Example of data analysis
Dataset `Pollution`

Data Mining
Master Degree in Computer Science
University of Padova

a.y. 2022/2023

Annamaria Guolo

These notes are an example of data analysis using the techniques illustrated during the course. This is just an example: as you know, there is no a unique way to analyse the data and the right model does not exist. Whichever the direction you take for the analysis, please be sure to apply correctly the appropriate instruments for the available data.

1 Dataset `Pollution`

Dataset included in `pollution.RData` refers to a study about the relationship between mortality in 60 US areas and air pollution. Some environmental and demographical information are collected.

- `mortality`: mortality rate (annual deaths for 100000 persons)
- `precipitation`: mean annual precipitation (inches)
- `humidity`: percent relative humidity
- `Jan.temp`: mean January temperature (Farenheit)
- `July.temp`: mean July temperature (Farenheit)
- `over65`: percentage of the population aged 65 years or over
- `house`: population per household
- `education`: median number of school years completed for persons 25 years or older

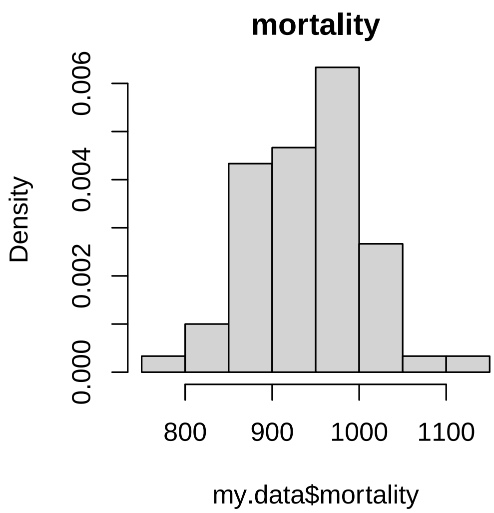
- `comfort`: percentage of the housing that is sound with all facilities
- `density`: population density (in persons per square mile)
- `office`: percentage of office workers
- `poor`: percentage of households with annual income under 3000 dollars
- `HC`: level of hydrocarbons
- `NOX`: dangerous level of oxides of nitrogen?: Si (Yes) ($> 30\ \mu g/mc$), No ($\geq 30\ \mu g/mc$)
- `SO2`: dangerous level of sulfur dioxide?: Si (Yes) ($> 125\ \mu g/mc$), No ($\geq 125\ \mu g/mc$)

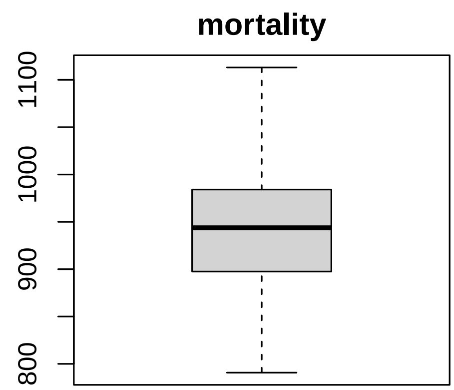
FIRST QUESTION.
Consider the dataset composed by `mortality`, `precipitation`, `humidity`, `HC`, `NOX`, `SO2`. Construct the most appropriate model for the purpose of the analysis. Which variables are associated to the mortality rate?
The response variable is mortality rate, so consider a linear regression model.

```
load('pollution.RData')
dim(pollution)

## [1] 60 15

my.data <- pollution[,c('mortality','HC', 'NOX', 'SO2', 'precipitation',
        'humidity')]
dim(my.data)

## [1] 60  6
```

Start with some graphical analyses, working on the response variable, to check the normality of its distribution

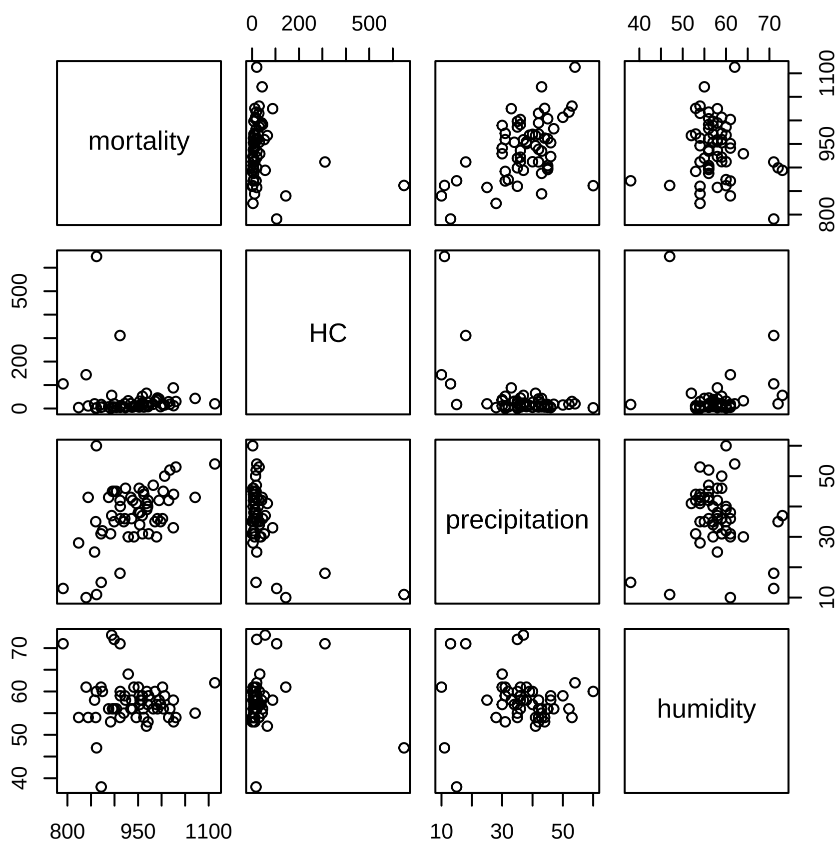
```
par(mfrow=c(1,2))
hist(my.data$mortality, prob=TRUE, main='mortality')
boxplot(my.data$mortality, main='mortality')
```





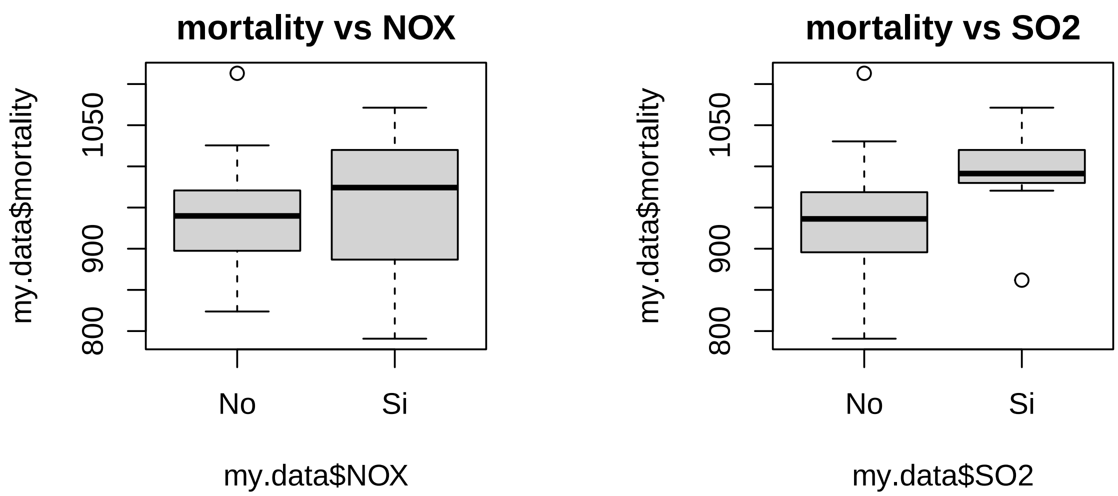
The hypothesis of normality seems to be satisfied.
Consider some graphs to evaluate relationships between variables.

```
pairs(my.data[,c(1,2,5,6)])
```

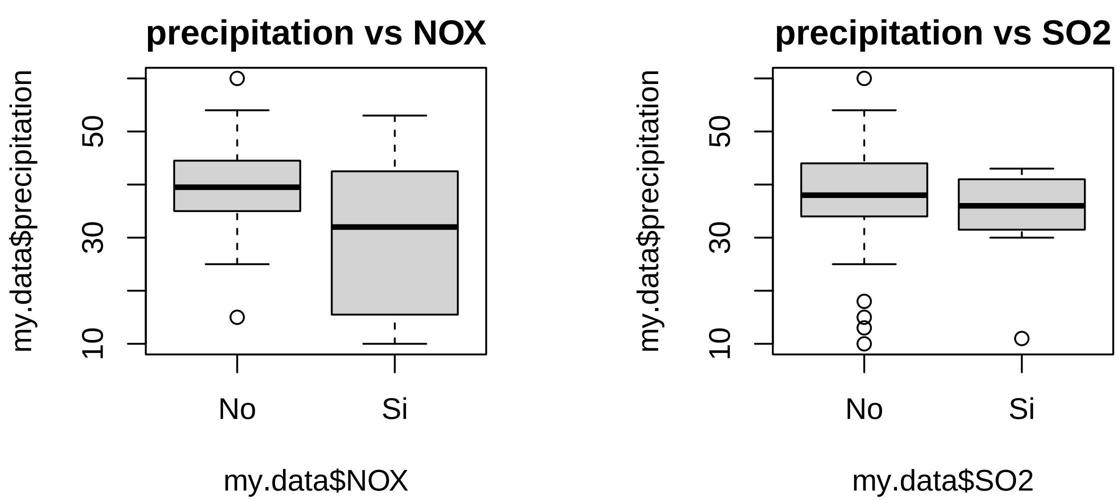


Graphs of interaction between factors and quantitative variables

```
par(mfrow=c(1,2))
boxplot(my.data$mortality~my.data$NOX, main='mortality vs NOX')
boxplot(my.data$mortality~my.data$SO2, main='mortality vs SO2')
```

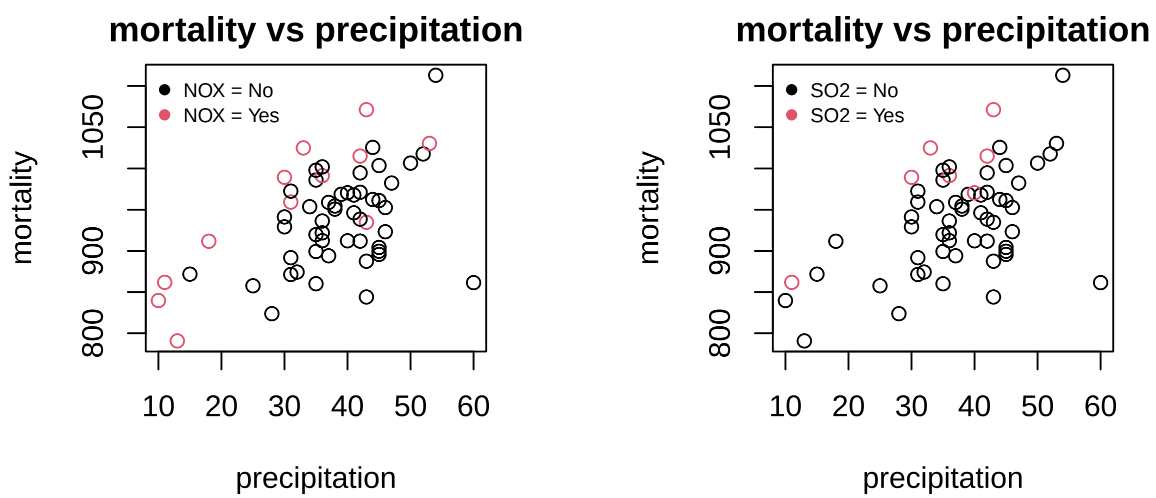



```
par(mfrow=c(1,2))
boxplot(my.data$precipitation~my.data$NOX, main='precipitation vs NOX')
boxplot(my.data$precipitation~my.data$SO2, main='precipitation vs SO2')
```

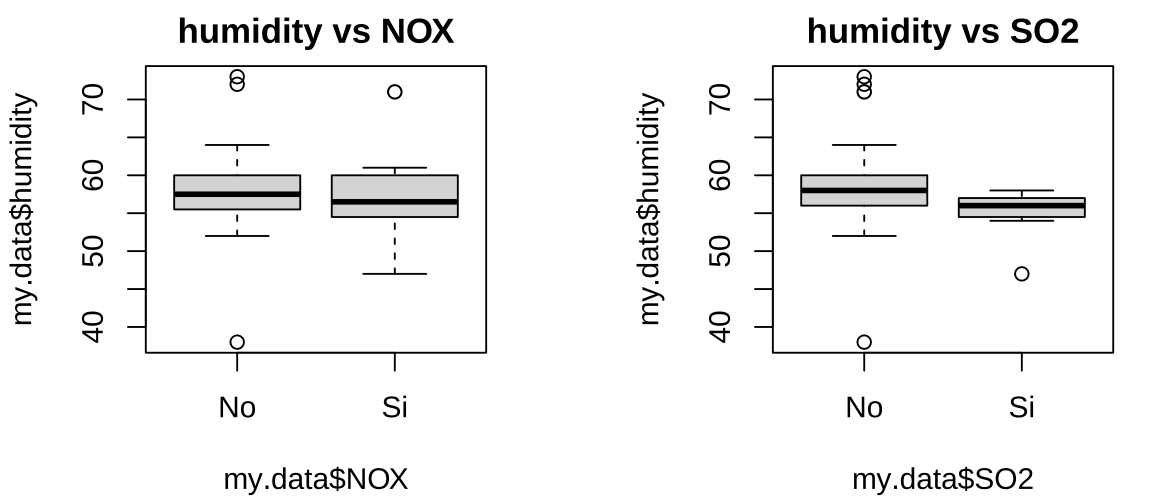



```
par(mfrow=c(1,2))
plot(my.data$precipitation, my.data$mortality, col=my.data$NOX,
        main='mortality vs precipitation', ylab='mortality',
        xlab='precipitation')
legend('topleft', legend=c('NOX = No','NOX = Yes'), col=c(1,2), pch=c(19,19),
```

```
        cex=0.7, bty='n')
plot(my.data$precipitation, my.data$mortality, col=my.data$SO2,
        main='mortality vs precipitation', ylab='mortality',
        xlab='precipitation')
legend('topleft', legend=c('SO2 = No','SO2 = Yes'), col=c(1,2), pch=c(19,19),
        cex=0.7, bty='n')
```

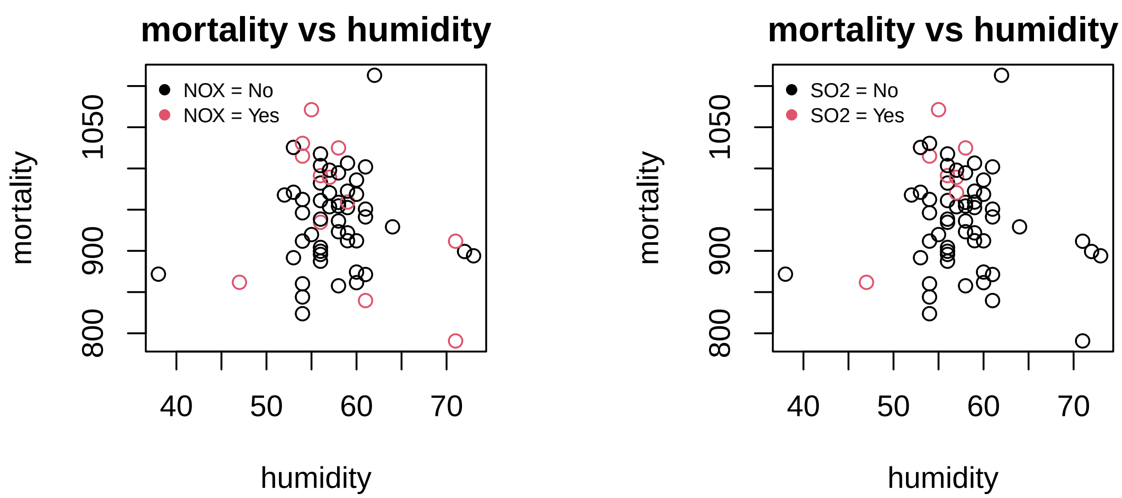



```
par(mfrow=c(1,2))
boxplot(my.data$humidity~my.data$NOX, main='humidity vs NOX')
boxplot(my.data$humidity~my.data$SO2, main='humidity vs SO2')
```

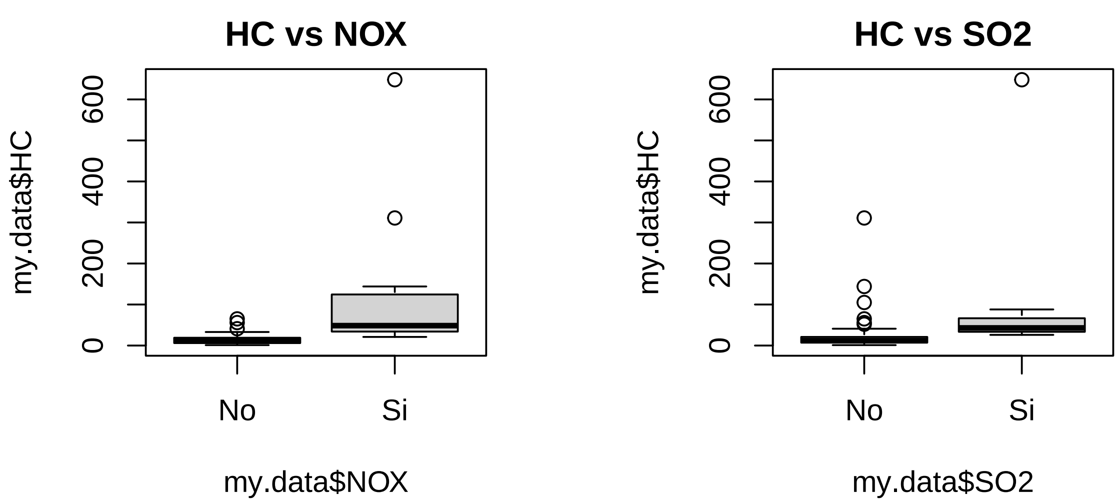



```
par(mfrow=c(1,2))
plot(my.data$humidity, my.data$mortality, col=my.data$NOX, main='mortality vs humidity',
        ylab='mortality', xlab='humidity')
```

```
legend('topleft', legend=c('NOX = No','NOX = Yes'), col=c(1,2), pch=c(19,19),
       cex=0.7, bty='n')
plot(my.data$humidity, my.data$mortality, col=my.data$SO2, main='mortality vs humidity',
       ylab='mortality', xlab='humidity')
legend('topleft', legend=c('SO2 = No','SO2 = Yes'), col=c(1,2), pch=c(19,19),
       cex=0.7, bty='n')
```

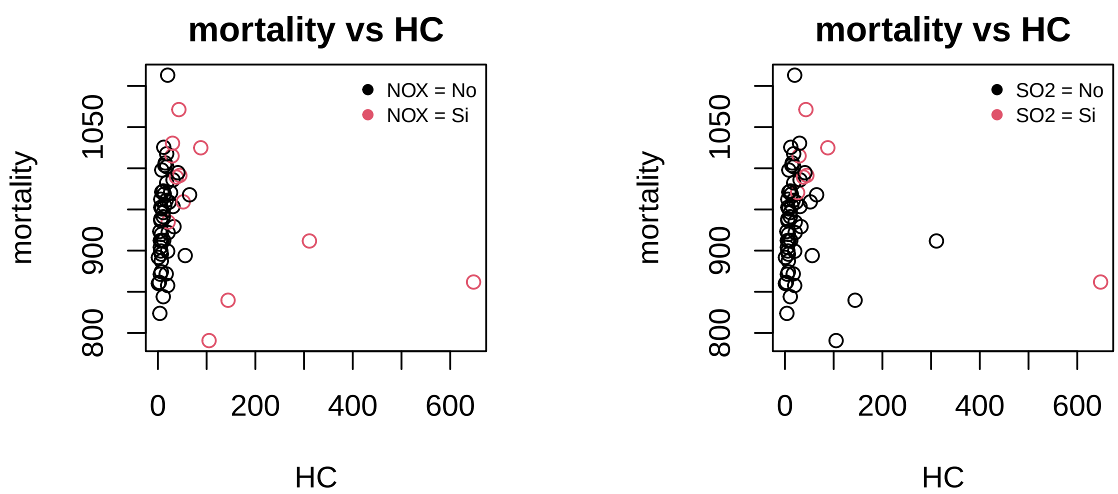



```
par(mfrow=c(1,2))
boxplot(my.data$HC~my.data$NOX, main='HC vs NOX')
boxplot(my.data$HC~my.data$SO2, main='HC vs SO2')
```



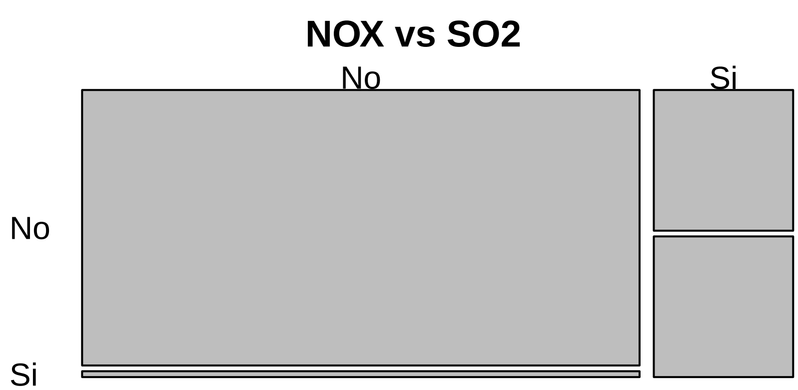

```
par(mfrow=c(1,2))
plot(my.data$HC, my.data$mortality, col=my.data$NOX, main='mortality vs HC',
       ylab='mortality', xlab='HC')
```

```
legend('topright', legend=c('NOX = No','NOX = Si'), col=c(1,2), pch=c(19,19),
       cex=0.7, bty='n')
plot(my.data$HC, my.data$mortality, col=my.data$SO2, main='mortality vs HC',
       ylab='mortality', xlab='HC')
legend('topright', legend=c('SO2 = No','SO2 = Si'), col=c(1,2), pch=c(19,19),
       cex=0.7, bty='n')
```



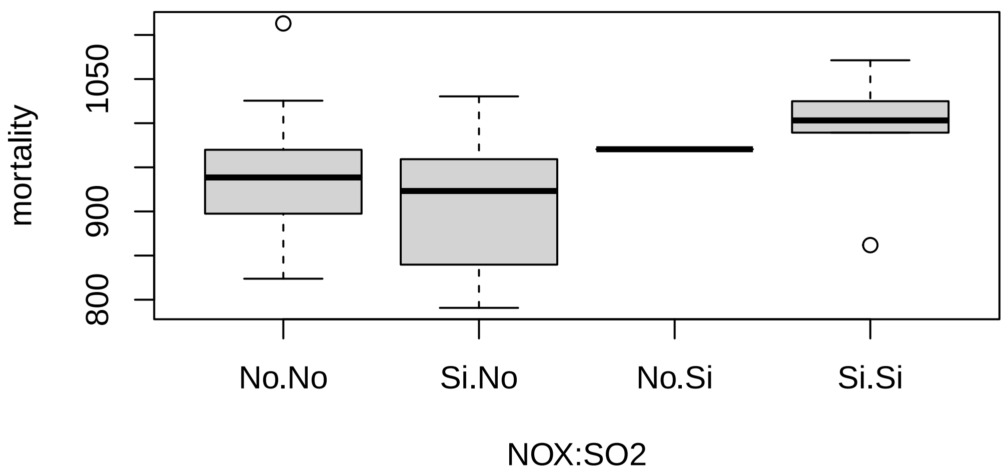
Mosaicplot for factors

```
mosaicplot(table(my.data$NOX, my.data$SO2), las=1,
       cex.axis=1, main='NOX vs SO2')
```



Boxplot of mortality with respect to factors

```r
boxplot(my.data$mortality ~ my.data$NOX*my.data$SO2, ylab='mortality', xlab='NOX:SO2')
```



it seems some interesting relationships can be inserted in the model.
Start with a model with interactions between covariates

```r
m <- lm(mortality ~ HC*NOX + HC*SO2+ NOX*SO2 +
        precipitation*NOX + humidity*NOX + precipitation*SO2 +
        humidity*SO2, data=my.data)
summary(m)

##
## Call:
## lm(formula = mortality ~ HC * NOX + HC * SO2 + NOX * SO2 + precipitation *
##     NOX + humidity * NOX + precipitation * SO2 + humidity * SO2,
##     data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -123.70  -23.30    0.94   21.46  125.23
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            829.8693    85.9000   9.661 9.64e-13 ***
## HC                       1.3385     0.5693   2.351   0.0229 *
## NOXSi                  213.5220   389.9062   0.548   0.5865
## SO2Si                -1269.4260  1324.4620  -0.958   0.3427
## precipitation            3.1382     0.9241   3.396   0.0014 **
## humidity                -0.6172     1.4865  -0.415   0.6799
## HC:NOXSi                -0.8982     0.6493  -1.383   0.1731
## HC:SO2Si                -0.1257     0.6046  -0.208   0.8362
```

```
## NOXSi:SO2Si            84.8317    65.7880   1.289   0.2035
## NOXSi:precipitation     1.3275     2.2492   0.590   0.5579
## NOXSi:humidity         -4.3185     5.9624  -0.724   0.4725
## SO2Si:precipitation     3.6625     6.8127   0.538   0.5934
## SO2Si:humidity         20.0289    19.4246   1.031   0.3078
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.62 on 47 degrees of freedom
## Multiple R-squared:  0.5132,Adjusted R-squared:  0.389
## F-statistic:  4.13 on 12 and 47 DF,  p-value: 0.0002117
```

Variable selection

```
m2 <- lm(mortality ~ HC*NOX + NOX*SO2 +precipitation*NOX +
        humidity*NOX + humidity*SO2, data=my.data)
summary(m2)

##
## Call:
## lm(formula = mortality ~ HC * NOX + NOX * SO2 + precipitation *
##     NOX + humidity * NOX + humidity * SO2, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -123.70  -25.45    1.10   23.91  125.23
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           829.8693    85.1871   9.742 4.76e-13 ***
## HC                      1.3385     0.5645   2.371  0.02171 *
## NOXSi                 -43.8206   305.5974  -0.143  0.88657
## SO2Si                -912.1087   757.5886  -1.204  0.23439
## precipitation           3.1382     0.9164   3.424  0.00125 **
## humidity               -0.6172     1.4742  -0.419  0.67727
## HC:NOXSi               -1.1201     0.5988  -1.871  0.06738 .
## NOXSi:SO2Si            68.1618    60.6394   1.124  0.26647
## NOXSi:precipitation     2.2077     2.0670   1.068  0.29072
## NOXSi:humidity         -0.1698     4.4568  -0.038  0.96977
## SO2Si:humidity         16.2730    13.2630   1.227  0.22570
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.22 on 49 degrees of freedom
## Multiple R-squared:  0.5009,Adjusted R-squared:  0.3991
```

```
## F-statistic: 4.918 on 10 and 49 DF,  p-value: 6.578e-05

m3 <- lm(mortality ~ HC*NOX + NOX*SO2+precipitation*NOX +
        humidity*SO2, data=my.data)
summary(m3)

##
## Call:
## lm(formula = mortality ~ HC * NOX + NOX * SO2 + precipitation *
##     NOX + humidity * SO2, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.674  -25.389    1.129   23.937  125.271
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            830.8328    80.5286  10.317 5.56e-14 ***
## HC                       1.3405     0.5566   2.408  0.01976 *
## NOXSi                  -55.0829    76.5293  -0.720  0.47502
## SO2Si                 -896.2700   626.9337  -1.430  0.15905
## precipitation            3.1401     0.9057   3.467  0.00109 **
## humidity                -0.6358     1.3772  -0.462  0.64634
## HC:NOXSi                -1.1229     0.5883  -1.909  0.06203 .
## NOXSi:SO2Si             68.5023    59.3748   1.154  0.25410
## NOXSi:precipitation      2.2422     1.8381   1.220  0.22825
## SO2Si:humidity          15.9946    10.9556   1.460  0.15056
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.73 on 50 degrees of freedom
## Multiple R-squared:  0.5009,Adjusted R-squared:  0.4111
## F-statistic: 5.576 on 9 and 50 DF,  p-value: 2.633e-05

m4 <- lm(mortality ~ HC*NOX + precipitation*NOX +
        humidity*SO2, data=my.data)
summary(m4)

##
## Call:
## lm(formula = mortality ~ HC * NOX + precipitation * NOX + humidity *
##     SO2, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -123.004  -27.043    4.352   24.727  126.907
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            832.1736    80.7811  10.302 4.64e-14 ***
## HC                       1.2771     0.5557   2.298  0.02569 *
## NOXSi                  -45.0271    76.2778  -0.590  0.55759
## precipitation            3.1187     0.9085   3.433  0.00119 **
## humidity                -0.6447     1.3817  -0.467  0.64275
## SO2Si                 -643.8957   589.4376  -1.092  0.27980
## HC:NOXSi                -1.0976     0.5898  -1.861  0.06851 .
## NOXSi:precipitation      2.2898     1.8436   1.242  0.21991
## humidity:SO2Si          12.3731    10.5303   1.175  0.24545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.89 on 51 degrees of freedom
## Multiple R-squared:  0.4876,Adjusted R-squared:  0.4072
## F-statistic: 6.067 on 8 and 51 DF,  p-value: 1.781e-05
```

```
m5 <- lm(mortality ~  HC*NOX + precipitation*NOX + humidity +SO2, data=my.data)
summary(m5)
```

```
##
## Call:
## lm(formula = mortality ~ HC * NOX + precipitation * NOX + humidity +
##     SO2, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.784  -27.825    3.738   29.109  125.486
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           811.4126    79.1131  10.256 4.31e-14 ***
## HC                      1.2560     0.5575   2.253  0.02850 *
## NOXSi                 -11.4834    70.9925  -0.162  0.87213
## precipitation           3.0837     0.9113   3.384  0.00137 **
## humidity               -0.2497     1.3450  -0.186  0.85345
## SO2Si                  48.0313    25.8178   1.860  0.06849 .
## HC:NOXSi               -1.2627     0.5749  -2.196  0.03254 *
## NOXSi:precipitation     1.6480     1.7673   0.932  0.35539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 48.06 on 52 degrees of freedom
## Multiple R-squared:  0.4738,Adjusted R-squared:  0.4029
## F-statistic: 6.687 on 7 and 52 DF,  p-value: 1.178e-05

m6 <- lm(mortality ~  HC*NOX + precipitation + humidity +SO2, data=my.data)
summary(m6)

##
## Call:
## lm(formula = mortality ~ HC * NOX + precipitation + humidity +
##     SO2, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.733  -25.461    4.649   29.974  120.225
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   817.8220    78.7168  10.389 2.17e-14 ***
## HC              1.3178     0.5528   2.384   0.0207 *
## NOXSi          50.6577    24.4451   2.072   0.0431 *
## precipitation   3.5568     0.7561   4.704 1.86e-05 ***
## humidity       -0.7002     1.2537  -0.558   0.5789
## SO2Si          50.9589    25.5946   1.991   0.0516 .
## HC:NOXSi       -1.3881     0.5582  -2.487   0.0161 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.01 on 53 degrees of freedom
## Multiple R-squared:  0.465,Adjusted R-squared:  0.4044
## F-statistic: 7.676 on 6 and 53 DF,  p-value: 5.843e-06

m7 <- lm(mortality ~  HC*NOX + precipitation+SO2, data=my.data)
summary(m7)

##
## Call:
## lm(formula = mortality ~ HC * NOX + precipitation + SO2, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -134.715  -27.120    3.017   28.223  117.194
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

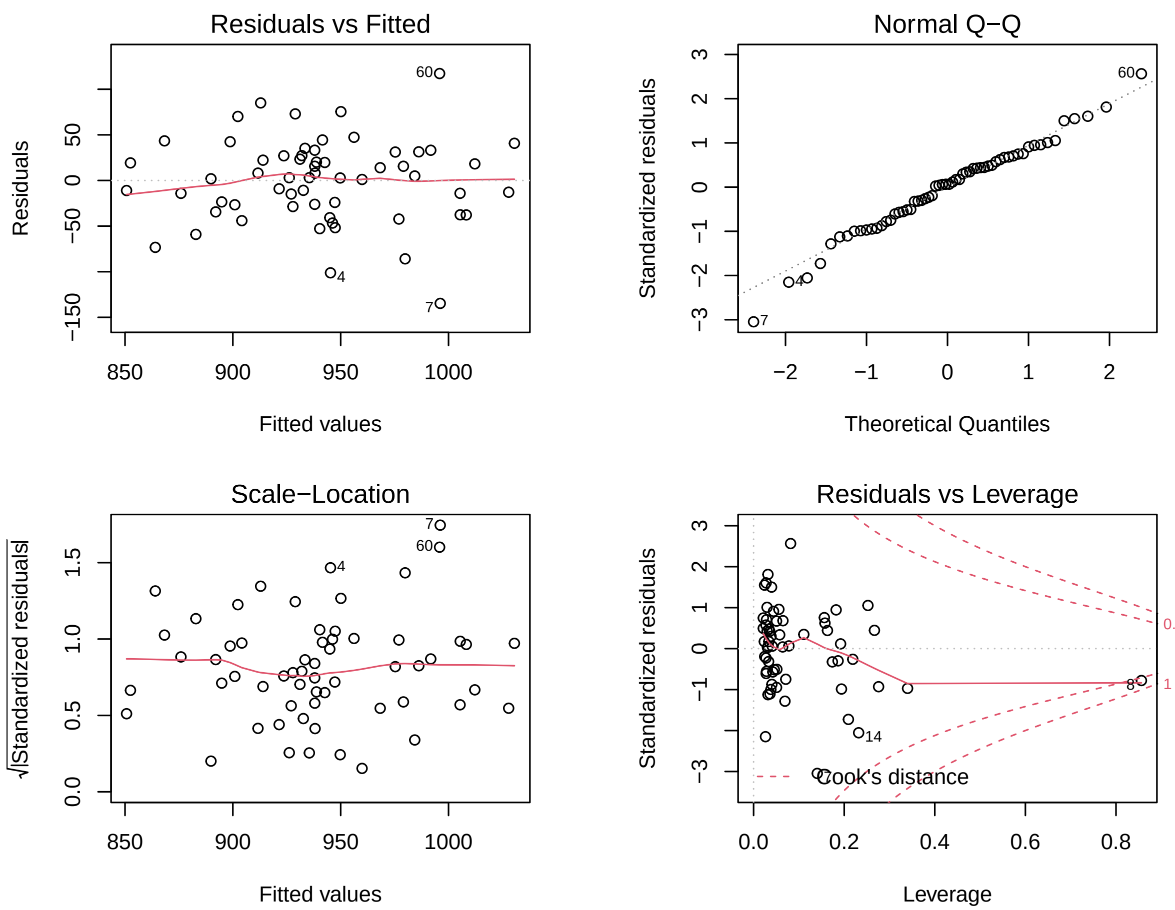
```
## (Intercept)   777.6338    31.6991  24.532  < 2e-16 ***
## HC              1.2464     0.5344   2.332   0.0234 *
## NOXSi          46.8190    23.3089   2.009   0.0496 *
## precipitation   3.5797     0.7502   4.772 1.43e-05 ***
## SO2Si          55.0097    24.3884   2.256   0.0282 *
## HC:NOXSi       -1.3126     0.5382  -2.439   0.0180 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.7 on 54 degrees of freedom
## Multiple R-squared:  0.4618,Adjusted R-squared:  0.412
## F-statistic: 9.267 on 5 and 54 DF,  p-value: 2.05e-06
```

Check the real advantage of using model `m7`

```
anova(m7, m)

## Analysis of Variance Table
##
## Model 1: mortality ~ HC * NOX + precipitation + SO2
## Model 2: mortality ~ HC * NOX + HC * SO2 + NOX * SO2 + precipitation *
##     NOX + humidity * NOX + precipitation * SO2 + humidity * SO2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     54 122857
## 2     47 111114  7     11744 0.7096  0.664
```

Residual analysis of model `m7`

```
par(mfrow=c(2,2))
plot(m7)
```



The graphical analysis seems satisfactory: there are no anomalous values, neither trend in residuals or deviations from normality. We can check the need for a polynomial associated to `precipitation`

```
m8 <- update(m7, .~. + I(precipitation^2))
summary(m8)

##
## Call:
## lm(formula = mortality ~ HC + NOX + precipitation + SO2 + I(precipitation^2) +
##     HC:NOX, data = my.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.486  -29.504    3.221   27.981  134.836
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         685.69395   69.28985   9.896 1.21e-13 ***
## HC                    1.19891    0.52944   2.265   0.0277 *
```

```
## NOXSi                 54.73688   23.65753   2.314   0.0246 *
## precipitation          8.94696    3.68284   2.429   0.0185 *
## SO2Si                 41.24401   25.83250   1.597   0.1163
## I(precipitation^2)  -0.07359    0.04946  -1.488   0.1427
## HC:NOXSi             -1.19035    0.53853  -2.210   0.0314 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.17 on 53 degrees of freedom
## Multiple R-squared:  0.4834,Adjusted R-squared:  0.4249
## F-statistic: 8.265 on 6 and 53 DF,  p-value: 2.48e-06

## it seems the polynomial is not useful
```

Model `m7` suggests that

- the mortality rate increases with precipitation
- the mortality rate increases with SO2
- for a small level of NOX, the mortality rate increases with HC, and viceversa.

Try to see if the model can be improved using splines.

```
library(splines)
sp.HC <- smooth.spline(x=my.data$HC, y=my.data$mortality, cv=TRUE)

## Warning in smooth.spline(x = my.data$HC, y = my.data$mortality, cv = TRUE):
cross-validation with non-unique 'x' values seems doubtful

## df=33
sp.precipitation <- smooth.spline(x=my.data$precipitation,
        y=my.data$mortality, cv=TRUE)

## Warning in smooth.spline(x = my.data$precipitation, y = my.data$mortality,
:  cross-validation with non-unique 'x' values seems doubtful

## df=2
library(gam)

## Loading required package:  foreach
## Loaded gam 1.20

m.gam <- gam(mortality ~ s(HC, 33)*NOX + SO2 +
        s(precipitation,2), data=my.data)
summary(m.gam)
```

```
##
## Call: gam(formula = mortality ~ s(HC, 33) * NOX + SO2 + s(precipitation,
##     2), data = my.data)
## Deviance Residuals:
##        Min         1Q     Median         3Q        Max
## -1.049e+02 -7.880e+00  2.680e-07  5.141e+00  9.413e+01
##
## (Dispersion Parameter for gaussian family taken to be 2371.666)
##
##     Null Deviance: 228275.4 on 59 degrees of freedom
## Residual Deviance: 49804.92 on 21 degrees of freedom
## AIC: 653.5642
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                       Df Sum Sq Mean Sq F value    Pr(>F)
## s(HC, 33)              1   8253    8253  3.4799 0.0761510 .
## NOX                    1   1722    1722  0.7261 0.4037529
## SO2                    1  36657   36657 15.4560 0.0007653 ***
## s(precipitation, 2)    1  51391   51391 21.6689 0.0001359 ***
## s(HC, 33):NOX          1  15286   15286  6.4452 0.0191057 *
## Residuals             21  49805    2372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                       Npar Df  Npar F Pr(F)
## (Intercept)
## s(HC, 33)                  32 0.98998 0.521
## NOX
## SO2
## s(precipitation, 2)         1 0.95727 0.339
## s(HC, 33):NOX
```

We do not need splines for `HC`.

```
m2.gam <- gam(mortality ~ HC*NOX + SO2 + s(precipitation,2), data=my.data)
summary(m2.gam)

##
## Call: gam(formula = mortality ~ HC * NOX + SO2 + s(precipitation, 2),
##     data = my.data)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

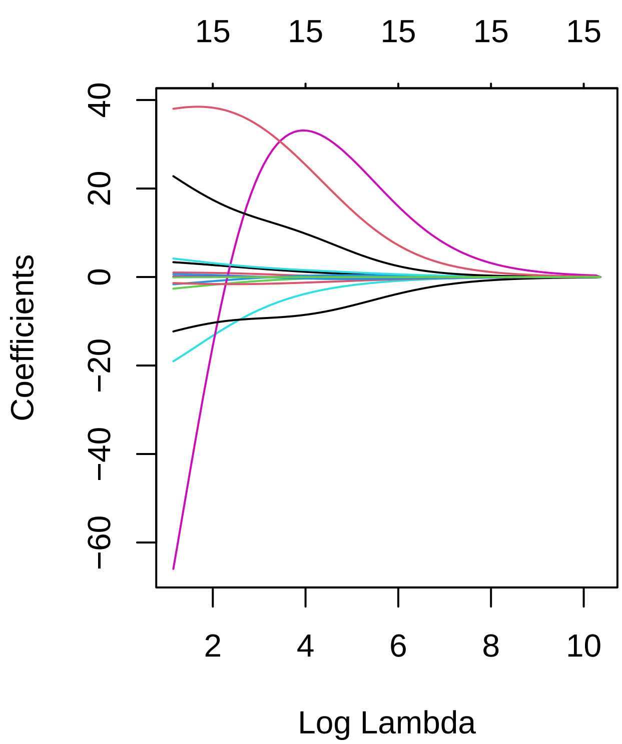
```
## -118.557  -29.873    2.896   30.243  124.691
##
## (Dispersion Parameter for gaussian family taken to be 2237.168)
##
##     Null Deviance: 228275.4 on 59 degrees of freedom
## Residual Deviance: 118569.8 on 53 degrees of freedom
## AIC: 641.6074
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                     Df Sum Sq Mean Sq F value    Pr(>F)
## HC                   1   4424    4424  1.9776   0.16548
## NOX                  1  10645   10645  4.7581   0.03362 *
## SO2                  1  19817   19817  8.8579   0.00439 **
## s(precipitation, 2)  1  58131   58131 25.9841 4.708e-06 ***
## HC:NOX               1  12322   12322  5.5078   0.02270 *
## Residuals           53 118570    2237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                     Npar Df Npar F Pr(F)
## (Intercept)
## HC
## NOX
## SO2
## s(precipitation, 2)       1 1.9165 0.172
## HC:NOX
```

We dot need splines for `precipitation`.

SECOND QUESTION.
Consider all the variables in the dataset. Construct the most appropriate model for the purpose of the analysis. Which variables are associated to the mortality rate?
Given the large number of covariates, we can use regularization techniques. And we can exploits some of the previous findings, as, for example, we can insert in the analysis the interaction between `HC` and `NOX`.

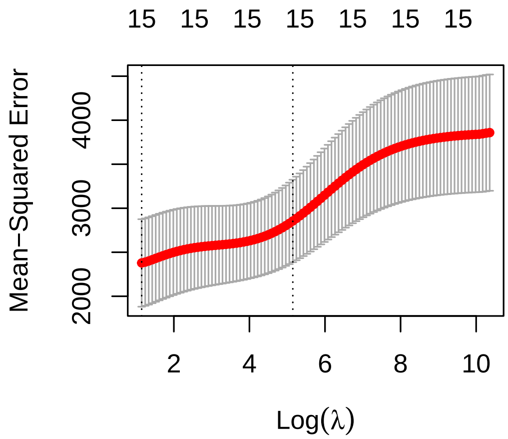
```
library(glmnet)
m.glm <- glm(mortality ~ .+HC:NOX, data=pollution)
X <- model.matrix(m.glm)[,-1]
m.ridge <- glmnet(x=X, y=pollution$mortality, alpha=0)
```

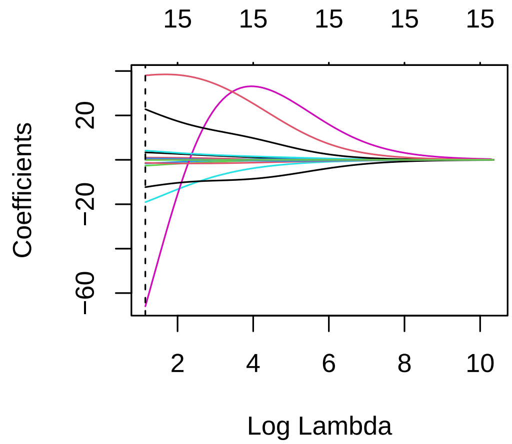
```
plot(m.ridge, xvar='lambda')
```




```
set.seed(222)
m.ridge.cv <- cv.glmnet(x=X, y=pollution$mortality, alpha=0)
```

```
par(mfrow=c(1,2))
plot(m.ridge.cv)
plot(m.ridge, xvar='lambda')
abline(v=log(m.ridge.cv$lambda.min), lty=2)
```



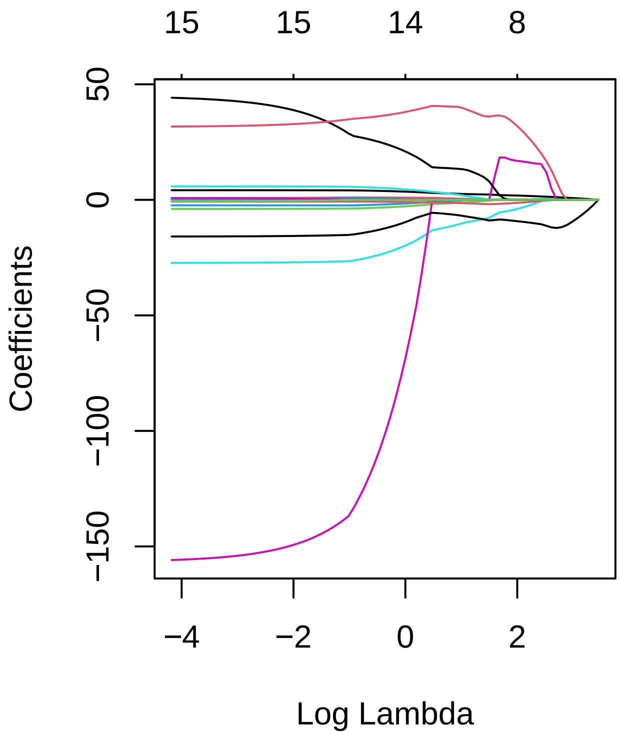



```
m.ridge.min <- glmnet(x=X, y=pollution$mortality, alpha=0,
        lambda=m.ridge.cv$lambda.min)
cbind(coef(m.glm), coef(m.ridge.min))

## 16 x 2 sparse Matrix of class "dgCMatrix"
##                                          s0
## (Intercept)    1.901397e+03  1.463837e+03
## precipitation  4.160419e+00  3.355093e+00
## humidity       7.525489e-01  1.014058e+00
## Jan.temp      -3.944194e+00 -2.625912e+00
## July.temp     -2.363294e+00 -1.633865e+00
## over65        -2.722341e+01 -1.906860e+01
## house         -1.560127e+02 -6.612185e+01
## education     -1.583265e+01 -1.231362e+01
## comfort       -8.118738e-01 -1.357872e+00
## density        5.243647e-03  8.017681e-03
## office         2.298015e-01  6.291360e-01
## poor           5.801315e+00  4.177718e+00
## HC             7.736219e-01  1.324555e-01
## NOXSi          4.526420e+01  2.289480e+01
## SO2Si          3.148585e+01  3.798483e+01
## HC:NOXSi      -6.893134e-01 -7.057876e-02
```

Ridge regression confirms the association of `NOX` and `SO2` on the mortality rate. We can use lasso for variable selection.

```
m.lasso <- glmnet(x=X, y=pollution$mortality, alpha=1)
```

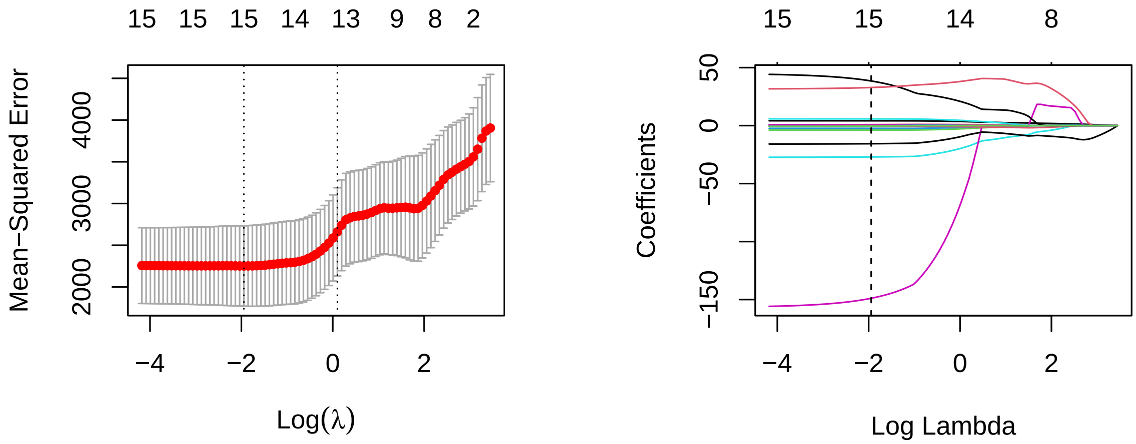
```
plot(m.lasso, xvar='lambda')
```




```
set.seed(222)
m.lasso.cv <- cv.glmnet(x=X, y=pollution$mortality, alpha=1)
m.lasso.cv

##
## Call:  cv.glmnet(x = X, y = pollution$mortality, alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min 0.1429    59    2252 482.3      15
## 1se 1.1067    37    2660 529.0      14
```

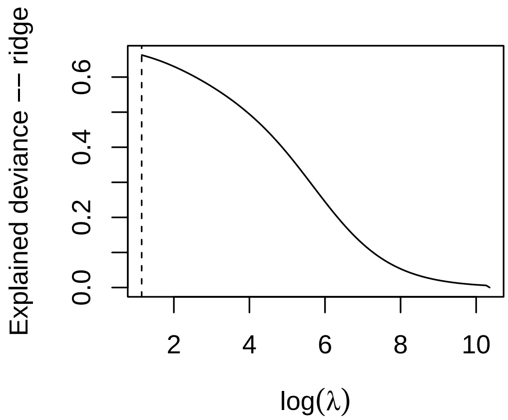
There is no relevant selection.

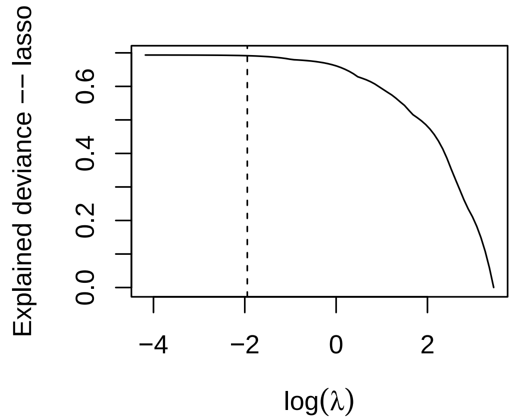
```
par(mfrow=c(1,2))
plot(m.lasso.cv)
plot(m.lasso, xvar='lambda')
abline(v=log(m.lasso.cv$lambda.min), lty=2)
```




```
m.lasso.min <- glmnet(x=X, y=pollution$mortality, alpha=1,
        lambda=m.lasso.cv$lambda.min)
cbind(coef(m.glm), coef(m.ridge.min), coef(m.lasso.min))

## 16 x 3 sparse Matrix of class "dgCMatrix"
##                                          s0            s0
## (Intercept)    1.901397e+03  1.463837e+03  1.857920e+03
## precipitation  4.160419e+00  3.355093e+00  4.140763e+00
## humidity       7.525489e-01  1.014058e+00  8.675517e-01
## Jan.temp      -3.944194e+00 -2.625912e+00 -3.888083e+00
## July.temp     -2.363294e+00 -1.633865e+00 -2.393603e+00
## over65        -2.722341e+01 -1.906860e+01 -2.702569e+01
## house         -1.560127e+02 -6.612185e+01 -1.488006e+02
## education     -1.583265e+01 -1.231362e+01 -1.562861e+01
## comfort       -8.118738e-01 -1.357872e+00 -7.706245e-01
## density        5.243647e-03  8.017681e-03  5.882628e-03
## office         2.298015e-01  6.291360e-01  4.374290e-01
## poor           5.801315e+00  4.177718e+00  5.758559e+00
## HC             7.736219e-01  1.324555e-01  5.053675e-01
## NOXSi          4.526420e+01  2.289480e+01  3.846837e+01
## SO2Si          3.148585e+01  3.798483e+01  3.284600e+01
## HC:NOXSi      -6.893134e-01 -7.057876e-02 -4.187426e-01
```

```
par(mfrow=c(1,2))
plot(log(m.ridge$lambda), m.ridge$dev.ratio, type='l',
        xlab=expression(log(lambda)), ylab='Explained deviance -- ridge')
abline(v=log(m.ridge.cv$lambda.min), lty=2)
plot(log(m.lasso$lambda), m.lasso$dev.ratio, type='l',
        xlab=expression(log(lambda)), ylab='Explained deviance -- lasso')
abline(v=log(m.lasso.cv$lambda.min), lty=2)
```





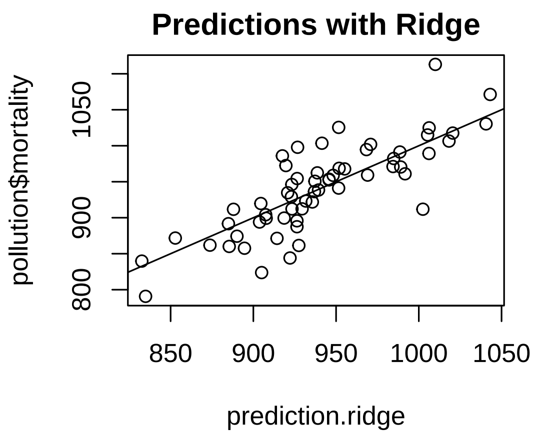
Compare the results from ridge regression and lasso.

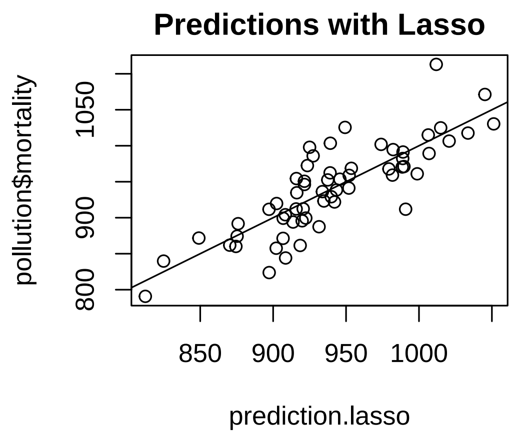
```
m.ridge$dev.ratio[m.ridge$lambda==m.ridge.cv$lambda.min]

## [1] 0.662835

m.lasso$dev.ratio[m.lasso$lambda==m.lasso.cv$lambda.min]

## [1] 0.691473
```

Lasso is preferable in terms of explained deviance.

```
prediction.ridge <- predict(m.ridge.min, newx=X)
prediction.lasso <- predict(m.lasso.min, newx=X)
```

```
par(mfrow=c(1,2))
plot(prediction.ridge, pollution$mortality,
        main='Predictions with Ridge')
abline(0,1)
plot(prediction.lasso, pollution$mortality,
        main='Predictions with Lasso')
abline(0,1)
```






```
min(m.ridge.cv$cvm)

## [1] 2378.412

min(m.lasso.cv$cvm)

## [1] 2251.617
```

We prefer lasso.
Compare the results with those from the linear model.

```
m.glm <- glm(mortality ~ .+HC:NOX, data=pollution)
library(boot)
set.seed(222)
m.glm.cv <- cv.glm(pollution, m.glm)
m.glm.cv$delta

## [1] 2516.973 2502.145
```

Lasso is preferable in terms of mean squared error. But since there is no substantial variable selection, it does not seem to be so interesting.

Try an automatic variable selection, using a `forward` procedure, adding the interactions suggested by the model of the first question.

```
library(leaps)
m.forward <- regsubsets(mortality ~ . + HC:NOX, data=pollution, nvmax=19,
        method='forward')
summary(m.forward)

## Subset selection object
## Call: regsubsets.formula(mortality ~ . + HC:NOX, data = pollution,
```

```
##     nvmax = 19, method = "forward")
## 15 Variables  (and intercept)
##               Forced in Forced out
## precipitation     FALSE      FALSE
## humidity          FALSE      FALSE
## Jan.temp          FALSE      FALSE
## July.temp         FALSE      FALSE
## over65            FALSE      FALSE
## house             FALSE      FALSE
## education         FALSE      FALSE
## comfort           FALSE      FALSE
## density           FALSE      FALSE
## office            FALSE      FALSE
## poor              FALSE      FALSE
## HC                FALSE      FALSE
## NOXSi             FALSE      FALSE
## SO2Si             FALSE      FALSE
## HC:NOXSi          FALSE      FALSE
## 1 subsets of each size up to 15
## Selection Algorithm: forward
##           precipitation humidity Jan.temp July.temp over65 house education comfort
## 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 ) "*"           "*"      "*"      "*"       "*"    "*"   "*"       " "
## 14  ( 1 ) "*"           "*"      "*"      "*"       "*"    "*"   "*"       "*"
## 15  ( 1 ) "*"           "*"      "*"      "*"       "*"    "*"   "*"       "*"
##           density office poor HC  NOXSi SO2Si HC:NOXSi
## 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 ) "*"      " "    "*"  "*" "*"   "*"   "*"
## 14  ( 1 ) "*"      " "    "*"  "*" "*"   "*"   "*"
## 15  ( 1 ) "*"      "*"    "*"  "*" "*"   "*"   "*"
```

RSS criterion

```
summary(m.forward)$rss

##  [1] 168671.67 148526.00 129820.13 115124.40 108617.31 102580.15  94046.62  85370.37
##  [9]  79775.09  75398.52  74454.73  70875.16  70282.13  69978.05  69954.69

which.min(summary(m.forward)$rss)

## [1] 15

coef(m.forward, 15)

##   (Intercept) precipitation      humidity      Jan.temp     July.temp        over65
##  1.901397e+03  4.160419e+00  7.525489e-01 -3.944194e+00 -2.363294e+00 -2.722341e+01
##         house     education       comfort       density        office          poor
## -1.560127e+02 -1.583265e+01 -8.118738e-01  5.243647e-03  2.298015e-01  5.801315e+00
##            HC         NOXSi         SO2Si      HC:NOXSi
##  7.736219e-01  4.526420e+01  3.148585e+01 -6.893134e-01
```

Adjusted R^2 criterion

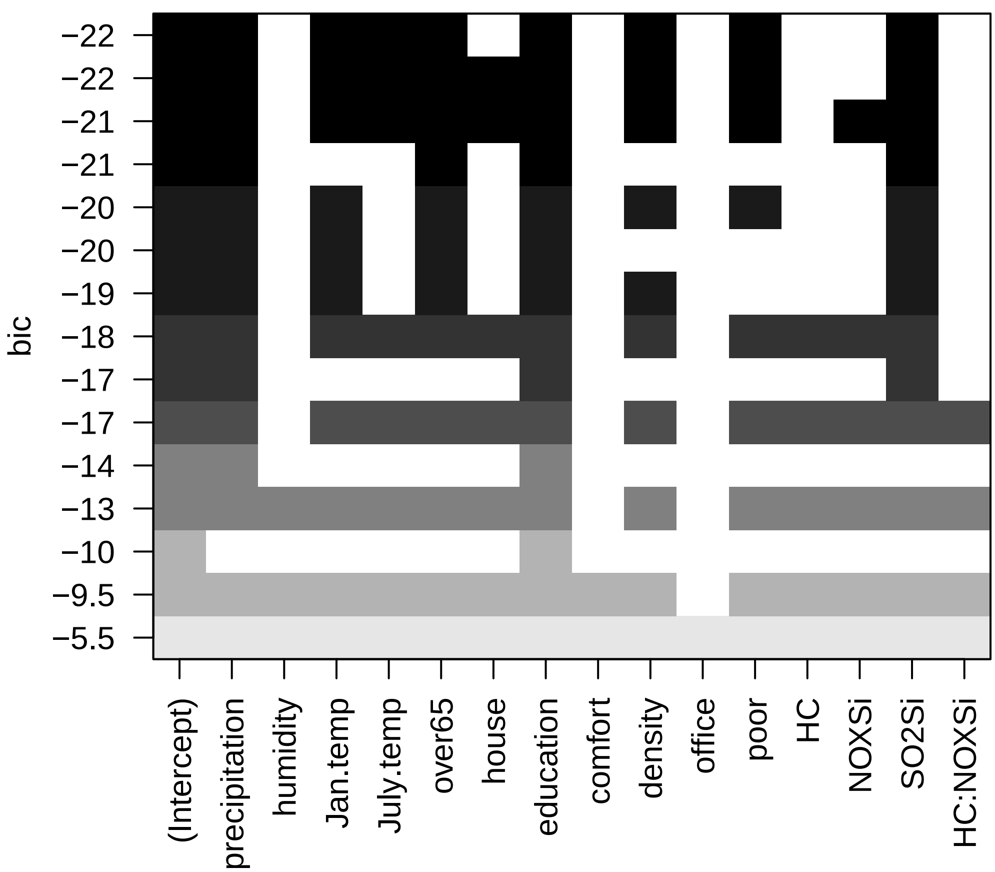
```
which.max(summary(m.forward)$adjr2)

## [1] 12

coef(m.forward, 12)

##   (Intercept) precipitation      Jan.temp     July.temp        over65         house
##  1.941417e+03  4.182912e+00 -3.951313e+00 -3.006279e+00 -2.743225e+01 -1.595139e+02
##     education       density          poor            HC         NOXSi         SO2Si
## -1.595196e+01  4.670546e-03  6.650527e+00  8.299267e-01  4.663277e+01  3.012948e+01
##      HC:NOXSi
## -7.676111e-01
```

BIC

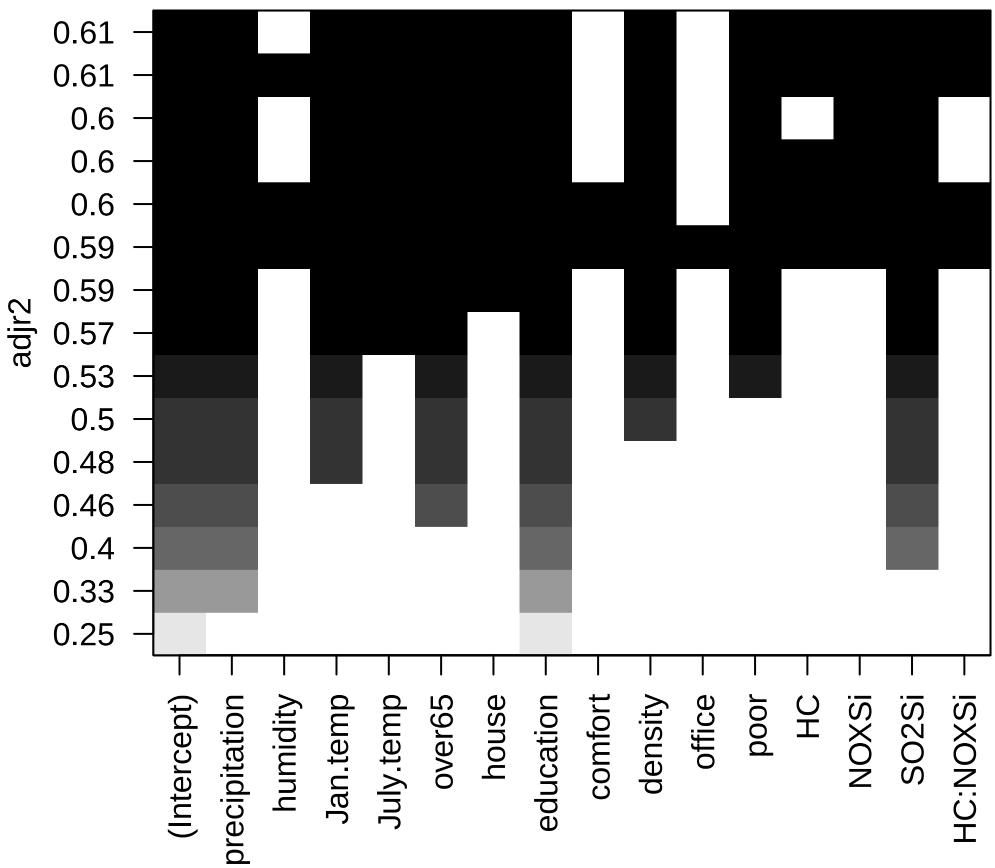
```r
which.min(summary(m.forward)$bic)

## [1] 8

coef(m.forward, 8)

##   (Intercept) precipitation      Jan.temp     July.temp        over65     education
## 1218.58525725    3.30033252   -2.27918048   -3.81679604  -17.45698605   -2.79524322
##       density          poor         SO2Si
##    0.01093273    6.67409276   56.13858512
```

Graphical evaluation of the automatic selection

```r
plot(m.forward)
```



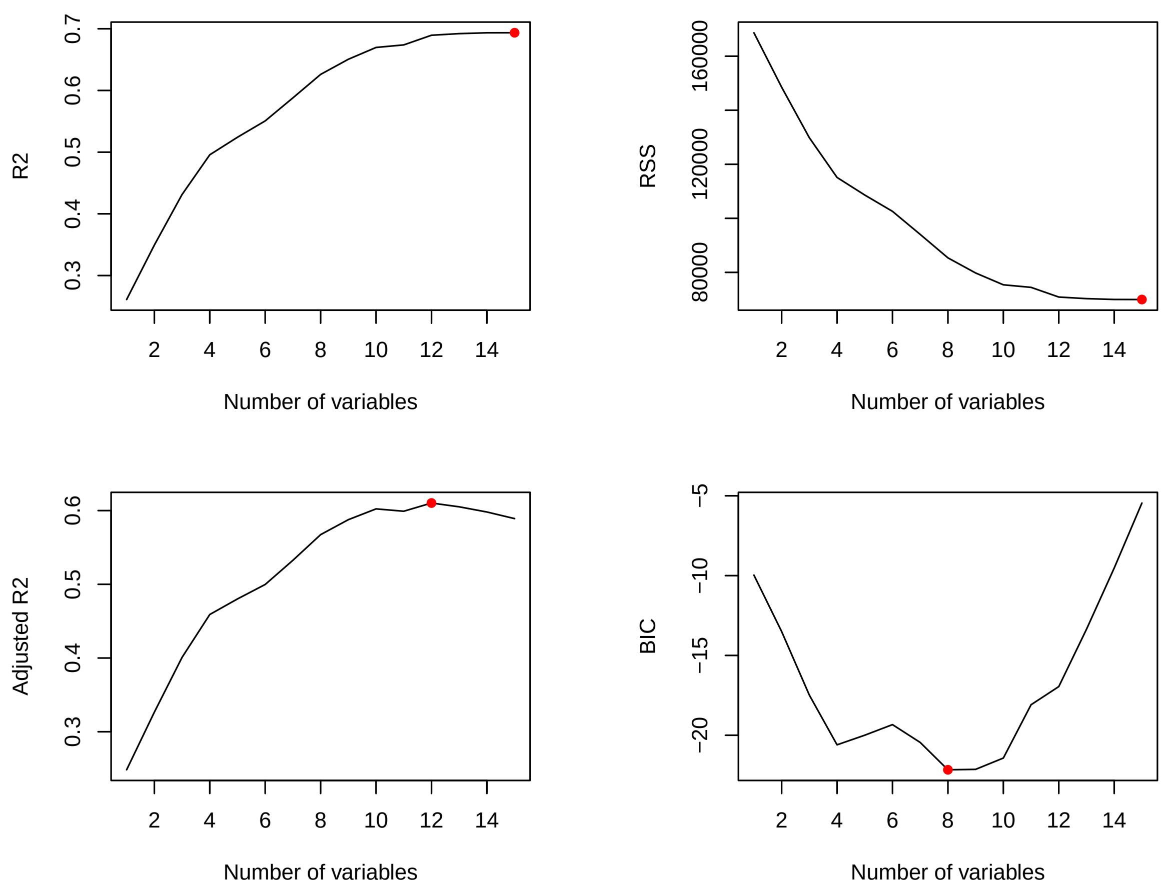
```r
plot(m.forward, scale='adjr2')
```



Selection based on BIC suggests an association between mortality with `precipitation`, `poor`, `SO2`. Selection based on adjusted R^2 also adds `density`, `HC` and `NOX`.
Model ranking

```r
par(mfrow=c(2,2))
## R2
plot(summary(m.forward)$rsq, xlab='Number of variables', ylab='R2', type='l')
## add the indication of the preferable model
points(which.max(summary(m.forward)$rsq),
summary(m.forward)$rsq[which.max(summary(m.forward)$rsq)], col='red', pch=16)
## RSS
plot(summary(m.forward)$rss, xlab='Number of variables', ylab='RSS', type='l' )
```

```r
points(which.min(summary(m.forward)$rss),
summary(m.forward)$rss[which.min(summary(m.forward)$rss)], col='red', pch=16)
## adjusted R2
plot(summary(m.forward)$adjr2, xlab='Number of variables',
        ylab='Adjusted R2', type='l')
points(which.max(summary(m.forward)$adjr2),
        summary(m.forward)$adjr2[which.max(summary(m.forward)$adjr2)],
        col='red', pch=16)
## BIC
plot(summary(m.forward)$bic, xlab='Number of variables', ylab='BIC', type='l')
points(which.min(summary(m.forward)$bic),
summary(m.forward)$bic[which.min(summary(m.forward)$bic)], col='red', pch=16)
```



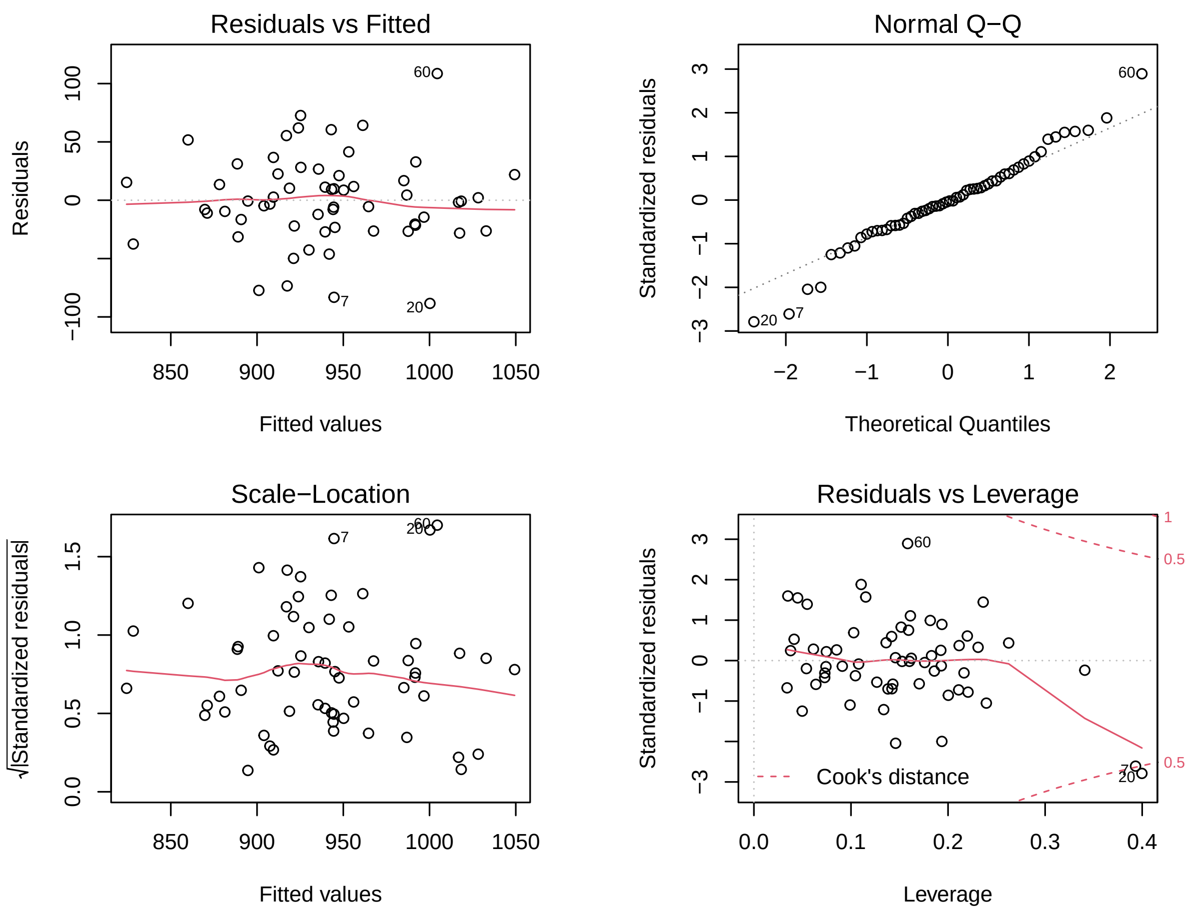
Evaluate the regression model chosen by BIC.

```r
m.bic <- lm(mortality ~ precipitation +Jan.temp + July.temp +
        over65+education+density+poor +SO2, data=pollution)
summary(m.bic)

##
```

```
## Call:
## lm(formula = mortality ~ precipitation + Jan.temp + July.temp +
##     over65 + education + density + poor + SO2, data = pollution)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.411 -22.365  -1.283  21.336 108.597
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.219e+03  1.721e+02   7.082 4.03e-09 ***
## precipitation  3.300e+00  7.701e-01   4.286 8.09e-05 ***
## Jan.temp      -2.279e+00  7.475e-01  -3.049 0.003635 **
## July.temp     -3.817e+00  1.676e+00  -2.277 0.027035 *
## over65        -1.746e+01  4.665e+00  -3.742 0.000464 ***
## education     -2.795e+00  9.059e+00  -0.309 0.758910
## density        1.093e-02  4.137e-03   2.642 0.010903 *
## poor           6.674e+00  2.336e+00   2.857 0.006172 **
## SO2Si          5.614e+01  1.808e+01   3.105 0.003107 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.91 on 51 degrees of freedom
## Multiple R-squared:  0.626,Adjusted R-squared:  0.5674
## F-statistic: 10.67 on 8 and 51 DF,  p-value: 1.165e-08
```

Check residuals

```
par(mfrow=c(2,2))
plot(m.bic)
```



We can check whether the principal component regression is useful.

```
library(pls)
set.seed(222)
m.pcr <- pcr(mortality ~ .+HC:NOX, scale=TRUE, validation='CV', data=pollution)
```

```
summary(m.pcr)

## Data:   X dimension: 60 15
##  Y dimension: 60 1
## Fit method: svdpc
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## CV           62.73    58.26    55.34    52.39    52.72    53.69    52.80    52.27
## adjCV        62.73    58.02    54.94    52.07    52.45    53.41    52.34    51.94
##        8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
```

```
## CV        53.91    51.89     58.56     58.97     63.07     69.82     50.54     49.53
## adjCV     53.57    51.99     57.43     57.80     61.76     68.22     49.63     48.63
##
## TRAINING: % variance explained
##            1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X            29.66    46.78    60.78    69.94    78.22     84.3    88.64    92.08
## mortality    18.89    27.21    39.73    39.75    39.77     46.8    47.17    48.57
##            9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X            94.31     96.50     97.79     98.84     99.49     99.97    100.00
## mortality    50.31     55.99     56.53     56.93     57.17     68.34     69.36
```

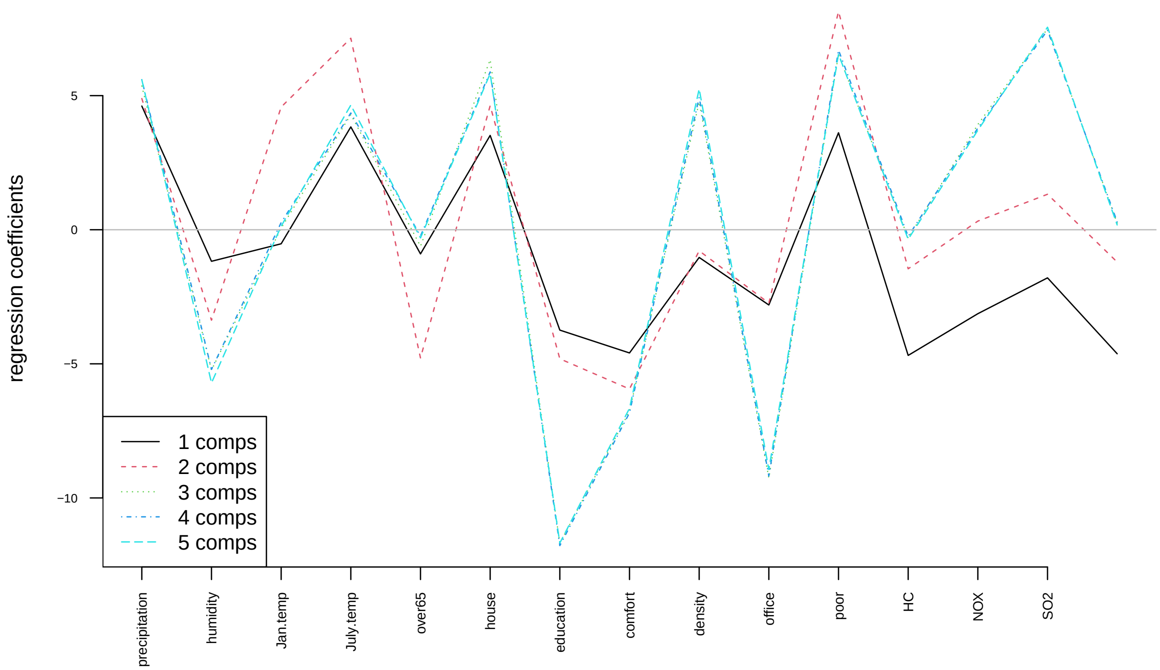
How many principal components?

```
selectNcomp(m.pcr, method='onesigma', ncomp=15)

## [1] 2
```

How much variance is explained by the two first principal components?

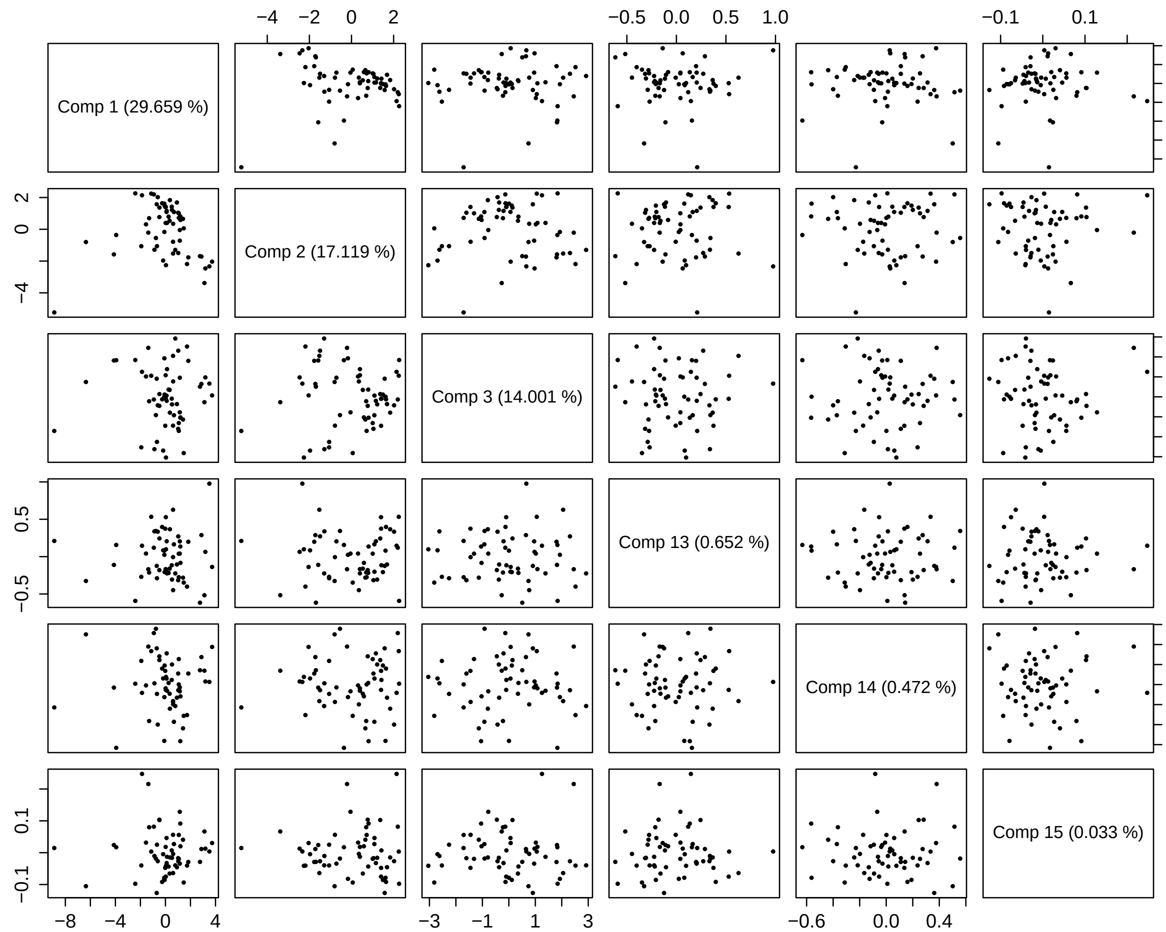
```
explvar(m.pcr)

##     Comp 1     Comp 2     Comp 3     Comp 4     Comp 5     Comp 6     Comp 7     Comp
## 29.6592716 17.1187196 14.0014794  9.1564337  8.2889907  6.0750224  4.3419359  3.43718
##     Comp 9    Comp 10    Comp 11    Comp 12    Comp 13    Comp 14    Comp 15
##  2.2271121  2.1930932  1.2876528  1.0556113  0.6520201  0.4721439  0.0333317
```

```
 coefplot(m.pcr, ncomp=1:5, legendpos='bottomleft', main='',
 xlab='', ylab='regression coefficients', axes=FALSE)
 axis(1, at=1:14, labels=colnames(pollution)[-1], las=2, cex=0.4, cex.axis=0.7)
axis(2, las=2, cex.axis=0.6)
```



On the basis of the first PC, mortality is associated to the temperature in July, to the features of the house (number of persons and comfort), to the poverty level of the area. The second PC gives weight also to the presence of pollutants `NOX` and `SO2`.
Check for the presence of groups or outliers using the scoreplot about some of the principal components.

```
scoreplot(m.pcr, comps=c(1,2,3,13,14,15), cex=0.5, cex.lab=1.4, cex.axis=1.4, pch=19)
```



There is no evidence of problems.