

An introduction to WS 2017/2018

Getting Started with R

Dr. Noémie Becker
Dr. Sonja Grath

Special thanks to:

Prof. Dr. Martin Hutzenthaler and Dr. Benedikt Holtmann for significant contributions to course development, lecture notes and exercises

What you should know after day 2

Part I: Getting started

- What is R
- Installation of R and RStudio
- How R is organized

Part II: Basics

- Organize your R session
- R as calculator
- What is a function
- What is an assignment in R
- How to plot a continuous function and how to make a scatterplot
- Getting help

What is R?

- R is a comprehensive statistical environment and programming language for professional data analysis and graphical display.
- It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories.
- Webpage: <http://www.r-project.org>

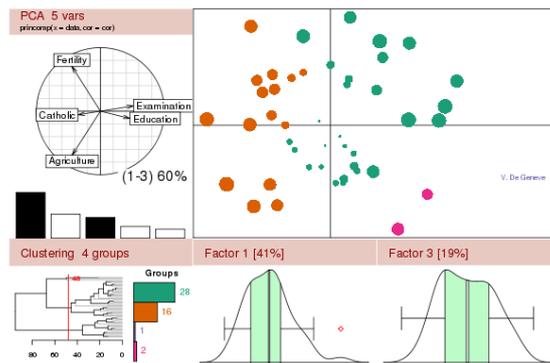
Advantages:

- R is free
- New statistical methods are usually first implemented in R
- Lots of help due to active community

Disadvantages:

- R has a long learning phase
- No 'undo'
- Work with scripts

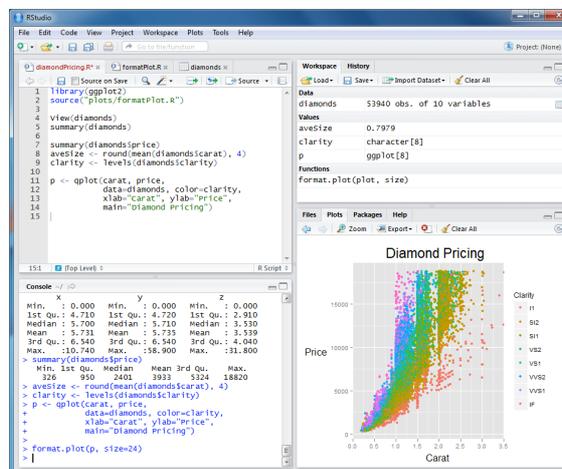
The R Project for Statistical Computing



3

R Studio

- Powerful IDE (Integrated Development Environment) for R
- It is free and open source, and works on Windows, Mac, and Linux and over the web
- Webpage: <https://www.rstudio.com/>



RProgramming.net

4

Literature

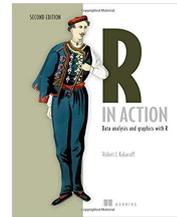
R in Action

Data Analysis and Graphics with R

2nd edition (2011)

Robert I. Kabacoff

<https://www.manning.com/books/r-in-action-second-edition>



Homework: Read Chapter 1 (freely available online as PDF)

Webpage

<http://www.statmethods.net/>

Getting started with R

An Introduction for Biologists

(2017)

Andrew P. Beckerman, Dylan Z. Childs & Owen L. Petchey



... and many, many more (also free web tutorials)

5

How R is organized

R commands are organized in packages (also called libraries)

Examples: stats, datasets, ggplot2, dplyr

To use a package, it has to be installed AND loaded!

Which packages are loaded at start?

```
library(lib.loc=.Library)
```

Which packages are installed?

```
installed.packages()
```

Load package:

```
library(packagename)
```

How to get help:

```
library(help="package")
```

```
??package
```

Try yourself:

```
library(help="ggplot2")
```

```
??ggplot2
```

6

Pre-Defined Datasets

R comes with a huge amount of pre-defined datasets, available in the package 'datasets' (usually available at start)

Examples: 'cars', 'mtcars', 'chickwts', ...

→ can be used for exercises, demonstration of in-built functions...

How to use a dataset:

```
data(cars)
```

How to get help on a dataset:

```
?cars
```

→ We will use pre-defined datasets in some of the exercises

7

Organize your R session

General advice:

- Organize your work in folders (e.g., "Rcourse/Day2")
- Save your commands in scripts

What is a script?

- A computer-readable **text** file (do not use MS Word or similar)
- For R, the conventional extension is .R

Example:

Example.R (see webpage)

8

How to organize a R session

- Open RStudio or a R console
- Open a new or pre-existing script in the text editor or RStudio (extension .R)
- Save the file (for example as 'Day2.R')
- Set your working directory (wd) with `setwd("path2directory")`
- Check your working directory with `getwd()`
- Load (and install) required packages
 - Install with `install.packages("name")` - only once, need to specify CRAN mirror
 - Load with `library(name)` – each session if required
- Comment your script with `#` – **REALLY IMPORTANT!**
- Write and execute your commands (with button or 'Ctrl+Enter' in Rstudio)
- Output is saved in your working directory (if folder unspecified)
- Save your script ('Ctrl+S')
- Quit your session and save workspace if required (`q()` in console)

9



R as calculator



Basic arithmetic operations

`2+3`

`7-4`

`3*5`

`7/3; 2^6`

Integer vs. modulo division

`5 %/% 3` # "5 divided by 3 without decimal positions"

`6 %% 2` # "if you divide 6 by 2 - what's the rest?"

Caution: German (Spanish, French..) decimal notation does not work!

`> 1,2`

Error: unexpected ',' in "1,"

`> 1.2` ✓

10

Functions/Commands

General form:

function()

Examples:

```
sqrt()  
exp()  
sum()  
prod()  
...
```

Functions can have pre-defined **parameters/arguments** with default settings

→ help page of the function

```
?read.table()
```

11



R as calculator



Important functions

```
exp()  
sin()  
cos()  
max()  
min()  
sum()  
prod()  
sqrt()  
factorial()  
choose()
```

factorial()
"4 factorial", 4!
→ 4*3*2*1

choose()
"5 choose 2", $\binom{a}{b}$
$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$



👉 Try yourself:

```
exp(1); exp(log(5))  
sin(pi/2)  
cos(pi/2)  
max(4,2,5,1); min(4,2,5,1)  
sum(4,2,5,1); prod(4,2,5,1)  
sqrt(16)  
factorial(4)  
choose(5,2)
```

12

How to plot a continuous function

You need

- Plotting function
- The continuous function you want to plot
- Range [a, b]

As plotting function you can use

```
plot()
```

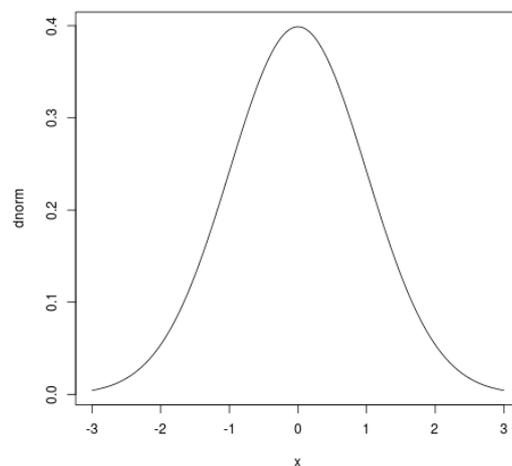
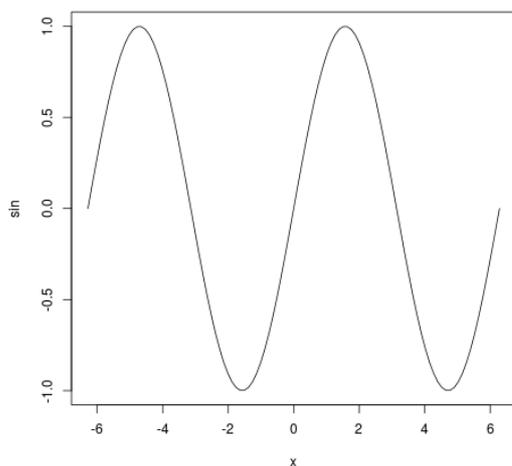
If *fun* is a function, then `plot(fun, from=a, to=b)` plots *fun* in the range [a, b]

Examples:

```
plot(sin, from=-2*pi, to=2*pi)
```

```
plot(dnorm, from=-3, to=3)
```

13



14

Assignments

General form:

`variable <- value`

Example:

```
x <- 5
```

“The **variable** 'x' is assigned the **value** '5’”

Valid variable names: contain numbers, '_', characters

Allowed:

`my.variable`, `my_variable`, `myVariable`
`favourite_color`, `a`, `b`, `c`, `data2`, `2test` ...

NOT allowed: '.' followed by number at the beginning

`.4you`

and neither are the reserved words, e.g: `if`, `else`, `repeat`, `while`
`function`, `for`, `FALSE`, `TRUE`, etc.

15

Assignments

You can write an assignment in three different ways:

```
x <- 5
```

```
5 -> x
```

```
x = 5
```

Have a look here:

```
plot(dnorm, from=-3, to=3)
```

Works with longer expressions:

```
x <- 2
```

```
y <- x^2 + 3
```

```
z <- x + y
```

And with complete vectors (more on vectors tomorrow):

```
x <- 1:10
```

```
y <- x^2
```

16

How to plot numerical vectors

You need

- Plotting function
- Two vectors

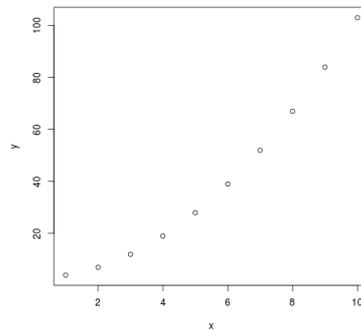
As plotting function you can use

```
plot()
```

If x and y are numerical vectors, then `plot(x, y)` produces a scatterplot of y against x

Examples:

```
x <- 1:10  
y <- x^2  
plot(x, y)
```



17

Help!



R console

```
help(solve)           #help page for command "solve"  
?(solve)             #same as help(solve)  
help("exp")  
help.start()  
help.search("solve") #list of commands which could  
                    #be related to string "solve"  
??solve              #same as help.search("solve")  
example(exp)         #examples for the usage of 'exp'  
example("*")         #special characters have to be in  
                    #quotation marks
```

18

What you should do now



If you have your own laptop or computer

1. Install R and RStudio (see web tutorials)
2. Read the first chapter of “R in Action” (course web page)
<http://www.manning.com/kabacoff/SampleCh-01.pdf>
3. Open a R session and try the commands we learned today (lecture slides)
→ if you have trouble with installing R, ask us/the tutors
4. Work on the first exercise sheet

If you don't have your own laptop or computer

1. Read the first chapter of “R in Action” (course web page)
<http://www.manning.com/kabacoff/SampleCh-01.pdf>
2. This afternoon in the exercise session: open a R session and try the commands we learned – there will be enough time
3. Work on the first exercise sheet