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 II

The problem of multiple comparisons

In the article by Martin Reite et al. which inspired our example to explain the mean comparison test, two other populations were studied: patients with bipolar disorder and patients with schizoaffective disorders. With what we have learned, we could compare these four groups with each other: control vs. schizophrenia, schizophrenia vs. bipolar, control vs. schizoaffective, etc. What would make a combination of 2 out of 4 = 6 tests.

```
combn(c("control","schizo","bipol","schizoaff"),2)
```

```
##      [,1]      [,2]      [,3]        [,4]     [,5]        [,6]
## [1,] "control" "control" "control"   "schizo" "schizo"    "bipol"
## [2,] "schizo"  "bipol"   "schizoaff" "bipol"  "schizoaff" "schizoaff"
```

For each of these tests, we have seen that the probability of making an error by concluding that the two means are different when they are not (*false positive*) corresponds to the α risk. The complementary probability therefore corresponds to the probability of not making an error: $1 - \alpha$

When two tests are performed, the probability of not making an error (of the false positive type) during the first test AND during the second test is the probability of not making an error at all. It is equal to the product of the two initial probabilities.

$$P(no\ error\ on\ 2\ tests) = (1 - \alpha)(1 - \alpha) = (1 - \alpha)^2$$

The complementary probability of this probability corresponds to the probability of making at least one false positive error in our study that these two tests constitute:

$$P(at\ less\ one\ error\ in\ the\ study) = 1 - (1 - \alpha)^2$$

And for k tests the **global α risk** is:

$$P(at\ less\ one\ error\ for\ k\ tests) = 1 - (1 - \alpha)^k$$

```
alpha <- 0.05
k <- 6
global_alpha <- 1 - (1 - alpha)^k
global_alpha
```

```
## [1] 0.2649081
```

So, if we successively do the 6 tests with, for each, an α threshold at 5%, we will actually have a global α risk of more than 26 chances out of 100 of being mistaken.

If we had chosen a 1% α threshold for each test, the global risk of error would have been:

```
alpha <- 0.01
k <- 6
global_alpha <- 1 - (1 - alpha)^k
global_alpha
```

```
## [1] 0.05851985
```

To correct this problem of multiple comparisons, be interested in Familywise Error Rate (FWER, Bonferroni Correction) and False Discovery Rate (FDR). This problem is very important in the statistical processing of fMRI, EEG or MEG data.

We are going to see that ANOVA (ANalysis Of VAriance) can be used to look for a difference across k group means as a whole. If there is a statistically significant global difference across these k means then a multiple comparison method will be used to look for specific differences between pairs of groups.

Data preparation

Before talking about ANOVA and practicing this method, we will change the example and learn some other useful functions of R. The data are those used for Reilly and Kean's "Formal Distinctiveness of High- and Low-Imaginability Nouns: Analyzes and Theoretical Implications" published in Cognitive Science in 2007. Their lab is kind enough to offer this data in the resource page of his site. The document containing all the data is an Excel file named **Reilly+Noun+Imageability+Dataset+(2013).xls**. I converted the main sheet in CSV format to make it easier to import into R.

The subject of this article is the imageability: words associated with perceptually salient, highly imageable concepts are learned earlier in life, more accurately recalled, and more rapidly named than abstract words.

Now create a new R script whose first line will start importing the .csv file into the *originalData* variable:

```
originalData <- read.csv2("imageability.csv")  # default: read.csv2(file, header =
TRUE, sep = ";", dec = ",")
colnames(originalData)

##  [1] "WORD"               "Description"        "ID"
##  [4] "block"              "BFRQ"               "CNC"
##  [7] "FAM"                "IMG"                "KFFRQ"
## [10] "NLET"               "nphon"              "NSYL"
## [13] "N.Com.Syll"         "AOA"                "NMORPH"
## [16] "Comp."              "tot.pref"           "tot.suff"
## [19] "Etymology"          "P.DENS"             "P.All"
## [22] "Bi.All.Fill"        "Stress."            "Freq_HAL"
## [25] "Log_Freq_HAL"       "I_Mean_RT"          "I_Mean_Accuracy"
## [28] "I_NMG_Mean_RT"      "I_NMG_Mean_Accuracy"
```

The *originalData* variable is a **data.frame**, an R structured object similar to a table but which can contain data of different types according to the columns. All lines must have the same length, that is, the same number of columns. Columns and lines have unique names.

The size of this data:

```
length(originalData)

## [1] 29
```

Oops! This is the size of the second dimension of our data.frame (29 columns, because R considers a data.frame as a list of vectors and length returns the vector length).

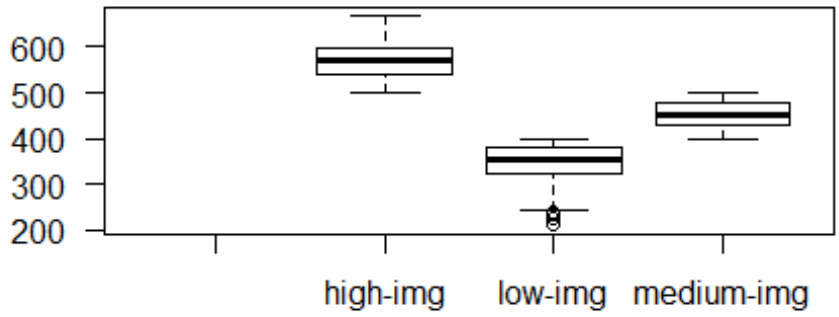
```
length(originalData[,1])

## [1] 3494

dim(originalData)

## [1] 3494   29
```

The **colnames()** function shows the names of the columns in the data.frame. This table can be seen by double clicking on the name of the variable in the top right panel. We find that there are many variables in this data. The meaning of some is not obvious even if we refer to the article and the various sheets of the Excel document.

The variables that interest us for the moment are:
* WORD
* I_NMG_Mean_RT = the mean naming latency (in msec) for a particular word.
* IMG = imageability (how easily a person can form an associated mental image).
* Etymology = origin of the word.

For my demonstration, I would like imageability to be a categorical variable. It seems to me that the column *Description* corresponds to that. Check ...

```
boxplot(IMG ~ Description, originalData, las=1)
```



That's right. Let's change the name of this second column. And select the 4 variables that interest us to put them in a new data.frame named *myData*.

```
names(originalData)[2]<-"Imageability"
selectedCol <- c("WORD", "I_NMG_Mean_RT", "Imageability", "Etymology")
myData <- originalData[, selectedCol]
# Let's see what happens:
summary(myData)
```

```
##           WORD      I_NMG_Mean_RT        Imageability    Etymology
##             : 617   Min.   : 510.9             : 617   Min.   :1.000
##  ABANDONMENT:   1   1st Qu.: 598.9   high-img  :1385   1st Qu.:1.000
##  ABDUCTION  :   1   Median : 634.3   low-img   : 636   Median :1.000
##  ABILITY    :   1   Mean   : 647.8   medium-img: 856   Mean   :1.833
##  ABODE      :   1   3rd Qu.: 682.3                     3rd Qu.:2.000
##  ABSCESS    :   1   Max.   :1070.1                     Max.   :5.000
##  (Other)    :2872   NA's   :638                        NA's   :617
```

R believes that *Etymology* is a digital datum! And its values are not very telling. Let's use the codes found in the 3rd sheet of the Excel document.

```
# recode Etymology
origins <- c("Latin","Germanic","Greek","Other","Unknown origin")
myData[,"Etymology"] <- origins[myData[,"Etymology"]]
```

```
# Etimology levels distribution:
table(myData[,"Etymology"])
```

```
##
##       Germanic          Greek          Latin          Other Unknown origin
##            907            151           1489            132            198
```

Let's see the result.

```
summary(myData)
```

```
##           WORD      I_NMG_Mean_RT        Imageability   Etymology
##             : 617   Min.   : 510.9              : 617   Length:3494
##  ABANDONMENT:   1   1st Qu.: 598.9   high-img   :1385   Class :character
##  ABDUCTION  :   1   Median : 634.3   low-img    : 636   Mode  :character
##  ABILITY    :   1   Mean   : 647.8   medium-img: 856
##  ABODE      :   1   3rd Qu.: 682.3
##  ABSCESS    :   1   Max.   :1070.1
##  (Other)    :2872   NA's   :638
```

This is not very good: there are many **missing values** (NA's) and the type of *Ethymology* is not the same as that of *Imageability*. Look at theses types:

```
str(myData)
```

```
## 'data.frame':    3494 obs. of  4 variables:
##  $ WORD         : Factor w/ 2878 levels "","ABANDONMENT",..: 2 4 7 8 13 14 15 17
18 19 ...
##  $ I_NMG_Mean_RT: num  795 580 696 630 737 ...
##  $ Imageability : Factor w/ 4 levels "","high-img",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Etymology    : chr  "Latin" "Latin" "Latin" "Latin" ...
```

Imageability is a 4-level **factor** and *Etymology* is a **string** of characters. Let's correct that and remove rows with missing values.

```
myData$Etymology <- as.factor(myData$Etymology)
myData <- myData[complete.cases(myData),]        # delete rows with NA's
summary(myData)
```

```
##           WORD      I_NMG_Mean_RT        Imageability
##  ABANDONMENT:   1   Min.   : 510.9              :   0
##  ABDUCTION  :   1   1st Qu.: 598.9   high-img   :1380
##  ABILITY    :   1   Median : 634.3   low-img    : 627
##  ABODE      :   1   Mean   : 647.8   medium-img: 849
##  ABSCESS    :   1   3rd Qu.: 682.3
##  ABSOLUTION :   1   Max.   :1070.1
##  (Other)    :2850
##           Etymology
##  Germanic      : 900
##  Greek         : 151
##  Latin         :1478
##  Other         : 132
##  Unknown origin: 195
```

```
##
##
```

How many lines have we lost due to missing values? Try to find the source of this missing data.

```
dim(originalData)[1] - dim(myData)[1]
```

```
## [1] 638
```

It remains a level in Imageability that is named an empty string and with no instance in our current data. It's annoying.

```
levels(myData$Imageability)
```

```
## [1] ""           "high-img"   "low-img"    "medium-img"
```

```
myData$Imageability <- droplevels(myData$Imageability)
levels(myData$Imageability)
```

```
## [1] "high-img"   "low-img"    "medium-img"
```

```
summary(myData)
```

```
##           WORD      I_NMG_Mean_RT        Imageability
##  ABANDONMENT:   1   Min.   : 510.9   high-img  :1380
##  ABDUCTION  :   1   1st Qu.: 598.9   low-img   : 627
##  ABILITY    :   1   Median : 634.3   medium-img: 849
##  ABODE      :   1   Mean   : 647.8
##  ABSCESS    :   1   3rd Qu.: 682.3
##  ABSOLUTION :   1   Max.   :1070.1
##  (Other)    :2850
##           Etymology
##  Germanic      : 900
##  Greek         : 151
##  Latin         :1478
##  Other         : 132
##  Unknown origin: 195
##
##
```

Ah! These are beautiful, clean data. But I would like to add a column: *I_NMG_Zscore* provides the standardized mean naming latency for a word. This metric allows the naming performance for different words to be directly compared, with more negative z-scores denoting shorter latencies.

```
# calculate standardized mean naming latency for each word and add as a column
# I_NMG_Zscore = (I_NMG_Mean_RT - mean(I_NMG_Mean_RT)) / sd(I_NMG_Mean_RT)
# the R scale() function does this for us:
myData$I_NMG_Zscore <- scale(myData$I_NMG_Mean_RT)
summary(myData)
```

```
##           WORD      I_NMG_Mean_RT        Imageability
##  ABANDONMENT:   1   Min.   : 510.9   high-img  :1380
##  ABDUCTION  :   1   1st Qu.: 598.9   low-img   : 627
##  ABILITY    :   1   Median : 634.3   medium-img: 849
##  ABODE      :   1   Mean   : 647.8
```

```
##  ABSCESS    :   1   3rd Qu.: 682.3
##  ABSOLUTION :   1   Max.   :1070.1
##  (Other)    :2850
##           Etymology      I_NMG_Zscore.V1
##  Germanic      : 900   Min.   :-1.968319
##  Greek         : 151   1st Qu.:-0.703129
##  Latin         :1478   Median :-0.194199
##  Other         : 132   Mean   : 0.000000
##  Unknown origin: 195   3rd Qu.: 0.496068
##                        Max.   : 6.072192
##
```

```
# to explain .V1 after I_NMG_Zscore
str(myData$I_NMG_Zscore)
```

```
##  num [1:2856, 1] 2.113 -0.969 0.691 -0.258 1.288 ...
##  - attr(*, "scaled:center")= num 648
##  - attr(*, "scaled:scale")= num 69.6
```

```
mean(myData$I_NMG_Mean_RT)
```

```
## [1] 647.7621
```

```
sd(myData$I_NMG_Mean_RT)
```

```
## [1] 69.5528
```

Analysis of variance

Principles

The reason this type of analysis is called analysis of variance and not analysis of multi-group means is that its approach is based on the comparison of estimated variances.

We will try to confirm, thanks to this analysis of variance, that the words denoting the most imageable concepts are more quickly named than the more abstract words.

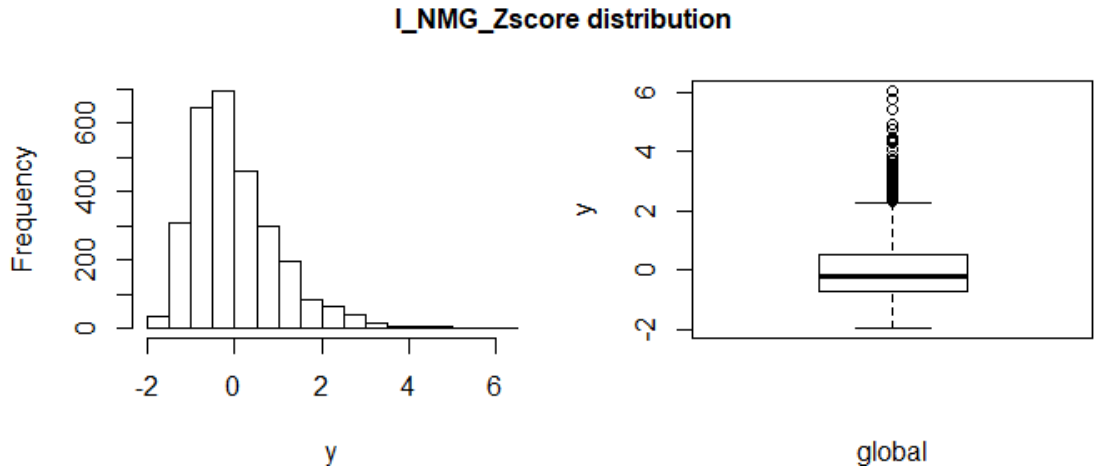
The variable I_NMG_Zscore takes values y_i which can be represented by the global mean of these values (μ) increased (or decreased) by a number ϵ_i representing the variability of this variable around its mean, different for each value of y_i. What we can **model** by the following equation:

$$y_i = \mu + \epsilon_i$$

In models, ϵ_i is often called the **sampling error**.

```
# par() function is used to set or query graphical parameters
par0 <- par(no.readonly = TRUE) # backup the whole list of settable default
parameters.
par(mfrow=c(1,2), oma=c(0,0,2,0), mar=c(4,4,1,1))
hist(myData$I_NMG_Zscore, xlab="y", main="")
```

```
boxplot(myData$I_NMG_Zscore, ylab="y", xlab="global")
title("I_NMG_Zscore distribution", outer=TRUE, cex.main=1)
```



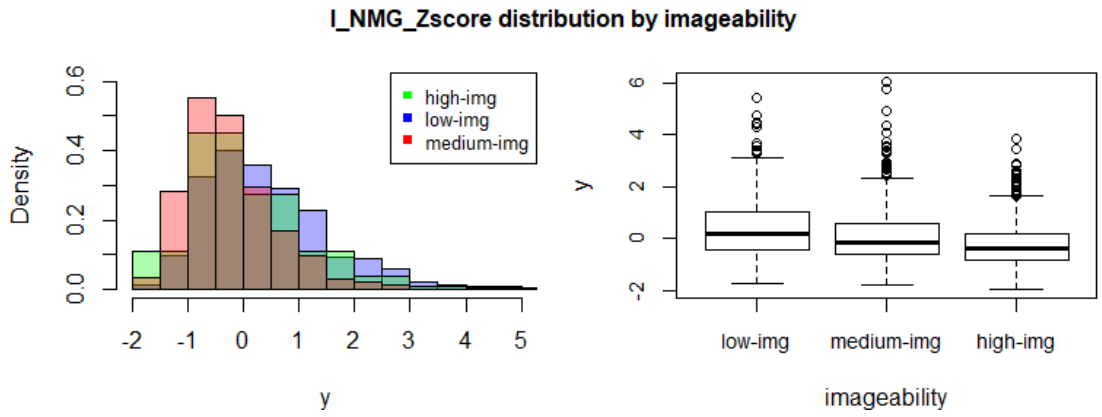

```
par(par0)                    # reset graphical parameters
```

If we take into account imageability because we suspect that it has an effect on naming speed, we assume that the three imageability groups have different means for this speed. The model then becomes:

$$y_{ji} = \mu_j + \epsilon_{ji}$$

with the index *j* indicating the imageability group.

```
par(mfrow=c(1,2), oma=c(0,0,2,0), mar=c(4,4,1,1))
myData.low <- myData[myData$Imageability=="low-img",]
myData.medium <- myData[myData$Imageability=="medium-img",]
myData.high <- myData[myData$Imageability=="high-img",]
hist(myData.low$I_NMG_Zscore, prob=TRUE, xlim=c(-2,5), ylim=c(0,0.6), xlab="y",
main="", col=rgb(0,0,1,0.33))
hist(myData.medium$I_NMG_Zscore, prob= TRUE, col=rgb(0,1,0,0.33), add=TRUE)
hist(myData.high$I_NMG_Zscore, prob= TRUE, col=rgb(1,0,0,0.33), add=TRUE)
legend("topright",legend=levels(myData$Imageability), pch=15,
col=c("green","blue","red"), cex=0.8)
# reorder the levels of the Imageability variable
myData$Imageability <- ordered(myData$Imageability, levels = c("low-img", "medium-
img", "high-img"))
boxplot(I_NMG_Zscore ~ Imageability, data=myData, ylab="y", xlab="imageability",
cex.axis=0.9)
title("I_NMG_Zscore distribution by imageability", outer=TRUE, cex.main=1)
```




```
par(par0)                          # reset graphical parameters
```

To highlight the terms responsible for the variability, we can break down the means μ_j into global mean μ and variation due to group membership τ_j (the imageability **effect**).

$$y_{ji} = \mu + \tau_j + \epsilon_{ji}$$

From this equation, knowing that for a given population the mean is a constant, we can write:

$$global\ variance = variance\ explained\ by\ groups + sampling\ error$$

Remember how to calculate the variance estimate of a population from an observed sample:

$$s^2 = \frac{\sum_{i=1}^{n}(y_i - \overline{y})^2}{n-1}$$

The quantity in the numerator is called the *sum of squares*. It is the sum of the squares of the deviations of all the observations y_i from their mean $\overline{y}$. In the context of ANOVA, this quantity is called the **total sum of squares** (abbreviated SS_T) because it relates to the total variance of the observations. Thus:

$$SS_T = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

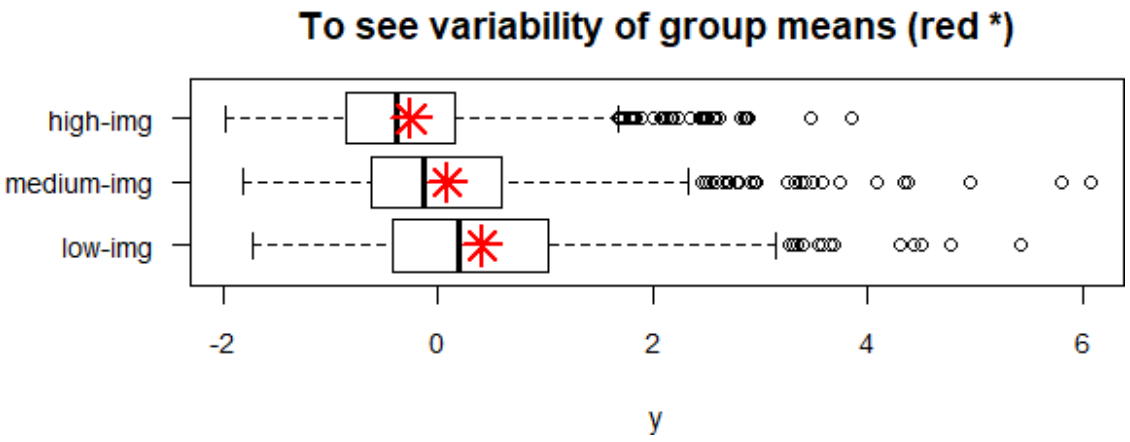
The ANOVA is based on the fact that two estimates of the population variance can be obtained from the sample data:

- one is sensitive to imageability effect and sampling error, it's the between groups estimate (**between sums of squares,** SS_B):

$$SS_B = \sum_{j=1}^{k}\sum_{i=1}^{n_j}(\overline{y_j} - \overline{y})^2$$

```
par(mar=c(4,5,3,0))
boxplot(I_NMG_Zscore ~ Imageability, data=myData, xlab="y", cex.axis=0.9,
```

```
horizontal=TRUE, las=1, main="To see variability of group means (red *)")
# get the group means
meansGrp <- by(as.numeric(myData$I_NMG_Zscore), myData$Imageability, mean)
points(y=1:3, x=meansGrp, pch = 8, lwd=2, cex = 2, col = "red")
```




```
par(par0)                          # reset graphical parameters
```

- and the other to sampling error, it's the within groups estimate (**within sums of squares,** SS_W):

$$SS_W = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (y_{ij} - \overline{y_j})^2$$

So, we understand the **sums of squares decomposition formula**:

$$SS_{total} = SS_{between} + SS_{within}$$

Let's check by the calculations:

```
n <- dim(myData)[1]
meanTotal <- 0    # by definition of z-score!
SST <- 0
for (i in 1:n){
  # square of difference between values and overall mean
  SST <- SST + (myData$I_NMG_Zscore[i] - meanTotal)^2
}
SST

## [1] 2855

# using aggregate() function with formula
nByGroup <- aggregate(I_NMG_Zscore ~ Imageability, data=myData, length)
nByGroup

##   Imageability   V1
## 1      low-img  627
## 2   medium-img  849
## 3     high-img 1380
```

```r
# we've done this in another way, with by()
meanByGroup <- aggregate(I_NMG_Zscore ~ Imageability, data=myData, mean)
meanByGroup
```

```
##   Imageability          V1
## 1      low-img  0.41644404
## 2   medium-img  0.09333869
## 3     high-img -0.24663403
```

```r
k <- length(levels(myData$Imageability))
SSB <- 0
for (j in 1:k){
  for (i in 1:nByGroup[j,2]) {
    # square of difference between group means and overall mean
    SSB <- SSB + (meanByGroup[j,2] - meanTotal)^2
  }
}
SSB
```

```
## [1] 200.0776
```

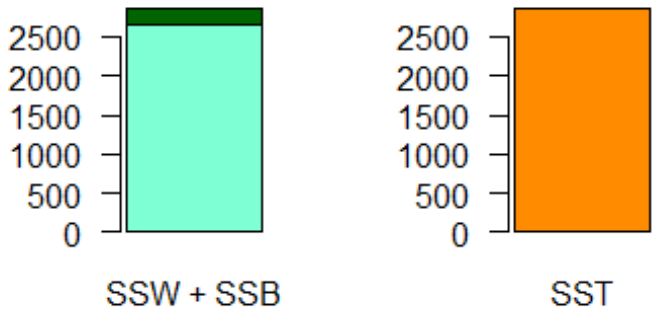
```r
SSW <- 0
for (j in 1:k){
  for (i in 1:nByGroup[j,2]) {
    group <- nByGroup[j,1]
    # square of difference between values and means of groups
    SSW <- SSW + (myData[myData$Imageability==group,"I_NMG_Zscore"][i] -
meanByGroup[j,2])^2
  }
}
SSW
```

```
## [1] 2654.922
```

```r
SSB + SSW
```

```
## [1] 2855
```

```r
par(mfrow=c(1,2))
barplot(as.matrix(c(SSW,SSB)), beside=FALSE, names.arg="SSW + SSB",
col=c("aquamarine","darkgreen"), las=1)
barplot(SST, names.arg="SST", col="darkorange", las=1)
```

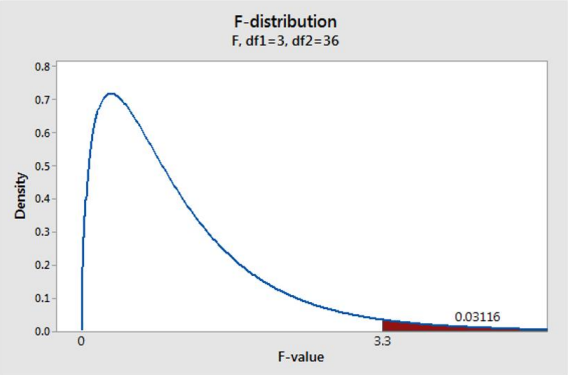


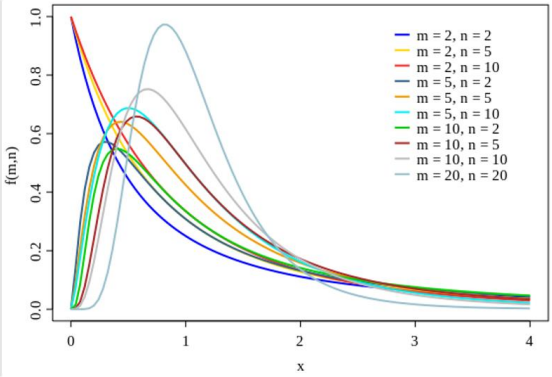
```
par(par0)
```

To obtain each variance estimate, we calculate the **mean square** (**MS**) by dividing the sum of the squares by the appropriate number of degrees of freedom: $n-1$ for the total variance, $k-1$ for the variance between the groups, and $n-k$ for the variance within groups.

In the null hypothesis (H_0) where there is no difference between the means of the 3 (sub)populations from which the 3 samples are drawn, the ratio between the between variance and the within variance follows a known distribution law, the Fisher-Snedecor's F law, with $\nu_1 = k-1$ and $\nu_2 = n-k$ degrees of freedom.

$$F = \frac{s^2_{between}}{s^2_{within}}$$





Since the distribution of F is known, a table or the R function **pf()** gives us the probability that a value of this ratio exceeds a given threshold.

```
MSB <- SSB / (k-1)
MSW <- SSW / (n-k)
f_statistic <- MSB / MSW
f_statistic
```

```
## [1] 107.5024
```

```
pf(f_statistic, k-1, n-k, lower.tail = FALSE)
```

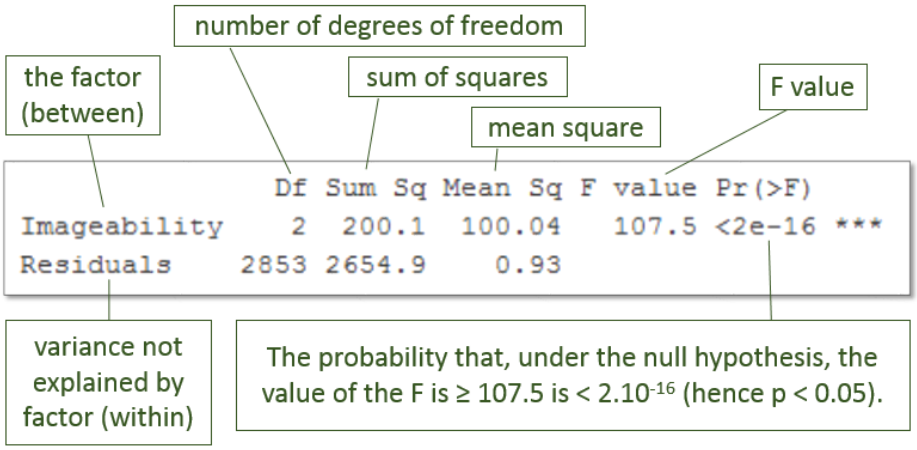
```
## [1] 9.723528e-46
```

Application

Fortunately, R offers us a function (and even several) that does all these calculations for us:

```
myAnova <- aov(I_NMG_Zscore ~ Imageability, data=myData)
summary(myAnova)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Imageability    2  200.1  100.04   107.5 <2e-16 ***
## Residuals    2853 2654.9    0.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



So we find that the imageability has a significant effect (p <5%) on the speed of naming. What we wanted to confirm.

There are actually a lot of things in the result of this analysis. This is often the case with the R functions. We shall see later one of the interests of this profusion of information.

```
str(myAnova)
```

```
## List of 13
##  $ coefficients : Named num [1:3] 0.08772 -0.46887 -0.00689
##   ..- attr(*, "names")= chr [1:3] "(Intercept)" "Imageability.L"
"Imageability.Q"
##  $ residuals    : atomic [1:2856] 1.696 -1.385 0.275 -0.674 0.871 ...
##   ..- attr(*, "scaled:center")= num 648
##   ..- attr(*, "scaled:scale")= num 69.6
##  $ effects      : atomic [1:2856] -2.31e-14 -1.41e+01 2.01e-01 -6.70e-01 8.75e-
01 ...
##   ..- attr(*, "scaled:center")= num 648
##   ..- attr(*, "scaled:scale")= num 69.6
##  $ rank         : int 3
##  $ fitted.values: atomic [1:2856] 0.416 0.416 0.416 0.416 0.416 ...
##   ..- attr(*, "scaled:center")= num 648
##   ..- attr(*, "scaled:scale")= num 69.6
##  $ assign       : int [1:3] 0 1 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:2856, 1:3] -53.4416 0.0187 0.0187 0.0187 0.0187 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2856] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:3] "(Intercept)" "Imageability.L" "Imageability.Q"
##   .. ..- attr(*, "assign")= int [1:3] 0 1 1
##   .. ..- attr(*, "contrasts")=List of 1
##   .. .. ..$ Imageability: chr "contr.poly"
##   ..$ qraux: num [1:3] 1.02 1.03 1.02
##   ..$ pivot: int [1:3] 1 2 3
##   ..$ tol  : num 1e-07
##   ..$ rank : int 3
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 2853
##  $ contrasts    :List of 1
```

```
##   ..$ Imageability: chr "contr.poly"
##  $ xlevels      :List of 1
##   ..$ Imageability: chr [1:3] "low-img" "medium-img" "high-img"
##  $ call         : language aov(formula = I_NMG_Zscore ~ Imageability, data =
myData)
##  $ terms        :Classes 'terms', 'formula'  language I_NMG_Zscore ~
Imageability
##   .. ..- attr(*, "variables")= language list(I_NMG_Zscore, Imageability)
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "I_NMG_Zscore" "Imageability"
##   .. .. .. ..$ : chr "Imageability"
##   .. ..- attr(*, "term.labels")= chr "Imageability"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(I_NMG_Zscore, Imageability)
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "nmatrix.1" "ordered"
##   .. .. ..- attr(*, "names")= chr [1:2] "I_NMG_Zscore" "Imageability"
##  $ model        :'data.frame':   2856 obs. of  2 variables:
##   ..$ I_NMG_Zscore: num [1:2856, 1] 2.113 -0.969 0.691 -0.258 1.288 ...
##   .. ..- attr(*, "scaled:center")= num 648
##   .. ..- attr(*, "scaled:scale")= num 69.6
##   ..$ Imageability: Ord.factor w/ 3 levels "low-img"<"medium-img"<..: 1 1 1 1 1
1 1 1 1 1 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language I_NMG_Zscore ~
Imageability
##   .. .. ..- attr(*, "variables")= language list(I_NMG_Zscore, Imageability)
##   .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:2] "I_NMG_Zscore" "Imageability"
##   .. .. .. .. ..$ : chr "Imageability"
##   .. .. ..- attr(*, "term.labels")= chr "Imageability"
##   .. .. ..- attr(*, "order")= int 1
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(I_NMG_Zscore, Imageability)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "nmatrix.1" "ordered"
##   .. .. .. ..- attr(*, "names")= chr [1:2] "I_NMG_Zscore" "Imageability"
##  - attr(*, "class")= chr [1:2] "aov" "lm"
```

What are the levels of imageability that differ?

Several tests of comparisons of the levels (modalities) of a factor exist. We will use the **Tukey's HSD test** (Honestly Significative Difference) which is a test called *post-hoc* because it is done after doing the global analysis (*omnibus test*).

We can apply this test on a factor if:
* the conditions of application of ANOVA are verified (see that later),

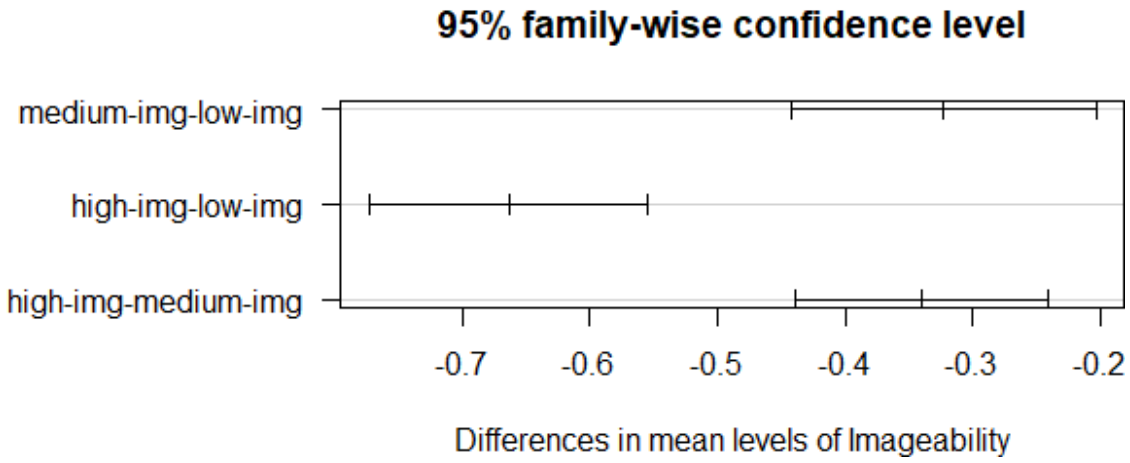
* the factor has a *fixed effect*, with at least 3 modalities (see that later),
* the factor has a significant effect on the response (which explains the term *post-hoc*).

For k groups there are k*(k-1)/2 possible pairwise comparisons, that is 3 in our case.

```
TukeyHSD(myAnova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = I_NMG_Zscore ~ Imageability, data = myData)
##
## $Imageability
##                             diff        lwr        upr p adj
## medium-img-low-img  -0.3231054 -0.4422186 -0.2039922     0
## high-img-low-img    -0.6630781 -0.7720224 -0.5541337     0
## high-img-medium-img -0.3399727 -0.4386384 -0.2413071     0
```

```
par(mar=c(4,9,3,0))
plot(TukeyHSD(myAnova), las=1)
```




```
par(par0)
```

The conditions of application

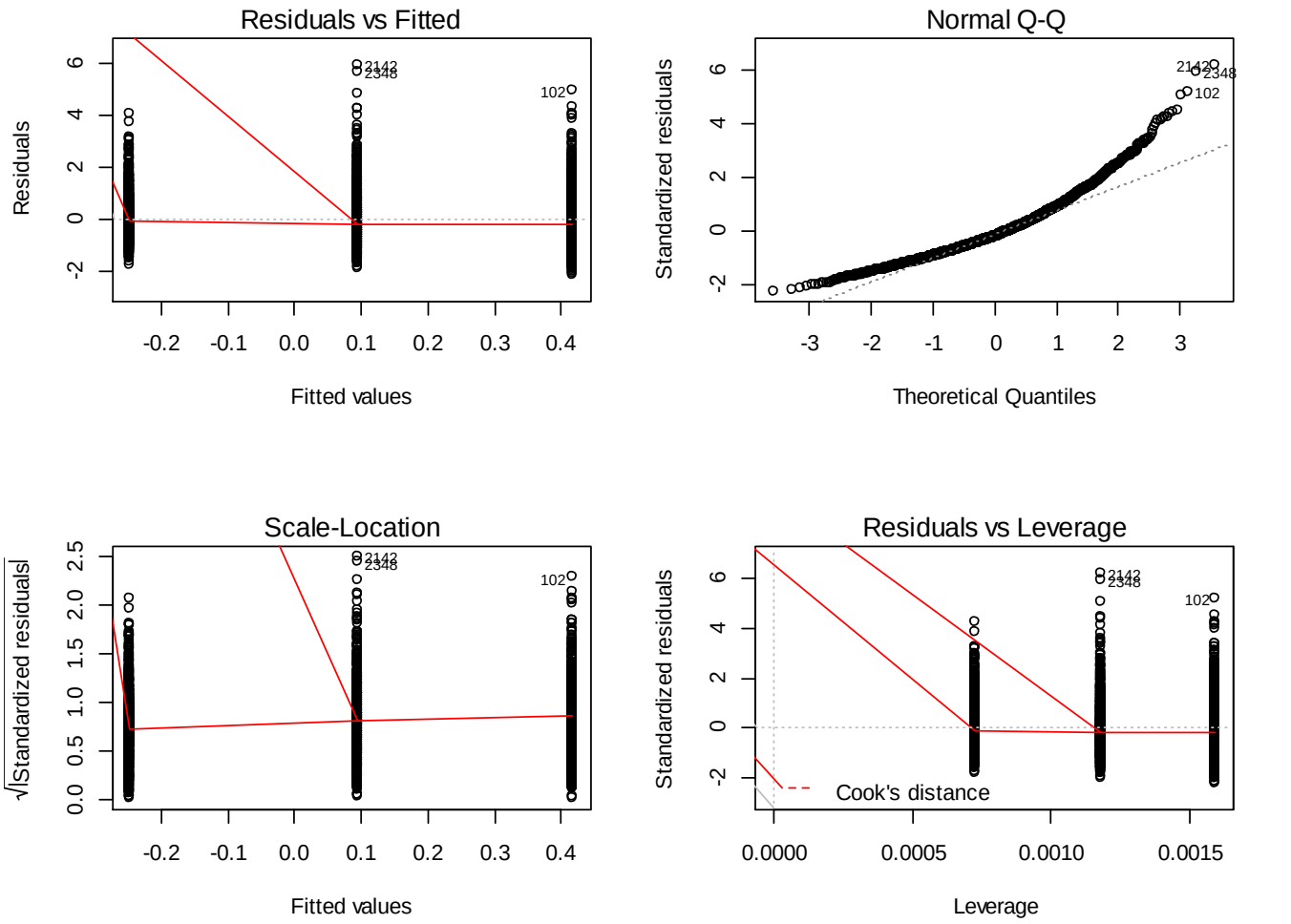
The use of the F law distribution is valid only under certain conditions:
* the groups (subpopulations defined by the factor levels) must be independent and data within each group also,
* the residuals must follow a Normal distribution,
* the group variances must be homogeneous or, better, equal (homoscedasticity).

Independence: the data must not be autocorrelated, one of the factors must not be calculated from another factor, there must not be repeated measurements for the same subject (otherwise we must change the model, see below), etc.

Some tests, such as the Shapiro-Wilk test, the Bartlett test or the Levene's test, may be useful for checking the other conditions. But the ANOVA is a robust method, beware of the rigor of these tests. In practice, it is better to take a look at some graphics first.

Here is one of the interests of the results of the ANOVA by the R function aov(): a simple call to the function plot() is used to check the application conditions of the ANOVA.

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




```
par(par0)
```

The first plot and the plot above show that there is no evident relationships between residuals and fitted values (the mean of each groups), which is good. So, we can assume the homogeneity of variances. The condition is not respected if the variability between the 3 groups seems very different or if the red line is moving away from the horizontal, which signs a relation between the residues and the predicted values. In the scale-location plot, above, the absolute value transforms all the residuals into a magnitude scale (removing direction) and the square-root helps you see differences in variability more accurately.

The Normal Q-Q plot of residuals is used to check the assumption that the residuals are normally distributed. It should approximately follow the reference dotted line. Outliers, skew, heavy and light-tailed aspects of distributions (all violations of normality) will show up in this plot.

The fourth graph, "Residuals vs. Leverage", is harder to interpret. See this link.

The parametric ANOVA F-test is more resistant to violations of the assumptions of the normality and equal variance assumptions if the design is *balanced*, that is, if group sizes are close enough.

If the conditions of application are not satisfied

If these conditions are not respected, the distribution of the MS_B/MS_W ratio in the null hypothesis moves away from the F law. In this case, it is possible to apply transformations on the dependant variable (log for example), or to use a *non-parametric* ANOVA (Kruskal-Wallis test), or to perform an ANOVA based on *permutation tests*.

Kruskal-Wallis test

Kruskal-Wallis test by rank is a non-parametric alternative to one-way (one factor) ANOVA test. It's recommended when the assumptions of ANOVA are not met.

A non parametric test (or *distribution free test*) does not assume anything about the underlying distribution (for example, that the data comes from a normal distribution). But it has less power than its parametric analogue.

```
myNonParamAnova <- kruskal.test(I_NMG_Zscore ~ Imageability, data=myData)
myNonParamAnova   # oddly summary() does not show what we want to see

##
##  Kruskal-Wallis rank sum test
##
## data:  I_NMG_Zscore by Imageability
## Kruskal-Wallis chi-squared = 188.62, df = 2, p-value < 2.2e-16
```

Permutation tests

In the null hypothesis, there is no difference between the means of the 3 imageability groups. Any naming speed values could be in any of these groups, it should not change its mean. After calculating the statistic MS_B/MS_W on our initial data, we will redo the calculation 1000 times (or more) by randomly moving some values from one group to another each time. It does not matter how these 1000 statistics are distributed. The question we will ask will be: what percentage of these 1000 values exceed the first calculated statistic (the one that interests us)? If it is less than 5%, we say that the test is significant at an α risk of 5%.

Here are some pages to help you understand the permutation tests and find the R functions that will make it easier for you to practice:
* Introduction in R: https://thomasleeper.com/Rcourse/Tutorials/permutationtests.html
* A good drawing is better than a long speech: https://www.ohbmbrainmappingblog.com/blog/a-brief-overview-of-permutation-testing-with-examples
* A R function for a permutation version of ANOVA: https://statmethods.wordpress.com/2012/05/21/permutation-tests-in-r/
* A more detailed explanation and more information about R functions relating to permutation tests, but in French: https://statistique-et-logiciel-r.com/tests-de-permutation-avec-le-logiciel-r/

And a paper to read before embarking on the statistical processing of fMRI, PET, MEG or EEG data: Nonparametric Permutation Tests For Functional Neuroimaging: A Primer with Examples

Two-way ANOVA

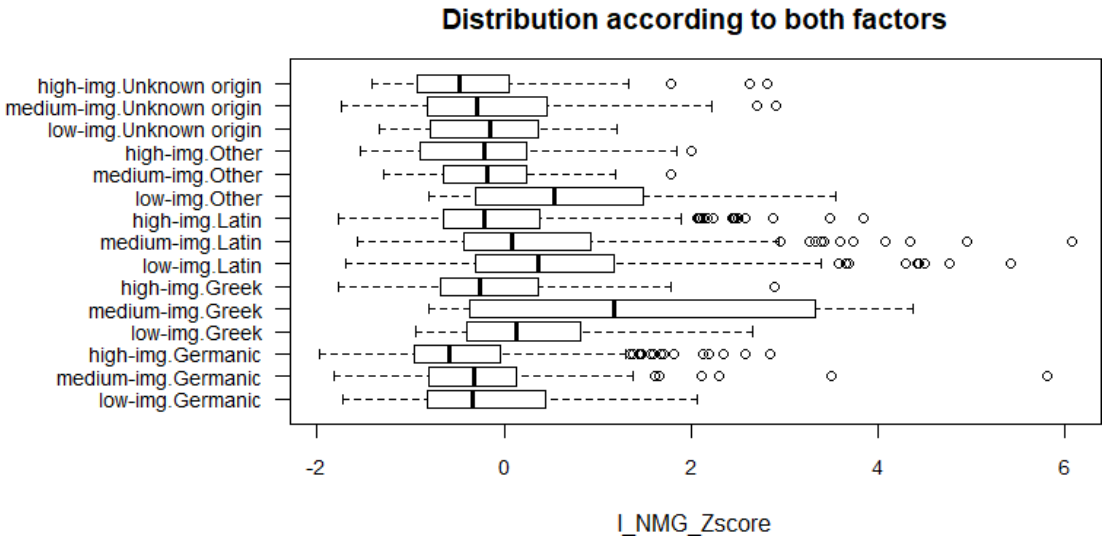
So far we have been working on an one-way ANOVA, where only one independent categorical variable, the factor, affects a dependent quantitative variable. Let's say a few words about the two-way ANOVA with the factors *Imageability* and *Etymology* of our example.

```
myAnova2 <- aov(I_NMG_Zscore ~ Imageability + Etymology, data=myData)
summary(myAnova2)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Imageability    2  200.1  100.04  113.16 <2e-16 ***
## Etymology       4  136.3   34.07   38.54 <2e-16 ***
## Residuals    2849 2518.6    0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The following boxplot taking into account the two factors suggests that depending on the origin of the words, the imageability does not quite have the same effect on the speed of naming. We talk about **interaction** between the two factors.

```
par(mar=c(4,11,3,0))
boxplot(I_NMG_Zscore ~ Imageability + Etymology, data=myData,  xlab="I_NMG_Zscore",
cex.axis=0.9, horizontal=TRUE, las=1, main="Distribution according to both
factors")
```




```
par(par0)
```

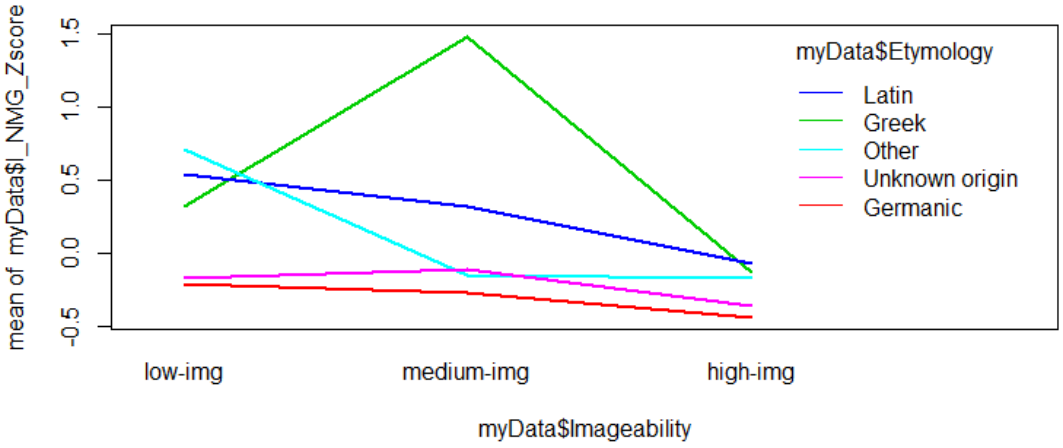
We can test the significance of this interaction by modifying the formula ("*" rather than "+" between the two factors in the formula):

```
myAnova2 <- aov(I_NMG_Zscore ~ Imageability * Etymology, data=myData)
summary(myAnova2)
```

```
##                          Df Sum Sq Mean Sq F value  Pr(>F)
## Imageability              2  200.1  100.04 113.853 < 2e-16 ***
## Etymology                 4  136.3   34.07  38.779 < 2e-16 ***
```

```
## Imageability:Etymology    8   22.3    2.79   3.178 0.00136 **
## Residuals              2841 2496.3    0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interaction.plot(myData$Imageability, myData$Etymology, myData$I_NMG_Zscore, lty=1,
col=2:6, lwd=2)
```



The plot of the interaction shows that moderately imageable Greek words are the most difficult to name and that moderately imageable words from other origins are as easy to name as highly imageable words.

This two-way ANOVA calculate **main effects** and an **interaction effect**. For the interaction effect, all factors are considered at the same time. And the calculation of the main effects seems similar to that of a one-way ANOVA: the effect of each factor seems to be considered separately. If it is true, the order of the factors in the formula does not matter. Check.

```
myAnova2bis <- aov(I_NMG_Zscore ~ Etymology * Imageability, data=myData)
summary(myAnova2bis)
```

```
##                          Df Sum Sq Mean Sq F value  Pr(>F)
## Etymology                 4  229.4   57.36  65.277 < 2e-16 ***
## Imageability              2  106.9   53.47  60.856 < 2e-16 ***
## Etymology:Imageability    8   22.3    2.79   3.178 0.00136 **
## Residuals              2841 2496.3    0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If the final conclusion given by the small p is the same, the intermediate results (sum square, mean square and F value) are not identical. This phenomenon is due to the fact that the size of different combinations of groups are not equal (the numbers are *not balanced*).

To be convinced of this, let us randomly draw a subsample of our sample in such a way that each combination of the two factors receives the same number of observations. Let's see what are the actual numbers of these combinations.

```
table(myData[,3:4])
```

```
##             Etymology
## Imageability Germanic Greek Latin Other Unknown origin
##   low-img          78    33   478    14             24
##   medium-img      254     4   479    21             91
##   high-img        568   114   521    97             80
```

The smallest subgroup (n=4) is that of the Greek words of medium imageability.

```
imgLevels <- levels(myData[,3])
etymLevels <- levels(myData[,4])

myBalancedData <- myData[0,]
for (i in imgLevels){
  for (j in etymLevels){
    selection <- myData[(myData$Imageability==i)&(myData$Etymology==j),]
    selectionSample4 <- selection[sample(nrow(selection), 4),]
    myBalancedData <- rbind(myBalancedData, selectionSample4)
  }
}

table(myBalancedData[,3:4])
```

```
##             Etymology
## Imageability Germanic Greek Latin Other Unknown origin
##   low-img           4     4     4     4              4
##   medium-img        4     4     4     4              4
##   high-img          4     4     4     4              4
```

```
myBalancedAnova <- aov(I_NMG_Zscore ~ Imageability * Etymology,
data=myBalancedData)
summary(myBalancedAnova)
```

```
##                       Df Sum Sq Mean Sq F value Pr(>F)
## Imageability           2   5.42  2.7085   2.908 0.0649 .
## Etymology              4   9.65  2.4114   2.589 0.0493 *
## Imageability:Etymology 8   7.31  0.9142   0.982 0.4627
## Residuals             45  41.91  0.9312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
myBalancedAnova2 <- aov(I_NMG_Zscore ~ Etymology * Imageability,
data=myBalancedData)
summary(myBalancedAnova2)
```

```
##                       Df Sum Sq Mean Sq F value Pr(>F)
## Etymology              4   9.65  2.4114   2.589 0.0493 *
## Imageability           2   5.42  2.7085   2.908 0.0649 .
## Etymology:Imageability 8   7.31  0.9142   0.982 0.4627
## Residuals             45  41.91  0.9312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because of the small size of each subgroup, the results are obviously not the same as before. The power of the test has dropped considerably. But what interests us is that the order of the factors has no more impact on the results.

When data is unbalanced, there are different ways to calculate the sums of squares for ANOVA. There are at least 3 approaches, commonly called Type I, II and III sums of squares. Which type to use has led to an ongoing controversy in the field of statistics. When data is balanced, the factors are *orthogonal*, and types I, II and III all give the same results. The anova and aov functions in R implement type I ANOVA by default (sequential sum of squares) but it is relatively simple to obtain the sum of squares of type II and a little more complicated to obtain the type III in R.

To learn more about these types of unbalanced ANOVA:
* Anova – Type I/II/III SS explained
* Type I, II, and III Sums of Squares

Exercices

- Look on Visualizing a One-Way ANOVA how the author takes advantage of the vector processing capabilities inherent to R to simplify calculations of sums of squares.

- Evaluate the normality of the distribution of values in the different groups using histograms and the Kolmogorov-Smirnov test.

- Can you find the residuals for doing histogram in each of the above analyzes?

- Check the other application conditions in the different analyzes provided above using graphs and tests. Look on the Internet for the right tests.

Brain break

The Ten Commandments for a well-formatted Excel database or in French.

"The most important aspect of a statistical analysis is not what you do with the data, it's what data you use." - *Andrew Gelman*

"If you torture the data long enough, it will confess to anything." - *Ronald Coase*