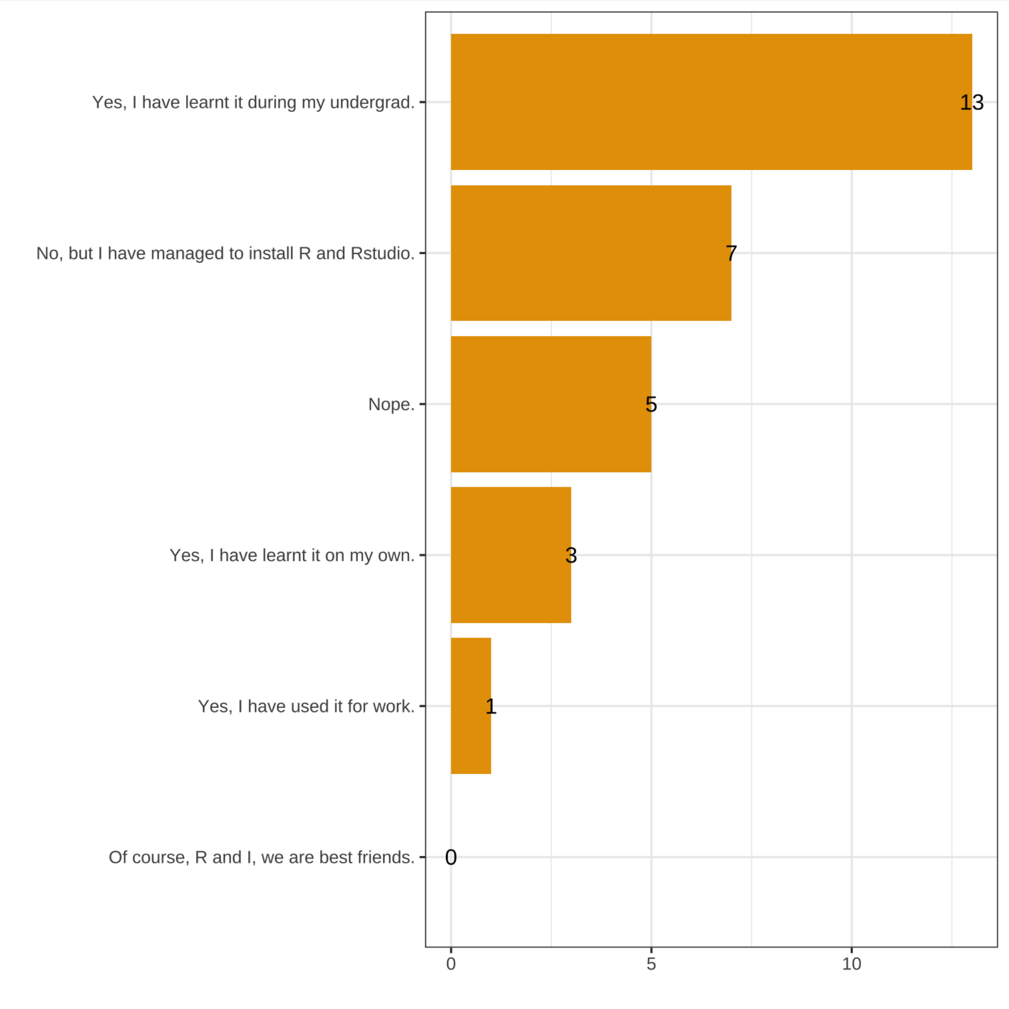
Introduction to Base R

Johann S. Schuur

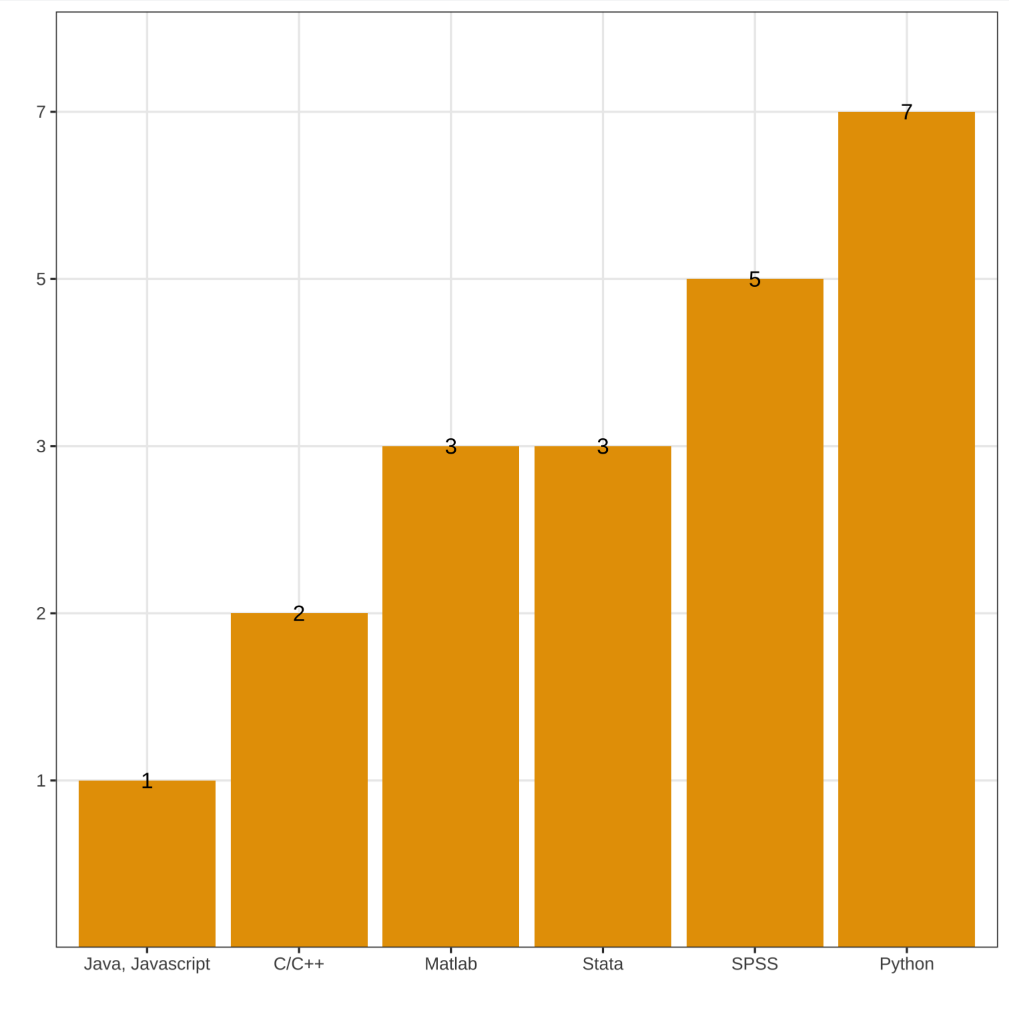
09.09.2020

Some poll results :)

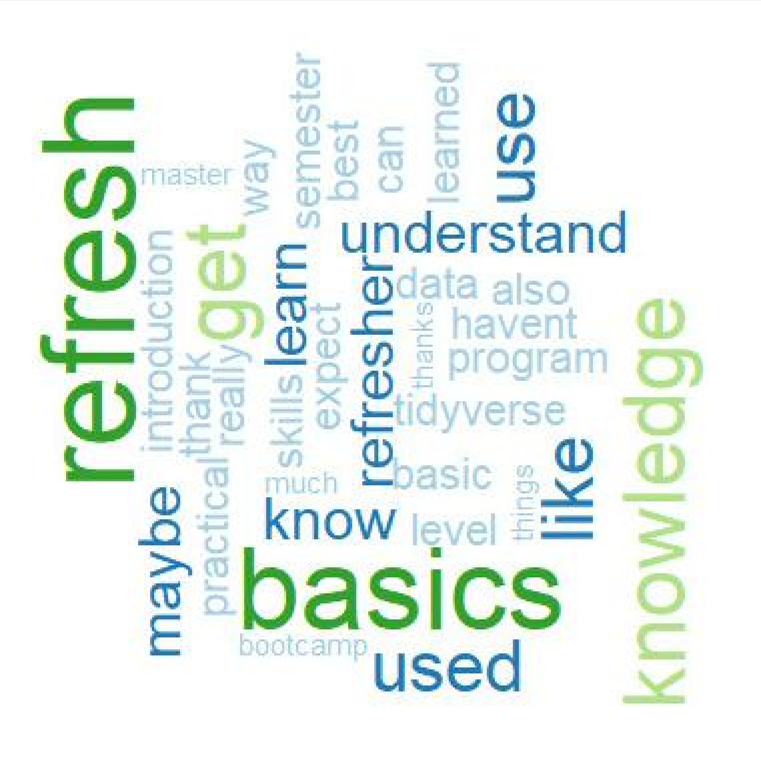
Previous R experience



Experience with other languages



R Bootcamp expectations



Outlook

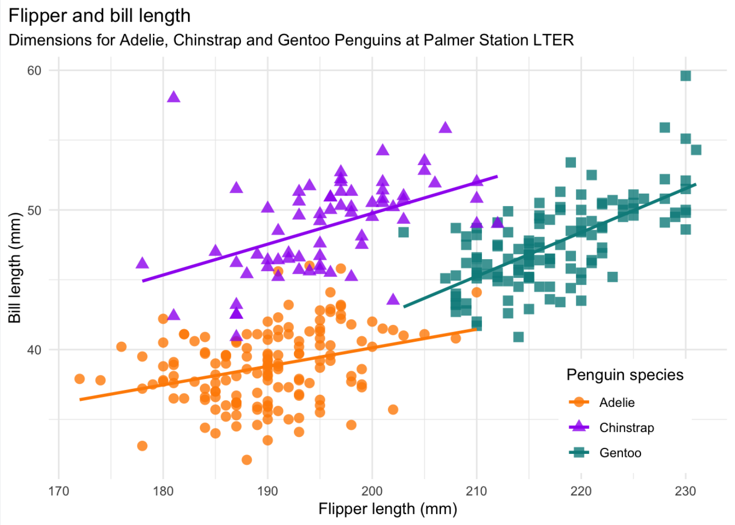
- 9:00 - 10:00 (warming up)
    - Introduction
    - R in your study program
    - other languages
    - installing R
    - the Integrated Development Environment (IDE)
- 10:00 - 10:15 (Virtual coffee break)
- 10:15 - 11:10 (Base R - Part 1)
    - variables and R as calculator
    - data structures (vectors, matrix, lists, indexing)
    - packages and libraries
    - base plotting
- 11:10 - 11:25 (Virtual coffee break)
- 11:25 - 12:10 (Base R - Part 2)
    - writing scripts (best practices)
    - data handling
    - if-else statements
- 12:10 - 12:30 (Virtual coffee break)
    - Time to exercise

The use of R

- Master in Comparative and International Studies (MACIS)
- Master in Science, Technology, and Policy (STP)

The use of R

- R is considered best for
  - graphing and visualizations,
  - data analysis and statistical computing



- Python goes beyond data analysis
  - developing and programming
  - web-scraping (STP course: Big data for Public Policy)
- both borrow from eachother however :)

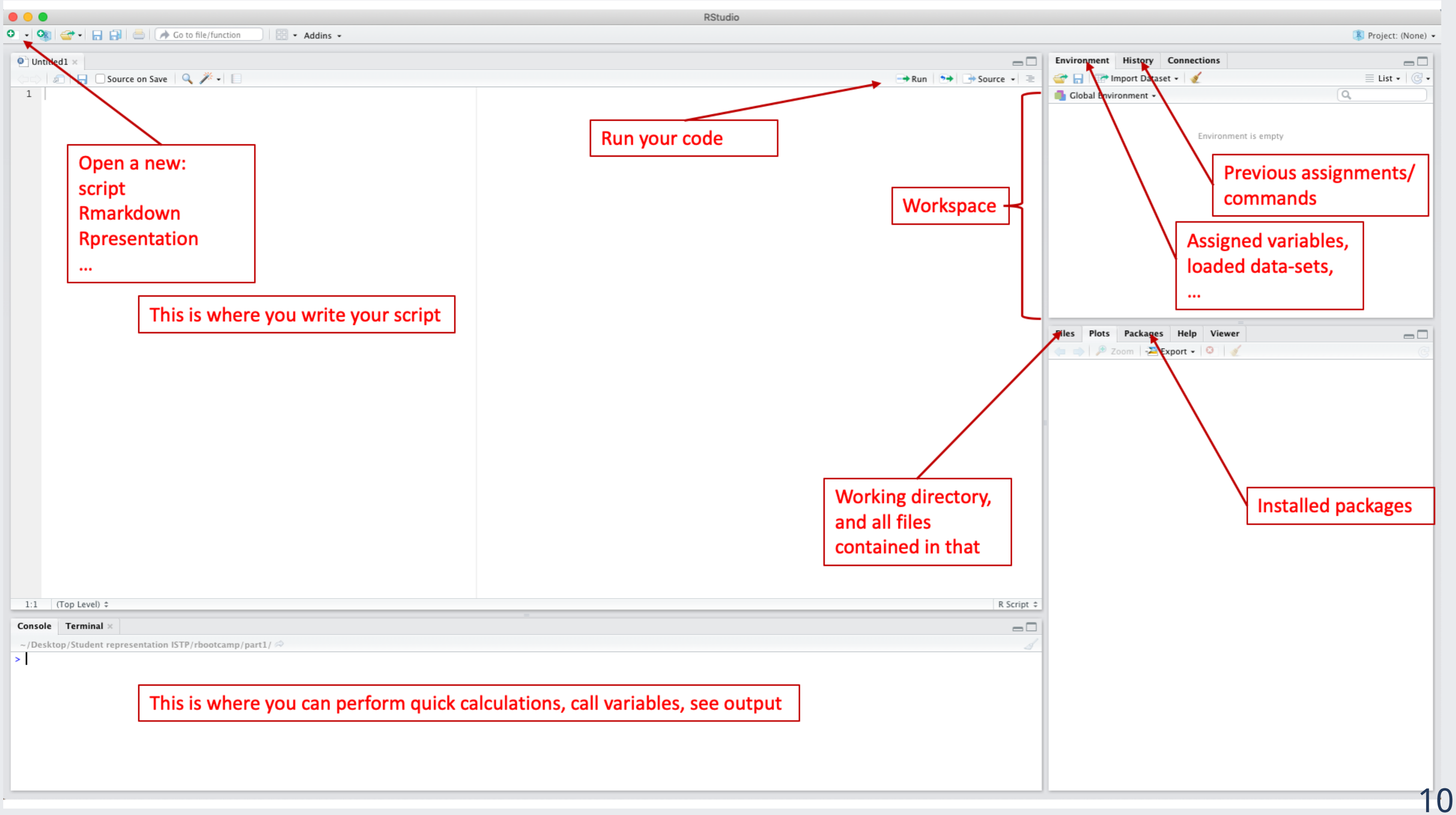
https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis#gs.k=J5=oY

Installing R

If you haven't done already, go to: https://courses.edx.org/courses/UTAustinX/UT.7.01x/3T2014/56c5437b88fa43cf828bff5371

- you need both!
  - R is the language
  - Rstudio is your IDE ("user interface")

The Integrated Development Environment (IDE)



Open a new:
script
Rmarkdown
Rpresentation
…

This is where you write your script

Run your code

Workspace

Previous assignments/ commands

Assigned variables, loaded data-sets, …

Working directory, and all files contained in that

Installed packages

This is where you can perform quick calculations, call variables, see output

Virtual coffee break (10:00 - 10:15)

Recap

- R in your study program
- other languages
- installing R
- the Integrated Development Environment (IDE)

Questions in break-out sessions

Variables and R as calculator

```r
# assigning variables
a <- 5
b <- 10

# performing a computation and calling the result
a + b # (Windows: ctrl + enter, Mac: cmd + enter)
```

```
## [1] 15
```

for more keyboard shortcuts, see: https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts

Data structures

- vector

```
# creating vectors
x <- c(1:9) # c(...), combines arguments into vector or list
```

- matrix
- array
- lists
- data frames

Data structures

- vector
- matrix

```r
# creating matrices
A <- matrix(x, nrow = 3, ncol = 3)

# assigning row, column and matrix names
rownames(A) <- c("penguin1", "penguin2", "penguin3")
colnames(A) <- c("species", "sex", "year")
matrix_names <- c("set1", "set2")
```

- array
- lists
- data frames

Data structures

- vector
- matrix
- array

```
array1 <- array(c(A,A), dim = c(3,3,2), dimnames = list(rownames(A),
                colnames(A), matrix_names))
                # takes multidimensional data objects
```

- lists
- data frames

Data structures

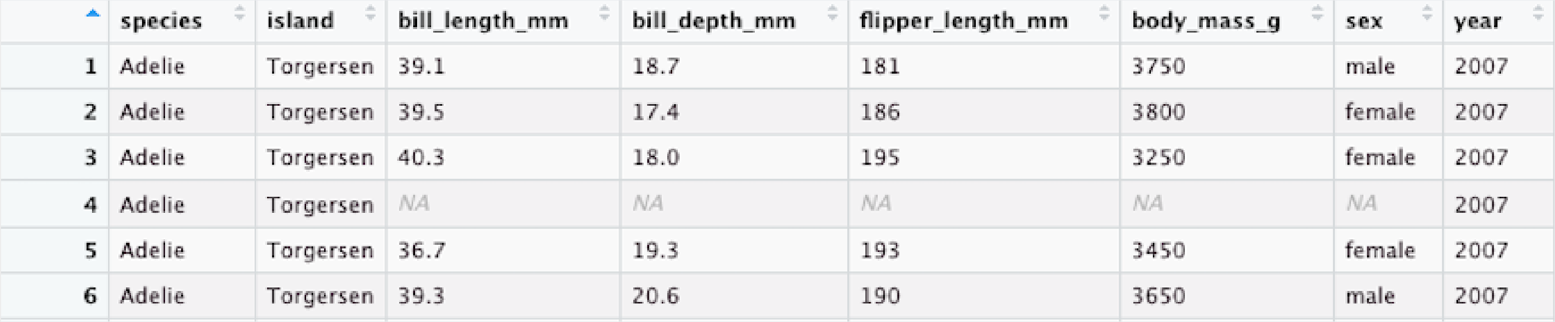
- vector
- matrix
- array
- lists

```r
l1 <- list(x, A, "hello", TRUE, FALSE) # elements of different types
```

- data frames

Data structures

- vector
- matrix
- array
- lists
- data frames
    - have column names
    - unique rownames
    - handle numeric, factor, or character data
    - each column contains same number of entries



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18.0	195	3250	female	2007
4	Adelie	Torgersen	NA	NA	NA	NA	NA	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	39.3	20.6	190	3650	male	2007

Indexing

```
x[5] # accessing fifth element of vector
```

```
## [1] 5
```

```
A[2,3] # accessing second row, third column of matrix
```

```
## [1] 8
```

```
A[c(1,2), c(2,3)] # ... 1st and 2nd row of 2nd and 3th column
```

```
##          sex year
## penguin1   4    7
## penguin2   5    8
```

```
array1[1,,1] # ... 1st row, all columns, 1st matrix
```

```
## species     sex    year
##       1       4       7
```

for more indexing, see: https://data-flair.training/blogs/data-structures-in-r/

Packages and libraries

Packages

- R is open source, we benefit from packages made (and updated) by the R community
- they organize work and typically contain:
  - code
  - specific functions
  - documentation ("README's")
  - data-sets
- to work with them they need to be installed (you do this once)

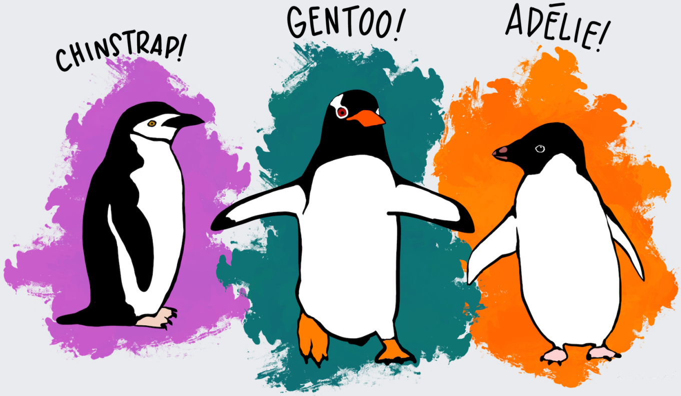
```
#install.packages("palmerpenguins") # we will need this package later
install.packages(c("palmerpenguins",
                   "tidyverse",
                   "ggplot2")) # we will need these packages later
```

Packages and libraries

Library

- using a package for a new project, we need to load it (you do this every time)

```r
library(palmerpenguins)
```



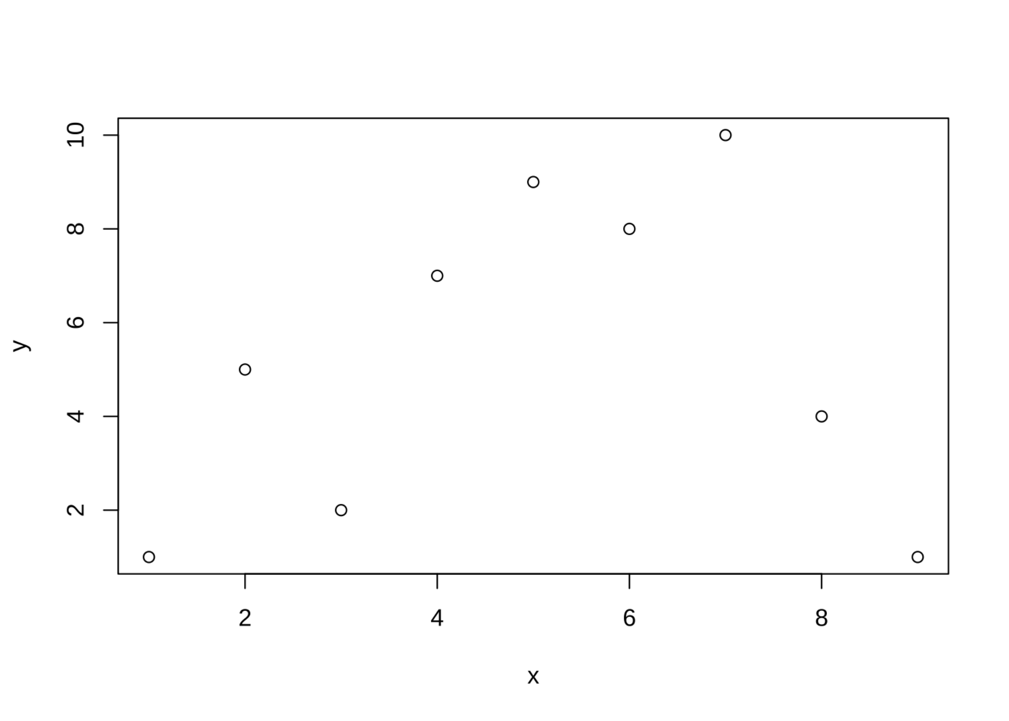
“Artwork by @allison_horst”

- you cannot input a number of libraries at the same time

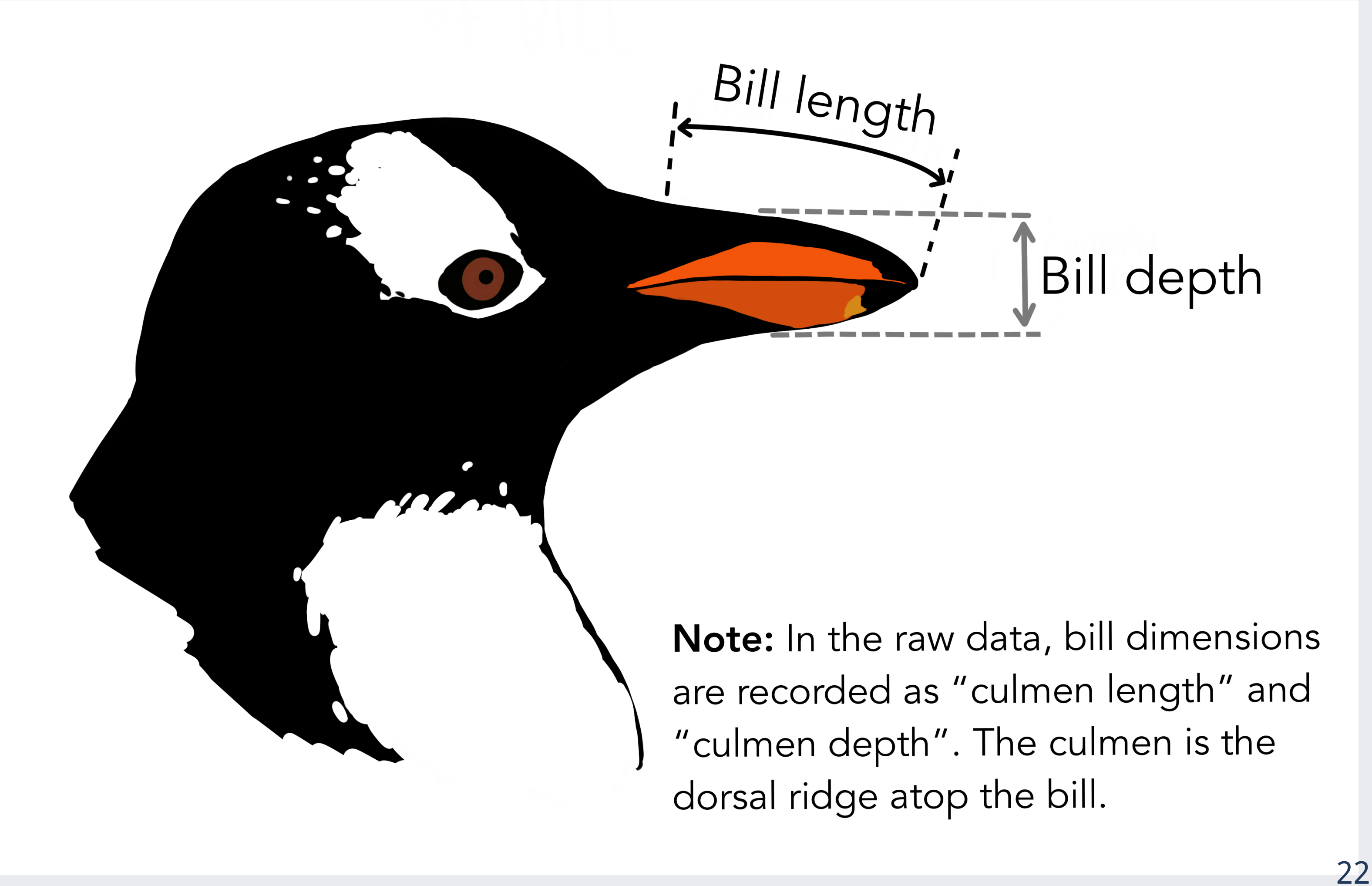
Base R plotting

- for a quick and simple visualization of your data (today)
- we will mainly use ggplot (tomorrow)

```
y <- c(1,5,2,7,9,8,10,4,1)
plot(x, y) # plotting y vs x
```



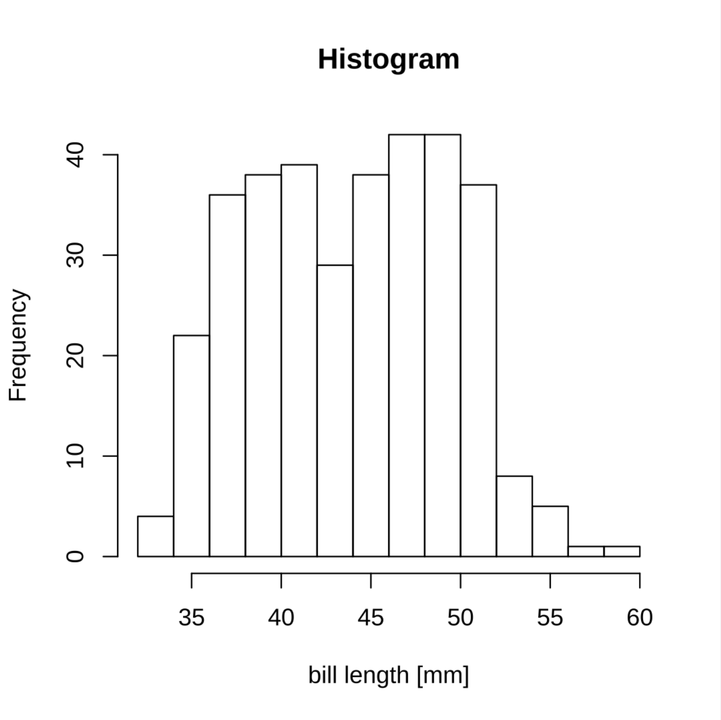
Base R plotting



Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

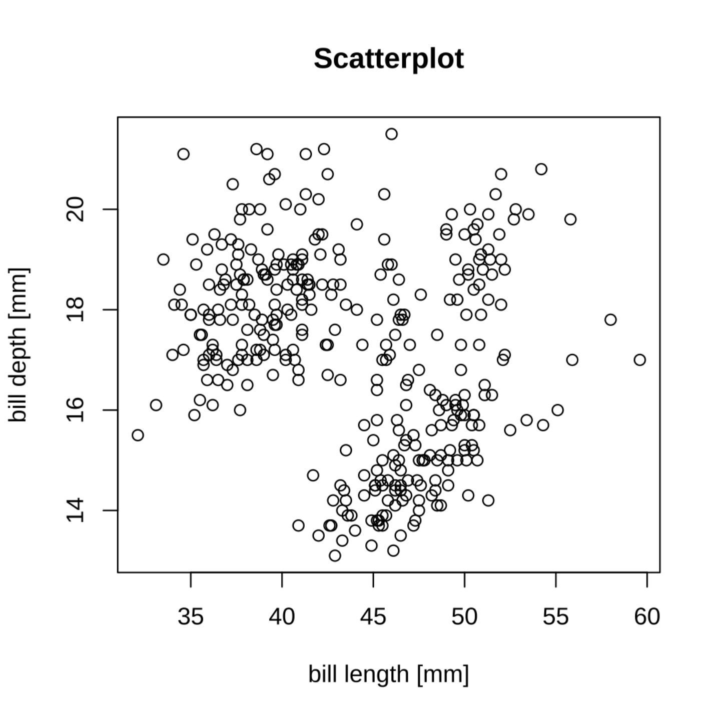
Base R plotting

```
# visualizing our dataset
hist(penguins$bill_length_mm,
     # hist(x-data, ...)
     main = "Histogram",
     xlab = "bill length [mm]")
```



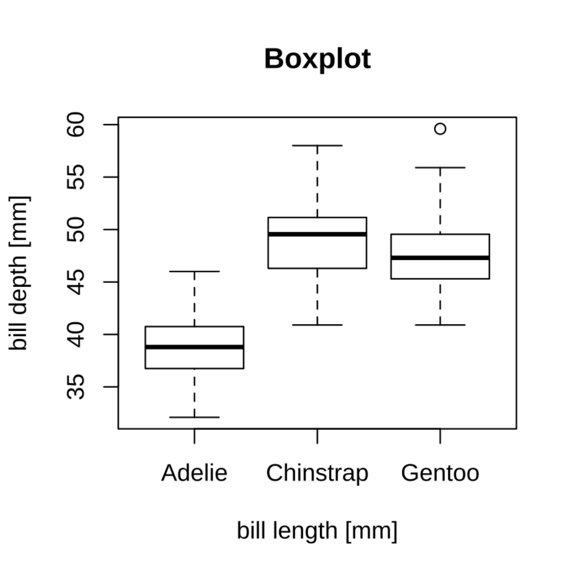
Base R plotting

```
plot(penguins$bill_length_mm, penguins$bill_depth_mm,
     # plot(x-data, y-data, ...)
     main = "Scatterplot",
     xlab = "bill length [mm]", ylab = "bill depth [mm]")
```



Base R plotting

```
boxplot(bill_length_mm ~ species, penguins,
        # boxplot(y-data ~ x-data, dataframe, ...)
        main = "Boxplot",
        xlab = "bill length [mm]", ylab = "bill depth [mm]")
```



- For more base R plotting, see: https://bookdown.org/rdpeng/exdata/the-base-plotting-system-1.html

Virtual coffee break (11:10 - 11:25)

Recap

- variables and R as calculator
- data structures (vectors, matrix, lists, indexing)
- packages and libraries
- base plotting

Questions in break-out sessions

Writing scripts

- Set up find and set your working directory

```
getwd() # tells you in which working directory you currently operate
setwd() # lets you set your desired working directory
```

Example:

```
setwd("~/Desktop/Student representation ISTP/rbootcamp/part1/")
```

- Make a separate folder for each project, that contains
  - data
  - figures
  - ...

Why?

- keep things tidy
- share your work
- have people replicate your work

Writing scripts (best practices)

- Best practices: Create a "New project"
  - this does all the above for you
  - allows you to work on mutiple projects simultaneously
  - DOESN'T CLUTTER YOUR WORKSPACE :)

for more, see: https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects

Writing scripts

- Indenting and commenting
    - comment where necessary, don't go overboard
    - use spaces after each input
    - object names: all lowercase, short, and yet informative
    - use indents to show that something is a part of something
    - ...
- Tips and tricks
    - Alt/Option + - for assignment operator
    - type "?nameoffunction" to access help center
    - use "tab" for autocompletion
    - use arrow-keys to navigate previous commands
    - Cmd/Ctrl + Shift + C to command out (several) lines
    - ...

If-else statements

- Let R perform an action based on if a condition is met or not
- Syntax:
    - if (this is true) {do this}
    - else if (an alternative is true) {then do this}
    - else (when none of the above is true) {then do this}

If-else statements

- Simplest example

```
x
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
# creating multiple condition statement
if(10 %in% x) {
  print("x is equal to 10")
} else if(20 %in% 10) {
    print("x is equal to 20")
} else if( x < 10 & x >= 0) {
    print("x is a number equal or greater than 0 and smaller than 10")
} else {
    print("the conditions are not med for x")
}
```

```
## [1] "x is a number equal or greater than 0 and smaller than 10"
```

NOTE: "if" only takes ONE logical value

If-else statements

- using "ifelse", we access the entire vector, element-by-element:

```
x
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
ifelse(x >= 6, "TRUE", "FALSE")
```

```
## [1] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "TRUE"  "TRUE"  "TRUE"  "TRUE"
```

Yet, another coffee break? (12:15-12:30)

Recap

- writing scripts
- data handling
- if-else statements

Questions in break-out sessions

Exercises

- set up a project (and check your working directory through the console)
- write name of author and date of latest access
- comment where necessary
- assign a variable (vector and matrix)
- access an index from the vector amd matrix
- plot a histogram from your vector
- plot a boxplot and histogram from the penguin data
    - choose interesting variables to analyse
    - include title and axis-lables
    - say something about what you see
- send the code to your peers for replication

That's all folks!

Any burning questions?