R Programming

Descriptive Statistics with R

École d'Ingénierie Digitale
et d'Intelligence Artificielle

الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

Importing data

We are interested here in basic concepts in statistics and in the description of data. It is possible to load data in several formats (.csv, .txt) or to define them by hand.

1. **Importing data from a csv file with the read.csv function**

There are several parameters to specify:

- Name of the file
- Presence of the *header*
- Type of seperator *sep*
- Specify the decimal ("." ou ","): *dec*

In [ ]: *# import a file*

data <- **read.csv**("nom_du_fichier.csv", header = **TRUE**, sep = ";",dec = ".")

Importing data

	height	weight	group
A	185	82	M
B	194	90	M
C	165	55	F
D	175	65	F
E	172	68	F
G	150	45	F
H	165	64	M

2. Defining data (by hand)

- In an array format
- In a vectorial format

In [19]: *# Declare a dataframe with data*

```
data = data.frame(height
=c(185,194,165,175,172,150,165),
weight=c(82,90,55,65,68,45,64),
group=c("M","M","F","F","F","F","M"),
row.names=c("A","B","C","D","E","G","H"))
```

 # Display Data

```
data
```

Importing data

In [17]: *# defining data as a vector*

```
X=c(14,18,40,43,45,112)
Y=c(280,350,470,500,560,1200)
```

The notation *c()* in R allows to define a vector, i.e. a list of values

3. Useful tips

To access a particular column in a dataframe, you can use the $ sign and the column name. This trick can be used in many situations with R objects. It will be used a lot during linear regression calculations.

In [21]: data$height

1. 185 2. 194 3. 165 4. 175 5. 172 6. 150 7. 165

Data vizualisation

1. Scatter Plot (using plot() function)

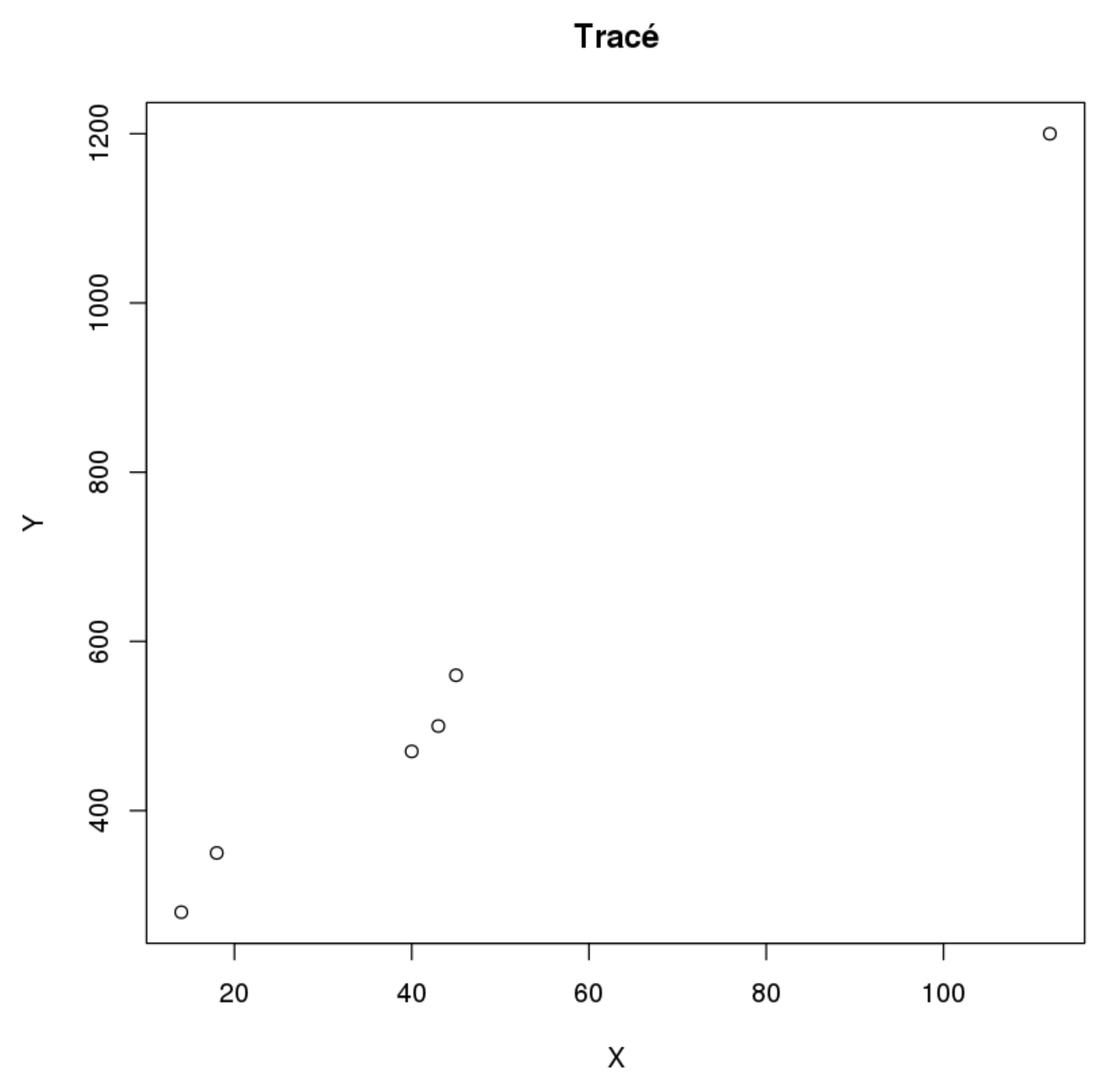
- *xlab* (*ylab*) is an optional parameter that defines a name for the x-axis (of the ordata)
- *main* defines the name of the graph

In [39]: *# Simple Scatter Plot of X and Y data*

plot(X, Y, xlab ="X" ,ylab = "Y", main = "Tracé")

To illustrate the commands seen in this script, I used the data « Notes ».

In [25]: Notes <- **read.csv** ("Notes.csv", header = TRUE, sep = ";", dec = ".")



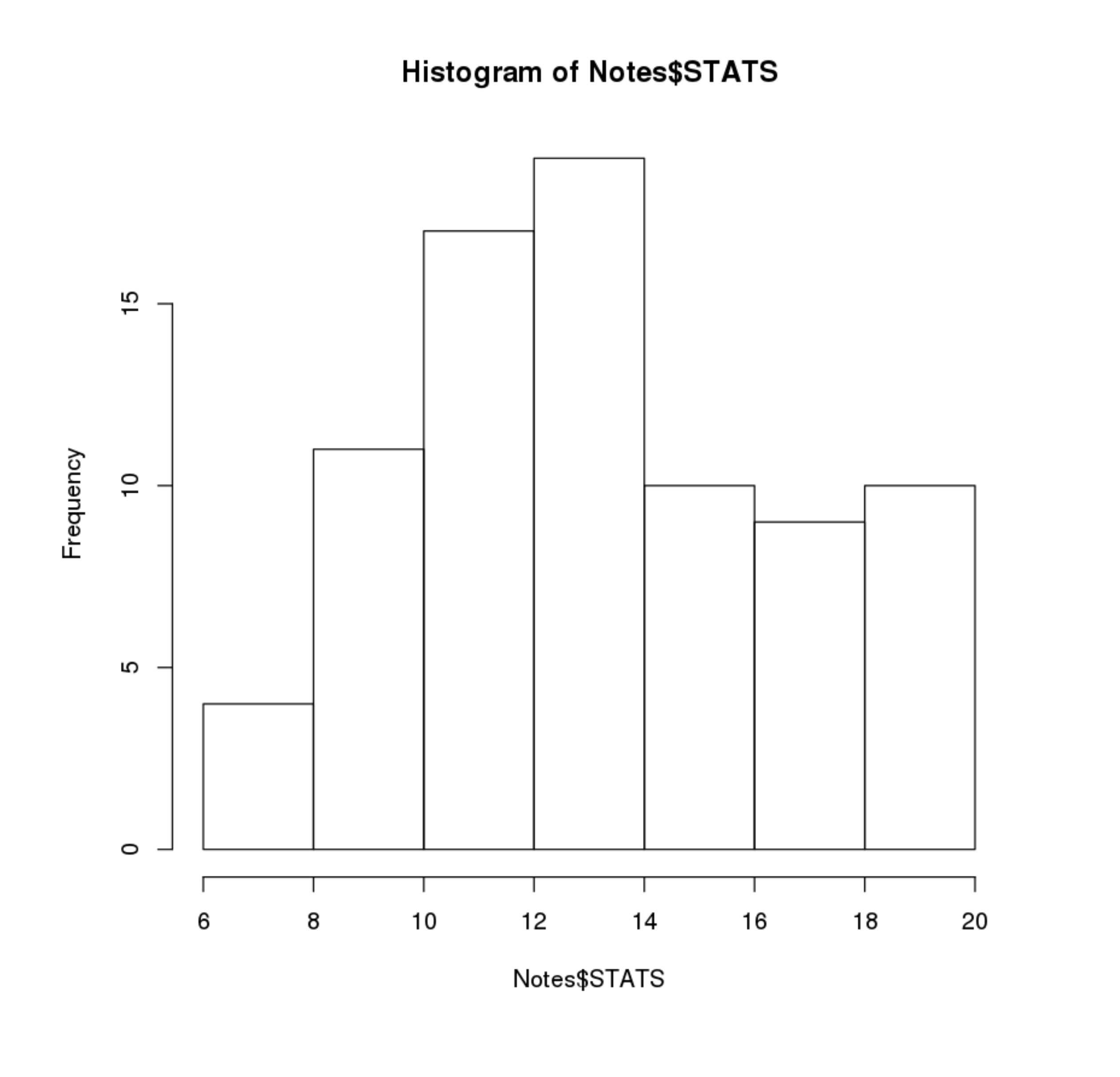
Data vizualisation

2. Histograms (using hist() function)

2.1. Drawing an histogram

In [26]: # simple histogram of a column of the data table

hist(Notes $ STATS)



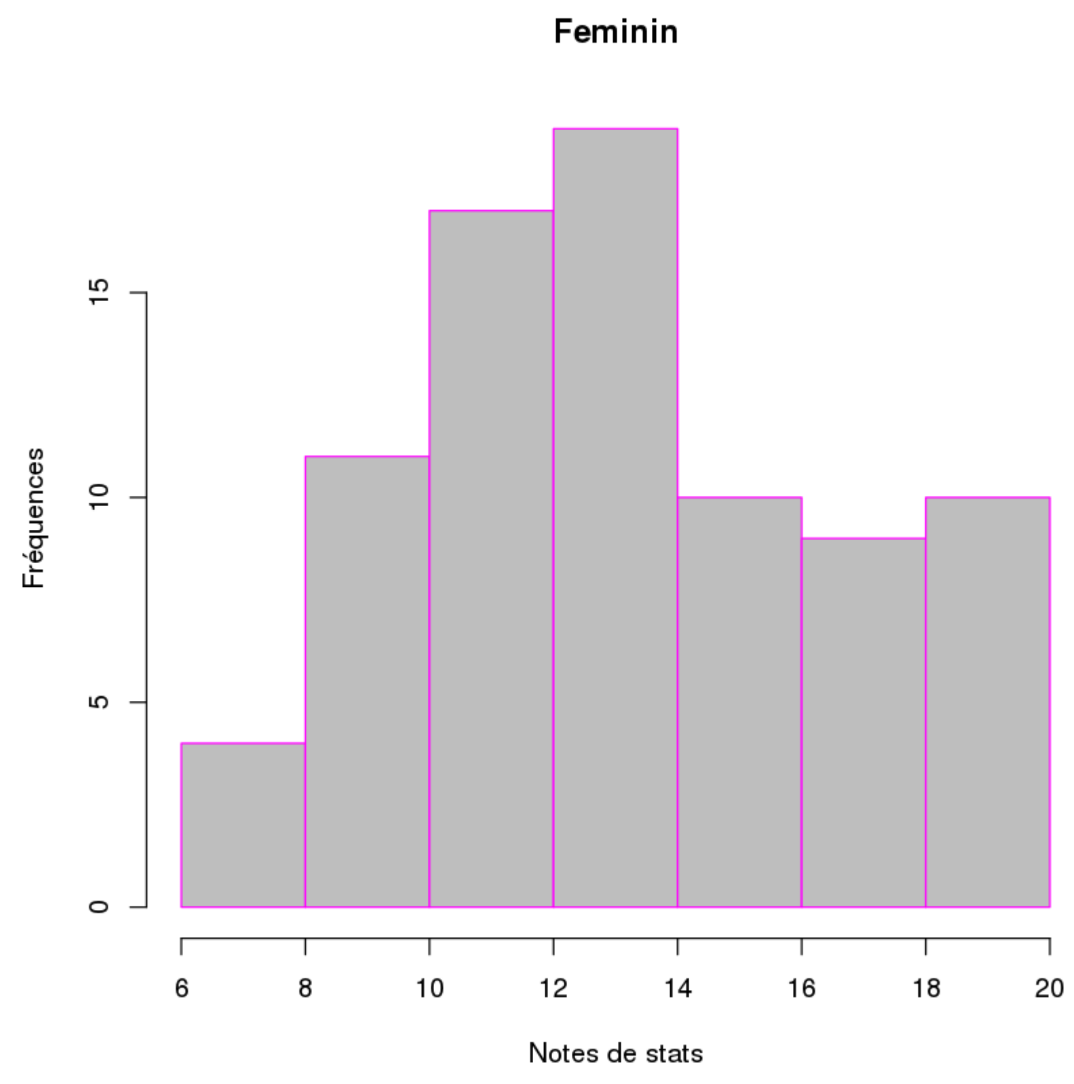
Data vizualisation

2. Histograms (using hist() function)

2.1. Drawing an histogram

In [27]: *# Histogram of the same data as above with additional display parameters*

hist(Notes$STATS, col="grey",border="magenta", main=**paste**("Feminin"), xlab="Notes de stats",ylab="Frequencies")



Data vizualisation

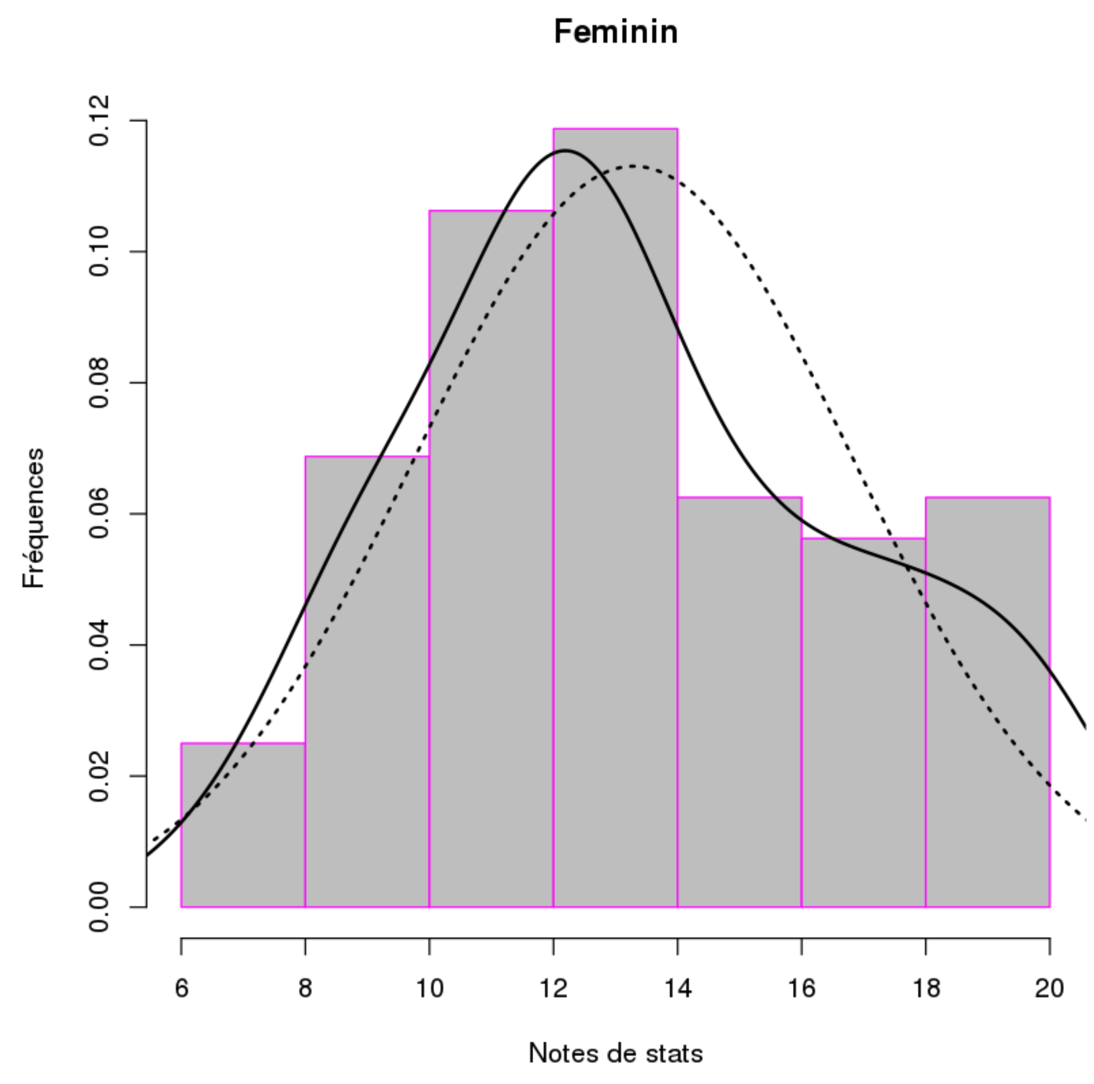
2. Histograms (using hist() function)

2.1. Drawing an histogram

We can add to the histogram the density curve of a normal distribution estimated for the data. When using hist () function, the freq parameter must be set to FALSE (freq = FALSE)

For the above curves, the parameters have been added:

- *lwd* which refers to the thickness of the curve
- *lty* which refers to the type of curve: dotted line, line, …



Data vizualisation

2. Histograms (using hist() function)

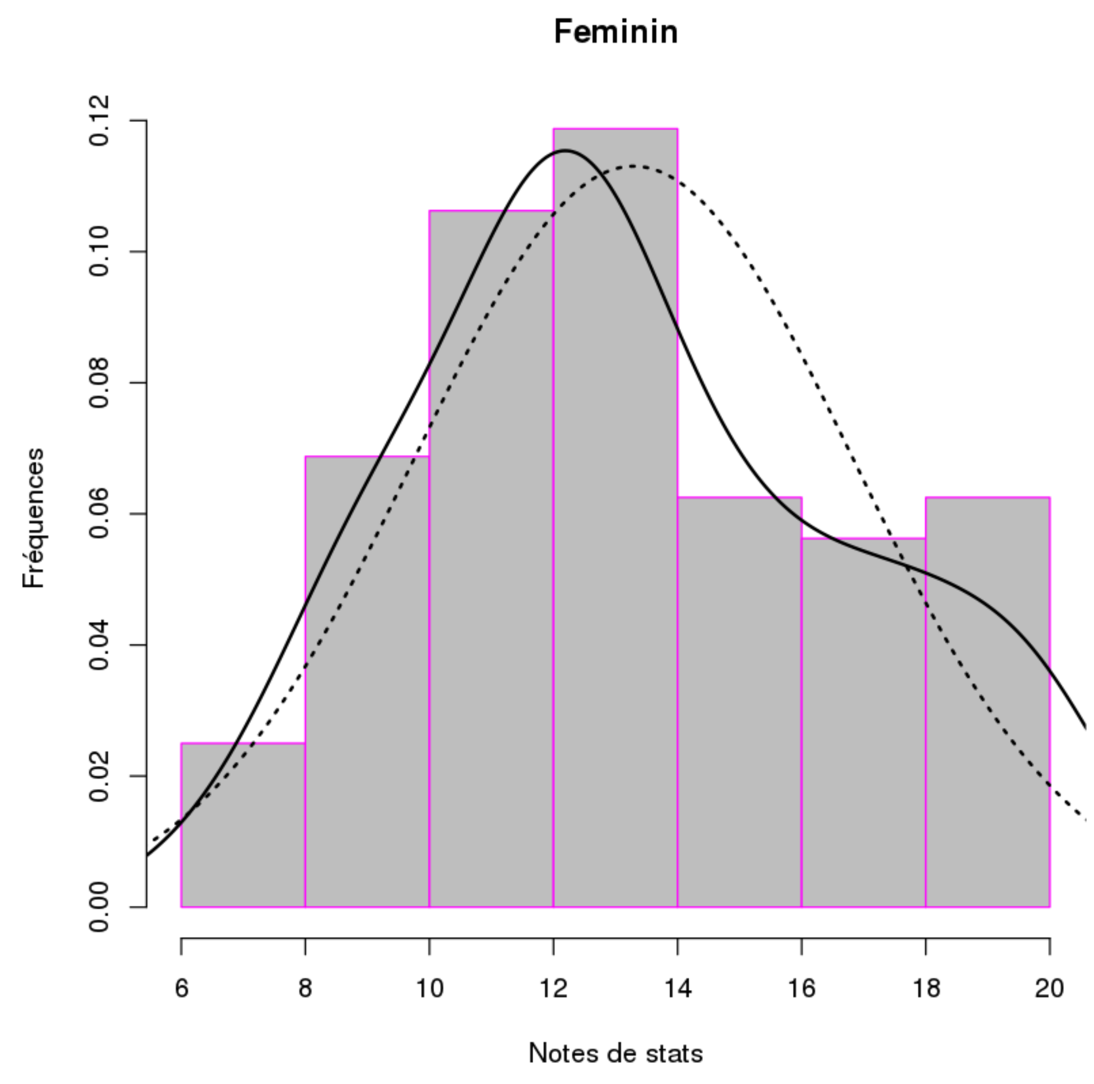
2.1. Drawing an histogram

In [33]: **hist**(Notes$STATS, col="grey",border="magenta", main=**paste**("Feminin"), xlab="Notes de stats",ylab="Fréquences", freq=**FALSE**)

lines(density(Notes$STATS), lwd=2) *# adjusted density*

x = **seq**(5,21,length.out=500)
 # density of a normal distribution

lines(x, dnorm(x, **mean**(Notes$STATS), sd(Notes$STATS)), lwd=2, lty=3)



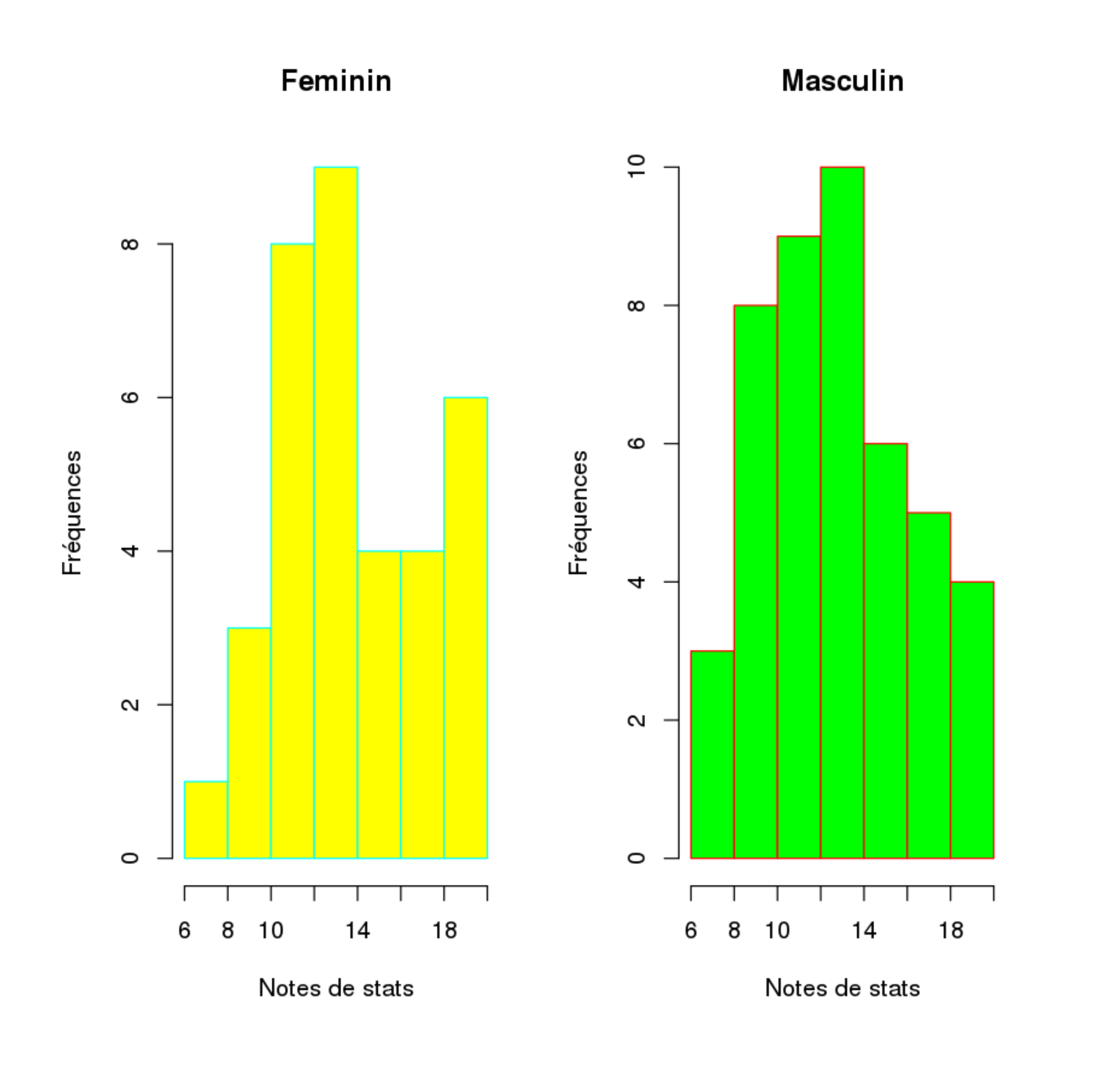
Data vizualisation

<u>2. Histograms (using hist() function)</u>

2.2. Drawing two juxtaposed histograms

In [28]: **layout**(**matrix**(1:2,1,2)) *# allows to divise the graphical output into 2 areas*

hist(Notes$STATS[Notes$SEXE=="F"], col="yellow",border="cyan", main=**paste**("Feminin"),xlab="Notes de stats",ylab="Frequencies")
hist(Notes$STATS[Notes$SEXE=="M"], col="green",border="red", main=**paste**("Male"),xlab="Notes de stats",ylab="Frequencies")

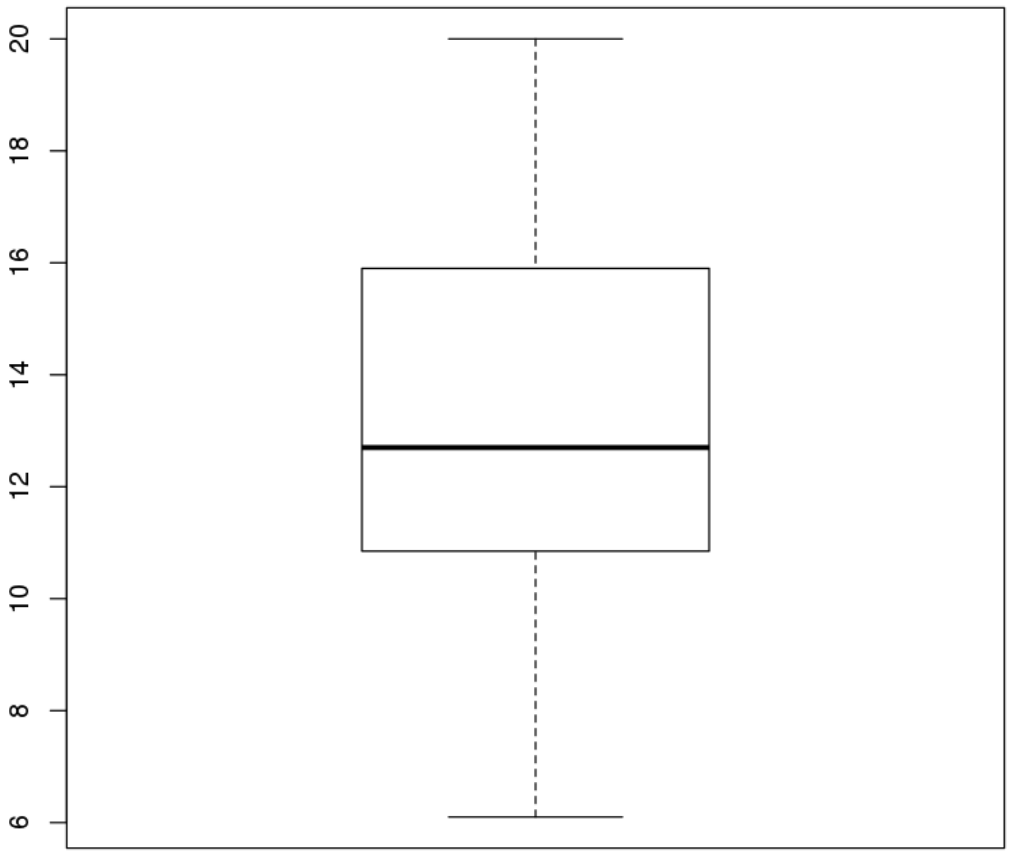


Data vizualisation

3. Tukey diagram: the boxplot () function

In [34]: *# Tukey diagram of statistics scores*

```
boxplot(Notes$STATS)
```



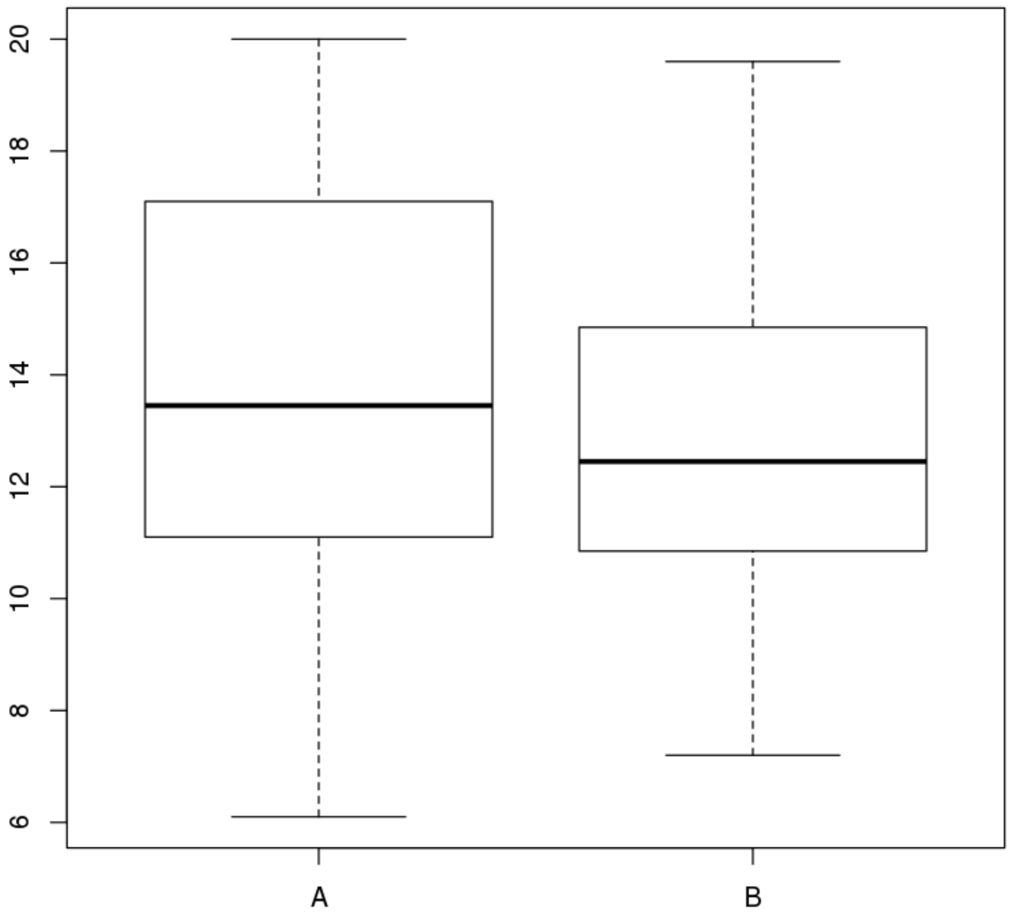
Data vizualisation

3. Tukey diagram: the boxplot () function

If the data can be partitioned according to a column, as it is the case here by group (A or B) or by sex (M or F), then we can draw the Tukey diagram for each of the values of the partition.

In [35]: *# Tukey diagram for two different groups in the data*

```
boxplot(STATS~GROUP, data=Notes)
```

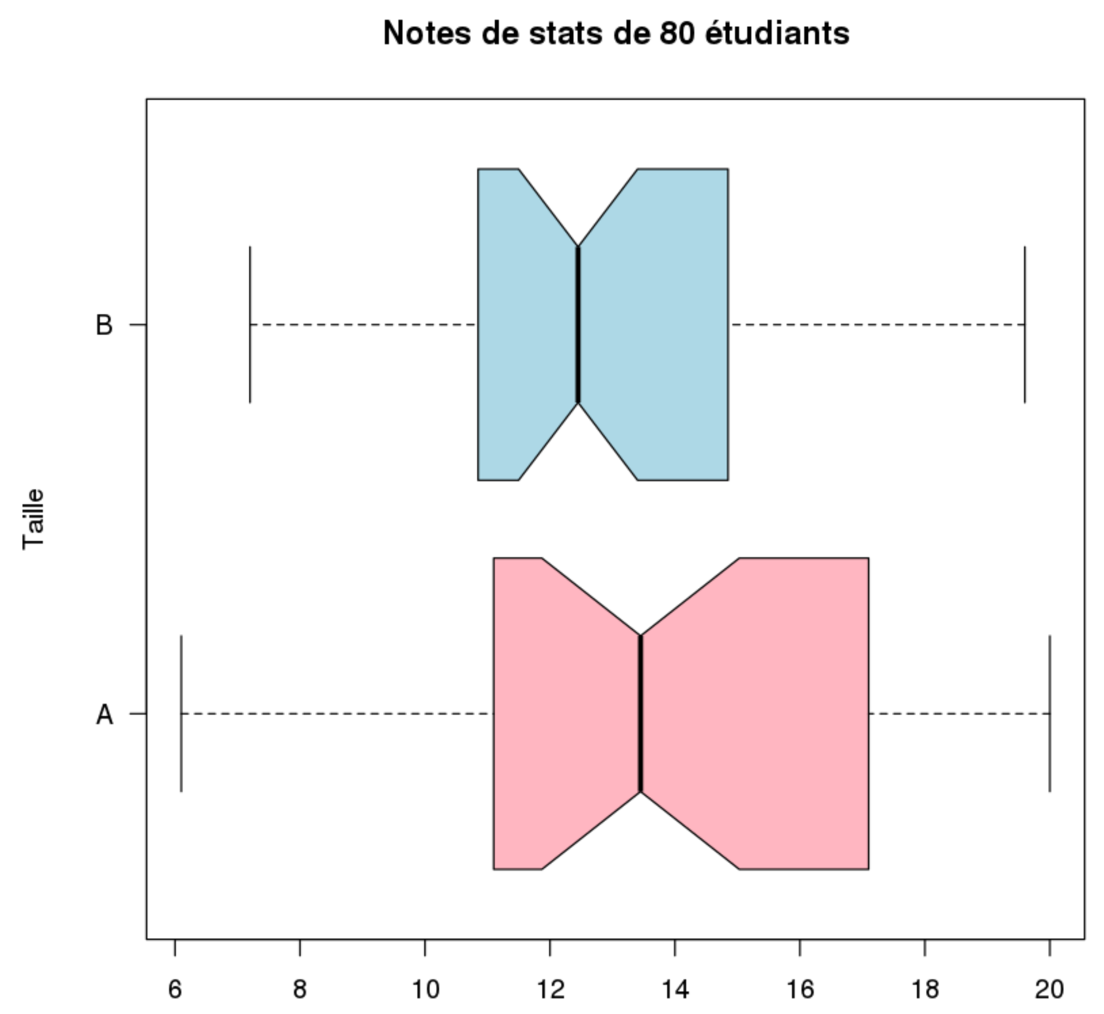


Data vizualisation

3. Tukey diagram: the boxplot () function

In [36]: *# You can modify the display of the diagrams to add color, a title,…*

```
boxplot(Notes$STATS~Notes$GROUP,
col=c("lightpink","lightblue"),
horizontal=TRUE,
notch=TRUE,
main=paste("Notes de stats de",nrow(Notes),"étudiants"),
ylab="Taille", las=1)
```



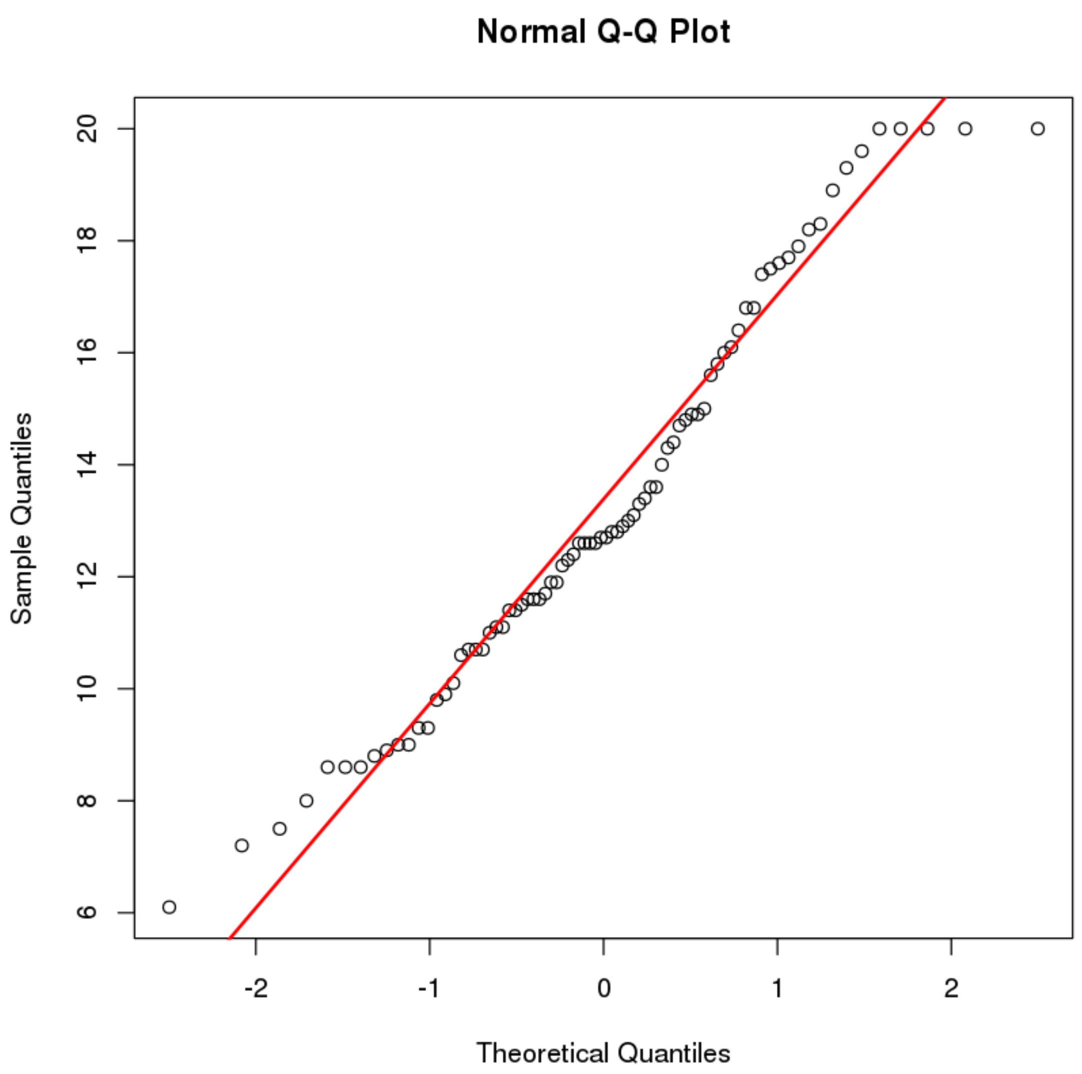
Data vizualisation

<u>4. The quantile-quantile diagram: qqnorm () function</u>

*Q*uantile-quantile diagram of Stats scores

In [42]: **qqnorm**(Notes$STATS)

qqline(Notes$STATS, col="red", lwd=2) *# straight line of the quantiles of the normal Dist.*

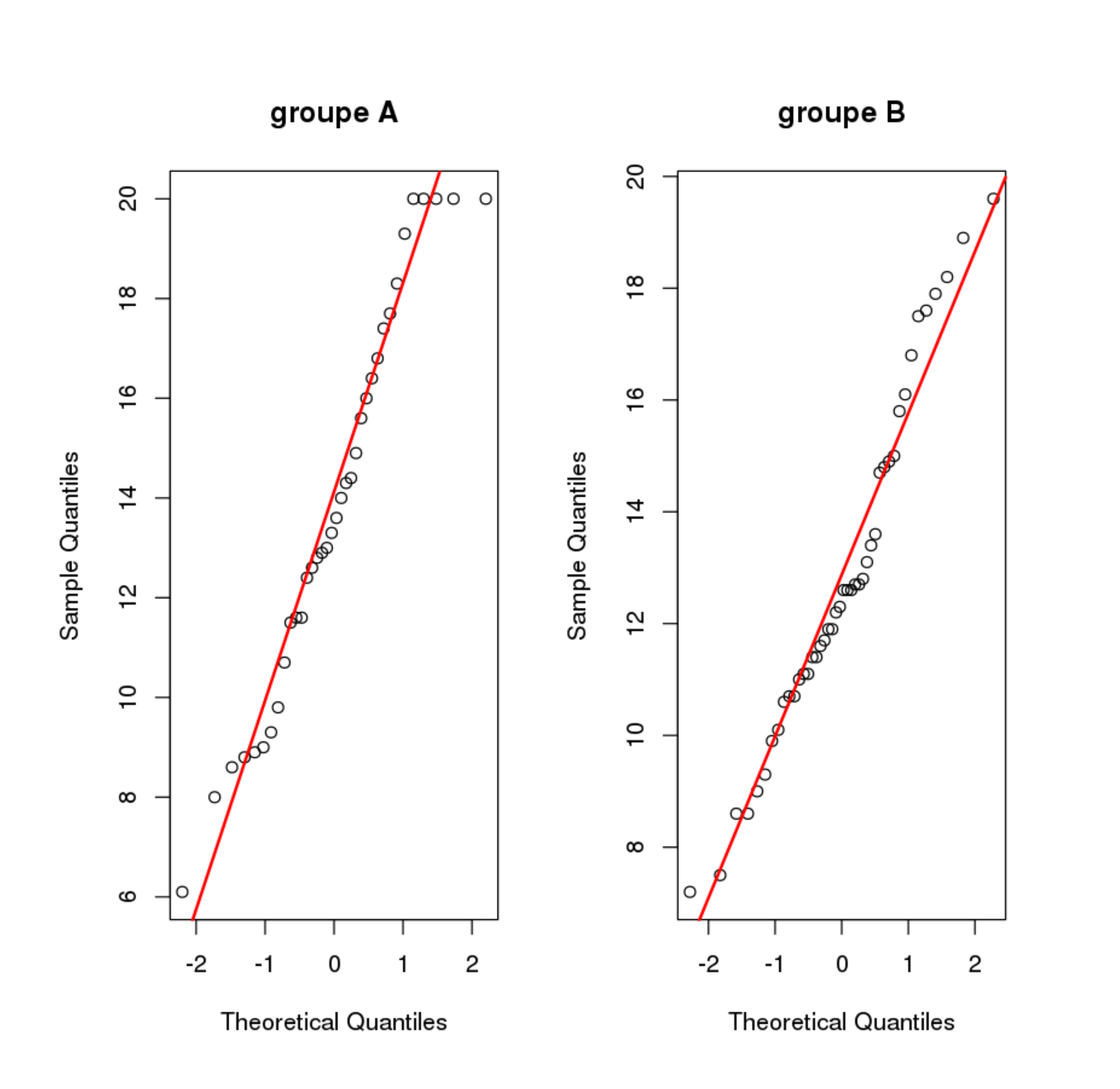


Data vizualisation

4. The quantile-quantile diagram: qqnorm () function

When we can partition the data on the values of a column:

```
In [41]: layout(matrix(1:2,1,2))
# divide the graphic output in two
# group A
qqnorm(Notes$STATS[Notes$GROUP=="A"], main="groupe A")
qqline(Notes$STATS[Notes$GROUP=="A"], col="red", lwd=2)
# groupe B
qqnorm(Notes$STATS[Notes$GROUP=="B"], main="groupe B")
qqline(Notes$STATS[Notes$GROUP=="B"], col="red", lwd=2)
```



Measurements on Defined Positions

1- Calculate the Mean

In [48]: *# Mean of stats scores*

```
mean(Notes$STATS)

# Mean of stats scores of group A
mean(Notes$STATS[Notes$GROUP=="A"])
13.29125
13.8777777777778
```

Measurements on Defined Positions

2- Calculate the Median

In [50]: *# The median of Stats scores*

```
median(Notes$STATS)


# The median of Stats scores of groupe A
median(Notes$STATS[Notes$GROUP=="A"])
```

12.7

13.45

Dispersion Measurements

1- Calculate quartiles

In [53]: *# Calculate des quartiles des notes de stats*

```
quantile(Notes$STATS)
```

0\% 6.1 **25\%** 10.925 **50\%** 12.7 **75\%** 15.85 **100\%** 20

2- Calculate standard deviation and variance

In [54]: *# Standard deviation of Stats scores*

```
sd(Notes$STATS)
# Variance of Stats scores
var(Notes$STATS)
```

3.52959257339671

12.4580237341772

Coefficient of Asymmetry and Coefficient of Flattening

Need a new library ("moments") to calculate the coefficient of asymmetry and flattening.

In [56]: install.packages("moments")

```
library(moments)
```

In [57]: *# skewness of Stats scores*

```
skewness(Notes$STATS)


# flattening (kurtosis) Stats scores
kurtosis(Notes$STATS) -3
```

0.300407083885415

-0.711856247025102

Correlation Coefficient

The *cor ()* command calculates the correlation coefficient between two data vectors. So we get the correlation between economics and statistics scores:

In [59]: *#Calculate the covariance between Stats and Economics scores*

cov(Notes$STATS, Notes$ECONO)

 # Calculate of the Pearson correlation coefficient between the stats scores and the eco scores

cor(Notes$STATS, Notes$ECONO)

6.68207120253165

0.730563270676972