Module 5 - Variable selection and regularization

Pawel Chilinski

December 3, 2013

Exercise 1.

File cigconsumption.txt contains data related to cigarettes sale per one person in 51 states of USA (variable Sales) and other variables such as:
Age - median of age of state population
HS - percentage of population having at least secondary education
Income - mean income per one person in a given state
Female - percentage of women in state population
State - name of state
Black - percentage of black people in state population
Price - weighted mean price of packet of cigarettes

```
> cig.data <- read.table(file="cigconsumption.txt",header=T)
> library(psych)
```

```
> cig.data.scaled<-scale(cig.data[,2:8])
> stripchart(data.frame(cig.data.scaled),vertical=T,method="jitter")
> for(row in which(abs(cig.data.scaled)>3)){
+         state<-row %% nrow(cig.data)
+         if(state==0){
+                 state=nrow(cig.data)
+         }
+         text(row%/%nrow(cig.data)+1.3,cig.data.scaled[row],cig.data$State[state])
+ }
```

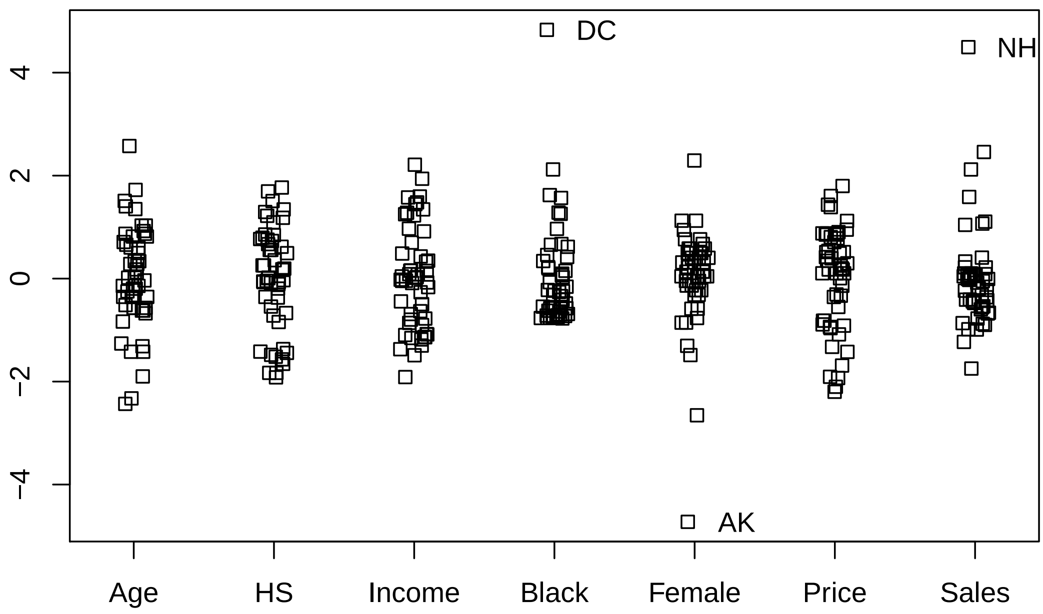

Figure 1: Scaled cig data to check its distribution and find potential outliers.

The data seems to have several observations that have unusual values (DC, AK, NH).

- Fit a linear regression model taking Sales as a response variable and the rest of variables in the data set as explanatory variables (excluding State).

```
> cig.lm<-lm(Sales~Age+HS+Income+Black+Female+Price,cig.data)
```

- Check diagnostic plots of the model.

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(cig.lm, which=1:4,labels.id=cig.data$State,id.n=8)
> par(op)
```

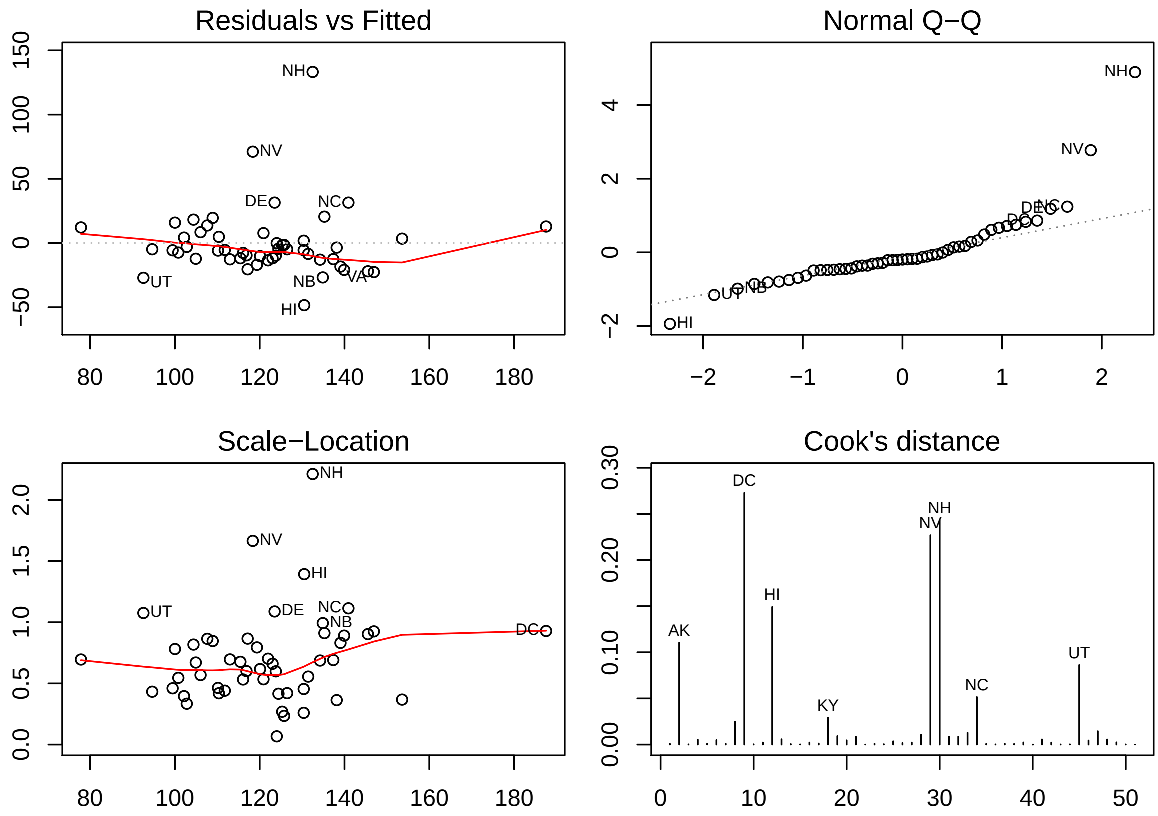

Figure 2: Diagnostic plots for cigconsumption model.

The Residuals vs Fitted plot shows that we have several values which seem extreme and should be further investigated (points labelled on the graph). The same conclusion can be made after looking at Normal Q-Q plot where we see most of the data is located approximately along ideal line but several points sticking out. Cooks distance plot shows influential observations (8 biggest).

- Find outliers in the data. Check if these observations are influential. If so exclude them from further analysis.

```
> par(mar=c(2,2,2,2))
> library(MASS)
> cig.lm.studres <- studres(cig.lm)
> plot(fitted(cig.lm),cig.lm.studres,ylab="studentized residual",xlab="fitted")
> cig.possible.outliers<-which(abs(cig.lm.studres)>2)
> points(fitted(cig.lm)[cig.possible.outliers],cig.lm.studres[cig.possible.outliers],col="red",cex=2)
> text(fitted(cig.lm)[cig.possible.outliers]+5,cig.lm.studres[cig.possible.outliers],
+                   labels=cig.data$State[cig.possible.outliers])
```

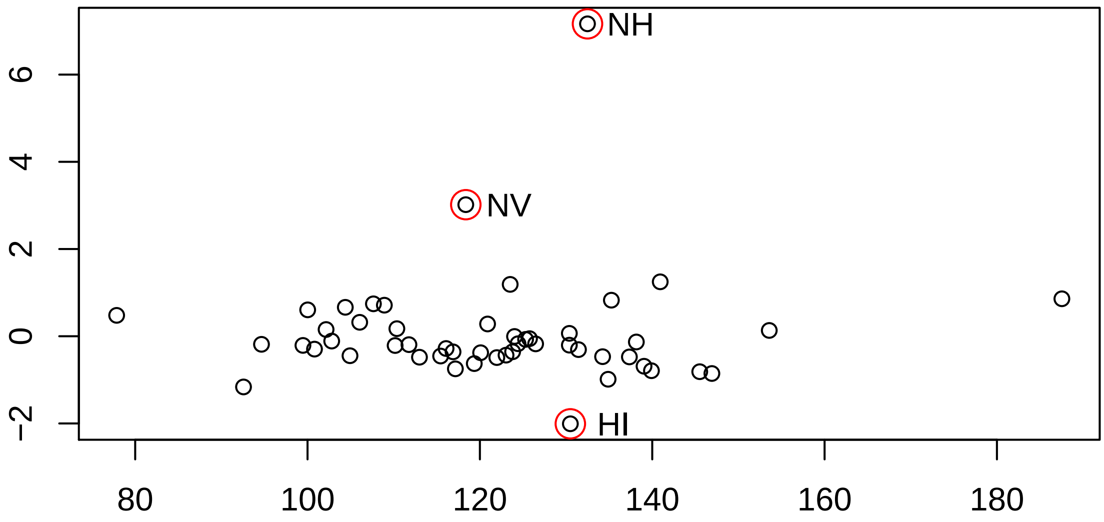

Figure 3: Finding outliers as studentized residual values greater then 2.

We can see that NH, NV and HI are outliers in the sense that they do not follow fitted linear model.

Find potential influential observations using leverages:

```
> h<-lm.influence(cig.lm)$hat
> names(h)<-cig.data$State
> rev(sort(h))
```

```
        DC         AK         UT         FL         KY         MS         HI
0.71971284 0.58016035 0.31057370 0.29848078 0.22927171 0.22026829 0.21690346
        NM         OR         NC         CT         NV         SC         AL
0.20516348 0.19482461 0.18961222 0.17467143 0.17115641 0.15738783 0.14883450
        LA         WV         AR         CO         NY         VA         NJ
0.14721698 0.13941373 0.13709423 0.13013894 0.12521079 0.12137818 0.11254359
        MA         PA         DE         RI         GA         ND         ID
0.11176850 0.11084828 0.10996426 0.10772871 0.09924626 0.09779421 0.09206081
        IN         TN         WY         MD         MI         IL         MT
0.08630311 0.08610596 0.08455731 0.08243815 0.07886005 0.07813892 0.07535750
        OK         TX         NB         SD         NH         VT         WA
0.07473124 0.07320930 0.07130620 0.07026368 0.06634506 0.06464328 0.06441387
        CA         IO         KA         MN         ME         MO         AZ
0.06010622 0.06008342 0.06005316 0.05954249 0.05922771 0.05381267 0.04969854
        OH         WI
0.04270237 0.03867070
```

```
> #observation potentially influential if hii>=2p/n
> 2*7/51
```

```
[1] 0.2745098
```

So based on leverages the DC, AK, UT, FL are potentially influential.They are among observations selected by Cook's distance but not FL.

```
> library(car)
> avPlots(cig.lm,labels=cig.data$State,id.n=3)
```

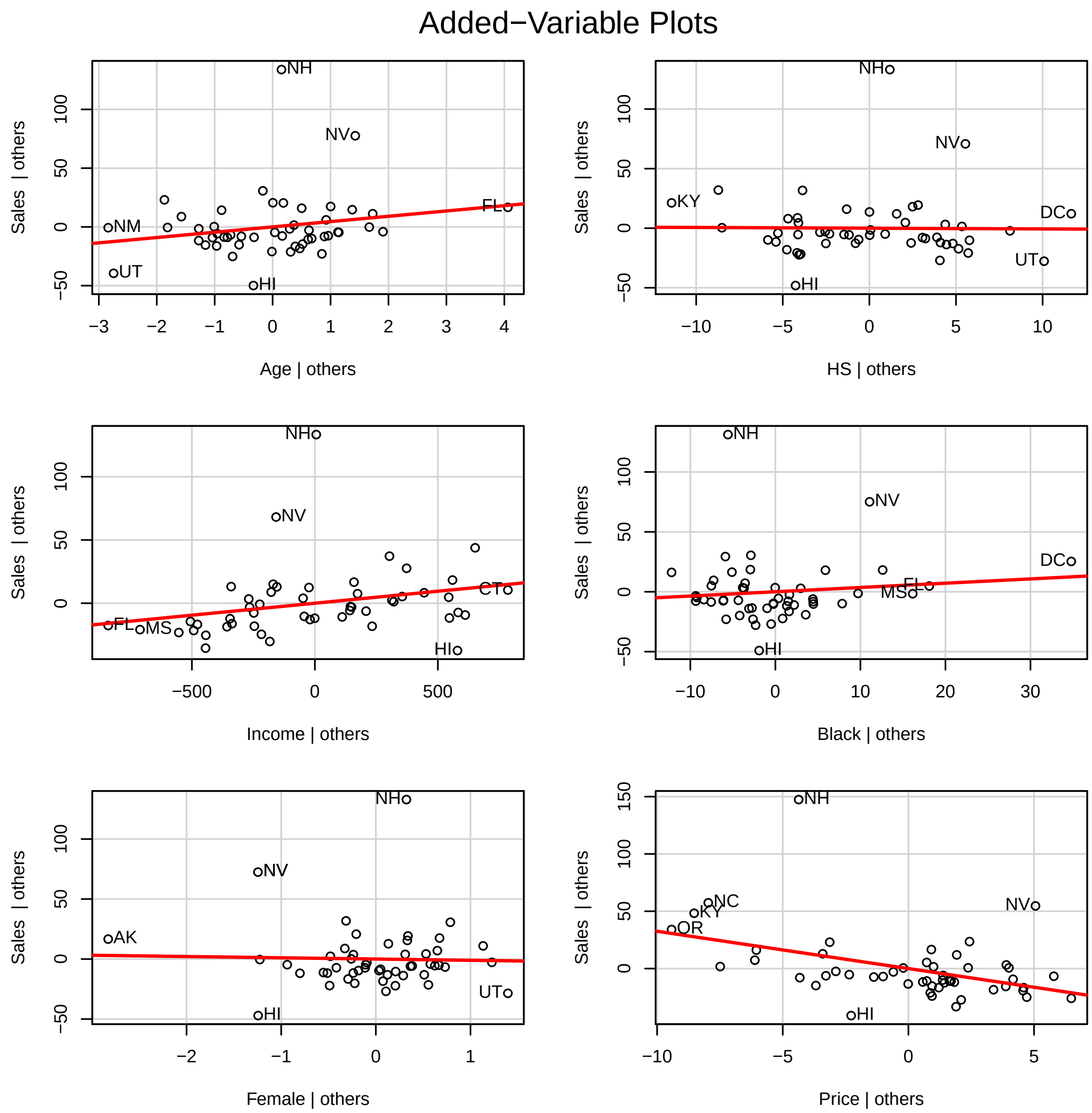

Figure 4: Partial regression plots

Based on partial regression plots checking if following states contain influential data: NH, NV, FL, UT, HI, KY, DC, AK, OR by creating models without given state and comparing how the model changed compared to the original model. The matrix coef.diffs contains in the first column full model's coefficients and in the subsequent columns the differences between model's coefficients without given observation and full model's coefficients.

```
> coef.diffs<-matrix(nrow = 7, ncol = 11)
> coef.diffs[,1]<-summary(cig.lm)$coef[,1]
> outliers<-c("NH", "NV", "FL", "UT", "HI", "KY", "DC", "AK", "OR","NC")
> colnames(coef.diffs)<-c("fullmodel",outliers)
> for(i in 1:length(outliers)){
+         coef.diffs[,i+1]<-summary(lm(Sales~Age+HS+Income+Black+Female+Price,
+                 cig.data[cig.data$State!=outliers[i],]))$coef[,1] - summary(cig.lm)$coef[,1]
+ }
> coef.diffs

        fullmodel            NH            NV            FL            UT
[1,] 103.34484573 70.2340702654 -1.305609e+02  1.8537126655 -81.776286331
[2,]   4.52045242 -0.2850920478 -1.599255e+00  0.1288885711  -1.407821257
[3,]  -0.06158605 -0.1407627081 -3.977661e-01  0.0164947849   0.331150985
[4,]   0.01894645 -0.0001044041  1.780343e-03 -0.0002684276  -0.002294837
[5,]   0.35753517  0.2380064501 -2.843040e-01  0.0131559056  -0.027310645
[6,]  -1.05285886 -1.7957044595  4.165352e+00 -0.1160504024   2.133956990
[7,]  -3.25491843  0.8348858721 -5.815813e-01  0.0149396639   0.099721400
                HI            KY            DC            AK            OR
```

```
[1,] 162.650274061 -32.632152642 48.336595474 -1.669167e+02 -1.143783e+00
[2,]  -0.267352910  -0.001625182 -0.600584515 -2.133083e-01 -9.580361e-02
[3,]  -0.220270884   0.254483269 -0.447222885 -1.155285e-04 -1.555252e-02
[4,]   0.004710896  -0.001306524  0.000144514 -6.797964e-04  1.252272e-04
[5,]  -0.035260032   0.097278452 -0.477544075 -1.227148e-01  3.045607e-05
[6,]  -2.994683056   0.215443681 -0.236999006  3.583204e+00  3.935379e-02
[7,]  -0.188683426   0.304585478  0.192575908 -1.743692e-01  5.362626e-02
                NC
[1,] -52.745681268
[2,]   0.085451010
[3,]   0.283190486
[4,]  -0.001548372
[5,]   0.033476945
[6,]   0.476847313
[7,]   0.414531475
```

Based on coef.diffs table we can conclude that NH, NV, UT, HI, KY, DC, AK, NC are influential observations (which is the same result as from the Cooks' distances graph) and we remove them from the model (i). As we can see not all influential observation were noticed on the studentized residual plot.

```
> cig.data.wo.influentials<-cig.data[!cig.data$State %in%
+                                   c("NH", "NV", "UT", "HI", "KY", "DC", "AK", "NC"),]
> cig.lm.wo.influentials<-lm(Sales~Age+HS+Income+Black+Female+Price,cig.data.wo.influentials)
```

- How did removing outliers influence values of R^2 and residual standard error ($\hat{\sigma}$)?

  As we can see after removing influential observations the R^2 increased by 0.21 (original model's value 0.32) and $\hat{\sigma}$ decreased by 17.12 (original model's value 28.17).

```
> cig.lm.sum<-summary(cig.lm)
> cig.lm.wo.influentials.sum<-summary(cig.lm.wo.influentials)
> cig.lm.wo.influentials.sum$sigma-cig.lm.sum$sigma

[1] -17.1259

> cig.lm.wo.influentials.sum$r.squared-cig.lm.sum$r.squared

[1] 0.2176803
```

- In a refitted model: do we reject hypothesis that all variables are insignificant? Yes we can reject this hypothesis because Income and Price variables are significant.

```
> summary(cig.lm.wo.influentials)

Call:
lm(formula = Sales ~ Age + HS + Income + Black + Female + Price,
    data = cig.data.wo.influentials)

Residuals:
    Min      1Q  Median      3Q     Max
-17.361  -5.790  -2.845   3.304  33.321

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.124e+02  2.064e+02  -1.029 0.310514
Age          2.842e-01  1.435e+00   0.198 0.844116
HS          -2.202e-01  4.883e-01  -0.451 0.654714
Income       1.657e-02  4.533e-03   3.656 0.000811 ***
Black       -2.451e-01  3.438e-01  -0.713 0.480399
Female       6.953e+00  4.502e+00   1.544 0.131234
Price       -2.173e+00  5.315e-01  -4.088 0.000233 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.05 on 36 degrees of freedom
Multiple R-squared:  0.5385,       Adjusted R-squared:  0.4616
F-statistic: 7.002 on 6 and 36 DF,  p-value: 5.431e-05
```

- Which predictors are insignificant for explaining Sales if all the other predictors are incorporated in the model?

  We can see looking at p-values that predictors Age, HS, Black, Female are insignificant for explaining Sales if all the other predictors are incorporated in the model.

- Using F test for comparing two nested models decide whether all of these variables can be removed from the model (use function anova()).

  From F test we can see that model with all predictors explains statistically significant more of the variability in the data than constant model (p-value < 0.05):

```
> cig.contstant.lm.wo.influentials<-lm(Sales~1,cig.data.wo.influentials)
> anova(cig.lm.wo.influentials,cig.contstant.lm.wo.influentials)
```

```
Analysis of Variance Table

Model 1: Sales ~ Age + HS + Income + Black + Female + Price
Model 2: Sales ~ 1
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     36 4394.1
2     42 9521.9 -6   -5127.8 7.0017 5.431e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Choose the best subset of predictors using stepwise procedures:

  – based o t-tests (backward elimination)
  Assuming p-to-remove as 0.05. First removing Age because it has the biggest p-value > 0.05 (0.84)

```
> wo.age.sum<-summary(update(cig.lm.wo.influentials,.~. - Age))
> wo.age.sum$coef[,4, drop=F]
```

```
                Pr(>|t|)
(Intercept) 0.2336028191
HS          0.6093499737
Income      0.0003730126
Black       0.3523429619
Female      0.0569389037
Price       0.0001923349
```

```
> wo.age.sum$adj.r.squared
```

```
[1] 0.4755904
```

  Adjusted R-squared increased. Next removing HS as it has the biggest p-value > 0.05 (0.6)

```
> wo.hs.sum<-summary(update(cig.lm.wo.influentials,.~. - Age-HS))
> wo.hs.sum$coef[,4, drop=F]
```

```
                Pr(>|t|)
(Intercept) 1.393462e-01
Income      5.044412e-05
Black       4.184212e-01
Female      3.690906e-02
Price       1.468671e-04
```

```
> wo.hs.sum$adj.r.squared
```

```
[1] 0.485725
```

  Adjusted R-squared increased. Next removing Black as it has the biggest p-value > 0.05 (0.42)

```
> wo.black.sum<-summary(update(cig.lm.wo.influentials,.~. - Age-HS-Black))
> wo.black.sum$coef[,4, drop=F]
```

```
                Pr(>|t|)
(Intercept) 2.083788e-01
Income      4.479931e-06
Female      4.884278e-02
Price       1.690691e-04
```

```
> wo.black.sum$adj.r.squared
```

```
[1] 0.4900867
```

  Adjusted R-squared increased and all predictors are significant. But checking what happens when we remove one more predictor which is on the limit of significance i.e. Female (0.049)

```
> wo.female.sum<-summary(update(cig.lm.wo.influentials,.~. - Age-HS-Black-Female))
> wo.female.sum$coef[,4, drop=F]
```

```
                Pr(>|t|)
(Intercept) 9.591939e-07
Income      2.424749e-06
Price       1.116641e-03
```

```
> wo.female.sum$adj.r.squared
[1] 0.4501189
```

Now Adjusted R-Squared decreased. The model achieved using backward elimination is Sales $\sim$ Income+Female+Price with Adjusted R-Squared 0.49.

– based on AIC criterion (use function step())

```
> step(lm(Sales~1,cig.data.wo.influentials), direction = c("forward"), k=2,
+                   scope=list(upper=.~.+Age + HS + Income + Black + Female + Price))

Start:  AIC=234.21
Sales ~ 1

         Df Sum of Sq    RSS    AIC
+ Income  1   2997.17 6524.7 219.95
+ HS      1    994.18 8527.7 231.47
+ Price   1    772.05 8749.9 232.57
+ Age     1    635.87 8886.0 233.24
<none>                9521.9 234.21
+ Female  1    252.93 9269.0 235.05
+ Black   1    213.73 9308.2 235.23

Step:  AIC=219.95
Sales ~ Income

         Df Sum of Sq    RSS    AIC
+ Price   1   1538.16 4986.6 210.39
<none>                6524.7 219.95
+ Female  1     15.33 6509.4 221.85
+ HS      1     15.25 6509.5 221.85
+ Age     1      6.65 6518.1 221.91
+ Black   1      0.03 6524.7 221.95

Step:  AIC=210.39
Sales ~ Income + Price

         Df Sum of Sq    RSS    AIC
+ Female  1    478.05 4508.5 208.06
<none>                4986.6 210.39
+ Age     1    223.02 4763.6 210.43
+ HS      1     89.25 4897.3 211.62
+ Black   1     10.66 4975.9 212.30

Step:  AIC=208.06
Sales ~ Income + Price + Female

        Df Sum of Sq    RSS    AIC
<none>               4508.5 208.06
+ Black  1    78.027 4430.5 209.31
+ Age    1    51.835 4456.7 209.56
+ HS     1     4.126 4504.4 210.02

Call:
lm(formula = Sales ~ Income + Price + Female, data = cig.data.wo.influentials)

Coefficients:
(Intercept)       Income        Price       Female
 -194.71971      0.01656     -2.03028      6.38502
```

So we can see the forward stepwise procedure based on the AIC criterion selects the same predictors as backward procedure based on t-tests i.e. Income, Price, Female. Running step function for backward and both directions gives the same results.

– based on BIC criterion (use function step())

```
> step(lm(Sales~1,cig.data.wo.influentials), direction = c("forward"),
+                   k=log(length(cig.data.wo.influentials)),
+                   scope=list(upper=.~.+Age + HS + Income + Black + Female + Price))

Start:  AIC=234.29
Sales ~ 1
```

```
         Df Sum of Sq    RSS    AIC
+ Income  1   2997.17 6524.7 220.11
+ HS      1    994.18 8527.7 231.62
+ Price   1    772.05 8749.9 232.73
+ Age     1    635.87 8886.0 233.39
<none>                9521.9 234.29
+ Female  1    252.93 9269.0 235.21
+ Black   1    213.73 9308.2 235.39

Step:  AIC=220.11
Sales ~ Income

         Df Sum of Sq    RSS    AIC
+ Price   1   1538.16 4986.6 210.63
<none>                6524.7 220.11
+ Female  1     15.33 6509.4 222.09
+ HS      1     15.25 6509.5 222.09
+ Age     1      6.65 6518.1 222.15
+ Black   1      0.03 6524.7 222.19

Step:  AIC=210.63
Sales ~ Income + Price

         Df Sum of Sq    RSS    AIC
+ Female  1    478.05 4508.5 208.38
<none>                4986.6 210.63
+ Age     1    223.02 4763.6 210.74
+ HS      1     89.25 4897.3 211.93
+ Black   1     10.66 4975.9 212.62

Step:  AIC=208.38
Sales ~ Income + Price + Female

        Df Sum of Sq    RSS    AIC
<none>               4508.5 208.38
+ Black  1    78.027 4430.5 209.71
+ Age    1    51.835 4456.7 209.96
+ HS     1     4.126 4504.4 210.42

Call:
lm(formula = Sales ~ Income + Price + Female, data = cig.data.wo.influentials)

Coefficients:
(Intercept)       Income        Price       Female
 -194.71971      0.01656     -2.03028      6.38502
```

So we can see the forward stepwise procedure based on the BIC criterion selects the same predictors as backward procedure based on t-tests and step procedure based on AIC criterion i.e. Income, Price, Female. Running step function for backward and both directions gives the same results.

– based on Adjusted R^2 criterion (use function regsubsets() from library leaps and then summary() of a returned object which contains components such as adjr2, bic, cp).

```
> library(leaps)
> subsets<-regsubsets(Sales~Age + HS + Income + Black + Female + Price,cig.data.wo.influentials)
> (regsubsets.sum<-summary(subsets))
Subset selection object
Call: regsubsets.formula(Sales ~ Age + HS + Income + Black + Female +
    Price, cig.data.wo.influentials)
6 Variables  (and intercept)
       Forced in Forced out
Age        FALSE      FALSE
HS         FALSE      FALSE
Income     FALSE      FALSE
Black      FALSE      FALSE
Female     FALSE      FALSE
Price      FALSE      FALSE
1 subsets of each size up to 6
```

```
Selection Algorithm: exhaustive
         Age HS  Income Black Female Price
1  ( 1 ) " " " " "*"    " "   " "    " "
2  ( 1 ) " " " " "*"    " "   " "    "*"
3  ( 1 ) " " " " "*"    " "   "*"    "*"
4  ( 1 ) " " " " "*"    "*"   "*"    "*"
5  ( 1 ) " " "*" "*"    "*"   "*"    "*"
6  ( 1 ) "*" "*" "*"    "*"   "*"    "*"
```

```
> regsubsets.sum$adjr2
```

```
[1] 0.2980521 0.4501189 0.4900867 0.4857250 0.4755904 0.4616101
```

```
> regsubsets.sum$bic
```

```
[1]  -8.731324 -16.530664 -17.102969 -14.092462 -10.638855  -6.924483
```

```
> regsubsets.sum$cp
```

```
[1] 14.455430  3.853743  1.937203  3.297950  5.039226  7.000000
```

We can also visualise selection of the best model based on different criterion:

```
> par(mar=c(1,1,1,1))
> par(mfrow=c(2,2))
> plot(subsets, scale="bic")
> plot(subsets, scale="r2")
> plot(subsets, scale="adjr2")
> plot (subsets, scale="Cp")
```

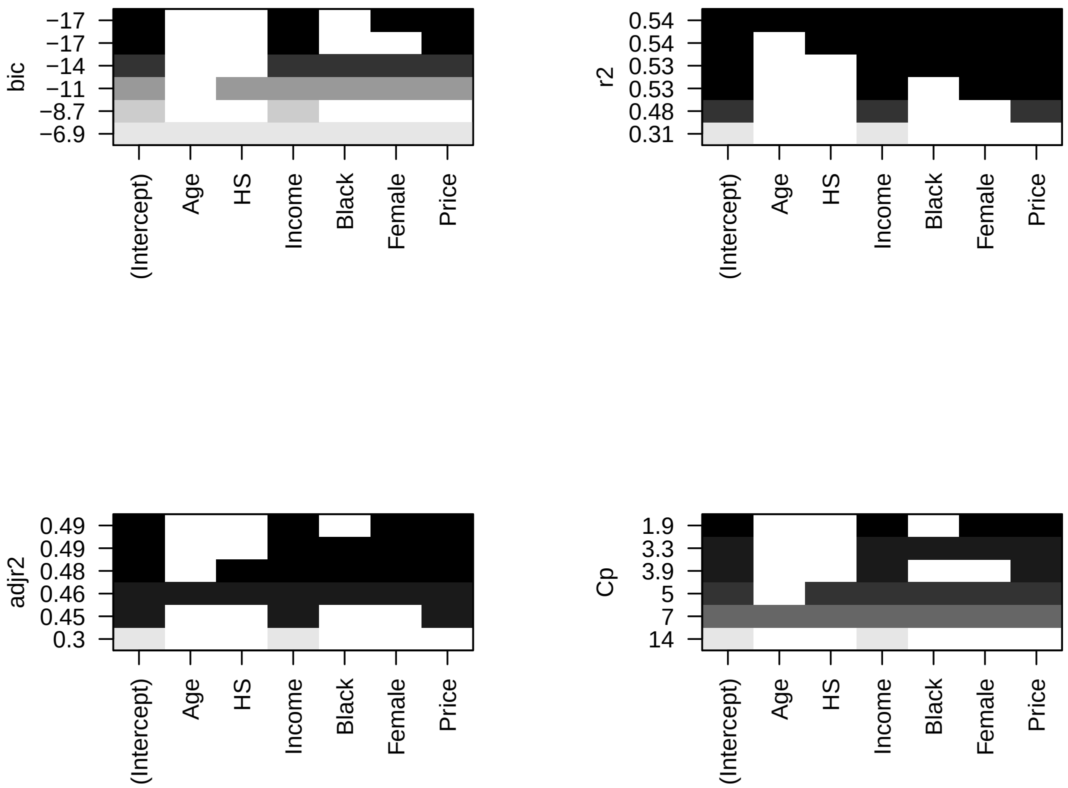

Figure 5: Finding best model using exhaustive search and different criteria.

As we can see again the best model is Sales $\sim$ Income+Female+Price (the only exception is R^2 which doesn't take into consideration number of predictors in the model and always prefers maximal model).

Checking C_p with respect to number of parameters:

```
> par(mar=c(2,2,1,1))
> plot(2:7,regsubsets.sum$cp,xlab="No. of Parameters", ylab="Cp Statistic")
> abline(0,1)
```

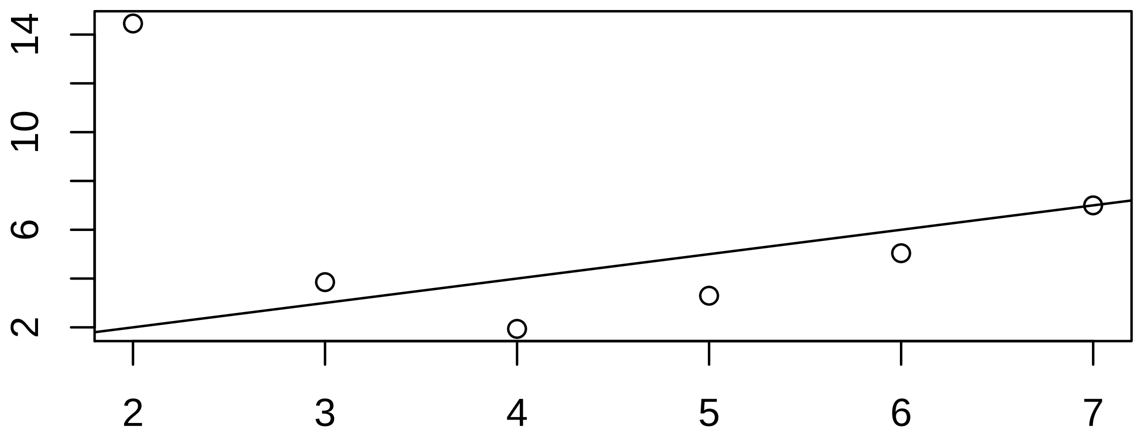

Figure 6: C_p against number of model parameters.

We can see that point for 3 predictors (i.e. 4 parameter model) lies below line p so the model fits data well.

- Compare the chosen model with initial one using F test.

```
> anova(lm(Sales~Income+Female+Price,cig.data.wo.influentials),cig.lm.wo.influentials)

Analysis of Variance Table

Model 1: Sales ~ Income + Female + Price
Model 2: Sales ~ Age + HS + Income + Black + Female + Price
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     39 4508.5
2     36 4394.1  3    114.39 0.3124 0.8163
```

F tests shows that there isn't significant difference between original model and sub-model selected (p-value 0.82).

- How did the standard errors of estimated coefficients (bi) change after removing insignificant predictors?

  Showing how much in percentage terms the standard errors of estimated coefficients (bi) changed after removing insignificant predictors (standard error decreased):

```
> std.err.selected.model<-
+                  summary(lm(Sales~Income+Female+Price,cig.data.wo.influentials))$coef[,2,drop=F]
> std.err.full.model<-
+                  cig.lm.wo.influentials.sum$coef[c("(Intercept)","Income","Female","Price"),2,drop=F]
> 100*(std.err.selected.model-std.err.full.model)/std.err.full.model

            Std. Error
(Intercept) -26.265214
Income      -31.383381
Female      -30.256458
Price        -8.190541
```

Exercise 2.

For the data uscrime.txt

```
> #Loading the data and checking if it looks valid
> crime.data <- read.table(file="uscrime.txt",header=T)
```

```
> crime.data.scaled<-scale(crime.data)
> stripchart(data.frame(crime.data.scaled),vertical=T,method="jitter")
> for(row in which(abs(crime.data.scaled)>3)){
+         obs<-row %% nrow(crime.data)
+         if(obs==0){
+                 obs=nrow(crime.data)
+         }
+         text(row%/%nrow(crime.data)+1.3,crime.data.scaled[row],obs)
+ }
```

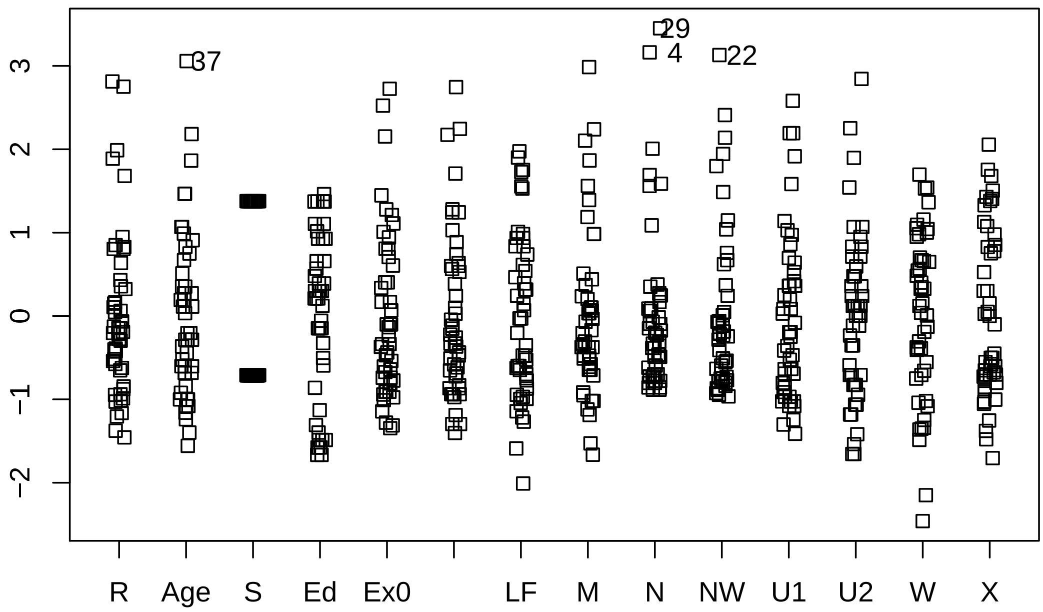

Figure 7: Scaled crime data to check its distribution and find potential outliers.

The data seems to have several observations that have unusual values (like observation 29).

- Fit a linear regression model taking crime rate as a response variable and all the other variables in the set as predictors.

```
> crime.lm<-lm(R~.,crime.data)
> (crime.lm.sum<-summary(crime.lm))

Call:
lm(formula = R ~ ., data = crime.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-34.884 -11.923  -1.135  13.495  50.560 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.918e+02  1.559e+02  -4.438 9.56e-05 ***
Age          1.040e+00  4.227e-01   2.460  0.01931 *  
S           -8.308e+00  1.491e+01  -0.557  0.58117    
Ed           1.802e+00  6.496e-01   2.773  0.00906 ** 
Ex0          1.608e+00  1.059e+00   1.519  0.13836    
Ex1         -6.673e-01  1.149e+00  -0.581  0.56529    
LF          -4.103e-02  1.535e-01  -0.267  0.79087    
M            1.648e-01  2.099e-01   0.785  0.43806    
N           -4.128e-02  1.295e-01  -0.319  0.75196    
NW           7.175e-03  6.387e-02   0.112  0.91124    
U1          -6.017e-01  4.372e-01  -1.376  0.17798    
U2           1.792e+00  8.561e-01   2.093  0.04407 *  
```

```
W             1.374e-01  1.058e-01   1.298  0.20332
X             7.929e-01  2.351e-01   3.373  0.00191 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.94 on 33 degrees of freedom
Multiple R-squared:  0.7692,	Adjusted R-squared:  0.6783
F-statistic: 8.462 on 13 and 33 DF,  p-value: 3.686e-07
```

- Remove all the variables which are redundant in the model. Use methods based on: t-tests, AIC, BIC, Adjusted R2, Mallows Cp criterion (use function regsubsets()). Before selecting the best subset of predictors remove outliers from the data.
  - Remove outliers

    Checking diagnostics plots for the model:

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(crime.lm, which=1:4)
```

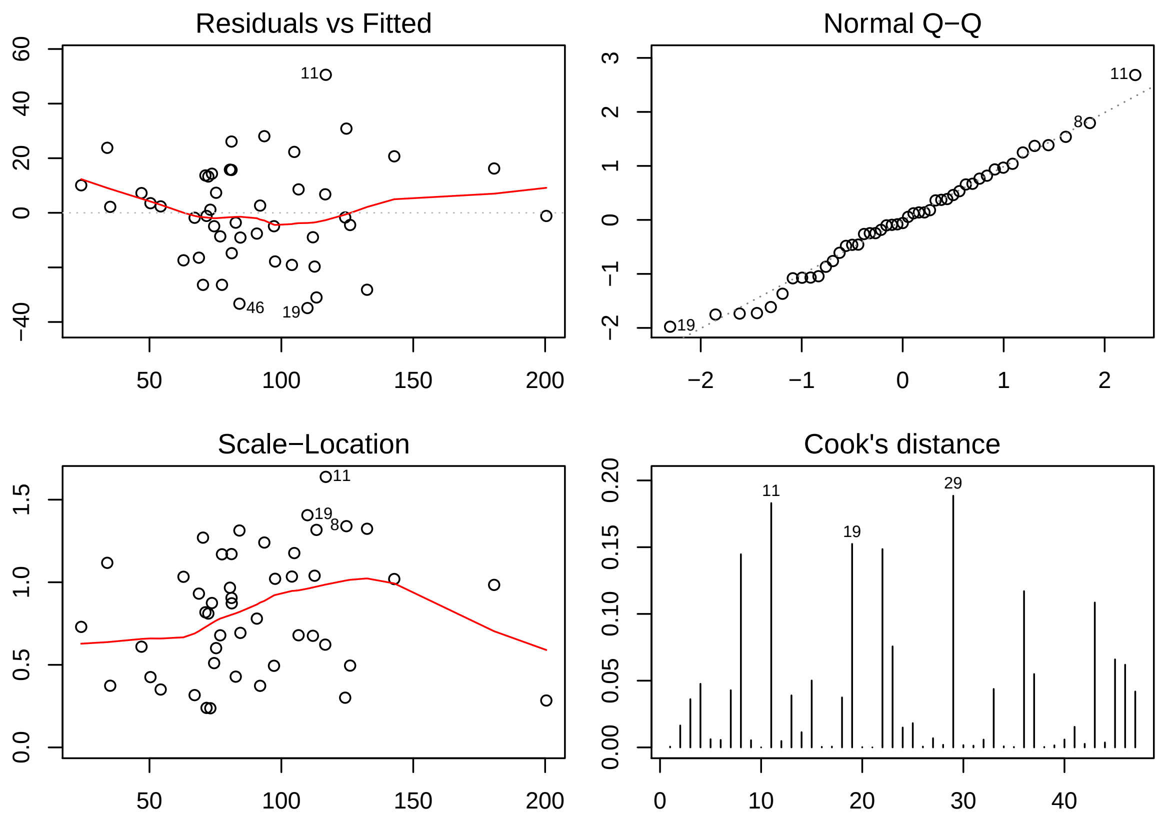

Figure 8: Diagnostic plots for uscrime model.

From the diagnostics plots we can suspect several outliers (11,19). The Q-Q plot without them looks reasonably normal. There are no significant Cook's values taking into consideration standard criteria (like $D_i > 1$ or $D_i > 4/n$ or $D_i > F_{p,n-p,1-\alpha}$)

Checking studentized residuals:

```
> par(mar=c(2,2,2,2))
> crime.lm.studres <- studres(crime.lm)
> plot(fitted(crime.lm),crime.lm.studres,ylab="studentized residual",xlab="fitted")
> crime.possible.outliers<-which(abs(crime.lm.studres)>2)
> points(fitted(crime.lm)[crime.possible.outliers],crime.lm.studres[crime.possible.outliers],col="red",cex=2)
> text(fitted(crime.lm)[crime.possible.outliers]+5,crime.lm.studres[crime.possible.outliers],
+                  labels=crime.possible.outliers)
```

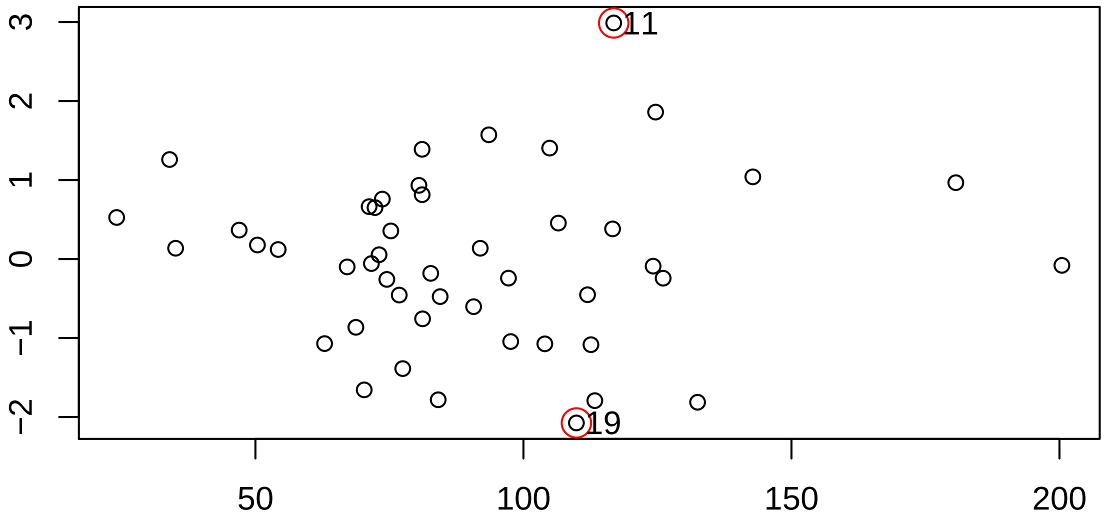

Figure 9: Finding outliers as studentized residual values greater then 2.

Here we can see two outliers (11,19).

Find potential influential observations using leverages:

```
> h<-lm.influence(crime.lm)$hat
> rev(sort(h))
```

```
       37        26        31        29        36        45        22         4
0.6752527 0.5745475 0.4888892 0.4622320 0.4607684 0.4476376 0.4441164 0.4158059
        7        15         8         5        19        47        43        35
0.4068346 0.3924670 0.3861394 0.3653295 0.3531691 0.3385852 0.3355858 0.3231805
       28        18        23        40        20         6        41        11
0.3141828 0.3099174 0.3091039 0.2805522 0.2758356 0.2709126 0.2699547 0.2624131
       32        30        27        25        33         3        16        24
0.2615685 0.2608636 0.2545720 0.2529633 0.2460662 0.2449432 0.2422287 0.2371708
       38        13        46        14        42        17        44         1
0.2364060 0.2260657 0.2257011 0.2162277 0.2142820 0.2123171 0.2004907 0.2002714
       10         2        34        21         9        39        12
0.1854205 0.1756965 0.1614667 0.1598966 0.1490065 0.1426685 0.1302939
```

```
> #observation potentially influential if hii>=2p/n
> 2*15/47
```

```
[1] 0.6382979
```

So based on leverages only 37th observation is potentially influential.But it is not selected by Cook's distance so not selecting it as influential.

Checking partial regression plots:

```
> avPlots(crime.lm,id.n=3,layout = c(4,4))
```

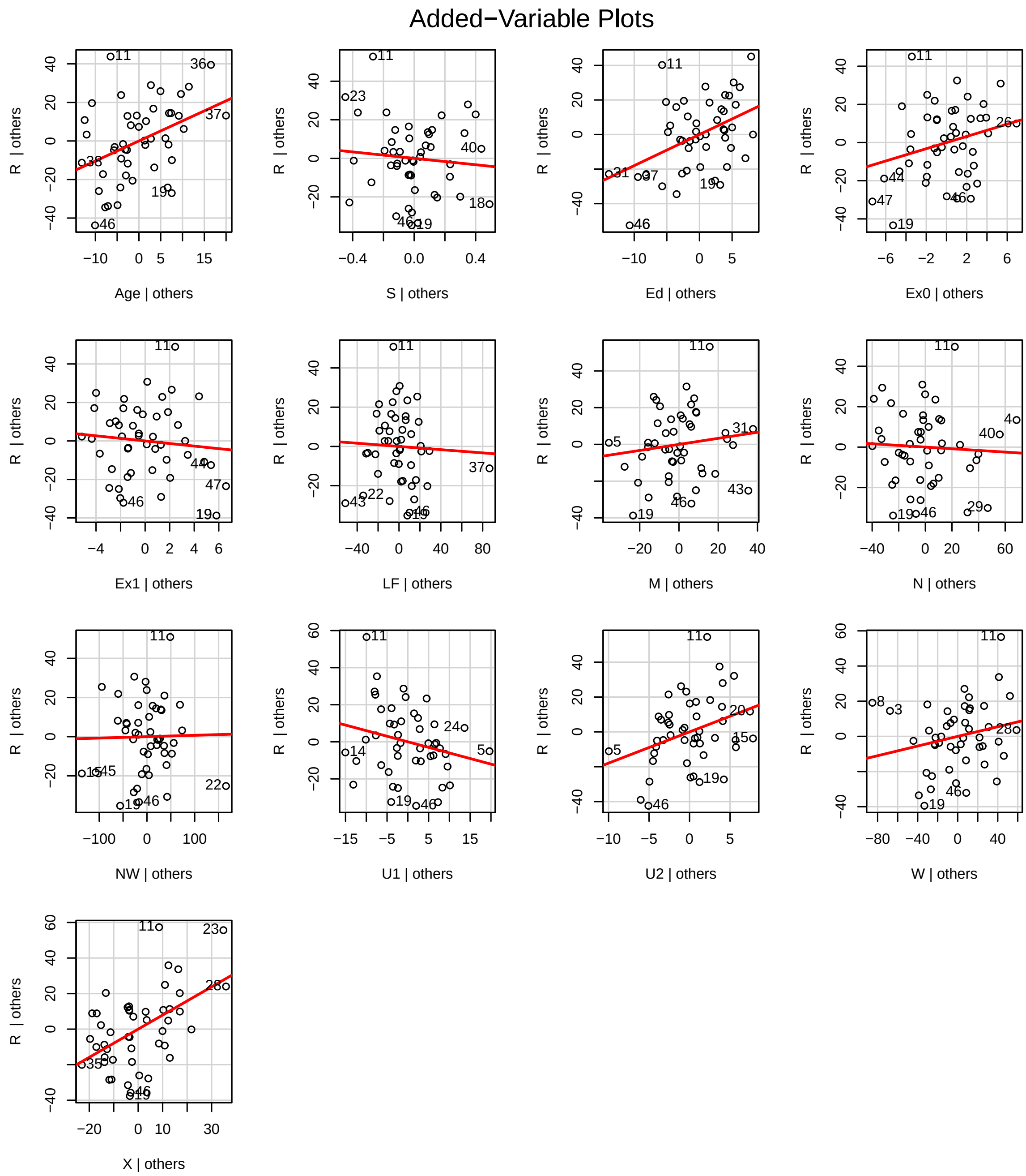

Figure 10: Partial regression plots

Partial regression also shows that 11 and 19 should be considered as outliers and possible influential observations. It looks that observations that should be removed are 11th and 19th. There aren't any significant influential observations.

```
> crime.data.wo.outliers<-crime.data[c(-11,-19),]
> crime.lm.wo.outliers<-lm(R~.,crime.data.wo.outliers)
```

As we can see after removing outliers the R^2 increased by 0.06 (original model's value 0.77) and $\hat{\sigma}$ decreased by 3.27 (original model's value 21.94):

```
> crime.lm.wo.outliers.sum<-summary(crime.lm.wo.outliers)
> crime.lm.wo.outliers.sum$sigma-crime.lm.sum$sigma

[1] -3.268544

> crime.lm.wo.outliers.sum$r.squared-crime.lm.sum$r.squared

[1] 0.05812826
```

– Remove all the variables which are redundant in the model.

  * based o t-tests (backward elimination)
    Assuming p-to-remove as 0.05.
    First removing LF because it has the biggest p-value > 0.05 (0.97)

```
> wo.lf.sum<-summary(update(crime.lm.wo.outliers,.~.-LF))
> pvals<-wo.lf.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
S           0.9780585032
M           0.8011651109
Ex1         0.7642091137
W           0.7314518128
NW          0.3989095390
U1          0.2344074194
N           0.2282096537
Ex0         0.1136465556
U2          0.0148180484
X           0.0009289970
Age         0.0007082232
Ed          0.0001676218
(Intercept) 0.0001573826
> wo.lf.sum$adj.r.squared
[1] 0.7626179
```

Adjusted R-squared increased. Next removing S as it has the biggest p-value > 0.05 (0.98)

```
> wo.s.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S))
> pvals<-wo.s.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
M           0.7989943954
Ex1         0.7548551067
W           0.7275612921
NW          0.3478392104
N           0.2176254296
U1          0.2162949363
Ex0         0.1027389690
U2          0.0114735781
Age         0.0005607046
X           0.0004645654
(Intercept) 0.0001237186
Ed          0.0001179338
> wo.s.sum$adj.r.squared
[1] 0.7698058
```

Adjusted R-squared increased. Next removing M as it has the biggest p-value > 0.05 (0.8)

```
> wo.m.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M))
> pvals<-wo.m.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
Ex1         7.659592e-01
W           7.381264e-01
NW          3.576877e-01
N           1.963689e-01
U1          1.325837e-01
Ex0         1.017705e-01
U2          8.293024e-03
Age         3.512363e-04
X           2.575996e-04
Ed          4.122633e-05
(Intercept) 3.737725e-07
> wo.m.sum$adj.r.squared
[1] 0.77613
```

Adjusted R-squared increased. Next removing Ex1 as it has the biggest p-value > 0.05 (0.76)

```
> wo.ex1.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M-Ex1))
> pvals<-wo.ex1.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
W           7.439300e-01
NW          3.005727e-01
```

```
N           1.922979e-01
U1          1.203475e-01
U2          6.540050e-03
Age         2.476081e-04
X           1.696265e-04
Ed          3.321276e-05
(Intercept) 2.287863e-07
Ex0         1.293448e-07
```

```
> wo.ex1.sum$adj.r.squared
[1] 0.7819504
```

Adjusted R-squared increased. Next removing W as it has the biggest p-value > 0.05 (0.74)

```
> wo.w.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M-Ex1-W))
> pvals<-wo.w.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
NW          2.518392e-01
N           1.922609e-01
U1          9.760294e-02
U2          3.822120e-03
Age         2.101679e-04
X           1.460949e-05
Ed          9.670832e-06
(Intercept) 1.972578e-08
Ex0         2.328076e-09
> wo.w.sum$adj.r.squared
[1] 0.7873508
```

Adjusted R-squared increased. Next removing NW as it has the biggest p-value > 0.05 (0.25)

```
> wo.nw.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M-Ex1-W-NW))
> pvals<-wo.nw.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
N           1.805597e-01
U1          1.240109e-01
U2          5.138834e-03
Age         3.333833e-04
X           1.867062e-05
Ed          2.754107e-06
(Intercept) 2.495305e-08
Ex0         2.021596e-09
> wo.nw.sum$adj.r.squared
[1] 0.785303
```

Adjusted R-squared unfortunately decreased but still there are insignificant predictors. Next removing N as it has the biggest p-value > 0.05 (0.18)

```
> wo.n.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M-Ex1-W-NW-N))
> pvals<-wo.n.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
U1          1.670811e-01
U2          7.967382e-03
Age         1.477655e-04
X           3.307777e-05
Ed          1.406441e-06
(Intercept) 1.616428e-08
Ex0         9.393496e-10
> wo.n.sum$adj.r.squared
[1] 0.780429
```

Adjusted R-squared unfortunately decreased but still there are insignificant predictors. Next removing U1 as it has the biggest p-value > 0.05 (0.17)

```
> wo.u1.sum<-summary(update(crime.lm.wo.outliers,.~.-LF-S-M-Ex1-W-NW-N-U1))
> pvals<-wo.u1.sum$coef[,4, drop=F]
> pvals[sort(pvals,index.return=T,decreasing=T)$ix,,drop=F]
                Pr(>|t|)
U2          6.398139e-03
```

```
Age         1.873055e-04
X           2.091141e-05
Ed          2.466289e-06
(Intercept) 1.907496e-08
Ex0         4.003605e-12
> wo.u1.sum$adj.r.squared
[1] 0.774888
```

Now all the predictors are significant so we are done with shrinking the model: R $\sim$ Age+Ed+Ex0+U2+X

* based on AIC criterion (use function step())

```
> step(lm(R~1,crime.data.wo.outliers), direction = c("forward"), k=2,
+                  scope=list(upper=.~.+Age+S+Ed+Ex0+Ex1+M+N+NW+U1+U2+W+X+LF))
Start:  AIC=327.68
R ~ 1

       Df Sum of Sq   RSS    AIC
+ Ex0   1   31223.3 31349 298.58
+ Ex1   1   29883.0 32690 300.47
+ W     1   11204.5 51368 320.81
+ Ed    1    7765.0 54808 323.72
+ N     1    5223.8 57349 325.76
+ M     1    3781.6 58791 326.88
<none>              62573 327.68
+ U2    1    2036.8 60536 328.19
+ X     1    1971.3 60601 328.24
+ LF    1    1862.6 60710 328.32
+ S     1     297.8 62275 329.47
+ Age   1     138.2 62435 329.58
+ NW    1      42.0 62531 329.65
+ U1    1      12.4 62560 329.68

Step:  AIC=298.58
R ~ Ex0

       Df Sum of Sq   RSS    AIC
+ Age   1    6967.9 24382 289.27
+ X     1    6281.6 25068 290.52
+ W     1    2323.5 29026 297.12
+ S     1    2225.4 29124 297.27
+ NW    1    1820.5 29529 297.89
+ M     1    1751.7 29598 298.00
<none>              31350 298.58
+ Ex1   1     627.2 30722 299.67
+ N     1     382.6 30967 300.03
+ LF    1     376.1 30973 300.04
+ U2    1     140.6 31209 300.38
+ U1    1      43.6 31306 300.52
+ Ed    1      16.8 31333 300.56

Step:  AIC=289.27
R ~ Ex0 + Age

       Df Sum of Sq   RSS    AIC
+ X     1   1952.67 22429 287.51
+ M     1   1915.55 22466 287.59
+ Ed    1   1617.56 22764 288.18
<none>              24382 289.27
+ LF    1    840.09 23542 289.69
+ U2    1    750.97 23631 289.86
+ U1    1    403.16 23978 290.52
+ N     1    377.51 24004 290.57
+ Ex1   1    255.40 24126 290.80
+ NW    1     57.80 24324 291.17
+ W     1     47.33 24334 291.18
+ S     1     47.25 24334 291.18

Step:  AIC=287.51
```

```
R ~ Ex0 + Age + X

       Df Sum of Sq   RSS    AIC
+ Ed    1    7285.0 15144 271.84
+ M     1    3170.9 19258 282.66
+ NW    1    1863.5 20565 285.61
+ W     1    1773.6 20655 285.81
+ LF    1    1697.3 20732 285.97
+ N     1    1399.7 21029 286.62
<none>              22429 287.51
+ S     1     626.4 21802 288.24
+ U1    1     368.2 22061 288.77
+ U2    1     218.3 22211 289.07
+ Ex1   1      52.0 22377 289.41

Step:  AIC=271.84
R ~ Ex0 + Age + X + Ed

       Df Sum of Sq   RSS    AIC
+ U2    1   2658.64 12485 265.15
+ U1    1    828.31 14316 271.31
<none>              15144 271.84
+ W     1    412.97 14731 272.60
+ N     1    350.30 14794 272.79
+ NW    1    290.29 14854 272.97
+ M     1    232.00 14912 273.15
+ Ex1   1    203.80 14940 273.23
+ LF    1    133.53 15010 273.44
+ S     1      5.66 15138 273.82

Step:  AIC=265.15
R ~ Ex0 + Age + X + Ed + U2

       Df Sum of Sq   RSS    AIC
+ U1    1    619.57 11866 264.86
<none>              12485 265.15
+ N     1    431.90 12053 265.57
+ NW    1    326.65 12159 265.96
+ W     1    162.93 12322 266.56
+ Ex1   1    117.91 12367 266.73
+ LF    1     19.35 12466 267.08
+ S     1      3.39 12482 267.14
+ M     1      0.39 12485 267.15

Step:  AIC=264.86
R ~ Ex0 + Age + X + Ed + U2 + U1

       Df Sum of Sq   RSS    AIC
+ N     1    568.71 11297 264.65
<none>              11866 264.86
+ NW    1    444.87 11421 265.14
+ M     1    159.98 11706 266.25
+ Ex1   1     92.84 11773 266.51
+ W     1     78.60 11787 266.56
+ S     1     36.87 11829 266.72
+ LF    1      6.71 11859 266.84

Step:  AIC=264.65
R ~ Ex0 + Age + X + Ed + U2 + U1 + N

       Df Sum of Sq   RSS    AIC
<none>              11297 264.65
+ NW    1    410.16 10887 264.99
+ W     1    101.41 11196 266.25
+ Ex1   1     96.83 11200 266.27
+ S     1     73.09 11224 266.36
```

```
+ LF    1      1.71 11295 266.65
+ M     1      1.65 11295 266.65

Call:
lm(formula = R ~ Ex0 + Age + X + Ed + U2 + U1 + N, data = crime.data.wo.outliers)

Coefficients:
(Intercept)          Ex0          Age            X           Ed           U2
  -562.3465       1.1980       1.1836       0.6071       2.3661       1.9204
         U1            N
    -0.4378      -0.1224
```

So we can see the forward stepwise procedure based on the AIC criterion doesn't remove predictors which removal causes to deteriorate criterion measuring the quality of the model (the method based on t-statistics removed some predictors even it meant that adjusted R^2 went down) and produced optimal model: R $\sim$ Ex0 + Age + X + Ed + U2 + U1 + N. Running step function for backward and both directions gives the same results.

* based on BIC criterion (use function step())

```
> step(lm(R~1,crime.data.wo.outliers), direction = c("forward"),
+                  k=log(length(crime.data.wo.outliers)),
+                  scope=list(upper=.~.+Age+S+Ed+Ex0+Ex1+M+N+NW+U1+U2+W+X+LF))
Start:  AIC=328.32
R ~ 1

       Df Sum of Sq   RSS    AIC
+ Ex0   1   31223.3 31349 299.86
+ Ex1   1   29883.0 32690 301.75
+ W     1   11204.5 51368 322.08
+ Ed    1    7765.0 54808 325.00
+ N     1    5223.8 57349 327.04
+ M     1    3781.6 58791 328.16
<none>              62573 328.32
+ U2    1    2036.8 60536 329.47
+ X     1    1971.3 60601 329.52
+ LF    1    1862.6 60710 329.60
+ S     1     297.8 62275 330.75
+ Age   1     138.2 62435 330.86
+ NW    1      42.0 62531 330.93
+ U1    1      12.4 62560 330.95

Step:  AIC=299.86
R ~ Ex0

       Df Sum of Sq   RSS    AIC
+ Age   1    6967.9 24382 291.19
+ X     1    6281.6 25068 292.44
+ W     1    2323.5 29026 299.04
+ S     1    2225.4 29124 299.19
+ NW    1    1820.5 29529 299.81
<none>              31350 299.86
+ M     1    1751.7 29598 299.91
+ Ex1   1     627.2 30722 301.59
+ N     1     382.6 30967 301.95
+ LF    1     376.1 30973 301.96
+ U2    1     140.6 31209 302.30
+ U1    1      43.6 31306 302.44
+ Ed    1      16.8 31333 302.48

Step:  AIC=291.19
R ~ Ex0 + Age

       Df Sum of Sq   RSS    AIC
+ X     1   1952.67 22429 290.07
+ M     1   1915.55 22466 290.15
+ Ed    1   1617.56 22764 290.74
<none>              24382 291.19
+ LF    1    840.09 23542 292.25
+ U2    1    750.97 23631 292.42
```

```
+ U1     1    403.16 23978 293.08
+ N      1    377.51 24004 293.12
+ Ex1    1    255.40 24126 293.35
+ NW     1     57.80 24324 293.72
+ W      1     47.33 24334 293.74
+ S      1     47.25 24334 293.74

Step:  AIC=290.07
R ~ Ex0 + Age + X

       Df Sum of Sq   RSS    AIC
+ Ed     1    7285.0 15144 275.04
+ M      1    3170.9 19258 285.85
+ NW     1    1863.5 20565 288.81
+ W      1    1773.6 20655 289.00
+ LF     1    1697.3 20732 289.17
+ N      1    1399.7 21029 289.81
<none>               22429 290.07
+ S      1     626.4 21802 291.44
+ U1     1     368.2 22061 291.96
+ U2     1     218.3 22211 292.27
+ Ex1    1      52.0 22377 292.61

Step:  AIC=275.04
R ~ Ex0 + Age + X + Ed

       Df Sum of Sq   RSS    AIC
+ U2     1   2658.64 12485 268.99
<none>               15144 275.04
+ U1     1    828.31 14316 275.14
+ W      1    412.97 14731 276.43
+ N      1    350.30 14794 276.62
+ NW     1    290.29 14854 276.80
+ M      1    232.00 14912 276.98
+ Ex1    1    203.80 14940 277.07
+ LF     1    133.53 15010 277.28
+ S      1      5.66 15138 277.66

Step:  AIC=268.99
R ~ Ex0 + Age + X + Ed + U2

       Df Sum of Sq   RSS    AIC
<none>               12485 268.99
+ U1     1    619.57 11866 269.34
+ N      1    431.90 12053 270.04
+ NW     1    326.65 12159 270.43
+ W      1    162.93 12322 271.04
+ Ex1    1    117.91 12367 271.20
+ LF     1     19.35 12466 271.56
+ S      1      3.39 12482 271.62
+ M      1      0.39 12485 271.63

Call:
lm(formula = R ~ Ex0 + Age + X + Ed + U2, data = crime.data.wo.outliers)

Coefficients:
(Intercept)          Ex0          Age            X           Ed           U2
  -572.5046       1.1952       1.2441       0.6019       2.2317       1.0592
```

So we can see the forward stepwise procedure based on the BIC criterion selects the same predictors as t-tests i.e. R $\sim$ Ex0 + Age + X + Ed + U2. Running step function for backward and both directions gives the same results.

* based on Mellows C_p criterion

  Here we can see the best model for number of predictors included in the model:

```
> subsets<-regsubsets(R~Age+S+Ed+Ex0+Ex1+M+N+NW+U1+U2+W+X+LF,crime.data.wo.outliers,nvmax=13)
> (regsubsets.sum<-summary(subsets))
```

```
Subset selection object
```

```
Call: regsubsets.formula(R ~ Age + S + Ed + Ex0 + Ex1 + M + N + NW +
    U1 + U2 + W + X + LF, crime.data.wo.outliers, nvmax = 13)
13 Variables  (and intercept)
    Forced in Forced out
Age     FALSE      FALSE
S       FALSE      FALSE
Ed      FALSE      FALSE
Ex0     FALSE      FALSE
Ex1     FALSE      FALSE
M       FALSE      FALSE
N       FALSE      FALSE
NW      FALSE      FALSE
U1      FALSE      FALSE
U2      FALSE      FALSE
W       FALSE      FALSE
X       FALSE      FALSE
LF      FALSE      FALSE
1 subsets of each size up to 13
Selection Algorithm: exhaustive
          Age S   Ed  Ex0 Ex1 M   N   NW  U1  U2  W   X   LF
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 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
> regsubsets.sum$cp
```

```
 [1] 48.965515 30.969270 16.520897  8.459246  2.829571  3.051562  3.419486
 [8]  4.242427  6.145959  8.063699 10.001788 12.001044 14.000000
```

We can also visualise selection of the best model (with 5 predictors):

```
> par(mar=c(1,1,1,1))
> plot (subsets, scale="Cp")
```

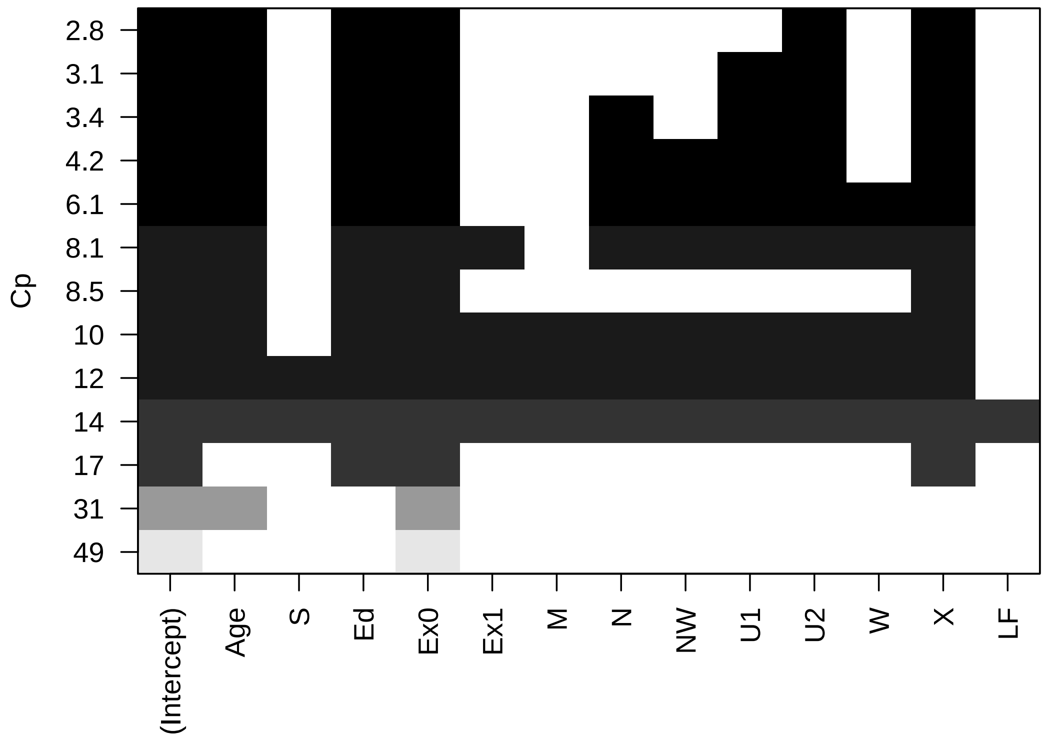

Figure 11: Finding best model using exhaustive search and C_p criterion.

The Mellows C_p criterion used in regsubsets selected the same model as BIC criterion using by the step function i.e. R $\sim$ Ex0 + Age + X + Ed + U2.

```
> par(mar=c(2,2,1,1))
> plot(2:14,regsubsets.sum$cp,xlab="No. of Parameters", ylab="Cp Statistic")
> abline(0,1)
```

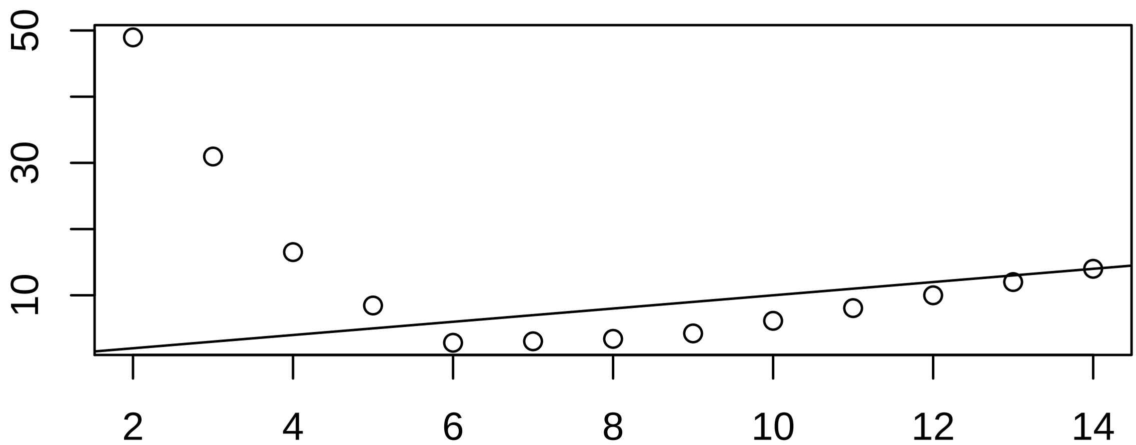

Figure 12: C_p against number of model parameters.

We can see that point for 5 predictors (i.e. 6 parameter model) lies below line p so the model fits data well.

Exercise 3.

Read in data longley from library MASS and fit a linear regression model taking variable Employed as response variable and the rest of variables as predictors.

```
> data(longley)
> describe(longley)

             var  n    mean    sd  median trimmed    mad     min     max  range
GNP.deflator   1 16  101.68 10.79  100.60  101.93  15.79   83.00  116.90  33.90
GNP            2 16  387.70 99.39  381.43  386.71 118.57  234.29  554.89 320.61
Unemployed     3 16  319.33 93.45  314.35  317.26 116.75  187.00  480.60 293.60
Armed.Forces   4 16  260.67 69.59  271.75  261.84  52.78  145.60  359.40 213.80
Population     5 16  117.42  6.96  116.80  117.22   8.17  107.61  130.08  22.47
Year           6 16 1954.50  4.76 1954.50 1954.50   5.93 1947.00 1962.00  15.00
Employed       7 16   65.32  3.51   65.50   65.31   4.31   60.17   70.55  10.38
              skew kurtosis    se
GNP.deflator -0.13    -1.40  2.70
GNP           0.02    -1.35 24.85
Unemployed    0.14    -1.30 23.36
Armed.Forces -0.37    -1.20 17.40
Population    0.26    -1.27  1.74
Year          0.00    -1.43  1.19
Employed     -0.09    -1.55  0.88

> longley.lm<-lm(Employed~.,longley)
```

- Check the data set for collinearity of predictors by making scatterplots for every pair of predictors and calculating variance inflation factors for every predictor (function vif() in library faraway).

  ```
  > vif(longley.lm)

  GNP.deflator          GNP   Unemployed Armed.Forces   Population         Year
     135.53244   1788.51348     33.61889      3.58893    399.15102    758.98060
  ```

  We have VIF values larger than 10 so we have colinear variables in the model.

  Checking pairwise colinearity:

```
> pairs(longley[1:(length(longley)-1)])
```

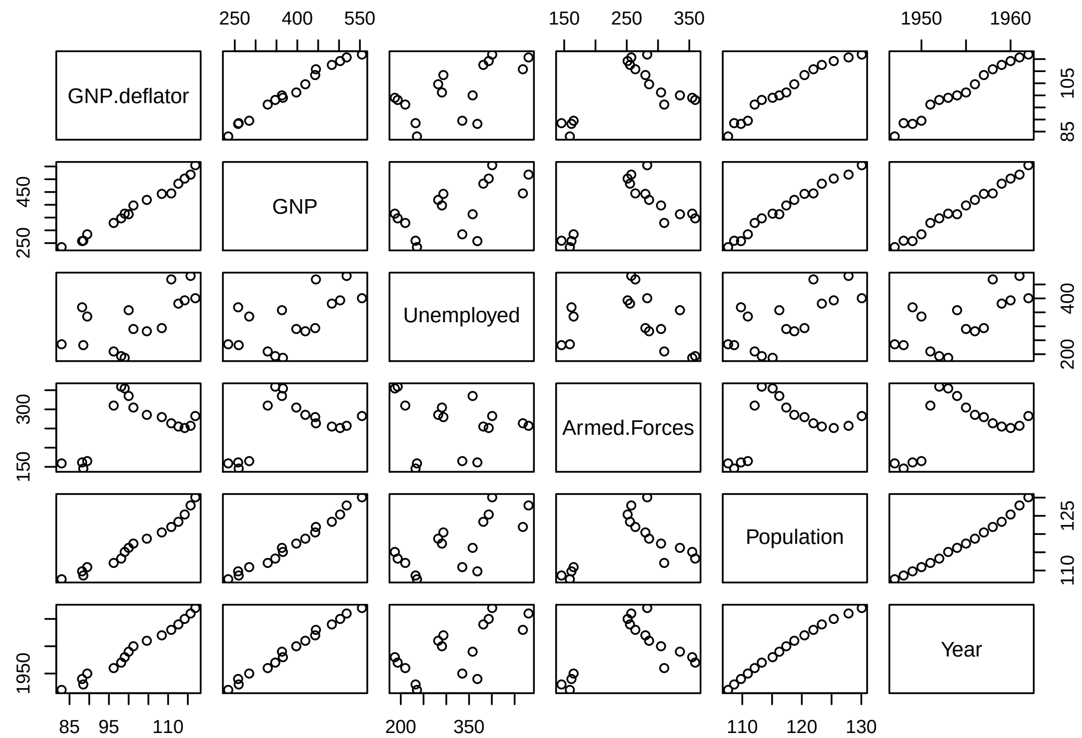

Figure 13: scatterplots for every pair of variables.

We can spot following coolinera pairs: (GNP.deflator, GNP), (GNP.deflator, Population), (GNP.deflator, Year), (GNP, Population), (GNP, Year), (Population, Year)

- Fit a model to the data using ridge regression method (function lm.ridge()). Take the range of λ as an interval [0, 0.2] with step 0.001.

```
> longley.lm.ridge<-lm.ridge(Employed~.,longley,lambda = seq(from=0,to=0.2,by=0.001))
```

- Plot fitted values of coefficients (bi) as a function of parameter λ (use function plot(fitted model)).

```
> plot(longley.lm.ridge,xlab="lambda",ylab="coef")
> legend("topright", names(longley)[-7], col = 1:6, lty = 1:6)
```

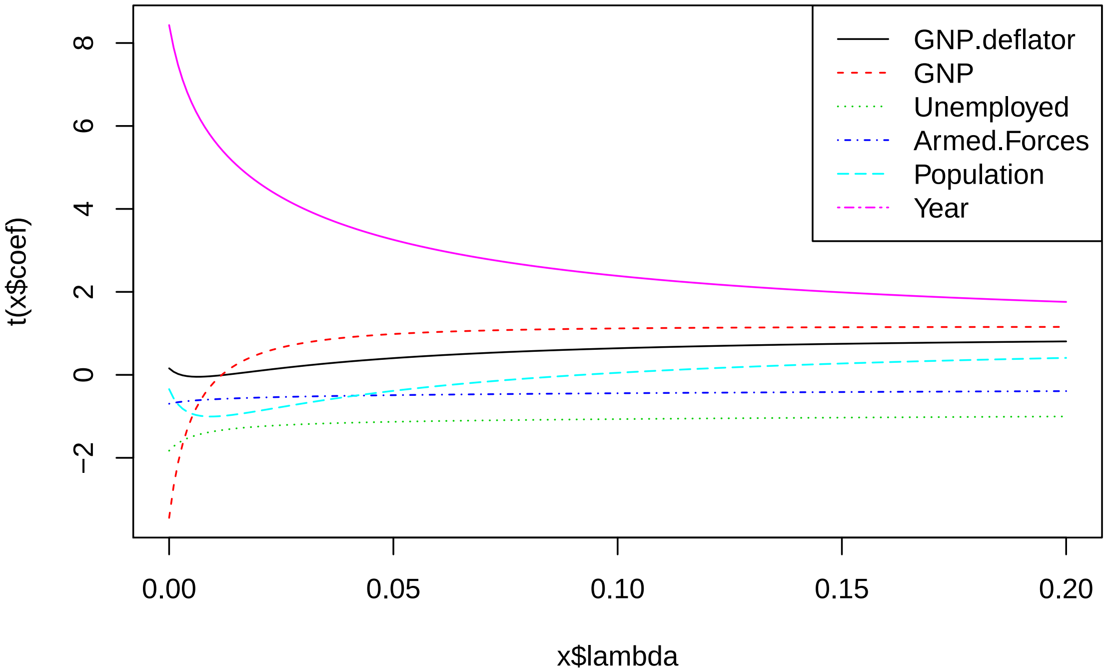

Figure 14: fitted values of coefficients (bi) as a function of parameter λ

- Choose the best value of penalty parameter λ using crossvalidation (use function select(fitted model))

```
> select(longley.lm.ridge)
```

```
modified HKB estimator is 0.004275357
modified L-W estimator is 0.03229531
smallest value of GCV  at 0.003
```

The smallest GCV is achieved at λ =0.003.

The impact of lambda on GCV can be visualized:

```
> par(mar=c(4,4,0,0))
> plot(longley.lm.ridge$lambda,longley.lm.ridge$GCV,xlab="lambda",ylab="GCV",pch=".",type="p",cex=3)
```

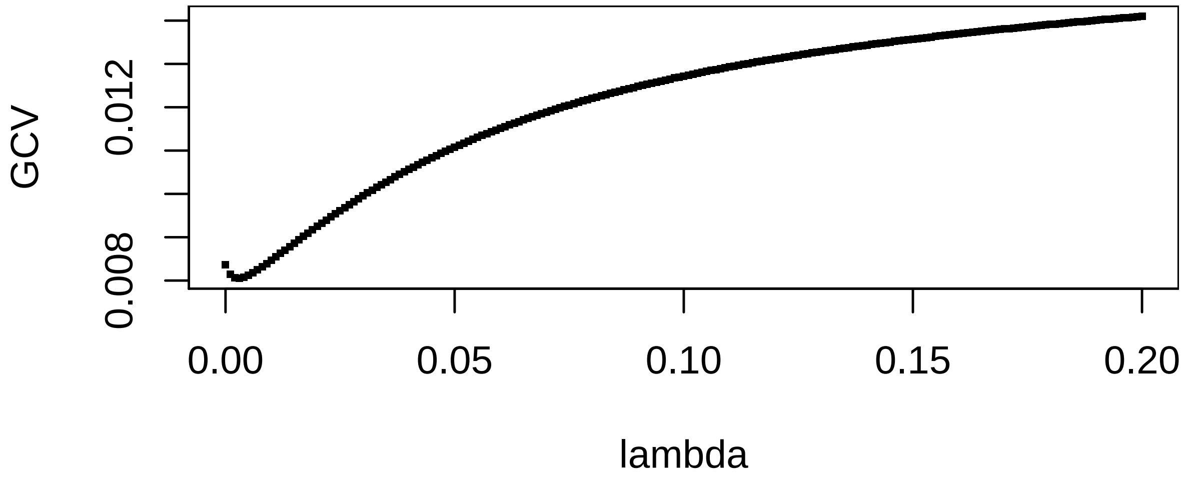

Figure 15: GCV with respect to lmbda

- Compare the fitted coefficient for variable GNP in the resulting model with the one fitted by regular least squares method.

```
> coef(longley.lm.ridge)["0.003","GNP"]

[1] -0.01739859

> lm(Employed~.,longley)$coef["GNP"]

        GNP
-0.03581918
```

The GNP coefcient from the ridge regression is smaller (in absolute value) from the LS regression GNP coeficient. It is the outcome of the penelty term added to the minimized function.

Exercise 4.

File prostate.txt contains data on prostate cancer for 97 men. We are interested in modelling the relationship between lpsa (logarithm of prostate specific antigen) with all the other variables in the data set (except train). Use LASSO method as a way to select the best subset of predictors in this model.

```
> prostate.data <- read.table(file="prostate.data",header=T)
```

- Use function lars() (library lars) which computes a sequence of all coefficients for different values of penalty parameter λ. This returns an object of a class lars.

```
> library(lars)
> x<-as.matrix(
+                 prostate.data[,c("lcavol","lweight","age","lbph","svi","lcp","gleason","pgg45")])
> y<-prostate.data$lpsa
> prostate.lars<-lars(x,y,type="lasso")
```

- Apply the following functions on the resulting object: print(), plot(), coef(), summary()

```
> print(prostate.lars)

Call:
lars(x = x, y = y, type = "lasso")
R-squared: 0.663
Sequence of LASSO moves:
     lcavol svi lweight pgg45 lbph age gleason lcp
Var       1   5       2     8    4   3       7   6
Step      1   2       3     4    5   6       7   8
```

```
> plot(prostate.lars)
> legend("topleft", names(prostate.data)[c(-9,-10)], col=1:6,lty=1:5,cex=.8)
```

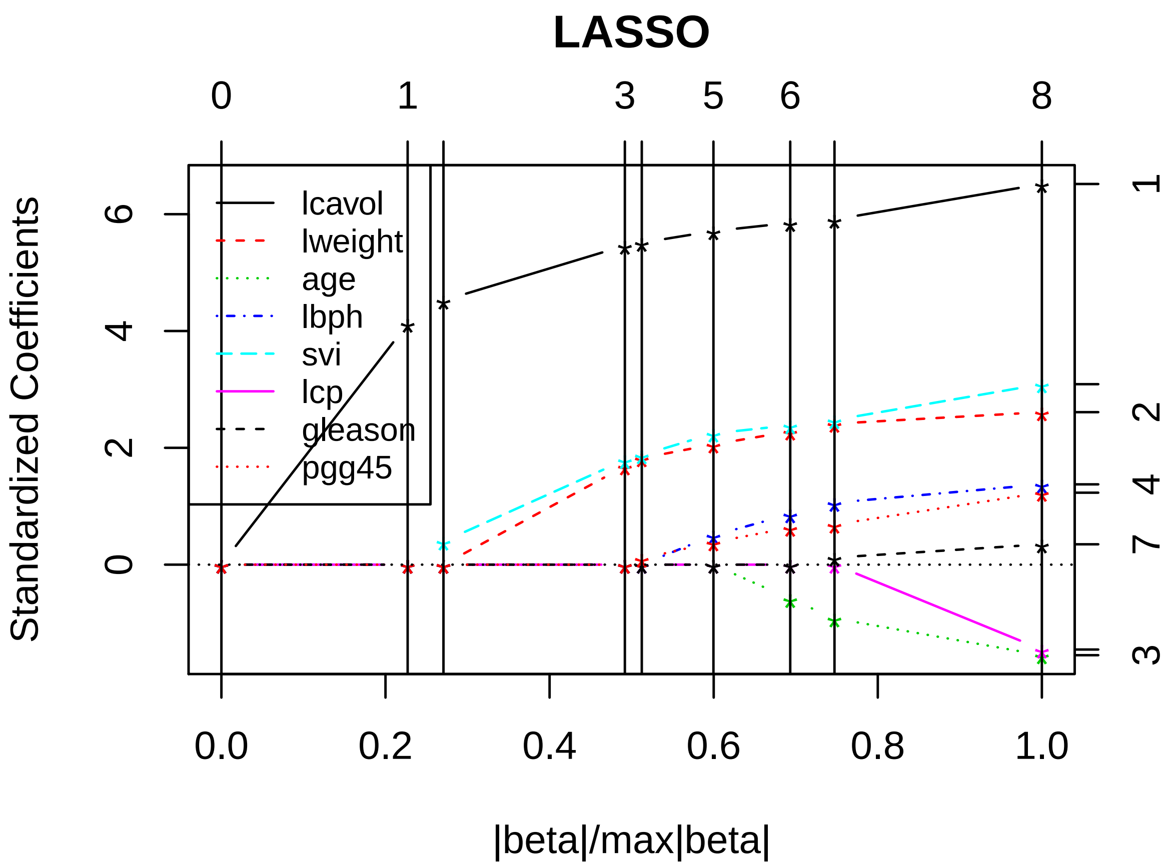

Figure 16: Visualization of the coefficients paths for LASSO

```
> coef(prostate.lars)

        lcavol  lweight          age       lbph        svi        lcp
 [1,] 0.0000000 0.0000000  0.000000000 0.00000000 0.00000000  0.0000000
 [2,] 0.3573072 0.0000000  0.000000000 0.00000000 0.00000000  0.0000000
 [3,] 0.3916323 0.0000000  0.000000000 0.00000000 0.09772183  0.0000000
 [4,] 0.4729686 0.4010287  0.000000000 0.00000000 0.44189300  0.0000000
 [5,] 0.4772307 0.4343405  0.000000000 0.00000000 0.46205106  0.0000000
 [6,] 0.4945025 0.4904080  0.000000000 0.03525646 0.55370843  0.0000000
 [7,] 0.5067509 0.5431376 -0.008040524 0.06070074 0.58841287  0.0000000
 [8,] 0.5115481 0.5748987 -0.012590999 0.07452697 0.61037529  0.0000000
 [9,] 0.5643413 0.6220198 -0.021248185 0.09671252 0.76167340 -0.1060509
         gleason        pgg45
 [1,] 0.00000000 0.0000000000
 [2,] 0.00000000 0.0000000000
 [3,] 0.00000000 0.0000000000
 [4,] 0.00000000 0.0000000000
 [5,] 0.00000000 0.0003752489
 [6,] 0.00000000 0.0013866469
 [7,] 0.00000000 0.0022898666
 [8,] 0.01704893 0.0024820450
 [9,] 0.04922793 0.0044575118

> summary(prostate.lars)

LARS/LASSO
Call: lars(x = x, y = y, type = "lasso")
  Df     Rss       Cp
0  1 127.918 166.4298
1  2  76.392  63.1249
2  3  70.247  52.5663
```

```
3  4  50.244  13.6861
4  5  49.257  13.6683
5  6  46.308   9.6411
6  7  44.621   8.1927
7  8  44.053   9.0321
8  9  43.058   9.0000
```

We can see as the L1 regularizaion constraint is loosened up then the model fits the data better and the RSS decreases and the coefficients increase.

- Choose the best subset of predictors in LASSO regresssion on the basis of Mallows Cp criterion (use functions: plot(lasso object,breaks=FALSE,plottype="Cp"), lasso_object$Cp where lasso_object is an object returned by function lars()).

```
> plot(prostate.lars,breaks=FALSE,plottype="Cp")
```

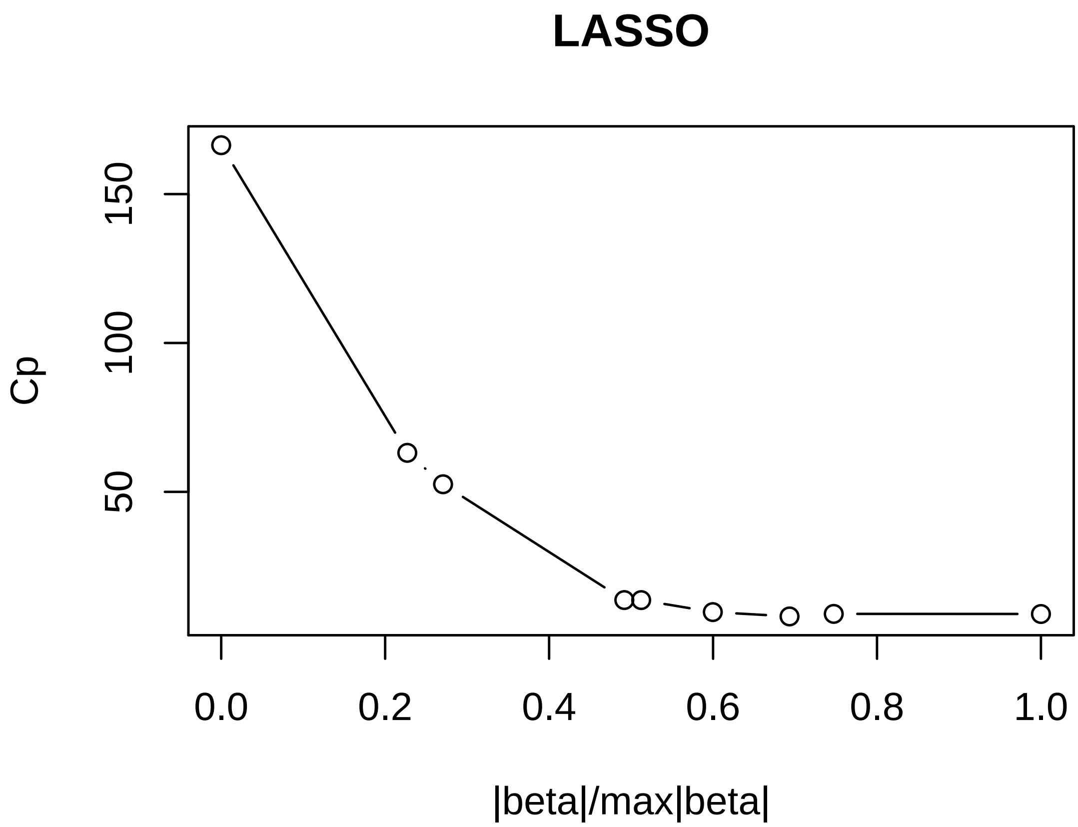

Figure 17: Visualization of Cp for LASSO

So to select the best coefficient based on C_p criterium we select the set with the smallest C_p:

```
> (coef.best<-as.numeric(which.min(prostate.lars$Cp)))
```

```
[1] 7
```

- What are the values of fitted coefficients (bi) in LASSO regression for the chosen model? In order to access the sequence of fitted coefficients use lasso_object$beta[number] where number is a number of a chosen step.

```
> prostate.lars$beta[coef.best,]
```

```
     lcavol     lweight          age         lbph          svi          lcp
0.506750858 0.543137561 -0.008040524  0.060700743  0.588412869  0.000000000
    gleason       pgg45
0.000000000 0.002289867
```

- Choose the best subset of predictors in LASSO regresssion on the basis of crossvalidation. Use function cv.lars().

```
> prostate.cv.lars <-cv.lars(x,y,K=10,type="lasso")
```

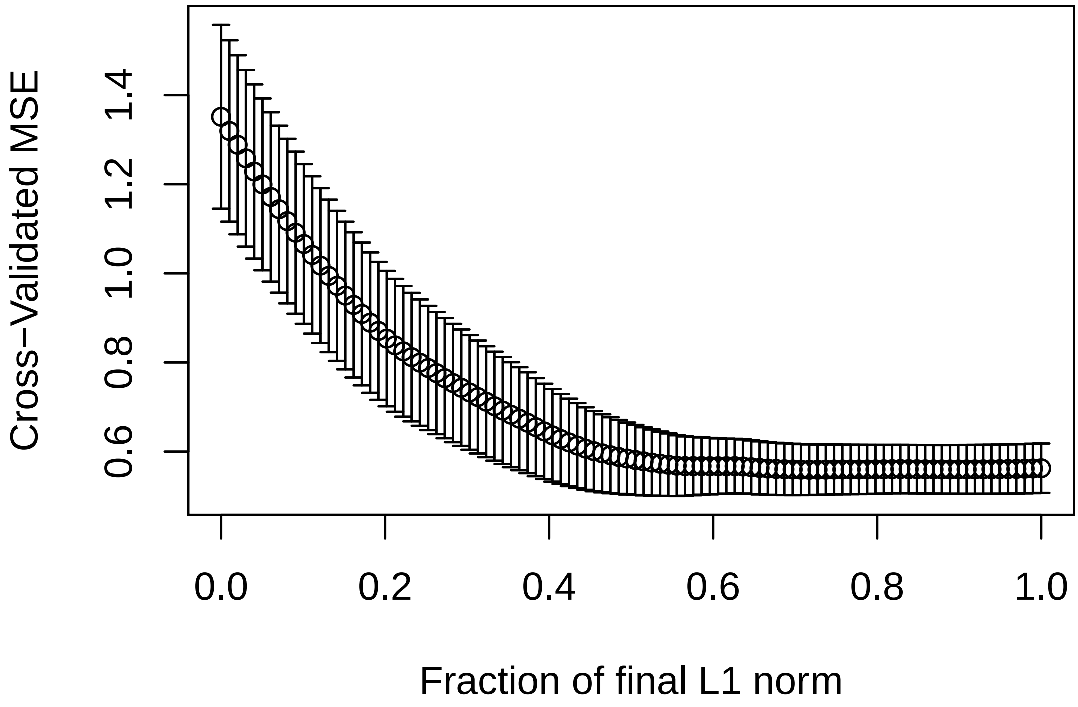

Figure 18: Visualisation of CV MSE

Getting coefficient for the minimum MSE:

```
> frac<-prostate.cv.lars$index[prostate.cv.lars$cv==min(prostate.cv.lars$cv)]
> predict.lars(prostate.lars, type="coefficients", mode="fraction", s=frac)$coef
```

```
     lcavol      lweight          age         lbph          svi          lcp
 0.508875493  0.557204220 -0.010055882  0.066824234  0.598139797  0.000000000
     gleason        pgg45
 0.007550794  0.002374980
```

Exercise 5.

File cities.txt contains values of the following attributes for 46 citites:
Work - weighted average value of number of work hours,
Price - cost of living index,
Salary - hour salary index.

```
> cities.data <- read.table(file="cities.txt",header=T)
> describe(cities.data)
```

```
       var  n    mean     sd  median trimmed    mad    min    max range skew
Work     1 46 1879.91 174.34 1849.00 1867.00 163.83 1583.0 2375.0 792.0 0.72
Price    2 46   70.10  21.39   70.95   68.88  20.83   30.3  115.5  85.2 0.38
Salary   3 46   39.55  24.76   43.65   38.74  28.84    2.7  100.0  97.3 0.20
       kurtosis    se
Work      -0.02 25.71
Price     -0.53  3.15
Salary    -0.85  3.65
```

- Transform the variables to have mean equal to 0 and std. deviation equal to 1 (use function scale()).

```
> cities.data.st<-scale(cities.data)
> describe(cities.data.st)
```

```
       var  n mean sd median trimmed  mad   min  max range skew kurtosis   se
Work     1 46    0  1  -0.18   -0.07 0.94 -1.70 2.84  4.54 0.72    -0.02 0.15
Price    2 46    0  1   0.04   -0.06 0.97 -1.86 2.12  3.98 0.38    -0.53 0.15
Salary   3 46    0  1   0.17   -0.03 1.16 -1.49 2.44  3.93 0.20    -0.85 0.15
```

- Make a scatterplot of variables Work and Price. Find a direction in which the variabiliy of data is the largest (first principal direction). Add the first and second principal directions to the plot. Tip: use function princomp() for the two first columns of the standardized data. Access the directions in which the variabiliy of data is the largest by using component $loadings of the returned object.

```
> work.price<-cities.data.st[,c("Work","Price")]
> plot(work.price)
> work.price.pc<-princomp(work.price)
> (work.price.pc.first<-work.price.pc$loadings[,1])

      Work      Price
-0.7071068  0.7071068

> (work.price.pc.second<-work.price.pc$loadings[,2])

      Work      Price
-0.7071068 -0.7071068

> abline(0,work.price.pc.first[2]/work.price.pc.first[1],col="red")
> abline(0,work.price.pc.second[2]/work.price.pc.second[1],col="green")
> legend("topright", c("first principal direction","second principal direction"), col = c("red","green"), lty
```

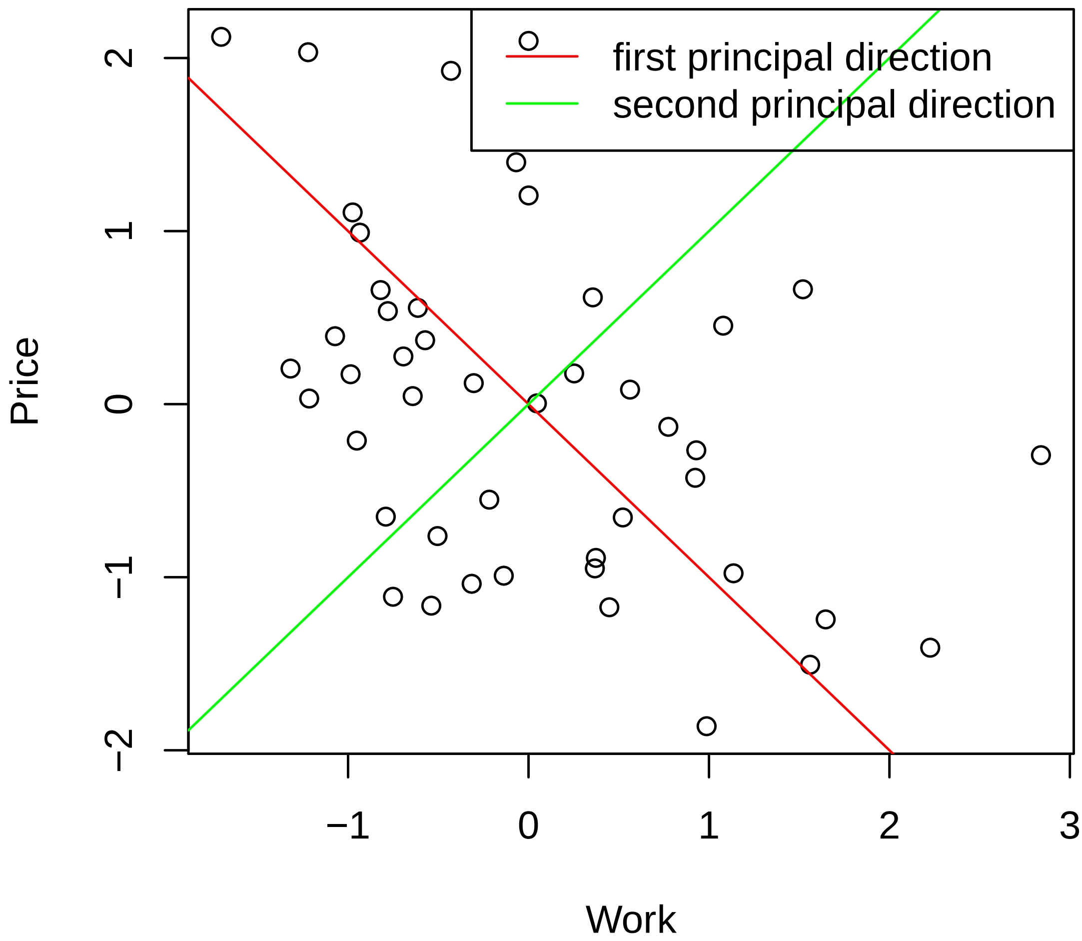

Figure 19: Price ∼ Work scatter plot

- Perform principal components analysis for all three variables (use function princomp() and summary() and plot() of the returned object).

```
> cities.data.st.pc<-princomp(cities.data.st)
> (cities.data.st.pc.sum <- summary(cities.data.st.pc))

Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     1.4536997 0.7935060 0.4380504
Proportion of Variance 0.7200679 0.2145480 0.0653841
Cumulative Proportion  0.7200679 0.9346159 1.0000000
```

```
> plot(cities.data.st.pc)
```

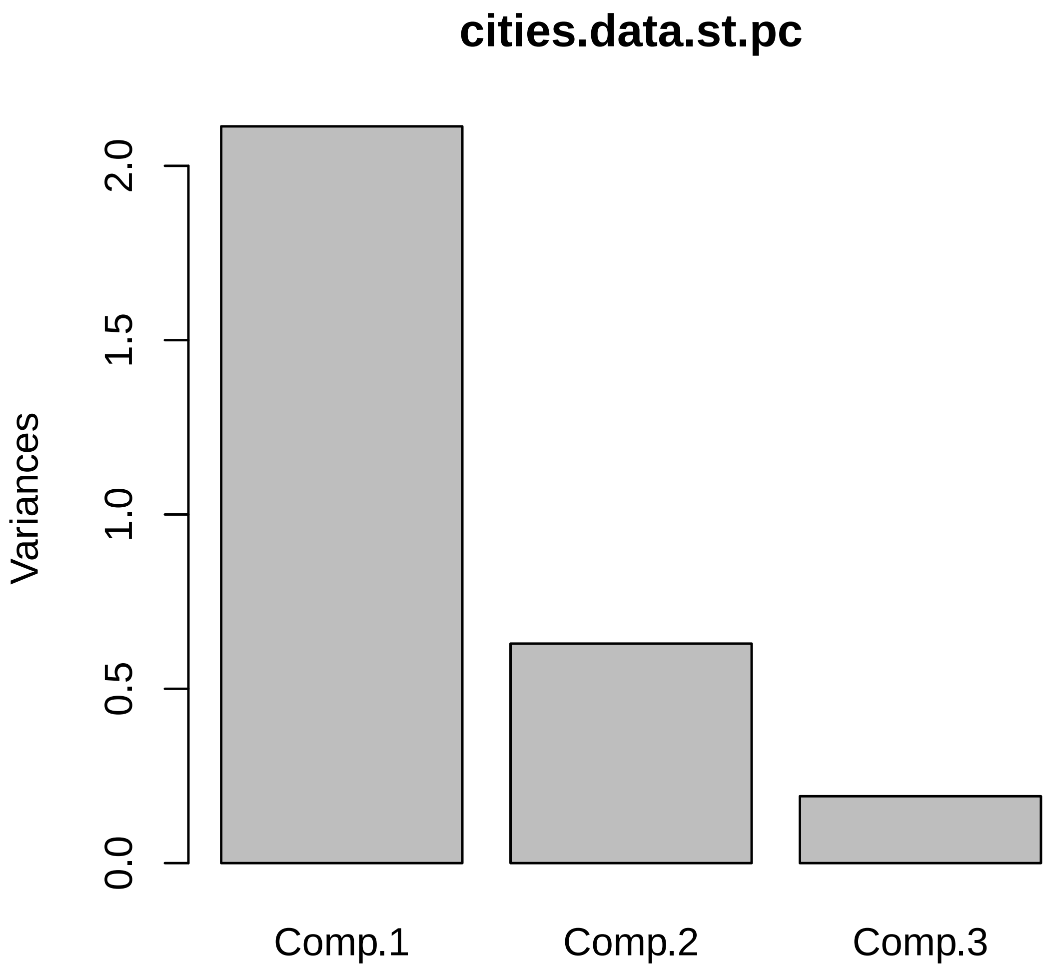

Figure 20: Principal components' variance of cities data

- What percent of data variability is contained in each component? Can we reduce the number of dimensions for this data?

  ```
  > cities.data.st.pc$sdev^2/sum(cities.data.st.pc$sdev^2)

     Comp.1    Comp.2    Comp.3
  0.7200679 0.2145480 0.0653841
  ```

  Because the first and second component constitute together more than 80% of data variability we can reduce the number of dimensions to 2.

- Which city has the largest value of first principal component? How can we interpret it? Manila has the largest value of the first principal component.

  ```
  > which.max(cities.data.st.pc$scores[,1])

  Manila
      25
  ```

The first loading and values for the Manila in the original coordnates are:

```
> cities.data.st.pc$loadings[,1]

      Work      Price     Salary
 0.4847566 -0.6178516 -0.6190884

> cities.data.st["Manila",]

     Work     Price    Salary
 2.226002 -1.407254 -1.435741
```

Because Manila has bigger work index then average city and smaller price and sallary, it is favoured by the first loading which has positive weight for work and negative for price and salary. So Manila is the city were people work more and enjoy smaller prices but earn less that average country in the data set.

Exercise 6.

Data yarn in library pls are related to PET test (Positron Emission Tomography). Data contains 28 observations and consists of three parts:
NIR - experiment matrix containing information of 268 wavelengths,
density - response variable,
train - logical vector, TRUE for training set observations, FALSE for test set observations.

```
> library(pls)
> data(yarn)
```

- Use principal components regression (PCR) method to fit a model describing dependence between response variable density and variables contained in matrix NIR (function pcr() in library pls). Use only training observations.

  Fitting the model and checking what is the RMSE for the model predicting training data and test data:

```
> yarn.pcr <- pcr(density ~ NIR, data = yarn, subset=train)
> summary(yarn.pcr)

Data:         X dimension: 21 268
        Y dimension: 21 1
Fit method: svdpc
Number of components considered: 20
TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X         52.053    98.78    99.51    99.74    99.89    99.98    99.99    99.99
density    5.173    98.21    99.47    99.77    99.95    99.99    99.99   100.00
         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
X          99.99       100       100       100       100       100       100
density   100.00       100       100       100       100       100       100
         16 comps  17 comps  18 comps  19 comps  20 comps
X             100       100       100       100       100
density       100       100       100       100       100

> #defining root mean square error
> rmse<-function(x,y) sqrt(mean((x-y)^2))
> #rmse of the train data
> rmse(yarn.pcr$fitted.values[,1,17],yarn$density[yarn$train])

[1] 0.02654927

> #rmse of the test data
> rmse(predict(yarn.pcr,yarn$NIR[!yarn$train,],ncomp=c(17))[,1,1],yarn$density[!yarn$train])

[1] 0.08253163
```

  As we can see the RMSE for the test data is bigger than for the train data as expected. I used 17 componet model becacause of the analysis performed in the next section.

- Make a selection of principal components that should be included in the model. Use method of crossvalidation (option validation="CV" in function pcr(), also use plot(fitted.model,plottype="validation"))

```
> yarn.pcr.cv.val <- pcr(density ~ NIR, data = yarn, validation="CV")
> summary(yarn.pcr.cv.val)
```

```
Data:        X dimension: 28 268
        Y dimension: 28 1
Fit method: svdpc
Number of components considered: 24

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           27.46    29.52    4.277    2.789    2.778    2.217   0.4926
adjCV        27.46    30.81    4.241    2.736    2.781    2.095   0.4795
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV      0.5235   0.2815   0.2797    0.2612    0.2689    0.2707    0.2541
adjCV   0.5145   0.2699   0.2698    0.2517    0.2582    0.2638    0.2441
       14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
CV       0.2610    0.2427    0.2494    0.2322    0.2223    0.2328    0.2040
adjCV    0.2489    0.2321    0.2376    0.2205    0.2114    0.2212    0.1955
       21 comps  22 comps  23 comps  24 comps
CV       0.1978    0.2134    0.2169    0.2123
adjCV    0.1884    0.2034    0.2068    0.2023

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X          52.17    98.60    99.47    99.70    99.88    99.97    99.98    99.99
density     5.50    98.15    99.40    99.58    99.95    99.99    99.99   100.00
         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
X          99.99     99.99     99.99       100       100       100       100
density   100.00    100.00    100.00       100       100       100       100
         16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
X             100       100       100       100       100       100       100
density       100       100       100       100       100       100       100
         23 comps  24 comps
X             100       100
density       100       100
```

```
> plot(yarn.pcr.cv.val,plottype="validation")
```

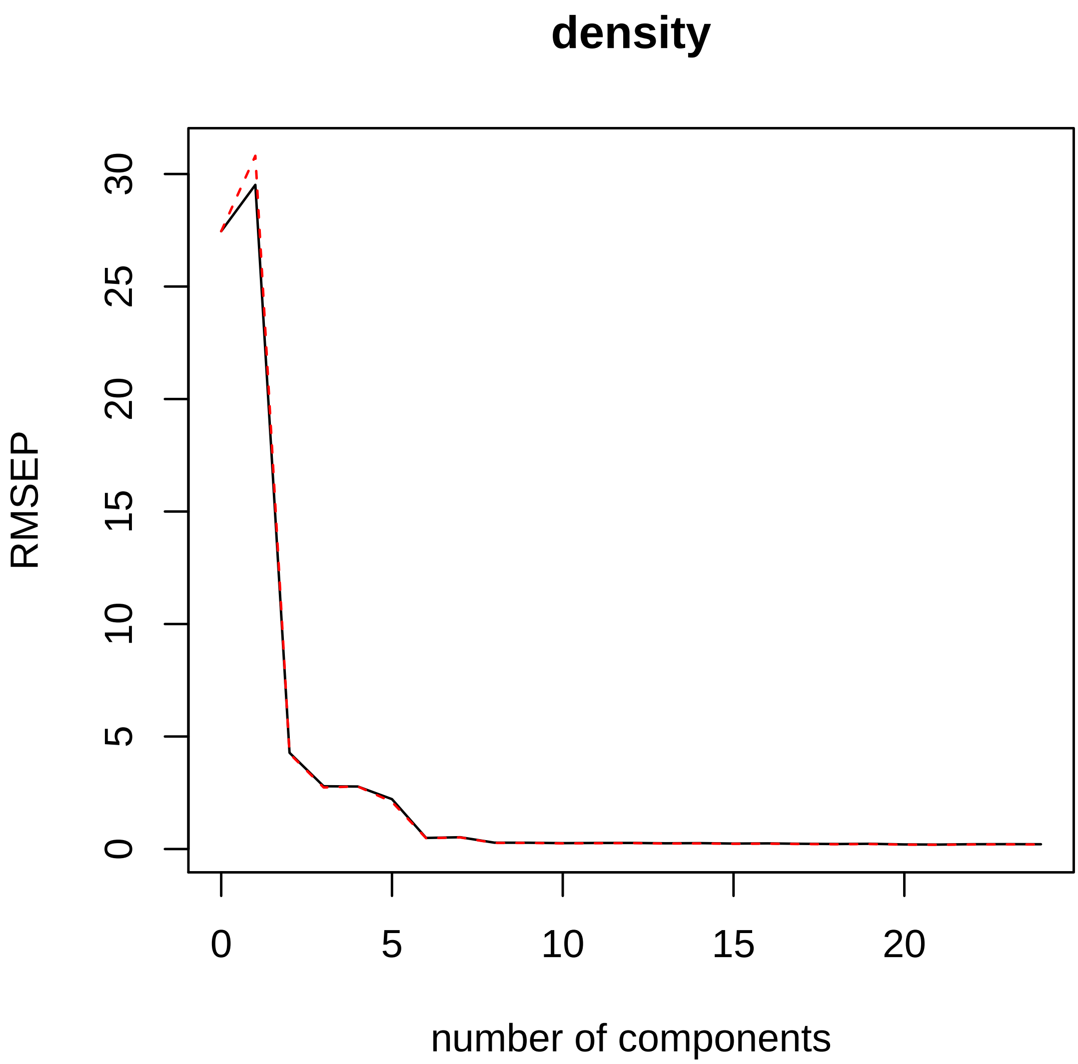

Figure 21: RMSEP ∼ number of componets

We can see that RMSEP(root mean squared error of prediction) for more then 16 components levels off so we can choose first 17 components as new dimensions for the model.

- Use partial least squares regression (PLSR) method to fit a model describing dependence between response variable density and variables contained in matrix NIR (function plsr() in library pls). Use only training observations.

```
> yarn.plsr <- plsr(density ~ NIR, data = yarn, subset=train)
```

- Make a selection of PLSR components that should be included in the model. Use method of crossvalidation (option validation = "CV" in function plsr()).

```
> yarn.plsr.cv.val <- plsr(density ~ NIR, data = yarn, validation="CV")
> summary(yarn.plsr.cv.val)

Data:           X dimension: 28 268
        Y dimension: 28 1
Fit method: kernelpls
Number of components considered: 24

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           27.46    5.354    4.046    2.022   0.6720   0.4800   0.4216
adjCV        27.46    4.657    4.015    2.018   0.6534   0.4712   0.4143
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV      0.2896   0.2485   0.2400    0.2176    0.2037    0.2070    0.2070
```

```
adjCV   0.2804   0.2417   0.2314    0.2088    0.1954    0.1983    0.1978
       14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
CV       0.2044    0.2061    0.2071    0.2067    0.2059    0.2071    0.2078
adjCV    0.1952    0.1967    0.1975    0.1971    0.1962    0.1975    0.1981
       21 comps  22 comps  23 comps  24 comps
CV       0.2080    0.2081    0.2081    0.2081
adjCV    0.1983    0.1984    0.1984    0.1984

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X          46.83    98.38    99.46    99.67    99.85    99.97    99.98    99.99
density    98.12    98.25    99.64    99.97    99.99    99.99   100.00   100.00
         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
X          99.99     99.99     99.99     99.99     99.99       100       100
density   100.00    100.00    100.00    100.00    100.00       100       100
         16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
X             100       100       100       100       100       100       100
density       100       100       100       100       100       100       100
         23 comps  24 comps
X             100       100
density       100       100
```

```
> plot(yarn.plsr.cv.val,plottype="validation")
```

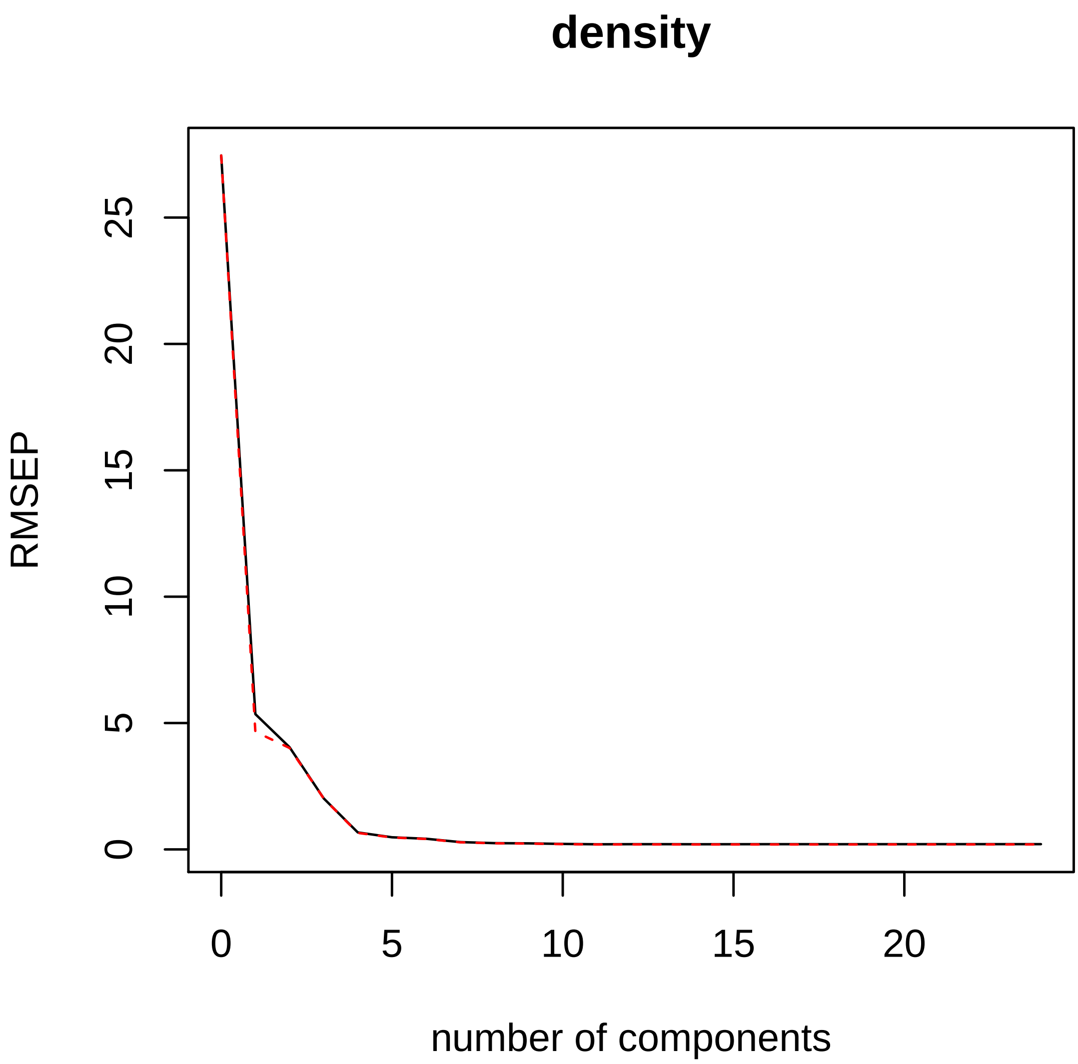

Figure 22: RMSEP ∼ number of componets

We can see that RMSEP(root mean squared error of prediction) for more then 11 components levels off so we can choose first 12 components as new dimensions for the model.

- Interpret two first PLSR components. We can visualize for first two components how they apply to orginal predictors.

```
> matplot(1:length(yarn.plsr$loadings[,1]),yarn.plsr$loadings[,1:2],type="l",xlab="NIR",ylab="loading value")
> abline(h=0,col="blue",lwd=0.5)
> legend("topright", c("Comp 1","Comp 2"), col = 1:2, lty = 1:2)
>
```

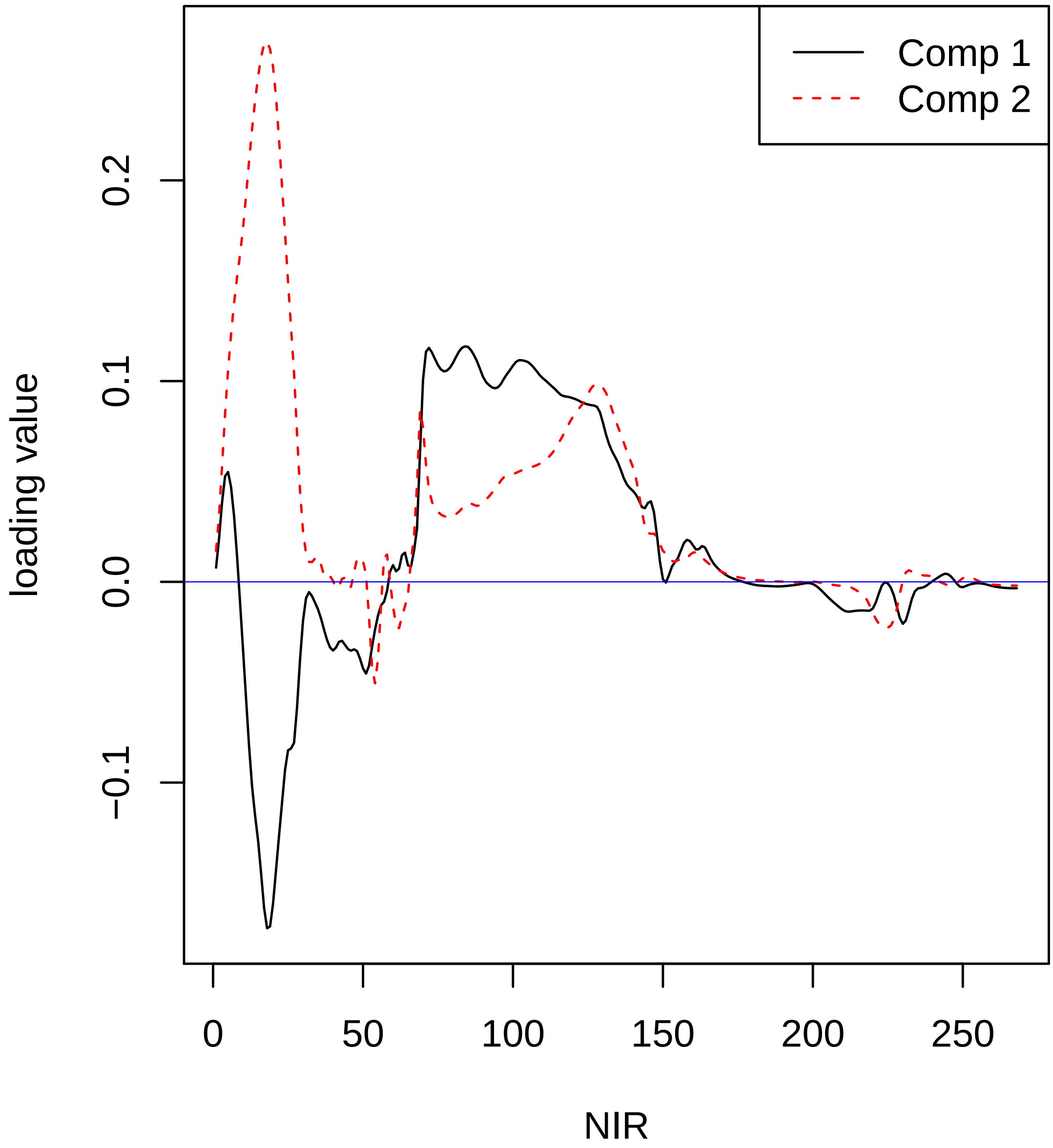

Figure 23: two first PLSR principal components as function of orginal predictors

So we can see that first component negates short wavelenghths and includes middle length waves. The second component carries signal from short wavelengths and also includes middle length waves.

- Make a prediction for test observations using both fitted models (use function predict()).

  Prediction using pcr model:

```
> predict(yarn.pcr, comps = 1:17, newdata = yarn[!yarn$train,])

     density
110 51.04785
22  50.15021
31  32.24931
```

```
41  34.67215
51  30.29867
61  20.36941
71  20.06522
```

Prediction using plsr model:

```
> predict(yarn.plsr, comps = 1:12, newdata = yarn[!yarn$train,])
```

```
     density
110 51.06042
22  50.15600
31  32.22192
41  34.65290
51  30.33657
61  20.36258
71  20.06334
```