David Smith
STAT 4630
Midterm

1.

a. I expect the standard error of the trimmed mean to be smaller than the standard error of the mean, since there will be less variation in the values being averaged. This is because any outliers or particularly large or small values are omitted from the calculation, leading to less variability in the estimate.

b. See R file. I used the bootstrap technique to estimate the standard error, since it was not feasible to derive an analytical solution. I simulated the data set and used the boot function to find the standard error of the trimmed mean, together with a function I wrote that returns the trimmed mean. The standard error was estimated as 1.6220.

ci. See R file. I used the replicate function simulate 100,000 data sets, used the apply function to get the trimmed mean of each data set, and found the standard error of these means by taking their standard deviation. This value represents the "true" value of the standard error in this simulation.

cii. See R file. I simulated 100 data sets for this part, since any significantly higher amount was too computationally intensive. I used the apply function to apply the boot function to each data set, and then used a for loop to extract and store the 100 standard errors. These represent a sample of standard errors, which we will compare to the "true" standard error found in part ci.

ciii. See R file. I used the formulas for the bias and variance to calculate these values. For the bias, I took the difference of each observed error and the "true" error (which represents the value that would be given by the Bayes predictor), and found the mean of these differences. For the variance, I simply used the var function on the observed standard errors.

2.

See R file. We first observe that we have eight predictors, each with 1030 observations. Since we are dealing with a prediction problem, we can rule out using any classification methods. We are led immediately to consider regression to predict, as we always should. We can rule out using ridge and lasso regression and principal component analysis, as we are not dealing with high-dimensional data, with the number of observations being much larger than the number of predictors. Furthermore, ridge regression will include all eight predictors, which may be unnecessary if some of them are insignificant and which leads to poor model interpretability. Lasso regression avoids this problem, but often gives performs worse than ordinary regression on low-dimensional data. We may want to consider polynomial regression if our data is highly nonlinear, so we proceed to examine the data more closely. Splines could also be helpful in this case, but could be very difficult to manage since we have eight predictors, so we will avoid using them unless the data strongly suggests otherwise.

First, by looking at the data in Excel, we observe many repeated values for our predictors, including the presence of a large number of zeros. This suggests that our predictors may sometimes not be relevant in the presence of certain values of other predictors, so we may want to consider omitting predictors if they are not significant in our regression. We now plot each predictor against each other and against the response variable in order to check for linearity, outliers, and any other observations that could be relevant. We see that the data is very much spread out, although appears vaguely linear, so we may want to use transformations to help emphasize the linearity of some of the predictors. Looking at the plots for the response variable against each predictor, we find the following. The data for the first predictor appears to curve a bit, and spreads out near large values, so a transformation could be appropriate. The data for the second predictor contains many zeros, and is fairly evenly spread. The data for the third predictor contains many zeros and some repeated low values, but most of the values are clustered between about 75 and 200. The data for the fourth predictor contains some repeated values, but is otherwise evenly spread. The data for the fifth predictor contains many zeros, and curves up and to the right, spreading out there, so a transformation might be helpful. The data for the sixth predictor appears evenly spread, along with the data for the seventh predictor. The data for the eighth predictor appears somewhat curved and spreads out towards the right. However, it consists of a large number of repeated values, so it may be better to consider it a categorical variable, rather than using a transformation. In summary, our data indicates that with a few tweaks and modifications, linear regression will be the most appropriate method to predict y.

We now check the correlations between variables. We do see some large values, particularly with the correlation of -0.657 between the fourth and fifth predictors. We may want to consider removing one of these variables, but for now will proceed with analyzing the regression. Before running a regression with the lm function, we convert the eighth predictor to a factor, so that R will treat it as a categorical variable in the regression. We then run a linear regression with all of the predictors. We see that the fifth predictor is highly insignificant, so we will remove it, which also solves the problem of the high correlation. The rest of the variables are quite significant. Our adjusted R squared value is 0.8319, which is quite high, and the residual standard error is 6.848. Looking at the residual plots, we do see some problems, including increasing variance in the residuals and a lack of normality, but it doesn't seem too bad. And in the end, we only care about the model performing well.

We proceed to remove the insignificant fifth predictor, and run the regression again. We see that the adjusted R squared value has increased slightly to 0.832, and the residual standard error has decreased to 6.847. Furthermore, all of our variables are significant. However, while this seems like a good model, we may be able to improve it by considering some of the transformations mentioned earlier. As already mentioned, the first predictor seemed like a good candidate for a transformation, particularly a log or square root in this instance. We plot both of these transformations, and observe that the log seems to make the data the most linear and evenly spread. We proceed to create a new column of values consisting of the logarithm of the first predictor, and replace any instances where the logarithm is negative infinity by 0 (i.e., when the original variable was zero). We now run a regression with all of the predictors, and the first predictor replaced by its logarithm. We see that now the fifth, sixth, and seventh predictors are all extremely insignificant, so we proceed to remove them one at a time, checking to make sure they do not become significant again as others are removed. We end up with a model consisting of the logarithm of the first predictor, the second predictor, the third predictor, the fourth predictor, and the eighth predictor as a categorical variable. We have that the adjusted R squared value is 0.8287, and the residual standard error is 6.915. These values are slightly worse than the model that had all predictors except for the fifth. However, the current model is simpler and has superior interpretability, and the differences in the adjusted R squared and residual standard error values are very small, so we choose to keep it as our final model. None of the remaining predictors would seem to benefit from a transformation, and while the residual plots show some issues, we will decide to reject our model only when it performs poorly on test data, so we have arrived at the final model.

To see how well the model predicts, we use it to predict a y value for each x value in the training data, and find the mean squared error, which turns out to be 46.982. This doesn't seem bad at all, considering the sizes of the data points. Thus, our model seems adequate to use for prediction. I decided not to use a validation test set approach since I wasn't comparing multiple different models; in this case it is probably better to use all of the data to train a model.

3.

a. The purpose of constraining the coefficients of the regression is to bias our model, in hopes of reducing the variance even more. We are deliberately choosing to penalize large coefficients, which gives a model with a larger training MSE than ordinary regression, but which will hopefully result in a lower test MSE by adding bias and reducing variance, optimizing the bias-variance tradeoff.

b. On the left-hand side of the sum we have the residual sum of squares, as in the ordinary lasso, but now on the right-hand side we are penalizing the sizes of the coefficients of groups of covariates as a whole, rather than the sizes of the coefficients individually. This is clear from the $||\beta^p||_2$ term, which takes the root sum of squares of all of the coefficients of each group, and it is that value that is penalized. The term $\sqrt{q_p}$ seems to account for the size of each group, so that all groups are penalized somewhat equally, regardless of size. Thus, the coefficients of covariates in each group are freer to vary than in ordinary lasso regression, but as a whole, they are still pulled to 0 as $\lambda$ increases. In summary, this method is called the group lasso because groups of coefficients are penalized and drawn towards 0 as a whole, rather than each coefficient individually.