Modern Data Mining - HW 2

Kexin Zhu
Yang Yi
Yifan Jiang

Overview / Instructions

This is homework #2 of STAT 471/571/701. It will be **due on Feb, 24, 2019 by 11:59 PM** on Canvas. You can directly edit this file to add your answers. Submit the Rmd file *and* a knitted (PDF, Word, or HTML) version, with only 1 submission allowed 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, parts (a) through (e), and write up the answer here. This question is designed to help us understanding model selection through simulations.

(a) Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

Answer: For predictor X, we generate 100 normal random numbers with arguments mean = 2 and sd = 2. For noise vector ϵ, we generate 100 normal random numbers with arguments mean = 0 and sd = 1.

```
set.seed(5)
x = rnorm(100, mean = 10, sd = 2)
noise_vector_epsilon = rnorm(100, mean = 0, sd = 1)
```

(b) Generate a response vector Y of length $n = 100$ according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

where β_0, β_1, β_2, and β_3 are constants of your choice.

Answer: We build the response vector by taking a sample of 1 to 100 from the elements of x with replacement.

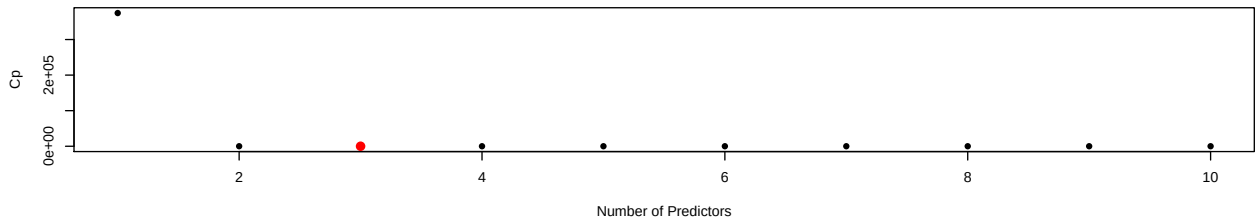
```
beta = sample(1:100, 4, replace = TRUE, prob = NULL)
y = beta[1] + beta[2] * x + beta[3] * x^2 + beta[4] * x^3 + noise_vector_epsilon
```

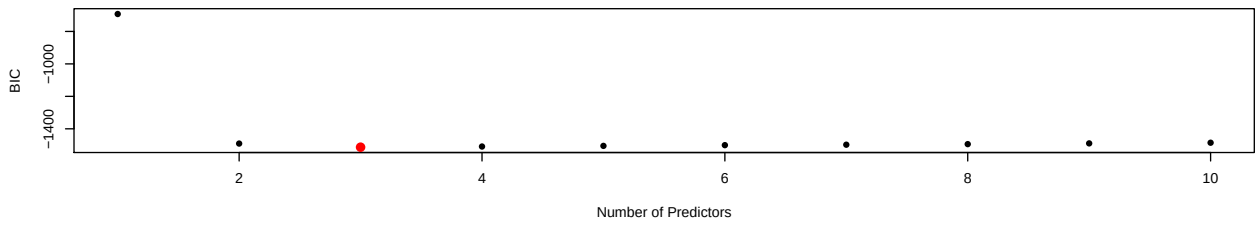
(c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, ..., X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the `data.frame()` function to create a single data set containing both X and Y.

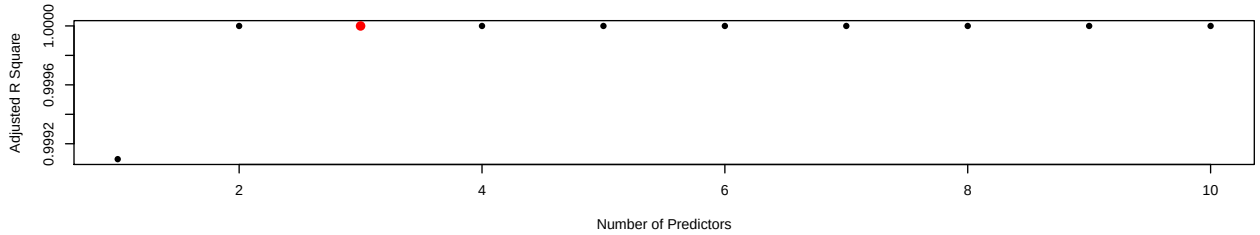
Answer: Here, R may mistakenly think that x^2 is x powered by 2, but according to the question, we need x^2 as a whole. Thus, we use `I()` function, which means *as.is*. We put `I(x^2)` to make x^2 as a variable. We compare different criteria by ploting them, and find minimum Cp, minimum BIC, and maximum $AdjustedRSquare$ points using `points()`.
Remark for `points()` function below: first segment is where the x is, second is where the y is. Thus, we

have to find the corresponding "x" for min/max number. `c(1:10)` means the number of predictors, while the argument in [ ] means a logic judgment which would pull out the correct integer for the point.

```
data_simulation <- data.frame(x = x, y = y)
fit.exhaustive <- regsubsets(y ~ I(x) + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) +
                         I(x^8) + I(x^9) + I(x^10), data = data_simulation, nvmax = 10)

summary(fit.exhaustive)
exhaustive.summary <- summary(fit.exhaustive)

par(mfrow=c(3,1)) # put plots together and compare different criterions
plot(exhaustive.summary$cp, xlab = "Number of Predictors", ylab="Cp", type = "p", pch = 16)
points(c(1:10)[exhaustive.summary$cp == min(exhaustive.summary$cp)], min(exhaustive.summary$cp),
       col = "red", pch = 16, cex = 1.5) # get and mark the min point

plot(exhaustive.summary$bic, xlab = "Number of Predictors", ylab = "BIC", type = "p", pch = 16)
points(c(1:10)[exhaustive.summary$bic == min(exhaustive.summary$bic)], min(exhaustive.summary$bic),
       col = "red", pch = 16, cex = 1.5)

plot(exhaustive.summary$adjr2,xlab = "Number of Predictors", ylab = "Adjusted R Square",
     type = "p", pch = 16)
points(c(1:10)[exhaustive.summary$adjr2 == max(exhaustive.summary$adjr2)], max(exhaustive.summary$adjr2
       col = "red", pch = 16, cex = 1.5)
```







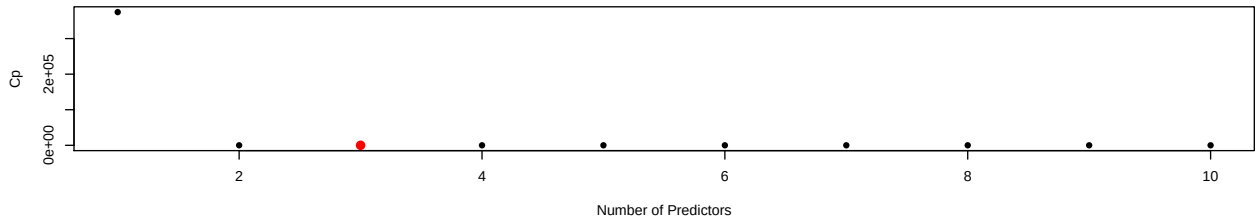
As concluded from the plots, the best model selected by C_p contains 3 predictors: X, X^2, X^3. The best model selected by BIC has 3 predictors: X, X^2, and X^3. The best model selected by adjusted R^2 also contains 3 predictors X, X^2, X^3.

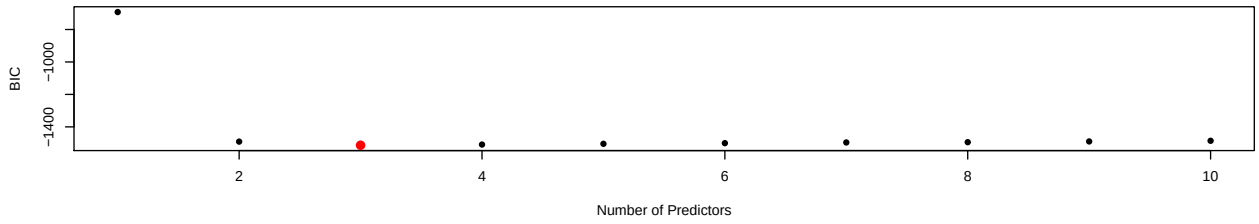
(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does

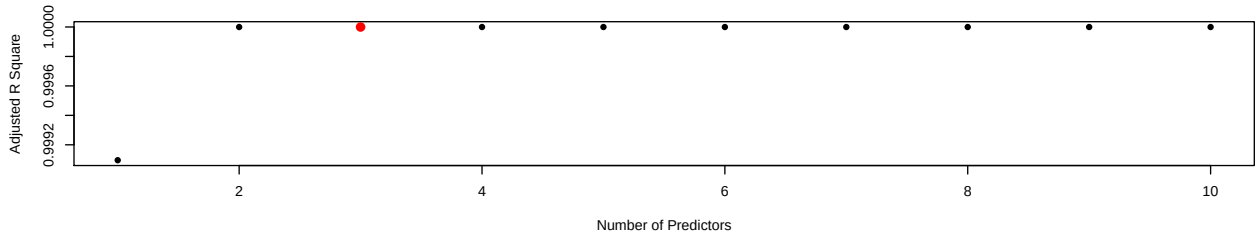
your answer compare to the results in (c)?

Answer:
Forward Selection: We change the default method argument in the resubsets from *exhaustive* to *forward*.

```
fit.forward <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) +
                          I(x^9) + I(x^10), data = data_simulation, nvmax = 10, method = "forward")
summary(fit.forward)
forward.summary <- summary(fit.forward)

par(mfrow=c(3,1)) # show the plots
plot(forward.summary$cp, xlab = "Number of Predictors", ylab ="Cp", type = "p", pch = 16)
points(c(1:10)[forward.summary$cp == min(forward.summary$cp)], min(forward.summary$cp),
       col = "red", pch = 16, cex = 1.5)

plot(forward.summary$bic, xlab = "Number of Predictors", ylab = "BIC", type = "p", pch = 16)
points(c(1:10)[forward.summary$bic == min(forward.summary$bic)], min(forward.summary$bic),
       col = "red", pch = 16, cex = 1.5)

plot(forward.summary$adjr2,xlab = "Number of Predictors", ylab = "Adjusted R Square",
     type = "p", pch = 16)
points(c(1:10)[forward.summary$adjr2 == max(forward.summary$adjr2)], max(forward.summary$adjr2),
       col = "red", pch = 16, cex = 1.5)
```





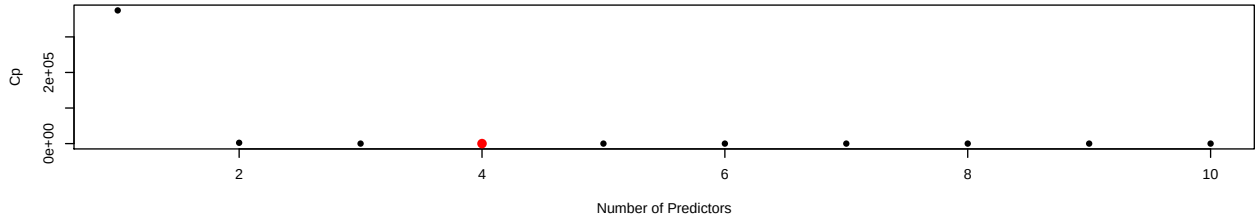


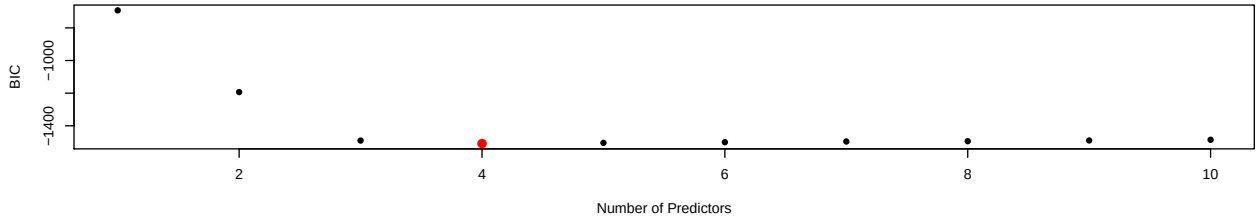
As concluded from the plots, the best model selected by C_p contains 3 predictors: X, X^2, X^3. The best model selected by BIC has 3 predictors: X, X^2, and X^3. The best model selected by $adjustedR^2$ also contains 3 predictors: X, X^2, X^3. The results are the same as (c).

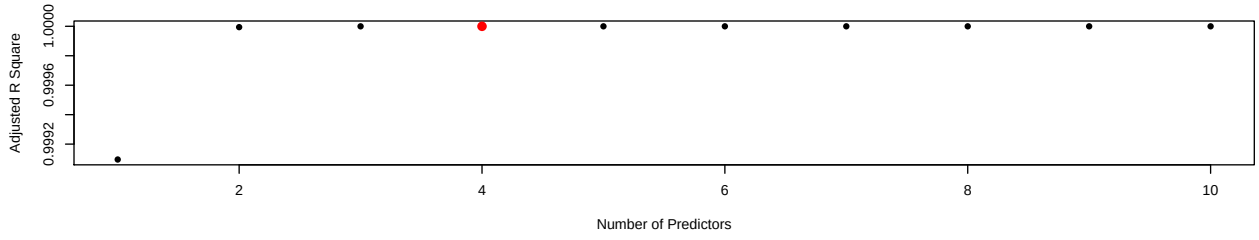
Backward Selection: We change the default method argument in the resubsets from *exhaustive* to

backward.

```
fit.backward <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) +
                           I(x^9) + I(x^10), data = data_simulation, nvmax = 10, method = "backward")
summary(fit.backward)
backward.summary <- summary(fit.backward)

par(mfrow=c(3,1)) # show the plots
plot(backward.summary$cp, xlab = "Number of Predictors", ylab ="Cp", type = "p", pch = 16)
points(c(1:10)[backward.summary$cp == min(backward.summary$cp)], min(backward.summary$cp),
       col = "red", pch = 16, cex = 1.5)

plot(backward.summary$bic, xlab = "Number of Predictors", ylab = "BIC", type = "p", pch = 16)
points(c(1:10)[backward.summary$bic == min(backward.summary$bic)], min(backward.summary$bic),
       col = "red", pch = 16, cex = 1.5)

plot(backward.summary$adjr2,xlab = "Number of Predictors", ylab = "Adjusted R Square",
     type = "p", pch = 16)
points(c(1:10)[backward.summary$adjr2 == max(backward.summary$adjr2)], max(backward.summary$adjr2),
       col = "red", pch = 16, cex = 1.5)
```





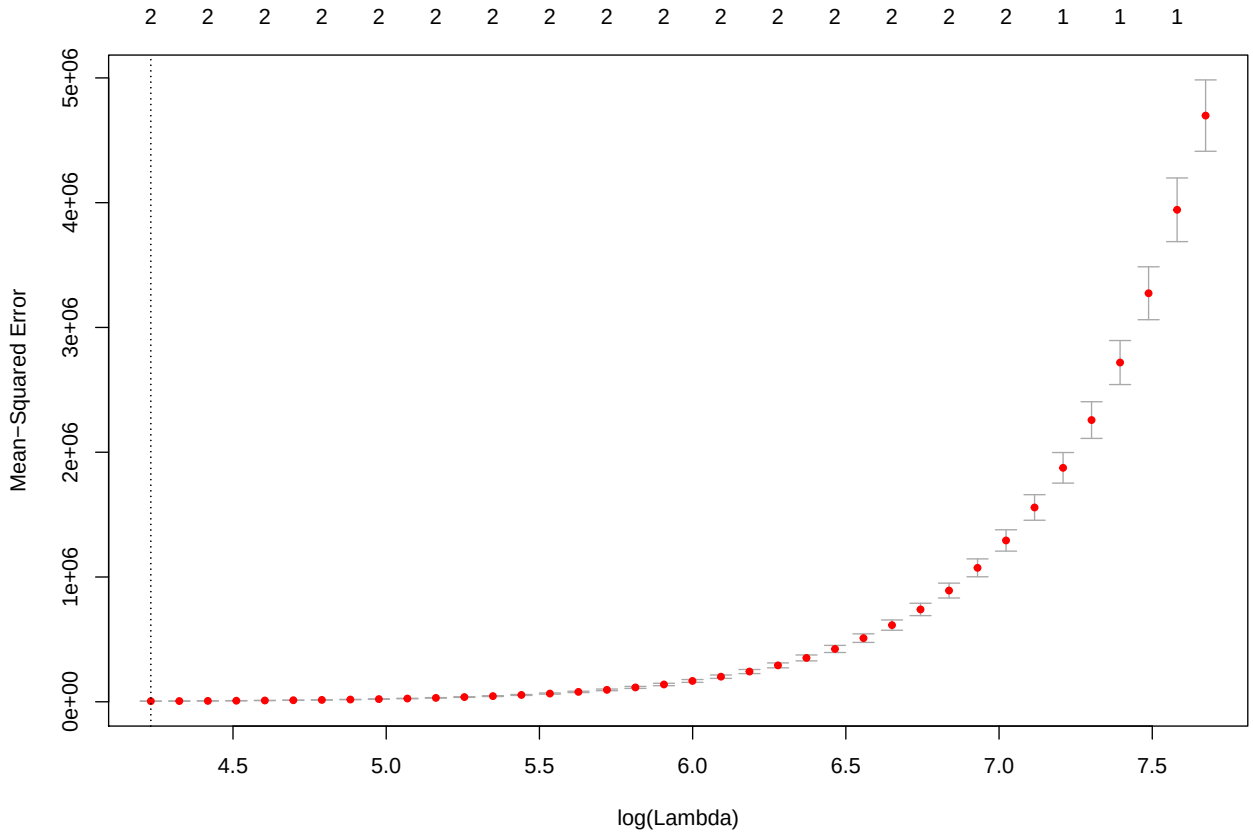


As concluded from the plots, the best model selected by C_p contains 4 predictors: X, X^4, X^5, and X^8. The best model selected by BIC has 4 predictors: X, X^4, X^5, and X^8. The best model selected by $adjustedR^2$ also contains 4 predictors: X, X^4, X^5, and X^8.

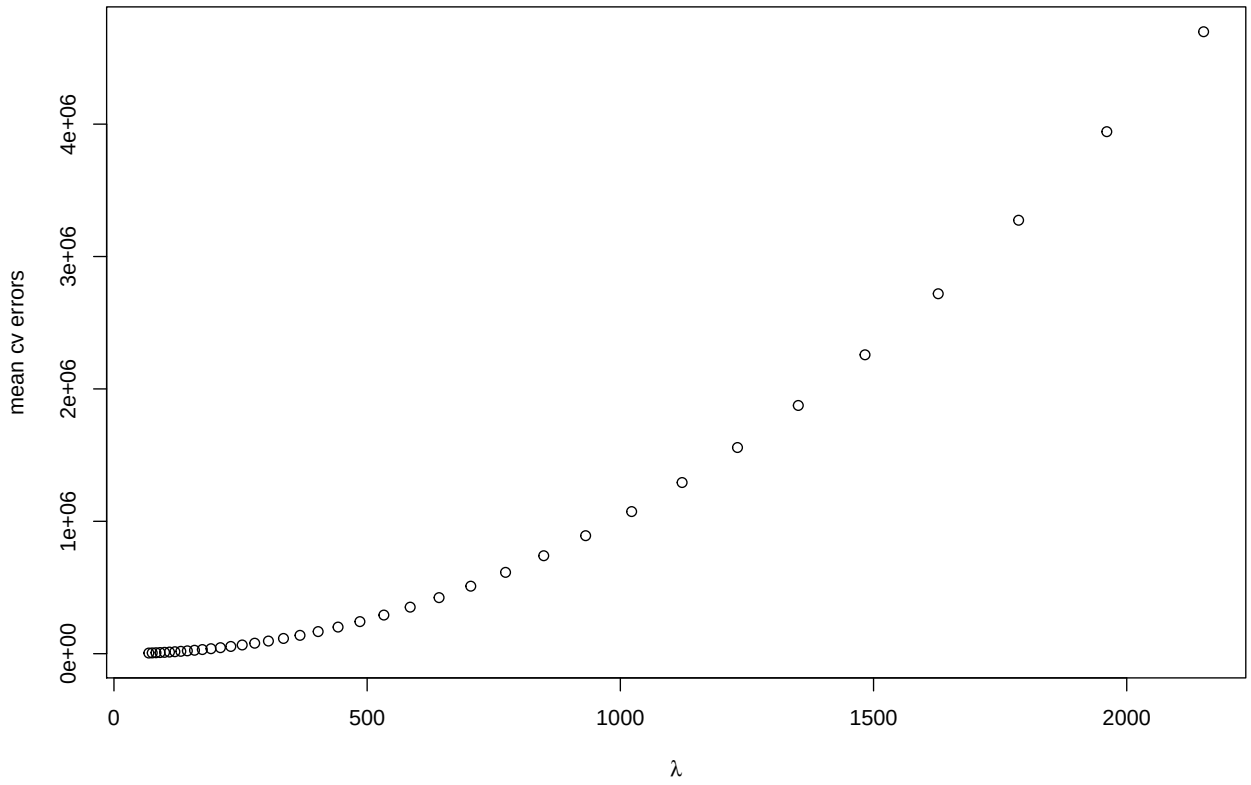
(e) Now fit a lasso model to the simulated data, again using $X, X_2, ..., X_{10}$ as predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

Answer: We use the function `cv.glmnet()` to accomplish the cross-validation, and then give `glmnet()` the λ we wish to use to generate output.

```
y <- y
x <- model.matrix(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) +
                    I(x^9) + I(x^10), data = data_simulation)

fit.cv = cv.glmnet(x, y, alpha = 1)
plot(fit.cv)
```




```
plot(fit.cv$lambda, fit.cv$cvm, xlab=expression(lambda), ylab="mean cv errors")
```




```
fit.cv$lambda.min
fit.cv$lambda.1se
fit.cv$nzero

fit.simulation <- glmnet(x, y, alpha = 1)
coef.min <- coef(fit.simulation, s=fit.cv$lambda.min)
coef.min <- coef.min[which(coef.min !=0),]
coef.min
```

The two plots above show the cross-validation error as a function of λ and MSE as a function of $log(\lambda)$ with the upper and lower boundary. The minimum CV value and the λ within one standard devation of the minimum CV error are 68.84151. We choose λ indicating the minimum CV error, and find that X^2 and X^3 are non-zero. Looking at this plot, we see that the smallest mean CV error occurs when λ is around 100. Specifically, our minimum error occurs when λ is 68.8415129.

(f) Describe as accurate as possible what C_p and BIC are estimating?

Answer: C_p is a criteria to assess fits when models with different numbers of parameters are being compared based on estimating the mean squared error (MSE). It is used to assess the fit of a regression model that has been estimated using ordinary least squares. It is applied in the context of model selection, where a number of predictor variables are available for predicting some outcome, and the goal is to find the best model involving a subset of these predictors. A small value of Cp means that the model is relatively precise. C_p is defined as:

$$C_p = \frac{1}{n}(RSS + 2d \cdot \hat{\sigma}^2)$$

BIC is the abbreviation of Bayesian information criterion, which is a criterion for model selection among a finite set of models. It is based, in part, on the likelihood function. When fitting models, it is possible to increase the likelihood by adding parameters, but doing so may result in overfitting. The BIC resolves this problem by introducing a penalty term for the number of parameters in the model. BIC is defined as:

$$BIC = \frac{1}{n}(RSS + k \cdot ln(n) \cdot \hat{\sigma}^2)$$

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 Cars data that we use in our lectures. To get the data, first install the package `ISLR`. The data set `Auto` should be loaded automatically. We use this case to go through methods learned so far.

You can access the necessary data with the following code:

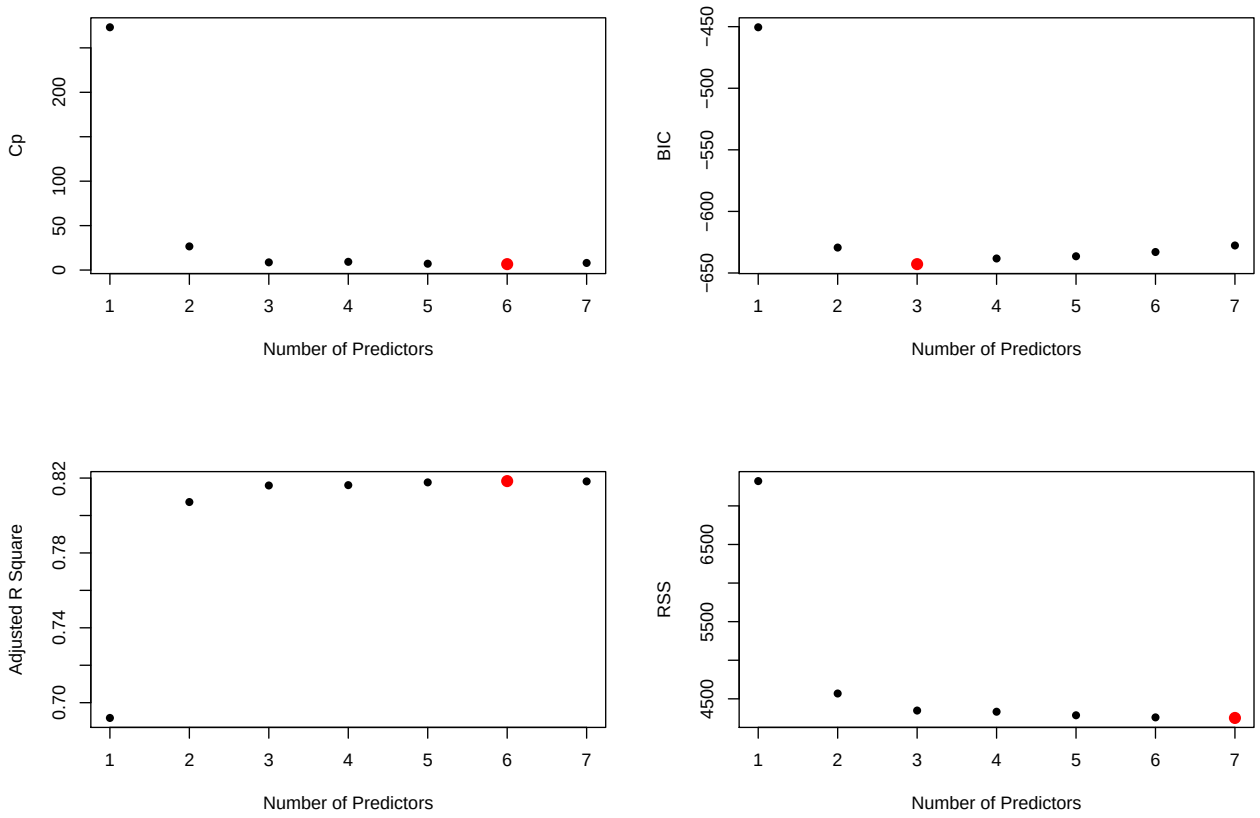
Final modelling question: We want to explore the effects of each feature as best as possible.

You may explore the possibility of variable transformations. We normally do not suggest to transform x for the purpose of interpreation. You may consider to transform y to either correct the violation of the linear model assumptions or if you feel a transformation of y makes more sense from an interpretaation pespective. You may also explore adding interactions and higher order terms. 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 C_p or BIC to select the model.

- Describe the final model and its accuracy. Include diagnostic plots with particular focus on the model residuals.

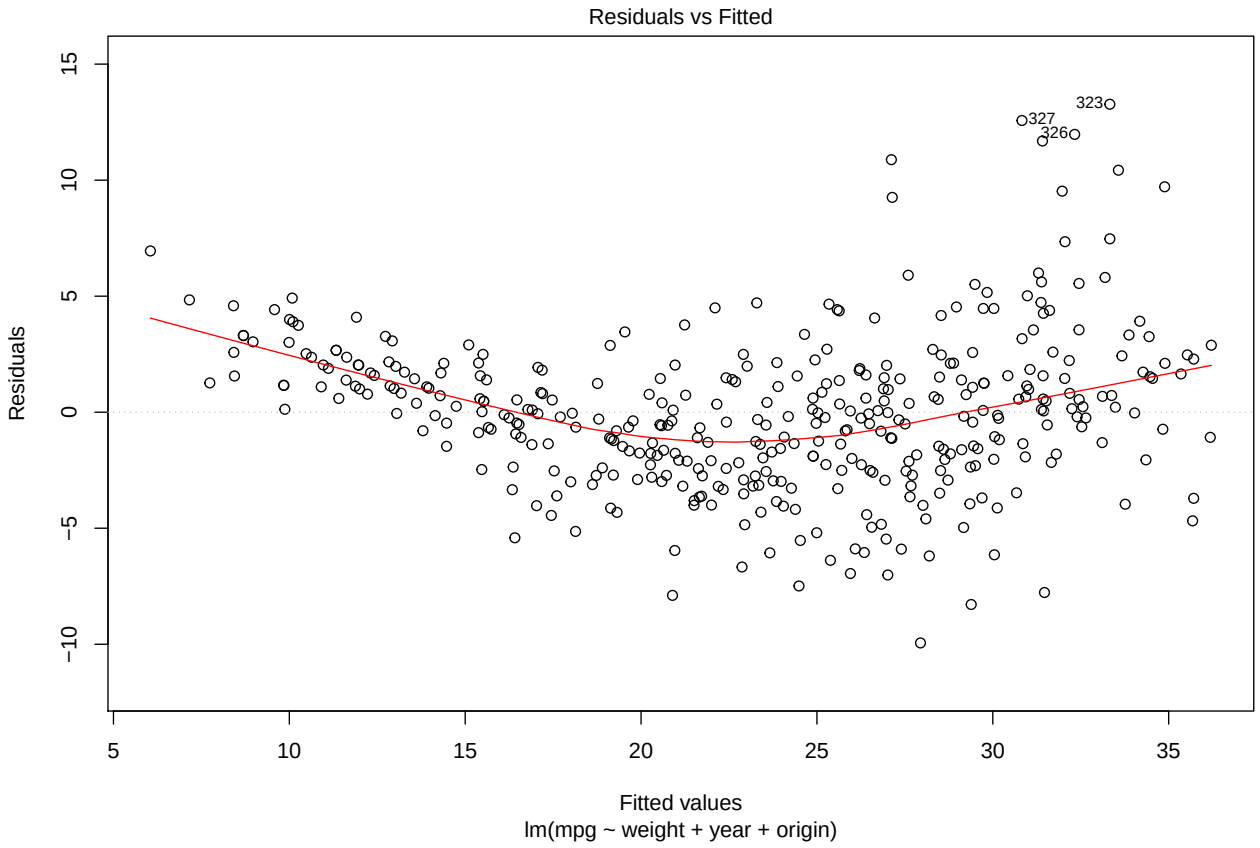
Answer: In this dataset, the number of predictors is 7, so we use all subset selection, which is the default setting in `regsubsets()`.

```
best.car <- regsubsets(mpg ~ cylinders + displacement + horsepower + weight + acceleration +
                         year + origin, data = auto_data)
summary(best.car)
fit.car <- summary(best.car)

par(mfrow=c(2,2))
plot(fit.car$cp, xlab = "Number of Predictors", ylab ="Cp", type = "p", pch = 16)
points(c(1:7)[fit.car$cp == min(fit.car$cp)], min(fit.car$cp),
       col = "red", pch = 16, cex = 1.5)

plot(fit.car$bic, xlab = "Number of Predictors", ylab = "BIC", type = "p", pch = 16)
points(c(1:7)[fit.car$bic == min(fit.car$bic)], min(fit.car$bic),
       col = "red", pch = 16, cex = 1.5)

plot(fit.car$adjr2,xlab = "Number of Predictors", ylab = "Adjusted R Square", type = "p", pch = 16)
points(c(1:7)[fit.car$adjr2 == max(fit.car$adjr2)], max(fit.car$adjr2),
       col = "red", pch = 16, cex = 1.5)

plot(fit.car$rss, xlab = "Number of Predictors", ylab ="RSS", type = "p", pch = 16)
points(c(1:7)[fit.car$rss == min(fit.car$rss)], min(fit.car$rss),
       col = "red", pch = 16, cex = 1.5)
```

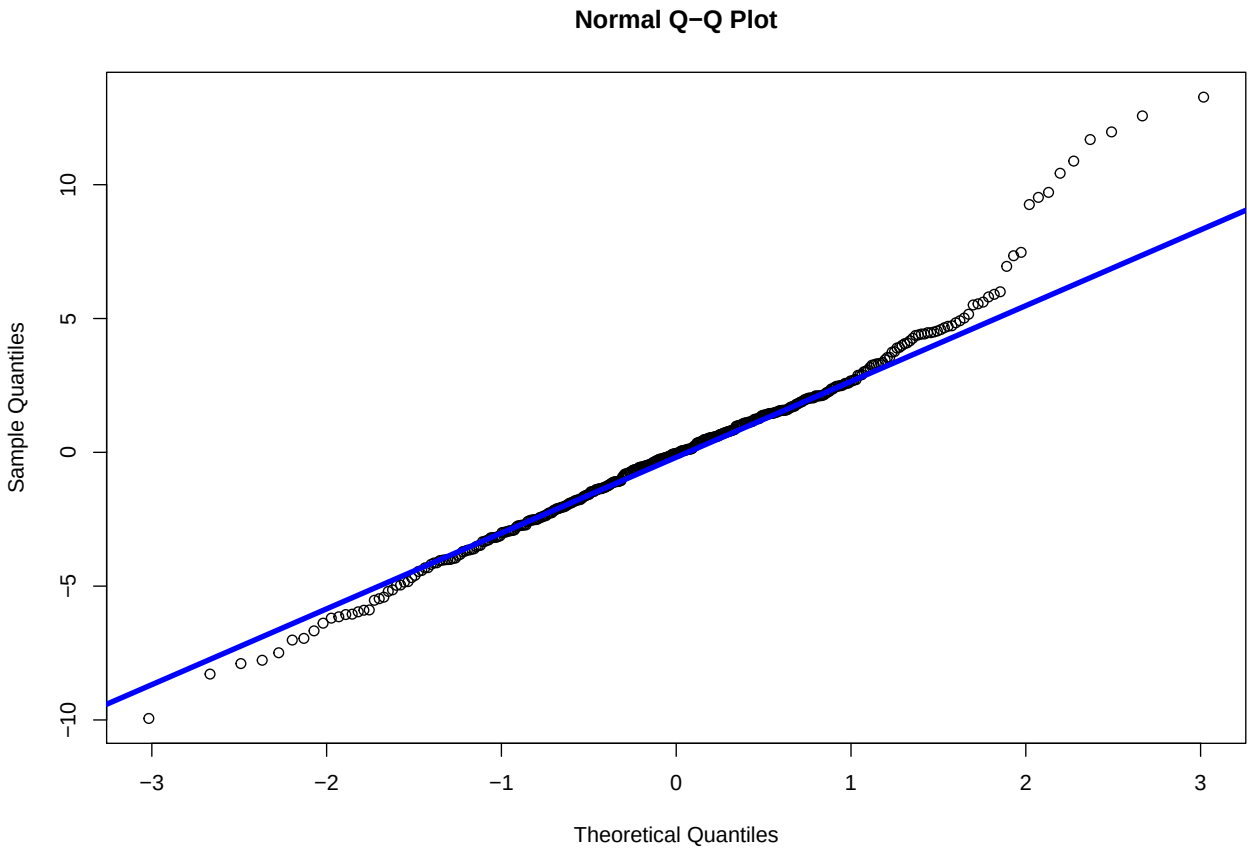


According to the model, we can see that both Cp and $AdjustedR^2$ predict that when we have 6 predictors in the model, the RSS would be the smallest. However, based on the BIC criteria, the selection process remains 3 predictors. We can also conclude that the RSS of models with 3 to 6 predictors are similar. It might be better to accept a model with three variables (here: weight, year, and origin) since all parameters manifest a sharp drop there.

```
best.fit.car <- lm(mpg ~ weight + year + origin,data = auto_data)
summary(best.fit.car)
#model diagnostics
plot(best.fit.car,1)
```




```
qqnorm(best.fit.car$residuals)
qqline(best.fit.car$residuals, lwd=4, col="blue")
```



Hence, supported by a large F-stat and a small p-value, we finalize our model as following:

$$mpg = -18.05 - 0.0059 * weight + 0.75 * year + 1.15 * origin$$

- Summarize the effects found.

Answer: According to the model shown above, we conclude that `mpg` is most correlated with car weight, the year manufactured, and the continent where the car is produced. With each 1 unit increase in weight, MPG would be supposed to decrease by 0.0059. If the car is manufactered one year closer, `mpg` would be supposed to increase by 0.75. Besides, given that the "origin" variable (1 = American car, 2 = European car, 3 = Asian car), Asian cars are supposed to have 1.15 `mpg` more that European ones, of which `mpg` is supposed to be 1.15 higher than American cars.

- Predict the `mpg` of a car that is: built in 1983, in the US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower, and weighs 4,000 pounds. Give a 95% CI.

Answer: We can calculate the predicted `mpg` with our model:

```
-18.05 - 0.0059 * 4000 + 0.75 * 83 + 1.15 * 1
```

The final answer is:

$$mpg = -18.05 - 0.0059 * 4000 + 0.75 * 83 + 1.15 * 1 = 21.75$$

The 95% CI is calculated as: $(15.054, 28.446)$, which can be calculated using the following R code.

```
lower <- 21.75 - 2*3.348
lower
upper <- 21.75 + 2*3.348
upper
```

- Any suggestions as to how to improve the quality of the study?

Answer: We propose several suggestions:

1. We could seek for more obervables (data points) to better predict the correlation between variables. Generally, we need at least 30 observables to correctly generate a one-variable model (Based on the rule of sum). Here, roughly speaking, we have 40 observables for each variable, which is pretty close to baseline of our requirement and has limited power in model prediction.
2. One caveat should be raised is that the explanatory variables our measure here may have strong correlations in between. For instance, the pairwise plot shows that displacement and weight hold a relatively strong correlation. This may lead to a problem, say, the predictors are not independent, which would lower down the power of prediction in models and add more noises to the selection process.

```
auto_data %>%
  select_if(is.numeric) %>%
  ggpairs()
summary(lm(horsepower ~ weight, auto_data))
```

3. Prepare some test data to test the model, such as using cross validation.
4. Reference more criterias to select the model, and compare models with test data.

Problem 3: LASSO

Part I: EDA

Crime data continuation: We continue to use the crime data analyzed in the lectures. We first would like to visulize how crime rate (`violentcrimes.perpop`) distributes by states. The follwoing `r`-chunk will read in the entire crime data into the `r`-path and it also creates a subset.

```
crime.all <- read.csv("CrimeData_clean.csv", stringsAsFactors = F, na.strings = c("?"))
crime <- dplyr::filter(crime.all, state %in% c("FL", "CA"))
```

Show a heatmap displaying the mean violent crime by state. You may also show a couple of your favorite summary statistics by state through the heatmaps. Write a brief summary based on your findings.

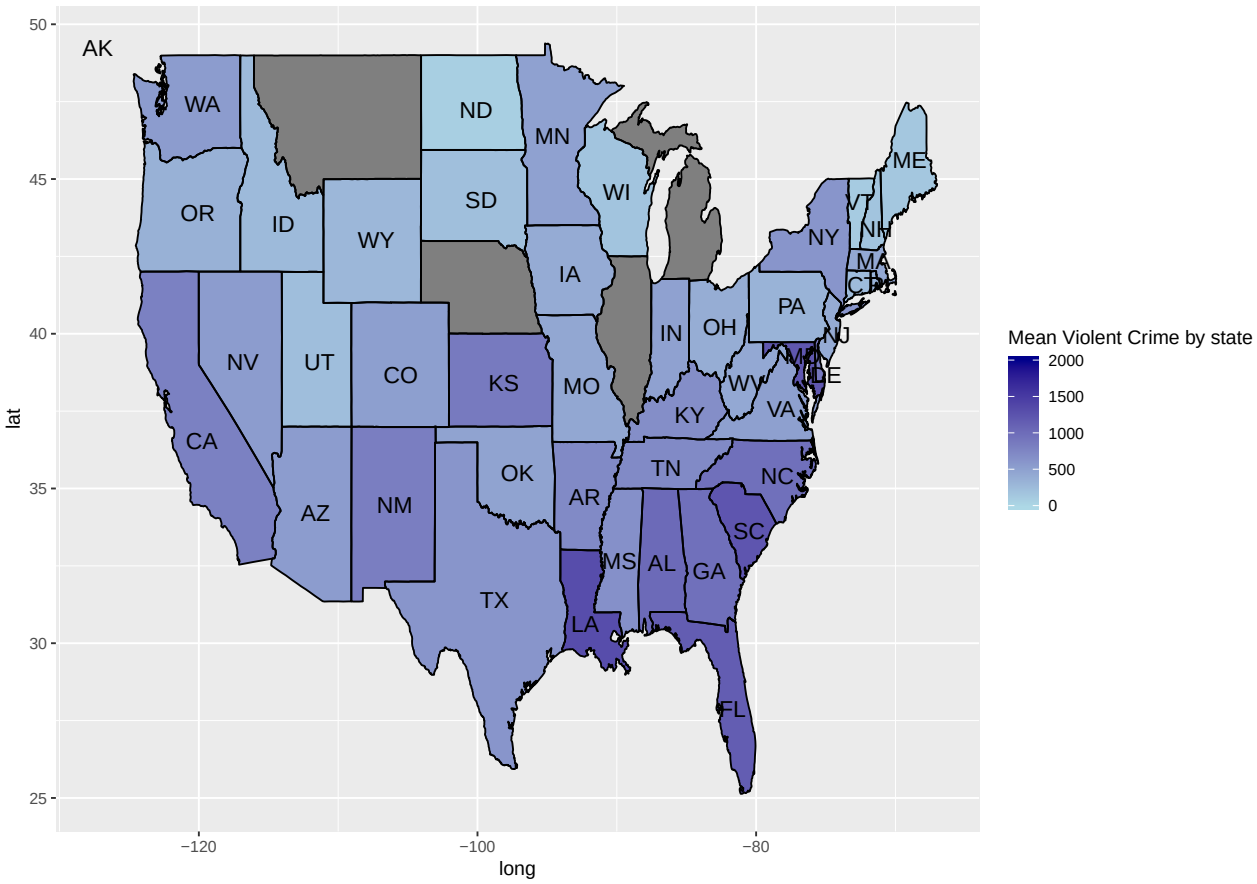
Answer: We create a heat map to display summary statistics at state level. We first extract the mean/max/min crime rate by state among other statistics. n=number of the obs'n in each state. Then we will display the mean violent crime by state in a heat map.

```
library(reshape2)
library(ggplot2)
library(dplyr)
data.s <- crime.all %>% group_by(state) %>% summarise(
  mean.crime=mean(violentcrimes.perpop, na.rm = TRUE),
  crime.min=min(violentcrimes.perpop),
  crime.max=max(violentcrimes.perpop),
  n=n())

crime_data <- data.s[, c("state", "mean.crime")]

crime_data$region <- tolower(state.name[match(crime_data$state, state.abb)])
crime_data$center_lat <- state.center$x[match(crime_data$state, state.abb)]
crime_data$center_long <- state.center$y[match(crime_data$state, state.abb)]

names(crime_data)
crime_data

states <- map_data("state")
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
map <- merge(states, crime_data , sort=FALSE, by="region", all.x=TRUE)

map <- map[order(map$order),]

ggplot(map, aes(x=long, y=lat, group=group))+ geom_polygon(aes(fill=mean.crime))+
  geom_path()+geom_text(data=crime_data, aes(x=center_lat, y=center_long, group=NA,
                                             label=state, size=2), show.legend =FALSE)+
  scale_fill_continuous(limits=c(0, 2000),name="Mean Violent Crime by state",
                        low="light blue", high="dark blue")
```



Summary: The heatmap displays that violent crimes happen more frequently in the southeast area of USA, with Louisiana state a significant high numbers. And the violent crime rate is higher in the east than that in the west. The five states with no values are marked as grey, without a state name.

Part II: LASSO selection

Our goal for the rest of the study is to find the factors that are related to violent crime. We will only use communities from two states `FL` and `CA` to assure the maximum possible number of variables.

1. Prepare a set of sensible factors/variables that you may use to build a model. You may show the R-chunk to show this step. Explain what varibles you may have excluded in the study and why? Or what other variables you have created to be included in the study.

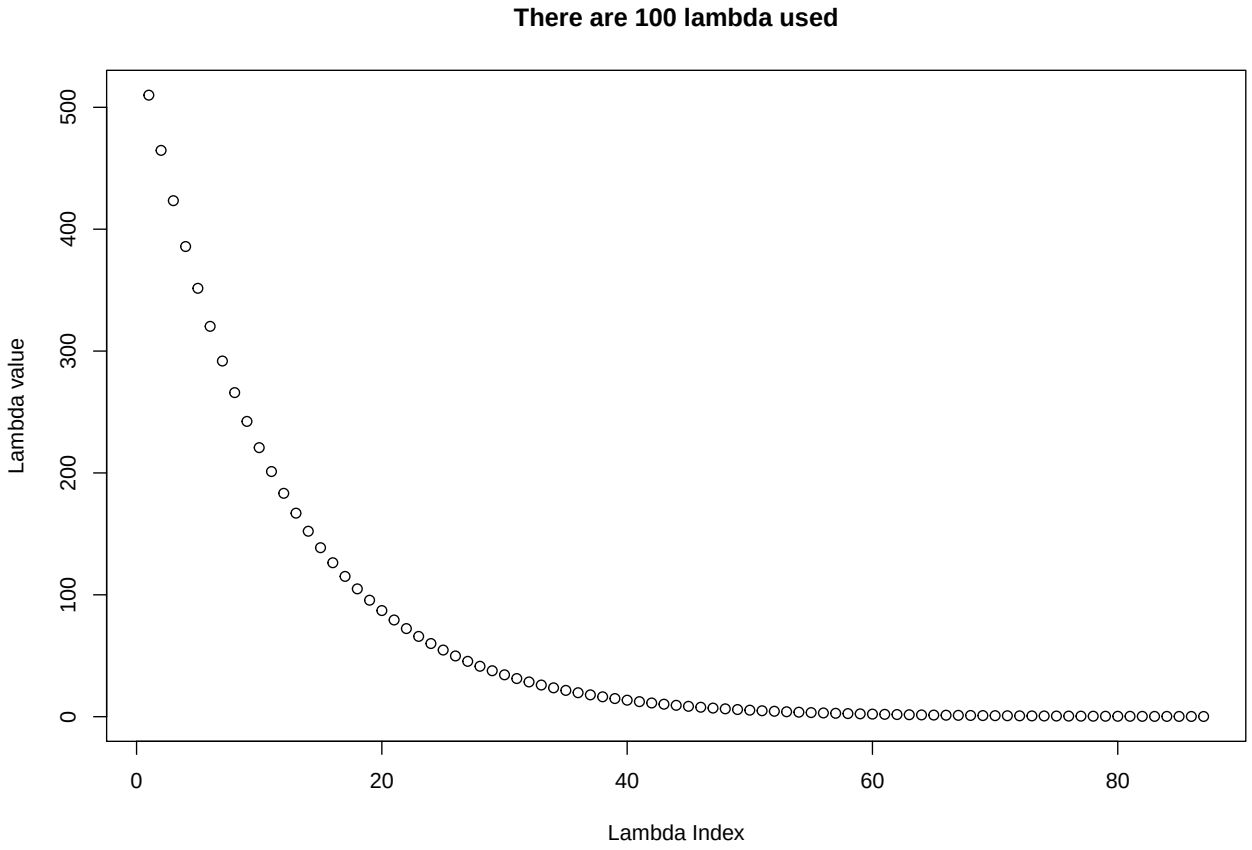
Answer: We use `filter()` funtion in `dplyr` to get the data from two states`FL` and `CA` and store it into *crime.new*. Then we get the predictors are all the variables except the last column and store it into $X.crime$. The response which is *violentcrimes.perpop* is stored in Y.

```
Y = crime[,99]
X <- model.matrix(violentcrimes.perpop~., data=crime)[, -1]
```

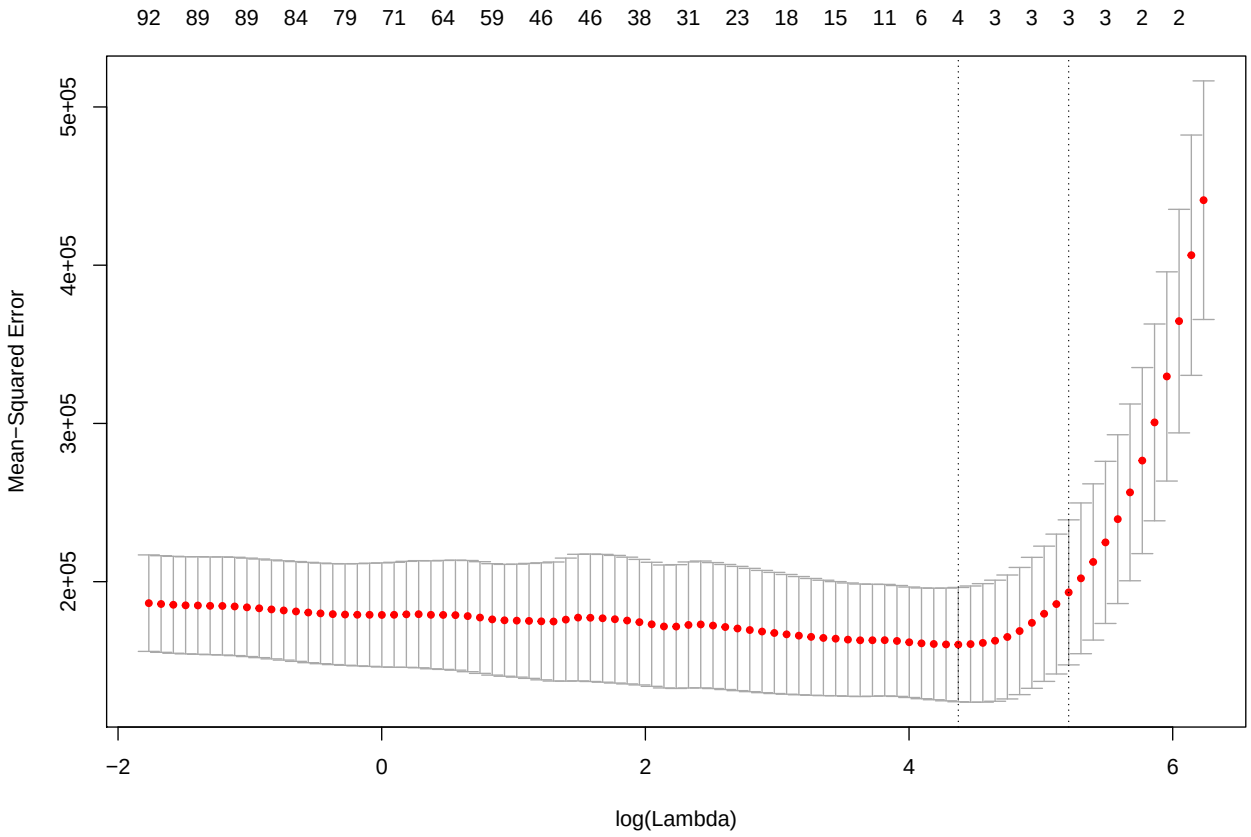
Then use LASSO to choose a reasonable, small model. Fit an OLS model with the variables obtained. The final model should only include variables with p-values < 0.05. Note: you may choose to use "lambda 1st" or "lambda min" to answer the following questions where applicable.

```
set.seed(5)
fit.cv <- cv.glmnet(X, Y, alpha=1, nfolds=10)
fit.cv$cvm # the mean cv error for each lambda
```

```
fit.cv$lambda.min # lambda.min returns the min point amoth all the cvm
fit.cv$nzero # number of non-zero coeff's returned for each lambda
plot(fit.cv$lambda , main = "There are 100 lambda used" , xlab = "Lambda Index" , ylab = "Lambda value")
```




```
plot(fit.cv)
```




```
#choose lambda min
coef.min <- coef(fit.cv, s="lambda.min")
coef.min <- coef.min[which(coef.min !=0),] # get the non=zero coefficients
#coef.min # the set of predictors chosen
rownames(as.matrix(coef.min)) # shows only names, not estimates

lasso<-glmnet(X, Y, alpha=1, lambda = fit.cv$lambda.min)
predict(lasso, s = 0,  type = 'coefficients')

coef.min
```

2. What is the model reported by LASSO?

Answer: By using "lambda min", we get four best predictors: race.pctblack, pct.kids2parents, pct.kids.nvrmarried, pct.house.vacant. The model reported by LASSO is:

$$violent\ crime\ rate = 1869.145 + 9.67 * race.pctblack - 19.58 * pct.kids2parents + 81.870 * pct.kids.nvrmarried + 5.14 * pct.house.va$$

3. What is the model after running OLS? Comment on the difference between the equation from questions (1) and (2)

```
#fit model
coef.min <- coef(fit.cv, s="lambda.min") #s=c("lambda.1se","lambda.min") or lambda val
coef.min <- coef.min[which(coef.min !=0),] # get the non=zero coefficients
var.min <- rownames(as.matrix(coef.min)) # output the names
lm.input <- as.formula(paste("violentcrimes.perpop", "~", paste(var.min[-1], collapse = "+")))
fit.min.lm <- lm(lm.input, data=crime)
lm.output <- coef(fit.min.lm) # output lm estimates
summary(fit.min.lm)
```

Answer: By using "lambda min", we get four best predictors: race.pctblack, pct.kids2parents, pct.kids.nvrmarried, pct.house.vacant. The model reported by LASSO is:

$$violent\ crime\ rate = 1999.426 + 13.112 * race.pctblack - 22.686 * pct.kids2parents + 85.510 * pct.kids.nvrmarried + 27.754 * pct.ho$$

LASSO model will be trained with biased, so we use lasso as a tool to select the predictors and then use OLS to train the unbiased model to ensure all the epislon and variables are independent.

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

Answer: Our final model is:

$$violent\ crime\ rate = 1999.426 + 13.112 * race.pctblack - 22.686 * pct.kids2parents + 85.510 * pct.kids.nvrmarried + 27.754 * pct.ho$$

a) What is your process of getting this final model?

Answer: To get this final model, we use the following process:
(1) use `glmnet` to produce LASSO estimate
(2) use cross validation to select a λ
(3) to help understanding the effect of λ, we plot several graphs
(4) choose "lambda.min" or "lambda.1st", to fit model with linear regression again
(5) compare results of different models as well as p-value
We find that the different between the result of "lambda.min" is the predictor `pct.house.vacant`, and it has p-value of 0.0123, which is not very low. So we excluded it in our final model.

b) Write a brief report based on your final model.

Answer: By using `glmnet` to produce LASSO estimate, we find that the race(percetage of black), percentage of kids with two parents, and percentage of kids that never married are three factors that related to the violent crime rate in state Florida and California.

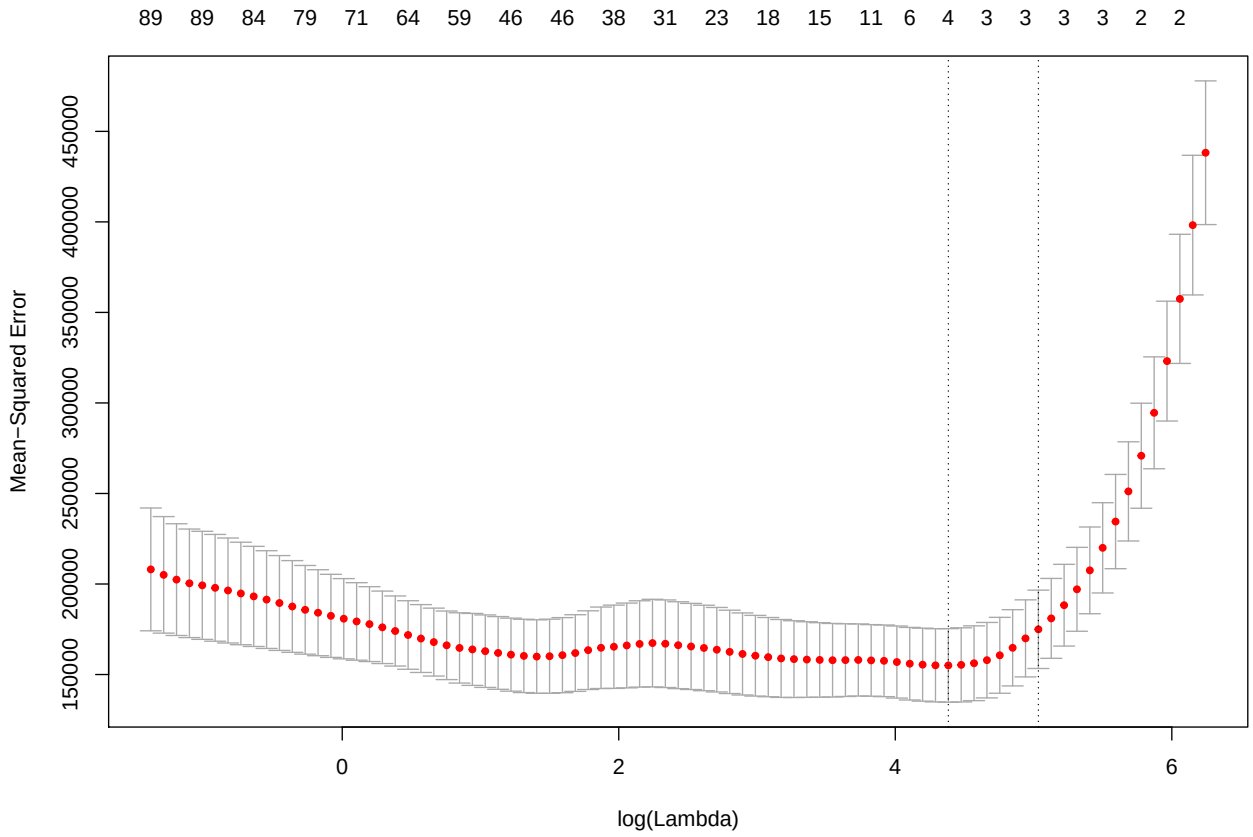
Part III: Elastic Net

Now, instead of LASSO, we want to consider how changing the value of α (i.e. mixing between LASSO and Ridge) will affect the model. Cross-validate between α and λ, instead of just λ. 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 parsimony.

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

Answer: We want α close to 1 so that it will do feature selection, yet still benefit from Ridge Regression.

```
fit.lambda <- glmnet(X, Y, alpha=.99)
fit.cv2 <- cv.glmnet(X, Y, alpha=.99, nfolds=10)
plot(fit.cv2)
```




```
#find the optimal lambda between
fit.cv2$lambda.min
fit.cv2$lambda.1se
```

Here, the $log(\lambda_{min})$ with the minimum mean squared error is 4.3837424, which means the $\lambda_{min} = 80.1373764$. The $log(\lambda_{1se})$ value is 5.0349786 and the $\lambda_{1se} = 153.6962964$. By looking at this we can get a better idea of our optimal λ.

```
set.seed(5)
#choose lambda to be 100
fit.final <- glmnet(X, Y, alpha=.99, lambda=100) # the final elastic net fit
beta.final <- coef(fit.final)
beta.final <- beta.final[which(beta.final !=0),]
beta.final <- as.matrix(beta.final)
rownames(beta.final)

coef <-coef(fit.final,s = "lambda.min")
coef
```

The α we choose is 0.99 and λ value is 100.

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

Answer: We choose the three best predictors and fit in the OLS model. Prediction error is 373.6.

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
fit.final_2=lm(violentcrimes.perpop~race.pctblack+pct.kids2parents+pct.kids.nvrmarried, crime)
summary(fit.final_2)

# prediction error
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model <- train(violentcrimes.perpop~race.pctblack+pct.kids2parents+pct.kids.nvrmarried,
               data = crime, method = "lm", trControl = train.control)
model
```

In conclusion, the equation for LASSO is:

$$violent\ crime\ rate(LASSO) = 1838.15 + 8.81 * race.pctblack - 18.83 * pct.kids2parents + 80.64 * pct.kids.nvrmarried$$

the equation for OLS is:

$$violent\ crime\ rate(OLS) = 2012.949 + 13.956 * race.pctblack - 22.678 * pct.kids2parents + 94.953 * pct.kids.nvrmarried$$

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

Answer: We find that race(percetage of black), percentage of kids with two parents, and percentage of kids that never married are three factors that related to the violent crime rate in state Florida and California. Each predictor in OLS will have less influence for the model, this maybe the reason that OLS model is unbiased while LASSO model is biased so it will have larger influence for the final prediction.