A Study of Survey Sampling Variable Selection Techniques in a Simulation Setting

Zachary A. King[1] and Eric V. Slud[2]

[1]*Department of Mathematics and Computer Science, Hampden-Sydney College, Hampden-Sydney, VA 23943 and*
[2]*Department of Mathematics, University of Maryland, College Park, MD 20742*

INTRODUCTION

As variable selection has been combed over extensively by many statisticians, it seemed necessary to change its flavor by experimenting with it in a survey sampling setting. This is a realm where far less progress has been made in regard to variable selection techniques. Issues such as non-response make selecting variables in this setting far more troublesome than with a complete given data-set. While non-response, an issue that commands a substantial amount of energy from the U.S. Census Bureau, was not a focal point of my project. The objective is to set up the framework to cope with it in future endeavors.

The framework is the simulations that created in R, a statistical programming platform. A major motivation for undertaking this project was to learn how to effectively code a working simulation, and it commanded the majority of my time. The importance of writing simulations is that they can easily be modified to accommodate and analyze real data. Benefits of data analysis skills can be reaped in a broad spectrum of mathematical and both hard and soft scientific subjects.

Specifically, these simulations create a design matrix comprised of values of all independent variables for each population unit. This is essentially a catalogue of the population in question with a full set of data comprising of values for any variable we could conceive of using. The design matrix multiplied by a user selected vector of parameters to weight the variables, with the addition of an error term, yields a vector of dependent values. The dependent values correspond to a single real quantity/attribute that the statistician attempts to estimate using a variety of correlating and non-correlating variables. Using this simulated data, parameter estimation, as well as variable selection, can be attempted in a variety of ways, and because of the nature of simulations, the real parameter values are known and can be used to verify the effectiveness of a given parameter estimation technique.

Knowledge and control of true parameter values, as occurs in the simulation setting, is extremely important for experimenting with variable selection. Ideally, one would want a reasonable variable selection algorithm to pick variables that are meaningful, i.e. variables that correspond to non-zero parameter values, and therefore influence the dependent variable. Whether or not a program is doing this successfully can be easily verified. This is not true with real data where the true parameters values are unknown.

The addition of survey sampling to the mix allows one to ask, and answer, the question of whether variables can be effectively selected on a consistent basis without access to the entire population (i.e. the entire design matrix), or even a significant fraction of it. One may also ask which variable selection techniques work best when dealing with moderate to large samples, or which parameter estimation techniques are most effective in this setting.

Theoretical variance formulas have been derived for the estimation of a dependent variable when we know values of independent values of only a sample of the population. In simulation, we can take many samples of the same population where we know the value of the dependent variable in question and evaluate both the mean and the variance of the resulting dependent variable estimations produced by the simulation. These quantities can then be compared with the theory to evaluate its validity.

The project investigates the relationship between ordinary least squares and ridge regression parameter estimation when dealing with survey samples. It is tested whether or not general theory of the ridge regression parameter estimator applies to a survey sampling setting when compared to the ordinary least squares estimator. The project also performs variable selection using two well-known selection criterion, AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion), as well as a few modified versions of the two and one more criterion of my own creation relating to ridge regression.

Definitions

For the purposes of this paper, I will define my shorthand.

N = total number of population units (example. number of citizens in the U.S.)
n = sample size
p = total number of variables in the design matrix
q = number of meaningful variables

$\hat{q}$ = number of estimated meaningful variables
ß = p-vector of user selected parameters
$\hat{\beta}$ = p- (or q- if post-variable selection) vector estimated parameters
e - error term; N-vector of normally distributed values about mean 0
X = Nxp design matrix of randomly generated values; the first column is always 1's
Y = N vector of dependent values

Parameters Estimation

As previously outlined in the introduction, my dependent values are generated by multiplying an arbitrary design matrix by a p-vector of user selected parameters with the addition of an error term:

$$Y = X\beta + \epsilon \qquad (1)$$

The dataset itself is just the Y and the X values, not the parameters or the error term. Note that the first value of ß is the intercept the same way b is the intercept in the equation of a line (y = mx + b). Think of the remaining p-1 ß values as "m" in the equation of a line. The intercept term creates the necessity for the first column of the design matrix to consist of only 1's such that the intercept term is preserved to be added to each individual Y value.

The immediate application of this data set is to recover the original parameters by a parameter estimation technique. To illustrate this more thoroughly, let's consider a modified equation of a line: $\underline{y} = m\underline{x} + b + \underline{\epsilon}$, where all underlined values are N-vectors (2).

Plotting y and x will result in a scatter plot of points for which the line of best fit is approximately y = mx + b. "m" is the only parameter here to be estimated and it can be sufficiently accurately estimated by utilizing simple linear regression. The ß0 and ß1 that minimize the following equation are the ordinary least squares intercept and slope estimates for the simple linear regression model.

$$f(\beta_0, \beta_1) = \sum_{i=1}^{N} [Y_i - (\beta_0 + \beta_1 X_i]^2 \ (3)$$

In multiple linear regression, where X no longer an N-vector, but an Nxp design matrix and m is not a constant but a p-vector ß, parameter estimation is similar to simple linear regression. The ß values (ß0, ß1,.., ßp) that minimize this equation are the ordinary least squares intercept and parameter estimations for the multiple linear regression model.

$$f(\beta_0, \beta_1, \dots, \beta_p) = \sum_{i=1}^{N} [Y_i - (\beta_0 + \beta_0 X_{i,2} + \cdots + \beta_p X_{i,p})]^2 \ (4)$$

This is the ordinary least squares method of parameter estimation on a given data set. In the R platform, this very simple - the "lm" function will minimize the above equation; entering a summary command will reveal the parameter estimates as well as yield other relevant information such as p values, f-statistic, etc.

Other methods for parameter estimation have been proposed and used selectively in the statistics community besides ordinary least squares. The most interesting method is ridge regression. It was designed to lower variance due to collinearity in parameter estimation. Consider the equation of $\hat{\beta}$ {ordinary least squares:

$$\hat{\beta}^{ols} = (XX')^{-1}X'Y \qquad (5)$$

This formula becomes quite messy when multiple variables (columns of X) are collinear; X'X becomes nearly singular. The values obtained from inverting it vary very widely because of the near singularity. Ridge regression aims to solve this problem by adding a "ridge" to X'X - this "ridge" is the result of the addition of a diagonal constant matrix to X'X. Let this diagonal matrix be C such that the diagonal entries are a constant and the rest are equal to zero. We now have:

$$\hat{\beta}^{ridge} = (XX' + C)^{-1}X'Y \qquad (6)$$

Unlike $\hat{\beta}^{ols}$, $\hat{\beta}^{ridge}$ is biased such that it tends to underestimate parameter values. However, the improvement in variance offsets the cost of bias. The Existence Theorem states that there will always exist a C such that the variance of $\hat{\beta}^{ridge}$will be less than $\hat{\beta}^{ols}$ (Breheny).

Dependent Value Total Estimation Techniques

The reason I've painstakingly gone over the process of parameter estimation is that it is essential for total dependent (Y) value estimation. With access to dependent values of an entire population, one could simply sum the dependent values to obtain the total. This is not so in the survey sampling world.

The general regression Y total estimator has been a focal point of my project: $t_{y,GREG} = X'_{tot}\hat{\beta}^{ols}$, where X_{tot} is p-row vector of the sum of the columns of X (7). The ridge regression Y total estimator is incorporated as follows: $t_{y,ridge} = X'_{tot}\hat{\beta}^{ridge}$(8).

Equations (7) and (8) assume one has access to the entire population and therefore is not a true estimator of Y total since there is no missing information to keep the "estimator" from reproducing the exact value of Y total. In the survey sampling realm, one has access to only a small chunk of the population, and thus a true estimator is used to attempt to find the correct value of Y total. To make equations (7) and (8) into estimators that can be appropriately used with a sample of a larger dataset, simply multiply each equation by (N/n) in order to scale up the estimate to the right order of magnitude.

$$\hat{t}_{y,GREG} = (\frac{N}{n})X'_{tot}\hat{\beta}^{ols} \qquad (9)$$

$$\hat{t}_{y,ridge} = (\frac{N}{n})X'_{tot}\hat{\beta}^{ridge} \qquad (10)$$

A key goal of the project was to evaluate the validity for a "variable model" (i.e. we don't know what the

model will be before variable selection) of a theoretical variance formula of the $\hat{t}_{y,GREG}$ values for fixed models that we call the naive variance.

$$\hat{V}_{naive} = \frac{N^2}{n}\left(1-\frac{n}{N}\right)\sigma_e^2[1+tr(X\text{^}'X)^{-1}S_{X,U}^2] \quad (11)$$

This was accomplished by taking "R" samples of the same simulated dataset and computing $\hat{t}_{y,GREG}$ for each sample. The empirical mean was compared to the true Y total value and the empirical variance was compared toV_{naive}. To do this in the most realistic fashion possible, variable selection must be executed before $\hat{t}_{y,GREG}$ computations can take place. First, a statistician would select a smaller model by filtering out useless variables through a variable selection technique. Note that the naive variance formula does not depend on how many variables are involved in the Y total estimation.

Variable Selection Criteria

Statisticians must devise ways of quantitative methods of comparing models to one another to determine which model is better. When programming a variable selection program, one must utilize a selection criterion to label models with a numerical value such that one can be picked over the rest. Generally RSS (Residual Sum of Squares) is a decent measure of goodness of fit, however, it will improve with each additional variable added to the model. That's not a good thing if the variable had nothing to do with the dependent values; the statistician is now fitting to noise. Good selection criterion for this reason have penalty terms for adding additional variables. This penalty term attempts to weed out the noise and only allow valuable variables into the final selected model. The selection criterion used in this project are AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), and several blends of the two as well as a criterion of my own creation.

All information criterion operate the same way - the selection criterion is calculated for a number of models; the model selected is the model with the lowest value of the selection criterion.

Each criterion is calculated by a different formula, each with its own distinct advantages and disadvantages. The formula for AIC is as follows:

$$AIC = n\,logRSS + 2(p+1) \quad (12)$$

Notice that the first term of the formula simply rewards goodness of fit; the better the fit, the lower the residual sum of squares. The second term is a penalty term that grows with each added parameter. "p" in this formula isn't necessarily the total number of candidate variables from the original dataset, but the number of variables in the model for which the AIC is being calculated. It is not quite accurate to label this "p" as q since we have not determined that the model in question is in fact the best model, which would ideally have q variables. The formula for BIC is very similar to AIC:

$$BIC = nlogRSS + (p+1)\ln(n) \quad (13)$$

The only formulaic difference between AIC and BIC is the harshness of the penalty term for adding additional variables to the candidate model. AIC uses a constant value of 2, whereas BIC uses the natural log of the sample size. When I refer to blends of AIC and BIC, I simply mean letting the penalty term for adding variables take on values that range roughly between those of AIC and BIC.

As it would be correct to expect that BIC with its harsher penalty term tends to allow selection processes to pick fewer variables as the cost of adding another variable is always high. Naturally, BIC selection processes will almost never result in a model with a noisy variable at the moderate cost of perhaps losing out on a couple of valuable variables. AIC processes will generally result in a model with all valuable variables at the cost of having a couple noisy variables tag along.

Variable Selection Procedures

The processes by which the aforementioned variable selection criterion are used are variable selection procedures. Three main popular variable selection procedures are forward selection, backward selection, and stepwise selection.

In a forward selection procedure, the algorithm will begin with an initial model, generally Y~1 or any other arbitrary constant and calculate the selection criterion for this model. Next the algorithm will select the variable with the greatest correlation to Y which will minimize the selection criterion from the set of all one-variable models. The next added variable will have the greatest partial correlation to the one variable model, thereby again minimizing the selection criterion from the set of all two-variable models. This process of adding one variable at a time will continue until the selection criterion can no longer be lowered by adding another variable. At this point the algorithm stops and whatever variables have been selected are the variables that comprise the final model.

Backward selection is simply the opposite of forward selection. Instead of beginning with no variables, we begin with the full model. Variables are subtracted one at a time until the selection criterion can no longer be improved by removing a single variable. The selection algorithm stops and the remaining variables become the constituents of the final model.

Stepwise selection blends forward and backward selection together. The initial model is arbitrary. Each iteration of the selection algorithm can either add or subtract a single variable from the

model; it will decide to add or subtract (and which variable) by calculating the selection criterion of the model resulting from all possible single variable additions and subtractions and then pick the move that results in the largest improvement of the selection criterion. The algorithm only stops when neither addition nor subtraction of a single variable can improve the selection criterion of the model. This procedure can take a lot of time and computing power due to the difficulty of arriving at the "perfect" model such that neither subtraction nor addition of variables improves it.

All three of these selection procedures are known as greedy algorithms because they add/subtract the variable that improves the selection criterion the most in a single step. These algorithms are fairly simple to write and execute, however, the best model isn't always found by a greedy algorithm. Sometimes smaller initial improvements lead to larger improvements later and a better final model. For my purposes, the simplicity of the greedy algorithm and the adequate results made its use appropriate.

The Simulation

Now that all the background is covered, I can begin to describe what it is exactly that I did to answer the questions I've previously mentioned.

I first created a dataset with a design matrix such that the first column is all 1s and the rest of the matrix is a collection of randomly generated numbers. The user selected p-vector beta was comprised of several 1s, .1s, -.1s, and 0s. I used variety of beta values to make it possible to evaluate my variable selection procedures based on how well they pick up variables with non-zero parameters as well as discard variables with corresponding parameter values of 0. I plugged this design matrix and beta into equation (1) to yield the dependent values.

Now to make this a study of a simulated survey sampling environment, I took 100 samples of size n from a data frame consisting of the Y values and X values without the column of ones. This is equivalent to picking n Y values and the corresponding n rows of the design matrix. I took 100 samples instead of one so that selection processes and parameter estimation could be done 100 times instead of one each time I ran my simulation.

For each of the 100 samples, I ran a canned forward selection program to yield the variables that the program found most valuable. Interestingly, the number of variables selected by the program was not constant over all 100 samples. The number of selected variables actually varied quite a bit. See Appendix C 1.1 for an example of this distribution. To estimate q, the number of meaningful variables, I allowed $\hat{q}$ to equal the floor of the average number of variables selected over the 100 samples. The $\hat{q}$ most commonly picked variables were logically nominated as the constituents of the final model.

I ran both ordinary least squares and ridge regression parameter estimation programs on the resulting variables for each of the 100 samples, the empirical variances of the average of those results may be found in Appendix F 1.1. With the parameter estimations, I was able to estimate the total Y values by equation 9 and 10 also 100 times, their means can be found in Appendix F 1.1 and a histogram of the 100 $\hat{t}_{y,GREG}$values from a complete execution of the simulation with an overlaid normal distribution can be found in Appendix D 1.2.

To gather data useful for answering questions about how sample size different selection criteria influence variable selection in a survey sampling setting, I had to run my program 15 different times varying the sample sizes and selection criteria. All results can be found in Appendix F 1.1. Note that k is the weight of the penalty term for selecting additional variables for a model (2 for AIC, log(n) for BIC). I let k take values in the set {2, 3, 4, 5, log(n)} and I let n take values in the set {200,500,800}. All possible 15 combinations of these n and k values were tested and recorded.

A Novel Penalty Term

Dr.Slud gave me the idea in one of his talks to try a non-constant penalty term; specifically what he said was to try to combine AIC and ridge regression to create a novel penalty term. Ridge regression operates by penalizing large ß values in order to minimize variance, however, it is because of this method that ridge regression produces biased estimates (Breheny). It always underestimates because it is designed to discriminate against the large ß values.

I wanted to add a penalty term to the traditional AIC formula for two reasons - first to decrease the number of selected variables to create a simpler, more economical model, and second to decrease the variance of the resulting population total estimates. My penalty term needed to be designed to find the estimated absolute values of all parameters in the candidate model, sum them, and add them to the AIC value of the candidate model. This theoretically makes it harder to select additional variables, especially heavily weighted variables.

To do this, inside of the forward selection function, the model parameters would need to be estimated for every iteration in order to add the correct penalty term to each candidate model. This could not be done using a canned selection program as I had been using previously. I was forced to write my own. In Appendix C 1.1 is a table of the results of

this function being executed several times without the extra penalty term with the purpose of convincing the readers that it produces comparable results to the canned selection program.

Conclusion

The GREG estimator really is unbiased:

In Appendix D 1.1, "bias tygreg" refers to the average GREG y total estimation from all 100 samples taken in a particular execution of the variable selection program. As you will see in Appendix A, "Bias tygreg" does not depend on k (A 1.1), or n (A 1.2). Furthermore, the values of "Bias tygreg" look to be approximately normally distributed about mean 0, and standard deviation ~200. To test this intuition, I plotted a histogram of all "Bias tygreg" values from all scenarios with an overlaid normal distribution of the empirical mean/standard deviation of the "Bias tygreg" values (~3, 222) (A 1.3). I can do this because of the lack of dependency of "Bias tygreg" on n and k, the only changing user controlled parameters of the simulation. The normal distribution does not fit perfectly, but keep in mind that there were only 15 values of "Bias tygreg" to draw from. Also note that the standard deviation is relatively very small as the real population total is on the order of 80,000.

I also wished to evaluate the accuracy of the theoretical naive variance of tyGREG-hat, because it does not depend on the selection criterion, I simply recorded empirical standard deviations of tyGREG-hat to compare to the theoretical calculations for each n value (200,500,800). These comparisons can be found in Appendix D 1.1.

The ridge regression estimator is biased:

In regard to the same 100 surveys as explained above, "Bias ty ridge" in Appendix D 1.1 refers to the bias of the average of all surveys in a particular execution of the variable selection program. Ridge regression theoretically yields smaller variances than using ordinary least squares at the price of adding bias to the estimator (citation). Theoretically ridge regression should underestimate Y totals as well as parameter values (citation). Like "bias tygreg," "Bias ty ridge" does not depend on n or k (see Appendix A 1.4 and A 1.5 if you need convincing). Therefore, it makes sense to average all values obtained over executions of the forward selection program in all user selected scenarios to deduce the typical y total estimation bias when using ridge regression to estimate parameters. Unlike the average of "bias tygreg" which was essentially 0 (3 is close enough to 0 when referring to totals on the order of tens of thousands), the average of "Bias ty ridge" was -834 with a standard deviation of 262. It is worth noting that each and every discrete value of "Bias ty ridge" was negative. It is difficult to explain why the standard deviation using ridge regression is actually higher than using ordinary least squares through general regression estimation as it should theoretically be opposite. However, the difference is very slight and is likely largely attributed to the small sample size of just 15 values for each estimation.

Larger sample sizes lead to more selected variables:

Each and every time I increased my survey sample size (by increments of 300) while keeping the selection criterion constant, at least one more variable was selected. When one looks at the AIC formula, one would be inclined to intuit that since the penalty term is made harsher by a larger n, fewer variables, not more, would be selected. However, as that is not the case, I hypothesize that while the penalty term is harsher, it is harsher for all models which does not lead to any differences in selection, but it is possible that the larger n values groups AIC values for models more closely together thereby making it "easier" for the program to squeeze in an extra variable before minimizing the AIC. See Appendix B 1.1,1.2, and 1.3 for graphs related to the above material.

Theoretical (Naive) Variance of the $t_{y,GREG}$ *estimator is constant across all large sample sizes up to N.*

In Appendix C 1.1, you will see the theoretical standard deviations of the $\hat{t}_{y,GREG}$ estimators calculated from input parameters of each execution of the forward selection program. While at first the graph can appear to be of just averages of the $t_{y,GREG}$ standard deviations across all survey samples of a given run, notice that these values are actually overlaid on top of the theoretical $t_{y,GREG}$ standard deviations of the whole population. The standard deviations are on the order of hundreds whereas the differences between them are single digits.

Ridge regression parameter estimation is as good as advertised.

Whereas the variance of "Bias ty ridge" wasn't as good as we would theoretically suppose it to be, the accuracy of the ridge parameter estimator ("beta hat ridge") was better than the least squares estimator as predicted. Referring to Appendix D 1.1, one can see that the ridge parameter estimator was at least as good, if not up to four times better, than the least squares estimator in every scenario studied. While the ridge estimator is biased, it does do a much better job of retrieving the original parameters.

The ridge regression inspired penalty term decreases the number of selected variables for a given sample size:

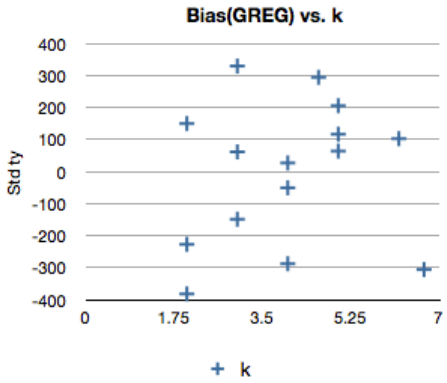
The penalty term did result in selection of one or more fewer variables for each discrete sample size, however, it did not lower the variance of the Y total estimations (it didn't raise it either, all variances

lie in line with those produced by processes not involving the extra penalty term). As expected, when I fed the $t_{y,GREG}$ estimator function the results of the ridge regression inspired forward selection process, it became biased and underestimated the Y total every time the program was executed. This bias was very minor, about a fifth of the mean of "Bias ty ridge." In Appendix D 1.2, 1.3, and 1.4 one can find the graphs from Appendix B (1.1, 1.2, 1.3) modified to include the selection processes including the extra penalty term. Note that k values are approximate as my program was designed to not hold k constant.

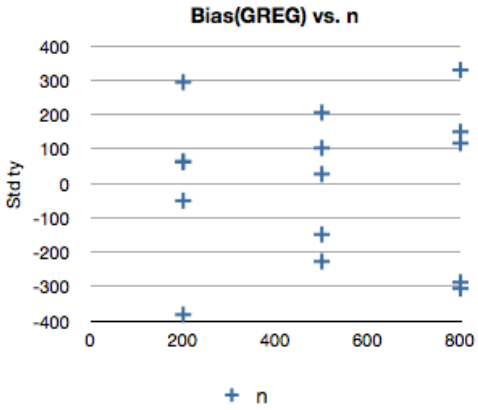
Appendices

Appendix A:

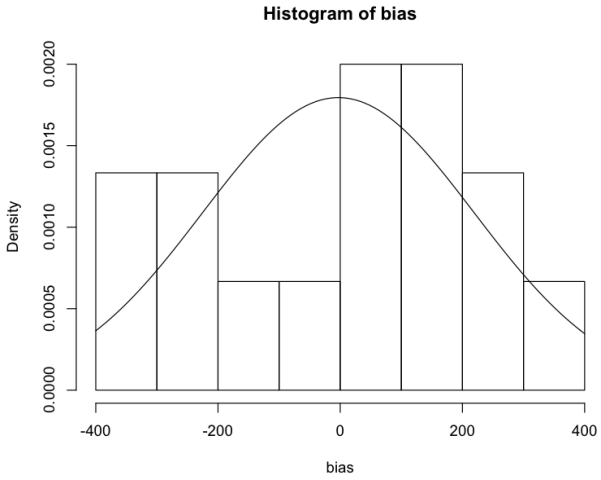
A1.1



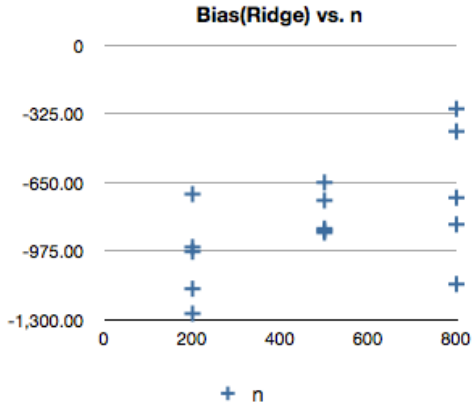
A1.2



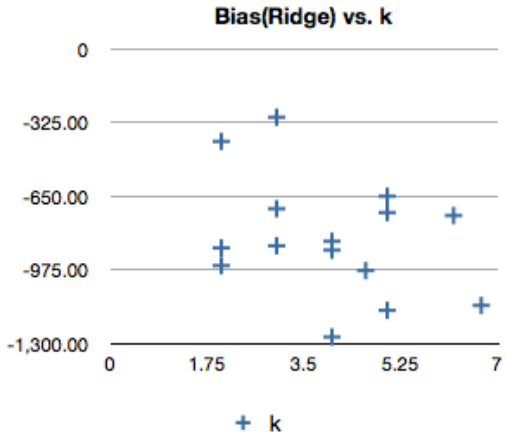
A1.3



A1.4

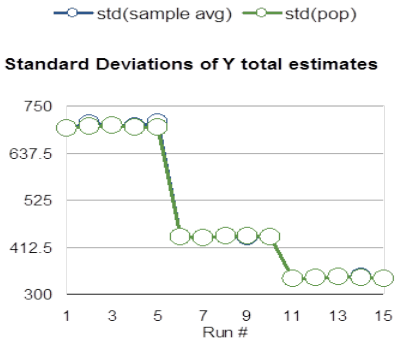


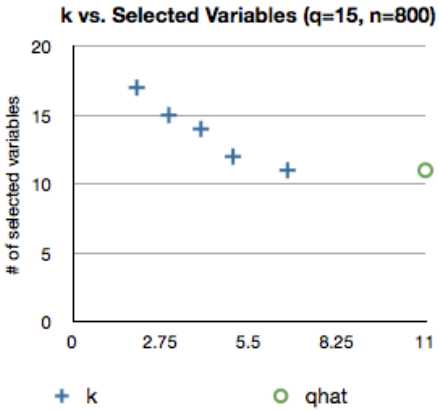
A1.5



Appendix B:

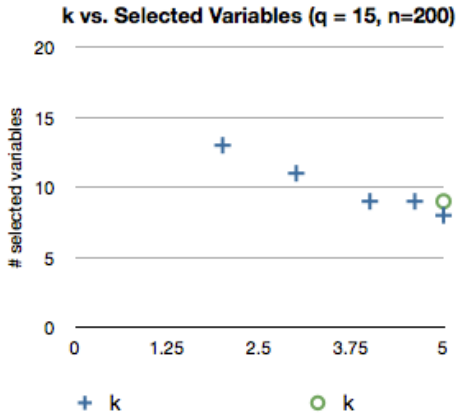
B1.1



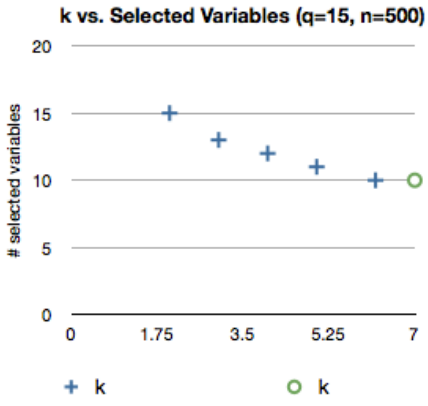


Appendix C:

C1.1



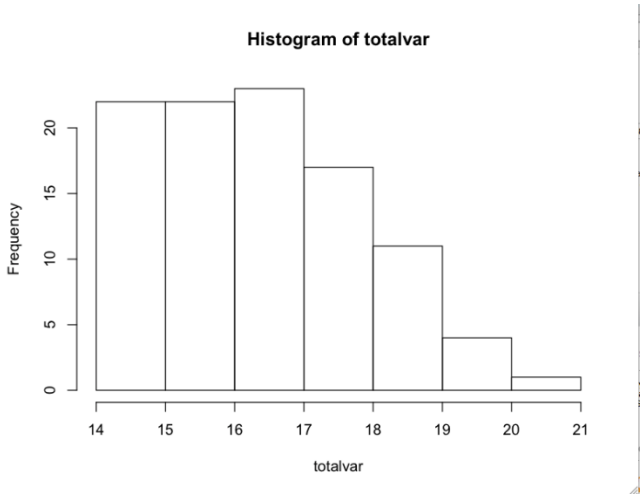
C1.2



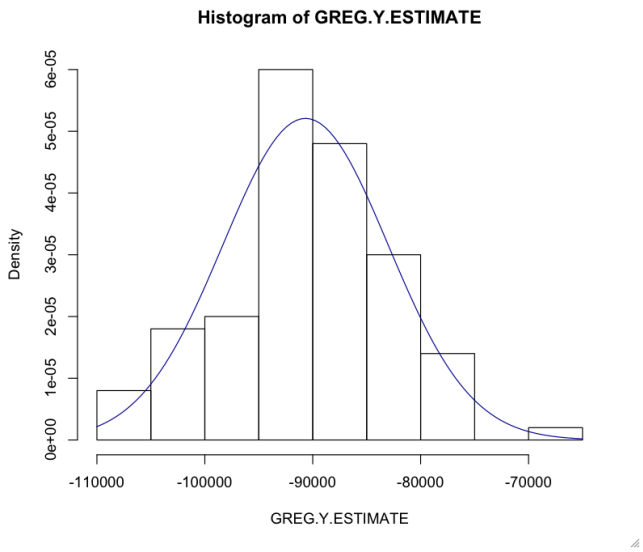
C1.3

Appendix D:

D1.1



D1.2



REFERENCES

1. Breheny, Patrick. “Ridge Regression.” University of Kentucky, 1st Sept. 2013. http://web.as.uky.edu/statistics/users/pbreheny/764-F11/notes/9-1.pdf
2. Lohr, Sharon L. “Sampling: design and analysis”. 2nd ed. Boston, MA. 2010.
3. Shedden, Kerby. “Multiple Linear Regression.” University of Michgan. 1st Jan. 2014. http://dept.stat.lsa.umich.edu/~kshedden/Courses/Stat401/Notes/401-multreg.pdf