# Homework #8

Justin Robinette

March 19, 2019

*No collaborators for any problem*

**Question 6.8.4, pg 260:** Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

for a particular value of $\lambda$. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

*i. Increase initially, and then eventually start decreasing in an inverted U shape.*
*ii. Decrease initially, and then eventually start increasing in a U shape.*
*iii. Steadily increase.*
*iV. Steadily decrease.*
*v. Remain constant.*

**Part A:** As we increase **lambda** from 0, the training RSS will:

**Results: iii** - Training error increases steadily because of less flexibility in the model

**Part B:** As we increase **lambda** from 0, the test RSS will:

**Results: ii** - Test error will decrease initially and then increase because, as the model becomes less flexible, the initial response will be a decrease in test RSS. Then it will increase again in a U shaped pattern.

**Part C:** As we increase **lambda** from 0, the variance will:

**Results: iv** Variance will decrease steadily because of more constraints

**Part D:** As we increase **lambda** from 0, the (sqaured) bias will:

**Results: iii** - Bias will steadily increase because of less flexibility in the model

**Part E:** As we increase **lambda** from 0, the ireeducible error will:

**Results: v** - Irreducible error is a constant value, therefore it remains constant

**Question 6.8.9, pg 263:** In this exercise, we will predict the number of applications received using the other variables in the **College** data set.

**Part A:** Split the data into a training and test set.

**Results:** The data set has been split with 70% of obs in training and 30% of obs in test. A table shows the number of observations in the total data set, and the training and test sets.

| College | College Training | College Test |
|---|---|---|
| 777 | 543 | 234 |

**Part B:** Fit a linear model using least squares on the training set, and report the test error obtained.

**Results:** I have fit a linear model on the training set and used it to predict for the test set. I then calculated the mse and reported it below.

```
## [1] "The test error rate obtained by the linear model is: 1508333.69105546"
```

**Part C:** Fit a ridge regression model on the training set, with lambda chosen by cross-validation. Report the test error obtained.

**Results:** First, I created matrices out of the train and test sets using *model.matrix()* and used *cv.glmnet()* to fit the ridge regression model. I then chose the best lambda from the model and used it in the prediction. Finally, I reported the MSE, which is higher than in was with the least squares linear model from Part A.

```
## [1] "The test error rate obtained by the ridge regression model is: 1593604.21005617"
```

**Part D:** Fit a lasso model on the training set, with lambda chosen by cross-validation. Report the test error obtained, as well as the number of non-zero coefficient estimates.

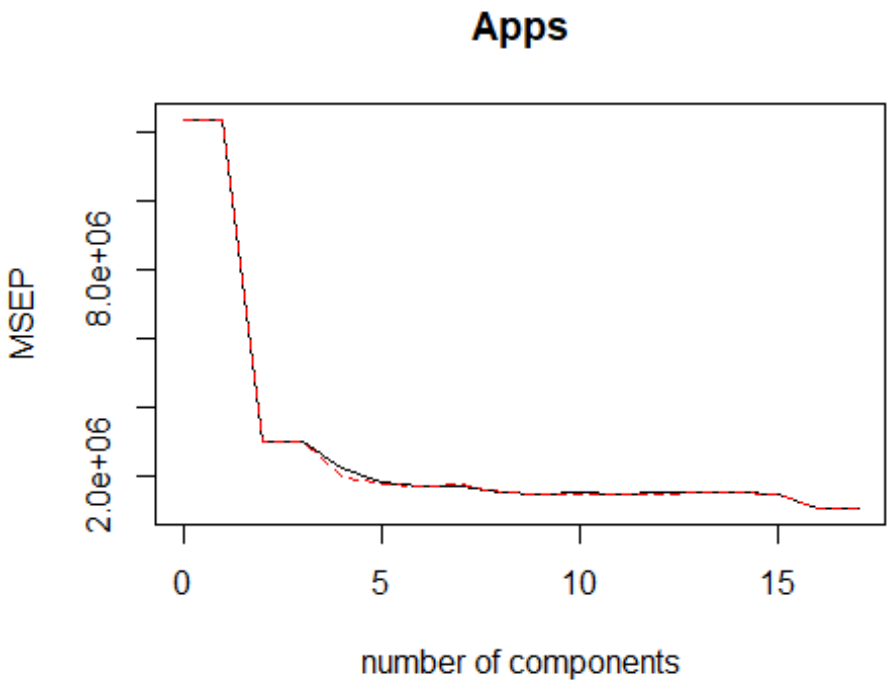
**Results:** Here I fit a lasso model on the training set and chose the best lambda from the model. I then reported the test error rate and the number of non-zero coefficient estimates (**15**).

```
## [1] "The test error rate obtained by the lasso model is: 1585895.4525594"

## [1] "The number of non-zero coefficient estimates is: 15"
```

**Part E:** Fit a PCR model on the training set, with *M* chosen by cross-validation. Report the test error obtained, along with the value of *M* selected by cross-validation.

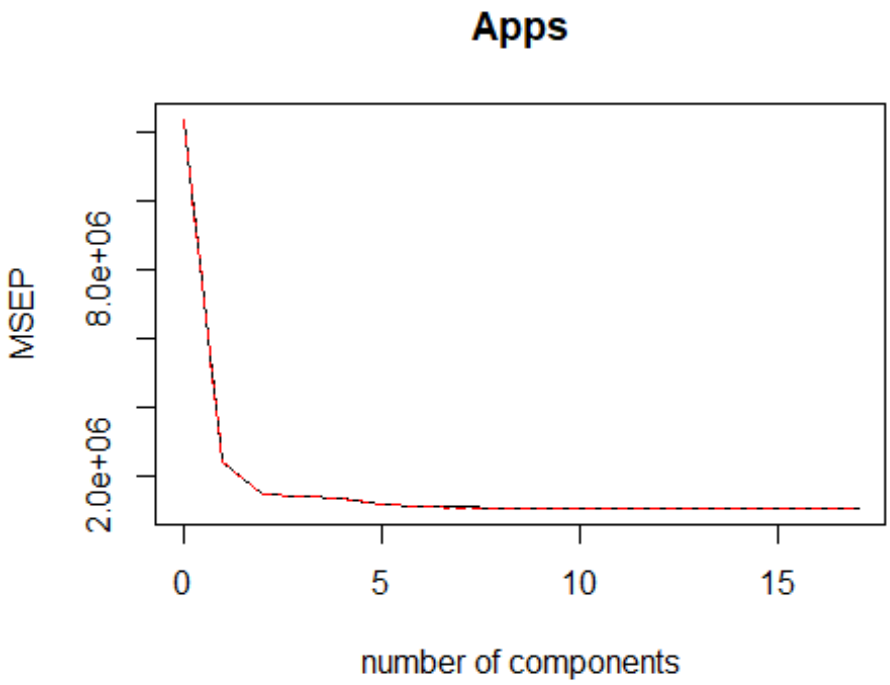
**Results:** First I fit the PCR model on the training set and plotted the validation plot showing Mean Square Error of Prediction by number of components. Based on this plot, it appears that the cross validation is suggesting that we should use all predictors in the model. I used *M* to include all predictors and the *predict()* function to obtain predictions and the MSE, which is reported below.




```
## [1] "The test error rate obtained by the PCR model is: 1508333.69105545"
```

**Part F:** Fit a PLS model on the training set, with *M* chosen by cross-validation. Report the test error obtained, along with the value of *M* selected by cross-validation.

**Results:** First I fit the PLS model on the training set and plotted the validation plot showing Mean Square Error of Prediction by number of components. Based on this plot, it appears that the MSEP remains relatively constant at 10 components. I used *M* to include 10 components and the *predict()* function to obtain predictions and the MSE, which is reported below.




```
## [1] "The test error rate obtained by the PLS model is: 1558091.75473337"
```

**Part G:** Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these 5 approaches?

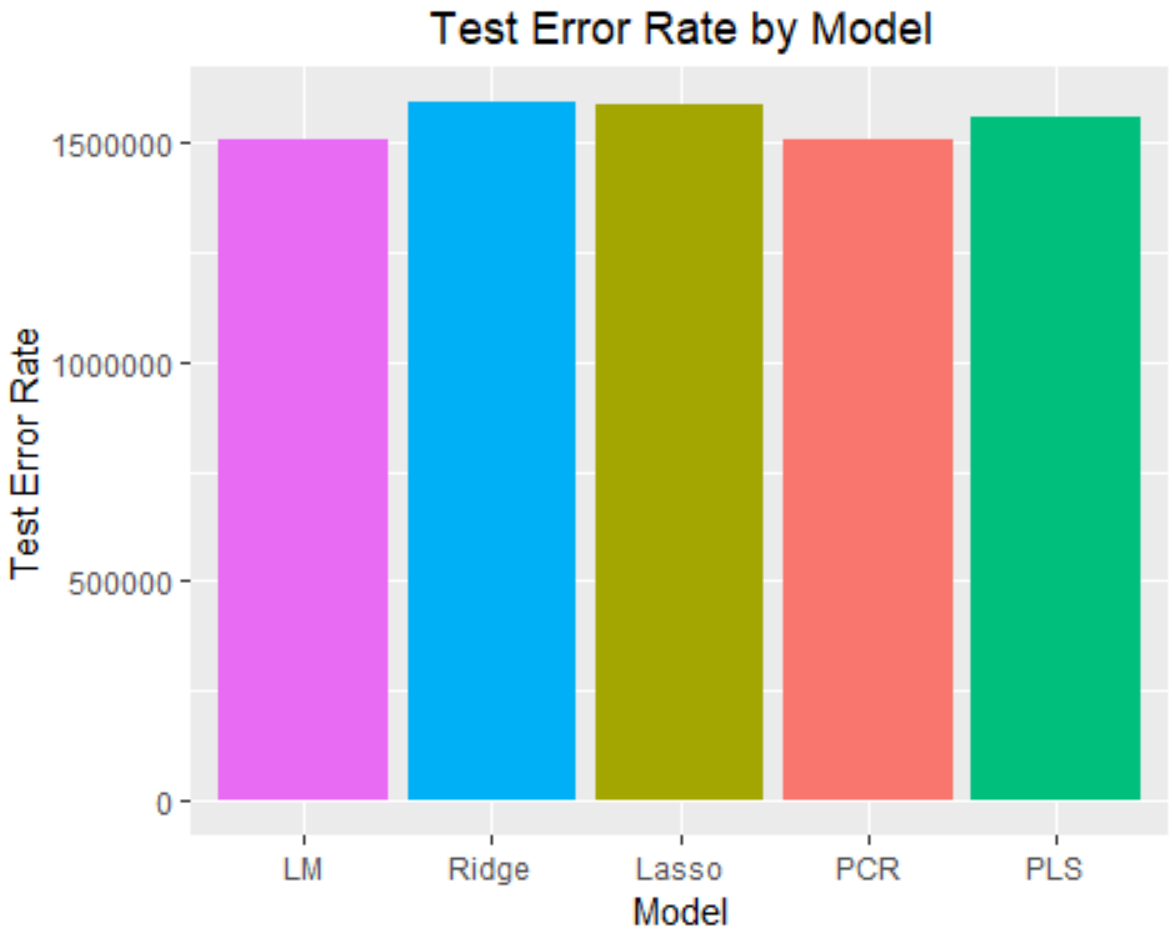
**Results:** As we can see from the table below, PCR and LM had the lowest test error rate, followed by PLS and then Lasso. Ridge was the worst performing.

I also constructed a barplot to show that there is not much different among the test errors resulting from the different methods.

Additionally, we are not able to predict the number of college applications received well.

***MSE by Model Type***

| LM | Ridge | Lasso | PCR | PLS |
|---|---|---|---|---|
| 1508334 | 1593604 | 1585895 | 1508334 | 1558092 |




**Question 6.8.11, pg 264:** We will now try to predict per capita crime rate in the **Boston** data set.

**Part A:** Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss resultls for the approaches that you consider.

**Results:** Here I've loaded the data set and split into test and train.

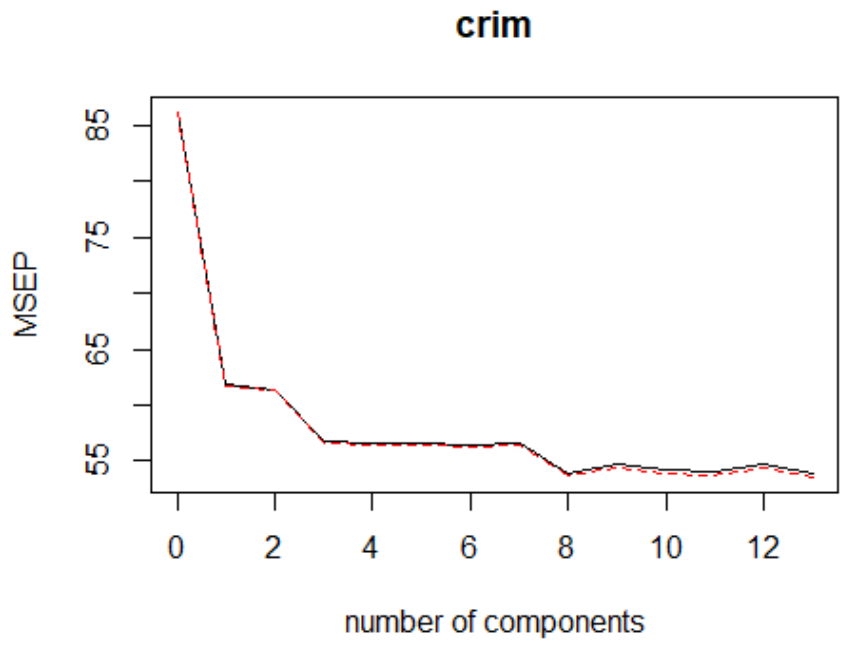
| Boston | Boston Training | Boston Test |
|---|---|---|
| 506 | 354 | 152 |

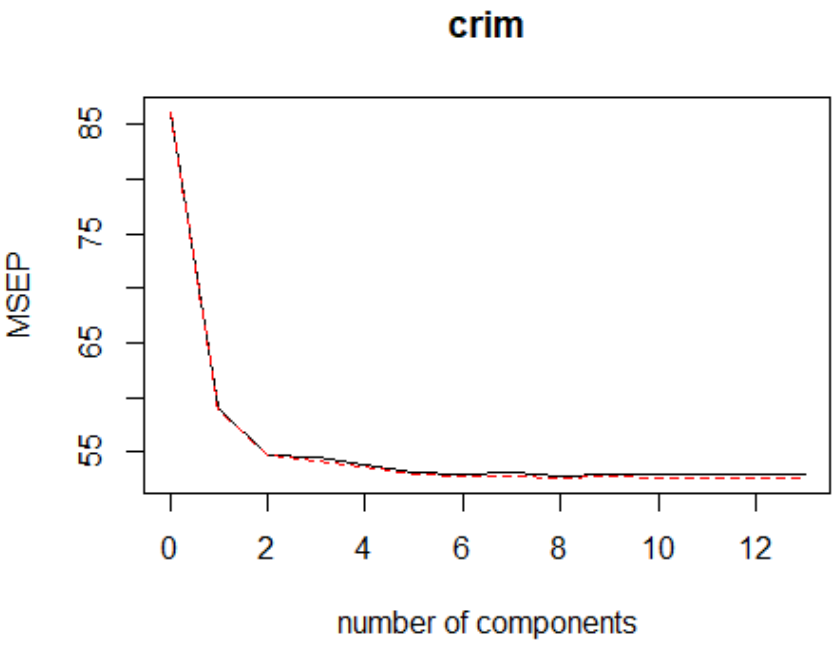
**Part B:** Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative (not training error).

**Results:** For this step, I used the following modeling methods: - Linear Model - Ridge Regression - Lasso Model - Forward Stepwise Selection - Backward Stepwise Selection - Principal Component Regression - Partial Least Squares Regression

From the table below, we can see that the Lasso model performed the best with a test error of **22.17753**. The worst performing models were the LM, PCR, and PLS with a test error of **23.49987**.

Not only did the Lasso model perform the best, but it also eliminates some predictors which simplifies the model.



***MSE by Method***

| LM | Ridge | Lasso | Fwd Stepwise | Back Stepwise | PCR | PLS |
|---|---|---|---|---|---|---|
| 23.49987 | 22.99736 | 22.17753 | 22.9561 | 22.1828 | 23.49987 | 23.49987 |

**Part C:** Does your chosen model involve all of the features in the data set? Why or why not?

**Results:** No, the Lasso model does not involve all predictors. As we can see below, *age* and *tax* are not included in the model as shown by the 0 coefficient estimates.

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  2.930221109
## (Intercept)  .
## zn           0.025011633
## indus       -0.049470844
## chas        -1.044851410
## nox         -0.609285907
## rm           0.190313168
## age          .
## dis         -0.378486434
## rad          0.484672879
## tax          .
## ptratio     -0.095523368
## black       -0.002582815
## lstat        0.189722651
## medv        -0.120892120
```