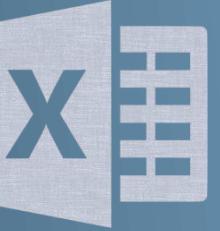


FROM  EXCEL TO 



WHO ARE WE?

Center for Health Data Science (**HeaDS**):

- Data Science Research Groups
- HDS Sandbox (Infrastructure Project)
- **SUND Data Lab**

<https://heads.ku.dk/>



Thilde Terkelsen



Helene Wegener



Chelsea Lennox



Tugce Karaderi

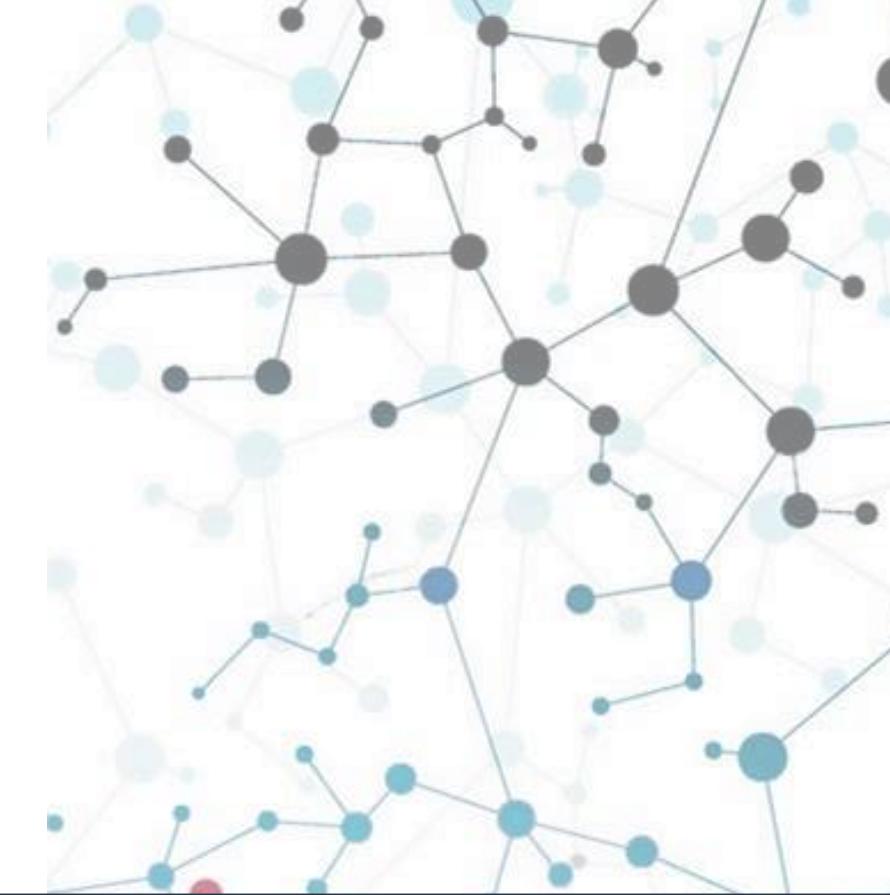
SUND Data Lab:

- Courses & Workshops, Seminars, etc.
- Health Data Science Consultations
- Commissioned Research & Supervision
- Hub for Data Science (Matchmaking)

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



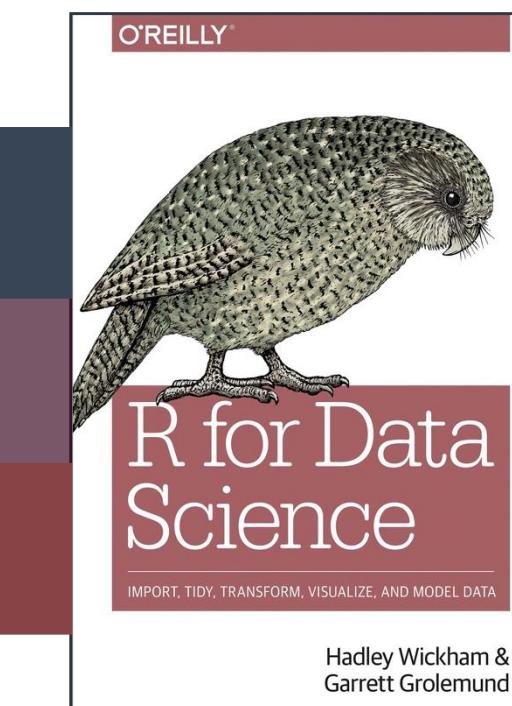
THE PRACTICALS



Two days: 9.00-16.00. There will be coffee breaks, we promise ☕

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

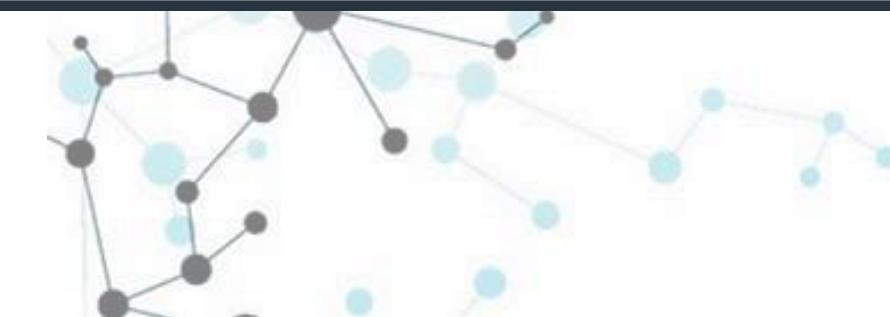
The course is build on hands-on presentations & exercises



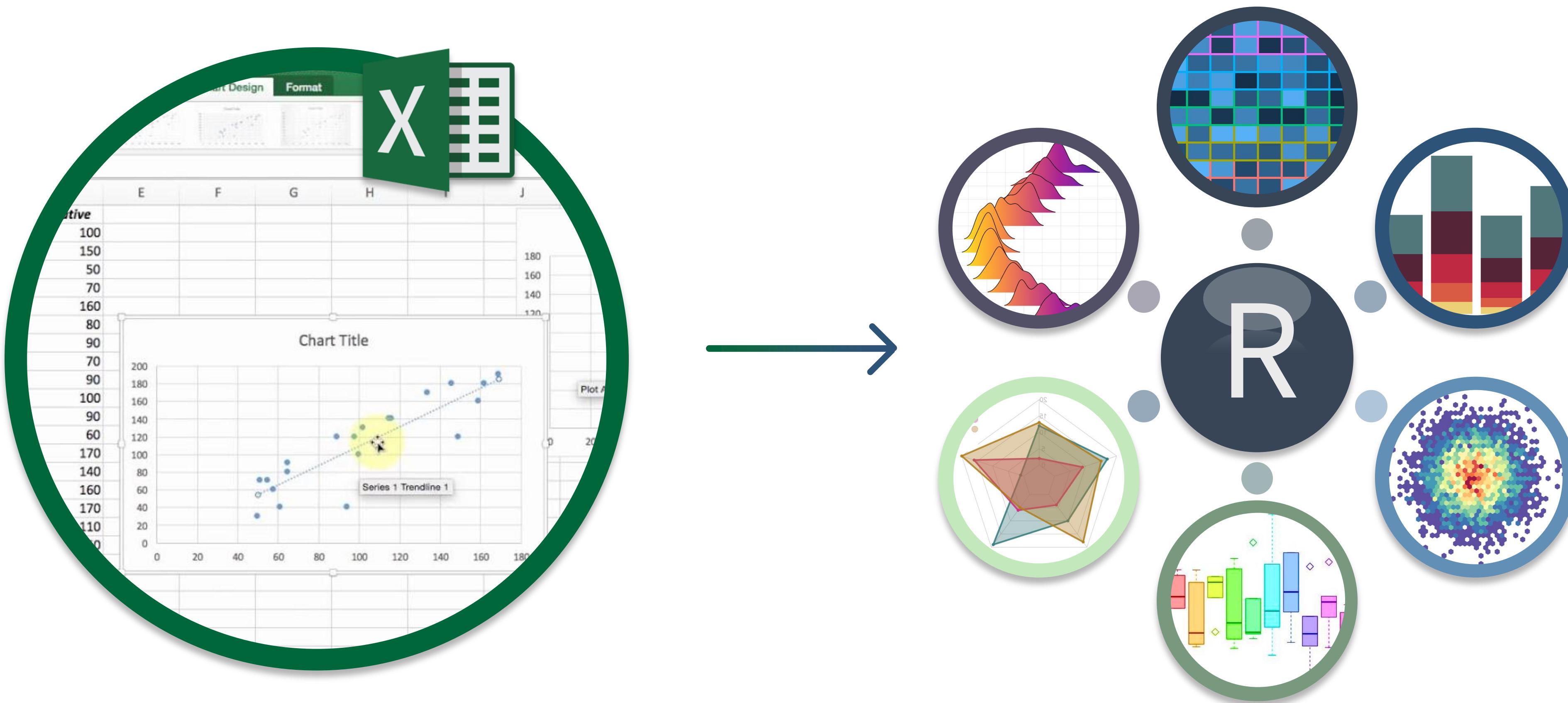
Download and install the newest version of R (<https://cran.r-project.org/>)

Download and install the newest version of R-studio (<https://posit.co/download/rstudio-desktop/>)

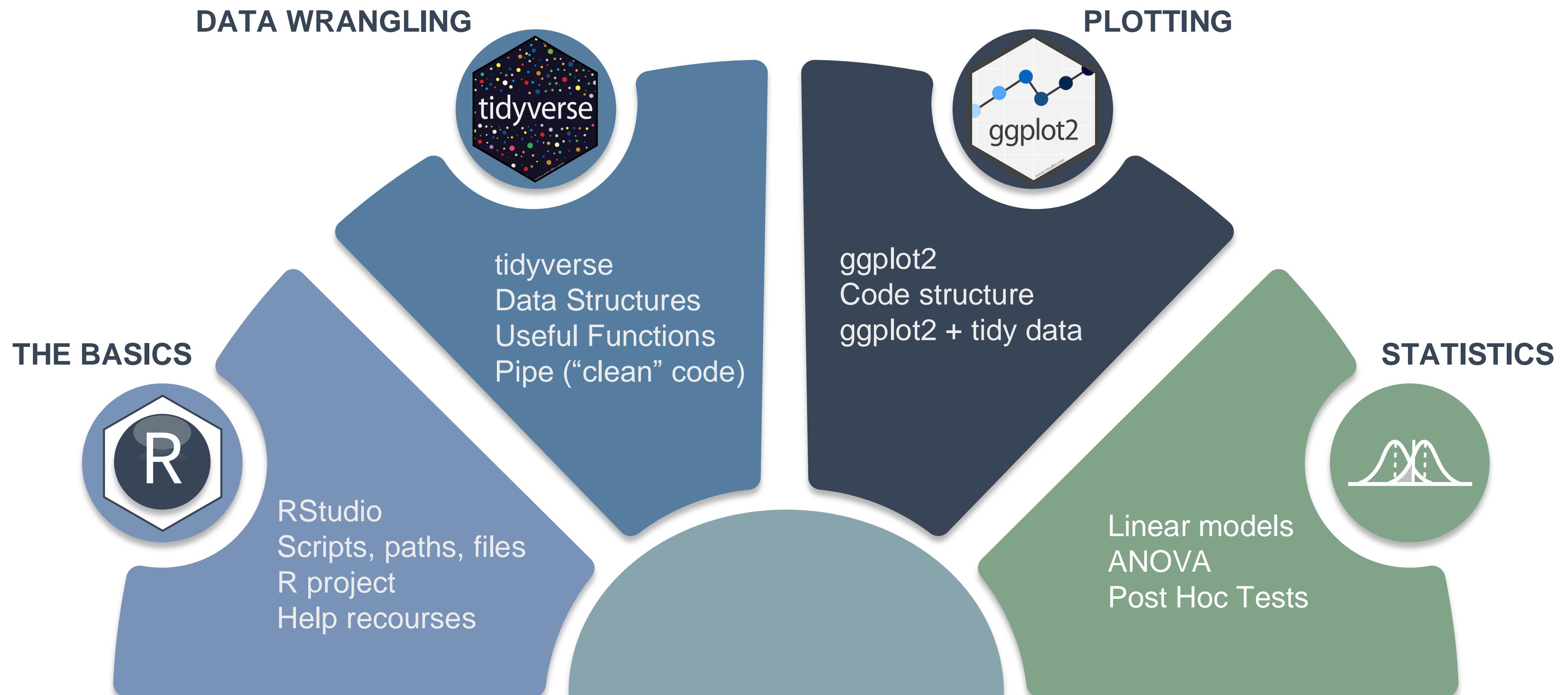
Go to course website: <https://center-for-health-data-science.github.io/FromExceltoR/>



— WELCOME TO
FROM EXCEL TO R



WHAT WILL I LEARN?



PROGRAM

DAY 1

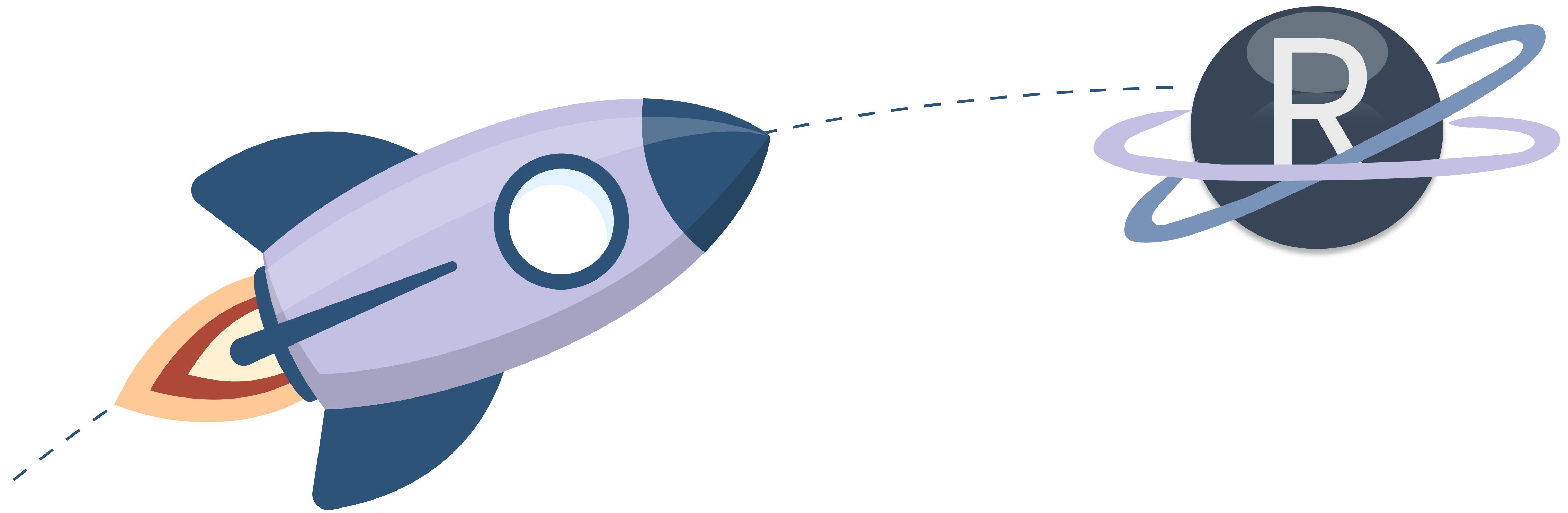
08:30 - Installation Issues + Coffee
09:00 - Introduction to R Basics
10:00 - Rstudio Exercise
10:45 - Break
11:00 - Tidyverse
12:00 - Lunch
13:00 - Tidyverse Exercise
14:30 - Break
14:45 - ggplot2
15:15 - ggplot2 Exercise
16:00 - See you tomorrow

DAY 2

08:30 - Coffee + Optional Q&A
09:00 - Applied Statistics
10:00 - Applied Statistics
10:45 - Break
11:00 - Applied Statistics
12:00 - Lunch
13:00 - Data Analysis Exercise
14:30 - Break
14:45 - Data Analysis Exercise
15:45 - Cool things in R + Course Evaluation
16:00 - Bye Bye Bye!

— FROM EXCEL TO R

LET'S GET STARTED



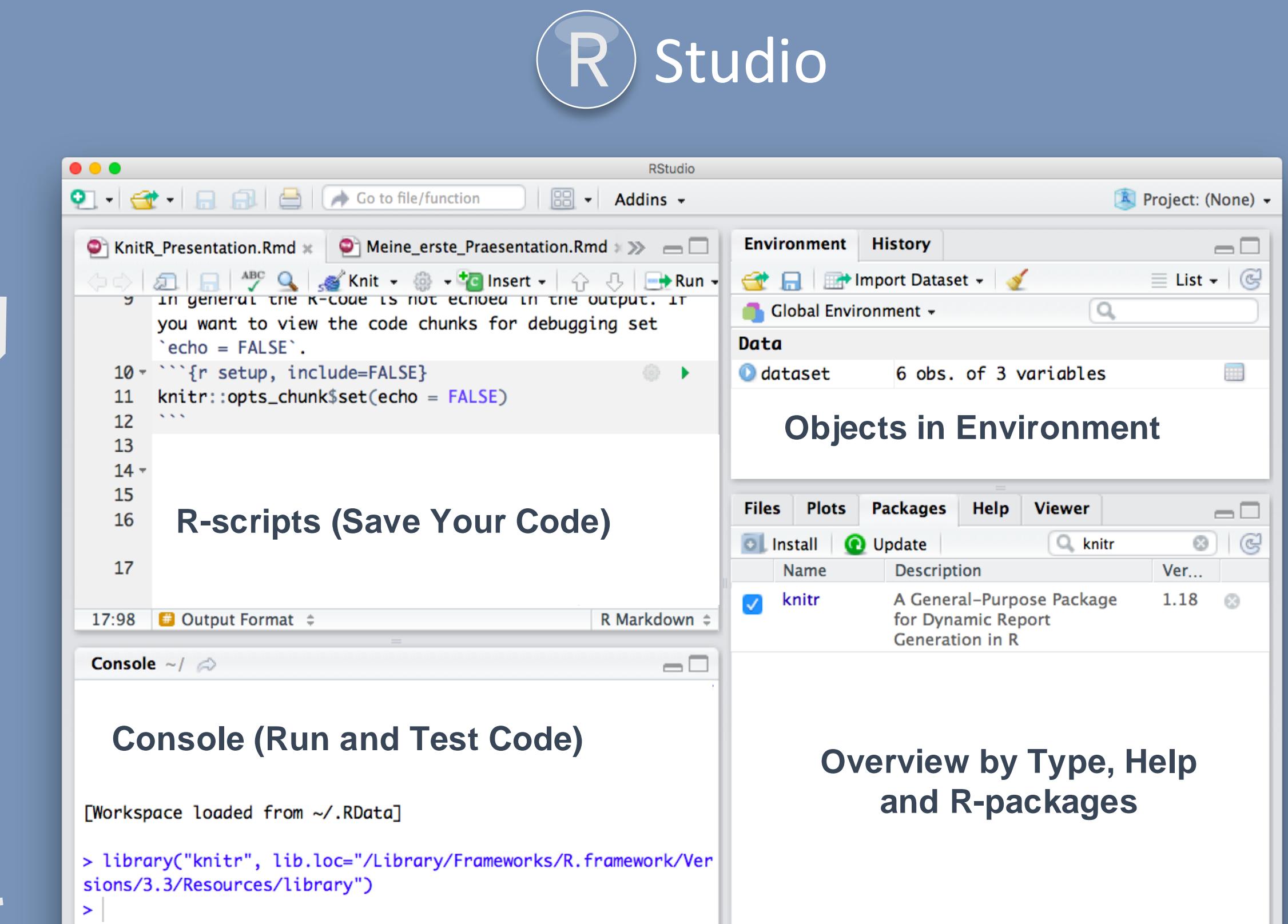
R & FRIENDS



Scripting / Programming Language



Reports (html, pdf, word)



R Code Interpreter and Editor

THE ANATOMY OF R

The **programming language**  may be used from within in the **Code Editor** 

The R code you write may be saved in **scripts** with extensions:



Functions perform actions based on **input arguments** and return an output:

`mean(x)`
`sd(x)`

Packages are bundles/libraries of functions.



QUARTO DOCUMENTS



- Tool which supports **reproducible** coding/publication (reports)
- Quarto works with a **markup language**
- A markup language is **text-coding system** which specifies structure & format of a document and the text within it
- We **render** (compile) a Quarto → html, pdf, word, website

Pointers for quarto:

- Separate sections with headers
- Do not overfill code chunks
- Write meaningful and precise comments

90% of all code comments:

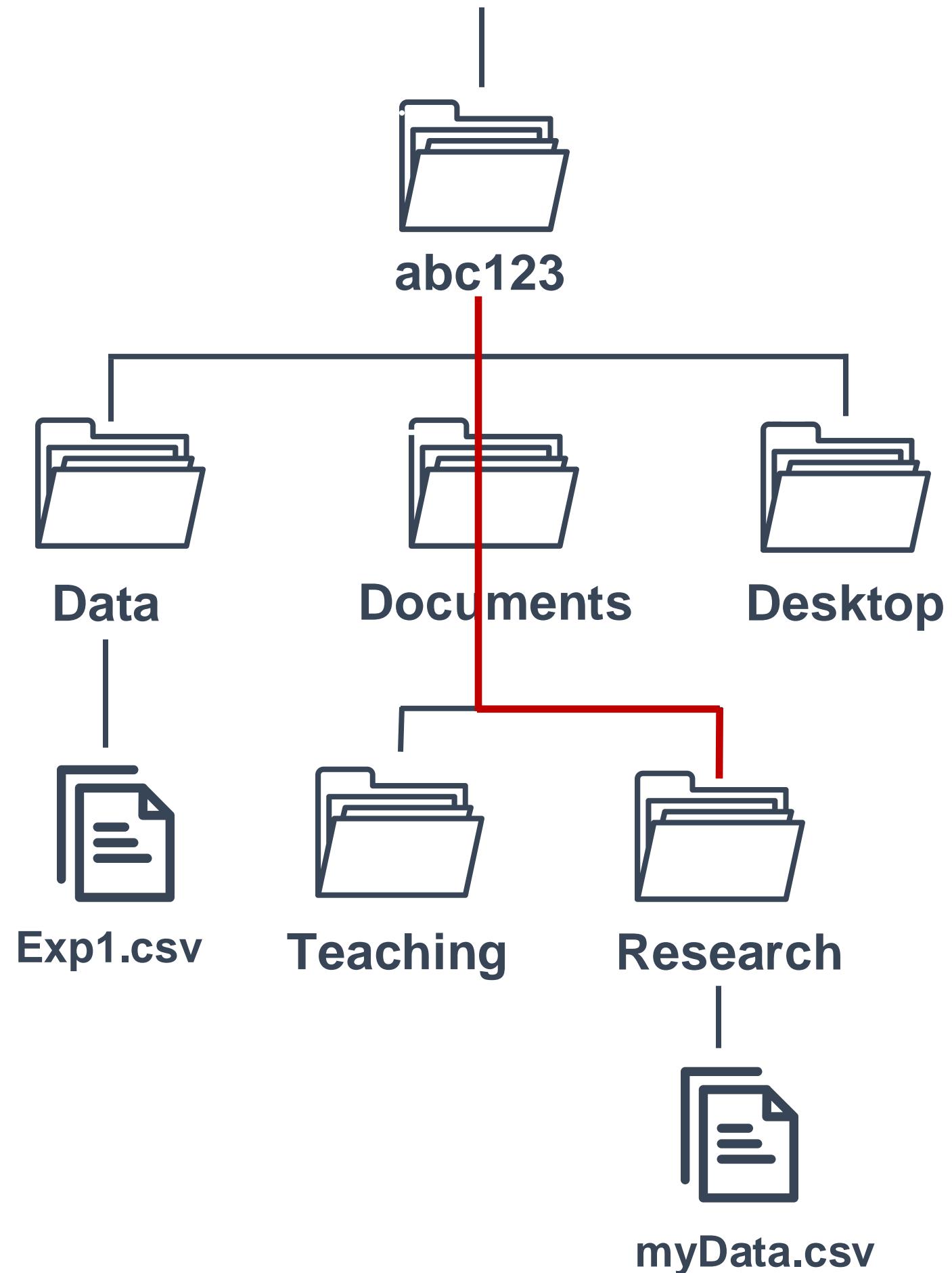


PATHS – LOCATION, LOCATION, LOCATION

- The directories on your computer are organized in a **File Tree**
- **Move** between directories or **point to** a place in the tree with a specific file

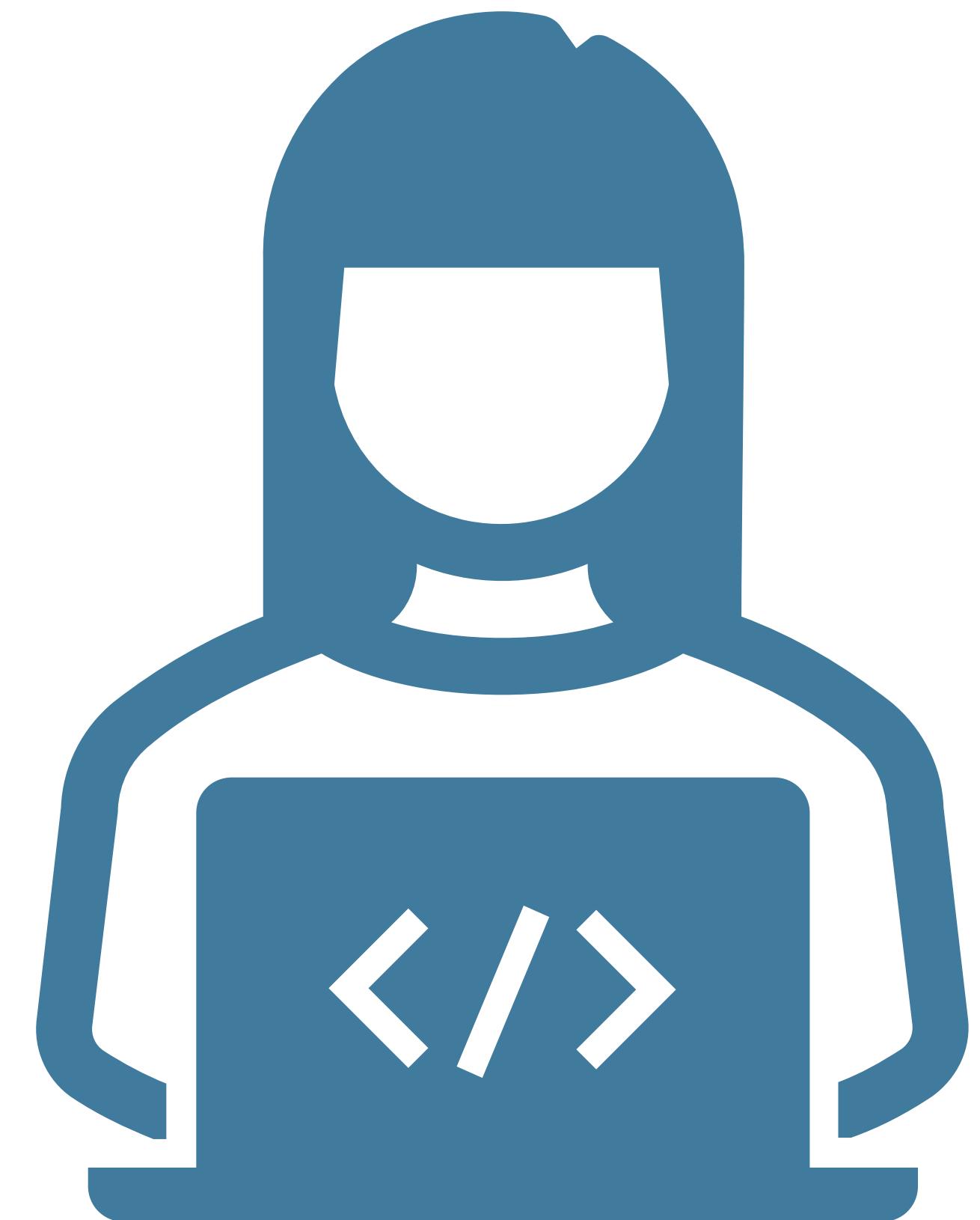
We do this by specifying a **path**:

- **Move to:** /abc123/Documents/Research
- **Point to:** /abc123/Documents/Research/myData.csv
- The place you are, right now, is called the **working directory** (wd)
Where am I: `getwd()`
Go to : `setwd('path')`



— FROM EXCEL TO R

LIVE CODING 1 – INTRO TO R



QUATO CHEAT SHEET

YAML parameter:

```
---
```

```
title: My Project Name
```

```
output:
```

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

```
---
```

Code Chunk:



Source mode:
```{r}  
# some R code  
```

Visual mode:
{r}
some R code

Code Options:

```
{r echo = FALSE} # don't print code (default is TRUE)  
{r eval = FALSE} # don't run code (default is TRUE)  
{r error = FALSE} # don't display error message (default is TRUE)  
(Can also be set for warning and message)
```

Figure Options:

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

GETTING STARTED

CHUNK OPTIONS

Source mode:

Header

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

Text

italics
bold
`highlighted`

Lists

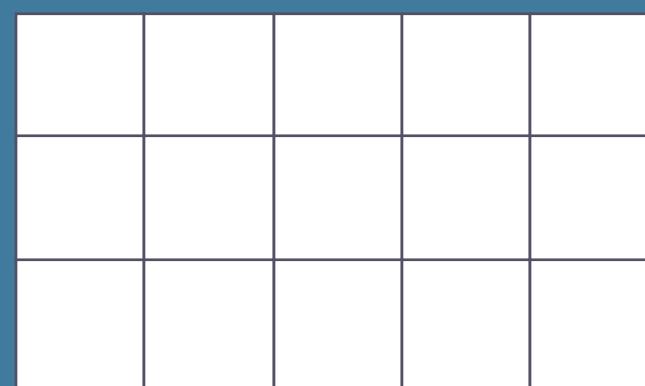
- * List item1 (filled dot)
 - + sub-item1 (open dot)
- 1. List item1 (numbered)
 - i) sub-item1 (roman)

TEXT

FROM EXCEL TO R

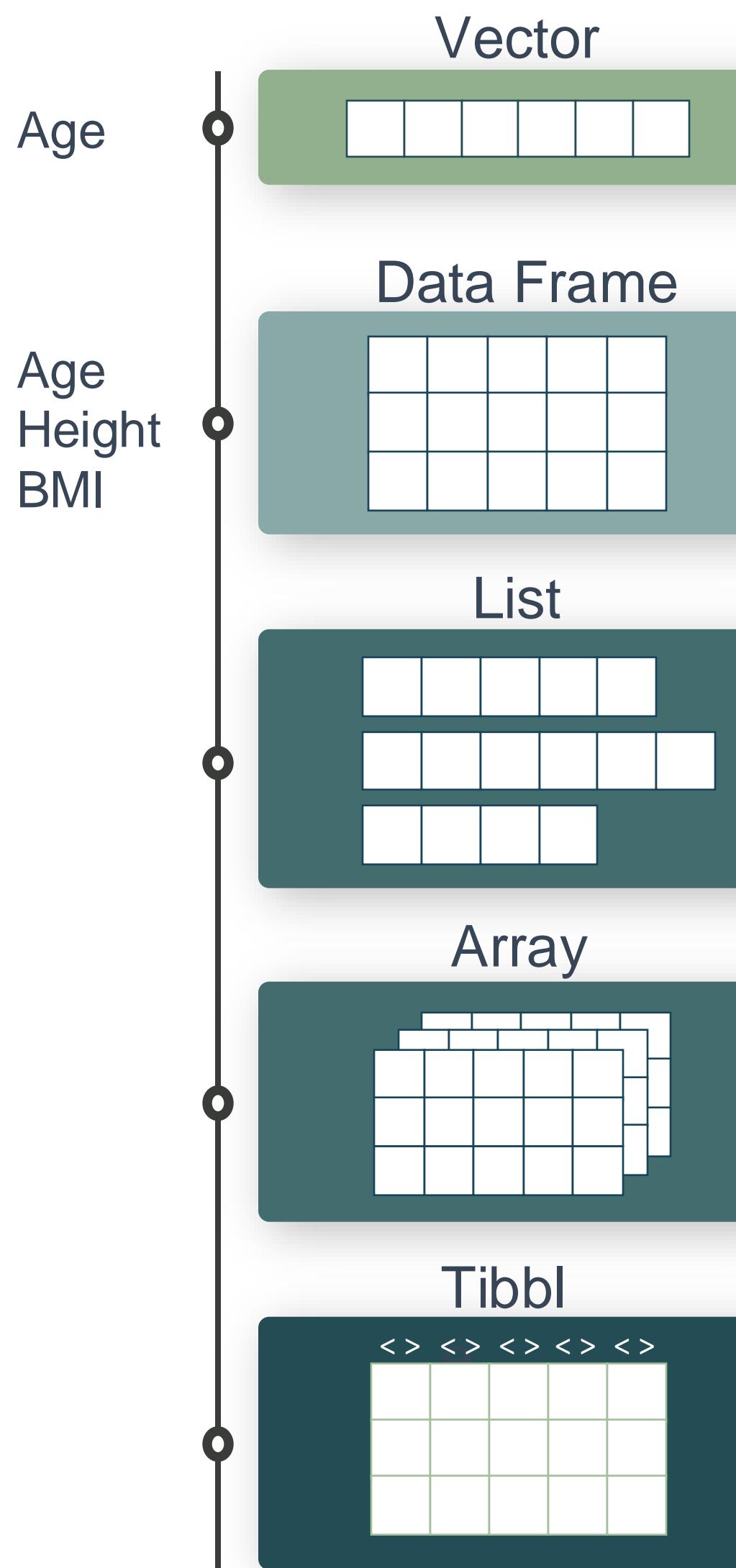
R DATA TYPES & STRUCTURES

VARIABLES

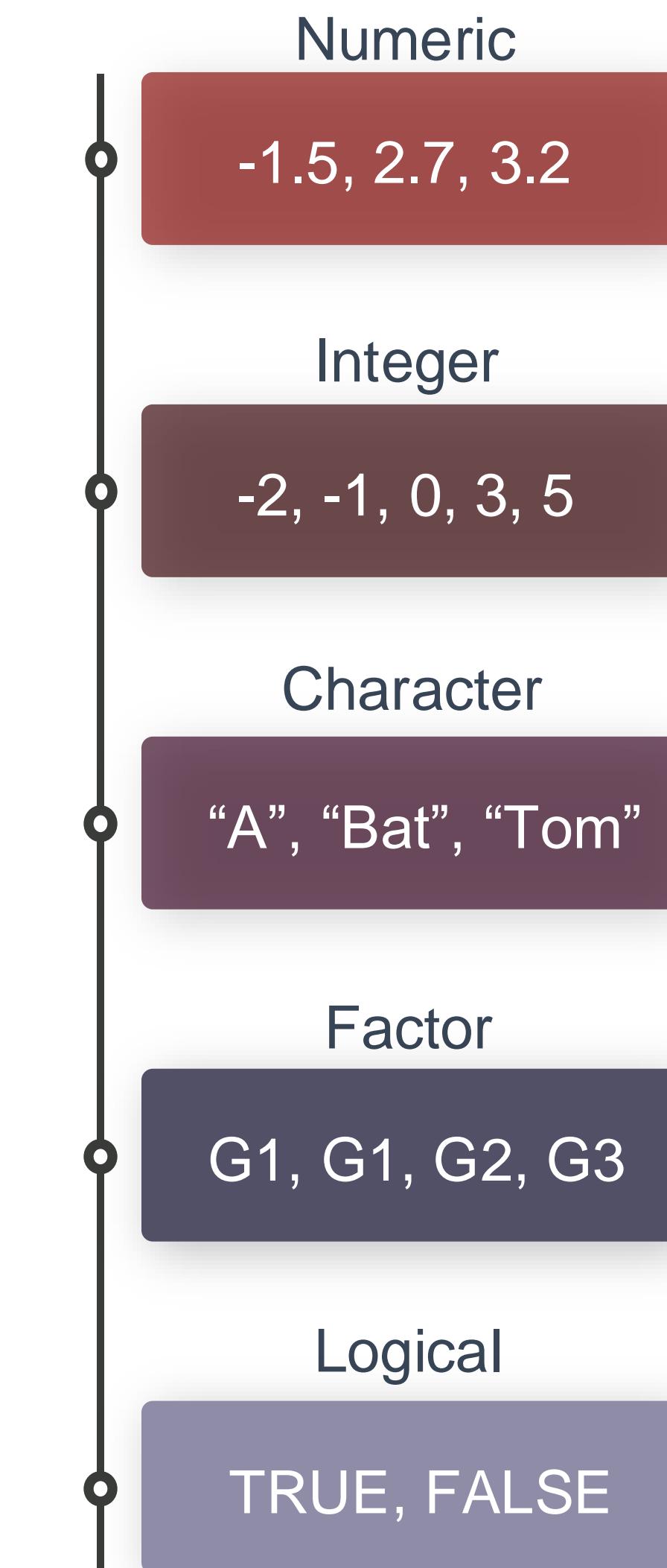


OBSERVATIONS

DATA STRUCTURES



DATA TYPES



INTRO CHEAT SHEET

Basics:

```
getwd() # location  
install.packages("package_name") # install packages  
library(package_name) # load packages
```

Overview:

```
head(df, n=10), tail(df, n=10) # first or last 10 rows  
unique(), table(), count() # unique vals, count vals
```

Type/Class:

```
class() # get data type/class  
is.numeric(x), is.character(), is.factor() # is data type  
as.numeric(x), as.character(), as.factor() # change data type
```

Summary Stats:

```
summary() # summary statistics  
mean(), median(), sd(), sum(), log(), max(),  
min(), quantile(), var()
```

Read in Data:

```
read.xlsx("name.xlsx") # library(readxl)  
read.delim("name.txt", sep ="\t")  
read.csv("name.csv", sep=";")
```

```
view() # view data as table  
df$col1 # extract column from dataframe  
nrow(df), ncol(df) # number of rows/columns
```

Conditions:

```
a == b    a != b    a > b    a < b  
a >= b   a <= b   is.na() is.null()
```

Plots:

```
plot(x)  
plot(x, y) or plot(col1, col2, df) # scatter  
hist(x) # histogram
```

GETTING STARTED

OVERVIEW

DATA TYPES & CONDITIONS

STATISTICS & BASE PLOTS

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/>



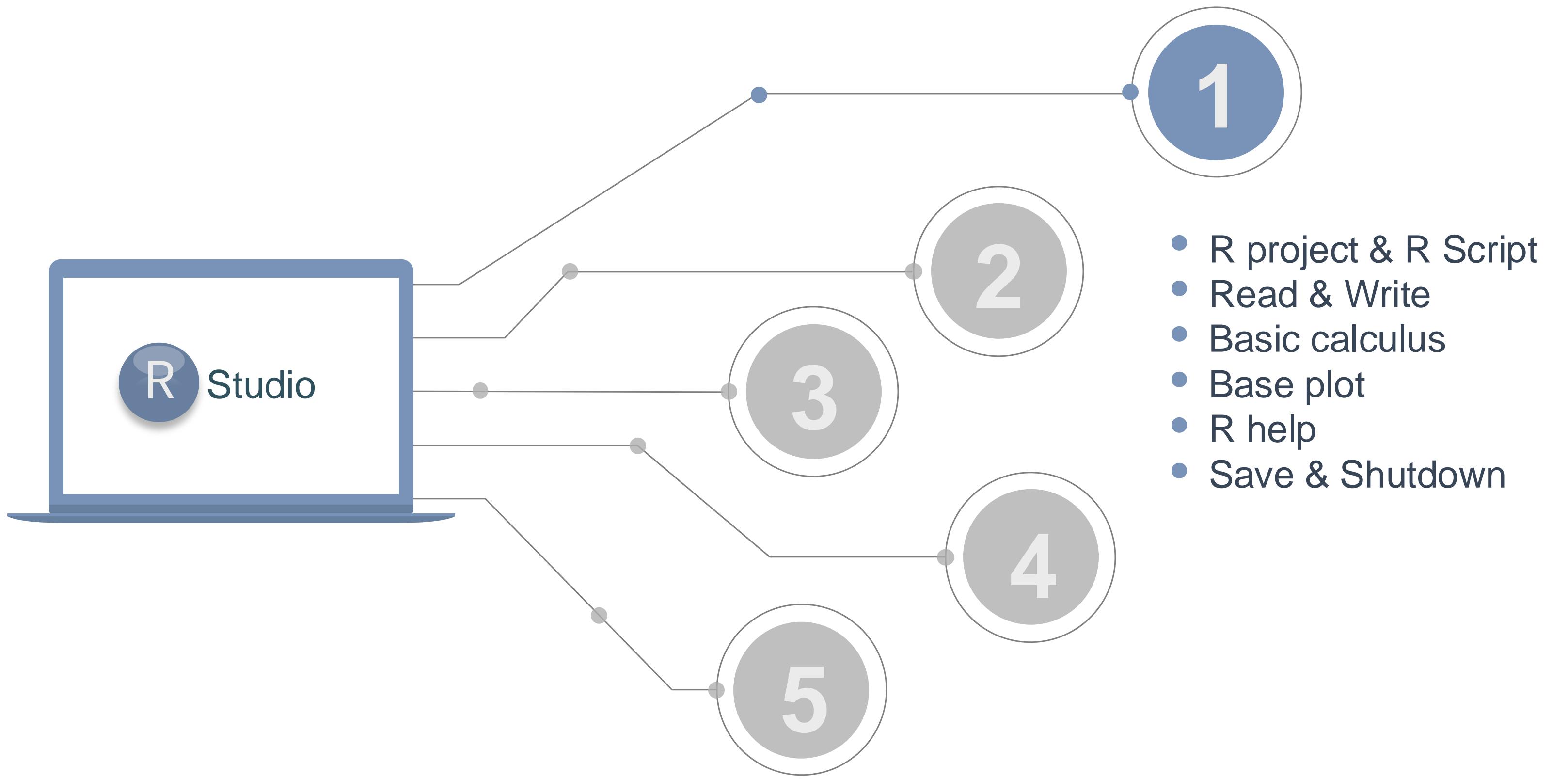
OTHER RESOURCES

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

https://blog.revolutionanalytic_s.com/

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





— FUNDAMENTALS
EXERCISE 1

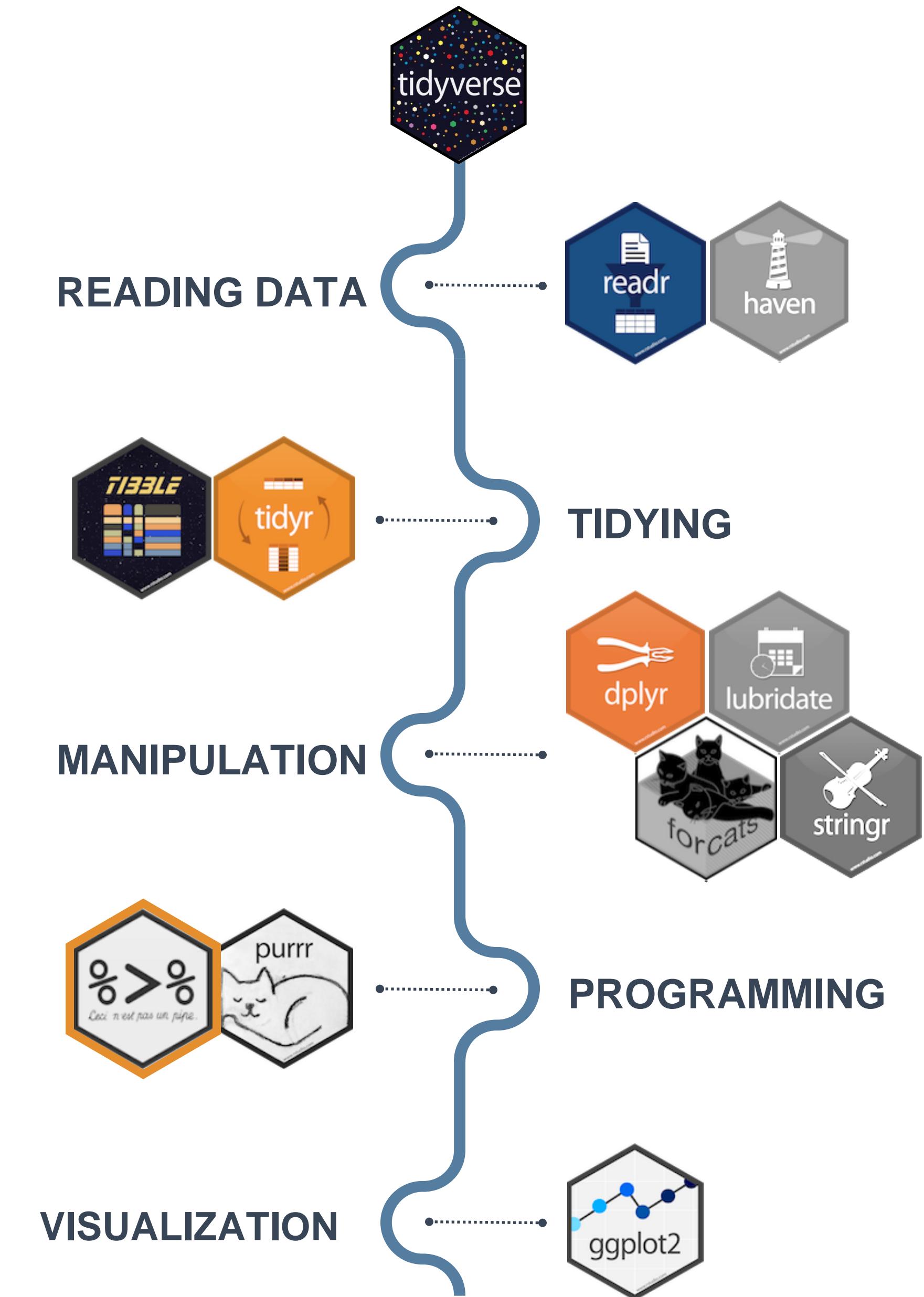
TIDYVERSE

<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



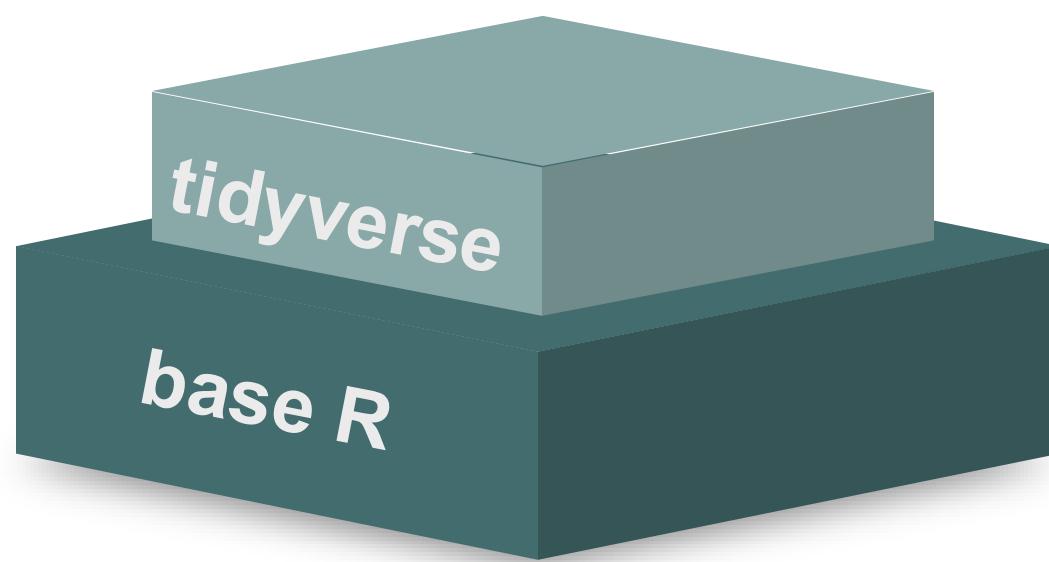
CECI N'EST PAS UNE PIPE

%>%

- 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”
- Can be computationally slow

base R

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

tidyverse

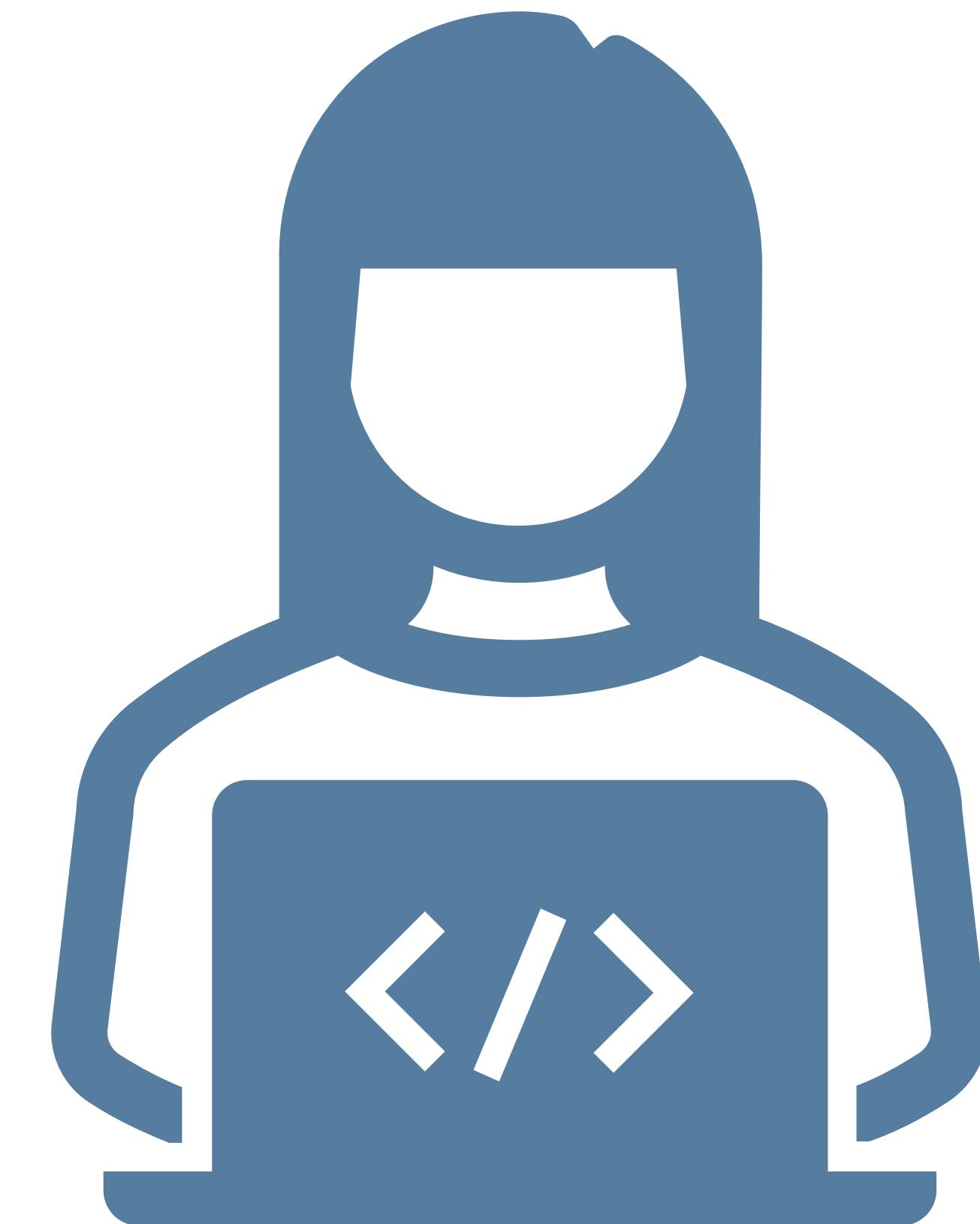
```
# no brain acrobatics  
x %>% f(y) %>% g(z)
```



pipe symbol

— FROM EXCEL TO R

LIVE CODING 2 – TIDYVERSE



TIDYVERSE CHEAT SHEET

readr, tidyverse, dplyr, ...

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: ...")

Workflow

Tidyverse workflow

```
df %>%  
  select(col1)  
  
df %>%  
  filter(col1 < x)  
  
df %>%  
  summarize(n_1 = n(),  
            avg_1 = mean(col1),  
            sd_1 = sd(col1))
```

Select column

```
df$col1
```

Data Manipulation (*dplyr*)

Summary

```
summarize()  
count()
```

Group

```
group_by()
```

Functions will manipulate each group separately and combine results.

Extract and sort observations (rows)

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

Manipulate variables (columns)

```
select() # subset rows by condition  
mutate(new_name = f(column))  
mutate(new_name = ifelse(col1 < x, "Yes", "No"))  
mutate(new_name = col1 + col2)
```

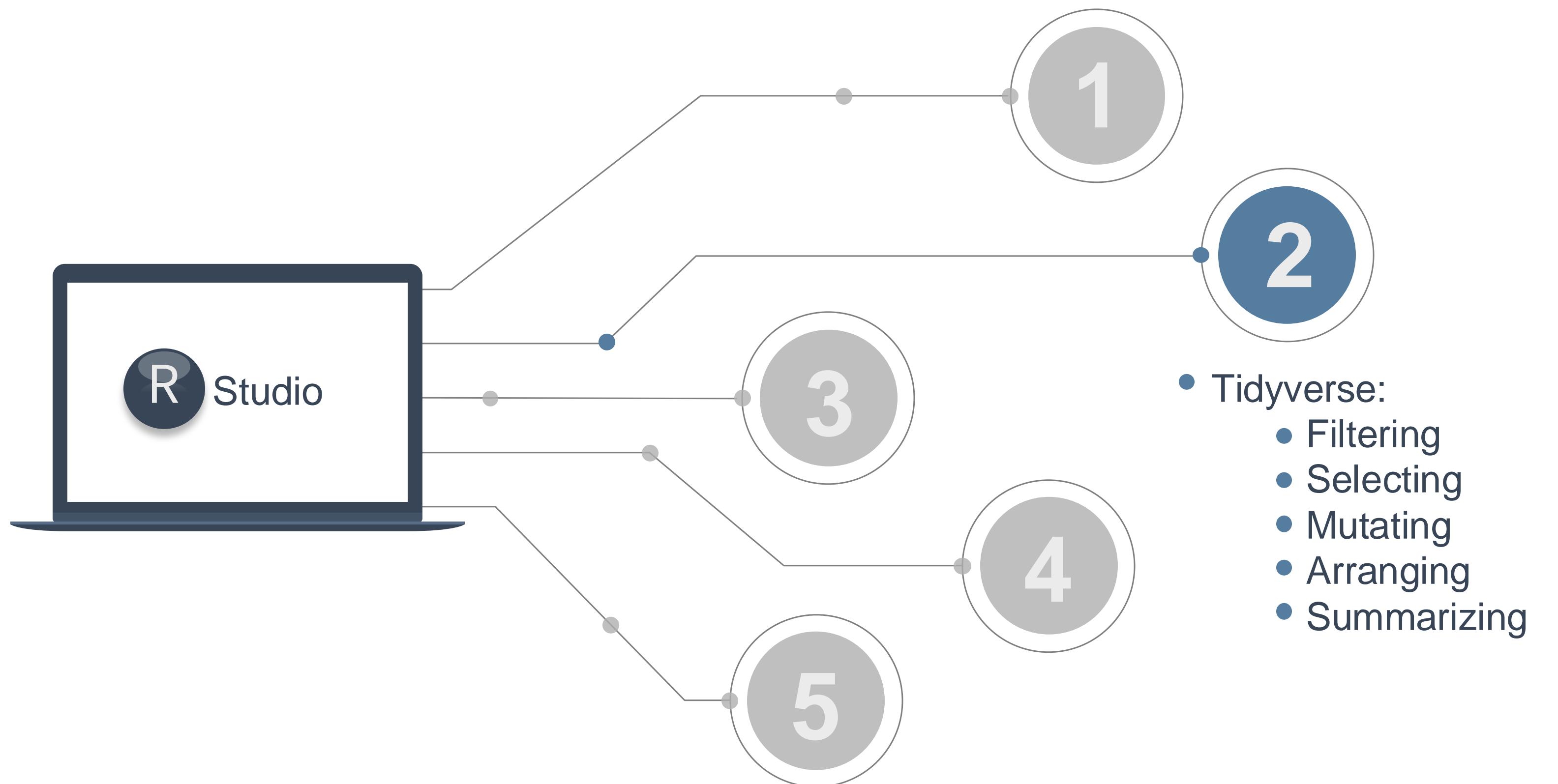
HELP

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

Much more info and detailed cheat sheets:

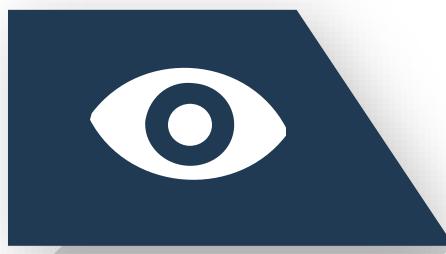
<https://brianward1428.medium.com/introduction-to-tidyverse-7b3dbf2337d5>

It also helps to google "tidyverse + whatever you want to do"

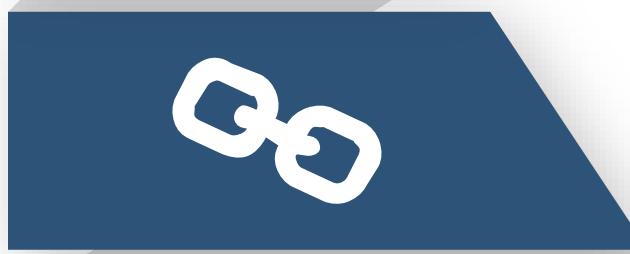
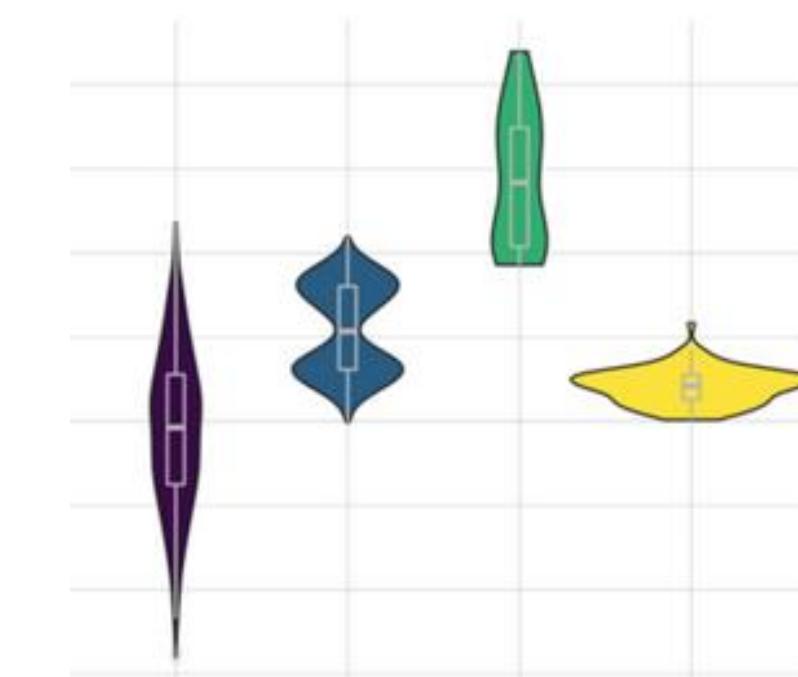


TIDYVERSE EXERCISE 2

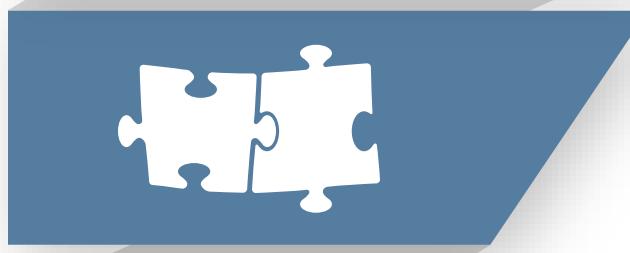
GGPLOT2 - EASY GRAPHICS



Aesthetically pleasing graphics.



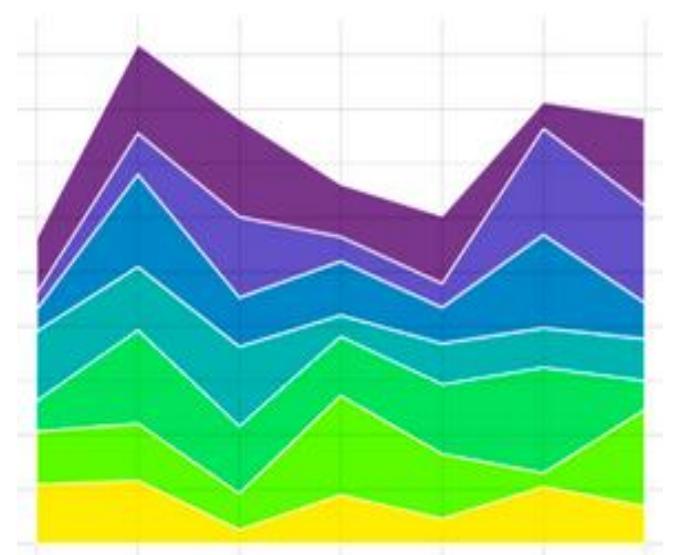
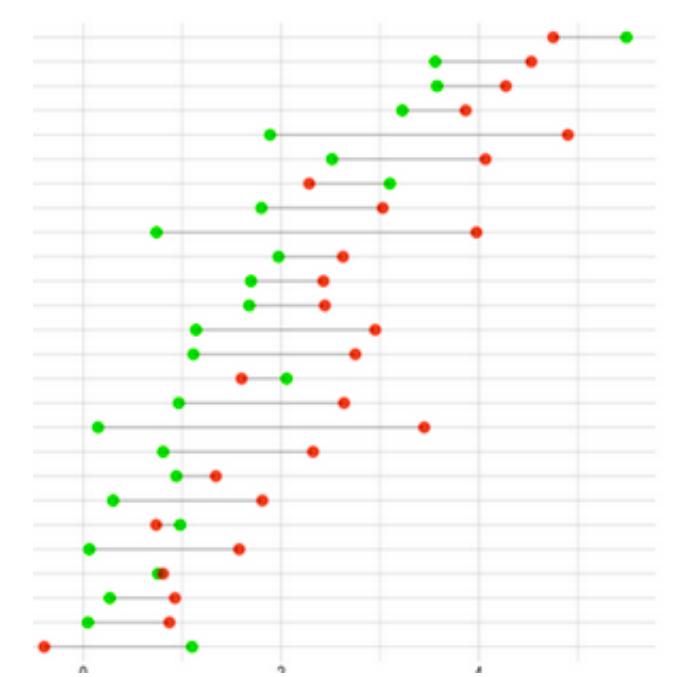
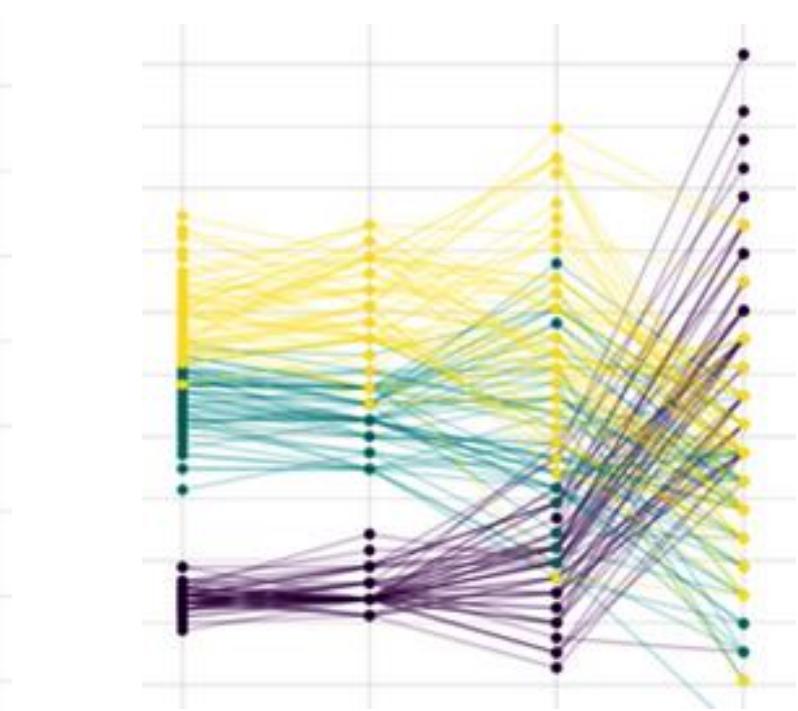
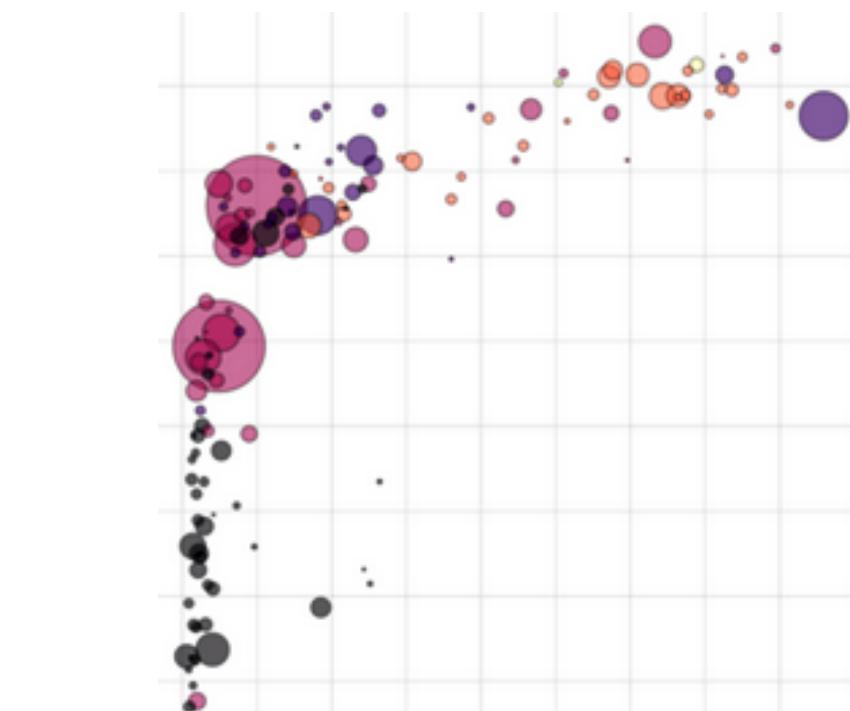
Well-defined “additive” (+) structure.



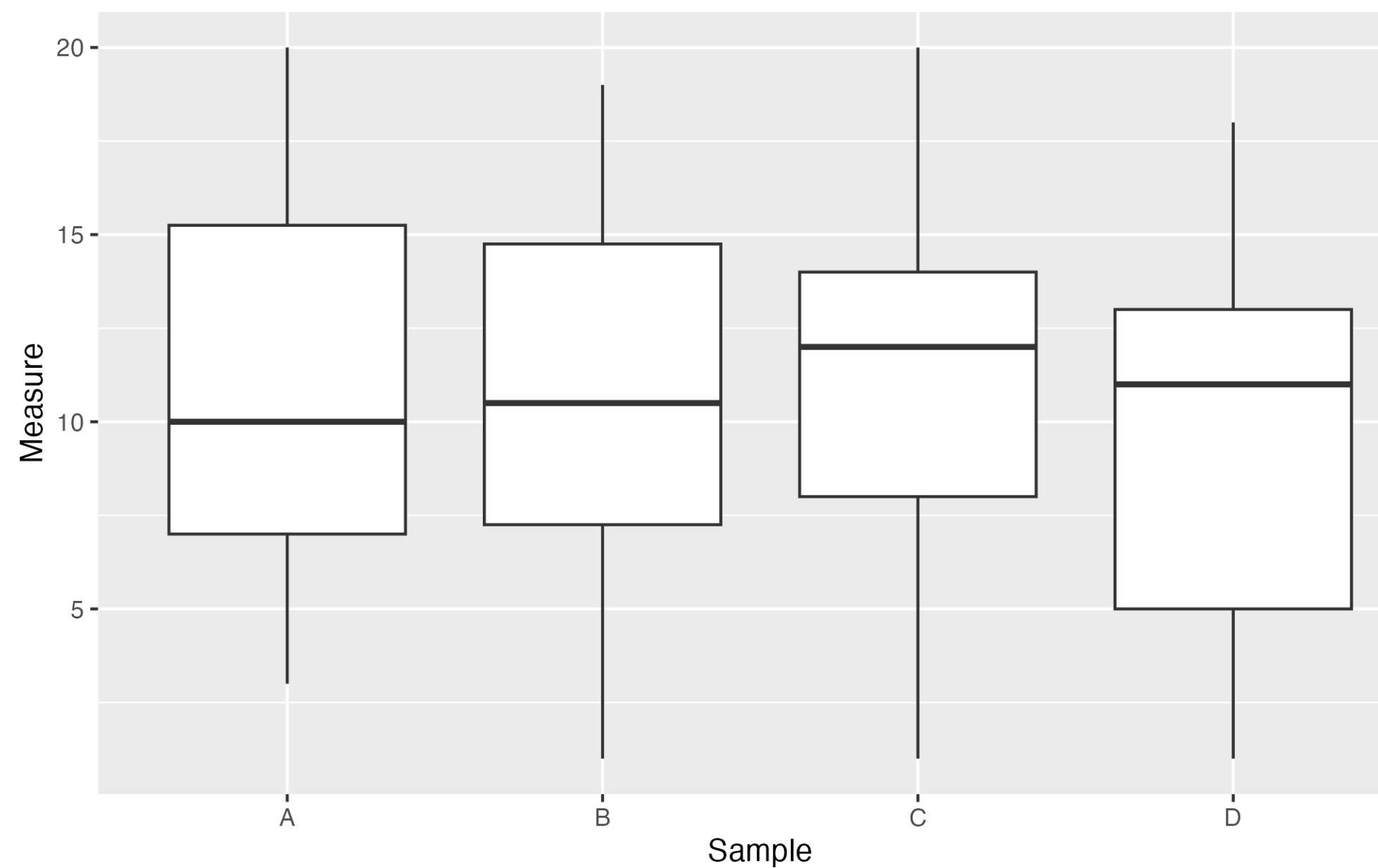
Integrates perfectly with tidy data.



Great documentation & community

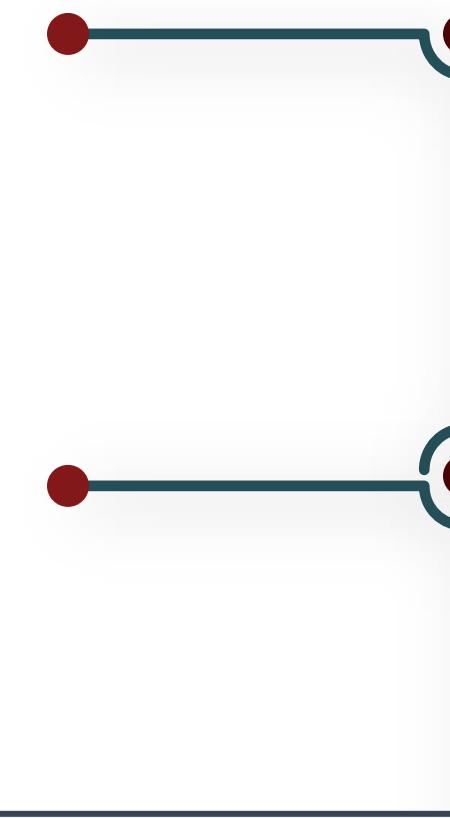


GGPLOT2 ADDITIVE STRUCTURE



DATASET, SAMPLES & OBSERVATIONS

DEFINE PLOT TYPE



```
ggplot(df,  
       aes(x = Sample, y = Measure))  
  
ggplot(df,  
       aes(x = Sample, y = Measure)) +  
  geom_boxplot()
```

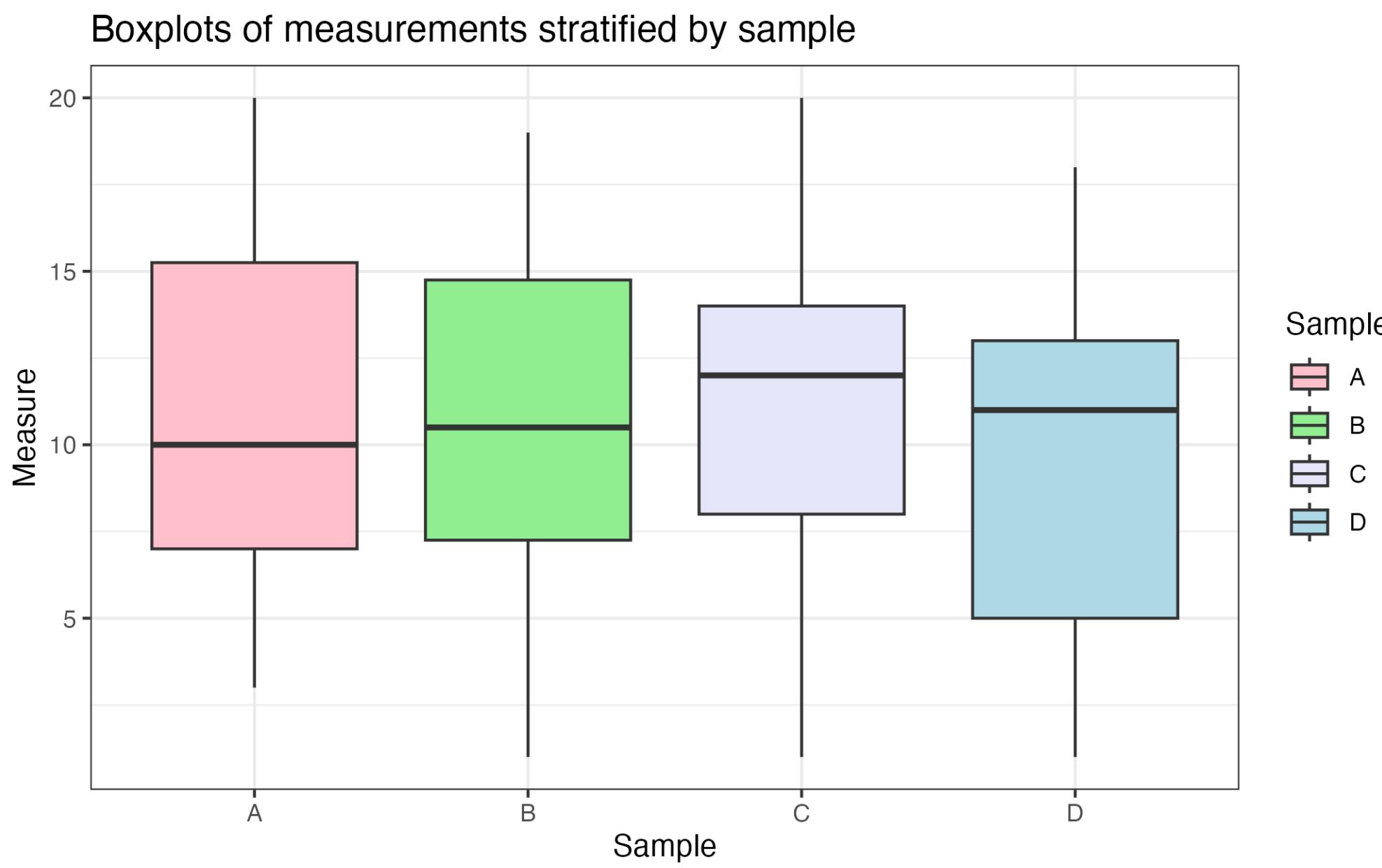
head(df)

Description: df [6 x 2]

| | Sample | Measure |
|---|--------|---------|
| 1 | C | 20 |
| 2 | C | 14 |
| 3 | C | 3 |
| 4 | B | 8 |
| 5 | C | 16 |
| 6 | B | 12 |

6 rows

GGPLOT2 ADDITIVE STRUCTURE



Dataset, Samples & Observations



```
ggplot(df,
```

```
  aes(x = Sample, y = Measure))
```

Define Plot Type



```
ggplot(df,
```

```
  aes(x = Sample, y = Measure)) +  
  geom_boxplot()
```

Color by Group



```
ggplot(df,
```

```
  aes(x = Sample, y = Measure,  
      fill = Sample)) +  
  geom_boxplot()
```

Title and Legend



```
... +  
  labs(title = "Boxplots of  
measurements stratified by sample")
```

Custom Colors



```
... +  
  scale_fill_manual(values =  
c("pink", "lightgreen", "lavender",  
"lightblue"))
```

Background

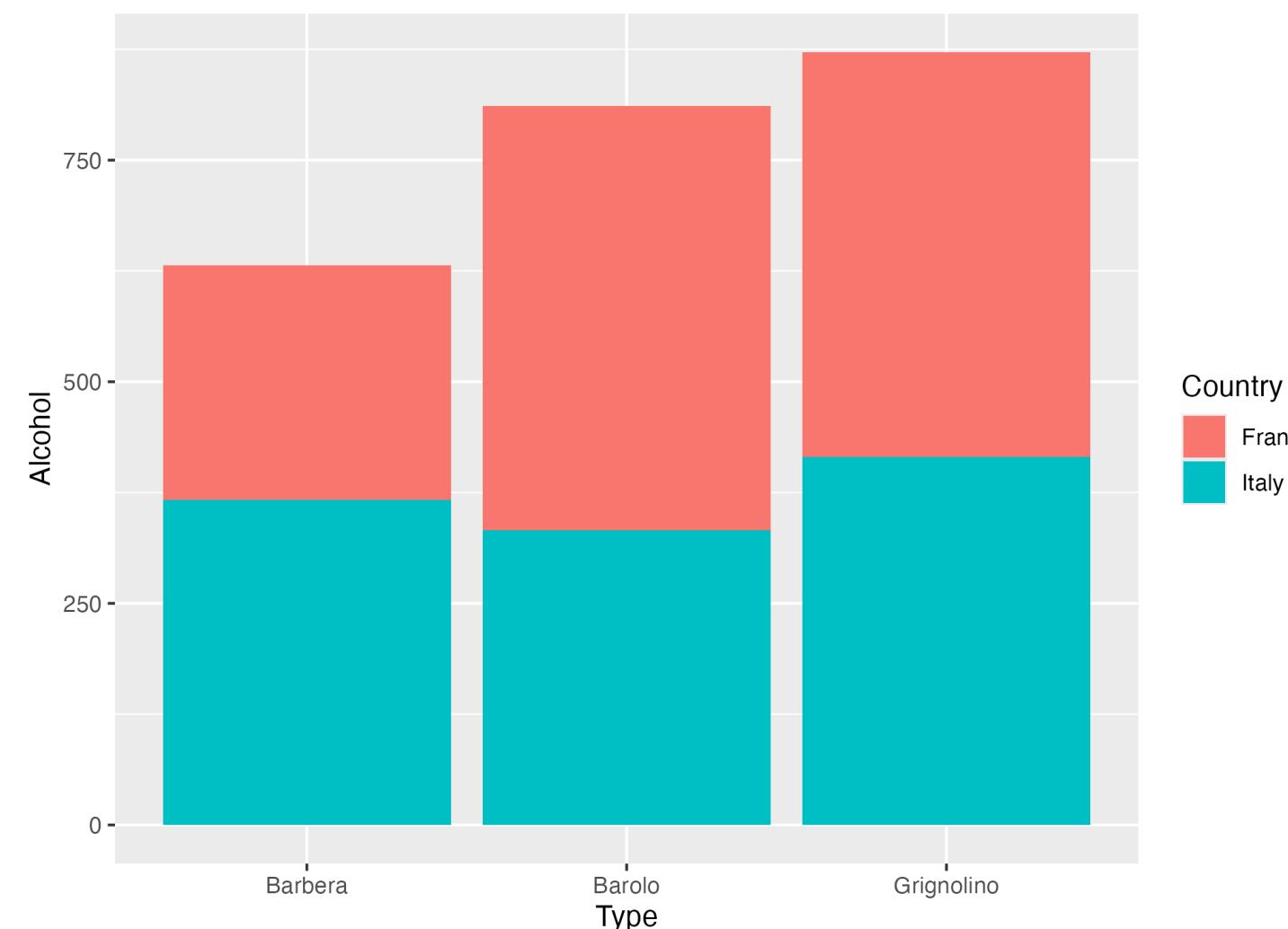


```
... +  
  theme_bw()
```

GGPLOT BASIC STRUCTURE

Geoms inherit the parameters from the ggplot they are added to:

```
ggplot(wine,
       aes(x = Type,
           y = Alcohol,
           fill = Country)) +
  geom_col()
```



head(wine)

Description: df [6 × 5]

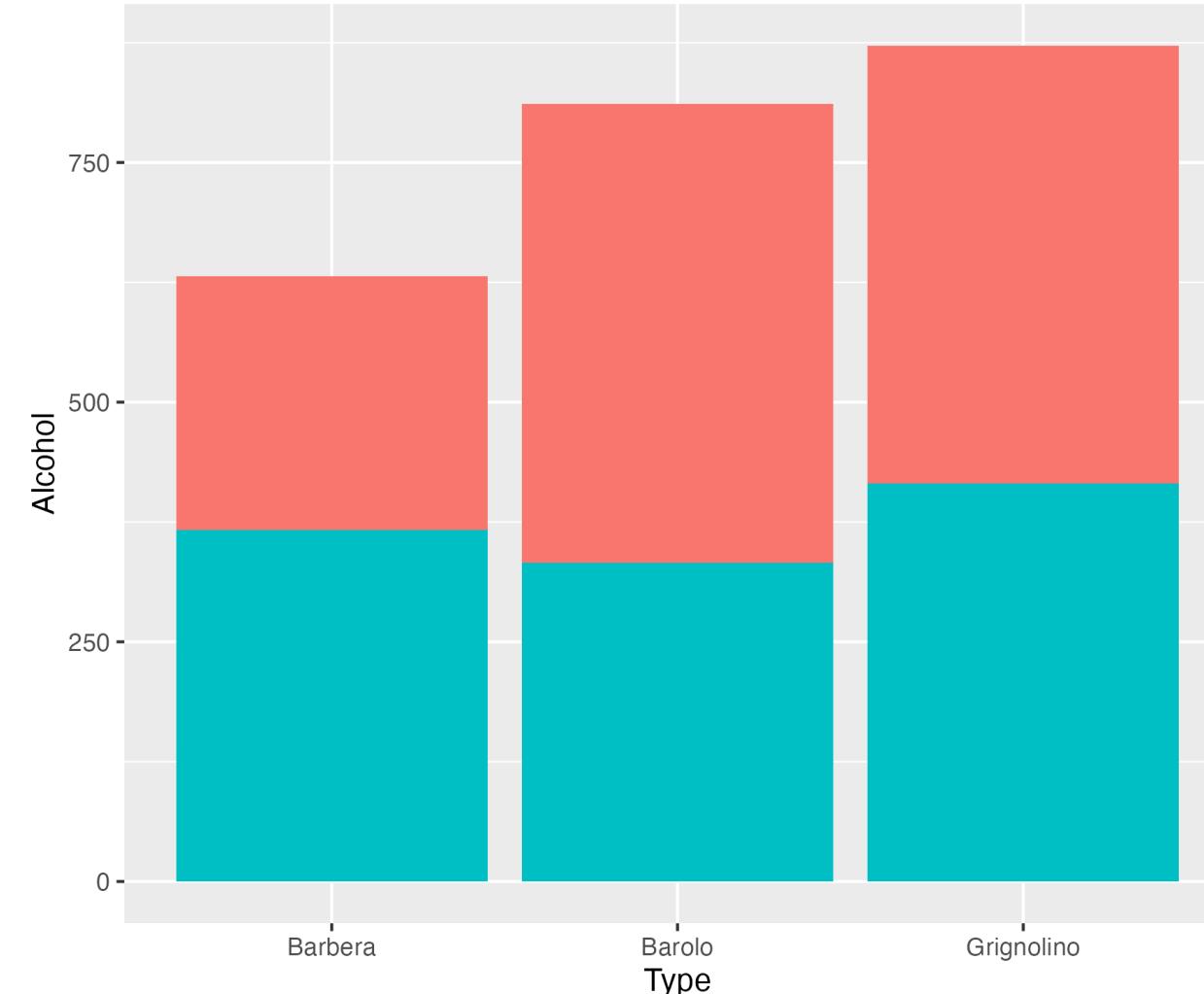
| | Type | Alcohol | Country | Magnesium | Phenols |
|---|--------|---------|---------|-----------|---------|
| | <fctr> | <dbl> | <chr> | <int> | <dbl> |
| 1 | Barolo | 14.23 | France | 127 | 2.80 |
| 2 | Barolo | 13.20 | France | 100 | 2.65 |
| 3 | Barolo | 13.16 | France | 101 | 2.80 |
| 4 | Barolo | 14.37 | Italy | 113 | 3.85 |
| 5 | Barolo | 13.24 | France | 118 | 2.80 |
| 6 | Barolo | 14.20 | Italy | 112 | 3.27 |

6 rows

GGPLOT BASIC STRUCTURE

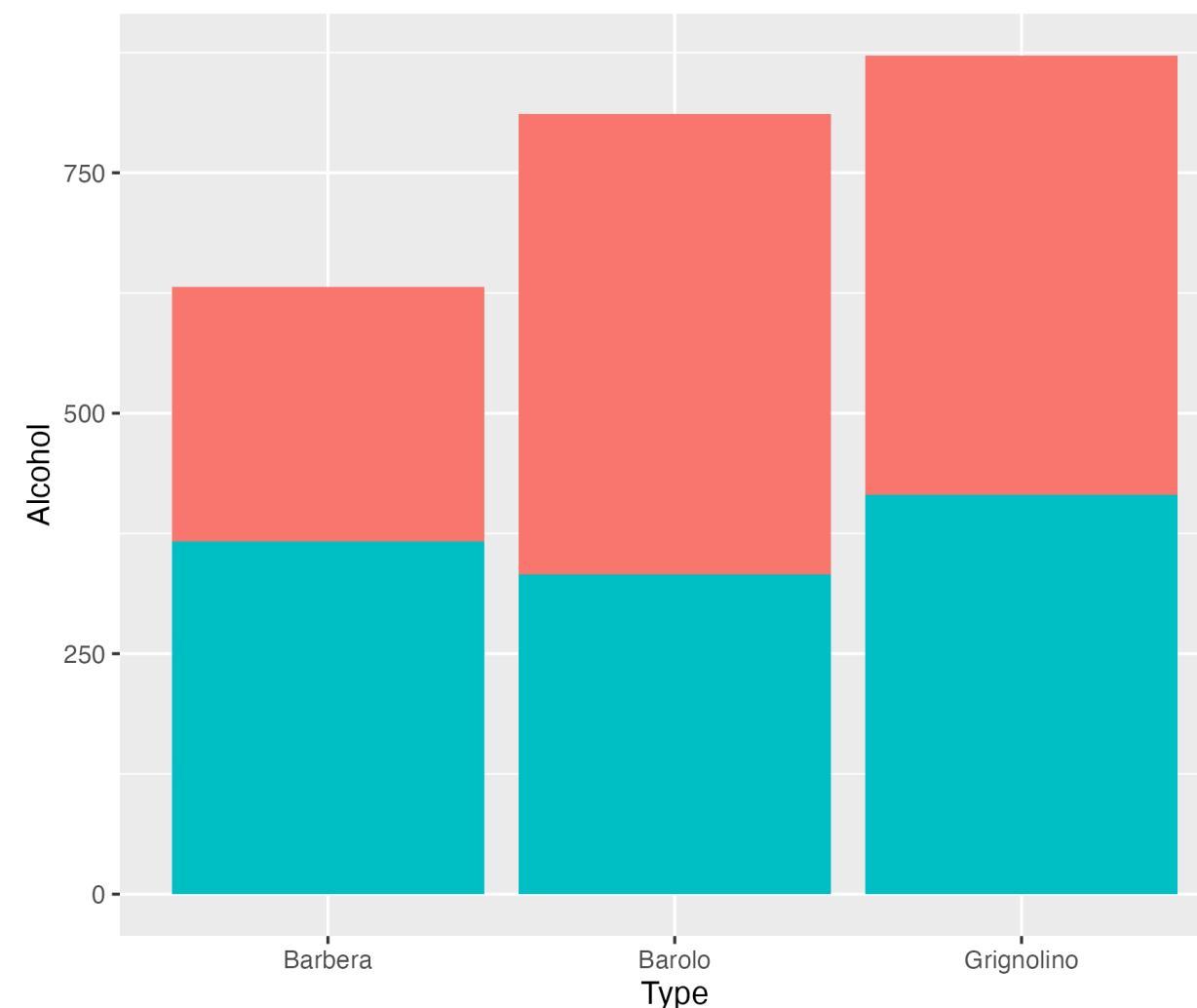
Geoms inherit the parameters from the ggplot they are added to:

```
ggplot(wine,
       aes(x = Type,
           y = Alcohol,
           fill = Country)) +
  geom_col()
```



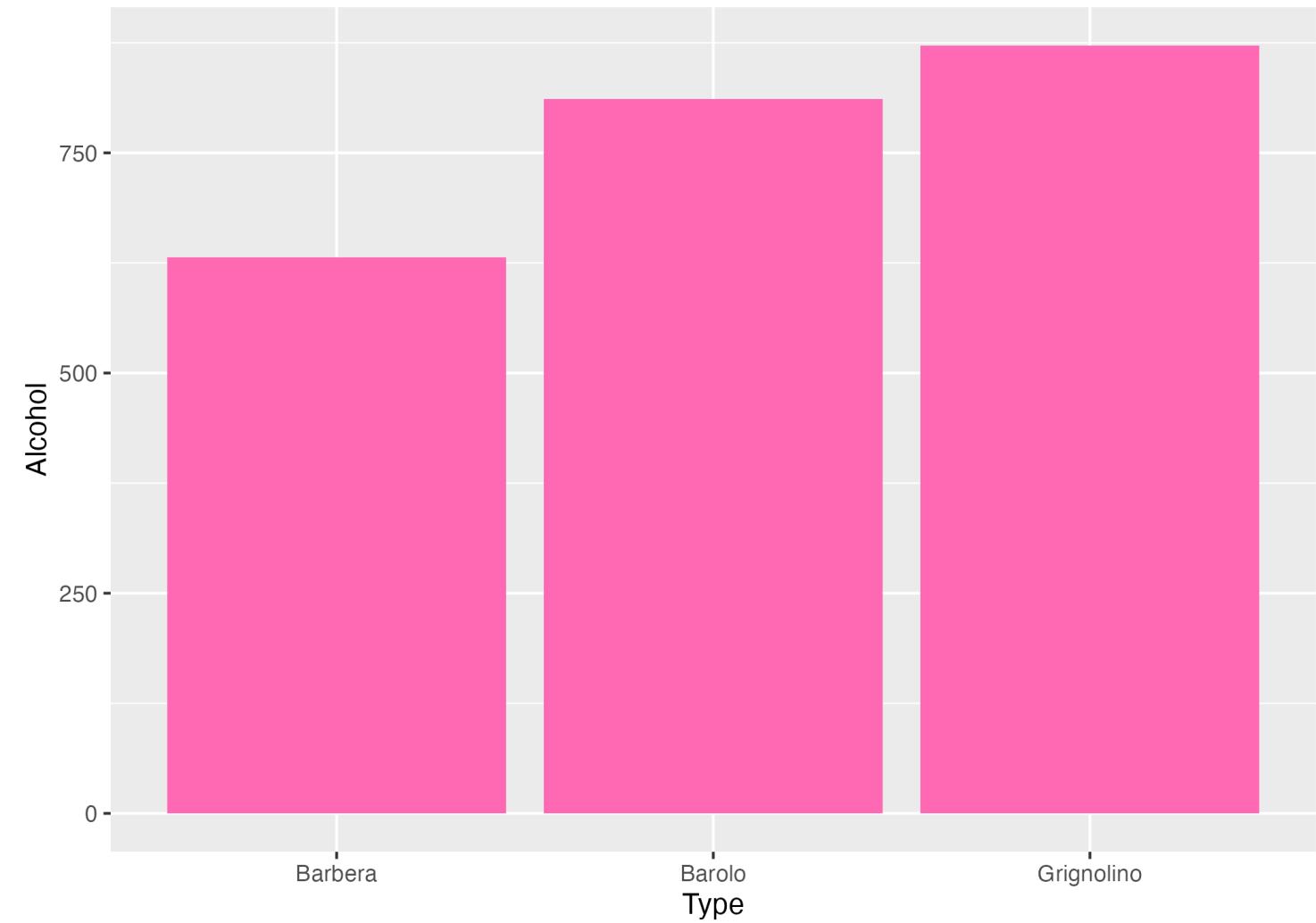
Alternatively, you can specify additional parameters directly in the geom:

```
ggplot(wine,
       aes(x = Type,
           y = Alcohol)) +
  geom_col(aes(fill = Country))
```



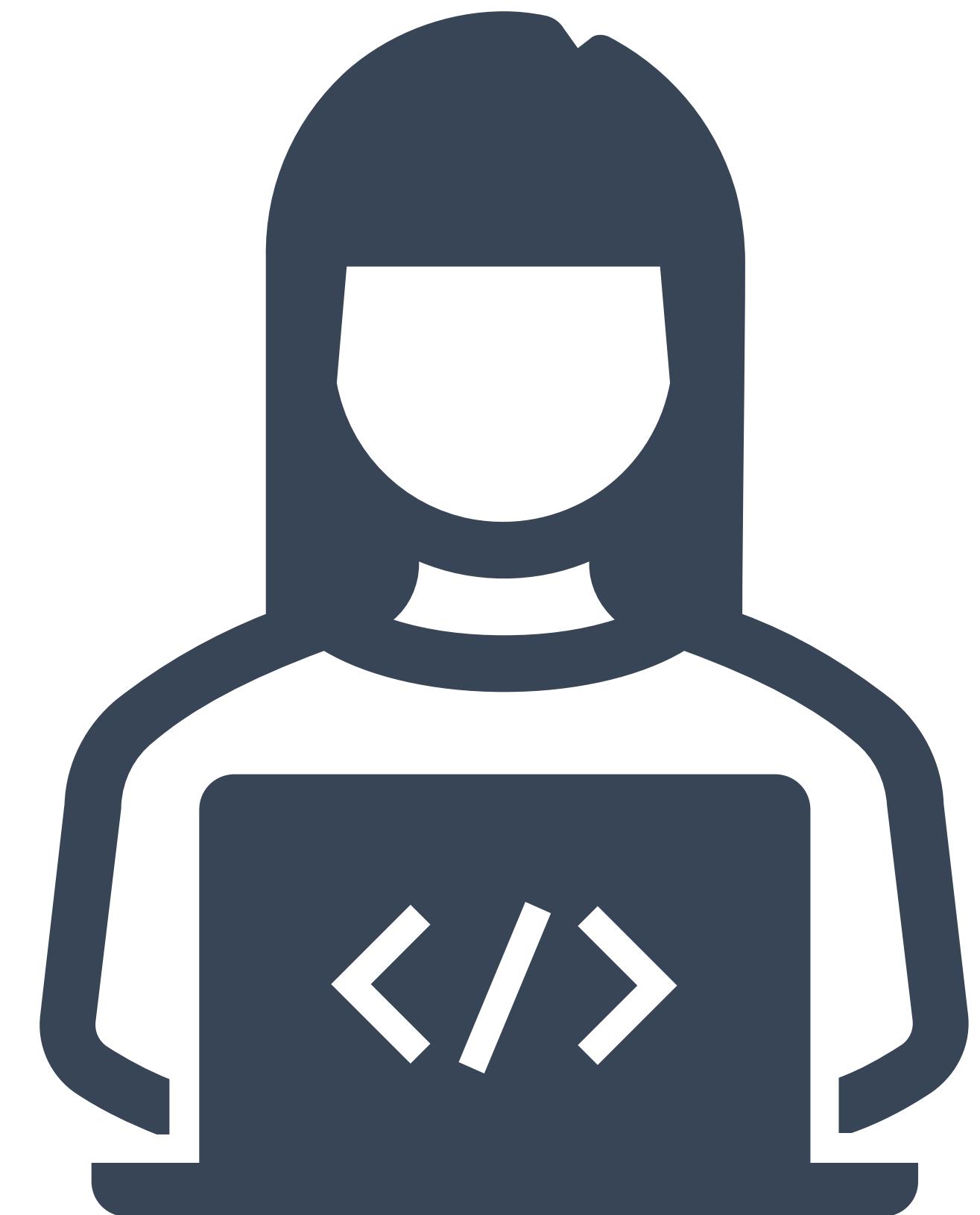
Things outside the aes are applied to everything!

```
ggplot(wine,
       aes(x = Type,
           y = Alcohol)) +
  geom_col(fill = "hotpink")
```



— FROM EXCEL TO R

LIVE CODING 3 – GGPLOT 2



GGPLOT CHEAT SHEET

Define Plot:

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

Add Plot Type:

```
... + geom_point() # scatter plot  
... + geom_line()  
... + geom_boxplot()
```

```
... + geom_col()  
... + geom_density()  
... + geom_histogram()
```

One Color:

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

Color Fill by Group:

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

Custom Colors:

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

More Colors:

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

Labels:

```
... + labs(title = "Title",  
          x = "X label",  
          y = "Y label")
```

Theme:

```
... + theme_bw()  
... + theme_minimal()  
... + theme_dark()  
... + theme_classic()
```

Text:

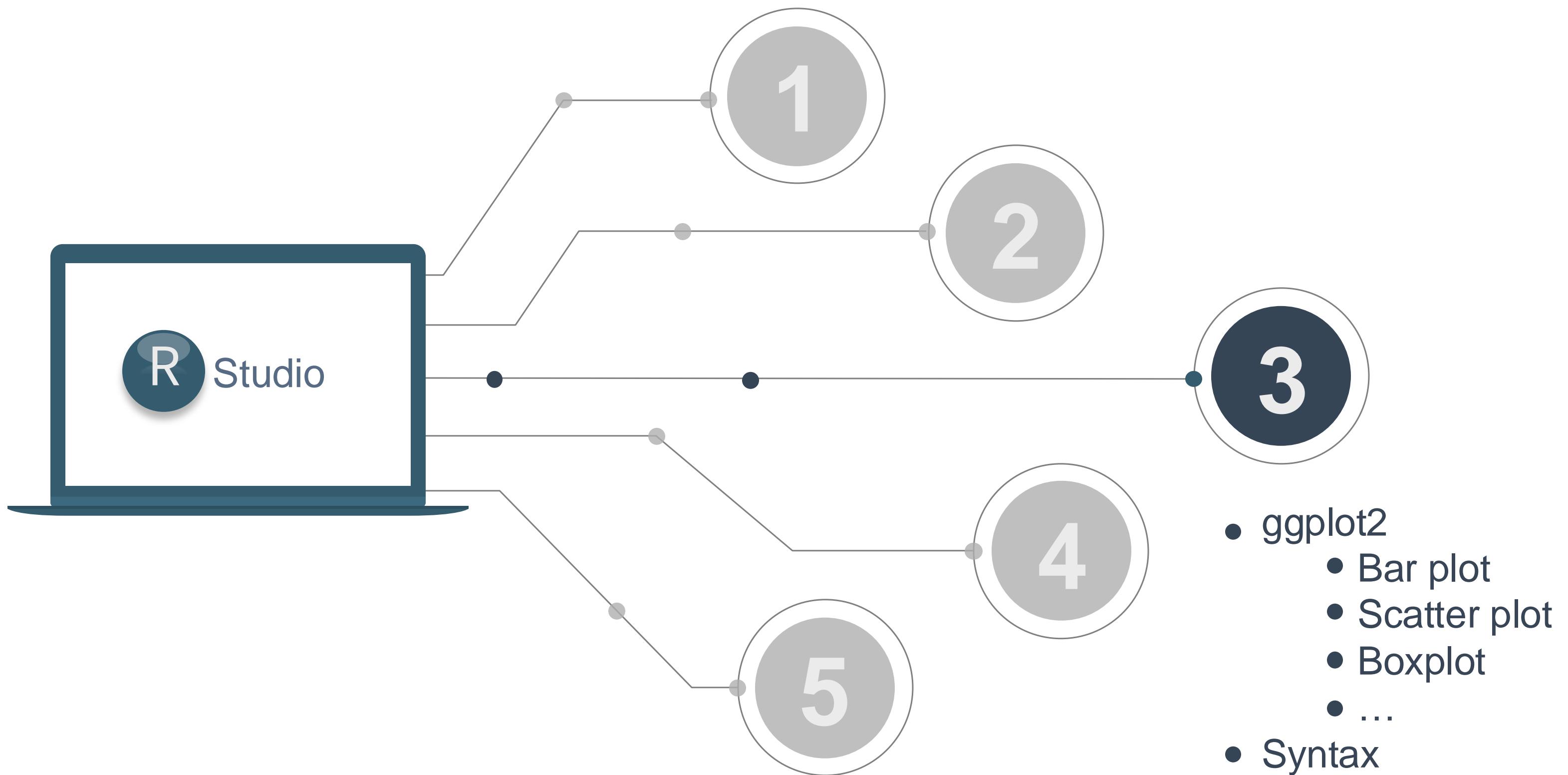
```
... + theme(legend.position = *)  
* = "none", "top", "bottom", "left", "right"
```

GET
STARTED

COLORS

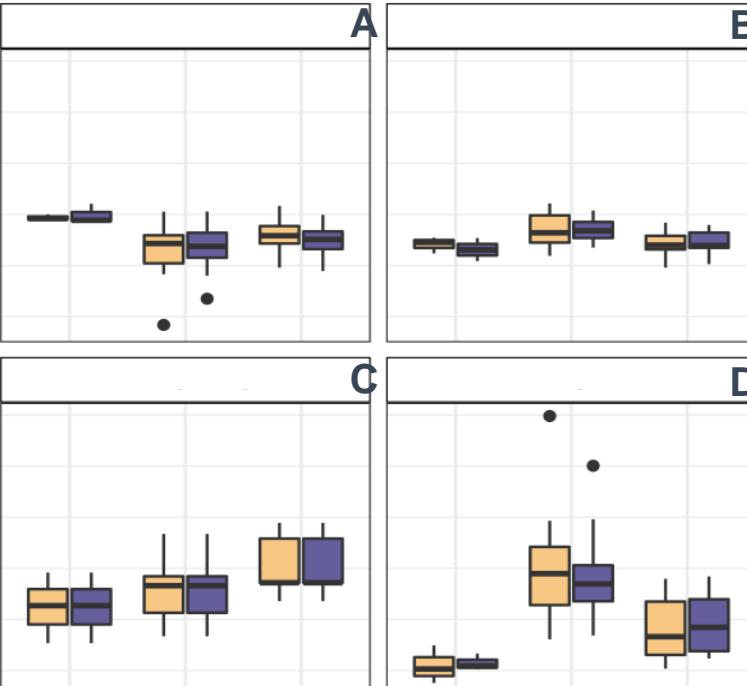
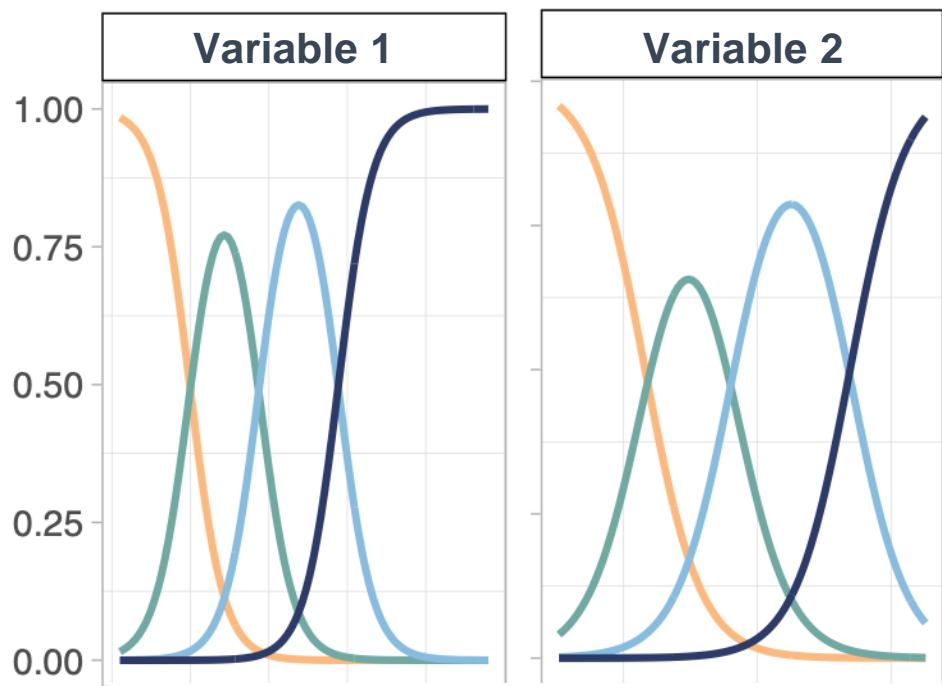
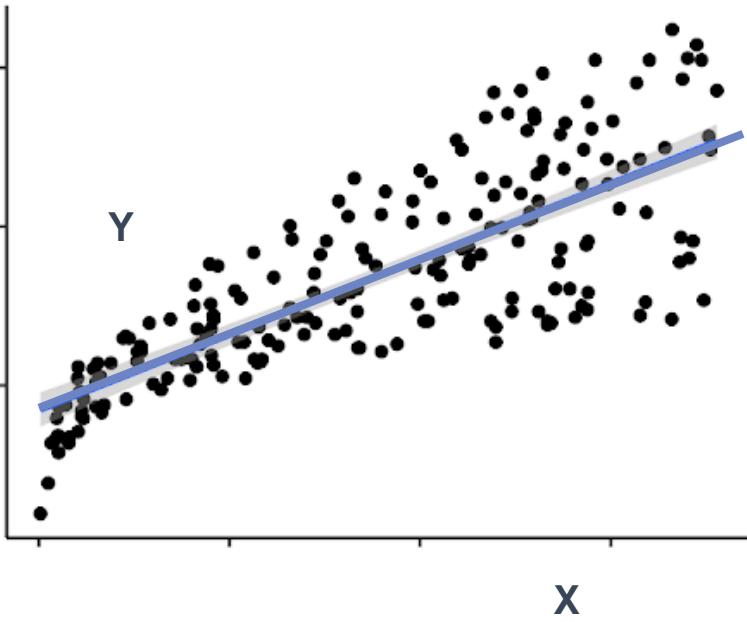
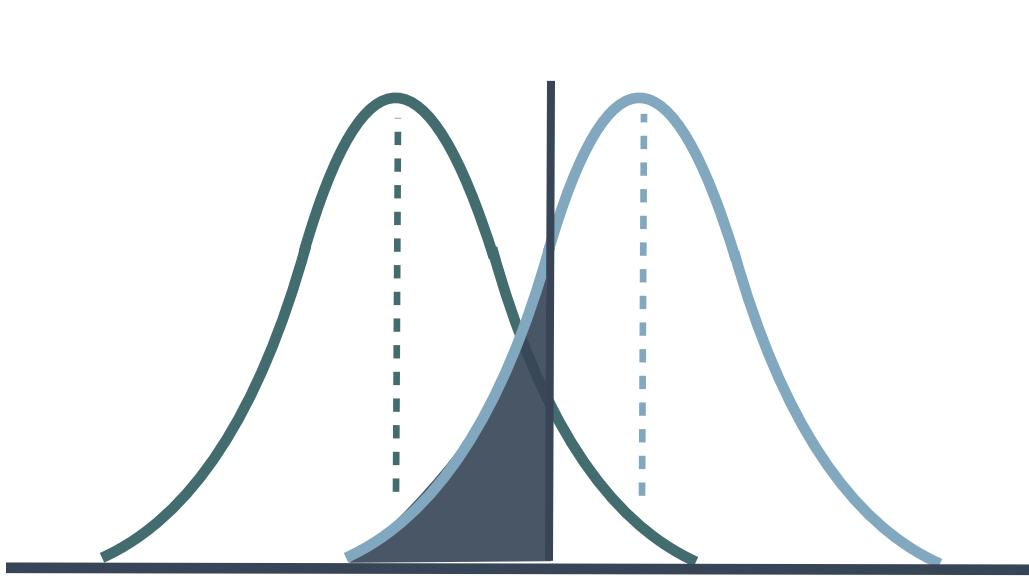
COLOR SCALES
& THEMES

TEXT



— GG PLOT2 EXERCISE 3

R - A STATISTICAL SCRIPTING LANGUAGE



MODEL FUNCTIONS

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

EMMEANS PACKAGE

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

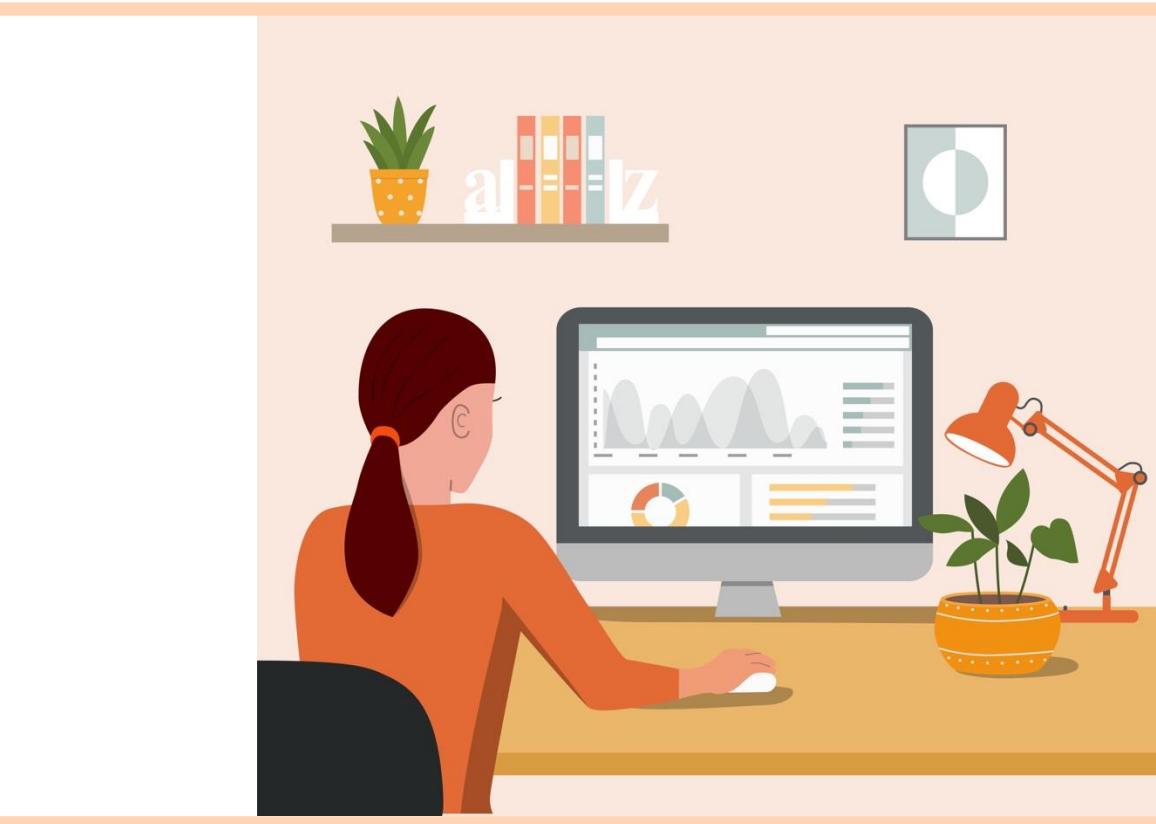
APPLY TO MODEL

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

MORE FUNCTIONS

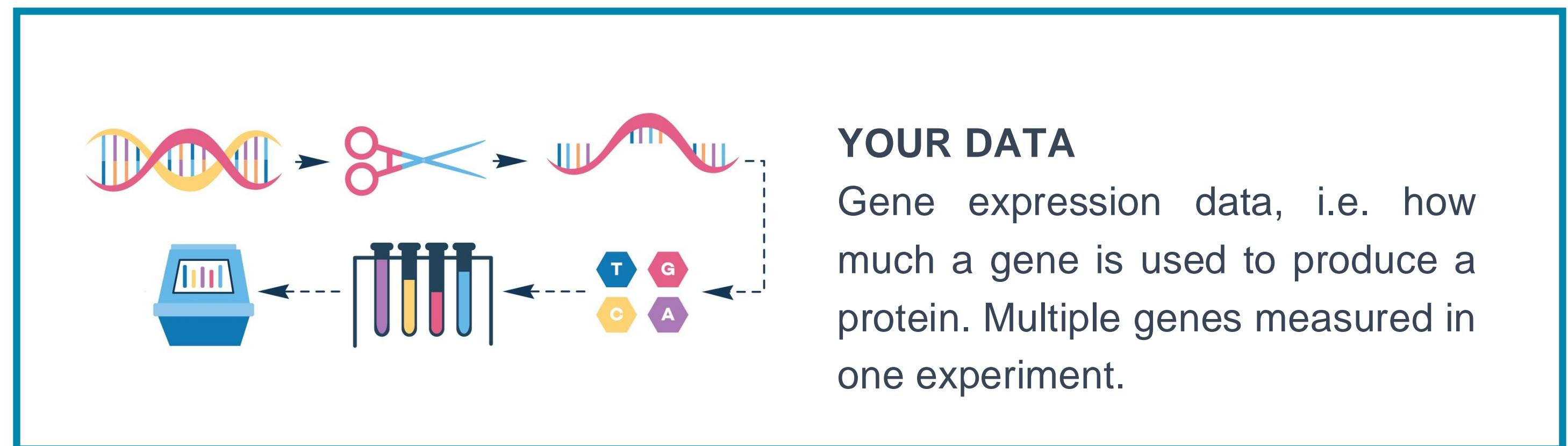
`t.test()`, `cor()`,
`cor.test()`, `aov()`,
`quantile()`,
`p.adjust()`,
`rank()`, ...

Let's use R in a statistical analysis



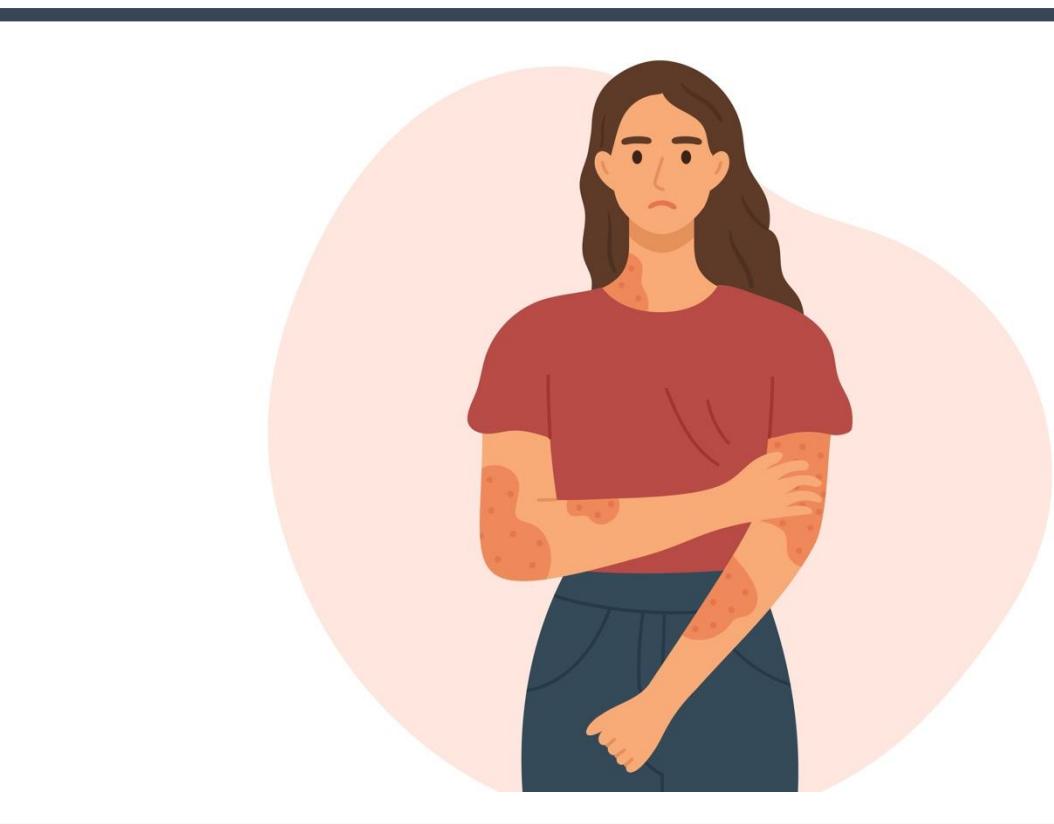
YOU

The researcher with R skills!



YOUR DATA

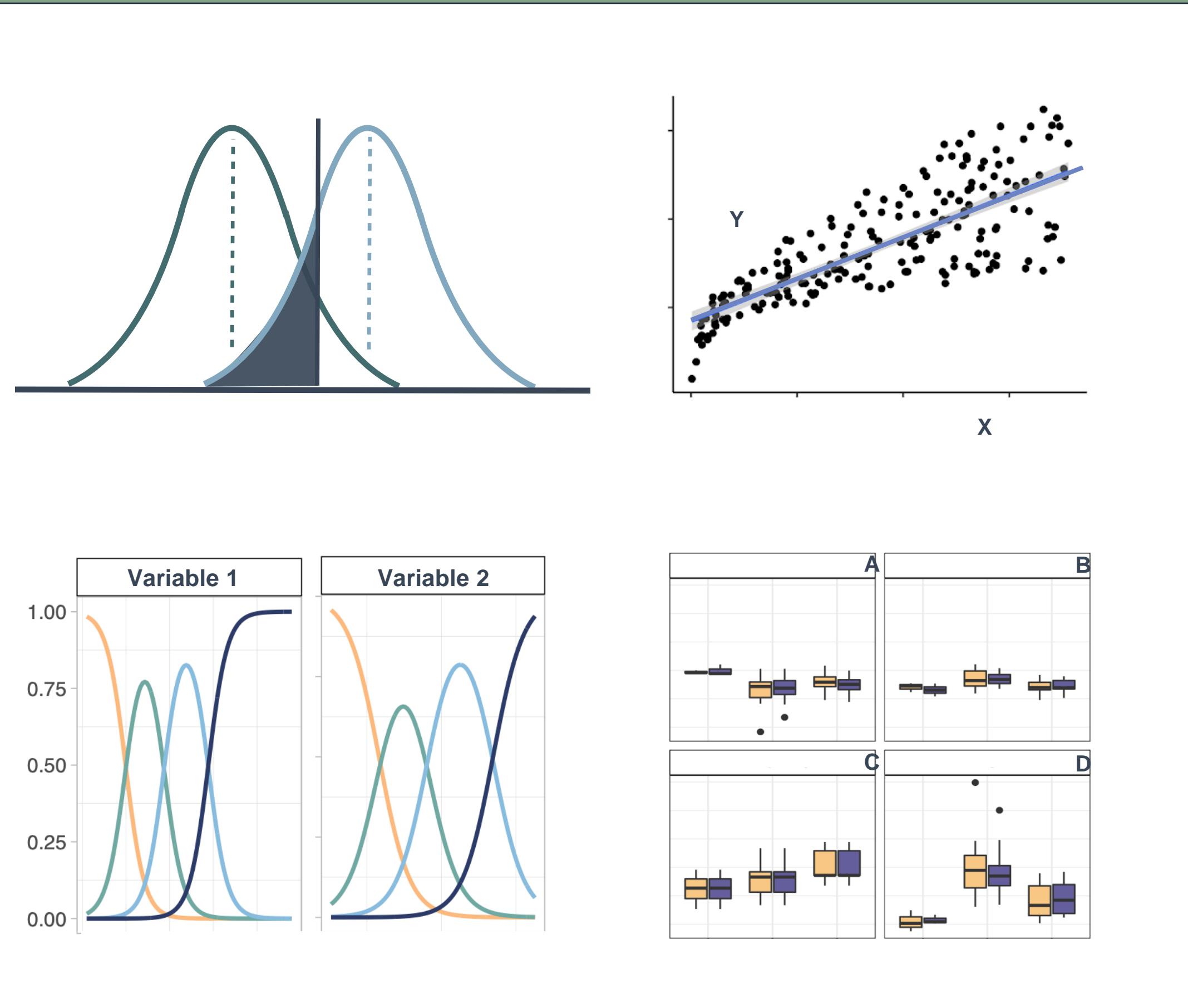
Gene expression data, i.e. how much a gene is used to produce a protein. Multiple genes measured in one experiment.



YOUR DISEASE OF INTEREST

Psoriasis, an immune-mediated skin disorder.

R - A STATISTICAL SCRIPTING LANGUAGE

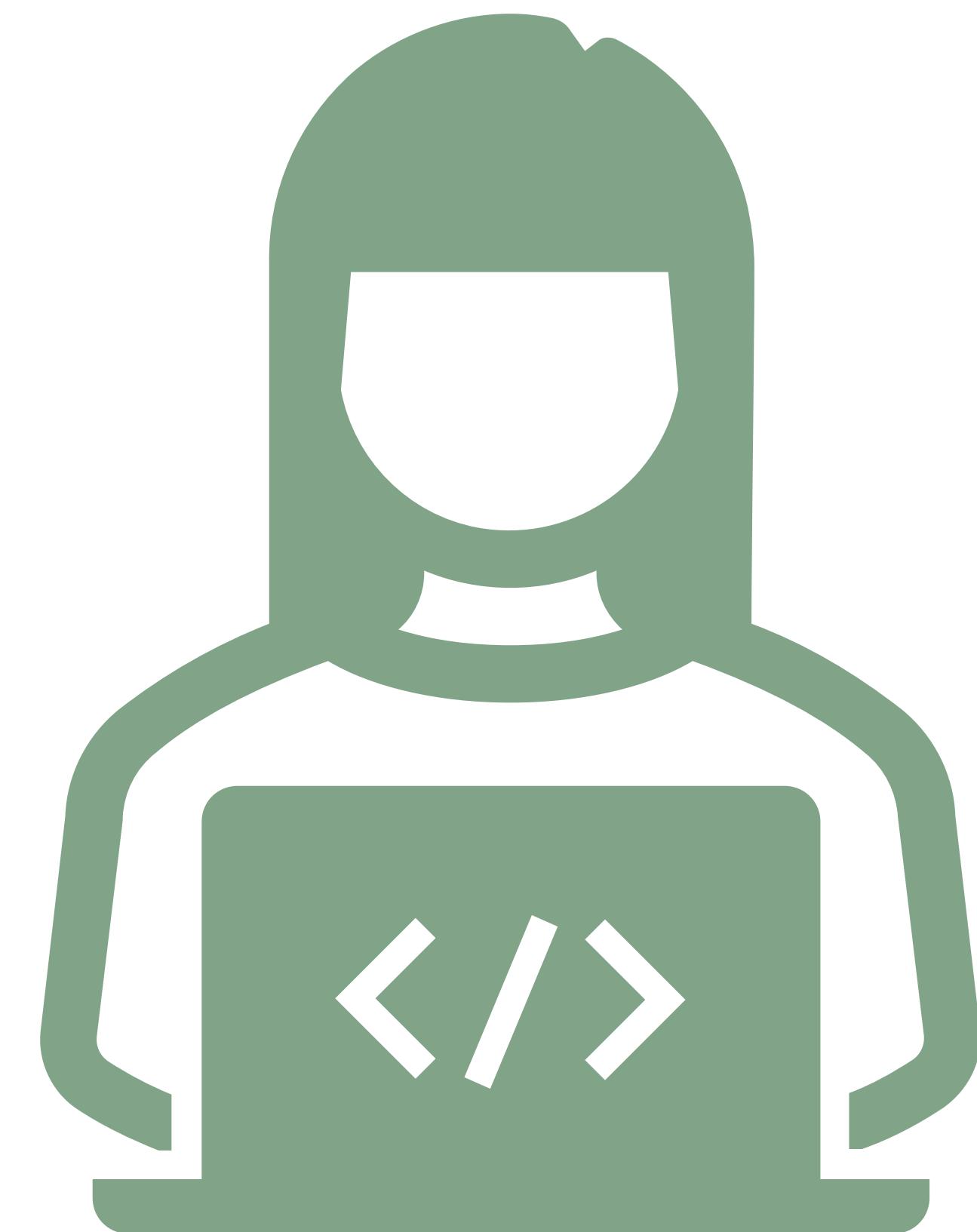


During this session:

- *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

— FROM EXCEL TO R

LIVE CODING 4 – APPLIED STATS



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)
```

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))
```

Check Type:

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

Change Type:

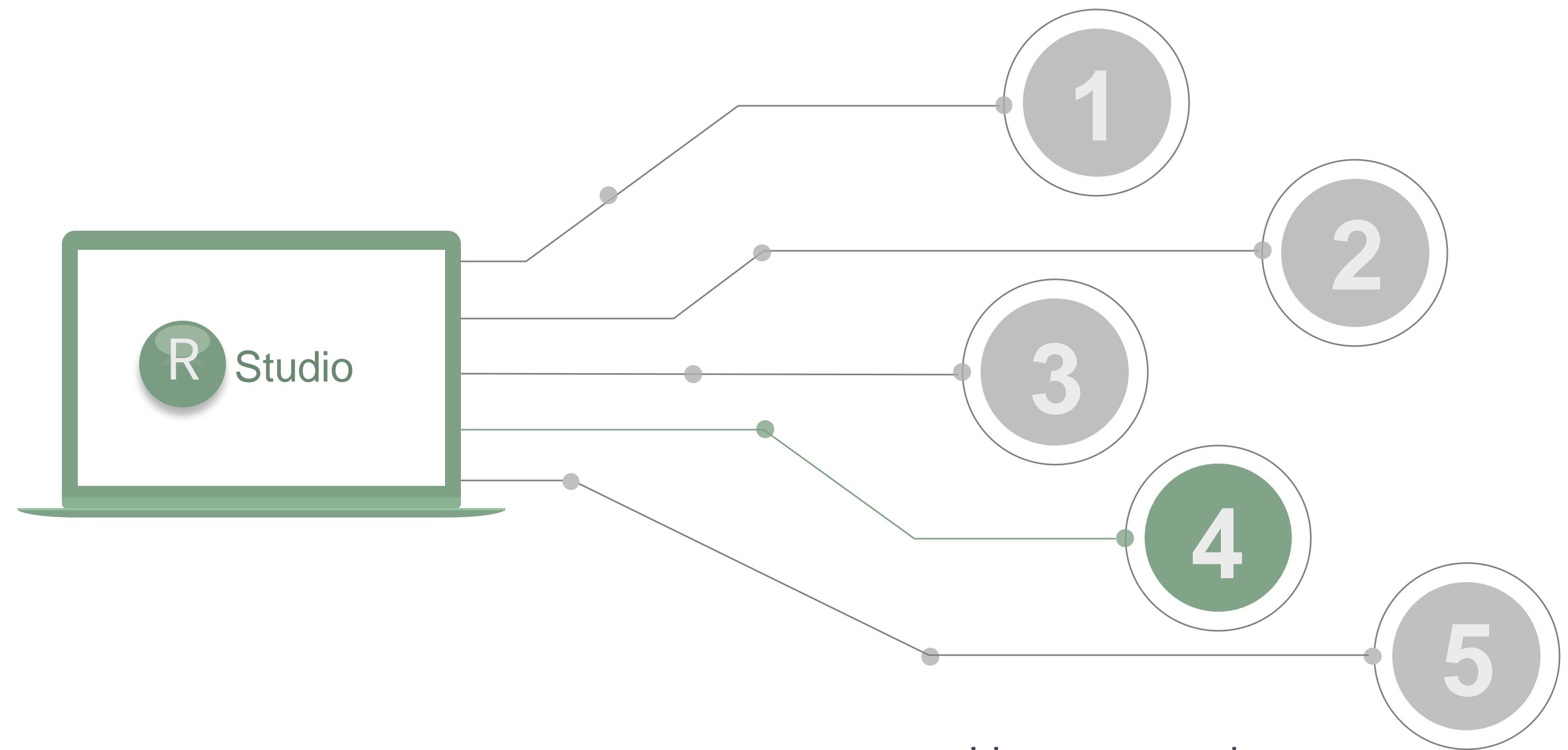
```
my.data <- mutate(my.data, x = factor(x))  
my.data$z <- as.numeric(my.data$z)
```

GET
STARTED

REGRESSION
MODELS

TESTS/COMPARISONS

VARIABLES

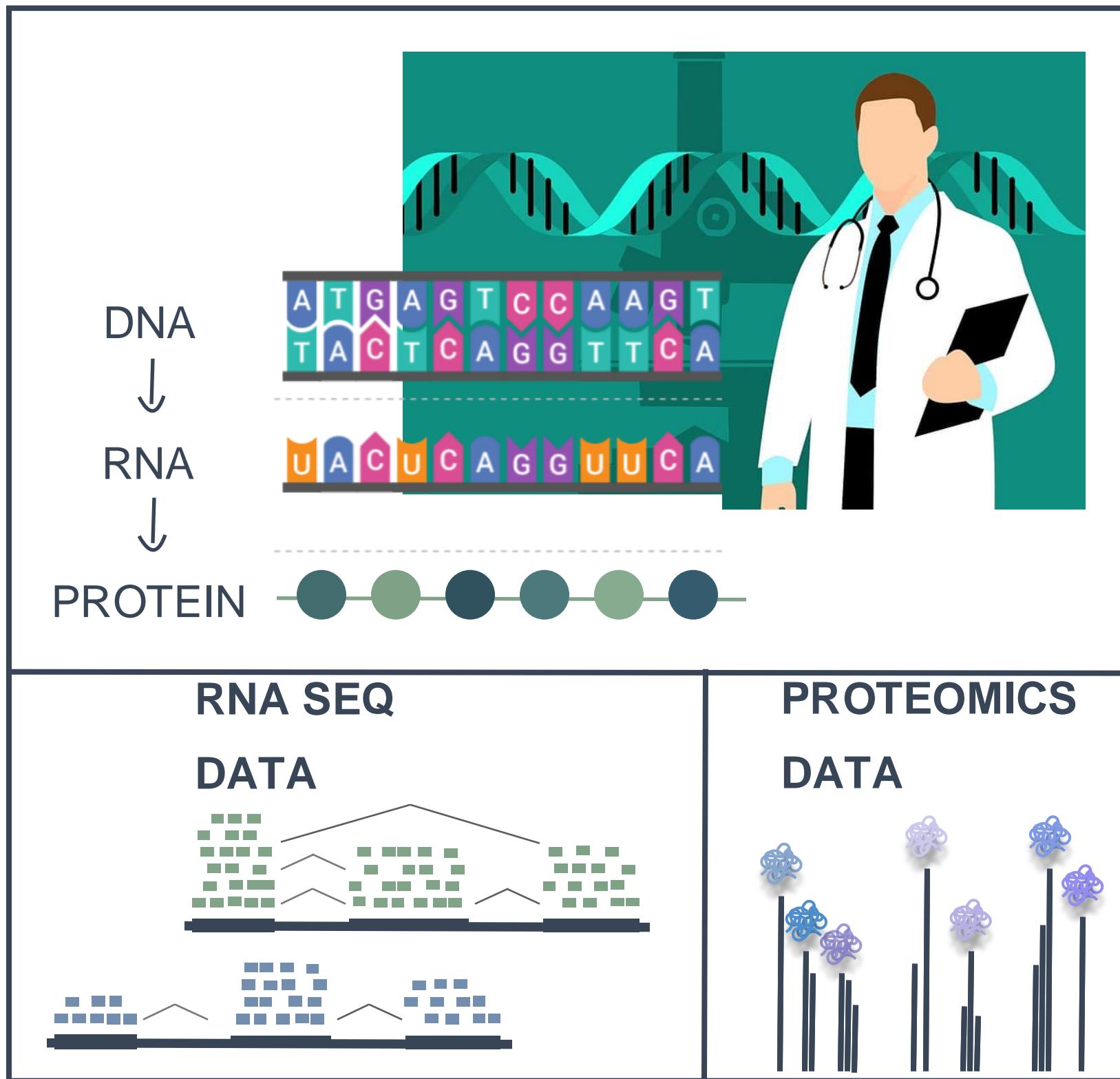


- Linear regression
- Summary Statistics
- ANOVA
- Logistic regression
- Clustering
- Correlation

— Statistics in R
EXERCISE 4

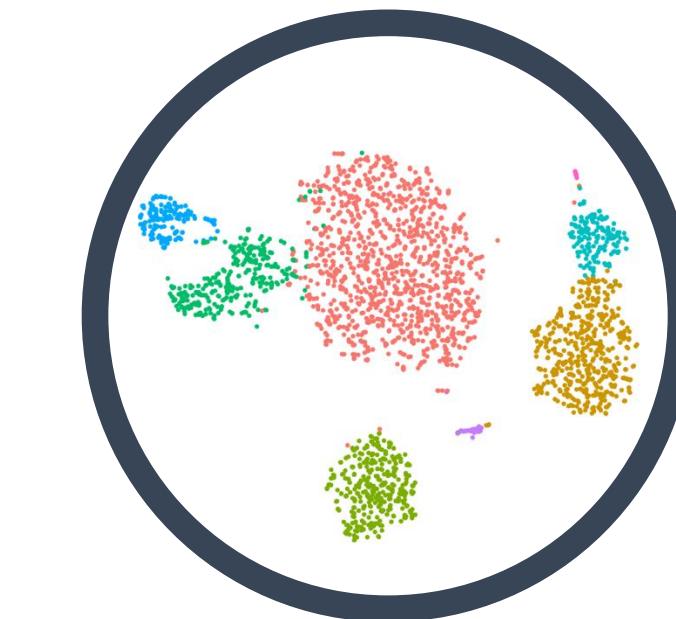
BIOINFORMATICS IN R

HIGH THROUGHPUT DATA



BIOINFORMATIC ANALYSIS

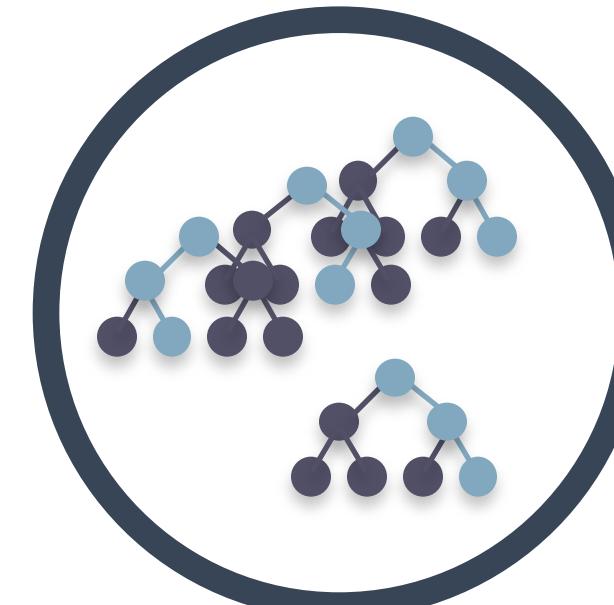
DIMENSIONALITY REDUCTION



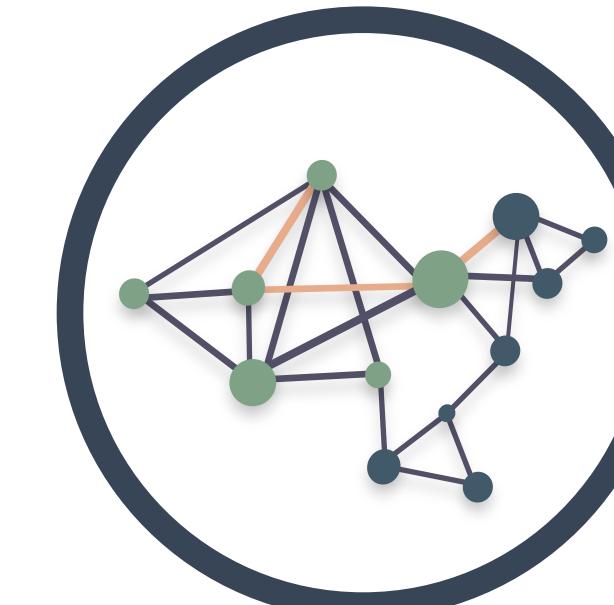
CLUSTERING

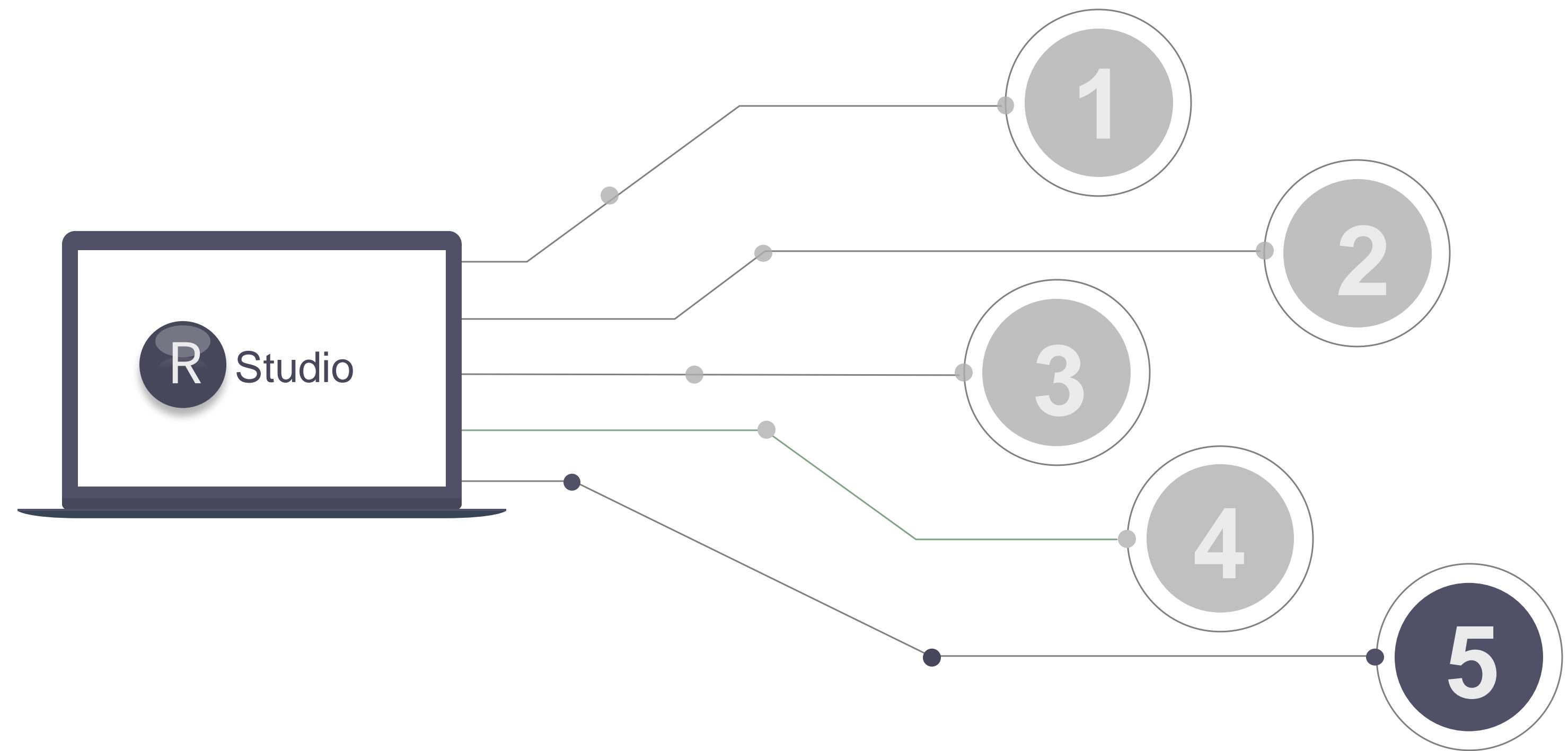


MACHINE LEARNING



NETWORK ANALYSIS





- Tidyverse
- ggplot
- Stats
- Quarto

— Statistics in R
EXERCISE 5

THE TOP OF THE R ICEBERG



STATISTICAL ANALYSIS

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

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

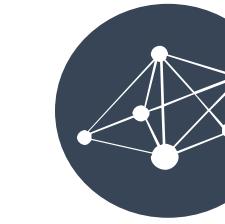
Survival analysis (Cox, Kaplan meier)



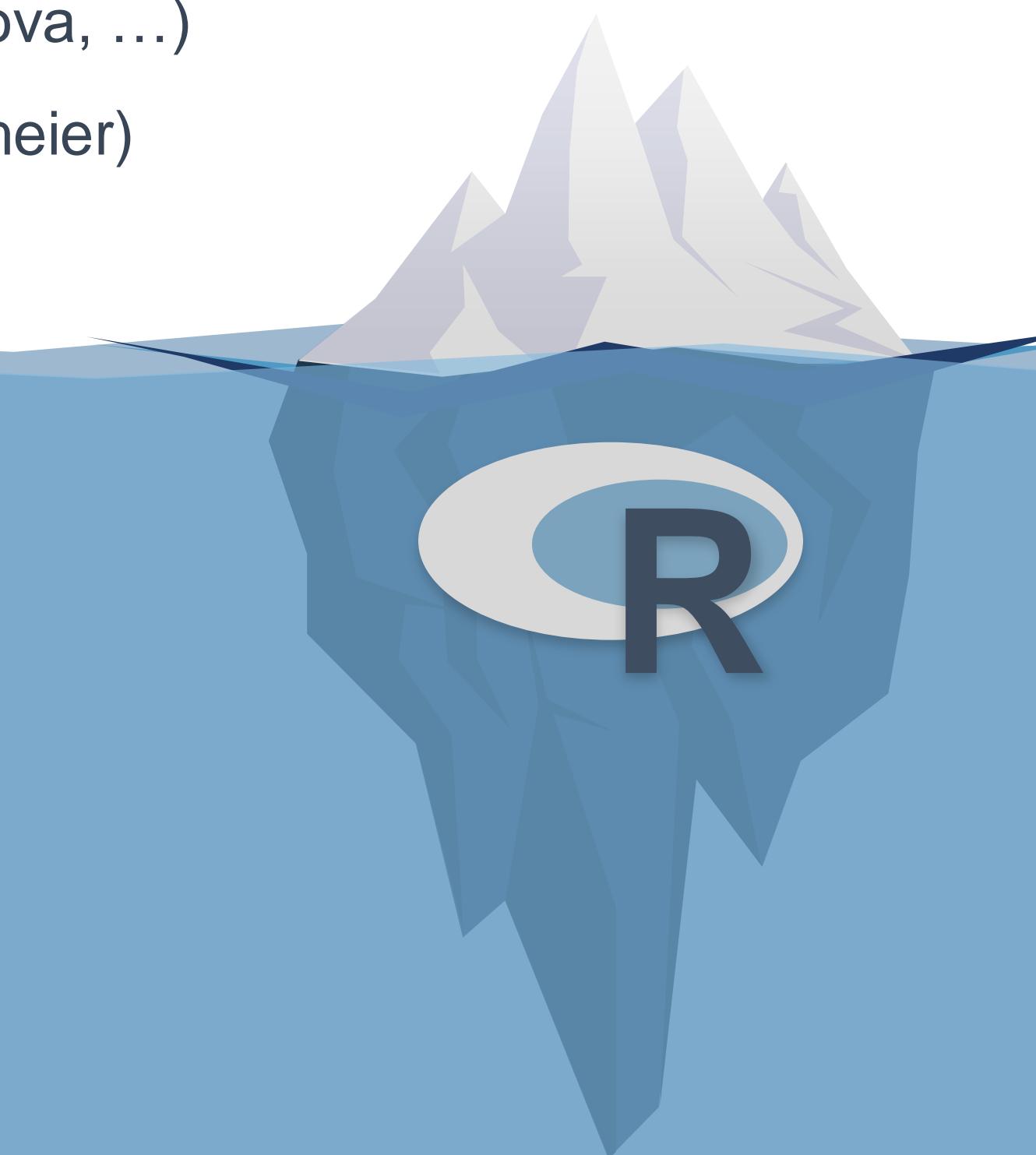
DATA MANGEMENT



EASY PLOTTING



BIOINFORMATIC ANALYSIS



— FROM EXCEL TO R

WANT MO-R-E?



The Section for Biostatistics offers a number statistics-oriented R courses:

Spring

[Basic Statistics for Health Researchers \(Danish course\)](#)

ECTS: 9,0

[Epidemiological methods in medical research](#)

ECTS: 7

[Advanced topics in health research B](#)

ECTS: 2,8

[Statistical methods in bioinformatics](#)

ECTS: 3,5

[Statistical analysis of survival data](#)

ECTS: 4,9

<https://publichealth.ku.dk/about-the-department/biostat/>

[Programming and statistical modelling in R](#)

ECTS: 1,6

[Bayesian methods in biomedical research](#)

ECTS: 2,4

[Psychometric validation of patient reported outcome measures](#)

ECTS: 2,5

[Introduction to validation of patient reported outcome measures](#)

[Causal inference I](#)

ECTS: 2,5

[Use of the statistical software R](#)

ECTS: 2,1

* These are screenshots. Go to the website and scroll down to 'Teaching'

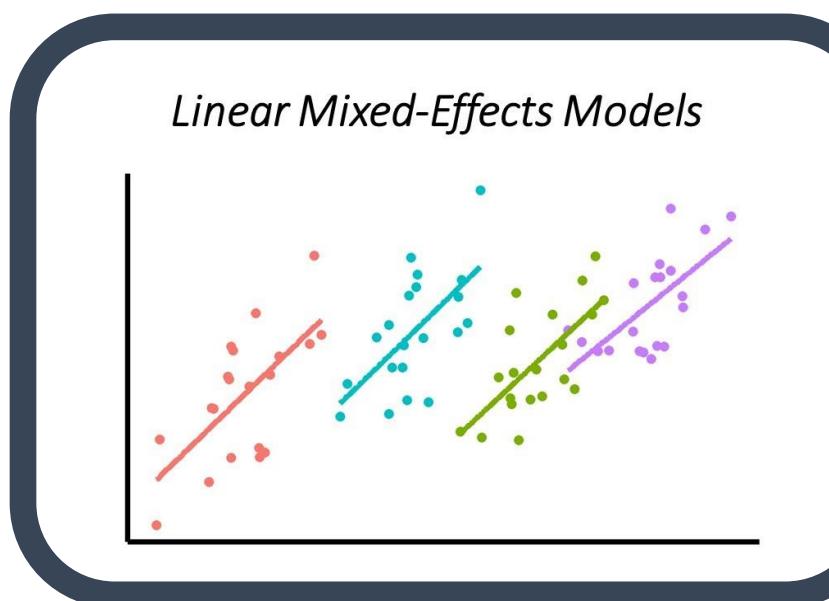


— TEASER STATISTICS in R



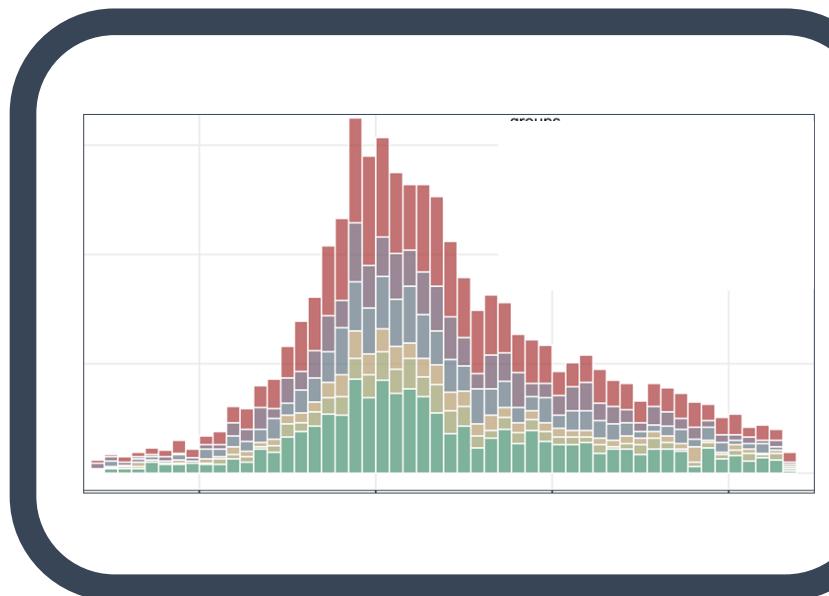
Survival Analysis

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/>)



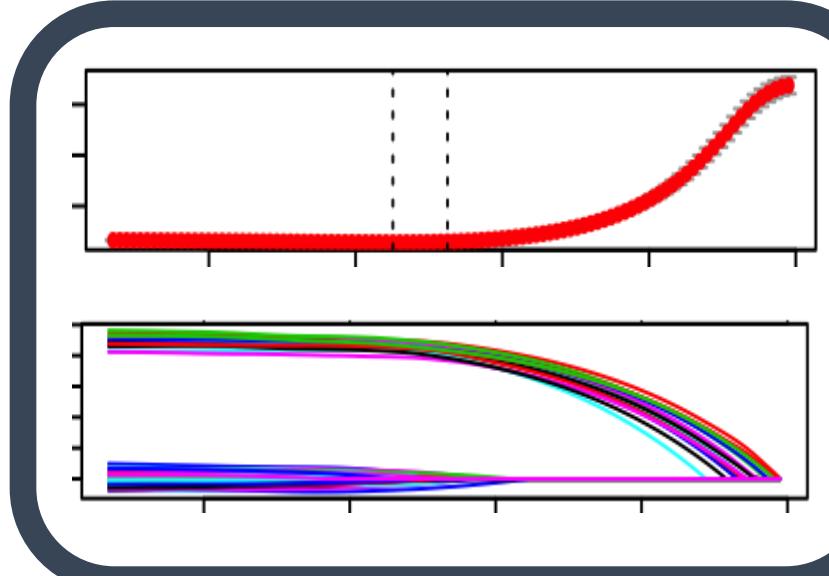
Mixed-Effects Models

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

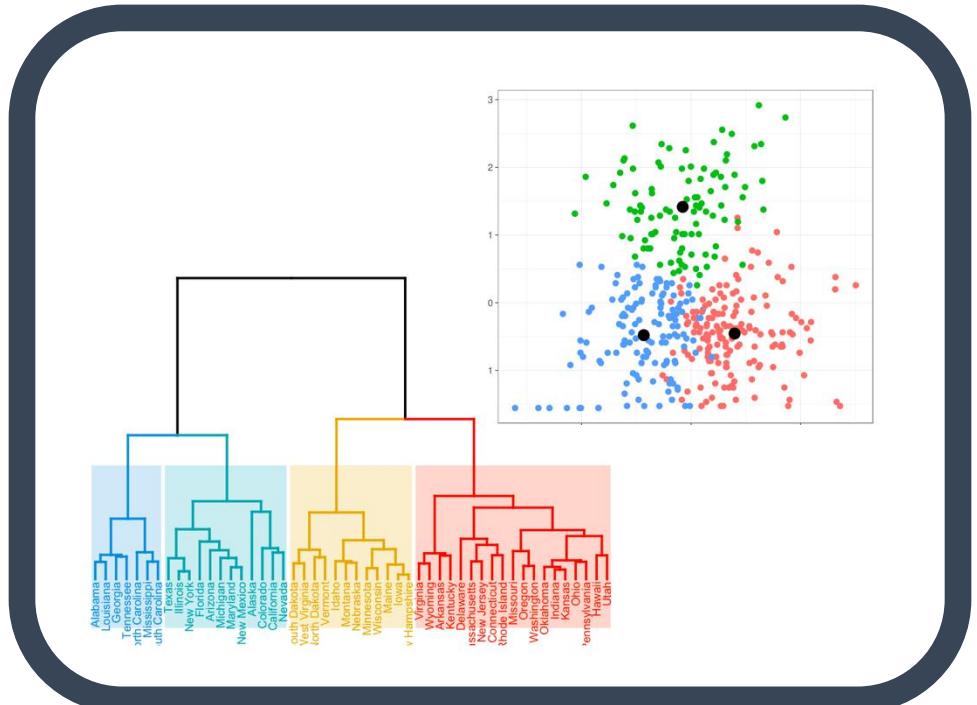
(R)
glmnet: <https://cran.r-project.org/web/packages/glmnet/glmnet.pdf>
elasticnet: <https://cran.r-project.org/web/packages/elasticnet/elasticnet.pdf>
<https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net>

TEASER

Machine Learning

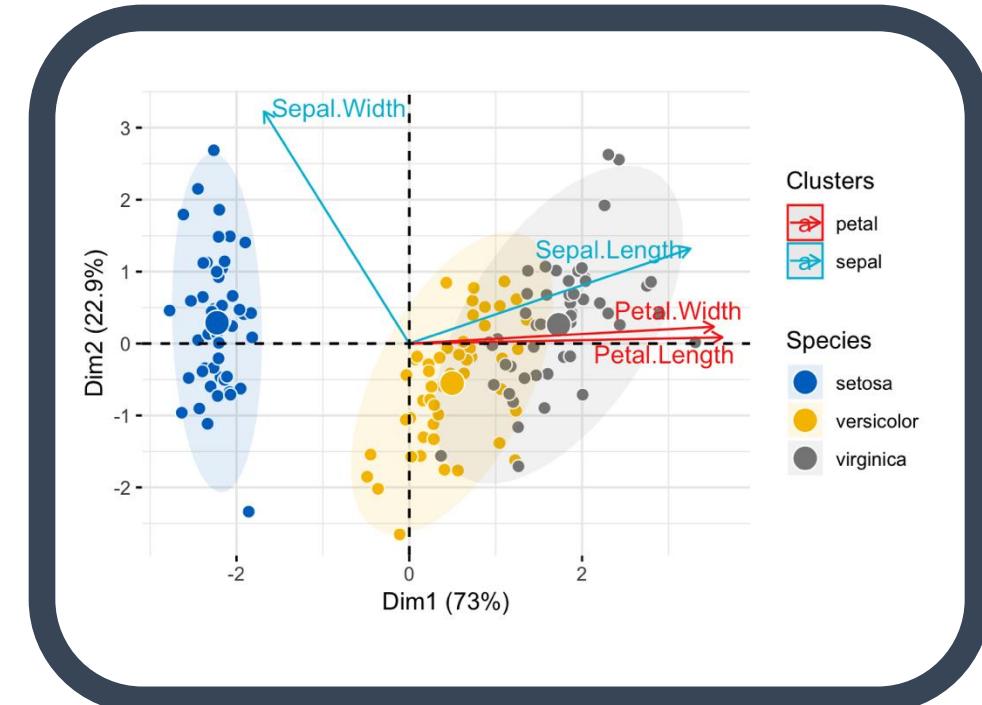
<https://lgatto.github.io/IntroMachineLearningWithR/an-introduction-to-machine-learning-with-r.html>

Clustering



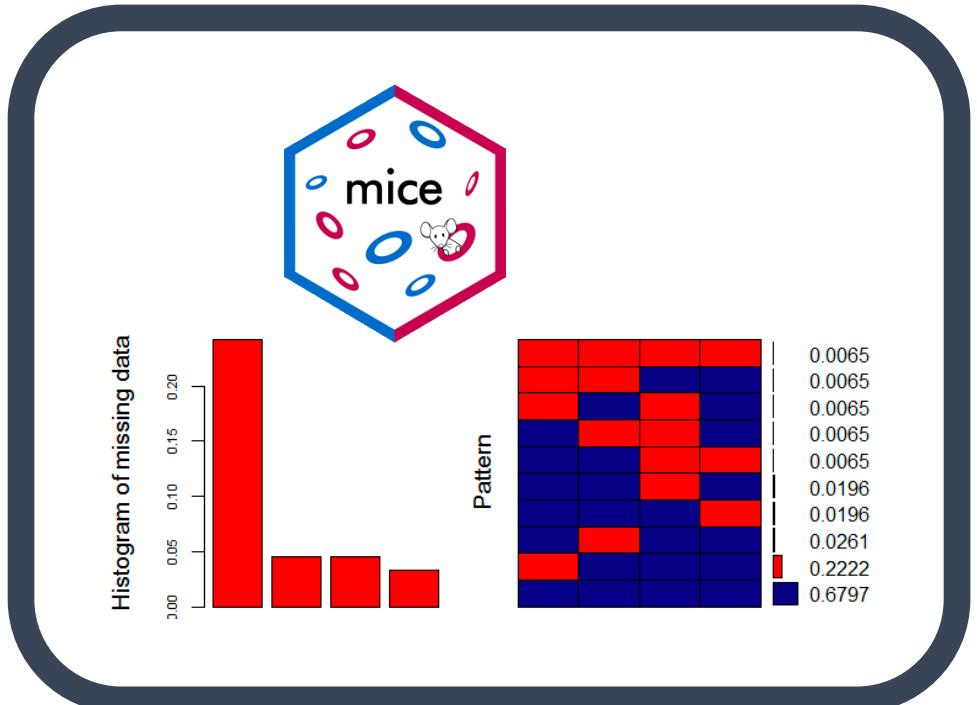
<https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/>

Feature Selection: PCA



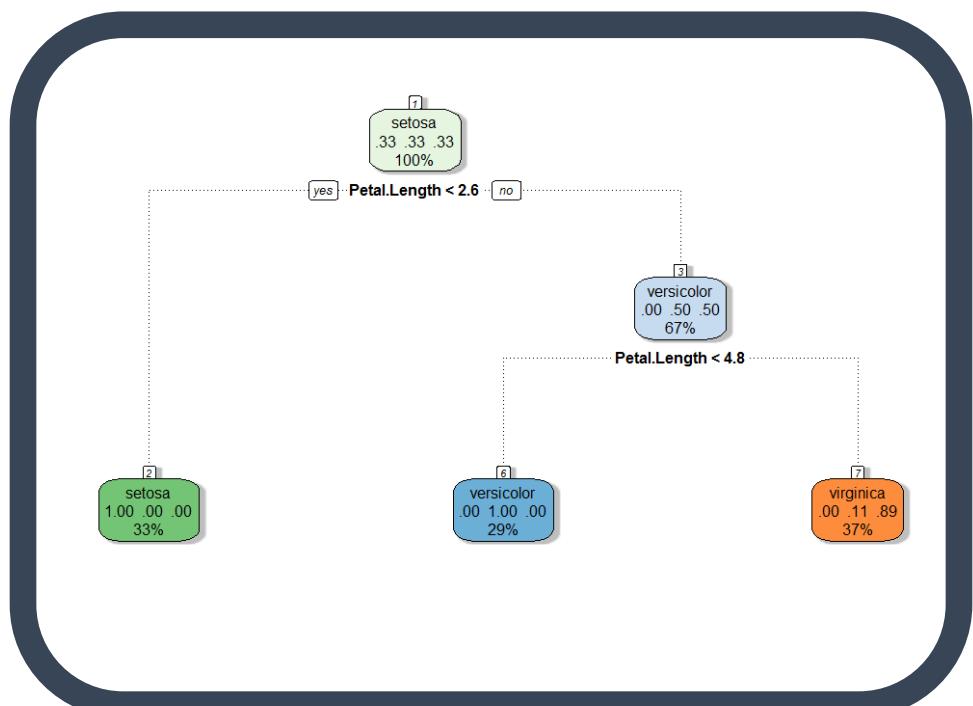
<https://bioconductor.org/packages/release/bioc/vignettes/PCAtools/inst/doc/PCAtools.html>

Missing Data



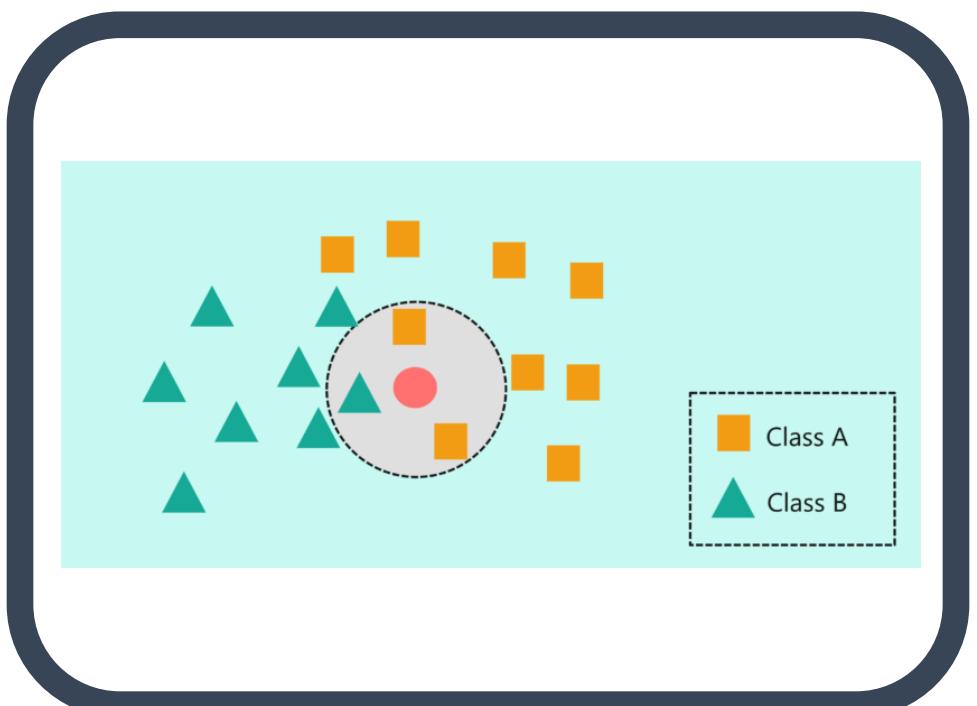
<https://amices.org/mice/>
<https://datascienceplus.com/imputing-missing-data-with-r-mice-package/>

Random Forest



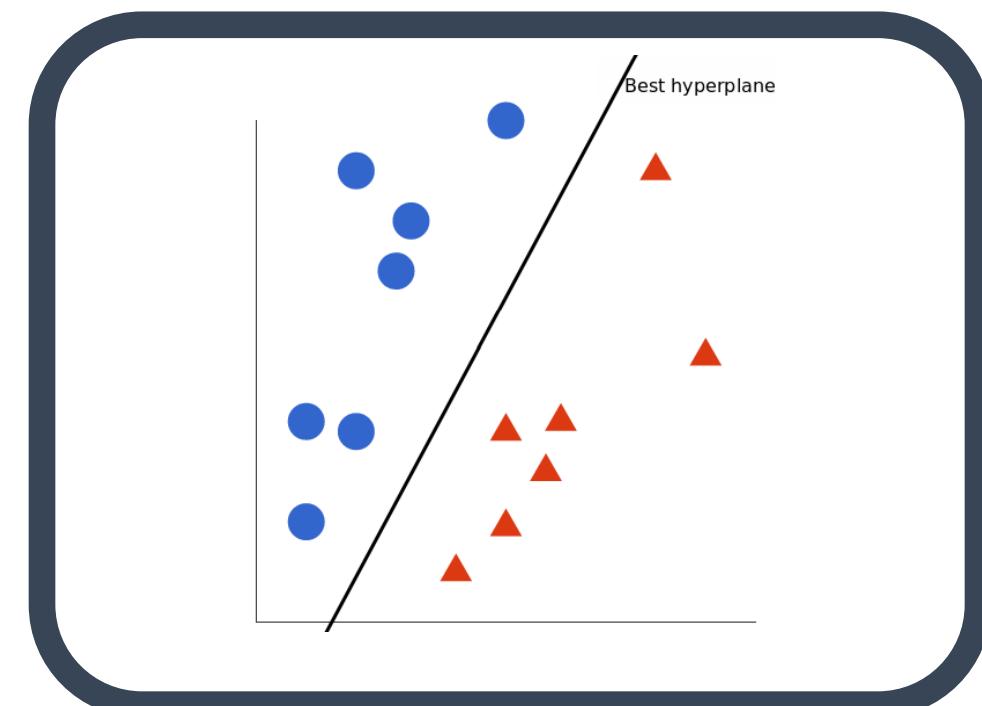
<https://www.blopig.com/blog/2017/04/a-very-basic-introduction-to-random-forests-using-r/>

kNN



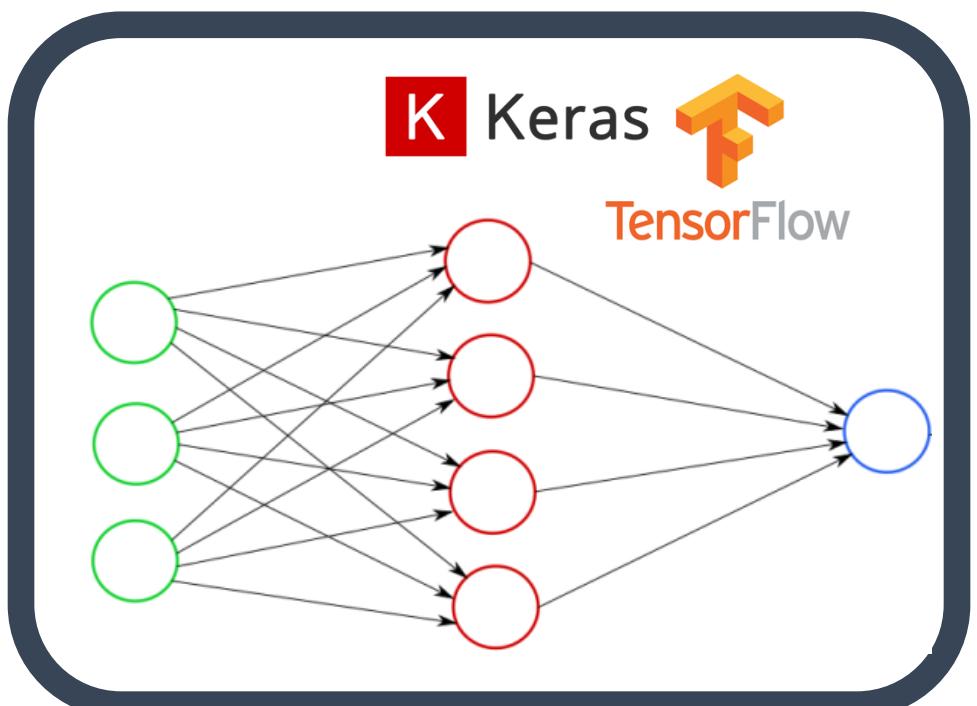
<https://www.edureka.co/blog/knn-algorithm-in-r/>

SVM



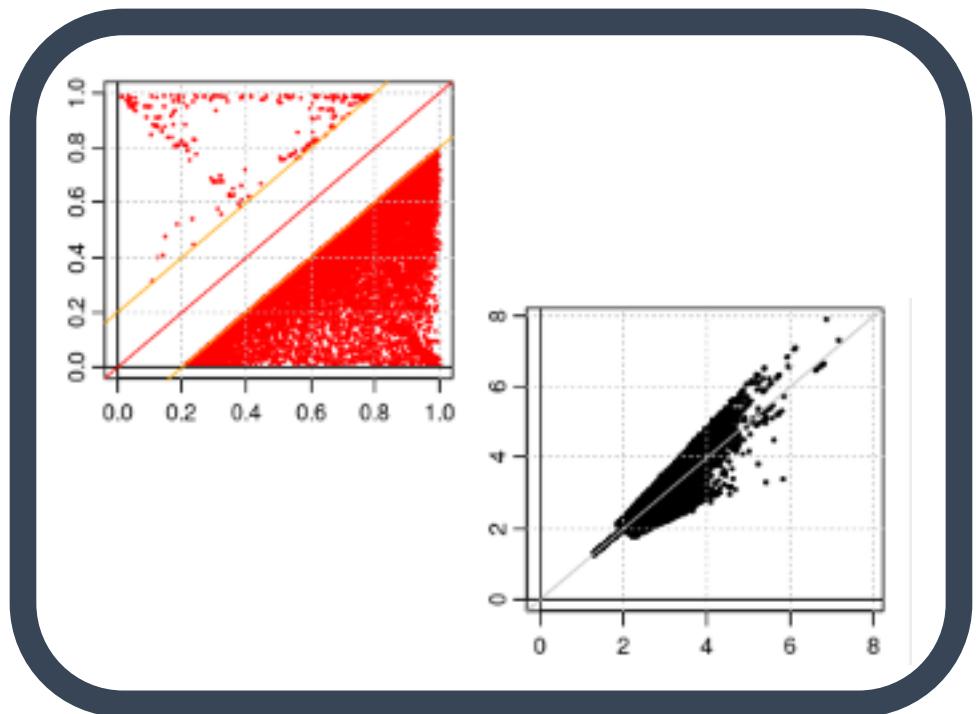
<https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>

Neural Networks



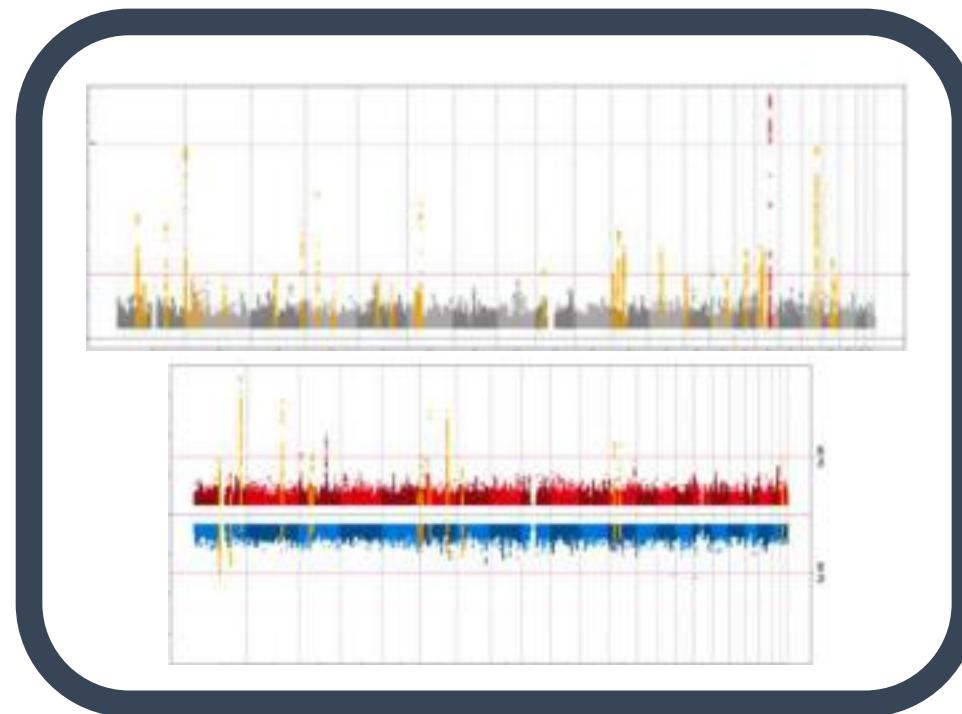
<https://keras.rstudio.com/>
[https://tensorflow.rstuio.com/](https://tensorflow.rstudio.com/)

GWAS - QC & Data Harmonization



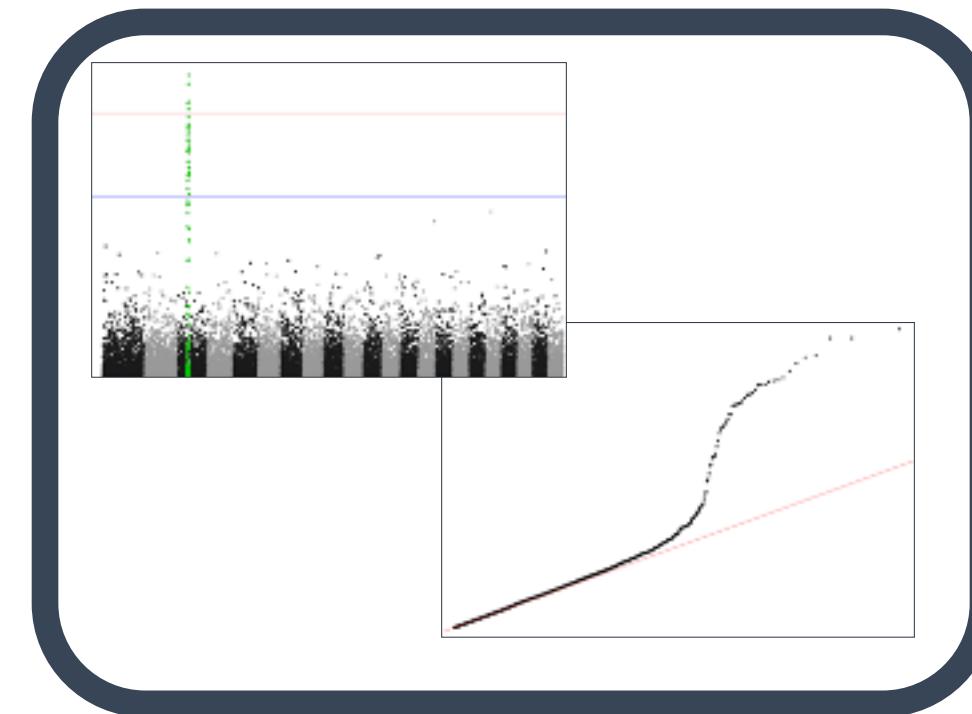
EasyQC: <https://www.uni-regensburg.de/medizin/epidemiologie-praeventivmedizin/genetische-epidemiologie/software/>

GWAS Data Management & Plots



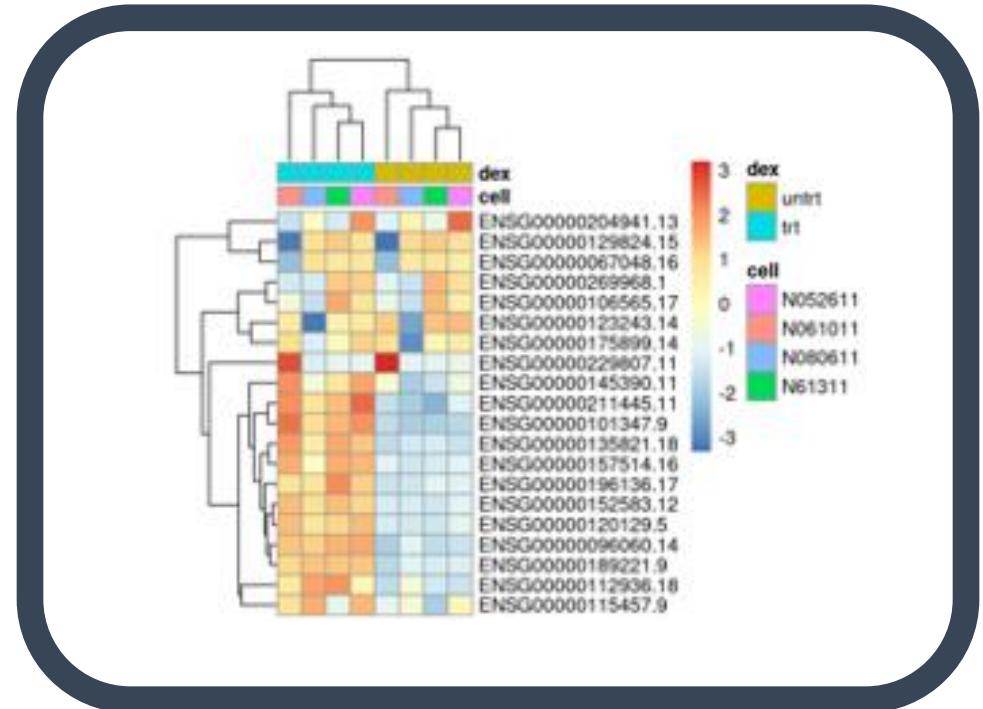
EasyStrata: <https://www.uni-regensburg.de/medizin/epidemiologie-praeventivmedizin/genetische-epidemiologie/software/>

More Plotting...



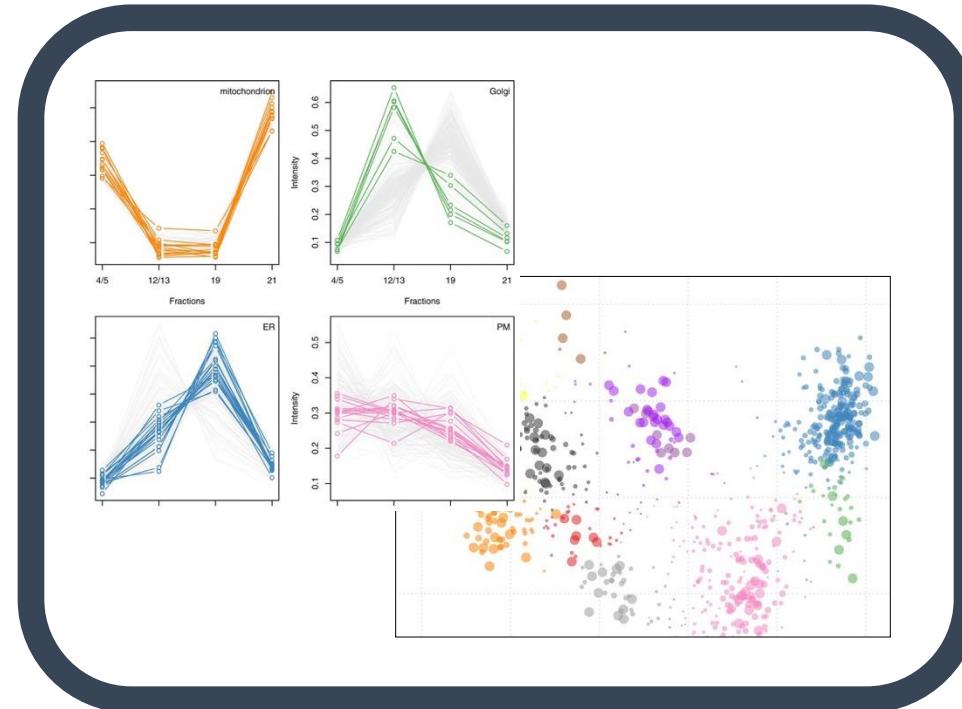
Manhattan and QQ plots:
<https://cran.r-project.org/web/packages/qqman/vignettes/qqman.html>

Gene Expression Analysis



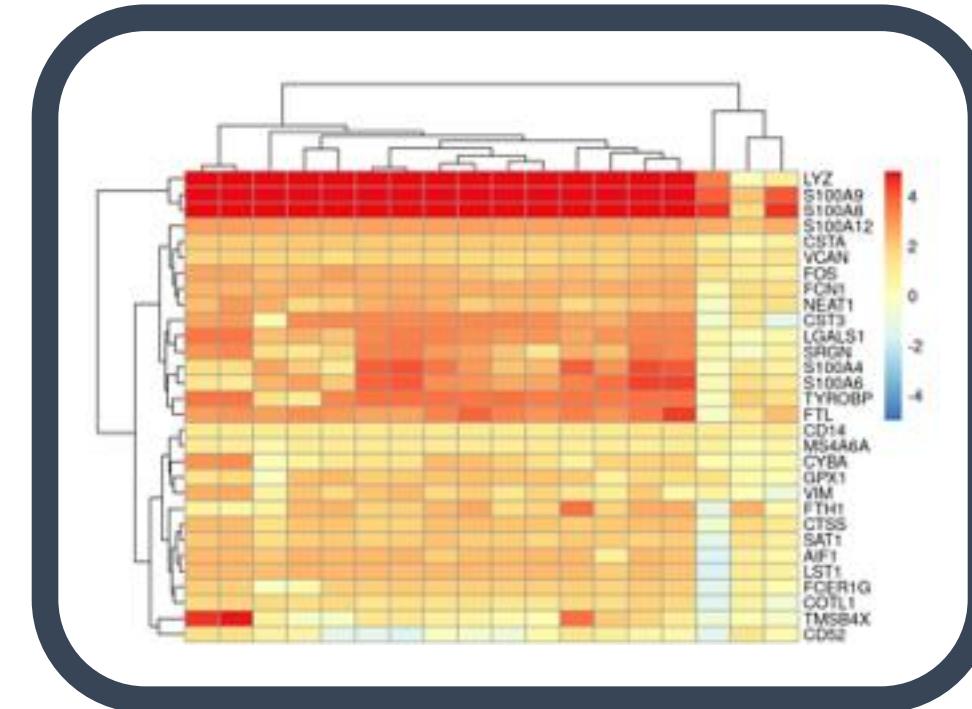
DESeq2, limma, EdgeR, etc.:
http://www.bioconductor.org/packages/release/BiocViews.html#_RNASeq

Proteomics Analysis



RforProteomics:
https://www.bioconductor.org/packages/release/BiocViews.html#_Proteomics

Single-Cell RNASeq



TEASER

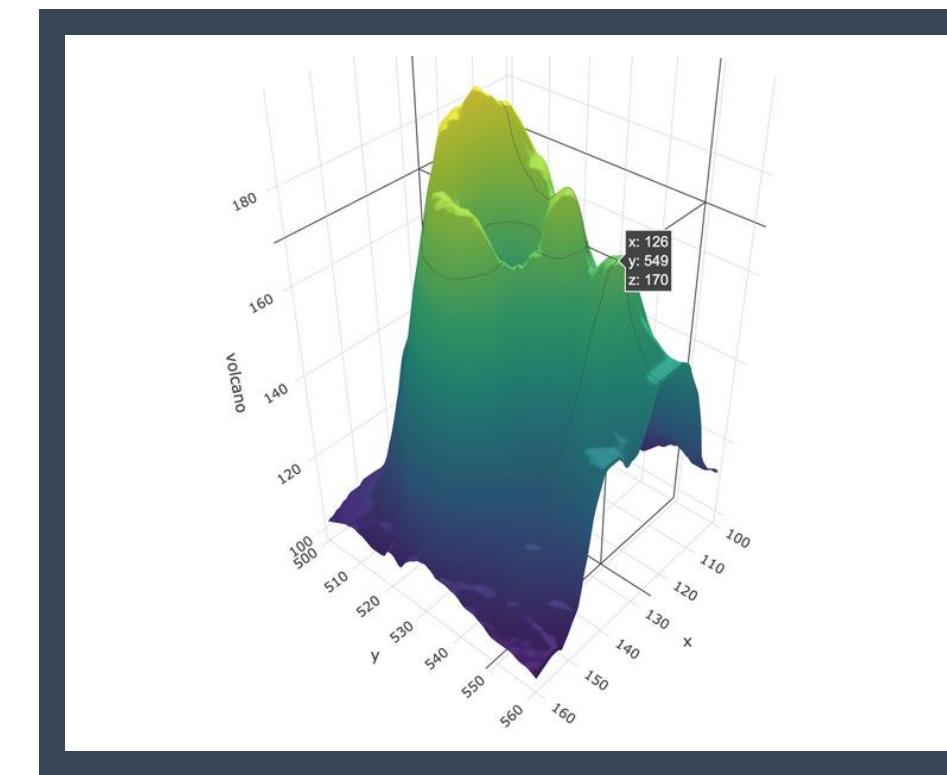
Omics Data

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

COOL STUFF IN R

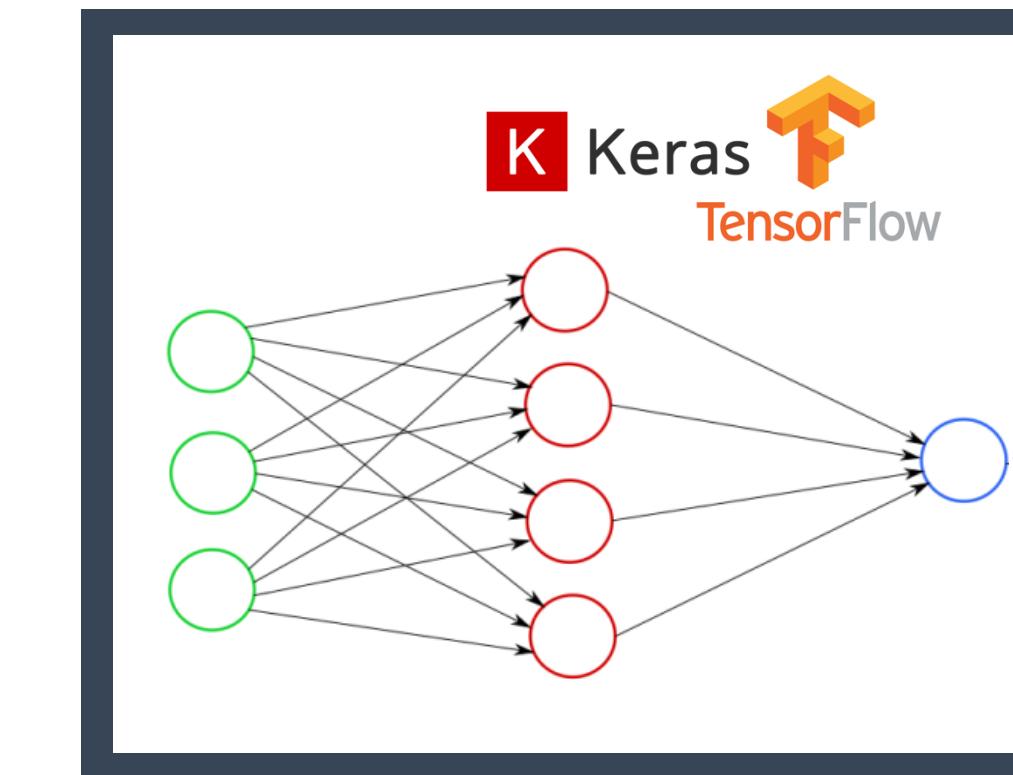
— FROM EXCEL TO R

PLOTTING IN 3D



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

DEEP LEARNING



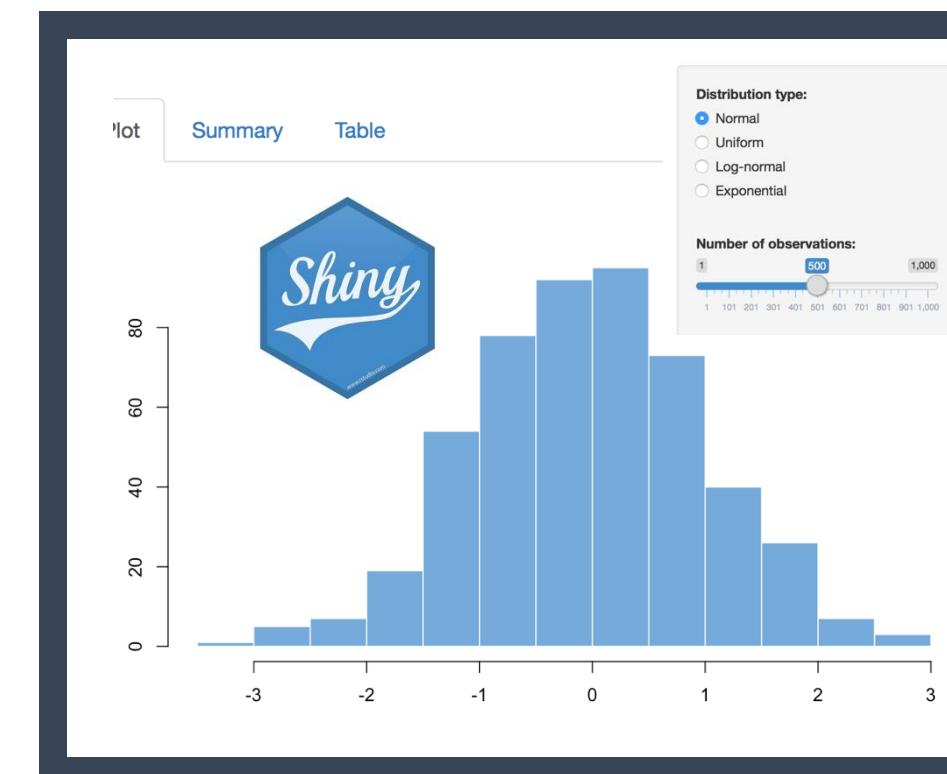
<https://keras.rstudio.com/>
<https://tensorflow.rstudio.com/>

BAYESIAN STATISTICS



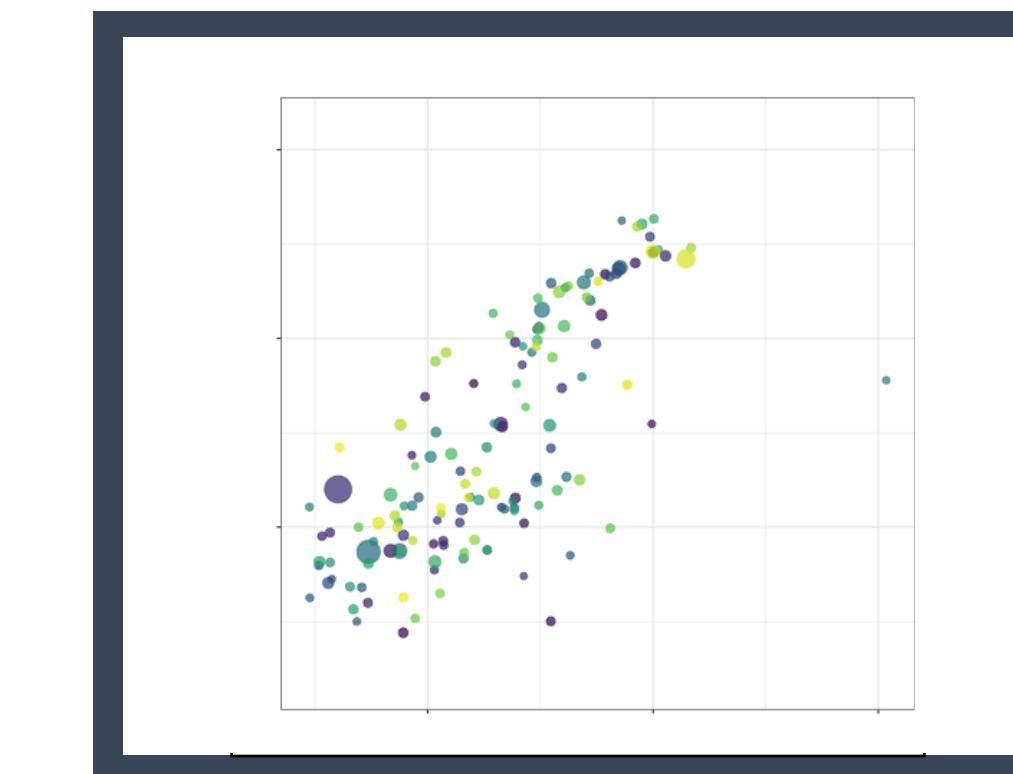
<https://mc-stan.org/users/interfaces/rstan>

WEBPAGE WITH R SHINY



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

INTERACTIVE PLOTS



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

MAIL AND MESSAGES



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

DATA SCIENCE WITH HEADS DATA LAB

- Drop-in Thursday (13-15)
- Consultations
- Commission Research & Supervision



Read more on HeaDS Website!

<https://heads.ku.dk>

- Events and Conferences
- Course Reminders



Follow HeaDS on LinkedIn!

<https://www.linkedin.com/company/ucph-heads/>

THANK YOU FOR LISTENING



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