ISE789/OR791 HW4

(Due: 11/4 before class)

1. Using *ozone* data, fit a model with *O3* as the response and *temp*, *humidity*, and *ibh* as predictors. Use the Box-Cox method to determine the best transformation on the response.

2. For the *prostate* data, fit a model with *lpsa* as the response and the other variables as predictors.
    a) Compute and comment on the condition numbers.
    b) Compute and comment on the correlation between the predictors.
    c) Compute the variance inflation factors.

3. For the same *prostate* data, do the following (You may directly use any package you prefer).
    a) Fit a ridge regression using GCV.
    b) Perform all subsets regression and select the best model using (i) Adjusted R^2 and (ii) Cp.
    c) Perform stepwise regression using AIC.
    d) Fit a nonnegative garrote using GCV.
    e) Fit lasso and select the model using leave-one-out cross validation.
    f) Fit LARS and select the model using Cp.

ISE 789 HW-4

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

1. Using ozone data, fit a model with O3 as the response and temp, humidity, and ibh as predictors. Use the Box-Cox method to determine the best transformation on the response.

Let's first get the ozone data and take a look at it

Hide

```
library(faraway)
data(ozone)
head(ozone)
```

	O3 <dbl>	vh <dbl>	wind <dbl>	humidity <dbl>	temp <dbl>	ibh <dbl>	dpg <dbl>	ibt <dbl>	vis <dbl>
1	3	5710	4	28	40	2693	-25	87	250
2	5	5700	3	37	45	590	-24	128	100
3	5	5760	3	51	54	1450	25	139	60
4	6	5720	4	69	35	1568	15	121	60
5	4	5790	6	19	45	2631	-33	123	100
6	4	5790	3	25	55	554	-28	182	250

6 rows | 1-10 of 10 columns

Let's fit the model now.

Hide

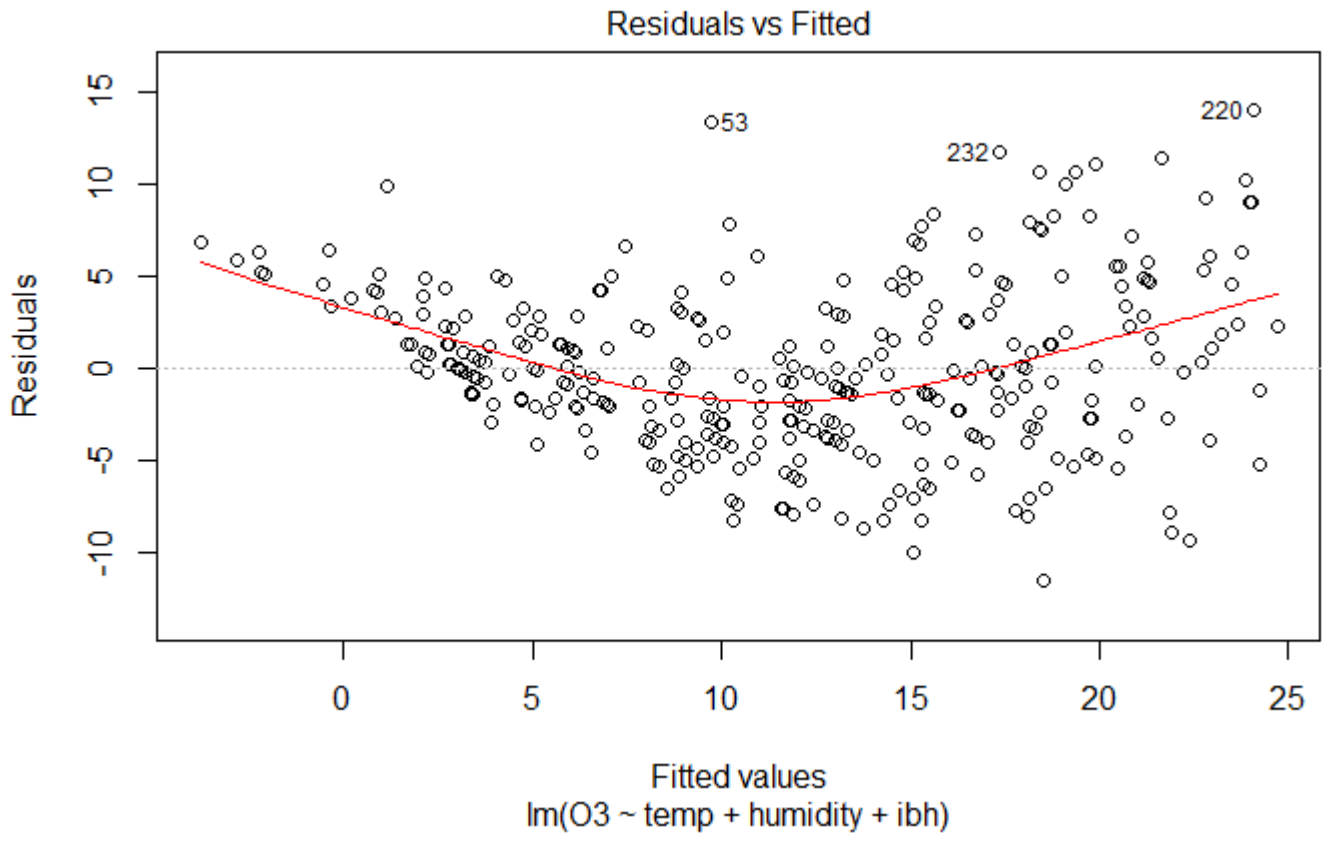
```
md <-lm(O3~temp+humidity+ibh,data=ozone)
summary(md)
```

```
Call:
lm(formula = O3 ~ temp + humidity + ibh, data = ozone)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5291  -3.0137  -0.2249   2.8239  13.9303

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.049e+01  1.616e+00  -6.492 3.16e-10 ***
temp         3.296e-01  2.109e-02  15.626  < 2e-16 ***
humidity     7.738e-02  1.339e-02   5.777 1.77e-08 ***
ibh         -1.004e-03  1.639e-04  -6.130 2.54e-09 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 4.524 on 326 degrees of freedom
Multiple R-squared:  0.684, Adjusted R-squared:  0.6811
F-statistic: 235.2 on 3 and 326 DF,  p-value: < 2.2e-16
```

Let's check if model satisfying all the assumptions or not.

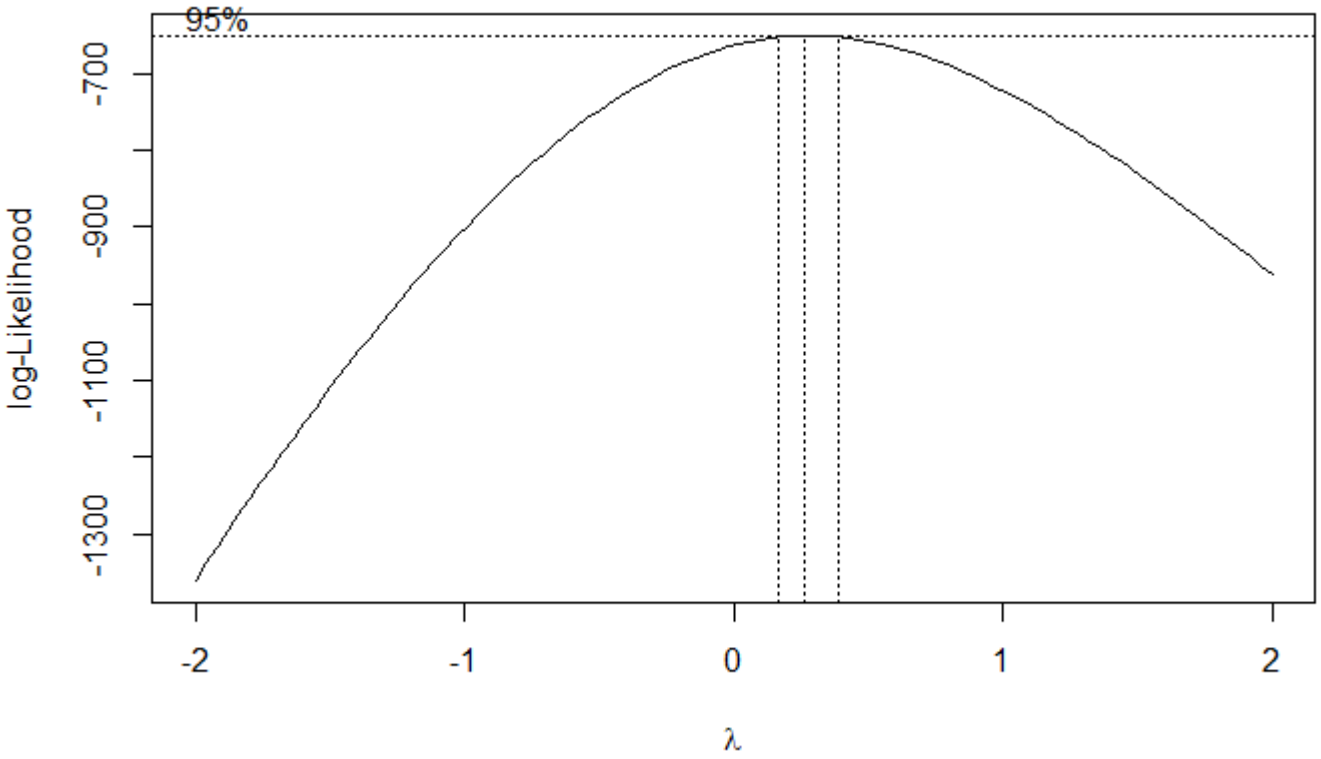
Hide

```
plot(md,which=1)
```



Hide

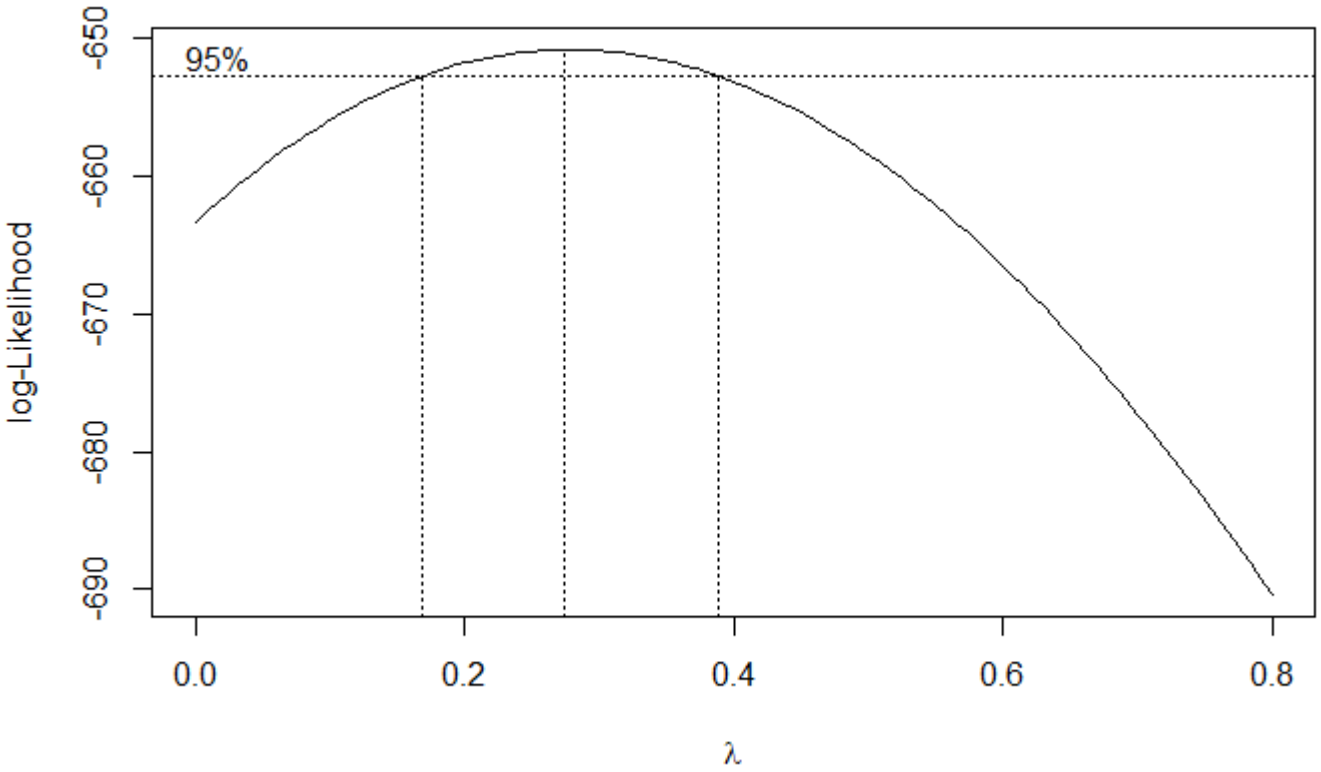
```
library(MASS)
bc<-boxcox(md,plotit=T)
```



Let's take a value of lambda and see how it improves the plot

Hide

```
bc<-boxcox(md,plotit=T,lambda=seq(0,0.8,by=0.1))
```



Let's find the best lambda

Hide

```
which.max(bc$y)
```

```
[1] 35
```

Hide

```
(lambda<-bc$x[which.max(bc$y)])
```

```
[1] 0.2747475
```

Now since we got our lambda, let's use this to rebuild the model and see the statistics

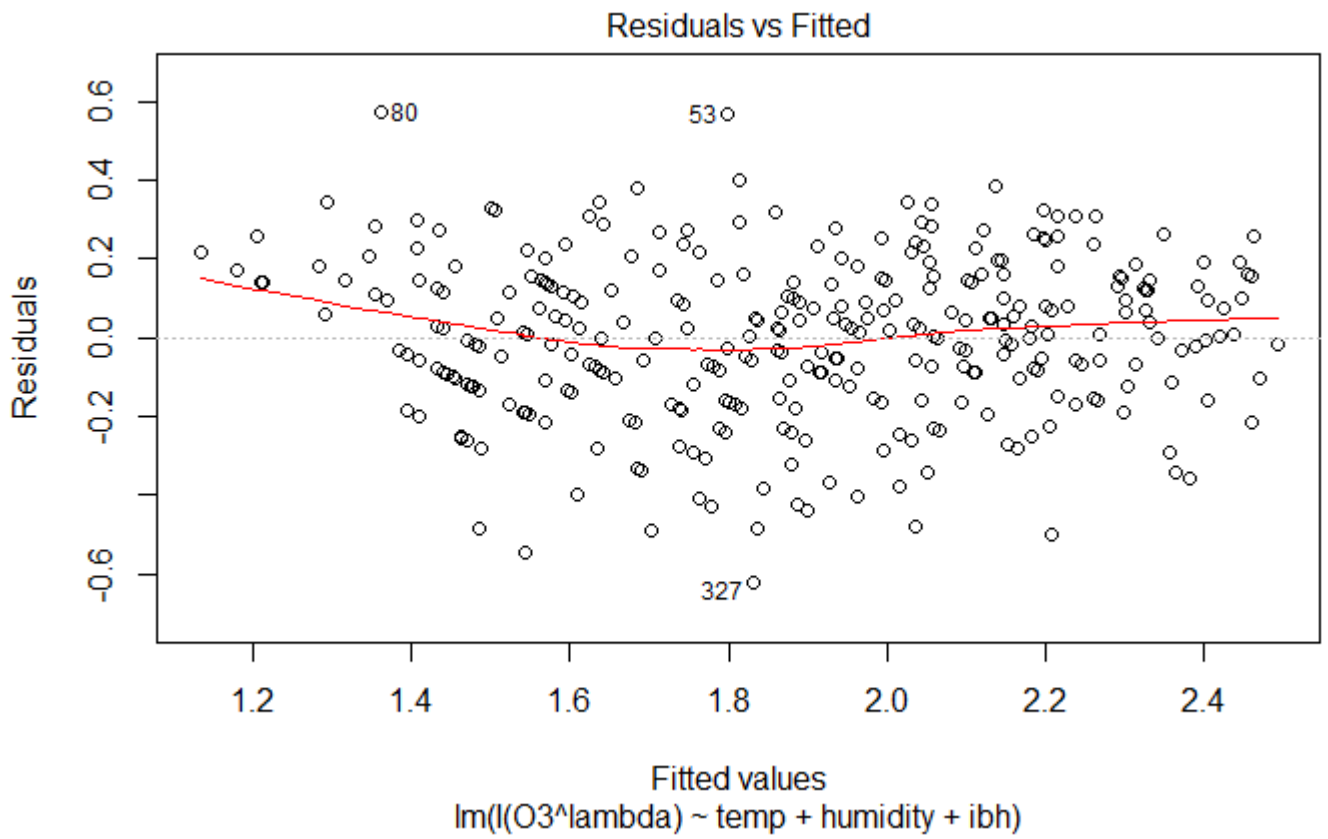
Hide

```
md_best<-lm(I(O3^lambda)~temp+humidity+ibh,data=ozone)
summary(md_best)
```

```
Call:
lm(formula = I(O3^lambda) ~ temp + humidity + ibh, data = ozone)

Residuals:
     Min       1Q   Median       3Q      Max
-0.62195 -0.12426  0.01026  0.14352  0.57016

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.779e-01  7.176e-02  12.233  < 2e-16 ***
temp         1.521e-02  9.365e-04  16.242  < 2e-16 ***
humidity     3.479e-03  5.946e-04   5.850 1.19e-08 ***
ibh         -5.610e-05  7.274e-06  -7.711 1.52e-13 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 0.2009 on 326 degrees of freedom
Multiple R-squared:  0.7161,    Adjusted R-squared:  0.7135
F-statistic: 274.1 on 3 and 326 DF,  p-value: < 2.2e-16
```

Hide

```
plot(md_best,which=1)
```



We, can see how because of transformations now, the variance is constant in our model

2. For the prostate data, fit a model with lpsa as the response and the other variables as predictors.

a. Compute and comment on the condition numbers.
b. Compute and comment on the correlation between the predictors.
c. Compute the variance inflation factors.

Hide

```
library(faraway)
attach(prostate)
```

```
The following objects are masked from prostate (pos = 8):

    age, gleason, lbph, lcavol, lcp, lpsa, lweight, pgg45, svi
```

Hide

```
a = lm(lpsa~.,prostate)
summary(a)
```

```

Call:
lm(formula = lpsa ~ ., data = prostate)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.7331 -0.3713 -0.0170  0.4141  1.6381 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.669337   1.296387   0.516  0.60693    
lcavol       0.587022   0.087920   6.677 2.11e-09 ***
lweight      0.454467   0.170012   2.673  0.00896 ** 
age         -0.019637   0.011173  -1.758  0.08229 .  
lbph         0.107054   0.058449   1.832  0.07040 .  
svi          0.766157   0.244309   3.136  0.00233 ** 
lcp         -0.105474   0.091013  -1.159  0.24964    
gleason      0.045142   0.157465   0.287  0.77503    
pgg45        0.004525   0.004421   1.024  0.30886    
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 0.7084 on 88 degrees of freedom
Multiple R-squared:  0.6548,    Adjusted R-squared:  0.6234 
F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

Hide

```
x=model.matrix(a)[,-1]
e=eigen(t(x)%*%x)
condition = sqrt(e$val[1]/e$val)
condition
```

```
[1]   1.00000   2.78186  47.66094  52.22787  85.98499 103.73114 153.85414 243.30248
```

There is multicollinearity in data because the condition values are higher than the threshold

Hide

```
round(cor(prostate),4)
```

```
        lcavol lweight    age    lbph     svi     lcp gleason  pgg45   lpsa
lcavol  1.0000  0.1941 0.2250  0.0273  0.5388  0.6753  0.4324 0.4337 0.7345
lweight 0.1941  1.0000 0.3075  0.4349  0.1088  0.1002 -0.0013 0.0508 0.3541
age     0.2250  0.3075 1.0000  0.3502  0.1177  0.1277  0.2689 0.2761 0.1696
lbph    0.0273  0.4349 0.3502  1.0000 -0.0858 -0.0070  0.0778 0.0785 0.1798
svi     0.5388  0.1088 0.1177 -0.0858  1.0000  0.6731  0.3204 0.4576 0.5662
lcp     0.6753  0.1002 0.1277 -0.0070  0.6731  1.0000  0.5148 0.6315 0.5488
gleason 0.4324 -0.0013 0.2689  0.0778  0.3204  0.5148  1.0000 0.7519 0.3690
pgg45   0.4337  0.0508 0.2761  0.0785  0.4576  0.6315  0.7519 1.0000 0.4223
lpsa    0.7345  0.3541 0.1696  0.1798  0.5662  0.5488  0.3690 0.4223 1.0000
```

We can see correlation is more than 50% in many of the predictors for example svi and lcp has correlation of 67.31%

Hide

```
round(vif(x),4)
```

```
 lcavol lweight     age    lbph     svi     lcp gleason   pgg45
 2.0541  1.3637  1.3236  1.3755  1.9569  3.0980  2.4734  2.9744
```

Hide

```
round(sqrt(vif(x)),4)
```

```
 lcavol lweight     age    lbph     svi     lcp gleason   pgg45
 1.4332  1.1678  1.1505  1.1728  1.3989  1.7601  1.5727  1.7246
```

Larger Variance inflation factor gives indication of multicollinearity between the variables

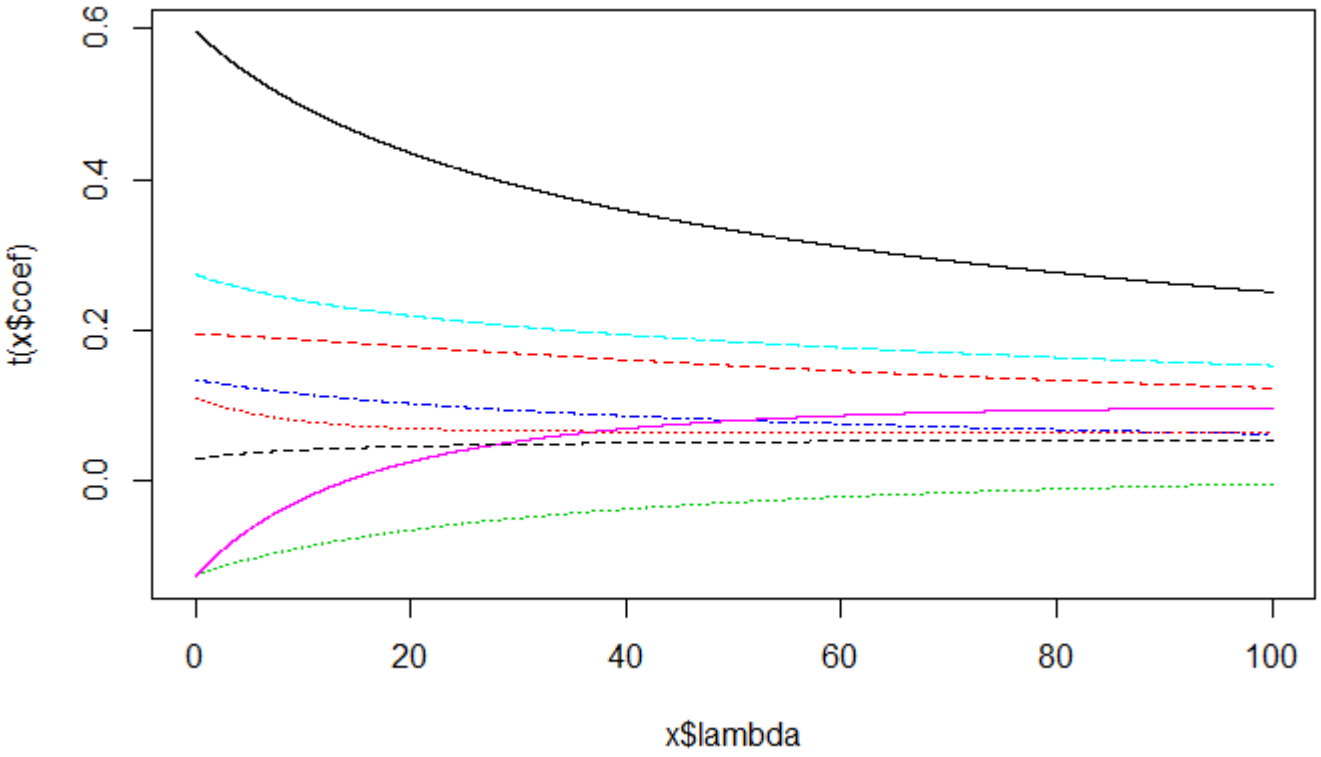
3. For the same prostate data, do the following (You may directly use any package you prefer).

a. Fit a ridge regression using GCV.
b. Perform all subsets regression and select the best model using (i) Adjusted ???? 2 and (ii) Cp.
c. Perform stepwise regression using AIC.
d. Fit a nonnegative garrote using GCV.
e. Fit lasso and select the model using leave-one-out cross validation.
f. Fit LARS and select the model using Cp.

a. let’s convert the data into dataframe and fit the ridge rigression

Hide

```
prostate.scale=data.frame(scale(prostate))
a.ridge=lm.ridge(lpsa~.,lambda=seq(0,100,.01),prostate.scale)
plot(a.ridge)
```



Hide

```
select(a.ridge)
```

```
modified HKB estimator is 4.256152
modified L-W estimator is 3.487311
smallest value of GCV  at 6.5
```

We can see how variables are converging as lambda values increases from 0 to 100

modified HKB estimator is 4.256152 modified L-W estimator is 3.487311 smallest value of GCV at 6.5 (optimal Lambda)

let's try to fit the model again with these parameters

Hide

```
a.ridge=lm.ridge(lpsa~.,lambda=6.5,prostate.scale)
coef(a.ridge)
```

```
                    lcavol       lweight           age          lbph           svi           lc
p
 4.085040e-17  5.279129e-01  1.908279e-01 -1.001433e-01  1.207332e-01  2.496503e-01 -5.154196e-0
2
      gleason         pgg45
 3.789531e-02  8.590828e-02
```

Hide

```
round(coef(a.ridge),4)
```

```
        lcavol lweight     age    lbph     svi     lcp gleason   pgg45
 0.0000  0.5279  0.1908 -0.1001  0.1207  0.2497 -0.0515  0.0379  0.0859
```

Seeing these coefficients lcavol seems to be the most important predictor

b. Let's select all the subset model and run regression on them

Hide

```
library(leaps)
b = regsubsets(lpsa~.,prostate)
rs=summary(b)
rs
```

```
Subset selection object
Call: regsubsets.formula(lpsa ~ ., prostate)
8 Variables  (and intercept)
        Forced in Forced out
lcavol      FALSE      FALSE
lweight     FALSE      FALSE
age         FALSE      FALSE
lbph        FALSE      FALSE
svi         FALSE      FALSE
lcp         FALSE      FALSE
gleason     FALSE      FALSE
pgg45       FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         lcavol lweight age lbph svi lcp gleason pgg45
1  ( 1 ) "*"    " "     " " " "  " " " " " "     " "
2  ( 1 ) "*"    "*"     " " " "  " " " " " "     " "
3  ( 1 ) "*"    "*"     " " " "  "*" " " " "     " "
4  ( 1 ) "*"    "*"     " " "*"  "*" " " " "     " "
5  ( 1 ) "*"    "*"     "*" "*"  "*" " " " "     " "
6  ( 1 ) "*"    "*"     "*" "*"  "*" " " " "     "*"
7  ( 1 ) "*"    "*"     "*" "*"  "*" "*" " "     "*"
8  ( 1 ) "*"    "*"     "*" "*"  "*" "*" "*"     "*"
```

Hide

```
names(rs)
```

```
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

Hide

```
rs$adjr2
```

```
[1] 0.5345838 0.5771246 0.6143899 0.6208036 0.6245476 0.6258707 0.6272521 0.6233681
```

the largest Adjusted R^2 is 0.6272521 which is 7th value and if we look at the which variables are included in 7th value, we see that it includeds all the variables expcept gleason.

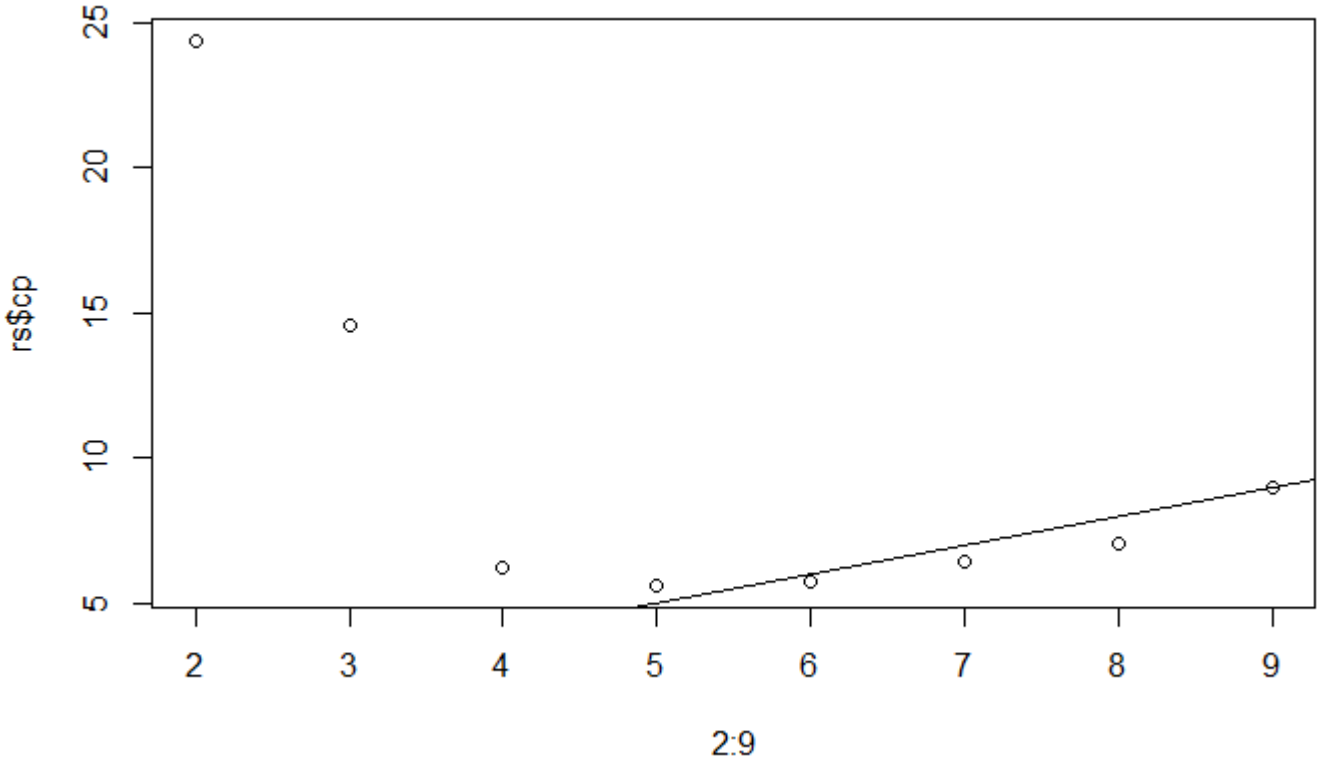
Hide

```
cbind(2:9,rs$cp)
```

```
     [,1]      [,2]
[1,]    2 24.394559
[2,]    3 14.541475
[3,]    4  6.216935
[4,]    5  5.626422
[5,]    6  5.715016
[6,]    7  6.401965
[7,]    8  7.082184
[8,]    9  9.000000
```

Hide

```
plot(2:9,rs$cp)
abline(0,1)
```



c.

Hide

```
a = lm(lpsa~.,prostate)
a.step= step(a)
```

```
Start:  AIC=-58.32
lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
    pgg45

          Df Sum of Sq    RSS     AIC
- gleason  1    0.0412 44.204 -60.231
- pgg45    1    0.5258 44.689 -59.174
- lcp      1    0.6740 44.837 -58.853
<none>                44.163 -58.322
- age      1    1.5503 45.713 -56.975
- lbph     1    1.6835 45.847 -56.693
- lweight  1    3.5861 47.749 -52.749
- svi      1    4.9355 49.099 -50.046
- lcavol   1   22.3721 66.535 -20.567

Step:  AIC=-60.23
lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45

          Df Sum of Sq    RSS     AIC
- lcp      1    0.6623 44.867 -60.789
<none>                44.204 -60.231
- pgg45    1    1.1920 45.396 -59.650
- age      1    1.5166 45.721 -58.959
- lbph     1    1.7053 45.910 -58.560
- lweight  1    3.5462 47.750 -54.746
- svi      1    4.8984 49.103 -52.037
- lcavol   1   23.5039 67.708 -20.872

Step:  AIC=-60.79
lpsa ~ lcavol + lweight + age + lbph + svi + pgg45

          Df Sum of Sq    RSS     AIC
- pgg45    1    0.6590 45.526 -61.374
<none>                44.867 -60.789
- age      1    1.2649 46.131 -60.092
- lbph     1    1.6465 46.513 -59.293
- lweight  1    3.5647 48.431 -55.373
- svi      1    4.2503 49.117 -54.009
- lcavol   1   25.4189 70.285 -19.248

Step:  AIC=-61.37
lpsa ~ lcavol + lweight + age + lbph + svi

          Df Sum of Sq    RSS     AIC
<none>                45.526 -61.374
- age      1    0.9592 46.485 -61.352
- lbph     1    1.8568 47.382 -59.497
- lweight  1    3.2251 48.751 -56.735
- svi      1    5.9517 51.477 -51.456
- lcavol   1   28.7665 74.292 -15.871
```

Hide

```
summary(a.step)
```

```
Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.83505 -0.39396  0.00414  0.46336  1.57888

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.95100    0.83175   1.143 0.255882
lcavol       0.56561    0.07459   7.583 2.77e-11 ***
lweight      0.42369    0.16687   2.539 0.012814 *
age         -0.01489    0.01075  -1.385 0.169528
lbph         0.11184    0.05805   1.927 0.057160 .
svi          0.72095    0.20902   3.449 0.000854 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 0.7073 on 91 degrees of freedom
Multiple R-squared:  0.6441,    Adjusted R-squared:  0.6245
F-statistic: 32.94 on 5 and 91 DF,  p-value: < 2.2e-16
```

The best model have AIC = -61.37

d.

Hide

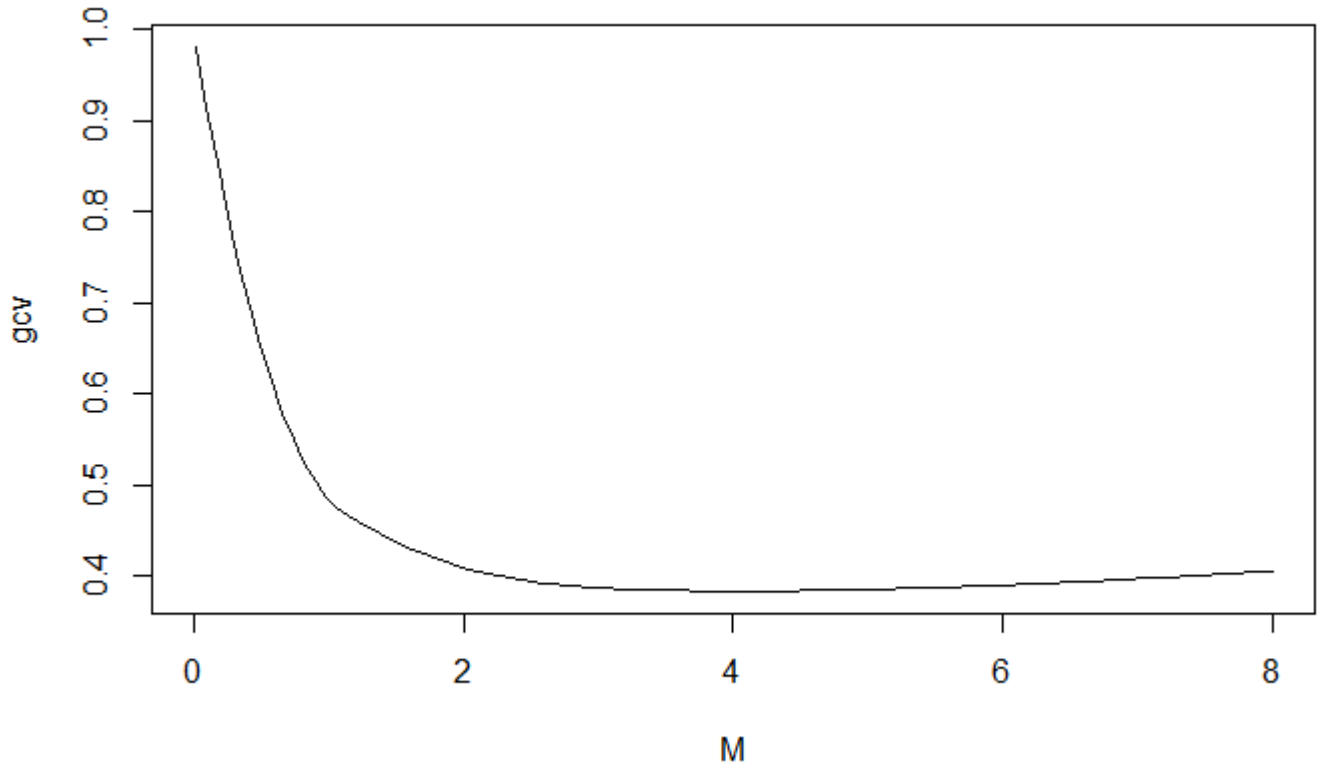
```
a=lm(lpsa~.-1,prostate.scale)
B=diag(a$coef)
Z=model.matrix(a)%*%B
D=t(Z)%*%Z
y=prostate.scale$lpsa
d=t(Z)%*%y
```

Hide

```
library(quadprog)
A=cbind(-1,diag(8))
M=seq(0.01,8,length=100)
gcv=numeric(100)
```

Hide

```
for(i in 1:100)
{
  b0=c(-M[i],rep(0,8))
  coef.nng=solve.QP(D,d,A,b0)$sol
  e=y-Z%*%coef.nng
  gcv[i]=sum(e^2)/(97*(1-M[i]/97)^2)
}
plot(M,gcv,type="l")
```



Hide

```
M=M[which.min(gcv)]
M
```

```
[1] 4.045354
```

Hide

```
b0=c(-M,rep(0,8))
coef.nng=round(solve.QP(D,d,A,b0)$sol,10)
beta.nng=B%*%coef.nng
e=y-Z%*%coef.nng
1-sum(e^2)/sum(y^2)
```

```
[1] 0.6438588
```

Hide

```
library(lars)
x = model.matrix(a)
a.lasso=lars(x,y)
summary(a.lasso)
```

```
LARS/LASSO
Call: lars(x = x, y = y)
  Df    Rss       Cp
0  1 96.000 159.8908
1  2 57.331  59.2198
2  3 48.858  38.7225
3  4 39.167  14.9926
4  5 38.833  16.1051
5  6 35.307   8.7453
6  7 34.172   7.7302
7  8 33.885   8.9690
8  9 33.144   9.0000
```

Hide

```
a.lasso$beta
```

```
     lcavol   lweight         age        lbph        svi        lcp     gleason      pgg45
0 0.0000000 0.0000000  0.00000000 0.000000000 0.00000000  0.0000000 0.000000000 0.00000000
1 0.3648277 0.0000000  0.00000000 0.000000000 0.00000000  0.0000000 0.000000000 0.00000000
2 0.4346808 0.0000000  0.00000000 0.000000000 0.06985316  0.0000000 0.000000000 0.00000000
3 0.4984110 0.1084529  0.00000000 0.000000000 0.15365575  0.0000000 0.000000000 0.00000000
4 0.5009110 0.1108315  0.00000000 0.004510448 0.15770304  0.0000000 0.000000000 0.00000000
5 0.5269756 0.1487100  0.00000000 0.064046516 0.20347485  0.0000000 0.000000000 0.03683705
6 0.5405115 0.1686722 -0.05036691 0.094832097 0.21539015  0.0000000 0.000000000 0.05889893
7 0.5445816 0.1775167 -0.07132454 0.107110293 0.22095756  0.0000000 0.007844154 0.06236310
8 0.5993774 0.1955262 -0.12665462 0.134549789 0.27477892 -0.1277618 0.028240038 0.11056623
attr(,"scaled:scale")
[1] 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959
```

Hide

```
round(cbind(a$coef,beta.nng,a.lasso$beta[7,]),4)
```

```
           [,1]    [,2]    [,3]
lcavol   0.5994  0.5659  0.5405
lweight  0.1955  0.1718  0.1687
age     -0.1267 -0.0486 -0.0504
lbph     0.1345  0.0940  0.0948
svi      0.2748  0.2259  0.2154
lcp     -0.1278  0.0000  0.0000
gleason  0.0282  0.0000  0.0000
pgg45    0.1106  0.0352  0.0589
```

e.

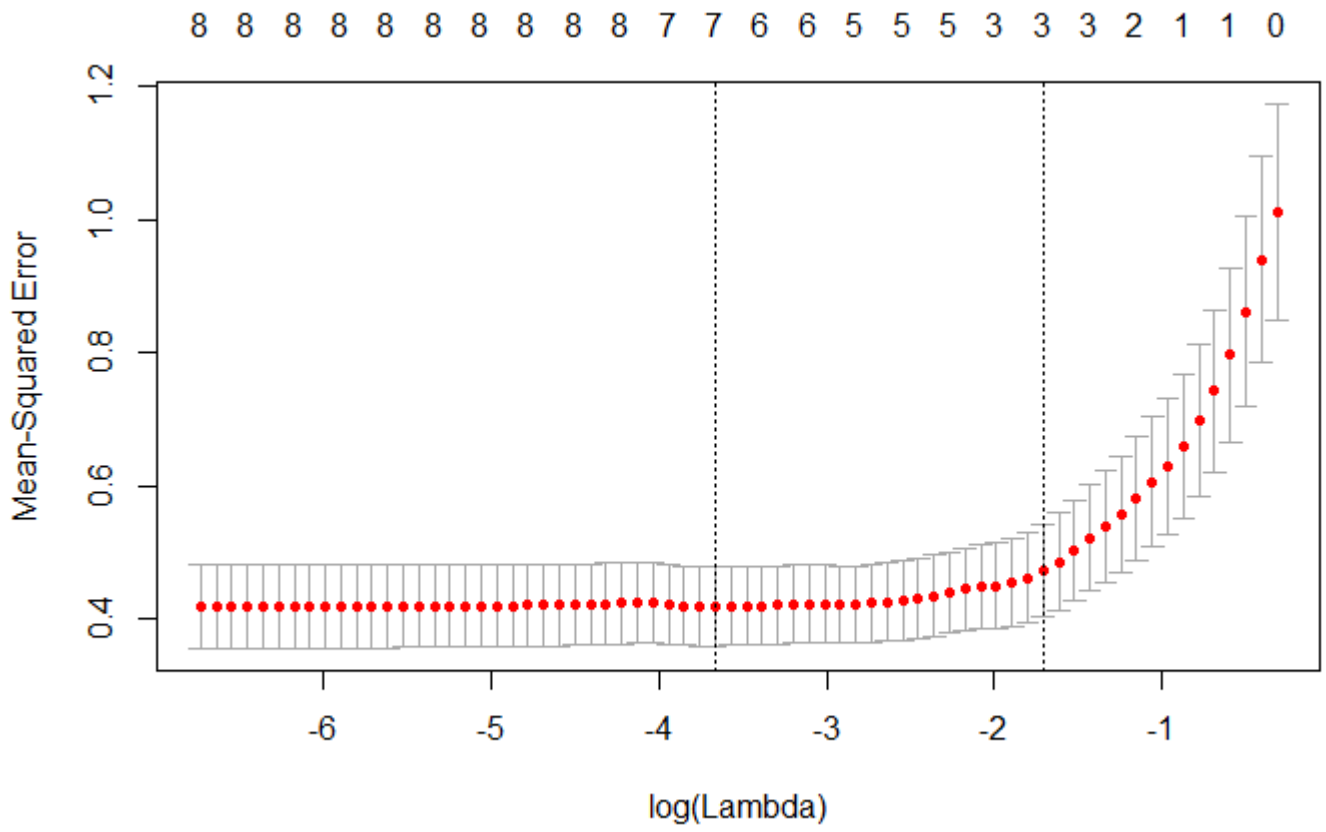
Hide

```
library(glmnet)
a.lasso=glmnet(x,y,family="gaussian")
a.cv=cv.glmnet(x,y,family="gaussian",nfolds=97)
```

```
Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per fold
```

Hide

```
plot(a.cv)
```



Hide

```
a.cv$lambda.min
```

```
[1] 0.02565504
```

Hide

```
round(coef(a.lasso,s=a.cv$lambda.min),5)
```

```
9 x 1 sparse Matrix of class "dgCMatrix"
                    1
(Intercept)  0.00000
lcavol       0.54163
lweight      0.17111
age         -0.05611
lbph         0.09819
svi          0.21690
lcp          .
gleason      0.00213
pgg45        0.05987
```

f.

Hide

```
library(lars)
y=prostate.scale$lpsa
x=as.matrix(prostate.scale[,1:8])
a.lasso=lars(x,y)
summary(a.lasso)
```

```
LARS/LASSO
Call: lars(x = x, y = y)
  Df    Rss       Cp
0  1 96.000 159.8908
1  2 57.331  59.2198
2  3 48.858  38.7225
3  4 39.167  14.9926
4  5 38.833  16.1051
5  6 35.307   8.7453
6  7 34.172   7.7302
7  8 33.885   8.9690
8  9 33.144   9.0000
```

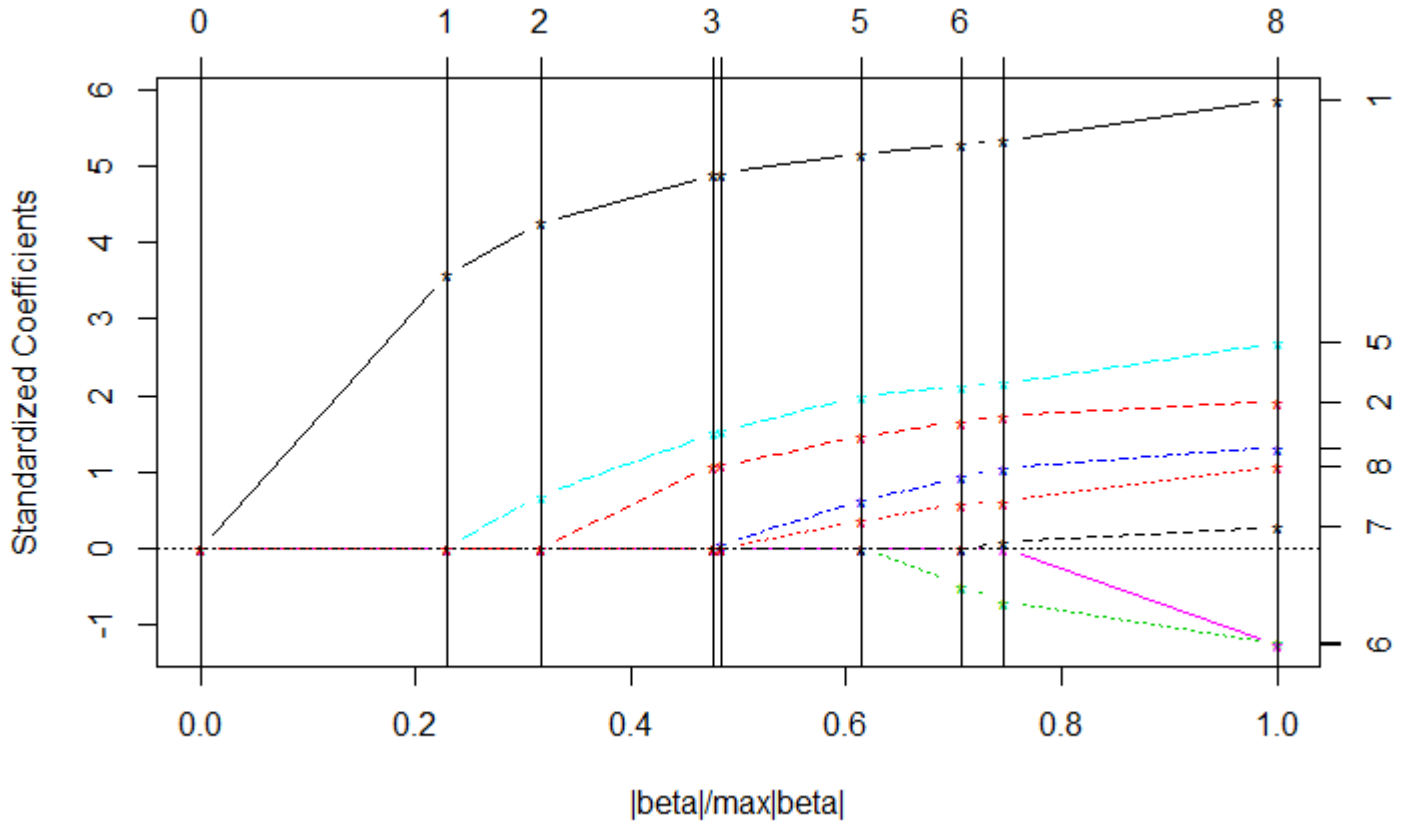
Hide

```
a.lasso$beta
```

```
     lcavol   lweight         age        lbph        svi        lcp     gleason      pgg45
0 0.0000000 0.0000000  0.00000000 0.000000000 0.00000000  0.0000000 0.000000000 0.00000000
1 0.3648277 0.0000000  0.00000000 0.000000000 0.00000000  0.0000000 0.000000000 0.00000000
2 0.4346808 0.0000000  0.00000000 0.000000000 0.06985316  0.0000000 0.000000000 0.00000000
3 0.4984110 0.1084529  0.00000000 0.000000000 0.15365575  0.0000000 0.000000000 0.00000000
4 0.5009110 0.1108315  0.00000000 0.004510448 0.15770304  0.0000000 0.000000000 0.00000000
5 0.5269756 0.1487100  0.00000000 0.064046516 0.20347485  0.0000000 0.000000000 0.03683705
6 0.5405115 0.1686722 -0.05036691 0.094832097 0.21539015  0.0000000 0.000000000 0.05889893
7 0.5445816 0.1775167 -0.07132454 0.107110293 0.22095756  0.0000000 0.007844154 0.06236310
8 0.5993774 0.1955262 -0.12665462 0.134549789 0.27477892 -0.1277618 0.028240038 0.11056623
attr(,"scaled:scale")
[1] 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959 9.797959
```

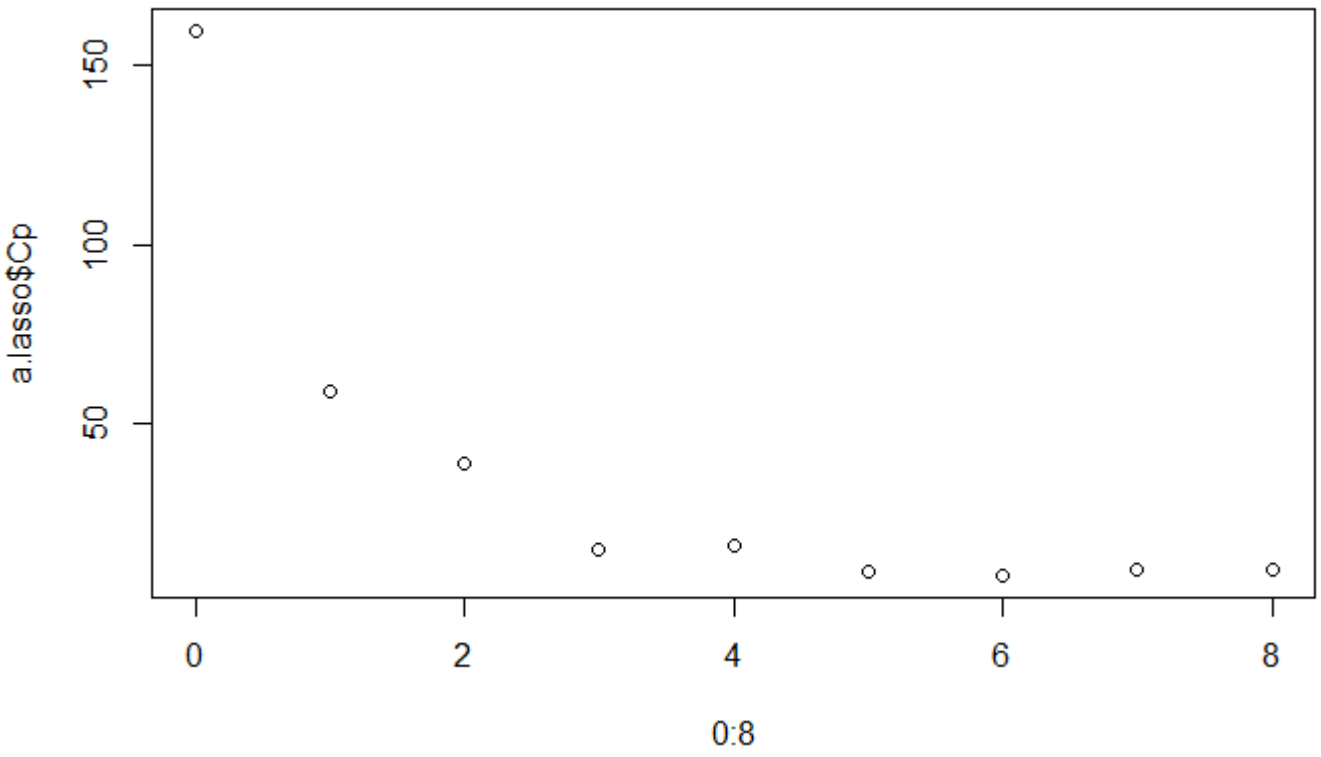
Hide

```
plot(a.lasso)
```



Hide

```
plot(0:8,a.lasso$Cp)
```



Hide

```
round(a.lasso$beta[7,],5)
```

```
  lcavol  lweight      age     lbph      svi      lcp  gleason    pgg45 
 0.54051  0.16867 -0.05037  0.09483  0.21539  0.00000  0.00000  0.05890 
```