Lab 2.1: Linear Regression in R

In this Lab we will learn how to:

- Run a simple linear regression model
- Run a multiple linear regression model

We will use also the knowledge we acquire in our previous labs and a new data set (Pnig_34.csv) that correspond with data from the Third Spanish National Forest Inventory (3SNFI) from the Palencia province and *Pinus nigra* Arn. as dominant species. As usual, first we must define our working directory and load the data sets we will use.

```
# establishing the working directory and loading the data sets

setwd('C:/your_desired_working_directoryR')
Stand.Pnigra.Palencia<-read.csv2('Pnig_34.csv',  header=TRUE)
```

Also we should explore the basic features of our dataframe (Stand.Pnigra.Palencia) by using this code:

```
# basic features of our dataframe

names(Stand.Pnigra.Palencia)
head(Stand.Pnigra.Palencia)
tail(Stand.Pnigra.Palencia)
```

To obtain the following result with the names function:

```
[1] "INVENTARIO" "PROVINCIA"  "PARCELA"    "N"        "DG"       "DM"       "DO"
[8] "HM"       "HO"       "AB"       "HART"      "REINEKE"
```

You can observe in your computer the outcomes of the head() and tail() functions. Additionally, you can find useful to apply the code studied in the previous lab (specially the function summary() and the hist() and boxtplot() graphs) to the variables in Stand.Pnigra.Palencia

Exploring data to insight on simple linear regression

In our previous labs we have knew how to draw scatterplots from two variable in our set. Now we can add to this scattlerplot a straight line showing the linear tendency of the point cloud. We will do this in 3 steps:

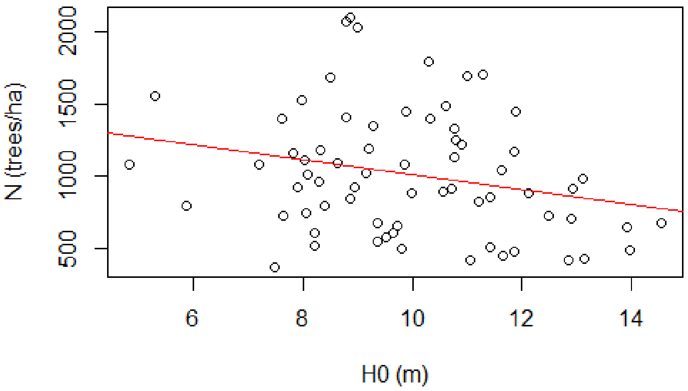
1. Fit a simple linear model (we will call this linearmodel_1)
2. Plot the two variables
3. Add a line representing the linearmodel_1

The script to do that is the following:

```
# Exploring data to insight on simple linear regression

linearmodel_1 <-   lm(Stand.Pnigra.Palencia$N~  Stand.Pnigra.Palencia$HO)
```

```
plot(Stand.Pnigra.Palencia$HO , Stand.Pnigra.Palencia$N ,
    xlab="H0 (m)", ylab="N (trees/ha)")
+ abline(linearmodel_1, col="red")
```

To obtain the following plot:



Fitting a simple linear regression

Already we have fitted a simple linear model by the lm () function but we did not see the result (only the straight line in the previous plot). To obtain the basic results (estimates values, p-values, R-squared,..) we can use the summary () function.

```
# obtaining the basic outcome from a linear regression
summary (linearmodel_1)
```

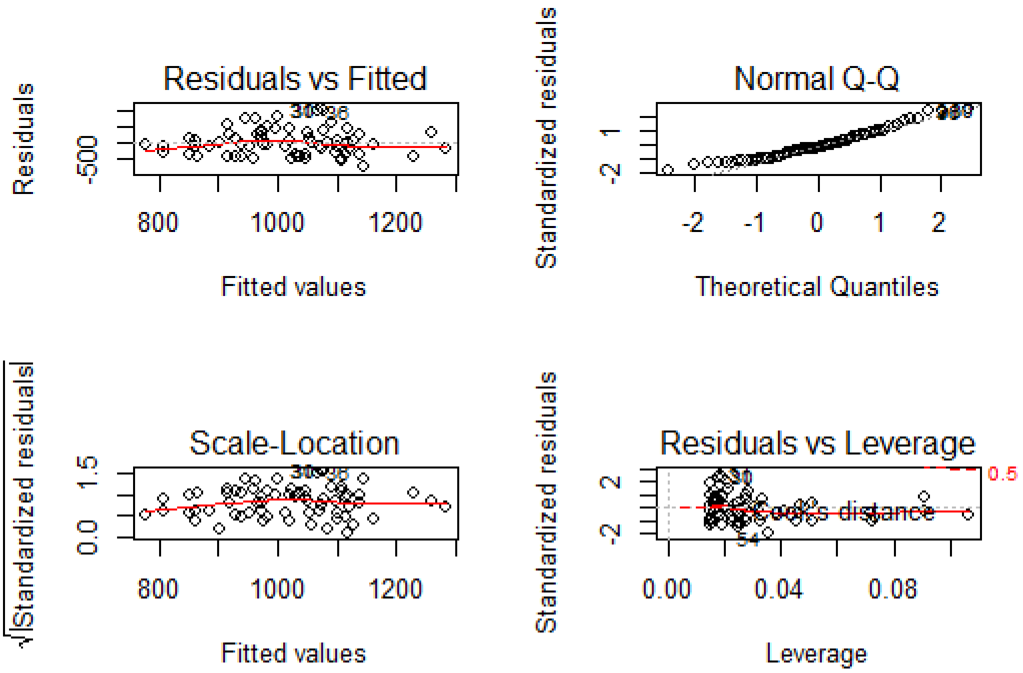
to obtain:

```
Coefficients:
                              Estimate   Std. Error   t value    Pr(>|t|)
(Intercept)                    1533.57     251.76      6.091     6.44e-08 ***
Stand.Pnigra.Palencia$HO        -51.96      24.86     -2.090     0.0405   *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 418.2 on 66 degrees of freedom
Multiple R-squared:  0.06207,        Adjusted R-squared:  0.04786
F-statistic: 4.368 on 1 and 66 DF,  p-value: 0.04048
```

Now we can obtaining the basic graph information from the simple linear model with the plot ( ) function.

```
# plotting the basic graph information of the linearmodel_1
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(linearmodel_1)  # Plot the model information
```

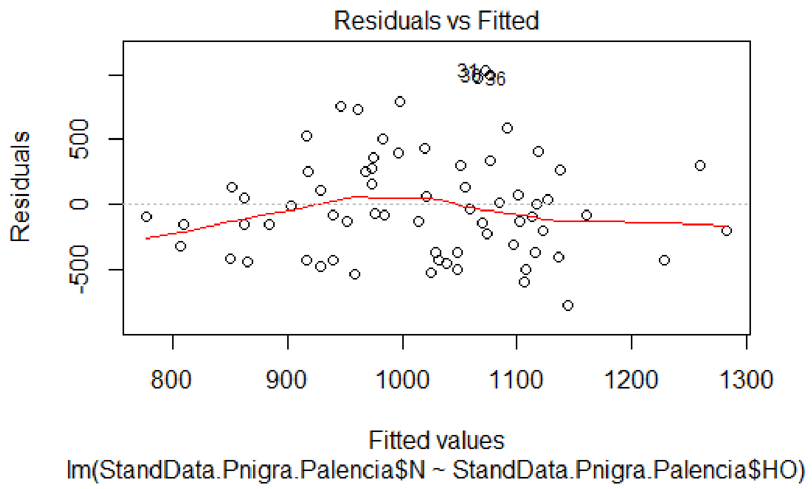
We will obtain the following graphs matrix:

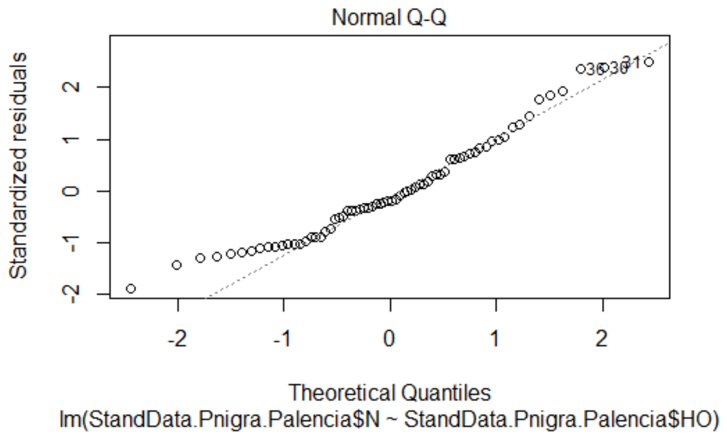


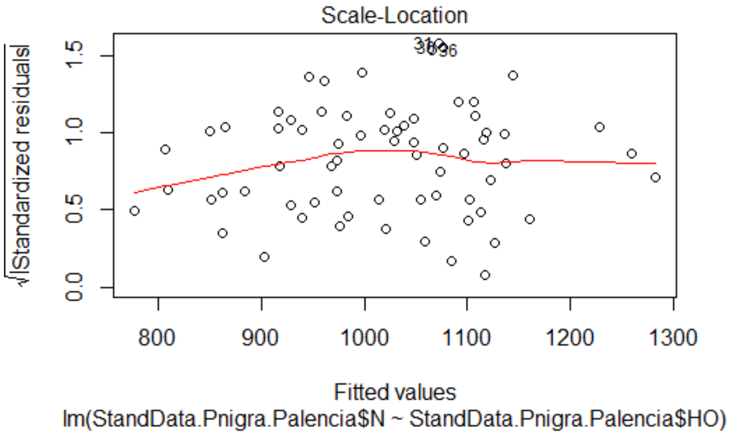
If you want to obtain the graphs individually then the code should be:

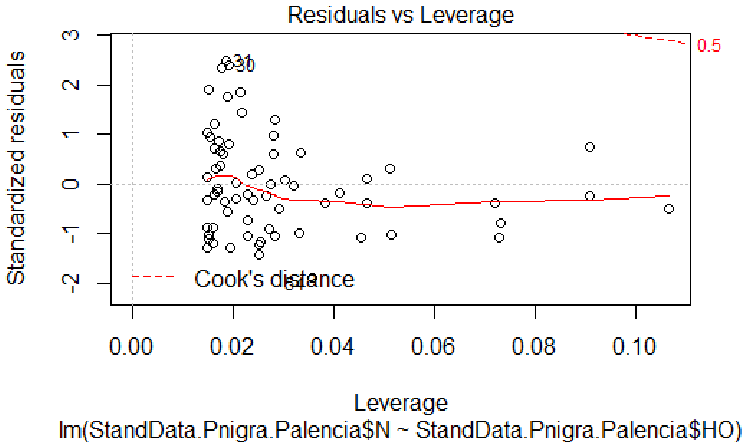
```
# plotting the basic graph information of the linearmodel_1
par(mfrow = c(1, 1))  #  Plot each graph individually
plot(linearmodel_1)  # Plot the model information
```

To obtain the following four graphs (*after hint return in the console window*):









Fitting a multiple linear regression

Sometime, you would like to explore if more than one independent variable (the x) is affecting the dependent variable (the y). When you assume that the relation between independent and dependent variables is linear you can use the multiple linear regression approach. In R you can expand our previous model by adding the new independent variables as follow:

```
# Multiple linear regression

linearmodel_2 <-   lm(N~  HO + DG  , data = Stand.Pnigra.Palencia)
summary (linearmodel_2)
```

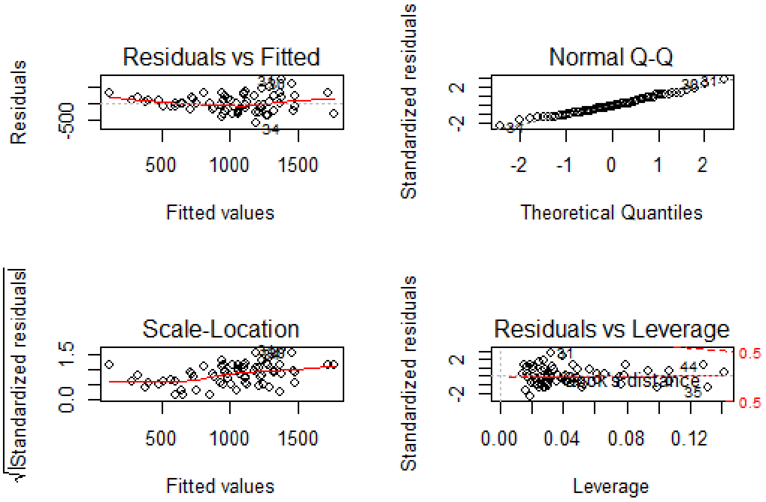
to obtain:

```
Call:
lm(formula = N ~ HO + DG, data = Stand.Pnigra.Palencia)

Residuals:
   Min     1Q Median     3Q    Max
-614.8 -194.3  -21.4  180.2  722.3

Coefficients:
                 Estimate          Std. Error          t value   Pr(>|t|)
(Intercept)      2108.35           170.92              12.335    < 2e-16   ***
HO                137.46            24.93               5.515    6.49e-07 ***
DG               -138.52            14.06              -9.853    1.60e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 266.9 on 65 degrees of freedom
Multiple R-squared:  0.6239,           Adjusted R-squared:  0.6123
F-statistic: 53.91 on 2 and 65 DF,  p-value: 1.58e-14
```

And with

```
# plotting the basic graph information of the linearmodel_2
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(linearmodel_2)  # Plot the model information
```

Obtain this plot:



You can obtain detailed procedures to draw residuals graphs and explore the linear regression outcomes at these links:

- https://www.r-bloggers.com/linear-regression-using-r/
- https://www.r-bloggers.com/visualising-residuals/