Homework 5

Qianbo Wang
uni: qw2180

Consider the Boston dataset, in R library MASS, on Housing Values in Suburbs of Boston, to fit a suitable model to predict medv (median value of owner-occupied homes in $1000s) using the following set of predictors: crim, zn, indus, nox, rm, age, tax.

Problem 1

Investigate whether there is any multicollinearity, and suggest remedial measures if appropriate.

- Detect multicollinearity

I just fit a multilinear regression on the data and then detect the multicollinearity.
First, use Variance Inflation Factor to detect if there is multicollinearity among the variables. And the VIFs are as follows:

ANOVA with VIF Result

```
Analysis of Variance Table

Response: medv
           Df  Sum Sq Mean Sq  F value  Pr(>F)    VIF
crim        1  6440.8  6440.8 179.5726 0.00000 1.1776
zn          1  3554.3  3554.3  99.0969 0.00000 1.0908
indus       1  2551.2  2551.2  71.1299 0.00000 1.0635
nox         1    28.7    28.7   0.7991 0.37180 1.0007
rm          1 11794.6 11794.6 328.8410 0.00000 1.3814
age         1    74.1    74.1   2.0656 0.15128 1.0017
tax         1   410.6   410.6  11.4491 0.00077 1.0097
Residuals 498 17861.9    35.9
```

Since from the result we can find that the VIFs of the variables are all < 10. Then calculate the mean of the VIF, $\overline{VIF} = 1.103626 > 1$. Since all $VIF < 10$, but the $\overline{VIF} > 1$, i.e. Average Variance of Inflation Factor is larger than 1. So there exists slight multicollinearity.

Second, use Condition Number to detect multicollinearity. The Condition Number of the correlation matrix is $19.45283 < 30$. So we can conclude that there is no serious multicollinearity among the explanatory variables. But there may exist slight multicollinearity.

- Multicollinearity Remedies

1. Use Ridge Regression
2. Do transformation on X and Y to get uncorrelated centered data for regression.
3. Use Principle Component Regression method to eliminate multicollinearity.

Problem 2

Compare models selected using LASSO and a stepwise procedure.

- Lasso Regression

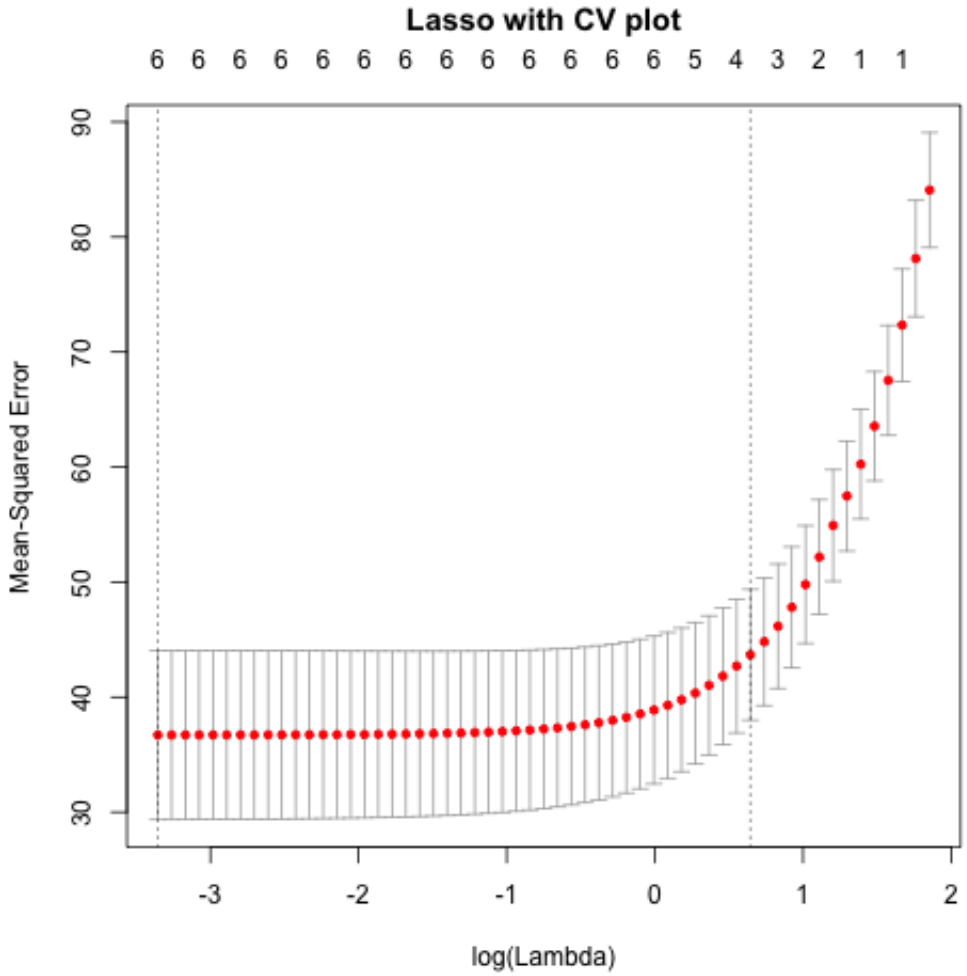
Use Lasso regression with cross validation to choose the best parameter λ, then we choose the best λ as

$$\lambda_{opt} = 0.0349$$

Then get the coefficients of lasso regression, the result is:

$$medv = -19.46 - 0.1303crim + 0.0213zn - 0.0157indus + 7.5757rm - 0.0228age - 0.0089tax$$

And the plot is as follows:



And then calculate the mean square error of Lasso regression, $MSE = 35.30283$.

- Stepwise Regression

Use stepwise regression and use AIC as the parameter to help choose the model. The forward stepwise regression result is as follows:

$$medv = -19.7132 - 0.1319crim + 0.0229zn + 7.6253rm - 0.0241age - 0.0093tax$$

Forward Stepwise Result

```
Call:  glm(formula = medv ~ rm + tax + crim + age + zn, data = data)

Coefficients:
(Intercept)           rm          tax         crim          age
 -19.713176     7.625253    -0.009323    -0.131852    -0.024121
         zn
   0.022947

Degrees of Freedom: 505 Total (i.e. Null);  500 Residual
Null Deviance:     42720
Residual Deviance: 17860  AIC: 3253
```

And the backward stepwise regression result is as follows:

$$medv = -19.7132 - 0.1319crim + 0.0229zn + 7.6253rm - 0.0241age - 0.0093tax$$

Backward Stepwise Result

```
Call:  glm(formula = medv ~ crim + zn + rm + age + tax, data = data)

Coefficients:
(Intercept)         crim           zn           rm          age
 -19.713176    -0.131852     0.022947     7.625253    -0.024121
        tax
  -0.009323

Degrees of Freedom: 505 Total (i.e. Null);  500 Residual
Null Deviance:     42720
Residual Deviance: 17860  AIC: 3253
```

Since both forward and backward stepwise regression give the same result, i.e. included crim, zn, tax, age and rm as explanatory variables. Then we should choose this as the stepwise result model. And then calculate the mean square error of stepwise regression, $MSE = 35.30361$.

- Compare

Compare the two methods, we can find that they have close MSE, but the stepwise model only have 5 explanatory variables, but the lasso contains all explanatory variables. And since lasso is more complex, so there may exist overfitting in lasso regression.

R Code:

```
rm(list=ls())
library(MASS)
multilinear<-lm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)
summary(multilinear)

sink("/Users/Qianbo/Google Drive/STAT W4201/HW5/multilinear.txt")
summary(multilinear)
sink()

#VIF detect multicollinearity
anova_table<-anova(multilinear)
SS<-anova(multilinear)$"Sum Sq"
VIF<-1/(1-SS[-length(SS)]/sum(SS))
anova_table$"VIF"<-c(VIF,"")
sink("/Users/Qianbo/Google Drive/STAT W4201/HW5/anova.txt")
anova_table
sink()
VIF_bar<-mean(VIF)

#correlation matrix

names<-c("crim","zn","indus","nox","rm","age","tax")
explanatory<-as.matrix(Boston[names])
dependent<-as.matrix(Boston["medv"])
corr_mat<-cor(explanatory)
eigen_values<-eigen(corr_mat)$values
con_number<-max(eigen_values)/min(eigen_values)

#ridge regression
ridge_reg<-lm.ridge(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)
ridge_coef<-ridge_reg$coef

#Lasso regression
library(glmnet)
names<-c("crim","zn","indus","nox","rm","age","tax")
explanatory<-as.matrix(Boston[names])
dependent<-as.matrix(Boston["medv"])

lasso_reg<-glmnet(explanatory,dependent)
cv_reg<-cv.glmnet(explanatory,dependent)
plot(cv_reg)

png(file="/Users/Qianbo/Google Drive/STAT W4201/HW5/lasso.png")
plot(cv_reg)
title("Lasso with CV plot",line = 2.5)
dev.off()

#choose model coefficient
lambda<-cv_reg$lambda.min
coeff<-coef(cv_reg,s="lambda.min")
predict_lasso<-predict(cv_reg,newx = explanatory,s="lambda.min")
MSE_lasso<-mean((dependent-predict_lasso)^2)

#stepwise regression
glm_model1<-glm(medv~1,data=Boston)
glm_model2<-glm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)

backward<-stepAIC(glm_model2,direction = "backward",scope=list(
  upper = glm_model2,lower = glm_model1),trace = F)

forward<-stepAIC(glm_model1,direction = "forward",scope=list(
  upper = glm_model2,lower = glm_model1),trace = F)

sink("/Users/Qianbo/Google Drive/STAT W4201/HW5/forward.txt")
forward
sink()
```

```
sink("/Users/Qianbo/Google Drive/STAT W4201/HW5/backward.txt")
backward
sink()

#calculate MSE
predict_back<-predict(backward,newx = explanatory)
predict_for<-predict(forward,newx = explanatory)
MSE_back<-mean((dependent-predict_back)^2)
MSE_for<-mean((dependent-predict_for)^2)
```