1st Summer School of the Institute for Language, Communication and the Brain

Applied mathematics, statistics and networks - Courses 2, 3 and 4 support

Bernard Giusiano

September 4-6, 2018 - Marseille, France

Table of Contents

This course presents the basic principles of statistical inference (estimation, mean comparison, variance analysis and linear regression) as well as a practical introduction to the R language. It corresponds to Bernard Giusiano's classes on Tuesday, Wednesday and Thursday.

PART III

Simple linear regression

Having seen with ANOVA how to test the relationship between a quantitative variable (*explained variable*) and one or more qualitative variables (*factors*, *explanatory variables*), let us now look at the relationship between two quantitative variables.

For that we will use other columns of the original data of the article of Reilly and Kean. Create a new script in which you will start by importing the Reilly and Kean data again and doing some initializations.

```
originalData <- read.csv2("imageability.csv")  # default: read.csv2(file, header =
TRUE, sep = ";", dec = ",")
par0 <- par(no.readonly = TRUE)  # backup the whole list of settable default
```

```
parameters.
set.seed(427)  # to initialize random generator at the same value each time
```

Do you remember how we list the names of the columns?

The variables that will interest us are:
* WORD
* BFRQ = verbal frequency,
* CNC = concretness,
* FAM = familiarity,
* IMG = imageability (how easily a person can form an associated mental image),
* KFFRQ = written frequency,
* NLET = number of letters,
* NSYL = number of syllables,
* Etymology = origin of the word,
* I_NMG_Mean_RT = the mean naming latency (in msec) for a particular word.

```
myData2 <- originalData[,c(1,5,6,7,8,9,10,12,19,28)]
summary(myData2)
```

```
##           WORD           BFRQ           CNC            FAM
##             : 617          :1718          : 617   Min.   :158.0
##  ABANDONMENT:   1   -      : 463   -      :  74   1st Qu.:441.0
##  ABDUCTION  :   1   1      : 371   595    :  24   Median :502.0
##  ABILITY    :   1   2      : 185   576    :  23   Mean   :488.9
##  ABODE      :   1   3      : 131   565    :  22   3rd Qu.:545.0
##  ABSCESS    :   1   4      :  76   590    :  22   Max.   :657.0
##  (Other)    :2872   (Other): 550   (Other):2712   NA's   :617
##       IMG           KFFRQ              NLET            NSYL
##  Min.   :210   Min.   :   1.00   Min.   : 2.000   Min.   :1.000
##  1st Qu.:410   1st Qu.:   5.00   1st Qu.: 5.000   1st Qu.:1.000
##  Median :495   Median :  15.00   Median : 6.000   Median :2.000
##  Mean   :485   Mean   :  47.64   Mean   : 6.297   Mean   :2.038
##  3rd Qu.:567   3rd Qu.:  47.00   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   :667   Max.   :3292.00   Max.   :14.000   Max.   :6.000
##  NA's   :617   NA's   :848       NA's   :617      NA's   :617
##    Etymology     I_NMG_Mean_RT
##  Min.   :1.000   Min.   : 510.9
##  1st Qu.:1.000   1st Qu.: 598.9
##  Median :1.000   Median : 634.3
##  Mean   :1.833   Mean   : 647.8
##  3rd Qu.:2.000   3rd Qu.: 682.3
##  Max.   :5.000   Max.   :1070.1
##  NA's   :617     NA's   :638
```

Let's clean up the data in the same way we did for ANOVA:

```
# recode Etymology
origins <- c("Latin","Germanic","Greek","Other","Unknown origin")
myData2[,"Etymology"] <- origins[myData2[,"Etymology"]]
myData2$Etymology <- as.factor(myData2$Etymology)
```

```
myData2 <- myData2[complete.cases(myData2),]        # delete rows with NA's
summary(myData2)
```

```
##           WORD           BFRQ          CNC            FAM
##  ABANDONMENT:   1          :951   -      :  69   Min.   :158.0
##  ABDUCTION  :   1   -      :396   595    :  22   1st Qu.:454.0
##  ABILITY    :   1   1      :361   565    :  21   Median :507.0
##  ABODE      :   1   2      :181   576    :  21   Mean   :495.1
##  ABSOLUTION :   1   3      :130   590    :  21   3rd Qu.:548.0
##  ABUNDANCE  :   1   4      : 75   558    :  19   Max.   :657.0
##  (Other)    :2636   (Other):548   (Other):2469
##       IMG             KFFRQ              NLET             NSYL
##  Min.   :210.0   Min.   :   1.00   Min.   : 2.000   Min.   :1.00
##  1st Qu.:409.0   1st Qu.:   5.00   1st Qu.: 4.000   1st Qu.:1.00
##  Median :494.0   Median :  15.00   Median : 6.000   Median :2.00
##  Mean   :484.6   Mean   :  47.71   Mean   : 6.269   Mean   :2.03
##  3rd Qu.:567.0   3rd Qu.:  47.00   3rd Qu.: 8.000   3rd Qu.:3.00
##  Max.   :667.0   Max.   :3292.00   Max.   :14.000   Max.   :6.00
##
##           Etymology    I_NMG_Mean_RT
##  Germanic      : 830   Min.   : 510.9
##  Greek         : 131   1st Qu.: 596.4
##  Latin         :1395   Median : 632.2
##  Other         : 116   Mean   : 645.0
##  Unknown origin: 170   3rd Qu.: 678.7
##                        Max.   :1070.1
##
```

BFRQ and CNC variables are not considered numerical because they have "-" values. Let's remove this problem.

```
myData2[(myData2$BFRQ=="")|(myData2$BFRQ=="-"),"BFRQ"] <- NA
myData2[(myData2$CNC=="")|(myData2$CNC=="-"),"CNC"] <- NA
myData2$BFRQ <- as.numeric(myData2$BFRQ)
myData2$CNC <- as.numeric(myData2$CNC)
summary(myData2)
```

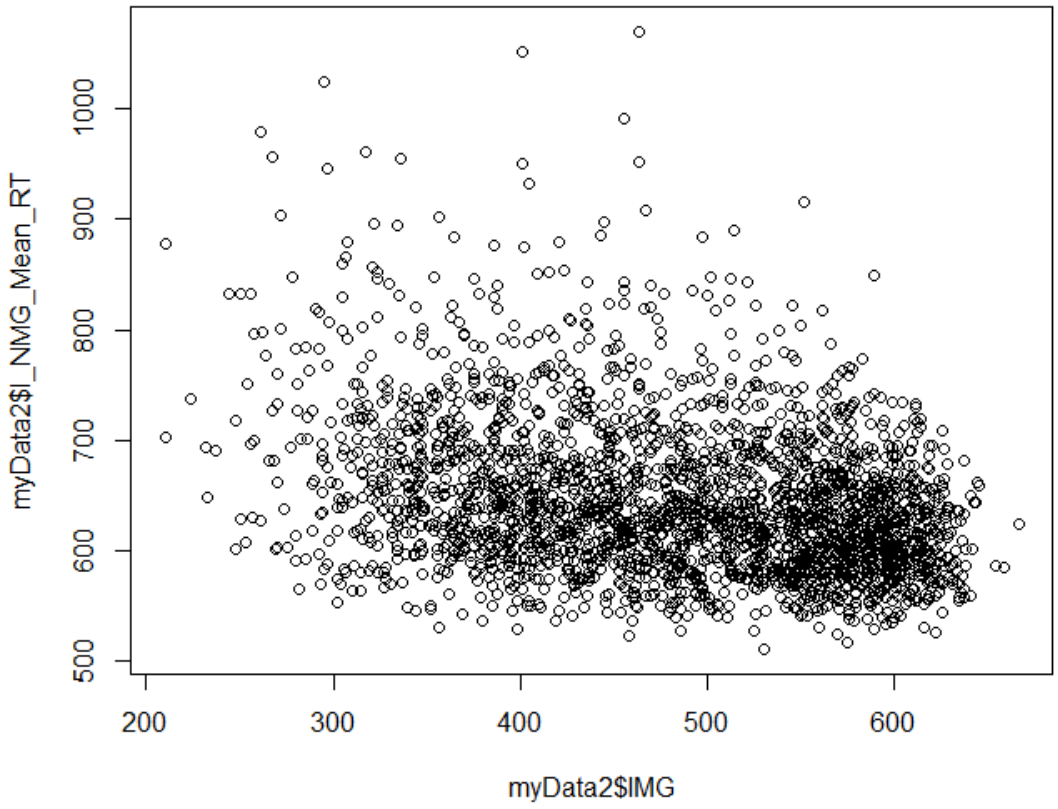
```
##           WORD           BFRQ           CNC            FAM
##  ABANDONMENT:   1   Min.   : 3     Min.   :  3.0   Min.   :158.0
##  ABDUCTION  :   1   1st Qu.: 3     1st Qu.:134.0   1st Qu.:454.0
##  ABILITY    :   1   Median :28     Median :254.0   Median :507.0
##  ABODE      :   1   Mean   :34     Mean   :235.8   Mean   :495.1
##  ABSOLUTION :   1   3rd Qu.:54     3rd Qu.:342.0   3rd Qu.:548.0
##  ABUNDANCE  :   1   Max.   :99     Max.   :419.0   Max.   :657.0
##  (Other)    :2636   NA's   :1347   NA's   :69
##       IMG             KFFRQ              NLET             NSYL
##  Min.   :210.0   Min.   :   1.00   Min.   : 2.000   Min.   :1.00
##  1st Qu.:409.0   1st Qu.:   5.00   1st Qu.: 4.000   1st Qu.:1.00
##  Median :494.0   Median :  15.00   Median : 6.000   Median :2.00
##  Mean   :484.6   Mean   :  47.71   Mean   : 6.269   Mean   :2.03
##  3rd Qu.:567.0   3rd Qu.:  47.00   3rd Qu.: 8.000   3rd Qu.:3.00
##  Max.   :667.0   Max.   :3292.00   Max.   :14.000   Max.   :6.00
```

```
##
##           Etymology    I_NMG_Mean_RT
##  Germanic      : 830   Min.   : 510.9
##  Greek         : 131   1st Qu.: 596.4
##  Latin         :1395   Median : 632.2
##  Other         : 116   Mean   : 645.0
##  Unknown origin: 170   3rd Qu.: 678.7
##                        Max.   :1070.1
##
```

Principle of simple linear regression

Let's start with a single quantitative variable explained, the naming latency (I_NMG_Mean_RT) and a single explanatory quantitative variable (IMG, imageability in its quantitative form).

```
plot(myData2$IMG, myData2$I_NMG_Mean_RT)
```



This cloud of points does not tell us much.
Would these two variables be *independent*? If they are, their **covariance** must be 0.
The covariance characterizes the simultaneous variations of two random variables: it will be positive when the differences between the variables and their means tend to be of the same sign, negative otherwise.

$$cov = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n}$$

```
cov(myData2$IMG, myData2$I_NMG_Mean_RT)
```

```
## [1] -2118.609
```

So there is a relationship between the two variables. They vary in the opposite direction. **Correlation coefficient** is a standardized form of covariance to evaluate the intensity of the relationship between the two variables. Its denominator is the product of the numerators of the standard deviation of the two variables. It is between -1 and +1 and is calculated by the following formula:

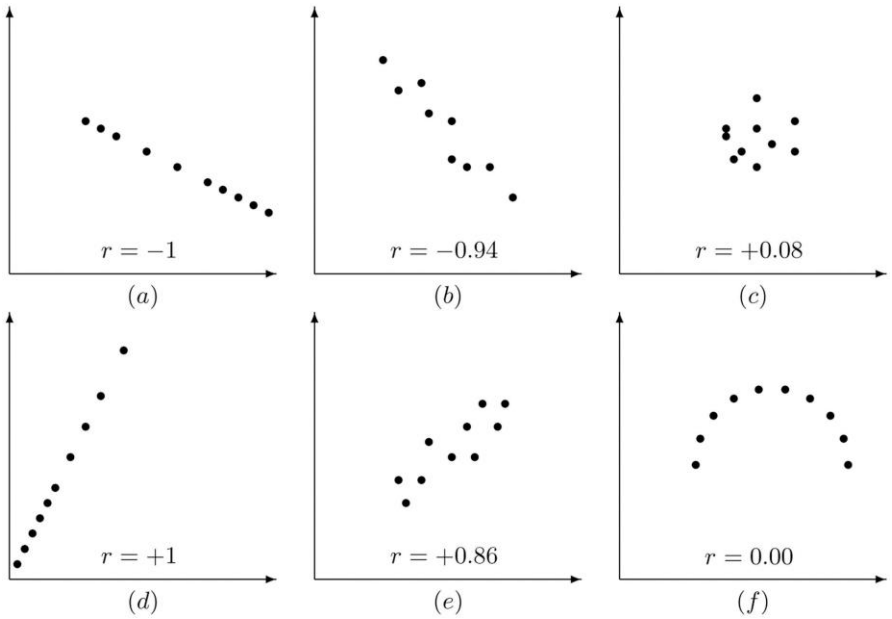
$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})\,(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

With x_i the values of the variable IMG, $\overline{x}$ its mean, y_i the values of I_NMG_Mean_RT and $\overline{y}$ its mean, we can evaluate the intensity of the relationship with the R function cor().

```
cor(myData2$IMG,myData2$I_NMG_Mean_RT)
```

```
## [1] -0.3221554
```

Binding that measures the correlation coefficient is symmetric: it is the same between x and y and between y and x. But this measure of correlation implies a **linear link** between the two variables, that is to say a link of the type $y = ax + b$. The shape of the link is often noticeable in the cloud of points drawn on a graph.



http://math.fdltcc.edu/wetherbee/books/m1030/IntroStatistics.pdf

A simple transformation on the correlation coefficient makes it possible to test its significance on the basis of Student's t-law, the null hypothesis being that the coefficient in the population is equal to 0 and that its value calculated on the sample is different from 0 only because of sampling fluctuations.

```
cor.test(myData2$IMG,myData2$I_NMG_Mean_RT)
```

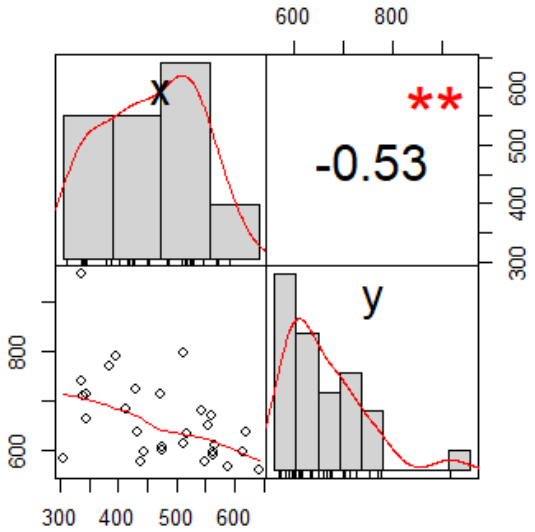
```
##
##  Pearson's product-moment correlation
##
## data:  myData2$IMG and myData2$I_NMG_Mean_RT
## t = -17.485, df = 2640, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3559173 -0.2875535
## sample estimates:
##       cor
## -0.3221554
```

Now that we know how to evaluate the relationship between two variables, the **linear regression** method allows us to go further and to *predict* the values of one variable according to the values of the other once we have chosen the *direction* of the relationship that interests us.

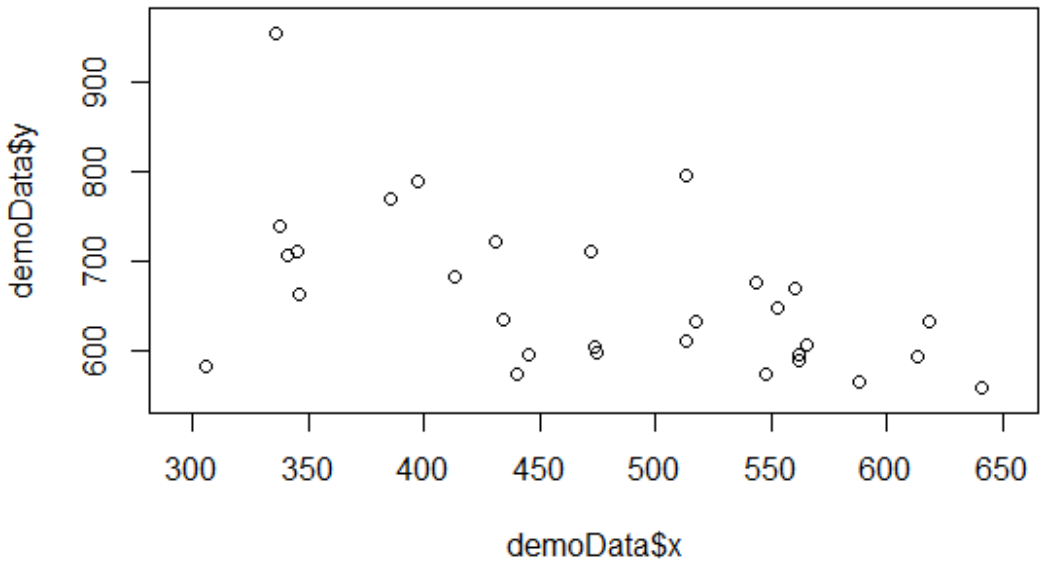
To simplify the plots of our demonstration, we will draw a small subsample of our data and rename *x* the imageability and *y* the naming latency. The variable *y* is designated in different ways: *response variable*, *variable to explain*, *dependent variable*, *outcome*; the variable *x*: *predictor*, *regressor*, *explanatory variable*, *independent variable*.

```
library(PerformanceAnalytics)

demoData <- myData2[sample(nrow(myData2), 30), c("IMG","I_NMG_Mean_RT")]
colnames(demoData) <- c("x","y")
chart.Correlation(demoData)
```




```
plot(demoData$x, demoData$y, xlim=c(min(demoData$x)-10, max(demoData$x)+10),
ylim=c(min(demoData$y)-10, max(demoData$y)+10))
```



Linear regression aims to *fit* a straight line to data that for any value of x gives the best prediction of y. In the equation of this line $y = ax + b$, the regression calculate the **slope** a of the line and its ordinate at the origin b, the **intercept**.

Many functions exist in R to compute many different types of regression. For now, we use the simplest function, lm() (for "linear model").

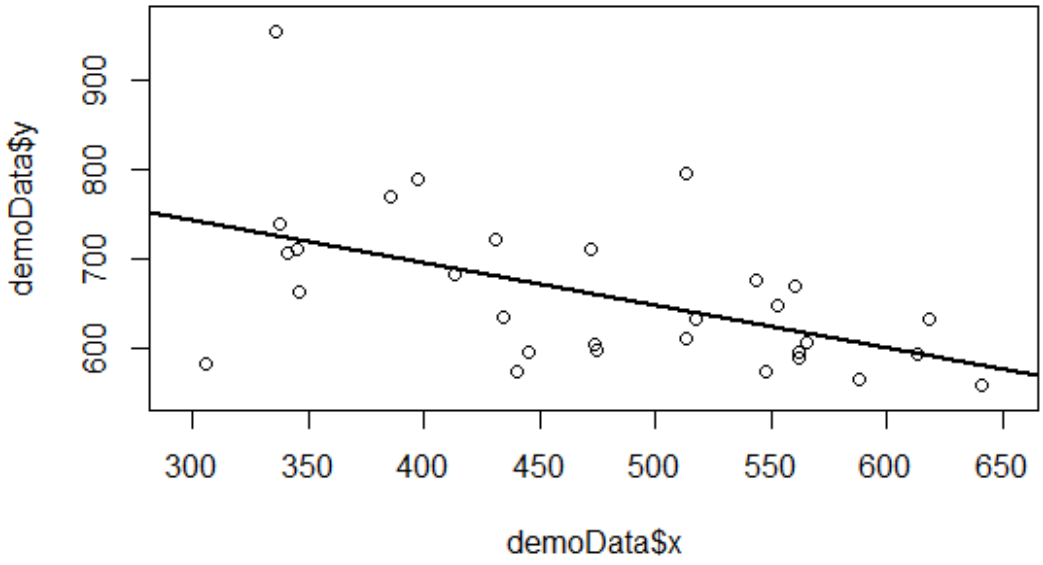
```
myDemoRegModel <- lm(y ~ x, data=demoData)
summary(myDemoRegModel)

##
## Call:
## lm(formula = y ~ x, data = demoData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -157.82  -43.27  -10.46   41.21  228.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 886.6218    70.5978  12.559 5.05e-13 ***
## x            -0.4751     0.1455  -3.266  0.00288 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.97 on 28 degrees of freedom
## Multiple R-squared:  0.2758, Adjusted R-squared:   0.25
## F-statistic: 10.67 on 1 and 28 DF,  p-value: 0.002881
```

We will detail these results later, but let's start with the statistical magic that finds this line that seems to summarize the link between the variable x and the variable y. This line has for equation ($\hat{y}$ not to confuse estimated y with observed y):

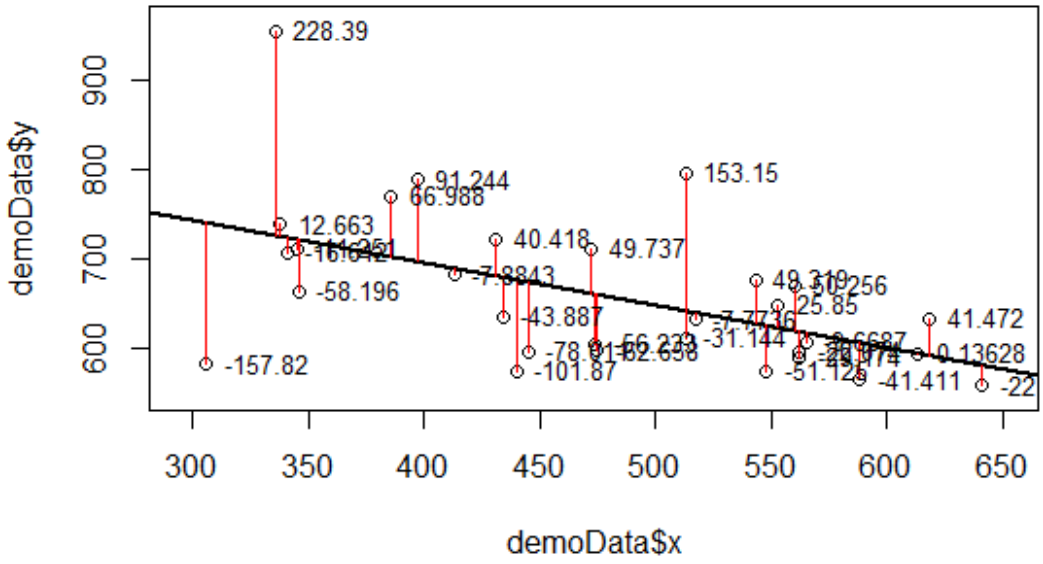
$$\hat{y} = -0.4751x + 886.6218$$

```
plot(demoData$x, demoData$y, xlim=c(min(demoData$x)-10, max(demoData$x)+10),
ylim=c(min(demoData$y)-10, max(demoData$y)+10))
# draw the regression line
abline(myDemoRegModel, lwd=2)
```



In the following graph, the red lines represent the **residuals**, i.e. the differences between the observed data and the data predicted by the regression, more precisely the differences between the ordinates since the values of y are predicted for the values of x observed.

```
plot(demoData$x, demoData$y, xlim=c(min(demoData$x)-10, max(demoData$x)+10),
ylim=c(min(demoData$y)-10, max(demoData$y)+10))
abline(myDemoRegModel, lwd=2)
# calculate residuals and predicted values
res <- signif(residuals(myDemoRegModel), 5)  # try "? signif"
pre <- predict(myDemoRegModel)
# plot distances between points and the regression line
segments(demoData$x, demoData$y, demoData$x, pre, col="red")
# add labels (res values) to points
text(demoData$x, demoData$y, labels=res, cex=0.8, pos=4)
```



The principle is to find the straight line which passes as close as possible to the set of points by turning the line on a hinge point having as coordinates $(\overline{x}, \overline{y})$. This fitting aims to minimize the sum of the squared errors (red lines) between observed points and predicted points (squared errors so that negative values do not negate positive values). For models with only one regressor, the calculation of parameters of this straight line is easy:

$$slope: a = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$intersect: b = \overline{y} - a\overline{x}$$

The slope corresponds to the variation of y when x varies by one unit.

```
yPred <- predict(myDemoRegModel,data.frame(x=c(500,501)))
yPred[2] - yPred[1]
```

```
##          2
## -0.4751029
```

This method of estimation for linear models is called the *ordinary least squares (OLS) linear regression*. It is the most common method, but there are others, especially the maximum likelihood method, that have extended the concept of regression to distributions other than normal distributions (*generalized linear model of regression*).

Results of simple linear regression in R

Let's go back to the application of the lm() function on our demo sample.
Did you try the str() function on its result?
Let's detail what gives us the function summary():

```
Call:
lm(formula = y ~ x, data = demoData)
```

Just recall the called function (*lm()*) with the formula expressing the model ($y \sim x$) and the name of the data frame containing the data of the formula.

The equation giving the observed y must add a term to the equation giving the estimated y ($\hat{y}$): the errors (ϵ_i) (or **residuals**) represented by the red lines on the graph seen above. It's the difference between the observed values and the values predicted by the model.

$$y_i = -0.4751 x_i + 886.6218 + \epsilon_i$$

```
Residuals:
    Min      1Q  Median      3Q     Max
-157.82  -43.27  -10.46   41.21  228.39
```

The Residuals section gives an idea of the symmetry of the distribution of residuals. We will see later that the residuals must be normally distributed.

```
# We can calculate this section like so:
summary(demoData$y - myDemoRegModel$fitted.values)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -157.80  -43.27  -10.46    0.00   41.21  228.40
```

The Coefficients section gives the estimate of intercept and slope (**coefficient** of the predictor x):

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 886.6218    70.5978  12.559 5.05e-13 ***
x            -0.4751     0.1455  -3.266  0.00288 **
```

The *standard error of the coefficient* captures how much uncertainty is associated with this coefficient.

```
n = length(myDemoRegModel$residuals)
SSE = sum(myDemoRegModel$residuals**2)
SSxx = sum((demoData$x - mean(demoData$x))**2)
# Standard Error of coefficient of x
sqrt((SSE/(n-2))/SSxx)

## [1] 0.1454752
```

This standard error is used to compute a t-value that is needed to statistically test the signifiance of the coefficient with H_0 is "the coefficient equal 0". If, as it's the case here, Pr(>|t|) <= 0.05 for x then we can conclude with an α risk of 5% that y depends on x.

```
# t-value for coefficient of x = estimate / std. error
coef(myDemoRegModel)[2] / sqrt((SSE/(n-2))/SSxx)

##         x
## -3.265869
```

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

The way to interpret the stars in terms of significativity (* significant, ** very significant, *** highly significant).

```
Residual standard error: 75.97 on 28 degrees of freedom
```

The *residual standard error* is like a global standard deviation of the error but with a number of degrees of freedom that take into account the number of predictors. It's a measure of the *quality* of the linear regression: the smaller it is, the closer the observed values are to the regression line.

```
k = 1  # number of predictors
df <- n - (1 + k)
df

## [1] 28

# Residual Standard Error
sqrt(SSE / df)

## [1] 75.96789

Multiple R-squared:  0.2758,    Adjusted R-squared:   0.25
```

The R-squared statistic, also called the *coefficient of determination*, provides a measure of the proportion of the variance in the data that's explained by the model. In our example, the R-squared we get is 0.2758; it means that roughly 28% of the variance found in the response variable y can be explained by the predictor variable x.

```
SSyy = sum((demoData$y - mean(demoData$y))**2)
# Multiple R-Squared (Coefficient of Determination):
(SSyy - SSE)/SSyy

## [1] 0.2758477
```

In multiple regression (that we will see later), the R^2 will increase as more variables are included in the model; that's why the adjusted R^2 is the preferred measure as it adjusts for the number of variables considered.

```
F-statistic: 10.67 on 1 and 28 DF,  p-value: 0.002881
```

The F-Statistic is a global test that checks if at least one of the coefficients are non-zero.

```
# F-statistic:
((SSyy - SSE)/k) / (SSE/(n-(k + 1)))

## [1] 10.6659
```

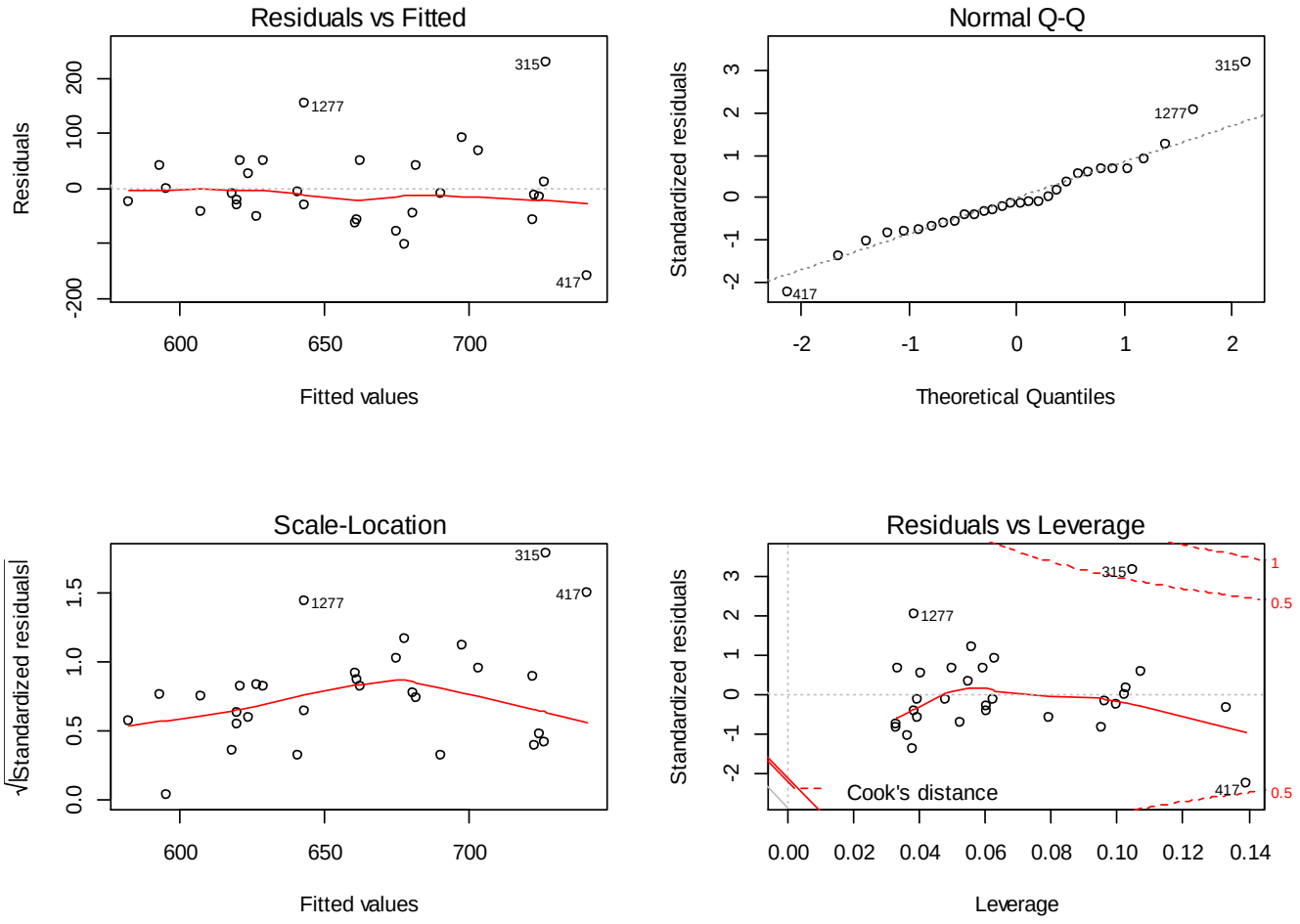
Conditions of application of linear regression

All the calculations we made about this linear regression are valid only if the following assumptions are true.
* The relationship between the two variables must be globally linear, at least roughly. This is the graphical representation that can convince us easily.
* The residuals must be independent. If the data come from different individuals, they are usually independent. But if the independent variable is temporal, the residuals are probably not independent, or if the values correspond to repeated measurements on the same subjects, then the residuals are not independent.

* The residuals have a homogeneous variance (*homoscedasticity*), i.e. the variance around the regression line is the same for all values of the independent variable.
* The residuals must follow a normal distribution with a mean of zero so that the coefficient significance tests are not biased.

As with ANOVA, the function plot() applied to the result of the regression gives us the means to check some of these conditions.

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

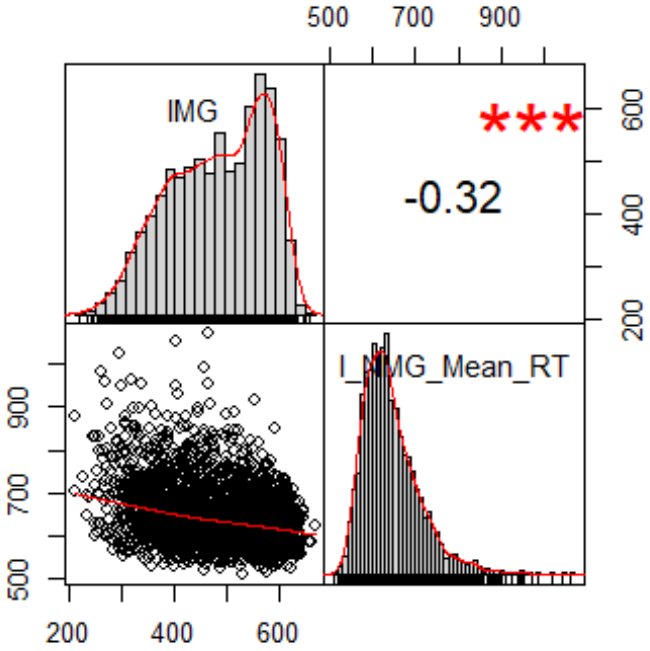



```
par(par0)
```

Application

Let us take the complete data that we have prepared at the beginning of this session.

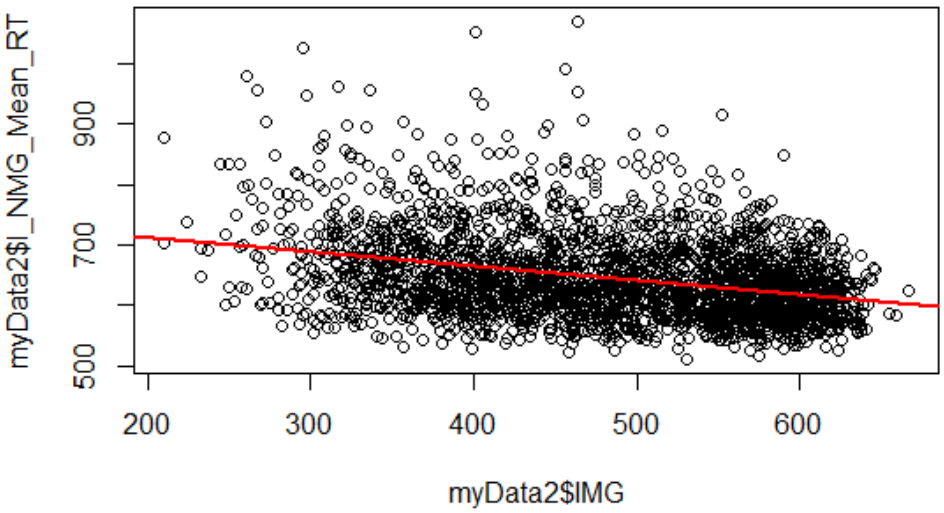
```
chart.Correlation(myData2[, c("IMG","I_NMG_Mean_RT")])
```



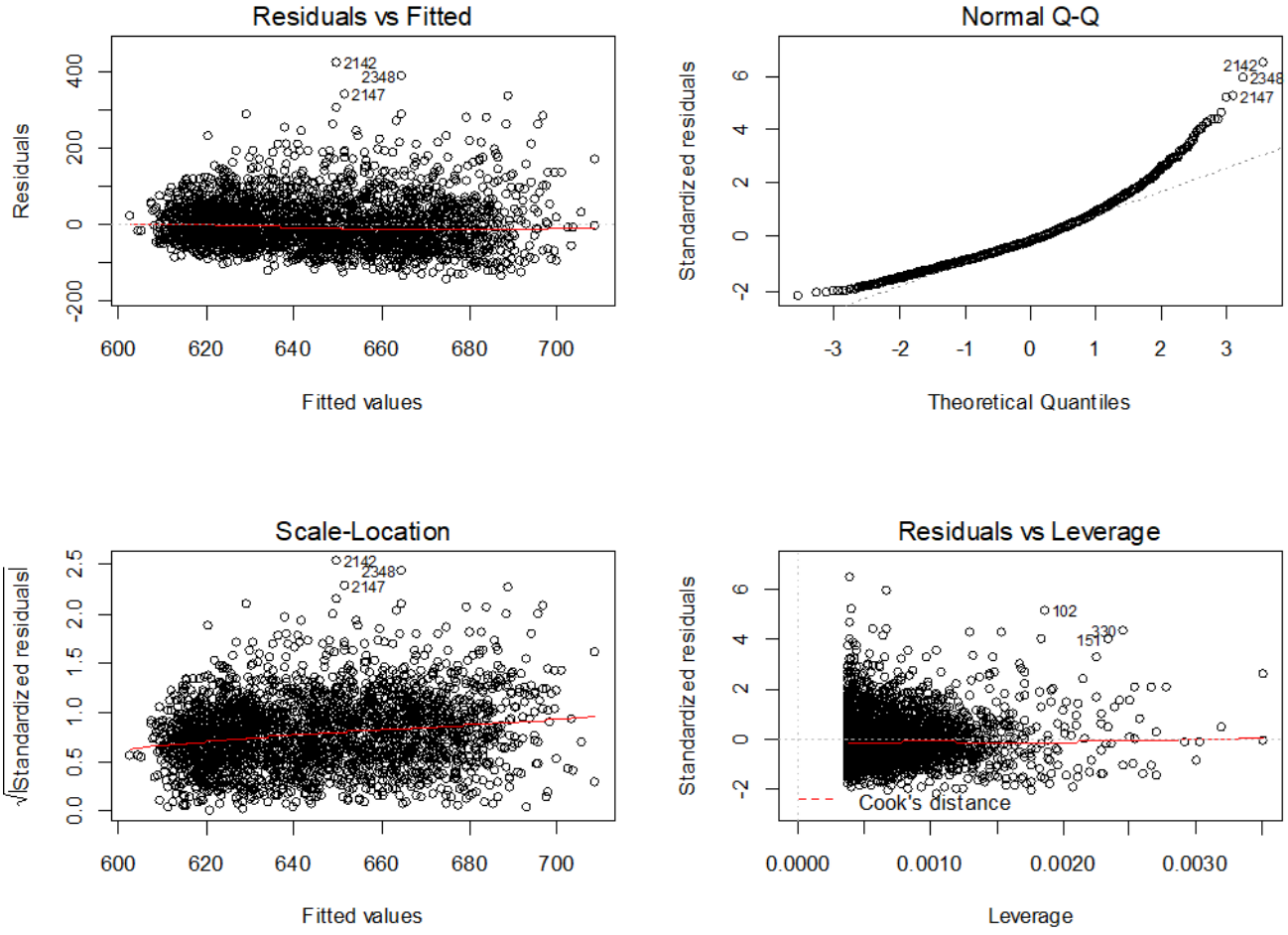

```
myRegModel <- lm(I_NMG_Mean_RT ~ IMG, data=myData2)
summary(myRegModel)

##
## Call:
## lm(formula = I_NMG_Mean_RT ~ IMG, data = myData2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144.21  -43.65  -11.38   33.48  420.33
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 757.6752     6.5691  115.34   <2e-16 ***
## IMG          -0.2326     0.0133  -17.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.24 on 2640 degrees of freedom
## Multiple R-squared:  0.1038, Adjusted R-squared:  0.1034
## F-statistic: 305.7 on 1 and 2640 DF,  p-value: < 2.2e-16

plot(myData2$IMG, myData2$I_NMG_Mean_RT)
# draw the regression line
abline(myRegModel, lwd=2, col="red")
```




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



```
par(par0)
```

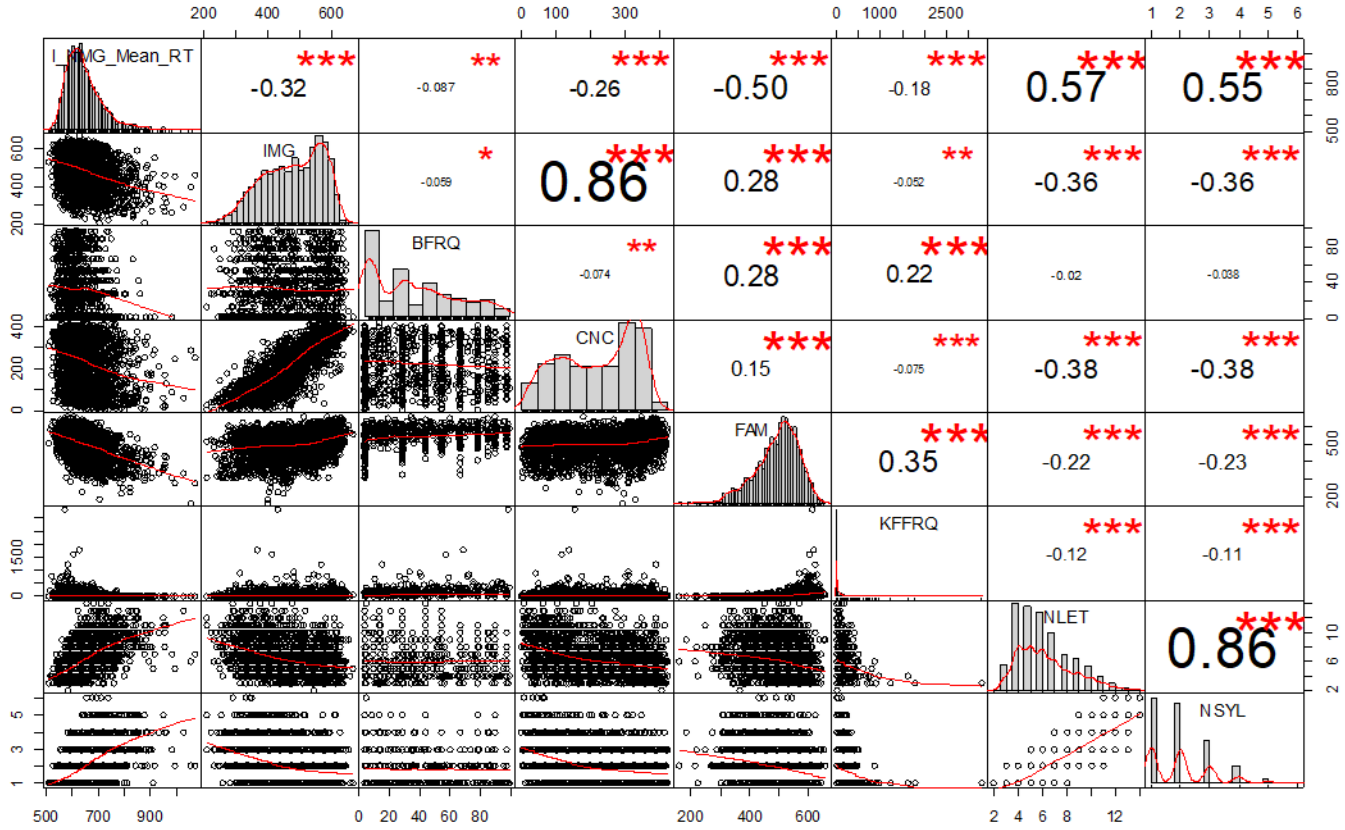
Compare these results of the regression with that of the demonstration sample.

Multiple linear regression

In the results of the analysis we have just done to study the link between naming latency and word imageability, we can see that, although this link is highly significant (Pr(>|t|) < 2e-16), it is not very strong (r = -0.32) and the imageability only explains 10% of the variance of the naming latency (R^2 = 0.1038).

The naming latency probably depends on other factors. Let us see if among the quantitative variables of our sample, some also have a link with this variable to explain. And look at some different ways to draw correlograms.

```
chart.Correlation(myData2[, c("I_NMG_Mean_RT", "IMG", "BFRQ", "CNC", "FAM",
"KFFRQ", "NLET", "NSYL")])
```



If we take as a reference the link with the imageability, the familiarity, the number of letters and the number of syllables have a greater correlation coefficient with the naming latency. Let's start by adding FAM to our model.

```
myMultiRegModel <- lm(I_NMG_Mean_RT ~ IMG + FAM, data=myData2)
summary(myMultiRegModel)
```

```
##
## Call:
## lm(formula = I_NMG_Mean_RT ~ IMG + FAM, data = myData2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -165.87  -41.25   -6.15   31.05  331.45
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 919.35153    8.61955  106.66   <2e-16 ***
## IMG          -0.14320    0.01240  -11.55   <2e-16 ***
## FAM          -0.41397    0.01614  -25.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.38 on 2639 degrees of freedom
## Multiple R-squared:  0.2826, Adjusted R-squared:  0.282
## F-statistic: 519.8 on 2 and 2639 DF,  p-value: < 2.2e-16
```

NLET and NSYL being very correlated, we will only take NLET and add it to the model.

```
myMultiRegModel2 <- lm(I_NMG_Mean_RT ~ IMG + FAM + NLET, data=myData2)
summary(myMultiRegModel2)
```

```
##
## Call:
## lm(formula = I_NMG_Mean_RT ~ IMG + FAM + NLET, data = myData2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115.42  -33.65   -5.21   27.72  335.38
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 749.95651    9.25873  81.000  < 2e-16 ***
## IMG          -0.03448    0.01124  -3.067  0.00219 **
## FAM          -0.35874    0.01400 -25.632  < 2e-16 ***
## NLET         14.25332    0.46696  30.524  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.2 on 2638 degrees of freedom
## Multiple R-squared:  0.4698, Adjusted R-squared:  0.4692
## F-statistic: 779.3 on 3 and 2638 DF,  p-value: < 2.2e-16
```

The three predictor have a significant effect on the dependent variable. The variance explained by this model is close to 47% now.

We can test with an ANOVA if the difference between the residual sum of squares of the two last models is significant.

```
# the function must be anova() (and no more aov())
anova (myMultiRegModel, myMultiRegModel2)
```

```
## Analysis of Variance Table
##
## Model 1: I_NMG_Mean_RT ~ IMG + FAM
## Model 2: I_NMG_Mean_RT ~ IMG + FAM + NLET
##   Res.Df     RSS Df Sum of Sq     F    Pr(>F)
## 1   2639 8994814
## 2   2638 6647149  1   2347665 931.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA shows that the addition of NLET significantly improves the model (RSS for model 2 is less than RSS for model 1).

Is there one or more interactions between these three regressors?

```
myMultiRegModel2interact <- lm(I_NMG_Mean_RT ~ IMG * FAM * NLET, data=myData2)
summary(myMultiRegModel2interact)
```

```
##
## Call:
## lm(formula = I_NMG_Mean_RT ~ IMG * FAM * NLET, data = myData2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -136.73  -32.04   -5.19   26.46  332.79
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.197e+02  9.911e+01   8.271  < 2e-16 ***
## IMG          -4.516e-01  2.131e-01  -2.119  0.03422 *
## FAM          -5.227e-01  1.969e-01  -2.654  0.00799 **
## NLET          2.571e+01  1.338e+01   1.921  0.05479 .
## IMG:FAM       8.853e-04  4.170e-04   2.123  0.03385 *
## IMG:NLET      1.640e-02  3.011e-02   0.545  0.58613
## FAM:NLET     -1.993e-02  2.709e-02  -0.736  0.46203
## IMG:FAM:NLET -4.093e-05  6.002e-05  -0.682  0.49534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.38 on 2634 degrees of freedom
## Multiple R-squared:  0.4876, Adjusted R-squared:  0.4863
## F-statistic: 358.1 on 7 and 2634 DF,  p-value: < 2.2e-16
```

The effect of NLET disappears in favor of the interaction between IMG and FAM. We can simplify the model by taking into account only the significant interaction.

```
myMultiRegModel2interact2 <- lm(I_NMG_Mean_RT ~ IMG * FAM + NLET, data=myData2)
summary(myMultiRegModel2interact2)
```

```
##
## Call:
## lm(formula = I_NMG_Mean_RT ~ IMG * FAM + NLET, data = myData2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -127.26  -32.92   -5.16   26.62  337.73
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.665e+02  3.177e+01  30.421  < 2e-16 ***
## IMG         -5.078e-01  6.742e-02  -7.532 6.83e-14 ***
## FAM         -7.971e-01  6.313e-02 -12.627  < 2e-16 ***
## NLET         1.426e+01  4.626e-01  30.832  < 2e-16 ***
## IMG:FAM      9.501e-04  1.335e-04   7.118 1.40e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.73 on 2637 degrees of freedom
## Multiple R-squared:  0.4798, Adjusted R-squared:  0.479
## F-statistic: 608.1 on 4 and 2637 DF,  p-value: < 2.2e-16
```

All regressors and interaction have a significant effect. Is this model better than the previous one when the R^2 is worse? The Akaike information criterion (AIC) can help us determine the best model.

```
AIC(myMultiRegModel2interact)
```

```
## [1] 28113.33
```

```
AIC(myMultiRegModel2interact2)
```

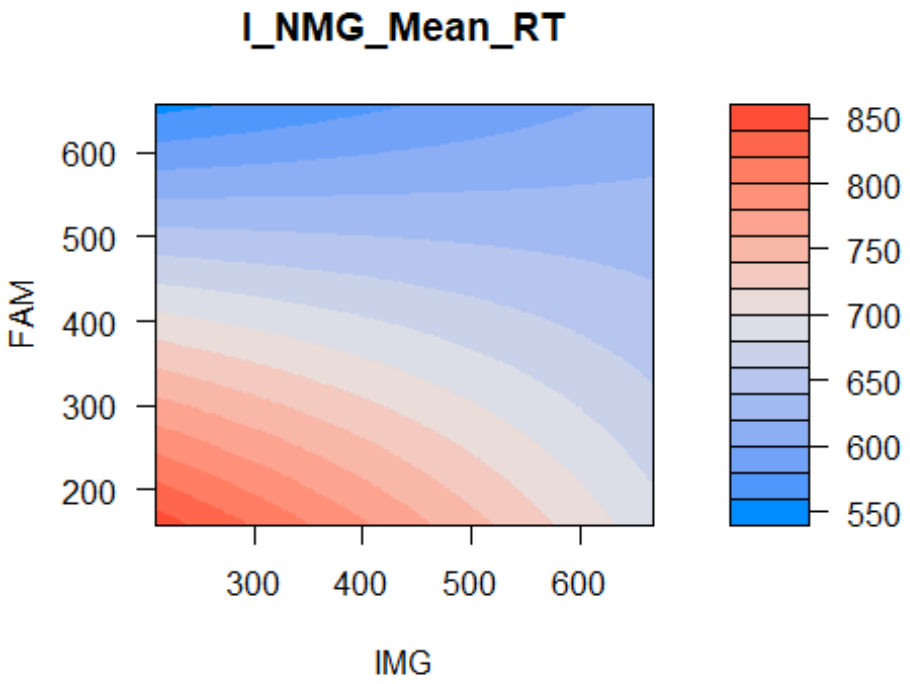
```
## [1] 28147.32
```

The best model is the one with the minimum AIC value. But *parsimony* is a rule of thumb in choosing our preferred regression model, so we utimately choose the simplest model with the fewest terms.

Let's illustrate the interaction with the visreg package.

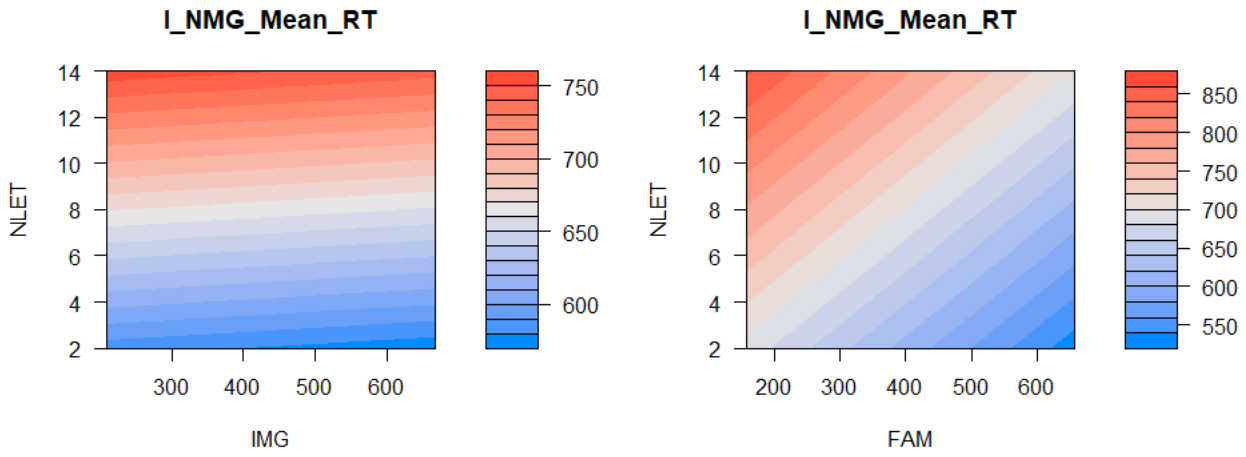
```
library(visreg)

visreg2d(myMultiRegModel2interact2,"IMG","FAM")
```



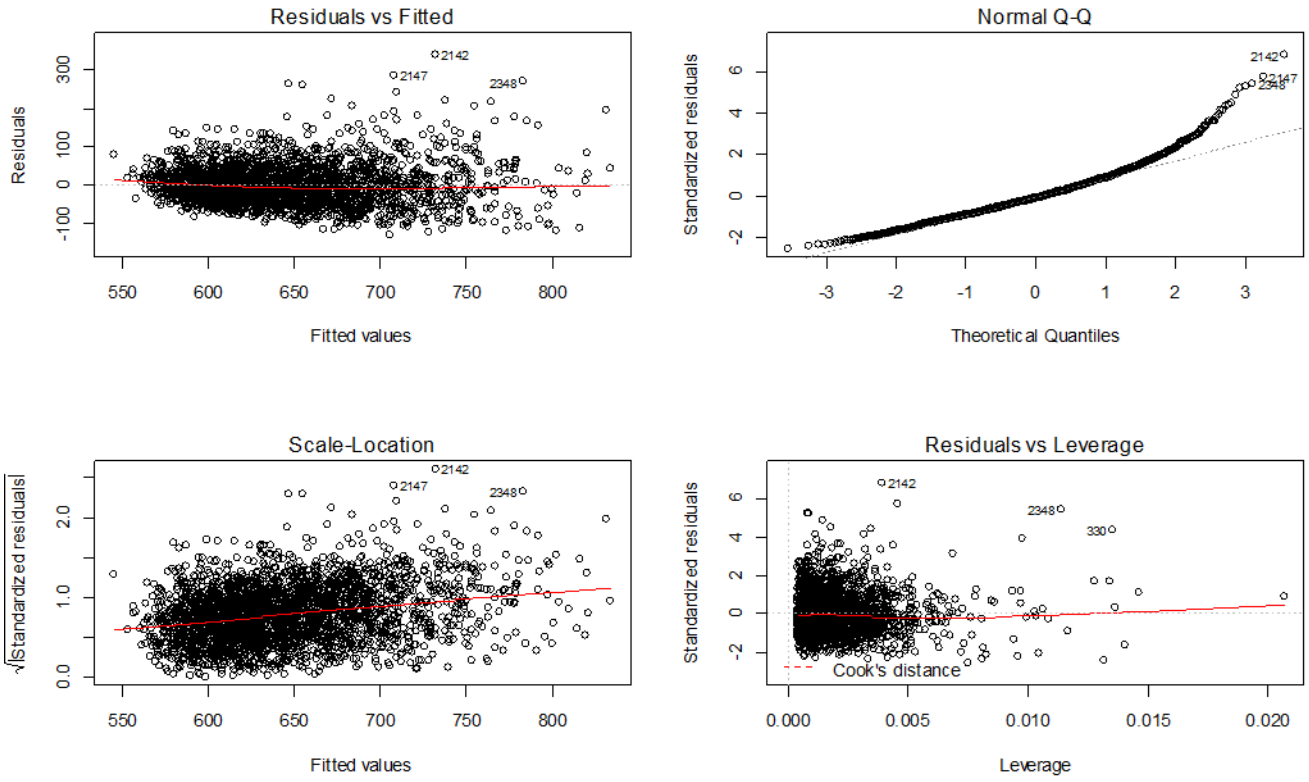
To see the differences with variables without interaction:

```
visreg2d(myMultiRegModel2interact2,"IMG","NLET")
visreg2d(myMultiRegModel2interact2,"FAM","NLET")
```



Finally, check if the application conditions are met with some graphics.

```
par(mfrow=c(2,2))
plot(myMultiRegModel2interact2)
```




```
par(par0)
```

ANOVA and linear regression are the same thing

Let's take our example of one-way ANOVA:

```
originalData4 <- read.csv2("imageability.csv")
names(originalData4)[2]<-"Imageability"
myData4 <- originalData4[, c("I_NMG_Mean_RT", "Imageability")]
myData4 <- myData4[complete.cases(myData4),]
myData4$Imageability <- droplevels(myData4$Imageab)
summary(myData4)
```

```
##  I_NMG_Mean_RT        Imageability
##  Min.   : 510.9   high-img  :1380
##  1st Qu.: 598.9   low-img   : 627
##  Median : 634.3   medium-img: 849
##  Mean   : 647.8
##  3rd Qu.: 682.3
##  Max.   :1070.1
```

Here is the analysis of variance:

```
fitByAnova <- aov(I_NMG_Mean_RT ~ Imageability, data=myData4)
summary(fitByAnova)
```

```
##                Df   Sum Sq Mean Sq F value Pr(>F)
## Imageability    2   967894  483947   107.5 <2e-16 ***
## Residuals    2853 12843431    4502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
meanByGroup <- by(myData4$I_NMG_Mean_RT, myData4$Imageability, mean)
meanByGroup
```

```
## myData4$Imageability: high-img
## [1] 630.608
## ------------------------------------------------------------
## myData4$Imageability: low-img
## [1] 676.7269
## ------------------------------------------------------------
## myData4$Imageability: medium-img
## [1] 654.254
```

```
TukeyHSD(fitByAnova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = I_NMG_Mean_RT ~ Imageability, data = myData4)
##
## $Imageability
##                           diff       lwr       upr p adj
## low-img-high-img     46.11894  38.54155  53.69632     0
## medium-img-high-img  23.64605  16.78358  30.50853     0
## medium-img-low-img  -22.47288 -30.75754 -14.18823     0
```

In ANOVA, the categorical variable is *effect coded*, which means that each group mean is compared to the global mean.

We can apply a linear regression to these data with the same formula:

```
fitByLinReg <- lm(I_NMG_Mean_RT ~ Imageability, data=myData4)
summary(fitByLinReg)
```

```
##
## Call:
## lm(formula = I_NMG_Mean_RT ~ Imageability, data = myData4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.40  -46.76  -12.15   33.11  415.85
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             630.608      1.806  349.15  < 2e-16 ***
```

```
## Imageabilitylow-img      46.119      3.231   14.27  < 2e-16 ***
## Imageabilitymedium-img   23.646      2.927    8.08 9.47e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.09 on 2853 degrees of freedom
## Multiple R-squared:  0.07008,    Adjusted R-squared:  0.06943
## F-statistic: 107.5 on 2 and 2853 DF,  p-value: < 2.2e-16
```

In the regression, the categorical variable is **dummy coded**. The dummy coding creates two 1/0 variables:
* Imageabilitylow-img = 1 for observations with the *low-img* level, 0 otherwise;
* Imageabilitymedium-img = 1 for observations with the *medium-img* level, 0 otherwise.
Observations with *high-img* level have a 0 value on both of these variables; *high-img* group is called the *reference group* (the first level by default).

Each group intercept is compared to the reference group intercept. Since the intercept is defined as the mean value when all other predictors = 0, and there are no other predictors, the three intercepts are just means.

We can see that:
* the F-statistic has the same value (107.5) in ANOVA and in linear regression,
* the intercept of linear regression is the mean of group *high-img*, the reference group,
* the coefficient estimate is the difference between the mean for the group and the intercept (or mean of the reference group) as we can verify in the mean differences given by the ANOVA post-hoc Tukey test.

```
intercept <- fitByLinReg$coefficient[1]
intercept
```

```
## (Intercept)
##     630.608
```

```
meanByGroup[2:3] - intercept
```

```
## myData4$Imageability
##    low-img medium-img
##   46.11894   23.64605
```

Exercices

- Re-do the simple regression graph to show the intercept.
- What do we get by multiplying the slope of a simple regression line by the variance of x?
- What is the effect of standardizing the two variables of our simple linear regression ?

Brain break

Green jelly beans linked to acne!

"Facts are stubborn, but statistics are more pliable." - Mark Twain

"To consult a statistician after an experiment is finished is often merely to ask him to conduct a post-mortem examination. He can perhaps say what the experiment died of." - R.A.Fisher

Some good Internet pages on linear regression :
* https://uc-r.github.io/linear_regression
* http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-tutorial-and-examples
* https://www.theanalysisfactor.com/13-steps-regression-anova/

To go further on ANOVA and linear regression with R, the free PDF book by Julian Faraway, a classic.