Multiple Linear Regression

2021

Multiple Linear Regression

Here we assume that $\mathbb{D}Y < \infty$ and $\mathbb{D}X^{(j)} < \infty, j = 1, 2, \dots, r$.

Regression analysis study the form of the relationship between numerical random variables $X^{(1)}, X^{(2)}, \dots, X^{(r)}$ and Y. More precisely its aim is by knowing $X^{(1)}, X^{(2)}, \dots, X^{(r)}$ and the regression model to predict Y.

For example, the price Y of a new home depends on many factors:

$X^{(1)}$ - the number of bedrooms,
$X^{(2)}$ - the number of bathrooms,
$X^{(3)}$ - the location of the house, etc.

People develop rules of thumb to help figure out the value. These may be:

$+\$30{,}000$ for an extra bedroom
$+\$15{,}000$ for an extra bathroom
$-\$10{,}000$ for the busy street.

These are intuitive uses of a multiple linear regression model to explain the cost of a house based on several variables.

$X^{(1)}, X^{(2)}, \dots, X^{(r)}$ are called **independent (or explanatory) variables (or predictors, or regressors) /независими променливи/**. I.e. we have multiple explanatory variables. If some of them are correlated we speak about **multicollinearity /мултиколинеарност/**. In such cases we would difficultly differentiate the clear effects of separate independent random variables. In presanse of multicollinearity the estimators considered here are again unbiased, however their standards errors will be bigger. **If there is no multicollinearity the coefficients of the models with more independent variables will be**

the same as the coefficients before the same variables in models with less independent variables.

Y is called **dependent (or outcome or response) variable (or regressand)** .

When there is a single dependent variable Y and multiple independent variables $X^{(1)}, X^{(2)}, \ldots, X^{(r)}$, and the dependence on the coefficients is linear the analysis is called a multiple linear regression analysis. More precisely the multiple linear regression model is

$$Y = \hat{Y} + \varepsilon = \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)} + \varepsilon = \vec{\beta}^T \vec{X} + \varepsilon,$$

where

$$\varepsilon \in N(0, \sigma_\varepsilon^2), \operatorname{cov}(X^{(j)}, \varepsilon) = 0, j = 1, 2, \ldots, r, \ \vec{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \ldots \\ \beta_r \end{pmatrix}, \vec{\beta} = \begin{pmatrix} 1 \\ X^{(1)} \\ X^{(2)} \\ \ldots \\ X^{(r)} \end{pmatrix}.$$

In practice, the above assumptions should be checked as part of the model-building process.

- ε **is the random residual (error) term /случайна грешка/**

$$\varepsilon = Y - \hat{Y} = Y - \beta_0 - \beta_1 X^{(1)} - \beta_2 X^{(2)} - \ldots - \beta_r X^{(r)} = Y - \vec{\beta}^T \vec{X}$$

- $\beta_1, \ldots, \beta_r$, are **unknown coefficients**. They will be estimated from the data by using the method of **least squares** (by minimizing the sum of square errors $\sum_{i=1}^{n} \varepsilon_i^2$).

By assumption

- $\mathbb{E}\varepsilon = 0$, and therefore,

$$\mathbb{E}Y = \beta_0 + \beta_1 \mathbb{E}X^{(1)} + \beta_2 \mathbb{E}X^{(2)} + \ldots + \beta_r \mathbb{E}X^{(r)}$$
$$\beta_0 = \mathbb{E}Y - \beta_1 \mathbb{E}X^{(1)} - \beta_2 \mathbb{E}X^{(2)} - \ldots - \beta_r \mathbb{E}X^{(r)}$$

- $\operatorname{cor}(X^{(i)}, \varepsilon) = 0$, $i = 1, 2, \ldots, r$, i.e. the independent variables $X^{(1)}, X^{(2)}, \ldots, X^{(r)}$ and the random error term ε are uncorrelated.

Therefore, $\hat{Y}$ and ε are uncorrelated and

$$\mathbb{E}(\varepsilon \,|\, X^{(1)}, X^{(2)}, \ldots, X^{(r)}) = \mathbb{E}\varepsilon = 0,$$

$$\mathbb{D}(\varepsilon \,|\, X^{(1)}, X^{(2)}, \ldots, X^{(r)}) = \mathbb{D}\varepsilon = \sigma_\varepsilon^2.$$

$$\begin{aligned}\hat{Y} &= \mathbb{E}(Y \,|\, X^{(1)}, X^{(2)}, \ldots, X^{(r)}) = \\ &= \mathbb{E}(\beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)} + \varepsilon \,|\, X^{(1)}, X^{(2)}, \ldots, X^{(r)}) = \\ &= \beta_0 + \beta_1 X^{(1)} + \beta_2 \mathbb{E} X^{(2)} + \ldots + \beta_r X^{(r)}\end{aligned}$$

$$\mathbb{E}\hat{Y} = \beta_0 + \beta_1 X^{(1)} + \beta_2 \mathbb{E} X^{(2)} + \ldots + \beta_r \mathbb{E} X^{(r)} = \vec{\beta}^T \mathbb{E} \vec{X} = \mathbb{E} Y,$$

and the corresponding **multiple linear regression equation** (the equation of the corresponding $r + 1$ dimensional hyperplane) is as follows:

$$y = \beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + \ldots + \beta_r x^{(r)}$$

By the model assumed it is easy to see that

- $\beta_0 = \mathbb{E}X(Y \,|\, X^{(1)} = 0, X^{(2)} = 0, \ldots, X^{(r)} = 0)$ is the **intercept** of the regression hyperplane from Oy axis.

- $$\begin{aligned}\beta_i &= \mathbb{E}(Y \,|\, X^{(i)} + 1, X^{(m)}, m \neq i) - \mathbb{E}(Y \,|\, X^{(i)}, X^{(m)}, m \neq i) = \\ &= \beta_i (X^{(i)} + 1) - \beta_i X^{(i)}\end{aligned}$$
  is the expected increment of the Y (in its units) when $X^{(i)}$ increases with 1 (in the units of $X^{(i)}$) and the other $X^{(m)}$, $m \neq i$, $m = 1, 2, \ldots, r$ are fixed.

When we consider the variances

$$\begin{aligned}\mathbb{D}(\hat{Y}) &= \mathbb{D}(\beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)}) = \mathbb{D}(\vec{\beta}^T \vec{X}) = \\ &= \operatorname{cov}(\vec{\beta}^T \vec{X}, \vec{\beta}^T \vec{X}) = \vec{\beta}^T \operatorname{cov}(\vec{X}) \vec{\beta}\end{aligned}$$

$$\mathbb{D}(Y) = \mathbb{D}(\hat{Y} + \varepsilon) = \mathbb{D}(\hat{Y}) + \mathbb{D}\varepsilon = \mathbb{D}(\beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)} + \mathbb{D}\varepsilon) =$$
$$= \mathbb{D}(\vec{\beta}^T \vec{X}) + \sigma_\varepsilon^2 = \mathrm{cov}(\vec{\beta}^T \vec{X}^T, \vec{\beta}^T \vec{X}^T) + \sigma_\varepsilon^2 = \vec{\beta}^T \mathrm{cov}(\vec{\beta}) + \sigma_\varepsilon^2$$

$$\frac{\mathbb{D}\hat{Y}}{\mathbb{D}Y} = \frac{\mathbb{D}Y - \mathbb{D}\varepsilon}{\mathbb{D}Y} = 1 - \frac{\mathbb{D}\varepsilon}{\mathbb{D}Y}$$

$$\mathrm{cov}(Y, \hat{Y}) = \mathrm{cov}(Y, \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)}) =$$
$$= \mathrm{cov}(Y, \vec{\beta}^T \vec{X}) = \vec{\beta}^T \mathrm{cov}(\vec{X}, Y) = \mathrm{cov}(Y, \vec{X})\,\vec{\beta}$$

Moreover,

$$\mathrm{cov}(Y, \hat{Y}) = \mathrm{cov}(\hat{Y} + \varepsilon, \hat{Y}) = \mathrm{cov}(\hat{Y}, \hat{Y}) + \mathrm{cov}(\varepsilon, \hat{Y}) =$$
$$= \mathrm{cov}(\hat{Y}, \hat{Y}) = \mathbb{D}\hat{Y}\,\vec{\beta}^T \mathrm{cov}(\vec{X})\,\vec{\beta}$$

Therefore,

$$\mathrm{cov}(Y, \vec{X}) = \vec{\beta}^T \mathrm{cov}(\vec{X}),\ \mathrm{cov}(\vec{X}, Y) = \mathrm{cov}(\vec{X})\,\vec{\beta}$$

The corresponding estimators of $\mathbb{E}Y$, $\mathbb{E}X$, $\mathbb{D}Y$, $\mathrm{cov}(\vec{X})$, $\mathrm{cov}(\vec{X}, Y)$ and $\mathrm{cor}(\vec{X}, Y)$ are already known. Therefore, we can estimate the coefficients $\vec{\beta}$

$$\vec{\beta}^T = \mathrm{cov}(Y, \vec{X})\mathrm{cov}(\vec{X})^{-1} \Leftrightarrow \vec{\beta} = \mathrm{cov}(\vec{X})^{-1}\mathrm{cov}(\vec{X}, Y). \qquad (1)$$

When we need to assess the quality of the model we need the following characteristic

$$\mathrm{cor}^2(\vec{X}, Y) := \frac{\mathrm{cov}(Y, \vec{X})\mathrm{cov}^{-1}(\vec{X})\mathrm{cov}(\vec{X}, Y)}{\mathbb{D}X} =$$
$$= \frac{\mathrm{cov}(Y, \vec{X})\,\vec{\beta}}{\mathbb{D}Y} = \frac{\mathbb{D}\hat{Y}}{\mathbb{D}Y} = 1 - \frac{\mathbb{D}\varepsilon}{\mathbb{D}Y}$$

(and the corresponding estimator R^2) is called **coefficient of determination /коефициент на определеност/**. And, as far as

$$\mathbb{D}Y = \mathbb{D}Y\,\mathrm{cor}^2(\overrightarrow{X}, Y) + \mathbb{D}\varepsilon$$

$\mathrm{cor}^2(\overrightarrow{X}, Y)$ shows what part of $\mathbb{D}Y$ which is due to regression.

$1 - \mathrm{cor}^2(\overrightarrow{X}, Y)$ is called **coefficient of indetermination / коефициент на неопределеност/**. It shows part of $\mathbb{D}Y$ is due to changes of the error term, i.e. variables that are not considered in the model.

When we use these coefficients $\overrightarrow{\beta} = \mathrm{cov}(\overrightarrow{X})^{(-1)}\mathrm{cov}(\overrightarrow{X}, Y)$, the minimal value of the **Residual Standard error** (between Y and $\hat{Y}$) of the model is

$$\sigma_\varepsilon = \sqrt{\mathbb{D}\varepsilon} = \sqrt{\mathbb{E}\varepsilon^2} = \sqrt{\mathbb{E}(Y - \hat{Y})^2} = \sqrt{\mathbb{E}(Y - \overrightarrow{\beta}^T\overrightarrow{X})^2} =$$
$$= \sqrt{\mathbb{D}Y(1 - \mathrm{cor}^2(\overrightarrow{X}, Y))}$$

The inequality

$$\mathbb{D}(Y\,|\,\overrightarrow{X} = \overrightarrow{x}) = \mathbb{D}(\overrightarrow{\beta}^T\overrightarrow{X} + \varepsilon\,|\,\overrightarrow{X} = \overrightarrow{x}) = \sigma_\varepsilon^2 \leq \overrightarrow{\beta}^T\mathrm{cov}(\overrightarrow{X})\overrightarrow{\beta} + \sigma_\varepsilon^2 = \mathbb{D}Y$$

means that the information for $\overrightarrow{X}$ can help us to improve the estimation for Y as far as by using $\overrightarrow{X}$ we will obtain shorter confidence intervals for $(Y\,|\,\overrightarrow{X} = \overrightarrow{x})$, than for Y.

The most important case of these models is when the errors $\varepsilon \in N(0,\sigma_\varepsilon^2)$. Then,

$$(Y\,|\,\overrightarrow{X} = \overrightarrow{x}) = (\overrightarrow{\beta}^T\overrightarrow{X} + \varepsilon\,|\,\overrightarrow{X} = \overrightarrow{x}) \in$$
$$\in \Big(\overrightarrow{\beta}^T\overrightarrow{x} = \mathbb{E}Y + \overrightarrow{\beta}^T(\overrightarrow{x} - \mathbb{E}\overrightarrow{X}) = \mathbb{E}Y = \mathrm{cov}(Y, \overrightarrow{X})\mathrm{cov}^{-1}(\overrightarrow{X})(\overrightarrow{x} - \mathbb{E}\overrightarrow{X});$$

$$\sigma_\varepsilon^2 = \mathrm{D}\,\varepsilon = \mathbb{D}Y - \mathrm{cov}(Y\vec{X})\mathrm{cov}^{(-1)}(\vec{X})\mathrm{cov}(\vec{X}, Y) = \mathbb{D}Y(1 - \mathrm{cor}^2(\vec{X}, Y))\Big)$$

and by knowing $\vec{X}$, $\vec{\beta}$ we can construct confidence interval for $(Y \mid \vec{X} = \vec{x})$ and its numerical characteristics.

Suppose we have n independent observations $(Y_i, X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(r)})$, $i = 1, 2, \ldots, n$ on the random vector $(Y, X^{(1)}, X^{(2)}, \ldots, X^{(r)})$. It is more compact to write the multiple LM using vectors and matrices:

$$\vec{Y} = \mathbb{X}\vec{\beta} + \vec{\varepsilon}, \vec{\varepsilon} \in N(\vec{0}, \sigma_\varepsilon^2\mathbb{I}), \mathbb{I} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix},$$

(The last mean that we have assumes that $\mathbb{D}\varepsilon_i = \sigma_\varepsilon^2$, $\mathrm{cov}(\varepsilon_i, \varepsilon_j) = 0$, $1 \le i < j \le n$.) where

$$\vec{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \ldots \\ Y_n \end{pmatrix}, \mathbb{X} = \begin{pmatrix} 1 & X_1^{(1)} & X_1^{(2)} & \ldots & X_1^{(r)} \\ 1 & X_2^{(1)} & X_2^{(2)} & \ldots & X_2^{(r)} \\ 1 & X_3^{(1)} & X_3^{(2)} & \ldots & X_3^{(r)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & X_n^{(1)} & X_n^{(2)} & \ldots & X_n^{(r)} \end{pmatrix}, \vec{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \ldots \\ \beta_r \end{pmatrix},$$

$$\vec{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_2 \\ \ldots \\ \varepsilon_n \end{pmatrix}.$$

$$\mathbb{X}^T\mathbb{X} = \begin{pmatrix} n & \sum_{i=1}^n X_i^{(1)} & \sum_{i=1}^n X_i^{(2)} & \dots & \sum_{i=1}^n X_i^{(r)} \\ \sum_{i=1}^n X_i^{(1)} & \sum_{i=1}^n \left(X_i^{(1)}\right)^2 & \sum_{i=1}^n X_i^{(1)}X_i^{(2)} & \dots & \sum_{i=1}^n X_i^{(1)}X_i^{(r)} \\ \sum_{i=1}^n X_i^{(2)} & \sum_{i=1}^n X_i^{(1)}X_i^{(2)} & \sum_{i=1}^n \left(X_i^{(2)}\right)^2 & \dots & \sum_{i=1}^n X_i^{(2)}X_i^{(r)} \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{i=1}^n X_i^{(r)} & \sum_{i=1}^n X_i^{(1)}X_i^{(r)} & \sum_{i=1}^n X_i^{(2)}X_i^{(r)} & \dots & \sum_{i=1}^n \left(X_i^{(r)}\right)^2 \end{pmatrix}$$

$\overrightarrow{\varepsilon} = \overrightarrow{Y} - \mathbb{X}\overrightarrow{\beta}$ is the vector form of the error terms.

Note that Errors in LMs are uncorrelated, normal with mean zero and constant variance. This is called **homoscedasticity / хомоскедастичност/**. Its opposite form **heteroscedasticity / хетероскедастичност/** is when $\mathbb{D}(\varepsilon_i)$ changes.

As far as

$$Y_i = \hat{Y} + \varepsilon_i = \beta_0 + \beta_1 X_i^{(1)} + \ldots + \beta - rX_i^{(r)} + \varepsilon_i = \overrightarrow{\beta}^T \overrightarrow{X}_i + \varepsilon_i,$$

$$\overrightarrow{X}_i = \begin{pmatrix} 1 \\ X_i^{(1)} \\ X_i^{(2)} \\ \dots \\ X_i^{(r)} \end{pmatrix}, i = 1, 2, \ldots, n$$

$$\hat{Y}_i = \mathbb{E}(Y_i \,|\, \overrightarrow{X}_i) = \beta_0 + \beta_1 X_i^{(i)} + \ldots + \beta_r X_i^{(r)}, i = 1, 2, \ldots, n$$

The corresponding **Estimator of the Residual Standard error (RSE) / Стандартна грешка на остатъците/** is

$$\hat{\sigma}_\varepsilon = RSE = S_\varepsilon = \sqrt{\frac{\sum_{i=1}^n \varepsilon_i^2}{n - r - 1}},$$

where r is the number of coefficients in front of the independent variables. Therefore, $(r + 1)$ is the number of the parameters in the model. S_ε^2 is a unbiased estimator for σ_ε^2 and is called **mean square error(MSE) of the model /усреднен квадрат на грешката на модела/**

We use the following notations

$$SSE = \sum_{i=1}^{n} \varepsilon_i^2, MSE = \frac{SSE}{n-r-1} = RSE^2 = S_\varepsilon^2.$$

Let us now explain briefly **the method of least squares /метод на най-малките квадрати/** which is the best way to estimate the coefficients. We are looking for constants

$$\overrightarrow{\beta} = \arg\min\left(\sum_{i=1}^{n} \varepsilon_i^2\right) = \arg\min\left(\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2\right) =$$
$$= \arg\min\left(\sum_{i=1}^{n} (Y_i - \overrightarrow{\beta}^T \overrightarrow{X}_i)^2 = \arg\min(\overrightarrow{Y} - \mathbb{X}\overrightarrow{\beta})^T(\overrightarrow{Y} - \mathbb{X}\overrightarrow{\beta})\right)$$

The solution is obtained when we solve the following system of equations with respect to $\overrightarrow{\beta}$

$$\left|\begin{array}{l} -2\mathbb{X}^T(\overrightarrow{Y} - \mathbb{X}\overrightarrow{\beta}) = \overrightarrow{0} \Leftrightarrow \\ \mathbb{X}^T\overrightarrow{Y} = \mathbb{X}^T\mathbb{X}\overrightarrow{\beta} \Leftrightarrow \\ \overrightarrow{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\overrightarrow{Y} \Leftrightarrow \\ \hat{\overrightarrow{\beta}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\overrightarrow{Y} \end{array}\right.$$

This corresponds to (1). It can be shown that these estimators are unbiased for $\overrightarrow{\beta}$, i.e. $\mathbb{E}\hat{\overrightarrow{\beta}} = \overrightarrow{\beta}$.

By using these coefficients we obtain that the vector of fitted values is

$$\hat{\overrightarrow{Y}} = \mathbb{X}\hat{\overrightarrow{\beta}} + \overrightarrow{\varepsilon} = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\overrightarrow{X} + \overrightarrow{\varepsilon}$$

These estimator correspond to the Maximum Likelihood Estimator because the errors are assumed to be normally distributed.

Example 1:

In the next data set $Y =$ `Earn` is the monthly salary in EUR of 48 people chosen at random from a population.

$X_1 =$ `s` are the years spent for education in school/university.
$X_2 =$ `c` are the results from a cognitive test for imagination.

a. Model the dependence of the monthly salary of a person from this population from the results from a cognitive test for imagination;
b. Model the dependence of the monthly salary of a person from this population from the years spent for education in school/university;
c. Model the dependence of the monthly salary of a person from this population from the results from a cognitive test for imagination and the years spent for education in school/university;
d. Determine the expected monthly salary of a person from this population of he/she had spent $16.$
 years in educating system and her/his results from the cognitive test are $89.$
e. Determine the expected monthly salary of these persons having in mind the years that he/she had spent in educating system and her/his results from the cognitive test.
f. Find and plot the errors(residuals): $\varepsilon_i, i = 1, 2, \ldots, n$ in the multiple regression model.
g. Determine the mean square error of the multiple model.
h. Compute the coefficient of deteremination.
i. Check if in the multiple model $\mathbb{E}\varepsilon = 0.$
j. Check if the errors in the multiple model are normal.

We have 2 regressors

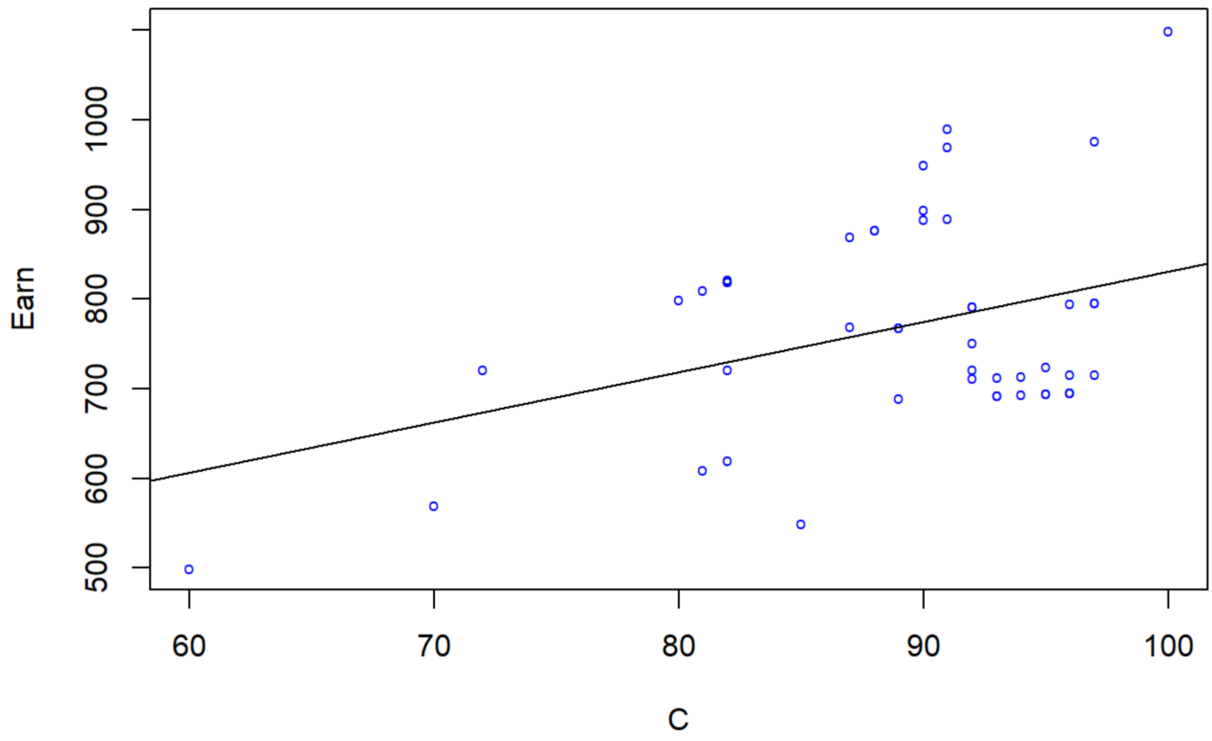
```
> S <- c(8, 8, 10, 13, 13, 13, 13, 13, 13, 13, 13, 12,
12, 12, 12,
+ 12, 12, 12, 12, 12, 12, 12, 15, 17, 18, 19, 19, 19, 15,
17, 17,
+ 17, 17, 17, 17, 17, 16, 16, 16, 16, 16, 16, 16, 16, 16,
16, 16, 13)
> C <- c(60, 70, 85, 87, 89, 90, 82, 81, 80, 87, 82, 81,
82, 82, 72,
+ 82, 92, 90, 92, 89, 89, 88, 88, 91, 91, 97, 100, 96,
92, 93, 94,
+ 95, 96, 97, 97, 97, 96, 96, 95, 93, 96, 94, 95, 92, 91,
90, 92, 93)
> n <- length(S); n
[1] 48
```

and the response as a linear function of the regressors

```
> Earn <- c(500, 570, 550, 770, 690, 900, 620, 610, 800,
870, 820,
+ 810, 820, 722, 722, 822, 722, 950, 752, 769, 769, 878,
878, 971,
+ 991, 977, 1100, 796, 712, 713, 714, 725, 716, 717, 797,
797,
+ 696, 696, 695, 693, 696, 694, 695, 792, 891, 890, 792,
693)
> df = data.frame(Earn, S, C);
```

a.

```
> plot(df$C, df$Earn, pch = "o", col='blue', cex = 0.6,
xlab = 'C', ylab = 'Earn')
> abline(lm(Earn ~ C))
```




```
> lm(Earn ~ C)

Call:
lm(formula = Earn ~ C)

Coefficients:
(Intercept)            C
    268.885        5.622
> mymodelEarnC <- lm(Earn ~ C)
> summary(mymodelEarnC)

Call:
lm(formula = Earn ~ C)

Residuals:
    Min      1Q  Median      3Q     Max
-196.75  -97.60  -14.91   90.61  268.92

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  268.885    184.919   1.454  0.15272
```

```
C                 5.622      2.067   2.720  0.00917 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 110.5 on 46 degrees of freedom
Multiple R-squared:  0.1386,    Adjusted R-squared:
0.1199
F-statistic: 7.401 on 1 and 46 DF,  p-value: 0.009172
```

The model is

$$Earn = 268.885 + 5.622C + \varepsilon$$

The `summary` function returns:

- the method
- the five-number summary of the residuals
- the coefficients - estimates, standard error, t-value and p-value ($H_0 : \beta_i = 0$, $h_A : \beta_i \neq 0$), small p-value is flagged with `***` and means that the coefficients are statistically significant.

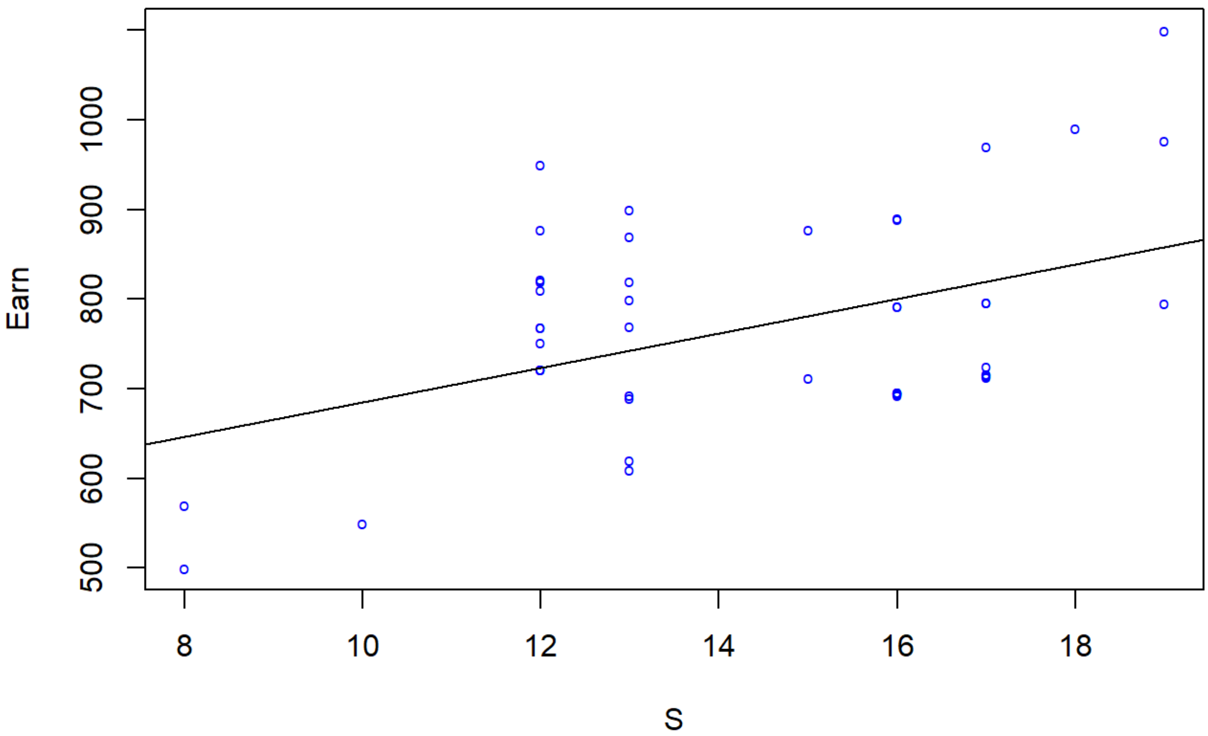
Other test of hypotheses are easily done knowing estimates, standard error and standard error for the residuals.

R^2 is interpreted as the **fraction of the variance explained by the model**.

Finally the F-statistic is given. The p-value for this is from the hypotheses test that $H_0 : \beta_1 = \beta_2 = \ldots = \beta_r = 0$. Meaning that the regression is not appropriate. The theory for this comes from that of the **analysis of variance (ANOVA)** that we will speak about in the next topic.

b.

```
> plot(S, Earn, pch = "o", col='blue', cex = 0.6, xlab =
'S', ylab = 'Earn')
> abline(lm(Earn ~ S))
```




```
> lm(Earn ~ S)

Call:
lm(formula = Earn ~ S)

Coefficients:
(Intercept)            S
     493.15        19.21
> mymodelEarnS <- lm(Earn ~ S)
> summary(mymodelEarnS)

Call:
lm(formula = Earn ~ S)

Residuals:
     Min       1Q   Median       3Q      Max
-146.811 -104.475   -8.475   89.775  241.901

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  493.146     84.799   5.815 5.47e-07 ***
```

```
S                19.208      5.784   3.321  0.00176 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 106.9 on 46 degrees of freedom
Multiple R-squared:  0.1934,    Adjusted R-squared:
0.1759
F-statistic: 11.03 on 1 and 46 DF,  p-value: 0.001763
```

The model is

$$Earn = 493.146 + 19.208S + \varepsilon$$

The summary function returns:

- the method
- the five-number summary of the residuals
- the coefficients - estimates, standard error, $t-value = t_{emp}$ and p-value for testing $H_0 : \beta_i = 0$, against $H_A : \beta_i \neq 0$. Small p-value is flagged with `***` and means that the coefficients are statistically significant.

Other test of hypotheses are easily done after knowing these estimates, their standard errors and the standard error for the residuals.

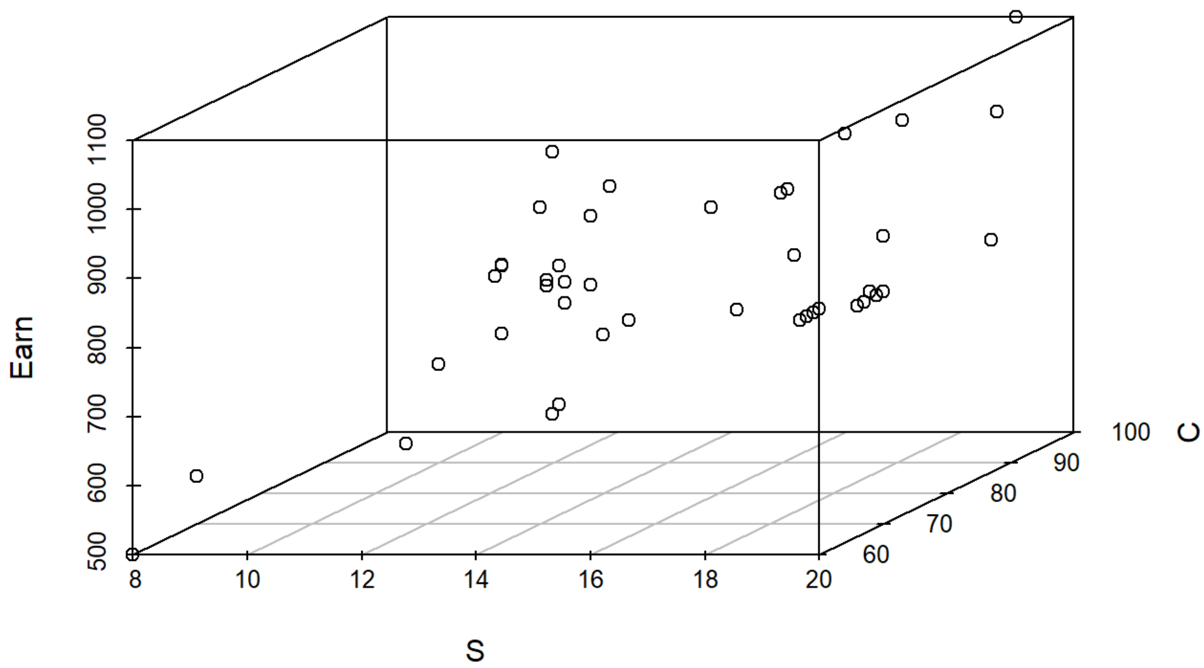
R^2 is the coefficient of determination and is interpreted as the **fraction of the variance explained by the model**.

Finally the F-statistic is given. It will be explained later on. Its p-value is for testing hypotheses $H_0 : \beta_1 = \beta_2 = \ldots = \beta_r = 0$. Meaning that the regression is not appropriate. The model is NOT adequate. The independent variables dos not determine Y at all. The theory for this comes from that of the **analysis of variance (ANOVA)** that we will speak about in the next topic.

c.

```
> library(scatterplot3d)
Warning: package 'scatterplot3d' was built under R
version 4.0.3
```

```
> scatterplot3d(S, C, Earn)
```




```
> library(rgl)
> open3d()
wgl
  1
> plot3d(S, C, Earn, col = "red", size = 3)
```

In order to estimate the coefficients in the model

$$Earn = \beta_0 + \beta_1 S + \beta_2 C + \varepsilon$$

we use again the function lm.

```
> mymodel <- lm(Earn ~ S + C, data = df)
> summary(mymodel)

Call:
lm(formula = Earn ~ S + C, data = df)

Residuals:
     Min          1Q    Median          3Q        Max
```

```
-139.897 -104.855   -7.961   91.739  241.778

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 451.0495   208.2405   2.166   0.0356 *
S            17.4389     9.8882   1.764   0.0846 .
C             0.7583     3.4189   0.222   0.8255
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 108 on 45 degrees of freedom
Multiple R-squared:  0.1943,    Adjusted R-squared:
0.1585
F-statistic: 5.425 on 2 and 45 DF,  p-value: 0.007748
```

The plane we are looking for is

$$Earn = 451.0495 + 17.4389S + 0.7583C.$$

If C and S were independent we would have one and the same coefficients in front of the same independents variables in the simple regression models considered in a) and b). The last means that **we observe multicollinearity**.

We can see the components of `mymodel` by

```
> ls(mymodel)
 [1] "assign"        "call"          "coefficients"
"df.residual"
 [5] "effects"       "fitted.values" "model"         "qr"
 [9] "rank"          "residuals"     "terms"
"xlevels"
```

d. By using this equation we obtain that

```
> Earn_16_89 <- mymodel$coefficients[1] +
mymodel$coefficients[2] * 16 +  mymodel$coefficients[3] *
89; Earn_16_89
(Intercept)
   797.5635
```

the expected monthly salary of a person from this population if he/she had spent 16 years in educating system and her/his results from the cognitive test are 89 is 797.5606 EUR.

We can see the coefficients via the function `summary`

```
> summary(mymodel)

Call:
lm(formula = Earn ~ S + C, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-139.897 -104.855   -7.961   91.739  241.778 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 451.0495   208.2405   2.166   0.0356 *
S            17.4389     9.8882   1.764   0.0846 .
C             0.7583     3.4189   0.222   0.8255
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 108 on 45 degrees of freedom
Multiple R-squared:  0.1943,    Adjusted R-squared:
0.1585
F-statistic: 5.425 on 2 and 45 DF,  p-value: 0.007748
```

e.

```
> yhat <- mymodel$fitted.values; yhat
       1        2        3        4        5        6
7        8
636.0606 643.6439 689.8967 743.7301 745.2468 746.0051
739.9385 739.1801
       9       10       11       12       13       14
15       16
738.4218 743.7301 739.9385 721.7412 722.4995 722.4995
714.9163 722.4995
      17       18       19       20       21       22
23       24
```

```
730.0828 728.5662 730.0828 727.8078 727.8078 727.0495
779.3663 816.5191
      25       26       27       28       29       30
31       32
833.9580 855.9469 858.2219 855.1886 782.3996 818.0358
818.7941 819.5524
      33       34       35       36       37       38
39       40
820.3108 821.0691 821.0691 821.0691 802.8718 802.8718
802.1135 800.5969
      41       42       43       44       45       46
47       48
802.8718 801.3552 802.1135 799.8385 799.0802 798.3219
799.8385 748.2801
```

or

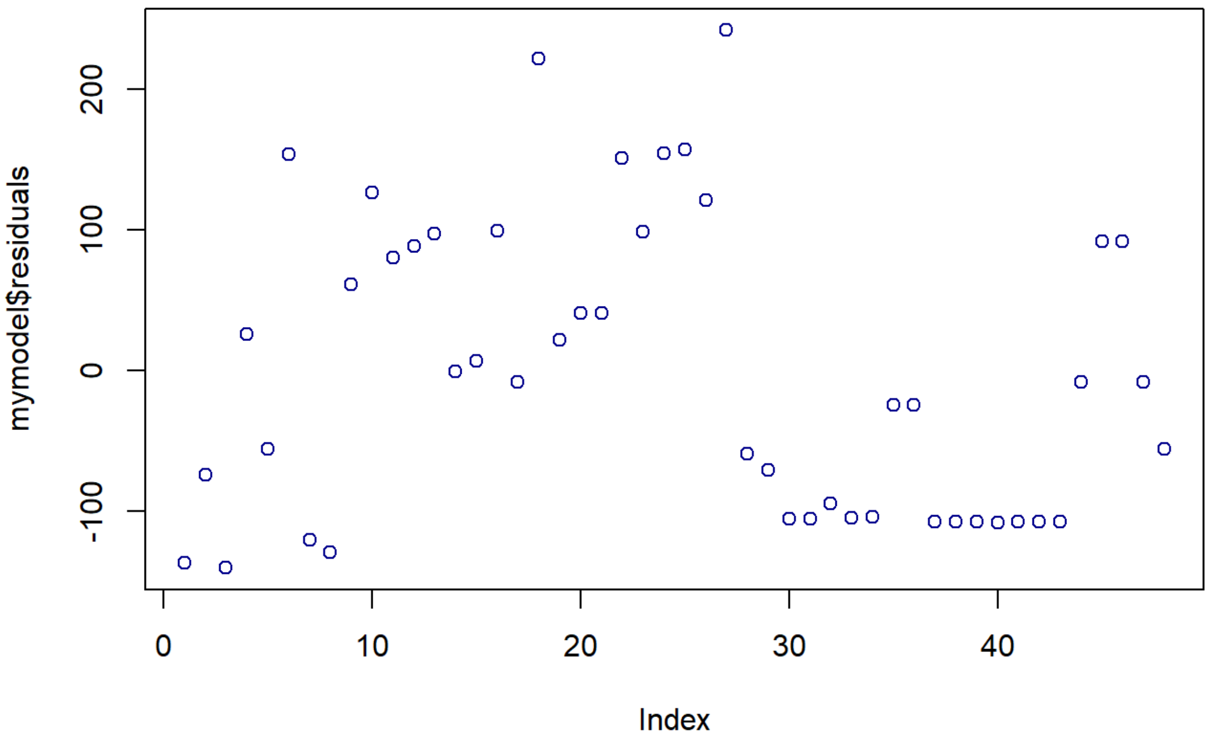
```
> yhat <- mymodel$coefficients[1] +
mymodel$coefficients[2] * S +  mymodel$coefficients[3] *
C; yhat
 [1] 636.0606 643.6439 689.8967 743.7301 745.2468
746.0051 739.9385 739.1801
 [9] 738.4218 743.7301 739.9385 721.7412 722.4995
722.4995 714.9163 722.4995
[17] 730.0828 728.5662 730.0828 727.8078 727.8078
727.0495 779.3663 816.5191
[25] 833.9580 855.9469 858.2219 855.1886 782.3996
818.0358 818.7941 819.5524
[33] 820.3108 821.0691 821.0691 821.0691 802.8718
802.8718 802.1135 800.5969
[41] 802.8718 801.3552 802.1135 799.8385 799.0802
798.3219 799.8385 748.2801
```

f.

```
> e <- resid(mymodel); e
           1            2            3            4
5            6
-136.0606242  -73.6439079 -139.8966818   26.2698889
-55.2467679  153.9949038
           7            8            9           10
11           12
```

```
-119.9384693 -129.1801409   61.5781875  126.2698889
80.0615307   88.2587833
          13           14           15           16
17           18
  97.5004549   -0.4995451    7.0837386   99.5004549
-8.0828288  221.4338279
          19           20           21           22
23           24
  21.9171712   41.1921563   41.1921563  150.9504847
98.6337122  154.4808787
          25           26           27           28
29           30
 157.0419545  121.0530601  241.7780750  -59.1886115
-70.3996013 -105.0357781
          31           32           33           34
35           36
-104.7941064  -94.5524348 -104.3107632 -104.0690915
-24.0690915  -24.0690915
          37           38           39           40
41           42
-106.8718390 -106.8718390 -107.1135106 -107.5968539
-106.8718390 -107.3551823
          43           44           45           46
47           48
-107.1135106   -7.8385255   91.9198029   91.6781312
-7.8385255  -55.2800813
```

```
> plot(mymodel$residuals, col = "darkblue")
```



or by using the formula

```
> e <- Earn - yhat; e
 [1] -136.0606242  -73.6439079 -139.8966818   26.2698889
-55.2467679
 [6]  153.9949038 -119.9384693 -129.1801409   61.5781875
126.2698889
[11]   80.0615307   88.2587833   97.5004549   -0.4995451
7.0837386
[16]   99.5004549   -8.0828288  221.4338279   21.9171712
41.1921563
[21]   41.1921563  150.9504847   98.6337122  154.4808787
157.0419545
[26]  121.0530601  241.7780750  -59.1886115  -70.3996013
-105.0357781
[31] -104.7941064  -94.5524348 -104.3107632 -104.0690915
-24.0690915
[36]  -24.0690915 -106.8718390 -106.8718390 -107.1135106
-107.5968539
[41] -106.8718390 -107.3551823 -107.1135106   -7.8385255
91.9198029
[46]   91.6781312   -7.8385255  -55.2800813
```

g.) It is time to determine the **mean square error** of the multiple model.

$$MSE = RSE^2 = S_\varepsilon^2 = \frac{1}{n-3}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 = \frac{1}{n-3}\sum_{i=1}^{n}\varepsilon_i^2$$

It is an unbiased estimator of σ_ε^2. The denominator $n-3$ comes from the fact that there are three values estimated from the data: β_0, β_1 and β_2.

Let us remind that

$$SSE = \sum_{i=1}^{n}\varepsilon_i^2, \quad MSE = \frac{SSE}{n-r} = \frac{SSE}{n-3}$$

```
> SSE <- sum(e^2); SSE
[1] 525183
> MSE <- SSE / (n - 3); MSE
[1] 11670.73
> s <- sqrt(MSE); s
[1] 108.0312
```

The **Residual Standard error** is

$$S_\varepsilon = \sqrt{MSE} = \sqrt{\frac{SSE}{n-3}} = 108.0312\ EUR$$

or we can extract it via the function `summary`

```
> summary(mymodel)

Call:
lm(formula = Earn ~ S + C, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-139.897 -104.855   -7.961   91.739  241.778

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 451.0495   208.2405   2.166   0.0356 *
S            17.4389     9.8882   1.764   0.0846 .
C             0.7583     3.4189   0.222   0.8255
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
Residual standard error: 108 on 45 degrees of freedom
Multiple R-squared:  0.1943,    Adjusted R-squared:
0.1585
F-statistic: 5.425 on 2 and 45 DF,  p-value: 0.007748
```

h. Via the function `summary` we can estimate also the **coefficient of determination**. Note that it is **Adjusted R-squared: 0.1585**

$\operatorname{cor}^2(X, Y) = 1 - \frac{\mathbb{E}\varepsilon^2}{\mathbb{D}Y}$, $R^2 = Adjusted\ R{-}suared = 01585$

The coefficient is not close to 1, therefore, we cannot say that the independent variables

X_1 - the years spent for education in school/university, and

X_2 - the results from a cognitive test for imagination are important for the value of the dependent variable

$Y = Earn$ - the monthly salary in EUR for peoples in this population.

We can determine it also via the formula

```
> Rsquare <- 1 - MSE/var(Earn); Rsquare
[1] 0.1584657
```

The other result **Multiple R-squared: 0.1942757** does not take into account that the denominators of the estimators S_ε^2 and $S_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y}_n)^2$ are different and computes

$$Multiple\ R{-}squared = 1 - \frac{SSE}{\sum_{i=1}^{n}(Y_i - \overline{Y}_n)^2} = 0.1942757$$

```
> Rsq <- 1 - SSE/sum((Earn - mean(Earn))^2); Rsq
[1] 0.1942757
```

i. In order to check if $\mathbb{E}\varepsilon = 0$ we use `t-test`.

$$H_0 : \mathbb{E}\varepsilon = 0$$
$$H_A : \mathbb{E}\varepsilon \neq 0$$

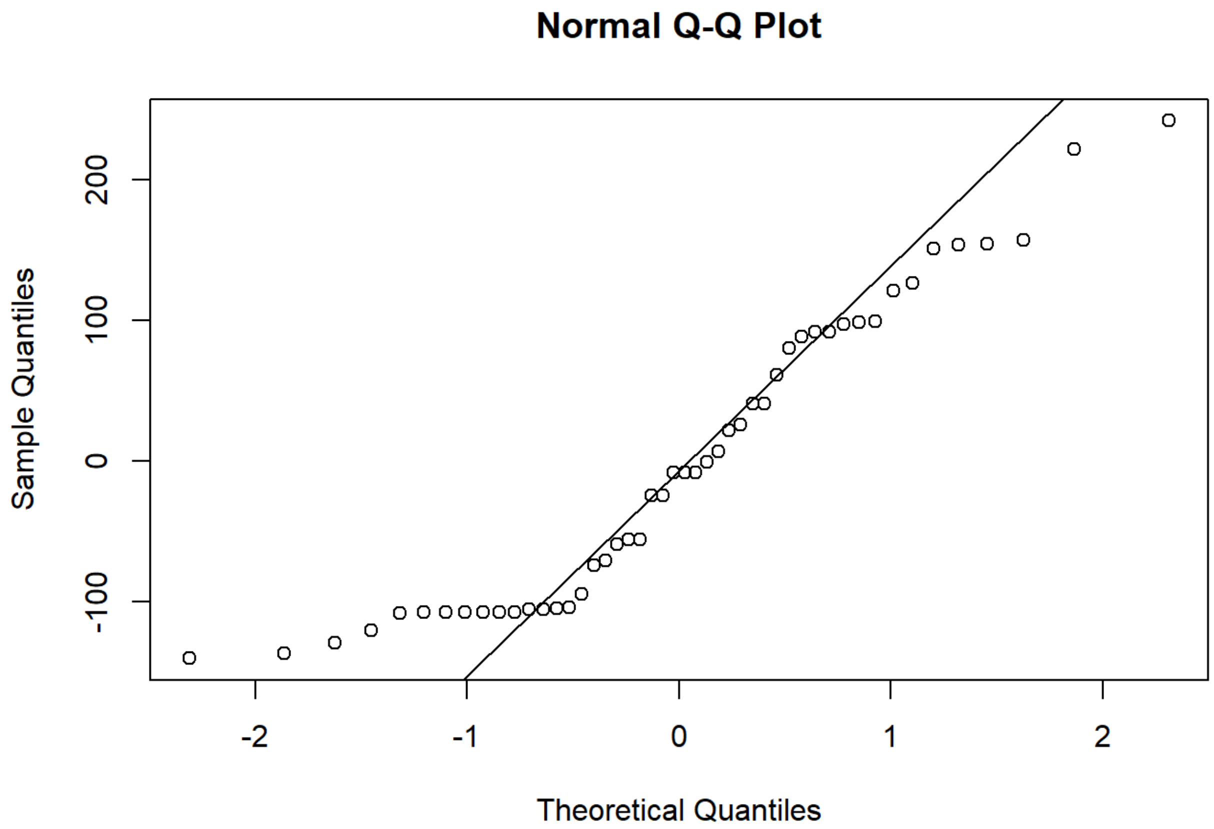
```
> mean(e)
[1] -3.434317e-13
> n <- length(e); n
[1] 48
> rse <- sqrt(MSE); rse
[1] 108.0312
> temp <- abs(mean(e) - 0) / (rse / sqrt(n)); temp
[1] 2.20248e-14
> pvalue <- 2 * pt(temp, n - 1, lower.tail = FALSE);
pvalue
[1] 1
```

The $p-value = 1 > 0.05 = \alpha$, so we have no evidence to reject H_0.

k. The next step is to test the assumptions of the model that the residuals are i.i.d. normally distributed $\varepsilon_i \in N(0,\sigma_\varepsilon^2)$

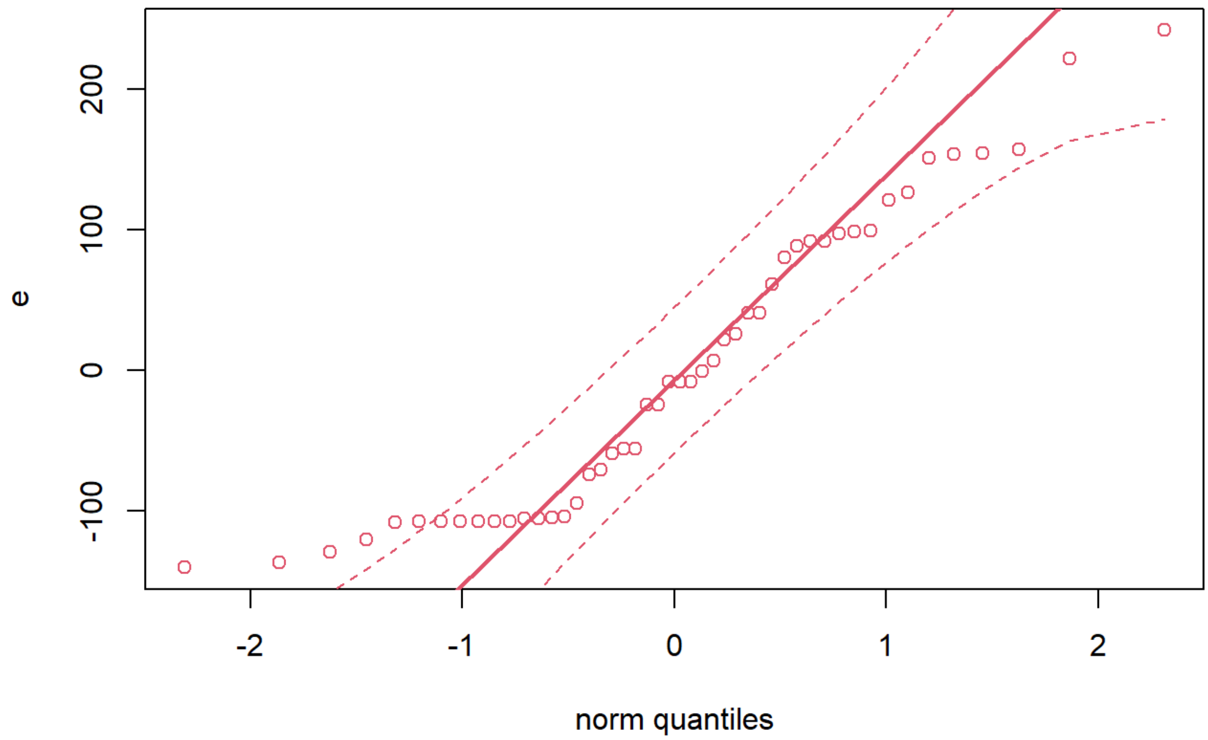
First we make the normal `qq-plot`

```
> qqnorm(e)
> qqline(e)
```




```
> library(StatDA)
Warning: package 'StatDA' was built under R version 4.0.3
Loading required package: sgeostat
Warning: package 'sgeostat' was built under R version 
4.0.3
Registered S3 method overwritten by 'geoR':
  method         from
  plot.variogram sgeostat
```

```
> qqplot.das(e)
```



We can perform also Shapiro test

H_0 : ε is normally distributed

H_A : ε is not normally distributed

We use the function `shapiro.test` in R

```
> shapiro.test(e)

	Shapiro-Wilk normality test

data:  e
W = 0.91997, p-value = 0.002966
```

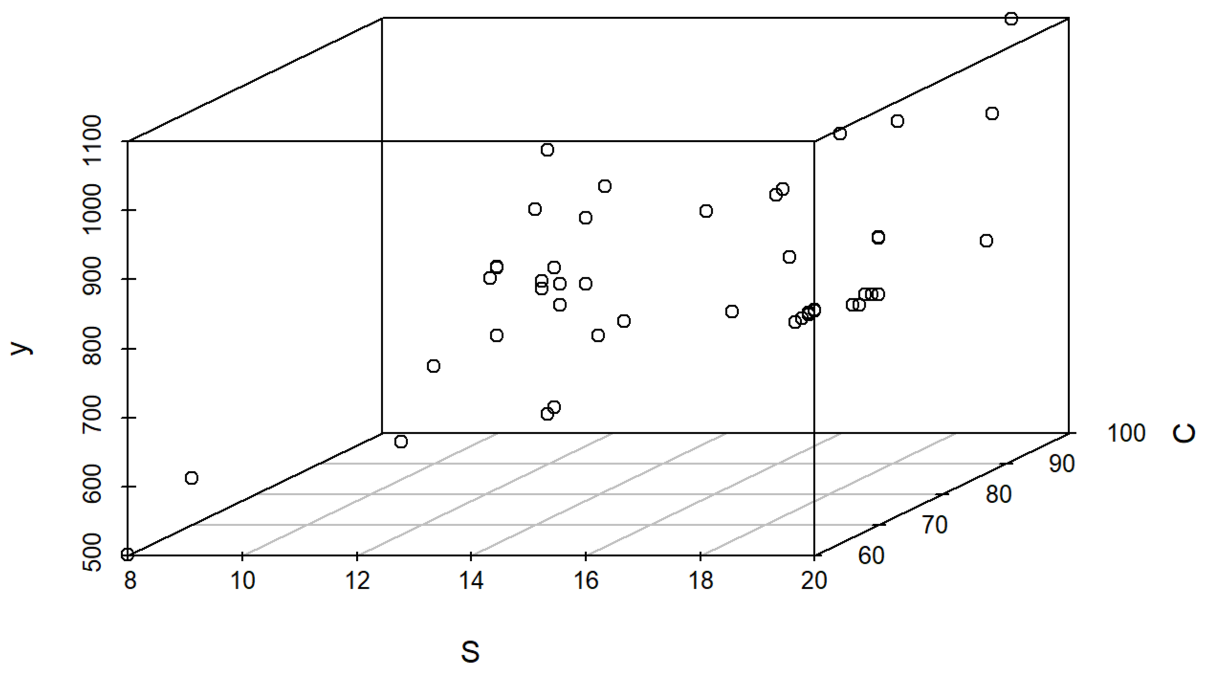
The $p-value = 0.002966 < 0.05 = \alpha$, so we reject H_0. We have no reason to assume that the data come from normal distribution.

Confidence intervals for $\hat{Y} = (Y \mid X)$

If the errors are normal we can estimate the accuracy of these considerations.

Let us add some normal noise $\varepsilon \in N(0, 2^2)$ with a small variance and see what will happen with the response variable $Y = Earn$

```
> y <- Earn + rnorm(n, 0, 2)
> scatterplot3d(S, C, y)
```



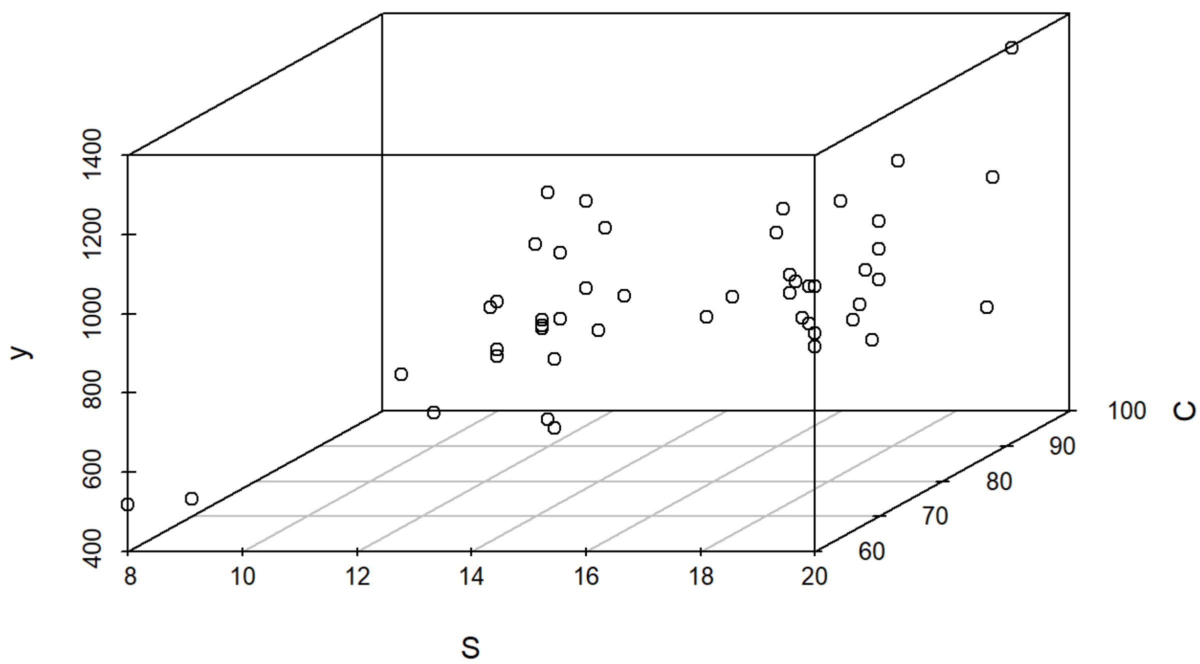

```
> mymodel <- lm(y ~ S + C, data = df)
> summary(mymodel)

Call:
lm(formula = y ~ S + C, data = df)

Residuals:
     Min        1Q    Median        3Q       Max 
-136.763 -105.727    -9.923    91.378   242.017 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 449.3321   208.2493   2.158   0.0363 *
S            17.2420     9.8886   1.744   0.0881 .
C             0.8045     3.4191   0.235   0.8151
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 108 on 45 degrees of freedom
Multiple R-squared:  0.1929,     Adjusted R-squared:
0.157
F-statistic: 5.378 on 2 and 45 DF,  p-value: 0.008051
```

We observe that the small variance almost does not change the model.

And what will happen if we add normal noise $\varepsilon \in N(0,\, 100^2)$ with a higher variance

```
> y <- Earn + rnorm(n, 0, 100)
> scatterplot3d(S, C, y)
```



```
> mymodel <- lm(y ~ S + C, data = df)
> summary(mymodel)

Call:
lm(formula = y ~ S + C, data = df)

Residuals:
    Min       1Q   Median       3Q      Max 
-239.25   -90.99   -30.88   110.70   415.91 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  121.329    295.688   0.410    0.684
S             13.538     14.041   0.964    0.340
C              5.196      4.855   1.070    0.290

Residual standard error: 153.4 on 45 degrees of freedom
Multiple R-squared:  0.1922,	Adjusted R-squared:  
0.1563 
F-statistic: 5.353 on 2 and 45 DF,  p-value: 0.008214
```

We observe that when we add more noise the guesses of $Y = Earn$ got worse and worse. The more noise the worse the confidence. Later on we will see that the more data the better the confidence.

For the confidence intervals we will need the estimator of

$$
\begin{aligned}
\operatorname{cov}(\hat{\vec{\beta}}) &= \operatorname{cov}((\mathbb{X}^T\mathbb{X})^{-1}\vec{Y} = \operatorname{cov}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(\mathbb{X})\vec{\beta} + \vec{\varepsilon} = \\
&= \operatorname{cov} \vec{\beta} + \mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\vec{\varepsilon} = \operatorname{cov}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\vec{\varepsilon}) = \\
&= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \operatorname{cov}(\vec{\varepsilon})((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T)^T = \\
&= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\sigma_\varepsilon^2\mathbb{I}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T)^T = \\
&= \sigma_\varepsilon^2(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T)^T = \\
&= \sigma_\varepsilon^2(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}((\mathbb{X}^T\mathbb{X})^{-1})^T = \sigma_\varepsilon^2((\mathbb{X}^T\mathbb{X})^{-1})^T = \\
&= \sigma_\varepsilon^2(\mathbb{X}^T\mathbb{X})^{-1}
\end{aligned}
$$

Therefore,

$$\hat{\overrightarrow{\beta}} \in N(\overrightarrow{\beta}; \sigma_\varepsilon^2(\mathbb{X}^T\mathbb{X})^{-1})$$

and the unbiased estimator of $\mathbb{D}\hat{\beta}_i$ is

$$S_{\beta_i}^2 := S_\varepsilon^2((\mathbb{X}^T\mathbb{X})^{-1})_{ii}, SE(\beta_i) = S_{\beta_i}, i = 1, 2, \ldots, r.$$

Correspondingly

$$\frac{\hat{\beta}_i - \beta_i}{S_{\beta_i}} \in t(n-r-1), i = 1, 2, \ldots, r$$

And the $(1-\alpha)100\,\%$ confidence interval for β_i is

$$[\hat{\beta}_i - t_{1-\frac{\alpha}{2};t(n-r-1)S_{\beta_i}}; \hat{\beta}_i + t_{1-\frac{\alpha}{2};t(n-r-1)S_{\beta_i}}], i = 1, 2, \ldots, r$$

The `summary` function returns the estimators $\hat{\beta}_i$ of the coefficients β_i, $i = 1, 2, \ldots, r$ their standard errors $SE(\beta_i) = S_{\beta_i}$, the corresponding $t-values = t_{emp} = \dfrac{\hat{\beta}_i - 0}{S_{\beta_i}}$ and the corresponding $t-values = \mathbb{P}(|\eta| > t_{emp})$, where $\eta \in t(n-r-1)$. They can be used for testing

$H_0 : \beta_i = 0$
$H_A : \beta_i \neq 0$

The small p-value is flagged with `***` and means that the coefficients are statistically significant.

Other test of hypotheses are easily done knowing estimates, standard error and standard error for the residuals.

When predict Y given $\overrightarrow{X}$ we will need

$$\mathbb{D}(\hat{Y}) = \mathbb{D}(Y\overrightarrow{X}) = \text{cov}(\hat{\overrightarrow{\beta}}^T\overrightarrow{X}, \hat{\overrightarrow{\beta}}^T\overrightarrow{X}) = \overrightarrow{X}\,\text{cov}(\hat{\overrightarrow{\beta}})\,\overrightarrow{X}^T =$$
$$= \overrightarrow{X}\sigma_\varepsilon^2(\mathbb{X}^T\mathbb{X})^{-1}\overrightarrow{X}^T = \sigma_\varepsilon^2\overrightarrow{X}(\mathbb{X}^T\mathbb{X})^{-1}\overrightarrow{X}^T$$

and therefore, we estimate it via

$$S_{\hat{Y}}^2 = S_\varepsilon^2\overrightarrow{X}(\mathbb{X}^T\mathbb{X})^{-1}\overrightarrow{X}^T$$

We also obtained that

$$\hat{Y} = (Y\,|\,\overrightarrow{X}) \in N(\overrightarrow{\beta}^T\overrightarrow{X};\ \sigma_\varepsilon^2\overrightarrow{X}(\mathbb{X}^T\mathbb{X})^{-1}\overrightarrow{X}^T)$$

therefore, the $(1-\alpha)100\,\%$

$$[\hat{\overrightarrow{\beta}}^T\overrightarrow{X} - t_{1-\frac{\alpha}{2};t(n-r-1)}S_{\hat{Y}};\ \hat{\overrightarrow{\beta}}^T\overrightarrow{X} + t_{1-\frac{\alpha}{2};t(n-r-1)}S_{\hat{Y}}].$$

Example 2

The `homeprice` data set contains information about homes that sold in a town of New Jersey in the year 2001. We want to figure out what are the appropriate prices in $1000\$$ (denoted by `list`) for homes.

```
> library(UsingR)
Warning: package 'UsingR' was built under R version 4.0.3
Loading required package: MASS
Loading required package: HistData
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2

Attaching package: 'Hmisc'
The following objects are masked from 'package:base':

    format.pval, units

Attaching package: 'UsingR'
```

```
The following object is masked from 'package:survival':

    cancer
> head(homeprice)
   list  sale full half bedrooms rooms neighborhood
1  80.0 117.7    1    0        3     6            1
2 151.4 151.0    1    0        4     7            1
3 310.0 300.0    2    1        4     9            3
4 295.0 275.0    2    1        4     8            3
5 339.0 340.0    2    0        3     7            4
6 337.5 337.5    1    1        4     8            3

> attach(homeprice)
```

a. Model the dependence of the prices of homes from this population from the number of full bathrooms.

b. Model the dependence of the prices of homes from this population from the number of bedrooms. What is the change of the price for one more bedroom? May we say that an additional bedroom increases the price with 15000$?

c. Model the dependence of the prices of homes from this population from the number of rooms. What is the influence of one more room on the price of the home?

d. Model the dependence of the prices of homes from this population from the points for neighbourhood. What is the change of the price for one more point for neighbourhood?

e. Model the dependence of the prices of homes from this population from the points for neighbourhood and rooms.

f. Model the dependence of the prices of homes from this population from the number of bedrooms and the points for neighbourhood.

g. Model the dependence of the prices of homes from this population from the number of full bathrooms, bedrooms and the points for neighbourhood. Check the hypothesis that we heed to pay 15000$ more per full bathroom?

h. Model the dependence of the prices of homes from this population from the number of full bathrooms, bedrooms and the points for neighbourhood (without cut). Is it acceptable the intercept to be 0?

i. Determine the expected price of a home from this population if it has 3 rooms, 2 bedrooms and 2 points for neighbourhood.

j. Determine the expected price of these homes having in mind the numbers of their rooms, full bathrooms and the points for neighbourhood.

k. Find and plot the errors(residuals): $\varepsilon_i, i = 1, 2, \ldots, n$ in the model in j).

l. Determine the mean square error (MSE) of the model in j).

m. Compute the coefficient of determination (R^2) of the model in j).

n. Check if in the model in j) $\mathbb{E}\varepsilon = 0$.

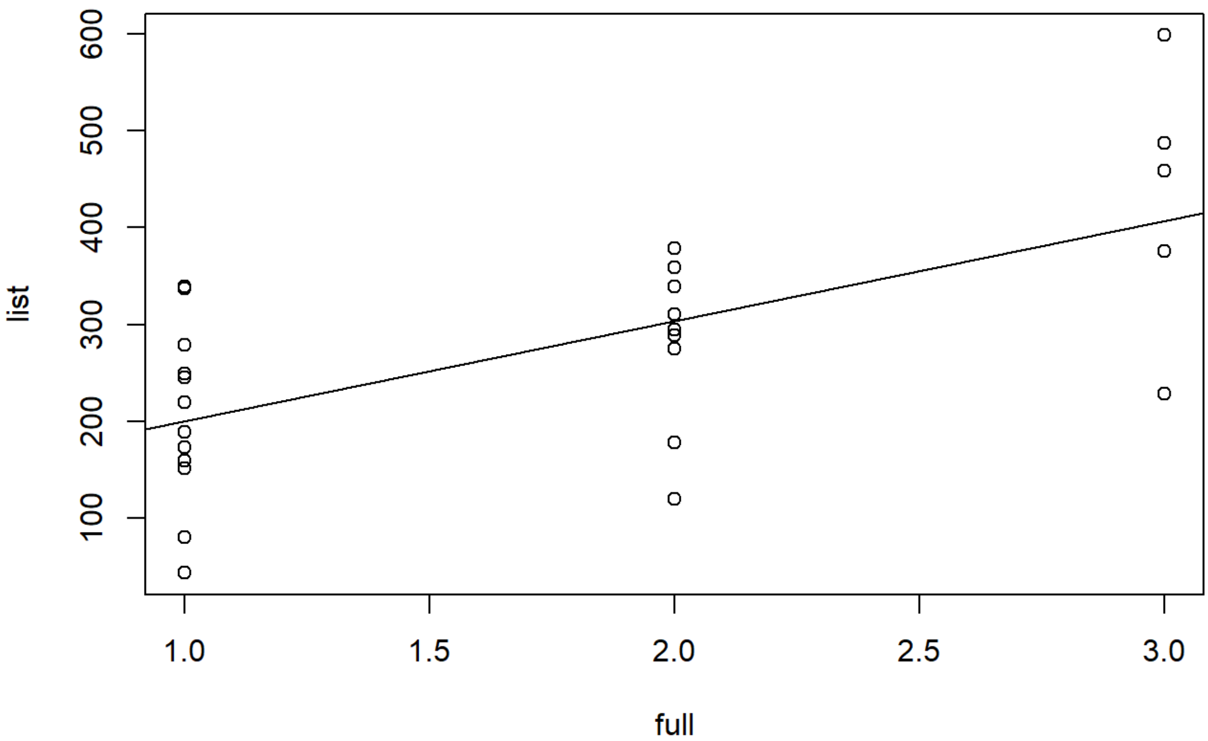
o. Check if the errors in the model in j) are normal.

p. Determine 95% confidence intervals for the expected price $\hat{Y}$ of these homes having in mind the numbers of their `rooms`, `full` bathrooms and the points for `neighbourhood`.

Solution:

a. Model the dependence of the prices (`list`) of homes from this population from the number of `full` bathrooms.

```
> modelPriceBathroom <- lm(list ~ full)
> plot(full, list)
> abline(lm(list ~ full))
```




```
> summary(modelPriceBathroom)

Call:
lm(formula = list ~ full)

Residuals:
     Min       1Q   Median       3Q      Max 
-184.435  -31.062   -8.435   51.938  191.938 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    96.18      44.16   2.178 0.038329 *  
full          103.63      23.55   4.400 0.000152 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 93.58 on 27 degrees of freedom
Multiple R-squared:  0.4177,    Adjusted R-squared:
0.3961 
F-statistic: 19.36 on 1 and 27 DF,  p-value: 0.0001523
```

$$list = 96.18 + 103.63 full + \varepsilon$$

One more `full` bathroom increases the price with $103.63 \times 1000\$$. In order to compute the $95\$$ confidence interval we use the function

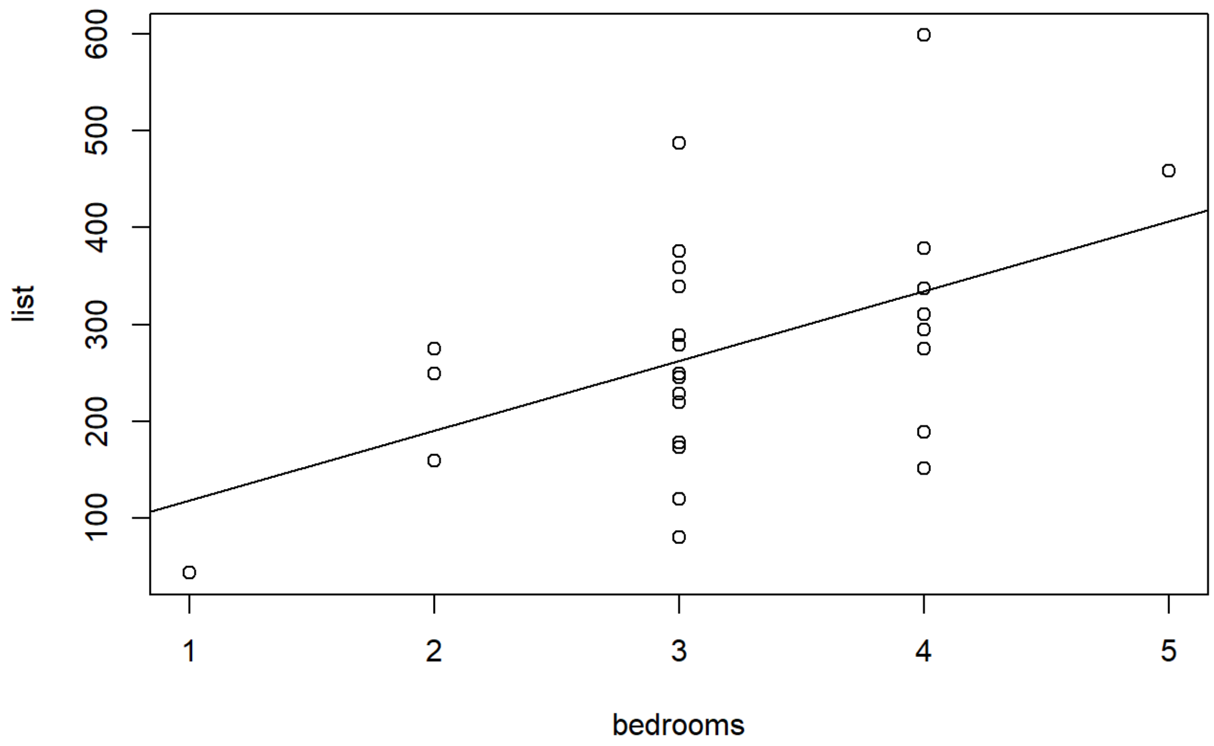
```
> myCI = function(b, SE, t) {
+   b + c(-1, 1) * SE * t
+ }
```

In this case first we have to compute

```
> e <- resid(modelPriceBathroom)
> n <- length(e)
> beta1hat <- modelPriceBathroom$coefficients[2];
beta1hat
    full
103.6266
> SSE <- sum(e^2)
> MSE <- SSE / (n-2)
> Seps <- sqrt(MSE)
> SEbeta1 <- Seps / sqrt(sum((full - mean(full))^2));
SEbeta1
[1] 23.54896
> alpha <- 0.05
> t <- qt(1 - alpha/2, n - 2, lower.tail = TRUE)
> myCI(beta1hat, SEbeta1, t)
[1]  55.30816 151.94512
```

b. Model the dependence of the prices `list` of homes from this population from the number of `bedrooms`. What is the change of the price for one more bedroom? May we say that an additional bedroom increases the price with 15000$?

```
> modelPriceBedrooms <- lm(list ~ bedrooms)
> plot(bedrooms, list)
> abline(lm(list ~ bedrooms))
```




```
> summary(modelPriceBedrooms )

Call:
lm(formula = list ~ bedrooms)

Residuals:
    Min      1Q  Median      3Q     Max 
-183.25  -59.65  -13.39   58.87  264.35 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)    45.61      82.44   0.553  0.58466   
bedrooms       72.26      25.21   2.866  0.00796 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 107.4 on 27 degrees of freedom
Multiple R-squared:  0.2332,	Adjusted R-squared:
0.2048
F-statistic: 8.213 on 1 and 27 DF,  p-value: 0.007962
```

.

$$list = 45.61 + 72.26\, bedrooms + \varepsilon$$

One more bedroom increases the price with $72.26 \times 1000\$$.

Let's compute $95\ \%$ confidence interval. In this case first we have to compute

```
> e <- resid(modelPriceBedrooms)
> n <- length(e)
> beta1hat <- modelPriceBedrooms$coefficients[2];
beta1hat
bedrooms
72.26065
> SSE <- sum(e^2)
> MSE <- SSE / (n-2)
> Seps <- sqrt(MSE)
> SEbeta1 <- Seps / sqrt(sum((bedrooms -
mean(bedrooms))^2)); SEbeta1
[1] 25.21457
> alpha <- 0.05
> t <- qt(1 - alpha/2, n - 2, lower.tail = TRUE)
> myCI(beta1hat, SEbeta1,t)
[1]  20.52462 123.99668
```

$H_0 : \beta_1 = 15$
$H_A : \beta_1 \neq 15$

Given α the critical area is

$$W_\alpha = \left\{ \frac{|\hat{\beta}_1 - 15|}{SE(\beta_1)} \geq t_{1-\frac{\alpha}{2};n-2} \right\}$$

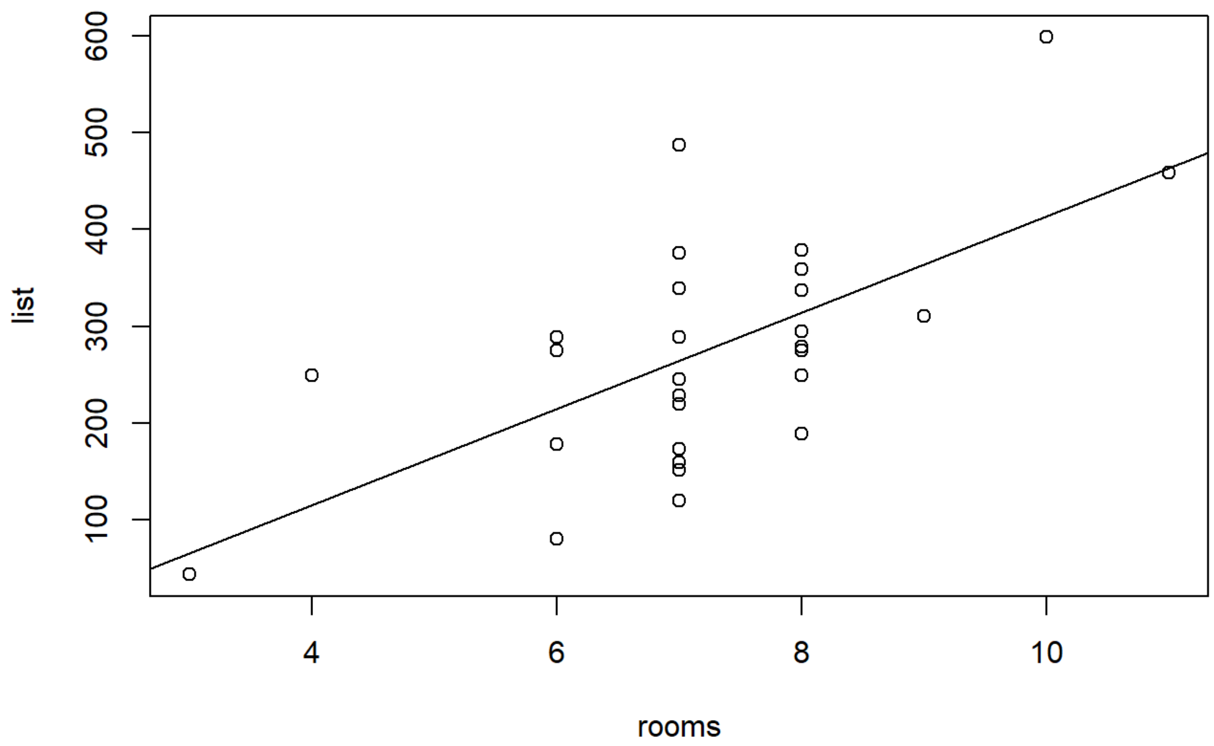
```
> b1 <- modelPriceBedrooms$coefficients[2]; b1
bedrooms
72.26065
> temp <- (b1 - 15) / SEbeta1; temp
bedrooms
2.270935
```

```
> pvalue <- 2 * pt(temp, n - 2, lower.tail =
FALSE);pvalue
  bedrooms
0.03134009
```

The $p-value = 0.03134009 < \alpha = 0.05$, so we reject H_0.

c. Let us now model the dependence of the prices `list` of homes from this population from the number of `rooms`.

```
> modelPriceRooms <- lm(list ~ rooms)
> plot(rooms, list)
> abline(lm(list ~ rooms))
```




```
> summary(modelPriceRooms )

Call:
lm(formula = list ~ rooms)

Residuals:
    Min       1Q   Median       3Q      Max
```

```
-145.54  -54.16  -19.54   64.65  223.46

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -84.10      87.20  -0.964 0.343355    
rooms          49.81      11.85   4.204 0.000257 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 95.34 on 27 degrees of freedom
Multiple R-squared:  0.3956,    Adjusted R-squared:
0.3732
F-statistic: 17.67 on 1 and 27 DF,  p-value: 0.0002575
```

$$list = -84.10 + 49.81\ rooms + \varepsilon$$

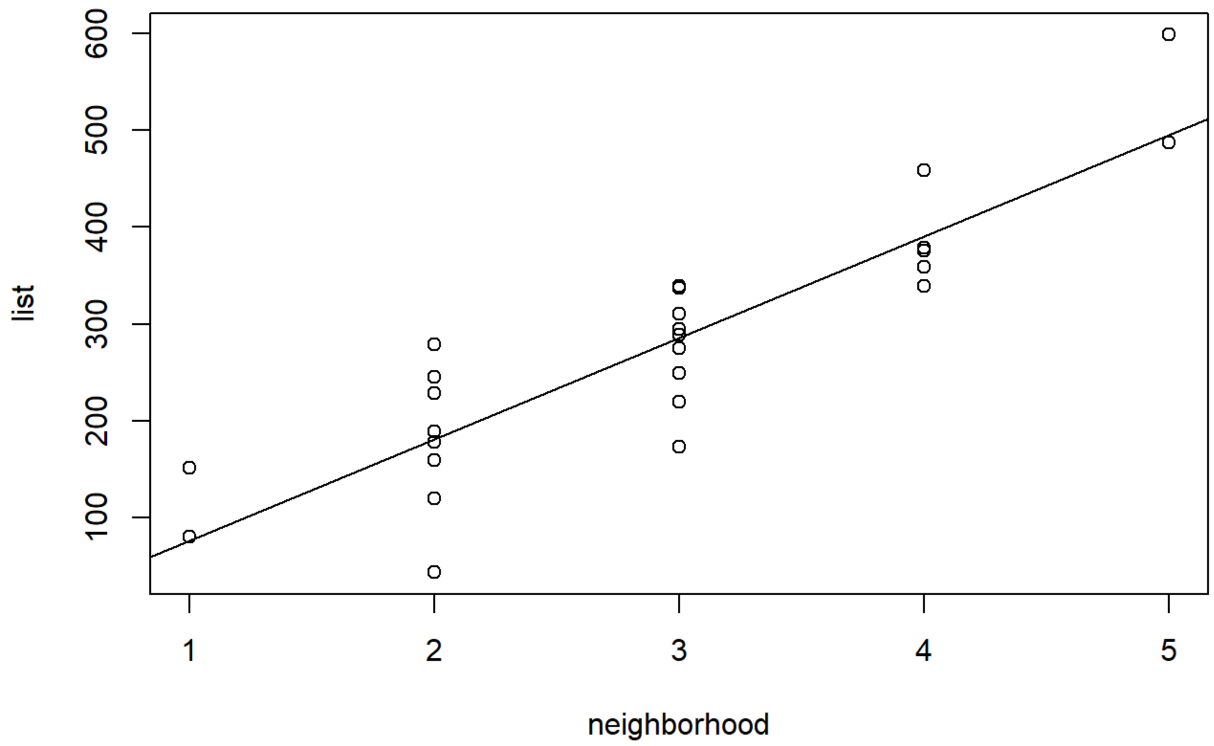
Let us now answer the question: What is the influence of one more room on the price of the home?

One more room increases the price with $49.81 \times 1000\$$. Let us now compute the corresponding $95\ \%$ confidence interval

```
> e <- resid(modelPriceRooms)
> n <- length(e)
> beta1hat <- modelPriceRooms$coefficients[2]; beta1hat
   rooms
49.80666
> SSE <- sum(e^2)
> MSE <- SSE / (n-2)
> Seps <- sqrt(MSE)
> SEbeta1 <- Seps / sqrt(sum((rooms - mean(rooms))^2));
SEbeta1
[1] 11.84735
> alpha <- 0.05
> t <- qt(1 - alpha/2, n - 2, lower.tail = TRUE)
> myCI(beta1hat, SEbeta1,t)
[1] 25.49791 74.11540
```

d. Let us now model the dependence of the prices `list` of homes from this population from the points for `neighbourhood`.

```
> modelPriceNeighbourhood <- lm(list ~ neighbourhood)
> plot(neighbourhood, list)
> abline(lm(list ~ neighbourhood))
```




```
> summary(modelPriceNeighbourhood)

Call:
lm(formula = list ~ neighbourhood)

Residuals:
     Min       1Q   Median       3Q      Max 
-137.878  -31.504   -2.878   47.822  103.683

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -28.75      33.17  -0.867    0.394
neighbourhood  104.81      10.83   9.676 2.86e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 58.02 on 27 degrees of freedom
```

```
Multiple R-squared:  0.7762,    Adjusted R-squared:
0.7679
F-statistic: 93.63 on 1 and 27 DF,  p-value: 2.863e-10
```

$$list = -28.75 + 104.81\ neighbourhood + \varepsilon$$

If we compare the models with one independent variable considered in a), b), c), d) we observe that here we have the biggest Adjusted R^2.
There for the `neighbourhood` is the most important variable for the price `list` within this set of independent random variables.
Let us now answer the question: What is the change of the price for one more point in neighbourhood?

One more point in neighbourhood increases the price with $104.81 \times 1000\$ = 104\,810\$$. Let us now compute the corresponding $95\,\%$ confidence interval

```
> e <- resid(modelPriceNeighbourhood)
> n <- length(e)
> beta1hat <- modelPriceNeighbourhood$coefficients[2];
beta1hat
neighbourhood
     104.8129
> SSE <- sum(e^2)
> MSE <- SSE / (n-2)
> Seps <- sqrt(MSE)
> SEbeta1 <- Seps / sqrt(sum((neighbourhood -
mean(neighbourhood))^2)); SEbeta1
[1] 10.8319
> alpha <- 0.05
> t <- qt(1 - alpha/2, n - 2, lower.tail = TRUE)
> myCI(beta1hat, SEbeta1,t)
[1]  82.58764 127.03808
```

e. Let us now model the dependence of the prices `list` of homes from this population from the points for `neighbourhood`s and the number of `rooms`.

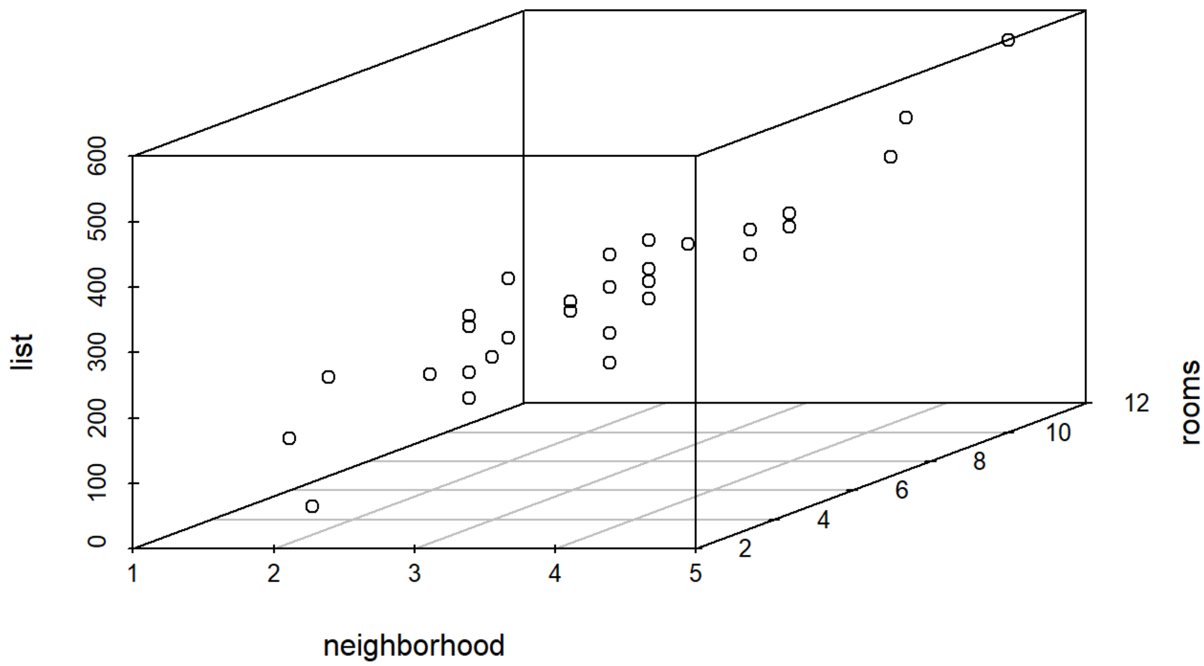
```
> modelPriceNeighbourhoodRooms <- lm(list ~ neighbourhood
+ rooms)
> summary(modelPriceNeighbourhoodRooms)
```

```
Call:
lm(formula = list ~ neighbourhood + rooms)

Residuals:
    Min      1Q  Median      3Q     Max 
-105.78  -29.34    5.01   35.78   65.31 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -167.477     43.232  -3.874 0.000649 ***
neighbourhood   89.115      9.465   9.416 7.32e-10 ***
rooms          25.559      6.300   4.057 0.000403 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 46.27 on 26 degrees of freedom
Multiple R-squared:  0.8629,	Adjusted R-squared:  
0.8524 
F-statistic: 81.85 on 2 and 26 DF,  p-value: 6.02e-12
```

$$list = -167.477 + 89.115\ neighbourhood + 25.559\ rooms + \varepsilon$$

The coefficients for $neighbourhood \neq 104.81$ and the coefficient for $rooms \neq 49.81$ as far as we have **multicollinearity**.

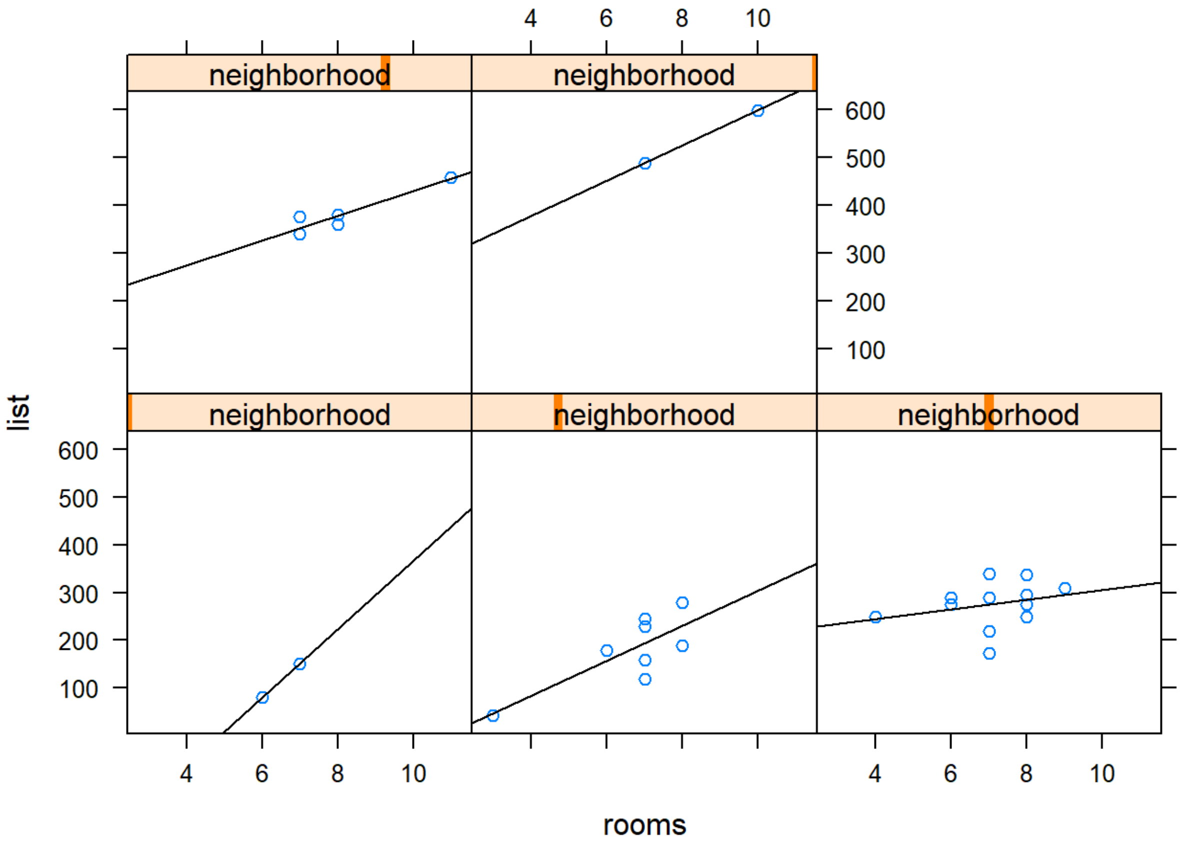
```
> scatterplot3d(neighbourhood,rooms,list)
```




```
> open3d()
wgl
  2
> plot3d(neighbourhood, rooms, list, col = "red", size =
3)
```

We can make regression models on different subsets. For example if we fix the number of neighbours we obtain.

```
> panel.lm <- function(x, y) {
+   panel.xyplot(x, y)
+   panel.abline(lm(y ~ x))
+ }
> xyplot(list ~ rooms | neighborhood, panel = panel.lm)
```



According to the data we observe that when we have the smallest number of points for neighbours the price is the most sensitive of the number of rooms.

f. Let us now model the dependence of the prices `list` of homes from this population from the number of `bedrooms` and points in `neighbourhood`.

```
> modelPriceNeighbourhoodBedrooms <- lm(list ~
neighbourhood + bedrooms)
> summary(modelPriceNeighbourhoodBedrooms)

Call:
lm(formula = list ~ neighbourhood + bedrooms)

Residuals:
     Min       1Q   Median       3Q      Max
-104.443  -34.765   -0.783   21.009   98.122

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

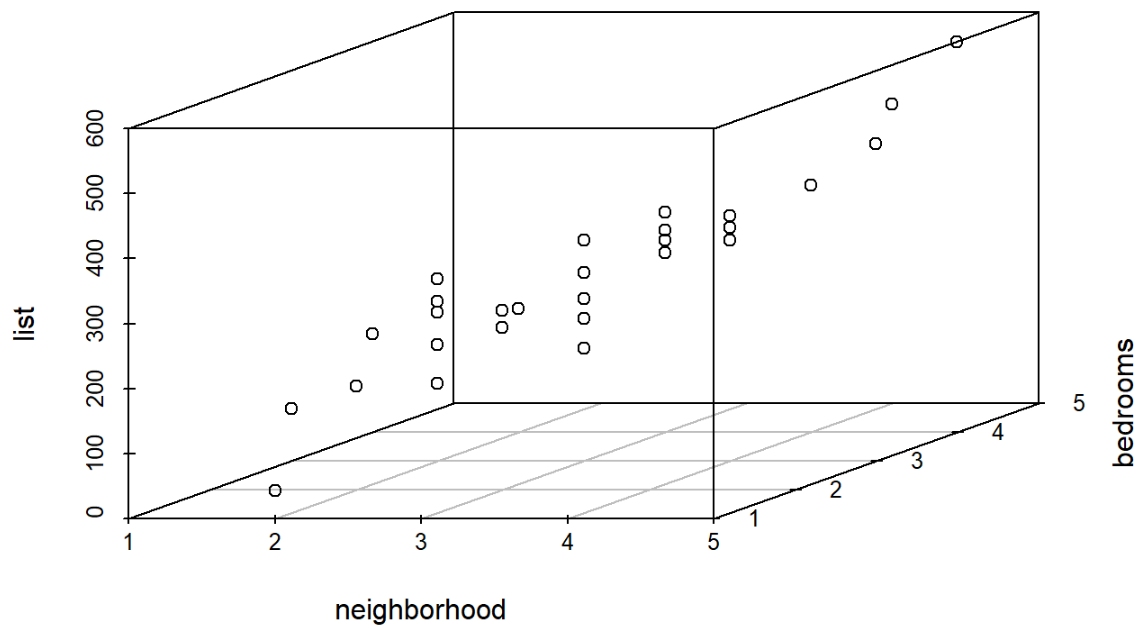
```
(Intercept)  -140.914     40.794  -3.454   0.0019 **
neighbourhood   96.565      9.203  10.493 7.71e-11 ***
bedrooms        42.887     11.574   3.705   0.0010 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 47.83 on 26 degrees of freedom
Multiple R-squared:  0.8535,    Adjusted R-squared:
0.8423
F-statistic: 75.75 on 2 and 26 DF,  p-value: 1.428e-11
```

$$list = -140.914 + 96.565\ neighbourhood + 42.887\ bedrooms + \varepsilon$$

The coefficients for $neighbourhood \neq 104.81$ and the coefficients for $bedrooms \neq 72.26$ as far as we have again **multicollinearity**.

If we compare the adjusted R^2 in this model and in the previous model considered in e) we observe that in e) adjusted R^2 is bigger. Therefore, the model in e) is better. So `rooms` is more important than `bedrooms`.

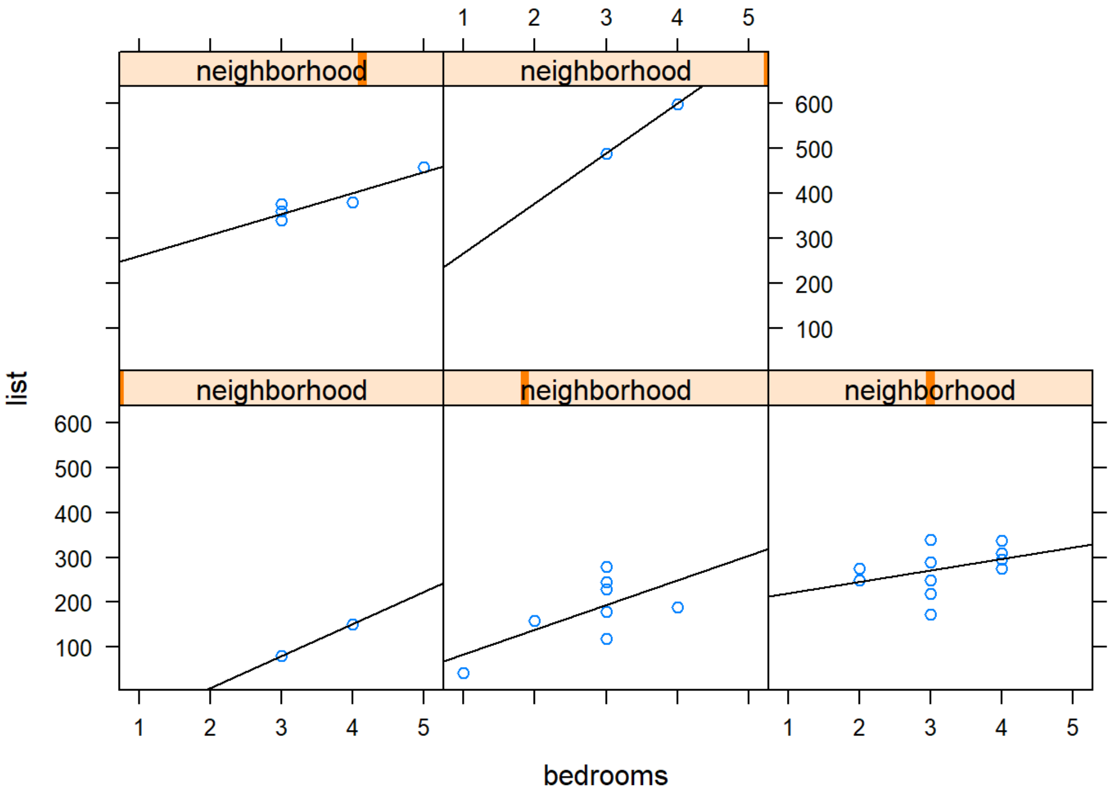
```
> scatterplot3d(neighbourhood, bedrooms, list)
```



```
> open3d()
wgl
  3
> plot3d(neighbourhood, bedrooms, list, col = "red", size
= 3)
```

We can make regression models on different subsets. For example if we fix the number of neighbours we obtain.

```
> xyplot(list ~ bedrooms | neighbourhood, panel =
panel.lm)
```

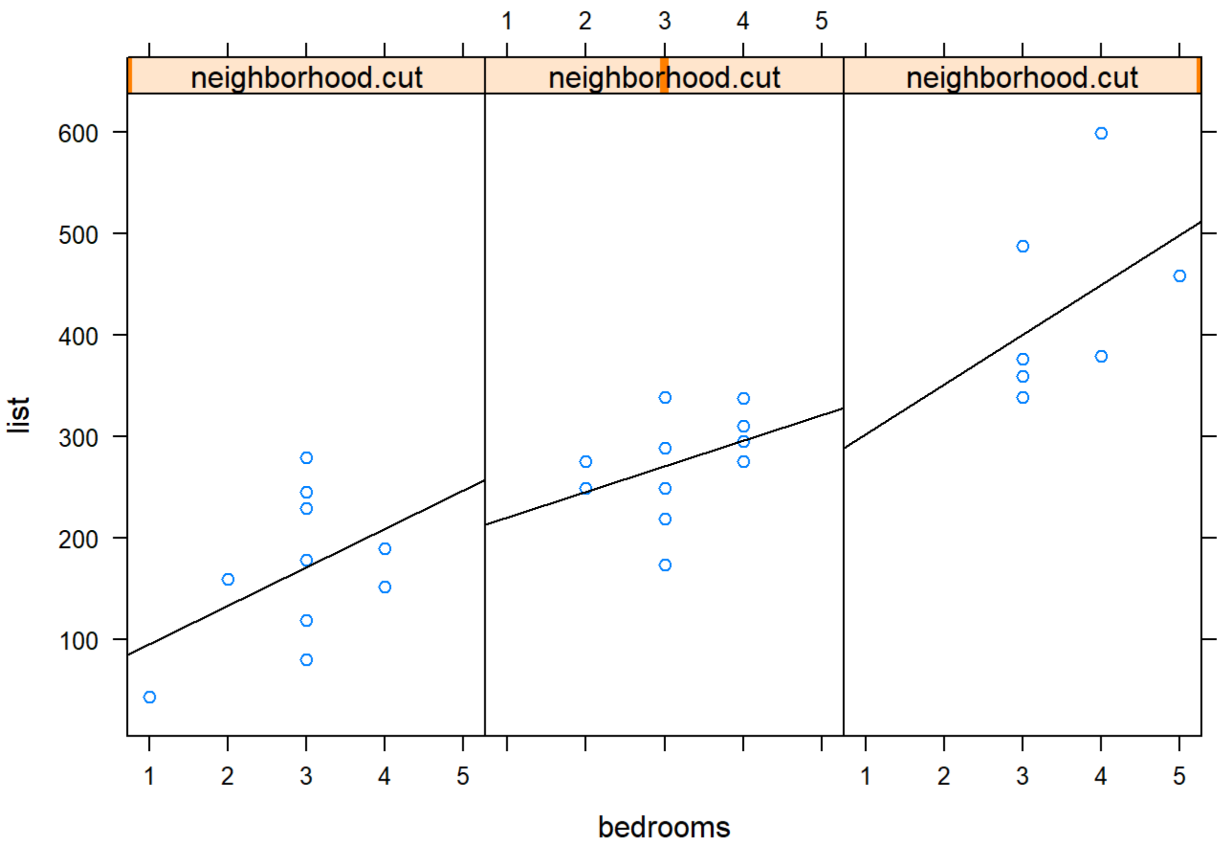


We keep the neighbourhood as a numerical variable to do the regression. The multiple linear regression model assumes that the regression line should have the same slope for all the levels.

Let us divide the population in three subsets with respect to the points for neighbourhoods and then to make regression models.

```
> neighbourhood.cut <- as.numeric(cut(neighbourhood, c(0,
2, 3, 5), labels = c(1, 2, 3)))
> table(neighbourhood.cut)
neighbourhood.cut
 1  2  3
10 12  7
```

```
> xyplot(list ~ bedrooms | neighbourhood.cut, panel =
panel.lm, layout = c(3, 1))
```




```
> model <- lm(list ~ bedrooms + neighbourhood.cut)
> summary(model)

Call:
lm(formula = list ~ bedrooms + neighbourhood.cut)

Residuals:
    Min       1Q   Median       3Q      Max 
-107.59   -44.83   -11.57    31.15   164.90 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -63.16      51.53  -1.226   0.2313    
bedrooms             36.74      15.86   2.317   0.0287 *  
neighbourhood.cut   116.76      16.53   7.062 1.69e-07
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
Residual standard error: 64.06 on 26 degrees of freedom
Multiple R-squared:  0.7372,	Adjusted R-squared:
0.717
F-statistic: 36.47 on 2 and 26 DF,  p-value: 2.846e-08
```

This mean that, if there are 0 `bedrooms` then the house is worth

```
> model$coefficients[1] + model$coefficients[3]*(1:3)
[1]  53.59894 170.36117 287.12340
```

if it has bad, neutral or good neighbours.

g. Let us now model the dependence of the prices `list` of homes from this population from the number of `full` bathrooms, `bedrooms` and points for `neighbourhood`.

```
> complex.model <- lm(list ~ full + bedrooms +
neighbourhood)

> summary(complex.model)

Call:
lm(formula = list ~ full + bedrooms + neighbourhood)

Residuals:
    Min      1Q  Median      3Q     Max 
-93.763 -27.845  -8.004  23.452 102.635 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -140.31      40.93  -3.428  0.00211 ** 
full             14.44      15.76   0.917  0.36815    
bedrooms         40.48      11.90   3.401  0.00226 ** 
neighbourhood     90.40      11.42   7.913 2.86e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 47.98 on 25 degrees of freedom
Multiple R-squared:  0.8583,	Adjusted R-squared:
0.8413
```

```
F-statistic: 50.47 on 3 and 25 DF,  p-value: 9.452e-11
```

$$list = -140.31 + 14.44\,full + 40.48\,bedrooms + 90.50\,neighbourhood + \varepsilon$$

This means that we need to pay $14.44 \times 1000\$ = 14\,440\$$ per full bathroom. Could it possibly be $15\,000\$$?

$$H_0 : \beta_1 = 15$$
$$H_A : \beta_1 > 15$$

```
> SE <- 15.76
> t <- (14.44 - 15) / SE; t
[1] -0.03553299
> pvalue <- pt(t, df = 25, lower.tail = FALSE); pvalue
[1] 0.5140315
```

The $p-value = 0.5140315 > 0.05 = \alpha$, so we have no evidence to reject H_0.

h. Model the dependence of the prices `list` of homes from this population from the number of `rooms`, `bedrooms` and points for `neighbourhood`.

```
> complex.model <- lm(list ~ rooms + bedrooms +
neighbourhood)
> summary(complex.model)

Call:
lm(formula = list ~ rooms + bedrooms + neighbourhood)

Residuals:
     Min       1Q   Median       3Q      Max 
-104.761  -29.449    1.635   31.158   73.909 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -168.136     43.598  -3.856 0.000716 ***
rooms          18.019     11.800   1.527 0.139299    
bedrooms       15.899     20.971   0.758 0.455452    
neighborhood   90.688      9.766   9.286  1.4e-09 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 46.65 on 25 degrees of freedom
Multiple R-squared:  0.866, Adjusted R-squared:  0.8499
F-statistic: 53.87 on 3 and 25 DF,  p-value: 4.706e-11
```

$$list = -168.146 + 18.019\,room + 15.899\,bedrooms + 90.688\,neighbourhood + \varepsilon$$

We can immediately answer the question:Is it acceptable the intercept to be 0?

$$H_0 : \beta_0 = 0$$
$$H_A : \beta_0 \neq 0$$

The $p-value = 0.000716 < 0.05 = \alpha$, so we reject H_0.

or

```
> SEb0 <- 43.598
> temp <- abs(-168.136  - 0) / SEb0; temp
[1] 3.856507
> pvalue<- 2 * pt(temp, df = 25, lower.tail = FALSE);
pvalue
[1] 0.0007155607
```

The $p-value = 0.000716 < 0.05 = \alpha$, so we reject H_0. The intercept β_0 is statistically significant.

Analogously we can check that the coefficients for `rooms` and `bedrooms` are not statistically significant or we can see this in `summary` output as far we have no `*` in the end of their rows. And although the adjusted R^2 is relatively high. If we compare this model with models in e), f) the model in e) is the best one.

i. Determine the expected price of a home from this population if it has 3 `rooms`, 2 `bedrooms` and 2 points for `neighbourhood`.

```
> -168.136 + 18.019*3 + 15.899*2 + 90.688*2
[1] 99.095
```

The estimated `list` by the model is $99\$$.

j. Determine the expected price `list` of these homes having in mind the numbers of their `rooms` and `full` bathrooms and the points for `neighbourhood`.

```
> complex.modelfull <- lm(list ~ full + rooms +
neighbourhood)
> summary(complex.modelfull)

Call:
lm(formula = list ~ full + rooms + neighbourhood)

Residuals:
    Min      1Q  Median      3Q     Max
-94.636 -26.574  -4.456  30.781  71.364

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -166.507     43.270  -3.848 0.000731 ***
full           14.882     15.133   0.983 0.334832
rooms          24.299      6.432   3.778 0.000875 ***
neighbourhood   83.056     11.298   7.351 1.06e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 46.29 on 25 degrees of freedom
Multiple R-squared:  0.868, Adjusted R-squared:  0.8522
F-statistic: 54.82 on 3 and 25 DF,  p-value: 3.894e-11
```

$$list = -166.507 + 14.882\,full + 25.299\,rooms + 83.056\,neighbourhood + \varepsilon$$

In this model the number of full bathrooms is not statistically significant and adjusted R^2 is less than in e) therefore the model in e) only with 2 independent variables is better.

```
> yhat <- complex.modelfull$fitted.values; yhat
        1         2         3         4         5
6         7         8
 77.22524 101.52431 331.11633 306.81726 365.57428
291.93557 214.34378 184.58040
```

```
        9         10         11         12         13
14         15         16
267.63650  87.38412 208.87948 536.40927 199.46209
258.21911 194.73928 175.16302
       17         18         19         20         21
22         23         24
184.58040 282.51818 463.51206 380.45597 291.93557
306.81726 258.21911 477.65225
       25         26         27         28         29
267.63650 389.87335 389.87335 208.87948 267.63650
```

or

```
> yhat <- complex.modelfull$coefficients[1] +
complex.modelfull$coefficients[2] * full +
complex.modelfull$coefficients[3] * rooms +
complex.modelfull$coefficients[4] * neighbourhood; yhat
 [1]  77.22524 101.52431 331.11633 306.81726 365.57428
291.93557 214.34378
 [8] 184.58040 267.63650  87.38412 208.87948 536.40927
199.46209 258.21911
[15] 194.73928 175.16302 184.58040 282.51818 463.51206
380.45597 291.93557
[22] 306.81726 258.21911 477.65225 267.63650 389.87335
389.87335 208.87948
[29] 267.63650
```

k. Find and plot the errors(residuals) ε in the model in j).

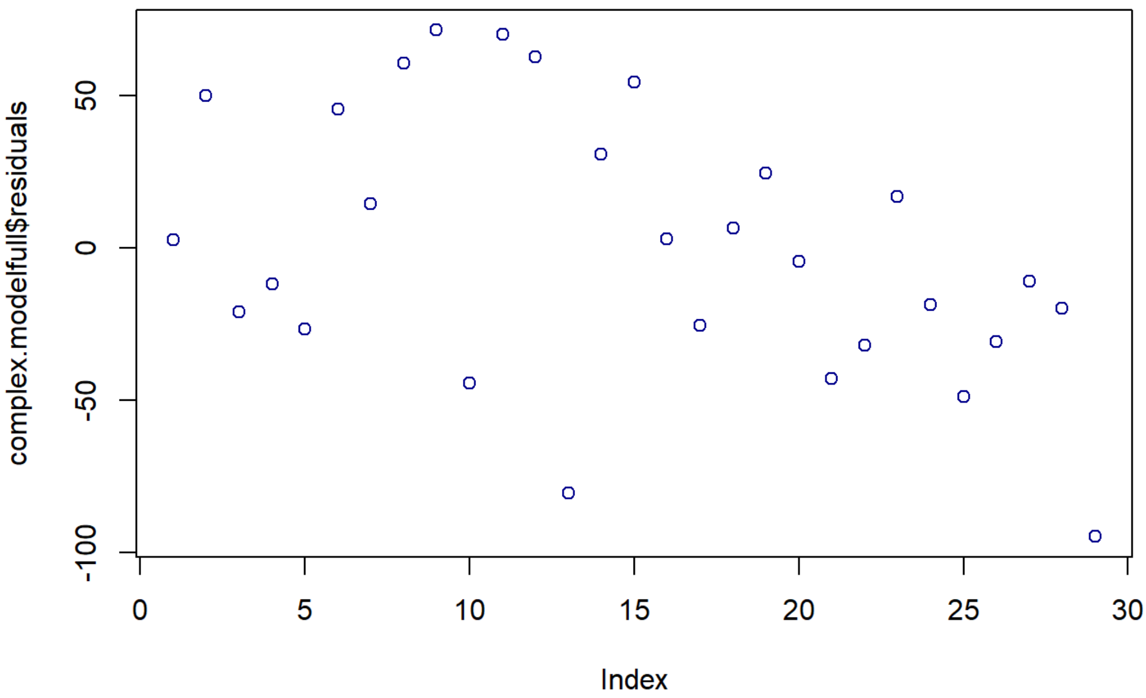
```
> e <- resid(complex.modelfull); e
         1          2          3          4          5
6          7
  2.774762  49.875690 -21.116328 -11.817256 -26.574278
45.564431  14.356221
         8          9         10         11         12
13         14
 60.419597  71.363503 -44.384116  70.120525  62.590726
-80.462091  30.780887
        15         16         17         18         19
20         21
 54.260719   2.836981 -25.580403   6.481815  24.487941
-4.455966 -42.935569
```

```
        22          23          24          25          26
27          28
-31.817256  16.780887 -18.652253 -48.636497 -30.873350
-10.873350 -19.879475
        29
-94.636497
```

or by using the formula

```
> e <- list - yhat; e
 [1]   2.774762  49.875690 -21.116328 -11.817256
-26.574278  45.564431
 [7]  14.356221  60.419597  71.363503 -44.384116
70.120525  62.590726
[13] -80.462091  30.780887  54.260719   2.836981
-25.580403   6.481815
[19]  24.487941  -4.455966 -42.935569 -31.817256
16.780887 -18.652253
[25] -48.636497 -30.873350 -10.873350 -19.879475
-94.636497
```

```
> plot(complex.modelfull$residuals, col = "darkblue")
```



l. It is time to determine the mean square error (MSE) of the multiple model in j).

$$MSE = RSE^2 = S_\varepsilon^2 = \frac{1}{n-4}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 = \frac{1}{n-4}\sum_{i=1}^{n}\varepsilon_i^2$$

It is an unbiased estimator of σ_ε^2. The denominator $n-4$ comes from the fact that there are four coefficients estimated from the data: β_0, β_1, β_2 and β_3.

Let us remind that

$$SSE = \sum_{i=1}^{n}\varepsilon_i^2, MSE = \frac{SSE}{n-r} = \frac{SSE}{n-4}$$

```
> SSE <- sum(e^2); SSE
[1] 53580.19
> MSE <- SSE / (n - 4); MSE
[1] 2143.208
> s <- sqrt(MSE); s
[1] 46.29479
```

The **Residual Standard error** is

$$S_\varepsilon = \sqrt{MSE} = \sqrt{\frac{SSE}{n-3}} = 46.29479\ EUR$$

or we can extract it via the function `summary`

```
> summary(complex.modelfull)

Call:
lm(formula = list ~ full + rooms + neighbourhood)

Residuals:
    Min      1Q  Median      3Q     Max 
-94.636 -26.574  -4.456  30.781  71.364 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  -166.507     43.270  -3.848 0.000731 ***
full           14.882     15.133   0.983 0.334832
rooms          24.299      6.432   3.778 0.000875 ***
neighborhood   83.056     11.298   7.351 1.06e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 46.29 on 25 degrees of freedom
Multiple R-squared:  0.868, Adjusted R-squared:  0.8522
F-statistic: 54.82 on 3 and 25 DF,  p-value: 3.894e-11
```

m. Compute the coefficient of determination of the model in j).
Via the function `summary` we can estimate also the coefficient of determination. Note that it is **Adjusted R-squared: 0.8522**

$$\mathrm{cor}^2(X, Y) = 1 - \frac{\mathbb{E}\varepsilon^2}{\mathbb{D}Y}, Adjusted\ R{-}squared = 0.8522,$$

The coefficient is close to 1, therefore, we can say that the independent variables are important for the value of the dependent variable Y `list` for homes in this population. We can determine it also via the formula

```
> Rsquare <- 1 - MSE/var(list); Rsquare
[1] 0.8522156
```

The other result **Multiple R-squared: 0.868** does not take into account that the denominators of the estimators S_ε^2 and $S_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y}_n)^2$ are different and computes

$$Multiple\ R{-}squared = 1 - \frac{SSE}{\sum_{i=1}^{n}(Y_i - \overline{Y}_n)^2} = 0.868$$

```
> Rsq <- 1 - SSE / sum((list - mean(list))^2); Rsq
[1] 0.8680496
```

n. In order to check if in the model in j) $\mathbb{E}\varepsilon = 0$ we use `t-test`.

$H_0 : \mathbb{E}\varepsilon = 0$
$H_A : \mathbb{E}\varepsilon \neq 0$

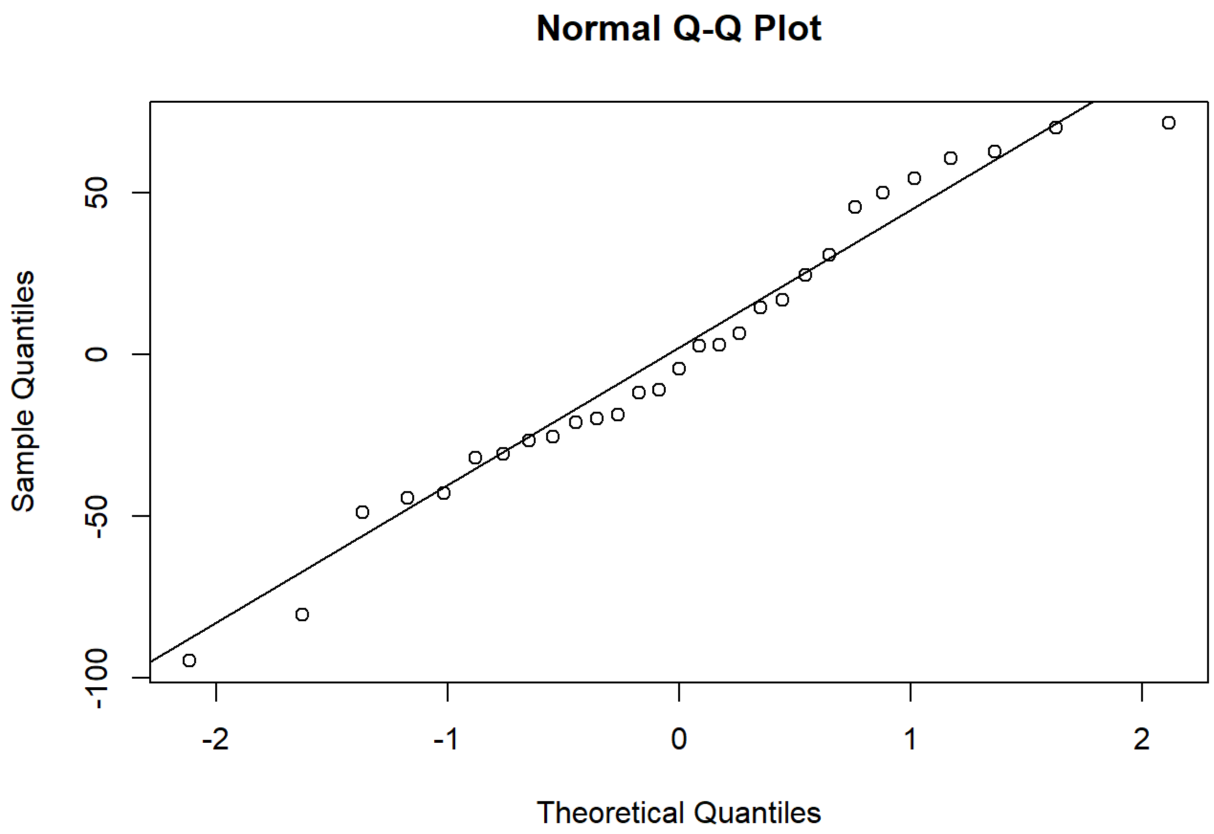
```
> mean(e)
[1] -2.508956e-13
> n <- length(e); n
[1] 29
> s <- sd(e); s
[1] 43.74447
> temp <- abs(mean(e) - 0) / (s/sqrt(n)); temp
[1] 3.088651e-14
> pvalue <- 2 * pt(temp, n - 1, lower.tail = FALSE);
pvalue
[1] 1
```

The $p-value = 1 > 0.05 = \alpha$, therefore, we have no evidence to reject H_0, so the requirement of the model $\mathbb{E}\varepsilon = 0$ is satisfied.

o. The next step is to test the assumptions of the model that the residuals ε are i.i.d. normally distributed $\varepsilon_i \in N(0, \sigma_\varepsilon^2)$

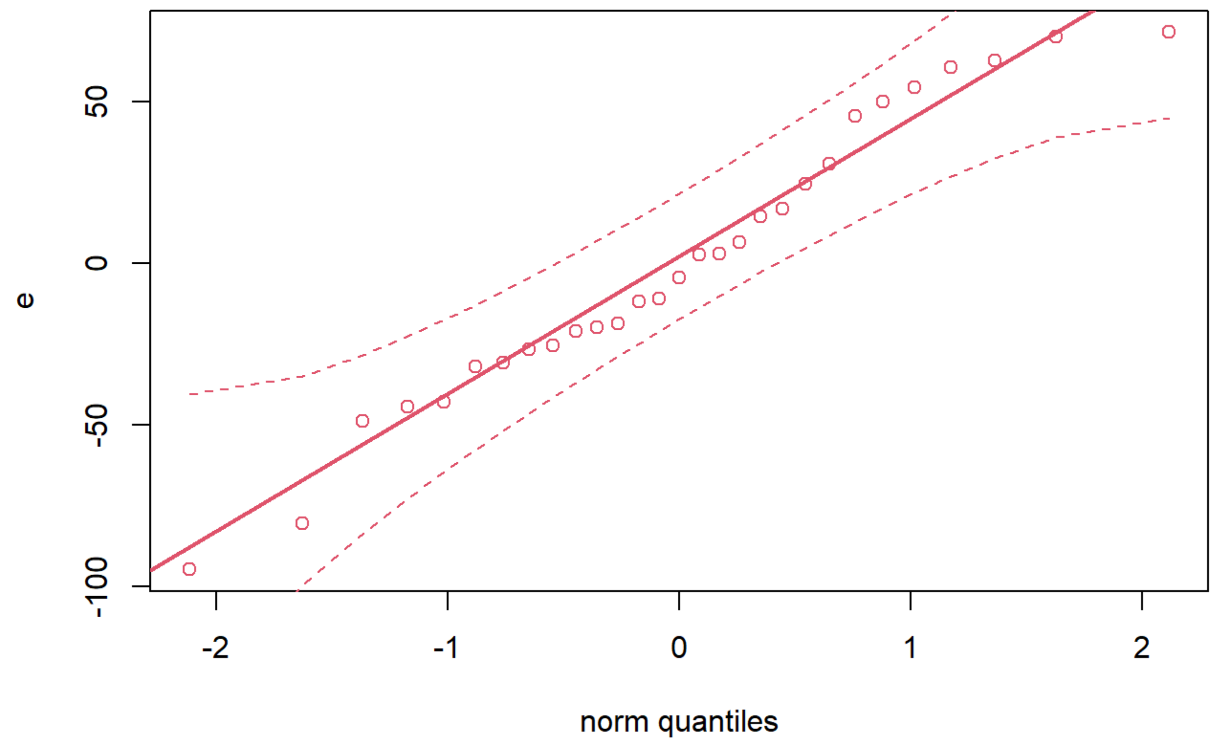
First we make the normal `qq-plot`

```
> qqnorm(e)
> qqline(e)
```



Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-100
-50
0
50
-2
-1
0
1
2


```
> qqplot.das(e)
```



e
norm quantiles
-100
-50
0
50
-2
-1
0
1
2

We can perform also Shapiro test

H_0 : ε is normally distributed
H_A : ε is not normally distributed

Now we use the function `shapiro.test` in R

```
> shapiro.test(e)

	Shapiro-Wilk normality test

data:  e
W = 0.96737, p-value = 0.4907
```

The $p-value = 0.4907 > 0.05 = \alpha$, therefore, we have no evidence to reject H_0, so the requirement ε to be normally distributed is satisfied.

p. Let us now determine 95% confidence intervals for the expected price $\hat{Y}$ of these homes having in mind the numbers of their `rooms`, `full` bathrooms and the points for `neighborhood`.

$$S^2_{\hat{Y}} = S^2_\varepsilon \overrightarrow{X} (\mathbb{X}^T\mathbb{X})^{-1} \overrightarrow{X}^T$$

Therefore,

$$\hat{Y} = (Y \,|\, \overrightarrow{X}) \in N(\hat{\overrightarrow{\beta}} \overrightarrow{X}; \sigma^2_\varepsilon \overrightarrow{X} (\mathbb{X}^T\mathbb{X})^{-1} \overrightarrow{X}^T)$$

And the $(1-\alpha)100\,\%$ confidence interval for $\hat{Y}$ is

$$[\, \overrightarrow{\beta}^T \overrightarrow{X} - t_{1-\frac{\alpha}{2};t(n-r-1)} S_{\hat{Y}};\ \overrightarrow{\beta}^T \overrightarrow{X} + t_{1-\frac{\alpha}{2};t(n-r-1)} S_{\hat{Y}}],\, i = 1, 2, \ldots, r$$

Now we use the function `predict` in R

```
> predict(complex.modelfull, interval = "confidence",
level = 0.95)
        fit       lwr      upr
1  77.22524  38.81385 115.6366
2 101.52431  61.37379 141.6748
3 331.11633 302.34844 359.8842
```

```
4   306.81726 285.99545 327.6391
5   365.57428 336.33731 394.8112
6   291.93557 259.05370 324.8174
7   214.34378 157.25044 271.4371
8   184.58040 158.09572 211.0651
9   267.63650 238.07703 297.1960
10   87.38412  34.26763 140.5006
11 208.87948 176.49141 241.2675
12 536.40927 490.28071 582.5378
13 199.46209 167.69450 231.2297
14 258.21911 231.79006 284.6482
15 194.73928 146.18554 243.2930
16 175.16302 140.85462 209.4714
17 184.58040 158.09572 211.0651
18 282.51818 262.75672 302.2796
19 463.51206 415.76753 511.2566
20 380.45597 341.42657 419.4854
21 291.93557 259.05370 324.8174
22 306.81726 285.99545 327.6391
23 258.21911 231.79006 284.6482
24 477.65225 426.82482 528.4797
25 267.63650 238.07703 297.1960
26 389.87335 362.34628 417.4004
27 389.87335 362.34628 417.4004
28 208.87948 176.49141 241.2675
29 267.63650 238.07703 297.1960
```

Polynomial models

Multiple Linear Regression Models are called in this way because the mean of Y is linear with respect to the parameters $\beta_0, \beta_1, \ldots, \beta_r$. Therefore, polynomial models (when $X_k = X^k, k = 1, 2, \ldots, r$)

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_r X^r + \varepsilon$$

are a particular case of Multiple Linear Regression Models.

Example 3:

In 1609 Galileo proved that the trajectory of a body falling with a horizontal component is a parabola. In the course of gaining insight into this fact, he set up an experiment which measured two variables, a `height` and a `distance`, yielding the following data

height (punti)	100	200	300	450	600	800	1000
dist (punti)	253	337	395	451	495	534	574

In plotting the data, Galileo apparently saw the parabola and with this insight proved it mathematically. Let's see if linear regression can help us find the coefficients.

```
> height <- c(100, 200, 300, 450, 600, 800, 1000)
> dist <- c(253, 337, 395, 451, 495, 534, 574)
```

The `I` function allows us to use the usual notation for power, because the `^` is used differently in the model notation.

```
> lm.2 <- lm(dist ~ height + I(height^2));
> summary(lm.2)

Call:
lm(formula = dist ~ height + I(height^2))

Residuals:
      1       2       3       4       5       6       7
-14.420   9.192  13.624   2.060  -6.158 -12.912   8.614

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.002e+02  1.695e+01  11.811 0.000294 ***
height       7.062e-01  7.568e-02   9.332 0.000734 ***
I(height^2) -3.410e-04  6.754e-05  -5.049 0.007237 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 13.79 on 4 degrees of freedom
Multiple R-squared:  0.9902,    Adjusted R-squared:
0.9852
```

```
F-statistic: 201.1 on 2 and 4 DF,  p-value: 9.696e-05
> points <- 100:1000
> quad.fit <- lm.2$coefficients[1] + lm.2$coefficients[2]
* points + lm.2$coefficients[3] * points^2
```

We observe that all coefficients are statistically significant. The model is

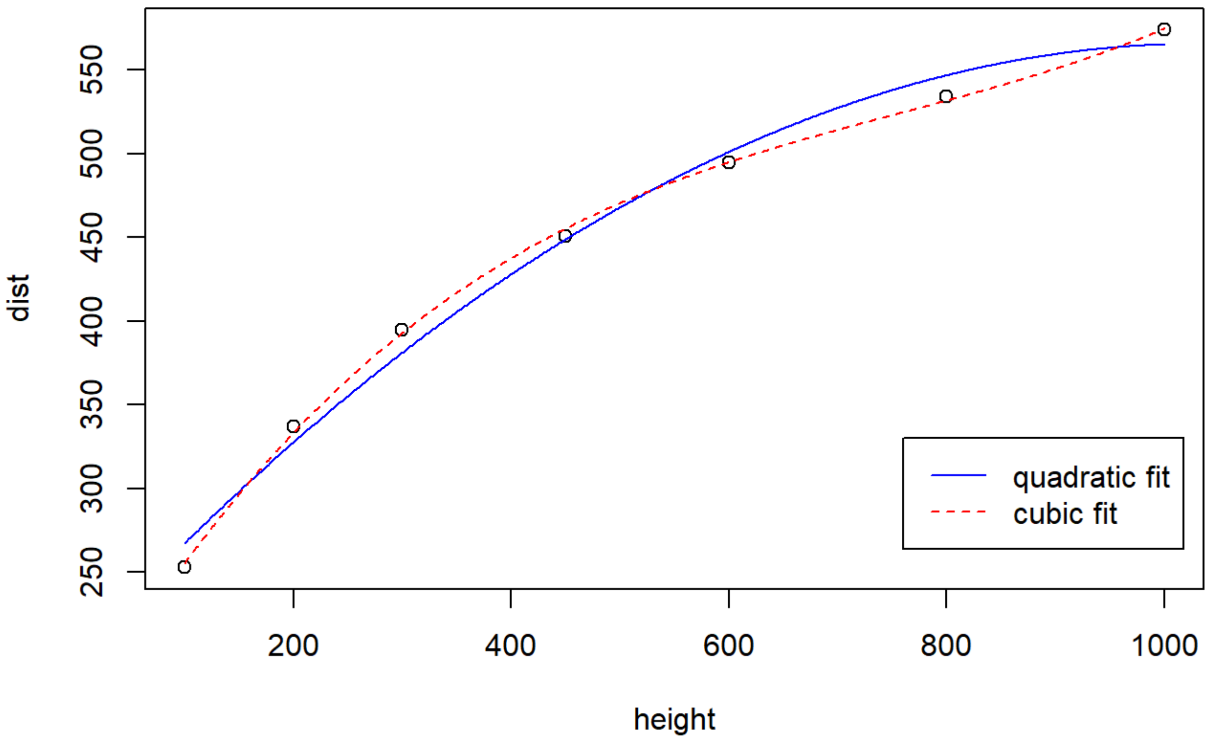
$$dist = 200.2 + 0.7062\, height - 0.000341\, height^2 + \varepsilon$$

```
> lm.3 <- lm(dist ~ height + I(height^2) + I(height^3));
> summary(lm.3)

Call:
lm(formula = dist ~ height + I(height^2) + I(height^3))

Residuals:
       1        2        3        4        5        6
7
-2.35639  3.52782  1.83769 -4.43416  0.01945  2.21560
-0.81001

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.555e+02  8.182e+00  19.003 0.000318 ***
height       1.119e+00  6.454e-02  17.332 0.000419 ***
I(height^2) -1.254e-03  1.360e-04  -9.220 0.002699 **
I(height^3)  5.550e-07  8.184e-08   6.782 0.006552 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 3.941 on 3 degrees of freedom
Multiple R-squared:  0.9994,    Adjusted R-squared:
0.9988
F-statistic:  1658 on 3 and 3 DF,  p-value: 2.512e-05
> cube.fit <- lm.3$coefficients[1] + lm.3$coefficients[2]
* points + lm.3$coefficients[3] * points^2 +
lm.3$coefficients[4] * points^3
```

Again we observe that all coefficients are statistically significant. The model is

$$dist = 155.5 + 1.119\, height - 0.001254\, height^2 + 0.000000555\, height^3 + \varepsilon$$

```
> plot(height, dist)
> lines(points, quad.fit, lty = 1, col = "blue")
> lines(points, cube.fit, lty = 2, col = "red")
> legend(x = 760, y = 330, c("quadratic fit", "cubic
fit"), lty = 1:2, col = c("blue", "red"))
```



Both curves seem to fit the data well. Which one to choose? A hypothesis test of β_3 will help us to decide between the two choices. Therefore we test,

$H_0 : \beta_0 = 0$
$H_A : \beta_0 \neq 0$

In the function `summary(lm.3)` the $p\!-\!value = 0.006552$ is flagged automatically by R. It is less than $\alpha = 0.05$, therefore, we reject H_0 and the alternative $\beta_3 \neq 0$ is accepted. According to this data we are tempted to attribute this cubic presence to resistance which is ignored in the mathematical solution which finds the quadratic relationship.

Example 4:

If there is no intercept term (β_0) in the model, you can explicitly remove it by adding `0` or `-1` to the formula.

```
> n <- 50
> x1 <- rnorm(n, 172, 7)
> x2 <- rnorm(n, 168, 7)
> eps <- rnorm(n, 0, 3)
> y <- x1 + x2 + eps
> lm.fit <- lm(y ~ x1 + x2 - 1)
> summary(lm.fit)
```

```
Call:
lm(formula = y ~ x1 + x2 - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.3482 -2.3311 -0.2261  2.4517  7.6135 

Coefficients:
   Estimate Std. Error t value Pr(>|t|)    
x1  0.93864    0.06204   15.13   <2e-16 ***
x2  1.06527    0.06307   16.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 3.431 on 48 degrees of freedom
Multiple R-squared:  0.9999,    Adjusted R-squared: 
0.9999 
F-statistic: 2.471e+05 on 2 and 48 DF,  p-value: < 
2.2e-16
```

You can compare the above model without intercept with the following model with intercept.

```
> summary(lm(y ~ x1 + x2))
```

```
Call:
lm(formula = y ~ x1 + x2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-7.4546 -1.9526 -0.2168  2.4126  7.1390 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 15.46230   22.09459   0.700    0.487    
x1           0.88406    0.09987   8.852 1.41e-11 ***
x2           1.02925    0.08167  12.603  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 
' ' 1

Residual standard error: 3.449 on 47 degrees of freedom
Multiple R-squared:  0.8336,	Adjusted R-squared:  
0.8265 
F-statistic: 117.7 on 2 and 47 DF,  p-value: < 2.2e-16
```

We observe that the model without intercept is better.

ANalysis Of VAriance (ANOVA)

If the residual of the model is $\varepsilon \in N(0,\sigma_\varepsilon^2)$ we can make **ANalysis Of VAriance (ANOVA) /дисперсионен анализ/** and to check if the influence of a group of independent variables $X^{(1)}, \ldots, X^{(k)}, k < r$ is statistically significant for Y.

Consider the models (longer)

$$Y = \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \ldots + \beta_r X^{(r)} + \varepsilon, \qquad (2)$$

and (shorter)

$$Y = \tilde{\beta}_0 + \tilde{\beta}_1 X^{(1)} + \tilde{\beta}_2 X^{(2)} + \ldots + \tilde{\beta}_k X^{(k)} + \tilde{\varepsilon} \qquad (3)$$

We can test the hypothesis
$H_0 : \beta_{k+1} = \beta_{k+2} = \ldots = \beta_r = 0$
H_A : At least one of these coefficients is statistically significantly different from 0.

If

- SSE_k is the sum of squares of the residuals in the shorter model (3), and
- SSE_r is the sum of squares of the residuals in the longer model (2).

$$\left(\frac{\frac{SSE_k - SSE_r}{r-k}}{\frac{SSE_r}{n-r-1}} \middle| H_0 \right) \in F(r-k;\ n-r-1)$$

Therefore, the critical area for H_0 is

$$W_\alpha = \left\{ \frac{\frac{SSE_k - SSE_r}{r-k}}{\frac{SSE_r}{n-r-1}} \geq x_{1-\frac{\alpha}{2};F(r-k;\ n-r-1)} \right\}$$

If

$$F_{emp} = \frac{\frac{SSE_k - SSE_r}{r-k}}{\frac{SSE_r}{n-r-1}},$$

is the computed value from the data, the $p-value = \mathbb{P}(\eta > F_{emp})$, where $\eta \in F(r-k;\ n-r-1)$.

If we have no multicollinearity and $k = r-1$ these test coincides with some $H_0 : \beta_i = 0$ for some $i \in 1, 2, \ldots, r$

Example 5

Which one of the independent variables C and S in Example 1 is more important for the model.

```
> S <- c(8, 8, 10, 13, 13, 13, 13, 13, 13, 13, 13, 12,
12, 12, 12,
```

```
+ 12, 12, 12, 12, 12, 12, 12, 15, 17, 18, 19, 19, 19, 15, 
17, 17,
+ 17, 17, 17, 17, 17, 16, 16, 16, 16, 16, 16, 16, 16, 16, 
16, 16, 13)
> C <- c(60, 70, 85, 87, 89, 90, 82, 81, 80, 87, 82, 81, 
82, 82, 72,
+ 82, 92, 90, 92, 89, 89, 88, 88, 91, 91, 97, 100, 96, 
92, 93, 94,
+ 95, 96, 97, 97, 97, 96, 96, 95, 93, 96, 94, 95, 92, 91, 
90, 92, 93)
> Earn <- c(500, 570, 550, 770, 690, 900, 620, 610, 800, 
870, 820,
+ 810, 820, 722, 722, 822, 722, 950, 752, 769, 769, 878, 
878, 971,
+ 991, 977, 1100, 796, 712, 713, 714, 725, 716, 717, 797, 
797,
+ 696, 696, 695, 693, 696, 694, 695, 792, 891, 890, 792, 
693)
> df <- data.frame(Earn, S, C)
> n <- length(S); n
[1] 48
```

In order to answer this question we can use the function `anova` in R. It can compare two models and report if they are significantly different. The output from `anova` includes a p-value. Conventionally, a $p-value < 0.05$ indicates that the models are significantly different, whereas a $p-value > 0.05$ provides no such evidence.

```
> EarnCS <- lm(Earn ~ C + S)
> EarnS <- lm(Earn ~ S)
> anova(EarnCS, EarnS)
Analysis of Variance Table

Model 1: Earn ~ C + S
Model 2: Earn ~ S
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     45 525183
2     46 525757 -1   -574.17 0.0492 0.8255
```

The $p-value = 0.8255 > 0.05 = \alpha$, therefore, the models are not significantly different. Therefore, `C` is not so important for `Earn`. In other words: if we add terms and the new model is essentially unchanged, then the extra terms are not worth the additional complications. This p-value shows the significance of the coefficient β_2 before the added independent variable `C` in the model

$$Earn = \beta_0 + \beta_1 S + \beta_2 C + \varepsilon$$

Or

```
> EarnCS <- lm(Earn ~ C + S)
> EarnC <- lm(Earn ~ C)
> anova(EarnCS, EarnC)
Analysis of Variance Table

Model 1: Earn ~ C + S
Model 2: Earn ~ C
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     45 525183
2     46 561483 -1    -36300 3.1103 0.08459 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

The $p-value = 0.08459 > 0.05 = \alpha$, therefore, the models are not significantly different. However, when we compare this p-value with the previous one we can say that now we are not so confident as in the previous case. Therefore, `S` is more important for `Earn`, than `C`.This p-value shows the significance of the coefficient β_1 for the added independent variable `S` in the model

$$Earn = \beta_0 + \beta_1 S + \beta_2 C + \varepsilon$$

In other words: if we add terms and the new model is essentially unchanged, then the extra terms are not worth the additional complications.

The `anova` function has one strong requirement when comparing two models: one model must be contained within the other. That is, all the terms of the smaller model must appear in the larger model. Otherwise, the comparison is impossible.

In order to make the same we can consider also only the larger model in function `anova`.

```
> EarnSC <- lm(Earn ~ S + C)
> anova(EarnSC)
Analysis of Variance Table

Response: Earn
          Df Sum Sq Mean Sq F value   Pr(>F)   
S          1 126058  126058 10.8012 0.001971 **
C          1    574     574  0.0492 0.825470   
Residuals 45 525183   11671                    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

The row `S` corresponds to the degrees of freedom, sum of squares of the errors $SSE_k = SSE_1$, mean square error $\frac{SSE_k}{n-r-1} = \frac{SSE_k}{n-2}$, F_{emp} and the $p-value$ for the model

$$Earn = \tilde{\beta}_0 + \tilde{\beta}_1 S + \tilde{\varepsilon}$$

The last p-value in this row shows the significance of $\tilde{\beta}_1$ in the above model.

In the row `C` the last p-value shows the significance of β_2 in the model

$$Earn = \beta_0 + \beta_1 S + \beta_2 C + \varepsilon$$

The more important independent variable is `S`. When we insert it in the model we can exclude `C`.

or

```
> EarnCS <- lm(Earn ~ C + S)
> anova(EarnCS)
Analysis of Variance Table

Response: Earn
          Df Sum Sq Mean Sq F value   Pr(>F)   
C          1  90332   90332  7.7400 0.007864 **
S          1  36300   36300  3.1103 0.084586 . 
Residuals 45 525183   11671                    
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

The inclusion only of `c` in the model does not lead us to such a high significance of the coefficient before `c` as in the previous case. When we insert after `c` the independent variable `s` in the model the coefficient before `s` is insignificant, however the change is not so huge as in the previous `anova`.

When compare the $p-value$ in the row `s` with the corresponding one in

```
> mymodel <- lm(Earn ~ S + C, data = df)
> summary(mymodel)
```

```
Call:
lm(formula = Earn ~ S + C, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-139.897 -104.855   -7.961   91.739  241.778 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 451.0495   208.2405   2.166   0.0356 *
S            17.4389     9.8882   1.764   0.0846 .
C             0.7583     3.4189   0.222   0.8255  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 108 on 45 degrees of freedom
Multiple R-squared:  0.1943,	Adjusted R-squared:
0.1585 
F-statistic: 5.425 on 2 and 45 DF,  p-value: 0.007748
```

we observe that they are the same.