Journal of Business and Economics, ISSN 2155-7950, USA
August 2016, Volume 7, No. 8, pp. 1204-1214
DOI: 10.15341/jbe(2155-7950)/08.07.2016/002
© Academic Star Publishing Company, 2016
http://www.academicstar.us

Doing Statistical Analysis with Spreadsheets

Jerzy Letkowski
(Western New England University, USA)

Abstract: VisiCalc, Lotus 1-2-3 and Quattro Pro had their days followed by long and almost absolute dominance of Microsoft Excel. In terms of the spreadsheet functionality, Excel has reached a state close to perfection as a problem solving software tool. Its operation is fast, smooth and efficient. It is equipped with a reach set of build in functions and services. It can virtually be infinitely extended by means of its macros (VBA). Today, which is around year 2016, an explosion of new spreadsheet alternatives is happening with interesting new solutions provided for different systems, ranging from Windows, through OS X, iOS, Linux, to Android and others. This paper attempts to explore capabilities of the most popular alternatives: Google Sheets and Open Office Calc with respect to solving statistical problems. Although its focus is on spreadsheet solutions for doing both the descriptive and inferential Statistics, using built-in functions, examples of user-defined functions are also given.

Key words: statistics; spreadsheet; built-in function; user-defined function
JEL codes: C46, C83, C88

1. Introduction

Business Statistics is one of many popular courses taught within business curricula.

A wide range of textbooks exists to support such courses some of which provide exclusive support for Excel as the primary computational tool (Anderson et al., 2011; Donnelly, 2014; Jaggia, 2012; Levine et al., 2013). Many other text book give students opportunities to apply Excel and statistical packages such as MINITAB (Black, 2011; Doane & Seward, 2010; Larose, 2010; Pelosi & Sandifer, 2003) and/or SPSS (McClave et al., 2012; (Sharpe at al., 2011). Other instructional materials act as guides or references (Dretzke, 2011) as well as cases for applying Excel to solve statistical problems (Pelosi et al., 1996, 1998; Letkowski, 2012, 2014).

Despite overwhelming usage and coverage of Excel, other alternatives become more viable. For example, in the author's classes, all students receive instructions for completing case studies in Excel, but increasingly more students elect to do their work, using other statistical programs, predominantly — Apache OpenOffice Calc (OpenOffice Calc, 2015) or Google Sheets (Google, 2015). Other worth mentioning contenders are LibreOffice (LibreOffice, 2015), whose spreadsheet package Calc is equipped with lots of statistical functions, including basic probability distribution functions. Beal (2015) evaluated 5 "mature", open-source alternatives to Excel, listing OpenOffice Calc and LibreOffice as top 2 of the five. Other three options include NeoOffice for Mac OS X, Google Docs (an online, browser based solution) and KOffice (available for all common operating systems).

This paper focuses on three popular spreadsheet programs: Excel (XL), and Google Sheets (GS), and

Jerzy Letkowski, Ph.D. in Economics, Western New England University; research areas/interests: data science and semantic web. E-mail: jerzy.letkowski@wne.edu.

OpenOffice Calc (OC). It examines basic functionality (without add-ins or plug-ins) of the programs with respect to meeting objectives for an introductory course in Statistics.

2. Spreadsheet Applications for Probability

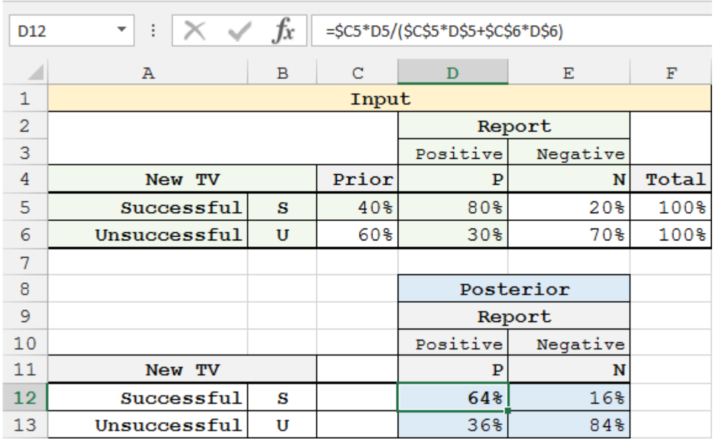
Assessing probabilities for typical events and random variables, using spreadsheets is quite straightforward. The three spreadsheet programs, explored in this paper, provide lots of computational power, including formula development and specialized functions. The scope of this paper is limited to basic functionality of Excel, Google Sheets and OpenOffice Calc. However, each of the programs can be expanded by means of scripting (programming) languages (Excel VBA, Google Apps Scripts, and Java, respectively). Such a capability makes is possible to solve even very sophisticated problems. If there are some functions missing, they can be added to the programs as custom-built functions.

The probability rules of Addition, Multiplication along with conditional probabilities, including the Bayes Theorem, can be easily applied in spreadsheets by means of formulas, involving basic arithmetic and logical expressions. As an example, consider a marketing case presented in Levine at al. (2013, p. 172):

> You can apply Bayes' theorem to the situation in which M & R Electronics World is considering marketing a new model of televisions. In the past, 40% of the new- model televisions have been successful, and 60% have been unsuccessful. Before introducing the new model television, the marketing research department conducts an extensive study and releases a report, either favorable or unfavorable. In the past, 80% of the successful new- model television(s) had received favorable market research reports, and 30% of the unsuccessful new- model television(s) had received favorable reports. For the new model of television under consideration, the marketing research department has issued a favorable (positive) report. What is the probability that the television will be successful?

Figure 1 captures Excel-based input and output for this problem, where cell D12 returns the probability in question according to the Bayes Theorem:

$$P(S|P) = P(P|S)\cdot P(S) / [P(P|S)\cdot P(S)+ P(P|U)\cdot P(U)].$$



	A	B	C	D	E	F
1	Input					
2				Report		
3				Positive	Negative	
4	New TV		Prior	P	N	Total
5	Successful	S	40%	80%	20%	100%
6	Unsuccessful	U	60%	30%	70%	100%
7						
8				Posterior		
9				Report		
10				Positive	Negative	
11	New TV			P	N	
12	Successful	S		64%	16%	
13	Unsuccessful	U		36%	84%	

Figure 1 An Excel Implementation of a Bayes Theorem's Application

A similar solution can be developed, using Google Sheets (GS) and OpenOffice Calc (OC) (Letkowski, 2015, Bayes Application).

More complex probability problems may require combinatorial analysis and formulas. The three spreadsheet program provide a similar support (Table 1).

Table 1 Combinatorial Functions

Function	Excel (XL)	Google Sheets (GS)	OpenOffice Calc (OC)
Factorial and Full Set Permutations	FACT(n)	FACT(n)	FACT(n)
Subset Permutations	PERMUT(n, m)	COMBIN(n, m)*FACT(n)	PERMUT(n; m)
Subset Permutations with Repetition	PERMUTATIONA(n; m)		PERMUTATIONA(n; m)
Combinations	COMBIN(n, m)	COMBIN(n, m)	COMBIN(n; m)
Subset Combinations with Repetition	COMBINA(n; m)		COMBINA(n; m)

Notice that GS does not have a specialized function for calculating the number of permutations of size m selected from a set of n elements with repetitions as well as the number of combinations of size m selected from a set of n elements with repetitions, provided by other programs as PERMUTATIONA() and COMBINA(), respectively. However, these functions can be implemented by the following formulas: = n^m and = COMBIN(n+m-1, n-1), respectively.

Table 2 Probability Distribution Functions

Probability Distribution	Excel (XL)	Google Sheets (GS)	OpenOffice Calc (OC)

Beta	BETA.DIST(x, α, β, cum,[a], [b])
	BETA.INV(p, α, β, [a], [b])
			BETADIST(x; α; β; [a]; [b]; cum)
			BETAINV(p; α; β; a; b)

Binomial	BINOM.DIST(k, n, p, cum)
	BINOM.INV(n, p, a)
	BINOM.DIST.RANGE(n, p, k_1, [k_2])
		BINOMDIST(k, n, p, cum)
		CRITBINOM(n, p, a)
			BINOMDIST(k; n; p; cum)
			CRITBINOM(n; p; a)
			B(n; p; k1; k2)

χ2	CHISQ.DIST(x, df, cum)
	CHISQ.DIST.RT(x, df)
	CHISQ.INV(p, df)
	CHISQ.INV.RT(p, df)
			CHIDIST(x; df)
			CHISQDIST(x; df; cum)
			CHISQINV(p; df)
			CHIINV(p; df)

Empirical (Discrete)	PROB(x, p, [start], [end])	PROB(x; p; start; [end])	PROB(x; p; start; [end])
Exponential		EXPONDIST(x, λ, cum)	EXPONDIST(x; λ; cum)

F	F.DIST(x, df_1, df_2, cum)
	F.INV(p, df_1, df_2)
	F.DIST.RT(x, df_1, df_2)
	F.INV.RT(p, df_1, df_2)
		F.DIST(x, df_1, df_2, cum)
		F.DIST.RT(x, df_1, df_2)
			FDIST(x; df_1; df_2)
			FINV(p; df_1; df_2)

Gamma	GAMMA.DIST(x; α; β; cum)
	GAMMA.INV(p; α; β)
			GAMMADIST(x; α; β; cum)
			GAMMAINV(p; α; β)

Hypergeometric	HYPGEOM.DIST(x; n; M; N)	HYPGEOMDIST(x, n, M, N)	HYPGEOMDIST(x; n; M; N)

Lognormal	LOGNORM.DIST(x, μ, σ, cum)
	LOGNORM.INV((x, μ, σ)
		LOGNORMDIST(x, μ, σ)
		LOGINV(x, μ, σ)
			LOGNORMDIST(x; μ; σ)
			LOGINV(p; μ; σ;)

Normal
(Gaussian)
	NORM.DIST(x, μ, σ, cum)
	NORM.INV(p, μ, σ)
	NORM.S.DIST(x, cum)
	NORM.S.INV(p)
	GAUSS(x)
	PHI(x)
		NORMDIST(x, μ, σ, cum)
		NORMINV(x, μ, σ)
		NORMSDIST(x)
		NORMSINV(x)
			NORMDIST(x; μ; σ; cum)
			NORMINV(p; μ; σ)
			NORMSDIST(x)
			NORMSINV(p)
			GAUSS(x)
			PHI(x)

Poisson	POISSON.DIST(x, λ, cum)	POISSON(x, λ, cum)	POISSON(x; λ; cum)

T	T.DIST(x, df, cum)
	T.INV(p, df)
	T.DIST.2T(x, df)
	T.INV.2T(p, df)
	T.DIST.RT(x, df)
		TDIST(x, df, tails)
		TINV(p, df)
		T.INV.2T(p, df)
		T.INV(p, df)
			TDIST(x; df; cum)
			TINV(p; df)

Weibull	WEIBULL.DIST(x, k, λ, cum)	WEIBULL(x, k, λ, cum)	WEIBULL(x; k; λ; cum)

An interesting application of the COMBIN() function is to assess the probability of hitting the Powerball® jackpot? Five numbers are selected from the set of numbers (1, 2, …, 59) plus one number (Powerball) from the set of numbers (1, 2, …, 35). Since each of the combination of 5 numbers selected from the 59 numbers can be combined with 1 of the 35 numbers, the total number of the combinations in a single Powerball® game is =35*COMBIN(59, 5) = 175,223,510. Thus the probability in question is equal to 0.00000000570699673805187 (Letkowski, 2015, Powerball).

Although not as powerful as professional statistical programs (MINITAB, SPSS, SAS, R, etc.), the three spreadsheet programs provide significant support for applications of typical probability distributions (Table 2).

XL and OC implement a similar set of the distribution functions. GS does not support four quite important distribution functions: Beta, x^2, F-inverse, and Gamma. This deficiency can be overcome by developing custom functions written in JavaScript or by utilizing GS' ImportData() function. The latter is a powerful tool that can take advantage of HTTP services. Letkowski (2012) shows how to use this function to retrieve a x^2 critical value (inverse of the x^2 probability distribution) from the following PHP service: http://doingstats.com/srv/chsqr.php?df="&F16&"&alpha="&F17, where F16 and F17 are cell references for values of parameters df (degrees of freedom) and α (the upper tail probability). The ImportData() function is applied as follows:

= ImportData ("http://doingstats.com/srv/chsqr.php?df="&F16&"&alpha="&F17)

This technique has another important advantage. It allows for spreadsheet applications to be distributes among many world-wide Web locations while sharing data and spreadsheet functionality.

There are many probability distributions that are not supported by the three spreadsheet programs. Table 3 shows just a few unimplemented functions.

Table 3 Selected Probability Distributions That Are not Sported by XL, GS and OC

Erlang, F(x, λ, k):	$f(x)=\frac{\lambda^k x^{k-1}e^{-\lambda x}}{(k-1)!}$ $F(x)=P(X\le x)=1-\sum_{i=0}^{k-1}\frac{1}{i!}e^{-\lambda x}(\lambda x)^{i}, x\ge 0$
Triangular F(x, a, b, c):	$F(x)=P(X\le x)=\begin{cases} 0 & for \quad x<a \\ \frac{(x-a)^2}{(c-a)(b-a)} & for \quad a\le x\le b \\ 1-\frac{(c-x)^2}{(c-a)(c-b)} & for \quad b<x\le c \\ 1 & for \quad x>c \end{cases}$
Trapezoidal F(x, a, b, c,d):	$F(x)=P(X\le x)=\begin{cases} 0 & for \quad x<a \\ g\frac{(x-a)^2}{b-a} & for \quad a\le x\le b \\ g[b-a+2(x-b)] & for \quad b<x\le c \\ 1-g\frac{(d-x)^2}{d-c} & for \quad c<x\le d \\ 1 & for \quad x>d \end{cases}$ where $g=\frac{1}{d+c-a-b}$

The Erlang's random variable is the sum of k random variables having an Exponential distribution with the same parameter λ. Thus by setting k to 1, "Erlang" becomes "Exponential". Moreover, by allowing the k parameter to take on real values, the Erlang distribution transforms itself into a Gamma distribution. Spreadsheet implantations of the above distributions are provided by Letkowski (2015, Erlang, Triangular, Trapezoidal). The Erlang probabilities are calculated in a GS spreadsheet, utilizing the following JavaScript function:

```
function ErlangDist(x, k, lambda) {
  var sum = 1.0;
  var z = 1.0;
  var b = lambda * x;
  for(j=1; j<k; j++) {
  z = z * b / j;
  sum = sum + z;
  }
  return 1-Math.exp(-b) * sum;
  }
```

The Triangular probabilities are crunched in an OC spreadsheet, utilizing the following formula:

```
=IF(_x<_a;0;
  IF(_x<=_b;(_x-_a)^2/((_c-_a)*(_b-_a));
    IF(_x<=_c;1-(_c-_x)^2/((_c-_a)*(_c-_b));1)))
```

Finally, the Trapezoidal probabilities are calculated in an Excel spreadsheet via the following formula:

```
=IF(_x<_a,0,
  IF(_x<=_b,_h * (_x - _a) ^ 2 / (_b - _a),
    IF(_x<=_c,_h*(_b-_a+2*(_x-_b)),
       IF(_x<=_d,1-_h*(_d-_x)^2/(_d-_c),1))))
```

References _a, _b, _c, _d, and _x represent named cells, containing values of the parameters of the distributions.

As shown above, there is plenty of computational power and diversified techniques for assessing probabilities in spreadsheets. There is also plenty of room for sharing and reusing already developed applications.

3. Spreadsheet Applications for Descriptive Statistics

Descriptive Statistic plays a very important role in solving problems within uncertain situations. Before one can reason and make decisions about such situations, using data that describe the situations, it is usually necessary to organize and/or process the data in order to provide richer, more convenient and more refined characteristics of the relevant aspects of the situations.

Table 4 shows spreadsheet functions for computing most frequently used summary measures. There are a few syntactical differences between some functions implemented by the three spreadsheet programs (XL, GS, and OC). Nonetheless most of the measures can be generated by the programs.

Table 4 Spreadsheet Functions of Summary Measures

Function	Excel	Google Sheets	OpenOffice Calc
Average of the absolute deviations from their mean	AVEDEV(x)	AVEDEV(x)	AVEDEV(x)
Average	AVERAGE(x)	AVERAGE(x)	AVERAGE(x)
Average, including numbers, text, and logical values	AVERAGEA(x)	AVERAGEA(x)	AVERAGEA(x)
Average of values that meet a given criteria	AVERAGEIF (xc, cx, [x])	AVERAGEIF (cx, xc, [x])	AVERAGEIF (cx, xc, [x])
Average of values that meet multiple criteria	AVERAGEIFS (xc1, cx1, [x1], […])	AVERAGEIFS (xc1, cx1, [x1], […])	AVERAGEIFS (xc1, cx1, [x1], […])
Correlation coefficient	CORREL(x,y)	CORREL(x,y)	CORREL(x,y)
Numeric count	COUNT(x)	COUNT(x)	COUNT(x)
Value (alpha-numeric) count	COUNTA(αx)	COUNTA(αx)	COUNTA(αx)
Blank cell count	COUNTBLANK(αx)	COUNTBLANK(αx)	COUNTBLANK(αx)
Counts of values that meet the given criteria	COUNTIF(αx, crit)	COUNTIF(αx, crit)	COUNTIF(αx, crit)
Counts of values that meet multiple criteria	COUNTIFS(αx1,crit1,[…])	COUNTIFS(αx1, crit1, […])	COUNTIFS (αx1,crit1,[…])
Counts unique values			COUNTUNIQUE (αnx)
Population covariance	COVARIANCE.P(x, y)	COVAR(x, y)	COVAR(x, y)
Sample covariance	COVARIANCE.S(x, y)		
Correlation coefficient	CORREL(x, y)	CORREL(x, y)	CORREL(x, y)
Sum of squares of deviations	DEVSQ(x)	DEVSQ(x)	DEVSQ(x)
Frequency distribution as a vertical array	FREQUENCY(x, bin)	FREQUENCY(x, bin)	FREQUENCY(x, bin)
Geometric mean	GEOMEAN(x)	GEOMEAN(x)	GEOMEAN(x)
Harmonic mean	HARMEAN(x)	HARMEAN(x)	HARMEAN(x)
Kurtosis of a data set	KURT(x)	KURT(x)	KURT(x)
k-th largest value	LARGE(x, k)	LARGE(x,k)	LARGE(x, k)
Maximum value	MAX(x)	MAX(x)	MAX(x)
Maximum value, including numbers, text, and logical values	MAXA(αx)	MAXA(αx)	MAXA(αx)
Median	MEDIAN(x)	MEDIAN(x)	MEDIAN(x)
Minimum value	MIN(x)	MIN(x)	MIN(x)
Smallest value, including numbers, text, and logical values	MINA(αx)	MINA(αx)	MINA(αx)
Array of the most frequently occurring, or repetitive values	MODE.MULT(x)		
Most common value in a data set	MODE.SNGL(x)	MODE(x)	MODE(x)
k-th percentile of values in a range, where k is in the range 0..1, exclusive	PERCENTILE.EXC(x,p)		
k-th percentile of values in a range	PERCENTILE.INC(x, p)	PERCENTILE(x, p)	PERCENTILE(x, p)
Rank of a value in a data set as a percentage (0..1, exclusive)	PERCENTRANK.EXC (x, v, [sd])	PERCENTRANK.EXC (x, v, [sd])	
Percentage rank of a value	PERCENTRANK.INC (x, v, [sd])	PERCENTRANK.INC (x, v, [sd])	PERCENTRANK(x, v)
Quartile of the data set, based on percentile values from 0..1, exclusive	QUARTILE.EXC(x,k)		
Quartile of a data set	QUARTILE.INC(x,k)	QUARTILE(x, k)	QUARTILE(x, k)
Rank of a number in a list of numbers with average option	RANK.AVG(v, x, [0|1])	RANK.AVG(v, x, [0|1])	
Rank of a number in a list of numbers with top option	RANK.EQ(v, x, [0|1])	RANK.EQ(v, x, [0|1])	RANK(v, x, [0|1])

Sample skewness	SKEW(x)	SKEW(x)	SKEW(x)
Population skewness	SKEW.P(x)		
k-th smallest value	SMALL(x, k)	SMALL(x, k)	SMALL(x, k)
Normalized value	STANDARDIZE (v, μ, σ)	STANDARDIZE (v, μ, σ)	STANDARDIZE (v,μ,σ)
Population standard deviation	STDEV.P(x)	STDEVP(x)	STDEVP(x)
Sample standard deviation	STDEV.S(x)	STDEV(x)	STDEV(x)
Sample standard deviation, including numbers, text, and logical values	STDEVA(αx)	STDEVA(αx)	STDEVA(αx)
Population standard deviation, including numbers, text, and logical values	STDEVPA(αx)	STDEVPA(αx)	STDEVPA(αx)
Mean of the interior of a data set	TRIMMEAN(x, p)	TRIMMEAN(x, p)	TRIMMEAN(x, p)
Population variance	VAR.P(x)	VARP(x)	VARP(x)
Sample variance	VAR.S(x)	VAR(x)	VAR(x)
Sample variance, including numbers, text, and logical values	VARA(αx)	VARA(αx)	VARA(αx)
Population variance, including numbers, text, and logical values	VARPA(αx)	VARPA(αx)	VARPA(αx)
Legend:			
αx	alpha-numeric range	k	integer value
x	numeric range	v	numeric value
xc	numeric range for criteria	crit	criterion
cx	criteria for numeric range	p	percentage value
bin	class interval limits	sd	number of sig. digits

Detail syntax and interpretation of the functions are well explained in the reference (help) documents of the programs. The level of the functional support of three spreadsheets with respect to the summary measures is close to identical. XL functions COVARIANCE.S(), MODE.MULT(), PERCENTILE.EXC() and SKEW.P() are unique. GS and OC do not implement them. Google's COUNTUNIQUE() does not have equivalent functions in XL or OC.

The programs are flexible enough to make up for the differences. Both GS and OC use function COVAR to calculate the population covariance. In order to calculate the sample covariance in GS and OC the value of the population covariance must be multiplied by n /(n-1), where n is the sample size. Thus = COVARIANCE.S(X, Y) in XL returns the same as = COVAR(X, Y)*COUNT(X)/(COUNT(X)-1) in GS and OC. In order to count unique numeric values in XL or OC one can count non zero outcomes of the FREQUENCY() function in which the bin range is the same as the sample: = FREQUENCY(x, x). Examples of such solutions are provided in (Letkowski, 2015, Summary Measures).

More interesting are serious limitations of some of the basic functions. It turns out that the MODE() function, in whichever shape or form, does not return the modal value as expected from its probabilistic definition, according to which the mode is a value that maximizes the density function. Letkowski (2014, A) shows many examples of incorrect assessment of this important summary measure by the spreadsheet MODE() function. A correct way of estimating the mode for continues data or for grouped-discrete data should be based not directly on a sample but on the frequency distribution derived from the sample (Krysicki, 2006, p. 17):

$$(b_{k-1}, b_k] : fn_k = \max_i \{fn_i\} \Rightarrow \text{mode} = b_{k-1} + w\frac{fn_k - fn_{k-1}}{(fn_k - fn_{k-1}) + (fn_k - fn_{k+1})}$$

Where b_k is the right limit of the k-th class interval, w is the interval width, and fn_i is the absolute frequency of the i-th interval. If the maximum is unique then the $(b_{k-1}, b_k]$ interval is referred to as the modal class interval. If the frequencies immediately to the left and to the right of the modal interval are identical, then the midpoint, mx_k = (b_{k-1} + b_k)/2, of the modal interval becomes the mode. If the data domain (population) is bottom unbounded then the absolute frequencies can be computed reliably by the FREQUENCY() function. Otherwise the COUNTIF() function must be employed (Letkowski, 2014, B). Spreadsheet examples of correct derivations of the modal value for a sample drawn from a continuous population are shown in (Letkowski, 2015, Mode).

4. Spreadsheet Applications for Inferential Statistics

Applications of inferential statistics depend heavily on probability distributions. The three spreadsheet programs (XL, GS and OC) are equipped with built-in probability functions (Table 2) for all basic probability distributions used in introductory Statistics courses. As shown in the Spreadsheet Applications for Probability section, GS does not support some of the basic probability functions but it can import such functions via HTTP services. Common inferential tasks, such as hypothesis tests, that utilize critical values (percentiles) or probability values (pv) can be setup using one the following probability distributions: Normal, Student-t, x^2, F. As shown above, these distributions are supported directly or indirectly by XL, GS, and OC. The following examples show how to obtain critical values (zcrit) and probability values (pv) for these distribution with at a significance level α.

The Normal Distribution (z_s is a value of the standardized test statistic)

Left-Tail:

XL: z_{crit} =Norm.S.Inv(α), p_v =Norm.S.Dist(z_s,True)

GS, OC: z_{crit} =NormSInv(α), p_v =NormSDist(z_s)

Two-Tail:

XL: z_{crit} =Abs(Norm.S.Inv(α/2)), p_v =2*Norm.S.Dist(-Abs(z_s),True)

GS, OC: z_{crit} =Abs(NormSInv(α/2)), p_v =2*NormSDist(-Abs(z_s))

Right-Tail:

XL: z_{crit} =Norm.S.Inv(1-α), p_v =1-Norm.S.Dist(z_s,True)

GS, OC: z_{crit} =NormSInv(1-α), p_v =1-NormSDist(z_s)

Tests requiring the Normal distribution can be conducted in all three spreadsheet programs. One has to pay attention to the minor syntactical differences (as shown above). Examples for one-sample Z-tests are provided in (Letkowski, 21015, Hypothesis Testing for the Mean).

Additionally, function ZTest() can be used to find out the p-value of a Z test. It returns the p-value (p_v) for the right-tail test of the mean. The following examples show how to use it for any tail:

Left-Tail: p_v = 1-ZTEST(x, μ_0, σ)

Two-Tail: p_v = 2*(1-ZTEST(x, μ_0, σ))

Right-Tail: p_v = ZTEST(x, μ_0, σ)

Parameters x and μ_0 stand for the sample range and hypothesized mean, respectively. In XL, identical results can also be obtained, using function Z.TEST().

T-tests are also well supported by the three spreadsheet programs. They involve the Student-t distribution.

The Student-*t* Distribution (t_s is a value of the standardized test statistic)

Left-Tail:

XL: t_{crit} = T.Inv(α,df), p_v = T.Dist(t_s,df,True)

GS: t_{crit} = T.Inv(α,df), p_v = TDist(-t_s,df,True)

OC: t_{crit} = -TINV(2* α;df), p_v = TDIST(-t_s; df; 1))

Two-Tail:

XL: t_{crit} = T.Inv.2T(α, df), p_v = T.DIST.2T(ABS(t_s), df)

GS: t_{crit} = T.Inv.2T(α, df), p_v = TDIST(ABS(t_s), df, 2)

OC: t_{crit} = TINV(α; df), p_v = TDIST(-t_s; df; 2)

Right-Tail:

XL: t_{crit} = T.INV(1-α, df), p_v = 1-T.Dist(t_s, df, True)

GS,: t_{crit} = T.INV(1-α, df), p_v = TDIST(t_s, df, 1)

OC: t_{crit} = TINV(2*α, df), p_v = TDIST(t_s; df; 1)

It is imperative to pay attention to many syntactical differences (as shown above). Examples for one-sample *t*-test are provided in (Letkowski, 2015, Hypothesis Testing for the Mean).

In order to run tests for differences between means of two populations, function T-Test (x_1, x_2, tails, type) can be employed, where x_1, x_2 are sample ranges, tails equals to "1" or "2", and type can be "1–paired", "2–populations with equal variances", and "3–populations with unequal variances". Examples for two-sample t-tests are provided in Letkowski (2015, Comparing Means of Two Populations).

Tests for variances utilize the F distribution which is fully supported by XL and OC. Surprisingly, GS does not have a function to calculate the inverse of the F distribution.

The F Distribution (*f*$_s$ is a value of the test statistic)

Left-Tail:

XL: f_{crit} = F.INV(α, df_1, df_2), p_v = F.DIST(f_s, df_1, df_2, True)

GS: f_{crit} = N/A p_v = F.DIST(f_s, df_1, df_2, True)

OC: t_{crit} = FINV(1-α; df_1; df_2), p_v = 1-FDIST(f_s; df_1; df_2)

Two-Tail:

XL: $f_{L\text{-}crit}$ = F.INV(α/2, df_1, df_2), $f_{U\text{-}crit}$ = F.INV(1-α/2, df_1, df_2)

p_v = IF(σ_1<σ_2,2*F.DIST(f_s, n_1-1, n_2-1, 1), 2*F.DIST.RT(f_s,n_1-1, n_2-1))

GS: f_{crit} = N/A

p_v = IF(σ_1<σ_2, 2*F.DIST(f_s,n_1-1,n_2-1,1), 2*F.DIST.RT(f_s, n_1-1, n_2-1))

OC: $f_{L\text{-}crit}$ = FINV(1-α/2; df_1; df_2), $f_{U\text{-}crit}$ = FINV(α/2; df_1; df_2)

p_v = IF(σ_1<σ_2, 2*(1-FDIST(f_s, n_1-1, n_2-1)), 2*FDIST(f_s, n_1-1, n_2-1))

Right-Tail:

XL: f_{crit} = F.INV(1-α, df_1, df_2), p_v = F.DIST.RT(f_s, df_1, df_2)

GS,: f_{crit} = N/A p_v = F.DIST.RT(f_s, df_1, df_2)

OC: f_{crit} = FINV(α, df_1, df_2), p_v = FDIST(f_s; df_1; df_2)

If data samples are provided, the p-value (p_v) can also be calculated for two-tail tests, using function F.Test(x_1, x_2), where x_1,x_2 are the sample ranges. Notice that some textbooks (Levine, 2013, pp. 369-372) do not provide the left tail option, working with the assumption that the test static (f_s) is not less than 1. F-test decisions must be made with GS only based on the p-value. Examples of F tests are shown in (Letkowski, 2015, F-Test).

Goodness-of-fit and independence tests are conducted, using the x^2 distribution. This distribution is well supported by XL and OC.

The x^2 Distribution (x_s is a value of the test statistic)

Left-Tail:

XL: χ_{crit} = CHISQ.INV(α, df), p_v = CHISQ.DIST(x^2_s, df, True)

GS: N/A

OC: χ_{crit} = CHISQINV(α; df), p_v = CHISQDIST(x^2_s, df)

Two-Tail:

XL: $f_{L\text{-}crit}$ = CHISQ.INV(α/2, df), $f_{U\text{-}crit}$ = CHISQ.INV(1-α/2, df)

GS: N/A

OC: $f_{L\text{-}crit}$ = CHISQINV(α/2; df), $f_{U\text{-}crit}$ = CHISQINV(1-α/2; df)

Right-Tail:

XL: $\mathbf{x}_{crit}$ = CHISQ.INV(1-α, df), p_v = CHISQ.DIST.RT(x^2_s, df)

GS: N/A

OC: $\mathbf{x}_{crit}$ = CHIINV(α; df), p_v = CHIDIST(x^2_s, df)

Typically, the right-tail test critical values and probabilities (p_v) are used to test variable independence, goodness of fit, etc. Again, it is always advisable to check the up-to-date references in order to apply appropriate syntax. Examples for x^2 tests are shown in (Letkowski, 2015, Chi-Square Test). Notice that the x^2_s critical value is calculated in GS, using the ImportData() function shown above (in section 2). In XL and OC, the p-value for independence tests can also be calculated by means of the CHITEST(af_1, of_2) function, where af_1 and of_2 are actual and observed frequencies, respectively.

5. Conclusion

This paper was set to explore only the function based solution to solving typical statistical problems. In addition to a rich collection of built-in functions, presented in this paper, the spreadsheet programs provide command or macro based opportunities. It is important to note that macro-based solutions can take advantage of both local and external (network) resources. GS makes the external access particularly easy by means of the ImportData() function presented above.

Arguably, spreadsheets represent the most commonly used software in business. This is why there are many other software solutions that attempt to work with rather than compete against spreadsheets. Many statistical textbooks select Excel enhanced by add-ins that fill gaps that exist in the common spreadsheet programs. Professional software developers have also recognized spreadsheet opportunities. Oracle (2015) provides and add-in for Excel to enable data import and export from and to MySQL. Probably the most noteworthy statistical extension of Excel is provided by an R based add-in, RExcel (Heiberger, 2009). As an open-source statistical package, R is becoming more and more popular "…thanks to a boost from big data application development…" (Krill, 2014). Currently, home and academic edition of RExcel is only available for a 32-bit installation of Excel. Spreadsheet programs and R should be considered as application programs complementing one another even if they are not tightly integrated. Deficiencies of the spreadsheet programs with respect to graphics and to handling more sophisticated statistical tests and procedures can be compensated by R. An excellent coverage of R that goes along with intuitive but rigorous introduction to the probability theory is provided by Baclawski (2008).

References

Anderson D. R., Sweeney D. J., Thomas A. and Williams T. A. (2011). *Essentials of Modern Business Statistics with Microsoft Excel* (5th ed.), Cengage Learning.

Apache (2015). "OpenOffice Calc", available online at: https://www.openoffice.org/product/calc.html.

Baclawski K. (2008). *Introduction to Probability with R. Chapman & Hall/CRC*, Taylor & Francis Group.

Beal V. (2012). "5 free open source alternatives to Microsoft Office", *PCWorld*, September 17, 2012, available online at: http://www.pcworld.com/article/2010005/5-free-open-source-alternatives-to-microsoft-office.html.

Black K. (2011). *Business Statistics: For Contemporary Decision Making* (7th ed.), John Wiley and Sons, Inc..

Doane D. and Seward L. (2010). *Applied Statistics in Business and Economics* (3rd ed.), Mcgraw-Hill/Irwin.

Donnelly R. (2014). *Business Statistics* (2nd ed.), Pearson.

Dretzke B. (2011). *Statistics with Microsoft Excel* (5th ed.), Pearson.

Google (2015). "Google sheets", available online at: http://www.google.com/sheets/about/.

Heiberger R. M. and Neuwirth E. (2009). *R Though Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics*, Springer.

Jaggia S. and Kelly A. (2012). *Business Statistics: Communicating with Numbers* (1st ed.), McGraw-Hill/Irwin.

Krill P. (2014). "R, Swift soar in language search popularity in 2014", *InfoWorld*, Dec 9, 2014, available online at: http://www.infoworld.com/article/2857133/application-development/r-swift-soar-in-language-popularity-in-2014.html.

Krysicki W., Bartos J., Dyczka W., Królikowska K. and Wasilewski M. (2006). *Rachunek Prowdopodobieństwa i Statystyka Matematyczna w Zadaniach*, Warszawa, Wydawnictwo Naukowe PWN.

Larose D. T. (2010). *Discovering Statistics*, W. H. Freeman and Company.

Levine D. M., Stephan D. L. and Szabat K. A. (2013). *Statistics for Managers Using Microsoft Excel* (7th ed.), Pearson Education, Inc., January 21, 2013.

Letkowski J. (2012). "Developing poisson probability distribution applications in a cloud", *AABRI Journal of Case Research in Business and Economics*, Vol. 5, available online at: http://www.aabri.com/jcrbe.html.

Letkowski J. (2014, A). "In search of the most likely value", *AABRI Journal of Case Studies in Education*, Vol. 5, available online at: http://www.aabri.com/jcse.html.

Letkowski J. (2014, B). "A spreadsheet based derivation of the probability distribution from a random sample", *AABRI Journal of Business Cases and Applications*, Vol. 11, available online at: http://www.aabri.com/jbca.html.

Letkowski J. (2015). "Resource references for this paper", available online at: http://letkowski.us/pub/conference/sobie/2015/sandestin/.

Letkowski J. and Letkowski N. (2015). "Learning problem solving with Excel, Access and MySQL", in: *Academic and Business Research Institute Conference*, Savannah, March 26-28, 2015.

McClave J. T., Benson P. G. and Sincich T. T. (2012). *Statistics for Business and Economics* (12th ed.), Pearson.

Oracle (2015). "MySQL for Excel". available online at: https://www.mysql.com/why-mysql/windows/excel/.

Pelosi M. T., Sandifer T. M. and Letkowski J. (1996). *Doing Statistics with Excel for Windows Version 5.0: An Introductory Course Supplement for Explorations in Data Analysis*, John Wiley & Sons, Inc..

Pelosi M., T. Sandifer T. M. and Letkowski J. (1998). *Doing Statistics with Excel 97: Software Instruction and Exercise Activity Supplement*, John Wiley & Sons, Inc..

Pelosi M. T. and Sandifer T. M. (2003). *Elementary Statistics: From Discovery to Decision*, John Wiley and Sons, Inc..

Salkind N. J. (2012). *Excel Statistics: A Quick Guide* (2nd ed.), SAGE Publications, Inc..

Sharpe N. D., De Veaux R. D. and Velleman P. (2011). *Business Statistics* (2nd ed.), Pearson.

Triola M. F. (2013). *Elementary Statistics Using Excel* (5th ed.), Pearson.