F. FABRI
Practical Tests (R Software)
Promotion 2019 - ISAFE

Programming with R: datasheet 2
The basics of R programming computing

I Data generation according to certain laws

The *nb* parameter corresponds to the number of observations desired. This parameter is essential regardless of the generated law. The other parameters have default values.

Law	R Command	Arguments
Binomial	*rbinom(nb,n,p)*	*n: Size and p: probability*
Chi-square	*rchisq(nb,df)*	*df* : degree of freedom
Exponential	*rexp(nb,lambda)*	*lambda* : parameter of the exponential law
Fisher	*rf(nb,df1,df2)*	*df1* et *df2* : 2 degrees of freedom of Fisher's law
Normal	*rnorm(nb,m,sd)*	*m: mean and sd: standard deviation*
Poisson	*rpois(nb,lambda)*	*lambda* : Poisson distribution parameter
Student	*rt(nb,df)*	d*f* : degree of freedom
Uniform	*runif(nb,min,max)*	*min: minimum and max: maximum*

Applications :

1) Run the *rnorm* command *(100,0,1)* or *x=rnorm(100,0,1); x*
2) Run the command *hist(rnorm(1000,0,1))*. Try other laws if you are curious!

The list of parameters exposed for graph functions is not limited …

II Histograms

Histogram of an ungrouped series

The command is hist () as we have just seen. Note that one parameter is mandatory: the series to display ... Below is a list of parameters that **can be used** for the hist () function.

Parameters	Descriptions
main	Histogram title (*main= "Mon titre"*)
breaks	Number of bars in the histogram. There are several ways to achieve this: 1) A vector giving the values of the limits of the rectangles (*breaks=c(0,5,10,15,20)*) 2) A function returning a vector (*breaks=seq(0,20,5)*) 3) A number giving the number of desired columns 4) Name of a slicing method (*breaks="sturges"*)
xlab	X-axis title (*xlab= " X-axis title "*)
ylab	Y-axis title (*ylab= " Y-axis title "*)
xlim	Delimits the abscissa limits to display the histogram (xlim = c (0.5))
ylim	Define the terminals of the orders for the efficiency of the histogramme (ylim = c (0.9))
labels	Gives class sizes (*labels=TRUE* by default they are not displayed)
col	Use colors to display the histogram (*color= "yellow"*). It will fill in the columns in

	yellow. If the density parameter is used, then only the hatched part is put in the requested color.
density	Default empty bars or hatched with a hatch frequency given by the value of density (*density=5*).
angle	It allows you to modify the angle of the bars hatching the histogram (*angle=30*). This parameter will only take effect if the density parameter is used.

Applications :

1) For this first application, we take a series of 1000 observations generated (by R) according to the reduced centered normal law.
Run the commands (one at time) and notice the differences.

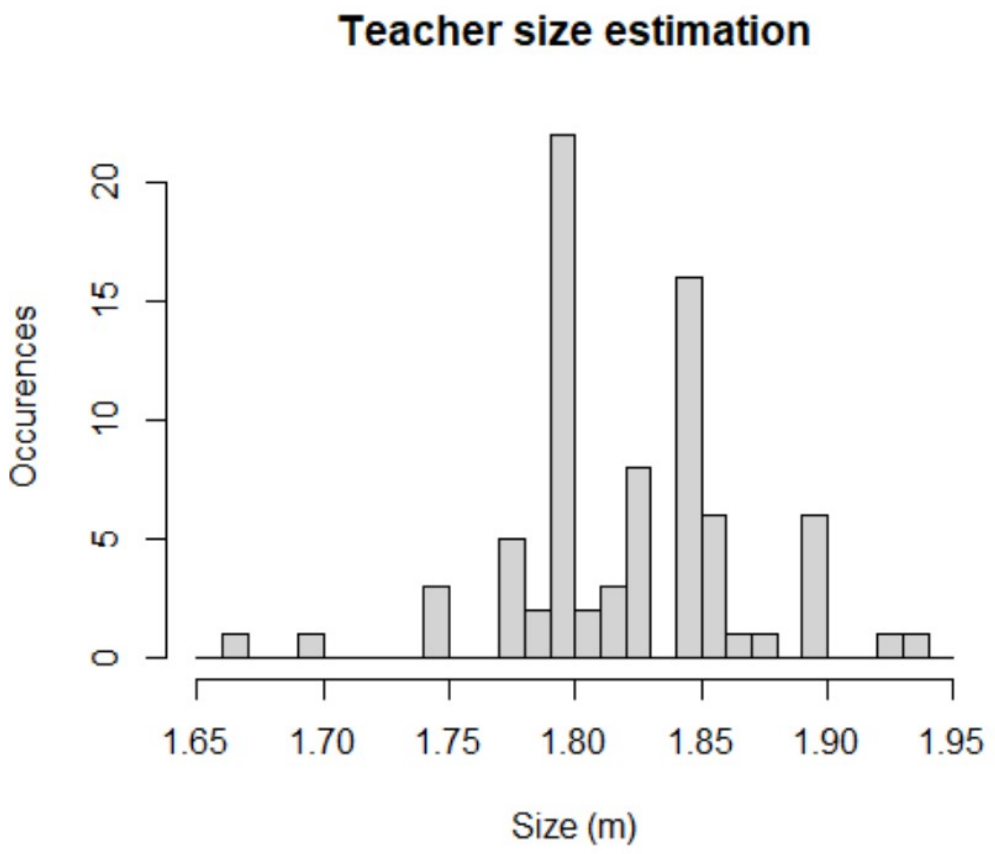
layout(matrix(1:6,3,2)) # the operation of this function is detailed at the end of the lab.
hist(rnorm(1000,0,1))
hist(rnorm(1000,0,1), xlab="X", ylab="Occurrences")
title=" Histogram of 1000 data n generated according to the law N(0,1) "
hist(rnorm(1000,0,1),main=title, xlab=" X ", ylab="Occurrences", col="blue",labels=TRUE)
hist(rnorm(1000,0,1),main=title, xlab=" X ", ylab="Occurrences", col="blue",labels=TRUE, breaks=seq(-4,4,.25))
hist(rnorm(1000,0,1),main=title, xlab=" X ", ylab="Occurrences", col="blue",labels=TRUE, breaks=6,xlim=c(-4,4))
hist(rnorm(1000,0,1),main=title, xlab=" X ", ylab="Occurrences", col="blue",labels=TRUE, breaks= seq(-4,4,.5),xlim=c(-4,4), ylim=c(0,250)) ; axis(side=1,at=seq(-4,4,1))
layout(1)

P.S :

The *axis* command (side = 1, at(seq(-4,4,1) add x-axis to the plot..
Side: an integer specifying which side of the plot the axis is to be drawn on. The axis is placed as follows: 1=below, 2=left, 3=above and 4=right.
For the y-axis we use (side=2)

2) Create the histogram of the series under R: Teacher size estimation

1.80,1.90,1.83,1.70,1.83,1.90,1.90,1.90,1.90,1.83,1.86,1.90,1.85,1.79,1.85,1.80,1.78,1.83,1.85,1.80,1.86,1.80,1.85,1.81,1.83,1.80,1.81,1.78,1.85,1.75,1.75,1.79,1.85,1.86,1.78,1.67,1.85,1.87,1.85,1.86,1.94,1.78,1.85,1.85,1.85,1.85,1.85,1.93,1.85,1.83,1.80,1.86,1.85,1.80,1.83,1.80,1.80,1.82,1.85,1.80,1.82,1.80,1.88,1.83,1.80,1.80,1.80,1.80,1.78,1.75,1.80,1.80,1.80,1.80,1.80,1.82,1.80,1.86,1.80



A) Histogram of a grouped series

Imagine that we want to plot the histogram of the following grouped series:

Size of people (m)	[1,5 ;1,7[	[1,7 ;1,8[	[1,8 ;1,9[	[1,9 ;2,0[
Classes	5	15	10	2

Just run the command

hist(c(rep(1.65,5),rep(1.75,15),rep(1.85,10),rep(1.95,2)),main= " Person size ", xlab="Size (m)",ylab=" Corrected classes",breaks=c(1.5,1.7,1.8,1.9,2), density=c(2,4,6,10), col=c("green","yellow","red","blue"),labels=paste(c(5,15,10,2)),angle=c(90,60,30,0))

P.S.: Note the command « *labels=paste(c(5,15,10,2))* » which displays of counts above the rectangles. Change with *labels = TRUE* and *labels = FALSE* to see the differences.

III Boxplots

The command for the function is boxplot (). The parameters *main*, *xlab*, *ylab*, *col*, *xlim* and ylim can be used for this type of graph (see the section on histograms).

Parameters	**Descriptions**
horizontal	By default, the graph is drawn vertically. To reverse it.
plot	By default, it draws the boxplot. If (*plot = FALSE*), then it returns the different characteristics allowing the plot of the boxplot.
outline	Outliers are not plotted if outline = FALSE (in case ...)

To juxtapose boxplots and use data stored in a *data.frame*, consider the following example. The commands to be executed are::

```
a=data.frame("N"=rnorm(1000,0,1), "E"=rexp(1000,1), "U"=runif(1000,0,1))
boxplot(a,main=" Comparison of different probability laws ")
text(3,5," Note the number \ n of outliers \ n for the exponential law ")
x11()
boxplot(subset(a,,1:2),names=c("Normal "," Exponential "),col=c("blue","red"),main="
Normal and exponential laws ",horizontal=TRUE)
```

P.S. : Note that the x11 () command keeps the last graphic plot visible.
Note the command *text* (*x, y, "text"*) which allows you to place text on the last graphic on the plot tab. The *x* parameter corresponds to the abscissa of the positioning of the text. The *y* parameter corresponds to that of the ordinate and *text* parameter is the text you want to insert.

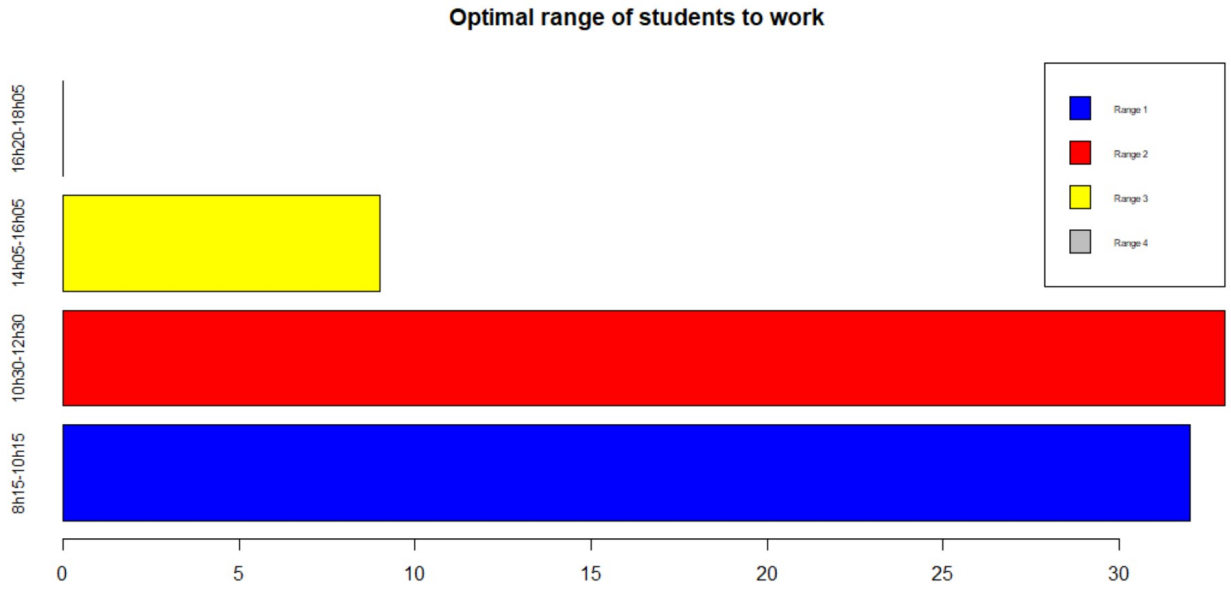
IV Barplot

The function is *barplot()*. The parameters: *main*, *xlab*, *ylab*, *xlim*, *ylim*, *density* et *angle* can be used (already seen in the histograms section).

Parameters	**Descriptions**
horiz	To place horizontal bars (*horiz=TRUE*)
names	To name the modalities (*names=c("modalitie1","modalitie2",…)*) A vector of names to be plotted below each bar or group of bars.
legend.text	To give a legend to each modality (legend.text=*c("legend1","legend2", …)*)

Applications :

1) Create the following graph (serie: 32,33,9,0)



2) Create barplot

a=matrix(c(18,19,3,0,20,14,5,2),4,2) # Definition of a matrix M(4,2)
color= c("blue","red","yellow","grey")
barplot(a,xlim=c(0,65),xlab="Classes",names=c("D1","D2"),horiz=TRUE,col=color, main="
Optimal range of students to work according to the half of the students ")
legend("right",legend=c("8h15-10h15","10h30-12h30","13h45-15h45","16h00-18h00"),
fill=color)

3) Barplot with juxtaposed bars

b=matrix(c(18,20,19,14,3,5,0,2),2,4) # Definition of a matrix M(2,4)
barplot(b,beside=TRUE,xlab=" Time slots ",ylab="Classes",legend.text=c("D1","D2"),
names=c("8h15-10h15","10h30-12h30","13h45-15h45","16h00-18h00"),col=c("blue","red"),
main=" Optimal range of students to work according to the half of the students ")

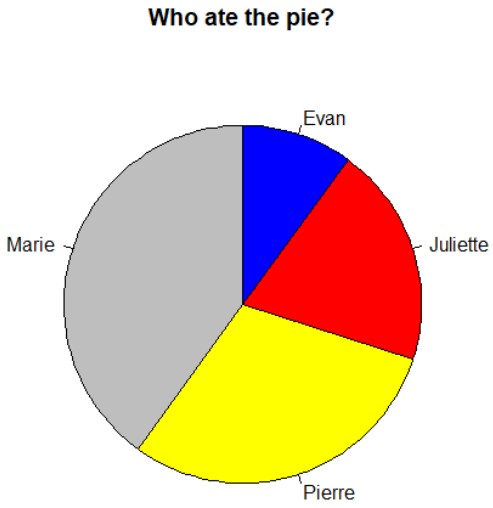
beside : a logical value. If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars.

V Pie chart

The function is *pie()*. The mandatory argument is the vector of numeric data (non-negative) that allows you to draw the circular diagram. We find the parameters *main*, *density*, *angle* et *col* that can also be used for this type of graph.

Parameters	Descriptions
clockwise	To go clockwise, you will need to lock this parameter to true (*horiz=TRUE*). By default, it is FALSE.
radius	Allows you to increase or decrease the size of the pie chart. By default it is 1. This can be useful in case the legends are too large and you need to reduce the disk. (*radius=0.5*).
init.angle	To initialize the start of the pie pieces at a given angleIt will be at 12 o'clock if it is clockwise and at 3 o'clock if it is counterclockwise by default.
labels	To give the label of the Pie chart slices.

Application: Create the following graph (data vector used c (1,2,3,4)):



VI Plot

Two variables x and y, if we want to plot points, just enter the command plot (x, y)

Example :

plot(runif(1000),runif(1000),main=" Generating evenly distributed points in a square ",xlab="Abs.",ylab="Ord.",pch=3)

There are many parameters exist for this function ... We will not detail them.

Just note the *pch* parameter : plotting 'character', i.e., symbol to use.

Symbols 1, 10, 13 and 16 use circles. The filled shapes 15:18 do not include a border.

Trendline

To add a trendline (a linear regression line for example ...) We can proceed as follows. Knowing the slope and the y-intercept of the linear regression line obtained by the least squares method (*y=pente*x+ord*). Once the point cloud has been drawn. We continue with the command *abline(a=ord,b=pente)*.

Examples :

plot(runif(1000),runif(1000),main=" Generating evenly distributed points in a square ",xlab="Abs.",ylab="Ord.")
abline(a=0.5,b=0.5)
abline(a=0.5,b=0.25) *** to add a second line.

plot(seq(5),c(4,8,6,9,7), type="l" ,*xlab="Abs.",ylab="Ord."*)
abline(a=6,b=.5)

VIII Divers

A) La fonction *layout()*

We have already saw this function in a previous example. Here is how it works:

layout(matrix(V,Nbligne,Nbcol))

The graphics window will be separated into *NbLigne*NbCol* : *NbLigne* lines and *NbCol* columns. The graphs will appear in the order given by the *V* vector.

Let's go back to the example used:

layout(matrix(1:6,3,2))

This command allows you to plot 6 graphs in the same plot tab according to the given positioning order (1:6=1,2,3,4,5,6)

Graph 1	Graph 4
Graph 2	Graph 5
Graph 3	Graph 6

To return to a graph by plots tab, run the command

layout(1)

B) Transformation of margins

We will notice, in the following code, the interest:

- to position the name of the labels in a different direction so that all the labels can be displayed (las=0 ou 1 ou 2 ou 3 ou 4)
- to change the margins (*par(mar=c(5,10,4.1,2.1))*). The parameters in *mar* apply to the bottom margins, then the left, then the top and finally the right.

```
country = c ("Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czechia",
"Denmark", "Estonia", "Finland", "France", "Germany", "Greece "," Hungary "," Ireland
"," Italy "," Latvia "," Lithuania "," Luxembourg "," Malta "," Netherlands "," Poland ","
Portugal "," Romania "," Slovakia ", "Slovenia", "Spain", "Sweden", "United Kingdom")
p_pom=c(185244,86236,44927,56570,4067,105280,20496,3648,6758,1710755,596666,28
2300,459612,21400,1921272,7464,87367,983,21,227000,2441393,329371,339570,32478,
80000,587034,22130,446400)
title = "Apple production in EU countries in 2017"
barplot(p_pom,names=country,xlab="Production (t)",main=title,horiz=TRUE) ; x11()
barplot(p_pom,names=country,las=1,horiz=TRUE,main=title,xlab="Production (t)") ;
x11()
par(mar=c(5,10,4.1,2.1))
barplot(p_pom,names=country,horiz=TRUE,main=title,las=1,xlab="Production (t)") ;
x11()
t=rank(p_pom)
names=c(); for (i in 1:28) {names=c(names,country[which(t==i)])}
par(mar=c(5,10,4.1,2.1))
barplot(sort(p_pom),names=names,las=1,main=title,horiz=TRUE,xlab="Production (t)")
```