Assignment -5

SURESH KUMAR .R

2024-10-03

```
library(ISLR)
library(MASS)
library(class)
library(boot)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

library(leaps)
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings

boston=Boston
```

11) We will now try to predict per capita crime rate in the Boston data set.

a) Try out some of the regression methods explored in this chapter,such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

   Linear model:

```
fit_Q11_lm_c6=lm(crim~.,data = boston)
summary(fit_Q11_lm_c6)

##
## Call:
## lm(formula = crim ~ ., data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
```

```
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

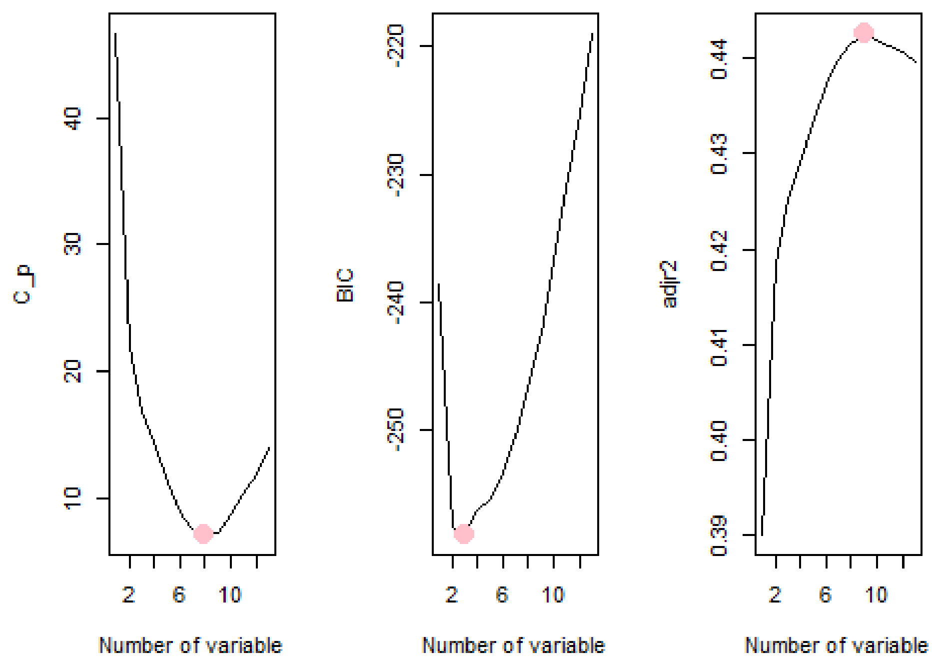
- In this model We can see that the variable zn,dis,rad,black,medv are having relationship with response.

  Subset selection - best:

```r
bs_Q11_c6=regsubsets(crim~.,data = boston,nvmax = 13)
bs_Q11_c6_summary<-summary(bs_Q11_c6)
bs_Q11_c6_summary$adjr2

##  [1] 0.3900489 0.4184935 0.4251977 0.4289661 0.4336665 0.4373321 0.4398956
##  [8] 0.4416149 0.4425053 0.4420078 0.4413928 0.4407131 0.4395838

par(mfrow=c(1,3))
plot(bs_Q11_c6_summary$cp,xlab = "Number of variable",ylab="C_p",type = "l")
points(which.min(bs_Q11_c6_summary$cp),bs_Q11_c6_summary$cp[which.min(bs_Q11_
c6_summary$cp)],col="pink",cex=3,pch=20)
plot(bs_Q11_c6_summary$bic,xlab = "Number of variable",ylab="BIC",type = "l")
points(which.min(bs_Q11_c6_summary$bic),bs_Q11_c6_summary$bic[which.min(bs_Q1
1_c6_summary$bic)],col="pink",cex=3,pch=20)
plot(bs_Q11_c6_summary$adjr2,xlab = "Number of variable",ylab="adjr2",type =
"l")
points(which.max(bs_Q11_c6_summary$adjr2),bs_Q11_c6_summary$adjr2[which.max(b
s_Q11_c6_summary$adjr2)],col="pink",cex=3,pch=20)
```




```
coef(bs_Q11_c6,which.min(bs_Q11_c6_summary$cp))
```

```
##   (Intercept)            zn           nox           dis           rad
##  19.683127801   0.043293393 -12.753707757  -0.918318253   0.532616533
##       ptratio         black         lstat          medv
##  -0.310540942  -0.007922426   0.110173124  -0.174207166
```

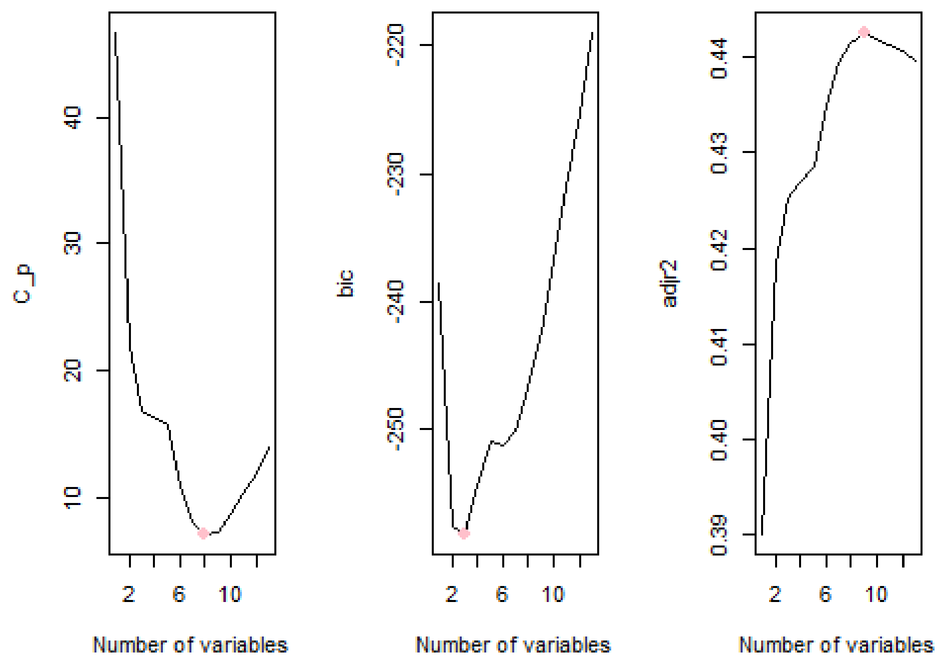
subset selection - forward:

```
fws_Q11_c6=regsubsets(crim~.,data = boston,nvmax = 13,method="forward")
fws_Q11_c6_summary<-summary(fws_Q11_c6)
fws_Q11_c6_summary$adjr2
```

```
##  [1] 0.3900489 0.4184935 0.4251977 0.4268474 0.4286801 0.4350129 0.4395053
##  [8] 0.4416149 0.4425053 0.4420078 0.4413928 0.4407131 0.4395838
```

```
par(mfrow=c(1,3))
plot(fws_Q11_c6_summary$cp,xlab = "Number of variables",ylab = "C_p",type =
"l")
points(which.min(fws_Q11_c6_summary$cp),fws_Q11_c6_summary$cp[which.min(fws_Q
11_c6_summary$cp)],col="pink",cex=2,pch=20)
plot(fws_Q11_c6_summary$bic,xlab = "Number of variables",ylab = "bic",type =
"l")
points(which.min(fws_Q11_c6_summary$bic),fws_Q11_c6_summary$bic[which.min(fws
_Q11_c6_summary$bic)],col="pink",cex=2,pch=20)
plot(fws_Q11_c6_summary$adjr2,xlab = "Number of variables",ylab =
"adjr2",type = "l")
```

```r
points(which.max(fws_Q11_c6_summary$adjr2),fws_Q11_c6_summary$adjr2[which.max
(fws_Q11_c6_summary$adjr2)],col="pink",cex=2,pch=20)
```




```r
coef(fws_Q11_c6,which.min(fws_Q11_c6_summary$cp))

##   (Intercept)            zn           nox           dis           rad
##  19.683127801   0.043293393 -12.753707757  -0.918318253   0.532616533
##       ptratio         black         lstat          medv
##  -0.310540942  -0.007922426   0.110173124  -0.174207166
```

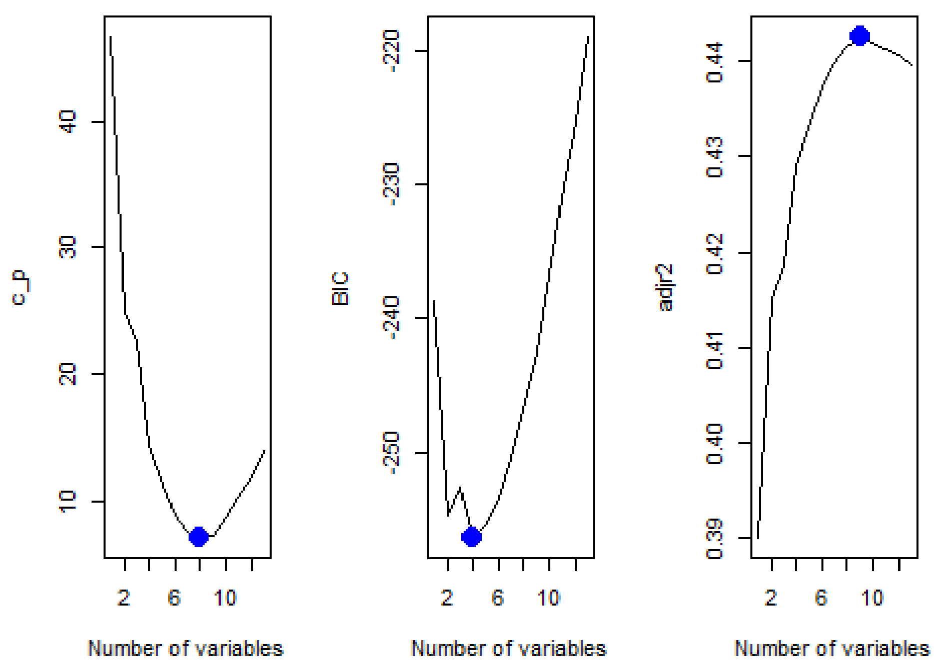
subset selection - backward:

```r
bws_Q11_c6=regsubsets(crim~.,data = boston,nvmax = 13,method="backward")
bws_Q11_c6_summary<-summary(bws_Q11_c6)
bws_Q11_c6_summary$outmat

##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " " 
## 2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   "*" 
## 3  ( 1 )  " " " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*" 
## 4  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*" 
## 5  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   " "   "*" 
## 6  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   " "   "*" 
## 7  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   " "   "*" 
## 8  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*" 
## 9  ( 1 )  "*" "*"   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*" 
## 10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*" 
## 11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*" 
```

```
## 12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"

par(mfrow=c(1,3))
plot(bws_Q11_c6_summary$cp,xlab = "Number of variables",ylab = "c_p",type =
"l")
points(which.min(bws_Q11_c6_summary$cp),bws_Q11_c6_summary$cp[which.min(bws_Q
11_c6_summary$cp)],col="blue",cex=3,pch=20)
plot(bws_Q11_c6_summary$bic,xlab = "Number of variables",ylab = "BIC",type =
"l")
points(which.min(bws_Q11_c6_summary$bic),bws_Q11_c6_summary$bic[which.min(bws
_Q11_c6_summary$bic)],col="blue",cex=3,pch=20)
plot(bws_Q11_c6_summary$adjr2,xlab = "Number of variables",ylab =
"adjr2",type = "l")
points(which.max(bws_Q11_c6_summary$adjr2),bws_Q11_c6_summary$adjr2[which.max
(bws_Q11_c6_summary$adjr2)],col="blue",cex=3,pch=20)
```




```
coef(bws_Q11_c6,which.max(bws_Q11_c6_summary$adjr2))

##   (Intercept)            zn         indus           nox           dis
##  19.124636156   0.042788127  -0.099385948 -10.466490364  -1.002597606
##           rad       ptratio         black         lstat          medv
##   0.539503547  -0.270835584  -0.008003761   0.117805932  -0.180593877
```

- the subset selection the three method give similar variable’s to use with response.

Regularization - ridge:

```r
set.seed(2)
boston_matrix_crim<-model.matrix(crim~.,data = boston)[,-1]

ridge_c6_Q11=cv.glmnet(boston_matrix_crim,boston$crim,alpha=0)
bestlam_c6_ridge<-ridge_c6_Q11$lambda.min
bestlam_c6_ridge

## [1] 0.5374992

coef(ridge_c6_Q11,s=bestlam_c6_ridge)

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  9.063048666
## zn           0.033002416
## indus       -0.082046152
## chas        -0.737684583
## nox         -5.393098508
## rm           0.335972073
## age          0.001962473
## dis         -0.702123643
## rad          0.422779055
## tax          0.003400607
## ptratio     -0.135911587
## black       -0.008483285
## lstat        0.142613436
## medv        -0.139604127
```

- The ridge method say's tax,black,age this are near to zero,so this variable's can exclude.

  Regularization - lasso:

```r
set.seed(1)
lasso_c6_Q11=cv.glmnet(boston_matrix_crim,boston$crim,alpha=1)
bestlam_c6_lasso<-lasso_c6_Q11$lambda.min
bestlam_c6_lasso

## [1] 0.05630926

coef(lasso_c6_Q11,s=bestlam_c6_lasso)

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) 12.319178096
## zn           0.035726832
## indus       -0.068876055
## chas        -0.577832639
## nox         -6.631559478
## rm           0.208676938
## age          .
## dis         -0.768388825
```

```
## rad          0.512333871
## tax          .
## ptratio     -0.179631375
## black       -0.007551172
## lstat        0.124630014
## medv        -0.154550130
```

- In the lasso method it say's age and tax is exact zero.

  Dimension Reduction - pcr:

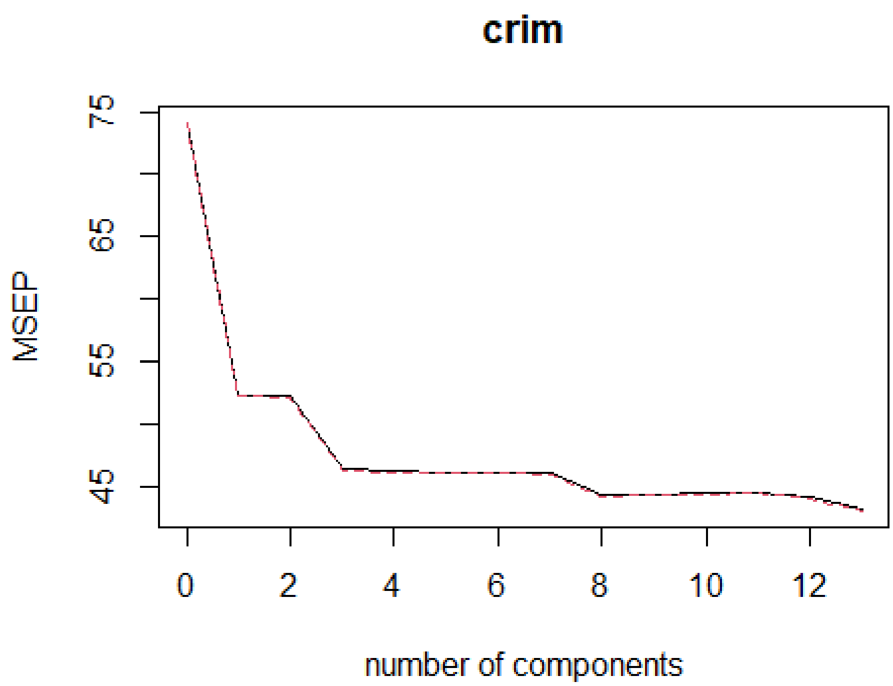
```
set.seed(2)
pcr_c6_Q11=pcr(crim~.,data=boston,scale=TRUE,validation="CV")
pcr_c6_Q11summary<- summary(pcr_c6_Q11)

## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.229    7.227    6.814    6.799    6.795    6.794
## adjCV         8.61    7.225    7.222    6.807    6.789    6.788    6.787
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.787    6.654    6.664     6.673     6.676     6.651     6.573
## adjCV    6.780    6.645    6.656     6.664     6.666     6.639     6.562
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X       47.70    60.36    69.67    76.45    82.99    88.00    91.14
93.45
## crim    30.69    30.87    39.27    39.61    39.61    39.86    40.14
42.47
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.40     97.04     98.46     99.52     100.0
## crim    42.55     42.78     43.04     44.13      45.4

pcr_c6_Q11summary

## NULL

validationplot(pcr_c6_Q11,val.type = "MSEP")
```




```
loadingspcr_c6<-pcr_c6_Q11$loadings[,1:8]
loadingspcr_c6

##                Comp 1      Comp 2      Comp 3       Comp 4       Comp 5
## zn      -0.260386431 -0.12232275 -0.38678511 -0.372729217  0.118916506
## indus    0.344970045  0.11705301  0.01606480 -0.006169698 -0.021737885
## chas    -0.002537395  0.40537659  0.20238642 -0.691013712 -0.529946773
## nox      0.337133326  0.24651528  0.02428149 -0.059281996  0.195467829
## rm      -0.212422422  0.45501084 -0.33953869  0.271293829 -0.009667228
## age      0.309700710  0.24541638  0.20333926  0.097307974  0.148147177
## dis     -0.309928466 -0.34687275 -0.16446671 -0.202636506 -0.103749280
## rad      0.303520488  0.05063099 -0.47074769  0.006961260 -0.228603393
## tax      0.327873512  0.02246586 -0.41371570 -0.020610143 -0.161224582
## ptratio  0.214014884 -0.31923649 -0.08428970  0.316836569 -0.617413009
## black   -0.197245373  0.01096616  0.43281439  0.264196294 -0.372847153
## lstat    0.320591379 -0.21252009  0.14991413 -0.230935416  0.179841594
## medv    -0.274450582  0.45649295 -0.12343134  0.178606624 -0.052157658
##               Comp 6      Comp 7      Comp 8
## zn      -0.41568749  0.31376763  0.40779707
## indus   -0.14617630 -0.28073637  0.68536670
## chas     0.16755735  0.04586120 -0.02611489
## nox     -0.19106738 -0.09130262  0.06443069
## rm       0.13455802  0.43665640  0.07685682
## age     -0.03372481  0.59530163  0.01637673
## dis     -0.02283442  0.10080440 -0.03153353
## rad     -0.19077101 -0.05510114 -0.45621119
## tax     -0.27651588 -0.11403113 -0.10347135
```

```
## ptratio  0.27380093  0.24947434  0.29757016
## black   -0.72392947  0.07846025 -0.08540903
## lstat   -0.08494270  0.38780273 -0.19278149
## medv    -0.04877358 -0.14919632 -0.01134994
```

- In pcr model say's that taking 3 or 8 component are best

  Dimension Reduction - pls:

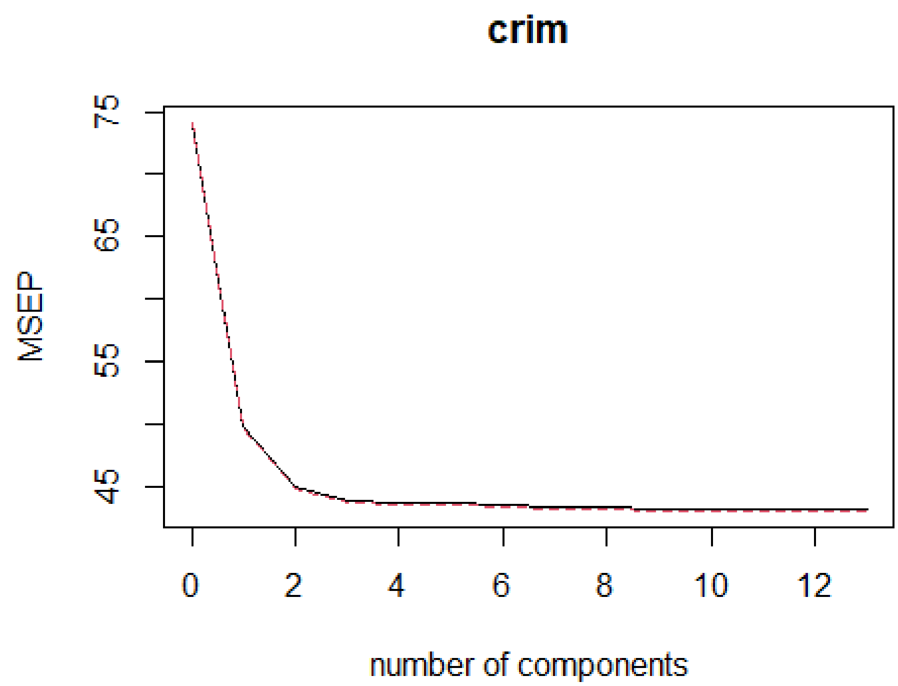
```
set.seed(2)
plsr_c6_Q11=plsr(crim~.,data=boston,scale=TRUE,validation="CV")
plsr_c6_Q11summary<- summary(plsr_c6_Q11)

## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: kernelpls
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.054    6.702    6.621    6.607    6.605    6.597
## adjCV         8.61    7.049    6.695    6.611    6.597    6.593    6.583
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.586    6.578    6.572     6.573     6.573     6.573     6.573
## adjCV    6.573    6.566    6.560     6.561     6.562     6.562     6.562
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X       47.27    56.79    61.38    71.13    76.41    79.78    83.99
86.27
## crim    34.32    41.81    44.03    44.58    44.94    45.24    45.33
45.38
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X        88.5     91.32     96.56     98.26     100.0
## crim     45.4     45.40     45.40     45.40      45.4

plsr_c6_Q11summary

## NULL

validationplot(plsr_c6_Q11,val.type = "MSEP")
```




```
loadingsplsr_c6_Q11=plsr_c6_Q11$loadings[,1:2]
loadingsplsr_c6_Q11

##               Comp 1      Comp 2
## zn      -0.24335054  0.53815945
## indus    0.34770483 -0.17781631
## chas    -0.01277319 -0.20332787
## nox      0.34017136 -0.15637797
## rm      -0.20731864  0.25770067
## age      0.30404305 -0.32036861
## dis     -0.30411964  0.32217571
## rad      0.33246649  0.42195276
## tax      0.35402888  0.34100696
## ptratio  0.22217831  0.01782896
## black   -0.22006673 -0.34948686
## lstat    0.32535380 -0.10839180
## medv    -0.28015730  0.06836246
```

- In plsr model say's that taking 2 component are best

b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, crossvalidation, or some other reasonable alternative, as opposed to using training error.

```
tr_bos=sample(nrow(boston),nrow(boston)*0.70)

tr_Q11b_bos=boston[tr_bos,]
```

```
te_Q11b_bos=boston[-tr_bos,]
```

Regularization - ridge:

```
set.seed(3)
tr_bos_matrix=model.matrix(crim~.,data = tr_Q11b_bos)[,-1]
te_bos_matrix=model.matrix(crim~.,data = te_Q11b_bos)[,-1]

Q_11b_ridge=cv.glmnet(tr_bos_matrix,tr_Q11b_bos$crim,alpha=0)
bestlam_Q11b_ridge=Q_11b_ridge$lambda.min

pred_Q11b_ridge=predict(Q_11b_ridge,s=bestlam_Q11b_ridge,newx =
te_bos_matrix)
test_error_Q11br=mean((te_Q11b_bos$crim- pred_Q11b_ridge)^2)

rmse_Q11b_ridge= sqrt(test_error_Q11br)
(rmse_Q11b_ridge/mean(te_Q11b_bos$crim))*100

## [1] 185.3637
```

Regularization - lasso:

```
Q_11b_lasso=cv.glmnet(tr_bos_matrix,tr_Q11b_bos$crim,alpha=1)
bestlam_Q11b_lasso=Q_11b_lasso$lambda.min

pred_Q11b_lasso=predict(Q_11b_lasso,s=bestlam_Q11b_lasso,newx =
te_bos_matrix)
test_error_Q11bl=mean((te_Q11b_bos$crim - pred_Q11b_lasso)^2)

rmse_Q11b_lasso= sqrt(test_error_Q11bl)
(rmse_Q11b_lasso/mean(te_Q11b_bos$crim))*100

## [1] 185.1713
```

c) Does your chosen model involve all of the features in the data set? Why or why not?

- The chosen model does not involve all of the features in the data set, because some not statistically significant to response.