STAT 4510/7510 - HW6

Yang, Anton - #14405729

Instructions: Please list your name and student number clearly. In order to receive credit for a problem, your solution must show sufficient detail so that the grader can determine how you obtained your answer.

Submit a single pdf generated using R Markdown. All R code should be included, as well as all output produced. Upload your work to the Canvas course site.

Problem 1 (there is only one problem, in several parts)

The file `CommunityCrime.csv` is a dataset containing 319 observations on 123 variables. The observations are communities within the United States. The data combines socioeconomic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI Uniform Crime Reporting program. A detailed description of all variables is available at http://archive.ics.uci.edu/ml/machine-learning-databases/communities/communities.names. We seek to predict the variable `ViolentCrimesPerPop`, the total number of violent crimes per 100,000 people.

Note: when asked to perform cross-validation to select a tuning parameter, be sure to conduct this cross-validation on the training data only, *then* see how well your cross-validated tuning parameter does on the test data.

a) Set a seed of 1 and split the data into a 90% training set, and a 10% test set.

```
set.seed(1)
data<-read.csv("CommunityCrime.csv")

split<-sample(1:nrow(data), size = 0.9*nrow(data))
training_set<-data[split,]
test_set<-data[-split,]
```

b) Fit a linear model using least squares on the training set. Report the test error obtained.

```
set.seed(1)

lm_model<-lm(ViolentCrimesPerPop~.,data=training_set)
linear_prediction<-predict(lm_model,newdata=test_set)
linear_errors<-mean((linear_prediction-test_set$ViolentCrimesPerPop)^2)
linear_errors

## [1] 0.03439512
```

c) Fit a ridge regression model on the training set with λ chosen by cross-validation. Report the test error obtained.

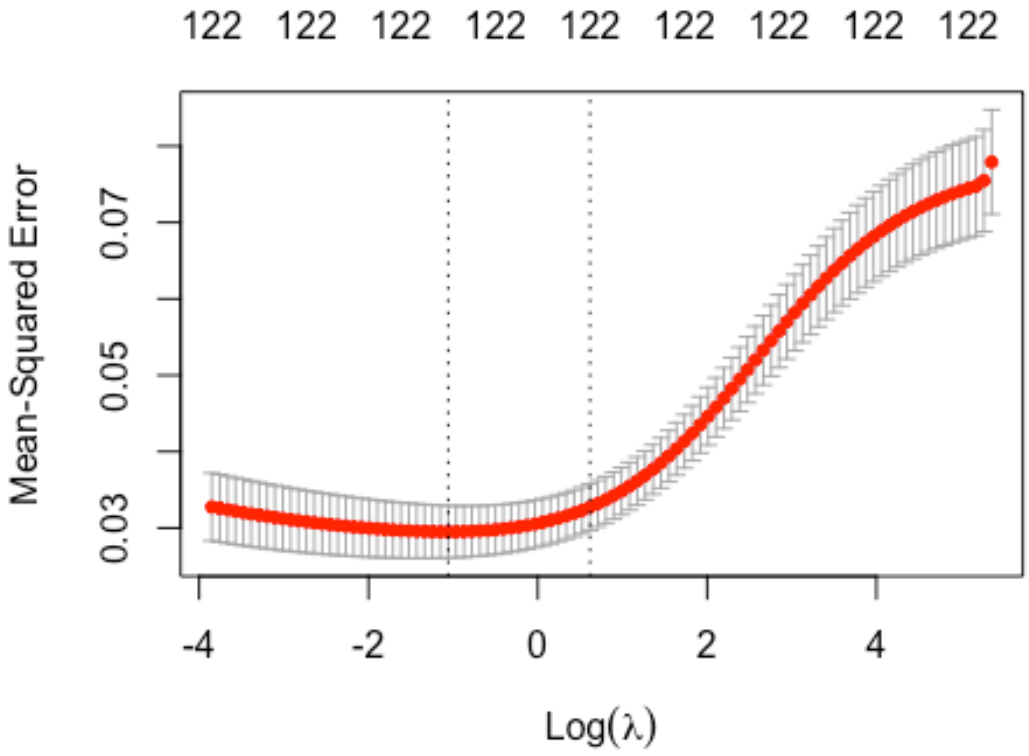
```
set.seed(1)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-7

grid<-10^seq(10,-2,length=100)
training_set_matrix<-model.matrix(ViolentCrimesPerPop~., data=training_set)
test_set_matrix<-model.matrix(ViolentCrimesPerPop~., data=test_set)

ridge_model<-glmnet(training_set_matrix, training_set$ViolentCrimesPerPop,
alpha = 0, lambda = grid)
set.seed(1)
ridge_cv<-cv.glmnet(training_set_matrix, training_set$ViolentCrimesPerPop,
alpha = 0)
plot(ridge_cv)
```




```
best_ridge_lambda<-ridge_cv$lambda.min
best_ridge_lambda

## [1] 0.3469928
```

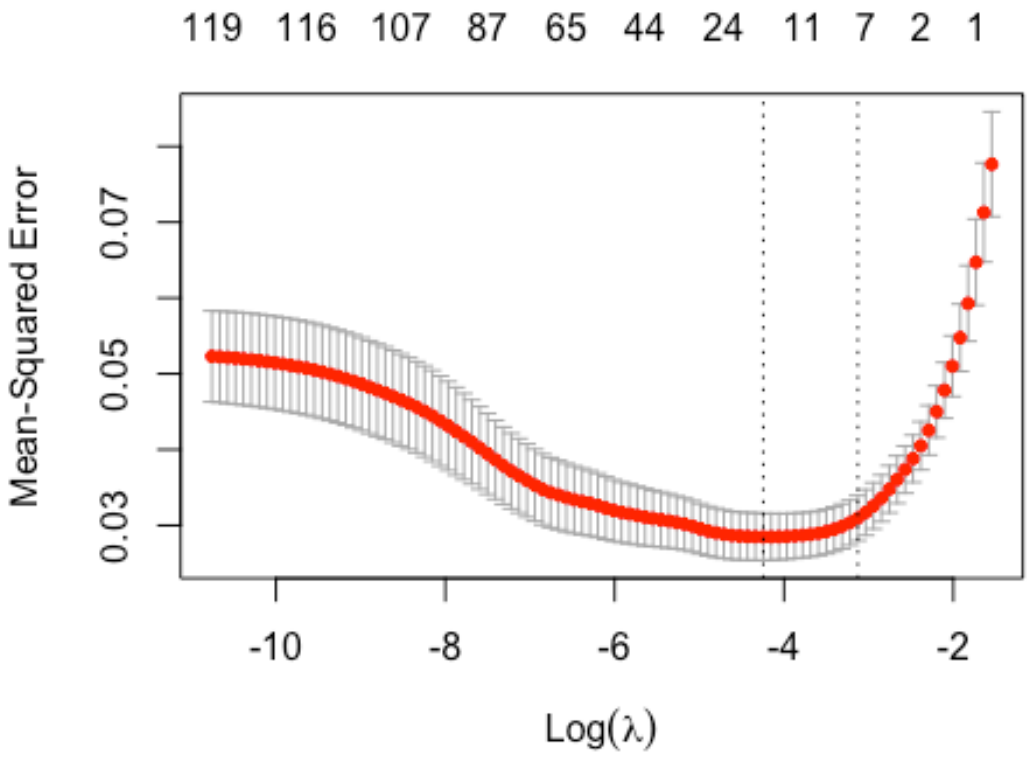
```
ridge_prediction =
predict(ridge_model,newx=test_set_matrix,alpha=0,s=best_ridge_lambda)
ridge_errors<-mean((ridge_prediction-test_set$ViolentCrimesPerPop)^2)
ridge_errors
```

```
## [1] 0.02597019
```

d) Fit a lasso model on the training set with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

Note: For parts e) and f) below: When you are identifying the value of M from the validation plot, you may need to zoom in to see what is happening with a small number of components. In your command, you can add in options like `xlim=c(0,20)` or `ylim=c(0,1)`, adjusting the numeric values as needed to provide a useful view.

```
lasso_model<-glmnet(training_set_matrix, training_set$ViolentCrimesPerPop,
alpha = 1, lambda = grid)
set.seed(1)
lasso_cv<-cv.glmnet(training_set_matrix, training_set$ViolentCrimesPerPop,
alpha = 1)
plot(lasso_cv)
```




```
best_lasso_lambda<-lasso_cv$lambda.min
best_lasso_lambda
```

```
## [1] 0.01433778
```

```
lasso_prediction<-predict(lasso_model, newx = test_set_matrix, alpha = 1,
s=best_lasso_lambda)
lasso_errors<-mean((lasso_prediction-test_set$ViolentCrimesPerPop)^2)
lasso_errors
```

```
## [1] 0.024541
```

```
data_matrix<-model.matrix(ViolentCrimesPerPop~., data=data)
lasso_full = glmnet(training_set_matrix, training_set$ViolentCrimesPerPop,
alpha=1, lambda=best_lasso_lambda)
lasso_coef = predict(lasso_full, type="coefficients", s=best_lasso_lambda)
length(which(lasso_coef!=0))
```

```
## [1] 19
```

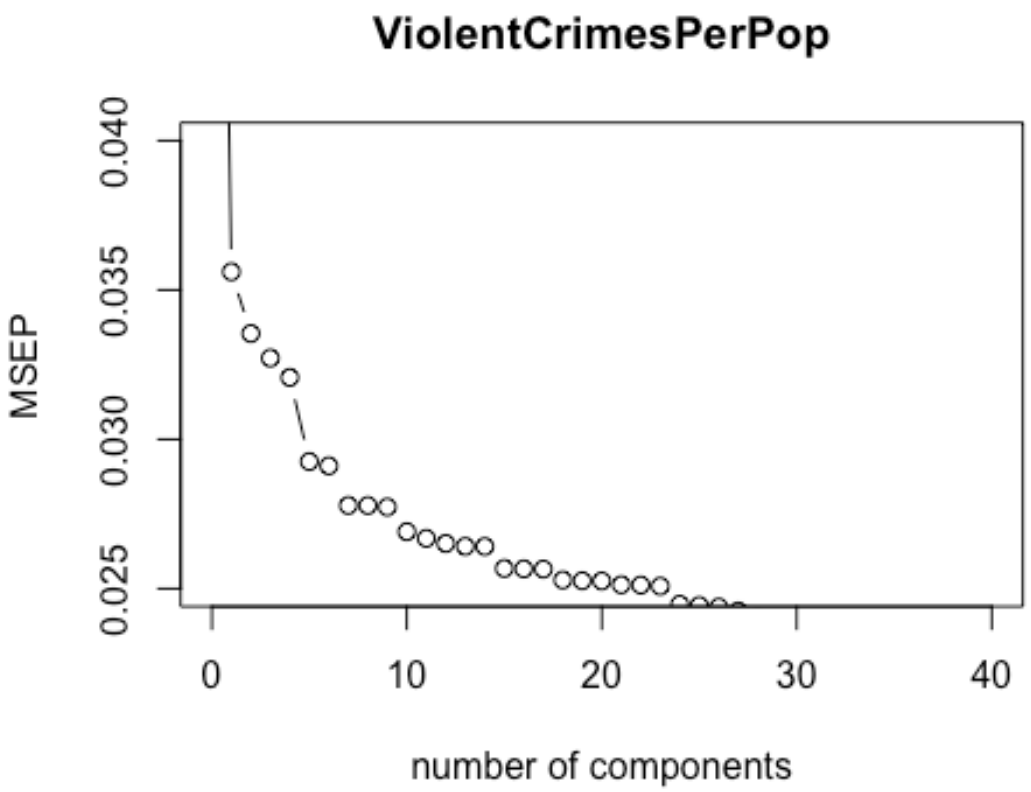
e) Fit a PCR model on the training set with M chosen by cross-validation. Report the test error obtained along with the value of M selected by cross-validation.

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
set.seed(1)
pcr_model<-pcr(ViolentCrimesPerPop~., data=training_set)
validationplot(pcr_model,val.type="MSEP", xlim = c (0,40), ylim = c(.025,
.04), type = "b")
```



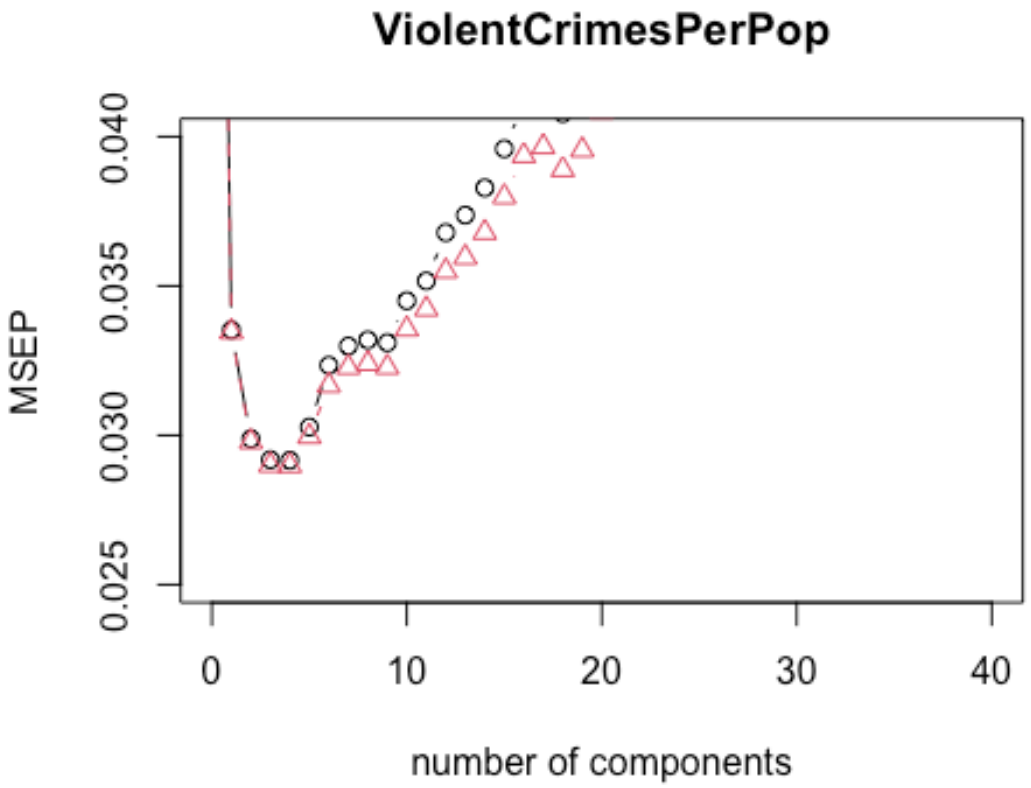

```
pcr_prediction<-predict(pcr_model,test_set,ncomp=8)
pcr_errors<-mean((pcr_prediction-test_set$ViolentCrimesPerPop))
pcr_errors
```

```
## [1] 0.05076923
```

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.

```
set.seed(1)

pls_model <- plsr(ViolentCrimesPerPop ~ ., data = training_set, scale = TRUE,
validation = "CV")
validationplot(pls_model, val.type="MSEP", xlim=c(0,40), ylim=c(.025,
.04),type="b")
```




```
pls_prediction <- predict(pls_model, newdata = test_set, ncomp = 4)
pls_errors <- mean((pls_prediction - test_set$ViolentCrimesPerPop)^2)

pls_errors
```

```
## [1] 0.0227858
```

g) Comment on the above parts and how well you believe we can predict violent crime rate using these methods.

I believe that these models are better than Ordinary Linear Model because these methods are able to eliminate coefficients that are not relevant to the response variable. We can see that all of the models are better than linear model except for the PCR. PCR does not perform compare to the other model because response Y is not used to help determine the principal component directions which there is no guarantee that the directions that best explain the predictors will be the best for predicting the response. The reason that PLS is better is because it's utilizing both the response and the predictors. We can see that the best model at predicting are PLS with lowest errors. Since there are many predictors, ridge, lasso, pls are good models to predict violent crime rate based on how small the errors are for each of these models.