Modern Data Mining - HW 2

Anirudh Bajaj
Esther Shin
Matt LeBaron

Overview / Instructions

This is homework #2 of STAT 471/571/701. It will be **due on Oct 10, 2018 by 11:59 PM** on Canvas. You can directly edit this file to add your answers. Submit the Rmd file, a PDF or word or HTML version with only 1 submission per HW team.

Problem 0

Review the code and concepts covered during lecture: multiple regression, model selection and penalized regression through elastic net.

Problem 1

Do ISLR, page 262, problem 8 only part (a) to (d) and write up the answer here. This question is designed to help us understanding model selections through simulations. (e) Describe as accurate as possible what Cp and BIC are estimating?

(a)

```
# Use rnorm() to generate a predictor X of length n = 100, and a noise vector of length n = 100
x <- rnorm(100)
noise <- rnorm(100)
```

(b)

```
# Generate a response vector Y of length n = 100 according to the model
y <- 1 + 2*x + 3*x^2 + 4*x^3 + noise
```

(c)

```
# Create the predictors x^2 through x^10
x2 <- x^2
x3 <- x^3
x4 <- x^4
x5 <- x^5
x6 <- x^6
x7 <- x^7
x8 <- x^8
x9 <- x^9
x10 <- x^10

# Use the regsubsets() function to perform best subset selection
new_data <- data.frame(x,x2,x3,x4,x5,x6,x7,x8,x9,x10,noise,y)
new_subset <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10)
new_summary <- summary(new_subset)
new_summary$which
```

```
##    (Intercept)     x    x2   x3    x4    x5    x6    x7    x8    x9   x10
## 1         TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## 3          TRUE  TRUE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4          TRUE  TRUE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## 5          TRUE  TRUE  TRUE TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
## 6          TRUE  TRUE  TRUE TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
## 7          TRUE  TRUE FALSE TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 8          TRUE  TRUE  TRUE TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 9          TRUE  TRUE FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 10         TRUE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
data.frame(variables=(1:length(new_summary$rsq)), r_squared=new_summary$rsq)
```

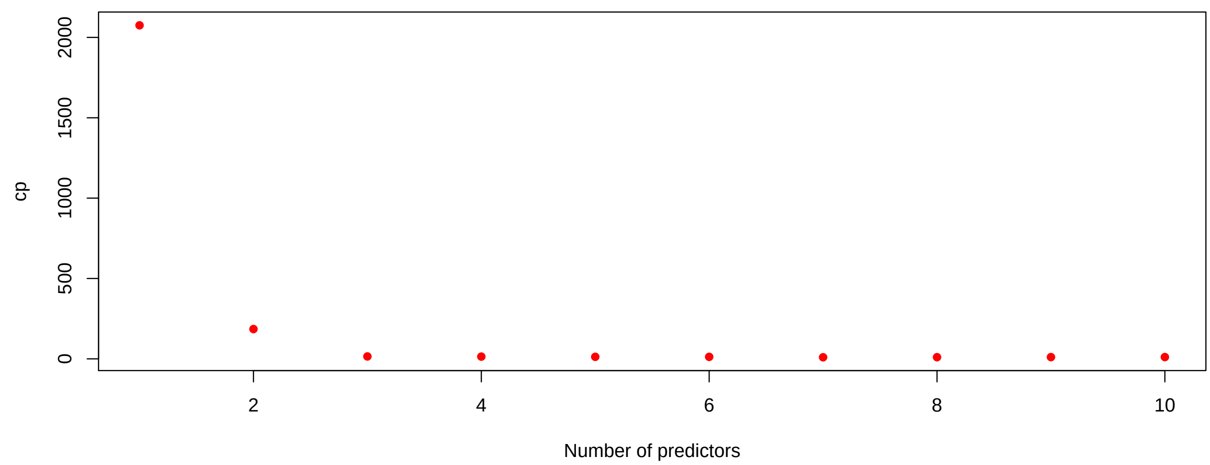
```
##    variables r_squared
## 1          1 0.9364260
## 2          2 0.9918209
## 3          3 0.9968693
## 4          4 0.9969594
## 5          5 0.9970558
## 6          6 0.9971198
## 7          7 0.9972558
## 8          8 0.9972972
## 9          9 0.9973397
## 10        10 0.9973942
```

```r
# What is the best model obtained?
data.frame(variables = (1:length(new_summary$rsq)),
           r_squared = new_summary$rsq,
           rss = new_summary$rss,
           bic = new_summary$bic,
           cp = new_summary$cp)
```

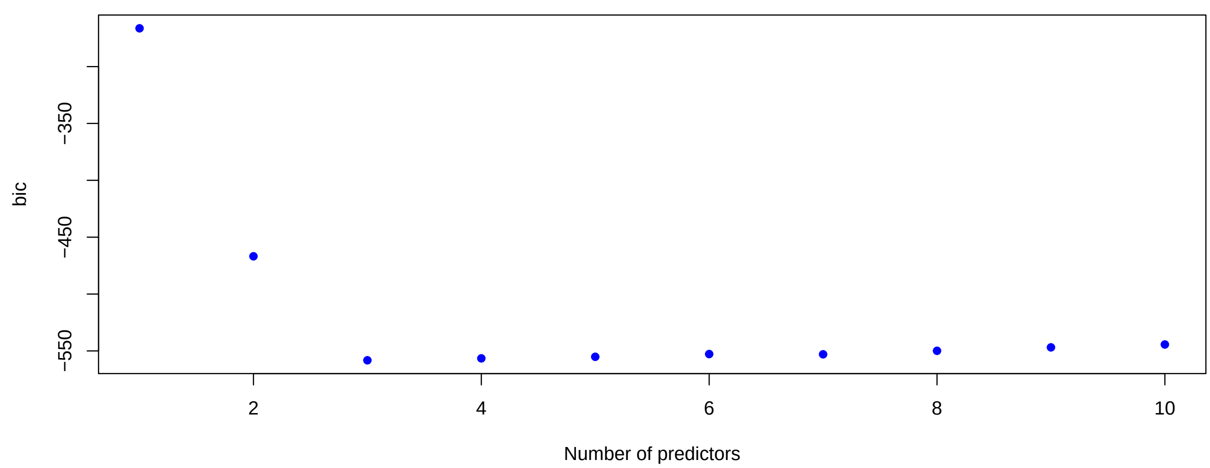
```
##    variables r_squared        rss       bic          cp
## 1          1 0.9364260 2057.43253 -266.3447 2075.362941
## 2          2 0.9918209  264.69958 -466.8014  185.357326
## 3          3 0.9968693  101.31939 -558.2280   14.929956
## 4          4 0.9969594   98.40250 -556.5440   13.851545
## 5          5 0.9970558   95.28195 -555.1614   12.558195
## 6          6 0.9971198   93.21076 -552.7540   12.372307
## 7          7 0.9972558   88.80940 -552.9859    9.727221
## 8          8 0.9972972   87.46931 -549.9012   10.312930
## 9          9 0.9973397   86.09341 -546.8815   10.860832
## 10        10 0.9973942   84.33021 -544.3456   11.000000
```

The best model obtained has three variables: x, x2, and x3. This makes sense because these three variables were used to generate Y in the first place, so they should be most predictive, although they aren't perfect because we added in noise as well.

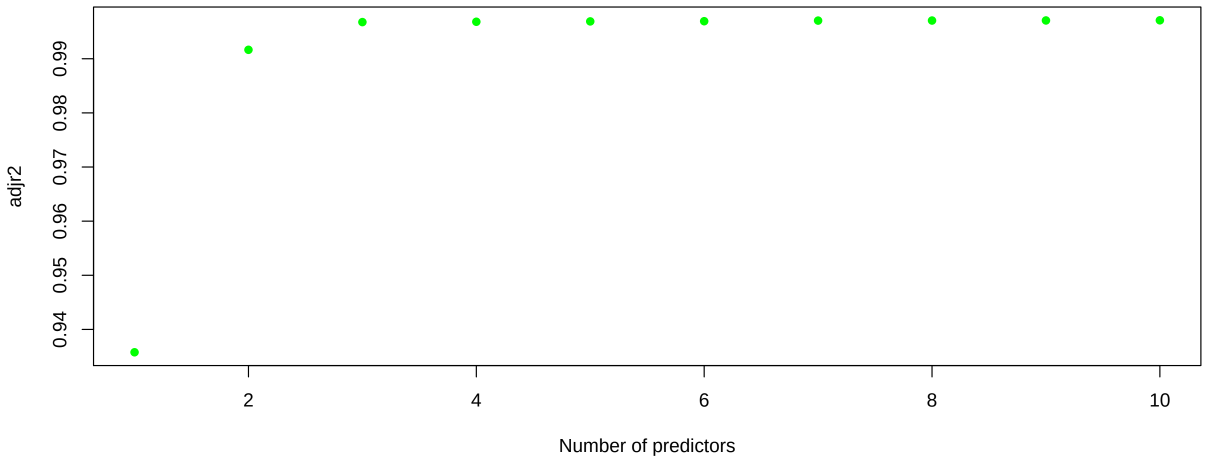
```r
# Comparing the plots for Cp, bic, and r2
plot(new_summary$cp, xlab="Number of predictors",
     ylab="cp", col="red", type="p", pch=16)
```




```r
plot(new_summary$bic, xlab="Number of predictors",
     ylab="bic", col="blue", type="p", pch=16)
```




```r
plot(new_summary$adjr2, xlab="Number of predictors",
     ylab="adjr2", col="green", type="p", pch=16)
```




```r
# Find the optimal model size for Cp
opt.size <- which.min(new_summary$cp)
opt.size
```

```
## [1] 7
```

```r
# Find the optimal model size for bic
opt.size <- which.min(new_summary$bic)
opt.size
```

```
## [1] 3
```

```r
# Find the optimal model size for r2
opt.size <- which.min(new_summary$rsq)
opt.size
```

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

```r
# Report the coefficients of the best model obtained
coef(new_subset, 3)
```

```
## (Intercept)           x          x2          x3
##    1.002750    2.137594    2.964058    3.943728
```

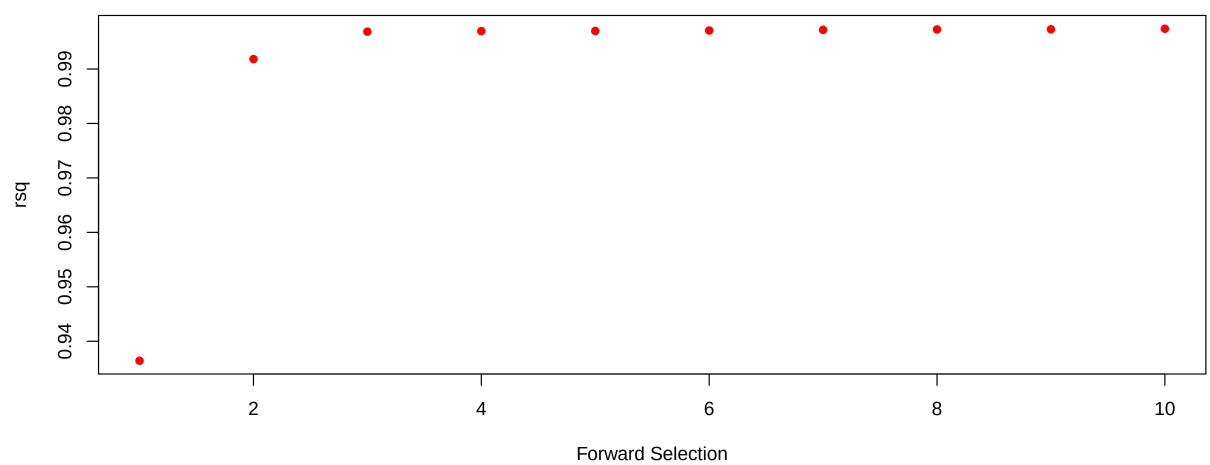
(d)

```r
# Repeat using forward selection
forward <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10, method = "forward")
forward_sum <- summary(forward)
forward_sum
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 +
##     x9 + x10, data = new_data, nvmax = 10, method = "forward")
## 10 Variables  (and intercept)
##     Forced in Forced out
## x       FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
```

```
## x5       FALSE      FALSE
## x6       FALSE      FALSE
## x7       FALSE      FALSE
## x8       FALSE      FALSE
## x9       FALSE      FALSE
## x10      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##             x   x2  x3  x4  x5  x6  x7  x8  x9  x10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" " " " " " " " " " " " " "*"
## 5  ( 1 )  "*" "*" "*" " " "*" " " " " " " " " "*"
## 6  ( 1 )  "*" "*" "*" " " "*" " " " " "*" " " "*"
## 7  ( 1 )  "*" "*" "*" " " "*" "*" " " "*" " " "*"
## 8  ( 1 )  "*" "*" "*" "*" "*" "*" " " "*" " " "*"
## 9  ( 1 )  "*" "*" "*" "*" "*" "*" "*" "*" " " "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
plot(forward_sum$rsq, ylab="rsq", col="red", type="p", pch=16,
     xlab="Forward Selection")
```




```
# Find the optimal model size for forward selection
opt.size <- which.min(forward_sum$rsq)
opt.size
```

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

```
# Report the coefficients for forward selection
coef(forward, 3)
```

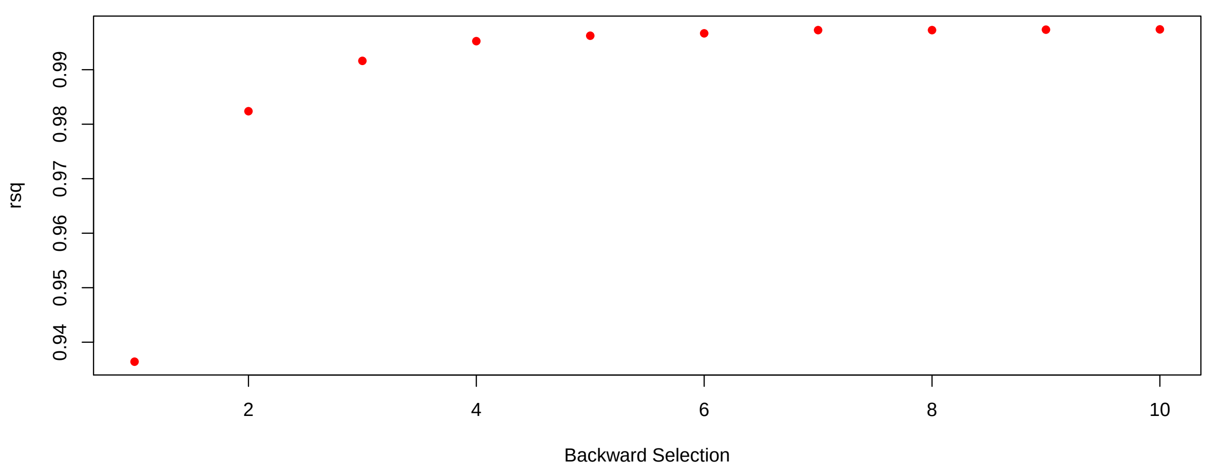
```
## (Intercept)           x          x2          x3
##    1.002750    2.137594    2.964058    3.943728
```

```
# Repeat using backward selection
backward <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10, method = "backwar
backward_sum <- summary(backward)
```

```r
backward_sum
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 +
##     x9 + x10, data = new_data, nvmax = 10, method = "backward")
## 10 Variables  (and intercept)
##     Forced in Forced out
## x       FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7      FALSE      FALSE
## x8      FALSE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##           x   x2  x3  x4  x5  x6  x7  x8  x9  x10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " " " "*" "*" " " " " " " " " " " " "
## 3  ( 1 )  " " " " "*" "*" " " "*" " " " " " " " "
## 4  ( 1 )  "*" " " "*" "*" " " "*" " " " " " " " "
## 5  ( 1 )  "*" " " "*" "*" " " "*" " " "*" " " " "
## 6  ( 1 )  "*" " " "*" "*" " " "*" " " "*" " " "*"
## 7  ( 1 )  "*" " " "*" "*" " " "*" "*" "*" " " "*"
## 8  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" " " "*"
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```r
plot(backward_sum$rsq, ylab="rsq", col="red", type="p", pch=16,
     xlab="Backward Selection")
```




```r
# Find the optimal model size for backward selection
opt.size <- which.min(backward_sum$rsq)
```

```
opt.size
```

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

```
# Report the coefficients for backward selection
coef(backward, 3)
```

```
## (Intercept)          x3          x4          x6
##   1.9359808   4.4838657   1.2966761  -0.1330395
```

Using forward and backward stepwise selection, we only use one variable (x3) as opposed to three variables (x, x2, x3).

(e)

Describe what Cp and BIC are estimating.

Cp: an unbiased estimator of average prediction errors.

BIC: uses k to estimate the number of parameters. It heavily penalizes additional variables when they are introduced into the model. Therefore, BIC models tend to have fewer variables.

Problem 2:

This will be the last part of the Auto data from ISLR. The original data contains 408 observations about cars. It has some similarity as the data CARS that we use in our lectures. To get the data, first install the package ISLR. The data Auto should be loaded automatically. We use this case to go through methods learnt so far.

You can access the necessary data with the following code:

```
# check if you have ISLR package, if not, install it
if(!requireNamespace('ISLR')) install.packages('ISLR')
auto <- ISLR::Auto
```

Final modelling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view. Use Mallow's Cp or BIC to select the model. * Describe the final model and its accuracy. Include diagnostic plots with particular focus on the model residuals. * Summarize the effects found. * Predict the mpg of a car that is: built in 1983, in US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower and weighs 4000 pounds. Give a 95% CI.

```
dim(auto)
```

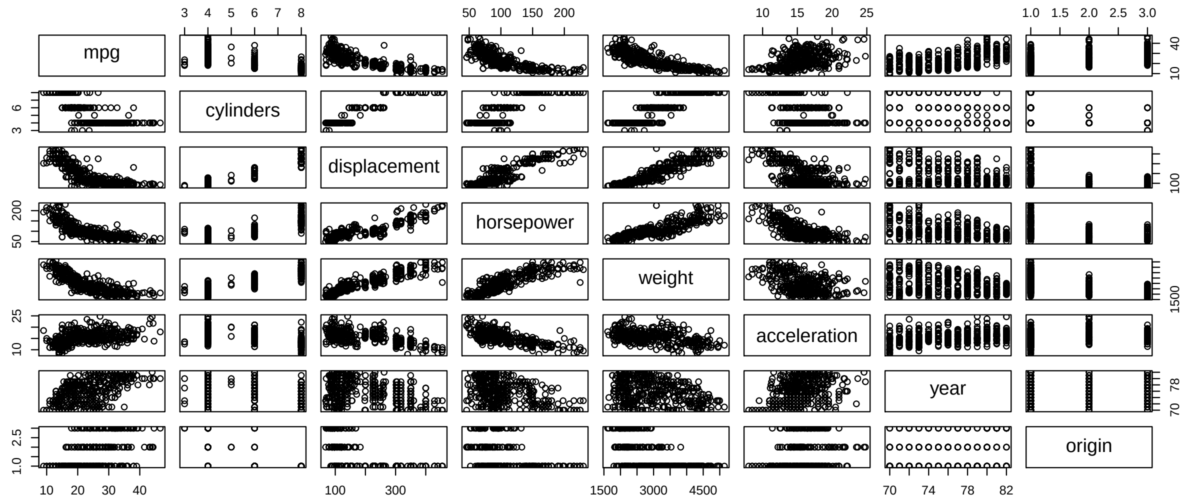
```
## [1] 392   9
```

```
str(auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241
```

```
#Exclude non-numeric values, and run pairwise scatter plot
auto %>%
```

```r
  select_if(is.numeric) %>%
  pairs()
```




```r
#From this we can see that MPG is highly correlated with Displacement, Horsepower and Weight
```

```r
#Use Regsubsets to find the model with the smallest RSS
names(auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
## [9] "name"
```

```r
data2 <- auto[,-9]
fit.exh <- regsubsets(mpg~ .,data2,nvmax = 25, method = "exhaustive")
names(fit.exh)
```

```
##  [1] "np"        "nrbar"     "d"         "rbar"      "thetab"
##  [6] "first"     "last"      "vorder"    "tol"       "rss"
## [11] "bound"     "nvmax"     "ress"      "ir"        "nbest"
## [16] "lopt"      "il"        "ier"       "xnames"    "method"
## [21] "force.in"  "force.out" "sserr"     "intercept" "lindep"
## [26] "nullrss"   "nn"        "call"
```

```r
#List the model with the smallest RSS among each size of the model
summary(fit.exh)
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data2, nvmax = 25, method = "exhaustive")
## 7 Variables  (and intercept)
##              Forced in Forced out
## cylinders        FALSE      FALSE
## displacement     FALSE      FALSE
## horsepower       FALSE      FALSE
## weight           FALSE      FALSE
## acceleration     FALSE      FALSE
## year             FALSE      FALSE
## origin           FALSE      FALSE
## 1 subsets of each size up to 7
```

```
## Selection Algorithm: exhaustive
##          cylinders displacement horsepower weight acceleration year origin
## 1  ( 1 ) " "       " "          " "        "*"    " "          " "  " "   
## 2  ( 1 ) " "       " "          " "        "*"    " "          "*"  " "   
## 3  ( 1 ) " "       " "          " "        "*"    " "          "*"  "*"   
## 4  ( 1 ) " "       "*"          " "        "*"    " "          "*"  "*"   
## 5  ( 1 ) " "       "*"          "*"        "*"    " "          "*"  "*"   
## 6  ( 1 ) "*"       "*"          "*"        "*"    " "          "*"  "*"   
## 7  ( 1 ) "*"       "*"          "*"        "*"    "*"          "*"  "*"
```

```r
f.e<-summary(fit.exh)
f.e$which
```

```
##   (Intercept) cylinders displacement horsepower weight acceleration  year
## 1        TRUE     FALSE        FALSE      FALSE   TRUE        FALSE FALSE
## 2        TRUE     FALSE        FALSE      FALSE   TRUE        FALSE  TRUE
## 3        TRUE     FALSE        FALSE      FALSE   TRUE        FALSE  TRUE
## 4        TRUE     FALSE         TRUE      FALSE   TRUE        FALSE  TRUE
## 5        TRUE     FALSE         TRUE       TRUE   TRUE        FALSE  TRUE
## 6        TRUE      TRUE         TRUE       TRUE   TRUE        FALSE  TRUE
## 7        TRUE      TRUE         TRUE       TRUE   TRUE         TRUE  TRUE
##   origin
## 1  FALSE
## 2  FALSE
## 3   TRUE
## 4   TRUE
## 5   TRUE
## 6   TRUE
## 7   TRUE
```

```r
data.frame(variables=(1:length(f.e$rsq)), r_squared=f.e$rsq)
```

```
##   variables r_squared
## 1         1 0.6926304
## 2         2 0.8081803
## 3         3 0.8174522
## 4         4 0.8180977
## 5         5 0.8200242
## 6         6 0.8211691
## 7         7 0.8214781
```

```r
#R2 increases as we increase number of variables (But much less incrementally so, after adding 2 variab
```

```r
data.frame(variables = (1:length(f.e$rsq)),
           r_squared = f.e$rsq,
           rss = f.e$rss,
           bic = f.e$bic,
           cp = f.e$cp)
```

```
##   variables r_squared      rss       bic         cp
## 1         1 0.6926304 7321.234 -450.5016 273.150806
## 2         2 0.8081803 4568.952 -629.3564  26.603456
## 3         3 0.8174522 4348.105 -642.8063   8.659680
## 4         4 0.8180977 4332.729 -638.2237   9.271088
## 5         5 0.8200242 4286.842 -636.4261   7.127265
## 6         6 0.8211691 4259.571 -632.9566   6.664509
```

```
## 7          7 0.8214781 4252.213 -627.6631   8.000000
```

```
coef(fit.exh, 6)
```

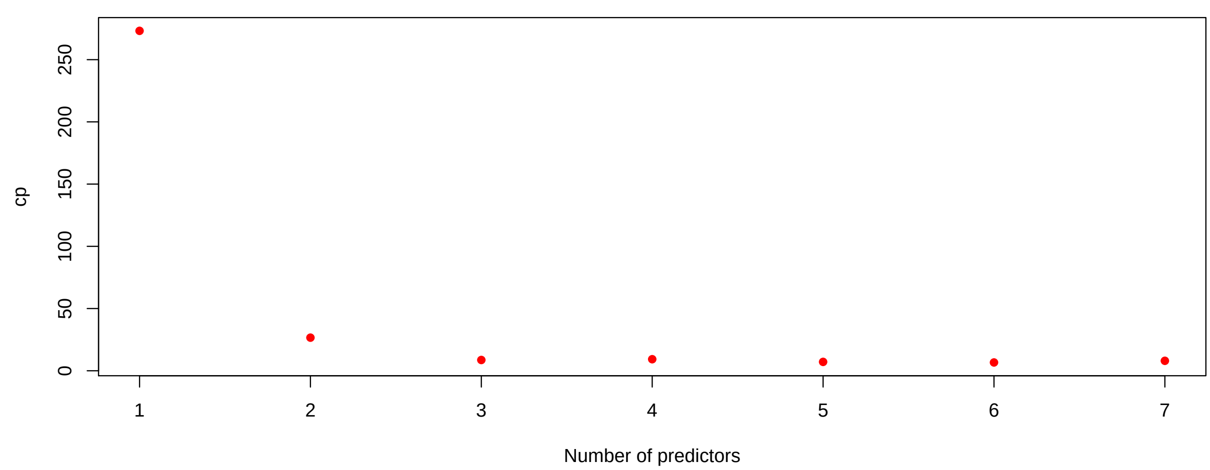
```
##   (Intercept)     cylinders  displacement    horsepower        weight
## -15.563492306  -0.506685137   0.019269286  -0.023895029  -0.006218311
##          year        origin
##   0.747515952   1.428241885
```

```
coef(fit.exh, 7)
```

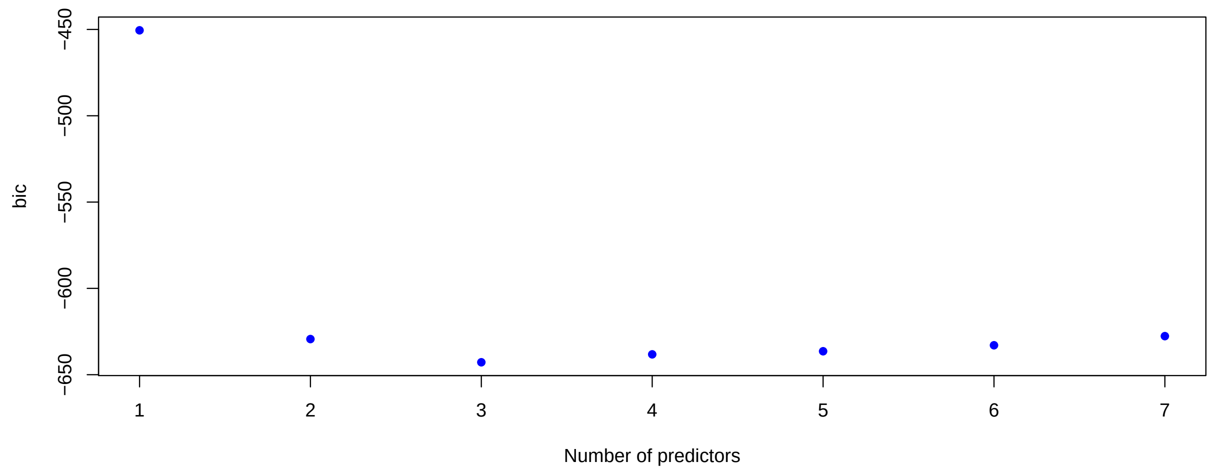
```
##   (Intercept)     cylinders  displacement    horsepower        weight
## -17.218434622  -0.493376319   0.019895644  -0.016951144  -0.006474043
##  acceleration          year        origin
##   0.080575838   0.750772678   1.426140495
```

plots of Cp vs number of predictors and plots of BIC vs number of the predictors

```
plot(f.e$cp, xlab="Number of predictors",
     ylab="cp", col="red", type="p", pch=16)
```




```
#CP smallest at 6 variables
plot(f.e$bic, xlab="Number of predictors",
     ylab="bic", col="blue", type="p", pch=16)
```




```r
#BIC smallest at 3 variables
```

```r
which.min(f.e$cp)
```

```
## [1] 6
```

```r
which.min(f.e$bic) #Choose BIC
```

```
## [1] 3
```

We may use 3 variable model (BIC tends to give the model with least number of predictors)

```r
fit.exh.var <- f.e$which[3,]
colnames(f.e$which)[fit.exh.var]
```

```
## [1] "(Intercept)" "weight"      "year"        "origin"
```

[1] "(Intercept)" "weight" "year" "origin"

So we now have the three variables

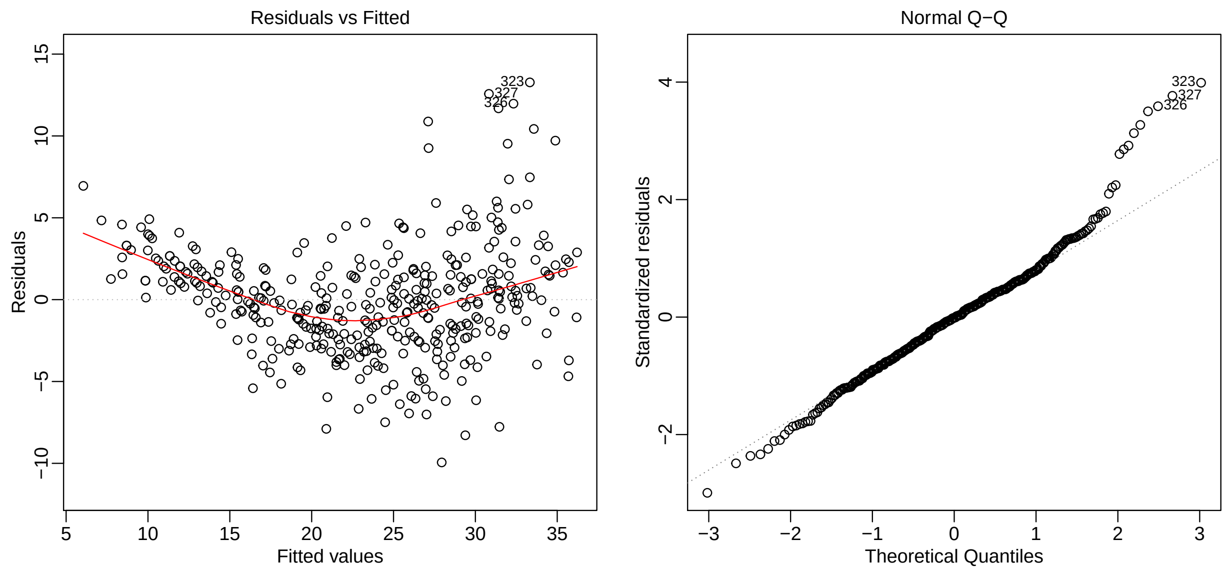
```r
fit.final <- lm(mpg ~ weight + year + origin, data2)
summary(fit.final)
```

```
##
## Call:
## lm(formula = mpg ~ weight + year + origin, data = data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9440 -2.0948 -0.0389  1.7255 13.2722
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.805e+01  4.001e+00  -4.510 8.60e-06 ***
## weight      -5.994e-03  2.541e-04 -23.588  < 2e-16 ***
## year         7.571e-01  4.832e-02  15.668  < 2e-16 ***
## origin       1.150e+00  2.591e-01   4.439 1.18e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.348 on 388 degrees of freedom
## Multiple R-squared:  0.8175, Adjusted R-squared:  0.816
## F-statistic: 579.2 on 3 and 388 DF,  p-value: < 2.2e-16
```

Final model with three variables

```
par(mfrow=c(1,2), mar=c(2.5,3,1.5,1), mgp=c(1.5,0.5,0))
plot(fit.final,1)
plot(fit.final,2)
```



Assessment of the model ("Describe the final model and its accuracy. Include diagnostic plots with particular focus on the model residuals. Summarize the effects found."):

MPG can be reasonably predicted by using three variables (weight, year, origin). The model has a good level of accuracy with an Adjusted Rsquare of 0.816. Diagnostic plot is included in the above chunk.

Making Predictions ("Predict the mpg of a car that is: built in 1983, in US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower and weighs 4000 pounds. Give a 95% CI.")

```
predicted.mpg <- data2[1,]
predicted.mpg$mpg <- NA
predicted.mpg$year <- 83
predicted.mpg$origin <- 1
predicted.mpg$cylinders <- 8
predicted.mpg$displacement <- 350
```

```r
predicted.mpg$horsepower <- 260
predicted.mpg$weight <- 4000

predicted.mpg.final.model <- predict(fit.final, predicted.mpg, interval="confidence", se.fit=TRUE)
print(predicted.mpg.final.model)
```

```
## $fit
##        fit      lwr      upr
## 1 21.96954 21.01736 22.92172
##
## $se.fit
## [1] 0.4842996
##
## $df
## [1] 388
##
## $residual.scale
## [1] 3.347605
```

fit lwr upr

1 21.96954 21.01736 22.92172

$se.fit

[1] 0.4842996

The predicted MPG for such car is 21.97 (21.02, 22.92)

Problem 3: Lasso

Crime data continuation: We use a subset of the crime data discussed in class, but only look at Florida and California. `crimedata` is available on Canvas; we show the code to clean here.

```r
cdata <- read.csv("CrimeData_clean.csv", stringsAsFactors = F, na.strings = c("?")) ## load crime data i
cdata <- dplyr::filter(cdata, state %in% c("FL", "CA")) ## filter data for Florida and California
dim(cdata)
```

```
## [1] 368  99
```

Our goal is to find the factors which relate to violent crime. This variable is included in crime as `crime$violentcrimes.perpop`.

Use LASSO to choose a reasonable, small model. Fit an OLS model with the variables obtained. The final model should only include variables with p-values < 0.05. Note: you may choose to use lambda 1st or lambda min to answer the following questions where apply.

1. What is the model reported by LASSO?

```r
Yvar <- cdata[,99] ## extracting Y
Xvar <- model.matrix(violentcrimes.perpop~., data = cdata)[,3:99] ## extracting X
colnames(Xvar)
```

```
##  [1] "population"                    "household.size"
##  [3] "race.pctblack"                 "race.pctwhite"
```

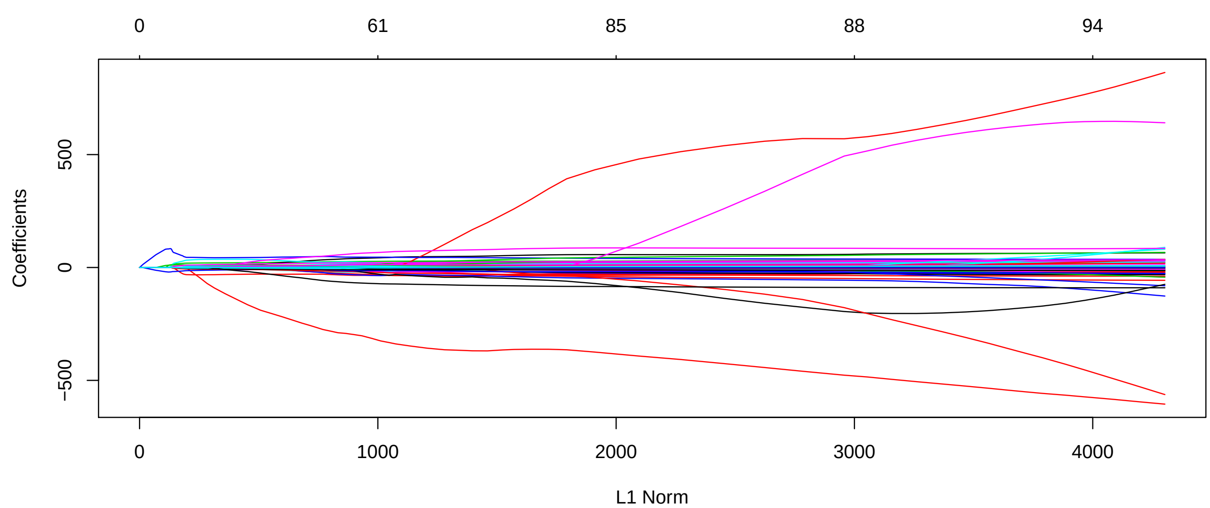
```
##  [5] "race.pctasian"                      "race.pcthisp"
##  [7] "age.pct12to21"                      "age.pct12to29"
##  [9] "age.pct16to24"                      "age.pct65up"
## [11] "pct.urban"                          "med.income"
## [13] "pct.wage.inc"                       "pct.farmself.inc"
## [15] "pct.inv.inc"                        "pct.socsec.inc"
## [17] "pct.pubasst.inc"                    "pct.retire"
## [19] "med.family.inc"                     "percap.inc"
## [21] "white.percap"                       "black.percap"
## [23] "indian.percap"                      "asian.percap"
## [25] "hisp.percap"                        "pct.pop.underpov"
## [27] "pct.less9thgrade"                   "pct.not.hsgrad"
## [29] "pct.bs.ormore"                      "pct.unemployed"
## [31] "pct.employed"                       "pct.employed.manuf"
## [33] "pct.employed.profserv"              "pct.occup.manuf"
## [35] "pct.occup.mgmtprof"                 "male.pct.divorce"
## [37] "male.pct.nvrmarried"                "female.pct.divorce"
## [39] "total.pct.divorce"                  "ave.people.per.fam"
## [41] "pct.fam2parents"                    "pct.kids2parents"
## [43] "pct.youngkids2parents"              "pct.teens2parents"
## [45] "pct.workmom.youngkids"              "pct.workmom"
## [47] "num.kids.nvrmarried"                "pct.kids.nvrmarried"
## [49] "num.immig"                          "pct.immig.recent"
## [51] "pct.immig.recent5"                  "pct.immig.recent8"
## [53] "pct.immig.recent10"                 "pct.pop.immig"
## [55] "pct.pop.immig5"                     "pct.pop.immig8"
## [57] "pct.pop.immig10"                    "pct.english.only"
## [59] "pct.no.english.well"                "pct.fam.hh.large"
## [61] "pct.occup.hh.large"                 "ave.people.per.hh"
## [63] "ave.people.per.ownoccup.hh"         "ave.people.per.rented.hh"
## [65] "pct.people.ownoccup.hh"             "pct.people.dense.hh"
## [67] "pct.hh.less3br"                     "med.num.br"
## [69] "pct.house.occup"                    "pct.house.ownoccup"
## [71] "pct.house.vacant"                   "pct.house.vacant.6moplus"
## [73] "med.yr.house.built"                 "pct.house.nophone"
## [75] "pct.house.no.plumb"                 "value.ownoccup.house.lowquart"
## [77] "value.ownoccup.med"                 "value.ownoccup.highquart"
## [79] "ownoccup.qrange"                    "rent.lowquart"
## [81] "rent.med"                           "rent.highquart"
## [83] "rent.qrange"                        "med.rent"
## [85] "med.rent.aspct.hhinc"               "med.owncost.aspct.hhinc.wmort"
## [87] "med.owncost.as.pct.hhinc.womort" "num.in.shelters"
## [89] "num.homeless"                       "pct.foreignborn"
## [91] "pct.born.samestate"                 "pct.samehouse1985"
## [93] "pct.samecity1985"                   "pct.samestate1985"
## [95] "land.area"                          "pop.density"
## [97] "pct.use.publictransit"
```

```r
lass <- glmnet(Xvar, Yvar, alpha = 1)
str(lass)
```

```
## List of 12
##  $ a0       : Named num [1:100] 896 904 1038 1160 1272 ...
##   ..- attr(*, "names")= chr [1:100] "s0" "s1" "s2" "s3" ...
##  $ beta     :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
```

```
##    .. ..@ i        : int [1:4484] 41 47 41 47 41 47 41 47 41 47 ...
##    .. ..@ p        : int [1:101] 0 0 2 4 6 8 10 12 15 18 ...
##    .. ..@ Dim      : int [1:2] 97 100
##    .. ..@ Dimnames:List of 2
##    .. .. ..$ : chr [1:97] "population" "household.size" "race.pctblack" "race.pctwhite" ...
##    .. .. ..$ : chr [1:100] "s0" "s1" "s2" "s3" ...
##    .. ..@ x        : num [1:4484] -0.872 14.625 -3.277 23.407 -5.467 ...
##    .. ..@ factors : list()
##  $ df        : int [1:100] 0 2 2 2 2 2 2 3 3 3 ...
##  $ dim       : int [1:2] 97 100
##  $ lambda    : num [1:100] 510 465 423 386 352 ...
##  $ dev.ratio: num [1:100] 0 0.104 0.198 0.276 0.34 ...
##  $ nulldev   : num 1.62e+08
##  $ npasses   : int 6877
##  $ jerr      : int 0
##  $ offset    : logi FALSE
##  $ call      : language glmnet(x = Xvar, y = Yvar, alpha = 1)
##  $ nobs      : int 368
##  - attr(*, "class")= chr [1:2] "elnet" "glmnet"
```

```
plot(lass) ## plot LASSO graph
```




```
crime <- cv.glmnet(Xvar, Yvar, alpha=1, nfolds=10) #cross validation
crime$cvm #  mean cv error
```

```
##  [1] 439269.2 405173.4 364601.0 329532.1 300417.6 276241.1 256201.5
##  [8] 239085.1 224277.6 211754.2 200975.9 191999.0 184543.4 178352.1
## [15] 173219.8 169002.9 165641.7 162977.7 160307.8 157409.9 155105.6
## [22] 153578.0 152624.3 152139.2 152162.8 152357.1 152263.9 151786.1
## [29] 151253.7 150791.9 150338.0 150005.5 149861.0 149946.4 150376.8
## [36] 151092.4 151903.1 152533.1 152658.6 152224.7 151982.4 152088.9
## [43] 152429.0 152893.2 153506.6 154382.0 155364.7 156008.2 156426.3
## [50] 156691.3 156926.7 156996.2 157061.5 157274.9 158140.6 159294.3
## [57] 159701.2 159766.0 159930.5 160110.1 160308.8 160737.2 161457.4
## [64] 162215.3 163051.5 163912.9 164781.7 165821.8 167123.3 168536.8
## [71] 169896.4 171440.4 173028.4 174768.3 176527.1 178444.0 180151.4
## [78] 182239.5 184264.5 186079.7 187725.3 189288.0 190493.4 191826.3
```

```
## [85] 193075.0 194257.5 195276.3 196382.2 197420.4
```

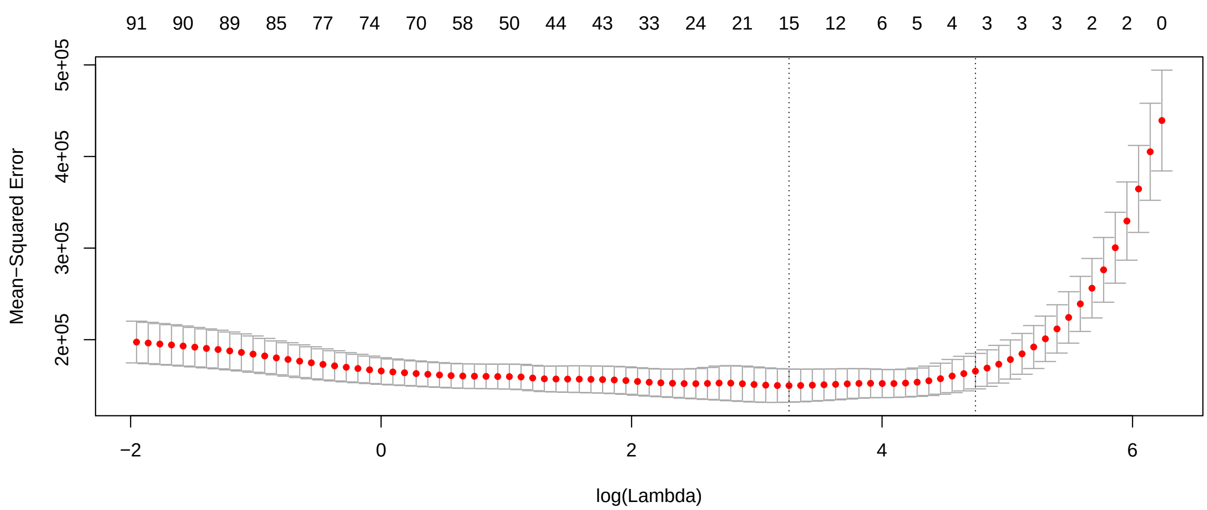
```r
crime$lambda.min #  min point among all the cvm
```

```
## [1] 25.97896
```

```r
crime$nzero # non-zero coeff's
```

```
##  s0  s1  s2  s3  s4  s5  s6  s7  s8  s9 s10 s11 s12 s13 s14 s15 s16 s17
##   0   2   2   2   2   2   2   3   3   3   3   3   3   3   3   3   3   3
## s18 s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35
##   4   4   4   5   5   6   6  10  11  11  12  14  15  15  15  16  18  19
## s36 s37 s38 s39 s40 s41 s42 s43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53
##  21  23  22  24  24  27  30  32  33  35  38  43  43  43  46  45  44  44
## s54 s55 s56 s57 s58 s59 s60 s61 s62 s63 s64 s65 s66 s67 s68 s69 s70 s71
##  45  48  50  54  56  55  58  61  64  65  70  69  70  73  74  75  75  75
## s72 s73 s74 s75 s76 s77 s78 s79 s80 s81 s82 s83 s84 s85 s86 s87 s88
##  77  77  81  83  85  88  89  89  89  90  88  90  90  91  91  90  91
```

```r
plot(crime)
```



2. What is the model after running OLS?

```r
cof <- coef(crime, s="lambda.1se") ## lambda 1se
cof <- cof[which(cof !=0),] # to get  non-zero coefficients
var <- rownames(as.matrix(cof))
input <- as.formula(paste("violentcrimes.perpop", "~", paste(var[-1], collapse = "+"))) # prepare input
```

```r
lmse <- lm(input, data=cdata)
output <- coef(lmse) # output lm estimates
summary(lmse)
```

```
##
## Call:
## lm(formula = input, data = cdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1115.29  -210.52   -37.48   155.25  1911.97
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2012.949    266.282   7.559 3.32e-13 ***
## race.pctblack         13.956      2.742   5.089 5.78e-07 ***
## pct.kids2parents     -22.678      3.371  -6.728 6.70e-11 ***
## pct.kids.nvrmarried   94.953     12.269   7.739 9.95e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 378.3 on 364 degrees of freedom
## Multiple R-squared:  0.6791, Adjusted R-squared:  0.6765
## F-statistic: 256.8 on 3 and 364 DF,  p-value: < 2.2e-16
```

```r
comp <- data.frame(cof, output )
names(comp) <- c("estimates from LASSO", "lm estimates")
comp
```

```
##                     estimates from LASSO lm estimates
## (Intercept)                  1810.355977   2012.94869
## race.pctblack                   7.952891     13.95606
## pct.kids2parents              -18.222789    -22.67806
## pct.kids.nvrmarried            78.476883     94.95302
```

3. What is your final model, after excluding high p-value variables?

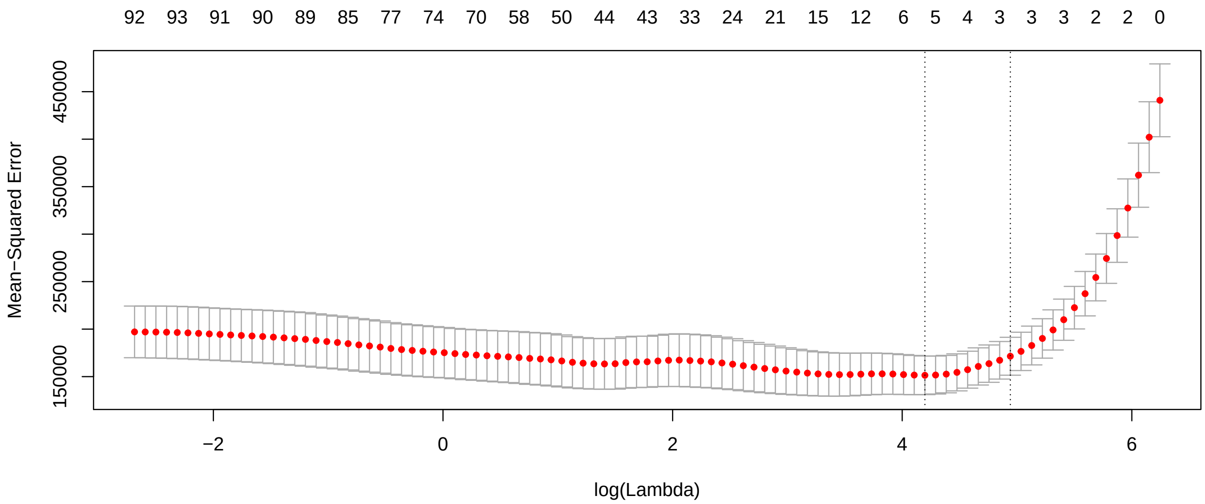
Now, instead of Lasso, we want to consider how changing the value of alpha (i.e. mixing between Lasso and Ridge) will affect the model. Cross-validate between alpha and lambda, instead of just lambda. Note that the final model may have variables with p-values higher than 0.05; this is because we are optimizing for accuracy rather than parsimoniousness.

1. What is your final elastic net model? What were the alpha and lambda values? What is the prediction error?

```r
fit2 <- glmnet(Xvar, Yvar, alpha=.99)
fit2.cv <- cv.glmnet(Xvar, Yvar, alpha=.99, nfolds=10)
fit2.cv$lambda.1se
```

```
## [1] 140.0423
```

```r
plot(fit2.cv)
```



2. Use the elastic net variables in an OLS model. What is the equation, and what is the prediction error.

```
cof2 <- coef(fit2.cv, s="lambda.1se")
cof2 <- cof2[which(cof2 !=0),] ##  non-zero coefficients
var2 <- rownames(as.matrix(cof2)) ## output names
input2 <- as.formula(paste("violentcrimes.perpop", "~", paste(var2[-1], collapse = "+"))) # prepare inp
lm2 <- lm(input2, data=cdata)
output2 <- coef(lm2) # output lm estimates
summary(lm2)
```

```
##
## Call:
## lm(formula = input2, data = cdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1115.29  -210.52   -37.48   155.25  1911.97
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2012.949    266.282   7.559 3.32e-13 ***
## race.pctblack         13.956      2.742   5.089 5.78e-07 ***
## pct.kids2parents     -22.678      3.371  -6.728 6.70e-11 ***
## pct.kids.nvrmarried   94.953     12.269   7.739 9.95e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 378.3 on 364 degrees of freedom
## Multiple R-squared:  0.6791, Adjusted R-squared:  0.6765
## F-statistic: 256.8 on 3 and 364 DF,  p-value: < 2.2e-16
```

3. Summarize your findings, with particular focus on the difference between the two equations.

```
comp2 <- data.frame(cof2, output2 )
names(comp2) <- c("estimates from Elastic Net", "lm estimates")
comp2
```

```
##                     estimates from Elastic Net lm estimates
```

```
## (Intercept)                           1767.966054   2012.94869
## race.pctblack                            6.760708     13.95606
## pct.kids2parents                       -17.291978    -22.67806
## pct.kids.nvrmarried                     74.920146     94.95302
```