Journal of Statistical Software

June 2018, Volume 85, Code Snippet 1. *doi: 10.18637/jss.v085.c01*

QXLA: Adding Upper Quantiles for the Studentized Range to Excel for Multiple Comparison Procedures

K. Thomas Klasson
United States Department of Agriculture

Abstract

Microsoft **Excel** has some functionality in terms of basic statistics; however it lacks distribution functions built around the studentized range (Q). The developed **Excel** add-in introduces two new user-defined functions, `QDISTG` and `QINVG`, based on the studentized range Q-distribution that expands the functionality of **Excel** for statistical analysis. A workbook example, demonstrating the Tukey, S-N-K, and REGWQ tests, has also been included. Compared with other options available, the method is fast with low error rates.

Keywords: Microsoft **Excel**, VisualBasics, VBA, add-in, studentized range, Q-distribution.

1. Introduction

Researchers in most fields of science are often faced with comparing means obtained with various treatments. The statistical analysis using multiple comparison procedures (MCPs) is often carried out by statistical software packages that are less suitable for data storage and manipulation. While Microsoft **Excel** offers excellent data storage and manipulation, it provides few built-in methods for statistical data analysis. **Excel**'s **Analysis ToolPak** add-in provides basic analysis of variance (single factor and two-factor) and also provides t-tests for two sample means. The availability of **Excel** functions such as `FDIST`, `FINV`, `TDIST`, and `TINV` for the F- and t-distributions allows for some capability to conduct simple multiple comparison procedures such as Fisher's least significant difference test. However, the lack of **Excel** support for the studentized range (Q) distribution does not allow for the Tukey honest significant difference, Student-Newman-Keuls (S-N-K), or Ryan-Einot-Gabriel-Welsch Q (REGWQ) tests to be carried out.

1.1. Brief overview of multiple comparison procedures

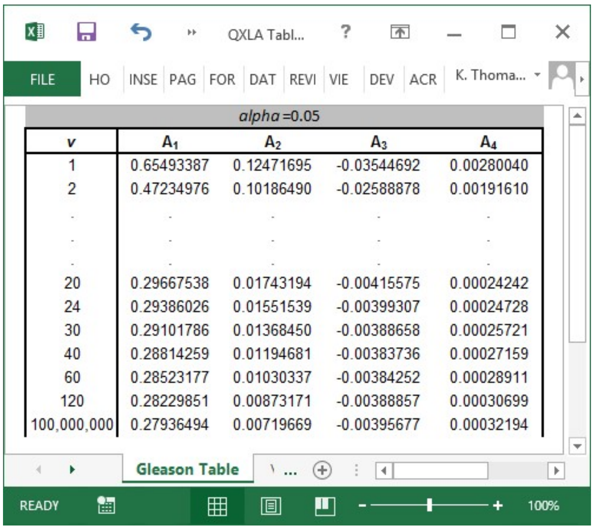
Comparisons of means of groups receiving different treatments often begin with a simple analysis of variance (ANOVA) followed by post-hoc analysis or MCP to determine which means are statistically different. The methods by which these tests are performed differ depending on factors such as planned versus unplanned comparisons, comparisons between groups of the same or different sizes, comparisons of groups with or without equal variances, and parametric versus stepwise comparisons. However, a common denominator is the use of statistical distributions such as the F-, t-, or Q-distributions. A comprehensive review of different methods was performed by Day and Quinn (1989) who described, provided equations for, and evaluated many of the commonly used methods. It should be noted that Day and Quinn recommended MCP methods that use the Q-distribution such as the Tukey, Ryan's Q (here expanded to REGWQ), and Games-Howell tests. As formulas for this distribution are not available in **Excel**, an add-in was constructed to provide the feature. As part of this manuscript, we also constructed an **Excel** workbook to demonstrate the use of the add-in with the Tukey, S-N-K, and REGWQ tests. Within this manuscript, we will use the term α to represent the upper percentile of the studentized range distribution; α is typically in the range of 0.1 to 0.01 in comparisons. We will use r to denote the total number of groups compared and v (or *df* in the VBA code) to denote the degrees of freedom within groups (available in the standard single factor ANOVA table). It should be noted that many statistical tables of the studentized range quantiles use one minus α (often denoted by p) to identify a table. However, as the upper percentile (α) is used in several other **Excel** functions (e.g., `FINV` and `TINV`), α was chosen as parameter for the user-defined formulas.

1.2. Methods for obtaining Q values and probabilities

As with many statistical properties related to distributions, statistical tables based on studentized ranges have long been part of statistical text books (Walpole and Myers 1989; Snedecor and Cochran 1967) but they tend to be limited to a few probabilities (e.g., $p = 0.95$, $\alpha = 0.05$). For an extensive set of Q tables, the reader is directed to Harter (1960).

Lund and Lund (1983) developed a numerical integration algorithm (AS 190) that could be used to estimate Q values for α values of 0.10–0.01 and also included a rough estimate algebraic algorithm (AS 190.2) for α values of 0.20–0.05. Copenhaver and Holland (1988) developed an algorithm in Fortran using Gauss-Legrendre quadrature that was later implemented in Pascal by Ferreira, Demetrio, Manly, and Machado (2007). It is the same algorithm used by the R environment for statistical computing and graphics (R Core Team 2018, C source code is freely available) and other statistical software packages. It is likely also the algorithm used by of the **Excel** add-in **RealStats-2007**, which is freely available (Zaiontz 2016). This last algorithm is computationally intensive.

The method described within this manuscript takes a different approach. It uses a method to calculate Q values proposed by Gleason (1998; 1999) that built on relationships between Student t quantiles and studentized range quantiles. Using this method, Gleason (1999) transformed traditional studentized range tables (Q tables) to a new set of tables (one for each probability). Each table listed four constants (a_1, a_2, a_3, a_4) for each degree of freedom (v). These constants could be used in a fourth-order polynomial with the parameter r to calculate the value of $Q(\alpha, r, v)$. Eight tables (corresponding to eight probabilities 0.50, 0.75, 0.90, 0.95, 0.975, 0.99, 0.995, and 0.999) were created by Gleason (see example in Figure 1) and,



alpha =0.05

v	A_1	A_2	A_3	A_4
1	0.65493387	0.12471695	-0.03544692	0.00280040
2	0.47234976	0.10186490	-0.02588878	0.00191610
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20	0.29667538	0.01743194	-0.00415575	0.00024242
24	0.29386026	0.01551539	-0.00399307	0.00024728
30	0.29101786	0.01368450	-0.00388658	0.00025721
40	0.28814259	0.01194681	-0.00383736	0.00027159
60	0.28523177	0.01030337	-0.00384252	0.00028911
120	0.28229851	0.00873171	-0.00388857	0.00030699
100,000,000	0.27936494	0.00719669	-0.00395677	0.00032194

Figure 1: Table of fourth order polynomial constants by Gleason (1999) for calculation of Q values. The above example is for $p = 0.95$, $\alpha = 0.05$. Entries for some values of v have been deleted for brevity. The last value of v is listed as 100,000,000; in the typical text book table it is listed with an infinity symbol.

while not all possible values of v and α were included, details were provided for interpolation mechanisms (Gleason 1999). Even accurate extrapolation was deemed possible: at least to $\alpha = 0.0001$ and $r = 200$, and the approach was included in the statistical software program Stata (Gleason 1998). The method by Gleason has also been written in Python with an expanded set of polynomial constants (Lew 2011). Briefly, the method used by Gleason and in this add-in to calculate Q values was:

1. If $r = 2$, set $y = 1$ and goto step 12.
2. Select two data lines from all eight Gleason tables corresponding to v's higher and lower (v_{high}, v_{low}) than the desired v. Each line will have four a constants.
3. If $r = 3$, calculate an adjustment term (Adj) as recommended by Gleason. For detailed equations, please consult Gleason (1999).
4. Calculate y^2_{low} and y^2_{high} for each of the eight α values using the appropriate four constants (a_1, a_2, a_3, a_4) and any adjustment due to $r = 3$. $y^2_{\text{low/high}} = (1 + a_1 \cdot \texttt{LOG}(1-r) + a_2 \cdot [\texttt{LOG}(1-r)]^2 + a_3 \cdot [\texttt{LOG}(1-r)]^3 + a_4 \cdot [\texttt{LOG}(1-r)]^4 + Adj)^2$. Note that `LOG` is an VBA function for the natural logarithm.
5. Interpolate y^2 from y^2_{low} and y^2_{high} and take the square root to find y for the desired v. This will yield eight y-values, one for each α. $y^2 = y^2_{\text{low}} + (1/v - 1/v_{\text{low}}) \cdot (y^2_{\text{high}} - y^2_{\text{low}})/(1/v_{\text{high}} - 1/v_{\text{low}})$.
6. Calculate x-values as $x = -1/[1 + 1.5 \cdot \texttt{NORMINV}(1 - \alpha/2, 0, 1)]$. `NORMINV` is an **Excel** function corresponding to the Z_p-function by Gleason (1999). This will yield eight x-values, one for each α.

7. Calculate z-values as $z = \texttt{LOG}(y + r/v)$. This will yield eight z-values, one for each α.

8. Assume that the variable z is dependent on x through a 2nd order polynomial, $z = m_1 + m_2 \cdot x + m_3 \cdot x^2$, or a 4th order polynomial for higher accuracy if $0.5 < \alpha < 0.001$. Gleason (1999) only used 2nd order.

9. Find the values of the constants (m_1, m_2, etc.) in the polynomial through matrix algebra.

10. Calculate x-value for the desired α (see step 6) and use the polynomial (see step 8) to find its z-value.

11. From the z-value, calculate the y-value from relationship in step 7.

12. Calculate the final Q value as $Q = y \cdot \texttt{SQR}(2) \cdot \texttt{TINV}(\alpha, v)$. Note that `SQR` is a VBA function and `TINV` is an **Excel** function.

The method of obtaining the probability (or α value) for a specific Q value with given r and v was not given by Gleason (1998; 1999). For this manuscript, an iterative process was developed based on a false position bracketing iterative interpolation technique for fast guaranteed convergence (Chapra and Canale 1985). Also, as Gleason reported that his method could be used to extrapolate to α values lower than 0.001, we extrapolated calculations to include $\alpha = 0.0001$, 0.00001, and 0.000001. This allowed for an expanded range when seeking probabilities for a given Q. The following procedure was used:

1. If $r = 2$, set $\alpha = \texttt{TDIST}(QVal/\texttt{SQR}(2), v, 2)$ and stop. Note that `TDIST` is an **Excel** function.

2. Follow steps 2–9 in the previous section. Only use 2nd order polynomial in step 8.

10. Define three additional α values equal to 0.0001, 0.00001, and 0.000001.

11. For each of the additional α values, calculate a temporary x-value and use the polynomial (see steps 8 and 9) to find a temporary z-value.

12. From the z-value, calculate the y-value (equation in step 7) for each of the three additional α values.

13. Calculate w-values as $w = y \cdot \texttt{SQR}(2) \cdot \texttt{TINV}(\alpha, v)$. This will yield eleven w-values, one for each α. Note that the w-values are the same as Q values, one for each α.

14. Find the α values that bracket the sought α based on the desired Q value and w-values.

15. Use false position bracketing iterative interpolation technique to find α corresponding to the desired Q value with an acceptable error of 0.1%.

1.3. Creating algorithms and an add-in in Excel

The **Excel** add-in was created using the Microsoft **Excel** 2013 built-in VisualBasics module. In addition to a small subroutine for loading Gleason tables into string variables, the functions `QINVG` and `QDISTG` were created as user-defined functions. They were so named to differentiate them from the user-defined functions (`QINV` and `QDIST`) in the **Excel** add-in **RealStats-2007** (Zaiontz 2016) and to recognize Gleason's contribution to this work (Gleason 1998, 1999). The code of the add-in is provided as part of this manuscript and has been annotated to correspond to the step-by-step methods listed above. After the code had been entered and tested, the spreadsheet was saved as an **Excel** 97–2003 add-in (`*.xla`).

1.4. User installation

The process of installation and activation of add-ins for Microsoft **Excel** is covered in **Excel**'s help files with slight differences between **Excel** versions. Examples for the add-in installation process for **Excel** 2007 and earlier have also been published by Buttrey (2009). The add-in presented within this manuscript has been tested for **Excel** versions 2003 through 2013.

2. Code verification, accuracy, and limitations

To verify that the `QINVG` user-defined function performed as expected, a limited set of conditions is presented here. To correspond to standard tables, `QINVG`($\alpha = 0.05$, $r = 5$, 10, 20, $v = 1(1)20$, 24, 30, 40, 60, 120, 99,999,999), where 1(1)20 is shorthand for 1, 2, 3, 4, 5, etc. to 20, was evaluated and compared to statistical tables (Harter 1960). The values were also generated by the R environment for statistical computing and graphics, version 3.3.0 (R Core Team 2018), using the following code:

```
R> QTable <- expand.grid(df = c(1:20, 24, 30, 40, 60, 120, Inf),
+    groups = c(5, 10, 20), alpha = 0.05)
R> QTable$QVal <- qtukey(1 - QTable$alpha, QTable$groups, QTable$df)
R> QTable
```

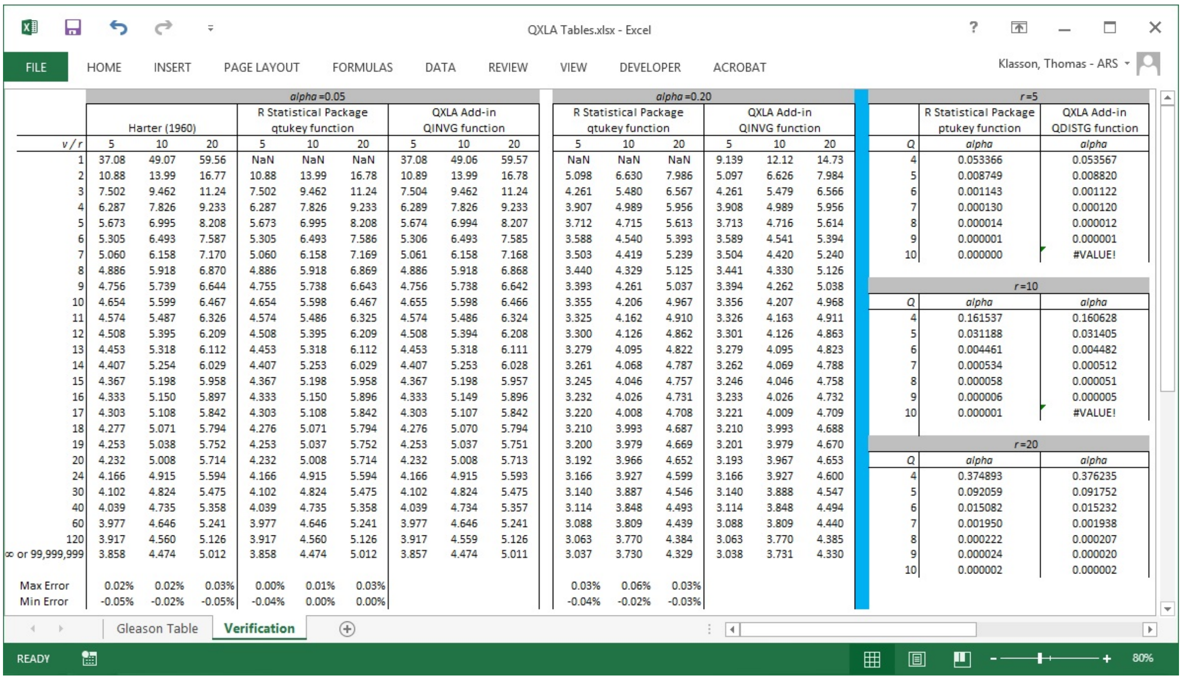
To test the α interpolation capability, `QINVG`($\alpha = 0.2$, $r = 5$, 10, 20, $v = 1(1)20$, 24, 30, 40, 60, 120, 99,999,999) was evaluated with the **QXLA** add-in and with R using the following code:

```
R> QTable <- expand.grid(df = c(1:20, 24, 30, 40, 60, 120, Inf),
+    groups = c(5, 10, 20), alpha = 0.20)
R> QTable$QVal <- qtukey(1 - QTable$alpha, QTable$groups, QTable$df)
R> QTable
```

As is noted, there is very good agreement with only minor deviations from expected values (Figure 2). The R implementation failed to provide Q values for $v = 1$, something also noted by Gleason (1999). Some of the Q values may have little practical use in MCPs where it would be unusual if $v < r$ in $Q(\alpha, r, v)$.

To verify that the `QDISTG` user-defined function performed as expected, `QDISTG`($QVal = 4(1)10$, $r = 5$, 10, 20, $v = 40$) was evaluated with the **QXLA** add-in and with R using the following code:

```
R> ATable <- expand.grid(df = 40, qvalue = seq(4, 10, by = 1),
+    groups = c(5, 10, 20))
R> ATable$AVal <- 1 - ptukey(ATable$qvalue, ATable$groups, ATable$df)
R> ATable
```



	alpha =0.05								
	Harter (1960)			R Statistical Package qtukey function			QXLA Add-in QINVG function		
v / r	5	10	20	5	10	20	5	10	20
1	37.08	49.07	59.56	NaN	NaN	NaN	37.08	49.06	59.57
2	10.88	13.99	16.77	10.88	13.99	16.78	10.89	13.99	16.78
3	7.502	9.462	11.24	7.502	9.462	11.24	7.504	9.462	11.24
4	6.287	7.826	9.233	6.287	7.826	9.233	6.289	7.826	9.233
5	5.673	6.995	8.208	5.673	6.995	8.208	5.674	6.994	8.207
6	5.305	6.493	7.587	5.305	6.493	7.586	5.306	6.493	7.585
7	5.060	6.158	7.170	5.060	6.158	7.169	5.061	6.158	7.168
8	4.886	5.918	6.870	4.886	5.918	6.869	4.886	5.918	6.868
9	4.756	5.739	6.644	4.755	5.738	6.643	4.756	5.738	6.642
10	4.654	5.599	6.467	4.654	5.598	6.467	4.655	5.598	6.466
11	4.574	5.487	6.326	4.574	5.486	6.325	4.574	5.486	6.324
12	4.508	5.395	6.209	4.508	5.395	6.209	4.508	5.394	6.208
13	4.453	5.318	6.112	4.453	5.318	6.112	4.453	5.318	6.111
14	4.407	5.254	6.029	4.407	5.253	6.029	4.407	5.253	6.028
15	4.367	5.198	5.958	4.367	5.198	5.958	4.367	5.198	5.957
16	4.333	5.150	5.897	4.333	5.150	5.896	4.333	5.149	5.896
17	4.303	5.108	5.842	4.303	5.108	5.842	4.303	5.107	5.842
18	4.277	5.071	5.794	4.276	5.071	5.794	4.276	5.070	5.794
19	4.253	5.038	5.752	4.253	5.037	5.752	4.253	5.037	5.751
20	4.232	5.008	5.714	4.232	5.008	5.714	4.232	5.008	5.713
24	4.166	4.915	5.594	4.166	4.915	5.594	4.166	4.915	5.593
30	4.102	4.824	5.475	4.102	4.824	5.475	4.102	4.824	5.475
40	4.039	4.735	5.358	4.039	4.735	5.358	4.039	4.734	5.357
60	3.977	4.646	5.241	3.977	4.646	5.241	3.977	4.646	5.241
120	3.917	4.560	5.126	3.917	4.560	5.126	3.917	4.559	5.126
∞ or 99,999,999	3.858	4.474	5.012	3.858	4.474	5.012	3.857	4.474	5.011
Max Error	0.02%	0.02%	0.03%	0.00%	0.01%	0.03%			
Min Error	-0.05%	-0.02%	-0.05%	-0.04%	0.00%	0.00%			

	alpha =0.20					
	R Statistical Package qtukey function			QXLA Add-in QINVG function		
v / r	5	10	20	5	10	20
1	NaN	NaN	NaN	9.139	12.12	14.73
2	5.098	6.630	7.986	5.097	6.626	7.984
3	4.261	5.480	6.567	4.261	5.479	6.566
4	3.907	4.989	5.956	3.908	4.989	5.956
5	3.712	4.715	5.613	3.713	4.716	5.614
6	3.588	4.540	5.393	3.589	4.541	5.394
7	3.503	4.419	5.239	3.504	4.420	5.240
8	3.440	4.329	5.125	3.441	4.330	5.126
9	3.393	4.261	5.037	3.394	4.262	5.038
10	3.355	4.206	4.967	3.356	4.207	4.968
11	3.325	4.162	4.910	3.326	4.163	4.911
12	3.300	4.126	4.862	3.301	4.126	4.863
13	3.279	4.095	4.822	3.279	4.095	4.823
14	3.261	4.068	4.787	3.262	4.069	4.788
15	3.245	4.046	4.757	3.246	4.046	4.758
16	3.232	4.026	4.731	3.233	4.026	4.732
17	3.220	4.008	4.708	3.221	4.009	4.709
18	3.210	3.993	4.687	3.210	3.993	4.688
19	3.200	3.979	4.669	3.201	3.979	4.670
20	3.192	3.966	4.652	3.193	3.967	4.653
24	3.166	3.927	4.599	3.166	3.927	4.600
30	3.140	3.887	4.546	3.140	3.888	4.547
40	3.114	3.848	4.493	3.114	3.848	4.494
60	3.088	3.809	4.439	3.088	3.809	4.440
120	3.063	3.770	4.384	3.063	3.770	4.385
∞ or 99,999,999	3.037	3.730	4.329	3.038	3.731	4.330
Max Error	0.03%	0.06%	0.03%			
Min Error	-0.04%	-0.02%	-0.03%			

r =5		
	R Statistical Package ptukey function	QXLA Add-in QDISTG function
Q	*alpha*	*alpha*
4	0.053366	0.053567
5	0.008749	0.008820
6	0.001143	0.001122
7	0.000130	0.000120
8	0.000014	0.000012
9	0.000001	0.000001
10	0.000000	#VALUE!
***r* =10**		
Q	*alpha*	*alpha*
4	0.161537	0.160628
5	0.031188	0.031405
6	0.004461	0.004482
7	0.000534	0.000512
8	0.000058	0.000051
9	0.000006	0.000005
10	0.000001	#VALUE!
***r* =20**		
Q	*alpha*	*alpha*
4	0.374893	0.376235
5	0.092059	0.091752
6	0.015082	0.015232
7	0.001950	0.001938
8	0.000222	0.000207
9	0.000024	0.000020
10	0.000002	0.000002

Figure 2: Verification table for `QINVG`($\alpha = 0.05$, $r = 5$, 10, 20, $v = 1(1)20$, 24, 30, 40, 60, 120, 99,999,999), `QINVG`($\alpha = 0.2$, $r = 5$, 10, 20, $v = 1(1)20$, 24, 30, 40, 60, 120, 99,999,999), and `QDISTG`($QVal = 4(1)10$, $r = 5$, 10, 20, $v = 40$).

In this case (see Figure 2), the `QDISTG` function was unable to calculate a value for α with $Q = 10$ when r was 5 and 10. This is due to the fact that the estimated α was outside the limits of the add-in. Currently, the lowest α value that can be calculated by `QDISTG` is 0.000001. The limitations of the **QXLA** add-in are as follows:

- α values used by `QINVG` should ideally be between 0.5 and 0.001 but can be used for $\alpha < 0.001$ without major concerns (Gleason 1999). The code automatically allows for this.

- α values calculated by `QDISTG` must be between 0.5 and 0.000001. Between 0.5 and 0.001, it uses the method by Gleason (1999) and was extended by extrapolation to 0.000001. Outside this range, a value `#VALUE` will be displayed in the cell. It would be possible to extend the range by further extrapolation but Gleason (1998) does not recommend extrapolation on the other side of the range, above $\alpha = 0.5$.

- v used by `QINVG` and `QDISTG` can be an integer between 1 and 99,999,999. It would be possible to extend the range further in the code. Outside this range, a value `#VALUE` will be displayed in the cell.

- r used by `QINVG` and `QDISTG` should ideally be an integer between 2 and 100. However, higher values will automatically be extrapolated and will not return an error value. According to Gleason (1998) extrapolation is feasible (and accurate) to $r = 200$.

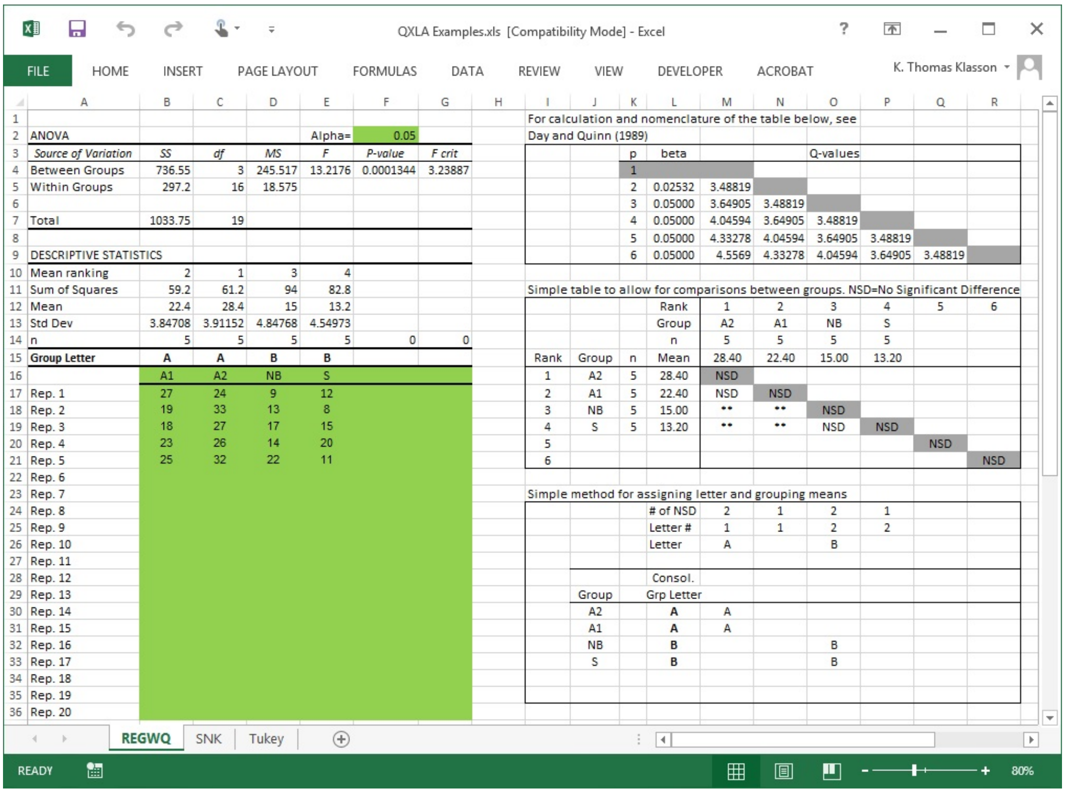


Figure 3: Spreadsheet with example implementing the REGWQ test for four treatment groups using the user-defined function `QINVG`(α, r, v). The green sections indicate user input sections which contain the raw data (cells `B16:G36`) and the desired α (cell `F2`). Other sheets in the downloadable workbook demonstrate S-N-K and Tukey tests.

3. Worked MCP example

Day and Quinn (1989), in their review of different MCPs, provide an example also used here. They presented four groups (A1, A2, NB, and S) that represented different treatments with five replicates in each group. Figure 3 is a depiction of a spreadsheet which uses the add-in and allows the user to enter data in the highlighted green sections. The workbook automatically constructs the basic single factor ANOVA table, performs the Tukey test, the stepwise REGWQ and S-N-K tests, and determines which mean belongs to which grouping (cells `B15:G15`). The Q values calculated by the add-in are in agreement with those listed by Day and Quinn (1989). The result from the REGWQ test shows that the means of groups A1 and A2 are significantly different from the other two but not from each other, and likewise for the means of the other two groups. The workbook is available for download and uses the **QXLA** add-in code. It should be noted that the workbook contains the VBA code for `QINVG` and `QDISTG` and can therefore function without installing the add-in.

4. Conclusions

- A subroutine and user-defined functions, `QINVG`(α, r, v) and `QDISTG`$(QVal, r, v)$, of the upper quantiles and the upper percentiles for the studentized range were successfully implemented in an **Excel** add-in using VBA programming.

- As indicated by Gleason (1999), the method "sacrifices some exactness for speed and simplicity." However, its easy programming, fast execution, and wide range of usability makes it an attractive option in **Excel**.
- Its efficiency was demonstrated by Lew (2011) who stated that R completed a data set of 1,216,000 points in 45 min using the `qtukey` function, which uses the algorithm by Copenhaver and Holland (1988), while a Python program using the Gleason method took 181 seconds (Lew 2011).
- The efficiency was also tested as part of this work against the **Excel** add-in **RealStats-2007** (Zaiontz 2016), which took 115 sec to complete the calculation of 78 Q values ($\alpha = 0.05$, $r = 5$, 10, 20, $v = 1(1)20$, 24, 30, 40, 60, 120, 99,999,999) compared with < 1 sec with the **QXLA** add-in.
- The user-functions allow for an expanded use of **Excel** in data analysis.

Acknowledgments

Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture. USDA is an equal opportunity provider and employer.

References

Buttrey SE (2009). "An **Excel** Add-In for Statistical Process Control Charts." *Journal of Statistical Software*, **30**(13), 1–12. doi:10.18637/jss.v030.i13.

Chapra SC, Canale RP (1985). *Numerical Methods for Engineers with Personal Computer Applications*. McGraw-Hill, New York.

Copenhaver MD, Holland B (1988). "Computation of the Distribution of the Maximum Studentized Range Statistic with Application to Multiple Significance Testing of Simple Effects." *Journal of Statistical Computation and Simulation*, **30**(1), 1–15. doi:10.1080/00949658808811082.

Day RW, Quinn GP (1989). "Comparisons of Treatments after an Analysis of Variance in Ecology." *Ecological Monographs*, **59**(4), 433–463. doi:10.2307/1943075.

Ferreira DF, Demetrio CGB, Manly BFJ, Machado AA (2007). "Quantiles From the Maximum Studentized Range Distribution." *Revista de Matematica e Estatistica*, **25**(1), 117–135.

Gleason JR (1998). "**dm64**: Quantiles of the Studentized Range Distribution." *Stata Technical Bulletin*, **46**, 6–10.

Gleason JR (1999). "An Accurate, Non-Iterative Approximation for Studentized Range Quantiles." *Computational Statistics & Data Analysis*, **31**(2), 147–158. doi:10.1016/s0167-9473(99)00002-x.

Harter HL (1960). "Tables of Range and Studentized Range." *The Annals of Mathematical Statistics*, **31**(4), 1122–1147. doi:10.1214/aoms/1177705684.

Lew R (2011). "**qsturng-py**." URL https://code.google.com/archive/p/qsturng-py/.

Lund RE, Lund JR (1983). "Algorithm AS 190: Probabilities and Upper Quantiles for the Studentized Range." *Journal of the Royal Statistical Society C*, **32**(2), 204–210. doi:10.2307/2347300.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Snedecor GW, Cochran WG (1967). *Statistical Methods*. 6th edition. The Iowa State University Press, Ames.

Walpole RE, Myers RH (1989). *Probability and Statistics for Engineers and Scientists*. 4th edition. Macmillan Publishing Company, New York.

Zaiontz C (2016). "Real Statistics Using **Excel**." URL http://www.real-statistics.com/.

Affiliation:

K. Thomas Klasson
U.S. Department of Agriculture
Agricultural Research Service
1100 Robert E. Lee Boulevard
New Orleans, LA 70124, United States of America
E-mail: thomas.klasson@ars.usda.gov

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/

June 2018, Volume 85, Code Snippet 1 *Submitted:* 2016-07-15
doi:10.18637/jss.v085.c01 *Accepted:* 2017-08-10