Problem Set 3

Sravan Ramaswamy

2/11/2021

https://github.com/SravanjR/bc-micro-methods/blob/main/psets/pset_model_selection/Problem-Set-3.pdf

```
#Clear Environment
rm(list = ls())

library(caTools)
library(glmnet)
library(dplyr)
 library(tidyverse)
 library(ISLR)
```

```
#Load Bid Data
DataDir = paste(getwd(), "/boston_cl.csv",sep = "")
Data = read.csv(DataDir,sep = ",")
```

```
set.seed(255)
sample = sample.split(Data$X, SplitRatio = .8)
Data = Data[2:15]
trainData = subset(Data, sample == TRUE)
testData  = subset(Data, sample == FALSE)
```

Question 1

How correlated are these variables?

```
res <- cor(Data)
round(res, 2)
```

```
##          crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio
## crim     1.00 -0.20  0.41 -0.06  0.42 -0.22  0.35 -0.38  0.63  0.58    0.29
## zn      -0.20  1.00 -0.53 -0.04 -0.52  0.31 -0.57  0.66 -0.31 -0.31   -0.39
## indus    0.41 -0.53  1.00  0.06  0.76 -0.39  0.64 -0.71  0.60  0.72    0.38
## chas    -0.06 -0.04  0.06  1.00  0.09  0.09  0.09 -0.10 -0.01 -0.04   -0.12
## nox      0.42 -0.52  0.76  0.09  1.00 -0.30  0.73 -0.77  0.61  0.67    0.19
## rm      -0.22  0.31 -0.39  0.09 -0.30  1.00 -0.24  0.21 -0.21 -0.29   -0.36
## age      0.35 -0.57  0.64  0.09  0.73 -0.24  1.00 -0.75  0.46  0.51    0.26
## dis     -0.38  0.66 -0.71 -0.10 -0.77  0.21 -0.75  1.00 -0.49 -0.53   -0.23
## rad      0.63 -0.31  0.60 -0.01  0.61 -0.21  0.46 -0.49  1.00  0.91    0.46
## tax      0.58 -0.31  0.72 -0.04  0.67 -0.29  0.51 -0.53  0.91  1.00    0.46
## ptratio  0.29 -0.39  0.38 -0.12  0.19 -0.36  0.26 -0.23  0.46  0.46    1.00
## black   -0.39  0.18 -0.36  0.05 -0.38  0.13 -0.27  0.29 -0.44 -0.44   -0.18
## lstat    0.46 -0.41  0.60 -0.05  0.59 -0.61  0.60 -0.50  0.49  0.54    0.37
## medv    -0.39  0.36 -0.48  0.18 -0.43  0.70 -0.38  0.25 -0.38 -0.47   -0.51
```

```
##         black lstat  medv
## crim    -0.39  0.46 -0.39
## zn       0.18 -0.41  0.36
## indus   -0.36  0.60 -0.48
## chas     0.05 -0.05  0.18
## nox     -0.38  0.59 -0.43
## rm       0.13 -0.61  0.70
## age     -0.27  0.60 -0.38
## dis      0.29 -0.50  0.25
## rad     -0.44  0.49 -0.38
## tax     -0.44  0.54 -0.47
## ptratio -0.18  0.37 -0.51
## black    1.00 -0.37  0.33
## lstat   -0.37  1.00 -0.74
## medv     0.33 -0.74  1.00
```

In general, we can see a large amount of correlation between some variables such as between "age" and "nox" or between "tax" and "rad" which may limit their explanatory power in a regression.

Question 2

Estimate the original HR model using the training data. Project the the log(Median House Price) onto all of the other variables. Everything should enter linearly, except for NOx and RM, which should only enter quadratically.

```
# Linear Projection

lm1 <- lm(log(medv) ~ crim + zn + indus + chas + poly(nox,2) + poly(rm,2) + age + dis + rad + tax + ptra
```

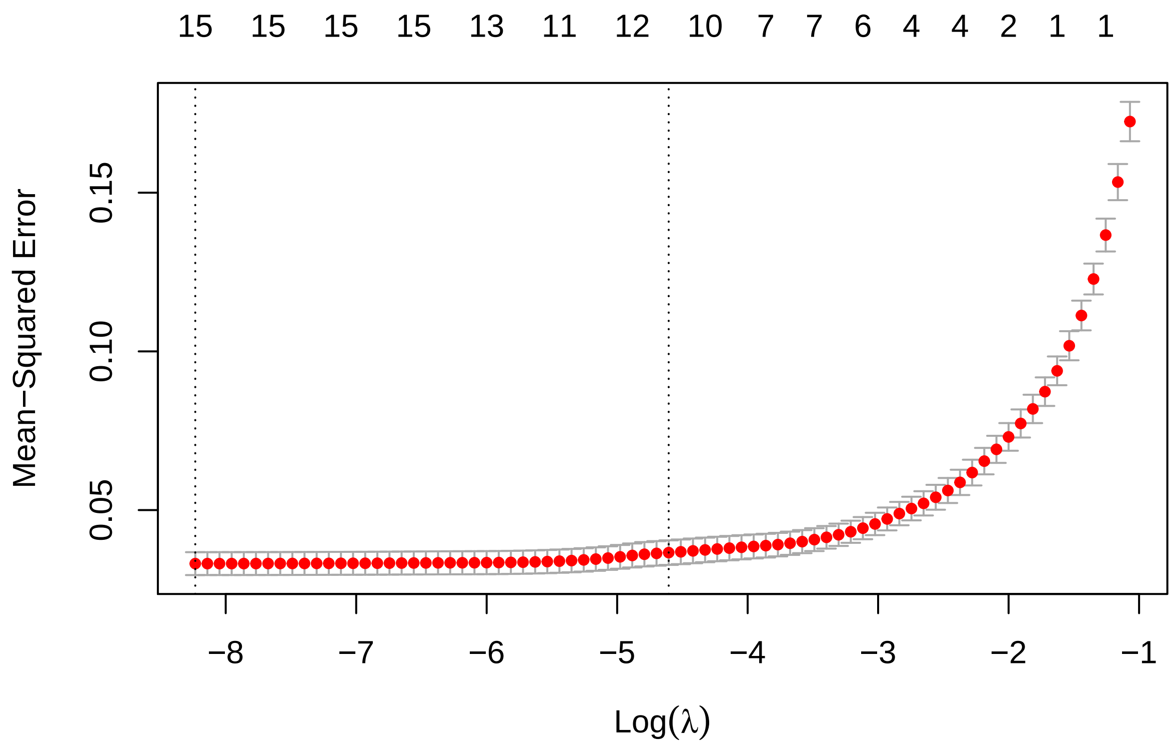
Question 3

Now estimate the model using LASSO. Use k=10 fold cross validation to select lambda. Select the model with the largest lambda (penalty) such that the MSE is within one standard error of the minimum MSE.

```
x_train = model.matrix(log(medv) ~ crim + zn + indus + chas + poly(nox,2) + poly(rm,2) + age + dis + rad

x_test = model.matrix(log(medv) ~ crim + zn + indus + chas + poly(nox,2) + poly(rm,2) + age + dis + rad

y_train = trainData %>%
  select(medv) %>%
  unlist() %>%
  as.numeric()

y_train <- log(y_train)

y_test = testData %>%
  select(medv) %>%
  unlist() %>%
  as.numeric()

y_test <- log(y_test)
```

```
lasso_mod = cv.glmnet(x_train,
                   y_train,
                   alpha = 1, nfolds = 10) # Fit lasso model on training data
```

```r
plot(lasso_mod)
```




```r
print(paste("MSE with the largest lambda within one standard error or the minimizing lambda: ", lasso_m
```

```
## [1] "MSE with the largest lambda within one standard error or the minimizing lambda:  0.036592957884
```

```r
print(paste("Log lambda for this MSE:",log(lasso_mod$lambda.1se))) # Log lambda for this MSE
```

```
## [1] "Log lambda for this MSE: -4.60508530952952"
```

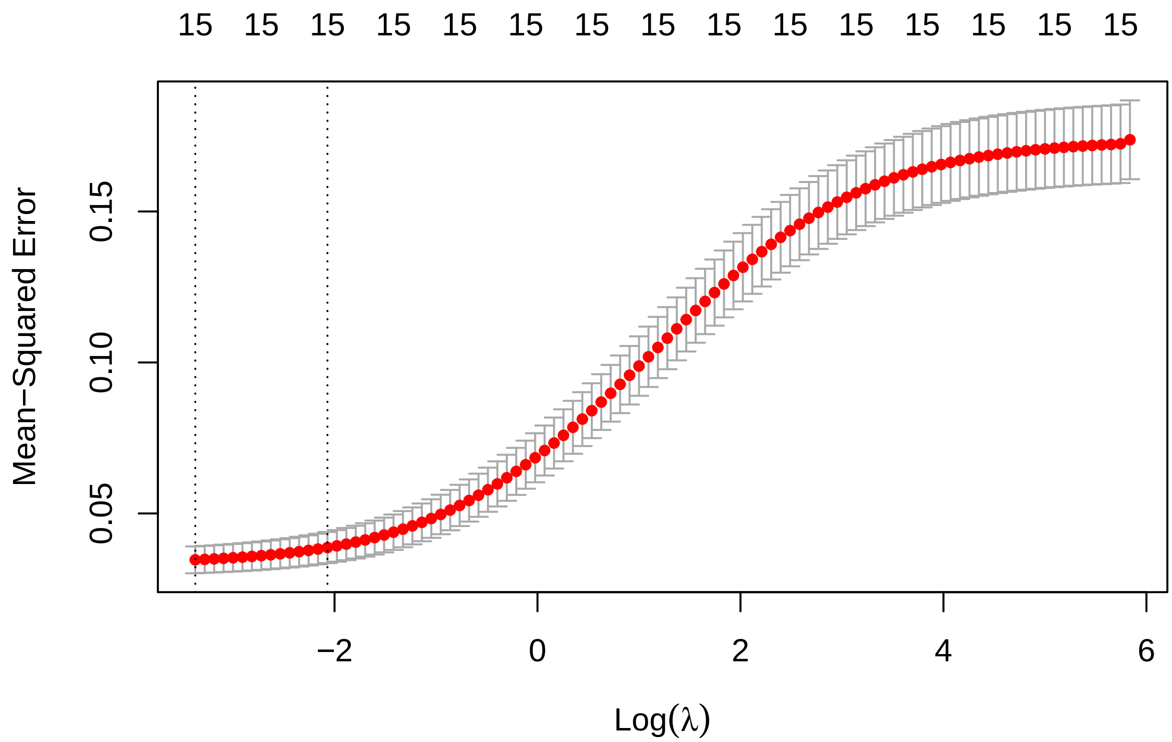
```r
print(paste("Number of Coefficients: ",lasso_mod$nzero[lasso_mod$lambda == lasso_mod$lambda.1se] )) # N
```

```
## [1] "Number of Coefficients:  10"
```

Question 4

Do the same thing for ridge regression. Select the model with the largest lambda (penalty) such that the MSE is within one standard error of the minimum MSE.

```r
ridge_mod = cv.glmnet(x_train,
                      y_train,
                      alpha = 0, nfolds = 10) # Fit ridge model on training data
plot(ridge_mod)
```




```
print(paste("MSE with the largest lambda within one standard error or the minimizing lambda: ", ridge_mo
```

```
## [1] "MSE with the largest lambda within one standard error or the minimizing lambda:  0.0387039685713
```

```
print(paste("Log lambda for this MSE:",log(ridge_mod$lambda.1se))) # Log lambda for this MSE
```

```
## [1] "Log lambda for this MSE: -2.06991585963551"
```

```
print(paste("Number of Coefficients: ",ridge_mod$nzero[ridge_mod$lambda == ridge_mod$lambda.1se]))
```

```
## [1] "Number of Coefficients:  15"
```

```
 # No. of coef | 1-SE MSE
```

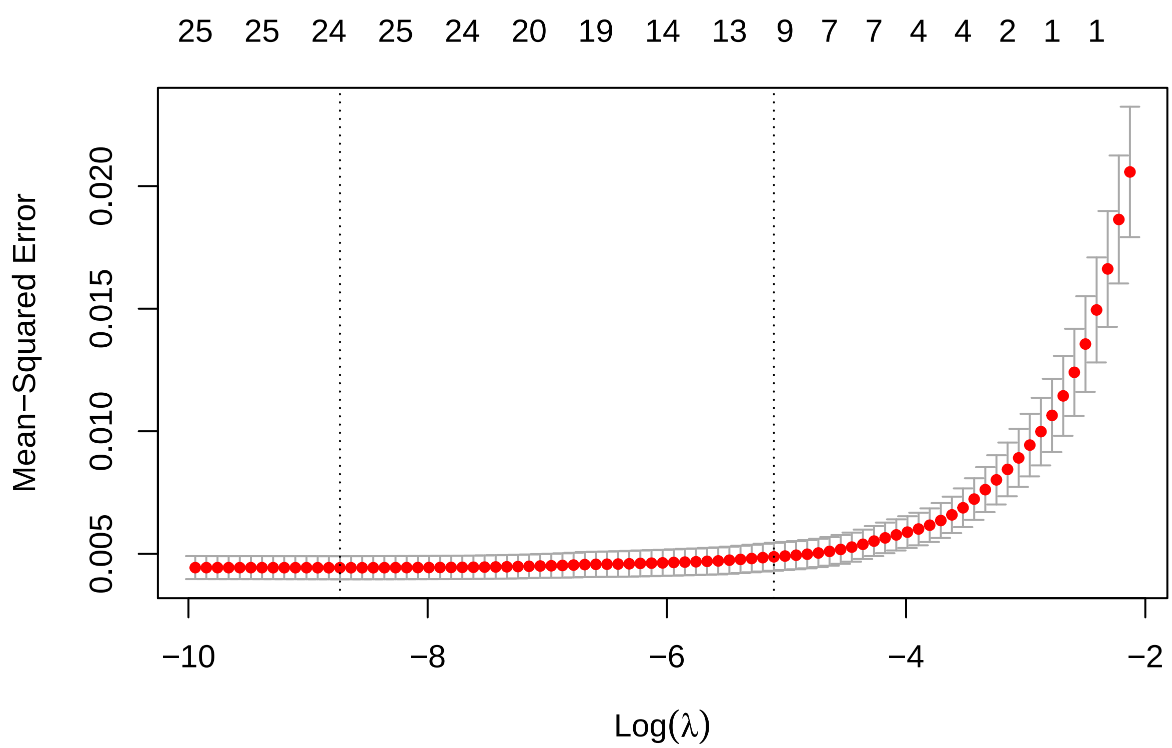
Question 5

HR's decision to have only NOx and RM enter quadractically seems sort of arbitrary. Expand the data to contain the square term of all variables. Then run Lasso on this expanded data set. Which coefficients survive now?

```
x_train2 = model.matrix(log(medv) ~ poly(crim,2) + poly(zn,2) + poly(indus,2) + chas + poly(nox,2) + pol

x_test2 = model.matrix(log(medv) ~ poly(crim,2) + poly(zn,2) + poly(indus,2) + chas + poly(nox,2) + poly

y_train2 = trainData %>%
  select(medv) %>%
  unlist() %>%
  as.numeric()
```

```
y_train2 <- log(y_train)

y_test2 = testData %>%
  select(medv) %>%
  unlist() %>%
  as.numeric()

y_test2 <- log(y_test)
```

```
lasso_mod2 = cv.glmnet(x_train2,
                   y_train2,
                   alpha = 1, nfolds = 10) # Fit lasso model on training data

plot(lasso_mod2)
```




```
print(paste("MSE with the largest lambda within one standard error or the minimizing lambda: ", lasso_mo
```

```
## [1] "MSE with the largest lambda within one standard error or the minimizing lambda:  0.004879887429
```

```
print(paste("Log lambda for this MSE:",log(lasso_mod2$lambda.1se))) # Log lambda for this MSE
```

```
## [1] "Log lambda for this MSE: -5.1054510170578"
```

```
print(paste("Number of Coefficients: ",lasso_mod2$nzero[lasso_mod2$lambda == lasso_mod2$lambda.1se] ))
```

```
## [1] "Number of Coefficients:  10"
```

```r
# No. of coef | 1-SE MSE


tmp_coeffs <- coef(lasso_mod2)
data.frame(name = tmp_coeffs@Dimnames[[1]][tmp_coeffs@i + 1], coefficient = tmp_coeffs@x)
```

```
##                  name   coefficient
## 1         (Intercept)  1.102025e+00
## 2     poly(crim, 2)1 -5.644095e-01
## 3                chas  2.304050e-02
## 4       poly(rm, 2)1  3.709702e-01
## 5       poly(rm, 2)2  2.314111e-01
## 6      poly(dis, 2)1 -4.782839e-05
## 7      poly(tax, 2)2  2.674922e-02
## 8  poly(ptratio, 2)1 -3.005344e-01
## 9    poly(black, 2)1  1.211741e-01
## 10   poly(black, 2)2 -3.432774e-02
## 11   poly(lstat, 2)1 -1.629408e+00
```

The surviving coefficients are crim, chas, rm, rm^2, dis, dis^2, tax^2, ptratio. black, black^2 and lstat.

Question 6

Report the internal MSE and test data MSE for HR's original model; Lasso and Ridge on the original covariates; and Lasso on the full set of second order terms. Which one fits best?

```r
## HR Original Model

# Internal
print(paste("HR Original Model Internal MSE: ",mean(lm1$residuals^2)))
```

```
## [1] "HR Original Model Internal MSE:  0.0296712125146004"
```

```r
# Test MSE
print(paste("HR Original Model Test MSE: ",mean((log(testData$medv) - predict.lm(lm1, testData))^2)))
```

```
## [1] "HR Original Model Test MSE:  0.038663245270006"
```

```r
##Internal MSE: Ridge
print(paste("Ridge Model Internal MSE: ",ridge_mod$cvm[ridge_mod$lambda == ridge_mod$lambda.1se]
))
```

```
## [1] "Ridge Model Internal MSE:  0.0387039685713887"
```

```r
##Test MSE: Ridge
ridge_pred = predict(ridge_mod, s =  ridge_mod$lambda.1se,  newx = x_test)
print(paste("Ridge Model Test MSE: ",mean((ridge_pred - y_test)^2)))
```

```
## [1] "Ridge Model Test MSE:  0.0388268324861102"
```

```r
##Internal MSE: Lasso
print(paste("LASSO Model Internal MSE: ",lasso_mod$cvm[lasso_mod$lambda == lasso_mod$lambda.1se]))
```

```
## [1] "LASSO Model Internal MSE:  0.0365929578843227"
```

```r
##Test MSE: Lasso
lasso_pred = predict(lasso_mod, s =  lasso_mod$lambda.1se, newx = x_test)

print(paste("LASSO Model Test MSE: ",mean((lasso_pred - y_test)^2)))
```

```
## [1] "LASSO Model Test MSE:  0.0463886375455343"
```

```r
##Internal MSE: Lasso Second Order
print(paste("LASSO Model Second Order Internal MSE: ",lasso_mod2$cvm[lasso_mod2$lambda == lasso_mod2$lam
```

```
## [1] "LASSO Model Second Order Internal MSE:  0.00487988742921054"
```

```r
##Test MSE: Lasso Second Order
lasso_pred2 = predict(lasso_mod2, s =  lasso_mod2$lambda.1se ,newx = x_test2)

print(paste("LASSO Model Second Order Test MSE: ",mean((lasso_pred2 - y_test2)^2)))
```

```
## [1] "LASSO Model Second Order Test MSE:  0.0201081410261233"
```

The best fitting model is the LASSO Model with second order terms according to the test MSE values as well as according to the internal training MSE values.