree with Smith's reference function values with LRE values from 4.5 to 7. The number of figures printed limits the low LRE values in some cases.

Not all of the Excel functions have comparable reference functions. The following tables give a comparison of ELV and IAS functions.

E represents Excel functions,
S represents Smith's functions and
K represents Knüsel's ELV functions.

Table 15-3: Discrete Probability Test Distributions Provided

Distribution	CDF	Comp CDF	Crit	Comp Crit	PDF	LCB	UCB
Binomial	EKS	KS	EKS	KS	ES	S	S
Gamma-Poisson	S	S	S	S	S		
Geometric	S	S	S	S	S	S	S
Hyper-geometric	KS	KS	KS	KS	ES	S	S
Negative Binomial	S	S	S	S	ES	S	S
Negative Hyper-geometric	S	S	S	S	S	S	S
Poisson	EKS	KS	KS	KS	ES	S	S

Explanations:

Cdf is the cumulative (integral) function

Pdf is the density function

Comp is the complement of the function

Crit is the inverse of the discrete cumulative function

LCB is the lower confidence boundary

UCB is the upper confidence boundary

Table 15-4: Continuous Probability Test Distributions Provided

Distribution	CDF	Comp CDF	Inv	Comp Inv	PDF
Beta	EKS	KS	EKS	KS	
Non-central Beta	KS	KS	KS	KS	
Chi-Square	EKS	KS	EKS	KS	
Non-central Chi-Square	KS	KS	KS	KS	
Error Function	E				E
Exponential	ES	S	S	S	E
F	EKS	KS	EKS	KS	
Non-central F	KS	KS	KS	KS	
Gamma	EKS	KS	EKS	KS	E
Non-central Gamma	K	K			
Log Normal	E		E		
Normal	EKS	K	EKS	K	E
T	EKS	K	EKS		
Non-central t	KS	KS	S	S	
Weibull	E				E

Explanations:

Cdf is the cumulative (integral) function

Pdf is the density function

Comp is the complement of the function

Inv is the inverse of the cumulative function

Consequently Smith's routines were primarily used to test the Excel routines. The LRE values shown in the charts below from random number inputs were all derived from Smith's reference routines when they were in his set. My reference routines were used when Smith did not include one. The differences due to the use of Smith's routines had almost no noticeable change in LRE values from my reference functions, since Excel's routines are generally below the accuracy range of both sets of reference functions. The exception is with the discrete probability functions, where the Excel functions have high accuracy.

Related to the problem of getting accurate test values, is a "sensitivity" problem. This is where small changes in parameter values cause large changes in the output value. There are situations where the parameter is expressed as more than 15 digits, and where differences in the 17th and 18th digit can be made, changes in the reference output in the 9th position are observed. This is just one of the problems in trying to do {IR} arithmetic in an {IF-754} environment

THEREFORE IT IS IMPORTANT THAT THE READER BE AWARE THAT SPECIFIC EXCEL ACCURACY ERRORS FROM TESTS MAY BE REFERENCE FUNCTION ERRORS AND/OR COMPUTER OBJECT ERRORS IN THE IEEE-754 ENVIRONMENT AND NOT TRUE EXCEL ERRORS.

TEST MEASURES

These are defined in Section 4.

COMPARISONS BETWEEN LRE AND LAE MEASURES:

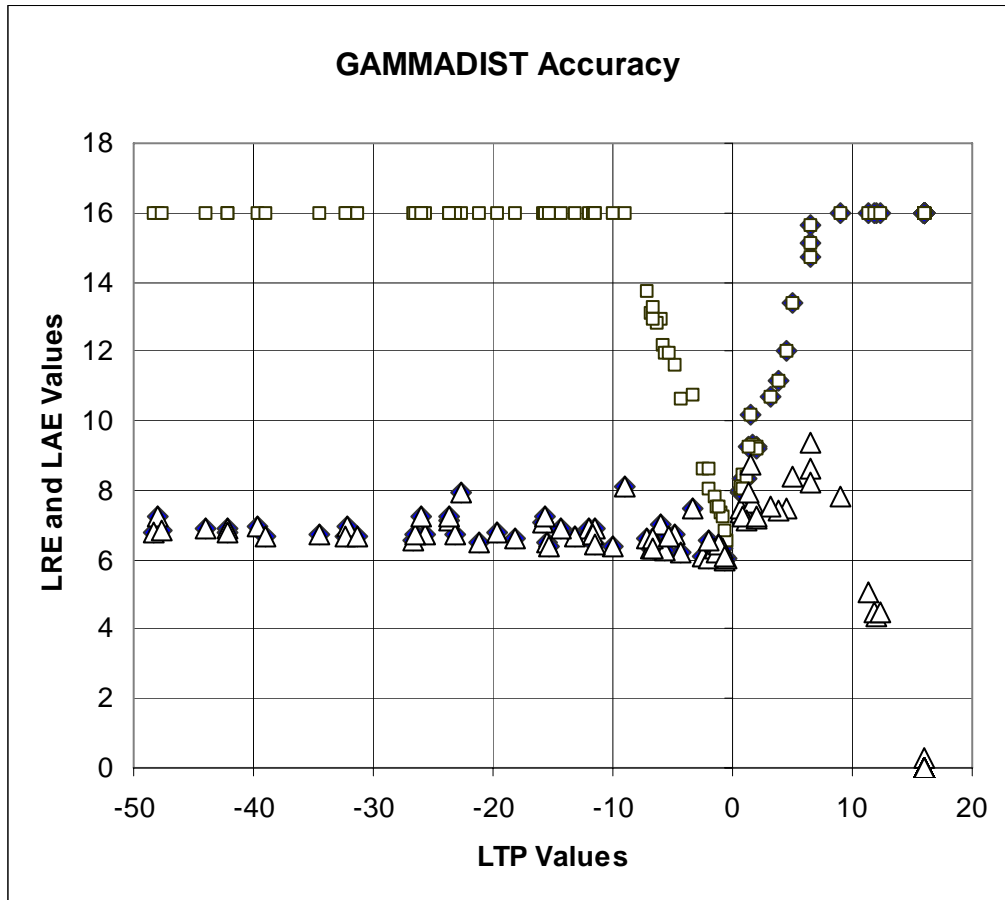
When errors are proportional to the value of the number returned, LRE values generally lie in a horizontal band when plotted versus LTP values. For this type of error, the LAE values will rise on the left side from the zero LTP axis. If errors are not relative, and essentially fixed over a wide range of function return values, the LAE plots will tend to be horizontal, since they are estimates of the fixed range of errors.

The figures below show how LRE and LAE values generally behave with respect to LTP values and certain function characteristics.

LRE/LAE BASED TEST METHODS

The tests based on LRE/LAE measurements are shown in the following figure. It shows test results from the three LRE/LAE methods on GAMMADIST. Note that the LTP = 0 line is for p=0.5.

Figure 15-1: Three LRE/LAE Test Methods on GAMMADIST Accuracy



Method 1: The open squares are points from the LAE method (McCullough (1998) uses the symbol LAR for these values). The LAE method is looking at the leading fixed-point decimals, and measures the differences between the test and reference function p values. In this case the LAE values for p values around 0.5 are about 7, and rise on each side of 0.5 to 16. 16 represents exact equality. The 16 value comes directly from the leading zeros (p less than 0.5) and leading nines (p values greater than 0.5). These leading constant digits dominate, and lead to zero differences when the p values get small enough (or large enough). This is the way Microsoft assess the Excel statistical functions.

Method 2: The diamonds represent the LRE method given as equation 2 in McCullough (1998). I use the exact same method for LRE values. For p values less than 0.5, they are a good measure of accuracy that fits the floating-point viewpoint. Above 0.5, the preceding nines dominate, reducing differences, without weighting the magnitude of these differences. Above 0.5, LAE and LRE measurements are essentially identical.

Method 3: The triangles represent a third method that represents Knusel's (2003) concern about the fact that complementing p for a q value gives a very inaccurate q value. Method 3 is a variation of the LRE method.

- a. For this test, the complement is one minus the p value, and the complement is tested. However the BETADIST, BINOMDIST, EXPONDIST, HYPGEOMDIST, NEGBINOMDIST and NORMSDIST distributions can

- provide a complement directly, by inverting or changing the inputs as shown in table 10-3. If the user chooses to use this method for obtaining p values > 0.5, then the results of the method 3 tests should be viewed differently. In this special case of complementing, the error charts should be viewed as being symmetrical about the LTP = 0 line, where the points above (positive LTP values) are to be viewed not as shown, but as a mirror image of the values to the left of the zero LTP line.
- b. The third method penalizes the Excel distributions for the loss in information when complementation is used to obtain the other tail area. The points in the region above p values of 0.5 (positive LTP values) are approximately bounded by the method 3 lines in figure 15-1.
 - c. The region above p=0.5 is, in-common-usage-of-the-function, infrequently found. It takes extreme values of the parameters to obtain function p values that correspond to LTP values greater than +4. Error returns are common in the area above LTP values of +5. Consequently it is difficult to fully populate the region of LTP values from +5 to +16 to show accuracies for p values close to 1.

Where the test method is not explicitly identified, the general practice has been to use Method 2 for all Excel 2000 function tests and for Excel 2003 function tests on the inverse functions. Method 3 is used on all Excel 2003 direct functions when both p and q reference functions are available. If only one reference function is available (such as the entire inverse function suite), method 2 is used. Method 1 is used to show accuracy as Microsoft views it.

ACCURATE DIGITS METHOD

Test method 4 is different, in that it deals with the accuracy-of-n-displayed-digits issue. The test is to round the Excel function to n significant digits, and to round the reference function output to n digits. When the two values are equal, then the n digits are accurate. If they are not equal, then the n digits of the Excel function are not accurate. This method was built into the testing procedure. Both Knüsel (1997 and 2000) and McCullough (2003) do not provide any guidance on how to do accurate-digits-method testing.

The other issue is, what constitutes acceptance of the statement, “this number is accurate to 4 digits”, which is as far as Knüsel and McCullough go. The accurate digits method test generates M lines where each line is a unique set of parameter values, Excel function output values and reference function output values. The parameter values are randomly generated within parameter space C. Now, how many lines with rounded values that don’t match is acceptable out of M lines? If one takes Knüsel’s or McCullough’s position that even one is unacceptable, then one can play the arbitrary M and C game to arrive at whatever conclusion that the tester wanted to prove.

Measurements on several functions (in section 14) suggest that given a rounding to N digits, the probability of a function, calculating values in parameter space C, returning a number that is not accurate to N decimal digits, can be measured. Instead of “this number is accurate to 4 digits”, the correct statement is that “this number is accurate to N digits with a probability of XX%”.

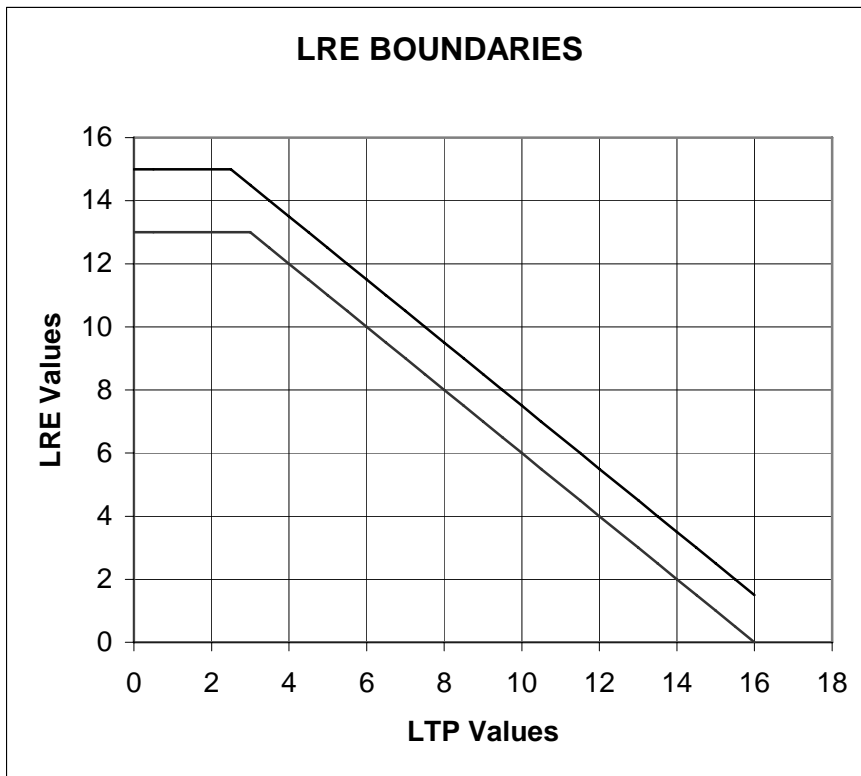
Recommendations for the number of digits to round to at the end of the tests on each distribution function are given in a recommendation block. The number given is an estimate based on rounding test outcomes and LRE values from testing

LIMITATIONS ON THE ACCURACIES OF COMPLEMENTATION

The nature of double precision numbers and the fact that p values above a p value of 0.5 are often obtained by complementation, introduces errors that cannot be measured. With complementation, the leading digits are all nines, and are not treated as “non-digits”. There is no shifting in the floating-point scheme, so the significant information to the right of the nines is lost.

This loss is shown in figure 15-2 in terms of LRE and LTP values.

Figure 15-2: Theoretical LRE Value Boundaries For p Values >0.5



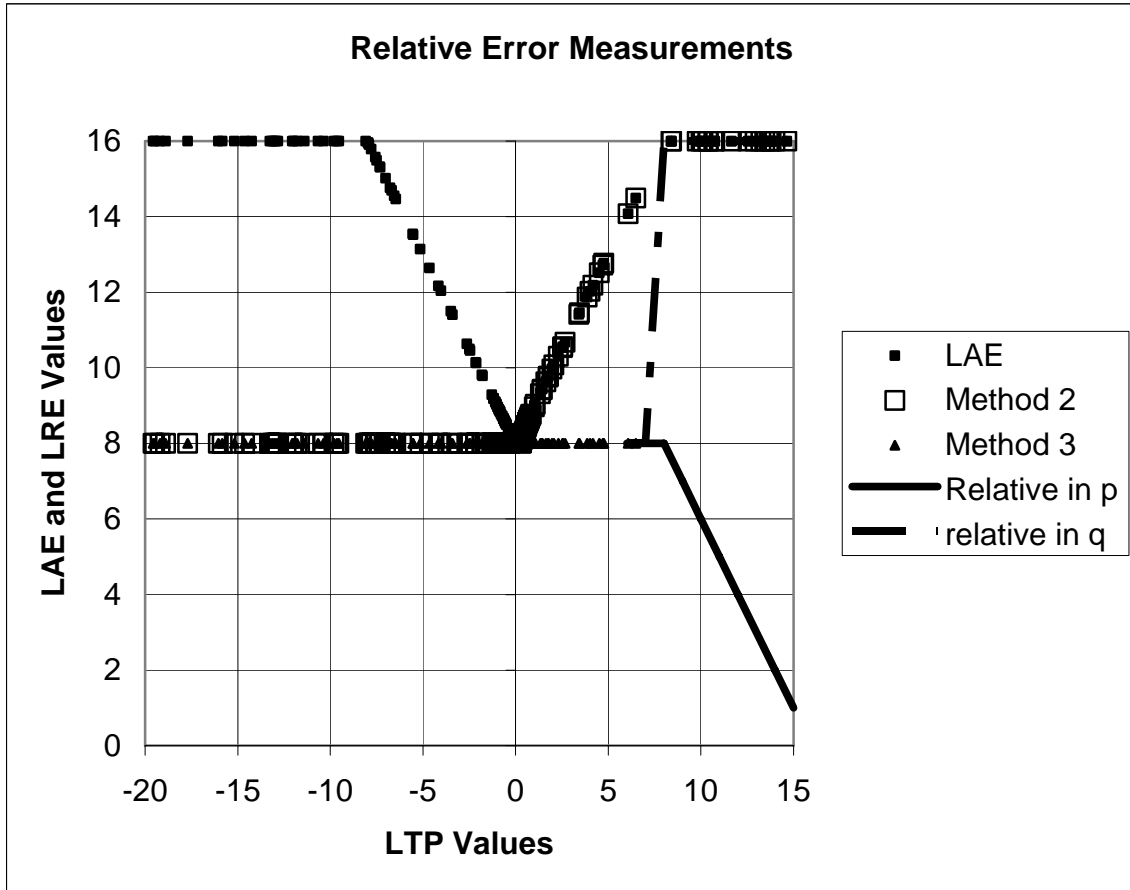
The upper line is a characteristic limit line from complementation. It represents the case where precise 15 digit p values are calculated only for p values from rmin to 0.5 and because of distribution symmetry, can be used to obtain p values above 0.5. This strictly is due to the nines generated by subtraction from one. It reduces the number of real significant values.

The lower line is more typical of reference functions, where the error between the reference function and accurate benchmark values are plotted. The inherent accuracy problems of doing computations in double precision over wide ranges of function input values, essentially limits reference function accuracies (using double precision) to lie between the upper and lower lines.

INTERPRETATION OF TEST RESULTS

In most cases, the charts in section 14 are shown of the distribution errors as LRE values versus LTP values. With the method of testing based on random values, the charts show scattered points, and it may be difficult to draw conclusions from them. Figure 15-3 shows some points from a theoretical distribution in which the error is proportional to the tail (small) area. Here the error is $1E-06$ times the true value and is added to the true value to give a function value.

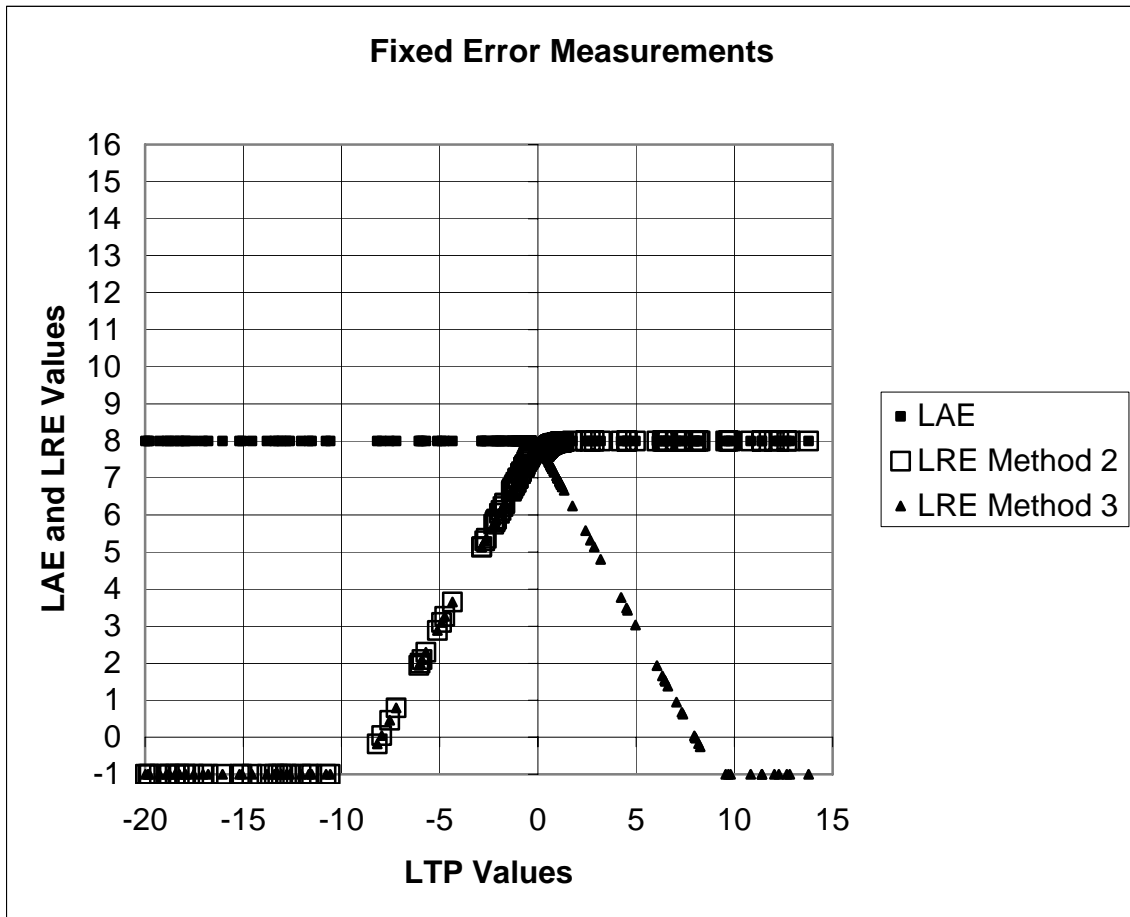
Figure 15-3: Theoretical Relative Errors For The Different Test Methods



Method 1 test results are the small solid squares. They form the v shape, symmetrical about the zero LTP axis. Results from method 2 and method 3 to the left of the axis are identical, and lie on a horizontal line out to $-300+$. On the right of the zero axis, method 2 points are identical with method 1 points. Method 3 points continue to be horizontal out to some point and then decline. If the relative error is on p, then the points decline as shown in figure 15-3. If the error is on q, then the points abruptly stop. With actual data, the points jump to either the 0 line or to the 16 line. The LRE 16 line (actually it is 15.7 for decimal digits, and is 16 for binary bits) is a perfect equality for two double precision numbers.

If the error is absolute, or defined as error digits, or if there are zero digits after n digits, then the data looks like that in figure 15-4.

Figure 15-4: Theoretical Absolute Errors



In general the p region from 0.2 to 0.3 represents a region where the p values seem to have the lowest LRE values. There also will be many points in the p region from 0.3 to 0.1 that are higher than the general band, up to LRE values of 16. The scatter band seems to be the widest in this region. However not all distributions show this effect.

Some charts in section 14 show a few scattered points well below the general band of points. When this occurs, it is very difficult to set accuracy limits in terms of significant digits. Since these points in all cases are rare (<0.1% occurrence), they are ignored, and the bottom of the general band used to set a recommended number-of-digits value.

RELIABILITY OF PROBABILITY FUNCTION OUTPUTS

Reliability is a general area that focuses on the ability of the function to perform. Reliability problems frequently occur in testing software.

FALSE ZEROS

The false zero is an incorrect zero return from the distribution function. This is when the function returns a zero when the true value is a small number. McCullough (2003) shows that this is a major problem with Excel 2003. The false zero return comes from several situations:

1. An {IF} association problem, where one of the terms in the algorithm is zero (value below rmin) or below some internal check, and when {IR} multiplied by larger terms results in a zero. Here the {IR} value is larger than rmin, but the {IF} process results in a zero.
2. A limit on the number of series terms that are to be summed exceeds an internal limit, and the algorithm was coded to return a zero in this case.
3. An internal input-parameter-range-testing-algorithm concluded that the p value was small and a zero was returned, when actually with the given parameter values, a {IR} calculation would result in a p value greater than rmin. This comes from the inability to fully predetermine the outcome of the full {IF} calculation, based on if-then-else-end-if structures on parameter values.
4. The algorithm is coded to return all p values below some threshold (i.e. 1E-15) as zero.

BINOMDIST is a good example of the occurrence of false zeros. Given the number of trials less than 1030, one can obtain accurate small tail values down to 1E-286. The occurrence of false zeros here is about 3.3%. When the number of trials is above 1030, there are no small tail areas, anything less than 1E-15 is returned as zero. To the user there is a sudden jump in what appears to be false zeros, when he is looking for small tail areas.

False zeros can be handled as an accuracy issue or as a reliability issue. McCullough (2003) treats it as an accuracy issue, as a penalty on not getting an exact value. Treated as an accuracy issue creates a greater problem in trying to fit accuracy to the number of digits displayed. Zeros don't have any digits. Knüsel (1999) defines a zero for a fixed-point reference as any value below 0.00005 and for a floating-point reference, any value below rmin. The first has a specified number of digits, and the second is a numerical limit for the number of specified digits above rmin. This example shows the conflicts in logic that occur when both a floating point and a fixed-point view are held.

NON-NUMERIC ERROR MESSAGE RETURNS

These are described in section 3.

The occurrence of these error messages may or may not be a frequent problem to a user, but it just prevents the user from obtaining a desired value. Frequent #NUM! occurrences when the inputs are valid, indicates poor reliability. #VALUE errors can be corrected by fixing inputs, but #NUM! Errors usually can't.

Although reliability can be measured as the ratio of #NUM!'s encountered to the total number of tests. The number is entirely dependent on the selected range of parameter values used in testing. With random number testing, a typical reliability is about 70%. However for a user, who may rarely get into the #NUM! range, his experienced reliability may exceed 99%. Consequently it does not seem to be realistic to report reliability values, unless there is a standard regarding ranges of input parameter values.

The tests in section 14 on each of the distributions, has indicated that #NUM! is a frequent output.

GROSS ERRORS

This is where the function does not properly terminate and a number comes out that is so far from the expected number, that the fact that it is wrong is obvious. For example this would be when an F distribution function returns a p value of -32.8934. The error is so obvious; it would never be taken as an accurate value.

NON-ENDING INTERNAL LOOPS OR LOGIC TRAPS

This is where the function is called in a cell and nothing happens. There is no indication that an internal exception, test or continuous loop had occurred. In Excel, pushing the ESC key will stop the computing sequence. If the function was a user Macro or VBA routine that was the source, the resulting message box will allow the user to locate the problem. The cause may sometimes be explained in Help or on one of Microsoft's KBAs.

Microsoft has carefully protected the internal routines of Excel, so that one can't locate where these kinds of problems occur in Microsoft software. This clearly defines the allowable parameter space when parameter combinations cause non-ending loops, or low level exceptions (occurs down on the bottom in some visibly inaccessible routine) that have no higher level error handling routines (the programmer didn't think that this could occur).

SYSTEM HALTS, CRASHES AND UNRESPONSIVE BEHAVIOR

This is an obvious occurrence. The newer programming structures in Excel 2003 and Windows XP have corrected a lot of these problems.