Understanding and Teaching Within-Cluster Correlation in Complex Surveys

Humberto Barreto and Manu Raghav[§]

DePauw University

Harrison Hall

7 E Larabee St

Greencastle, Indiana 46135

Email: hbarreto@depauw.edu, manuraghav@depauw.edu

This Version: February 20, 2015

Abstract: This econometrics pedagogy paper demonstrates the importance of using cluster standard errors with data generated from complex surveys. Simulation is used to show that both classic ordinary least squares and robust standard errors perform poorly in the presence of within-cluster correlated errors, while cluster standard errors perform much better. We take advantage of Excel's spreadsheet interface to produce clear and intuitive visuals of the data generation process and intuitively explain key results. Customizable Stata and R implementations, which help in further analysis by taking advantage of the unique different capabilities of Stata and R, are also provided. We conclude with suggestions for how to use these files in the classroom.

JEL codes: A2, A22, A23, C80, C81, C83, C87

Keywords: complex survey, simulation, cluster sampling, estimation, survey regression

[§] Author names are arranged alphabetically.

1. Introduction

Once taught only at the graduate level and even then also in an advanced elective, complex survey methodology has become so common that it deserves to be considered as a special topic in an undergraduate, upper-level econometrics course. This paper will show how to teach the concept of within-cluster or intra-cluster correlation, an essential element of complex surveys, using Excel, Stata, and R. Highlights include clear presentation of the data generation process (DGP), simulation to demonstrate sampling distributions, and emphasis on the estimated standard error (SE) as a random variable.

The exposition is sensitive to the intended audience. Instead of a mathematically rigorous approach, with a variety of abstract modeling scenarios and error structures cast in formal language, this paper will consider a single, concrete within-cluster correlated error DGP (in the family of cluster-specific random effects models) based on Moulton (1990). We further restrict the analysis by making the ordinary least squares (OLS) estimator the main focus of the paper. Our goals include explaining why OLS estimated SEs (which we call classic SEs in this paper) do not do a good job in estimation, why robust SEs do not do much better, and how cluster SEs, which give good results, work. We include, as optional, advanced material in the Excel workbook, explicit matrix derivations of the various SEs, including exact SEs.

Complex surveys may consist of two or more of the following: stratified sampling, cluster sampling, and unequal probability of selection. Stratified sampling increases the precision of the estimates but increases the cost of the survey. On the other hand, cluster sampling and unequal probability of selection are utilized for convenience in conducting the survey and to lower the cost of the survey. All the above non-simple random sampling methods lead to increased complexity in calculating standard errors. Cluster sampling increases the standard error of the estimates as sampling variations arise from different clusters of observations being chosen in different samples instead of different individual observations, as is the case with sampling methods that do not involve cluster sampling. Since all of the observations in the clusters are likely to be fairly similar to one another, we do not get as much independent information when we select a cluster and measure individual observations inside that cluster. Broadly speaking, there are two types of violations: unequal probability of selection and errors that are not identically and independently distributed (iid). Both of these complications are characteristics of complex surveys, but they are separate issues and can be isolated for individual exposition.

Barreto and Raghav (2013) showed how unequal probability of selection can lead to biased and inconsistent OLS coefficient estimates even if there is no cluster sampling that causes errors to be non-iid. The paper went on to explain why probability-weighted least squares out performs OLS in such situations.

This sequel focuses on how cluster sampling causes errors to be not iid, while assuming equal probability of selection. It uses simple random sampling (implying equal probability of selection) with a DGP that features within-cluster correlation of errors. By explicitly modeling and displaying errors that violate the classic iid assumption, students can see and understand the meaning of "correlated errors" produced by complex surveys (and, in similar fashion, panel data). Furthermore, they can directly observe the implications of the within-cluster correlated error structure on OLS coefficients and a variety of estimated SEs.

The Excel workbook contains several user-defined functions that enable implementation of the DGP and Monte Carlo simulation. We use these functions for simulation to develop intuition and provide analytical results for confirmation. With the *Cluster.xlsm* workbook open, these functions can be accessed by other open workbooks and provide a convenient way to use Excel to analyze data from complex surveys.

The next section describes the model and its implementation in Excel. Section three explains how to run a variety of simulations to show the primary results: OLS estimated SEs are biased and inconsistent when errors are within-cluster correlated, robust SEs adjust for heteroscedasticity, but not within-cluster error correlation, and cluster SEs can offer better estimates, but they are not a perfect solution. The next two sections describe the implementations in Stata and R. The Excel, Stata, and R files are all freely available at www.depauw.edu/learn/stata. The last section offers teaching tips.

2. Modeling a Within-cluster Correlated Errors DGP in Excel

The Excel workbook, *Cluster.xlsm*, is a macro-enabled file that requires Microsoft Excel 2007 or greater. The file will work in earlier versions of Excel (back to 1997), but the number of clusters and error correlation matrix must not exceed 256 columns. With Excel 2007 or greater, having hundreds of clusters with many observations per cluster is possible, but the bigger the data set, the slower the simulations. Experimenting with large numbers of clusters and many observations per cluster is better done in R because the simulations run much faster. Download *Cluster.xlsm* from www.depauw.edu/learn/stata and be sure to enable macros when opening this file.

The *DGP* sheet has a six observation data set, with three clusters of two observations each. This sheet gives a bird's eye view of the data generation process. Although complex surveys are based on sampling from finite populations, the core logic of within-cluster error correlation is easier to grasp by modifying the classical linear model that forms the foundation of regression analysis. Thus, in the *DGP* sheet in *Cluster.xlsm*, the data are generated by $Y_{ci} = \beta X_{ci} + \varepsilon_{ci}$ where $\varepsilon_{ci} \sim N(0, \sigma_c)$ and $Cor(\varepsilon_{ci}, \varepsilon_{cj}) = \rho_\varepsilon$ for $i \neq j$ with c clusters of size n_c yielding $n = c\ n_c$ total observations. The terminology is conventional: error and epsilon, ε, are synonymous; Greek letters (in red text) are unknown parameters, while Latin letters represent sample analogues; and residuals (e) are found by subtracting predicted Y ($\hat{Y}$) from actual Y.

Figure 1 shows how this DGP is implemented in Excel. Click on one of the *Y* cells to see that the coefficients, β_0 and β_1, are used in the familiar way to produce the deterministic component of *Y* and a random error term is added to create observed *Y* in column N. Notice that the *Xs* are fixed in repeated sampling and do not change as *F9* is pressed. The cluster indicator variables, *C1*, *C2*, and *C3* and *Cluster ID* help communicate the clustered nature of the data set.

<<Insert Figure 1 Here>>

The key to clustering is, of course, the error correlation matrix in cells G11:L16 and its contribution to the epsilons realized in column M. Click the *Cluster 1*, *2*, and *3* radio buttons in the *Highlight* box for a visual display of how the errors are clustered. The *epsilon* (ε) values are produced by three separate uses of the MULTIVARNORMAL array function in cells M11:M12, M13:M14, and M15:M16. Click in cell M11 or M12 to highlight precedent cells, showing how MULTIVARNORMAL outputs a pair of random variables that conform to the average, SD, and within-cluster error correlation matrix. Press the *Esc* (escape) key to exit an array function. Pressing the *F9* key recalculates the sheet and draws new epsilons, which makes the observations in the charts bounce. Press *F9* repeatedly and observe how the errors bounce with no obvious relationship. This is the hallmark of independent and identically distributed errors.

To show that the errors are in fact normally distributed, click the MC Sim button and track two epsilons (e.g., cells M11 and M12). Results from this simulation should offer convincing evidence for the claim that the errors are normally distributed with mean zero and SD of seven.

To demonstrate the impact of within-cluster correlation on the errors and observed *Y*, change ρ_ε in cell G6 to 0.99 and press *F9*. The error correlation matrix updates, populating the off-diagonal terms in each cluster with 0.99 and the epsilons now show a marked pattern that is discernible in the 3 pairs of errors that are drawn with every press of *F9*. Each pair of epsilons is clearly connected—a high (low) value of one epsilon is now followed by another high (low) value and the pairs are either positive or negative, but not mixed as was common when there was no within-cluster correlation. The chart to the right of the epsilons provides a strong visual, showing the pairwise connection that is the very definition of within-cluster correlated errors. A Monte Carlo simulation of the within-cluster correlation of the errors (tracking cell Q28) shows that the MULTIVARNORMAL function is working (roughly) as advertised.

To emphasize the difference between the classical linear model and the within-cluster correlated errors DGP, simply copy the *DGP* sheet and reset cell G6 to its initial value of zero. Now switch back and forth between the two *DGP* sheets, pointing out the similarities and differences. The parameters and Xs are the same, but the error correlation matrix is different and this is what is driving the difference in the errors.

Scroll right until columns AC:AK are on screen. Excel's native LINEST function is used to show OLS regression results. The orange background cell is the OLS estimated SE, which we will refer to as the *Classic SE*. It is computed by using the root mean squared error (RMSE) as an estimate of the SD of the errors. LINESTR is a user-defined function that reports the *Robust SE* in the blue background cell. It uses each observation's residual squared as an estimate of the individual variance for each observation. LINESTW is a user-defined function that computes the *Cluster SE* based on the conventional Taylor linearization formula. Finally, the *Exact SE* of the OLS estimated slope coefficient, i.e., the true precision of the OLS estimator based on exact knowledge of ρ_ε and σ_i, is displayed in cell AC31 as an output of the OLSEXACTSE user-defined function. The red text reminds students that the *Exact SE* is based on unknown parameter values and is unobservable. It is also useful to note that the *Exact SE* is not a random variable. The formulas for the covariance matrices are provided below.

Classic SE:

$$\hat{V}(b) = \frac{n}{n-k}\hat{\sigma}^2\,(X'X)^{-1}.$$

Robust SE:

$\hat{V}(b) = \frac{n}{n-k}(X'X)^{-1}X'\hat{\Omega}X(X'X)^{-1}$, where $\hat{\Omega}$ is a diagonal matrix with each observation's estimated error variance ($\hat{\sigma}_i^2$). Note that $X'\hat{\Omega}X = \sum_{i=1}^{n} x'e'ex$.

Cluster SE:

$\hat{V}(b) = \frac{c}{c-1}(X'X)^{-1}D(X'X)^{-1}$, where $D = \sum_{i=1}^{c} x'e'ex$.

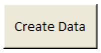
Each cluster's $x'e'ex$ is calculated by first summing $x'e'$ and ex over the observations in each cluster and then multiplying. So, there will be *c* of these $x'e'ex$ products, one for each cluster. These are then added together to get *D.*

Exact SE:

$\hat{V}(b)=\frac{c}{c-1}(X'X)^{-1}\left(\sum_{1}^{c}X_c'\Omega_c Rho_c X_c\right)(X'X)^{-1}$, where Ω_c is a diagonal matrix with each observation’s error variance (σ_i^2) and *Rho$_c$* is the within-cluster correlation matrix. Similar to the *D* matrix in *Cluster SE* formula above, when calculating the term in the parentheses, only data within each particular cluster are used.

Matrix manipulations that reveal detailed computations for the SEs of the sample slope displayed in column AC are available by scrolling right. The gray shaded columns separate the different standard errors. The *Robust SE* is calculated in two equivalent ways, with the first an extension of the *Classic SE* and the second a natural lead-in to the *Cluster SE*. Notice that the *Robust SE* sums $x'e'ex$ over all of the observations, while the *Cluster SE* creates individual $x'e'ex$ products, one for each cluster and then sums these individual products. This is the critical step in how the *Cluster SE* incorporates within-cluster correlation—it essentially treats each cluster as an observation for purposes of variance estimation.

With columns AC:AK visible, pressing the *F9* key repeatedly confirms that the estimated coefficients are the same, but the three estimated SEs are different. Both estimated coefficients and estimated SEs change with each recalculation, forcefully demonstrating that they are random variables. Their sampling distributions are of utmost interest and will determine which estimator is preferable. Consistent with our intuitive, visual pedagogical approach, simulation will be our primary mode of analysis. Having the *Exact SE* available enables confirmation of the simulation and provides a clear signal of which statistics are random variables and which are not.

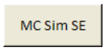
We have, however, reached the limits of this DGP—three clusters with two observations per cluster is simply too small to explore the properties of the various estimators. Scroll back left to the beginning of the *DGP* sheet. Set ρ_ε = 0 (in cell G7) if needed and confirm that cells D22 and D23 are set to 25 and 4, respectively. Click the  button.

Excel inserts two new, color-coded sheet tabs into the workbook. The fixed *Xs* in column B come from the matching *25X4* sheet. Scroll right to column DY in the *25Y4* sheet, noticing along the way how the DGP has been implemented in exactly the same way as the tiny data set in the *DGP* sheet. Click on cells to see formulas as needed. Column DY has four calls to the MULTIVARNORMAL function, one for each cluster. Column DZ has the familiar formula for the *Y* as the sum of the deterministic and stochastic components. Further right are two charts with LINEST, LINESTR, and LINESTW below. In addition, the Exact OLS SE is reported.

The tight cloud is a function of the parameters chosen. Visit the *25X4* sheet to see how the fixed *Xs* were generated. With 25 clusters, a *Step X* of 3 (in cell B2) is producing a great deal of spread in the Xs. Lowering *Step X* will produce less spread in the *Xs* and increase the variability of the OLS slope estimator. Exploring the relationship between the spread of the *Xs* and the SE of the OLS slope estimator would be an interesting open-ended assignment.

Return to the *25Y4* sheet and press *F9* repeatedly to confirm that the *Exact SE* does not bounce, while the other three estimated SEs vary. We are ready to explore the properties of each estimator of the OLS SE.

3. Monte Carlo simulation to evaluate estimators of the SE

With ρ_ε = 0 and homoscedastic errors, the *Classic SE* should perform well. Click the  MC Sim SE button and accept the default 1,000 repetitions to run a simulation. The resulting output is in a new sheet, conveniently color-coded and placed between the *Y* and *X* sheets for the 25, 4 (number of clusters and observations per cluster) pair. Although your output will not be exactly the same, Figure 2 shows typical results.

The results from the 1,000 samples are listed in the simulation output sheet in rows 2 to 1001. In Figure 2, rows 997 to 1001 contain slope coefficients (column B) and three estimated SEs (columns C, D, and E). Summary statistics are displayed in rows 1003 to 1006.

<<Insert Figure 2 Here>>

Cell B1003 in Figure 2 has the average of the 1,000 slope coefficients, which is close to 2 (the parameter value) and suggests that OLS is unbiased. The SD of the slopes in cell B1004 is an approximation of the true, exact OLS SE and, for convenience, it is also reported in cell H1003, next to the *Exact SE*. The fact that the *Approx SE* is close to the *Exact SE* is evidence that the simulation is working well.

The average of the 1,000 estimated SEs reported in row 1003 of Figure 2 should be compared to the *Exact SE*. Not surprisingly, given that the *DGP* obeys the classical linear model, the *Classic SE* is quite close to the *Exact SE*. This is evidence that all is well.

Next, we turn attention to a simple violation of the classical model: unequal variance of the errors. Return to the *DGP* sheet and click the Heteroscedasticity button. The *Y on X* chart displays the familiar horn-shape that is indicative of heteroscedasticity because the formula for *Y* has been altered. Click on a *Y* cell in column DZ to see that that stochastic component is now the error multiplied by the square of the value of *X*. Thus, bigger *X* values have bigger error variances. Notice that the *Exact SE* is no longer displayed because the way we have implemented heteroscedasticity by modifying the formula for *Y* is not properly incorporated into the matrix computation for the *Exact SE*.

Click the MC Sim SE button and accept the default 1,000 repetitions to run a simulation. The variability in b_1 is so high that it is likely that the average of merely 1,000 estimated slopes is not near 2, the population parameter. In fact, the OLS slope estimator is unbiased even in the presence of heteroscedasticity and this can be demonstrated by increasing the number of repetitions in the simulation. More important for our purposes is a comparison of the *Classic* and *Robust SEs*. Simulation shows that the *Classic SE* severely underestimates the true variability (as reflected in the *Approx SE*), while the *Robust SE* does fairly well. This is a clear demonstration of why the *Robust SE* has come dominate econometric practice.

We are now ready to show the implications of the within-cluster correlated errors. Begin by returning to the *25Y4* sheet and clicking the Homoscedasticity button to return the DGP to the classical model. Scroll left to the beginning of the sheet and change cell D4 to 0.99. Click the Update Data button and scroll right to the end of the sheet, noting how the error correlation matrix has been changed.

The two charts now offer a strong visual of the effects of within-cluster correlation. Press F9 repeatedly to redraw errors and observe how they seem to be clumped together, as shown in Figure 3. Each cluster has highly correlated errors so they are not evenly distributed like before. This translates into the grouping effect in the *Y on X* chart.

<<Insert Figure 3 Here>>

Notice that, before running a simulation, it is easy to see that the *Classic* and *Robust SEs* are roughly half of the *Exact SE*, while the *Cluster SE* seems to be doing much better. Pressing F9 repeatedly emphasizes the effect of clustering in the charts and shows that the *Cluster SE* seems best, but a simulation is needed to make the case more concretely.

Click the MC Sim SE button and examine the results, which will be similar to Figure 4. There are three critical points to be made:

1) The *Classic SE* is badly biased—comparing 0.029666 to 0.061731 shows that it is, on average, roughly half of the *Exact SE*. Using an estimated SE computed by the conventional algorithm for OLS will underestimate the true precision of the variability in the estimated slope, compromising confidence intervals and hypothesis testing.
2) The *Robust SE* (with average value of 0.028926 in Figure 4) does not fix this problem. As Cameron and Trivedi (2005, p. 707) point out, "The term 'robust' standard error can confusion." The *Robust SE* adjusts only for heteroscedasticity and is powerless against within-cluster correlated errors in a complex survey (or panel data set).

3) The *Cluster SE* (with average value of 0.058111 in Figure 4) does much better in this case. Although not exactly on target (and this is not an artifact of simulation), it is much closer to the *Exact SE* than its rivals.

<<Insert Figure 4 Here>>

4. Stata Implementation

Stata is one of the more popular statistical packages and many instructors may want to use it instead of Excel. As Stata is more geared towards programming rather than visualization, the implementation in Stata will not be as graphical as in Excel. The Stata do-file *Cluster.do* (available at www.depauw.edu/learn/stata) replicates the simulations described above.

This explanation below assumes that line numbers are displayed in the do-file editor of Stata. In case they are not, they can be enabled by clicking on the *Edit* menu of the do-file editor window of Stata, then selecting Preferences. Select the *Editor* tab of the dialog box and check *Line Numbers*, then click *OK*.

All of the customizable parameters are in the form of local macros and are defined at the very beginning of *Cluster.do* between lines 16 and 31. Each set of parameters are separated by a blank line. The first set contains local variables *clusters* and *obs_per_cluster*. The former sets the number of clusters in the simulations and the latter sets the number of observations within each cluster. The next set contains parameters regarding the *X* variable. Variables *corr_x*, *mean_x*, and *sd_x* respectively define the within-cluster correlation, the mean, and the standard deviation of the *X* variable. The next set contains information about the distribution of the errors. Variables *mean_ep*, *sd_ep*, and *corr_ep* respectively set the value of the mean, standard deviation, and the within-cluster (or intra-cluster) correlation of the epsilons. The values of betas are set with local macros *beta0* and *beta1*. Finally, the number of repetitions is set using the variable *reps*. The code used to calculate *Exact SEs* is located between lines 355 and 371.

Unlike the Excel file, the do-file has the additional option of drawing different sets of *X* values and then running Monte Carlo simulations with each set of *X*. The number of draws of the *X* variable is set by *x_reps*. To match the Excel implementation, *x_reps* is initially set at one. However, if *x_reps* is changed to 5 and *reps* is set at 1000, then Stata will run 5000 regressions, with 1000 simulations for each set of X. We use an algorithm in Stata that does not require us to create large matrices to store results as this can be problematic in some versions of Stata such as the small or the IC version. Instead, our method updates the value of means and standard deviations of slope coefficients of individual regressions in each iteration during that iteration. The formulas for which we update the values are located between lines 311 and 337.

We recommend not setting Monte Carlo repetitions to more than 1000 as that could take a lot of time and, therefore, may not be suitable for classroom use. In order to make sure that results are exactly replicated, we should set the seed to a specific number (we have set it to 5000 but it can be set to any other number).

<<Insert Figure 5 Here>>

As Figure 5 shows, running *Cluster.do* will give results that enable easy analysis. Information about the parameter values used in the simulation is reported, including the within-cluster correlation of the epsilons (.7). The average OLS slope coefficient (2.0095557) is followed by five standard errors. The *Classic SE* (the ordinary OLS standard error), *Robust SE*, and *Cluster SE* are the average of the estimated standard errors in the 100 repetitions. Finally, as explained previously in the paper, the Stata output also reports the exact standard error of the OLS estimator and the SD of the OLS slope coefficients from each repetition, which we call the approximate standard error.

These five SEs are not to be compared to each other. The *Exact SE* is the true precision of the OLS estimator and the *Approx SE* is being used here to confirm that the simulation is running as expected. The other three SEs are evaluated by comparing them to the *Exact SE*. Even with only 100 repetitions, Figure 5 offers evidence in favor of the *Cluster SE* since it is much closer to the *Exact SE* than its two rivals.

5. R Implementation

In addition to Excel and Stata, we have also provided an R script file called *Cluster.R* (available at www.depauw.edu/learn/stata) to implement the DGP in R. The R programming language is open source, available across many platforms, and has many resources for help and support (Racine and Hyndman, 2002). Anyone can download R for free from www.r-project.org.

Unlike Stata, some versions of which have limitations on matrix size, R has no such constraints. This is why we have used a different algorithm than the one we used for the implementation in Stata. This method takes advantage of the ability of R to create larger matrices where we store the results of each repetition (estimated slope and various SEs). These matrices, which are named as *slope_coef_array*, *se_OLS_array*, *se_robust_array*, and *se_cluster_array* will enable further analysis using the individual results of each simulation. Just like in the Stata do-file, all of the customizable variables are displayed at the beginning of the R script file.

The implementation in R is much faster than Stata so we can experiment with a larger number of repetitions, a larger number of clusters, and/or a larger number of observations per cluster with R. Figure 6 shows sample output.

<<Insert Figure 6 Here>>

Notice that 1000 clusters of 3 observations each are drawn 10,000 times. This would be impractical in Excel and would take a long time in Stata. The results in Figure 6 confirm the results from earlier simulations (see Figures 4 and 5): the *Cluster SE* is clearly much superior to the *Classic* and *Robust SEs*.

6. Teaching tips

This paper focuses on a limited range of options and content of the econometrics of complex survey design, an area with many models, specifications, and error structures (see Cameron and Trivedi (2005) and Lohr (2009)). The exposition was driven by a desire to convey the heart of the issue of within-cluster correlation to a student audience. We think this is the best way to teach this material. Instead of trying to do too much, focus on core lessons and ideas with strong visuals and concrete, numerical examples.

After reviewing the *DGP*, we ran three simulations. First, under the classical model, it is clear (Figure 2) that the *Classic SE* does reasonably well. Incorporating heteroscedasticity and simulating demonstrates the power and popularity of the *Robust SE*. The third simulation is the most important: as Figure 4 shows, both the *Classic* and *Robust SEs* do badly, while the *Cluster SE* wins the race. This key result was replicated in Stata and R.

Under both heteroscedasticity and within-cluster correlated errors, OLS remains consistent, but it is inefficient. We ignored this point because our focus was on the fixing the estimated SE. Under a complex survey design with unequal probability of selection, Barreto and Raghav (2013) show that OLS yields biased slope coefficients. In this case, OLS must be replaced by an unbiased estimator (such as the user-defined function LINESTW or Stata's svyset and svy: reg approach).

The Excel implementation is best for introducing and explaining the effects of within-cluster correlated errors. The spreadsheet shows the effect of cluster sampling and graphs allow for strong visuals that update when parameter values are changed. To simulate a real-world complex survey, however, with thousands of clusters, and tens of thousands of observations, R is the best option.

The larger the number of clusters, the better will be the accuracy of the *Cluster SE*. Even though the cluster standard error always does better than the usual classic (OLS) and robust standard error estimators in the presence of within-cluster correlated errors, it does not estimate the *Exact SE* very well when the number of clusters is small. However, in practical applications of cluster sampling, the number of clusters is quite large. National surveys such the Current Population Survey (CPS) and National Health Interview Survey (NHIS) use cluster sampling to save time and money. The CPS has 2,025 clusters and NHIS has 1,900 clusters. As we have shown above, the cluster standard error does well under these circumstances. Discussing the sampling design and methodology of the CPS and NHIS in class or asking students to read how these surveys are conducted as a part of a homework assignment can be useful exercises. This helps students understand the practical application of cluster sampling and the difficulties encountered in complex survey design.

The R-script file can also be used in a variety of homework assignments and for independent projects. Unlike Excel and Stata, R is free to download and use. R is also faster than both Excel and Stata in running simulations. Students can use the R-script file for exploratory exercises, such as running simulations for different combinations of number of clusters, observations per cluster, and different values of epsilon and the slope variable. The R-script allows for looping over the *X* variable during a simulation. This enables tracking the effects of different *Xs* on the standard errors. Such exercises are likely to take a lot of time to finish and therefore are not suitable for classroom activities. They are best left as homework assignments or independent studies to be finished at home by students and preferably left to run overnight.

References

Barreto, H. and Howland, F. (2010), *Introductory Econometrics: Using Monte Carlo Simulation with Microsoft Excel*. Cambridge University Press.

Barreto, H. and Raghav, M. (2013), "Understanding and Teaching Unequal Probability of Selection," *Journal of Econometric Methods* **2**(1), 101-112.

United States Census Bureau, *Current Population Survey – Methodology*. http://www.census.gov/cps/methodology/, accessed October 22, 2013.

Cameron, A. and Trivedi, P. (2005), *Microeconometrics: Methods and Applications.* Cambridge University Press.

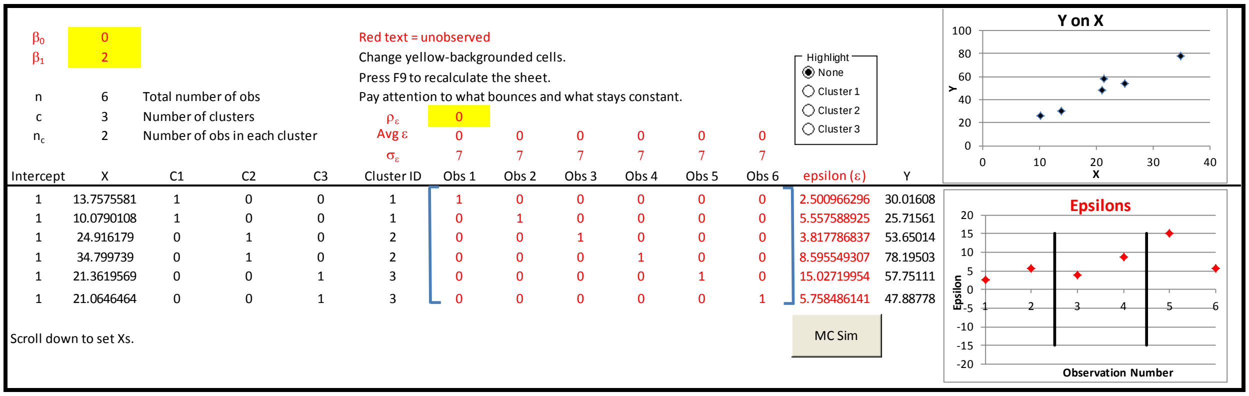
Lohr, S. (2009), *Sampling: design and analysis*. Thomson.

Moulton, B. (1990), "An Illustration of a Pitfall in Estimating the Effects of Aggregate Variables on Micro Units," *The Review of Economics and Statistics,* **72**(2), 334-338.

Office of Information Services, Center for Disease Control and Prevention, *About the National Health Interview*. http://www.cdc.gov/nchs/nhis/about_nhis.htm, accessed October 22, 2013.

Racine, J and Hyndman, R. (2002), "Using R to Teach Econometrics," *Journal of Applied Econometrics,* **17**(2), 175-189.

Figure 1: Implementing within-cluster correlated errors in the *DGP* sheet in *Cluster.xlsm*.



β_0	0	
β_1	2	
n	6	Total number of obs
c	3	Number of clusters
n_c	2	Number of obs in each cluster

Red text = unobserved

Change yellow-backgrounded cells.

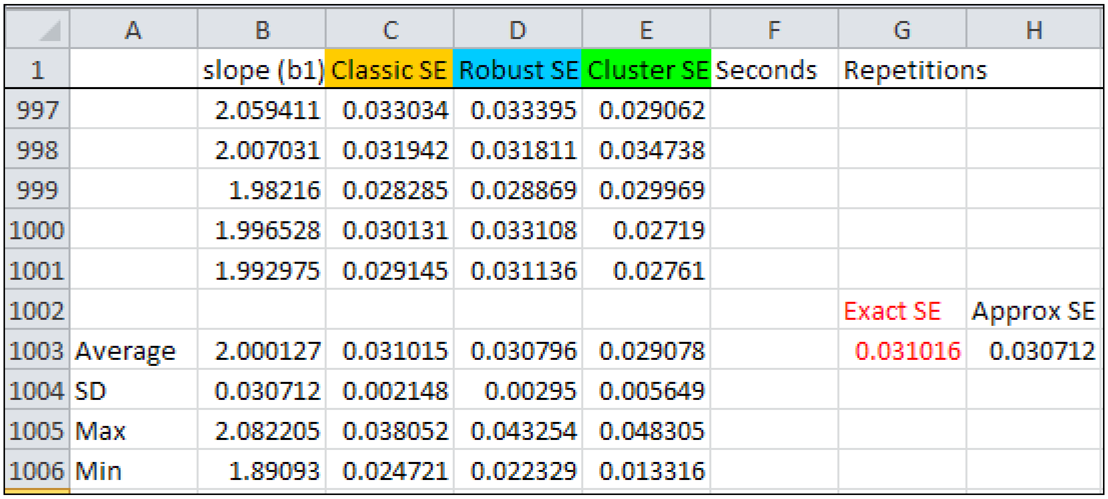
Press F9 to recalculate the sheet.

Pay attention to what bounces and what stays constant.

					ρ_ε	0							
					Avg ε	0	0	0	0	0	0		
					σ_ε	7	7	7	7	7	7		
Intercept	X	C1	C2	C3	Cluster ID	Obs 1	Obs 2	Obs 3	Obs 4	Obs 5	Obs 6	epsilon (ε)	Y
1	13.7575581	1	0	0	1	1	0	0	0	0	0	2.500966296	30.01608
1	10.0790108	1	0	0	1	0	1	0	0	0	0	5.557588925	25.71561
1	24.916179	0	1	0	2	0	0	1	0	0	0	3.817786837	53.65014
1	34.799739	0	1	0	2	0	0	0	1	0	0	8.595549307	78.19503
1	21.3619569	0	0	1	3	0	0	0	0	1	0	15.02719954	57.75111
1	21.0646464	0	0	1	3	0	0	0	0	0	1	5.758486141	47.88778

Scroll down to set Xs.

Figure 2: Simulation results for $\rho_\varepsilon = 0$ and constant $\sigma_\varepsilon = 7$.



	A	B	C	D	E	F	G	H
1		slope (b1)	Classic SE	Robust SE	Cluster SE	Seconds	Repetitions	
997		2.059411	0.033034	0.033395	0.029062			
998		2.007031	0.031942	0.031811	0.034738			
999		1.98216	0.028285	0.028869	0.029969			
1000		1.996528	0.030131	0.033108	0.02719			
1001		1.992975	0.029145	0.031136	0.02761			
1002							Exact SE	Approx SE
1003	Average	2.000127	0.031015	0.030796	0.029078		0.031016	0.030712
1004	SD	0.030712	0.002148	0.00295	0.005649			
1005	Max	2.082205	0.038052	0.043254	0.048305			
1006	Min	1.89093	0.024721	0.022329	0.013316			

Figure 3: Showing the effects of within-cluster correlation.

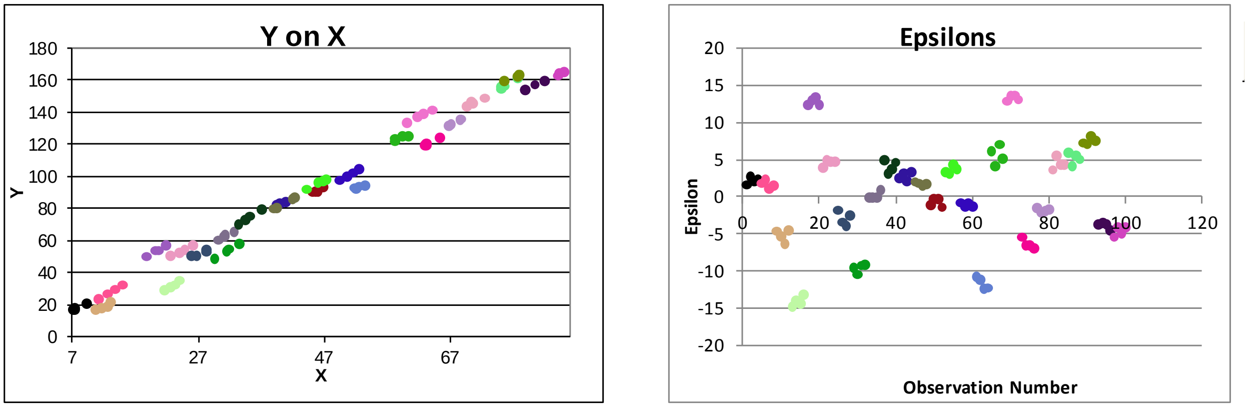
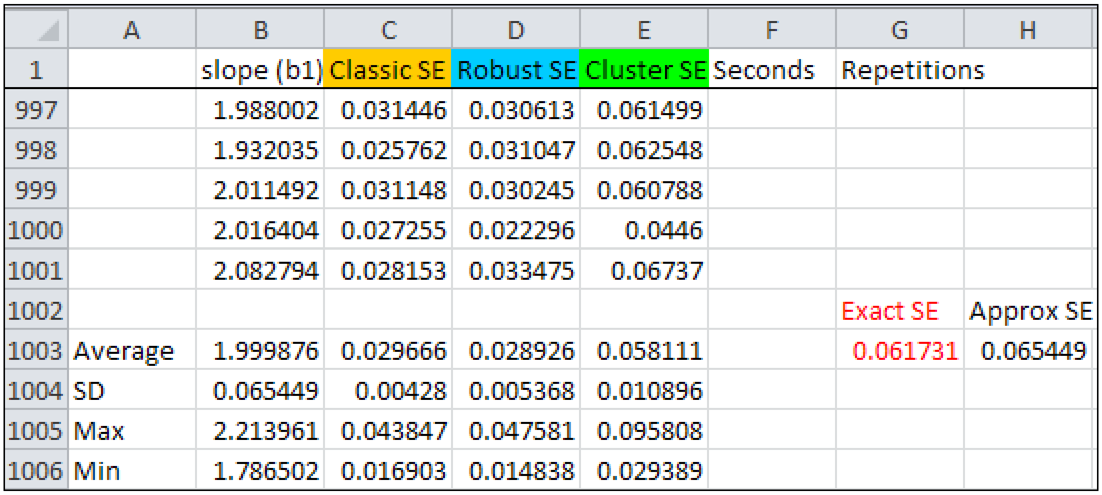
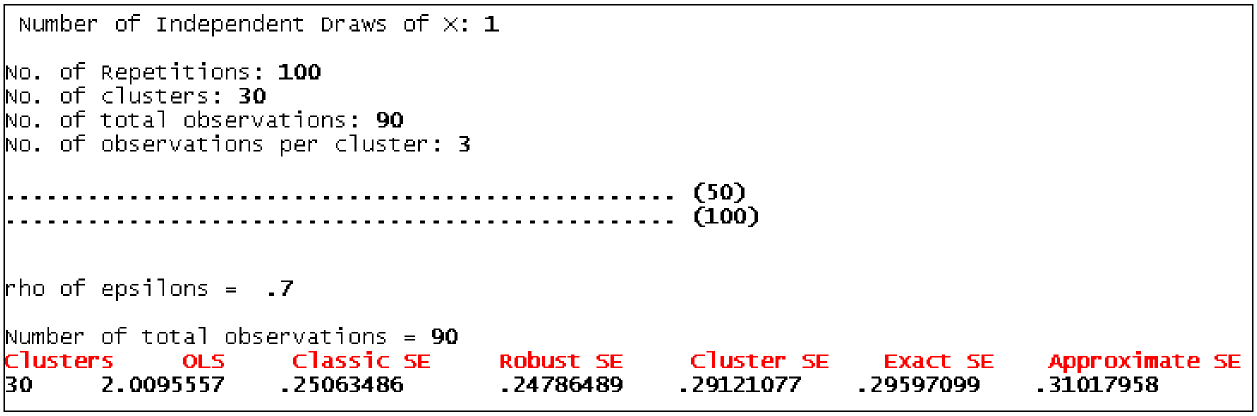


Figure 4: Simulation results for ρ_ε = 0.99 and constant σ_ε = 7.



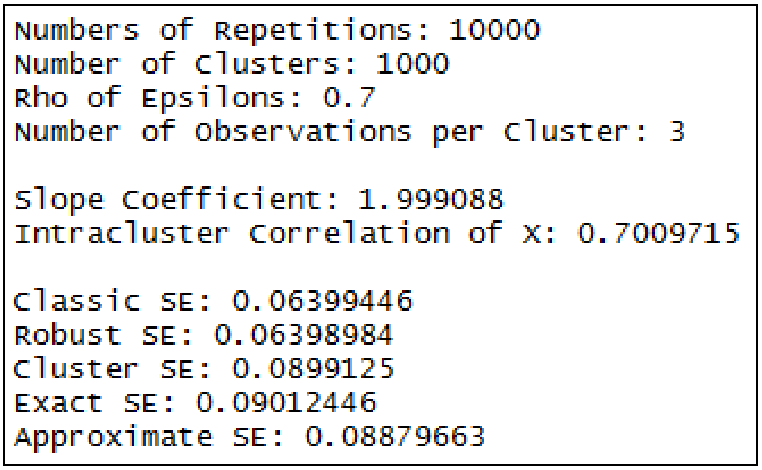
	A	B	C	D	E	F	G	H
1		slope (b1)	Classic SE	Robust SE	Cluster SE	Seconds	Repetitions	
997		1.988002	0.031446	0.030613	0.061499			
998		1.932035	0.025762	0.031047	0.062548			
999		2.011492	0.031148	0.030245	0.060788			
1000		2.016404	0.027255	0.022296	0.0446			
1001		2.082794	0.028153	0.033475	0.06737			
1002							Exact SE	Approx SE
1003	Average	1.999876	0.029666	0.028926	0.058111		0.061731	0.065449
1004	SD	0.065449	0.00428	0.005368	0.010896			
1005	Max	2.213961	0.043847	0.047581	0.095808			
1006	Min	1.786502	0.016903	0.014838	0.029389			

Figure 5: Simulation results for ρ_ε = 0.7 with 30 clusters of 3 observations each.



```
 Number of Independent Draws of X: 1

No. of Repetitions: 100
No. of clusters: 30
No. of total observations: 90
No. of observations per cluster: 3

.................................................. (50)
.................................................. (100)


rho of epsilons =  .7

Number of total observations = 90
Clusters      OLS      Classic SE      Robust SE      Cluster SE     Exact SE      Approximate SE
30      2.0095557    .25063486       .24786489     .29121077     .29597099     .31017958
```

Figure 6: Simulation results for $\rho_\varepsilon = 0.7$, with 1000 clusters of 3 observations each.



```
Numbers of Repetitions: 10000
Number of Clusters: 1000
Rho of Epsilons: 0.7
Number of Observations per Cluster: 3

Slope Coefficient: 1.999088
Intracluster Correlation of X: 0.7009715

Classic SE: 0.06399446
Robust SE: 0.06398984
Cluster SE: 0.0899125
Exact SE: 0.09012446
Approximate SE: 0.08879663
```