COMP828 Introduction to R and R Studio

Nuttanan Wichitaksorn

Department of Mathematical Sciences
Auckland University of Technology

Introduction to R[1]

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hard-copy.

[1] https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

Introduction to R Studio

- `RStudio` is a user interface or an integrated development environment for `R`.
- Generally, `R` needs to be first installed before `RStudio` installation.
- Usually, `RStudio` interface layout shows or displays 4 panels: Script, Console, Environment, and information on Files, Plots, Packages, etc.
- See more details in https://rladiessydney.org/courses/ryouwithme/01-basicbasics-1/

R and R Studio Installations

- Download and install `R` through https://cran.r-project.org/bin/windows/base/

- Or download and install `R` and `RStudio` from https://posit.co/download/rstudio-desktop/

Working with R/R Studio

- Learning by coding
- R vs Excel
- Writing R scripts
- Libraries and Packages
- (Language Key setting: Try not to use ~ key on your keyboard.)

Some Basics[2]

You can use R as a calculator

```
1 / 200 * 30
```

```
[1] 0.15
```

```
(59 + 73 + 2) / 3
```

```
[1] 44.66667
```

```
sin(pi / 2)
```

```
[1] 1
```

You can create new objects with <−:

```
x <- 3 * 4
```

[2] https://r4ds.had.co.nz/workflow-basics.html

Some Basics (cont.)

All R statements where you create objects, **assignment** statements, have the same form:

```
object_name <- value
```

- You will make lots of assignments and <- is a pain to type.
- Don't be lazy and use =, it will work, but it will cause confusion later.
- Instead, use RStudio's keyboard shortcut Alt + - (the minus sign).
- Notice that RStudio automagically surrounds <- with spaces, which is a good code formatting practice.

Some Basics (cont.)

Object names must start with a letter, and can only contain letters, numbers, _ and .

You want your object names to be descriptive, so you'll need a convention for multiple words. We recommend **snake_case** where you separate lowercase words with _.

```
i_use_snake_case
otherPeopleUseCamelCase
some.people.use.periods
And_aFew.People_RENOUNCEconvention
```

Be careful! `R` code is case-sensitive.

Do you need help? Then, use `help()` or `?`

Some Basics (cont.)

R has a large collection of built-in functions that are called like this:

```
function_name(arg1 = val1, arg2 = val2, ...)
```

Example:

```
seq(1, 10)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

Try this:

```
x <- "hello world"
```

What if you see the + sign in the *Console*:

```
> x <- "hello
+
```

Press **ESCAPE** button!

Some Basics (cont.)[3]

Assigning a value to a name in R - the <- symbol is an assignment operator

```
a <- 1
```

Print the value using the name

```
a
```

```
[1] 1
```

Alternatively print the value using the print function

```
print(a)
```

```
[1] 1
```

[3] https://edbrooks.github.io/R_for_sports_data/scripts/basics_in_r.R

Some Basics (cont.)

Check the class of the object

```
class(a)
```

```
[1] "numeric"
```

Create an integer value of 1

```
b <- 1L
```

Print the value using the name

```
b
```

```
[1] 1
```

Check the class of the object

```
class(b)
```

```
[1] "integer"
```

Some Basics (cont.)

Create a character value using text

```
c <- "text"
```

Print the value using the name

```
c
```

```
[1] "text"
```

```
class(c)
```

```
[1] "character"
```

Some Basics (cont.)

Create a vector of values

```
vector_a <- c(1,2,3,4,5)
```

Print the vector

```
vector_a
```

```
[1] 1 2 3 4 5
```

Some Basics (cont.)[4]

An array can be considered as a multiply subscripted collection of data entries, for example numeric.

R allows simple facilities for creating and handling arrays, and in particular the special case of matrices.

A dimension vector is a vector of non-negative integers. If its length is *k* then the array is *k*-dimensional, e.g. a matrix is a *2*-dimensional array. The dimensions are indexed from one up to the values given in the dimension vector.

For example:

```
xx <- rnorm(10)      # Generate 10 normal random numbers
dim(xx)              # Check the dimension
X <- matrix(xx,2,5) # Put the numbers in a 2x5 matrix
dim(X)               # Check the dimension (again)
```

[4] https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

More Tutorials on R and R Studio

- Try R Code School http://tryr.codeschool.com/
- DataCamp Introduction to R https://www.datacamp.com/courses/free-introduction-to-r
- R tutorial (Clarkson University) http://www.cyclismo.org/tutorial/R/
- Coursera R Programming https://www.coursera.org/course/rprog
- R for Sports data https://edbrooks.github.io/R_for_sports_data/