Jiahong Hu

Jh3561

HW5 STAT 4201

Problem 1

```
> summary(fit)

Call:
lm(formula = medv ~ crim + zn + indus + nox + rm + age + tax,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-16.625  -3.161  -0.833   2.089  41.042

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.615259   3.221482  -6.089 2.27e-09 ***
crim         -0.132538   0.038482  -3.444 0.000621 ***
zn            0.022103   0.014823   1.491 0.136547
indus        -0.014980   0.072282  -0.207 0.835909
nox           0.010643   4.230468   0.003 0.997994
rm            7.606508   0.418424  18.179  < 2e-16 ***
age          -0.023198   0.014893  -1.558 0.119964
tax          -0.009006   0.002662  -3.384 0.000772 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.989 on 498 degrees of freedom
Multiple R-squared:  0.5818, Adjusted R-squared:  0.576
F-statistic: 98.99 on 7 and 498 DF,  p-value: < 2.2e-16
```

```
> x_data<-data[,c("crim","zn","indus","nox","rm","age","tax")]
> names(x_data)
[1] "crim"  "zn"    "indus" "nox"   "rm"    "age"   "tax"
> correlation<-cor(x_data)
> correlation
            crim         zn      indus        nox         rm
crim   1.0000000 -0.2004692  0.4065834  0.4209717 -0.2192467
zn    -0.2004692  1.0000000 -0.5338282 -0.5166037  0.3119906
indus  0.4065834 -0.5338282  1.0000000  0.7636514 -0.3916759
nox    0.4209717 -0.5166037  0.7636514  1.0000000 -0.3021882
rm    -0.2192467  0.3119906 -0.3916759 -0.3021882  1.0000000
age    0.3527343 -0.5695373  0.6447785  0.7314701 -0.2402649
tax    0.5827643 -0.3145633  0.7207602  0.6680232 -0.2920478
             age        tax
crim   0.3527343  0.5827643
zn    -0.5695373 -0.3145633
indus  0.6447785  0.7207602
nox    0.7314701  0.6680232
rm    -0.2402649 -0.2920478
age    1.0000000  0.5064556
tax    0.5064556  1.0000000
```

The correlation matrix for the explanatory matrix shows there are relative strong linear associations between “ indus” and “age” ,” indus” and “tax”,”nox” and “age”, “nox” and “tax”.

Then, I use VIF – Variance Inflation Factors to detect the multicollinearity.

```
> x_data<-data[,c("crim","zn","indus","nox","rm","age","tax")]
> library(usdm)
> vif(x_data)
  Variables      VIF
1      crim 1.542630
2        zn 1.682654
3     indus 3.462196
4       nox 3.383524
5        rm 1.216923
6       age 2.474575
7       tax 2.833196
```

The chart indicates that all the VIF for the seven explanatory variables are above 1, therefore the average VIF of the seven explanatory variables is sure above 1. Hence, there exists serious multicollineariy.

Remedy: Ridge regression produces a slight biased estimator with smaller variance, which leads to a smaller MSE overall.

First, I use lambda =2.

```
> fit1<-lm.ridge(medv~crim+zn+indus+nox+rm+age+tax,data=data,lambda=2)
> fit1
                    crim           zn        indus          nox           rm          age          tax 
-19.376675030 -0.132631826  0.022139221 -0.017326852 -0.061887534  7.571260182 -0.022937453 -0.008929141 
> 
```

The difference of coefficients produced by ridge and OLS is not very obvious.

Then, I use cross validation to find the optimal lambda that produces the lowest MSE.

```
> ridge.opt<-ridge.cv(x_data,y_data,lambda=c(1,5,10,50,100),plot.it=TRUE)
> library(parcor)
> ridge.opt<-ridge.cv(x_data,y_data,lambda=c(1,5,10,15,20),plot.it=TRUE)
> fit1<-lm.ridge(medv~crim+zn+indus+nox+rm+age+tax,data=data,lambda=2)
> library(parcor)
> ridge.opt<-ridge.cv(x_data,y_data,lambda=c(1,5,10,15,20,25,30),plot.it=TRUE)
> ridge.opt$coefficients
       Xcrim          Xzn       Xindus         Xnox          Xrm         Xage         Xtax 
-0.132830213  0.022492415 -0.035110813 -0.639584891  7.271602847 -0.021019229 -0.008353003 
> ridge.opt$lambda.opt
[1] 20
```

```
> ridge.opt$intercept

-17.34498
```

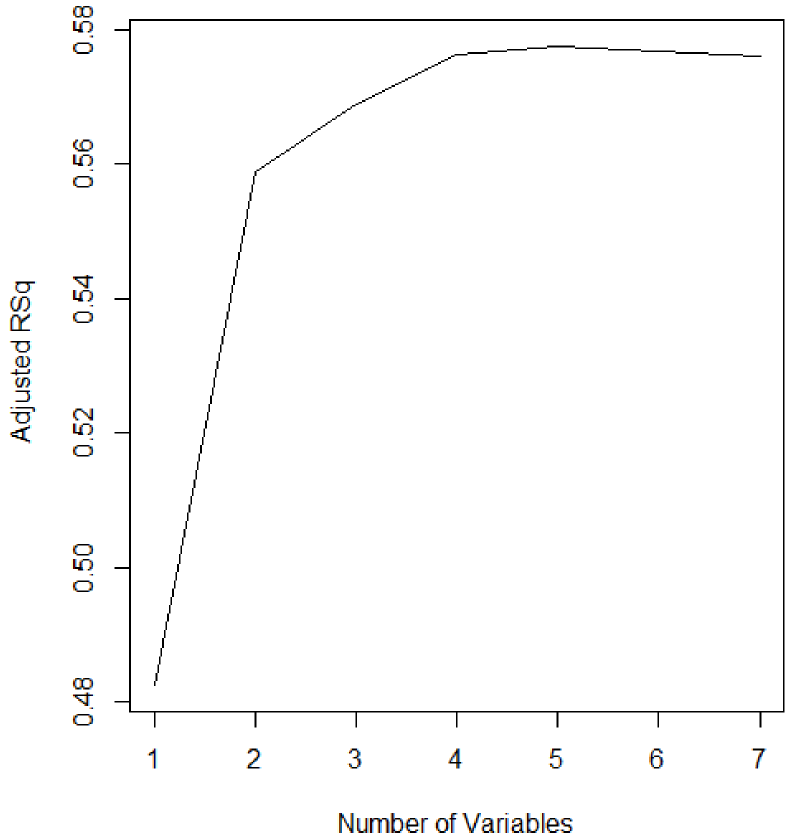
The optimal lambda is 20 and the regression result is

Y = -17.34 – 0.13 crim + 0.02 zn-0.04indus-0.06nox+7.27xrm-0.02age-0.01tax

Problem 2

1. Best subset selection – Exhaustive Search

```
> library(leaps)
> data_new<-data[,c("medv","crim","zn","indus","nox","rm","age","tax")]
> data_new<-data.frame(data_new)
> regfit.full=regsubsets(medv~.,data_new )
> summary(regfit.full)
Subset selection object
Call: regsubsets.formula(medv ~ ., data_new)
7 Variables  (and intercept)
      Forced in Forced out
crim      FALSE      FALSE
zn        FALSE      FALSE
indus     FALSE      FALSE
nox       FALSE      FALSE
rm        FALSE      FALSE
age       FALSE      FALSE
tax       FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: exhaustive
         crim zn  indus nox rm  age tax
1  ( 1 ) " "  " " " "   " " "*" " " " "
2  ( 1 ) " "  " " " "   " " "*" " " "*"
3  ( 1 ) "*"  " " " "   " " "*" " " "*"
4  ( 1 ) "*"  " " " "   " " "*" "*" "*"
5  ( 1 ) "*"  "*" " "   " " "*" "*" "*"
6  ( 1 ) "*"  "*" "*"   " " "*" "*" "*"
7  ( 1 ) "*"  "*" "*"   "*" "*" "*" "*"
> reg.summary=summary (regfit.full)
> plot(reg.summary$adjr2 ,xlab =" Number of Variables ",ylab=" Adjusted RSq",type="l")
> which.max (reg.summary$adjr2)
[1] 5
> fit2<-lm(medv~crim+zn+rm+age+tax,data=data)
> summary(fit2)

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

Residuals:
    Min      1Q  Median      3Q     Max 
-16.669  -3.167  -0.808   2.075  41.083 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -19.713176   2.862677  -6.886 1.73e-11 ***
crim         -0.131852   0.038261  -3.446 0.000617 ***
zn            0.022947   0.014231   1.612 0.107487    
rm            7.625253   0.408770  18.654  < 2e-16 ***
age          -0.024121   0.012709  -1.898 0.058271 .  
tax          -0.009323   0.002139  -4.358 1.59e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.977 on 500 degrees of freedom
Multiple R-squared:  0.5818,	Adjusted R-squared:  0.5776 
F-statistic: 139.1 on 5 and 500 DF,  p-value: < 2.2e-16

> 
```



According to the exhaustive research, the best model with lowest adjusted R^2 has 5 variables.

The mode:

Y = -19.71-0.13crim+0.02zn+7.62rn-0.02age-0.01tax

2. Forward research

```
> full=lm(medv~crim+zn+indus+nox+rm+age+tax,data=data_new)
> null=lm(medv~1,data=data_new)
> step(null, scope=list(upper=full, lower=null), direction='forward', trace=TRUE)
Start:  AIC=2246.51
medv ~ 1

        Df Sum of Sq   RSS    AIC
+ rm     1   20654.4 22062 1914.2
+ indus  1    9995.2 32721 2113.6
+ tax    1    9377.3 33339 2123.1
+ nox    1    7800.1 34916 2146.5
+ crim   1    6440.8 36276 2165.8
+ age    1    6069.8 36647 2171.0
+ zn     1    5549.7 37167 2178.1
<none>               42716 2246.5

Step:  AIC=1914.19
medv ~ rm

        Df Sum of Sq   RSS    AIC
+ tax    1    3290.8 18771 1834.5
+ crim   1    2496.1 19566 1855.4
+ indus  1    2254.3 19808 1861.7
+ nox    1    2217.5 19844 1862.6
+ age    1    1997.0 20065 1868.2
+ zn     1     974.5 21087 1893.3
<none>               22062 1914.2

Step:  AIC=1834.45
medv ~ rm + tax

        Df Sum of Sq   RSS    AIC
+ crim   1    472.61 18299 1823.5
+ age    1    403.42 18368 1825.5
+ zn     1    311.59 18460 1828.0
+ nox    1    189.01 18582 1831.3
+ indus  1    120.36 18651 1833.2
<none>               18771 1834.5

Step:  AIC=1823.55
medv ~ rm + tax + crim

        Df Sum of Sq   RSS    AIC
+ age    1    341.95 17957 1816.0
+ zn     1    306.14 17992 1817.0
+ nox    1    164.24 18134 1821.0
+ indus  1    141.92 18157 1821.6
<none>               18299 1823.5

Step:  AIC=1816
medv ~ rm + tax + crim + age

        Df Sum of Sq   RSS    AIC
+ zn     1    92.895 17864 1815.4
<none>               17957 1816.0
+ indus  1    14.350 17942 1817.6
+ nox    1     4.132 17952 1817.9

Step:  AIC=1815.38
medv ~ rm + tax + crim + age + zn

        Df Sum of Sq   RSS    AIC
<none>               17864 1815.4
+ indus  1   1.71889 17862 1817.3
+ nox    1   0.17872 17863 1817.4

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

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

Under the forward selection, the best model is also with five variables and is close to the model under the best subset research.

Y = -19.71+7.72rm-0.01tax-0.13crim-0.02age+0.02zn

3. Backwards

```
> step(full, scope=list(upper=full, lower=null), direction='backward', trace=TRUE)
Start:  AIC=1819.33
medv ~ crim + zn + indus + nox + rm + age + tax

        Df Sum of Sq   RSS    AIC
- nox    1       0.0 17862 1817.3
- indus  1       1.5 17863 1817.4
<none>               17862 1819.3
- zn     1      79.8 17942 1819.6
- age    1      87.0 17949 1819.8
- tax    1     410.6 18273 1828.8
- crim   1     425.5 18287 1829.2
- rm     1   11853.2 29715 2074.9

Step:  AIC=1817.33
medv ~ crim + zn + indus + rm + age + tax

        Df Sum of Sq   RSS    AIC
- indus  1       1.7 17864 1815.4
<none>               17862 1817.3
- zn     1      80.3 17942 1817.6
- age    1     106.8 17969 1818.3
- crim   1     425.9 18288 1827.2
- tax    1     432.9 18295 1827.5
- rm     1   11853.4 29715 2072.9

Step:  AIC=1815.38
medv ~ crim + zn + rm + age + tax

       Df Sum of Sq   RSS    AIC
<none>              17864 1815.4
- zn    1      92.9 17957 1816.0
- age   1     128.7 17992 1817.0
- crim  1     424.3 18288 1825.2
- tax   1     678.6 18542 1832.2
- rm    1   12432.3 30296 2080.7

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

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

The mode under the backward research:

Medv = -19.71-0.13crim+0.02zn+7.62rm-0.02age-0.01tax

4. Efroymson's method

```
> step(null, scope=list(upper=full, lower=null), direction='both', trace=TRUE)
Start:  AIC=2246.51
medv ~ 1

        Df Sum of Sq   RSS    AIC
+ rm     1   20654.4 22062 1914.2
+ indus  1    9995.2 32721 2113.6
+ tax    1    9377.3 33339 2123.1
+ nox    1    7800.1 34916 2146.5
+ crim   1    6440.8 36276 2165.8
+ age    1    6069.8 36647 2171.0
+ zn     1    5549.7 37167 2178.1
<none>               42716 2246.5

Step:  AIC=1914.19
medv ~ rm

        Df Sum of Sq   RSS    AIC
+ tax    1    3290.8 18771 1834.5
+ crim   1    2496.1 19566 1855.4
+ indus  1    2254.3 19808 1861.7
+ nox    1    2217.5 19844 1862.6
+ age    1    1997.0 20065 1868.2
+ zn     1     974.5 21087 1893.3
<none>               22062 1914.2
- rm     1   20654.4 42716 2246.5

Step:  AIC=1834.45
medv ~ rm + tax

        Df Sum of Sq   RSS    AIC
+ crim   1     472.6 18298 1823.5
+ age    1     403.4 18368 1825.5
+ zn     1     311.6 18459 1828.0
+ nox    1     189.0 18582 1831.3
+ indus  1     120.4 18651 1833.2
<none>               18771 1834.5
- tax    1    3290.8 22062 1914.2
- rm     1   14567.9 33339 2123.1

Step:  AIC=1823.55
medv ~ rm + tax + crim

        Df Sum of Sq   RSS    AIC
+ age    1     341.9 17957 1816.0
+ zn     1     306.1 17992 1817.0
+ nox    1     164.2 18134 1821.0
+ indus  1     141.9 18157 1821.6
<none>               18298 1823.5
- crim   1     472.6 18771 1834.5
- tax    1    1267.3 19566 1855.4
- rm     1   14181.2 32480 2111.9

Step:  AIC=1816
medv ~ rm + tax + crim + age
Step:  AIC=1815.38
medv ~ rm + tax + crim + age + zn

        Df Sum of Sq   RSS    AIC
<none>               17864 1815.4
- zn     1      92.9 17957 1816.0
- age    1     128.7 17992 1817.0
+ indus  1       1.7 17862 1817.3
+ nox    1       0.2 17863 1817.4
- crim   1     424.3 18288 1825.2
- tax    1     678.6 18542 1832.2
- rm     1   12432.3 30296 2080.7

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

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

The model under this method is medv=-19.71+7.62rm-0.01tax-0.13crim-0.02age+0.02zn

In summary , all the results are very close to each other.

Part b)

Lasso

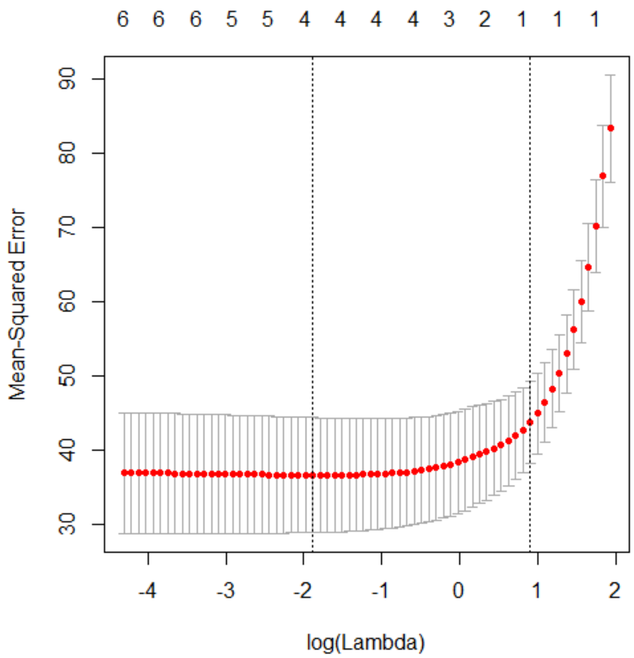
By using the 10-fold cross validation, we have optimal lambda = 0.03, and the model has 6 variables.

Y =-19.46-0.13crim+0.02zn-0.02indus+7.57rm -0.02 age-0.01tax

```
> mylars(x_data,y_data,k=10)
> lasso.opt=mylars(x_data,y_data,k=10)
> lasso.opt$lambda
 [1] 6.38897522 5.82139627 5.30423947 4.83302546 4.40367279 4.01246264 3.65600652 3.33121698 3.03528084
[10] 2.76563486 2.51994348 2.29607864 2.09210135 1.90624483 1.73689929 1.58259795 1.44200431 1.31390062
[19] 1.19717731 1.09082338 0.99391763 0.90562073 0.82516788 0.75186224 0.68506887 0.62420924 0.56875621
[28] 0.51822948 0.47219140 0.43024322 0.39202160 0.35719548 0.32546322 0.29654996 0.27020528 0.24620099
[37] 0.22432917 0.20440038 0.18624202 0.16969679 0.15462140 0.14088526 0.12836940 0.11696542 0.10657453
[46] 0.09710674 0.08848005 0.08061972 0.07345769 0.06693191 0.06098586 0.05556805 0.05063154 0.04613357
[55] 0.04203519 0.03830090 0.03489835 0.03179808
> lasso.opt$lambda.opt
[1] 0.03179808
> lasso.opt$cofficients
NULL
> lasso.opt=mylars(x_data,y_data,k=10)
> lasso.opt$lambda.opt
[1] 0.03179808
> lasso.opt$coefficients
           1            2            3            4            5            6            7 
-0.130487200  0.021396857 -0.015646811  0.000000000  7.578436157 -0.022818091 -0.008950752 
> lasso.opt$intercept
[1] -19.47349
>
```

By use 400 data as training data and 106 as testing data, the model has 6 variables.
The model medv = Y =-18.95-0.12crim+0.02zn-0.02indus+7.47rm -0.02 age-0.01tax

```
> library(glmnet)
> grid=10^seq(10,-2,length=100)
> x_train=x_data[1:400,]
> y_train=y_data[1:400]
> x_test=x_data[401:506,]
> y_test=y_data[401:506]
>
> lasso.mod=glmnet(x_train,y_train,alpha =1,lambda=grid)
> plot(lasso.mod)
>
>
> set.seed(1)
> cv.out=cv.glmnet(x_train,y_train,alpha=1)
> plot(cv.out)
> bestlam=cv.out$lambda.min
> out=glmnet(x_data,y_data,alpha=1,lambda=grid)
> lasso.coef=predict(out,type="coefficients",s=bestlam)
> lasso.coef
8 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept) -18.953267463
crim         -0.122941605
zn            0.018802811
indus        -0.018329591
nox           .
rm            7.472120555
age          -0.021436186
tax          -0.008741906
>
```



In summary, the ridge regression produces a model with 7 variables, lasso with 6 variables, forwards and backwards with 5 variables. The best subset model has five variables. Overall, they all give similar results.