



DATA ANALYSIS

WITH R AND THE TIDYVERSE

Slides by:

JEAN FONTAINE, MAX SPRANG and STEFFEN ALBRECHT

Support by:

MERT CIHAN



2 LET'S QUICKLY CHECK THE INSTALLATION

- We'll work with **Rstudio**
- We'll need the two packages “**tidyverse**” and “**readxl**”
- Run the following in the Rstudio **Console**
 - `install.packages("tidyverse")`
 - `install.packages("readxl")`



3 THE COURSE

- Objectives
 - Powerful data analysis and visualization from little programming skills
- R and the tidyverse
 - R: core statistical software
 - Tidyverse: extensions to R + more readable code
- Content
 - Brief overview of the basics (this document)
 - Tidyverse Tutorial
 - Extra exercises



4 WHY R???

- It is frequently used in Life Sciences
 - e.g. the **Bioconductor** project for
 - Molecular Biology
 - Next-generation sequencing data
 - And much more...
 - <https://www.bioconductor.org/>

5

BRIEF OVERVIEW OF THE BASICS IN R



6 R STUDIO INTERFACE

The screenshot displays the RStudio interface with several key components highlighted by red boxes:

- Source code or data tables:** A red box highlights the main workspace area, which contains a data table with 49 observations of 4 variables (bmi, age, male, exmin).
- Variables:** A red box highlights the Environment pane, which lists the variables in the current environment, including 'Global Environment', 'Data', and 'FUNCTIONS'.
- R console:** A red box highlights the Console pane, which displays the R startup message and the prompt '> |'.
- Files, plots or help (FI):** A red box highlights the Files pane, which shows the file explorer view of the current workspace, including folders like '.RData', '.Rhistory', and files like 'Confounding and Effect Measure Modification_AL.docx'.

The data table in the workspace is as follows:

	bmi	age	male	exmin
1	24	18	0	360
2	25	44	1	250
3	24	32	1	350
4	25	34	1	360
5	24	22	0	280
6	25	28	0	200
7	24	66	1	150
8	25	34	0	377
9	24	44	1	510
10	25	36	0	432
11	24	50	0	200
12	25	55	0	250
13	24	30	0	450
14	25	34	0	200
15	24	21	1	457

The Console output shows the following text:

```
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

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.

[Workspace loaded from ~/.RData]
> |
```

7 TIPS

- The # symbol indicates to R that what follows is a comment
- Commands are separated by a new line
 - Lists, piped commands and content within parentheses or brackets can be spread over different lines
- Parentheses and brackets must be closed
- R is case sensitive

```
# LOAD LIBRARY
library(tidyverse)

# LOAD AND DISPLAY TABLE
column_names <- c("probe_id", "expression", "present")

cd103minus <- read_tsv(
  "file.tsv",
  col_names=column_names
)

cd103minus %>%
  select(probe_id, expression) %>%
  head()
```

8 TIPS

- Some keyboard shortcuts
 - **[TAB]** for text auto-completion of function or object names
 - **[CTRL]+[ENTER]** to run the selected lines from the code editor to the console
 - **[ALT]+[-]** to insert assignment symbols (<-)
 - **[CTRL]+[SHIFT]+[m]** to insert the pipe symbols (%>%)
 - In the console, use arrow keys to traverse through the history of commands
 - "Up arrow" – traverse backwards (older commands)
 - "Down arrow" – traverse forward (newer commands)

9 AS CALCULATOR

Symbol	Meaning
+	add
-	subtract
*	multiply
/	divide
^	power (e.g. 2^3 is equal to 8)

- Simple operations
 - $1+3$ # will return 4
- Statistical functions
 - `sqrt(81)` # will the square root of 81 that is 9
 - $1 + \text{abs}(-4)$ # will return absolute value of -4 plus 1 = 5
- The Order of Operations
 - Do calculations inside Parentheses first
 - $6 \times (5 + 3) = 6 \times 8 = 48$
 - Then compute Exponents (Powers, Roots) before multiplication and division
 - $5 \times 2^2 = 5 \times 4 = 20$
 - Then Multiply or Divide (before you Add or Subtract)
 - $2 + 5 \times 3 = 2 + 15 = 17$
 - Otherwise just go left to right

10 OBJECTS AND VARIABLES

- R stores everything in objects having defined types
- Some common object types
 - numeric (e.g. 1, 25.5, 1e-6)
 - character (e.g. "ABCD", "Hello World 24!")
 - logical (TRUE or T, FALSE or F, NA for not applicable)
 - factor: categorical values (numeric index associated to character labels)
 - vector: used to store a set of objects
 - function: (e.g. library, abs, sqrt)
- Variables
 - Names to remember and reuse objects
 - `x <- 2 # variable x is assigned value 2`
- The class function returns the type
 - `class(x) # R returns "numeric"`
 - `class("ABC") # R returns "character"`

II VECTORS

```
# Vector of 4 numeric objects
x <- c(1.2, # 1st object
      2.3, # 2nd object
      0.2, # 3rd object
      1.1) # 4th object

# Indexing some values
X          # 1.2 2.3 0.2 1.1
x[1]      # 1.2
x[ length(x) ] # 1.1
x[3]      # 0.2
x[ c(2,3,4) ] # 2.3 0.2 1.1
x[ 2:4 ]   # 2.3 0.2 1.1
```

12 TABLES

- Several types of tables
- matrix - table of objects of the same type
- data.frame - table of objects of same or different types
- tibble – extension of data.frame for working with large tables
 - tibbles have a refined print method (default few rows and columns fit screen)
 - each column reports its type
 - numerics may be detailed as integers (int) or double precision values (dbl)

13 TABLES

TABLE [ROW, COL]

TABLE\$COL[ROW]

```
df <- data_frame(
  x = 1:5, y = 1, z = x^2+y
)
```

```
df
```

```
# A tibble: 5 x 3
```

	x	y	z
	<int>	<dbl>	<dbl>
1	1	1	2
2	2	1	5
3	3	1	10
4	4	1	17
5	5	1	26

```
df[1,1] # 1
```

```
df[1,2] # 1
```

```
df[1,3] # 2
```

```
df[1,] # 1 1 2
```

```
df[1,3] # 2
```

```
df[2,3] # 5
```

```
df[,3] # 2 5 10 17 26
```

```
df$z # 2 5 10 17 26
```

```
df[ 4:5 , 1:2 ]
```

```
# A tibble: 2 x 2
```

	x	y
	<int>	<dbl>
1	4	1
2	5	1

14 FOR LOOPS

```
# Vector of character objects
x <- c("file1.csv", # 1st object
      "file2.csv", # 2nd object
      "file3.csv") # 3rd object

# Looping on some values
for(my_file in x){
  text <- paste("We have", my_file)
  print(text)
}

[1] "We have file1.csv"
[1] "We have file2.csv"
[1] "We have file3.csv"
```

15 LOGICAL OPERATIONS

```

1<2                # TRUE

!(1<3)            # FALSE (logical NOT: !)

1 != 3            # TRUE

(3 != 1) & (2 >= 1.9) # TRUE (logical AND: &)

(3 == 1) | (3 < 5) # TRUE (logical OR: |)

y <- c(TRUE, FALSE, 5>2)

Y                # TRUE FALSE TRUE

```

Symbol	Meaning
==	logical equals
!=	not equal
!	logical NOT
&	logical AND
	logical OR
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

Command	Result
TRUE	TRUE
FALSE	FALSE
!TRUE	FALSE
!FALSE	TRUE
TRUE & TRUE	TRUE
TRUE & FALSE	FALSE
FALSE & FALSE	FALSE
TRUE TRUE	TRUE
TRUE FALSE	TRUE
FALSE FALSE	FALSE

For more
details see
help pages

16 SOME USEFUL FUNCTIONS

- Table statistics
 - `rownames(x)`
 - `colnames(x)`
 - `dim(x)` – number of rows and columns
 - `summary(x)` – summary statistics
- Table join / merge
 - `left_join()`
 - `full_join()`
 - `rbind()` – add rows (base R)
 - `cbind()` – add columns (base R)
- Statistics
 - `length(x)`
 - `max(x)`
 - `min(x)`
 - `sum(x)`
 - `mean(x)`
 - `median(x)`
 - `var(x)`
 - `sd(x)`
 - `cor()` – correlation coefficient
 - `t.test()` – Student's t-test
- Other
 - `seq()` – sequence of numbers

17

R TUTORIAL

LINK TO TUTORIAL AND DATA FILES AVAILABLE AT:

[HTTPS://CBDM.UNI-MAINZ.DE/MB22B/](https://cbdm.uni-mainz.de/mb22b/)

18 THE TUTORIAL WAS CREATED AND EXTENDED BY:



**Gregorio
Alanís Lobato**



**Jean-Fred
Fontaine**



**Maximilian
Sprang**



**Steffen
Albrecht**

