Lecture 1

Daniel Ballesteros-Chávez

October 4, 2021

Contents

1 Introduction to R

- What is R?

R is an open source statistical software.

- How to get it?

You can get R directly on their website `https://www.r-project.org/`

- What is RStudio?

RStudio is an integrated development environment (IDE) for R.

It is a good tool to get started if you are not used to scripting programming. It has very helpful visual features.

I use emacs and it is difficult for me to get used to Rstudio. My advice: use whatever you find more comfortable.

- Basic commands
  - getwd()
  - setwd()
  - install.packages()
  - library()
  - df <- read.dbf()
  - df$NAME
  - dim(df)
  - class(df)
  - ls()
  - rm(list=ls())
  - sum(df$NAME)
  - as.numeric()
  - as.character()
  - df$NEWVAR <- df$NAME
  - table()
  - tapply, sapply, lapply, etc.

2 The Working Directory

Get my Working Directory (a.k.a. where am I?)

```
getwd()
```

Set my Working Directory (a.k.a. change directory)

```
setwd("/home/ennaniux/Documents/R_Modelling")
```

How does this work in Windows?

```
setwd("C:/home/ennaniux/Documents/R_Modelling")
```

or maybe

```
setwd("C:\\home\\ennaniux\\Documents\\R_Modelling")
```

3 Installing packages

The Packages are sets of tools that can be downloaded from different servers around the world. Different packages have different R functions for specific purposes.

The Comprehensive R Archive Network (CRAN) `https://cran.r-project.org/`. CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R.

For example, the foreign package allows R to read different data set files like .sav, .dbf, and other file extensions.

In order to install the package foreign we type in the console

```
install.packages("foreign")
```

this will provide a list of possible servers to choose from, and you can select one close to your location.

4 Simple manipulations; numbers and vectors

The simplest data structure R operates on is the **numeric vector**, which is a single entity consisting of an ordered collection of numbers.

```
x <- c(1,3,5,9)
x
```

```
1
3
5
9
```

The syntax is equivalent to

```
c(11,31,15,19) -> y
y
```

```
11
31
15
19
```

and

```
assign("z",c(-1,33,2.5,9) )
z
```

```
 -1
 33
2.5
  9
```

Linear operations:

```
z * y + x
```

```
 -10
1026
42.5
 180
```

Definition of a sequence

```
3:10
```

```
 3
 4
 5
 6
 7
 8
 9
10
```

If we want to know the number of entries in our vector, then we use the function `length`

```
length(c(3:10))
```

```
8
```

R tries to apply operations on vectors

```
x <- c(3:10)
x^2
```

```
  9
 16
 25
 36
 49
 64
 81
100
```

The sum of the first 100 natural numbers, if we want to perform the sum one can do for instance

```
x <- c(1:100)
sum(x)
```

```
5050
```

Missing values are denoted by `NA`. Whenever there is a missing value, the default behaviour is to be reminded:

```
x <- c(4, 4, NA, 2, 3, NA, 5)
sum(x)
```

```
nil
```

Then, if you want to omit the missing values in the operation you have to specify it

```
x <- c(4, 4, NA, 2, 3, NA, 5)
sum(x, na.rm=TRUE)
```

```
18
```

A vector can also consists of strings or character type entries:

```
x <- c("Uno", "Dos", NA, "Cuatro", "Dos", "Cuatro")
unique(x)
```

```
Uno
Dos
nil
Cuatro
```

How to we know if we have a missing value?

```
x <- c(4, 4, NA, 2, 3, NA, 5)
is.na(x)
```

FALSE
FALSE
TRUE
FALSE
FALSE
TRUE
FALSE

5 Pre-loaded data

In R there are several pre-loaded data

```
## In case the code below does not work
## you may need to install the package dataset
## For a list of available  datasets, type library(help = "datasets")
head(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

We can also obtain a summary of the data set

```
summary(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa :50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicolor:50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	nil
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	nil
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	nil

6 Writing a data frame

Write a data frame by specifying the columns:

```
df <- data.frame(
"NAME" =  c("Aleksandra", "Hugo", "Piotr", "Ewa"),
"AGE"  =  c(29,35, 39, 33),
"HEIGHT"= c(1.68, 1.83, 2.03, 1.66) )
df
```

NAME	AGE	HEIGHT
Aleksandra	29	1.68
Hugo	35	1.83
Piotr	39	2.03
Ewa	33	1.66

What is the dimension of the data frame?

```
dim(df)
```

What are the variable names of the data frame?

```
names(df)
```

7 Reading a data frame

From a .csv file

```
df <- read.csv('./path_to/file.csv')
```

From a .dbf file

```
library(foreign)
df <- read.csv('./path/to/file.dbf')
```

From a .sav file

```
library(foreign)
     df <- read.spss(file='./path/to/file.sav', to.data.frame=TRUE)
     str(df)   # show the structure of the data frame
```

From the clipboard

```
df2 <- read.table(file = "clipboard", sep = "\t", header=TRUE)
```

8 Creating a new variable

- Graphics
- Reading data
- Markdown