Introduction to R

23.11. – 25.11.2020

Christiane Hassenrück (chassenrueck@marum.de)

Course overview

Monday,
23.11.2020

- General background
- Obtaining R and R studio
- R data and object types
- Data table organization, reading data into R
- Understanding errors
- Good coding practice, sustainable and collaborative coding (git and github)

Tuesday,
24.11.2020

- Data manipulation in base R: simple calculations, summaries, loops, functions
- Data manipulation in the R tidyverse
- Which manipulations do you need to perform with your data?

Wednesday,
25.11.2020

- Data visualization in R: par(), layout(), plot()
- Creating (interpolated) maps in R: *OceanView*, *marmap*, *sf*
- Open questions?
- Resources online and in Bremen

Course aim: To enable you to find solutions to your R problems independently!

Statistical computing software

	Graphical user interface (GUI)	Command line
Commercial	sas SIGMAPLOT Exact Graphs and Data Analysis SPSS StatSoft STATISTICA PRIMER-E jmp	S 8 Insightful MATLAB
Free	PAST	RStudio R

Why R?

“R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.” (https://www.r-project.org/)

Pro’s	Con’s
• Versatile	• Command-line
• Platform-independent	• Needs some getting used to…
• Data exploration and visualization	• Won’t always tell you what to do…
• Hypothesis testing	
• Advanced graphics	
• Large data sets	
• Reproducible	
• Open source	
• Good documentation and online support	
… many more	

We are going to do something about that!

Why R?

“R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.” (https://www.r-project.org/)

R console:

CRAN Mirrors
(https://cran.r-project.org)

Download R and R packages:

(https://www.bioconductor.org)

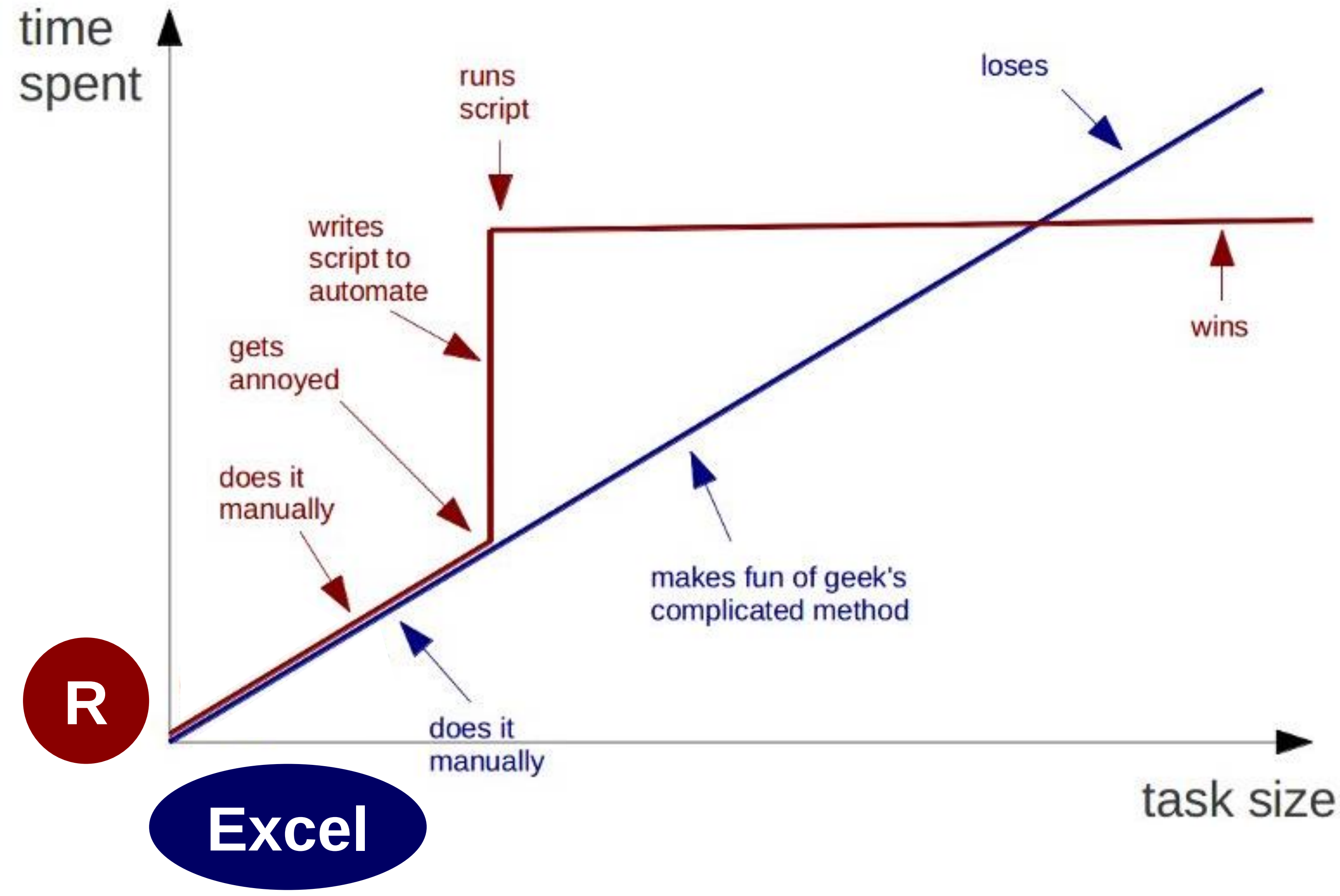
(https://github.com)

More user-friendly implementation:

R markdown
(http://rmarkdown.rstudio.com/)

Online help: https://stackoverflow.com/
https://stats.stackexchange.com/

Why R?



http://www.howtogeek.com/102420/geeks-versus-non-geeks-when-doing-repetitive-tasks-funny-chart/

Let‘s get started…

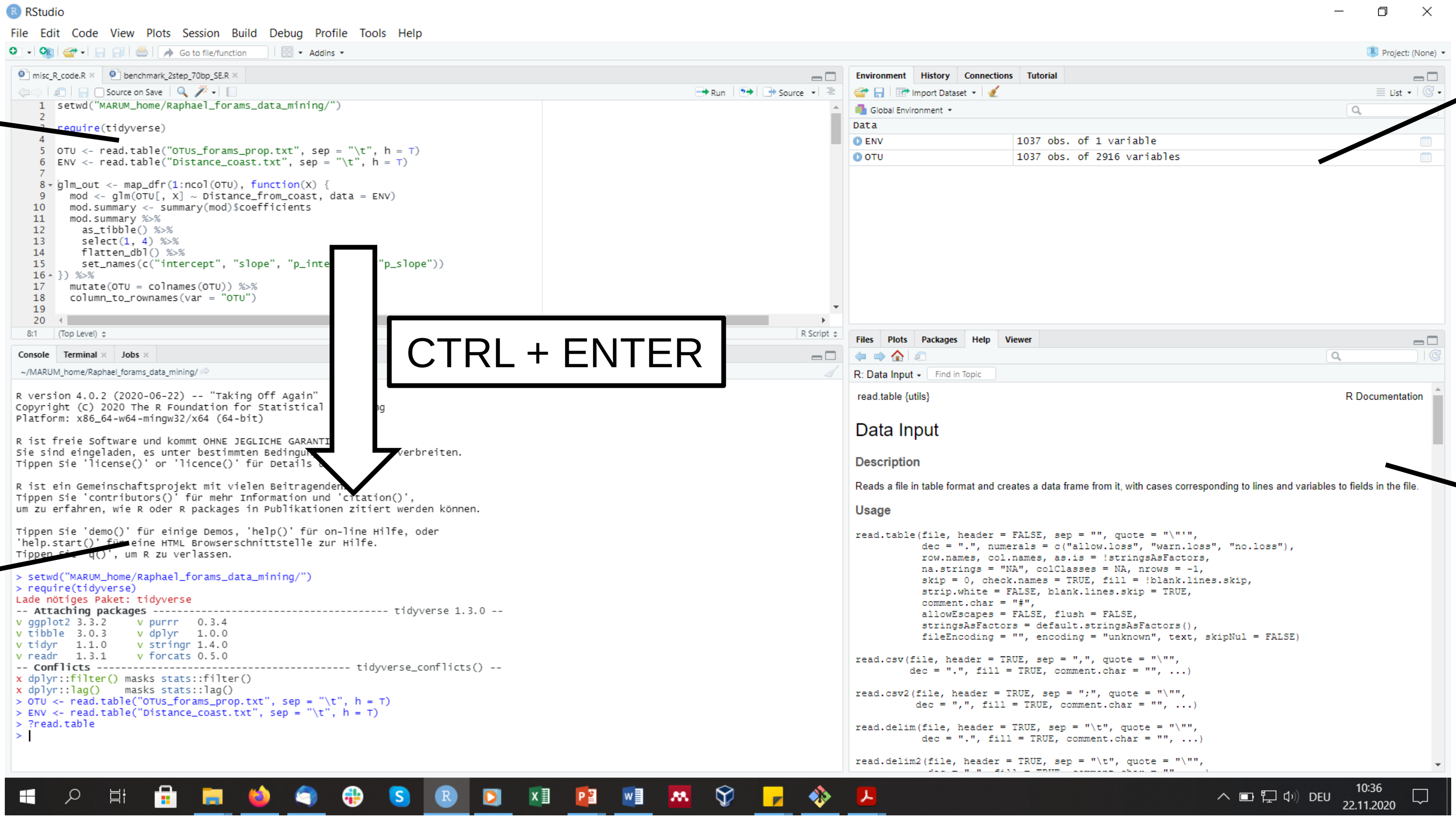
Download and install R: https://cloud.r-project.org/

Download and install Rstudio: https://rstudio.com/products/rstudio/download/

Let‘s get started…

R script:
Writing and
documentation

Workspace:
R objects (data
and functions)



```r
setwd("MARUM_home/Raphael_forams_data_mining/")

require(tidyverse)

OTU <- read.table("OTUs_forams_prop.txt", sep = "\t", h = T)
ENV <- read.table("Distance_coast.txt", sep = "\t", h = T)

glm_out <- map_dfr(1:ncol(OTU), function(x) {
  mod <- glm(OTU[, x] ~ Distance_from_coast, data = ENV)
  mod.summary <- summary(mod)$coefficients
  mod.summary %>%
    as_tibble() %>%
    select(1, 4) %>%
    flatten_dbl() %>%
    set_names(c("intercept", "slope", "p_inte    "p_slope"))
}) %>%
  mutate(OTU = colnames(OTU)) %>%
  column_to_rownames(var = "OTU")
```

Data

ENV	1037 obs. of 1 variable
OTU	1037 obs. of 2916 variables

~/MARUM_home/Raphael_forams_data_mining/

```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical          g
Platform: x86_64-w64-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTI
Sie sind eingeladen, es unter bestimmten Bedingun              verbreiten.
Tippen Sie 'license()' or 'licence()' für Details 

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> setwd("MARUM_home/Raphael_forams_data_mining/")
> require(tidyverse)
Lade nötiges Paket: tidyverse
-- Attaching packages --------------------------------------- tidyverse 1.3.0 --
v ggplot2 3.3.2     v purrr   0.3.4
v tibble  3.0.3     v dplyr   1.0.0
v tidyr   1.1.0     v stringr 1.4.0
v readr   1.3.1     v forcats 0.5.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
> OTU <- read.table("OTUs_forams_prop.txt", sep = "\t", h = T)
> ENV <- read.table("Distance_coast.txt", sep = "\t", h = T)
> ?read.table
> 
```

read.table {utils} R Documentation

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
```

CTRL + ENTER

Console:
Execution and
(error) messages

Various:
File browser
Plotting panel
Help
Packages
…

R data and object types

		R
Numerical		
	Continuous (weight, temperature, length)	`numeric`
	Discrete (species counts)	`numeric, integer`
Categorical		
	Ordinal (categories of increasing impact)	`ordered factor`
	Nominal (unweighted treatments)	`factor, character`
Character strings (names)		`character`

R data and object types

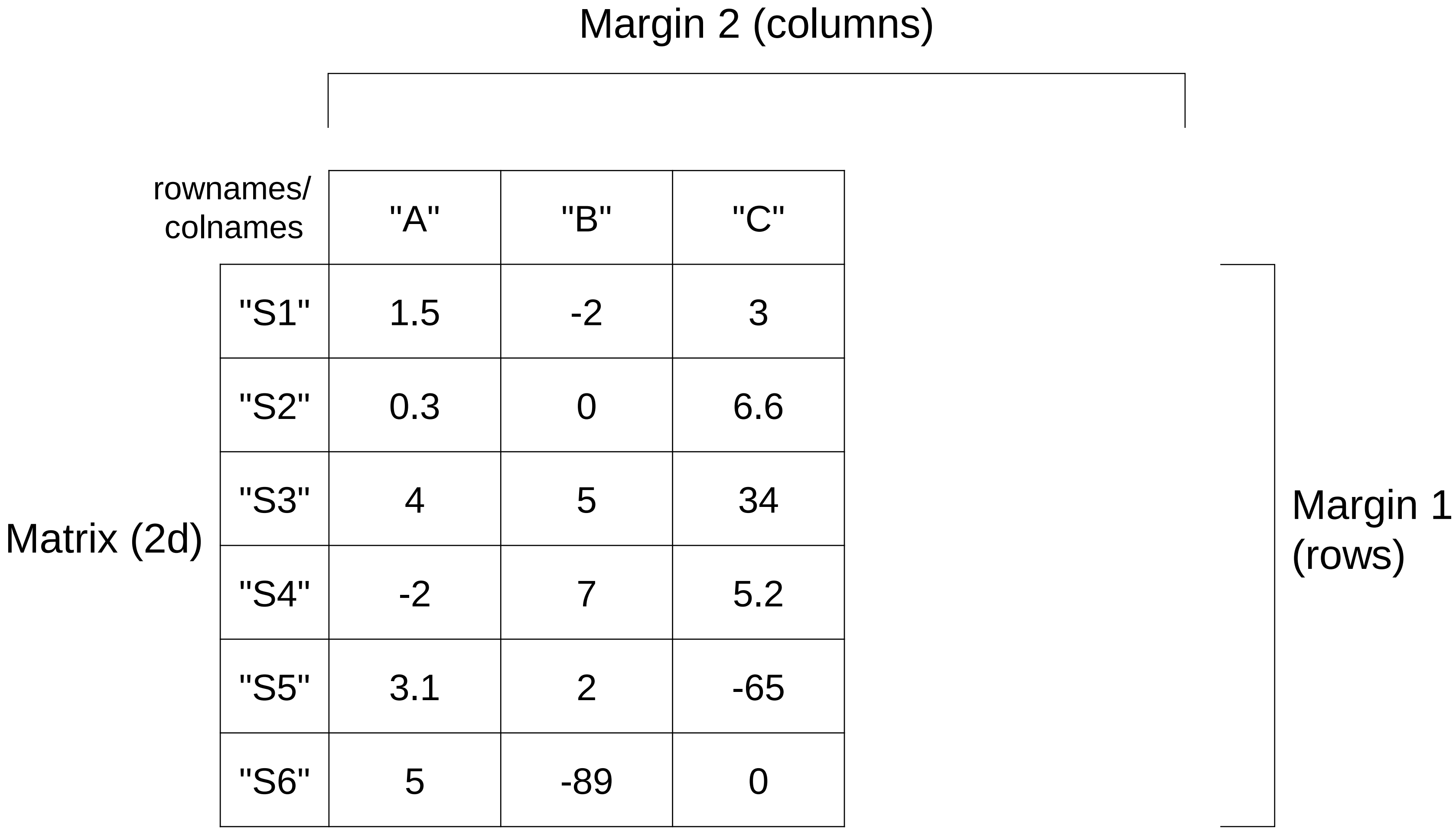
	What we are used to	What R understands
`numeric`	-2, 0.5, 1.452345, 50	-2, 0.5, 1.452345, 50
`integer`	-3, 2, 0, 6, 3, 54	-3, 2, 0, 6, 3, 54
`character`	sampleA, sampleB exp1, exp2	"sampleA", "sampleB" "exp1", "exp2"
`factor`	exp1, exp2	1, 2 Levels: "exp1", "exp2"
`ordered factor`	exp1, exp2	1, 2 Levels: "exp1" < "exp2"
`logical`	TRUE, FALSE	TRUE, FALSE 1, 0

R data and object types

Vector (1d)

"A"
1.5
0.3
4
-2
3.1
5

R data and object types



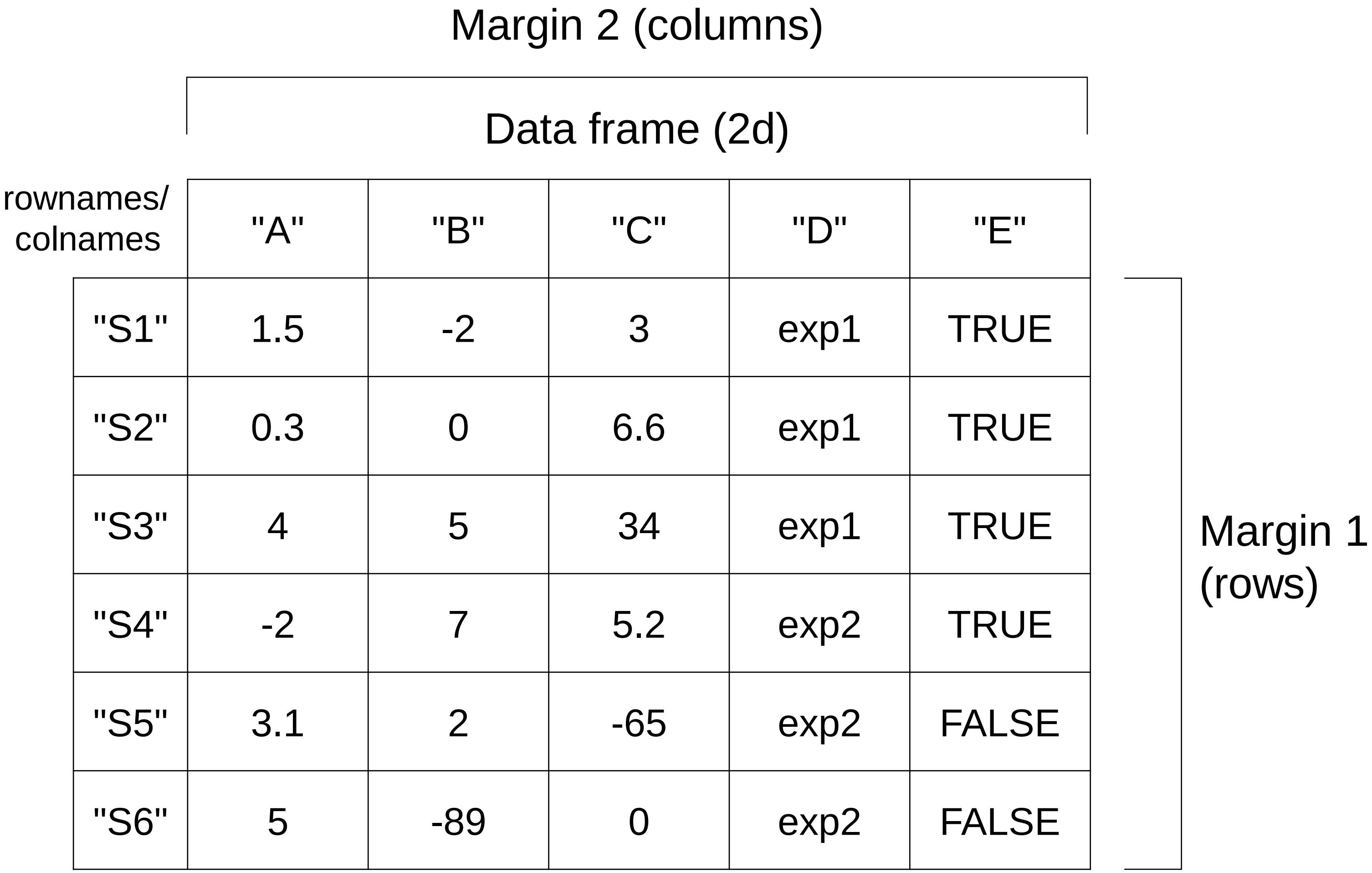
Margin 2 (columns)

rownames/ colnames	"A"	"B"	"C"
"S1"	1.5	-2	3
"S2"	0.3	0	6.6
"S3"	4	5	34
"S4"	-2	7	5.2
"S5"	3.1	2	-65
"S6"	5	-89	0

Margin 1 (rows)

Matrix (2d)

R data and object types



Margin 2 (columns)

Data frame (2d)

rownames/ colnames	"A"	"B"	"C"	"D"	"E"
"S1"	1.5	-2	3	exp1	TRUE
"S2"	0.3	0	6.6	exp1	TRUE
"S3"	4	5	34	exp1	TRUE
"S4"	-2	7	5.2	exp2	TRUE
"S5"	3.1	2	-65	exp2	FALSE
"S6"	5	-89	0	exp2	FALSE

Margin 1
(rows)

R data and object types

List (1d): vector of R objects

	"A"	"B"	"C"	"D"	"E"
"S1"	1.5	-2	3	exp1	TRUE
"S2"	0.3	0	6.6	exp1	TRUE
"S3"	4	5	34	exp1	TRUE
"S4"	-2	7	5.2	exp2	TRUE
"S5"	3.1	2	-65	exp2	FALSE
"S6"	5	-89	0	exp2	FALSE

data.frame

	"A"	"B"	"C"	"D"	"E"
"S1"	1.5	-2	3	exp1	TRUE
"S2"	0.3	0	6.6	exp1	TRUE
"S3"	4	5	34	exp1	TRUE
"S4"	-2	7	5.2	exp2	TRUE
"S5"	3.1	2	-65	exp2	FALSE
"S6"	5	-89	0	exp2	FALSE

vector

"A"	1.5	0.3	4	-2	3.1

value

1.5

matrix

	"A"	"B"	"C"
"S1"	1.5	-2	3
"S2"	0.3	0	6.6
"S3"	4	5	34
"S4"	-2	7	5.2
"S5"	3.1	2	-65
"S6"	5	-89	0

Let‘s move to R…

Exercise 1: R data and object types

Collect unfamiliar commands!

Collect your error messages!

Data table organization

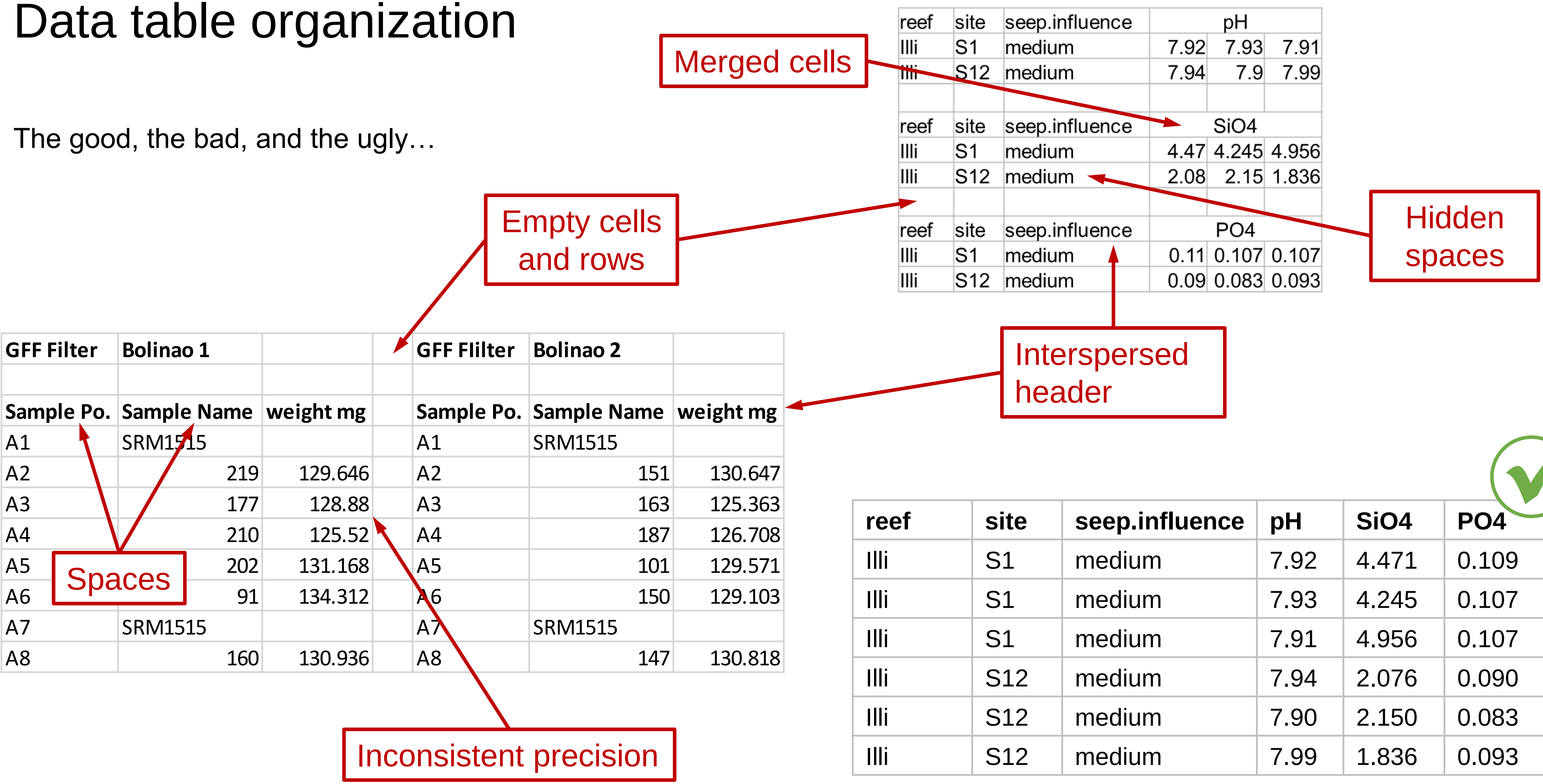
- Most common input format for tabular data:
  - .txt
  - .csv
  - .tsv

- Include variable names in first row (header)
- Don’t start row or column names with numbers
- Values are usually tab, space, or comma separated
- Avoid special characters and spaces in data values, variable names, and file names

	Bad	Good
Variable name	mean temperature mean-temperature	temperature.mean temperature_mean
Data value	day 1	day1 1 (variable name: day)

Data table organization

The good, the bad, and the ugly…



reef	site	seep.influence	pH		
Illi	S1	medium	7.92	7.93	7.91
Illi	S12	medium	7.94	7.9	7.99
reef	site	seep.influence	SiO4		
Illi	S1	medium	4.47	4.245	4.956
Illi	S12	medium	2.08	2.15	1.836
reef	site	seep.influence	PO4		
Illi	S1	medium	0.11	0.107	0.107
Illi	S12	medium	0.09	0.083	0.093

Merged cells

Hidden spaces

Empty cells and rows

Interspersed header

GFF Filter	Bolinao 1			GFF Fliilter	Bolinao 2	
Sample Po.	Sample Name	weight mg		Sample Po.	Sample Name	weight mg
A1	SRM1515			A1	SRM1515	
A2	219	129.646		A2	151	130.647
A3	177	128.88		A3	163	125.363
A4	210	125.52		A4	187	126.708
A5	202	131.168		A5	101	129.571
A6	91	134.312		A6	150	129.103
A7	SRM1515			A7	SRM1515	
A8	160	130.936		A8	147	130.818

Spaces

Inconsistent precision

reef	site	seep.influence	pH	SiO4	PO4
Illi	S1	medium	7.92	4.471	0.109
Illi	S1	medium	7.93	4.245	0.107
Illi	S1	medium	7.91	4.956	0.107
Illi	S12	medium	7.94	2.076	0.090
Illi	S12	medium	7.90	2.150	0.083
Illi	S12	medium	7.99	1.836	0.093

✓

Data table organization

Long data format:

- One data value per line per measurement variable
- Additional comlums with contextual data (usually categories)

reef	site	seep.influence	measurement	value
Illi	1	medium	pH	7.92
Illi	1	medium	pH	7.93
Illi	1	medium	pH	7.91
Illi	12	medium	pH	7.94
Illi	12	medium	pH	7.90
Illi	12	medium	pH	7.99
Illi	1	medium	SiO4	4.471
Illi	1	medium	SiO4	4.245
Illi	1	medium	SiO4	4.956
Illi	12	medium	SiO4	2.076
Illi	12	medium	SiO4	2.150
Illi	12	medium	SiO4	1.836
Illi	1	medium	PO4	0.109
Illi	1	medium	PO4	0.107

Wide data format:

- More easily readable
- Values either rearrangement of or summaries calculated from long data format

Original data - rearranged

reef	site	seep.influence	pH	SiO4	PO4
Illi	S1	medium	7.92	4.471	0.109
Illi	S1	medium	7.93	4.245	0.107
Illi	S1	medium	7.91	4.956	0.107
Illi	S12	medium	7.94	2.076	0.090
Illi	S12	medium	7.90	2.150	0.083
Illi	S12	medium	7.99	1.836	0.093

Mean values

reef	site	seep.influence	pH	SiO4	PO4
Illi	S1	medium	7.92	4.539	0.108
Illi	S12	medium	7.94	2.021	0.089

Let‘s move to R…

Exercise 2: Reading data into R
Example data set from shallow hydrothermal vents in Papua New Guinea

Optional: Read your own data into R (homework)

Collect your error messages!

R errors

Syntax errors

- When R doesn't understand you, because the command doesn't make sense…
- R returns an error message
- Majority of errors
- E.g.: Trying to calculate the mean of categorical data, typos

Semantic errors

- When R doesn't do what you want, although the command makes sense…
- R will not return an error message, because the command is valid
- More dangerous errors
- E.g.: Calculating percentages over columns, and not rows

Google is your new best friend ☺

Sustainable and collaborative coding

Project directory organization:

∟ Input (raw) data → Data archiving:

∟ Intermediate (temporary) files

∟ Final results (plots, tables, etc.)

∟ Scripts

PANGAEA.

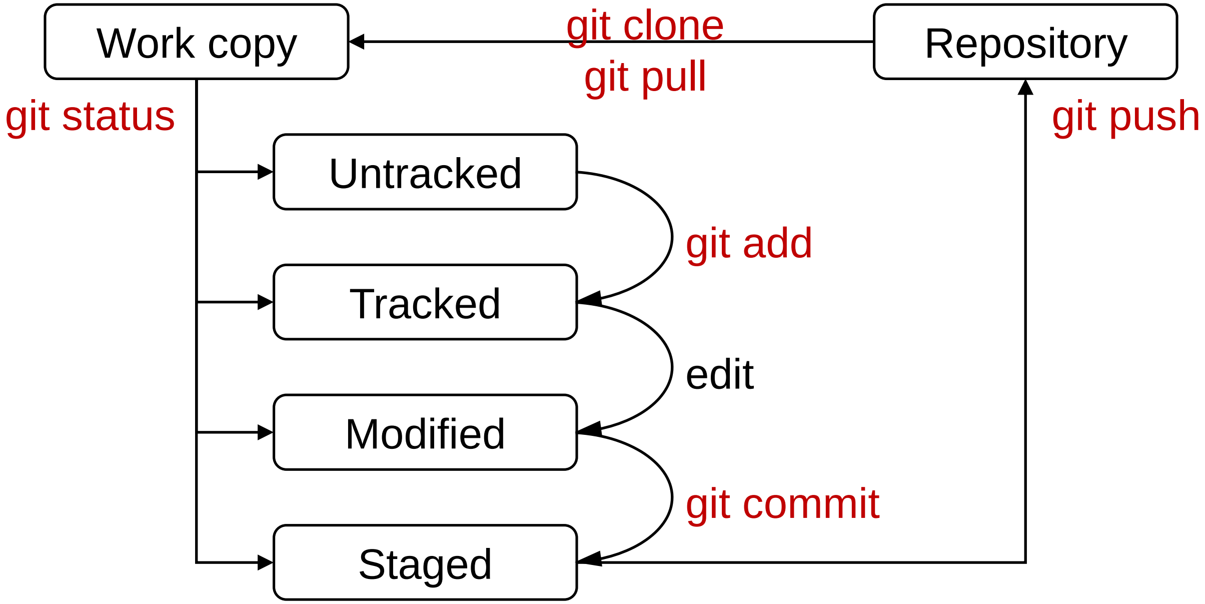
gfbio GERMAN FEDERATION FOR BIOLOGICAL DATA

MetaboLights

ENA European Nucleotide Archive

GitHub

(https://github.com)



Git plug-in for Rstudio: https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

Plot()-ing in R

Configure your plotting area:

- par()
- layout()
- plot()

Plotting elements:

- points()
- segments()
- lines()
- rect()
- polygon()
- image()
- etc.

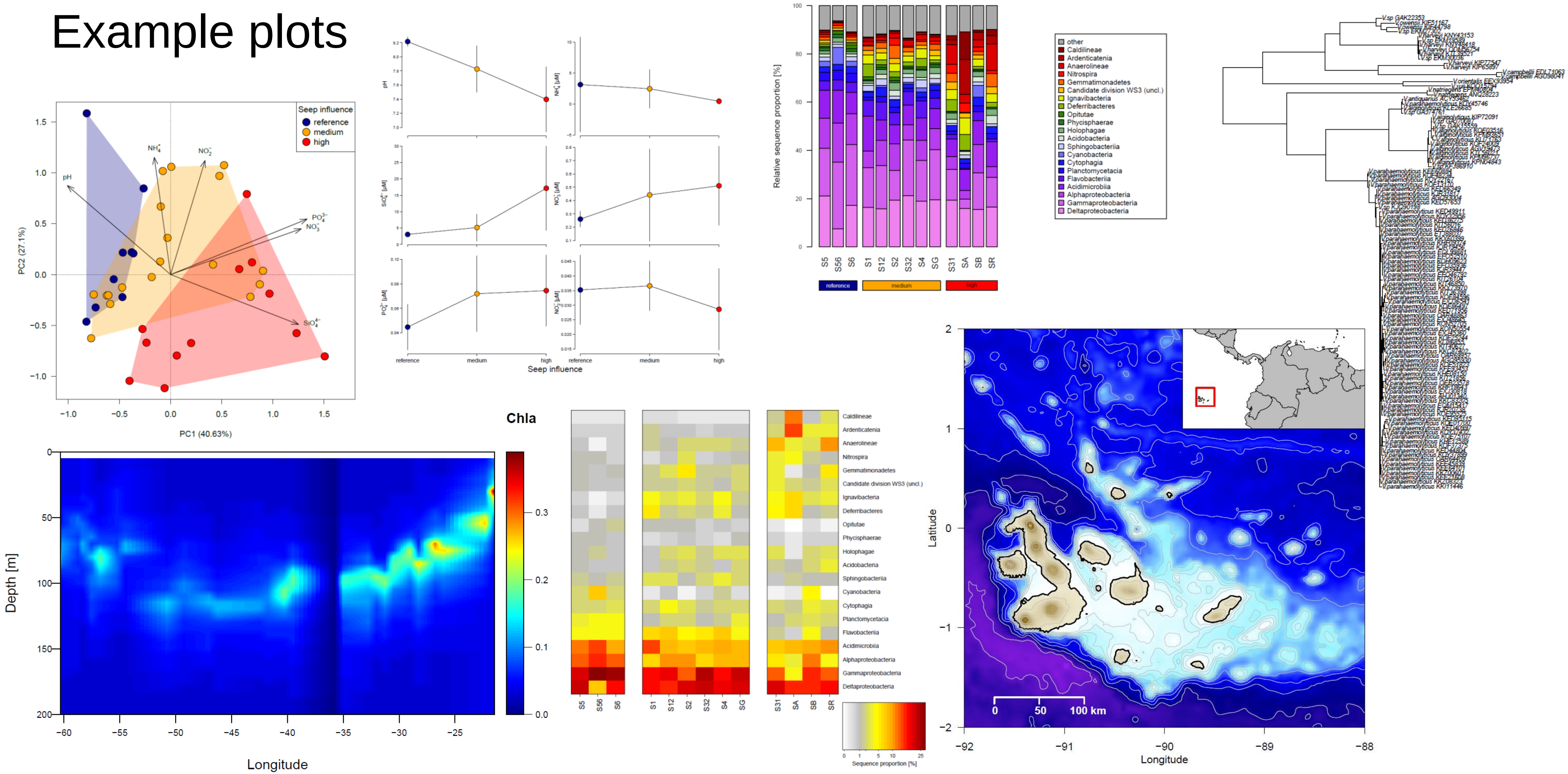
Packages for creating maps in R:

- maps
- OceanView
- marmap
- sp, sf (more advanced spatial analysis)

Most time-consuming step when plotting:

Getting data into shape!

Example plots



Further resources

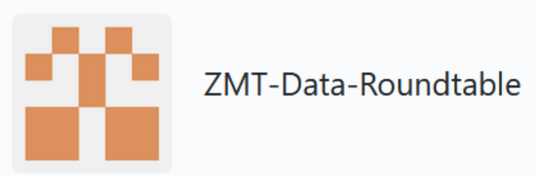
Tidyverse tutorials: https://www.tidyverse.org/

Stackexchange: https://stats.stackexchange.com/

Stackoverflow: https://stackoverflow.com/

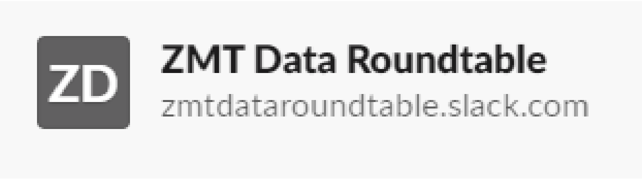
ZMT data roundtable: https://www.leibniz-zmt.de/de/neuigkeiten/veranstaltungen/data-round-table.html

GitHub



ZMT-Data-Roundtable

slack



ZMT Data Roundtable
zmtdataroundtable.slack.com

Contact: tobias.poprick@leibniz-zmt.de, arjun.chennu@leibniz-zmt.de