University of Belgrade – School of Electrical Engineering (ETF), https://www.etf.bg.ac.rs/
PSSOH Conference, http://pssoh.etf.rs/

R for Data Science

Assist. Prof. Nadica Miljković, PhD

nadica.miljkovic@etf.bg.ac.rs

WHAT TO EXPECT?

Introductory course

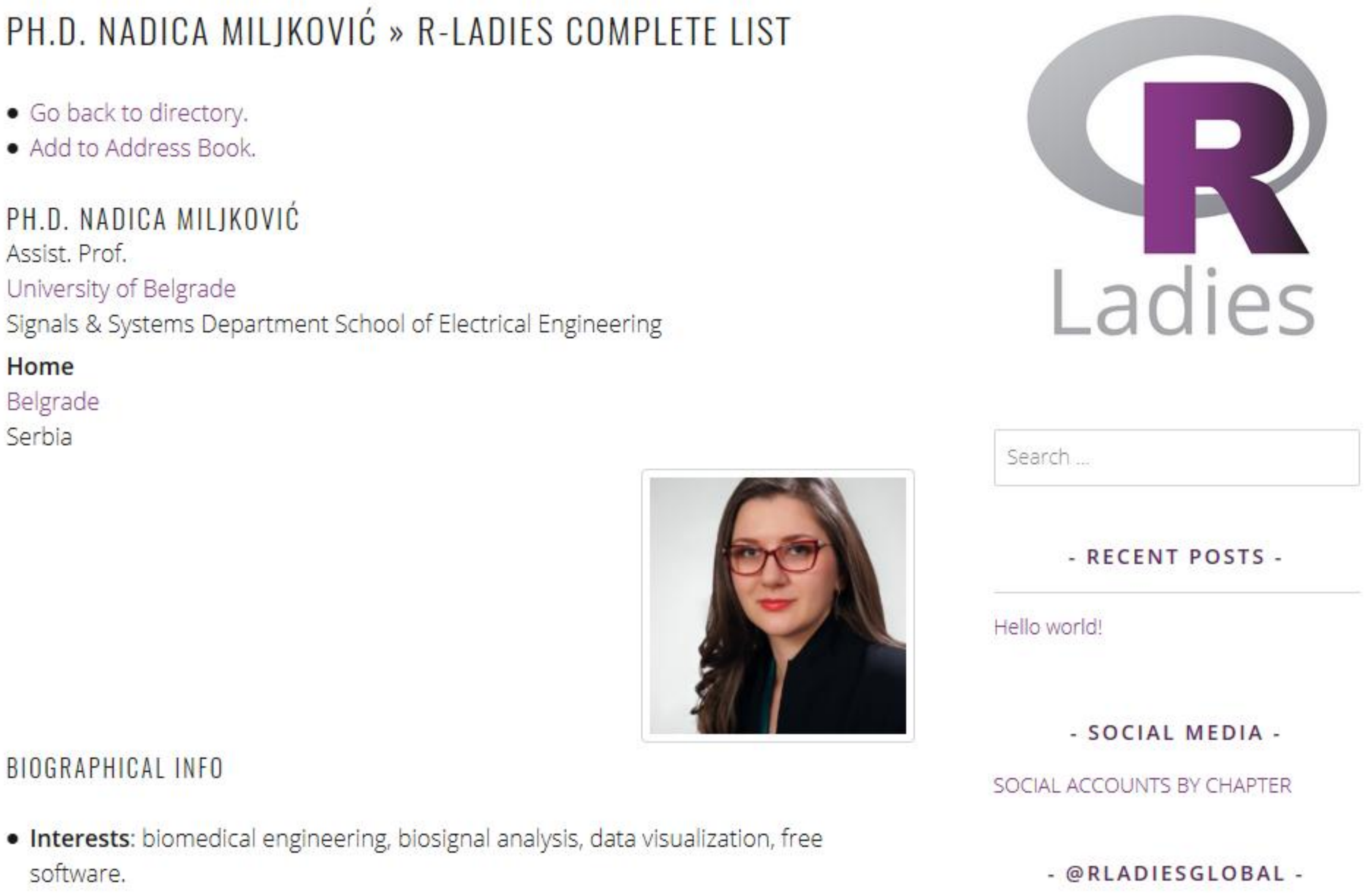
- R programming (https://www.r-project.org/about.html)
  - What is R? What is Data Science?
  - Data Science & R-Ladies Communities in Serbia
  - data types & grammar
- Hands-on
  - Data manipulation
  - Data visualization
- Final remarks
  - Data Ethics
  - Recommended books
  - Recommended resources

Goals

- Main goal: to help you all start with R programming and to pursue your career in data science… maybe you decide it’s not for you…
- Indirect goal: to promote PSSOH, R-Ladies, Data Science Communities, and to enhance R programming and Data Science in Serbia and at the University of Belgrade – School of Electrical Engineering.

Who am I?

- My name is Nadica Miljković.
- I am Assist. Prof. of Biomed. Eng. at the ETF.
- I have 10 yrs. long experience in Academia and 6 yrs. in industry.
- For a long time (> decade) I used Matlab mainly.
- Prof. Predrag Pejović introduced me with:
  - free software and
  - R.
- I searched for R. No results. Tried “R programming” instead.
- Personal website: http://automatika.etf.bg.ac.rs/en/department-personnel/98-english/content/faculty/615-phd-nadica-miljkovi%C4%87
- Short disclaimer: this course is mostly based on the course that I teach on R at the ETF: http://automatika.etf.bg.ac.rs/sr/13m051tobs.



PH.D. NADICA MILJKOVIĆ » R-LADIES COMPLETE LIST

- Go back to directory.
- Add to Address Book.

PH.D. NADICA MILJKOVIĆ

Assist. Prof.
University of Belgrade
Signals & Systems Department School of Electrical Engineering

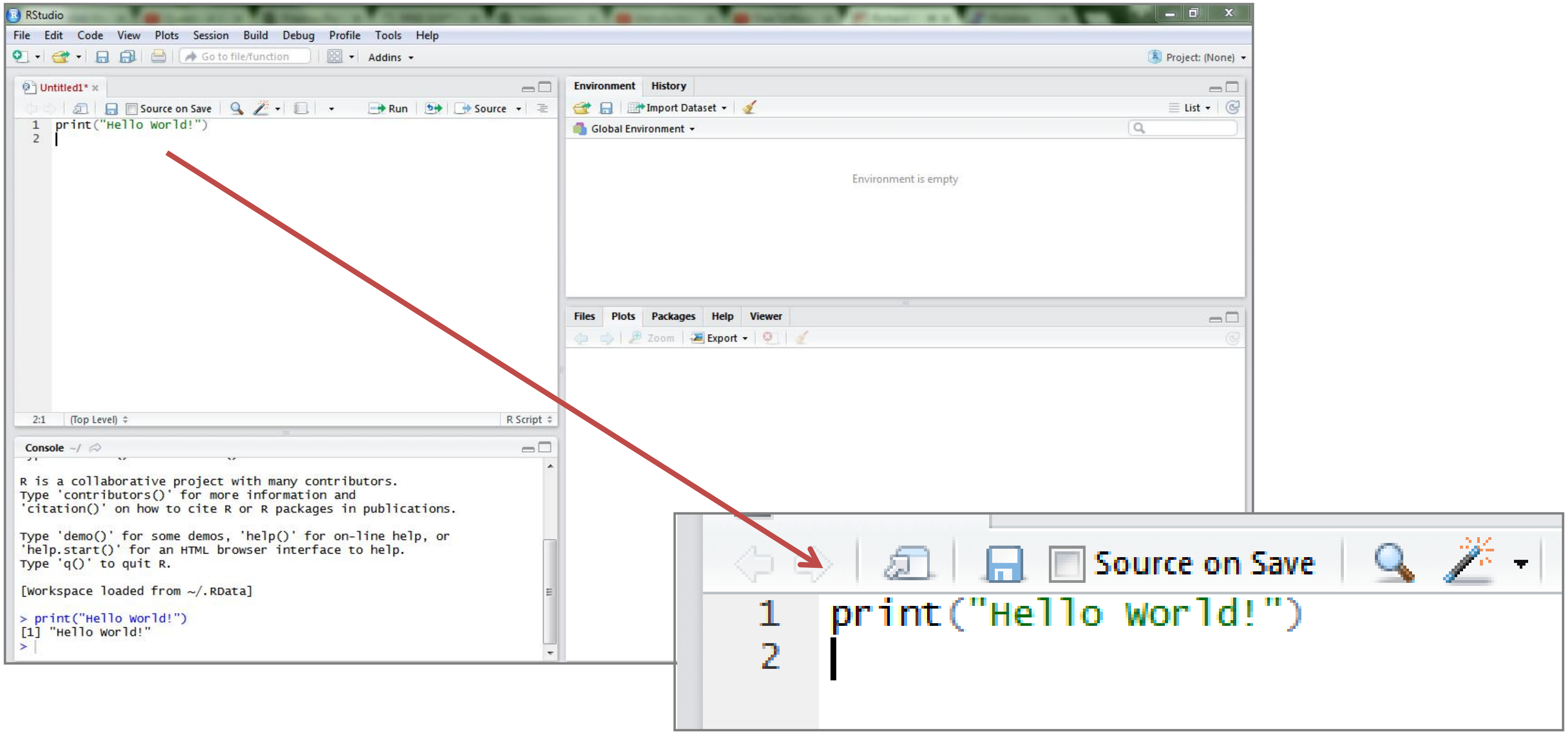
Home
Belgrade
Serbia

BIOGRAPHICAL INFO

- **Interests**: biomedical engineering, biosignal analysis, data visualization, free software.

EYES ON SLIDES & HANDS ON KEYBOARD

Shall we start?



```
print("Hello World!")
```

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

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

[Workspace loaded from ~/.RData]

> print("Hello World!")
[1] "Hello World!"
>
```

- First part: Solve BSS (Blank Screen Syndrome)
- Second part: Gentle Introduction to R
- Third part: Apply gathered knowledge and do something useful

What is R?

- Past, present, and future of R (https://simplystatistics.org/2018/07/12/use-r-keynote-2018/, [Online], Assessed October 17, 2018):
  - “R is a free software environment for statistical computing and graphics”, https://www.r-project.org/, [Online], Assessed October 17, 2018.
  - “The tidyverse is an opinionated collection of R packages designed for data science.”, https://www.tidyverse.org/, [Online], Assessed October 17, 2018.
- Maybe it’s accurate to say that R is a programming language for data analysis.

History & facts

- 1991: created by Ross Ihaka and Robert Gentleman
- 1995. R became free/libre
- 2000. R 1.0.0 was available
- R is *lingua franca* of data science
- R is not used just for statistics
- It is useful to learn R ... guess you know that
- Many R packages
- Large R community
- R is relatively simple to learn even for non coders/programmers
- Great for research

Free software

- Free software guarantees four essential freedoms (http://www.fsf.org/):
  - to run the program as you wish, for any purpose (**freedom 0**)
  - to study how the program works, and change it so it does your computing as you wish (**freedom 1**)
  - to redistribute copies so you can help others (**freedom 2**)
  - to distribute copies of your modified versions to others (**freedom 3**)
- More about FS: https://www.gnu.org/philosophy/free-software-even-more-important.html.
- Talk by Richard Stallman the founder of FSF: https://www.fsf.org/blogs/rms/20140407-geneva-tedx-talk-free-software-free-society.
- Open source ≠ Free software
- About open source initiative: https://opensource.org/osd.

CRAN is useful!

- Comprehensive R Archive Network
- Contains R packages:
  - base packages
  - recommended packages
- R packages are all around us. You can look at the:
  - Bioconductor (https://www.bioconductor.org/)
  - Personal web pages of data scientists, analytics, engineers, …
  - Repositories (GitHub, BitBucket, Zenodo, ...)

R advantages

- Free software
- Scalability & functionality
  - as a user you can contribute to CRAN (10065 CRAN packages - 10.02.2017., 12162 packages - 17.02.2018., 13217 packages – 17.10.2018., 15328 – 25.02.2020, and for current information check https://cran.r-project.org/web/packages/)
- Active, supportive, and large community
- Graphical capacity
  - Exceptional quality of graphs
  - Visualization of complex data
  - “lattice”, “ggplot2”, and other famous packages

and disadvantages

- Objects must be stored in internal memory, but:
  - in most cases memory is large enough and
  - specialized packages can help
- If you want to do pioneering work, and there is no package for that:
  - you should write your own code

RAM *vs*. our data

- Increase of RA memory is larger than size of most data sets. (izvor: “RAM is eating bid data”, https://www.linkedin.com/pulse/ram-eating-big-data-size-data-sets-used-analytics-chaaranpall-lambba, [Online], Assessed October 17, 2018)
- How “big” are our data? Up to TB and/or PB... or maybe not
- You can always use computer clusters as Apache^{TM} Hadoop® (http://hadoop.apache.org/, logo by Apache Software Foundation , Apache License 2.0, https://commons.wikimedia.org/w/index.php?curid=63919822).

Data Science software polls

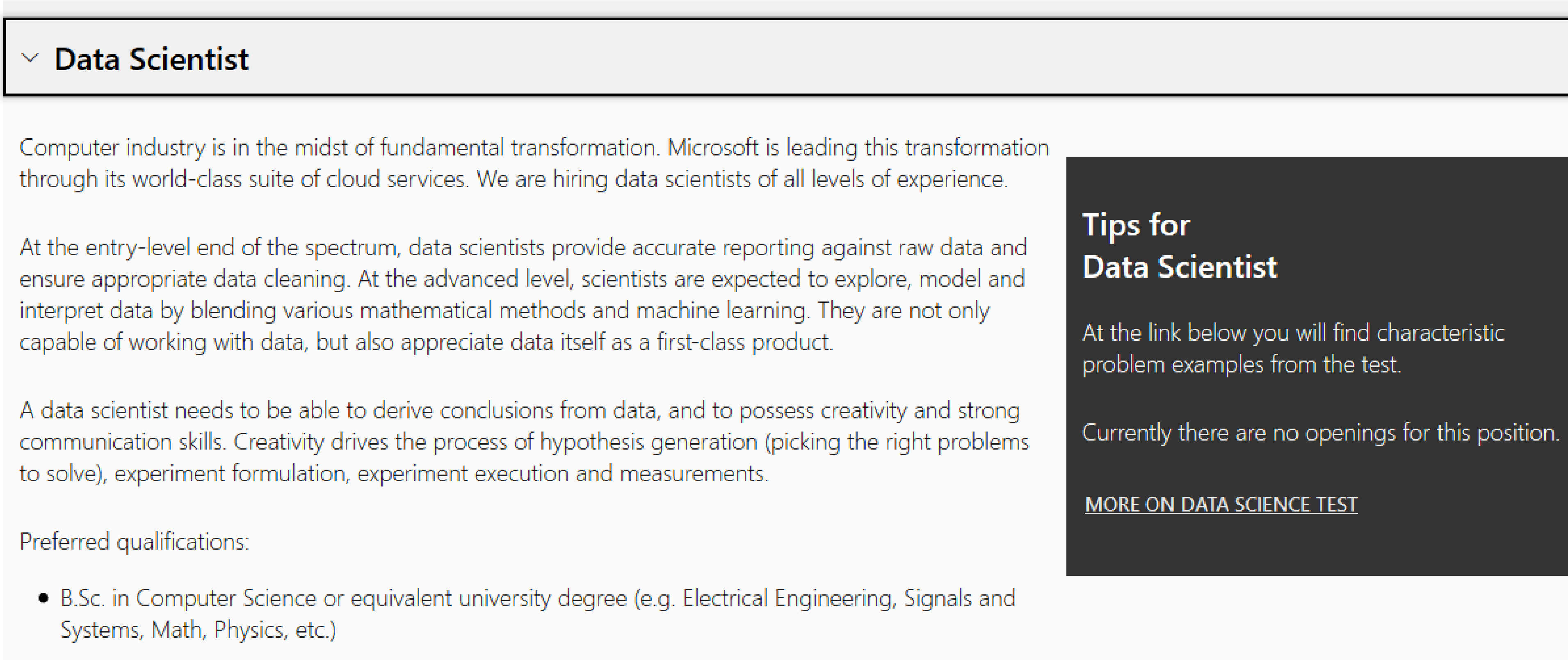
all over the Internet show that there is growing interest and application of R for data science…

Who uses R?

both companies and Academia

there are many reports on Internet

Sample data scientist position



Data Scientist

Computer industry is in the midst of fundamental transformation. Microsoft is leading this transformation through its world-class suite of cloud services. We are hiring data scientists of all levels of experience.

At the entry-level end of the spectrum, data scientists provide accurate reporting against raw data and ensure appropriate data cleaning. At the advanced level, scientists are expected to explore, model and interpret data by blending various mathematical methods and machine learning. They are not only capable of working with data, but also appreciate data itself as a first-class product.

A data scientist needs to be able to derive conclusions from data, and to possess creativity and strong communication skills. Creativity drives the process of hypothesis generation (picking the right problems to solve), experiment formulation, experiment execution and measurements.

Preferred qualifications:

- B.Sc. in Computer Science or equivalent university degree (e.g. Electrical Engineering, Signals and Systems, Math, Physics, etc.)

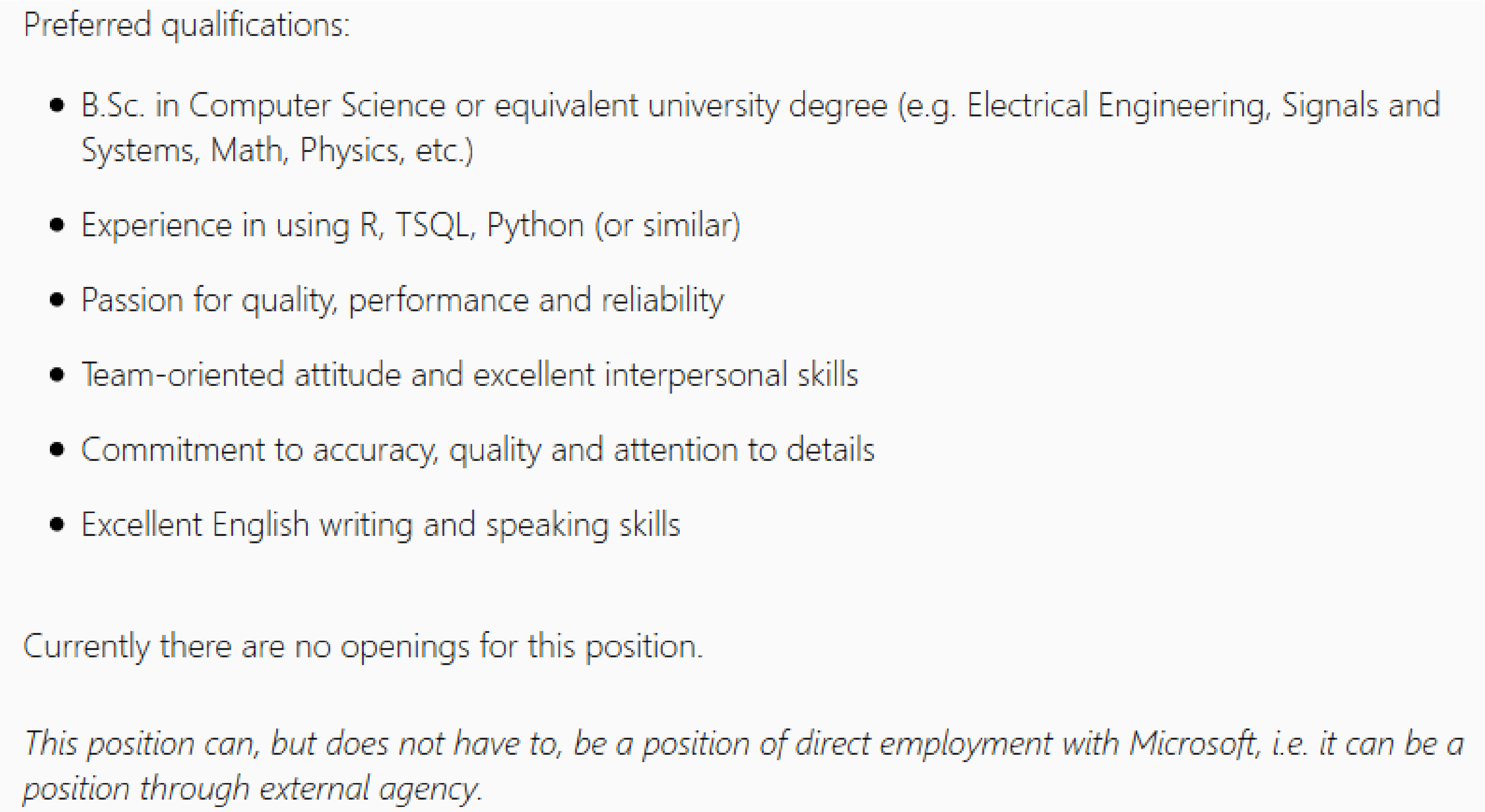
Tips for Data Scientist

At the link below you will find characteristic problem examples from the test.

Currently there are no openings for this position.

https://www.microsoft.com/sr-latn-rs/mdcs/jobs.aspx

Sample data scientist position



Preferred qualifications:

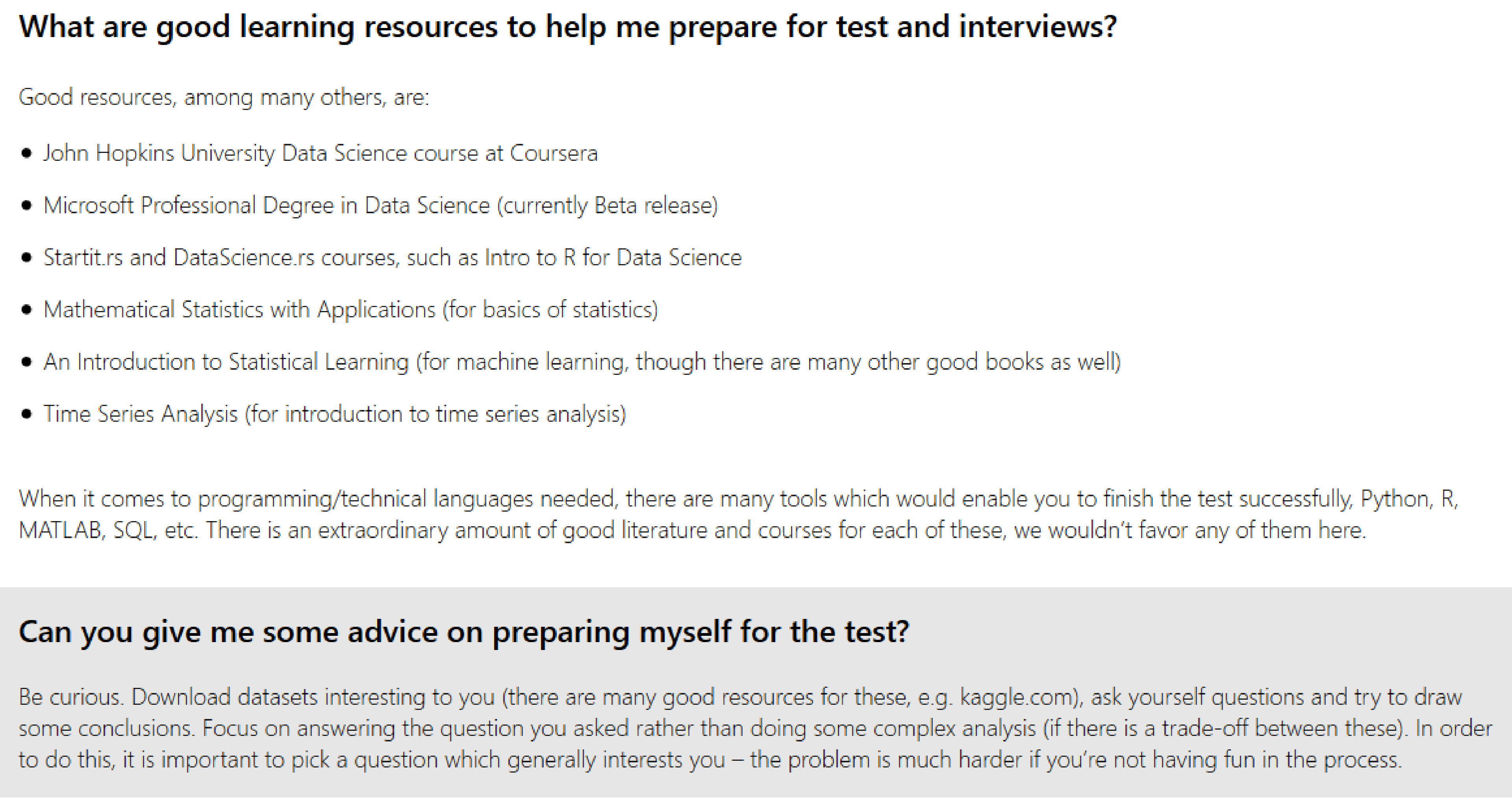
- B.Sc. in Computer Science or equivalent university degree (e.g. Electrical Engineering, Signals and Systems, Math, Physics, etc.)
- Experience in using R, TSQL, Python (or similar)
- Passion for quality, performance and reliability
- Team-oriented attitude and excellent interpersonal skills
- Commitment to accuracy, quality and attention to details
- Excellent English writing and speaking skills

Currently there are no openings for this position.

This position can, but does not have to, be a position of direct employment with Microsoft, i.e. it can be a position through external agency.

https://www.microsoft.com/sr-latn-rs/mdcs/jobs.aspx

Sample data scientist position



What are good learning resources to help me prepare for test and interviews?

Good resources, among many others, are:

- John Hopkins University Data Science course at Coursera
- Microsoft Professional Degree in Data Science (currently Beta release)
- Startit.rs and DataScience.rs courses, such as Intro to R for Data Science
- Mathematical Statistics with Applications (for basics of statistics)
- An Introduction to Statistical Learning (for machine learning, though there are many other good books as well)
- Time Series Analysis (for introduction to time series analysis)

When it comes to programming/technical languages needed, there are many tools which would enable you to finish the test successfully, Python, R, MATLAB, SQL, etc. There is an extraordinary amount of good literature and courses for each of these, we wouldn't favor any of them here.

Can you give me some advice on preparing myself for the test?

Be curious. Download datasets interesting to you (there are many good resources for these, e.g. kaggle.com), ask yourself questions and try to draw some conclusions. Focus on answering the question you asked rather than doing some complex analysis (if there is a trade-off between these). In order to do this, it is important to pick a question which generally interests you – the problem is much harder if you're not having fun in the process.

https://www.microsoft.com/sr-latn-rs/mdcs/prepare-for-ds-test.aspx

Abdominal and Pelvic MRI
Functional MRI
Circuits, Signals, and Systems for Bioengineers
Design and Analysis of Experiments
Statistical Methods for the Analysis of Biomedical Data
Second Edition
Magnetic Resonance Imaging
The MRI Manual
3D Imaging in Medicine
Nuclear Medicine
Third Edition
Medical Instrumentation: Application and Design
Webster, editor
IC User's Casebook
Handbook of Neural Engineering
Akay
Biomechanics VII-B
Biomechanics VII-A
Shepherd
Neurobiology
Oxford
Gross Anatomy
Kyung Won Chung
From Basic Motor Control to Functional Recovery III
Nikolai Gantchev
Motor Control VII
Metabolic and Endocrine Physiology
Tepperman
Neuroanatomy

"*There are only two kinds of languages: the ones people complain about and the ones nobody uses*"
Bjarne Stroustrup (from R.G. Peng. R programming for data science, LeanPub, 2015.)

Who are data scientists?

many definitions

hard to define

I’d ask Hilary Mason: https://hilarymason.com/speaking/

OK?

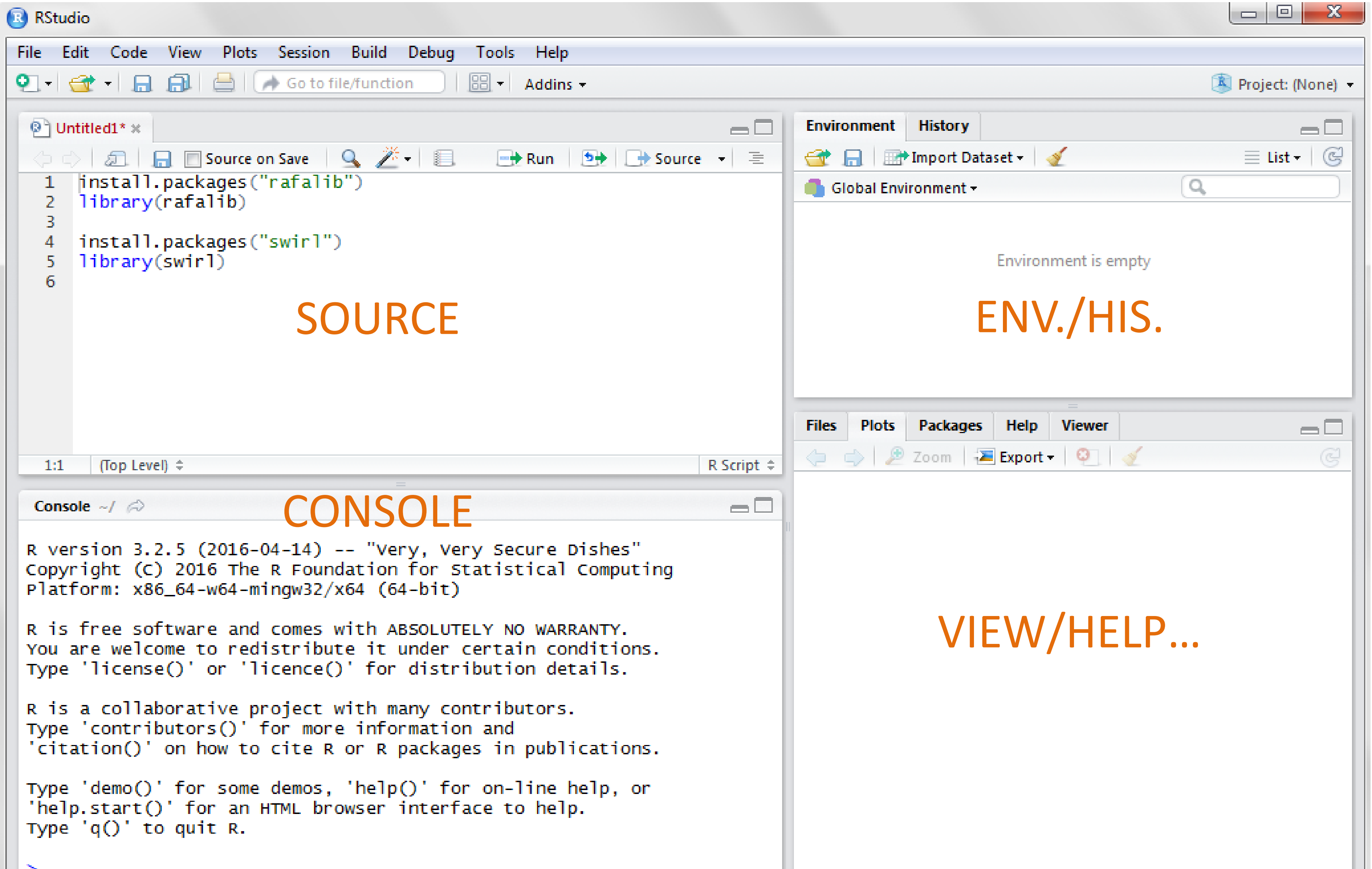
INSTALLING IS THE MOST UNCOMMON,
BUT THE MOST EFFECTIVE STEP

DO IT!

Install!

- Download from CRAN and install R https://cran.r-project.org/ (*Download R for Windows / Base link / Download*).
  - Note: use default installation parameters.
- Then, install R Studio IDE (Integrated Development Environment) from https://www.rstudio.com/ (Desktop option).
  - Note: your explorer should recognize the operating system automatically. If not, you'll be using Windows. Yes, I know… but I do combine free, proprietary, and open… story for some other presentation and occasion.

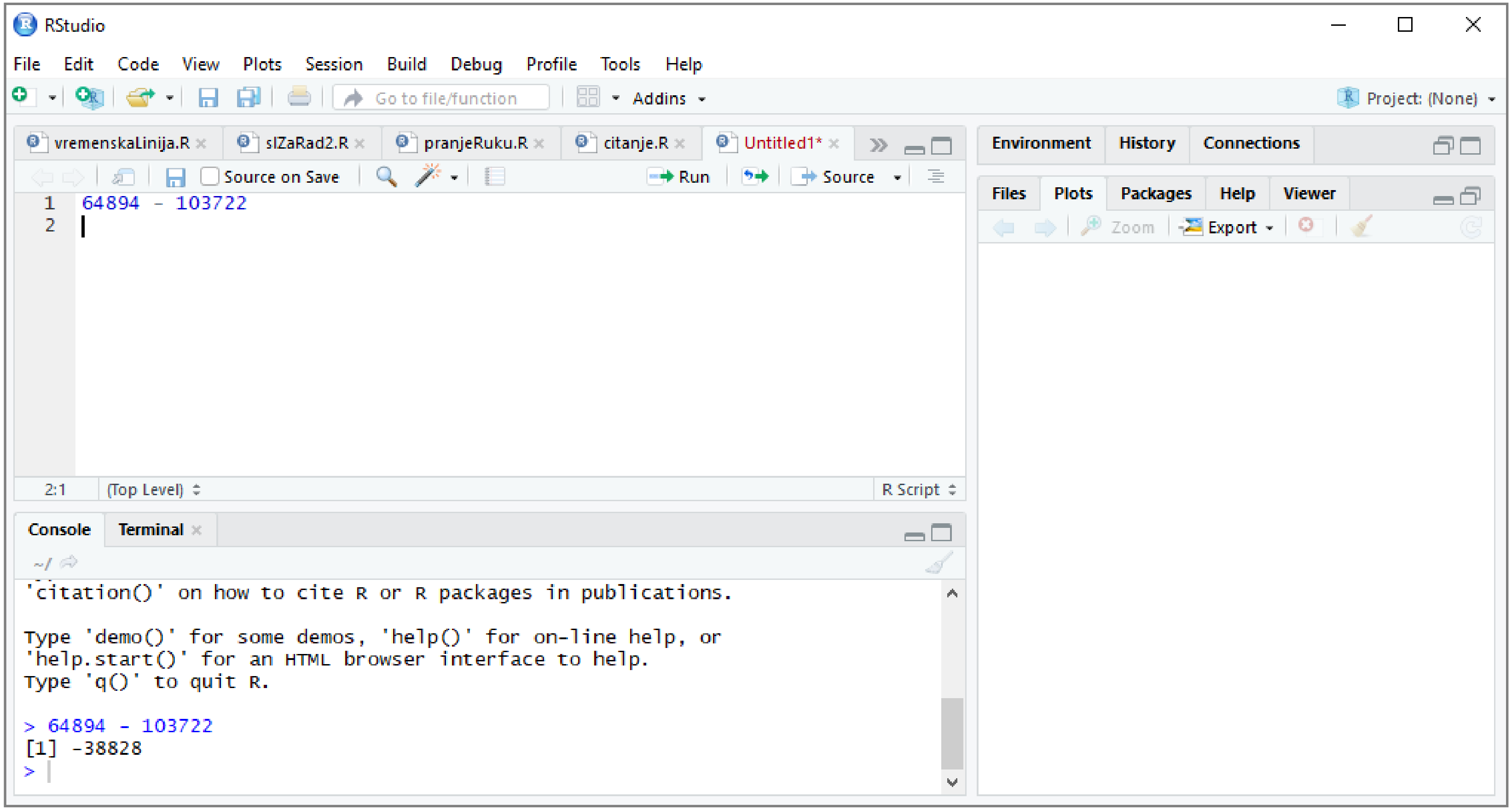
R Studio IDE



```r
install.packages("rafalib")
library(rafalib)

install.packages("swirl")
library(swirl)

```

```
R version 3.2.5 (2016-04-14) -- "Very, Very Secure Dishes"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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

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

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

R studio as calculator



```
64894 - 103722
```

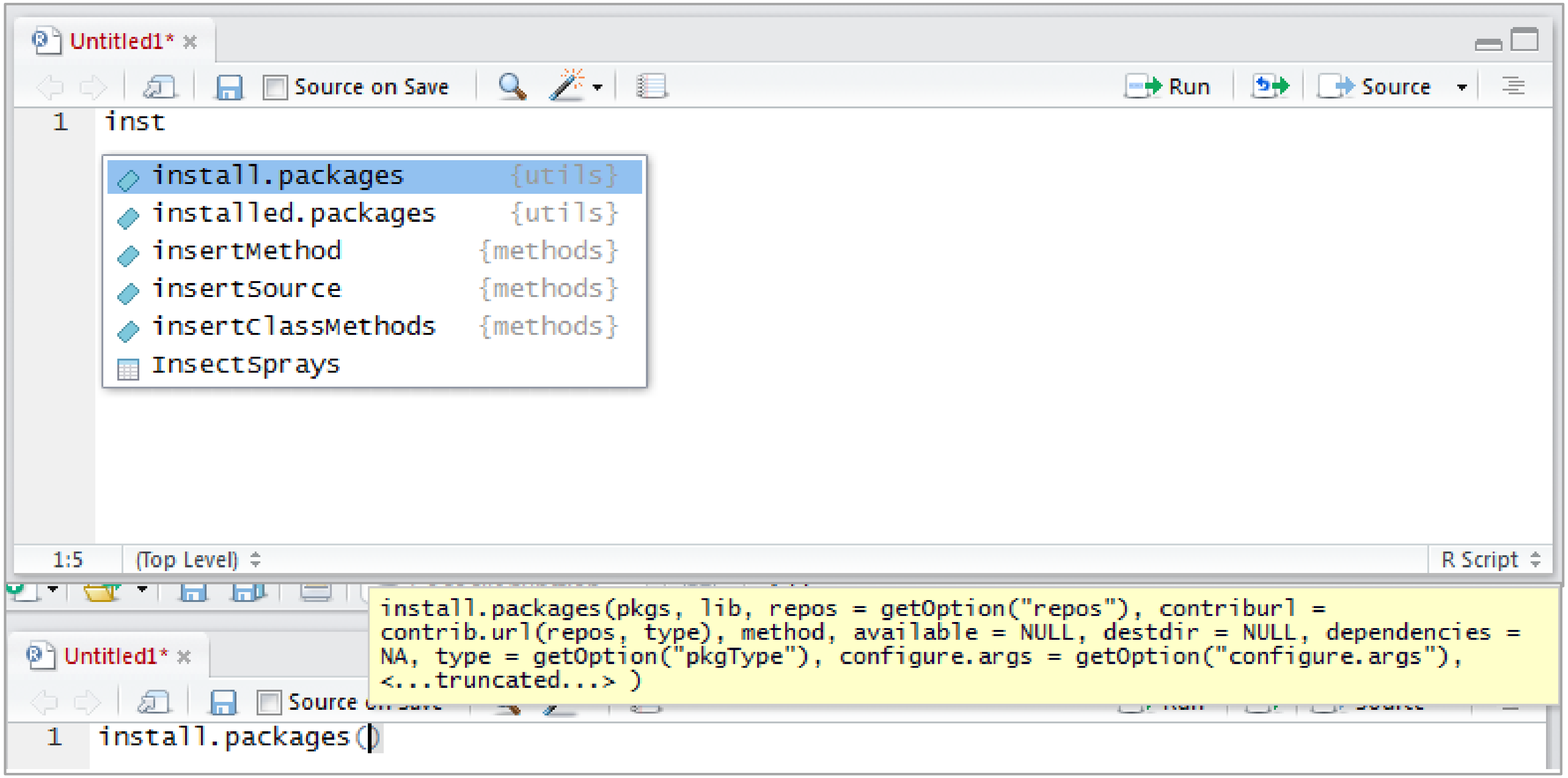
```
'citation()' on how to cite R or R packages in publications.

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

> 64894 - 103722
[1] -38828
>
```

- Subtract/add/multiply… play with numbers… R can be used as calculator.
- Presented numbers are related to the vital statistics. More at: http://www.stat.gov.rs/en-US/oblasti/stanovnistvo/rodjeni-i-umrli, [Online], Assessed October 17, 2018.
- Do it in R script too! Use Ctrl-ENTER to evaluate the expression.

R packages

- R Studio is the most popular IDE for R programming.
- Base R has relatively small number of packages… so you need to install them.
- You can use *install.packages* function.
- Type in console of R Studio *install.packages*… I am spoiled with auto complete option and you should be too.
- By typing the package name, you are connected to the CRAN, and package is being downloaded and installed on your computer.
- To use the functionality of the installed package inside your R script you should use *library()* function (without quotation).

Install packages



```
1  inst
```

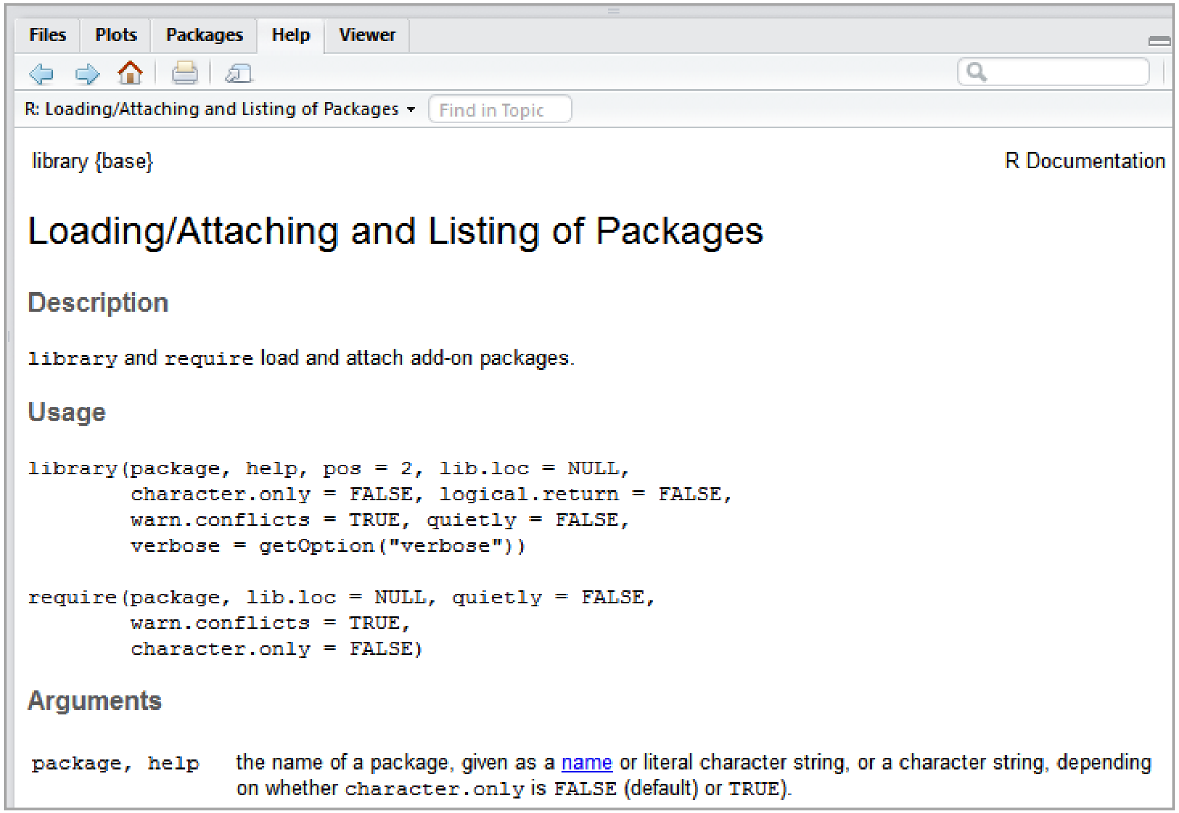
```
install.packages        {utils}
installed.packages      {utils}
insertMethod          {methods}
insertSource          {methods}
insertClassMethods    {methods}
InsectSprays
```

```
install.packages(pkgs, lib, repos = getOption("repos"), contriburl =
contrib.url(repos, type), method, available = NULL, destdir = NULL, dependencies =
NA, type = getOption("pkgType"), configure.args = getOption("configure.args"),
<...truncated...> )
```

```
1  install.packages()
```

- Let’s install some packages: type “ggplot2”, “readODS”, and then “dplyr”.
- Useful link for installation instructions: https://github.com/genomicsclass/windows, [Online], Assessed October 17, 2018.

Help (?)



library {base}   R Documentation

Loading/Attaching and Listing of Packages

Description

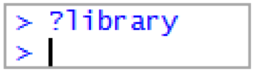
`library` and `require` load and attach add-on packages.

Usage

```
library(package, help, pos = 2, lib.loc = NULL,
        character.only = FALSE, logical.return = FALSE,
        warn.conflicts = TRUE, quietly = FALSE,
        verbose = getOption("verbose"))

require(package, lib.loc = NULL, quietly = FALSE,
        warn.conflicts = TRUE,
        character.only = FALSE)
```

Arguments

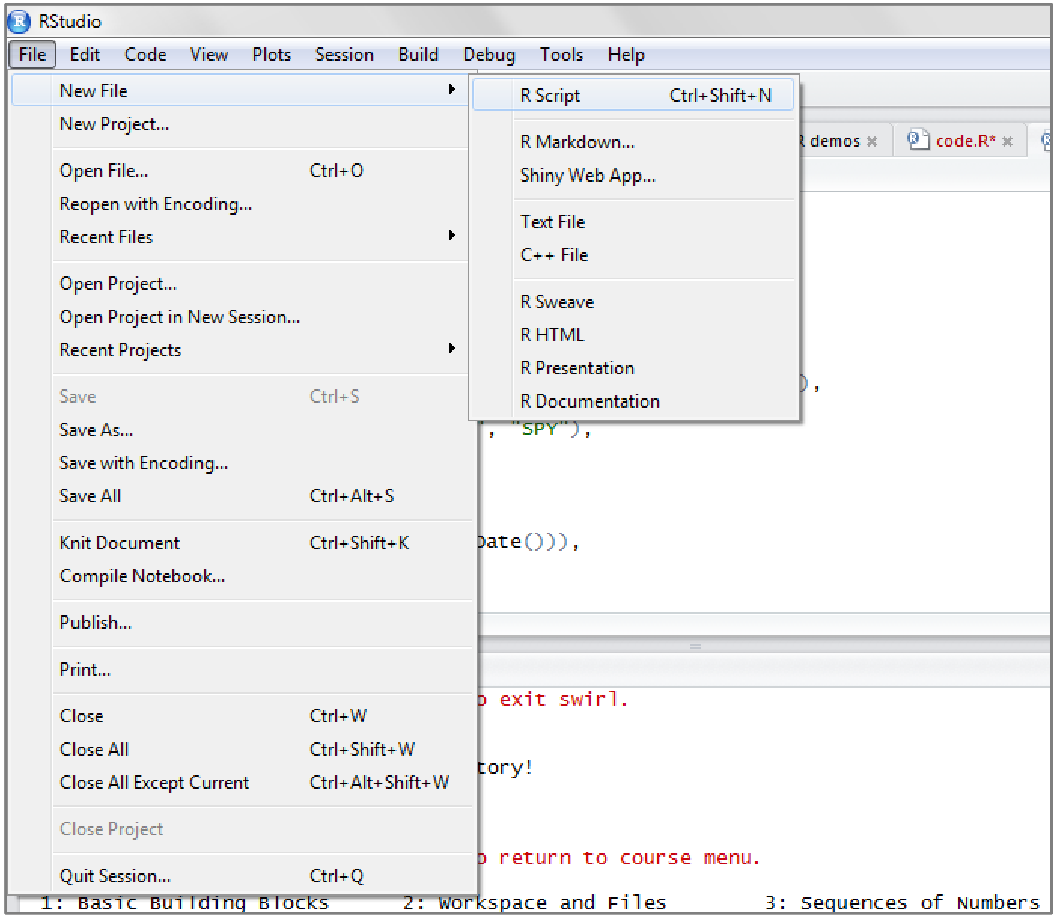
`package, help` the name of a package, given as a name or literal character string, or a character string, depending on whether `character.only` is `FALSE` (default) or `TRUE`.



```
> ?library
> |
```

- Just type *?library* and see what happens.
- Check Help tab for R documentation and use options: Next topic, Previous topic, Show R help, Print topic, Show in new window.
- Use search (Find in Topic and Find in documentation).
- Use Internet!

R Studio – new R script



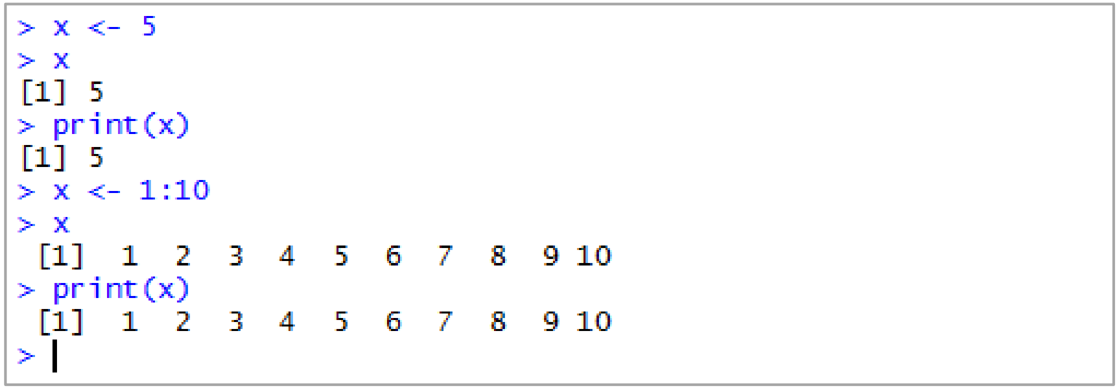
```
),
, "SPY"),
Date())),
```

```
o exit swirl.
tory!
o return to course menu.
 1: Basic Building Blocks      2: Workspace and Files        3: Sequences of Numbers
```

- File / New File / R Script
- It has “surprisingly” *.R* extension.

LET THE CODING BEGIN

Data entry



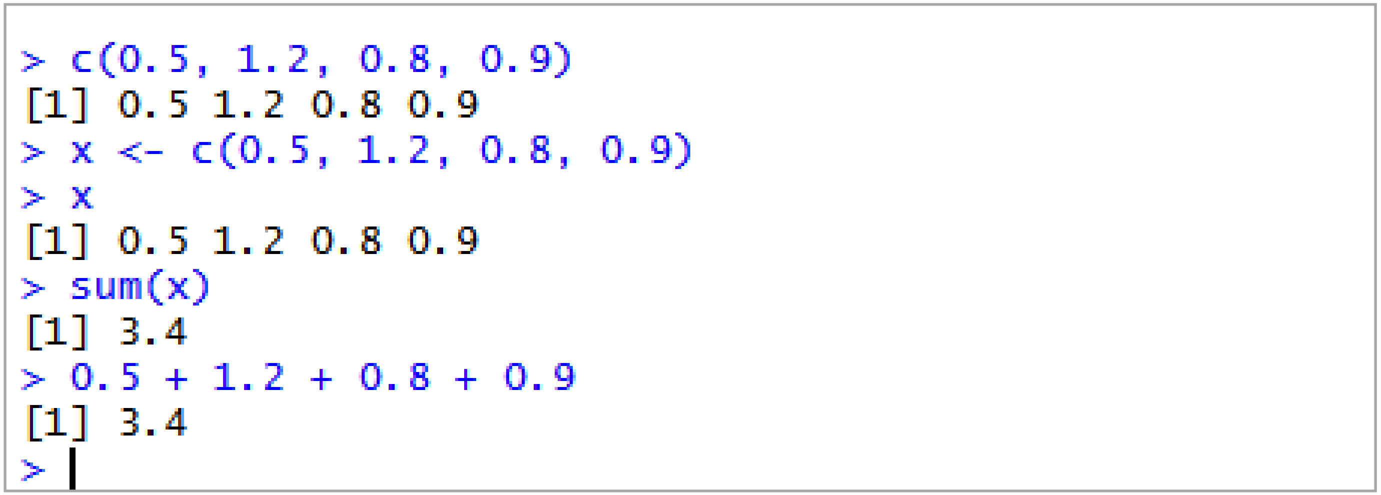
```
> x <- 5
> x
[1] 5
> print(x)
[1] 5
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> print(x)
 [1]  1  2  3  4  5  6  7  8  9 10
> 
```

- Use # hashtag for comments in your code.
- Use explicit and implicit print.
- Assignment operator is used in R to assign value to variable.

Data types

- Data types in R are:
  - Atomic class (there are 5): numeric, logic, character, integer, and complex;
  - arrays, lists;
  - factors;
  - missing values;
  - data frames and matrices.
- All R objects have attributes.
- Some attributes (*e.g.* dimension) can change object's property.

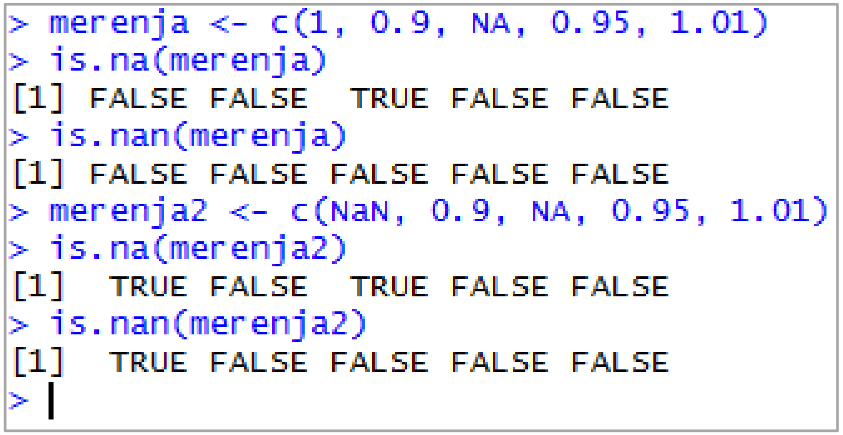
Create arrays



```
> c(0.5, 1.2, 0.8, 0.9)
[1] 0.5 1.2 0.8 0.9
> x <- c(0.5, 1.2, 0.8, 0.9)
> x
[1] 0.5 1.2 0.8 0.9
> sum(x)
[1] 3.4
> 0.5 + 1.2 + 0.8 + 0.9
[1] 3.4
> 
```

- One of the most commonly used functions in R is *c() i.e.* combine.
- Arrays can contain elements of the same data type.
- Lists can contain various data types.

Missing values



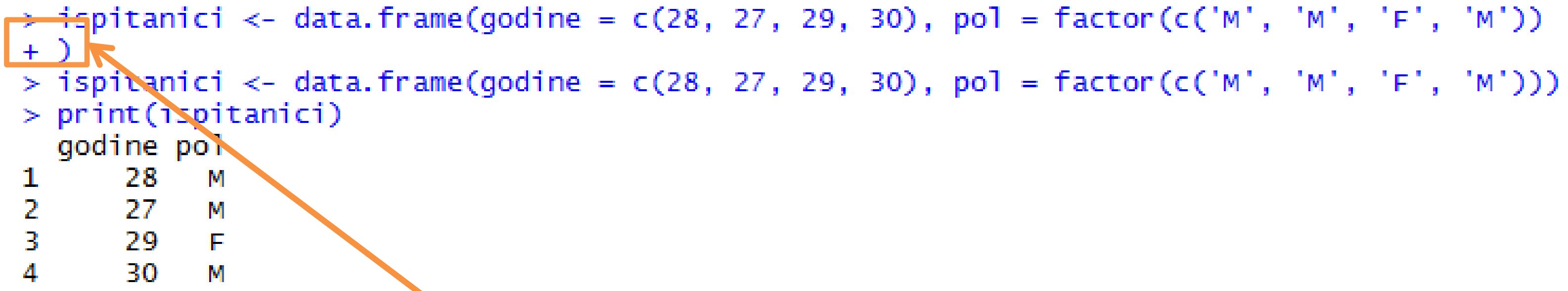
```
> merenja <- c(1, 0.9, NA, 0.95, 1.01)
> is.na(merenja)
[1] FALSE FALSE  TRUE FALSE FALSE
> is.nan(merenja)
[1] FALSE FALSE FALSE FALSE FALSE
> merenja2 <- c(NaN, 0.9, NA, 0.95, 1.01)
> is.na(merenja2)
[1]  TRUE FALSE  TRUE FALSE FALSE
> is.nan(merenja2)
[1]  TRUE FALSE FALSE FALSE FALSE
> |
```

- Missing values can be NA or NaN.
- Functions for testing missing values are *is.na()* and *is.nan()* and their results is logical Boolean result.
- How many NAs are there in the array *merenja*?

Data frame data type

- For data in a form of tables data frames are used.
- Specifically designed dplyr package provides many useful functions for data frame manipulations (https://github.com/hadley/dplyr).
- Data frame can contain various types of atomic data types.
- Data science uses data frames extensively.

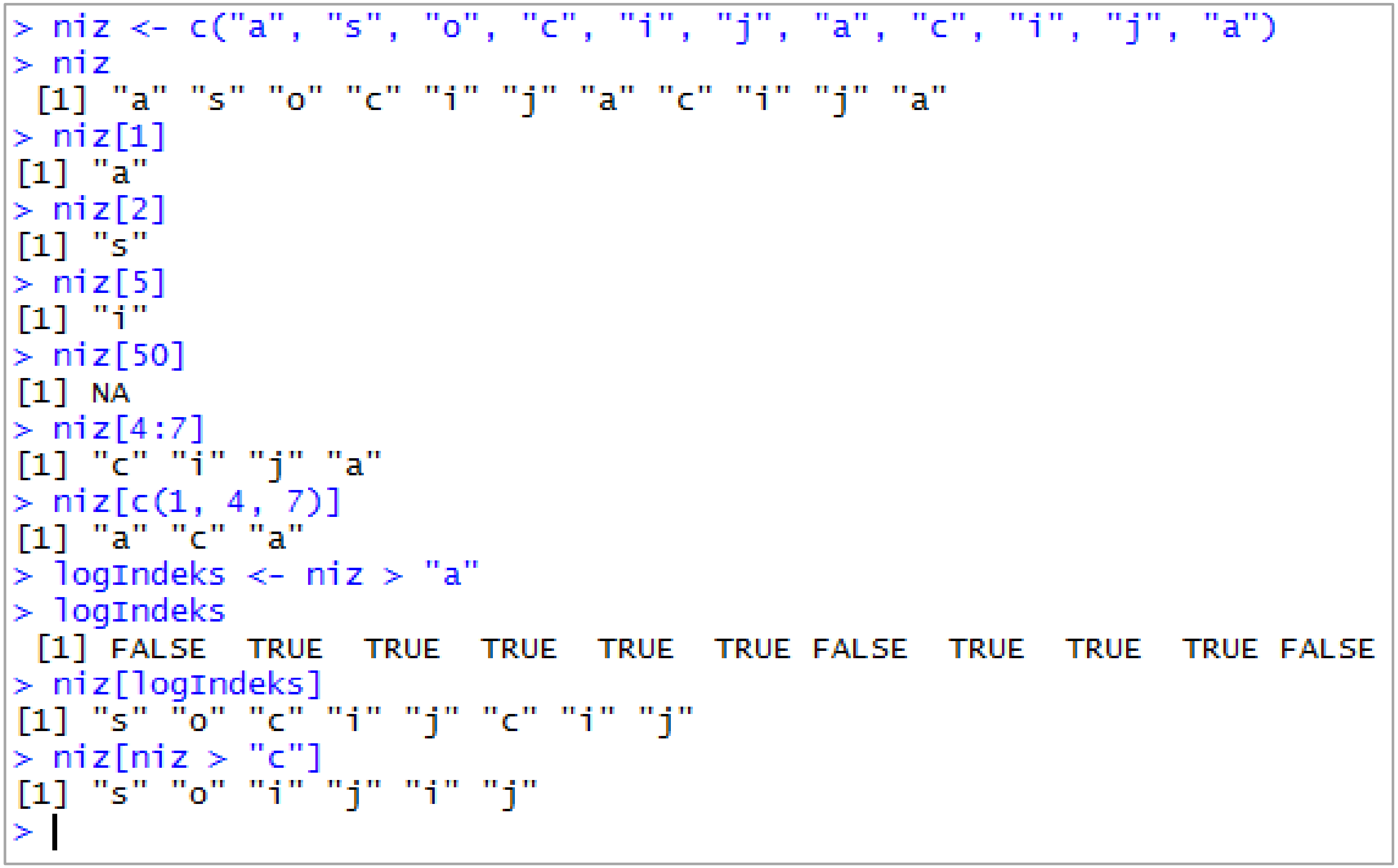
Let’s create simple data frame



```
> ispitanici <- data.frame(godine = c(28, 27, 29, 30), pol = factor(c('M', 'M', 'F', 'M'))
+ )
> ispitanici <- data.frame(godine = c(28, 27, 29, 30), pol = factor(c('M', 'M', 'F', 'M')))
> print(ispitanici)
  godine pol
1     28   M
2     27   M
3     29   F
4     30   M
```

What is “+” used for?

Subsetting data



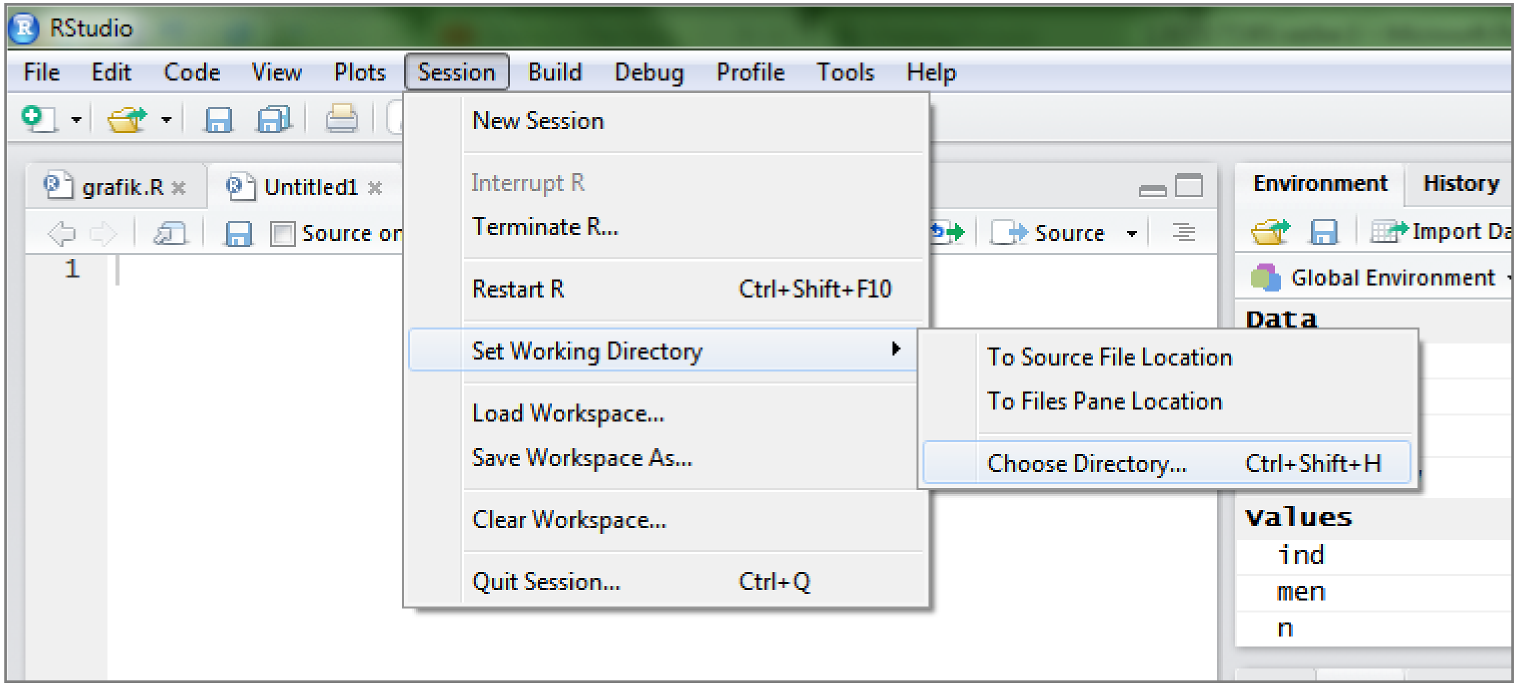
```
> niz <- c("a", "s", "o", "c", "i", "j", "a", "c", "i", "j", "a")
> niz
 [1] "a" "s" "o" "c" "i" "j" "a" "c" "i" "j" "a"
> niz[1]
[1] "a"
> niz[2]
[1] "s"
> niz[5]
[1] "i"
> niz[50]
[1] NA
> niz[4:7]
[1] "c" "i" "j" "a"
> niz[c(1, 4, 7)]
[1] "a" "c" "a"
> logIndeks <- niz > "a"
> logIndeks
 [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
> niz[logIndeks]
[1] "s" "o" "c" "i" "j" "c" "i" "j"
> niz[niz > "c"]
[1] "s" "o" "i" "j" "i" "j"
> 
```

- Operators: “[“, “[[“, and “$” (most commonly used!!!).
- Indexes can be both numerical and logical.
- Apply the operator “$” to previously created data frame.

Reading data from file

- *read.table()*, *read.csv()* – have as a result data frame. They are most commonly used to read table data from files.
- Input of *read.table():*
  - *file*,
  - *header*: T/F vrednost,
  - *sep*: separator for columns,
  - *colClasses*: data type,
  - ..
- Input parameters of *read.csv()* can have different default values than for *read.table()*, so use help extensively.
- There are other functions, but will not be used here (*readLines(), writeLines()*, *source()*, *dump()*, *dget()*, *dput()*, *load()*, *save()*, *unserialize()*, *serialize()*).

R Studio – working directory



- Setting Working Directory will save your time in reading data.
- You can also define Projects, but that's for more advanced users.
- Use Session / Set Working Directory / Choose Directory or Ctrl+Shift+H or *setwd()* function.

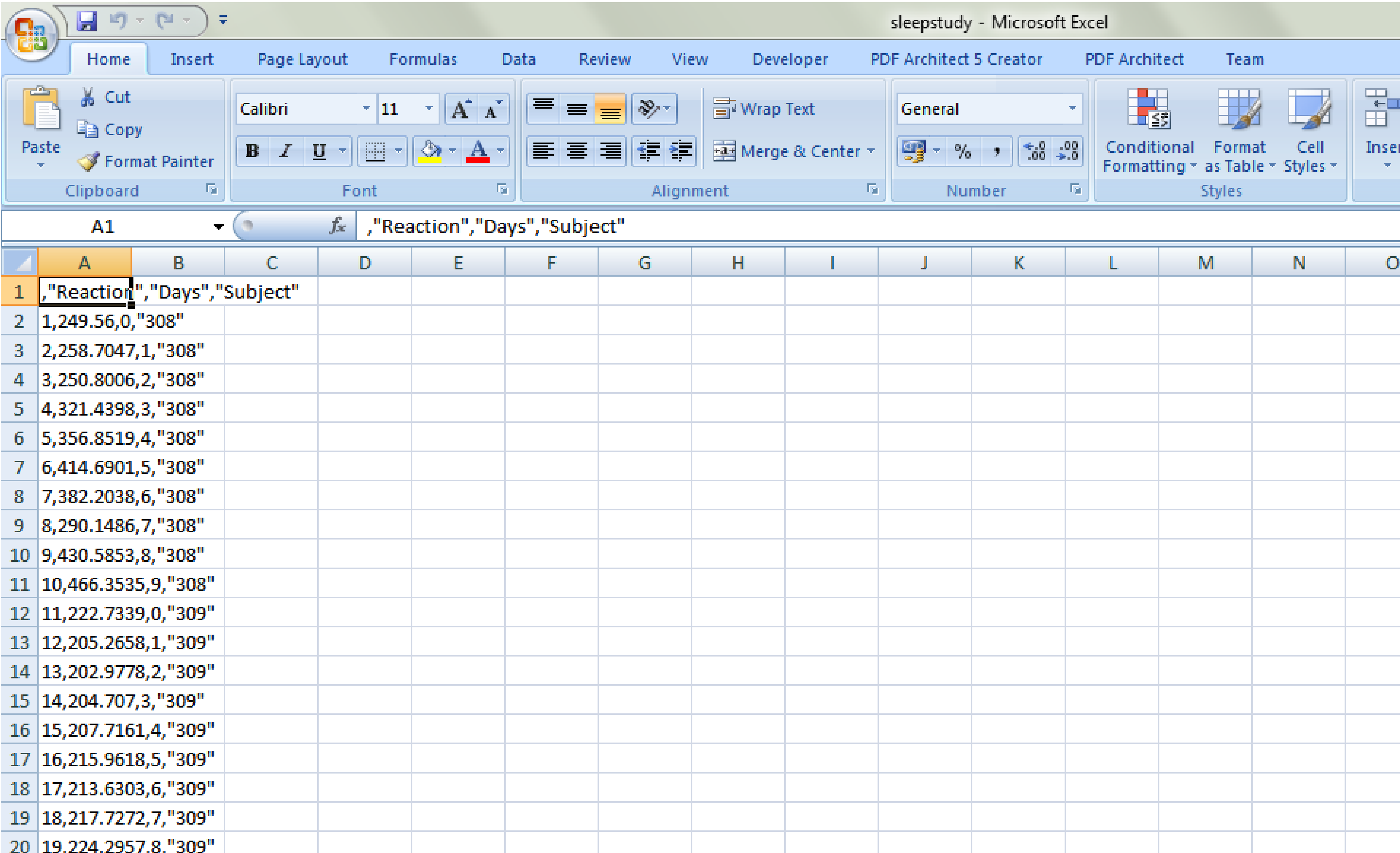
Case study sleep study

- Data from “Reaction times in a sleep deprivation study” study are available at: https://vincentarelbundock.github.io/Rdatasets/datasets.html, [Online], Assessed October 18, 2018. Download “sleepstudy.csv”.
  - The detailed description can be found at: https://vincentarelbundock.github.io/Rdatasets/doc/lme4/sleepstudy.html.
  - Results are published in: Belenky, Gregory, et al. "Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study." *Journal of sleep research* 12.1 (2003): 1-12, http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2869.2003.00337.x/full.

Sleep study

- “On day 0 the subjects had their normal amount of sleep. Starting that night they were restricted to 3 hours of sleep per night. The observations represent the average reaction time on a series of tests given each day to each subject. “
- Data frame has 180 observations with 3 variables:
  - ReactionAverage reaction time (ms)
  - DaysNumber of days of sleep deprivation
  - Subject number on which the observation was made.
- For assessment of sleep restriction PVT (Psychomotor Vigilance Task Performanse) was used.

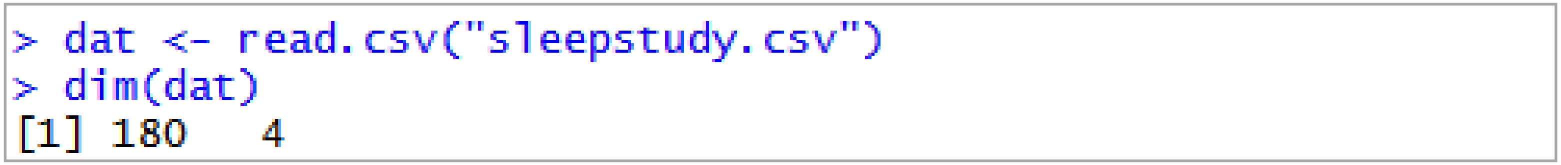
Data



A1 ,"Reaction","Days","Subject"

	A
1	,"Reaction","Days","Subject"
2	1,249.56,0,"308"
3	2,258.7047,1,"308"
4	3,250.8006,2,"308"
5	4,321.4398,3,"308"
6	5,356.8519,4,"308"
7	6,414.6901,5,"308"
8	7,382.2038,6,"308"
9	8,290.1486,7,"308"
10	9,430.5853,8,"308"
11	10,466.3535,9,"308"
12	11,222.7339,0,"309"
13	12,205.2658,1,"309"
14	13,202.9778,2,"309"
15	14,204.707,3,"309"
16	15,207.7161,4,"309"
17	16,215.9618,5,"309"
18	17,213.6303,6,"309"
19	18,217.7272,7,"309"
20	19,224.2957,8,"309"

Read your data



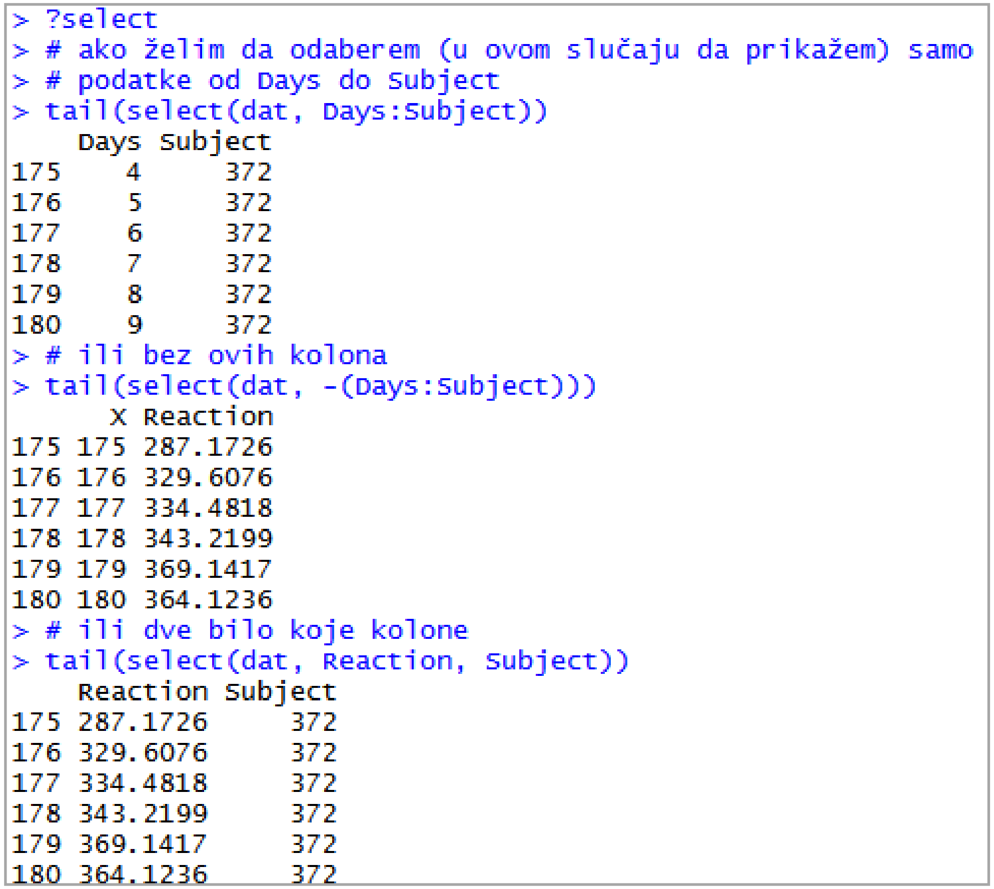
```
> dat <- read.csv("sleepstudy.csv")
> dim(dat)
[1] 180   4
```

- This is just a part of the original study.
- NOTE: Here, only data from subjects with sleep restriction to 3 hours is presented. Reaction is averaged value of PVT tests during one day.
- There are 18 subjects.
- Use functions *head()*, *tail()*, and *unique()*. What are these functions used for?

Let’s manipulate data: dplyr

- Hypothesis for data format, before dplyr (od eng. *deep layer*) application are:
  - there is one observation per row
  - each column represents one variable
  - ...
- These package was introduced by Hadley Wickham (Rstudio, https://cran.r-project.org/web/packages/dplyr/dplyr.pdf).
- Note: There is no novelty in functionality compared to the base R. However, the code is efficient since it is developed in C++ and the functions are simple and intuitive to use.
- Let’s try some of these functions.

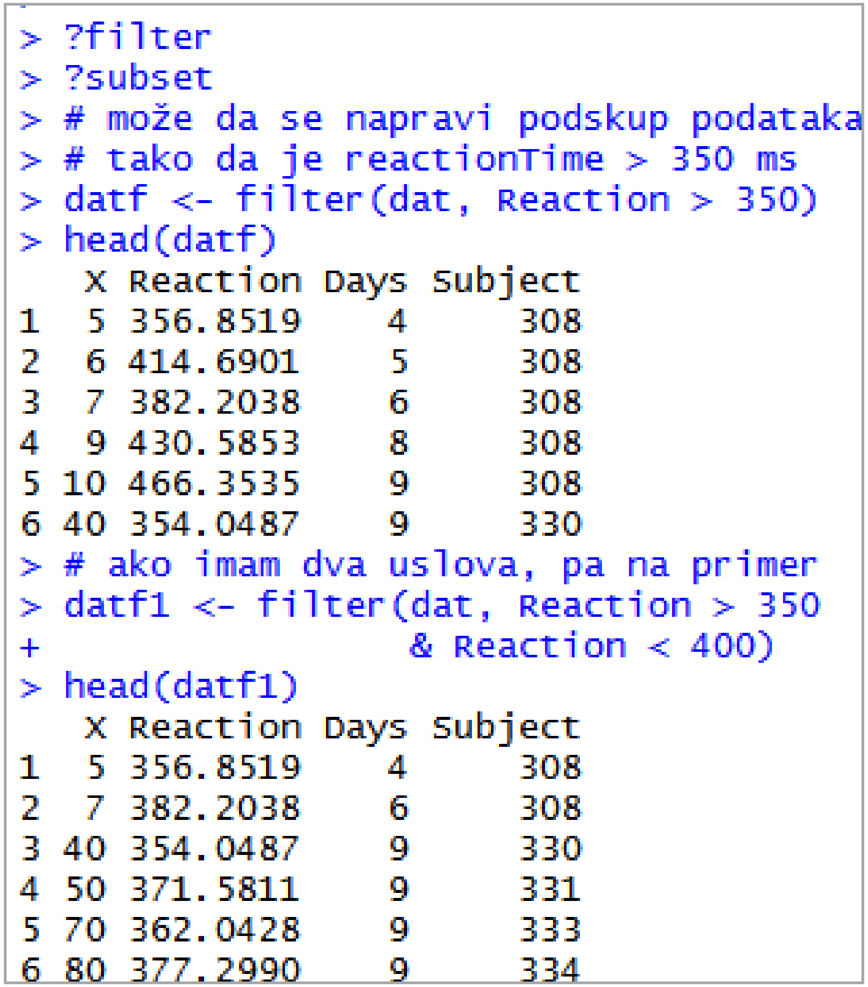
dplyr::select()



```
> ?select
> # ako želim da odaberem (u ovom slučaju da prikažem) samo
> # podatke od Days do Subject
> tail(select(dat, Days:Subject))
    Days Subject
175    4     372
176    5     372
177    6     372
178    7     372
179    8     372
180    9     372
> # ili bez ovih kolona
> tail(select(dat, -(Days:Subject)))
      X Reaction
175 175 287.1726
176 176 329.6076
177 177 334.4818
178 178 343.2199
179 179 369.1417
180 180 364.1236
> # ili dve bilo koje kolone
> tail(select(dat, Reaction, Subject))
    Reaction Subject
175 287.1726     372
176 329.6076     372
177 334.4818     372
178 343.2199     372
179 369.1417     372
180 364.1236     372
```

- Funtion *select()* helps to select data subset for further analysis.
- It can be used to eliminate data subset.

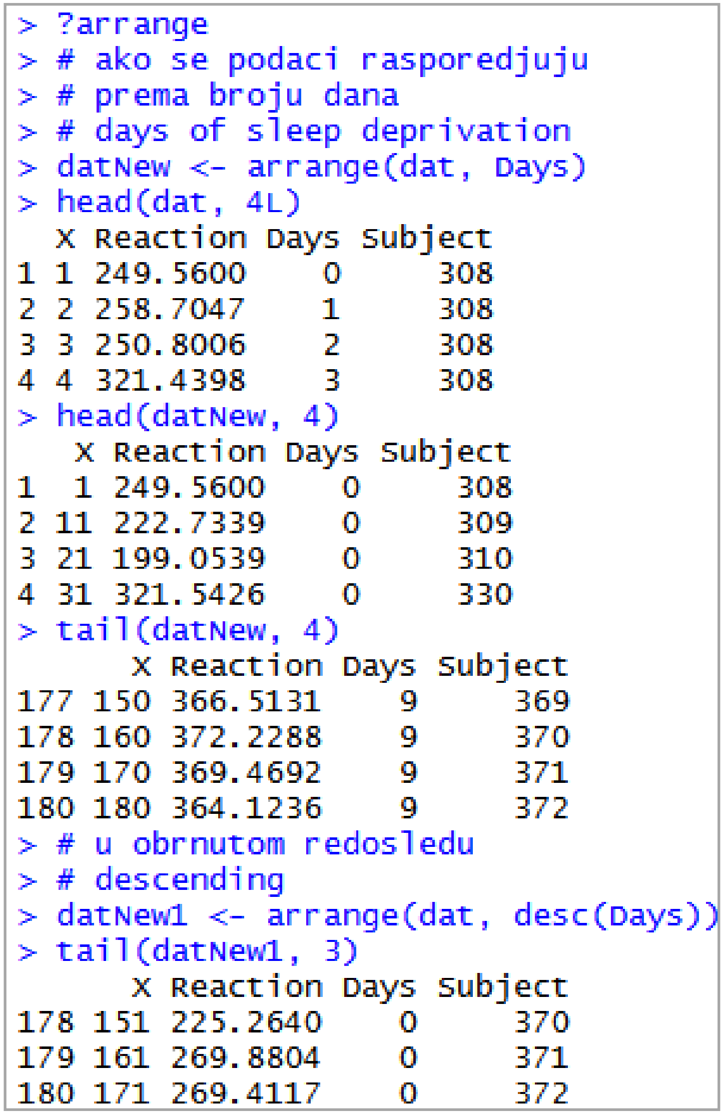
dplyr::filter()



```
> ?filter
> ?subset
> # može da se napravi podskup podataka
> # tako da je reactionTime > 350 ms
> datf <- filter(dat, Reaction > 350)
> head(datf)
   X Reaction Days Subject
1  5 356.8519    4     308
2  6 414.6901    5     308
3  7 382.2038    6     308
4  9 430.5853    8     308
5 10 466.3535    9     308
6 40 354.0487    9     330
> # ako imam dva uslova, pa na primer
> datf1 <- filter(dat, Reaction > 350
+                 & Reaction < 400)
> head(datf1)
   X Reaction Days Subject
1  5 356.8519    4     308
2  7 382.2038    6     308
3 40 354.0487    9     330
4 50 371.5811    9     331
5 70 362.0428    9     333
6 80 377.2990    9     334
```

- Most commonly used dplyr function.
- Logical conditions can be defined within this function.

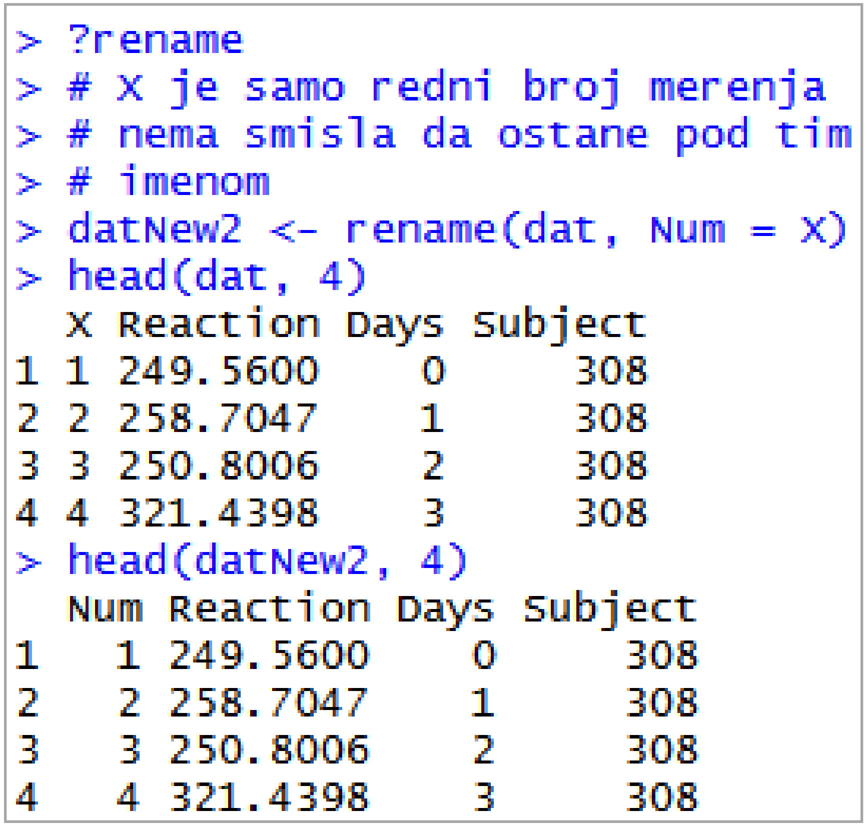
dplyr::arrange()



```
> ?arrange
> # ako se podaci rasporedjuju
> # prema broju dana
> # days of sleep deprivation
> datNew <- arrange(dat, Days)
> head(dat, 4L)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
4 4 321.4398    3     308
> head(datNew, 4)
   X Reaction Days Subject
1  1 249.5600    0     308
2 11 222.7339    0     309
3 21 199.0539    0     310
4 31 321.5426    0     330
> tail(datNew, 4)
      X Reaction Days Subject
177 150 366.5131    9     369
178 160 372.2288    9     370
179 170 369.4692    9     371
180 180 364.1236    9     372
> # u obrnutom redosledu
> # descending
> datNew1 <- arrange(dat, desc(Days))
> tail(datNew1, 3)
      X Reaction Days Subject
178 151 225.2640    0     370
179 161 269.8804    0     371
180 171 269.4117    0     372
```

- Descending and ascending ordering of data frames.
- Can be very useful for comparison.

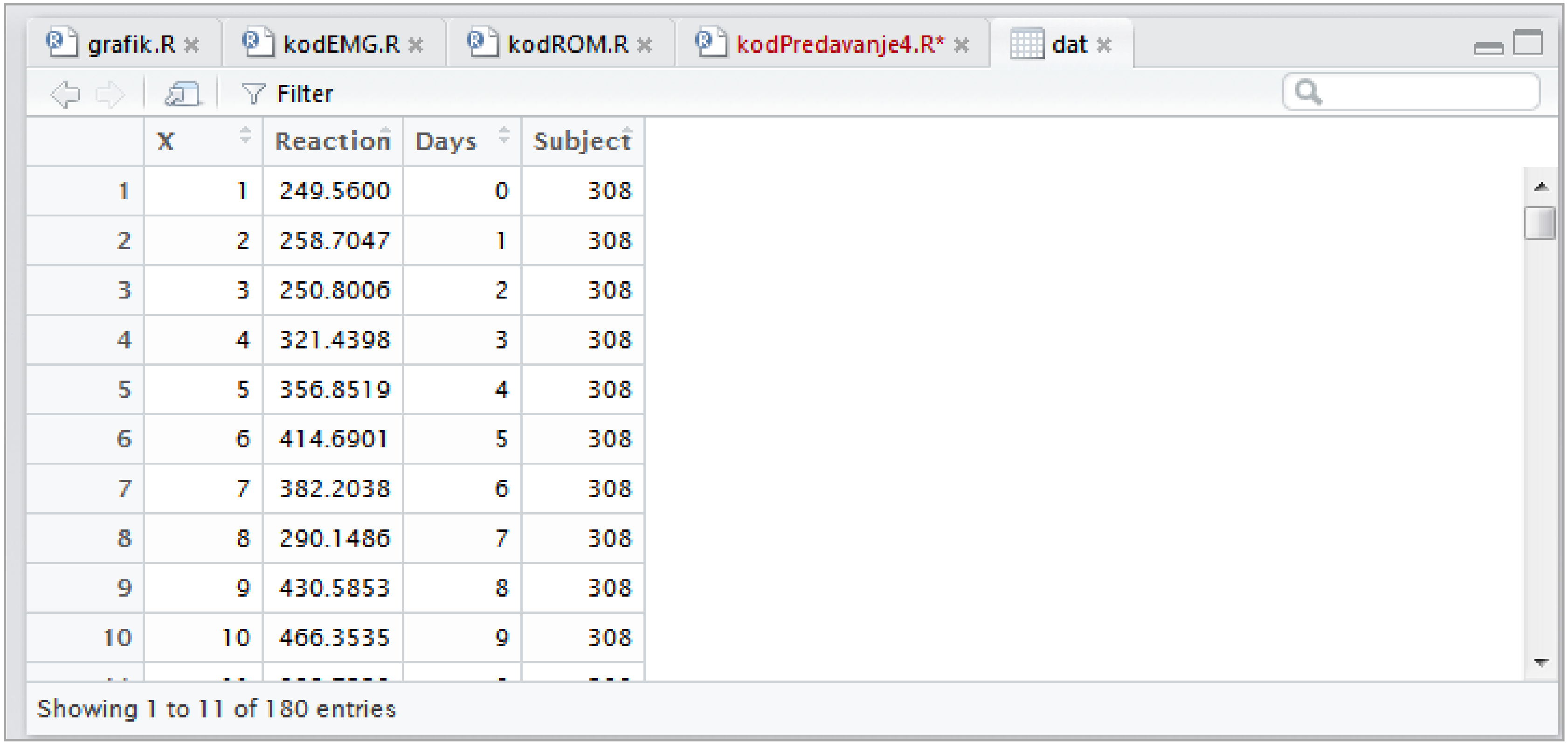
dplyr::rename()



```
> ?rename
> # X je samo redni broj merenja
> # nema smisla da ostane pod tim
> # imenom
> datNew2 <- rename(dat, Num = X)
> head(dat, 4)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
4 4 321.4398    3     308
> head(datNew2, 4)
  Num Reaction Days Subject
1   1 249.5600    0     308
2   2 258.7047    1     308
3   3 250.8006    2     308
4   4 321.4398    3     308
```

- It is recommended to use intuitive and logical names whenever possible.
- Simpler use of “$” operator.
- Simpler data sharing.
- …

View() function

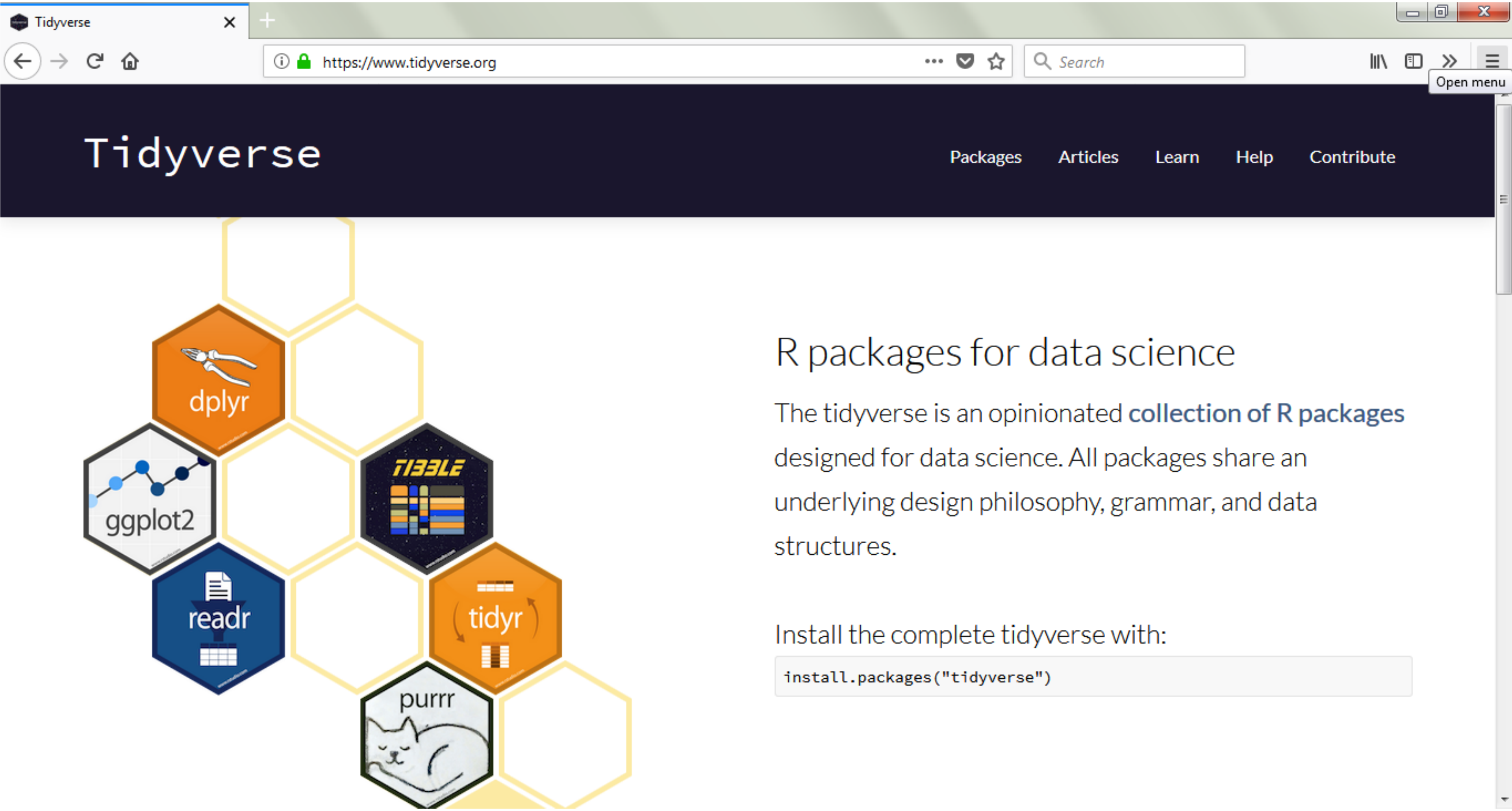


	X	Reaction	Days	Subject
1	1	249.5600	0	308
2	2	258.7047	1	308
3	3	250.8006	2	308
4	4	321.4398	3	308
5	5	356.8519	4	308
6	6	414.6901	5	308
7	7	382.2038	6	308
8	8	290.1486	7	308
9	9	430.5853	8	308
10	10	466.3535	9	308

Showing 1 to 11 of 180 entries

- Enables an easy-to-use examination of data.
- Useful function.

dplyr is just a first step … tidyverse



R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

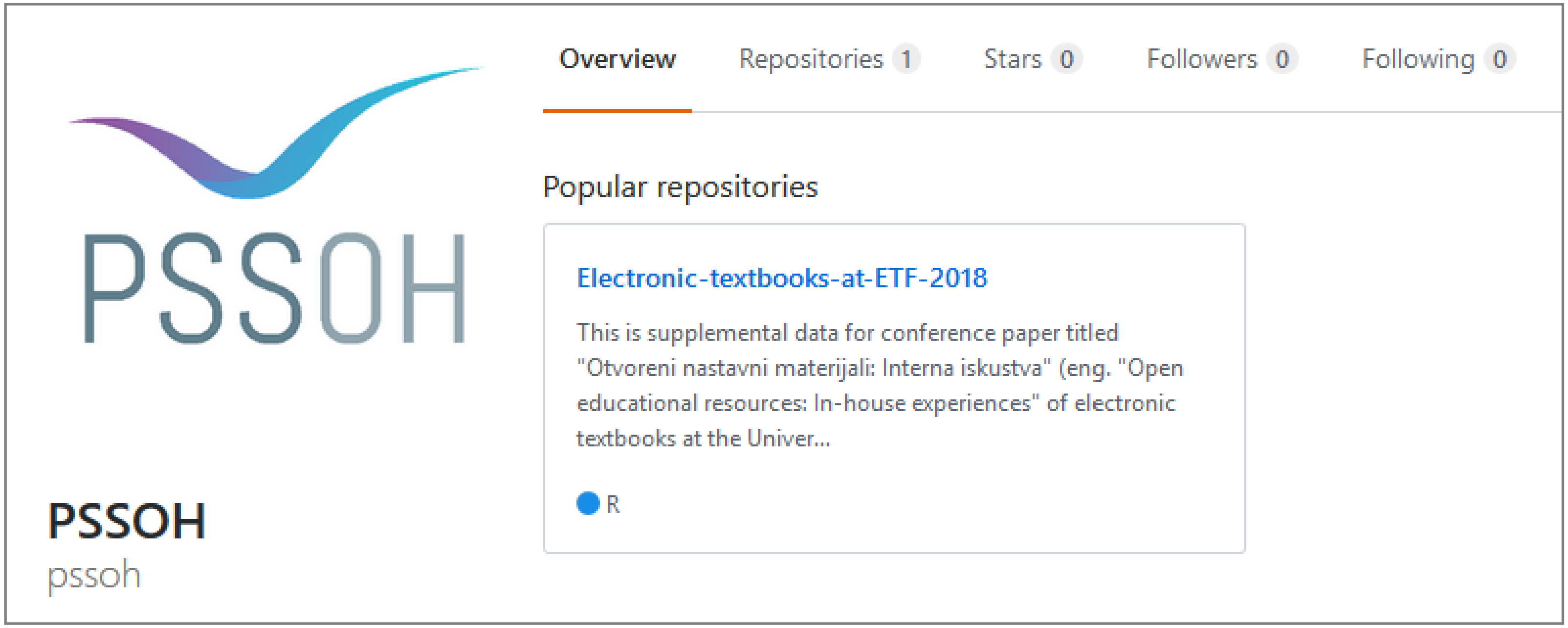
```
install.packages("tidyverse")
```

For homework

- Calculate average Reaction time for all days.
- Present the average with standard deviation.
- Comment results. How does sleep deprivation affects our reaction? Is one day enough to recover from the sleep deprivation?
- Optional: plot error bar

VISUALIZATION

Case study: OER at ETF



Popular repositories

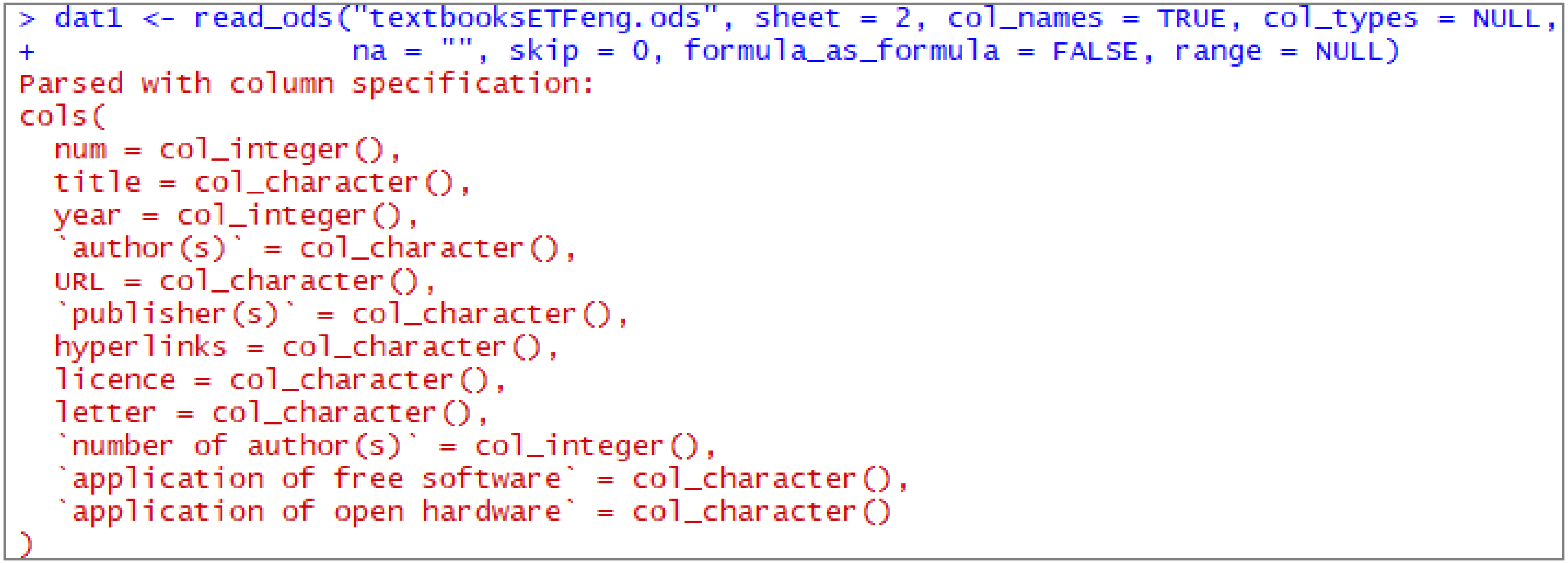
Electronic-textbooks-at-ETF-2018

This is supplemental data for conference paper titled "Otvoreni nastavni materijali: Interna iskustva" (eng. "Open educational resources: In-house experiences" of electronic textbooks at the Univer...

● R

- We will be using data from GitHub page of PSSOH Conference (https://github.com/pssoh/Electronic-textbooks-at-ETF-2018)
- The data are presented in the paper titled "Otvoreni nastavni materijali: Interna iskustva" (eng. "Open educational resources: In-house experiences“.
- The data are stored in *.ods* files (OpenDocument Spreadsheet).

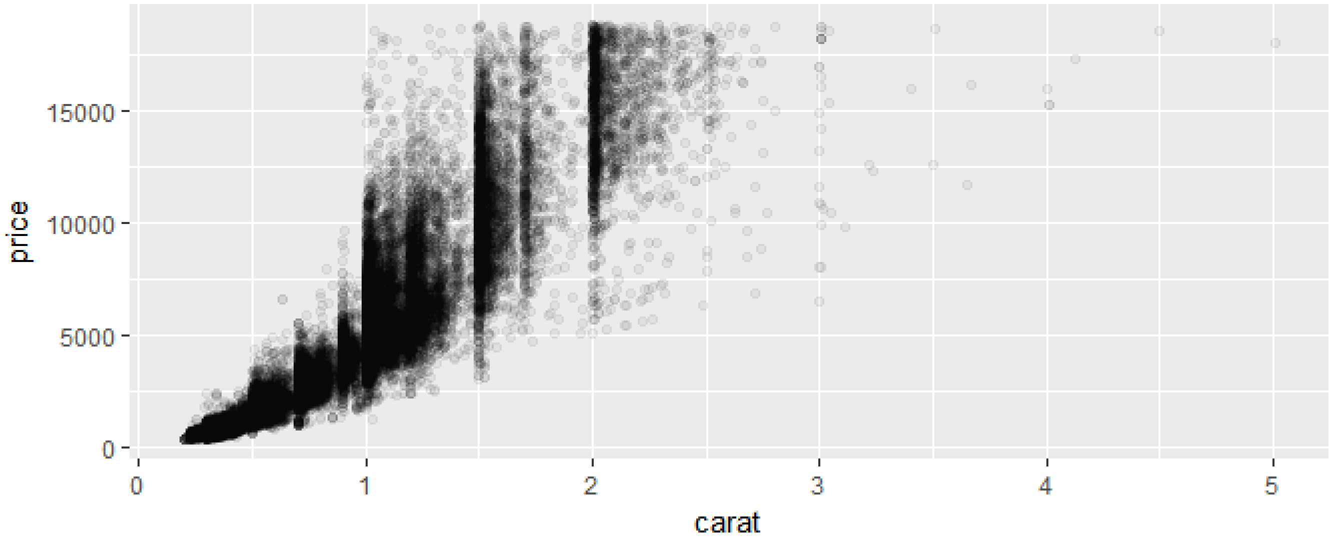
Let’s read the data



```
> dat1 <- read_ods("textbooksETFeng.ods", sheet = 2, col_names = TRUE, col_types = NULL,
+                   na = "", skip = 0, formula_as_formula = FALSE, range = NULL)
Parsed with column specification:
cols(
  num = col_integer(),
  title = col_character(),
  year = col_integer(),
  `author(s)` = col_character(),
  URL = col_character(),
  `publisher(s)` = col_character(),
  hyperlinks = col_character(),
  licence = col_character(),
  letter = col_character(),
  `number of author(s)` = col_integer(),
  `application of free software` = col_character(),
  `application of open hardware` = col_character()
)
```

- I used readODS package and you can too.
- Perform following steps:
  - Install package
  - Download files
  - Set working directory
  - Read files
- Yes, there is a warning, but we will ignore it this time. Use *head()* to check the data frame content.

Before we start plotting… ggplot2

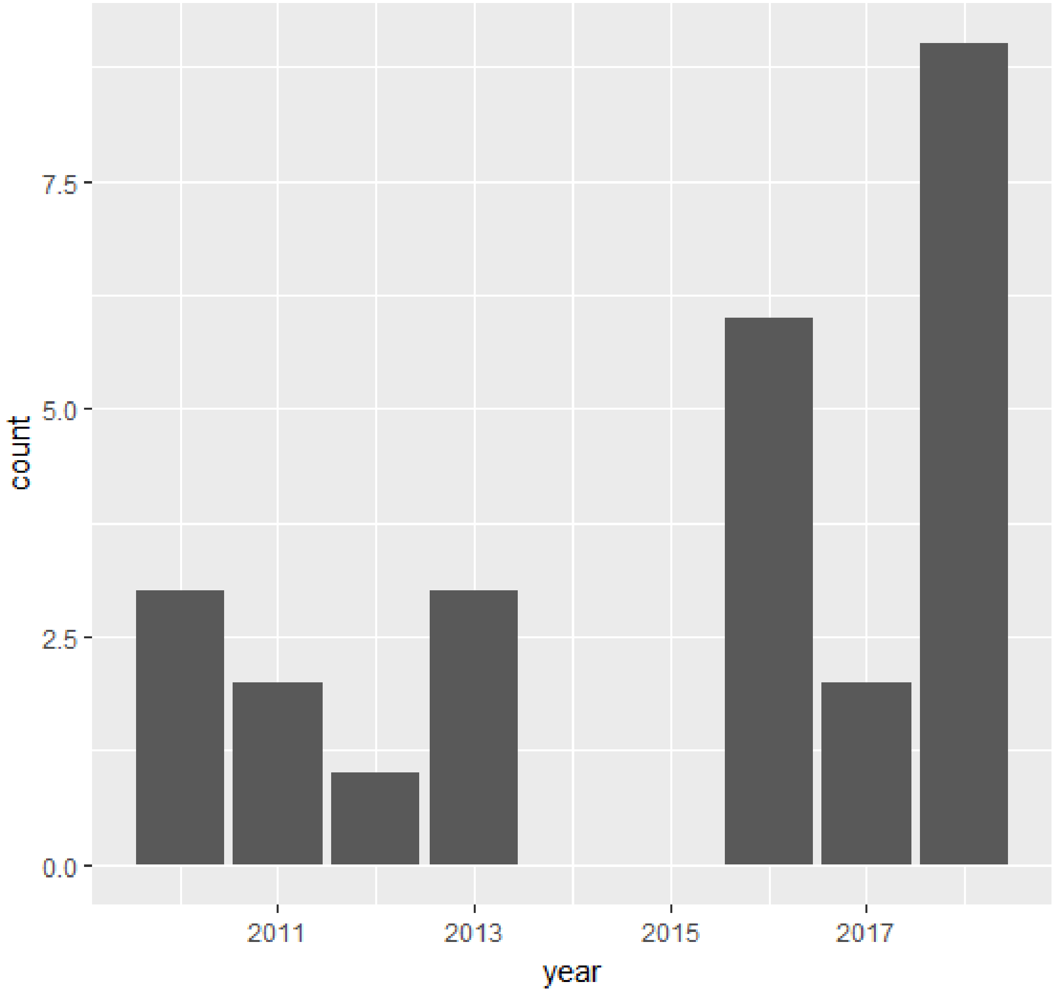


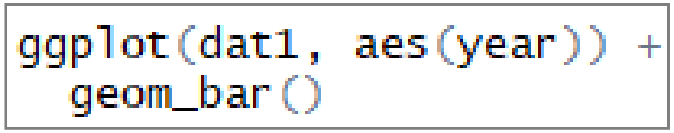
- Sample ggplot2 graph with *example()* function is presented.
- ggplot2 package (http://ggplot2.org/) is both flexible an has high level of abstraction.
- Though it is highly popular, you should not use it for:
  - 3D visualization (use **rgl** instead, https://cran.r-project.org/web/packages/rgl/index.html),
  - graphs from theory of graphs and
  - interactive graphs (use **ggvis** instead, https://cran.r-project.org/web/packages/ggvis/index.html).
- I recommend: http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html.

ggplot2 layering and anatomy

- Some basic graph elements include:
  - data,
  - aesthetic mapping,
  - geometric shapes,
  - statistical transformations,
  - scalling,
  - coordinate system,
  - positioning, and
  - faceting.
- The most important operator is “+”.
- Used with data frames. Usually prepared by dplyr package.
- Matlab and Python were jealous (☺) of ggplot2 and eager to get it, so you can now use ggplot2 functionality with them. Check these GitHub pages:
  - https://github.com/yhat/ggpy i
  - https://github.com/piermorel/gramm.

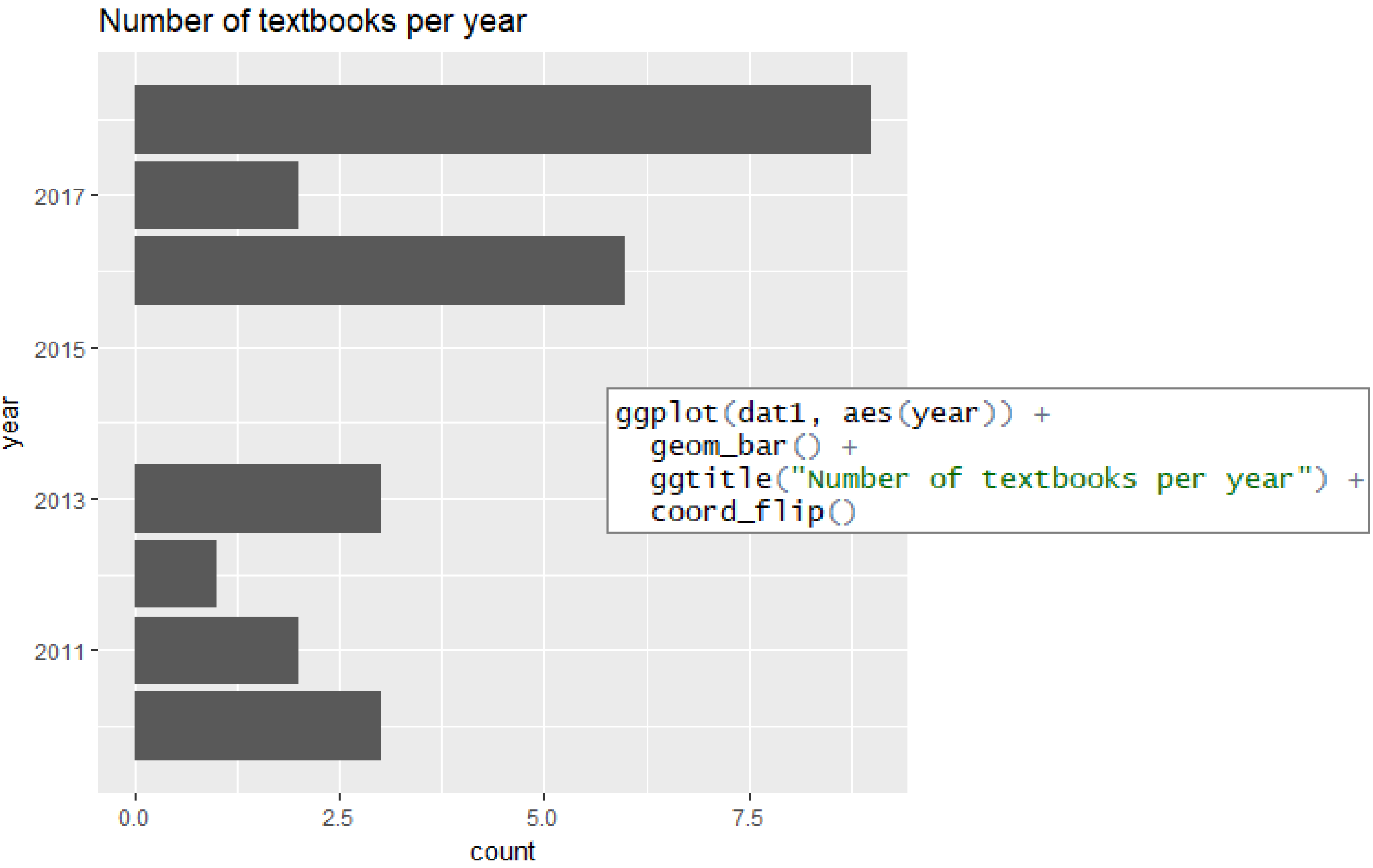
Let’s try it





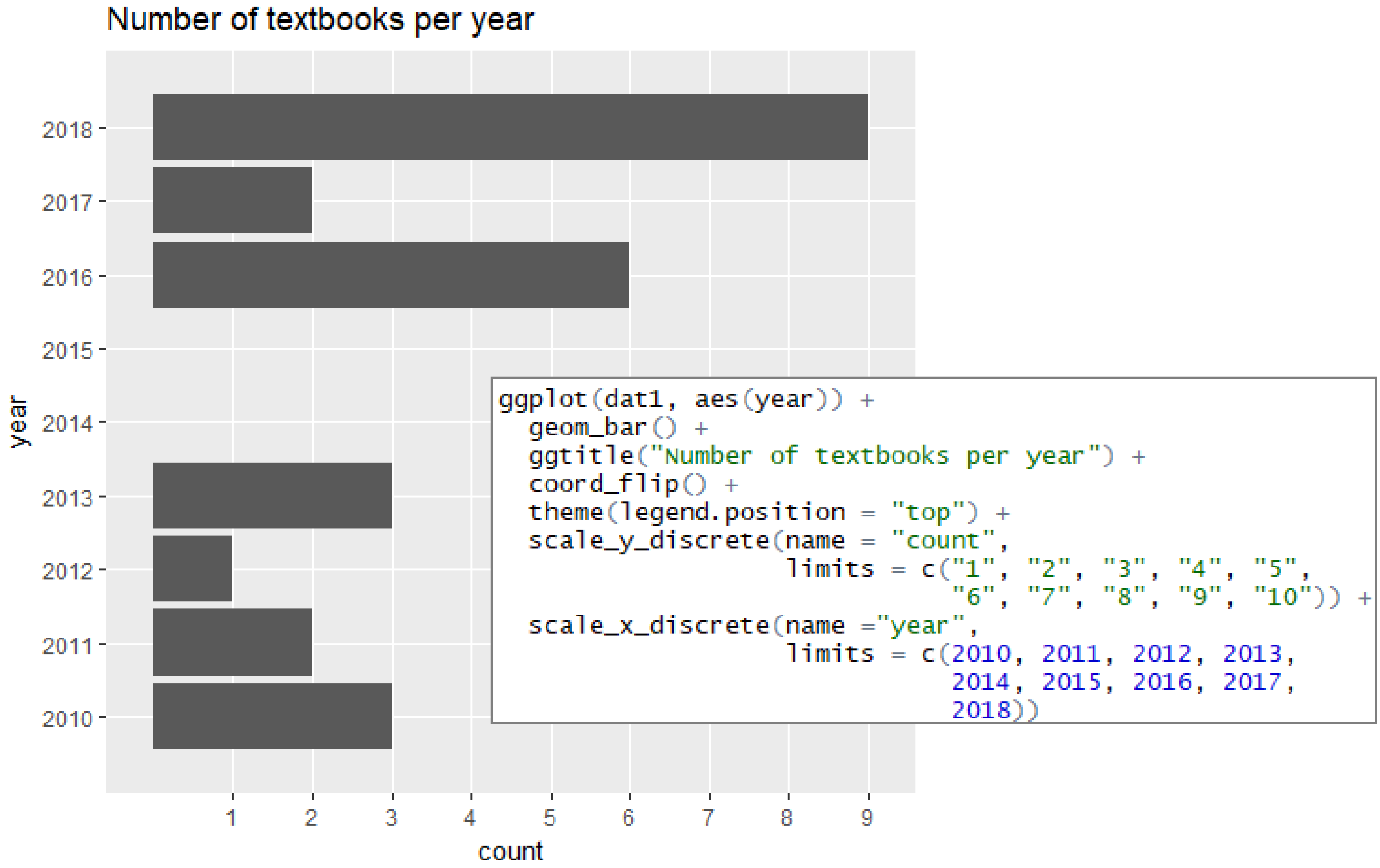
```
ggplot(dat1, aes(year)) +
  geom_bar()
```

I use makeup. Do you?



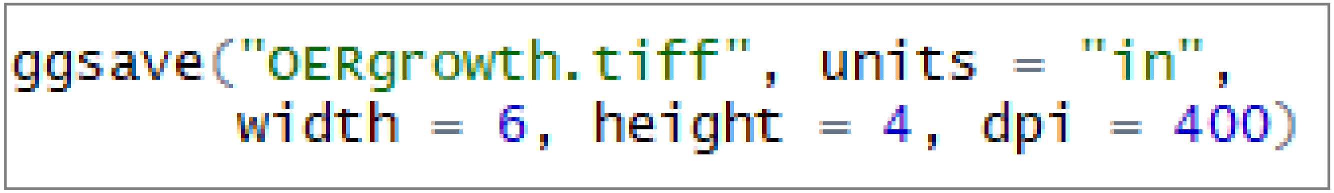
```
ggplot(dat1, aes(year)) +
  geom_bar() +
  ggtitle("Number of textbooks per year") +
  coord_flip()
```

Eyeliner is for special occasion



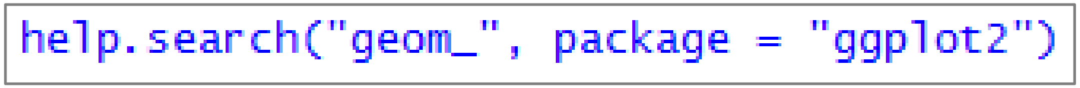
```
ggplot(dat1, aes(year)) +
  geom_bar() +
  ggtitle("Number of textbooks per year") +
  coord_flip() +
  theme(legend.position = "top") +
  scale_y_discrete(name = "count",
                   limits = c("1", "2", "3", "4", "5",
                              "6", "7", "8", "9", "10")) +
  scale_x_discrete(name ="year",
                   limits = c(2010, 2011, 2012, 2013,
                              2014, 2015, 2016, 2017,
                              2018))
```

Save your graphs smart way!



```
ggsave("OERgrowth.tiff", units = "in",
       width = 6, height = 4, dpi = 400)
```

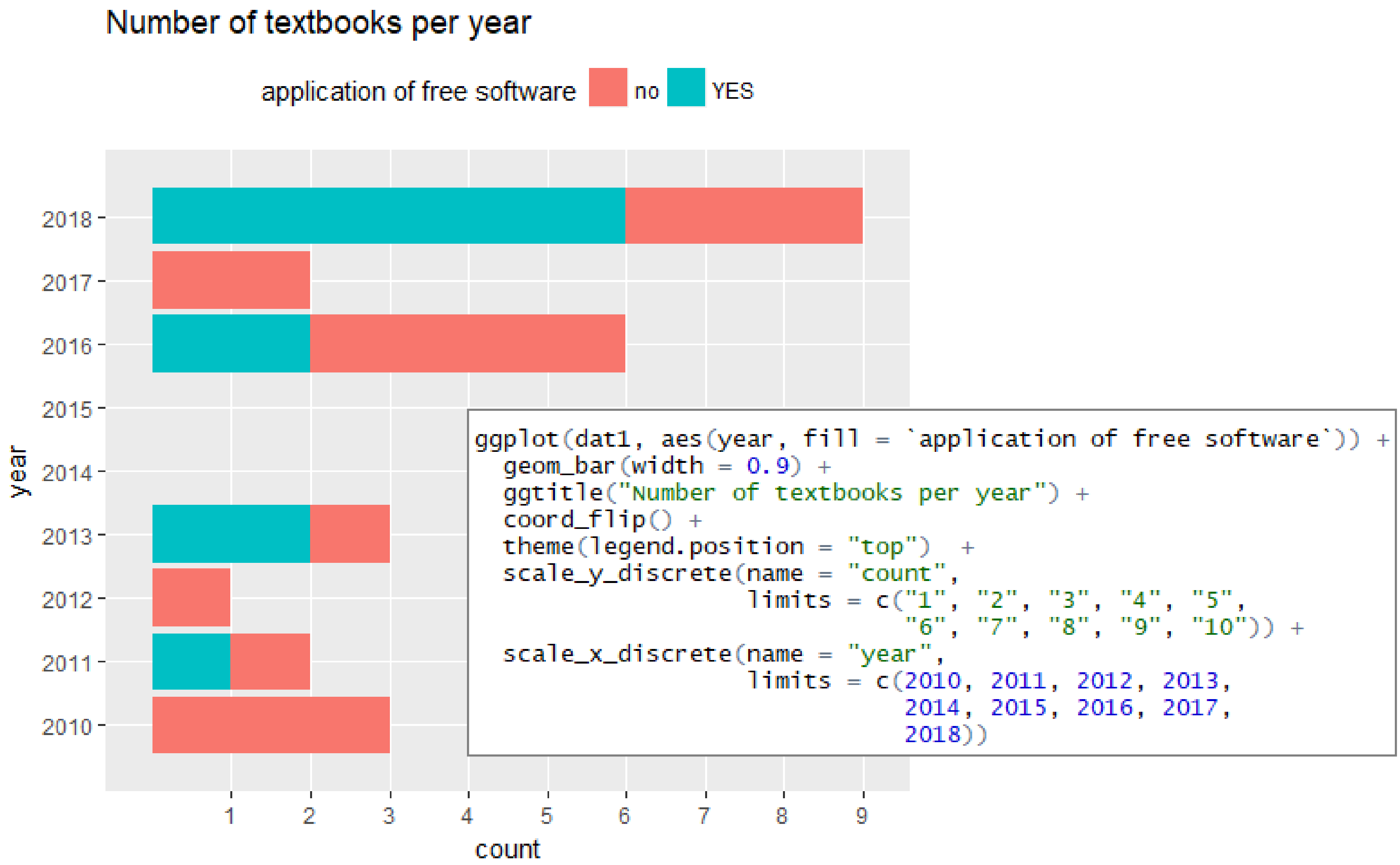
Geometry and aesthetics



```
help.search("geom_", package = "ggplot2")
```

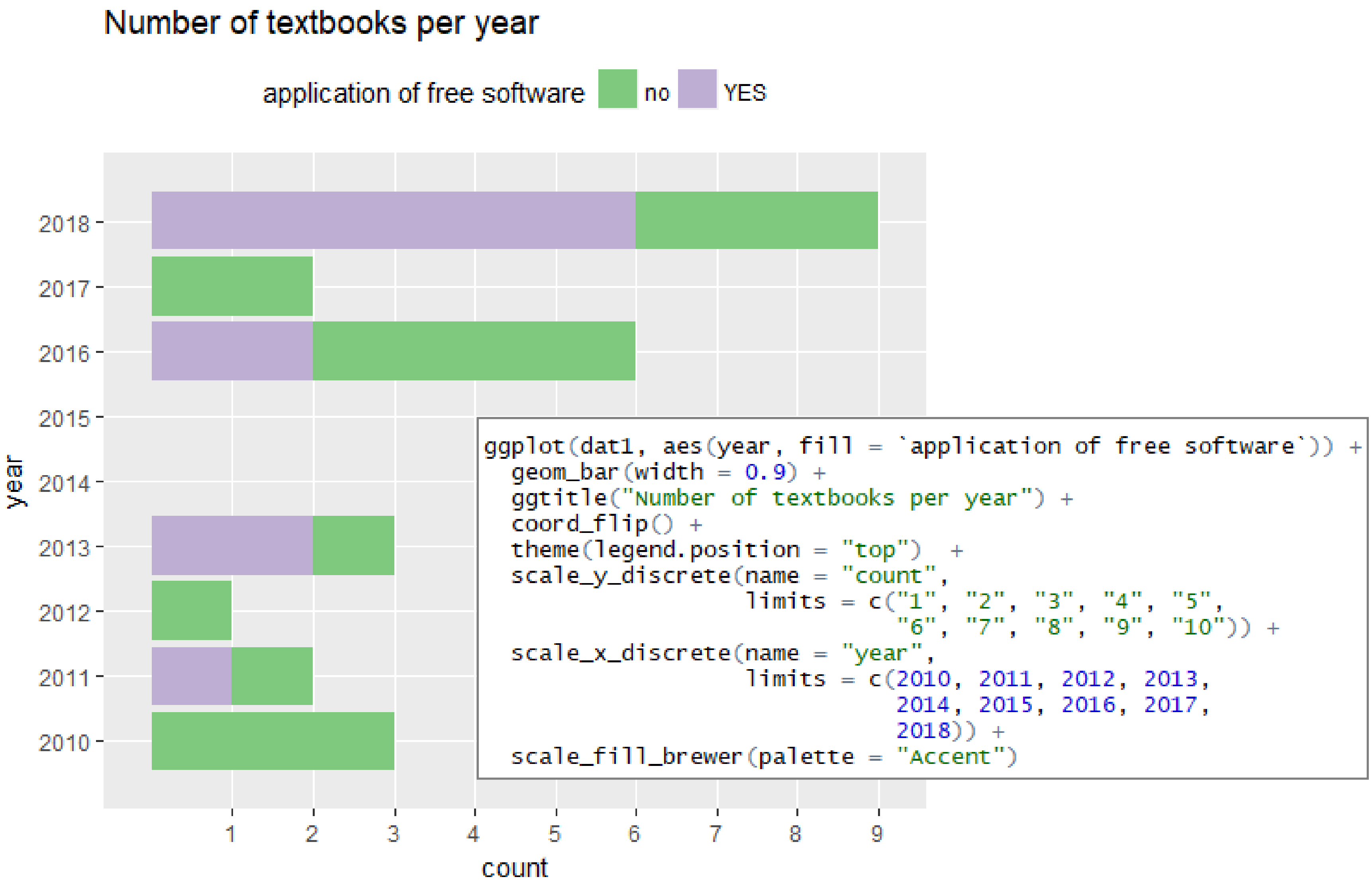
- Use *aes()*:
  - to define position (on abscise and ordinate axis),
  - color (“outside”/“inside” color),
  - shape,
  - line tipe, and
  - size.
- There are many *geom_()* functions that come with unique aesthetics:
  - dots: *geom_point()* – used for scatter plot and dot plot
  - lines: *geom_line()* – for time series and line trends
  - boxplot: *geom_boxplot()*
- If you add functionality than use operator “+”.

Gray shades are fine, but color matters



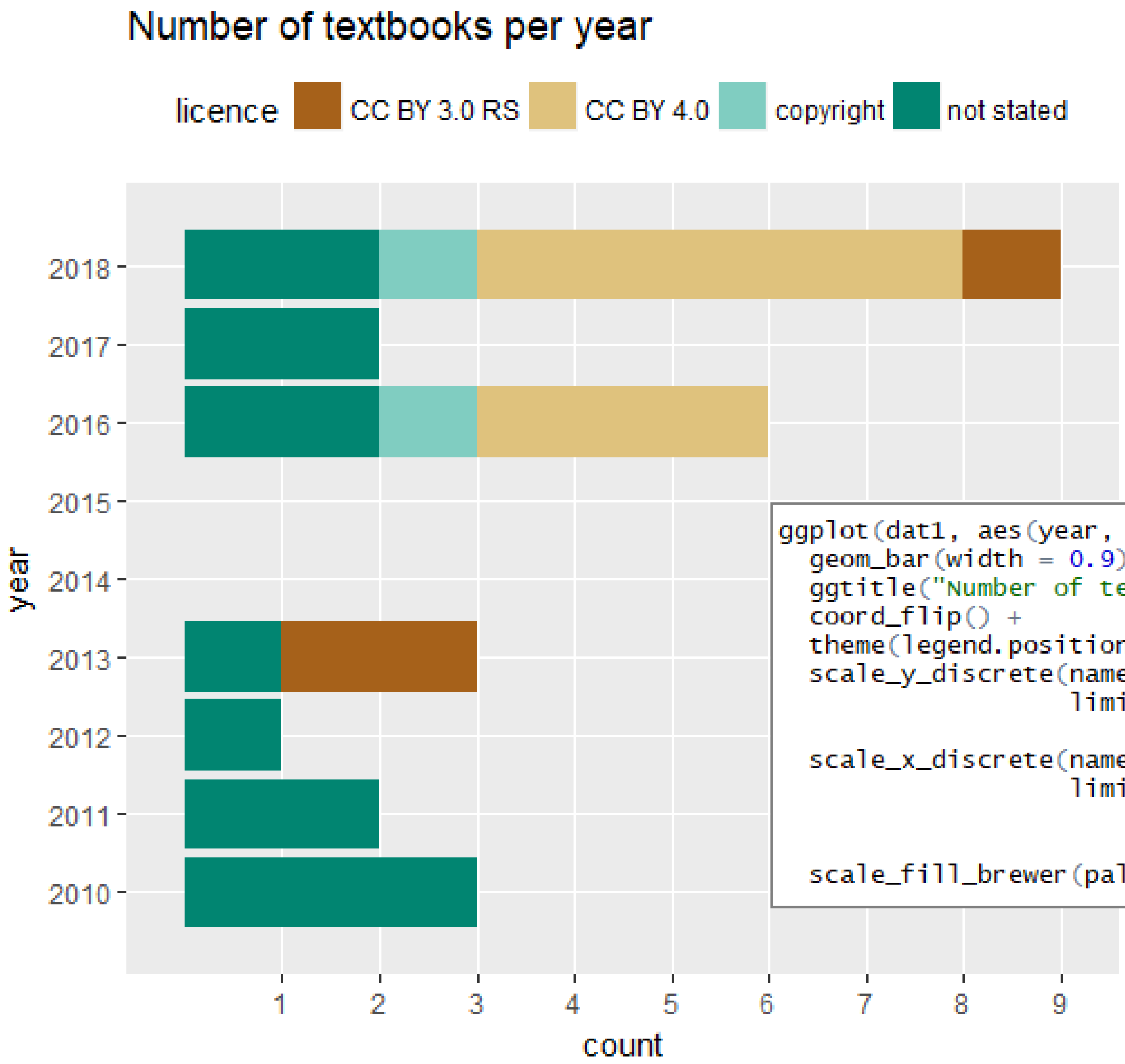
```
ggplot(dat1, aes(year, fill = `application of free software`)) +
  geom_bar(width = 0.9) +
  ggtitle("Number of textbooks per year") +
  coord_flip() +
  theme(legend.position = "top")  +
  scale_y_discrete(name = "count",
                   limits = c("1", "2", "3", "4", "5",
                              "6", "7", "8", "9", "10")) +
  scale_x_discrete(name = "year",
                   limits = c(2010, 2011, 2012, 2013,
                              2014, 2015, 2016, 2017,
                              2018))
```

Both cold & hot can hurt you



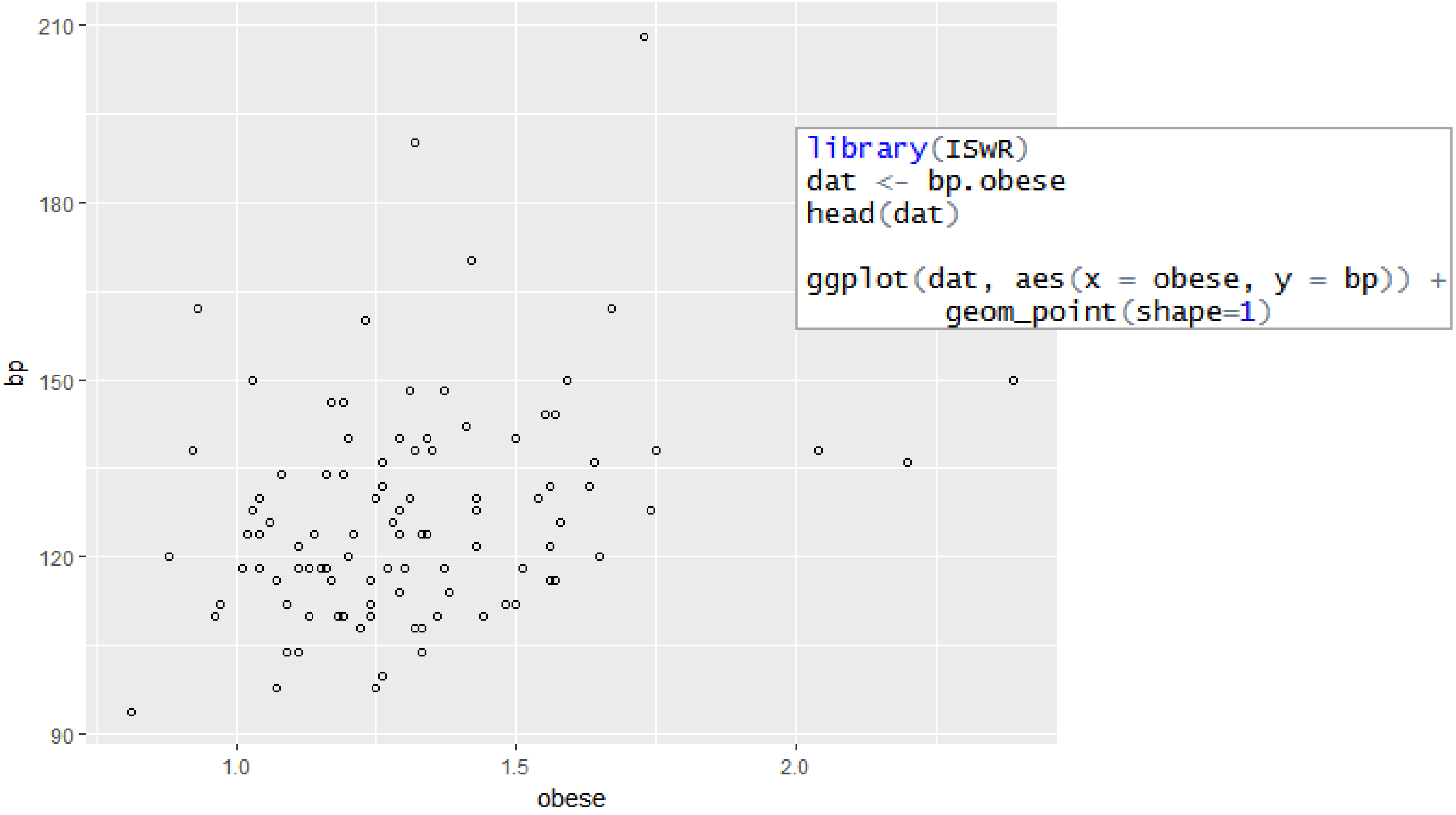
```
ggplot(dat1, aes(year, fill = `application of free software`)) +
  geom_bar(width = 0.9) +
  ggtitle("Number of textbooks per year") +
  coord_flip() +
  theme(legend.position = "top")  +
  scale_y_discrete(name = "count",
                   limits = c("1", "2", "3", "4", "5",
                              "6", "7", "8", "9", "10")) +
  scale_x_discrete(name = "year",
                   limits = c(2010, 2011, 2012, 2013,
                              2014, 2015, 2016, 2017,
                              2018)) +
  scale_fill_brewer(palette = "Accent")
```

Another example



```
ggplot(dat1, aes(year, fill = licence)) +
  geom_bar(width = 0.9) +
  ggtitle("Number of textbooks per year") +
  coord_flip() +
  theme(legend.position = "top") +
  scale_y_discrete(name ="count",
                   limits=c("1", "2", "3", "4", "5",
                            "6", "7", "8", "9", "10")) +
  scale_x_discrete(name ="year",
                   limits=c(2010, 2011, 2012, 2013,
                            2014, 2015, 2016, 2017,
                            2018)) +
  scale_fill_brewer(palette = "BrBG")
```

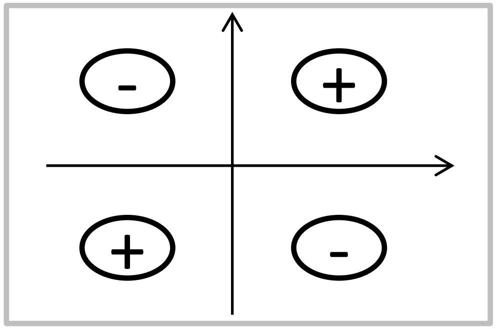
For homework: scatter plot



```
library(ISwR)
dat <- bp.obese
head(dat)

ggplot(dat, aes(x = obese, y = bp)) +
        geom_point(shape=1)
```

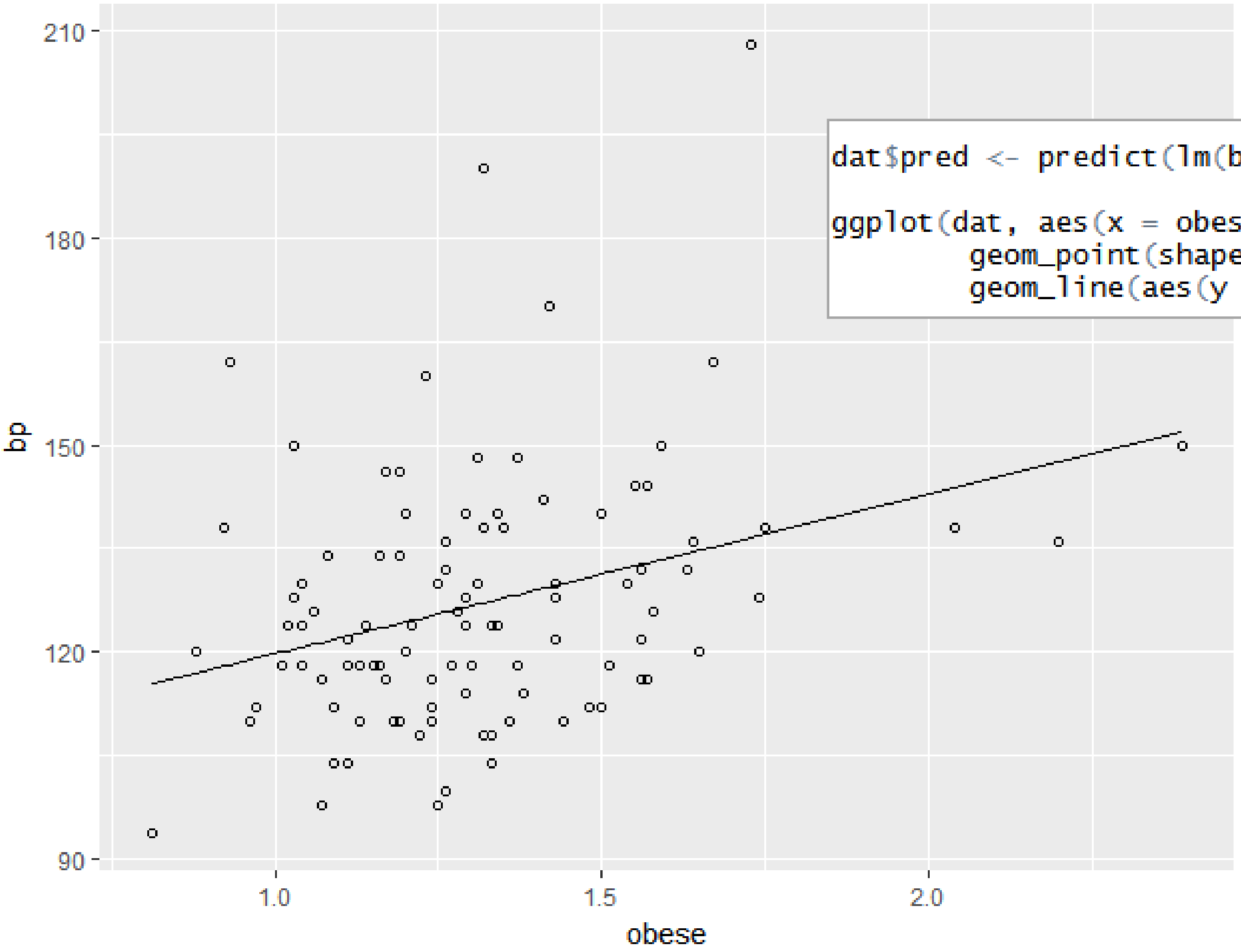
- Relation of BMI (Body Mass Index) and systolic blood pressure is presented ("bp.obese" data from ISwR package).
- It is a must in any data analysis (https://en.wikipedia.org/wiki/Scatter_plot).
- Try using ggplot2 package without lm() function and plot non-linear model by your choice.

Scatter plot anatomy



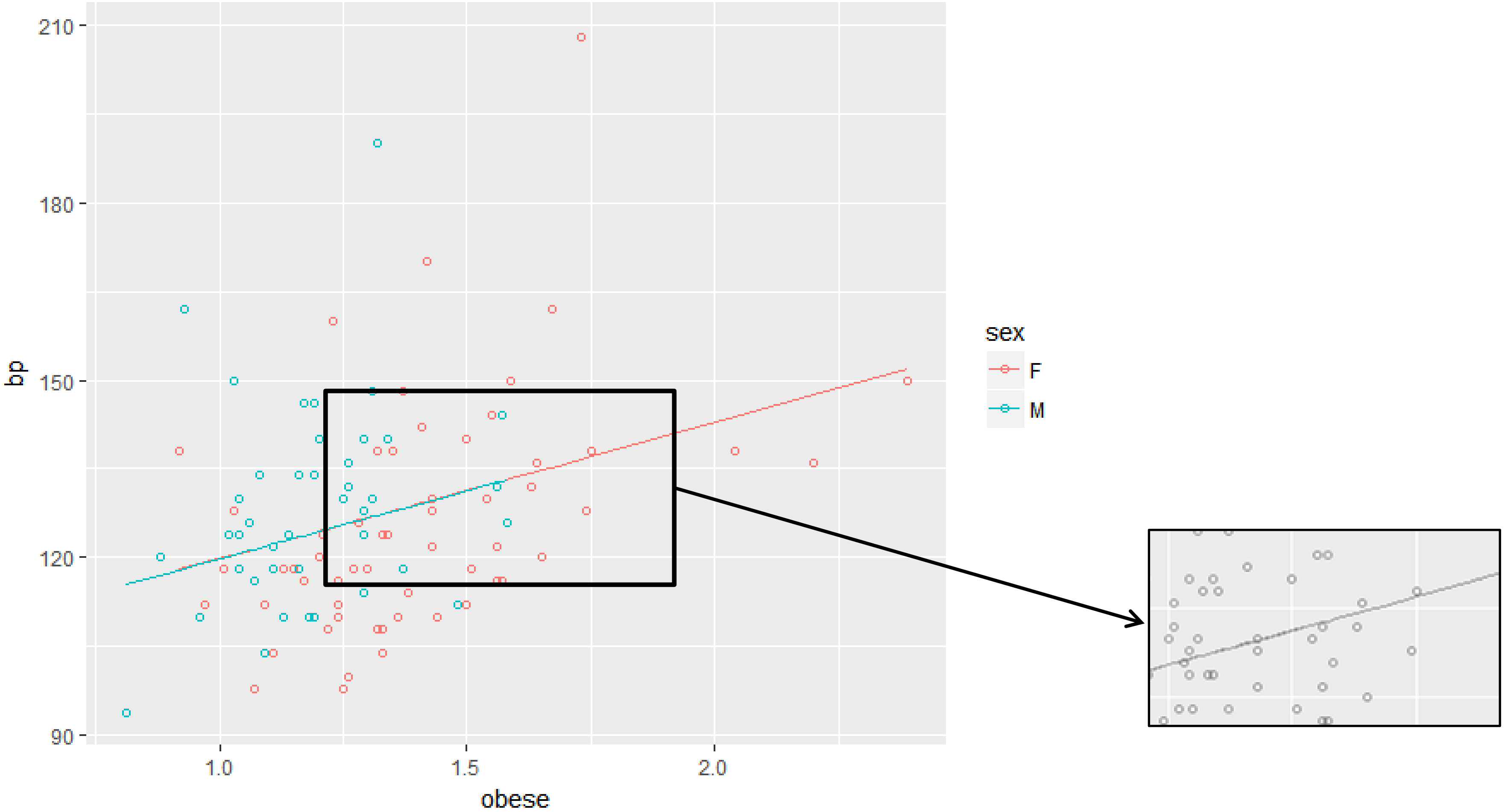
- Scatter plots are commonly used to test the correlation between two variables.
- Graphical representation of correlation is given at the bottom panel.

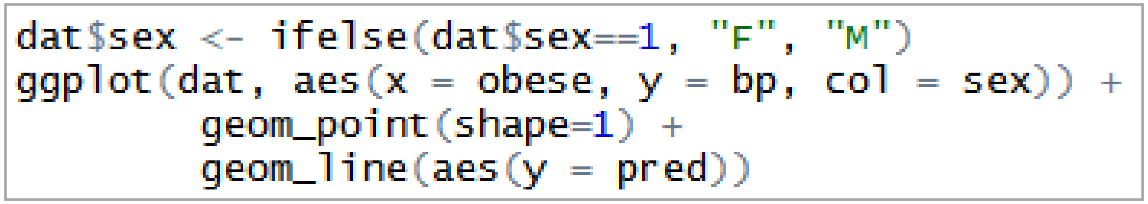
Linear models…



```
dat$pred <- predict(lm(bp ~ obese, data = dat))

ggplot(dat, aes(x = obese, y = bp)) +
        geom_point(shape=1) +
        geom_line(aes(y = pred))
```

...





```
dat$sex <- ifelse(dat$sex==1, "F", "M")
ggplot(dat, aes(x = obese, y = bp, col = sex)) +
        geom_point(shape=1) +
        geom_line(aes(y = pred))
```

VISUALIZATION: RECOMMENDATIONS AND CONCLUSIONS

Rules for visualization

- First visualization, then analysis!
- Visualization is the most important part of data analysis.
- Picture: By Randri87 - https://www.triopticaonline.com/comprar/gafas-de-vista/, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=52841451.

Other rules

- *De gustibus non disputandum est*, but ...
- Main goal of visualization is to present the information in appropriate way. There are some basic rules.
- You should NEVER EVER:
  - use pie chart,
  - colors that do not have “good” contrast,
  - 3D graph if there is an appropriate 2D graph,
  - colors for information that is presented by other means
  - use graph that does not present or only partially presents important information
- I recommend video tutorial by prof. Rafael Irizarry “Statistics for Genomics: Useful plots and bad plots”, https://www.youtube.com/watch?v=46-t2jOYsyY, [Online], Assessed October 18, 2018.

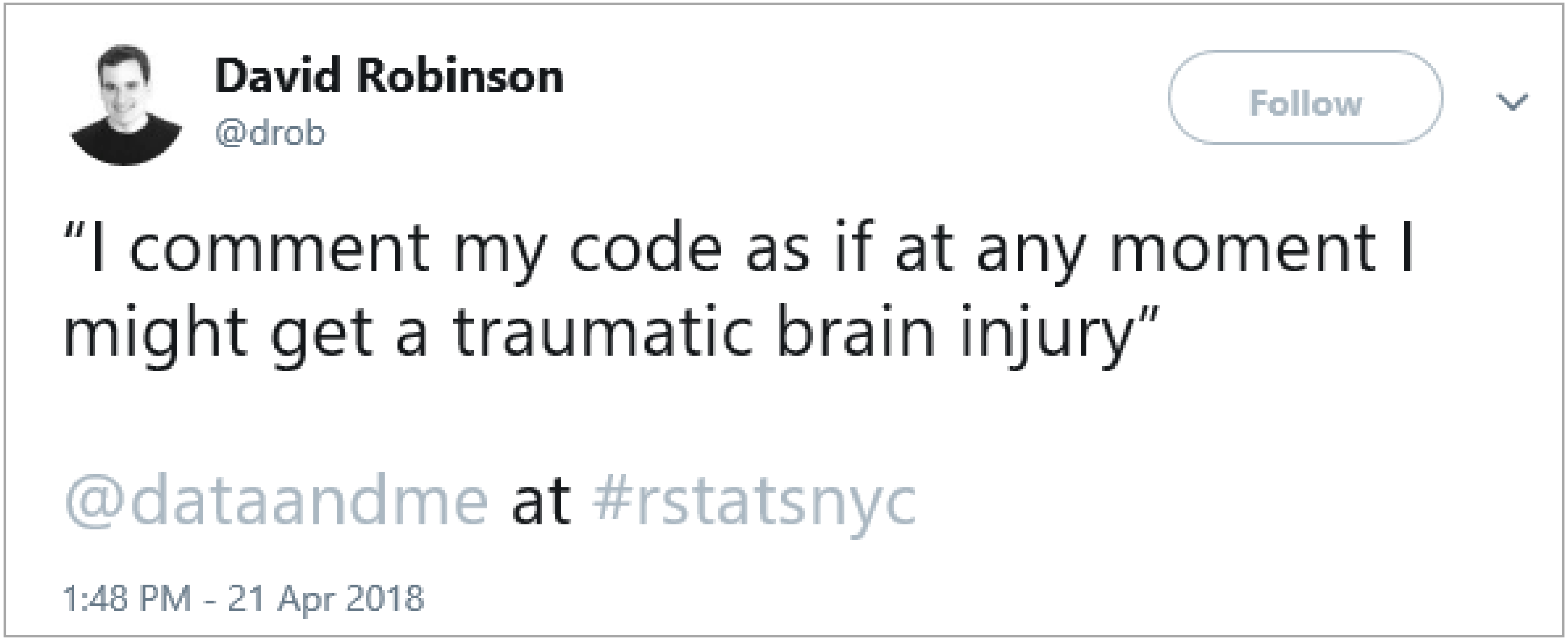
When you should use pie chart?

By Scott Bauer - http://www.ars.usda.gov/is/graphics/photos/mar01/k7633-3.htm, Јавно власништво, https://commons.wikimedia.org/w/index.php?curid=79501

What is worse than pie chart?

- Rotated pie chart
- Two pie charts
- …

Rules, rules, rules



David Robinson
@drob

"I comment my code as if at any moment I might get a traumatic brain injury"

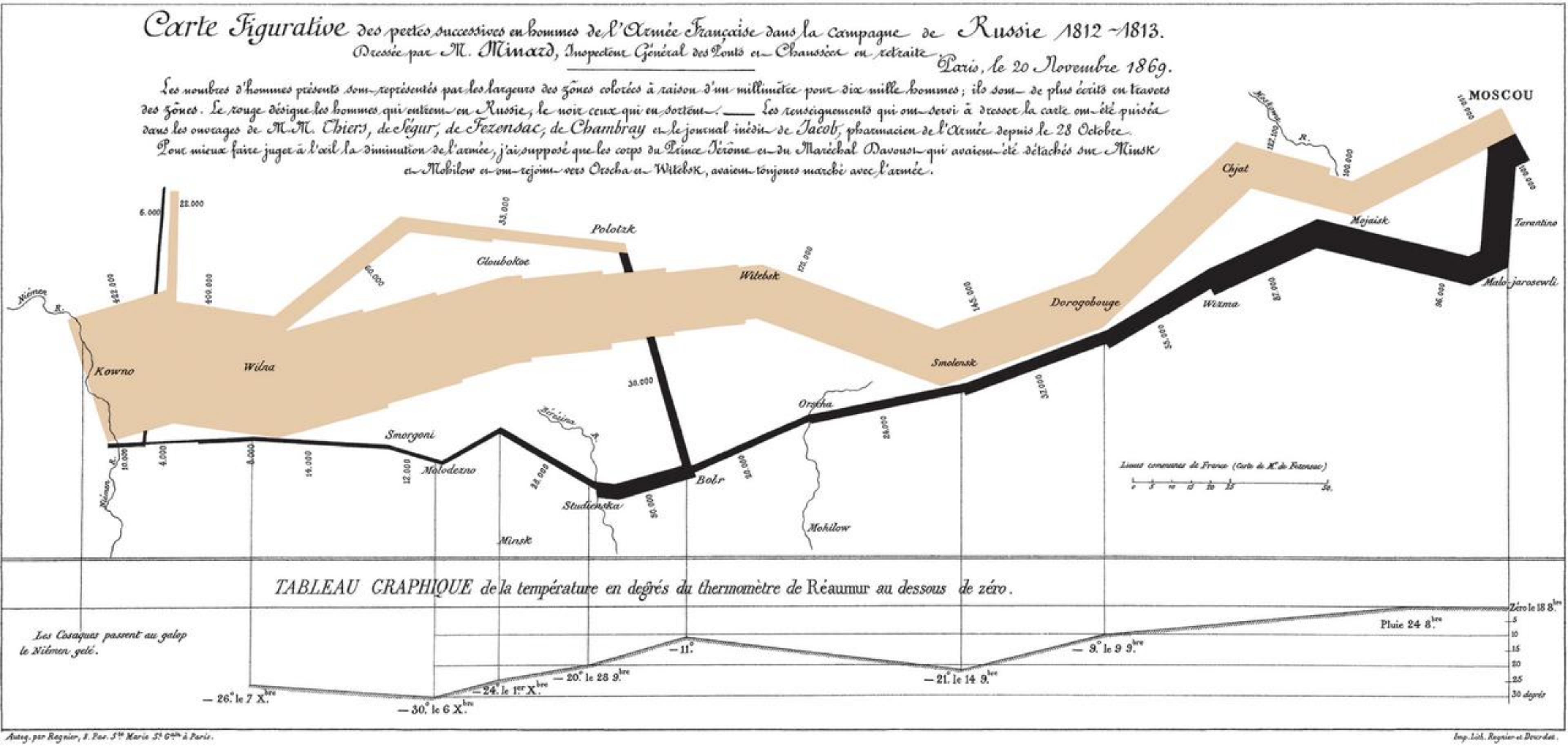
@dataandme at #rstatsnyc

1:48 PM - 21 Apr 2018

https://twitter.com/drob/status/987795355659112453

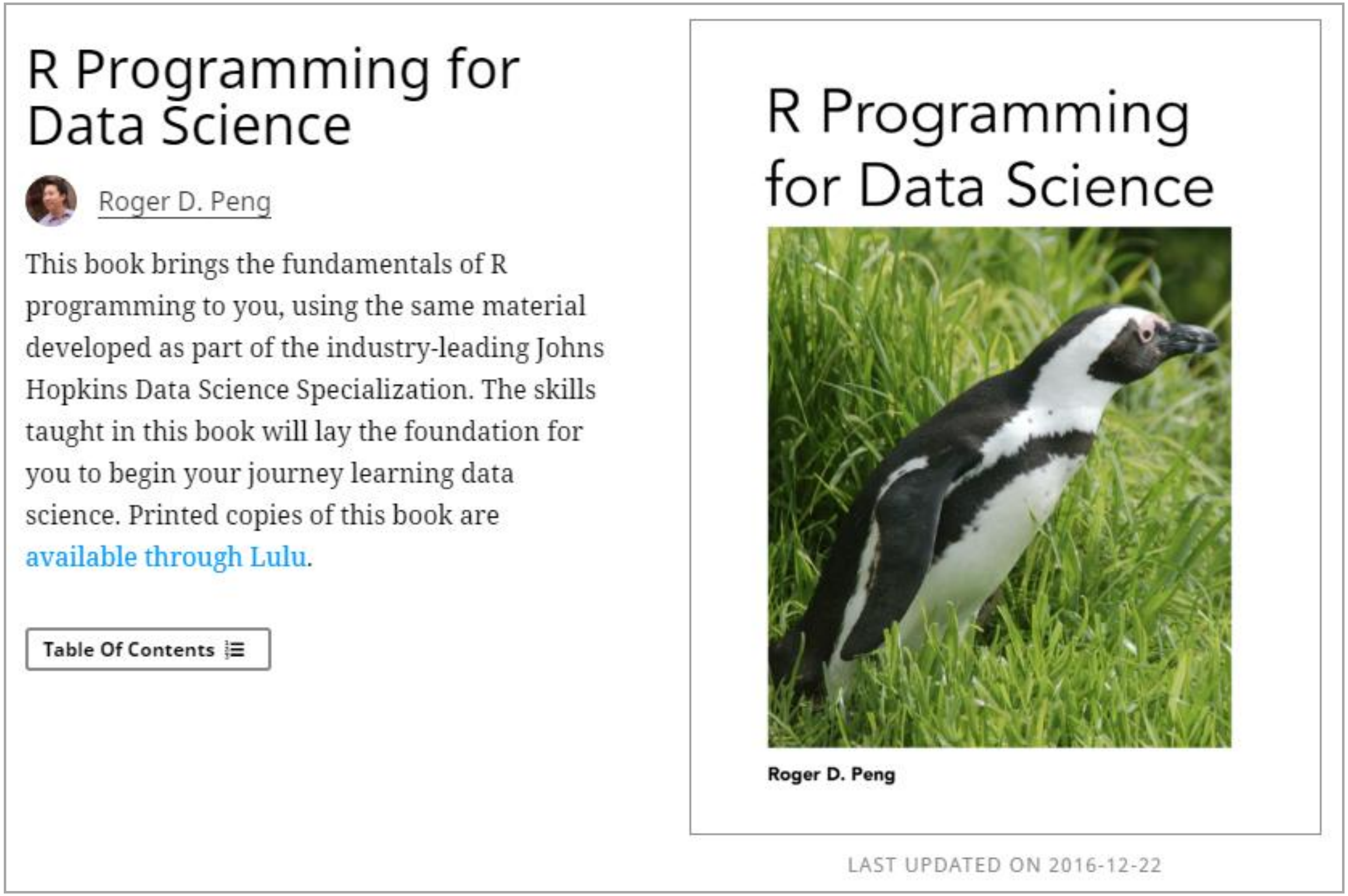
WHERE TO GO NOW FROM HERE?

Inspirational Minard’s map


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Niémen R.
Kowno
Wilna
Smorgoni
Molodezno
Minsk
Studienska
Bérézina R.
Bobr
Polotzk
Gloubokoe
Witebsk
Orscha
Mohilow
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Moskowa R.
422.000
400.000
60.000
33.000
6.000
22.000
125.000
145.000
175.000
100.000
100.000
96.000
87.000
55.000
37.000
24.000
20.000
28.000
50.000
30.000
14.000
8.000
12.000
4.000
10.000
Lieues communes de France (Carte de M. de Fezensac)
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8.bre
Pluie 24 8.bre
— 9.° le 9 9.bre
— 21.° le 14 9.bre
— 11.°
— 20.° le 28 9.bre
— 24. le 1er X.bre
— 30.° le 6 X.bre
— 26.° le 7 X.bre
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. Ste Marie St Gain à Paris.
Imp. Lith. Regnier et Dourdet.

By Charles Minard (1781-1870) - see upload log, Public Domain, https://commons.wikimedia.org/w/index.php?curid=297925.

R book (for learning R)



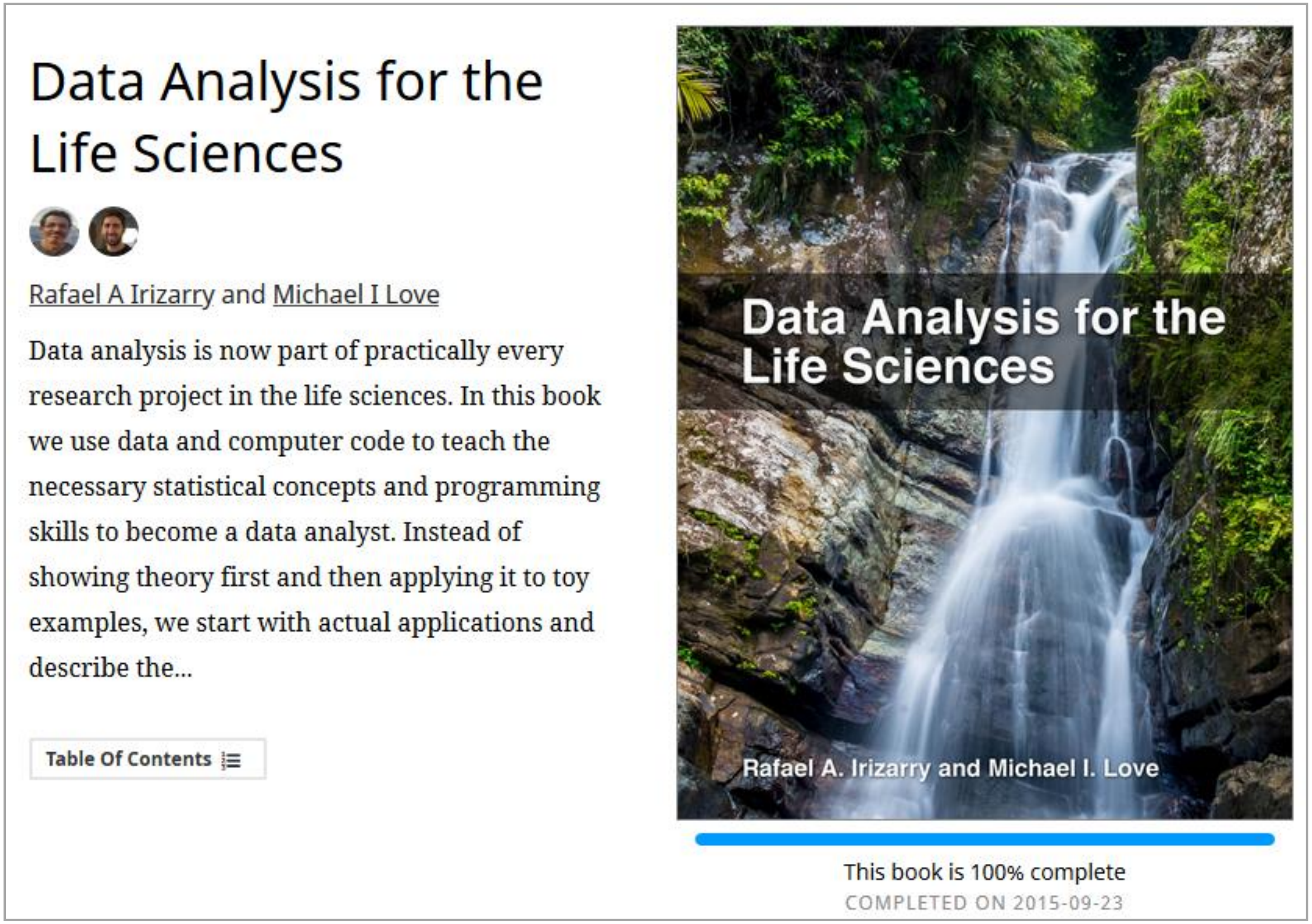
R Programming for Data Science

Roger D. Peng

This book brings the fundamentals of R programming to you, using the same material developed as part of the industry-leading Johns Hopkins Data Science Specialization. The skills taught in this book will lay the foundation for you to begin your journey learning data science. Printed copies of this book are available through Lulu.

- Peng R. D. R programming for data science, Leanpub book, 2014-2016.
- https://leanpub.com/rprogramming.

R book (for understanding R)



Data Analysis for the Life Sciences

Rafael A Irizarry and Michael I Love

Data analysis is now part of practically every research project in the life sciences. In this book we use data and computer code to teach the necessary statistical concepts and programming skills to become a data analyst. Instead of showing theory first and then applying it to toy examples, we start with actual applications and describe the...

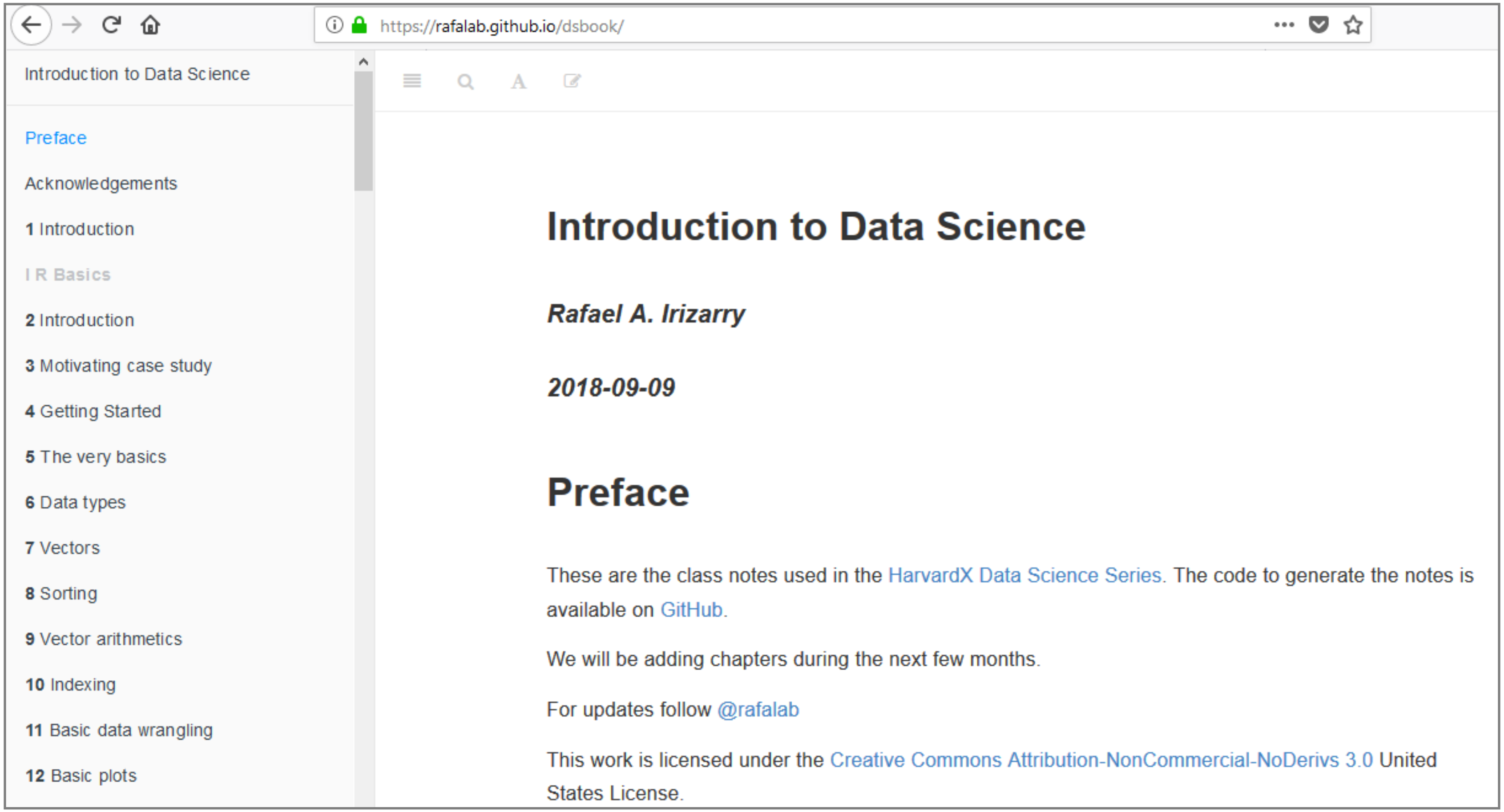
Table Of Contents
Data Analysis for the Life Sciences
Rafael A. Irizarry and Michael I. Love

This book is 100% complete

COMPLETED ON 2015-09-23

- Irizarry R. and Love M. I. Data analysis for the life sciences, Leanpub book, 2015.
- https://leanpub.com/dataanalysisforthelifesciences.

R book (for Data Science)



Introduction to Data Science

Rafael A. Irizarry

2018-09-09

Preface

These are the class notes used in the HarvardX Data Science Series. The code to generate the notes is available on GitHub.

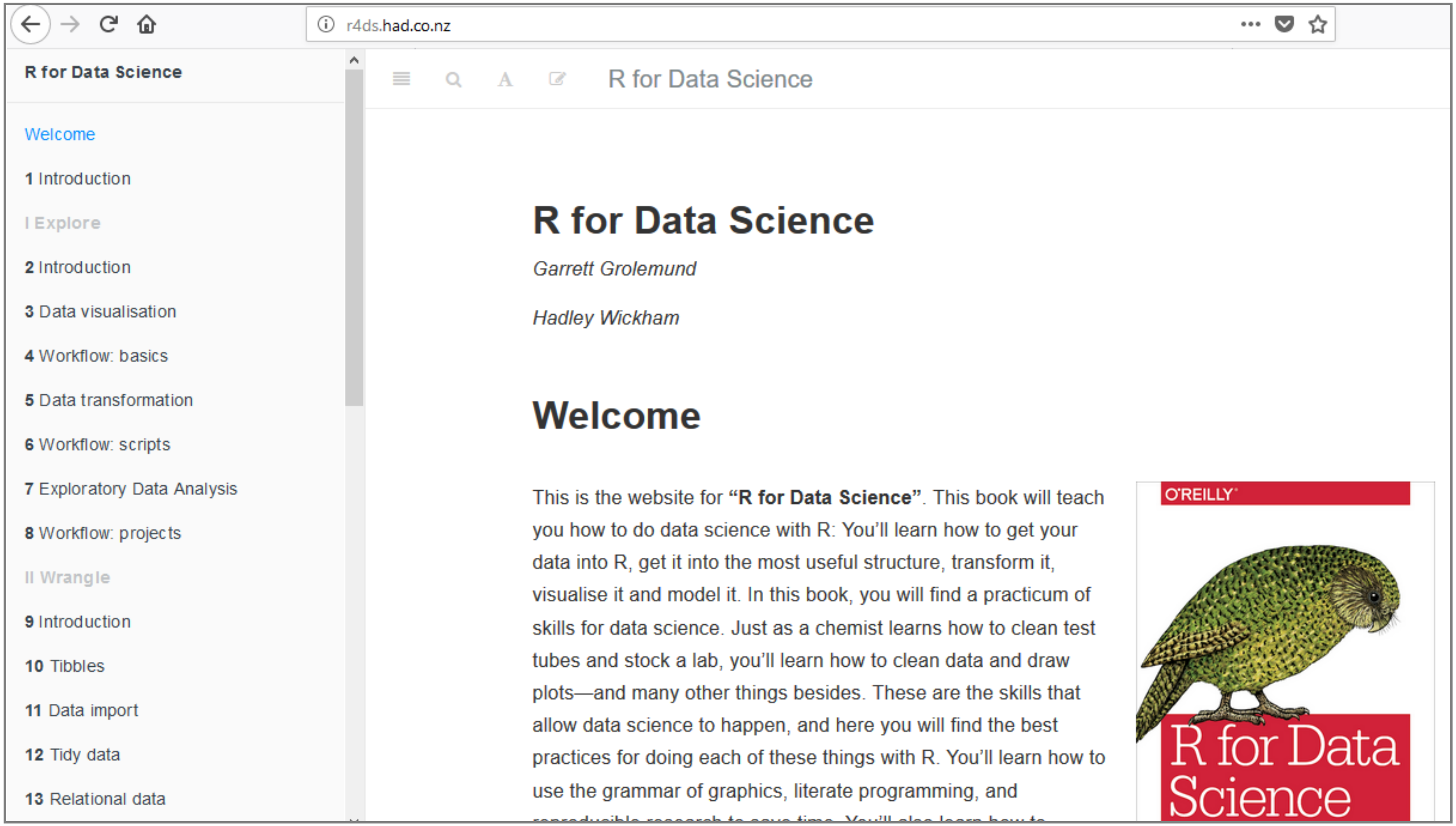
We will be adding chapters during the next few months.

For updates follow @rafalab

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 United States License.

Irizarry R. A. Introduction to data science, https://rafalab.github.io/dsbook/, 2018.

R book (for Data Science)



R for Data Science

Garrett Grolemund

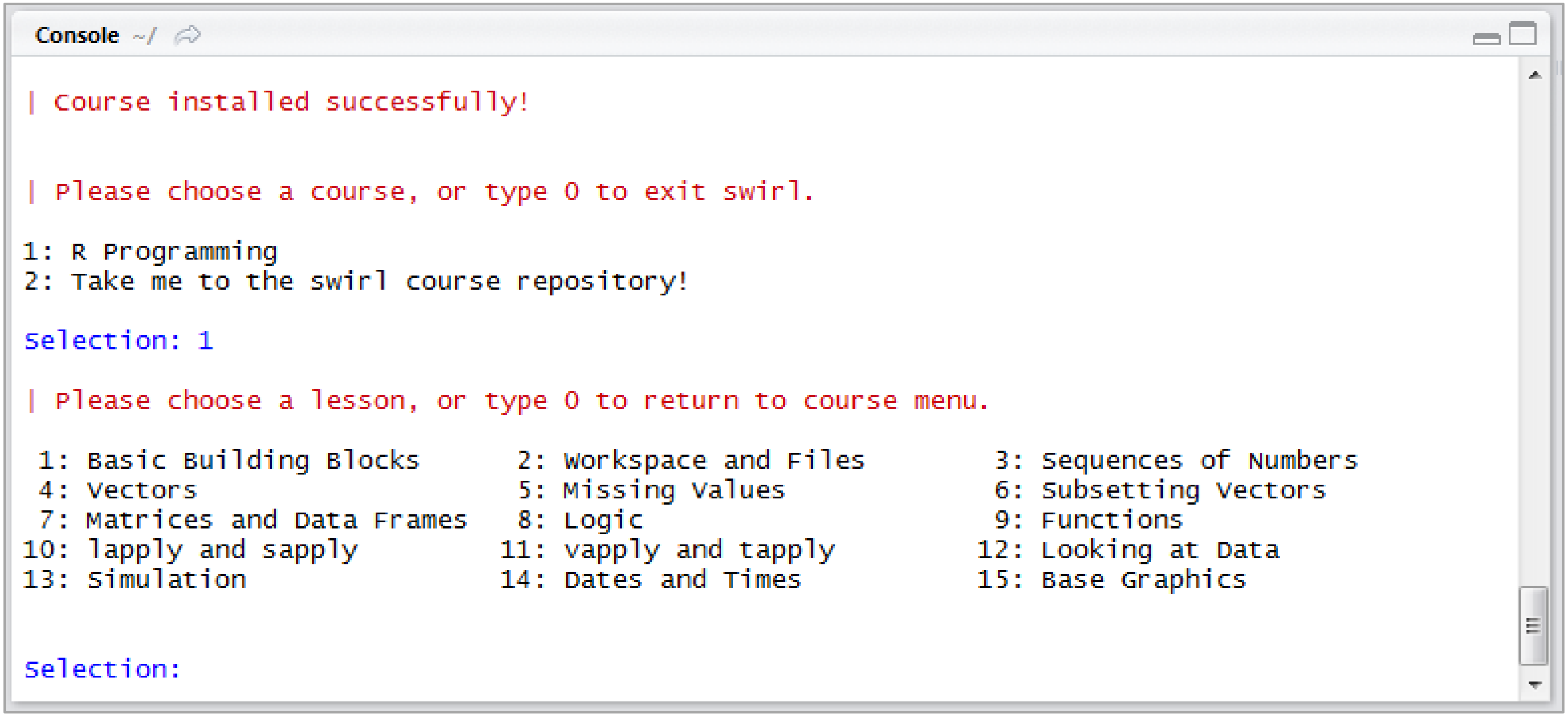
Hadley Wickham

Welcome

This is the website for **"R for Data Science"**. This book will teach you how to do data science with R: You'll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you'll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You'll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You'll also learn how to

Garrett G. and H. Wickham, R for Data Science, http://r4ds.had.co.nz/, 2017.

Interactive learning



```
| Course installed successfully!


| Please choose a course, or type 0 to exit swirl.

1: R Programming
2: Take me to the swirl course repository!

Selection: 1

| Please choose a lesson, or type 0 to return to course menu.

 1: Basic Building Blocks      2: Workspace and Files       3: Sequences of Numbers
 4: Vectors                    5: Missing Values            6: Subsetting Vectors
 7: Matrices and Data Frames   8: Logic                     9: Functions
10: lapply and sapply         11: vapply and tapply        12: Looking at Data
13: Simulation                14: Dates and Times          15: Base Graphics


Selection:
```

- By installing swirl package (http://swirlstats.com/) you can use R's interactivity to practice more.
- I used "R programming E" swirl course with interactive exercises.