Linear regression with R

Data Mining
Master Degree in Computer Science
University of Padova
a.y. 2017/2018

Annamaria Guolo

Start the R session and make sure there are no objects in the workspace

```
ls()
```

Eventually remove existing objects

```
rm(list=ls())
```

1 Boston Dataset

Upload the Boston dataset, that is inside library (or package) MASS.

```
library(MASS)
data(Boston)
```

The dataset contains information about 506 houses in the area of Boston. For other information about the dataset we can use the help online

```
?Boston
```

or

```
help(Boston)
```

First look at the variables...look at the information about the first 3 houses. We can access them through the *square brackets*, that are used to access the elements of vectors, matrices, datasets.

```
Boston[1:3,]

##       crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03 34.7
```

Dimension of the dataset

```
dim(Boston)

## [1] 506  14
```

For convenience we can assign the information about about the number of houses `n` to an object

```
n <- nrow(Boston)
n

## [1] 506
```

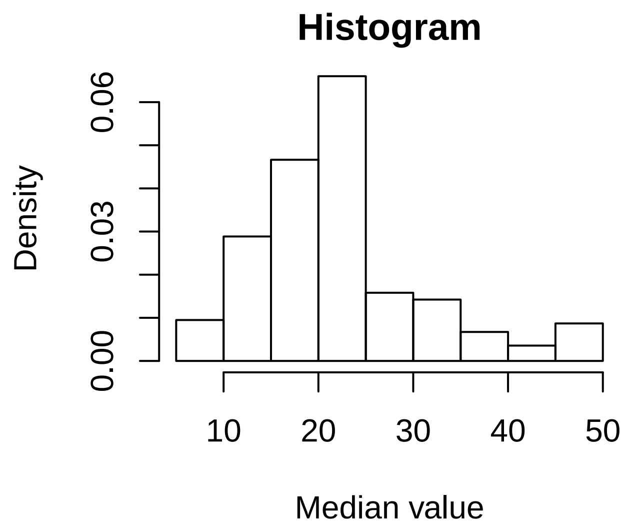
Consider only variables

- `medve`: median values of the houses (1000 $)
- `lstat`: lower status of the population (percent)

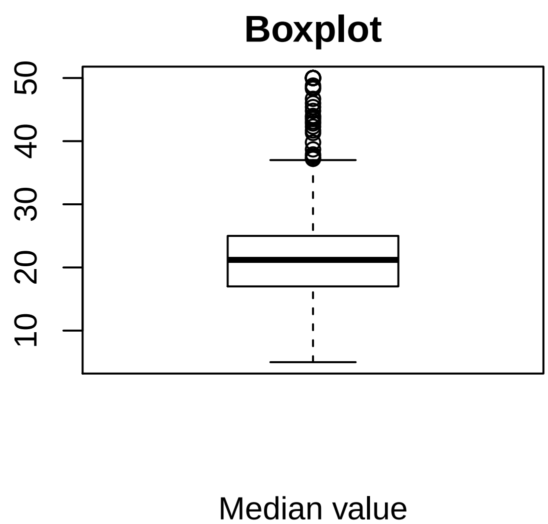
We want to evaluate whether and how the value `medve` can be predicted using `lstat`. Start with some characteristics about the value

```
summary(Boston$medv)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   17.02   21.20   22.53   25.00   50.00
```

```
## histogram of the distribution
## xlab=graphical option to assign a label to the x-axis
## main: title of the graph
hist(Boston$medv, prob=TRUE, xlab='Median value', main='Histogram')
```

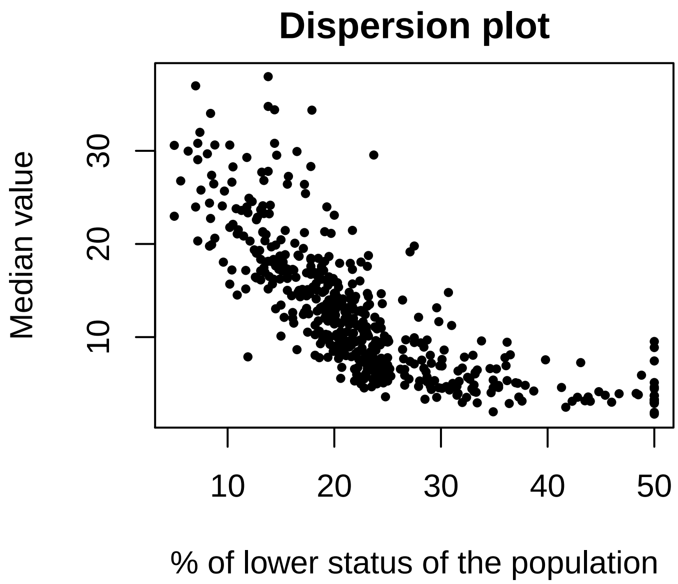



```
## boxplot of the distribution
boxplot(Boston$medv, xlab='Median value', main='Boxplot' )
```



Graphical evaluation of the relationship between `medv` and `lstat`

```
## dispersion plot
## pch=19 type of point;
## cex=0.5 reducing the dimension of the points (defaulting to 1)
## ylab: analogous to xlab but relative to y-axis
plot(Boston$medv, Boston$lstat, main='Dispersion plot',
        xlab='% of lower status of the population',
        ylab='Median value', pch=19, cex=0.5)
```



The plot shows an inverse relationship between the variables.
Correlation between the variables

```
cor(Boston$medv, Boston$lstat)

## [1] -0.7376627
```

What can we conclude?
Try to estimate a simple linear regression model

$$\texttt{medv} = \beta_0 + \beta_1 \texttt{lstat} + \varepsilon$$

Construct it step by step

```
beta1 <- cov(Boston$medv, Boston$lstat)/var(Boston$lstat)
beta1

## [1] -0.9500494

beta0 <- mean(Boston$medv) - beta1* mean(Boston$lstat)
beta0

## [1] 34.55384
```

Note that the variance of `lstat`

```
mean(Boston$lstat^2)-(mean(Boston$lstat)^2)

## [1] 50.89398
```

is equal to

```
var(Boston$lstat)*(n-1)/n

## [1] 50.89398
```

as R computes variances and covariances by dividing them by $n-1$ instead of n in order to provide unbiased estimates (it works at a sample level, not at the population level). The R function needed to fit linear regression models is `lm()`

```
model <- lm(medv ~ lstat, data=Boston)
```

The output provides an object (`model`) with many details.

```
## basic information: estimate of the coefficients
model

##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

Much of the information can be visualised through command `summary`

```
summary(model)

##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441,Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

The output contains:

- information about residuals
- estimate, standard error, significance test on the parameters
- information about the accuracy of the model
- test F for the significance of all the parameters

How can we comment on the output?
Other information in `model`

```
names(model)

##  [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
##  [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
## [11] "terms"         "model"
```

How can we access the components?

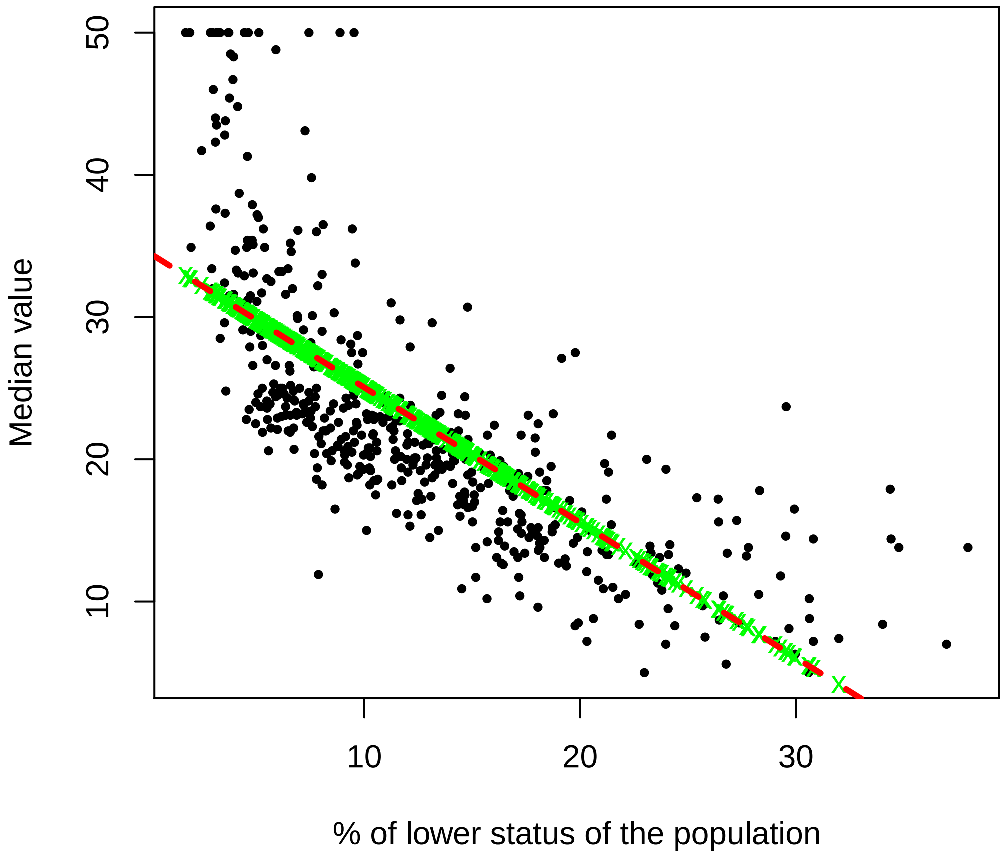
```
model$coefficients

## (Intercept)       lstat
##  34.5538409  -0.9500494
```

Model-based estimated median values

```
est.values <- fitted(model)
```

Observations, model-based estimated values and linear regression fit

```
plot(Boston$lstat, Boston$medv, pch=19, cex=0.5,
        xlab='% of lower status of the population', ylab='Median value')
## add on the estimated values
points(Boston$lstat, est.values, pch='x', col='green')
## add on the least squares regression line
abline(coef(model)[1], coef(model)[2], lty=2, col='red', lwd=3)
## equal to
## abline(beta0, beta1, lty=2, col='red')
## lty=2 specifies dashed line (defaulting to lty=1 solid line)
## lwd=3 specifies line width (defaulting to lwd=1)
```

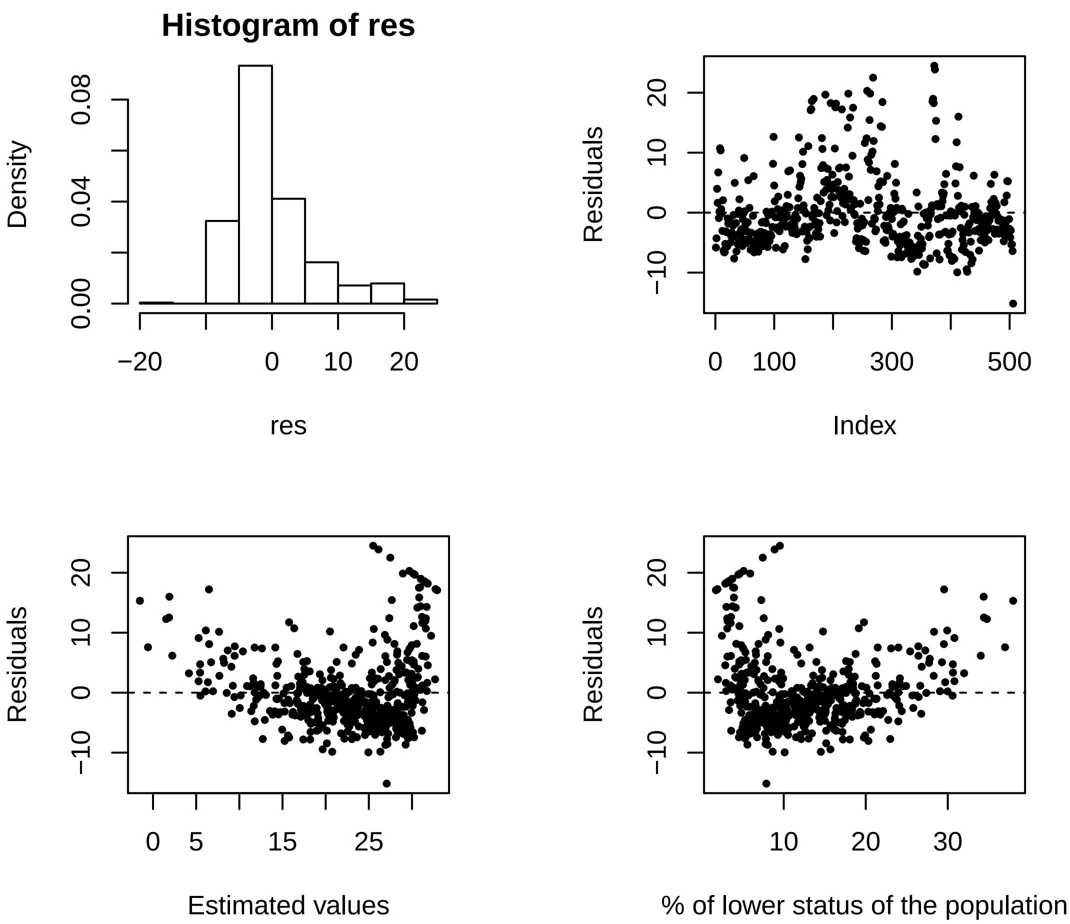


Residuals

```
res <- residuals(model)
```

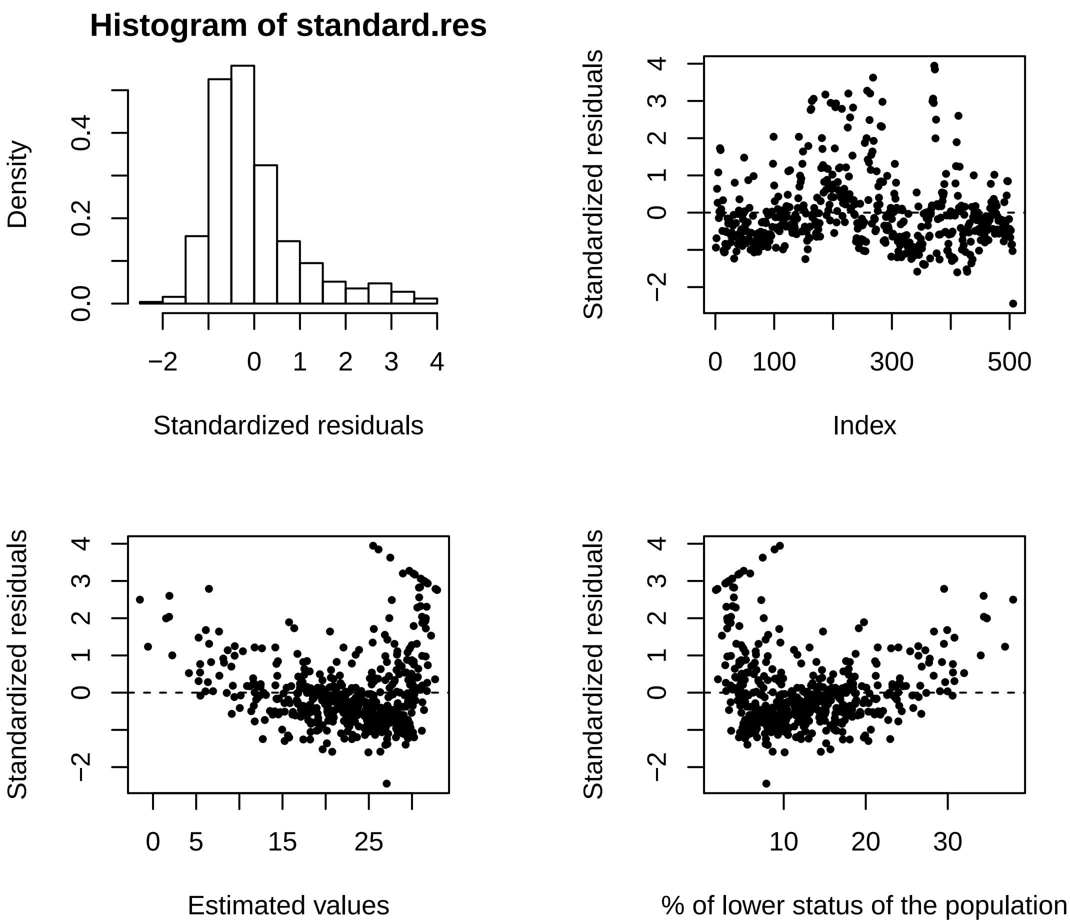
Graphical evaluation of the residuals

```
# subdivide the window into 4 parts, 2 rows and 2 columns
par(mfrow=c(2,2))
hist(res, prob=TRUE)
plot(res, pch=19, cex=0.5, ylab='Residuals')
## add on the line parallel to the x-axis
abline(h=0, lty=2)
plot(est.values, res, pch=19, cex=0.5, xlab='Estimated values',
        ylab='Residuals')
abline(h=0, lty=2)
plot(Boston$lstat, res, ylab='Residuals',
        xlab='% of lower status of the population', pch=19, cex=0.5)
abline(h=0, lty=2)
```



Graphical evaluation of the standardized residuals

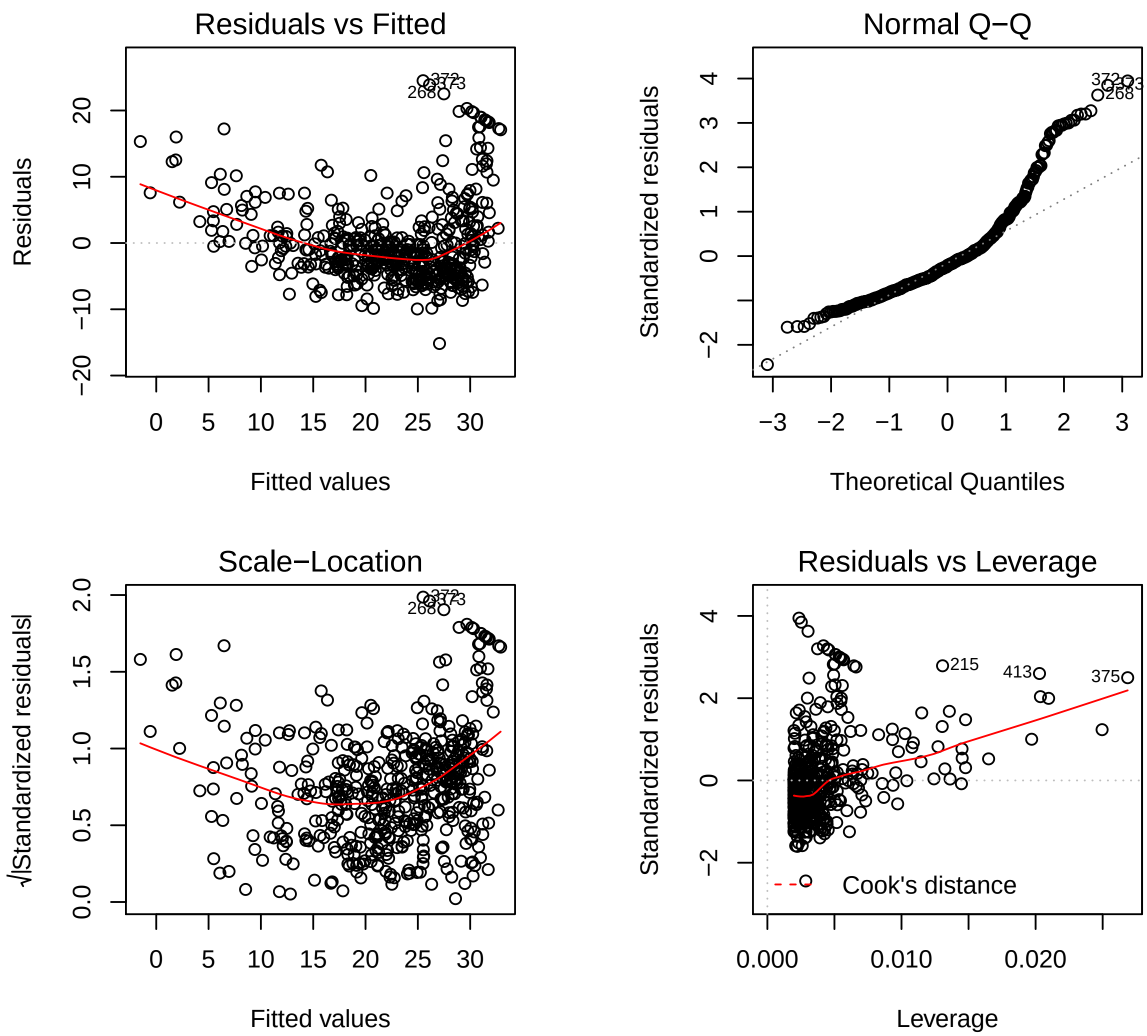
```
# subdivide the window into 4 parts, 2 rows and 2 columns
par(mfrow=c(2,2))
standard.res <- rstandard(model)
hist(standard.res, prob=TRUE, xlab='Standardized residuals')
plot(standard.res, pch=19, cex=0.5, ylab='Standardized residuals')
## add on the line parallel to the x-axis
abline(h=0, lty=2)
plot(est.values, standard.res, pch=19, cex=0.5,
        xlab='Estimated values', ylab='Standardized residuals')
abline(h=0, lty=2)
plot(Boston$lstat, standard.res, ylab='Standardized residuals',
        xlab='% of lower status of the population',
                pch=19, cex=0.5)
abline(h=0, lty=2)
```



Comments?

Graphical evaluation of the accuracy of the model provided by `R`

```
# subdivide the window into 4 parts, 2 rows and 2 columns
par(mfrow=c(2,2))
plot(model)
```



Are there any anomalies? There are "suspicious" values that `R` indicates through the corresponding row number in the dataset, but they are not anomalous on the basis of the Cook's distance (contour is zero).
Confidence interval at level 0.95 for β_1

```
## variance/covariance matrix associated to the parameter estimates
vcov(model)

##              (Intercept)       lstat
## (Intercept)   0.31654954 -0.018983106
## lstat        -0.01898311  0.001500278

## standard error
se <- sqrt(diag(vcov(model)))
se

## (Intercept)       lstat
##  0.56262735  0.03873342
```

```r
## for beta1
beta1-qt(0.975, df=n-2)*se[2]

##     lstat
## -1.026148

beta1+qt(0.975, df=n-2)*se[2]

##      lstat
## -0.8739505

## or using the operator c()
c(beta1-qt(0.975, df=n-2)*se[2],  beta1+qt(0.975, df=n-2)*se[2])

##      lstat      lstat
## -1.0261482 -0.8739505
```

Given the large values of n, the standard normal approximation can be used as well

```r
c(beta1-qnorm(0.975)*se[2],  beta1+qnorm(0.975)*se[2])

##      lstat      lstat
## -1.0259655 -0.8741333
```

Using R functionalities

```r
confint(model)

##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505

## change the confidence level, for example 90%
confint(model, level=0.90)

##                   5 %       95 %
## (Intercept) 33.626697 35.4809847
## lstat       -1.013877 -0.8862212
```

Hypothesis test on $H_0 : \beta_1 = -1$ against $H_1 : \beta_1 \neq -1$ at significance level 0.05

```r
statistic.t <- (beta1-(-1))/se[2]
statistic.t

##    lstat
## 1.289601
```

```r
qt(0.025, df=n-2)

## [1] -1.964682
```

There is no empirical evidence against H_0 at significance level 0.05.

```r
##p-value of the test
2*min(pt(statistic.t, n-2), 1-pt(statistic.t, n-2))

## [1] 0.1977807
```

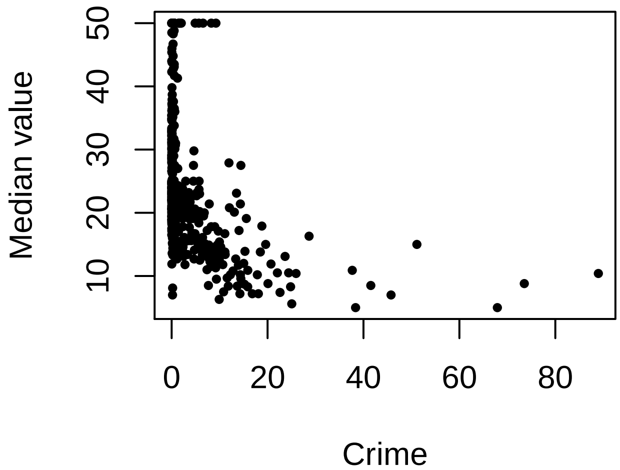
We confirm the previous result.
Predictions on a new dataset

```r
predict(model, newdata=data.frame(list(lstat=c(5, 10, 25))))

##        1        2        3
## 29.80359 25.05335 10.80261

## Predictions with prediction interval
predict(model, newdata=data.frame(list(lstat=c(5, 10, 25))),
        interval='prediction')

##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 10.80261 -1.457504 23.06272
```

1.1 Multiple linear regression model

Consider variable `crim` that includes the information about per capita crime rate by town.
Relationship between `crim` and `medv`

```r
plot(Boston$crim, Boston$medv, ylab='Median value',
        xlab='Crime', pch=19, cex=0.5)
```



Estimation of the model

$$\texttt{medv} = \beta_0 + \beta_1 \texttt{lstat} + \beta_2 \texttt{crim} + \varepsilon$$

```
model.mv <- lm(medv ~ lstat + crim, data=Boston)
summary(model.mv)

##
## Call:
## lm(formula = medv ~ lstat + crim, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -3.987  -1.513   2.138  25.017
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.31921    0.57374  59.816   <2e-16 ***
## lstat       -0.91139    0.04339 -21.004   <2e-16 ***
## crim        -0.07045    0.03602  -1.956   0.0511 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.198 on 503 degrees of freedom
## Multiple R-squared:  0.5476,Adjusted R-squared:  0.5458
## F-statistic: 304.4 on 2 and 503 DF,  p-value: < 2.2e-16
```

The significance of β_2 is questionable.
How do we interpret the parameter estimates? How is `medv` related to `lstat`?

1.2 Model with polynomials

Consider the model without `crim`. Given the dispersion plot between `medv` and `lstat` we can try to insert a quadratic term, that is, we estimate model

$$\texttt{medv} = \beta_0 + \beta_1 \texttt{lstat} + \beta_2 \texttt{lstat}^2 + \varepsilon$$

```
model2 <- lm(medv ~ lstat + I(lstat^2), data=Boston)
## or
model2 <- update(model, .~.+I(lstat^2))
summary(model2)

##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407,Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

The new covariate has an associated coefficient significantly different from 0.

Compare the two models, with and without the quadratic term, using the F statistic

```
rss0 <- (6.216^2)*504
## or
## sum(model$residuals^2)
rss <- (5.524^2)*503
f <- (rss0 - rss)/rss * (503/1)
f
```

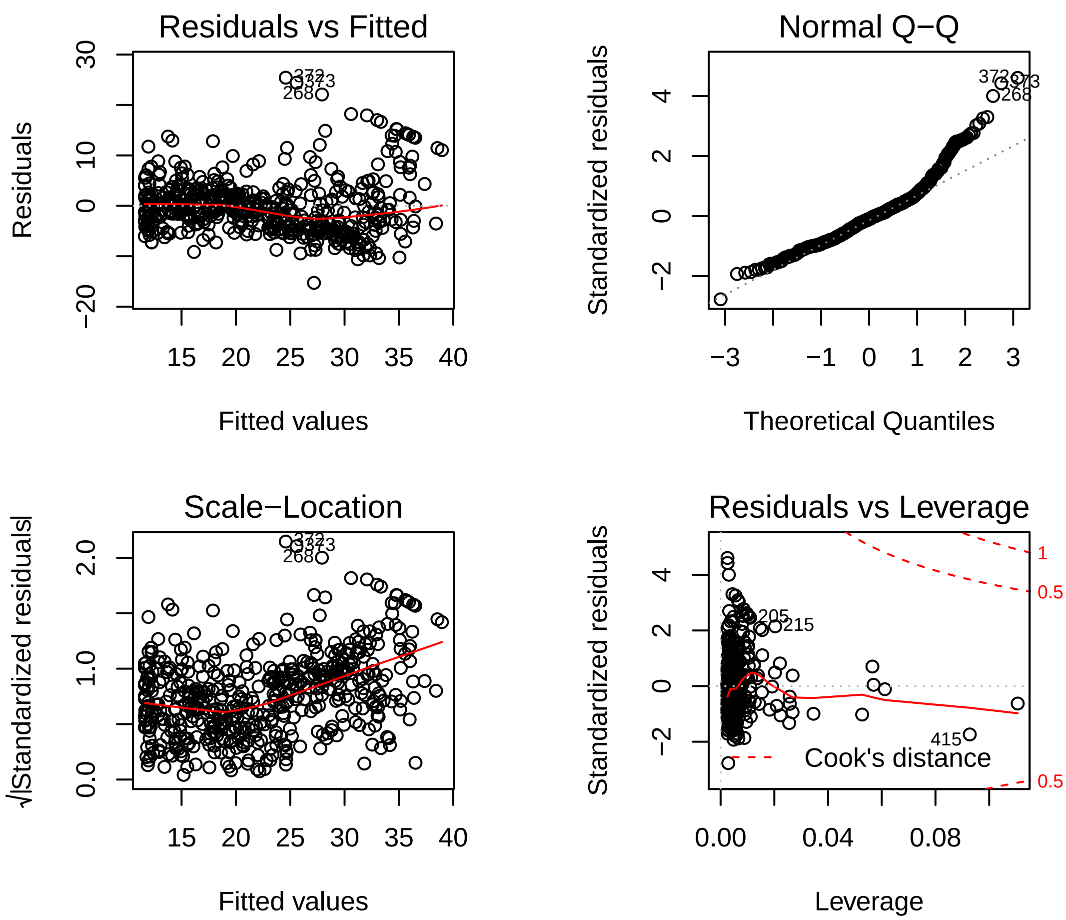
```
## [1] 135.183

qf(0.95, 1, 503)

## [1] 3.860012

## There is empirical evidence against H0 that suggests
## to move to the simplest model with a single covariate
## p-value
1-pf(f, 1, 503)

## [1] 0

## the p-value confirms the rejection of H0
```

In R we can use function `anova()`

```
anova(model, model2)

## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that in this case statistic F corresponds to the square of statistic t for the significance of the coefficient associated to the square of `lstat` in `model2`.
Residuals of the updated model

```
par(mfrow=c(2,2))
plot(model2)
```



2 *Gender discrimination* dataset

Data in file `Gender_Discrimination.csv` contain the information about gender, experience and annual salary in $ for some employees of a company. We want to evaluate whether the salary differs between males and females, given the experience.

```
my.data <- read.csv('Gender_Discrimination.csv', sep=',')
```

First look at the data

```
my.data[1:3,]

##   Gender Experience Salary
## 1 Female         15  78200
## 2 Female         12  66400
## 3 Female         15  61200

dim(my.data)
```

```
## [1] 208   3
```

```r
summary(my.data)
```

```
##     Gender      Experience        Salary      
##  Female:140   Min.   : 2.00   Min.   : 53400  
##  Male  : 68   1st Qu.: 7.00   1st Qu.: 66000  
##               Median :10.00   Median : 74000  
##               Mean   :12.05   Mean   : 79844  
##               3rd Qu.:16.00   3rd Qu.: 88000  
##               Max.   :39.00   Max.   :194000
```

Variable `Gender` is a qualitative variable with 2 levels, `Female` and `Male`

```r
is.factor(my.data$Gender)
```

```
## [1] TRUE
```

```r
levels(my.data$Gender)
```

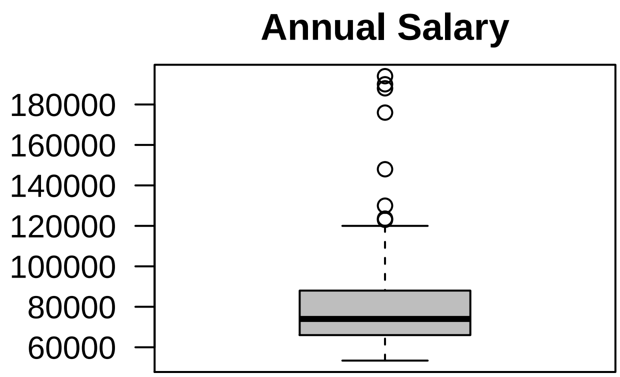
```
## [1] "Female" "Male"
```

```r
table(my.data$Gender)
```

```
## 
## Female   Male 
##    140     68
```

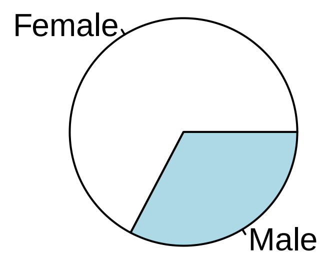
Initial description of the data.
Boxplot of the salary

```r
boxplot(my.data$Salary, las=2, col='grey', main='Annual Salary')
## las=2 plots the y-labels horizontally, to make them readable
```



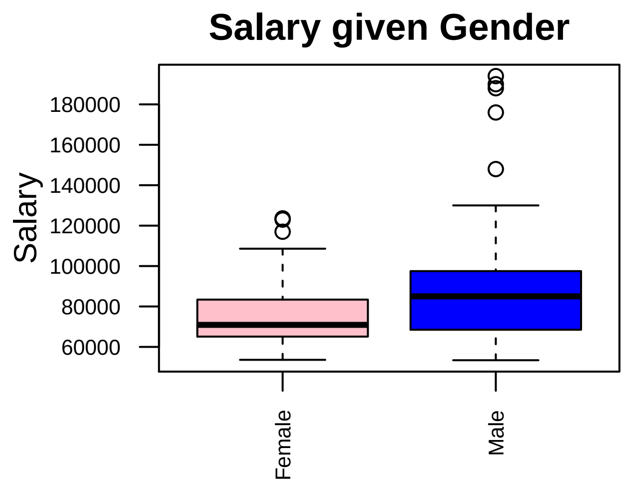
Gender distribution

```
pie(table(my.data$Gender), labels=c('Female','Male'))
```



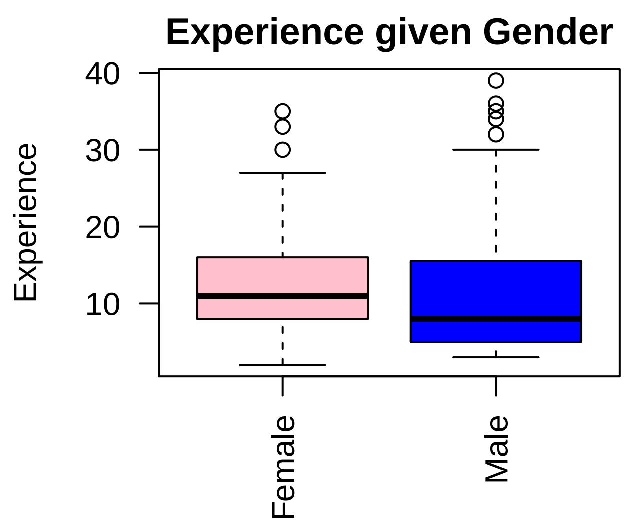
Distribution of salary given gender

```
boxplot(my.data$Salary~my.data$Gender, main='Salary given Gender',
        col=c('pink','blue'), las=2, ylab='Salary', cex.axis=0.7)
## cex.axis: modify the dimension of the labels, default is 1
```



Distribution of experience given gender

```
boxplot(my.data$Experience~my.data$Gender, main='Experience given Gender',
        col=c('pink','blue'), las=2, ylab='Experience')
```

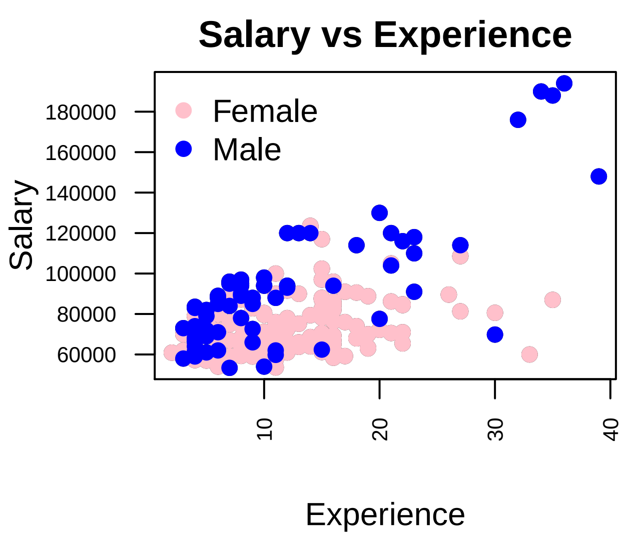


Dispersion plot of salary and experience

```
plot(my.data$Experience, my.data$Salary, main='Salary vs Experience',
        xlab='Experience', ylab='Salary', las=2, cex.axis=0.7)
```

Dispersion plot of salary and experience by distinguishing gender

```
plot(my.data$Experience, my.data$Salary, main='Salary vs Experience',
        xlab='Experience', ylab='Salary', las=2, cex.axis=0.7)
points(my.data$Experience[my.data$Gender == 'Female'],
        my.data$Salary[my.data$Gender == 'Female'], col='pink', pch=19)
points(my.data$Experience[my.data$Gender == 'Male'],
        my.data$Salary[my.data$Gender == 'Male'], col='blue', pch=19)
legend('topleft', pch=c(19,19), c('Female','Male'),
        col=c('pink','blue'), bty='n')
```



Estimate a multiple linear with covariates `Gender` and `Experience`. Consider that `Gender` is codified so that it assumes value 0 if `Gender=Female` and value 1 if `Gender=Male` (`R` follows the alphabetical order; it can be changed). The model is

$$\texttt{Salary} = \beta_0 + \beta_1 \texttt{Gender} + \beta_2 \texttt{Experience} + \varepsilon$$

or

$$\texttt{Salary} = \beta_0 + \beta_1 I(\texttt{Gender=Male}) + \beta_2 \texttt{Experience} + \varepsilon$$

if we want to explicit that Gender has an associated binary/indicator variable (dummy variable). Thus, if Gender=Female, the model is

$$\texttt{Salary} = \beta_0 + \beta_2 \texttt{Experience} + \varepsilon,$$

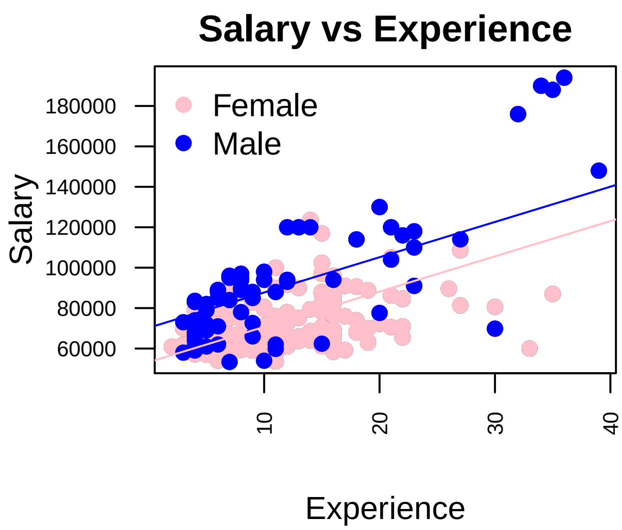
while if Gender=Male, the model is

$$\texttt{Salary} = \beta_0 + \beta_1 + \beta_2 \texttt{Experience} + \varepsilon,$$

```
model <- lm(Salary ~ Gender + Experience, data=my.data)
summary(model)

##
## Call:
## lm(formula = Salary ~ Gender + Experience, data = my.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52779  -9806   -121   8347  60913
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  53260.0     2416.6  22.039  < 2e-16 ***
## GenderMale   17020.6     2499.6   6.809 1.06e-10 ***
## Experience    1744.6      160.7  10.858  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16910 on 205 degrees of freedom
## Multiple R-squared:  0.4413,Adjusted R-squared:  0.4359
## F-statistic: 80.98 on 2 and 205 DF,  p-value: < 2.2e-16
```

Note that in the summary we have the estimate of β_1, the parameter in case gender is male. Female level is considered as *reference level*. The linear regression fit for females is $\widehat{\texttt{Salary}} = 5.3260001 \times 10^4 + 1744.6288555 * \texttt{Experience}$, while that for males is $\widehat{\texttt{Salary}} = 5.3260001 \times 10^4 + 1.7020585 \times 10^4 + 1744.6288555 * \texttt{Experience} = 7.0280587 \times 10^4 + 1744.6288555 * \texttt{Experience}$.
Graphical visualization

```r
plot(my.data$Experience, my.data$Salary, main='Salary vs Experience',
        xlab='Experience', ylab='Salary', las=2, cex.axis=0.7)
points(my.data$Experience[my.data$Gender == 'Female'],
        my.data$Salary[my.data$Gender == 'Female'], col='pink', pch=19)
points(my.data$Experience[my.data$Gender == 'Male'],
        my.data$Salary[my.data$Gender == 'Male'], col='blue', pch=19)
legend('topleft', pch=c(19,19), c('Female','Male'),
        col=c('pink','blue'), bty='n')
abline(coef(model)[1], coef(model)[3], col='pink')
abline(coef(model)[1]+coef(model)[2], coef(model)[3], col='blue')
```



Model with interaction between `Gender` and `Experience`

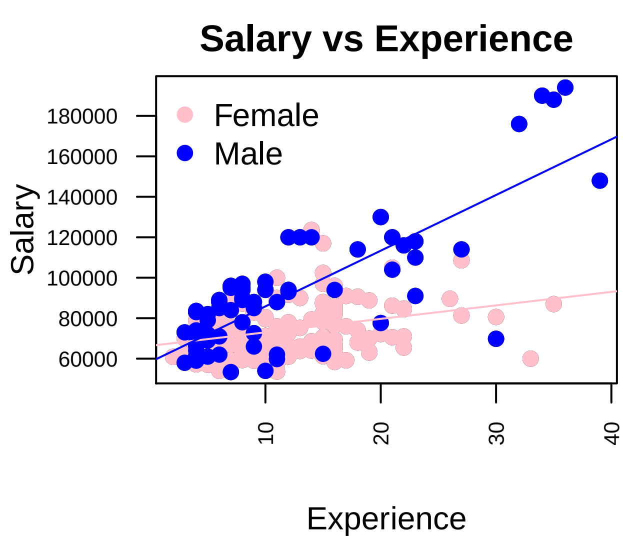
```r
model2 <- lm(Salary ~ Gender * Experience, data=my.data)
summary(model2)

##
## Call:
## lm(formula = Salary ~ Gender * Experience, data = my.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -71048  -9278  -1701   9166  47932
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              66333.6     2811.7  23.592  < 2e-16 ***
## GenderMale               -8034.3     4110.6  -1.955  0.05201 .
## Experience                 666.7      206.5   3.228  0.00145 **
## GenderMale:Experience     2086.2      287.3   7.261 7.95e-12 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15110 on 204 degrees of freedom
## Multiple R-squared:  0.5561,Adjusted R-squared:  0.5495
## F-statistic: 85.18 on 3 and 204 DF,  p-value: < 2.2e-16
```

What can we infer from the model? Is it preferable to the model without interaction? Why? A proper answer uses the value of R^2, the residual analysis, test F with `anova()`, ...

Graphical inspection of the model

```
plot(my.data$Experience, my.data$Salary, main='Salary vs Experience',
        xlab='Experience', ylab='Salary', las=2, cex.axis=0.7)
points(my.data$Experience[my.data$Gender == 'Female'],
        my.data$Salary[my.data$Gender == 'Female'], col='pink', pch=19)
points(my.data$Experience[my.data$Gender == 'Male'],
        my.data$Salary[my.data$Gender == 'Male'], col='blue', pch=19)
legend('topleft', pch=c(19,19), c('Female','Male'),
        col=c('pink','blue'), bty='n')
abline(coef(model2)[1], coef(model2)[3], col='pink')
abline(coef(model2)[1]+coef(model2)[2],
        coef(model2)[3]+coef(model2)[4], col='blue')
```

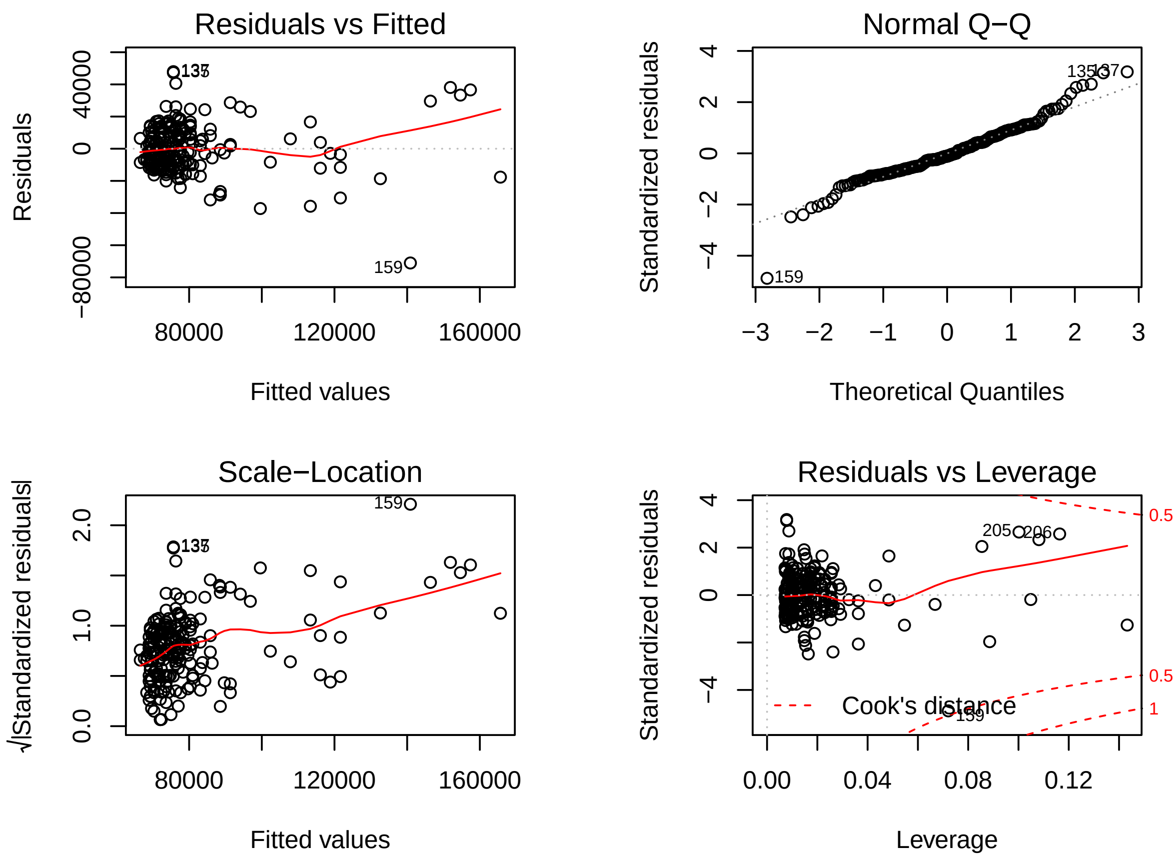


Does it make sense to include a polynomial term associated to `Experience`?

```r
## let's try with the square of Experience
model3 <- update(model2, . ~ . +I(Experience^2))
summary(model3)

##
## Call:
## lm(formula = Salary ~ Gender + Experience + I(Experience^2) +
##     Gender:Experience, data = my.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -71177  -9603  -1653   9365  48286
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           67736.622   3966.649  17.077  < 2e-16 ***
## GenderMale            -7858.634   4133.012  -1.901   0.0587 .
## Experience              433.976    507.375   0.855   0.3934
## I(Experience^2)           7.661     15.249   0.502   0.6159
## GenderMale:Experience  2040.852    301.713   6.764  1.4e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15140 on 203 degrees of freedom
## Multiple R-squared:  0.5566,Adjusted R-squared:  0.5479
## F-statistic: 63.71 on 4 and 203 DF,  p-value: < 2.2e-16
```

Comments?
We still need the residual analysis of `model2`.

```r
par(mfrow=c(2,2))
plot(model2)
```



How can we comment on the plot?
Using `model2` we can predict the salary for a male and a female with 20 years of experience

```
predict(model2, newdata=data.frame(list(Experience=20, Gender='Male')))

##        1
## 113358.4

predict(model2, newdata=data.frame(list(Experience=20, Gender='Female')))

##        1
## 79667.82
```

without using `predict()`

```
## prediction for male
coef(model2)[1]+ coef(model2)[2]+coef(model2)[3]*20+coef(model2)[4]*20

## (Intercept)
##    113358.4
```

```r
## prediction for female
coef(model2)[1]+ coef(model2)[3]*20

## (Intercept)
##    79667.82
```

3 Hald cement dataset

File `Hald.dat` contains the information about 13 cement mixture:

- column 1: Heat (cals/gm) evolved in setting, recorded to nearest tenth
- column 2: calcium aluminate
- column 3: tricalcium silicate
- column 4: tricalcium aluminoferrite
- column 5: dicalcium silicate

it is well known that some of the chemicals are partly equivalent.
It is of interest the relationship between the heat evolved in setting and the chemicals.
Upload the data

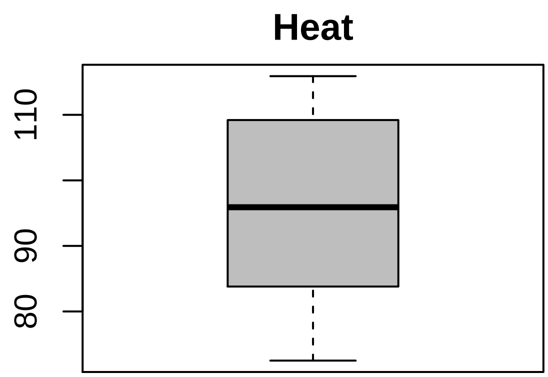
```
cement <- read.table('hald.dat')
cement

##       V1 V2 V3 V4 V5
## 1   78.5  7 26  6 60
## 2   74.3  1 29 15 52
## 3  104.3 11 56  8 20
## 4   87.6 11 31  8 47
## 5   95.9  7 52  6 33
## 6  109.2 11 55  9 22
## 7  102.7  3 71 17  6
## 8   72.5  1 31 22 44
## 9   93.1  2 54 18 22
## 10 115.9 21 47  4 26
## 11  83.8  1 40 23 34
## 12 113.3 11 66  9 12
## 13 109.4 10 68  8 12
```

Assign a name to the variables

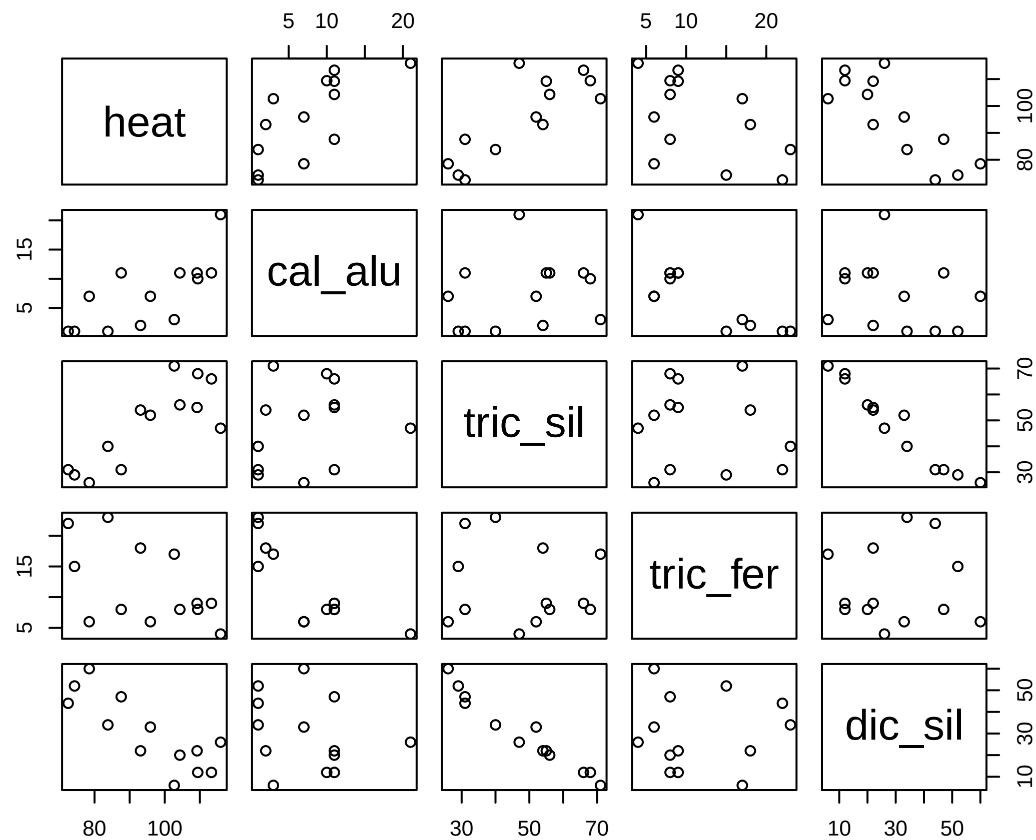
```
colnames(cement) <- c('heat','cal_alu', 'tric_sil', 'tric_fer', 'dic_sil')
```

Some preliminary graphical analyses

```
boxplot(cement$heat, col='grey', main='Heat')
```




```
pairs(cement)
```



Construct a first model with `cal_alu` as covariate

```r
m.cement <- lm(heat ~ cal_alu, data=cement)
summary(m.cement)

##
## Call:
## lm(formula = heat ~ cal_alu, data = cement)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.061  -9.048   1.339   7.883  15.614
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  81.4793     4.9273   16.54 4.07e-09 ***
## cal_alu       1.8687     0.5264    3.55  0.00455 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.73 on 11 degrees of freedom
## Multiple R-squared:  0.5339,Adjusted R-squared:  0.4916
## F-statistic:  12.6 on 1 and 11 DF,  p-value: 0.004552
```

Add on `tric_sil`

```r
m.cement2 <- lm(heat ~ cal_alu + tric_sil, data=cement)
summary(m.cement2)

##
## Call:
## lm(formula = heat ~ cal_alu + tric_sil, data = cement)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## cal_alu      1.46831    0.12130   12.11 2.69e-07 ***
## tric_sil     0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787,Adjusted R-squared:  0.9744
```

```
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

Both the variables are significant; including `tric_sil` moved $R^2 = 0.533948$ to $R^2 = 0.9786784$.
Add on the remaining variables

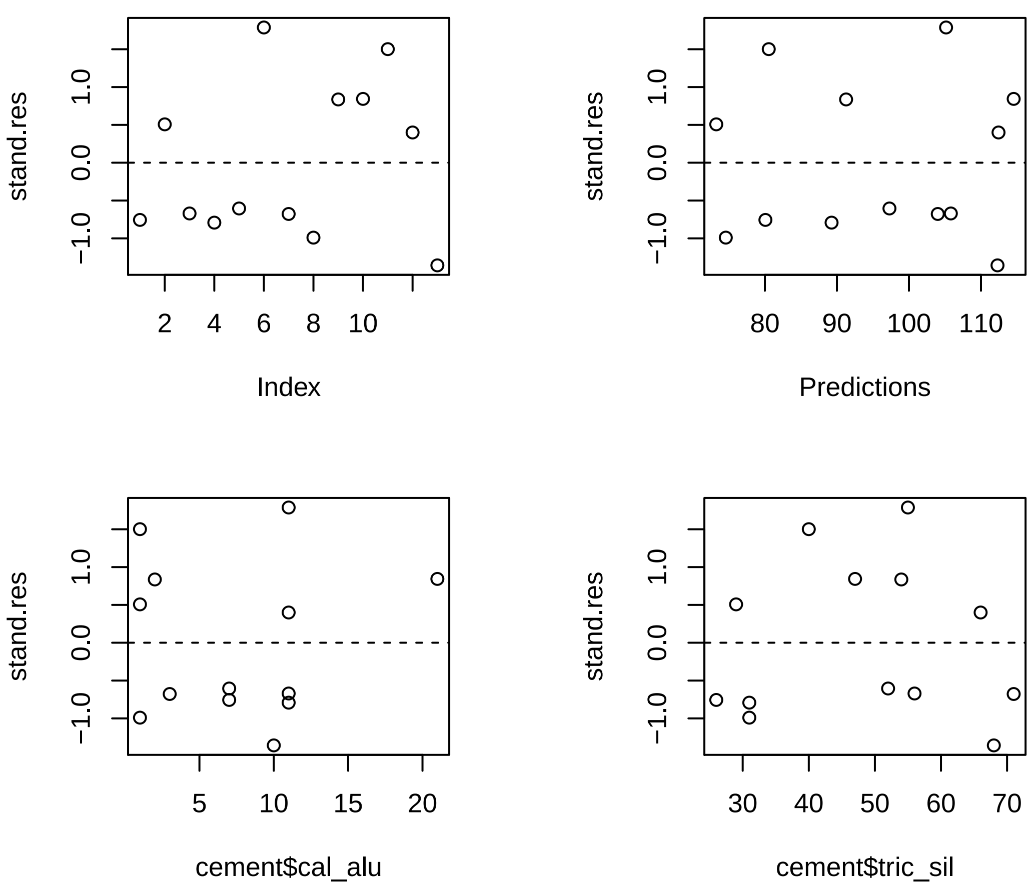
```
m.cement3 <- lm(heat ~ cal_alu + tric_sil + tric_fer + dic_sil, data=cement)
summary(m.cement3)

##
## Call:
## lm(formula = heat ~ cal_alu + tric_sil + tric_fer + dic_sil,
##     data = cement)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1750 -1.6709  0.2508  1.3783  3.9254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.4054    70.0710   0.891   0.3991
## cal_alu       1.5511     0.7448   2.083   0.0708 .
## tric_sil      0.5102     0.7238   0.705   0.5009
## tric_fer      0.1019     0.7547   0.135   0.8959
## dic_sil      -0.1441     0.7091  -0.203   0.8441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.446 on 8 degrees of freedom
## Multiple R-squared:  0.9824,Adjusted R-squared:  0.9736
## F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07
```

No significant variable anymore...but R^2 is still large...and F statistic would lead to reject the hypothesis of non-significant coefficients associated to all the covariates.... what's wrong in the model?
Check the correlations among the variables

```
cor(cement)

##                heat    cal_alu   tric_sil   tric_fer    dic_sil
## heat      1.0000000  0.7307175  0.8162526 -0.5346707 -0.8213050
## cal_alu   0.7307175  1.0000000  0.2285795 -0.8241338 -0.2454451
## tric_sil  0.8162526  0.2285795  1.0000000 -0.1392424 -0.9729550
## tric_fer -0.5346707 -0.8241338 -0.1392424  1.0000000  0.0295370
## dic_sil  -0.8213050 -0.2454451 -0.9729550  0.0295370  1.0000000
```

Variables `cal_alu` and `tric_fer` are highly correlated as well as `tric_sil` and `dic_sil`. Including highly correlated variables in the model hides the effects on the response...the phenomenon is called **multicollinearity**. The practical solution is to maintain just one of the two correlated variables in the model. So we will refer to model `m.cement2`.
Residuals of the model

```
stand.res <- rstandard(m.cement2)
predictions <- fitted(m.cement2)
par(mfrow=c(2,2))
plot(stand.res)
abline(h=0, lty=2)
plot(predictions, stand.res, xlab='Predictions')
abline(h=0, lty=2)
plot(cement$cal_alu, stand.res)
abline(h=0, lty=2)
plot(cement$tric_sil, stand.res)
abline(h=0, lty=2)
```



There are no deterministic patterns.

4 Carseats dataset

Dataset `Carseats` contains the information about 400 carseats. Data are included in package `ISLR` associated to the textbook Gareth J, Witten D, Hastie T, Tibshirani R. *An Introduction to Statistical Learning with Applications in R* (ISLR, from hereon). Springer, 2013. The following analysis answers to questions in exercise 10, chapter 3 of the textbook.

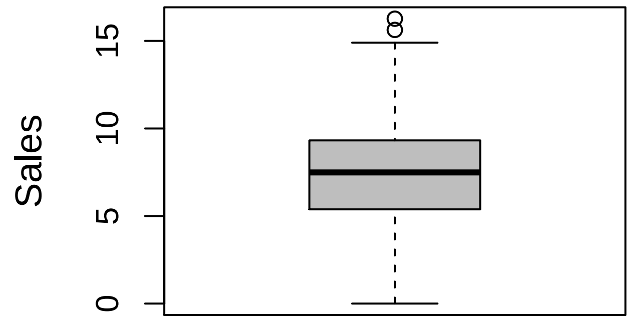
```
## upload library ISLR
library(ISLR)
## and the dataset
data(Carseats)
## dimension of the data
dim(Carseats)

## [1] 400  11

## variables
names(Carseats)

##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"  "Price"
##  [7] "ShelveLoc"   "Age"         "Education"   "Urban"       "US"
```

Extract the variables of interest, namely, `Sales`, `Price`, `Urban`, `US`, `ShelveLoc`.

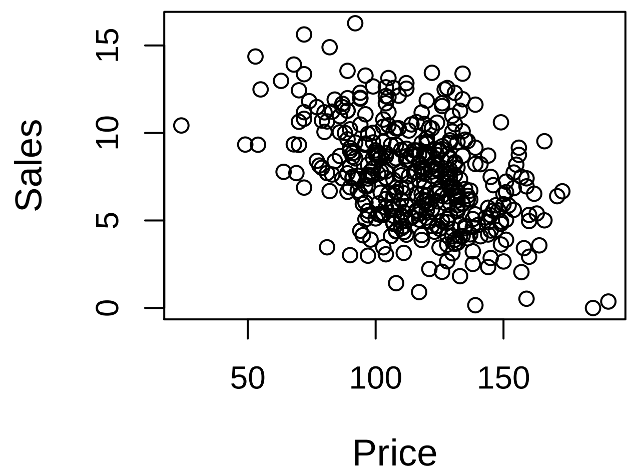
```
my.data <- Carseats[, c('Sales', 'Price', 'Urban', 'US', 'ShelveLoc')]
my.data[1:3,]

##   Sales Price Urban  US ShelveLoc
## 1  9.50   120   Yes Yes       Bad
## 2 11.22    83   Yes Yes      Good
## 3 10.06    80   Yes Yes    Medium
```

Some graphical analyses to evaluate the relationship between the response (`Sales`) and the covariates

```
boxplot(my.data$Sales, col='grey', ylab='Sales', cex.lab=1.2)
```

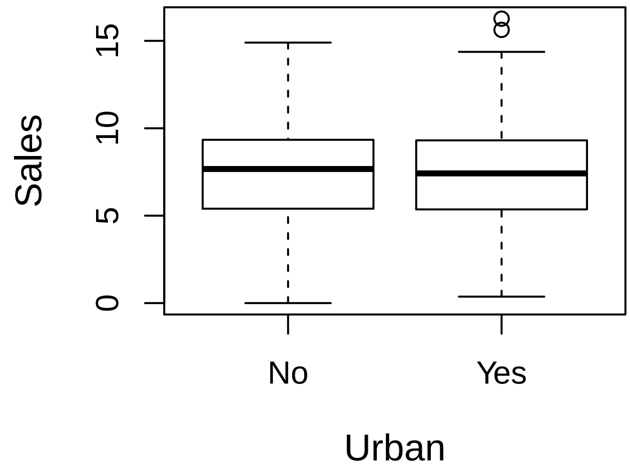



```
plot(my.data$Price, my.data$Sales, cex.lab=1.2, xlab='Price', ylab='Sales')
```



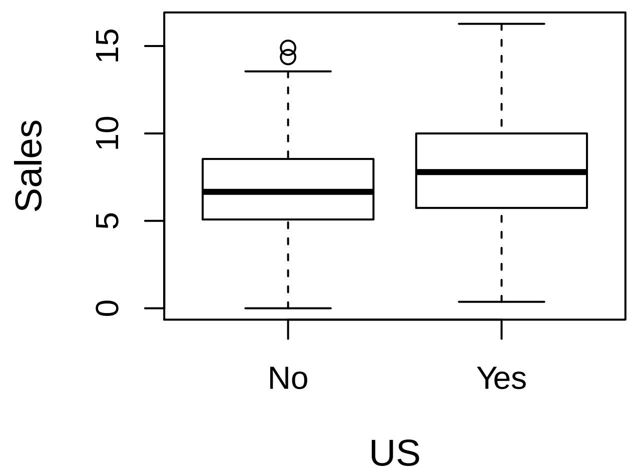
It seems there is an inverse relationship.

```
boxplot(my.data$Sales~my.data$Urban, cex.lab=1.2, xlab='Urban', ylab='Sales',
cex.names=1.2)
```



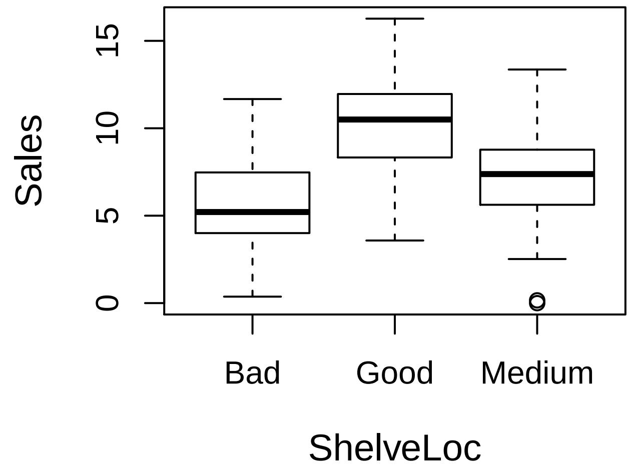
It seems there are no variations of `Sales` with respect to `Urban`, on average.

```
boxplot(my.data$Sales~my.data$US, cex.lab=1.2, xlab='US', ylab='Sales',
cex.names=1.2)
```



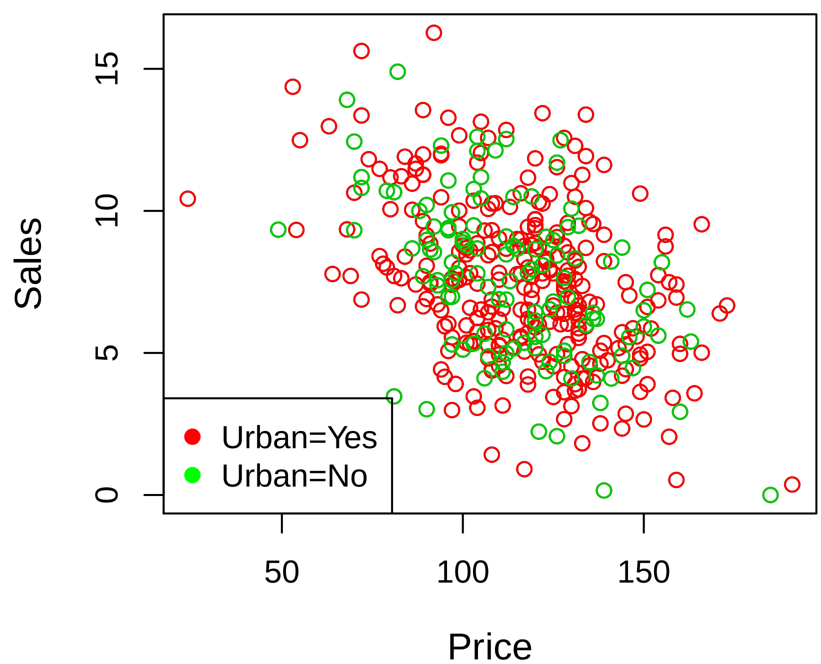
Is there anything interesting?

```
boxplot(my.data$Sales~my.data$ShelveLoc, cex.lab=1.2, xlab='ShelveLoc', ylab='Sales',
cex.names=1.2)
```



Is there anything interesting? Dispersion plot according to the levels of `Urban`

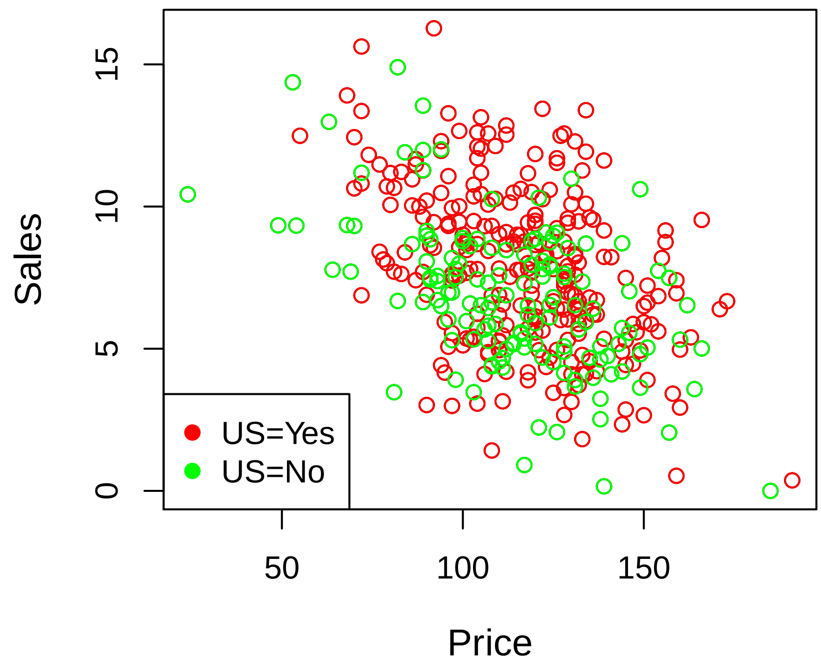
```
plot(my.data$Price, my.data$Sales, cex.lab=1.2, xlab='Price', ylab='Sales')
points(my.data$Price[my.data$Urban=='Yes'], my.data$Sales[my.data$Urban=='Yes'], col=2)
points(my.data$Price[my.data$Urban=='No'], my.data$Sales[my.data$Urban=='No'], col=3)
legend('bottomleft', col=c('red','green'), pch=c(19,19),
       legend=c('Urban=Yes','Urban=No'))
```



The partial overlapping of the observations belonging to the two groups suggests that there would not be interactions between the covariates.

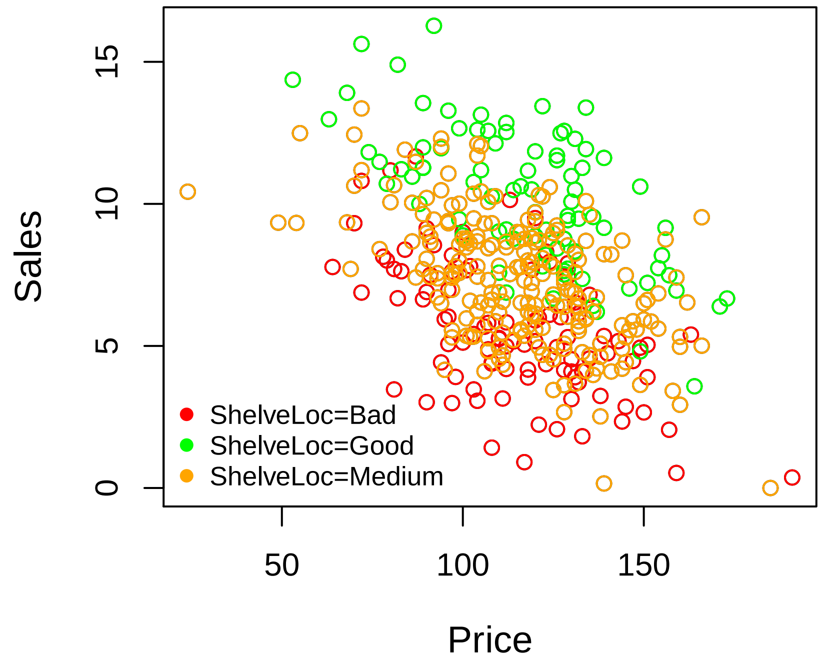
Dispersion plot according to the levels of US

```
plot(my.data$Price, my.data$Sales, cex.lab=1.2, xlab='Price', ylab='Sales')
points(my.data$Price[my.data$US=='Yes'], my.data$Sales[my.data$US=='Yes'], col='red')
points(my.data$Price[my.data$US=='No'], my.data$Sales[my.data$US=='No'], col='green')
legend('bottomleft', col=c('red','green'), pch=c(19,19),
       legend=c('US=Yes','US=No'))
```



How can we interpret the plot?
Dispersion plot according to the levels of ShelveLoc

```
plot(my.data$Price, my.data$Sales, cex.lab=1.2, xlab='Price', ylab='Sales')
points(my.data$Price[my.data$ShelveLoc=='Bad'], my.data$Sales[my.data$ShelveLoc=='Bad'],
       col='red')
points(my.data$Price[my.data$ShelveLoc=='Good'], my.data$Sales[my.data$ShelveLoc=='Good'
       col='green')
points(my.data$Price[my.data$ShelveLoc=='Medium'],
       my.data$Sales[my.data$ShelveLoc=='Medium'], col='orange')
legend('bottomleft', col=c('red', 'green', 'orange'), pch=c(19,19, 19),
       legend=c('ShelveLoc=Bad', 'ShelveLoc=Good', 'ShelveLoc=Medium'),
       bty='n', cex=0.8)
```



How can we interpret the plot?
Estimate the multiple linear regression model

```
model.sales <- lm(Sales~Price + Urban + US + ShelveLoc, data=my.data)
summary(model.sales)

##
## Call:
## lm(formula = Sales ~ Price + Urban + US + ShelveLoc, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0042 -1.2829 -0.0053  1.2471  4.6856
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     11.320199   0.514569  21.999  < 2e-16 ***
## Price           -0.058053   0.003941 -14.731  < 2e-16 ***
## UrbanYes         0.245370   0.204700   1.199    0.231
## USYes            1.002308   0.195132   5.137 4.41e-07 ***
## ShelveLocGood    4.853360   0.278001  17.458  < 2e-16 ***
## ShelveLocMedium  1.913316   0.227969   8.393 8.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.856 on 394 degrees of freedom
## Multiple R-squared:  0.5734,Adjusted R-squared:  0.568
## F-statistic: 105.9 on 5 and 394 DF,  p-value: < 2.2e-16
```

How do we interpret the coefficients associated to the qualitative variables?
Eliminate variable `Urban`

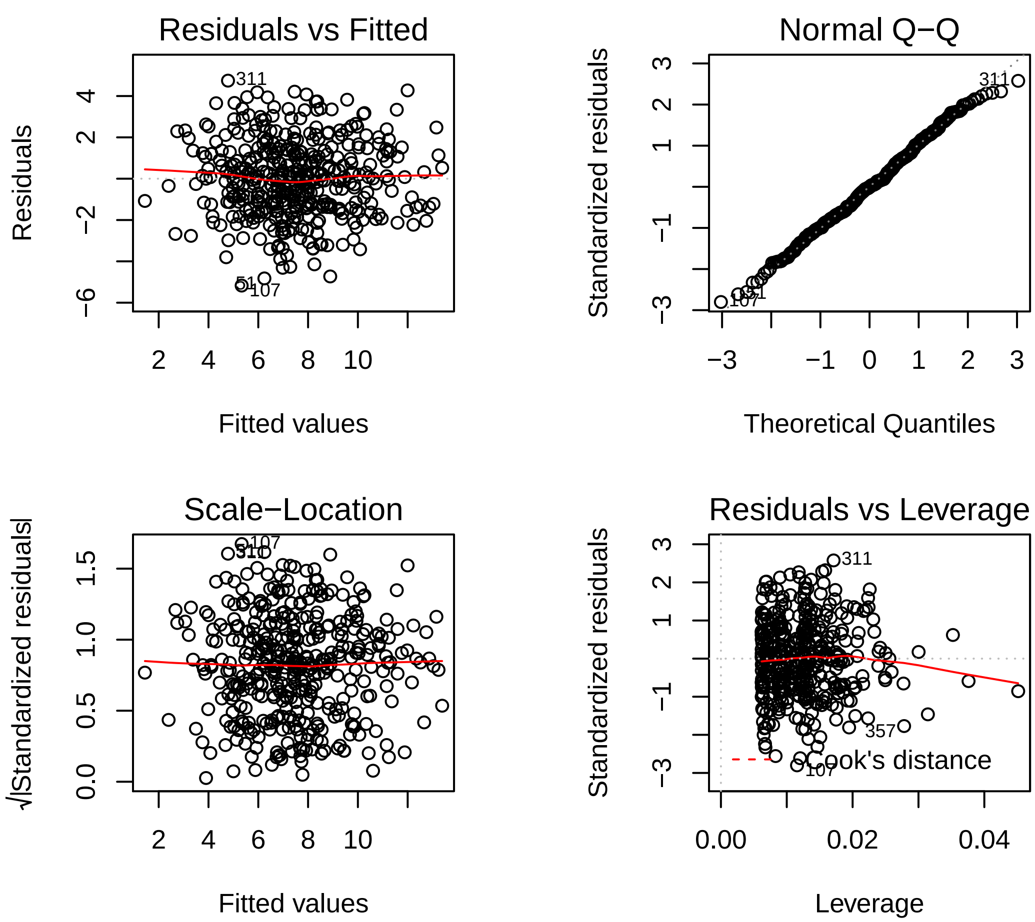
```
model.sales2 <- update(model.sales, .~.-Urban)
summary(model.sales2)

##
## Call:
## lm(formula = Sales ~ Price + US + ShelveLoc, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1720 -1.2587 -0.0056  1.2815  4.7462
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     11.476347   0.498083  23.041  < 2e-16 ***
## Price           -0.057825   0.003938 -14.683  < 2e-16 ***
## USYes            1.013071   0.195034   5.194 3.30e-07 ***
## ShelveLocGood    4.827167   0.277294  17.408  < 2e-16 ***
## ShelveLocMedium  1.893360   0.227486   8.323 1.42e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.857 on 395 degrees of freedom
## Multiple R-squared:  0.5718,Adjusted R-squared:  0.5675
## F-statistic: 131.9 on 4 and 395 DF,  p-value: < 2.2e-16
```

Evaluate the accuracy of `model.sales2` with respect to `model.sales` using statistic F

```
anova(model.sales2, model.sales)

## Analysis of Variance Table
##
## Model 1: Sales ~ Price + US + ShelveLoc
## Model 2: Sales ~ Price + Urban + US + ShelveLoc
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    395 1362.6
## 2    394 1357.6  1     4.951 1.4368 0.2314
```

Residual analysis from `model.sales2` provided by `R`

```r
par(mfrow=c(2,2))
plot(model.sales2)
```



A complete residual analysis would require further plots, as we have seen before; for example, histogram of the residuals, dispersione plot of the residuals against the covariates, ...

How do we interpret the model?

Confidence interval at 95% level:

```r
confint(model.sales2)
```

```
##                      2.5 %      97.5 %
## (Intercept)     10.49712308 12.45557170
## Price           -0.06556772 -0.05008219
## USYes            0.62963699  1.39650421
## ShelveLocGood    4.28200999  5.37232383
## ShelveLocMedium  1.44612467  2.34059480
```

Predictions of sales for a store in US, when the price of the carseat is 115 $ and when `ShelveLoc` is `Medium`:

```
estimate <- coef(model.sales2)
estimate[1] + estimate[2]*115 + estimate[3] + estimate[5]

## (Intercept)
##    7.732908
```

How does the prediction change when `ShelveLoc` is `Bad`?

```
estimate[1] + estimate[2]*115 + estimate[3]

## (Intercept)
##    5.839548
```

Evaluate whether interactions in the model make sense...what do the previous plots suggest?

```
## for example...
model.sales3  <- lm(Sales~Price * ShelveLoc + US, data=my.data)
summary(model.sales3)

##
## Call:
## lm(formula = Sales ~ Price * ShelveLoc + US, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2497 -1.2567 -0.0158  1.2418  4.5909
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           11.365656   0.940335  12.087  < 2e-16 ***
## Price                 -0.056816   0.008027  -7.079 6.75e-12 ***
## ShelveLocGood          5.870530   1.350791   4.346 1.77e-05 ***
## ShelveLocMedium        1.645606   1.135419   1.449    0.148
## USYes                  1.005919   0.195317   5.150 4.12e-07 ***
## Price:ShelveLocGood   -0.008877   0.011384  -0.780    0.436
## Price:ShelveLocMedium  0.002128   0.009697   0.219    0.826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.859 on 393 degrees of freedom
## Multiple R-squared:  0.5732,Adjusted R-squared:  0.5667
## F-statistic: 87.98 on 6 and 393 DF,  p-value: < 2.2e-16
```

The interaction is not significant.

Suppose we want to change the baseline level for one qualitative variabile. For example, we change the baseline level of `ShelveLoc` from `Bad` to `Good`. There are two possibilities

```
## first possibility
new.shelveloc <- my.data$ShelveLoc
contrasts(new.shelveloc) <- contr.treatment(levels(new.shelveloc),
        base=which(levels(new.shelveloc) == 'Good'))
## second possibility
new.shelveloc2 <- relevel(Carseats$ShelveLoc, ref='Good')
```

Function `contrasts()` allows more possibilities to specify *contrasts* (levels, relationships among levels), while `relevel()` only allows to change the reference level. They are equivalent for our purpose.

```
model.sales4 <- update(model.sales2, .~. - ShelveLoc + new.shelveloc2)
summary(model.sales4)

##
## Call:
## lm(formula = Sales ~ Price + US + new.shelveloc2, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1720 -1.2587 -0.0056  1.2815  4.7462
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           16.303514   0.518219  31.461  < 2e-16 ***
## Price                 -0.057825   0.003938 -14.683  < 2e-16 ***
## USYes                  1.013071   0.195034   5.194  3.3e-07 ***
## new.shelveloc2Bad     -4.827167   0.277294 -17.408  < 2e-16 ***
## new.shelveloc2Medium -2.933807   0.238289 -12.312  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.857 on 395 degrees of freedom
## Multiple R-squared:  0.5718,Adjusted R-squared:  0.5675
## F-statistic: 131.9 on 4 and 395 DF,  p-value: < 2.2e-16
```

Note that the results are coherent with those from `model.sales2`, with obvious changes in signs and values of the coefficients associated to the dummies in `Shelveloc`.