Using R for my Research
modern uses: Markdown and Shiny app

Elrozy Andrinopoulou
Department of Biostatistics, Erasmus University Medical Center

e.andrinopoulou@erasmusmc.nl

What is this Workshop About

- Statistics have flourished in the recent years mainly due to the possibility of doing complex analysis using computers

- The most valuable tool of a modern quantitative researcher is his/her personal computer
  - ▷ Many statistical software exist to do simple and specialized analysis

- Analysts must learn not only how to use the software but also what is behind

What is this Course About (cont'd)

- Therefore, the aim of this workshop is twofold:

- Aim I: Introduce R
  - ▷ learn/refresh

- Aim II: Modern uses
  - ▷ Markdown and Shiny app

Agenda

- **Part I:** Introduce R
  - ▷ How does R look like ?
  - ▷ What is R ?
  - ▷ Why R ?
  - ▷ Where do I get R ?
  - ▷ How does R work ?
  - ▷ How to get help in R ?

Agenda (cont'd)

- **Part I:** Introduce R
  - ▷ Using R
  - ▷ Importing Data
  - ▷ Starting with examples
  - ▷ Using R Commands
  - ▷ Most Frequently Used R Objects
  - ▷ Functions
  - ▷ Loops and Control Flow

Agenda (cont'd)

- **Part II:** Modern uses
  - ▷ Markdown
  - ▷ Shiny app

Structure & Material

- Slides and Practicals
  - ▷ you will be asked to perform small tasks
  - ▷ solutions of the practicals available beforehand

Structure & Material (cont'd)

- You are welcome to try along

- You are welcome to interrupt and ask questions

References

- Intro with applications in statistics
  - ▷ Dalgaard, P. (2008) *Introductory Statistics with R, 2nd Ed.* New York: Springer-Verlag. (moderate)
  - ▷ Venables, W. and Ripley, B. (2002) *Modern Applied Statistics with S*. New York: Springer-Verlag. (advanced)

- Programming
  - ▷ Venables, W. and Ripley, B. (2000) *S Programming*. New York: Springer-Verlag.
  - ▷ Chambers, J. (2008) *Software for Data Analysis Programming with R*. New York: Springer-Verlag.

References (cont'd)

- More books that use R (or S) can be found at:
  http://www.r-project.org/doc/bib/R-books.html, or
  http://www.r-project.org/doc/bib/R-jabref.html

References (cont'd)

- R ships with a number of helpful manuals (illustrated later)

- Other manuals and helpful material are available on-line via CRAN: `http://cran.r-project.org/other-docs.html`
  - ▷ 'Simple R' by John Verzani (`http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf`)
  - ▷ 'R reference card' by Tom Short (`http://cran.r-project.org/doc/contrib/Short-refcard.pdf`)

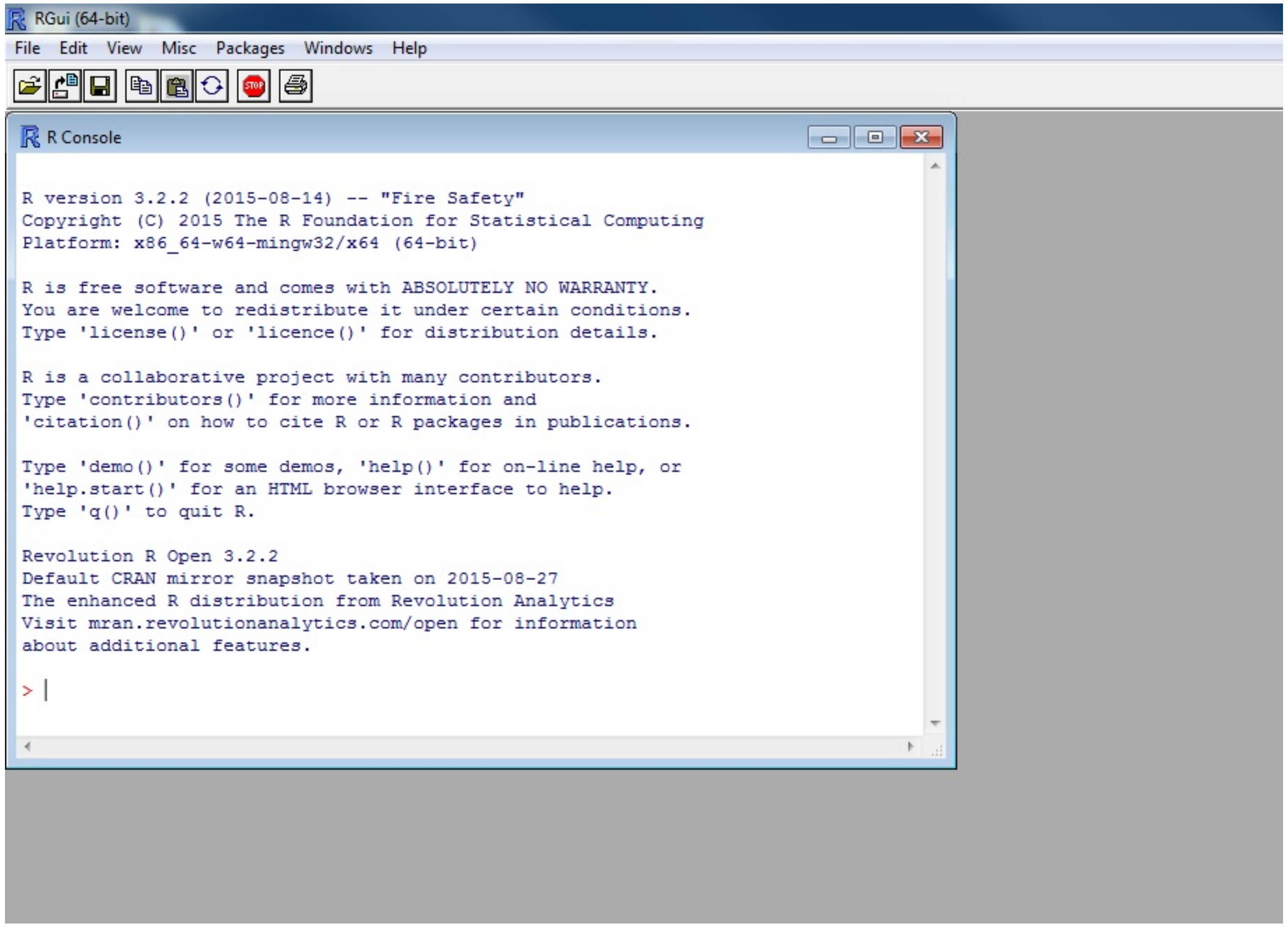
Part I

1 Introduction

- A little bit of history
  - ▷ it was initiated in 1992 by Ross Ihaka and Robert Gentleman at University of Auckland, New Zealand
  - ▷ in 1997 the R Core Team was established with renowned members of the statistical computing community
  - ▷ nowadays, the R Core Team has grown and consists of about 20 members, experts in computing

1 Introduction (cont'd)

- How does R look like ?



```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Revolution R Open 3.2.2
Default CRAN mirror snapshot taken on 2015-08-27
The enhanced R distribution from Revolution Analytics
Visit mran.revolutionanalytics.com/open for information
about additional features.

> |
```

1 Introduction (cont'd)

- What is R ?
  - ▷ is a software environment for statistical computing and graphics.
  - ▷ Unlike SPSS, R is purely command driven

1 Introduction (cont'd)

- Why R ?
  - ▷ R is a free software environment for statistical computing and graphics.
  - ▷ it compiles and runs on LINUX, Windows and MacOS
  - ▷ R has extensive and powerful graphics & data manipulation capabilities
  - ▷ it can easily interface with low-level programming languages, e.g., C/C++ or Fortran
  - ▷ it can be easily extended via R packages
  - ▷ the source code is available
  - ▷ users are allowed to modify and redistribute the code

1 Introduction (cont'd)

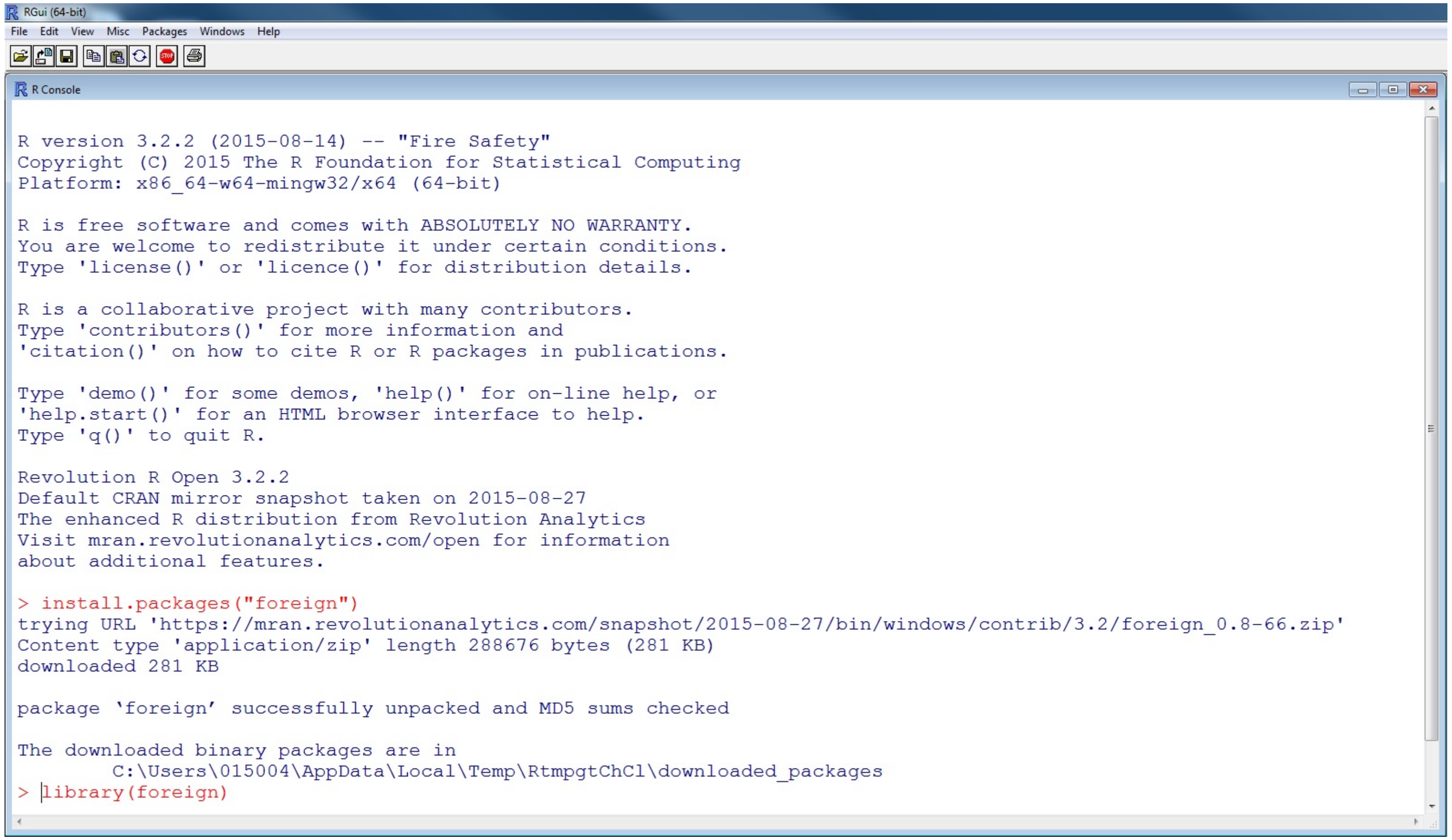
- Where do I get R ?
  - ▷ `http://cran.r-project.org`
  - ▷ choose your platform, e.g., Windows, Linux
  - ▷ e.g., for Windows: `Windows` $\rightarrow$ `base` $\rightarrow$ `Download R 3.4.3 for Windows`
  - ▷ Install . . .

1 Introduction (cont'd)

- How does R work ?
  - ▷ Packages built for specific tasks
  - ▷ Download R packages from the CRAN web site ⇒ within R
    - * Packages
    - * Install package(s) . . .
    - * make your choice(s)
    - * load the package using `library()` (note: install does not mean load)

1 Introduction (cont'd)

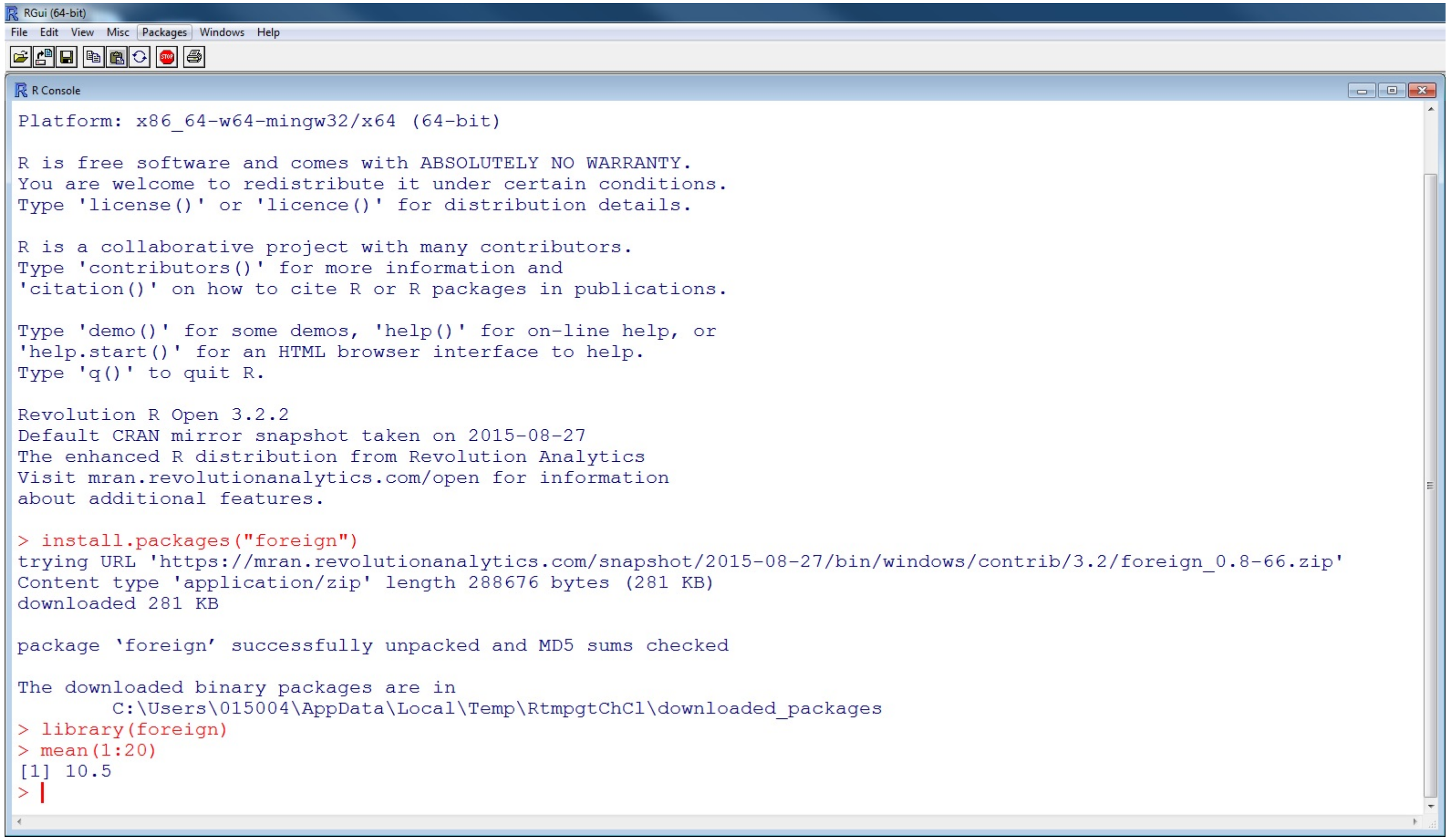
- How does R work ?



```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Revolution R Open 3.2.2
Default CRAN mirror snapshot taken on 2015-08-27
The enhanced R distribution from Revolution Analytics
Visit mran.revolutionanalytics.com/open for information
about additional features.

> install.packages("foreign")
trying URL 'https://mran.revolutionanalytics.com/snapshot/2015-08-27/bin/windows/contrib/3.2/foreign_0.8-66.zip'
Content type 'application/zip' length 288676 bytes (281 KB)
downloaded 281 KB

package ‘foreign’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\015004\AppData\Local\Temp\RtmpgtChCl\downloaded_packages
> library(foreign)
```

1 Introduction (cont'd)

- How does R work ?



```
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Revolution R Open 3.2.2
Default CRAN mirror snapshot taken on 2015-08-27
The enhanced R distribution from Revolution Analytics
Visit mran.revolutionanalytics.com/open for information
about additional features.

> install.packages("foreign")
trying URL 'https://mran.revolutionanalytics.com/snapshot/2015-08-27/bin/windows/contrib/3.2/foreign_0.8-66.zip'
Content type 'application/zip' length 288676 bytes (281 KB)
downloaded 281 KB

package ‘foreign’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\015004\AppData\Local\Temp\RtmpgtChCl\downloaded_packages
> library(foreign)
> mean(1:20)
[1] 10.5
> 
```

1 Introduction (cont'd)

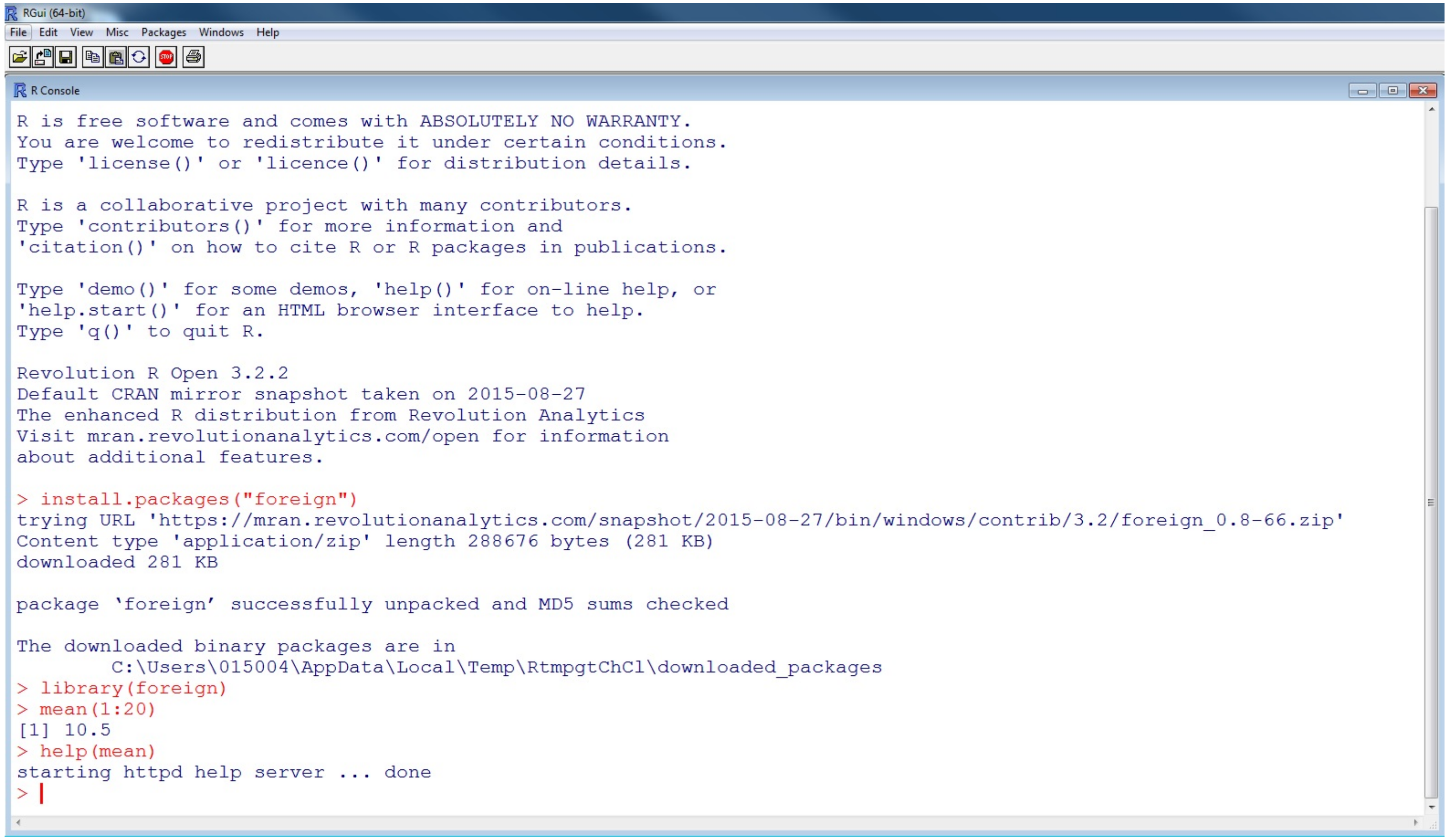
- How to get help in R ?
  - ▷ Within R
    - * `help.search("topic")` or `??"topic"` (depends on the installed packages)
    - * `RSiteSearch("topic")` (requires internet connection)
    - * `help()` or `?` invoke the on-line help file for the specified function
    - * checking the FAQ
  - ▷ On the internet
    - * R-help (`https://stat.ethz.ch/mailman/listinfo/r-help` – mailing list)
    - * R-seek (`http://www.rseek.org` – Google-like searched engine)
    - * R-wiki (`http://rwiki.sciviews.org/doku.php`)

1 Introduction (cont'd)

- How to get help in R ?
  - ▷ On the internet
    - * CRAN Task Views (http://cran.r-project.org/web/views/ – categorization of packages)
    - * Crantastic (http://crantastic.org/ – categorization of packages + reviews)
    - * Equalis (http://www.equalis.com/forums/ – R forum)
    - * R4stats (http://www.r4stats.com/ – examples of basic R programs)
    - * R related Blogs (http://www.r-bloggers.com/ – many useful illustrations of R and R packages)

1 Introduction (cont'd)

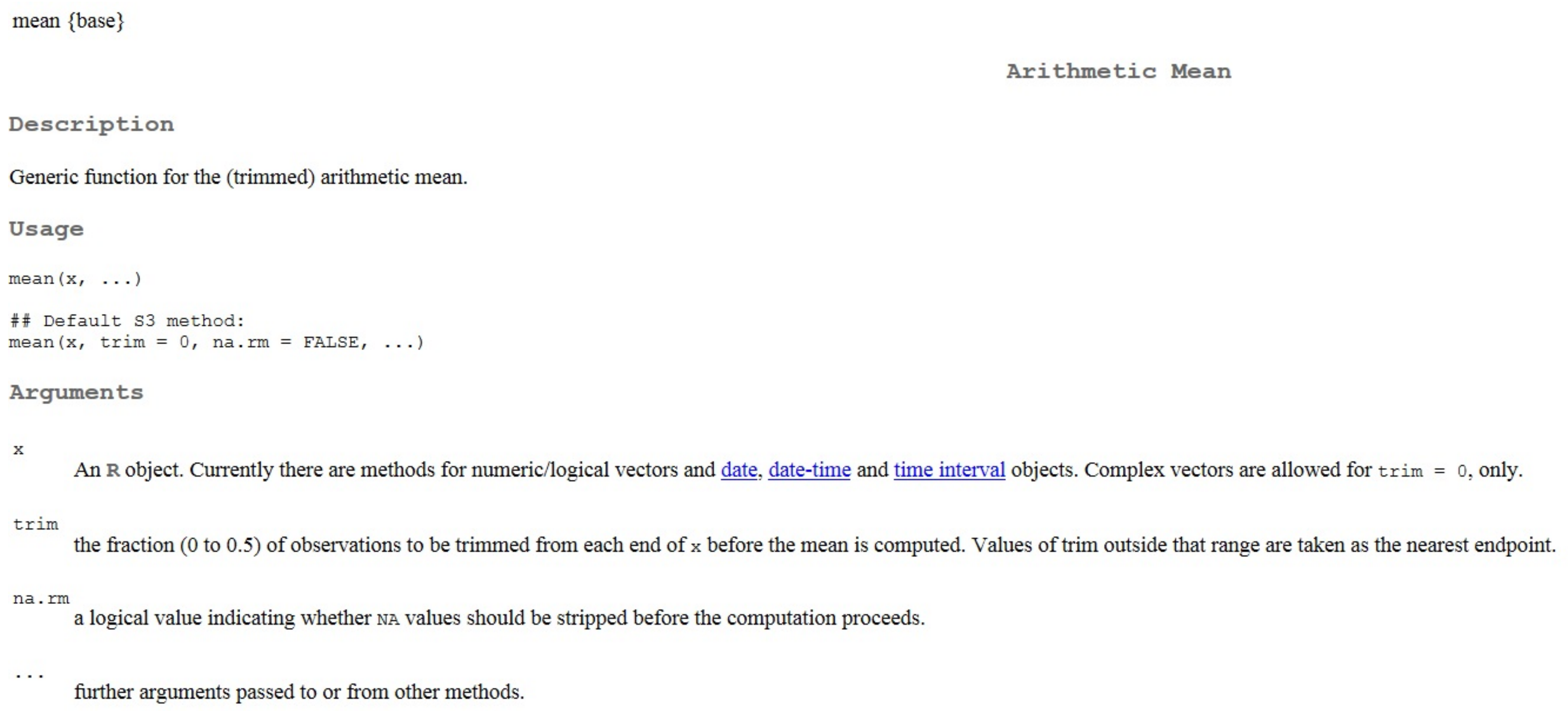
- How to get help in R ?



```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Revolution R Open 3.2.2
Default CRAN mirror snapshot taken on 2015-08-27
The enhanced R distribution from Revolution Analytics
Visit mran.revolutionanalytics.com/open for information
about additional features.

> install.packages("foreign")
trying URL 'https://mran.revolutionanalytics.com/snapshot/2015-08-27/bin/windows/contrib/3.2/foreign_0.8-66.zip'
Content type 'application/zip' length 288676 bytes (281 KB)
downloaded 281 KB

package 'foreign' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\015004\AppData\Local\Temp\RtmpgtChCl\downloaded_packages
> library(foreign)
> mean(1:20)
[1] 10.5
> help(mean)
starting httpd help server ... done
> 
```

1 Introduction (cont'd)

- How to get help in R ?



mean {base}

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

`x`	An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.
`trim`	the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
`na.rm`	a logical value indicating whether `NA` values should be stripped before the computation proceeds.
`...`	further arguments passed to or from other methods.

1 Introduction (cont'd)

- Disadvantages of R
  - ▷ appears intimidating to the first-time user
  - ▷ output is not so nice looking (but there are some alternatives)
  - ▷ exporting output is more difficult
  - ▷ cannot easily handle very very big data sets (depends on the installed RAM)
  - ▷ a lot of things are available but it is sometimes hard to find your way
  - ▷ the quality of the available packages is greatly varying

2 Using R

- R is a command-based functional language
  - ▷ write and execute commands
  - ▷ use and define functions

- You may write the commands in the R console (Windows) or in a shell (Linux)

You will become more familiar with the syntax as you use it

2 Using R (cont'd)

- Strongly advisable to use a suitable text editor – Some available options:
  - ▷ RWinEdt (for Windows; you also need WinEdt installed)
  - ▷ Tinn-R (for Windows; `http://sciviews.org/Tinn-R/`)
  - ▷ Rkward (for Linux)
  - ▷ Rstudio (all major platforms; `http://www.rstudio.org/`)
  - ▷ for more check `http://www.sciviews.org/_rgui/projects/Editors.html`

2 Using R (cont'd)

- For this course: Rstudio (`http://www.rstudio.org/`)
  - ▷ free
  - ▷ works fine in Windows, MacOS and Linux
  - ▷ helpful with errors

3 Importing Data

- `read.table()` and its variants
  - ▷ note: use forward slashes or double backward slashes in the file names, e.g., `"C:/Documents and Settings/User/Data/file.txt"` or `"C:\\Documents and Settings\\User\\Data\\file.txt"`

- Specialized functions for importing data from other statistical packages
  - ▷ packages `foreign` & `Hmisc`
  - ▷ `read.spss()`, `read.csv()`, `read.dta()`, `sas.get()`, etc.

```
read.spss("C:\\Documents and Settings\\User\\Data\\file.sav")
```

4 Starting with Examples

```
> dat
> set.seed(2015+1)
> patient <- c(1:20)
> height <- rnorm(20, 1.70, 0.1)
> weight <- rnorm(20, 70, 10)
> sex <- sample(1:2, 20, replace = TRUE)
> sex <- factor(sex, levels = 1:2, labels = c("male", "female"))

> dat <- data.frame(patient, height, weight, sex)
```

4 Starting with Examples

patient	height	weight	sex
1	1.608526	65.80858	male
2	1.800125	63.45247	male
3	1.694358	66.75113	female
4	1.729665	75.95356	female
5	1.420853	71.79690	male
⋮			

4 Starting with Examples (cont'd)

- $\boxed{\text{What is a matrix/vector?}}$

4 Starting with Examples (cont'd)

patient	**height**	weight	sex
1	**1.608526**	65.80858	male
2	**1.800125**	63.45247	male
3	**1.694358**	66.75113	female
4	**1.729665**	75.95356	female
5	**1.420853**	71.79690	male
⋮			

4 Starting with Examples (cont'd)

- Common questions
  - ▷ What is the average weight:

    ```
    > mean(dat$weight)
    ```
  - ▷ What is the average height:

    ```
    > mean(dat$height)
    ```
  - ▷ ...

5 Using R Commands (cont'd)

- Elementary commands: **expressions** and **assignments**

- An **expression** given as command is evaluated printed and discarded

- An **assignment** evaluates an expression and passes the value to a variable – the result is not automatically printed

5 Using R Commands (cont'd)

- Expression is given as a command,

```
> 10
[1] 10
```

- However, it cannot be viewed again unless the command is rerun.

- In order to store information, the expression should assign the command

```
> x <- 10
> x
[1] 10
> mean_weight <- mean(dat$weight)
```

5 Using R Commands (cont'd)

- R is case sensitive, e.g.,
  - ▷ `"sex"` is different than `"Sex"`
- Commands are separated by a semi-colon or by a newline
- Comments can be put anywhere, starting with a hashmark (#): everything to the end of the line is a comment
- Assign a value to an object by <- or =

5 Using R Commands (cont'd)

- Missing values
  - ▷ are coded as `NA` (i.e., not available) `is.na()`

    ```
    is.na(dat)
    ```

- Infinity
  - ▷ is coded as `Inf` (plus infinity) or `-Inf` (minus infinity) `is.finite()`

6 Most Frequently Used R Objects

- Main types of `modes`
  - ▷ `integer` & `numeric`: quantitative data
  - ▷ `character`: qualitative data
  - ▷ `logical`: TRUE or FALSE

    ```
    > dat$weight_65higher <- dat$weight > 65
    ```
  - ▷ `function`: see later

- There are also other types of storage modes

6 Most Frequently Used R Objects (cont'd)

patient	height	weight	sex	weight 65higher
1	1.608526	65.80858	male	TRUE
2	1.800125	63.45247	male	FALSE
3	1.694358	66.75113	female	TRUE
4	1.729665	75.95356	female	TRUE
5	1.420853	71.79690	male	TRUE
⋮				

6 Most Frequently Used R Objects (cont'd)

- In order to list the created objects use `objects()`

```
> objects()
# Now remove all objects
> rm(list=ls(all=TRUE))
> objects()
character(0)
```

- In order to investigate a specific object **dat** `str(dat)`

6 Most Frequently Used R Objects (cont'd)

- Basic arithmetics
  - ▷ `+`, `-`, `*`, `/`, `^`

    ```
    > dat$BMI <- dat$weight/(dat$height^2)
    ```

- Complicated arithmetics
  - ▷ `%*%`, `t()`, `%/%`

6 Most Frequently Used R Objects (cont'd)

- R is able to distinguish between vectors, matrices, dataframes and lists

```
> #Creating a vector consisting of 1,2, and 3
> vector1 <- c(1,2,3)
> vector1
[1] 1 2 3
> vector1 <- 1:3
> vector1
[1] 1 2 3

> #Creating a non-numeric vector
> non.num.vector <- c("A","B","C")
> non.num.vector
[1] "A" "B" "C"
```

6 Most Frequently Used R Objects (cont'd)

- R is able to distinguish between vectors, matrices, lists and dataframes

```
> #Running basic arithmetic operations
> vector1 + vector1
[1] 2 4 6
```

6 Most Frequently Used R Objects (cont'd)

- **Matrices** have the same type of elements
- **Data.frames** and **lists** do not need to have the same type of elements
- **Lists** may have elements of different length

```
> #create a matrix
> vector2 <- 1:4
> matrix(vector2, 2, 2)
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

6 Most Frequently Used R Objects (cont'd)

- **Matrices** have the same type of elements
- **Data.frames** and **lists** do not need to have the same type of elements
- **Lists** may have elements of different length

```
> #create a matrix
> vector2 <- 1:4
> matrix(vector2, 2, 2, byrow = TRUE)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

6 Most Frequently Used R Objects (cont'd)

- **Matrices** have the same type of elements
- **Data.frames** and **lists** do not need to have the same type of elements
- **Lists** may have elements of different length

```
> mylist = list(names = c("Jack","Mary"),
child.ages = c(4,7,9,10,11))
> mylist
$names
[1] "Jack" "Mary"
$child.ages
[1]  4  7  9 10 11
```

6 Most Frequently Used R Objects (cont'd)

Differences between **Matrices**, **Data.frames** and **lists**

$$\textbf{Matrix}\left\{\begin{array}{cc} \text{height} & \text{weight} \\ 1.608526 & 65.80858 \\ 1.800125 & 63.45247 \\ 1.694358 & 66.75113 \\ 1.729665 & 75.95356 \\ 1.420853 & 71.79690 \end{array}\right.$$

6 Most Frequently Used R Objects (cont'd)

Differences between **Matrices**, **Data.frames** and **lists**

Data frame

patient	height	weight	sex	weight 65higher
1	1.608526	65.80858	male	TRUE
2	1.800125	63.45247	male	FALSE
3	1.694358	66.75113	female	TRUE
4	1.729665	75.95356	female	TRUE
5	1.420853	71.79690	male	TRUE

6 Most Frequently Used R Objects (cont'd)

Differences between **Matrices**, **Data.frames** and **lists**

List

height	weight	format	#patients per gender
1.608526	65.80858	short format	female 65
1.800125	63.45247		male 47
1.694358	66.75113		
1.729665	75.95356		
1.420853	71.79690		

7 Functions

- What are functions & how to define them
  - ▷ each package has several functions
  - ▷ functions ⇒ a group of (organized) R commands

- Why to define functions
  - ▷ R is a functional language
  - ▷ organize your code: easier to reuse and distribute to others
  - ▷ formal definition of arguments
  - ▷ it protects from dangerous use of variable names

7 Functions (cont'd)

- How to use functions in packages
  - ▷ study the on-line help file `(?mean)`, especially sections
    - * Arguments
    - * Value
    - * Examples

7 Functions (cont'd)

- Some great and useful R functions for **datasets**
  - ▷ What is the number of males and females?

```
> table(dat$sex)
```

  - ▷ What is the mean weight and height?

```
> apply(cbind(dat$weight, dat$height), 2, median)
```

  - ▷ How many patients are included in the dataset?

```
> length(dat$weight)
```

  - ▷ Divide the dataset into groups

```
> spl.dat <- split(dat, dat$sex)
```

7 Functions (cont'd)

- Some great and useful R functions for **vectors** in general

  ▷ Create a sequence

  ```
  > seq(1, 10, by = 2)
  ```

  ▷ Create a vector with repeated values

  ```
  > rep(1:2, time = 2)
  ```

  ▷ Combine vectors by columns or rows

  ```
  > cbind(dat$weight, dat$height)
  > rbind(dat$weight, dat$height)
  ```

7 Functions (cont'd)

You can also create your own functions!

- ▷ Easily use the code again for different data
- ▷ Distribute to others

8 Loops and Control Flow

- Loops:
  - ▷ Repeat a statistical test or a computation
    - `for()`: loop over a sequence of values, e.g., iterations
    - `while()`: loop as long as a prespecified condition is satisfied
    - `replicate()`: replicate a number of times
- Control flow:
  - ▷ Perform a statistical test or a computation in specific cases
    - `if-else`: standard control flow
    - `ifelse()`: conditional element selection (vectorized version)
    - `switch()`: select one of a list of alternatives

8 Loops and Control Flow (cont'd)

```
> for (i in 1:10){ print(i) }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

8 Loops and Control Flow (cont'd)

```
> for (i in 1:10){
   if (i < 5) {
     print(i)
   }
}
[1] 1
[1] 2
[1] 3
[1] 4
```

8 Loops and Control Flow (cont'd)

- Try to vectorize (instead of for-loop)
  - ▷ **for-loop**

```
> x <- rnorm(1000000,10,10)
> cSum <- 0
> for (i in 1:length(x)){
    cSum <- cSum + x[i]
    cSum
  }
```

  - ▷ **vectorize**

```
> cumsum(x)
```

8 Loops and Control Flow (cont'd)

- Try to vectorize (instead of for-loop)
  - ▷ **for-loop**

    ```
    > cSum <- 0
    > for (i in 1:length(x)){
        cSum <- cSum + x[i]
        cSum
      }
    ```

    0.06 seconds
  - ▷ **vectorize**

    ```
    > cumsum(x)
    ```

    0.02 seconds

Part II

1. Markdown

- R Markdown is a format for writing reproducible, dynamic reports with R
- Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more

1. Markdown (cont'd)

- In **Rstudio**

  - ▷ File → New File → R Markdown...
  - ▷ Insert title and author
  - ▷ Select format

1. Markdown (cont'd)

- **Writing-part**

  Header

  ▷ Input

```
# Methods
## Data collection
## Statistical analysis
# Results
```

  ▷ Output

  Methods

  Data collection

  Statistical analysis

  Results

1. Markdown (cont'd)

- **Writing-part**

  Emphasis

  ▷ Input

  ```
  *Methods*
  _Methods_
  **Methods**
  __Methods__
  ```

  ▷ Output

  Methods
  Methods
  Methods
  Methods

1. Markdown (cont'd)

- **Writing-part**

  Bullets

  ▷ Input

```
* Method 1
- Method 2
+ Method 3
```

  ▷ Output

  - Method 1
  - Method 2
  - Method 3

1. Markdown (cont'd)

- **Writing-part**

  Bullets

  ▷ Input

```
* Method 1
- Method 2
    + Method 3
```

  ▷ Output

  - Method 1
  - Method 2
    - Method 3

1. Markdown (cont'd)

- **Writing-part**

  Links

  - ▷ Input

    ```
    [link](http://example.com)
    ```

  - ▷ Output

    link

1. Markdown (cont'd)

- **Writing-part**

  Images

  ▷ Input

  ```
  ![Rsymbol](https://
  upload.wikimedia.org/
  wikipedia/commons/1/12/
  R_logo_2000.png)
  ```

  ▷ Output

  Guess???

1. Markdown (cont'd)

- **Writing-part**

  Images

  - ▷ Input

    ```
    ![Rsymbol](https://
    upload.wikimedia.org/
    wikipedia/commons/1/12/
    R_logo_2000.png)
    { width=10% }
    ```

  - ▷ Output

    Guess???

1. Markdown (cont'd)

- **Writing-part**

  Horizontal rules

  ▷ Input

  ```
  ---
  ```

  ▷ Output

  ---

1. Markdown (cont'd)

- **Writing-part**

  Escaping

  ▷ Input

  ```
  \*Method\*
  ```

  ▷ Output

  Method

Escaping Markdown characters with a back-slash allows you to use any characters which might be getting accidentally converted into HTML.

1. Markdown (cont'd)

- **Writing-part**

  Highlights

  - ▷ Input

    ```
    `Methods`
    ```

  - ▷ Output

    Methods

Check the difference between pdf, word and html.

1. Markdown (cont'd)

- **R-part (Global options chunks)**

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
~~~

1. Markdown (cont'd)

- **R-part (Global options chunks)**

  ▷ eval: If FALSE, knitr will not run the code in the code chunk.

  ▷ include: If FALSE, knitr will run the chunk but not include the chunk in the final document.

1. Markdown (cont'd)

- **R-part (Global options chunks)**
  - ▷ echo: If FALSE, knitr will not display the code in the code chunk above its results in the final document.
  - ▷ results: If 'hide', knitr will not display the codes results in the final document. If 'hold', knitr will delay displaying all output pieces until the end of the chunk. If 'asis', knitr will pass through results without reformatting them (useful if results return raw HTML, etc.)
  - ▷ error: If FALSE, knitr will not display any error messages generated by the code.
  - ▷ message: If FALSE, knitr will not display any messages generated by the code.
  - ▷ warning: If FALSE, knitr will not display any warning messages generated by the code.

1. Markdown (cont'd)

- **R-part (Global options chunks)**

  ▷ cache: If TRUE, knitr will cache the results to reuse in future knits. Knitr will reuse the results until the code chunk is altered.

1. Markdown (cont'd)

- **R-part (Global options chunks)**

  - ▷ fig.cap: A character string to be used as a figure caption in LaTex.
  - ▷ fig.height, fig.width: The width and height to use in R for plots created by the chunk (in inches).

1. Markdown (cont'd)

- **R-part (Global options chunks)**

  ▷ . . .
  https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf

2. Shiny app

- Shiny is an R package that makes it easy to build interactive web apps straight from R
- Shiny combines the computational power of R with the interactivity of the modern web
- Shiny apps are easy to write - No web development skills are required

2. Shiny app (cont'd)

- In **Rstudio**
  - ▷ File → New File → Shiny Web App...
  - ▷ Insert application name

2. Shiny app (cont'd)

- **Example I**

```
> library(shiny)
> runExample("01_hello")
```

2. Shiny app (cont'd)

- **Shiny structure**

  ▷ **a user interface object (ui)**

  ▷ **a server function**

  ▷ **a call to the shinyApp function**

2. Shiny app (cont'd)

- **Shiny structure**

  ▷ **a user interface object (ui)**: layout and appearance of your app

  ▷ **a server function**: instructions that your computer needs to build your app

  ▷ **a call to the shinyApp function**: creates Shiny app objects

2. Shiny app (cont'd)

- **Shiny structure**

```
> library(shiny)

# Define UI ----
> ui <- fluidPage(

)

# Define server logic ----
> server <- function(input, output) {

}

# Run the app ----
> shinyApp(ui = ui, server = server)
```

Thank you!

e.andrinopoulou@erasmusmc.nl

https://github.com/erandrinopoulou

@ERandrinopoulou