FAST R CHECKLIST

General. For your own download, go to

`http://www.r-project.org/`

and follow the instructions there. For **R Studio**

`http://www.rstudio.com/products/rstudio/download/`

This is the program that we will follow in class, because it is customized to English. For useful guidance on **R**, you may check among others:

`http://cran.r-project.org/doc/manuals/R-intro.html`

`http://www.statmethods.net/index.html`

`https://wiki.mobilizingcs.org/rstudio`

1. Using slashes and quotation marks. Refer first to the Spanish keyboard setting. The usual Windows address specification

`c:\mydocuments\myfile.txt`

that uses the backslash `'\'`, must be replaced by

`c:\\mydocuments\\myfile.txt`

that doubles the backslash. Or by

`c:/mydocuments/myfile.txt`

that uses the standard slash `'/'`. Quotation mark `'"'` in Spanish keyboard is above key for number 2 (+ shift). It can be replaced by a single `'''` (below question mark `'?'`).

2. Working directory. For selecting the working directory, type

```
setwd('C:\\Users\\g100**\\Documents')
```

For confirmation,

```
getwd()
```

For files in the working directory,

```
list.files()
```

3. Packages. These can be treated online. However, it is useful to remember that:

```
library()                       # see all packages installed
search()                        # see packages currently loaded
install.packages('openxlsx')    # Install a package
library('openxlsx')             # Load a package
detach('package:openxlsx')      # Unload a package in session
```

As an application, we will do

```
mydata = read.xlsx('countries95.xlsx')
```

4. In case you are stuck. Do `'Esc'`.

5. Data. Type

```
data(package = .packages(all.available = T))
```

to list the data sets in *all available* packages. Type

```
data(package = 'ISLR')
```

to obtain the data sets *only* in package `'ISLR'`. The following are some instructions for managing data sets:

```
data()                                  # A collection of data
                                         sets available in R
datasetname                             # See data set
attributes(datasetname)                 # Features of a data
                                        set
names(datasetname)                      # Column names
row.names(datasetname)                  # Row names
str(datasetname)                        # Structure of a data
                                        set
dim(datasetname)                        # Dimensions
class(datasetname)                      # Class
dtfr = data.frame(datasetname)            # Create a data frame
                                          called dtfr
dtfr = read.table('mydata.txt')           # Read data in file
                                          mydata.txt to create
                                          data frame dtfr
dtfr = read.table('mydata.txt', h = T)    # As above with column
                                          names
```

```
row.names(dtfr) = Y                    # Row names changed as
                                       indicated by (char)
                                       variable Y

tolower(Y)                             # Change to lowercase
                                       uppercase names in Y

toupper(Y)                             # Change to uppercase
                                       lowercase names in Y

library('Hmisc'); capitalize(Y)        # Change to uppercase
                                       only first letter in
                                       lowercase names of Y

X = data.matrix(dtfr)                  # Create a data matrix
                                       for numerical work

X = as.matrix(dtfr)                    # Same

as.numeric(Y)                          # Convert to numeric a
                                       character variable Y

as.character(Z)                        # Convert to character
                                       a numeric variable Z

dtfr[, 1:2]                            # First two columns of
                                       dtfr

dtfr[,c(3,7)]                          # Columns 3 and 7 of
                                       dtfr

dtfr[c(1,2, 15), ]                     # Rows 1, 2, and 15 of
                                       dtfr

dtfr[c(c(1:8), 15), ]                  # Rows 1 to 8, and 15
                                       of dtfr

subset(dtfr, Y == 'Char')              # Select rows of dtfr
                                       in which character
                                       variable Y = 'Char'
```

```
subset(dtfr, Z1 == 1 & Z2 > 10)          # Select rows of dtfr
                                         in which the numerical
                                         variables Z1 = 1 and
                                         Z2 > 10

subset(dtfr, Z1 < 2 |  Z1 > 7)           # Select rows of dtfr
                                         in which the numerical
                                         variable Z1 <2 or
                                         Z1 > 7

dtfr[order(dtfr$Z), ]                    # order by increasing
                                         values of Z

dtfr[order(-dtfr$Z), ]                   # order by descending
                                         values of Z

dtfr[order(row.names(dtfr)), ]           # order alphabetically
                                         by row names

replace(dtfr[ ,2], c(1,20),'today')      # replace first 20 values
                                         in column 2 of dtfr by
                                         word 'today'

scale(dtfr)                              # Standardized data
                                         frame

scale(dtfr, scale = F)                   # Centered data frame
```

6. Complements. Other indications are given in the file

'matrices_R.txt' ,

available at Aula Global. For clearing the screen after a long computation, go to 'Edit' and 'Clear Console'. Or type

'cat('\014')' .

Last computed value is retrieved by doing

'.Last.value' .

Nesting commands is always a convenient strategy, as for example in

```
> S = round(var(dtfr), digits = 4); S
```

Or `sequencing`, as in

```
> S = var(dtfr); S = round(S, digits = 4); S
```

When doing computations, it is advisable to generally follow the order

```
> Assignment; Evaluation
```

7. Relevant ASCII codes. '|'= 124; '∼ '= 126; 'Ø'= 0216; 'ø'= 0248. '∼' can be also obtained by doing 'AltGr + 4 + Space'.

8. Logical operators.

```
<                # less than
<=               # less than or equal to
>                # greater than
>=               # greater than or equal to
==               # exactly equal to
!=               # not equal to
!x               # Not x
x | y            # x OR y
x & y            # x AND y
isTRUE(x)        # test if x is TRUE
```

9. Built-in Functions.

```
abs(x)                  # absolute value
sqrt(x)                 # square root
round(x, digits=n)      # round(3.475, digits=2) is 3.48
cos(x), ...             # cosine of x, ...
log(x)                  # natural log
log10(x)                # log in basis 10
exp(x)                  # exponential of x
```

10. Plotting. An interesting example is

```
pairs(USArrests, panel = panel.smooth, main = 'USArrests data')
```

11. Saving and loading. Do

```
save.image('myresults.RData')
```

to save your results: data frames, lists, values, ..., etc. in a file called `'myresults.RData'`, that will be located at the current working directory. For reading it back into the Global Environment , do

```
load('myresults.RData') .
```

Or simply open it from any of the two upper panes of **R** studio.

12. Help. Help of various types can be obtained by typing

```
help.start()            # General ideas
help(mean)              # Help on the mean
help(datasetname)       # Help on a data set
```