An introduction to R
WS 2014/2015

Dr. Noémie Becker (AG Metzler)
Dr. Sonja Grath (AG Parsch)

Special thanks to: Prof. Dr. Martin Hutzenthaler
(previously AG Metzler, now University of Duisburg-Essen)
course development, lecture notes, exercises

Course Organisation

Webpage:

http://evol.bio.lmu.de/_statgen/Rcourse/ws1415/

→ see **Syllabus_R-Course_2015.pdf** with all information!

→ all course material will be posted here

First session: March 3 14:00 to 17:00

Then:

In the mornings: Lectures and discussion of exercises

March 4-6 and March 9-12 9:00 to 12:00

B 01.027

In the afternoons: Preparation of exercises, practice

In total: 6 exercise assignments (3 this week, 3 next week)

Course Certificate

2 Possibilities:

Certificate of Attendance

→ ACTIVE attendance (be there/be prepared/be active)

→ Do not forget to sign attendance sheet

→ 8 lectures, **you can only miss 1**

Graded Certificate

→ Written exam about lectures + exercises

→ No compulsory attendance

Final Exam: March 13, 10-12 am (B 01.027)

Make-up Exam: March 30, 10-12 am (room tba)

→ please register with Sonja until March 25, grath@bio.lmu.de

R is programming...

… you can only be a good programmer if you practice a lot!

Course Outline – Week 1

Goals:

- Use R console and Rstudio
- Organize your R session
- Load and use packages and predefined data sets
- R basics (calculations, assignments...)
- Basic statistics in R
- Get your data into R and work with data
- Basic plotting

Lecture Notes:
(see webpage, R-Course_Lecture-Notes_2015.pdf)
Chapter 1-4 + some additional material

Exercise Assignments:
Sheet 1-3 + some additional practical tasks from the lectures

- you can work on the exercises in the afternoon in **C 00.005** or **G 00.037** or elsewhere (no compulsory attendance)
- solutions to the exercises will be presented and discussed in the morning sessions

Course outline – Day 1

Getting started

What is R?

Downloading/Installing R

Literature

Libraries

Lecture notes, pp 3-6

Basics

Organize your R session

R as calculator

Getting help

Assignments, comparisons and logical expressions

Printing and plotting (1)

Vectors

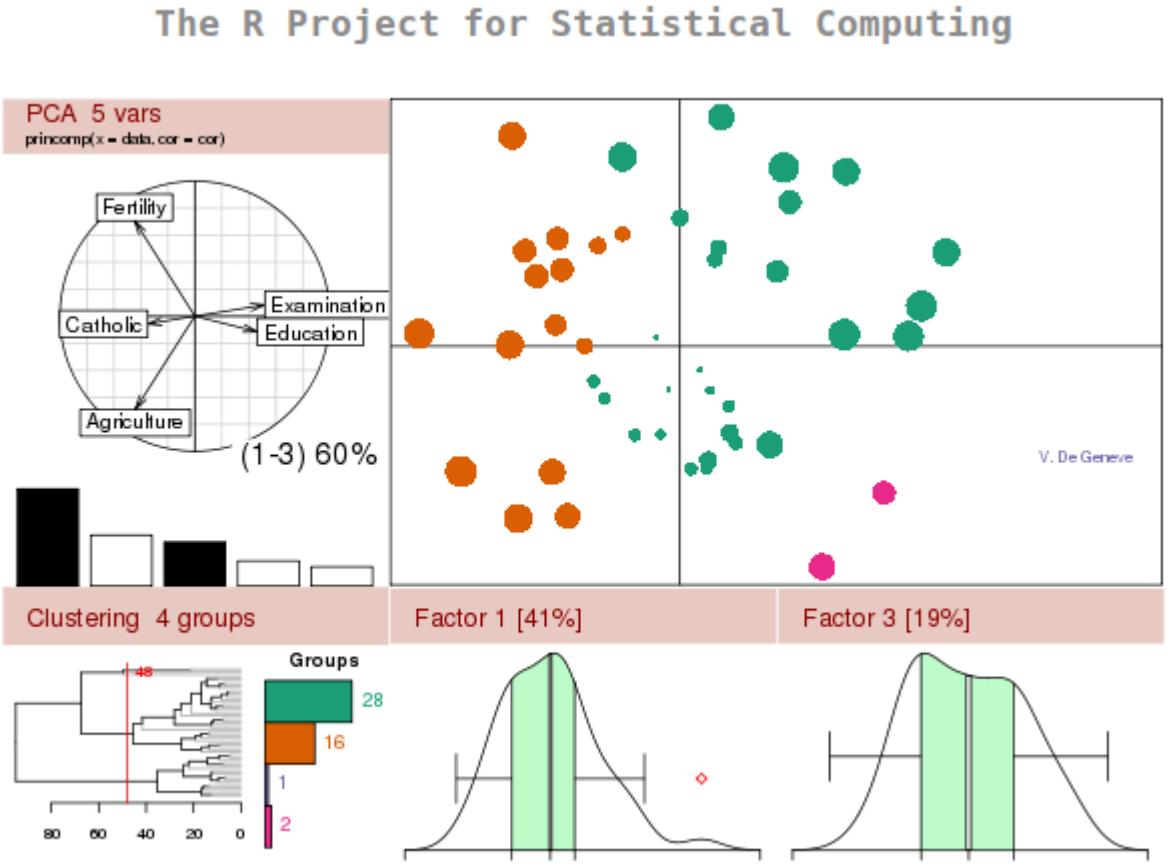
Lecture notes, pp 6-12

Getting started

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



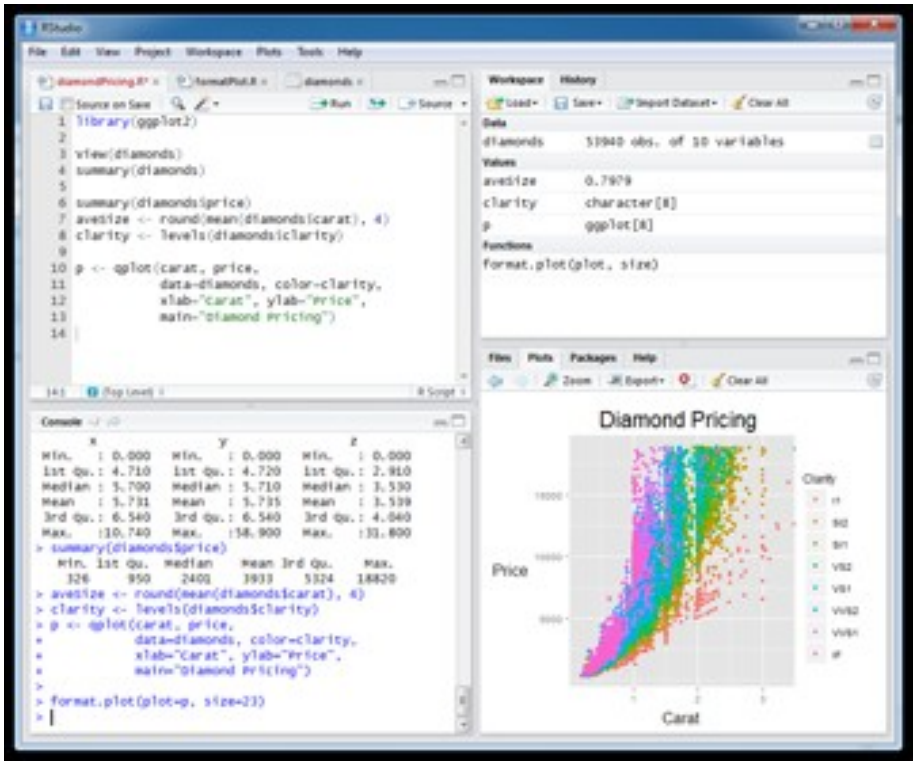
Advantages:
- R is free
- New statistical methods are usually first implemented in R

Disadvantages:
- R has a long learning phase
- No 'undo'

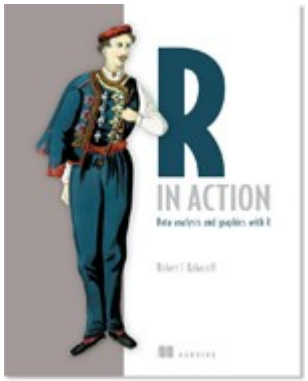
R Studio

- Powerful IDE for R
- It's free and open source, and works on Windows, Mac, and Linux and over the web
- Short introduction tomorrow and on the course webpage

Webpage:
https://www.rstudio.com/

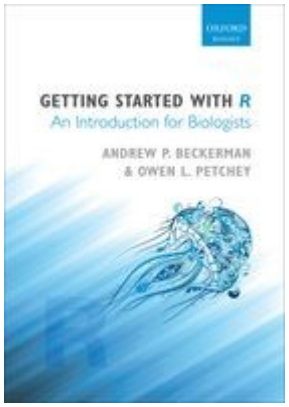


Literature



R in Action
Data Analysis and Graphics with R
2nd edition (2011)
Robert I. Kabacoff
http://www.manning.com/kabacoff/
→ Sample Chapter 1 (PDF)
→ "Homework"

Webpage
http://www.statmethods.net/



Getting started with R
An Introduction for Biologists
(2012)
Andrew P. Beckerman & Owen L. Petchey

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

Libraries

R commands are organized in libraries (packages)

Examples: 'stats', 'datasets'

Loaded at start:

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

Load packages:

```
library(packagename)
```

Further useful commands:

```
library(help="packagename")
installed.packages()
download.packages()
install.packages()
```

Basics

Organize your R session

General advice:

Organize your work in folders

Save your commands in scripts (text files)

Example:

```
# Define a vector with age values

age <- c(1,3,5,2,11,9,3,9,12,3)

# Define a vector with weight values

weight <- c(4.4,5.3,7.2,5.2,8.5,7.3,6.0,10.4,10.2,6.1)

# Calculate the mean weight value

mean(weight)

# Quit R session

q()
```

Organize your R session

General advice:

Organize your work in folders

Save your commands in scripts (text files)

Working directory:

```
getwd()
```

```
setwd()
```

Recipe:

(1) Open your favourite text editor

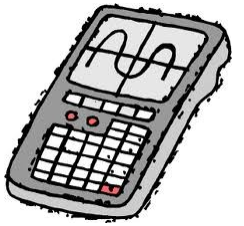
(2) Save the file (*e.g.* example.R)

(3) Define first comments for your workflow

(4) Write your R commands and test them step-by-step

more on organisation: tomorrow and in the exercises!

R as calculator



Basic arithmetic operations

2+3

7-4

3*5

7/3; 2^6

Comments

This is a comment

Integer vs. modulo divison

5 %/% 3 # “5 divided by 3 without decimal positions” → 1

5 %% 3 # “if you divide 5 by 3 – what's the rest?” → 2

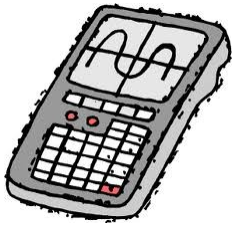
Caution: German (Spanish, French..) decimal notation does not work!

> 1,2

Error: unexpected ',' in “1,”

> 1.2 ✔

R as calculator



Important functions

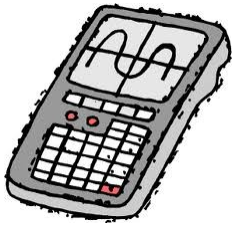
```
exp(1)

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)
```

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

choose()
"5 choose 2", $\binom{a}{b}$

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}.$$

R as calculator



Further functions

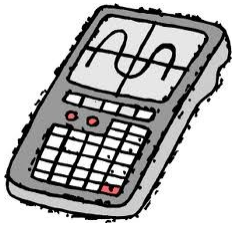
log10(), log2(), tan(), asin(), acos(), atan(), sinh(), cosh(), tanh(), asinh(), acosh(), atanh(), abs(), round(), floor(), ceiling(), trunc(), signif()...

and many, many more!

→ help pages

→ write your own functions [next week]

R as calculator



Further functions

log10(), log2(), tan(), asin(), acos(), atan(), sinh(), cosh(), tanh(), asinh(), acosh(), atanh(), abs(), round(), floor(), ceiling(), trunc(), signif()...

and many, many more!

→ help pages

→ write your own functions [next week]

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
```

Assignments

General form:

variable <- *value*

Example:

```
x <- 5
```

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

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

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

.4you

Allowed:

my.variable, my_variable, myVariable

favourite_color, a, b, c, data2, 2data ...

Assignments

```
x <- 5        # The variable x is assigned the value 5

5 -> x        # The same assignment but unusual

x = 5         # The same assignment but unusual
```

Works with longer expressions:

```
x <- 2

y <- x^2 + 3

y

[1] 7
```

… or to define functions:

```
myfunction <- sqrt

myfunction(81)

[1] 9
```

Comparisons

```
4 == 4                  #Are both sides equal?
[1] TRUE                #TRUE is a constant in R
4 == 5                  #Are both sides equal?
[1] FALSE               #FALSE is a constant in R
2 != 3                  #! is negation, != is 'not equal'
3 != 3
3 <= 5
5 >= 2*2
5 > 2+3
5 < 7*45
```

Caution: Never compare 2 numerical values with ==

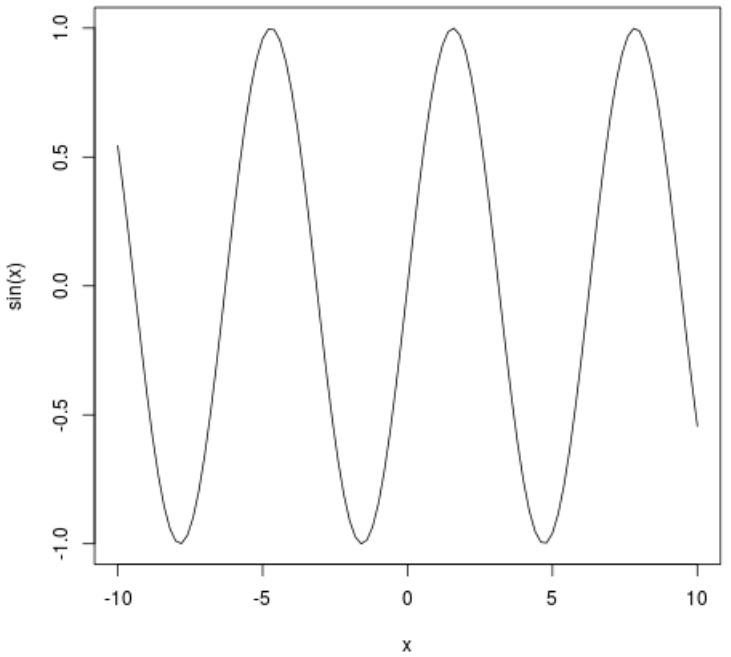
```
cos(pi/2) == 0
[1] FALSE

cos(pi/2)
[1] 6.123234e-17     #R does not answer with 0
```

Logical expressions

```
TRUE & TRUE    #& is the logical AND
[1] TRUE
TRUE & FALSE
[1] FALSE
TRUE | FALSE      #| is the logical OR
5 > 3 & 0 != 1
5 > 3 & 0 != 0
as.integer(TRUE); as.integer(FALSE)
[1] 1    #the internal representation of TRUE is 1
[1] 0    #the internal representation of FALSE is 0
```

Printing and Plotting

```
x <- 3

print(x)

x

print(sqrt(2),digits=5)

y <- 42

cat("And the answer is ",y,".\n")

plot(sin, from = -10, to = 10)
```



Vectors

Vectors are enumerations of arbitrary objects

For example:

```
(2,5,3,7)

(1,4,7,10)

("green","blue","red")

("A","T","G","C")

("A","A","A","G","G")
```

Vectors

Vectors are enumerations of arbitrary objects

To create vectors, you can use functions in R: 'c()', 'seq()', 'rep()'

```
c(2,5,3,7)

seq(from=1,to=10,by=3)

seq(from=3,to=7)

seq(1,11,3)

seq(3,7)

seq(7,3)

3:7

c(2:5, 3:7)

rep(3,5)

rep(0:2,3)

rep(7:9,2:4)
```

Operations on vectors

You assess elements of a vector with the [ ]-Operator:

```
x <- c(12,15,13,17,11)

x[4]

[1] 17

x[3:5]

x[-2]

X[-(3:5)]
```

Standard operations on vectors are element by element:

```
c(2,5,3) + c(4,2,7)

[1] 6 7 10

2 + c(2,5,3)

c(2,5,3)^2
```

Operations on vectors

```
sum(5:7)

prod(4:6)

x <- 1:5

x[3:5]

x[-2]

x > 3

[1] FALSE FALSE FALSE TRUE TRUE
```

Useful commands on vectors:

```
length(x)

rev(x)

sort(x)

unique(x)
```

Operations on vectors

Some tricky but very useful commands on vectors:

```
x <- c(12,15,13,17,11)      # vector definition

x[x>12] <- 0                # set all values that are
                            # greater than 12 to 0

x[x==0] <- 2                # set all values that are
                            # equal to 0 to 2

X==2                        # test where the vector x is
                            # equal to 2
```

```
[1] FALSE TRUE TRUE TRUE FALSE
```

```
as.integer(x==2)
```

```
[1] 0 1 1 1 0
```

```
sum(x==2)
```

```
[1] 3
```

If you are new to R and programming, this was pretty much...

What you should do now

If you have your own laptop or computer

1. Install R and Rstudio (see tutorial on the web)
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

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. Tomorrow afternoon in the exercise session: open a R session and try the commands we learned – there will be enough time

Keep in mind:
Programming needs a lot of practice!

Outlook – Day 2

Review session day 1

Basics

- Matrices
- Data types in R

Basic statistics with R

- Some distributions implemented in R
- Examining the distribution of a set of data
- Random number generators

A sample R session