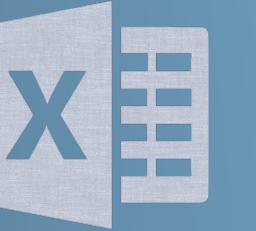


FROM  EXCEL TO 



WHO ARE WE?

Center for Health Data Science (HeaDS) - <https://heads.ku.dk/>

SUND Center, which includes a KU data lab

- **Consultation & Collaboration:**
 - Data science and bioinformatics analyses, e.g. big data, -omics analysis, machine learning.
- **Teaching; Courses & Workshops, Seminars, etc.**

Data Science Laboratory (DSL) - <https://datalab.science.ku.dk/>

Dep. of Math and Computer Science, Faculty of SCIENCE



Thilde Terkelsen



Diana Andrejeva



Tugce Karaderi



Henrike Zschach



Bo Markussen



Helle Sørensen



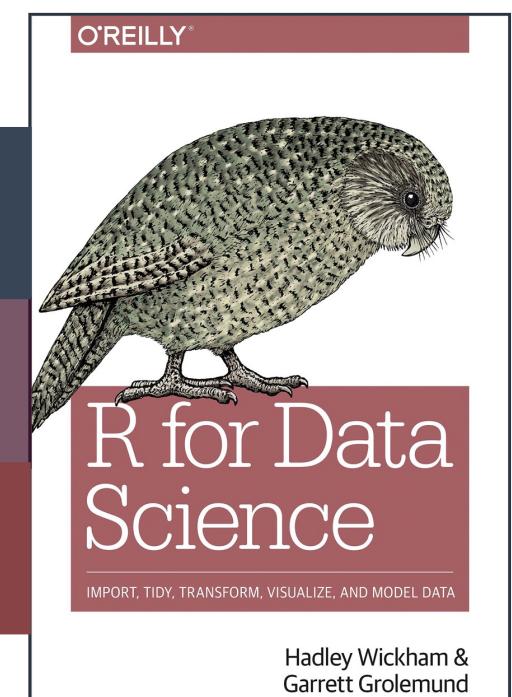
THE PRACTICALS



Two days: 9.00-16.30. There will be coffee breaks, we promise ☕

“R for Data Science” - a generally useful book on R, also for this course

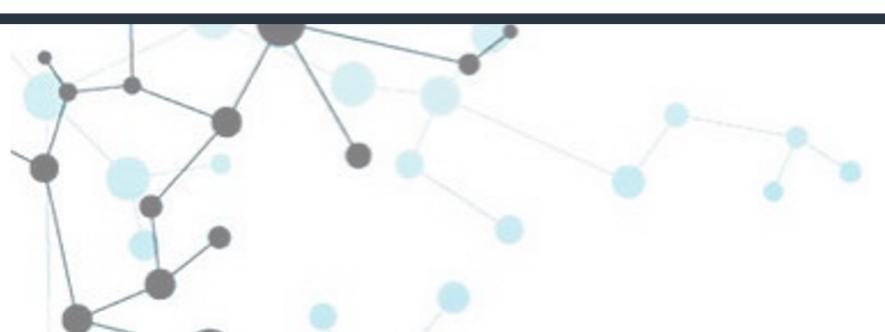
The course is build on hands-on presentations (.R, .Rmd) & exercises



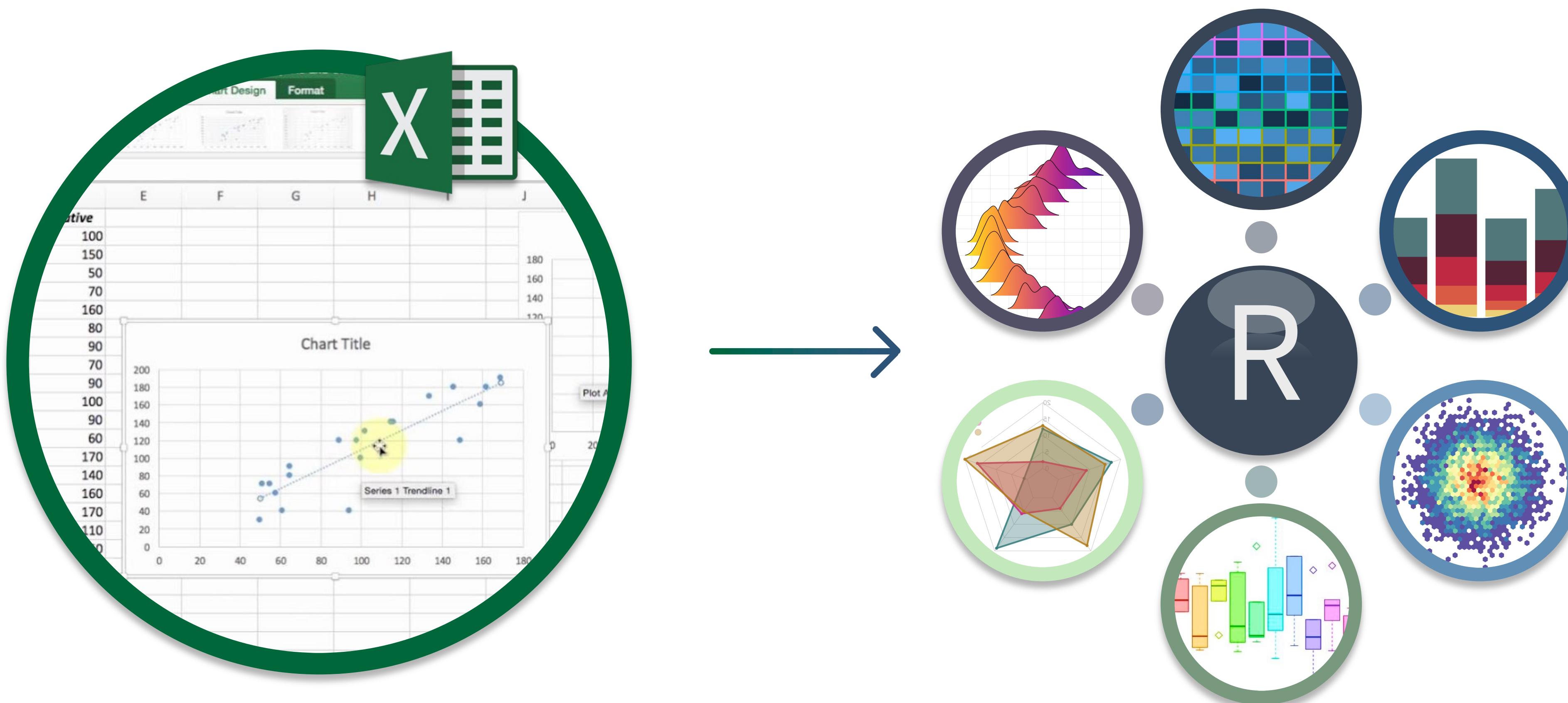
→ Download and install the newest version of R (<https://cran.r-project.org/>)

→ Download and install the newest version of R-studio (<http://www.rstudio.com/download>)

→ Download the course material and place it somewhere you can find it again!
<https://github.com/Center-for-Health-Data-Science/FromExceltoR>



WELCOME TO FROM EXCEL TO R

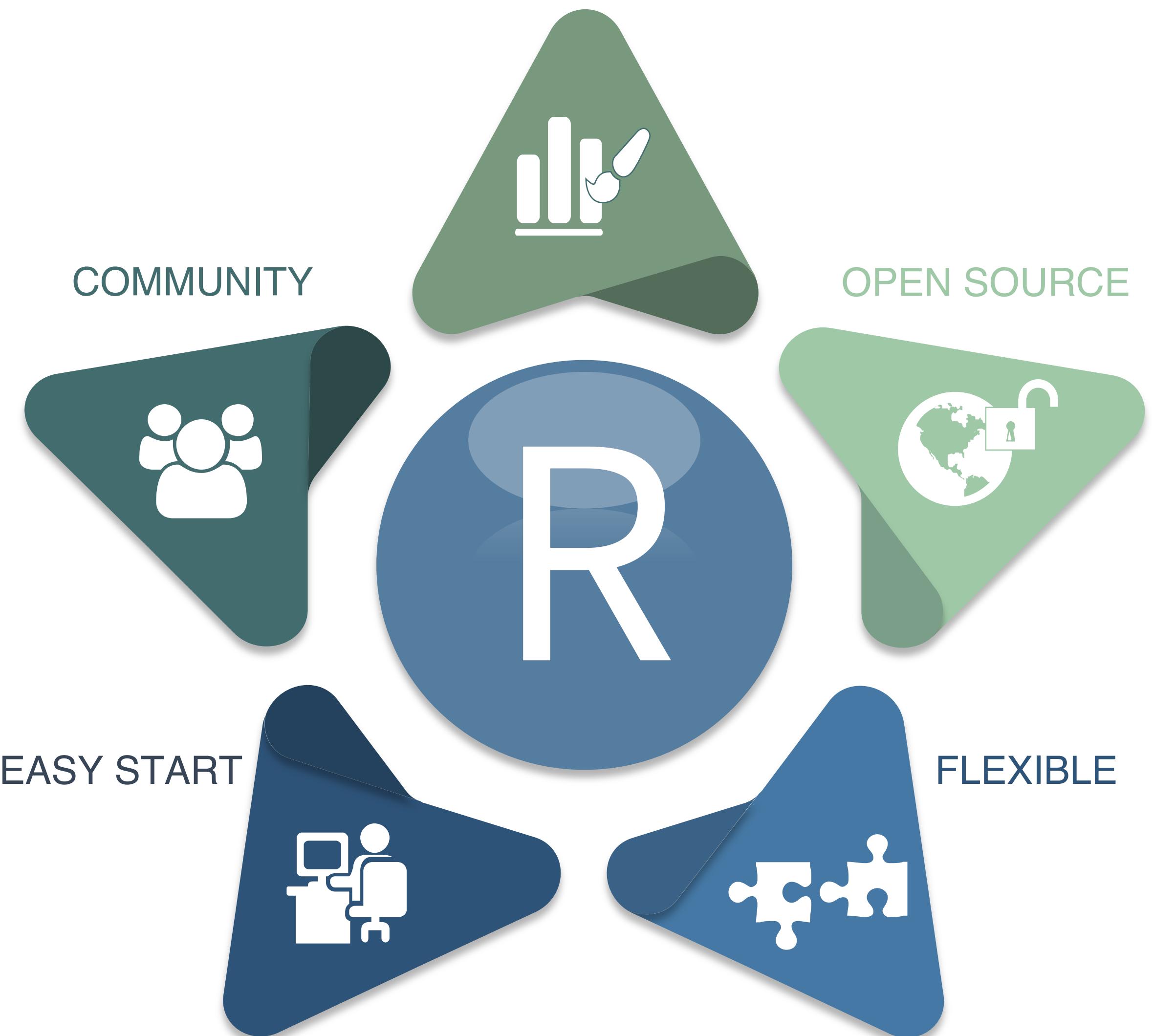


WHY R ?

- **Open Source**
- **Easy to get started with:**
Compatible with all systems, great support
- **Large Community:**
R-packages, pipelines, tutorials, help pages
- **Flexible Language:**
Plugins, git/github, R Shiny, Rmarkdown, ...
- **Customisable Graphics**

R has its **limitations**, but now fewer than ever

GRAPHICS



A COMPARISON

FROM EXCEL TO R



WHAT WILL YOU LEARN IN THIS COURSE?

THE BASICS



R base syntax
R Studio
Scripts, paths, files
R project
Help resources

DATA WRANGLING



tidyverse
Data Structures
Useful Functions
Pipe (“clean” code)



STATISTICS IN R

PLOTTING



ggplot2
Code structure
ggplot2 + tidy data

REPRODUCIBILITY



R Markdown
Doc. types
Good practices
Other cool things

PROGRAM

DATES: 11-10 & 12-10, 2022

PLACE: Faculty of Health and Medical Sciences,
Panum, Blegdamsvej 3B, 2200 København

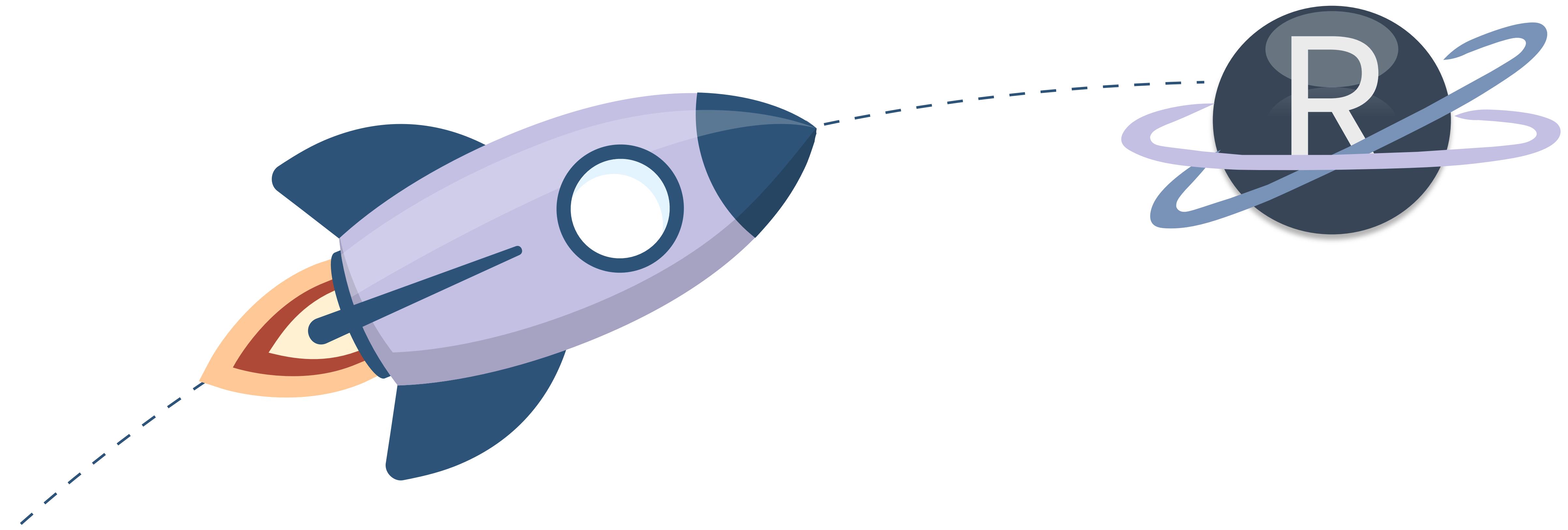
DAY 1

08:30 - Installation Issues & Coffee
09:00 - Introduction to R Basics
09:45 - Rstudio Exercise
10:15 - Break
10:30 - Tidyverse
11:00 - Tidyverse Exercise
12:00 - Lunch
13:00 - Tidyverse Exercise
14:00 - Break
14:15 - ggplot2
15:00 - ggplot2 Exercise
16:30 - Q&A - See you tomorrow

DAY 2

08:45 - Coffee
09:00 - Rmarkdown
09:30 - Rmarkdown Exercise
10:00 - Break
10:15 - Statistics in R
11:00 - Statistics Exercise
12:00 - Lunch
13:00 - Statistics Exercise
13:45 - Course Evaluation & Break
14:00 - R (Bring Your Own) Dataset Exercise
15:45 - Other cool things in R
16:00 - Wrap up

— FROM EXCEL TO R
LET'S GET STARTED



R & FRIENDS



Scripting / Programming Language



Reports (html, pdf,
latex)

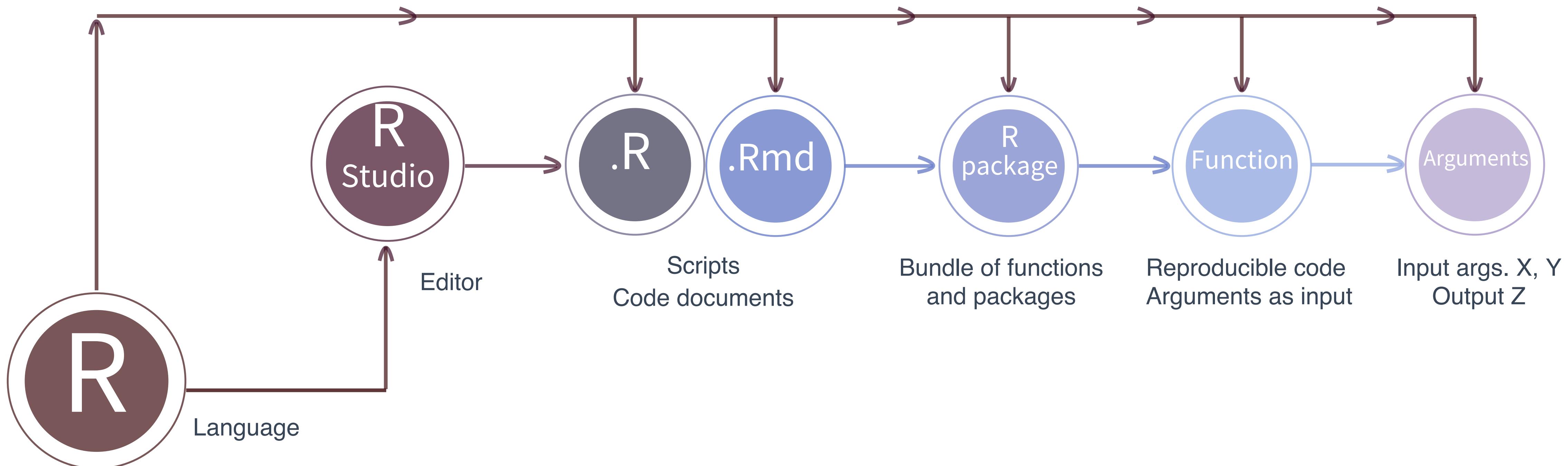


R Studio

A screenshot of the RStudio interface. The top bar shows 'RStudio' and 'Project: (None)'. The left pane displays two R Markdown files: 'KnitR_Presentation.Rmd' and 'Meine_erste_Praesentation.Rmd'. The main pane shows code snippets for 'R-scripts (Save Your Code)' and 'Console (Run and Test Code)'. The right pane shows the 'Environment' tab with 'dataset' and 'Objects in Environment' sections, and the 'Packages' tab showing 'knitr' as selected. A large callout text 'Overview by Type, Help and R-packages' points to the right pane.

R Code Interpreter and Editor

THE ANATOMY OF R



FIRST TIME IN R?

PACKAGES & FUNCTIONS

?my.package, ?my.function

What is it? Input?

install.packages(), remove.packages()

TIPS

Arrows↑↓ to find the code you ran

R studio tips: view, diagnostics

✖️⚠️ Auto-complete with tab

R-cheat sheets

(<https://rstudio.com/resources/cheatsheet>
s/)



WORKING DIRECTORY

setwd(), getwd(), list.files(), list.dirs()

Where am I working from? Full/relative path.

SAVE YOUR WORK

.R, (or .Rmd)

The file with my code. Save it!

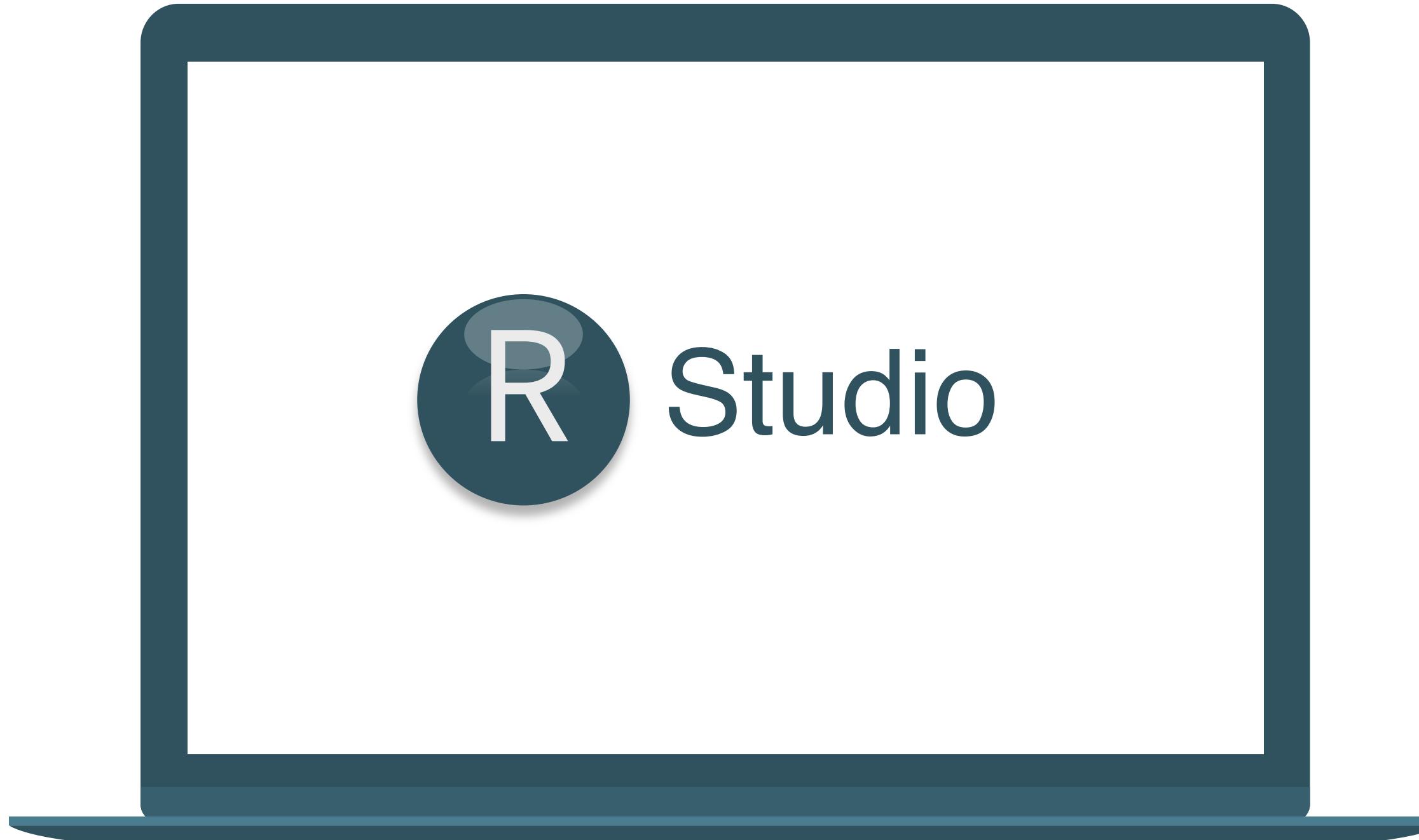
R project

Save Session, everything together

RUN CODE

Run button, highlight enter, short-cut

R STUDIO BASICS



1. R Project

File → New Project → New/Existing → Create Project

2. Set Path

getwd() - Get directory

setwd() - Set directory

setwd("/Users/Tom/Rstuff") - Full path

setwd("./Rstuff") - Relative path

Session → Set Working Directory → Choose Directory

3. R Script

Script Icon → R Script → File → Save as ...

4. Install and load a R package.

install.packages("my.package")

library (my.package) - Load package.

Tools → Install packages → my.package

ONLINE RESOURCES FOR R

<https://www.r-project.org/>



GET STARTED

<https://rseek.org/>

https://rstudio.com/resources/c_heatsheets/

<http://www.cookbook-r.com/>

<https://www.statmethods.net/r-tutorial/index.html>



GRAPHICS

<https://www.r-graph-gallery.com/>

<http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html>



BOOKS & COURSES

<https://www.r-bloggers.com/best-books-to-learn-r-programming/>

<https://www.datacamp.com/>

<https://www.codecademy.com/>

<https://www.coursera.org/>



OTHER RESOURCES

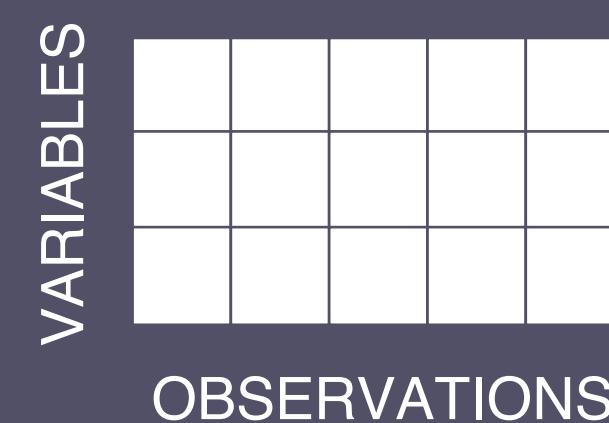
<https://github.com/trending/r>

<https://blog.revolutionanalytics.com/>

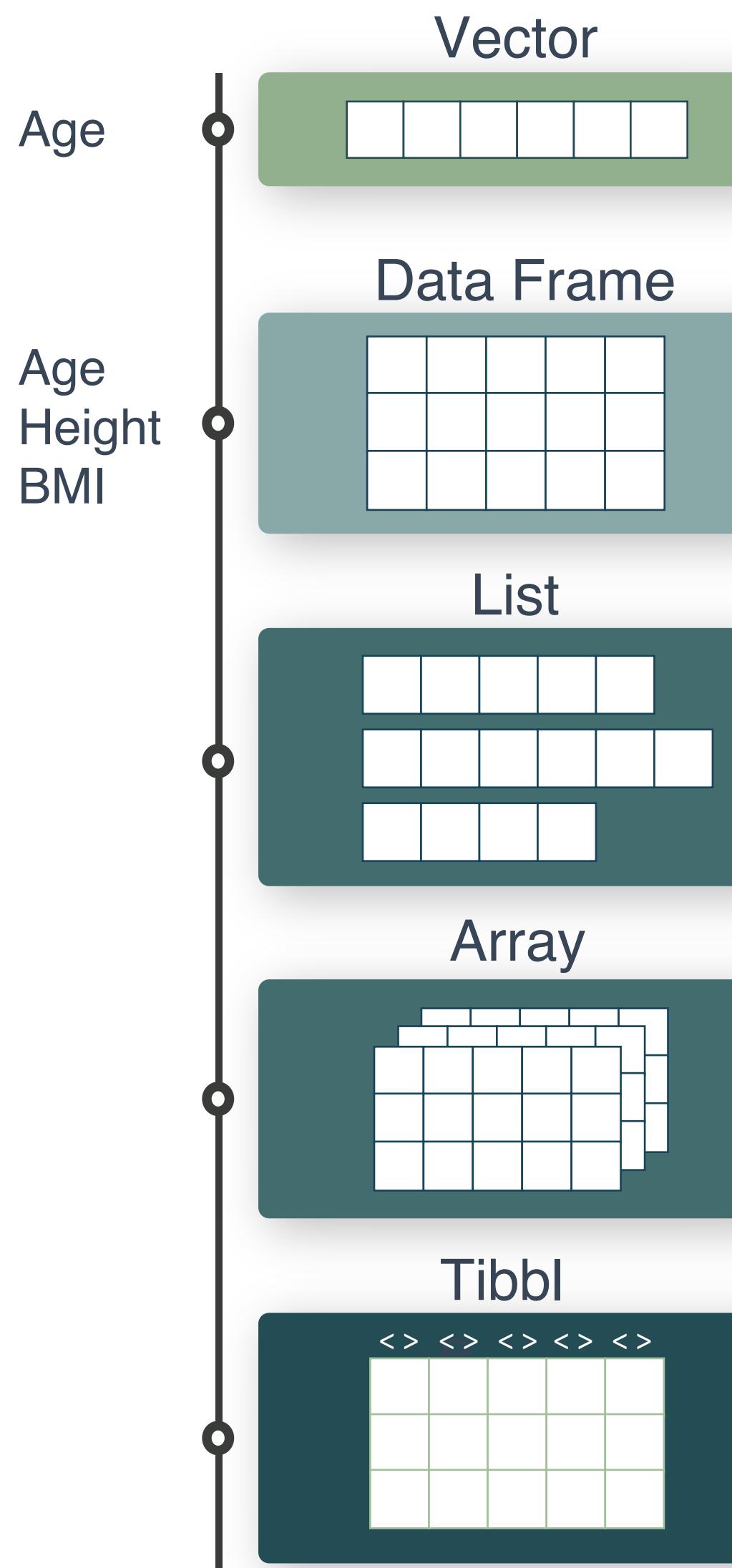
<https://stackoverflow.com/questions/tagged/r>



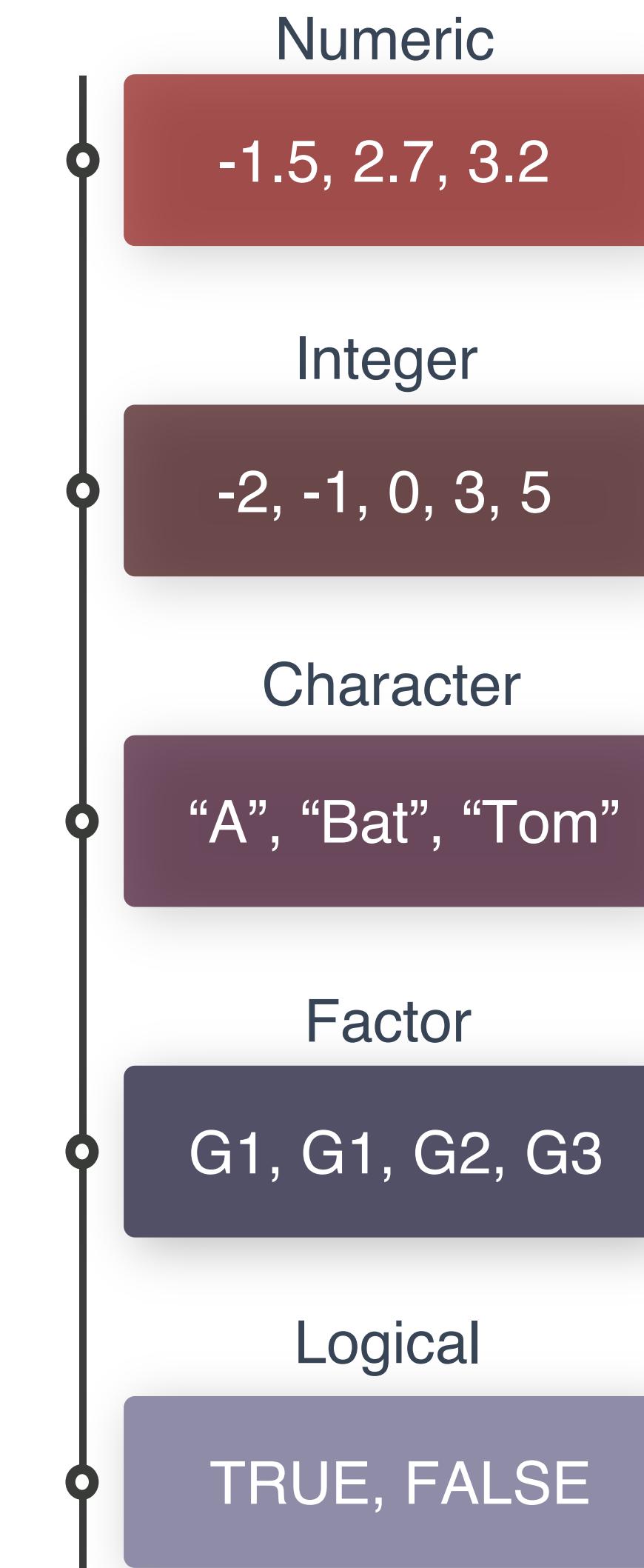
— FROM EXCEL TO R R DATA TYPES & STRUCTURES



DATA STRUCTURES



DATA TYPES



R BASE SYNTAX - RUN THROUGH

VARIABLE ASSIGNMENT

```
> a <- 'apple'  
> a  
[1] 'apple'
```

```
> x <- c(1.5, 2.6, 1.7, 3.2, 3.0, 2.9, ...)  
> x  
[1] 1.5 2.6 1.7 3.2 3.0 2.9 ...
```

READING AND WRITING DATA

DON'T USE

Input	Output	Description
<code>df <- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>	Read and write a delimited text file.
<code>df <- read.csv('file.csv')</code>	<code>write.csv(df, 'file.csv')</code>	Read and write a comma separated value file. This is a special case of read.table/write.table.
<code>load('file.RData')</code>	<code>save(df, file = 'file.Rdata')</code>	Read and write an R data file, a file type special for R.

Spaces in names
Special characters % ? / | \ & \$ @
Unspecific names
Short/long names

R BASE SYNTAX - RUN THROUGH

SELECTING ELEMENTS

x[4] The fourth element.

x[-4] All but the fourth.

x[2:4] Elements two to four.

x[!(2:4)] All elements except two to four.

x[c(1, 5)] Elements one and five.

By Value

x[x == 10] Elements which are equal to 10.

x[x < 0] All elements less than zero.

x[x %in% c(1, 2, 5)] Elements in the set 1, 2, 5.

R-BASE FUNCTIONS

log(x) Natural log.

sum(x) Sum.

exp(x) Exponential.

mean(x) Mean.

max(x) Largest element.

median(x) Median.

min(x) Smallest element.

quantile(x) Percentage quantiles.

round(x, n) Round to n decimal places.

rank(x) Rank of elements.

sig.fig(x, n) Round to n significant figures.

var(x) The variance.

cor(x, y) Correlation.

sd(x) The standard deviation.

CONDITIONS

a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

BASE R CHEAT SHEET

Basics: `getwd()`, `setwd() # location`
`install.packages('pname')`, `library(pname)`
`ls()`, `rm() # list, remove objects`
`load()`, `data()`, `save() # load, save as .Rdata`

Overview: `head(df, n=10)`, `df[1:10,]` `tail(df, n=10)`
first or last 10 rows
`class() # data structure`
`unique()`, `table() # unique vals, count vals`

Is/As type:
`is.numeric(x)` (character, factor, integer, etc.)
`as.numeric(x)` (factor, matrix, data.frame, etc.)

Other:
`seq(1, 10, by = 1.0) # sequence from-to`
`rep(x, times) # replicate n times`
`sort(), reverse() # sort or reverse vector`

Read in data:
`read.xlsx('name.xlsx')`,
`read.delim('name.txt', sep = '\t')`
`read.csv('name.csv', sep=';')`

Make Data: `c() # vector`
`data.frame(x=x, y=y)`
`matrix(x, nrow = 3, ncol = 3)`
`list(x=x, y=y)`

Strings:
`paste(x, y, sep = '')`
`grep('pattern', x) # find str pattern`
`gsub('pattern', 'replace', x) # replace with`

Plots:
`plot(x)`
`plot(x,y) # scatter`
`hist(x) # histogram`

GETTING
STARTED

DATA STRUCTURES
& OVERVIEW

DATA TYPES &
STRINGS

VECTORS &
BASE PLOTS

GETTING HELP



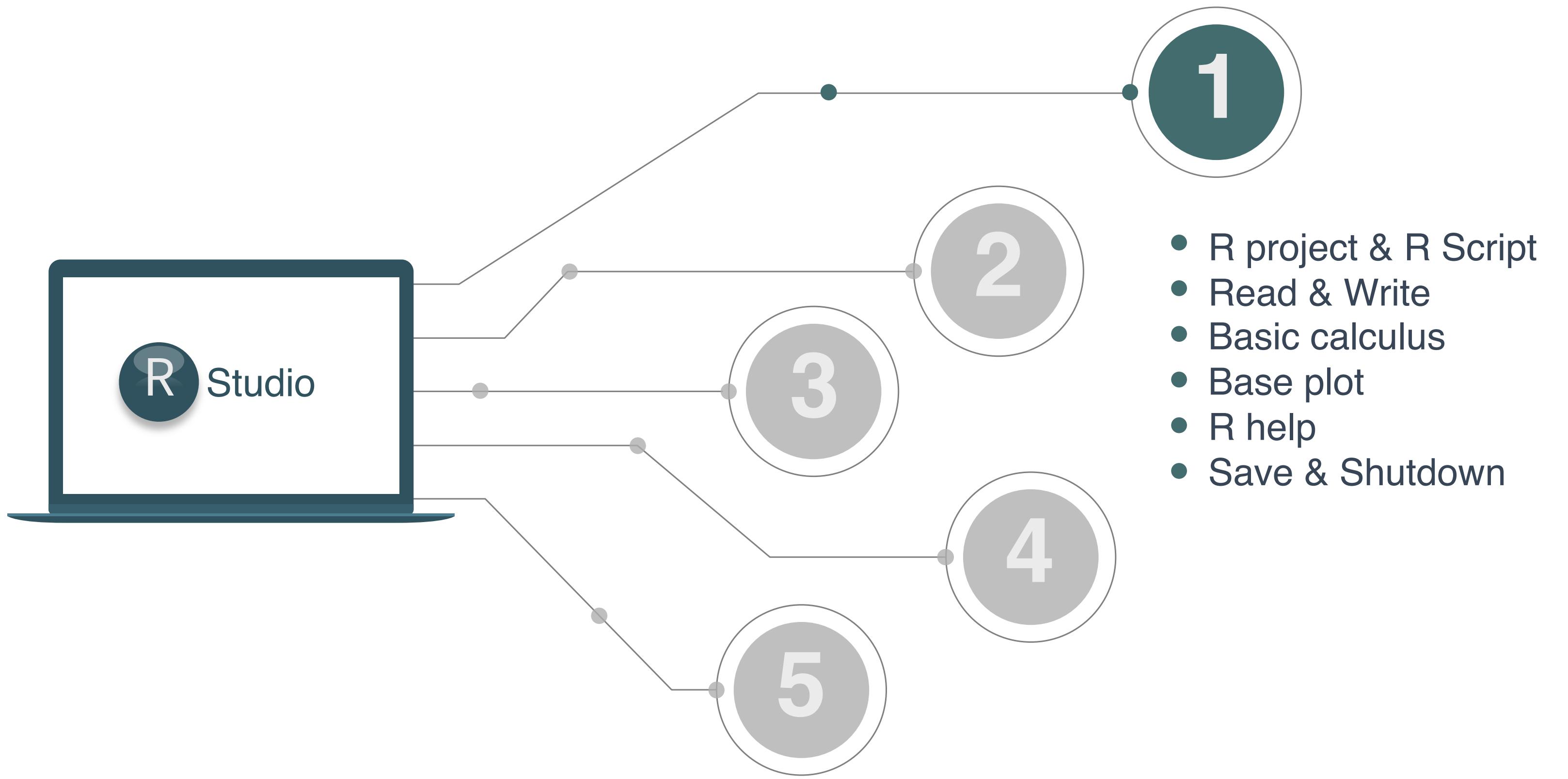
02

GETTING HELP

1. Ask us, and ask people next to you.
2. Look at the R presentations we go through.
3. Use the **cheat sheets** in this presentation and online:
<https://rstudio.com/resources/cheatsheets/>

Google it! Most important skill of all.

You will get the **solutions** to the exercises near the end of the day.



— FUNDAMENTALS
EXERCISE 1

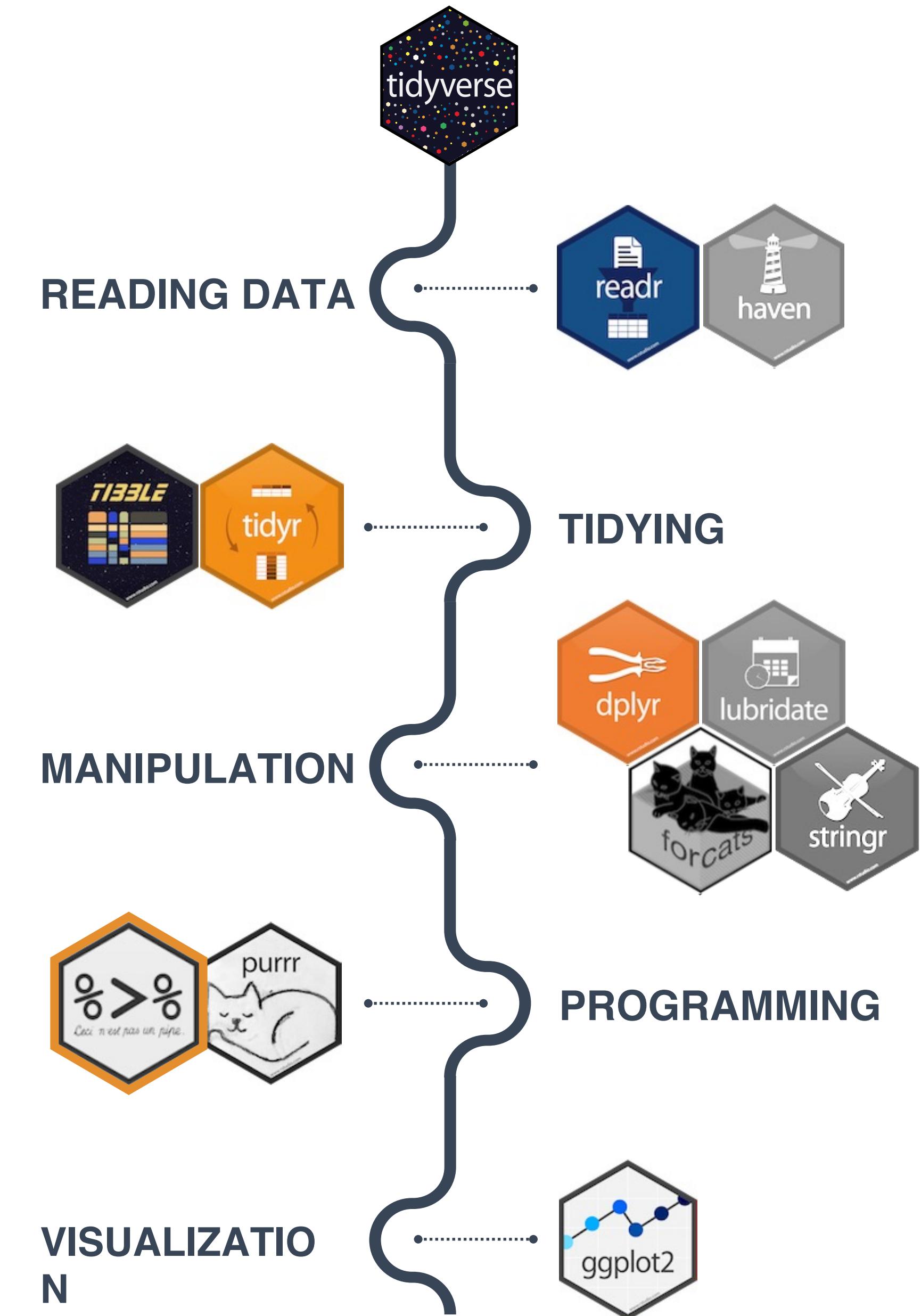
TIDYVERSE

<https://www.tidyverse.org/>

tidyverse is a collection of R packages for data science

“The packages share an underlying design philosophy, grammar, and data structures.” *Wickham and Grolemund*

tidyverse is used to “tidy up” your datasets, so they are easy to work with



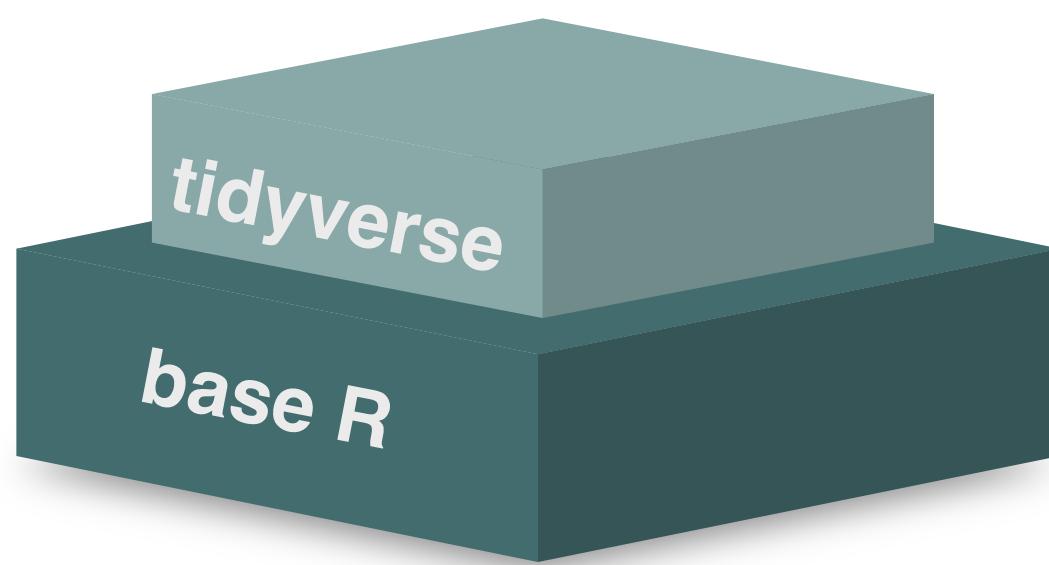
CECI N'EST PAS UNE PIPE

%>%

- You do NOT have to “choose” between tidyverse and base R

BENEFITS

- Short & well-organised code
- Tidy datasets, easy to work with
- Great documentation
- Functions with logical names & inputs



CONSIDERATIONS

- Can be less stable
- “Different syntax”
- Remember what tidyverse is made for!

base R

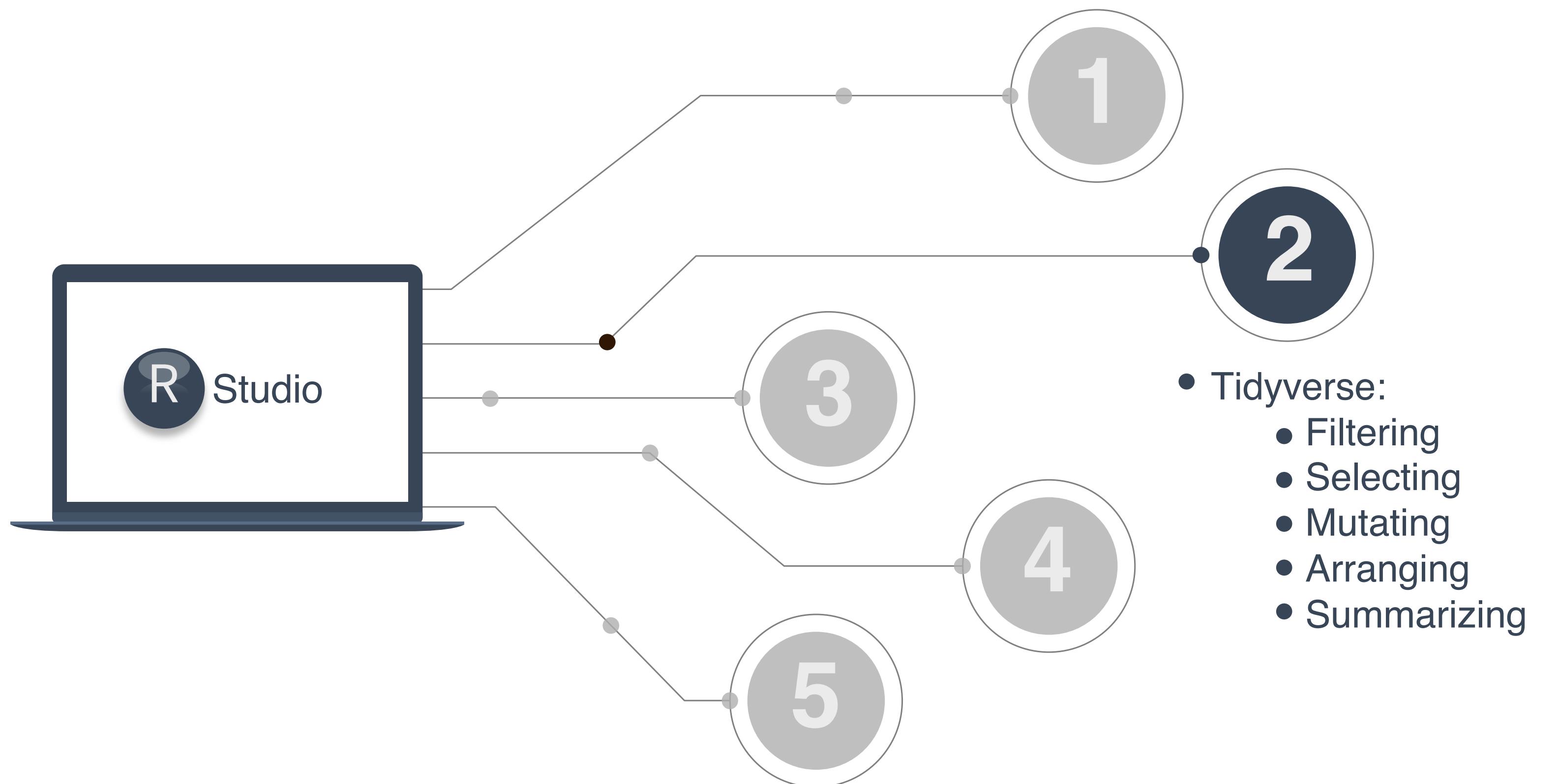
```
# think from the inside out  
g(f(x,y),z)
```

tidyverse

```
# no brain acrobatics  
x %>% f(y) %>% g(z)
```



pipe symbol



TIDYVERSE EXERCISE 2

TIDYVERSE CHEAT SHEET

Read Data (*readr*)

Reading tabular data

There are solutions for multiple data types
`read_excel()` # using *readxl* package
`read_table()`
`read_csv()`

Useful arguments

Skip lines: `read_csv(file, skip=1)`
Read subset: `read_csv(file, n_max=1)`

Data types

readr guesses the types of each column and tells you about it
(“Parsed with column specifications: ...”)

HELP

R Documentation (e.g. enter `?dplyr::filter` and see examples)

Much more info and detailed cheat sheets:

<https://brianward1428.medium.com/introduction-to-tidyverse-7b3dbf2337d5>

It also helps to google “tidyverse + whatever you want to do”

Data Tidying (*tidyr*)

Handle missing values

`drop_na()`
`fill()`
`replace_na()`

Subsetting

`tibble[, 1:5]` # returns a tibble
`tibble$colname` # returns a vector
(same as `tibble[[colname]]`)

Reorganize layout

Change between long and wide format
`gather()` # wide to long
`spread()` # long to wide

Data Manipulation (*dplyr*)

Summary

`summarise()`/`summarize()`
`count()`

Group

`group_by()`

Functions will manipulate each group separately and combine results.

Extract and sort observations # i.e. rows

`filter()` # subset by condition
`distinct()` # subset to unique values
`top_n()` # subset by position
`arrange()` # sort low->high, other way with `desc()`

Manipulate variables # i.e. columns

`select()`
`mutate(new_name = f(column))`

Vectorised functions

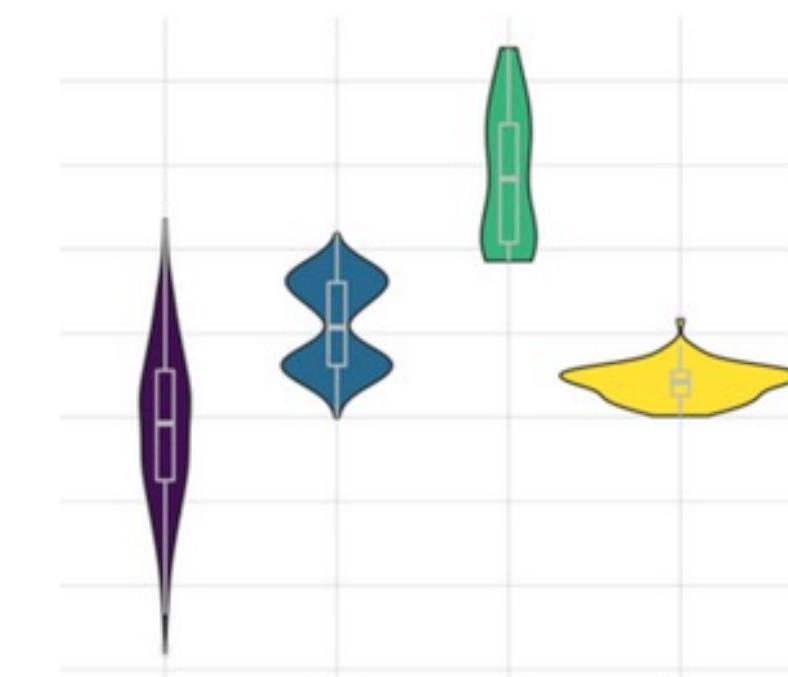
Ranking: `percent_rank()`
Math: Any arithmetic or logical operations, `between()`,
`near()`
`if_else()`

<https://www.r-graph-gallery.com/>

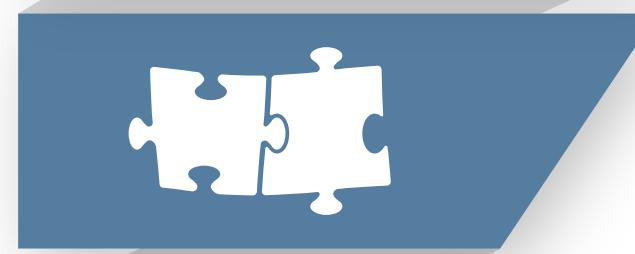
GGPLOT2 - EASY GRAPHICS



Aesthetically pleasing graphics.



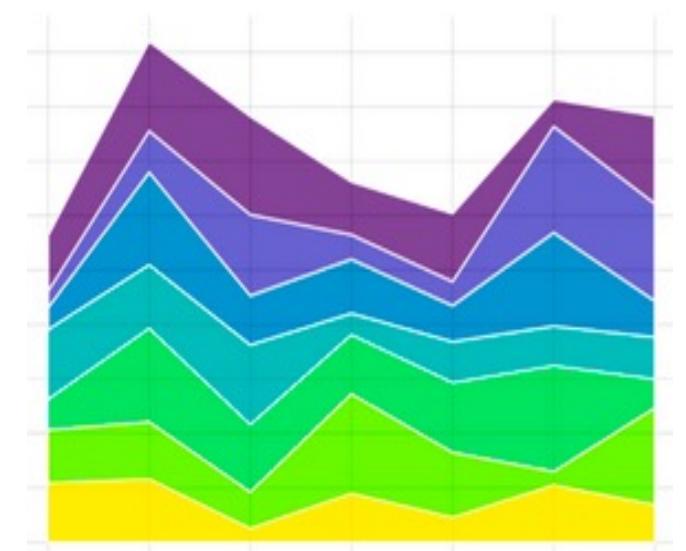
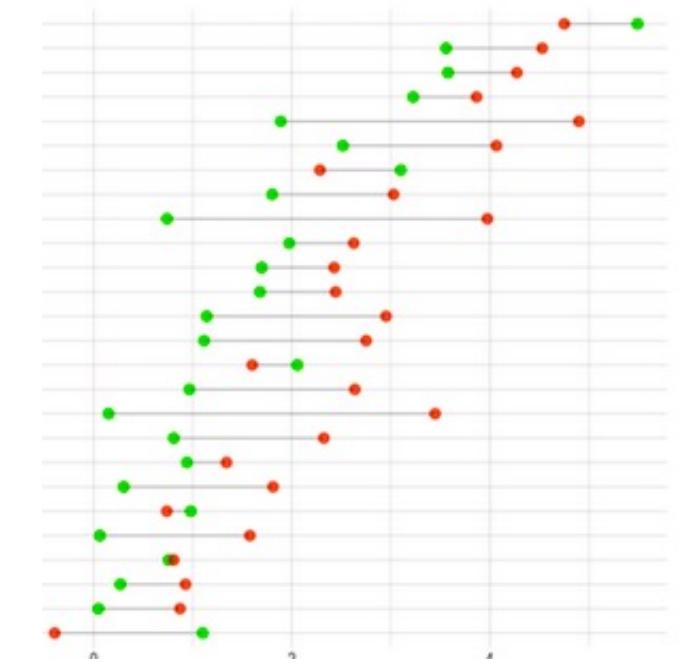
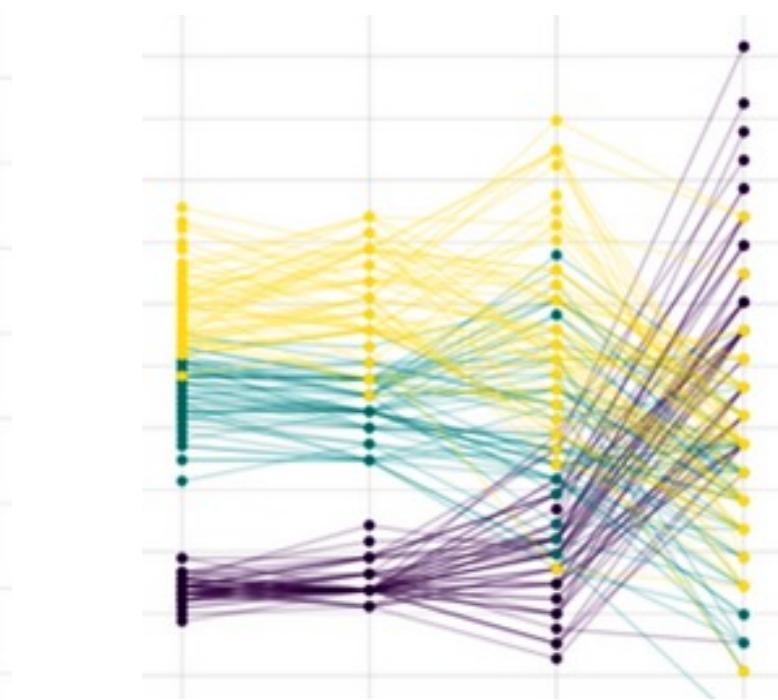
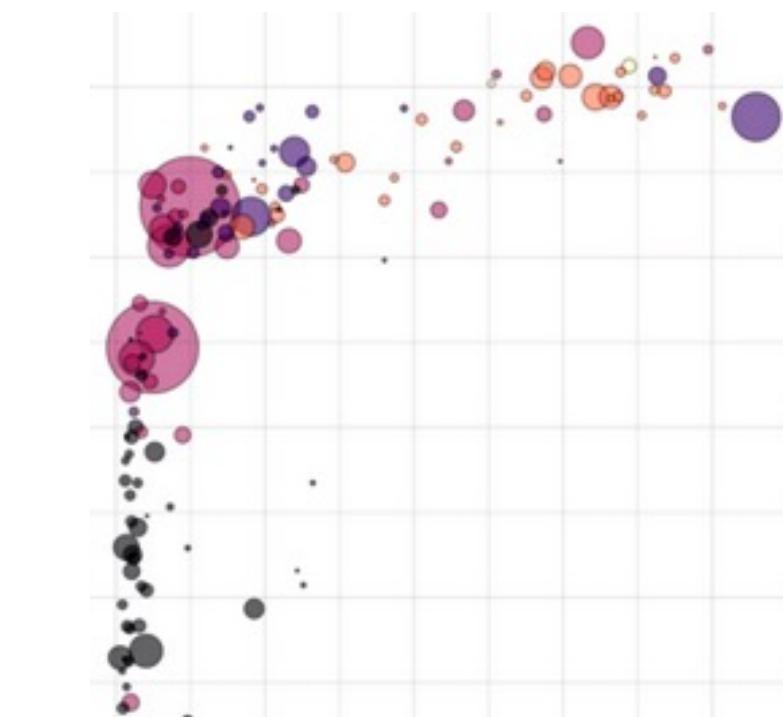
Well-defined “additive” (+) structure.



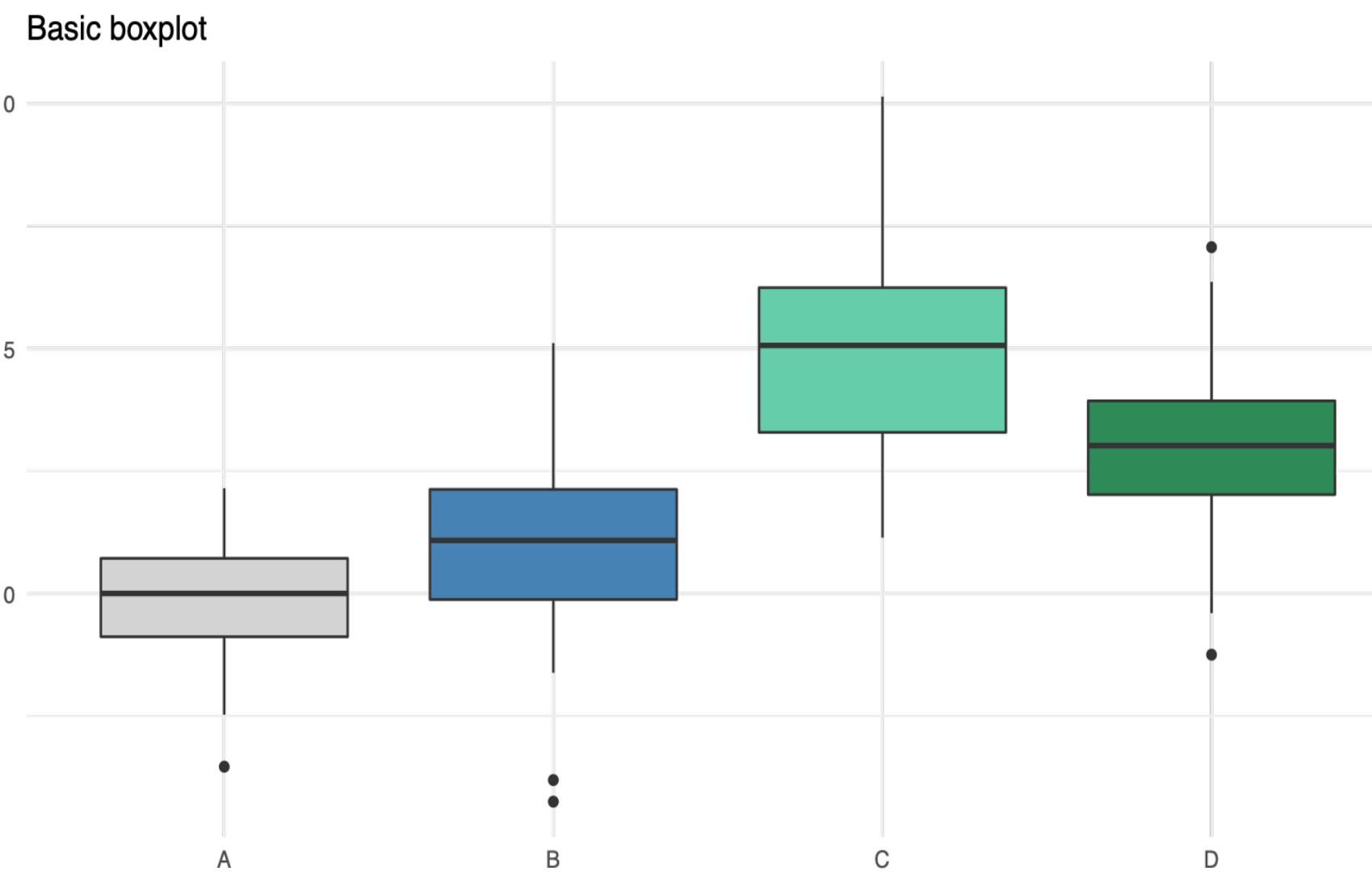
Integrates perfectly with tidy data.



Great documentation & community



GGPLOT2 ADDITIVE STRUCTURE



DATASET, SAMPLES
& OBSERVATIONS



```
ggplot(my.DS, aes(x=alphabet,  
y=measure))
```

DEFINE PLOT TYPE



```
ggplot(my.DS, aes(x=alphabet,  
y=measure))  
+ geom_boxplot()
```

COLOR BY GROUP



```
ggplot(my.DS, aes(x=alphabet,  
y=measure, fill=alphabet))  
+ geom_boxplot()
```

TITLE AND LEGEND



```
... + ggtitle("Basic boxplot") +  
theme(legend.position="none",  
plot.title =  
element_text(size=11))
```

CUSTOM COLORS

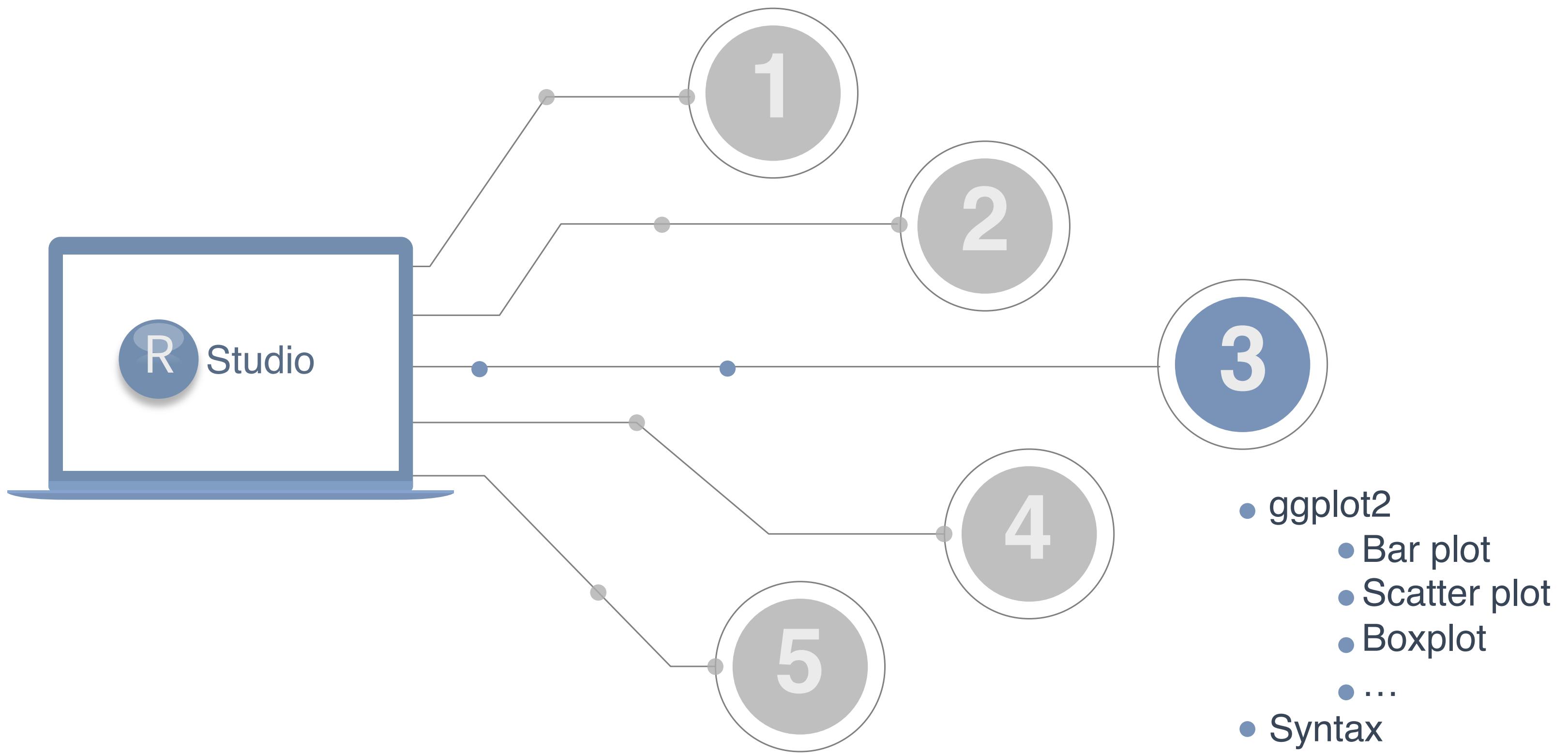


```
... + scale_fill_manual(values =  
c("lightgray", "steelblue",  
"aquamarine3", "seagreen4"))
```

BACKGROUND



```
+ theme_minimal()
```



— GG PLOT 2
EXERCISE 3

GGPLOT CHEAT SHEET

Define Plot:

```
ggplot(data = my.data,  
aes(x = x.var, y = y.var))
```

Add Plot Type:

- + geom_point()
- + geom_line()
- + geom_boxplot()
- + geom_col()
- + geom_density()
- + geom_histogram()

One Color:

```
ggplot(..., aes(...,  
color = "green"))
```

Color Fill by Group:

```
ggplot(..., aes(...,  
fill = group.var))
```

More Colors:

- + scale_fill_grey(start = 0.2, end = 0.8)
- + scale_fill_gradient(low="white", high="red")

Labels:

- + ggtitle("...")
- + xlab("...")
- + ylab("...")

Custom Colors:

- + scale_*_manual(values = c())
- ex: scale_color_manual(values = c("blue", "pink"))

Grid Theme:

- + theme_bw()
- + theme_minimal()
- + theme_dark()
- + theme_*

Text:

- + theme(* = element_text())
- + theme(axis.title = element_text(angle = 90, colour= "red"),
legend.text = element_text(size = 8, face = "bold"))

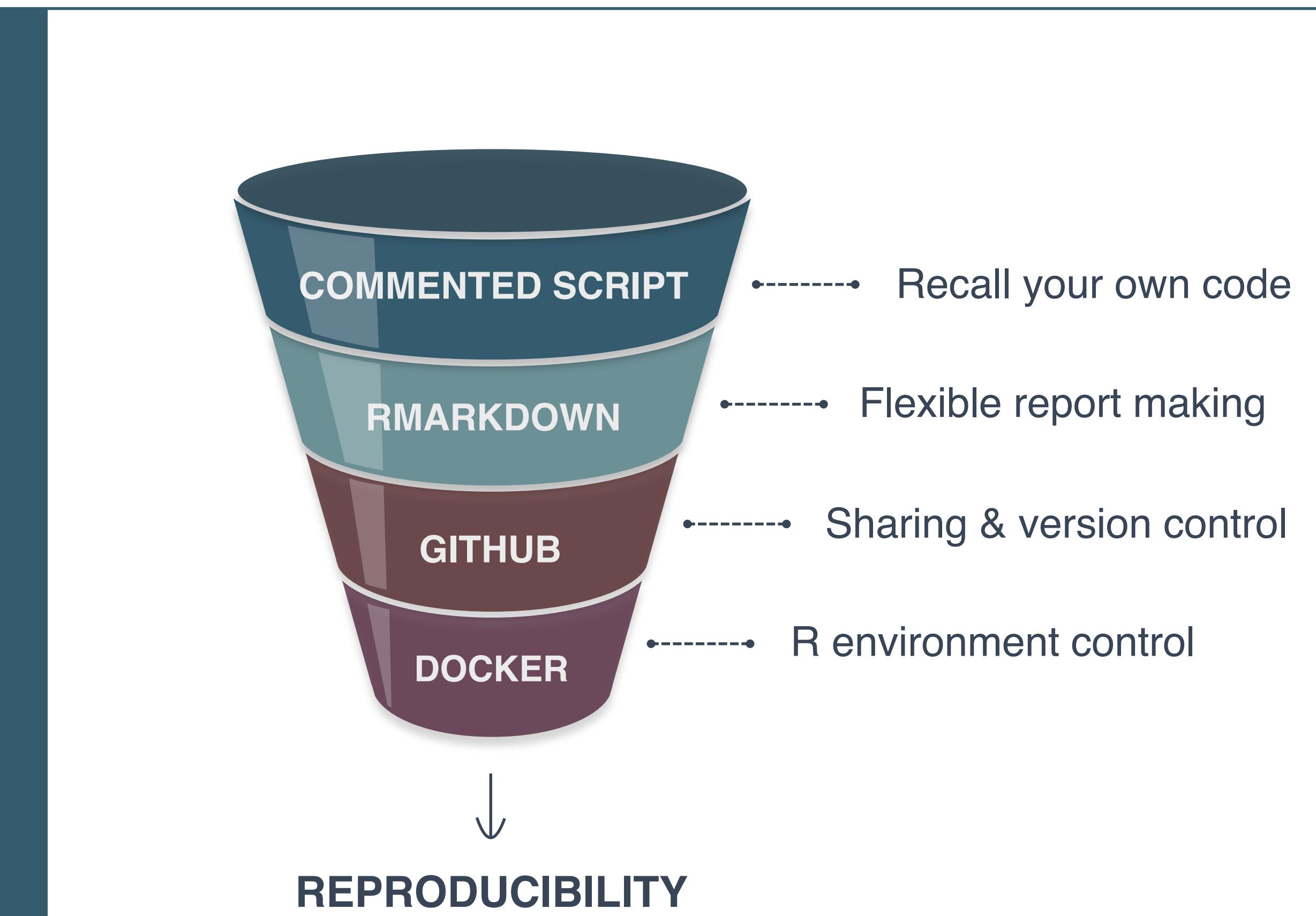
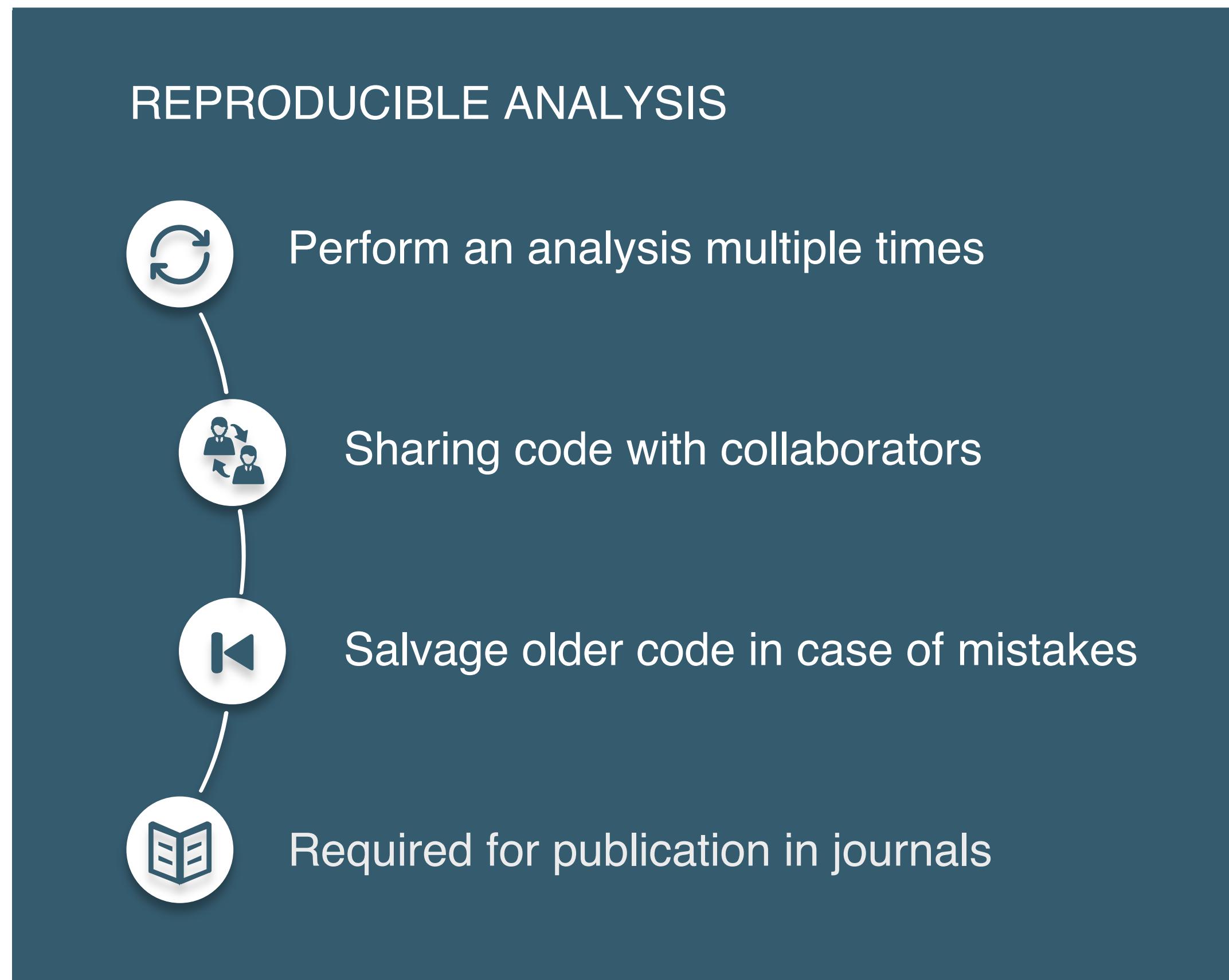
GET
STARTED

COLORS

COLOR SCALES
& THEMES

TEXT

REPRODUCIBILITY IN R





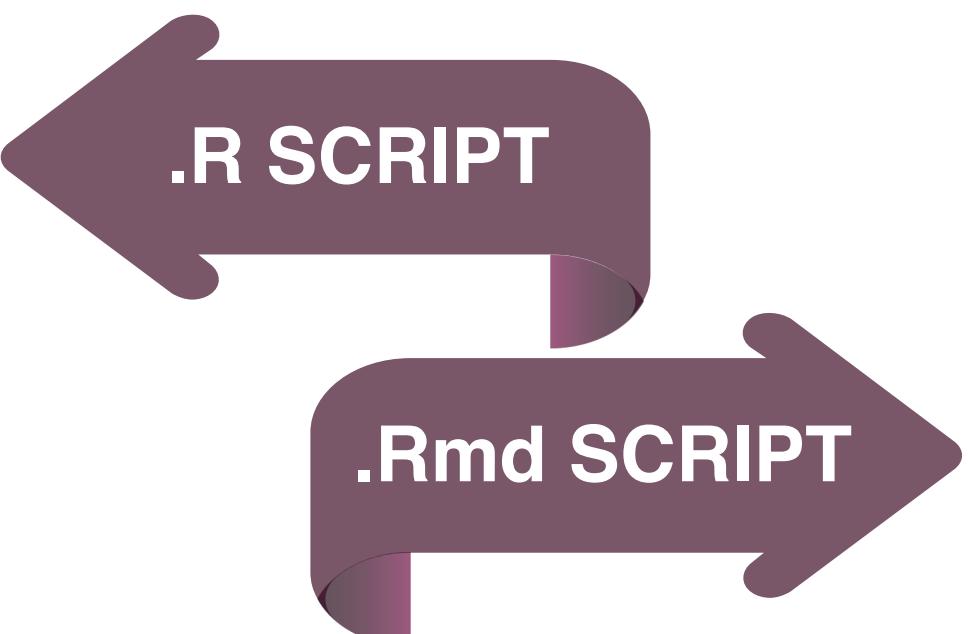
- A markup language
 - Plain text format
 - Converted into HTML, PDF, LATEX
-
- Finds code embedded in text
 - Evaluates the code
 - “Knits” the results into the document

— <https://www.markdowntutorial.com/>

R SCRIPT & RMARKDOWN

HOW TO

- Write code as normal
- Comment text with #



HOW TO

- Write text as normal
- Embed code `'{r} my.code'`

USE FOR

- Testing new code
- Big data analysis
- Software development

USE FOR

- Reports for yourself
- Reports for collaborators
- Tutorials

RMarkdown

The diagram illustrates the RStudio interface for R Markdown. On the left, the R Markdown editor shows the following code:

```
1 ---  
2 title: "Rcourse"  
3 author: "Data Science Lab"  
4 date: "9/14/2020"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. The document is a flat text type document which can be read without opening it in RStudio. Making a markdown document is easy, For example, if you want to make something bold use two stars, For italics use underscore, etc. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 Here is a summary of the cars dataset  
19```{r cars}  
20 summary(cars)  
21```  
22  
23 Here is some math  
24```{r, eval=TRUE}  
25 ((5+6+7)/3)*12  
26```  
27  
28 Here is a plot
```

The right pane shows the generated HTML output:

Data Science Lab
9/14/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. The document is a flat text type document which can be read without opening it in RStudio. Making a markdown document is easy, For example, if you want to make something **bold** use two stars, For *italics* use underscore, etc. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Here is a summary of the cars dataset

```
summary(cars)
```

```
##      speed      dist  
##  Min.   : 4.0   Min.   : 2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Here is some math

```
((5+6+7)/3)*12
```

```
## [1] 72
```

<https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

<https://rmarkdown.rstudio.com/lesson-1.html>

RMarkdown Cheat Sheet

Begin .Rmd:

```
---
```

```
title: My Project Name
```

```
output:
```

```
    html_document (pdf_document, ...)
```

```
---
```

Code Chunk:

```
```{r}  
some R code
```
```

Global Option:

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

GETTING
STARTED

Code Options:

```
echo (= TRUE or FALSE - print my code)  
eval (= TRUE or FALSE - run my code)  
warning (= TRUE or FALSE display warning messages)
```

Figure Options:

```
fig.align (= 'left', 'right', 'center')  
fig.cap (= 'my figure caption')  
fig.height (= n), fig.width (= n)
```

CHUNK
OPTIONS

Header:

Header size ranging from largest (one #) to smallest (six #):
my.text, ## my.text, ### my.text, etc.

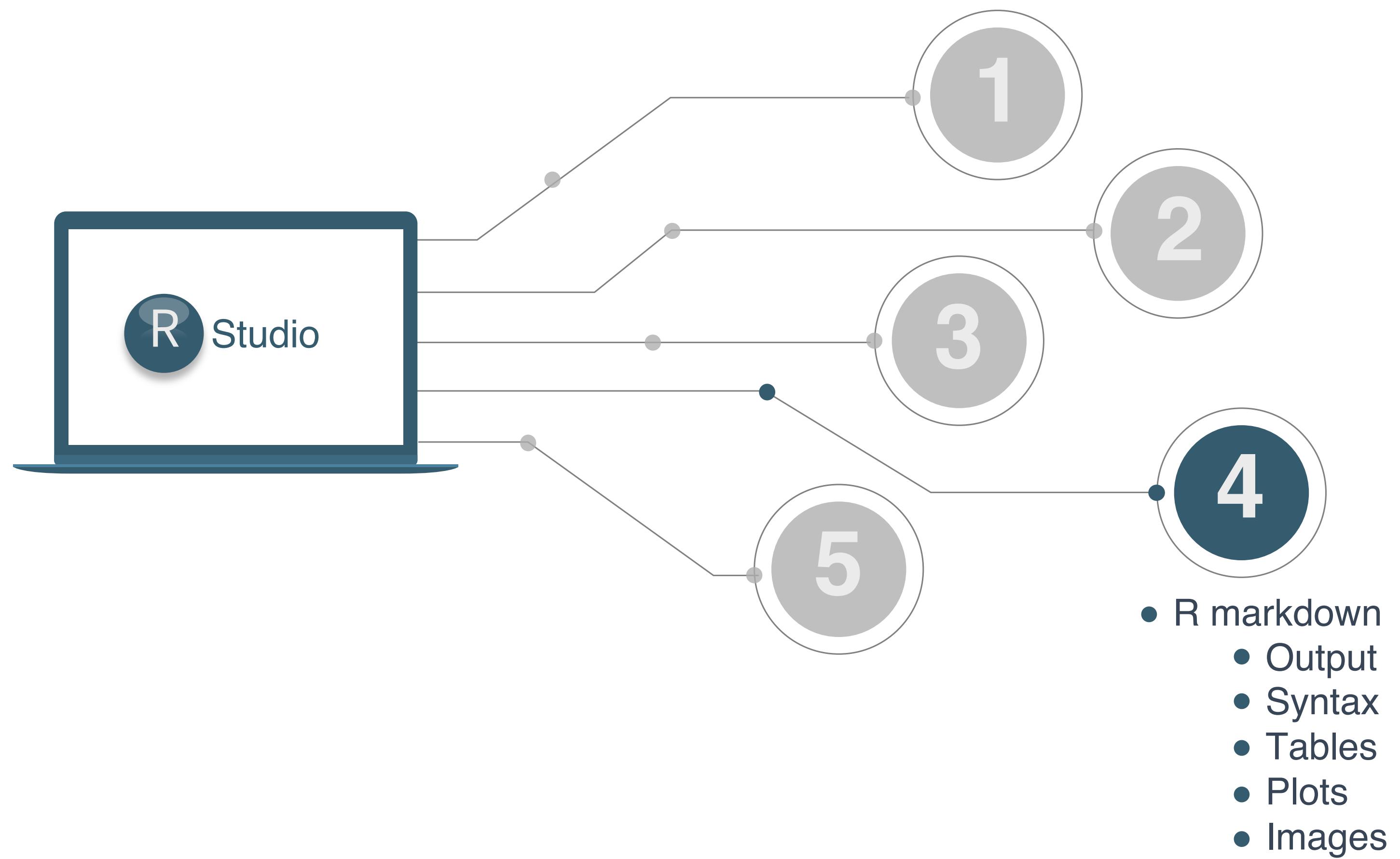
Text:

italics
bold
`highlighted`

Lists:

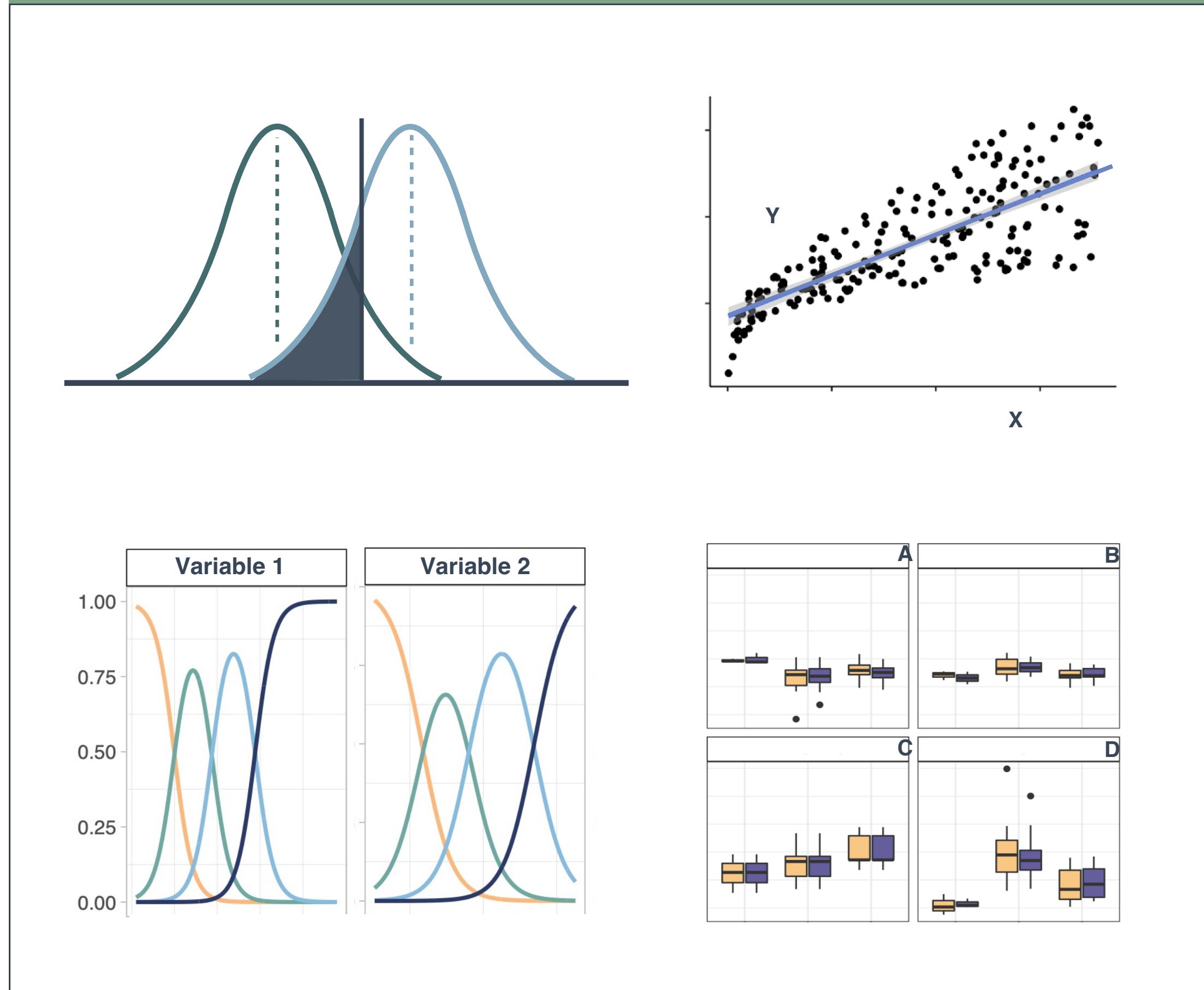
- * List item1 (filled dot)
 - + sub-item1 (open dot)
- 1. List item1 (numbered)
 - i) sub-item1 (roman)

TEXT



— R markdown
EXERCISE 4

R - A STATISTICAL SCRIPTING LANGUAGE



MODEL FUNCTIONS

`lm()`, `glm()`
`lmer()`, `glmer()`,
`nls()`, ...

EMMEANS PACKAGE

`emmeans()`,
`pairs()`, `cld()`

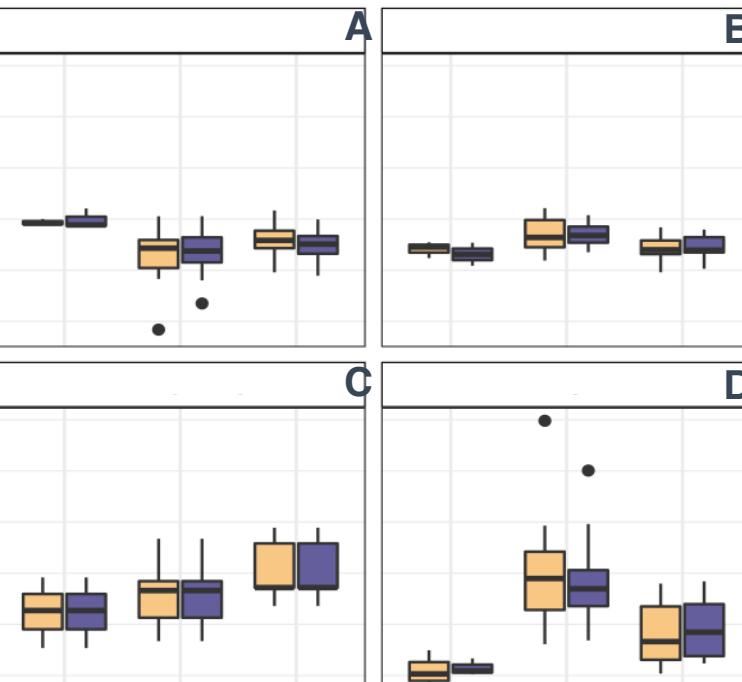
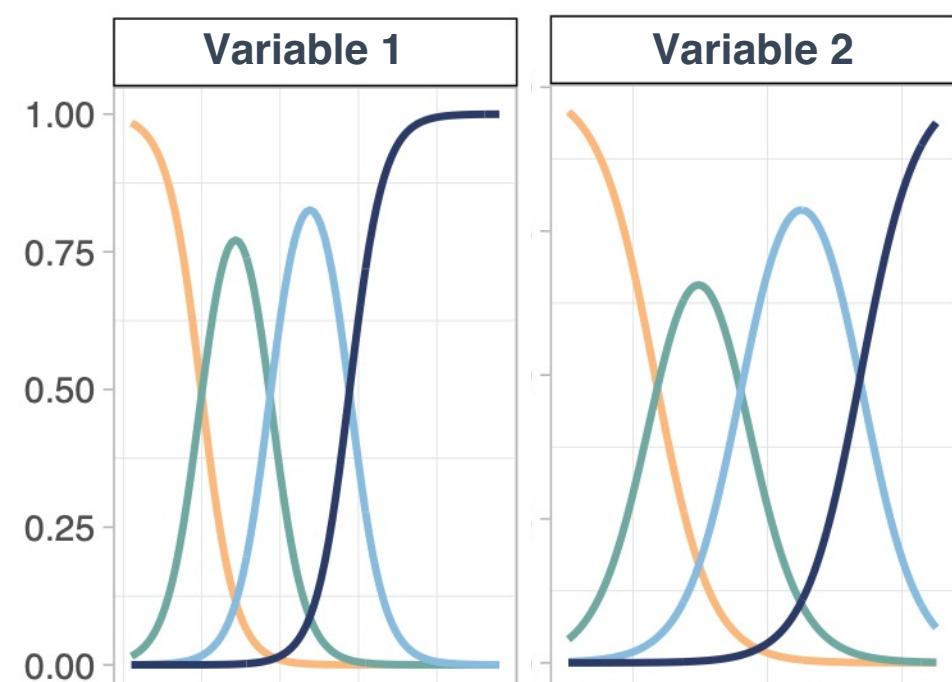
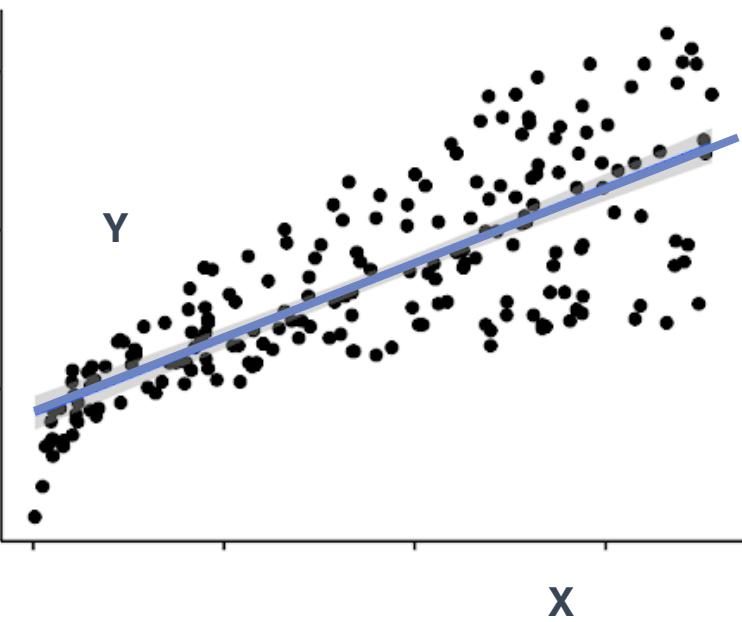
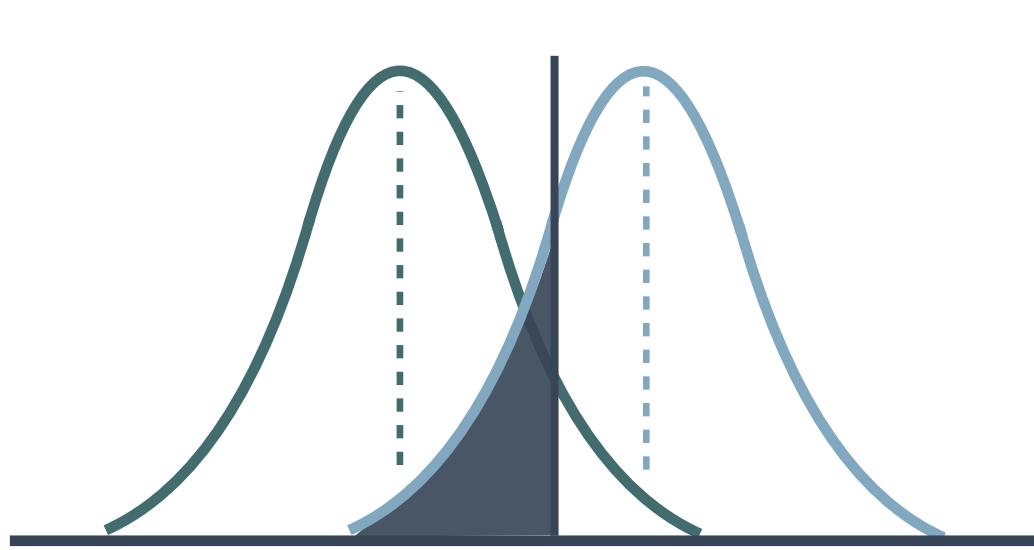
APPLY TO MODEL

`summary()`, `anova()`,
`confint()`, `predict()`,
`drop1()`, `update()`,
`step()`, ...

MORE FUNCTIONS

`t.test()`, `cor()`,
`cor.test()`, `aov()`,
`quantile()`,
`p.adjust()`,
`rank()`, ...

R - A STATISTICAL SCRIPTING LANGUAGE



During this session:

- Cooperatively discuss and share ideas about the data
- Apply steps of basic statistical analysis for hypothesis testing consistent with the given data
- Suggest conclusions, based on your analysis, regarding the association between psoriasis and gene expression levels

Let's use R to perform a statistical analysis



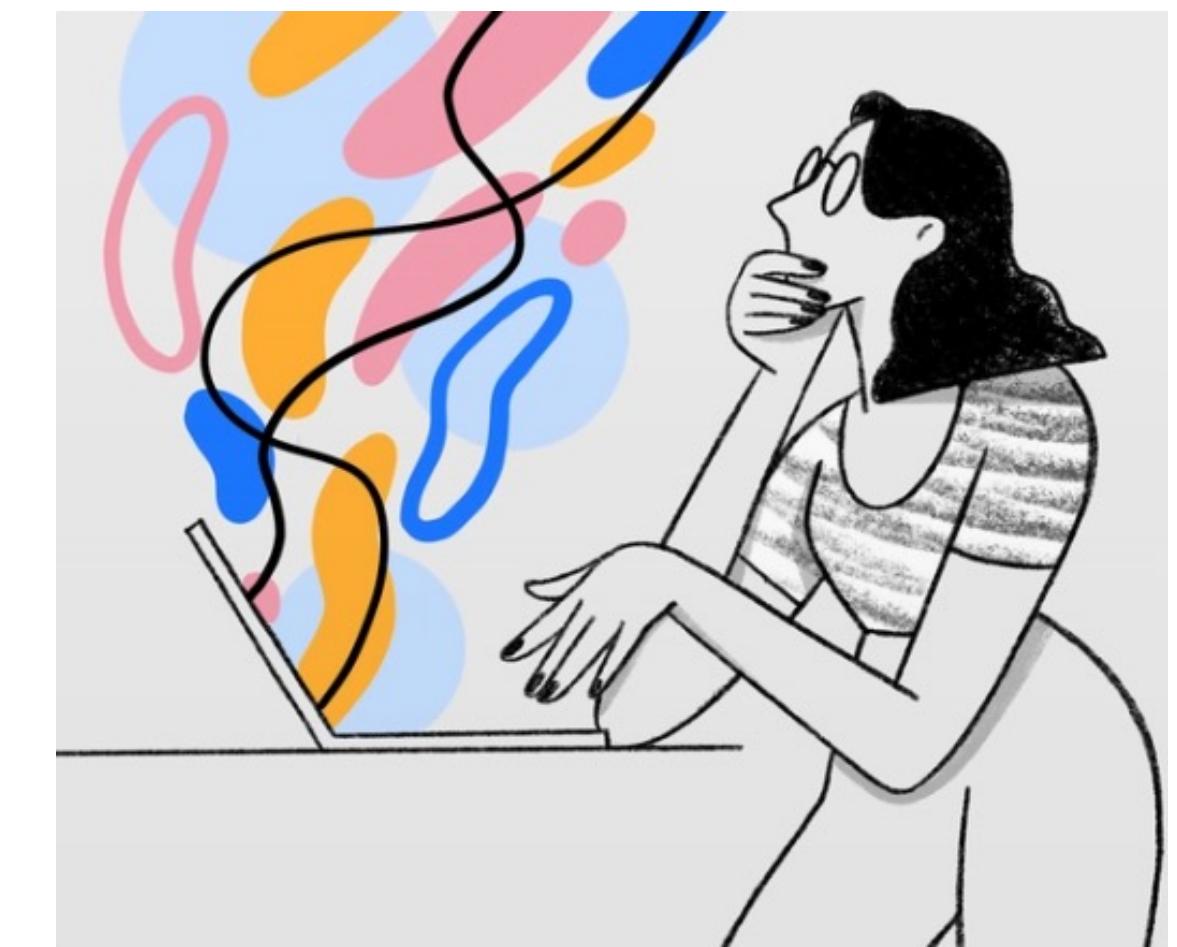
Your data

Gene expression data i.e. how much a gene is used to produce a protein. Multiple genes measured in an array.

Your disease of interest

Psoriasis, an immune-mediated skin disorder

You
The researcher with R skills!



STATS CHEAT SHEET

Import Data:

```
read_excel("my.data.xlsx")
```

Overview of Data:

```
summary(my.data)  
nrow(my.data)
```

```
length(my.data)  
names(my.data)
```

GET
STARTED

Linear:

```
lm(y~x, data=my.data)  
confint(model)
```

Logistic:

```
glm(y~x,  
data=my.data)
```

Linear Mixed:

```
lmer(y~x + (1|z),  
data=my.data)
```

Check Model:

```
summary(model)  
par(mfrow=c(2,2))  
plot(model)
```

ANOVA:

```
anova(model2, model1)
```

F-Test:

```
drop1(model, test="F")
```

Emmeans:

```
emmeans(model, ~x)  
pairs(emmeans(model, ~x))
```

REGRESSION
MODELS

Check Type:

```
table(my.data$x)  
is.numeric(my.data$x)  
is.factor(my.data$x)
```

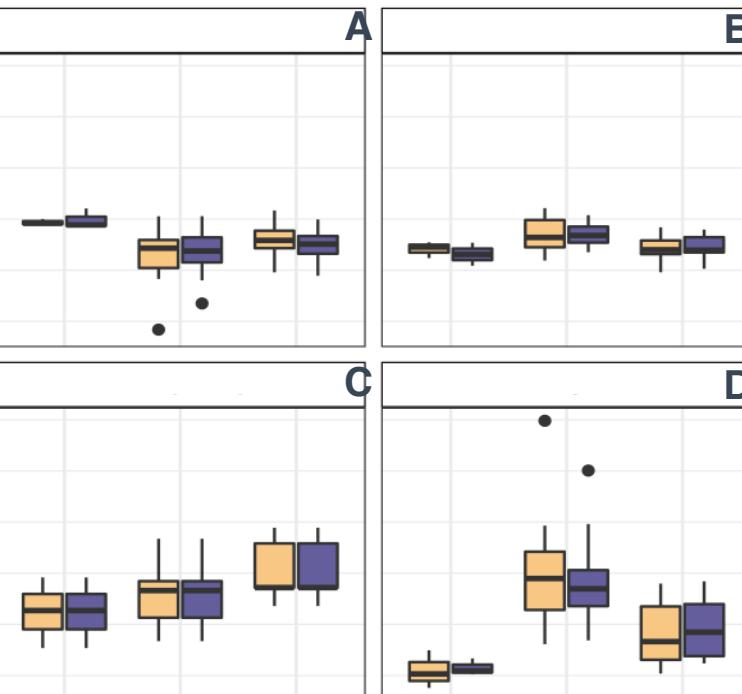
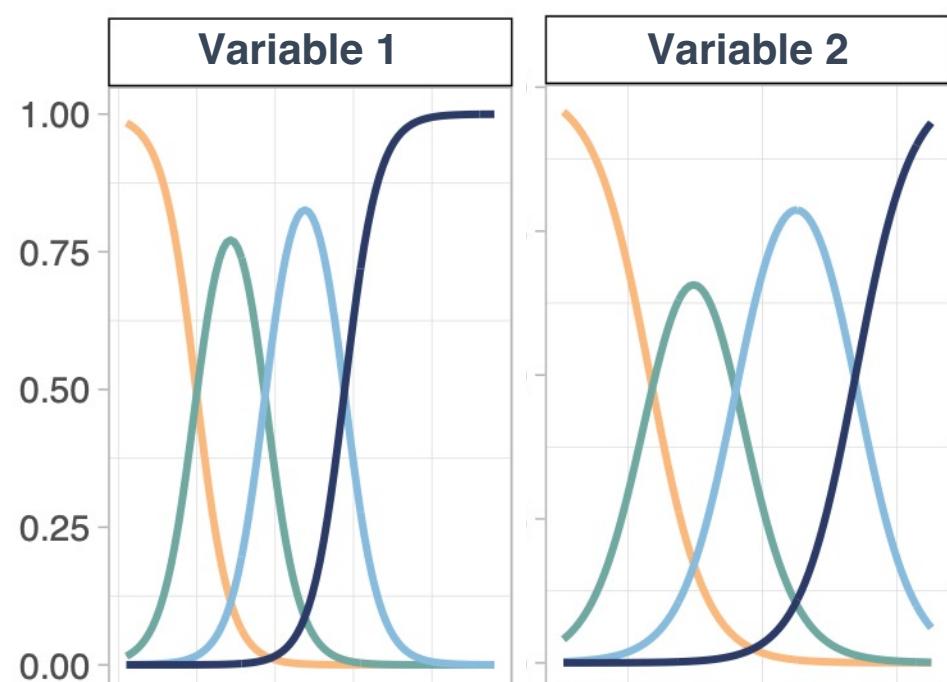
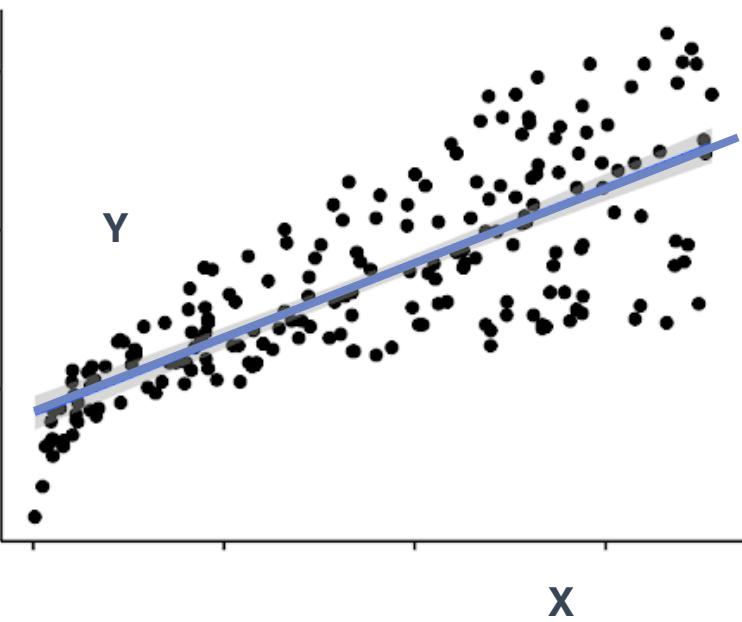
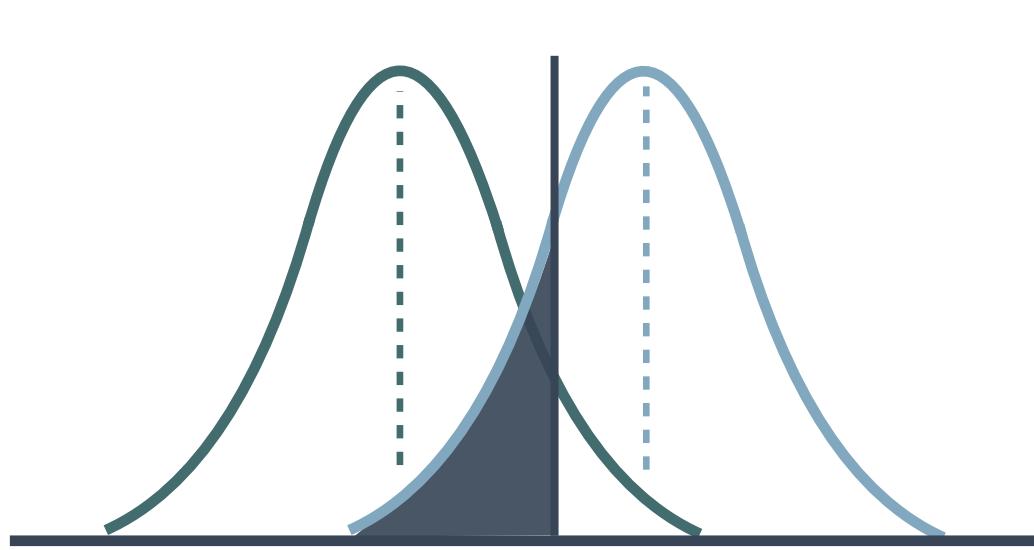
Change Type:

```
my.data <- mutate(my.data, x = factor(x))  
my.data$z <- as.numeric(my.data$z)
```

TESTS/
COMPARISONS

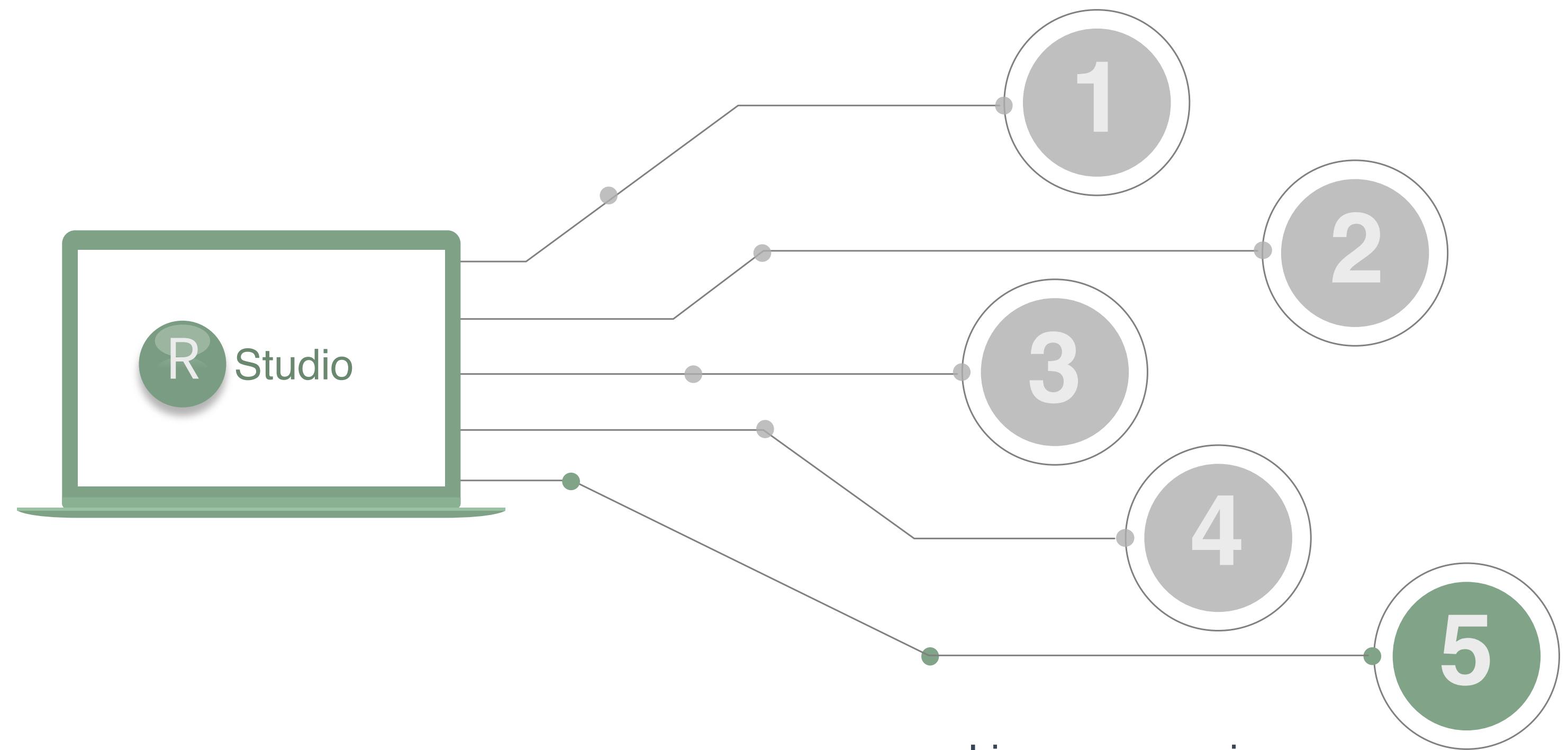
VARIABLES

R - A STATISTICAL SCRIPTING LANGUAGE



During this session:

- Cooperatively discuss and share ideas about the data
- Apply steps of basic statistical analysis for hypothesis testing consistent with the given data
- Suggest conclusions, based on your analysis, regarding the association between psoriasis and gene expression levels

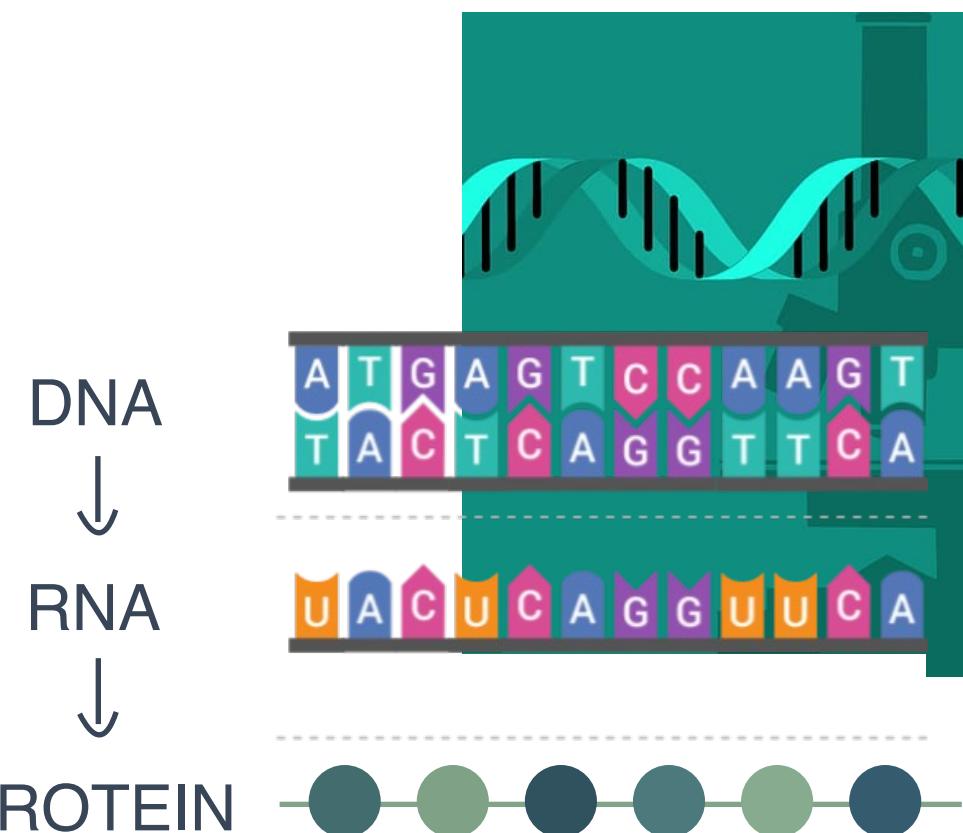


- Linear regression
- Summary Statistics
- ANOVA
- Logistic regression
- Clustering
- Correlation

— Statistics in R
EXERCISE 5

BIOINFORMATICS IN R

HIGH THROUGHPUT DATA



RNA SEQ

DATA



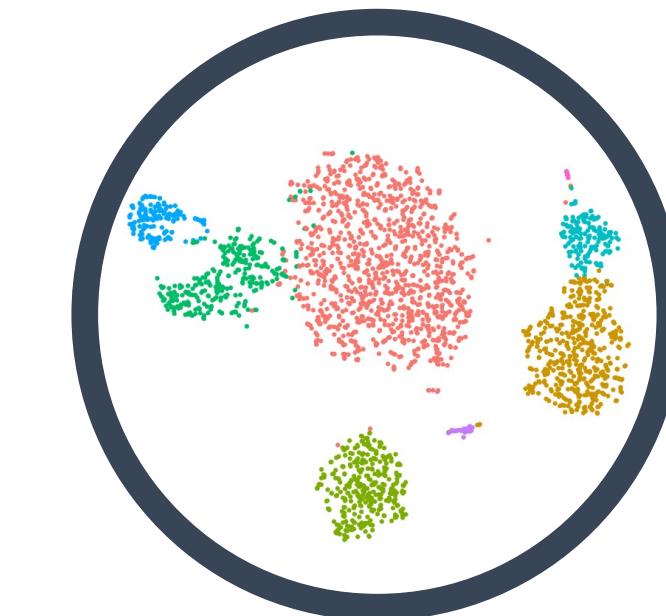
PROTEOMICS

DATA

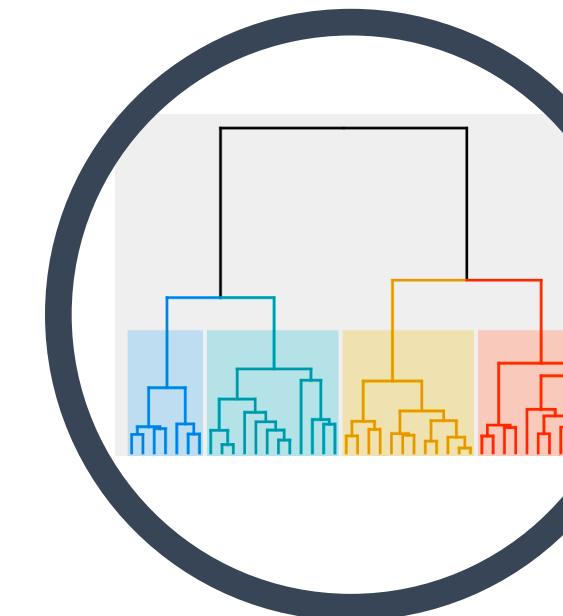


BIOINFORMATIC ANALYSIS

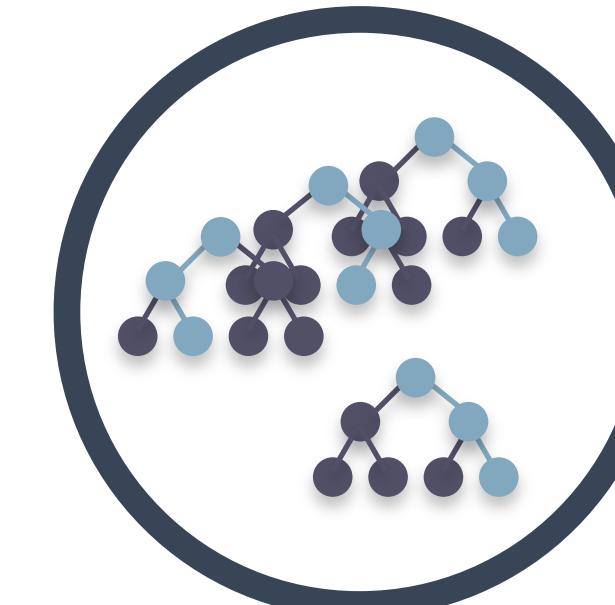
DIMENSIONALITY REDUCTION



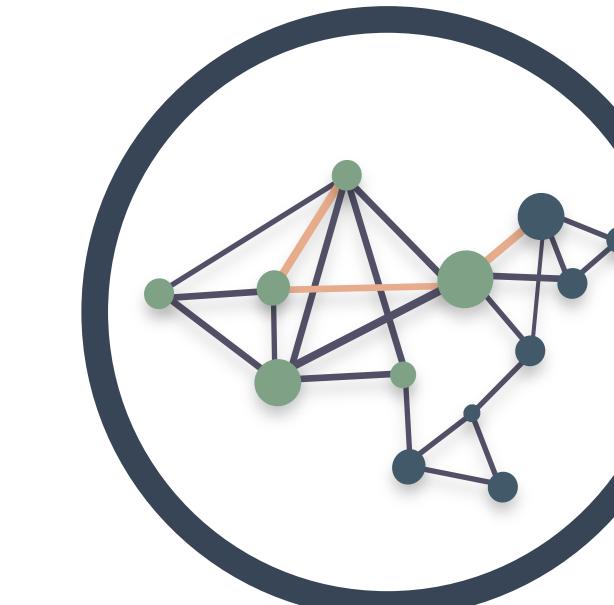
CLUSTERING



MACHINE LEARNING



NETWORK ANALYSIS



THE TOP OF THE R ICEBERG



STATISTICAL ANALYSIS

Statistical models (linear, generalized, mixed, ...)

Statistical tests (t-test, chisq, anova, ...)

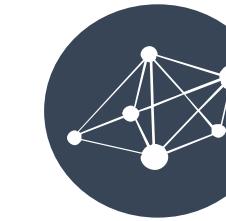
Survival analysis (Cox, Kaplan meier)



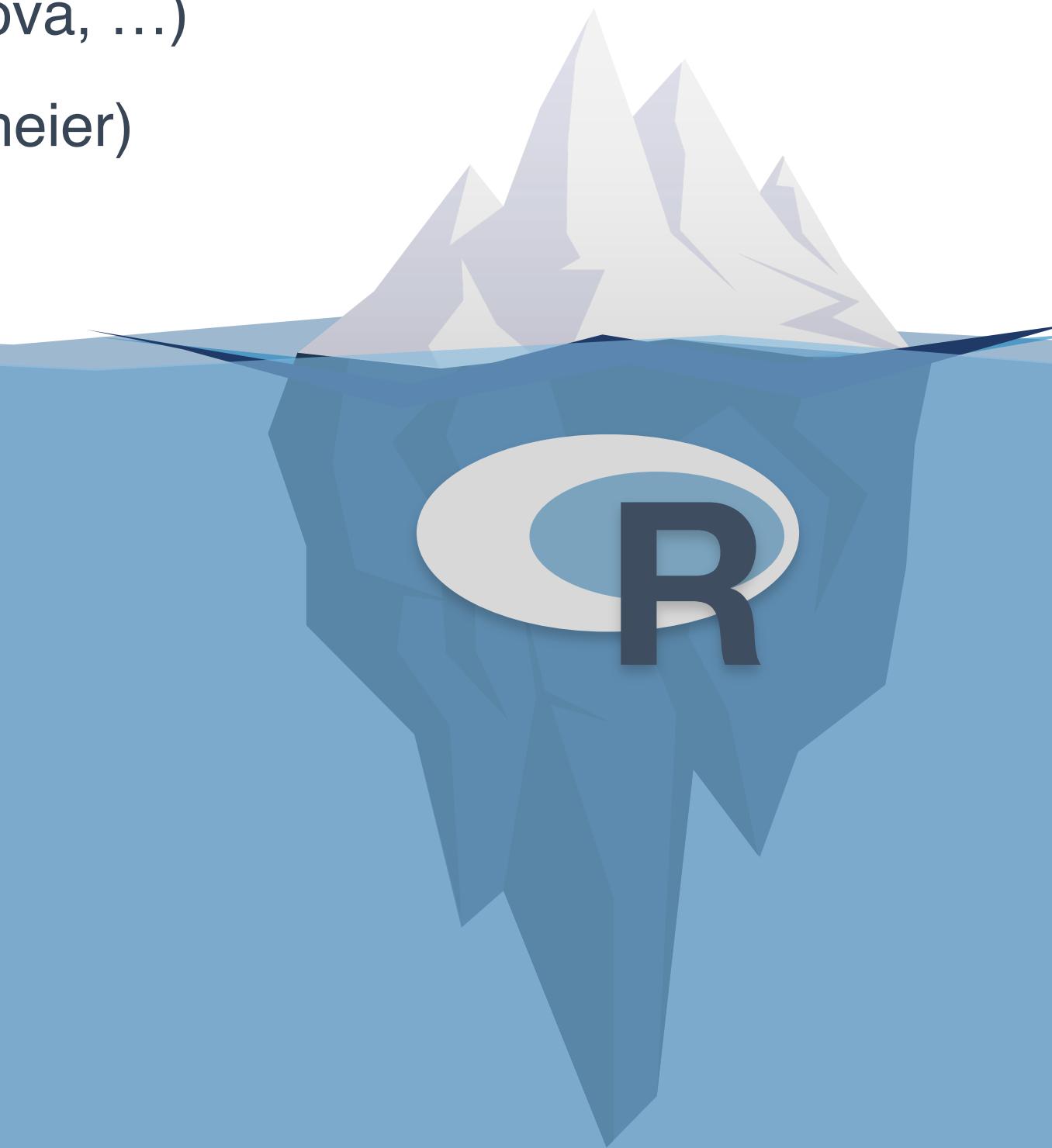
DATA MANGEMENT



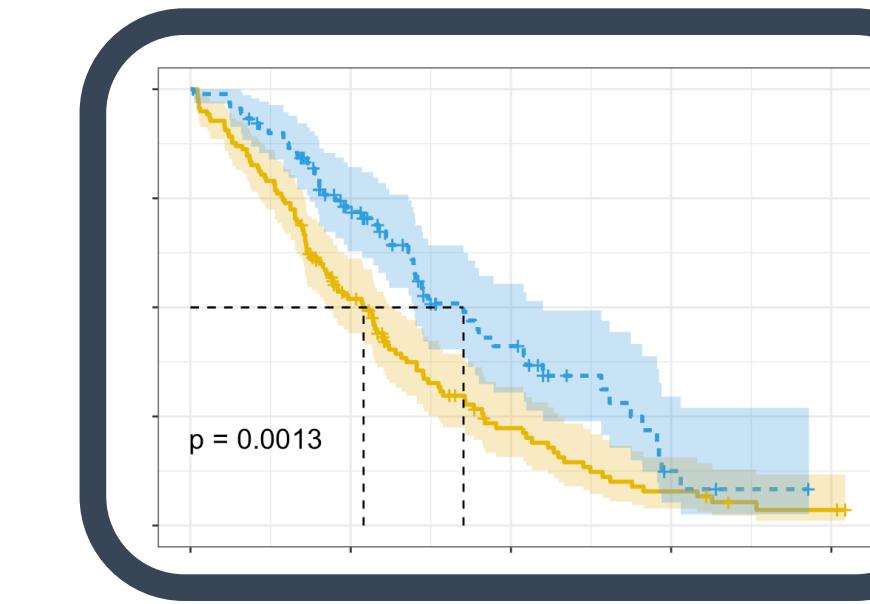
EASY PLOTTING



BIOINFORMATIC ANALYSIS

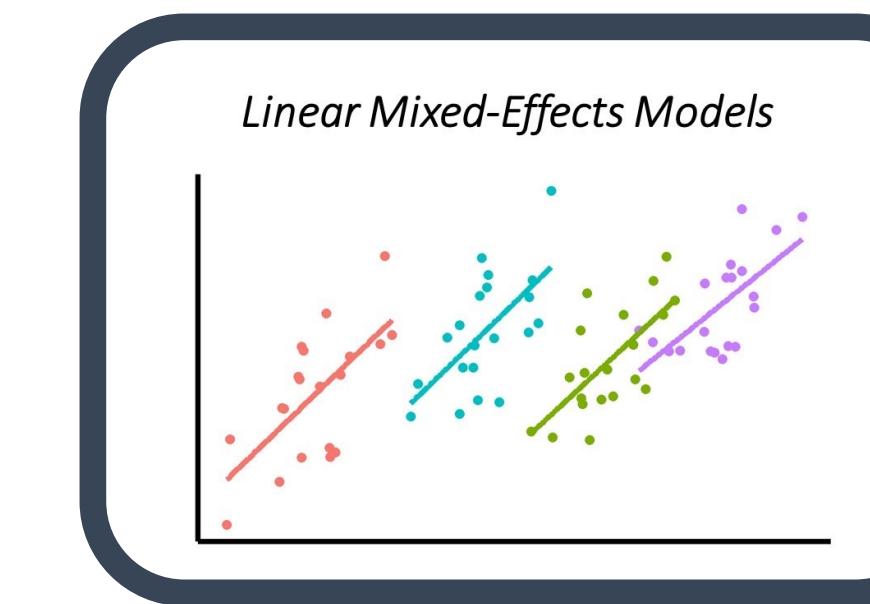


— TEASER STATISTICS in R



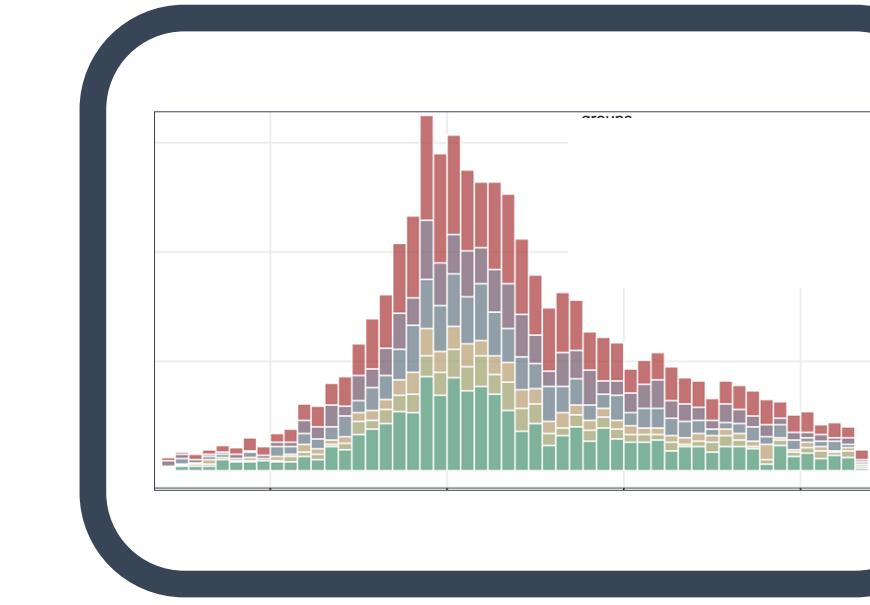
Survival Analysis

survival: <https://rviews.rstudio.com/2017/09/25/survival-analysis-with-r/>
survminer: <https://cran.r-project.org/web/packages/survminer/survminer.pdf>
(<https://rpkgs.datanovia.com/survminer/>)



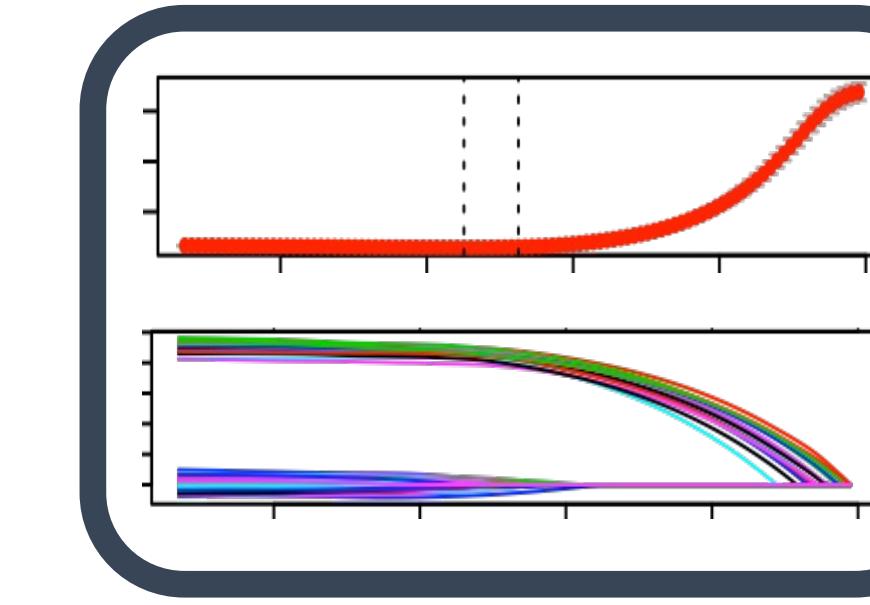
Mixed-Effects Models

lme4: <https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf>
<https://cran.microsoft.com/snapshot/2017-08-01/web/packages/sjPlot/vignettes/sjplmer.html>
glmmTMB: <https://cran.r-project.org/web/packages/glmmTMB/index.html>



Epidemiological Analysis

Epi: <https://cran.r-project.org/web/packages/Epi/index.html>
pubh: <https://rviews.rstudio.com/2020/03/05/covid-19-epidemiology-with-r/>
https://cran.r-project.org/web/packages/incidence/vignettes/customize_plot.html
<https://rviews.rstudio.com/2020/03/05/covid-19-epidemiology-with-r/>



Elastic-Net Regression

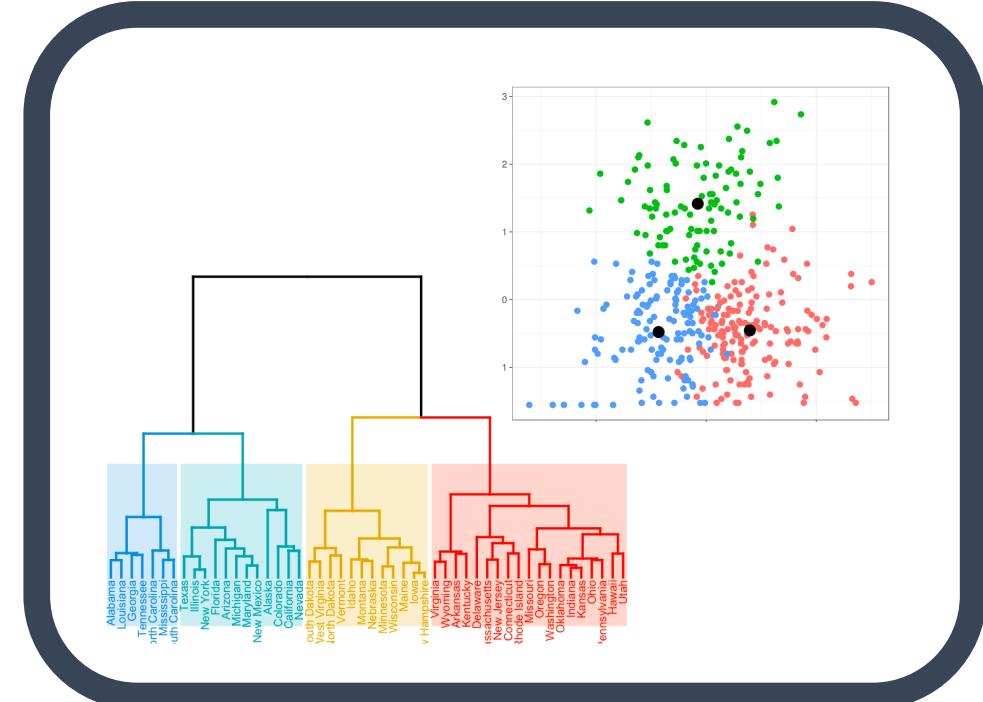
(R
glmnet: <https://cran.r-project.org/web/packages/glmnet/glmnet.pdf>
elasticnet: <https://cran.r-project.org/web/packages/elasticnet/elasticnet.pdf>
<https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net>

TEASER

Machine Learning

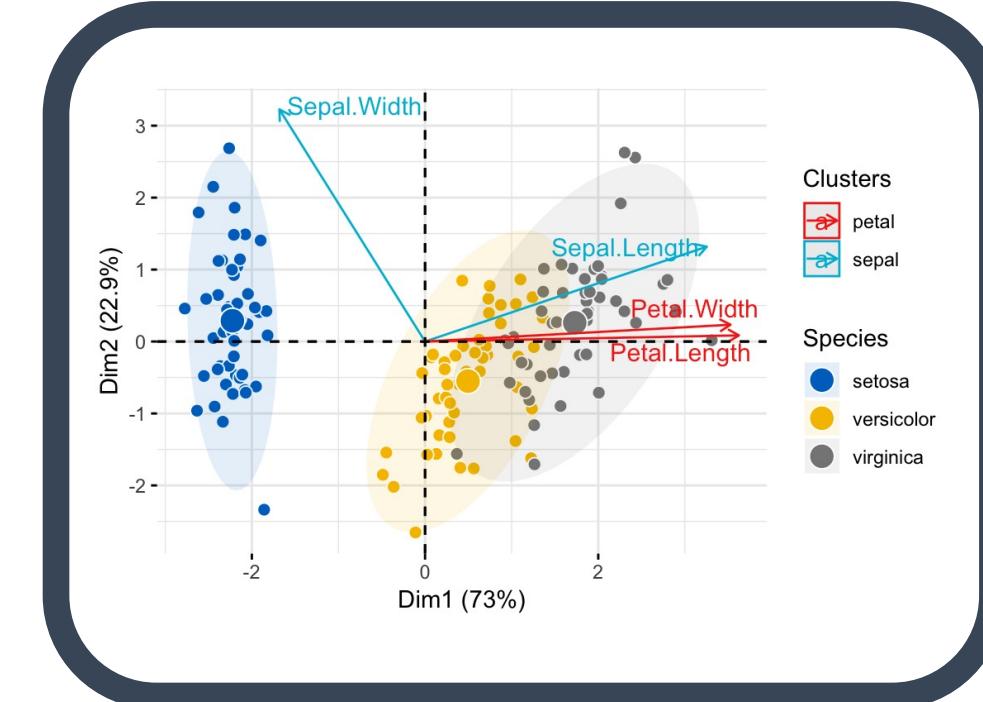
<https://lgatto.github.io/IntroMachineLearningWithR/an-introduction-to-machine-learning-with-r.html>

Clustering



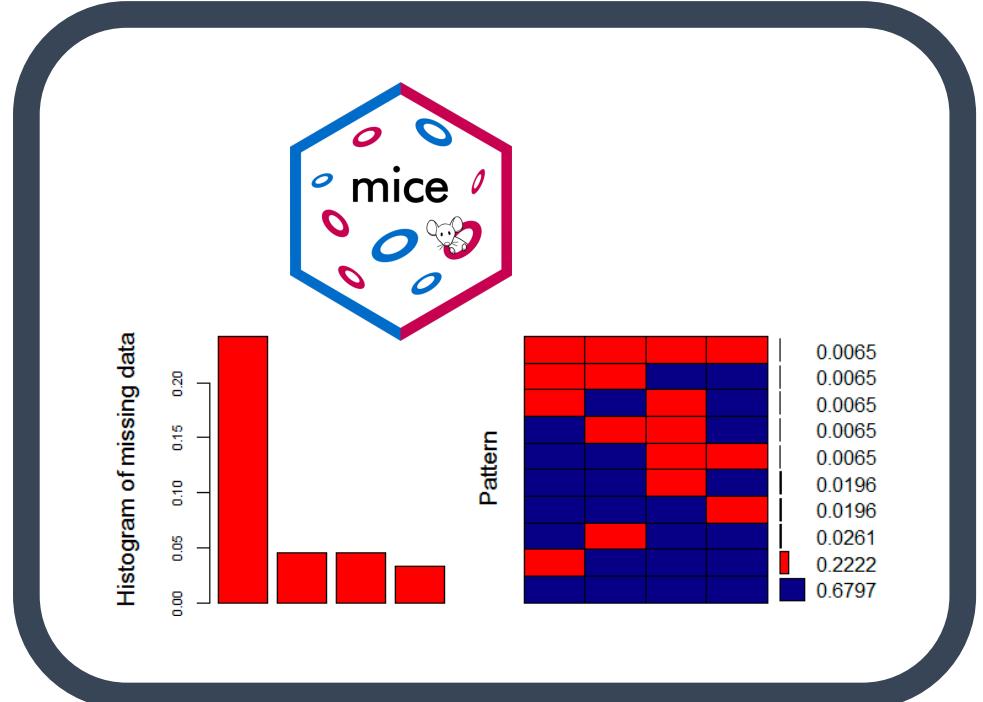
<https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/>

Feature Selection: PCA



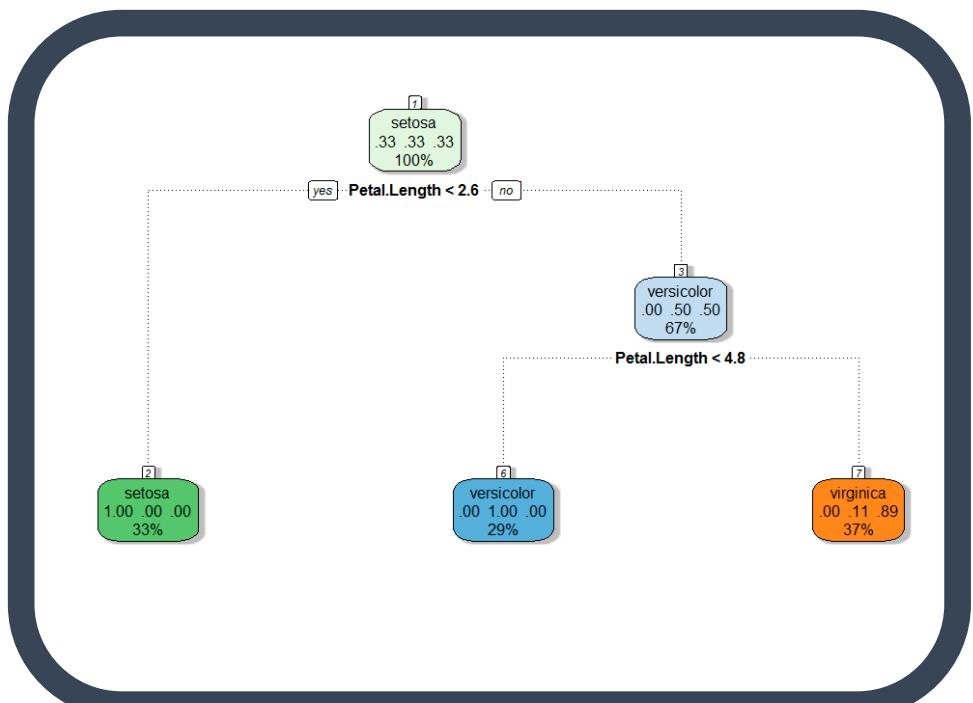
<https://bioconductor.org/packages/release/bioc/vignettes/PCAtools/inst/doc/PCAtools.html>

Missing Data



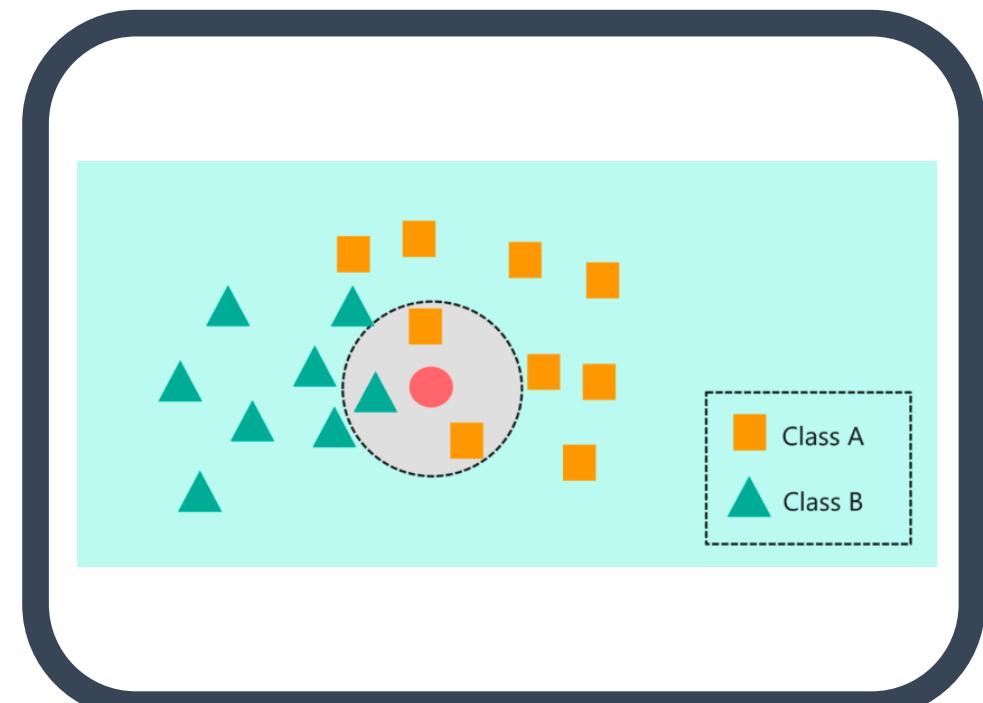
<https://amices.org/mice/>
<https://datascienceplus.com/imputing-missing-data-with-r-mice-package/>

Random Forest



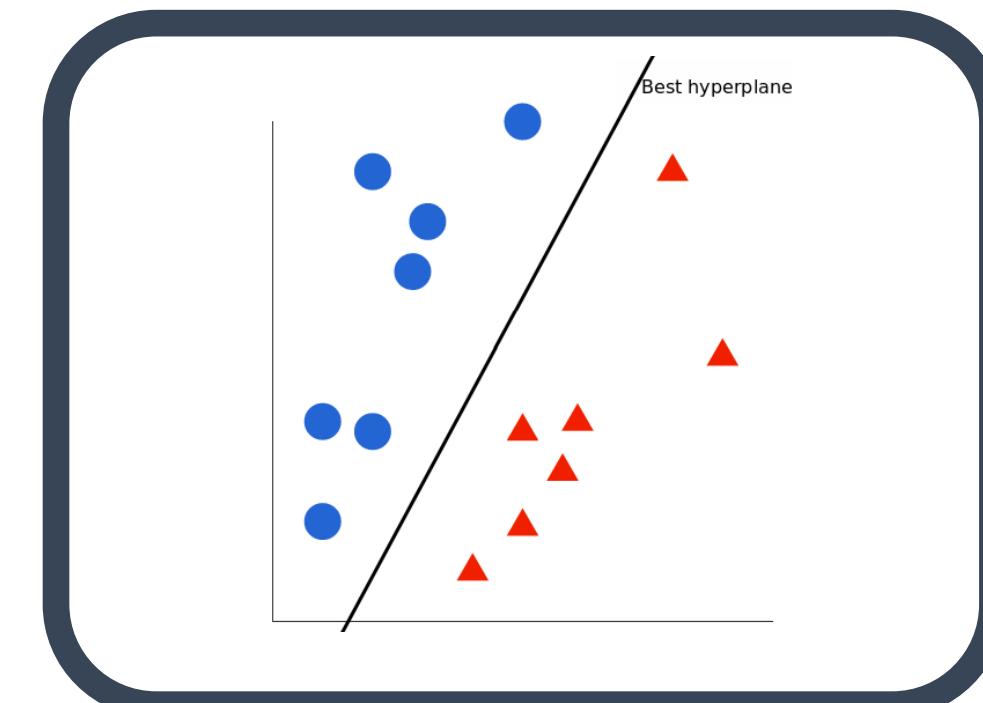
<https://www.blopig.com/blog/2017/04/a-very-basic-introduction-to-random-forests-using-r/>

kNN



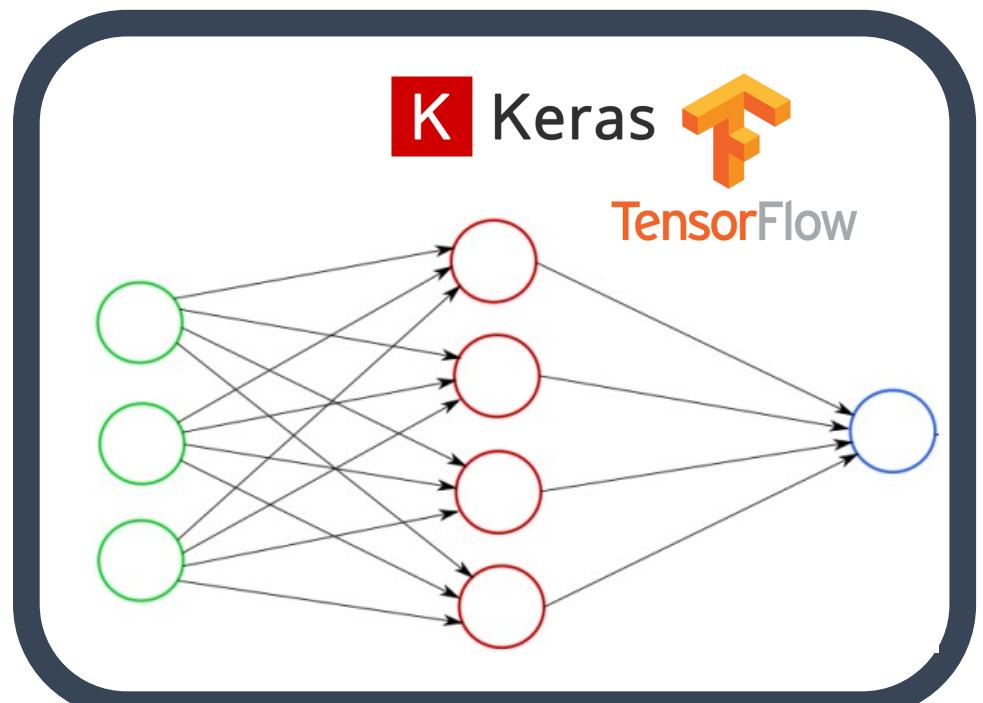
<https://www.edureka.co/blog/knn-algorithm-in-r/>

SVM



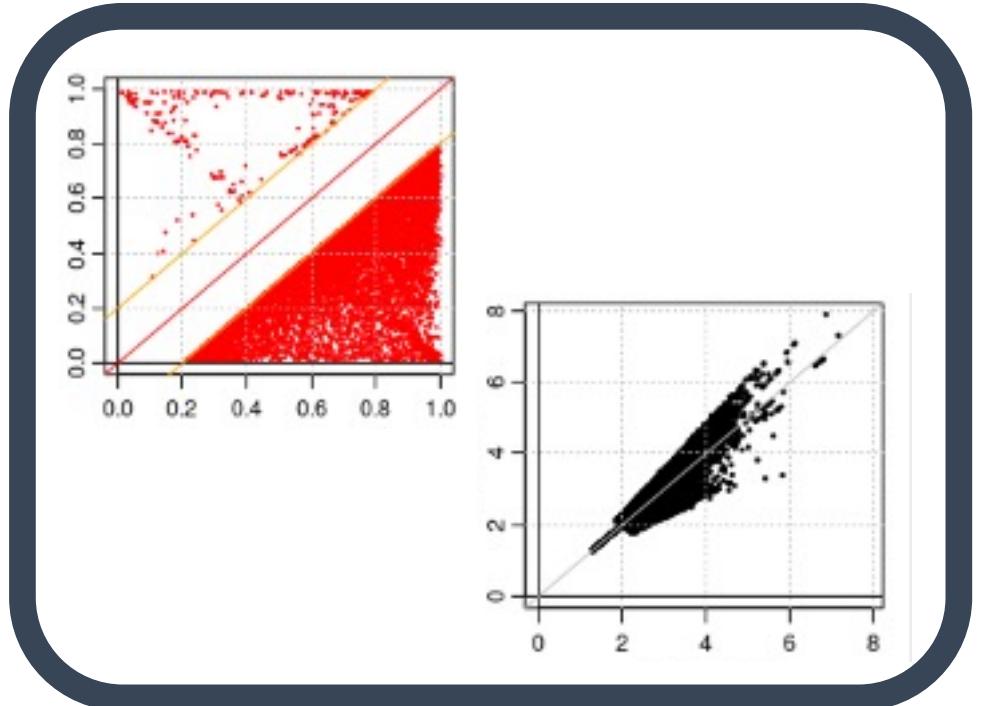
<https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>

Neural Networks



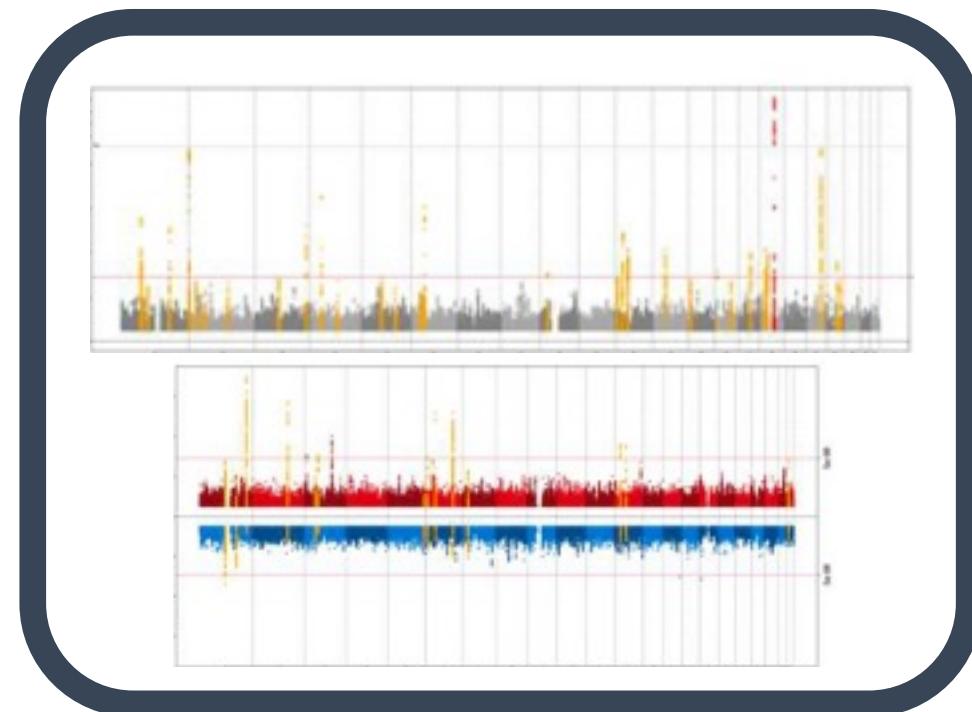
<https://keras.rstudio.com/>
<https://tensorflow.rstudio.com/>

GWAS - QC & Data Harmonization



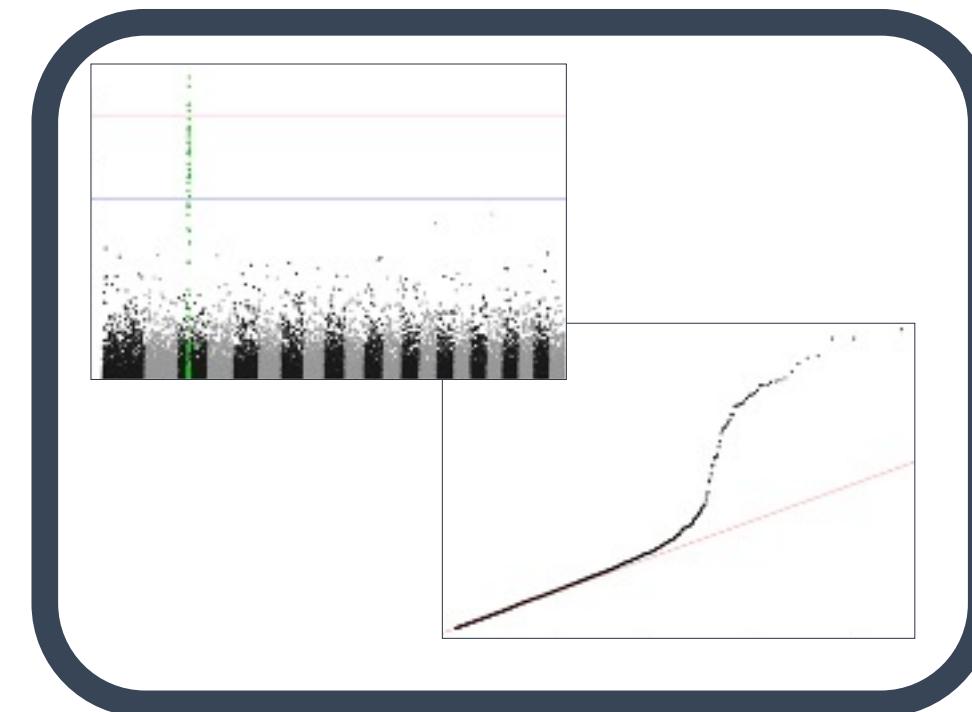
EasyQC: <https://www.uni-regensburg.de/medizin/epidemiologie-praeventivmedizin/genetische-epidemiologie/software/>

GWAS Data Management & Plots



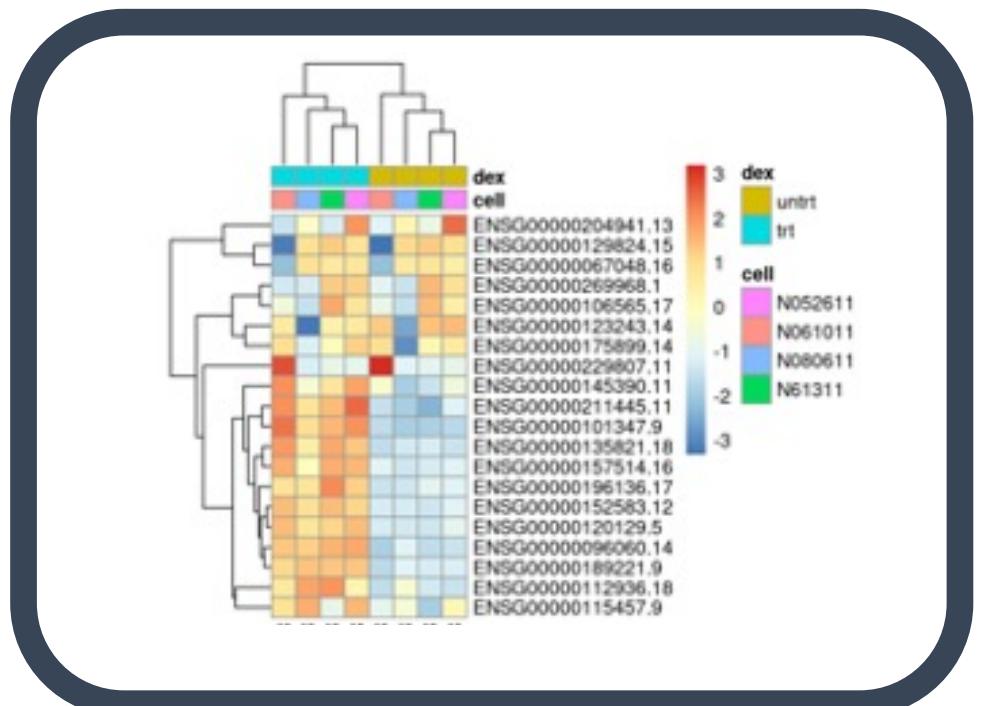
EasyStrata: <https://www.uni-regensburg.de/medizin/epidemiologie-praeventivmedizin/genetische-epidemiologie/software/>

More Plotting...



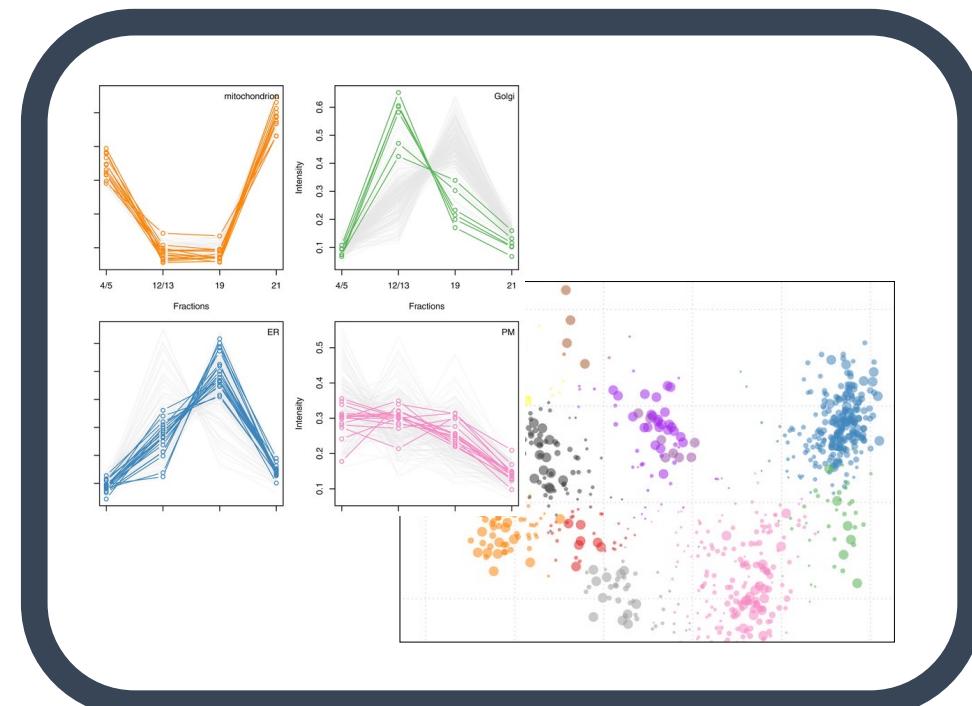
Manhattan and QQ plots:
<https://cran.r-project.org/web/packages/qqman/vignettes/qqman.html>

Gene Expression Analysis



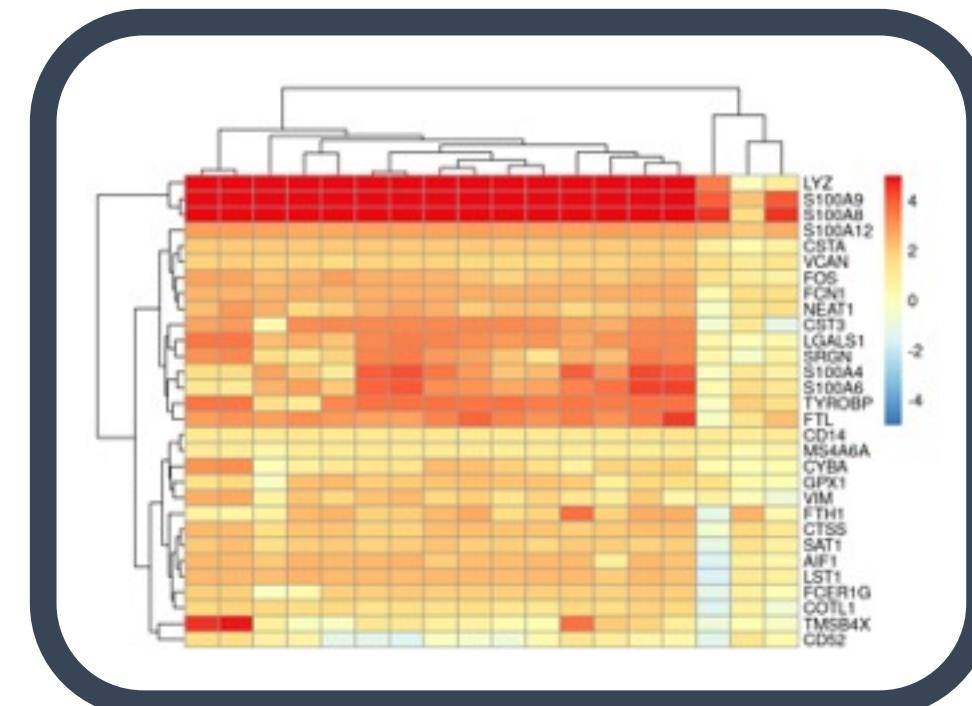
DESeq2, limma, EdgeR, etc.:
http://www.bioconductor.org/packages/release/BiocViews.html#_RNASeq

Proteomics Analysis



RforProteomics:
http://www.bioconductor.org/packages/release/BiocViews.html#__Proteomics_RforProteomics.html

Single-Cell RNASeq



<https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>

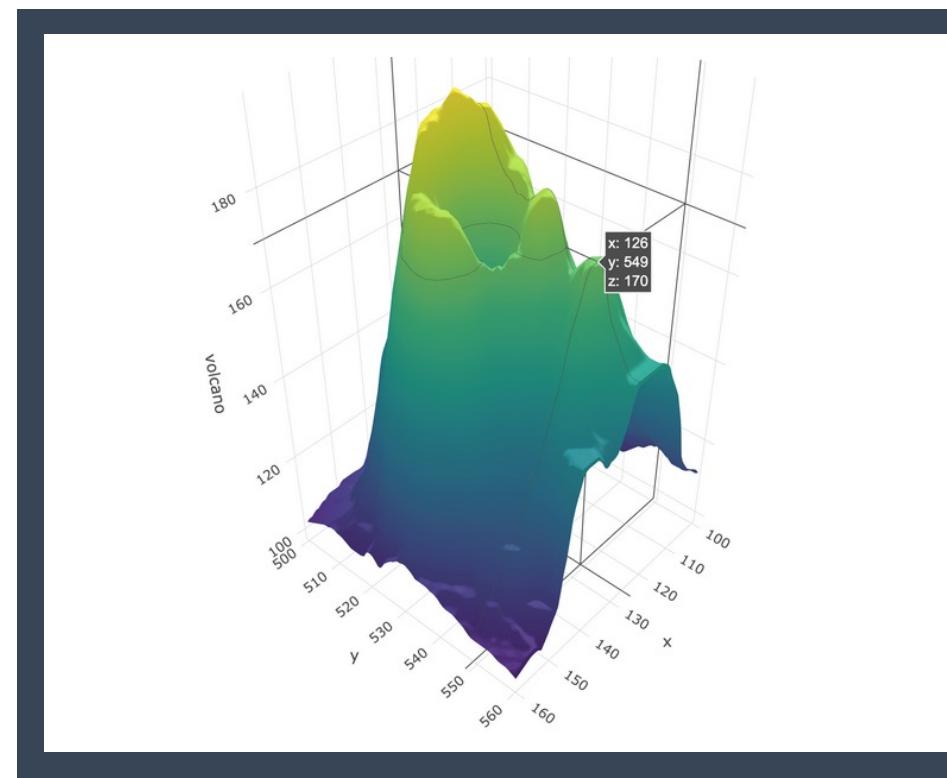
— TEASER Omics Data

<http://www.bioconductor.org/packages/release/BiocViews.html>

COOL STUFF IN R

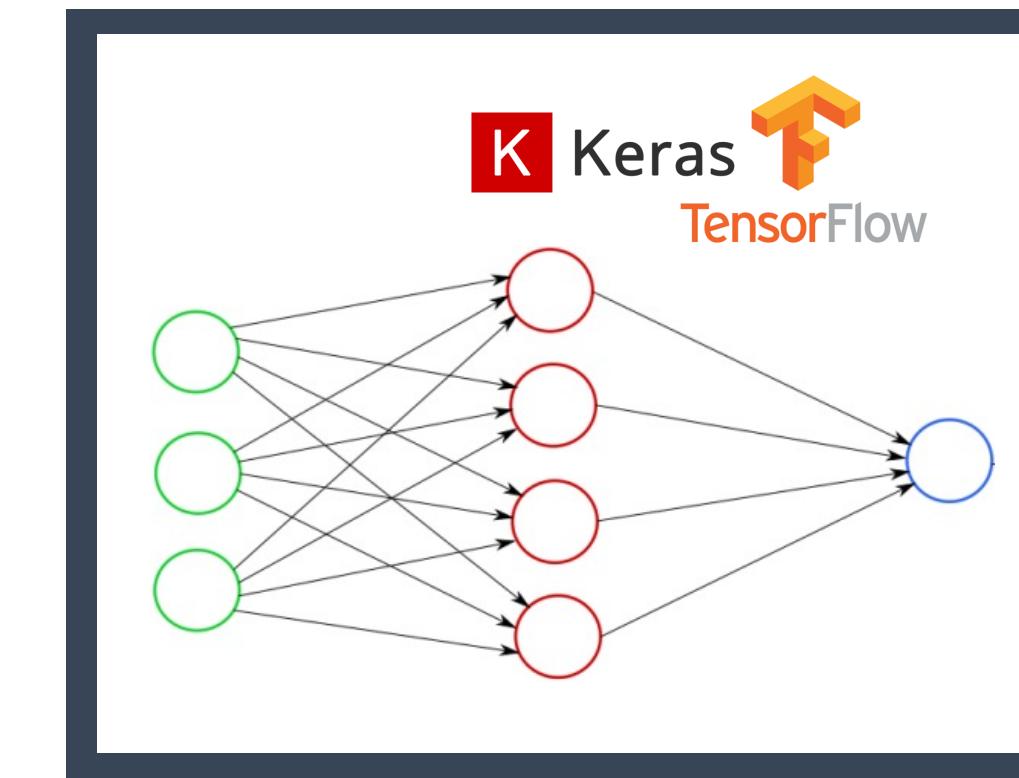
— FROM EXCEL TO R

PLOTTING IN 3D



<https://plotly-r.com/d-charts.html>

DEEP LEARNING



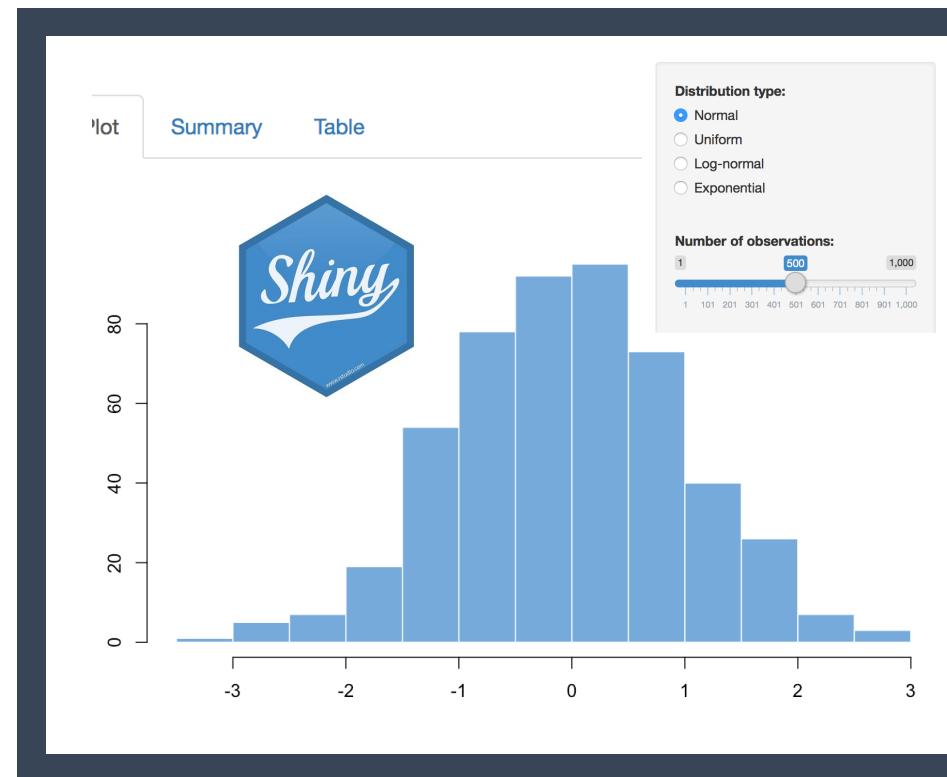
<https://keras.rstudio.com/>
<https://tensorflow.rstudio.com/>

BAYESIAN STATISTICS



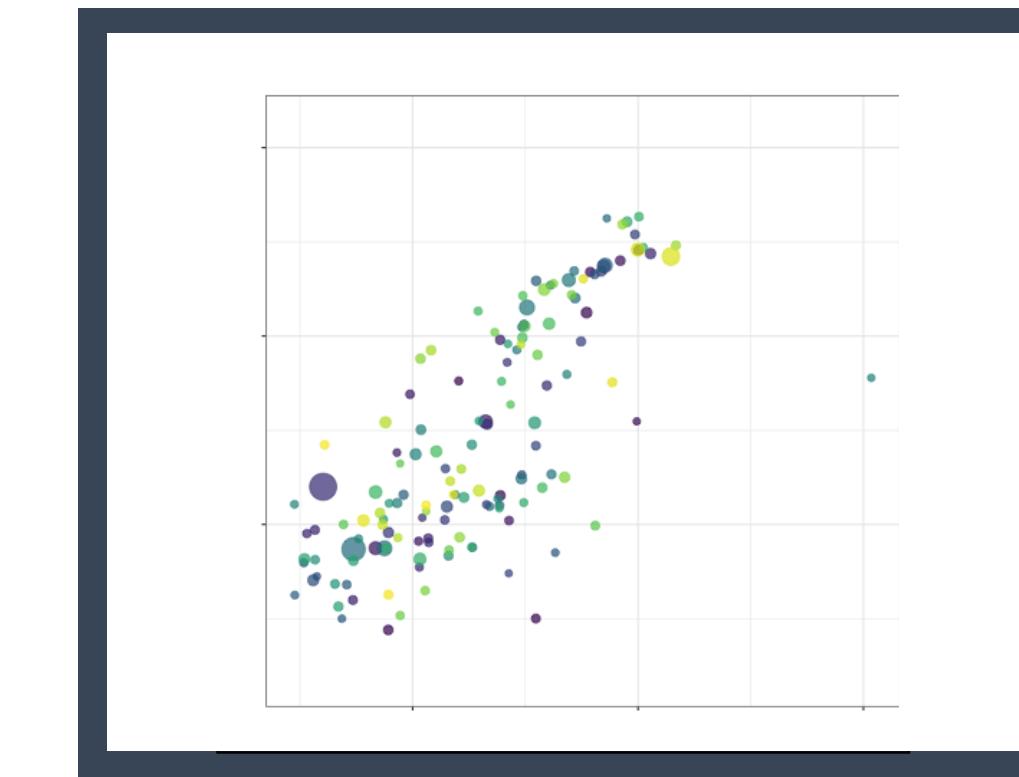
<https://mc-stan.org/users/interfaces/rstan>

WEBPAGE WITH R SHINY



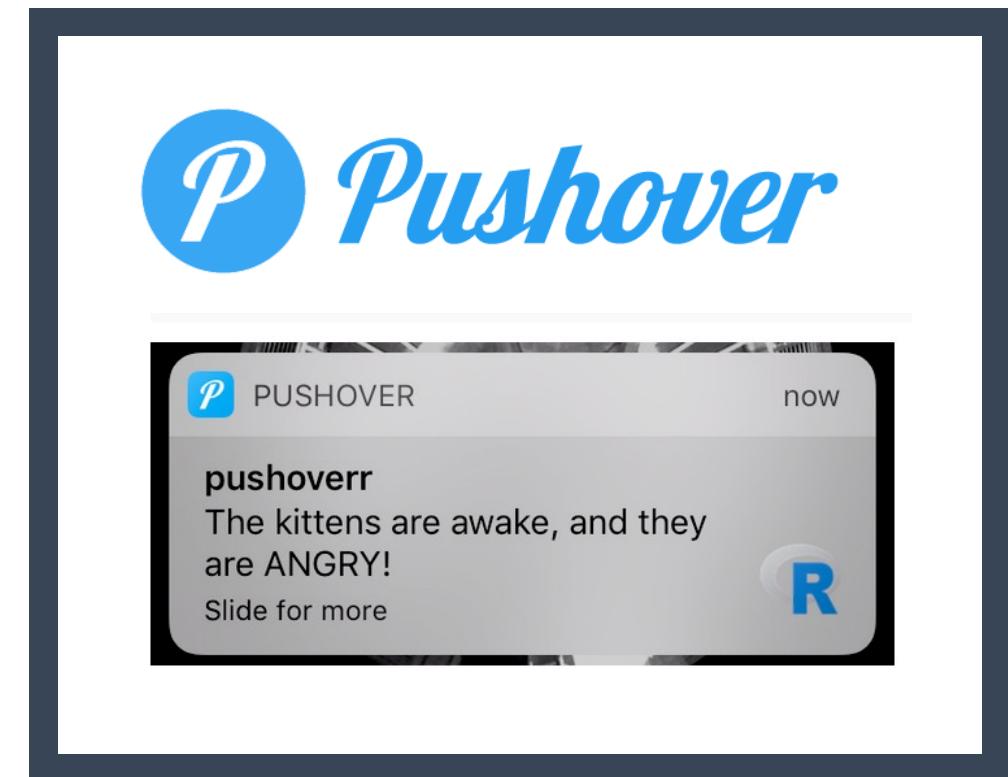
<https://shiny.rstudio.com/>

INTERACTIVE PLOTS



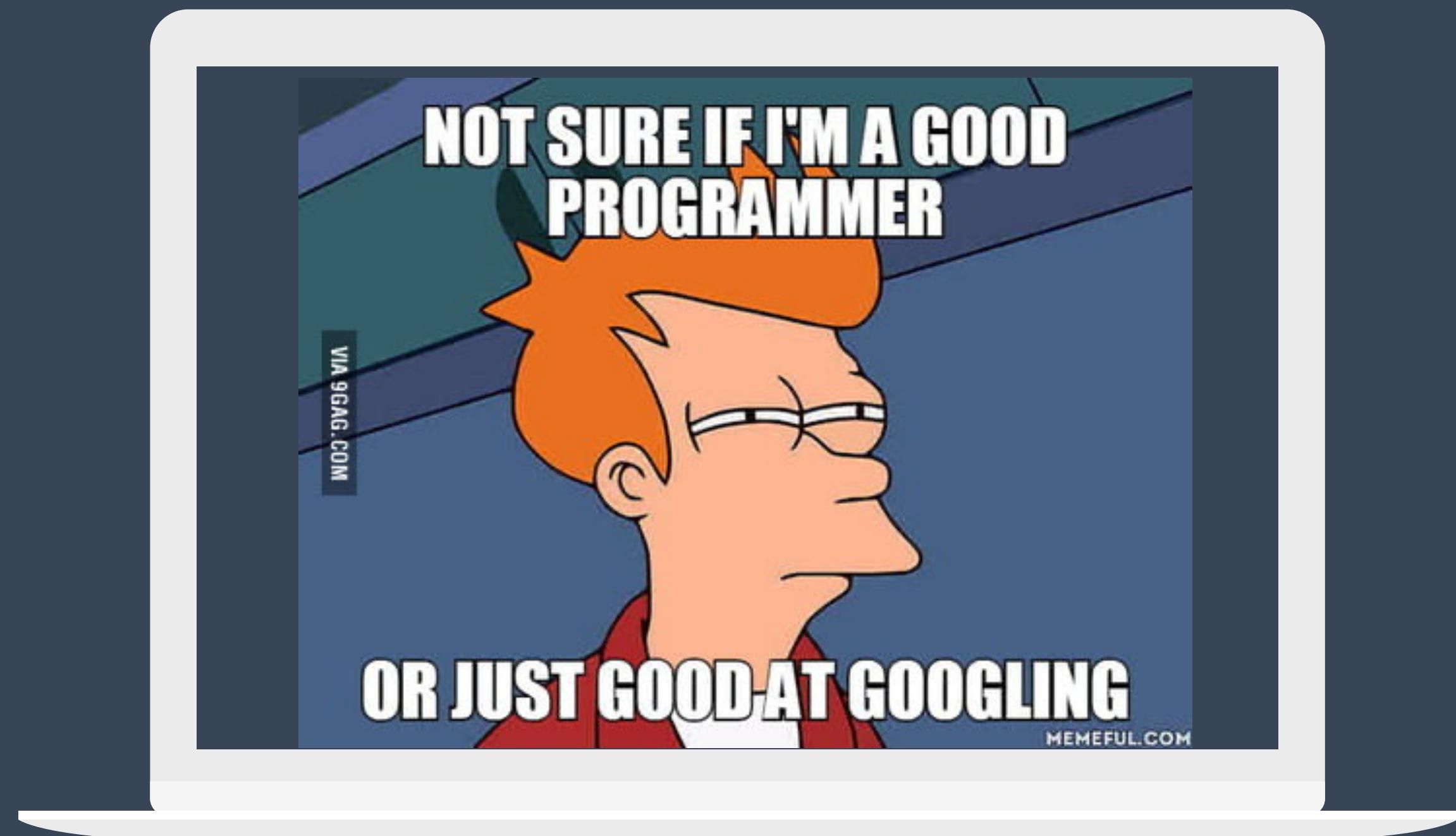
<https://gganimate.com/articles/gganimate.html>

MAIL AND MESSAGES



<https://github.com/briandconnelly/pushoverr>

THANK YOU FOR LISTENING



This keynote presentation was created by Thilde Terkelsen,
Data Scientist, Center for Health Data Science, SUND, KU.
For internal use at KU only, do not distribute commercially.