An introduction to R
WS 2013/2014

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

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

Course outline – Day 1

Organisation

Getting started

- What is R?
- Downloading/Installing R

Basics

- R as calculator
- Getting help
- Assignments, comparisons and logical expressions
- Vectors
- Operations on vectors
- Matrices
- Functions, arguments/parameters
- Literature

Course Organisation

Webpage:

http://evol.bio.lmu.de/_statgen/Rcourse/ws1314/

Lectures/Solutions to Exercises (B01.027):

March 3	**9:00 to 12:00 am**
March 4-7 and March 12-14	**9:00 to 10:30 am (lectures)**
	10:30 to 12:00 am (solutions to exercises)

March 10: NO course, March 11: 14:00 to 17:00 (B01.019)

In the afternoon: Preparation of exercises

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

→ discussion/solution together in the exercise hours in the morning

Course Certificate

2 Possibilities:

Certificate of Attendance

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

→ Do not forget to sign attendance sheet

Graded Certificate

→ Written exam about lectures + exercises

Final Exam: March 17, 10-12 am (room tba)

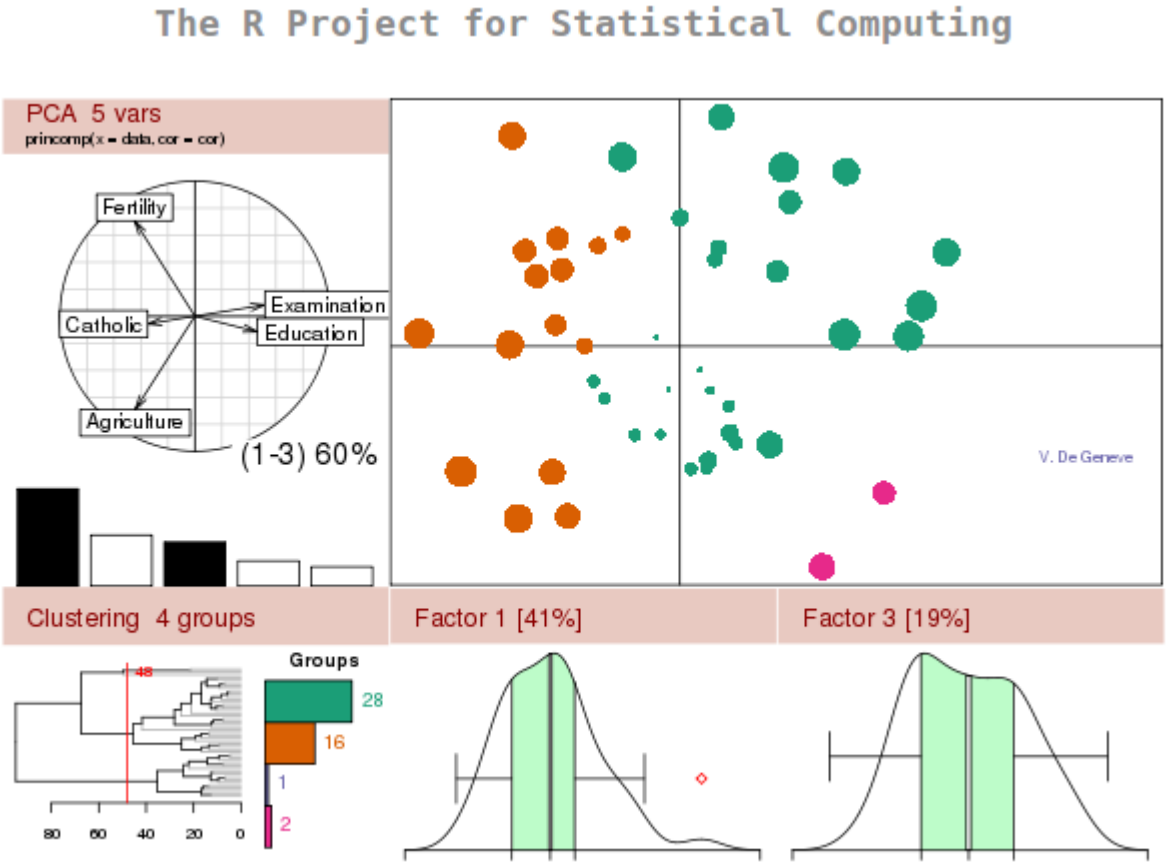
Make-up Exam: April 7 (room/time tba)

→ please register with Noémie until April 1, nbecker@bio.lmu.de)

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.

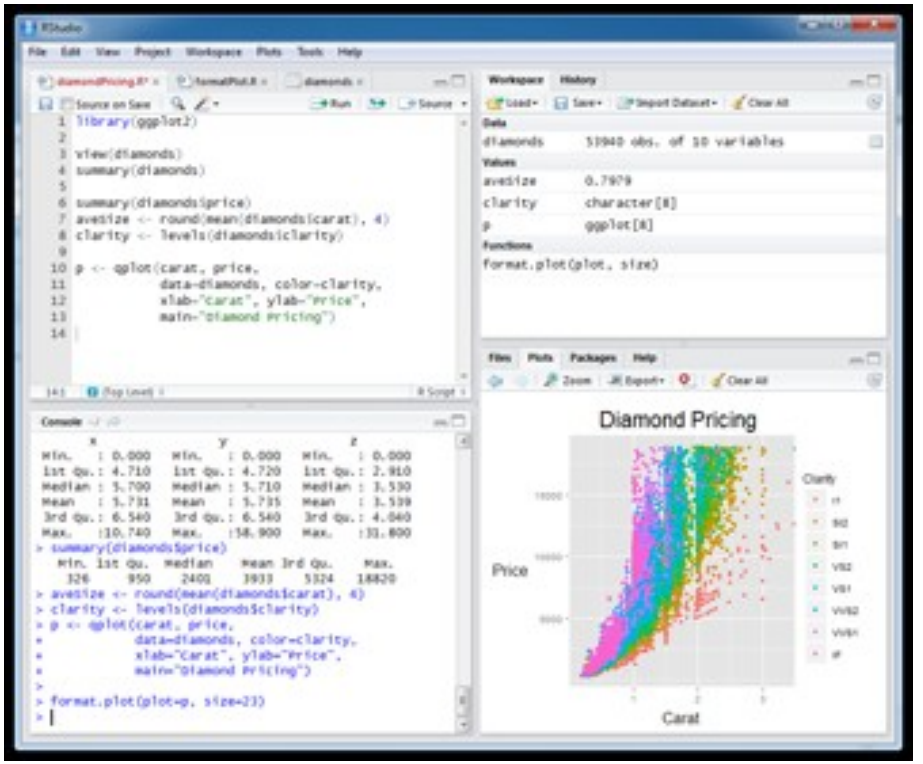
Webpage:
http://www.r-project.org



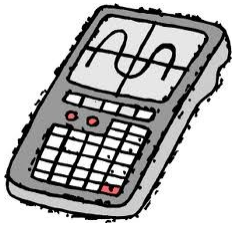
R Studio

- Powerful IDE for R
- It’s free and open source, and works on Windows, Mac, and Linux and over the web
- Short introduction today (Paul Staab, AG Metzler)

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



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

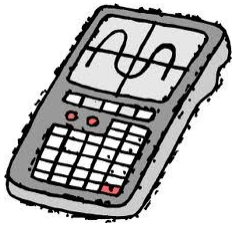
Caution: German (Spanish, French..) decimal notation does not work!

> 1,2

Error: unexpected ',' in “1,”

> 1.2 ✔

R as calculator



Important functions

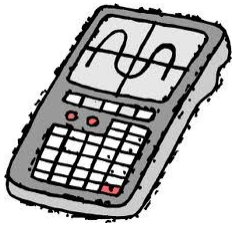
```
>exp(1)
>exp(log(5))
>sin(pi/2)
>cos(pi/2)
>max(4,2,5,1)
>min(4,2,5,1)
>sum(4,2,5,1)
>prod(4,2,5,1)
>sqrt(16)
>factorial(4)
>choose(5,2)
```

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]

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
```

Vectors

Vectors are enumerations of arbitrary objects

To create vectors: '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)

>x[x>12] <- 0

>x[x==0] <- 2

>x==2

[1] FALSE TRUE TRUE TRUE FALSE

>as.integer(x==2)

[1] 0 1 1 1 0

>sum(x==2)

[1] 3
```

Matrices - Basics

You can create matrices by:

1. matrix()
2. converting a vector into a matrix
3. binding together vectors

```
>m <- matrix(data = 1:8, nrow=4, ncol=2)

>m <- matrix(1:8,4,2)

>z <- as.matrix(1:6)

>cbind(1:3,5:7)

>rbind(1:3,5:7,10:12)
```

Matrices - Basics

You can create matrices by:

1. matrix()
2. converting a vector into a matrix
3. binding together vectors

Indexing is 'row by column':

```
>m[3,2]

>m[2,]

>m[,2]

>m[2:3,1:2]
```

Matrices - Basics

Examples:

```
>m <- matrix(data = 1:8, nrow=4, ncol=2)
>m
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
>m[2:3,1:2]
     [,1] [,2]
[1,]    2    6
[2,]    3    7
```

Matrices - Basics

Examples:

```
>z <- as.matrix(1:6)

> z

     [,1]

[1,]    1

[2,]    2

[3,]    3

[4,]    4

[5,]    5

[6,]    6
```

Matrices - Basics

Examples:

```
>cbind(1:3,5:7)
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
>rbind(1:3,5:7,10:12)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    5    6    7
[3,]   10   11   12
```

More in the exercises!

Functions/Commands

General form:

function()

Examples:

```
>sqrt()
>exp()
>c()
>matrix()
```

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

→ help page of the function

Parameters/Arguments

Example: matrix()

Which arguments can be used with this function?

>?matrix()

matrix {base}

Matrices

Description

matrix creates a matrix from the given set of values.

as.matrix attempts to turn its argument into a matrix.

is.matrix tests if its argument is a (strict) matrix.

Usage

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
       dimnames = NULL)
```

>matrix(data=(1:6),nrow=2, ncol=3)

Parameters/Arguments

Example: matrix()

Which arguments can be used with this function?

matrix(data=NA, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)

```
>matrix(data=(1:6),nrow=2, ncol=3)

>matrix(ncol=3,data=(1:6),nrow=2)

>matrix(1:6,2,3)

>matrix(1:6,ncol=3, nrow=2)

>matrix(1:6,nr=2, nc=3)

>matrix(d=(1:6),nr=2, nc=3)
```

Fehler in matrix(d = (1:6), nr = 2, nc = 3) :

Argument 1 passt auf mehrere formale Argumente

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

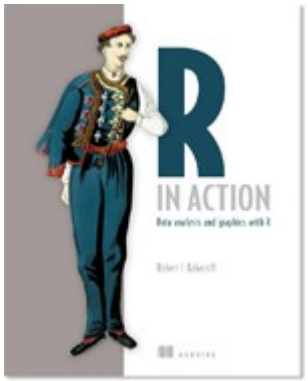
(2) Save the file (*e.g.* MyFirstAnalysis.R)

(3) Define first comments for your workflow

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

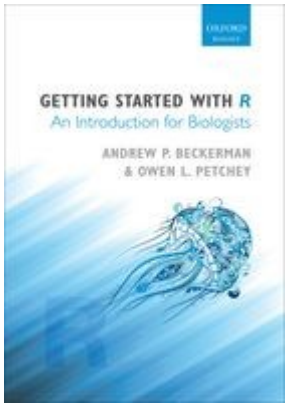
more on organisation: tomorrow and in the exercises!

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 more