

A robust function to return the cumulative density of non-central F distributions in Microsoft Office Excel

*James Byron Nelson**

University of the Basque Country (UPV/EHU)

The manuscript presents a Visual Basic® for Applications function that operates within Microsoft Office Excel® to return the area below the curve for a given F within a specified non-central F distribution. The function will be of use to Excel users without programming experience wherever a non-central F distribution is required, such as conducting power analyses for analysis of variance designs and constructing confidence intervals on effect sizes. Tests show the function to produce results comparable to those obtained with the commercial software SPSS and the popular free R environment for statistical computing. Spreadsheets for use in Excel and OpenOffice Calc® are included with example usages.

This paper presents a Visual Basic for Applications function that can be used in Microsoft Office Excel (hereafter referred to as Excel) to compute the cumulative distribution frequency function (CDF) for non-central F distributions. This paper will focus on the use of the function in Excel, though it will also function in OpenOffice Calc with minor modifications mentioned in Appendix B. The function is most often used in determining power, as described by most graduate textbook in statistics

* *Acknowledgments:* The work presented here were made possible by Grant No. PSI2011-24231 from the Spanish Ministry of Science and Innovation and Grant No. IT-694-13 from the Basque government. I thank Juan Manuel Rosas, Gabriel Rodriguez, and Maria del Carmen Sanjuan in their assistance with the preparation of this manuscript. Excel, in reference to the Microsoft Office Excel spreadsheet program as well as Visual Basic in reference to the development system are registered trademarks of Microsoft Cooperation. SAS is a registered trademark of SAS Institutes. Matlab is a registered trademark of Mathworks. SPSS is a registered trademark of IBM Corporation. OpenOffice Calc is an open-sourced product of Apache Software Foundation.

Corresponding author: J. B. Nelson. Depto. Procesos Psicológicos Básicos y su Desarrollo. Universidad de País Vasco (UPV/EHU). Avenida de Tolosa, 70, San Sebastián, España 20018. E-mail: JamesByron.Nelson@ehu.es

(e.g., Howell, 1987). It is also used in computing confidence intervals on effect-size estimates that are based on the proportion of variance explained by a model, such as the commonly used eta-squared (η^2), and partial-eta squared (η^2_p) effect-size estimates. The paper is intended for researchers who understand the use of non-central distributions and the concepts of power and effect size. The function is written so that it can be implemented by those with basic spreadsheet proficiency (e.g., able to enter formula into cells), as well as those with programming experience.

Fully reporting effect sizes is one illustration of the need for an easily accessible CDF function for non-central F distributions. Several publishing authorities recommend reporting effect sizes in research (e.g., JARS Group, 2008; Wilkinson et al., 1999), and confidence intervals are an important part of that report. Confidence intervals indicate a space that, with a specified probability, covers the true parameter in a population. Imagine an experiment with 4 groups of 16 subjects. Suppose that an analysis of variance on the dependent variable produced an F of 5.27. The resulting effect size, as measured by η^2 , would be .21. Calculating 90% confidence intervals around that estimate would produce η^2 values of .05 - .32. That interval should have about a 90% chance of encompassing the true fixed size of the effect in the population (ignoring issues of bias in η^2).

Despite the importance attributed to them, such intervals are still underreported in research (see Fritz, Morris, & Richler, 2012). Smithson (2001) discusses that one of the limiting factors in the full report of effect sizes is that it requires the use of non-central distributions, and there has been no easy way to access such distributions. Though access has improved, that conclusion is still very likely valid for many researchers. Consider central distributions, those that assume that there are no differences among the variables being compared such as the familiar central t and F distributions used in null-hypothesis significance testing. In addition to being accessible in statistical packages such as SPSS or R, they are also available as easily-used functions in Excel as *TDIST(...)* and *FDIST(...)*. Non-central distributions assume that differences among the variables do exist, and the distributions are shifted by a “non-centrality parameter” (NCP) that is a function of those differences. Though access to an accurate CDF function for non-central t became available for Excel users in the instructional ESCI software (Exploratory Software for Confidence Intervals) presented by Cummings and Finch (2001), a robust function for the non-central F distribution is not available in spreadsheets.

There are resources available that offer the same functionality as the function to be presented here such as *NCDF.F* in SPSS® (IBM Corp, 2011), *ncfcdf* in MatLab® (The MathWorks Inc, 2014), *CDF* in SAS® (SAS Institute Inc, 2014), and the *pf* function in the statistical software R (R Core Team, 2014). The function presented here operates in Excel, a product whose widespread use provides an excellent supplement to commercial statistical applications which are often available only on university installations. Users of Excel typically have Excel available on their personal computers, thus it is on hand in times when they may be working away from the university. Excel is certainly no complete substitute for the statistical applications listed above. Rather, the function's use in Excel complements existing software. After an analysis of variance in SPSS is completed, for example, the results can be easily copied and pasted into Excel. There the function can be used at any time to rapidly find confidence intervals on an effect size. Such a scenario compares favorably to the multi-stage alternative of completing all planned analyses in SPSS; recording the results; saving the original data file; creating a new file in which to hold the computed intervals; then loading and executing a separate SPSS syntax script (e.g., Smithson, 2001).

A spreadsheet also has an advantage in that no programming is required once the function is made available in the spreadsheet. No loading of scripts or defining of variables need be involved. After copying and pasting the function into the appropriate place (see Appendix C), or loading a workbook template containing the function, it is available for use and results can be obtained almost immediately. Making changes in the value of a parameter (e.g., degrees of freedom) results in near-instant feedback. As such, the largest advantages of a spreadsheet are that its interactive nature facilitates rapid data exploration and users can tailor the spreadsheet design to their own preferences. Rather than have a single Excel workbook devoted to calculating confidence limits as a stand-alone application the functions can be integrated within different workbooks as a part of the analysis of different experiments. Other advantages, and limitations, of Excel can be found discussed in Teixeira, Rosa, and Calapez (2009).

A CDF for the non-central F distribution is available for Excel as the function *nf_dist* in a free “add-in” for Excel from www.real-statistics.com (Zaiontz, 2015). The source code for that function is not available, and the robustness of the function is compromised apparently due to the computational limitations of Excel's “double” data type (see the *Evaluation* section of this manuscript). The function presented here avoids many of Excel's computational limitations and produces results comparable to those of the *NCDF.F* function in SPSS. The source code is presented and

thoroughly commented allowing individuals with programming knowledge great flexibility in its implementation. The example spreadsheet that accompanies this manuscript (described in more detail later) contains examples of using the function in power calculations and effect-size confidence interval calculations. The sheet by itself is a useful tool. But the most useful aspect is the CDF function itself which allows users complete flexibility in how they tailor the design of their spreadsheets to their analysis strategies in their research.

The functions listed in the appendices can be copied and pasted into Visual Basic for Applications modules in Excel under “Visual Basic” found on the “Developer Tab” (See Appendix C) or a spreadsheet containing the functions may be downloaded in Appendix D. The spreadsheet contains both functions described in Appendices A and B, as well as the helper functions listed and described in Table 1. The spreadsheet can be used as is, modified, saved as a template, or the functions copied and pasted to new workbooks. For advanced Excel users all source code for these functions is available in the spreadsheet in Visual Basic for Applications (see Appendix C for the steps to view and manipulate this code). As described in Appendix B, the function will work under OpenOffice Calc (www.openoffice.org) with minor modifications. A spreadsheet containing all the functions is also provided for OpenOffice Calc in Appendix E.

Non-Central F

The Visual Basic for Applications code implements the formula for the CDF of a non central F distribution with a non-centrality parameter λ , with numerator and denominator degrees of freedom $v1, v2$, listed below.

$$CD_{F,v1,v2,\lambda} = e^{-\frac{1}{2}\lambda} \sum_{j=0}^{\infty} \left[\frac{\left(\frac{\lambda}{2}\right)^j}{j!} I\left(\frac{Fv1}{v2 + Fv1} \mid \frac{v1}{2} + j, \frac{v2}{2}\right) \right]$$

As can be observed, the majority of the function is easily implemented with summation and basic math functions. The part that presents some difficulty is the regularized incomplete beta function, $I\left(\frac{Fv1}{v2 + Fv1} \mid \frac{v1}{2} + j, \frac{v2}{2}\right)$. Here, the Excel built-in worksheet function $BETADIST(y, \frac{v1}{2} + j, \frac{v2}{2})$ can be used where $y = Fv1/(v2 + Fv1)$. Appendix A presents one implementation of the CDF function written for its readability. The more robust form of the function is presented in Appendix B. These functions were developed and tested using Excel 2007 (12.0.6712.5000).

The Excel implementation shown in Appendix A is accurate across a wide range of parameters, but has failings. The shape of the non-central distribution is affected by the non-centrality parameter and the degrees of freedom. The distribution is shifted toward the non-centrality parameter and its variance is increased. The increase in variance has the overall effect of lowering the height of the distribution and expanding its width. Expanding its width requires more iterations of the formula for accuracy, which creates problems due to Excel's use of the "double" data type for calculations. The number of total iterations is limited to 170, at which time $j!$ will exceed the limits of the double data type producing an "overflow" error. Other operations using j (e.g., $(\lambda/2)^j$) may also exceed the capabilities of Excel. Accuracy becomes insufficient to allow enough iteration to calculate accurate probabilities when the probability density of the tails of the non-central distributions is lessened, such as when increases in degrees of freedom move more of the distribution into the peak, or the entire distribution is depressed by large non-centrality parameters.

The function listed in Appendix A is presented as it easily relates the code to the CDF function above, and provides a transition to understanding the more robust version in Appendix B. It is also listed as tests described in the *Evaluation* section will show that it is likely isomorphic to how the *nf_dist* function from the real-statistics add-in (version 3.1.2 for Excel 2007) implements the CDF. The slightly more complex function in Appendix B uses logarithms to reduce overflow problems and allow for more iterations and an improvement in accuracy. Although $(\lambda/2)^j$ and $j!$ individually may exceed the double data type storage limit, it is seldom the case that their ratio will do so. Moreover, the ratio of these terms multiplied by $e^{-\frac{1}{2}\lambda}$ should be in the range of zero to 1. Thus, with these terms expressed as logarithms overflow problems in the intermediate calculations should be eliminated. Referring to Appendix B, the term $e^{-\frac{1}{2}\lambda}$ is converted to a base-10 logarithm on line 33, while the result of $I\left(\frac{Fv1}{v2+Fv1} \mid \frac{v1}{2} + j, \frac{v2}{2}\right)$ is converted on lines 43-50. The *Betadist* function can return zero, for which calculation of a logarithm is impossible. In that case, the logarithm is set to a value such that its inverse will be extremely small, but still within the double data-type limit. The function $\left(\frac{\lambda}{2}\right)^j$ is accomplished first by finding the log of $\frac{\lambda}{2}$ on line 35, and adding it to itself on each iteration of j (line 52). The factorial of j is calculated simply by summing its logarithms (line 54).

Between lines 66 – 103 the function implements a simple algorithm to allow it to exit the summation loop when further calculations will have

negligible impact on the cumulative density. Increment sizes tend to follow an asymmetric inverted U. They increase from iteration j to iteration $j+1$ slowly, undergo a period of rapid increase which peaks, and then the size of the increments begins to rapidly decrease and then level off. The algorithm finds that final leveling-off period and monitors the decrease until the size of the between-iteration decreases is below a threshold. Completing 2000 iterations may take negligible, almost unnoticeable, time on most systems for a few CDF calculations. When implementing the function across many cells, or calling the function repeatedly in other operations, the results can sometimes be noticeably slow and the early-out algorithm leads to a dramatic performance improvement with very little cost in accuracy. The overall accuracy is determined in the *Evaluation* section of this manuscript. Steps to convert the program for use in OpenOffice Calc are included at the end of Appendix B.

Use in Excel.

The following section will show how the function can be used in Excel to calculate confidence intervals using η^2 as an example. Then, it will show how to accomplish a-priori and post-hoc power analyses using a simple one-way design to illustrate the process. Readers will recognize that there are numerous effect-size statistics that can be used in analysis of variance designs. The examples used here are only to demonstrate ways in which the non-central F CDF function can be used in Excel. It is not the purpose of this manuscript to discuss the adequacy of, nor advocate, the statistics that the example employs. Such a debate is beyond the manuscript's scope, and irrelevant to the presentation of the CDF function. Regardless of the effect-size statistic from the variance-explained class (e.g., η^2 , ω^2 , f^2) that a researcher deems appropriate for their report, access to a non-central F CDF is desirable for construction of confidence intervals. The section summarizes the logic and general steps of the parametric approach to computing effect sizes (as opposed to bootstrap methods, see Finch & French, 2012), and how those steps are accomplished with the Excel function. More detailed, though highly accessible, overviews of the general process are available in Cummings and Finch (2001), Fidler and Thompson (2001), and Smithson (2001).

When calculating confidence intervals we assume that there is a true effect in the population, that is, the obtained test statistic (e.g., F) came from a non-central distribution. For a confidence interval (e.g., 90%), the question then becomes what would be the necessary NCP to shift the distribution to the right so that obtained F would cut off the bottom 5% of

the distribution (i.e., the “upper” NCP). Then, what would be the necessary NCP be to shift the distribution left until the obtained F cut off the top 5% of the distribution (i.e., 95% below, the “lower” NCP). These NCPs are then converted to the desired effect size. The *NCF_Dist(...)* function can be used to quickly find the necessary non-centrality parameters by changing NCP until the required probability is obtained. The resulting NCP is then converted into the preferred effect size using simple linear transforms. While the task can easily be accomplished manually, the spreadsheet provides helper functions to accomplish these tasks (see Table 1).

Table 1. Listing of functions.

Function name and parameters	Purpose
NCF_Dist(F,df1,df2,NCP)	Returns the area below F with df1,df2 degrees of freedom and a non-centrality parameter (NCP).
NCP_to_Eta(NCP, df1,df2)	Calculates η^2 from an NCP and degrees of freedom where $\eta^2 = \text{NCP}/(\text{NCP}+\text{df1}+\text{df2}+1)$.
F_to_etasquared(F,df1,df2)	Calculates η^2 from F where $\eta^2 = F*\text{df1}/(F*\text{df1}+\text{df2})$.
Etasquared_to_F(etasquared,df1,df2)	Calculates F from η^2 where $F = \eta^2 * \text{df2} / (\text{df1} * (-\eta^2 + 1))$.
F_to_NCP(F,df1,df2)	Takes an F and calculates its respective NCP where $\text{NCP} = F*(\text{df1}/\text{df2})*(\text{df1}+\text{df2}+1)$.
Eta_to_CohenF(etasquared)	Takes eta-squared and converts to Cohens <i>f</i> using $f = \sqrt{\eta^2 / (1-\eta^2)}$.
CohenF_to_eta(CohenF)	Takes Cohen's <i>f</i> and converts to η^2 with $\eta^2 = (f^2)/(1+f^2)$.
Omegasq_from_etasq(etasq,mseffect,df1,df2)	Takes η^2 , MS effect, the degrees of freedom and calculates ω^2 .
P_to_ncp(cutoff,F,df1,df2)	Determines the necessary NCP so that the specified F will cut off the lower portion of the distribution determined by “cutoff.”
Find_NCPs(F,df1,df2,confidence,direction)	Returns the upper (direction = 1) or lower (direction = 0) NCP for a given F and confidence interval.
Find_critical_F(alpha,df1,df2)	Finds the critical value of F associated with a given alpha and degrees of freedom.

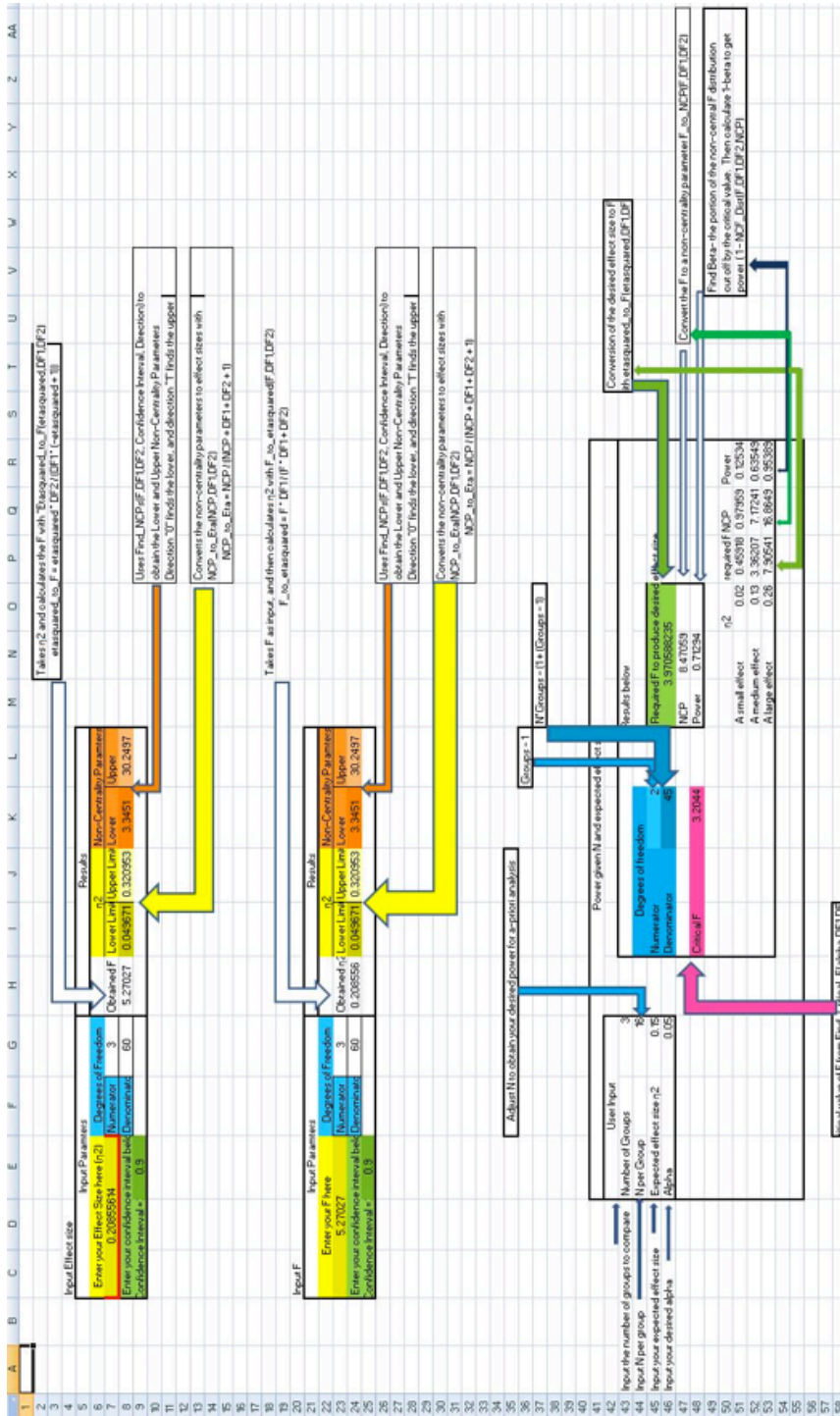


Figure 1. Screenshot of the “function usage explanations” worksheet from the spreadsheet available in Appendix D.

Figure 1 shows the spreadsheet in Appendix D. The first section demonstrates how to use the functions to obtain confidence intervals on η^2 . The top box “Input Effect size” allows the user to input an effect size (.209), desired confidence interval (.9), degrees of freedom (3,60) and returns several results. Cell H8 shows the F (5.27) associated with the effect size as calculated by *Etasquared_to_F(...)*. Cells K8-L8 show the necessary non-centrality parameters (3.346 & 30.25). These upper and lower non-centrality parameters can be found by entering *NCF_Dist(F,Df1,Df2,NCP)* into a cell where NCP is a reference to a different cell containing the non-centrality parameter. The value of the NCP can then be changed until the function returns the necessary cumulative density (e.g., 5% for the upper NCP). Though a user can accomplish that task manually in minimal time, the function *Find_NCPs(...)*, used in the spreadsheet in cells K8 and L8, automates those steps. It calls the function *NCF_Dist(...)* with different non-centrality parameters until the desired cumulative density is obtained (see Table 1 for more details on the *Find_NCPs(...)* function). Cells I8 and J8 show the lower (.05) and upper confidence (.32) limits on η^2 returned by *NCP_to_Eta(...)*.

The lower section, “Input F” allows users to input their obtained F and degrees of freedom and returns the associated η^2 with *F_to_Etasquared(...)* in cell H24 along with the confidence limits and non-centrality parameters. To obtain confidence limits on ω^2 or partial ω^2 , η^2 or η^2_p and their respective limits can be converted to ω^2 or partial ω^2 (see Fidler & Thompson, 2001) using the function *omegasq_from_etasq(...)*.

The bottom section of the spreadsheet shows the use of the CDF function in power calculations. Power is an estimate of the probability of correctly rejecting the null hypothesis. In calculating power the desired or expected effect size is converted to F, and assuming that F came from a non-central F distribution, we use it to find the non-centrality parameter to be able to calculate cumulative densities from that distribution. Then, we locate the critical value of F necessary to produce a “significant” result with a given alpha in that distribution. Any values of F that come from that portion of the curve below that critical value would not meet the rejection criterion, and would thus represent the proportion of Type II errors. The portion of the curve above that F ($1 - NCF_dist(...)$) represents the power.

In column G the user can input the number of groups (3) and N per group (16) in the design, the expected effect size (.15) as η^2 , and alpha (.05). On the right, in column K, the spreadsheet calculates the degrees of freedom in the design and the critical value of F with *Find_critical_F(...)*. In H46, the spreadsheet uses *eta_to_F(...)* to calculate the F necessary to

produce the desired effect size (3.97). The NCP associated with that F is then calculated (8.47) in K25 with $F_to_NCP(...)$, and the power ($1 - NCF_Dist(...)$) = .71) in N48. Below in rows 50-53, the spreadsheet also shows the power of the design for small, medium, and large “rough-guideline” effect sizes of η^2 given the design. The values calculated by the spreadsheet correspond to those produced by the stand-alone program G-Power version 3.1.9.2 (Faul, Erdfelder, Lang, & Buchner, 2007) to at least the fourth decimal, when η^2 is converted to Cohen’s f which is used by G-Power.

To determine the size of the samples necessary to obtain a given effect with a specified power, N per group can be adjusted by entering new values, or using the “N-Adjuster” scroll bar (not shown in Figure 1 but available in the spreadsheet), until the desired power is reached in cell N48. For example, to have a power of .8 to detect an effect size of $\eta^2 = .15$ in a 3-group experiment with alpha = .05, 19 subjects per group would be required (power = .795).

Evaluation

To test the accuracy of the functions listed in the appendices, a range of Fs were generated with SPSS (version 20) to cut off cumulative densities within multiple non-central F distributions. Within each non-central F distribution 10 Fs were obtained, each cutting off a respective interval beginning at .05 and incrementing by .1 to .95. Each F was incremented by .01 until the density below it exceeded the specified interval. That procedure was conducted on 5700 different non-central F distributions spanning λ between 1 and 300 in increments of 1, $v1$ values of 1-10 in increments of 1, and $v2$ values of 20-200 in increments of 10. The resulting 570,000 cumulative densities were compared to those produced by the functions in Appendices A and B, as well as the nf_dist function from the Real-Statistics add in for the same Fs and non-central F distributions.

The function in Appendix A appears to be equivalent to that used by the nf_dist function available from the Real-Statistics add in. Using 150 iterations the two functions returned exactly the same values and both failed with non-centrality parameters of 228 and higher. Prior to failure, the functions were largely accurate with the following statistics based on the absolute deviation from the results produced by SPSS (average = 1.27×10^{-6} , median = 2.22×10^{-10} , min = 5.41×10^{-9} , max = .00037). To test the accuracy prior to failure, the listing in Appendix A was modified to return the last valid result when it reached the point of failure. The overall absolute accuracy was poor on these last 138,700 tests (average = .088, median =

.016, min = 2.83×10^{-9} , max = 0.71,) and the code in Appendix A will not be considered further.

The function in Appendix B was robust. It tended to slightly over-estimate the cumulative density with only .04% of the estimates being below those produced by SPSS. After converting the deviations from SPSS to their absolute values the average deviation was 2.67×10^{-07} (median = 2.39×10^{-07} , min = 2.13×10^{-13} , max = 8.86×10^{-07}). The same set of comparisons of the Appendix B function to the cumulative densities produced by the *pf* function in R yielded nearly identical accuracy (mean = 2.67×10^{-7} , median = 2.39×10^{-7} , min = 9.12×10^{-15} , max = 8.85×10^{-7}). Commenting out the code that allows the function to exit early (lines 89-96), as would be expected, improved accuracy. The statistics on the absolute deviations with SPSS were as follow, mean = 3.16×10^{-10} , median 1.71×10^{-10} , min= 0, max = 5.6×10^{-9} . Comparisons with R produced accuracy to the same decimal place in all summary statistics as reported with SPSS with the exception of the minimum, which was accurate to the 15th decimal, the accuracy limit for double data types.

As a second test, the function was used as it might be used to calculate confidence intervals (%90) on a variety of effect sizes and one-way designs. Effect size (η^2) values were created between .04 and .98 in increments of .01. Each of those was paired with designs consisting of 2 to 11 groups (i.e., numerator degrees of freedom from 1 to 10). Imaginary sample sizes were adjusted so that the F required to produce the specified effect size, ($F = \eta^2 Df_{denom.} / (Df_{num.}(1 - \eta^2))$), would be significant at at least $p < .025$. To accomplish that adjustment, for each numerator degrees of freedom, a sample size beginning at $n = 4$ was specified. The resulting denominator degrees of freedom were calculated and the necessary F to obtain the specified effect size and its accompanying probability were calculated. If necessary, the n of each group was incremented by 1 and the calculations repeated until the probability of the F or greater was $p < .025$. Upper and lower non-centrality parameters were obtained by incrementing the non-centrality parameters by .001 and obtaining the cumulative probability of the specified F using the function in the Appendix B. The accuracy of the obtained non-centrality parameters was compared to those obtained using the same procedure in SPSS.

The function listed in Appendix B was good on all tests. Of the 1900 test values there were minor deviations from the results obtained by SPSS on 5% of the cases (.001 in 4.6% of the cases, .002 in .3% of the cases and .003 in .2% of the cases) and occurred with extremely large effect sizes ($\eta^2 > .78$). The deviations were particularly small when considered as a

proportion of the size of the parameter estimated by SPSS. Among the non-zero deviations the average deviation was .0002% when considered as such.

RESUMEN

Una función robusta para calcular con Microsoft Office Excel la densidad acumulada de las distribuciones F no centrales. Este manuscrito presenta una función de Visual Basic® para aplicaciones que opera en el entorno de Microsoft Office Excel® computando el área bajo la curva de un valor de F determinado dentro de una distribución F no central. Es una función útil para usuarios de Excel con o sin experiencia de programación en aquellas ocasiones en las que se requiere una distribución F no central, como cuando se realizan análisis de la potencia para diseños de análisis de varianza o cuando se necesita calcular los intervalos de confianza en el tamaño del efecto. Las pruebas realizadas con esta función encuentran resultados equivalentes a los obtenidos con el programa comercial SPSS y el popular entorno gratuito R para el cálculo estadístico. Se incluye una hoja de cálculo con ejemplos de uso.

REFERENCES

- Cummings, G., & Finch, S. (2001). A primer on the understanding, use and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*, 633-649.
- Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods, 39*, 175-191.
- Fidler, F., & Thompson, B. (2001) Computing correct confidence intervals for ANOVA fixed- and random-effects effect sizes. *Educational and Psychological Measurement, 61*, 575-604.
- Finch, W.H., & French, B.F. (2012). A comparison of methods for estimating confidence intervals for Omega-Squared effect size. *Educational and Psychological Measurement, 72*, 68-77.
- Fritz, C. O., Morris, P. E., & Richler, J.L. (2012). Effect size estimates: Current use, calculations and interpretation. *Journal of Experimental Psychology: General, 141*, 2-18.
- Howell, D.C. (1987) *Statistical Methods for Psychology, 2nd Edition*. Boston: PWS-Kent.
- IBM Corp. (2011). *IBM SPSS Statistics for Windows, Version 20.0*. Armonk, NY: IBM Corp
- R Core Team (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>
- SAS Institute Inc (2014). *SAS, Version 9.4*. Cary, NC.
- Smithson, M. (2001). Correct confidence intervals for various regression effect sizes and parameters: The importance of noncentral distributions in computing intervals. *Educational and Psychological Measurement, 61*, 603-630.

- Teixeira, A., & Rosa, A., & Calapez (2009). Statistical power analysis with Microsoft Excel: Normal tests for one or two means as a prelude to using non-central distributions to calculate power. *Journal of Statistics Education*, 17, 1-21.
- JARS Group: The American Psychological Association Working Group on Journal Article Reporting Standards. (2008). Reporting Standards for Research in Psychology: Why Do We Need Them? What Might They Be? *American Psychologist*, 63(9), 839-851.
- TheMathWorks Inc (2014). *MATLAB- The language of technical computing, Version R2014B*, Natick, Massachusetts.
- Wilkinson, L., & the Task Force on Statistical Inference. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, 54, 594-604.
- Zaiontz C. (2015) *Real Statistics Using Excel*. www.real-statistics.com

APPENDIX A

```

1  Function NCF_dist1(F As Double, v1 As Double, _
2  v2 As Double,u As Double) as Double
3
4  Dim j As Double:      Dim y As Double
5  Dim Iy As Double:     Dim v1_over2 As Double
6  Dim v2_over2 As Double: Dim u_over2 As Double
7  Dim sum As Double:    Dim Ratio as Double
8  \y-----
9  y = (v1 * F) / (v1 * F + v2)
10 \lambda-----
11 u_over2 = u*.5
12 \frac{v1}{2}-----
13 v1_over2 = v1 * 0.5
14 \frac{v2}{2}-----
15 v2_over2 = v2 * 0.5
16
17 For j = 0 To 150
18 \I_y\left(\frac{v1}{2} + j, \frac{v2}{2}\right)-----
19 Iy = Application.WorksheetFunction._
20 BetaDist(y, v1_over2 + j, v2_over2)
21 \left(\frac{\lambda}{2}\right)^j-----
22 Ratio = (u_over2 ^ j) / _
23 Application.WorksheetFunction.Fact(j)
24 \sum_{j=0}^{\infty} \left[ \left(\frac{\lambda}{2}\right)^j I_y\left(\frac{v1}{2} + j, \frac{v2}{2}\right) \right]-----
25 sum = sum + (Ratio * Iy)
26
27 Next j
28 \ e^{-\frac{1}{2}\lambda} \sum_{j=0}^{\infty} \left[ \left(\frac{\lambda}{2}\right)^j I_y\left(\frac{v1}{2} + j, \frac{v2}{2}\right) \right]-----
29 NCF_dist1 = Exp(-u_over2) * sum
30 End Function

```

APPENDIX B

```

1  Function NCF_Dist(F As Double,v1 As Double,v2 As Double, _
2    u As Double) As Double
3  'The following variables are used in the CDF calculations---
4  Dim j As Double:           Dim Iy As Double
5  Dim v1_over2 as Double:    Dim v2_over2 as Double
6  Dim u_over2 As Double:     Dim u_over2_raisedJ As Double
7  Dim J_fact As Double:     Dim Ratio As Double
9  Dim Raise_Power As Double: Dim Sum As Double
9  Dim y As Double:           Dim increment As Double
10 Dim final_log As Double:   Dim Exp_neg_u_over2 As Double
11
12 'These variables are used to exit the function early when
13 'further calculations are relatively meaningless -----
14 Dim change As Double:      Dim last_increment As Double
15 Dim Max As Double:        Dim abs_change As Double
16 Dim percent_max As Double: Dim past_middle As Boolean
17 Dim at_middle As Boolean:  Dim accuracy_constant As Double
18
19 'Start by setting starting values outside the main loop ---
20 y = (v1 * F) / (v1 * F + v2):   v1_over2 = v1 * 0.5
21 v2_over2 = v2 * 0.5:           u_over2 = u * 0.5
22 Sum = Application.WorksheetFunction. _
23     BetaDist(y, v1_over2, v2_over2)*Exp(-u_over2)
24
25 accuracy_constant = 0.0001
26 'Now we use log10() of the terms involved in factorials,
27 'exponents, and multiplications. Note that we will be
28 'taking the Log10() of Exp(-u_over2) which will fail when
29 'u_over2 is > ~745.13321 as exp(-u_over2)will return zero.
30 'To avoid problems, we'll take advantage of Log10(Exp(-X))
31 'being equal to X * -0.434294481903252.
32
33 Exp_neg_u_over2 = u_over2 * -0.434294481903252
34
35 Raise_Power = Application.WorksheetFunction. _
36     Log10(u_over2)
37
38 'Main loop- iterates and sums. 2000 can be increased for
39 'gigantic NCPs. t is unlikely ever to be necessary, however
40
41 For j = 1 To 2000
42
43 Iy = Application.WorksheetFunction. _
44     BetaDist(y, v1_over2 + j, v2_over2)
45 'if Iy is zero or practically zero then
46 If Iy < 2.2251E-308 Then
47 Iy = -307.6526556 'set its log to the minimum possible
48 Else'else calculate its log
49 Iy = Application.WorksheetFunction.Log10(Iy)
50 End If
51 'raise u/2 to the j
52 u_over2_raisedJ = u_over2_raisedJ + Raise_Power
53 'get j!
54 J_fact = J_fact + Application.WorksheetFunction. _

```



```

55             Log10(j)
56 Ratio = u_over2_raisedJ - J_fact
57
58 final_log = Ratio + Iy + Exp_neg_u_over2
59 'Power(), as opposed to 10^x will not overflow with large
60 'negative values of final_log, but will return zero, so no
61 'value checking is necessary here.
62 increment = Application.WorksheetFunction. _
63     Power(10, final_log)
64
65 Sum = Sum + increment
66 'Check to see if we can exit early- Keep track of the
67 'maximum amount of change that has occurred
68 change = increment - last_increment
69 abs_change = Abs(change)
70 'check to see if we've past the middle of the hump
71 If (at_middle) Then
72     If (Not past_middle) Then
73         past_middle = True
74     End If
75 End If
76
77 'check to see if we're -at- the middle so that on the next
78 'pass we will be past the middle. We have to be past the
79 'middle or lines 92 & 96 can trip the exit too soon
80 If (Not at_middle) Then
81     If change < 0 Then
82         at_middle = True
83     End If
84 End If
85
86 'Keep track of the maximum change so far, because a
87 '"small" change is relative.
88 If abs_change > Max Then Max = abs_change
89 'if past the small changes at the hump then
90 If (past_middle) Then
91 'if decreasing increments
92 If (Abs(increment) < Abs(last_increment)) Then
93 'scale to relative increment size
94     percent_max = abs_change / Max
95 'if below accuracy constant
96     If (percent_max < accuracy_constant) Then
97         Exit For 'then quit the "For j = 1 to 2000" loop
98     End If
99 End If
100 End If
101
102 last_increment = increment
103 'Ends of the early-exit checking section of the code--
104 Next j
105 NCF_Dist = Sum 'return the cumulative sum
106 End Function

```

Conversion to Open Office

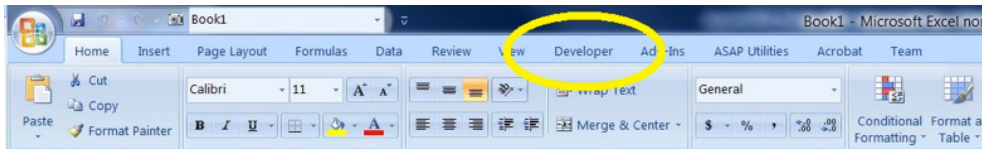
- 1: Line 18 add: `Dim Functions`
- 2: Line 19 add:
`Functions = createunosevice("com.sun.star.sheet.FunctionAccess")`
- 3: replace lines 22-23:
`Sum = Functions.
_callFunction("BETADIST",array(y,v1_over2,v2_over2))_
*Exp(-u_over2)`
- 4: Replace line 35:
`Raise_Power = Functions.callFunction("LOG10",array(u_over2))`
- 5: Replace line 43:
`Iy = Functions.
_callfunction("BETADIST",array(y,v1_over2+j,v2_over2))`
- 6: In line 46 change 308 to 300.
- 7: Replace line 49: `Iy =Functions.callFunction("LOG10",array(Iy))`
- 8: Replace line 54:
`J_fact= J_fact+ Functions.callFunction("LOG10",array(j))`
- 9: Replace line 62:
`increment = Functions.callFunction("POWER",array(10,final_log))`

APPENDIX C

To use any of the code from Appendix A or B, it must be available as a Function in a Module in the Visual Basic for Applications Project of the spreadsheet. A spreadsheet is available Appendix D that contains all the code (along with all the functions in Table 1) and is ready to be used. The easiest way to have access to all the functions in any new workbook is to load the supplementary workbook and delete the first four sheets, and rename the remaining sheet as sheet1, effectively creating a blank startup workbook. Then, save that workbook as a “template” (Office-Button→Save as→Other Formats→Excel Macro-Enabled Template *.xlsm). Once saved, that template can be used to create any new workbooks under “Office-Button→New→Templates” which will create a new workbook from the template that will have all the functions available.

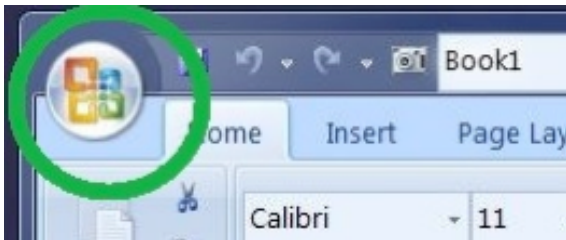
As an alternative, the code for the functions can be copied from the workbook’s Visual Basic for Application module and pasted into that of another workbook. The code can also be copied and pasted directly from this manuscript into a Visual Basic for Applications Module. To create a module into which to paste code, follow the steps below, which are described for users with even little experience.

Look for the “Developer tab” on the ribbon.

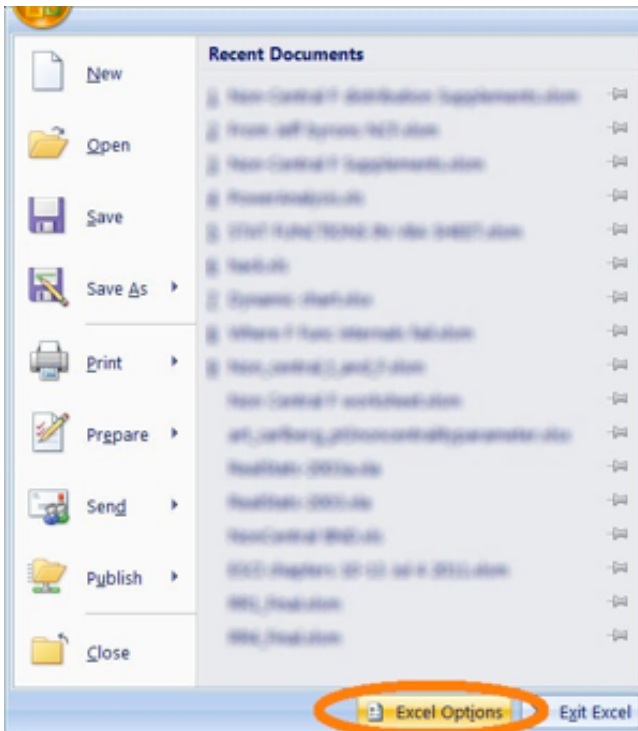


If the tab is not visible, follow the next four steps.

1: Click the Office button, circled in Green in Figure 3.

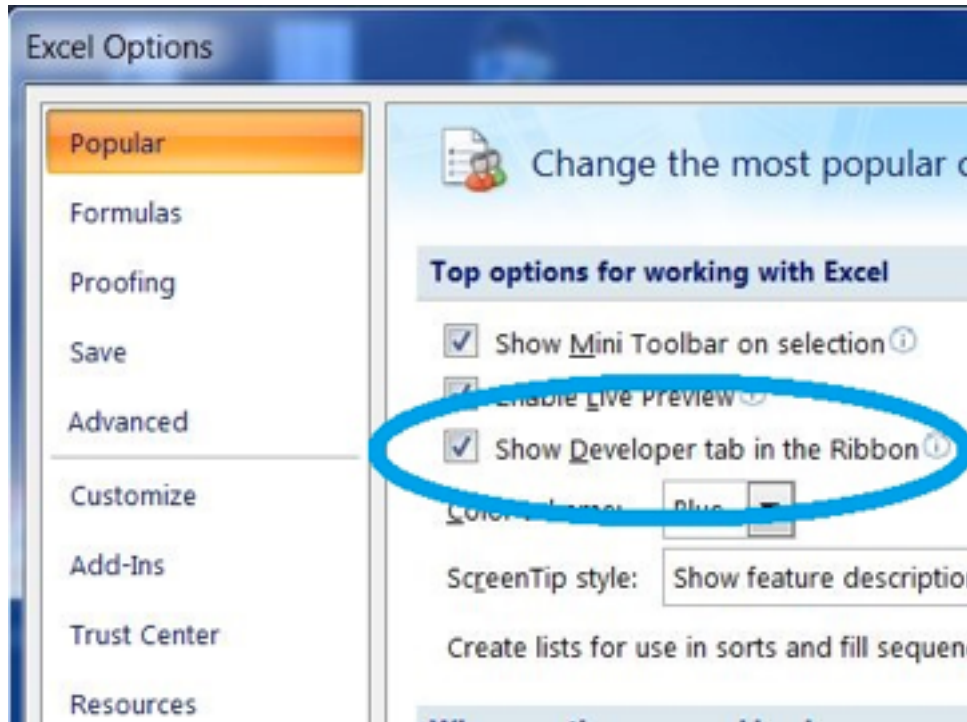


2: Select “Excel Options” circled in orange in Figure 4.



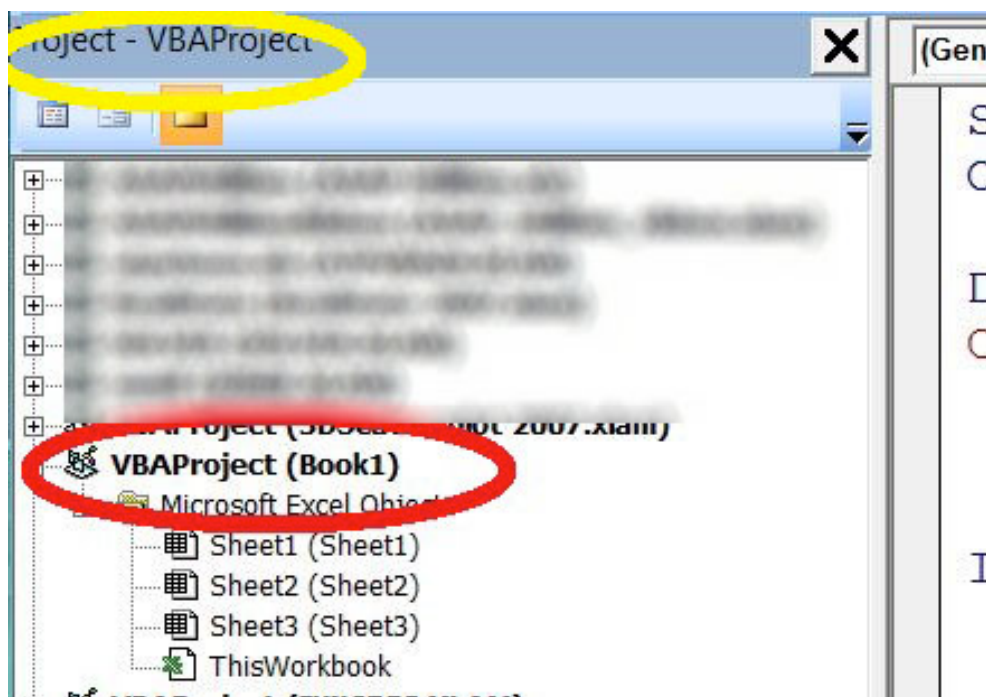
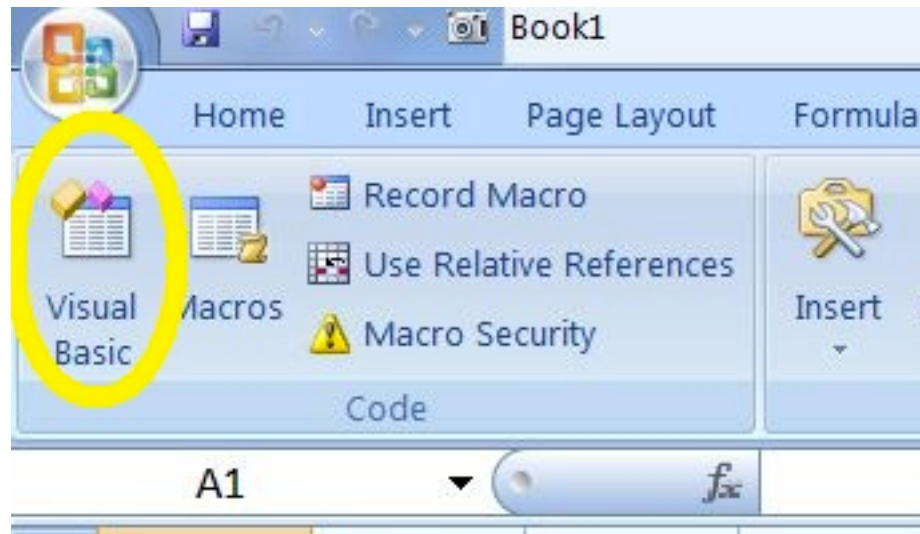
3: Under those options select “Popular” and... (highlighted top-left in Figure 5)

4: Check “Show Developer tab in the Ribbon” (circled in blue in Figure 5)



On the ribbon, click the “Developer tab” as in Figure 2 above.

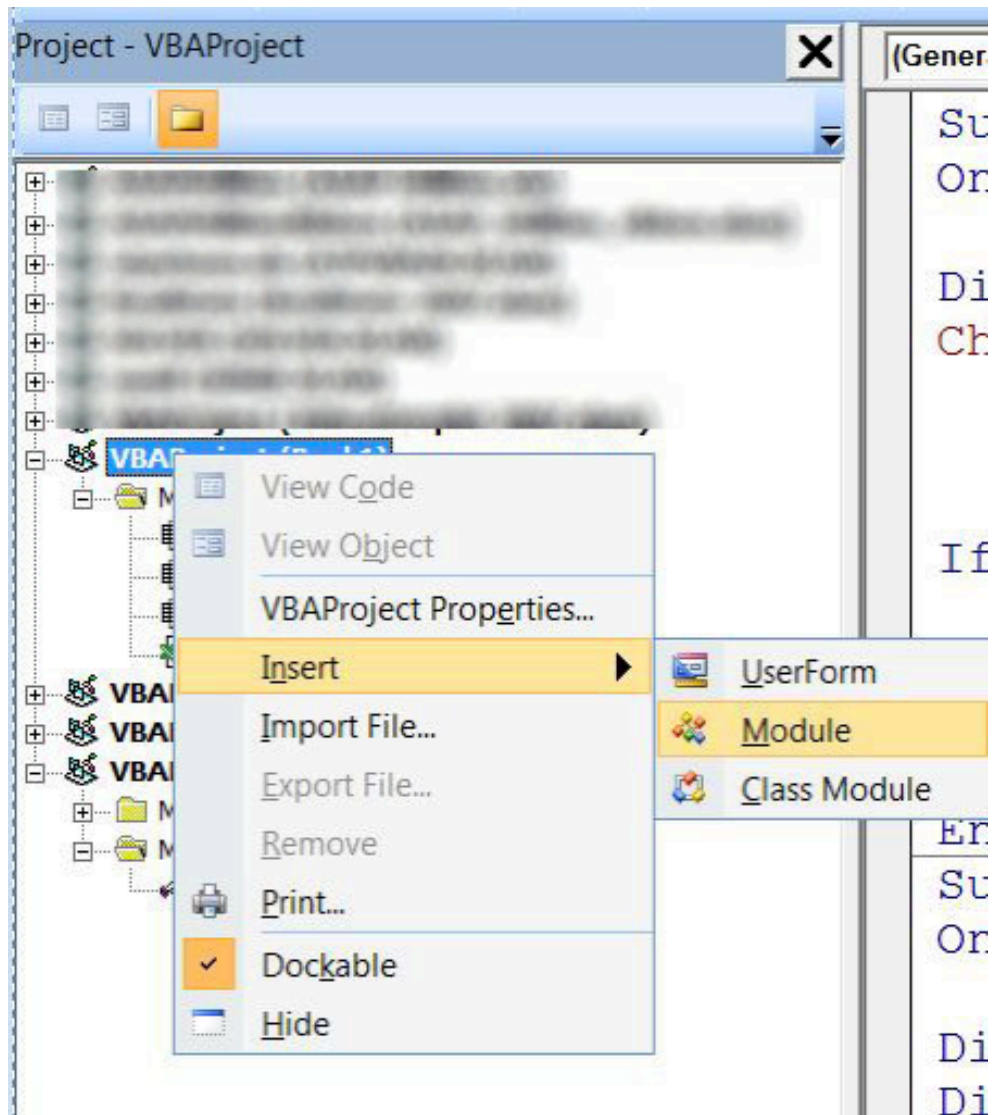
On the Developer tab, click “Visual Basic” circled in yellow in Figure 6. Clicking “Visual Basic” will open a new window containing the Visual Basic for Applications Editor, part of which appears in Figure 7.



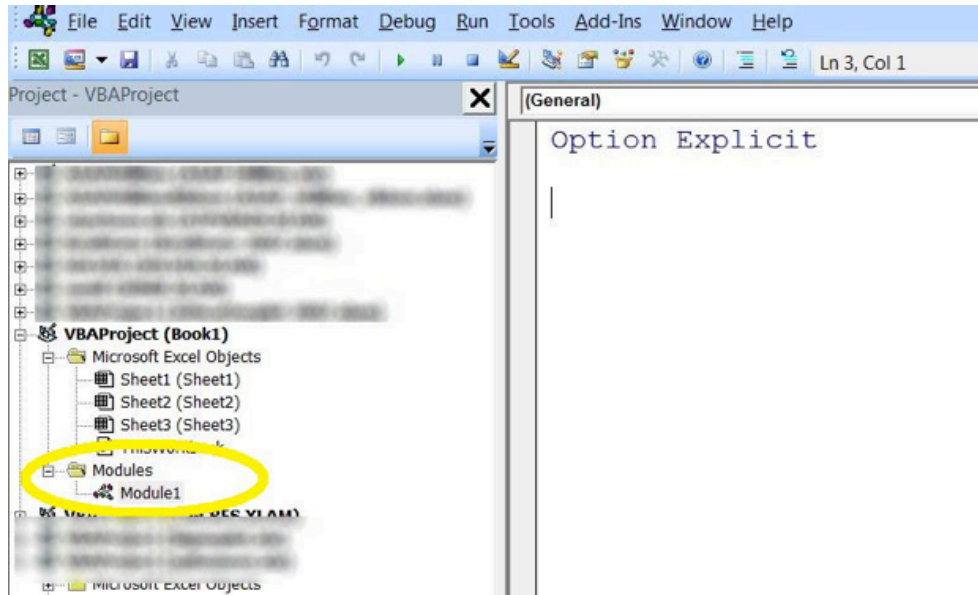
In the editor, look for a window to the left titled “Project-VBAProject” as circled in Yellow in Figure 7. If the window is not visible, then it can be made visible by clicking “View” and selecting “Project Explorer.”

In the Project Explorer window, look for VBAProject(book1) as circled in red in Figure 7 (when you save your file, the name in parentheses will change).

Right Click on the “VBAProject (Book1)” text and select “Insert” and then “Module” as shown in Figure 8.



A new module (Module1) should appear in the Project Explorer window (Circled in Yellow in Figure 9), and a blank page should appear to the right. There may be text in this window (e.g., “Option Explicit” at the top) or not. This window is where you will put the code.



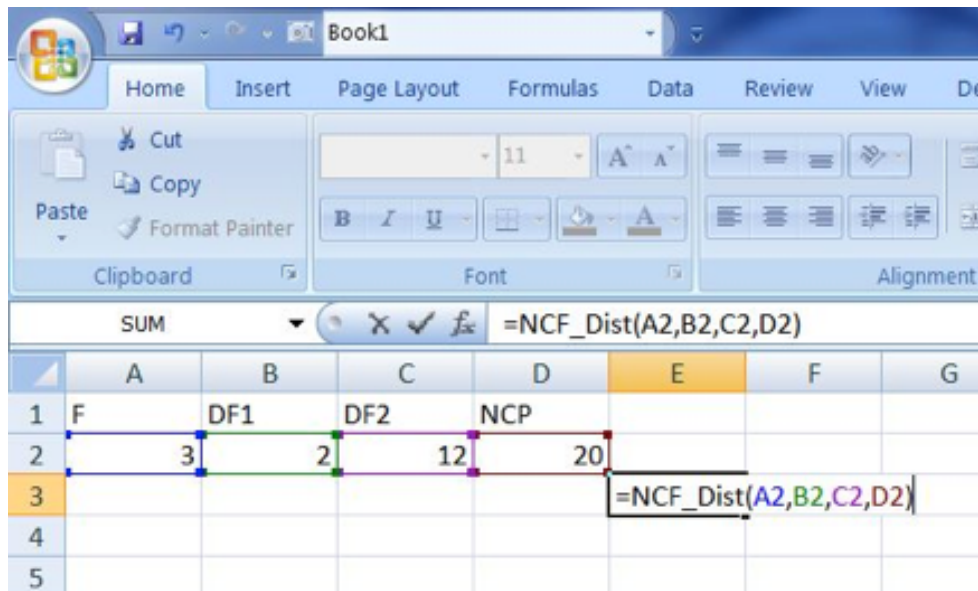
Copy the code you wish to use from Appendix A or B (the NCF_Dist function of Appendix B is the more accurate and preferred), and paste it below any text appearing in the window. Begin copying with the word “Function” and end with the words “End Function.” Paste directly into the window. The result should appear something like below (see Figure 10, colors may be different). *If any page numbers, line numbers, page headings etc. were copied and pasted (e.g., see 1 2 3... running down the window below, circled in purple in Figure 10), be sure to delete them.* If no errors were made in the copying and pasting operation (e.g., any extraneous characters and page materials etc. were removed) the function will now work in the spreadsheet. **To preserve the formatting, you may have to copy to an editor such as Word first, and then copy to the VBA window.**

```

Option Explicit
1 Function NCF_Dist(F As Double, v1 As Double, v2 As Double, u As
2 'The following variables are used in the CDF calculations-----
3 Dim j As Double: Dim Iy As Double
4 Dim v1_over2 As Double: Dim v2_over2 As Double
5 Dim u_over2 As Double: Dim u_over2_raisedJ As Double
6 Dim J_fact As Double: Dim Ratio As Double
7 Dim Raise_Power As Double: Dim Sum As Double
8 Dim y As Double: Dim increment As Double
9 Dim final_log As Double: Dim Exp_neg_u_over2 As Double

```

The function is accessed in the spreadsheet in the same way as any other built-in spreadsheet function by typing “=” and then typing the function name (see Figure 11). The function should appear in a list where it can be selected by pressing “tab.” From there, select the cells containing the parameters for the function (F, degrees of Freedom for the numerator, degrees of freedom for the denominator and the non-centrality parameter) or type the numbers in directly and then press “enter.” If the function was successfully entered into the Module, you should be able to enter =NCF_Dist(3,2,12,20) and obtain 0.024057 for a result.



To maintain functionality, the spreadsheet must be saved under “Save as->Excel Macro enabled workbook” otherwise the functions will be disabled.

(In OpenOffice Calc, the Basic Editor is found under Tools→Macros→Organize Macros→OpenOffice Basic→New.)

(Manuscript received: 13 March 2015; accepted: 23 June 2015)