

# R FOR BEGINNERS



Image credits:[Kelly Sikkema](#)



Dr Ajay Kumar Koli  
Assistant Professor  
Department of Travel & Tourism  
Vishwakarma University - Pune  
✉ [@ajay\\_kolii](https://twitter.com/@ajay_kolii)  
✉ [koliajaykumar@gmail.com](mailto:koliajaykumar@gmail.com)  
🌐 <https://koliajay.netlify.app/>

Hello! 😊

# Modules:

1. [Basics of R & RStudio](#)
2. [Dynamic Documents using R Markdown](#)
3. [Data Visualisation using ggplot2](#)
4. [Data Wrangling using dplyr](#)
5. [Slide Crafting using xaringan](#)

A close-up photograph of a white skein of yarn and a blue knitting needle. The skein is a tight coil of white, textured yarn. A blue wooden knitting needle lies next to it, with its tip pointing towards the bottom left. The background is a dark, textured surface, possibly denim or a similar fabric.

# R and RStudio: Basics

SESSION - 1

Image credits:[Kelly Sikkema](#)





...but now it's like...



Source



# R is FREE

- R is a language and environment for statistical computing and graphics. (R project)
- In August 1993, designed by



Ross Ihaka  
(New Zealand Statistician)



Robert Gentleman  
(Canadian Statistician)

# Download R from CRAN

## CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main page](#), [windows release](#), [windows old release](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

### 0-Cloud

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

Automatic redirection to servers worldwide, currently sponsored by Rstudio

### Algeria

<https://cran.usthb.dz/>

University of Science and Technology Houari Boumediene

### Argentina

<http://mirror.fcaglp.unlp.edu.ar/CRAN/>

Universidad Nacional de La Plata

### Australia

<https://cran.csiro.au/>

CSIRO

<https://mirror.aarnet.edu.au/pub/CRAN/>

AARNET

<https://cran.ms.unimelb.edu.au/>

School of Mathematics and Statistics, University of Melbourne

<https://cran.curtin.edu.au/>

Curtin University

### Austria

<https://cran.wu.ac.at/>

Wirtschaftsuniversität Wien

### Belgium

<https://www.freestatistics.org/cran/>

Patrick Wessa

<https://ftp.belnet.be/mirror/CRAN/>

Belnet, the Belgian research and education network

### Brazil

<https://nbcgib.uesc.br/mirrors/cran/>

Computational Biology Center at Universidade Estadual de Santa Cruz

<https://cran-r.c3sl.ufpr.br/>

Universidade Federal do Parana

<https://cran.fiocruz.br/>

Oswaldo Cruz Foundation, Rio de Janeiro

<https://vps.fmvz.usp.br/CRAN/>

University of Sao Paulo, Sao Paulo

<https://briege.rsalq.usp.br/CRAN/>

University of Sao Paulo, Piracicaba

### Bulgaria

<https://ftp.uni-sofia.bg/CRAN/>

Sofia University

### Canada

<https://mirror.rcg.sfu.ca/mirror/CRAN/>

Simon Fraser University, Burnaby

<https://muug.ca/mirror/cran/>

Manitoba Unix User Group

<https://mirror.its.dal.ca/cran/>

Dalhousie University, Halifax

<https://utstat.toronto.edu/cran/>

University of Toronto

Chile

# R Console

- R version
- R name
- R licence
- prompt >

# Never Save R "Workspace Image":

- It helps in "freshly minted R sessions".
- "put more trust in your script than in your memory"

[Source](#)

# R as a BIG calc

WHAT YOU CODE

```
1
```

WHAT YOU SEE

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

# R as a BIG calc

WHAT YOU CODE

```
1  
1 + 1
```

WHAT YOU SEE

```
## [1] 1  
## [1] 2
```

# R as a BIG calc

WHAT YOU CODE

```
1  
1 + 1  
34 / 40
```

WHAT YOU SEE

```
## [1] 1  
## [1] 2  
## [1] 0.85
```

# R as a BIG calc

WHAT YOU CODE

```
1  
1 + 1  
34 / 40  
5 < 4
```

WHAT YOU SEE

```
## [1] 1  
## [1] 2  
## [1] 0.85  
## [1] FALSE
```

# R as a BIG calc

WHAT YOU CODE

```
1  
1 + 1  
34 / 40  
5 < 4  
16 == 16
```

WHAT YOU SEE

```
## [1] 1  
## [1] 2  
## [1] 0.85  
## [1] FALSE  
## [1] TRUE
```

# Plot using R

---

Code

---

Output

```
plot(1:100)
```

# Plot using R

Code

Output



# Functions

# R Function

- "A function, in a programming environment, is a set of instructions. A programmer builds a function to avoid repeating the same task, or reduce complexity."

---

Code

---

Output

---

```
round(9.1565, 2)
```

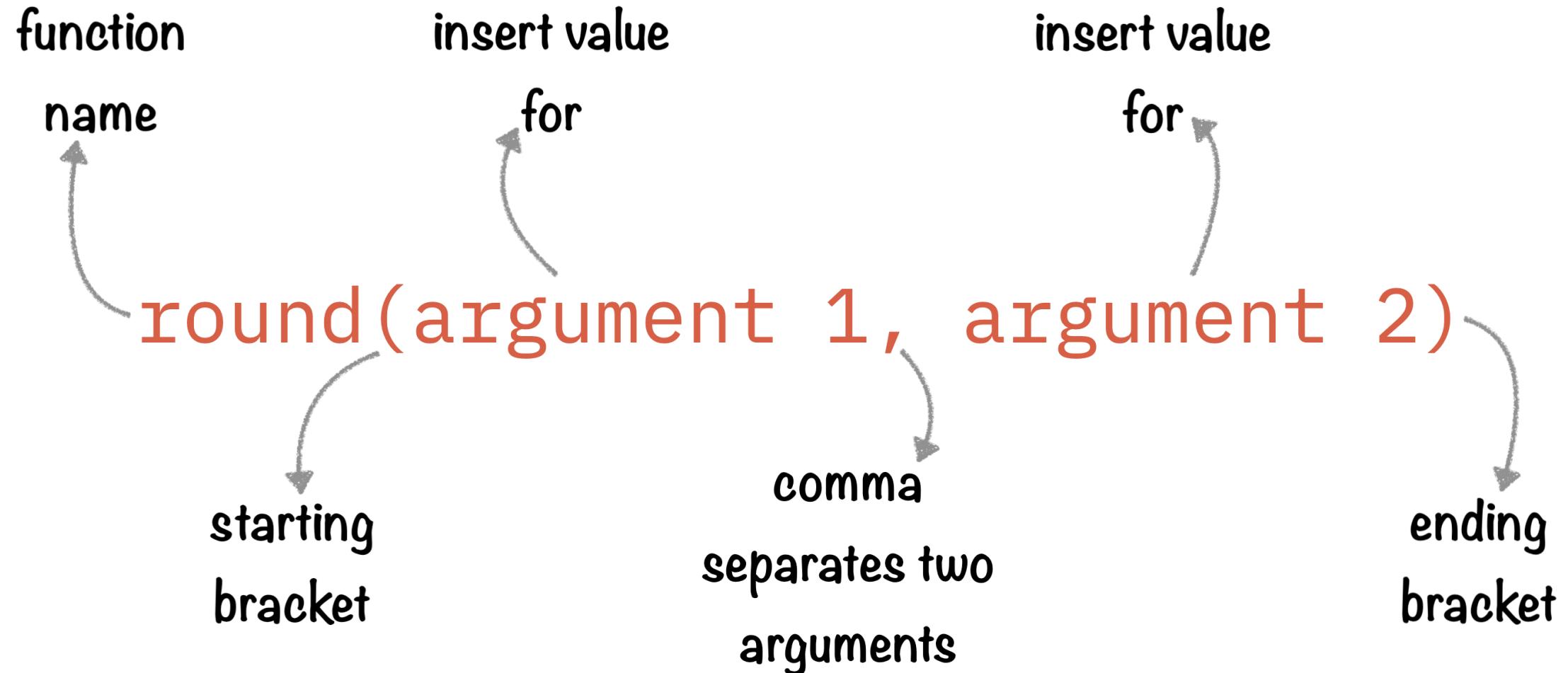
# R Function

- "A function, in a programming environment, is a set of instructions. A programmer builds a function to avoid repeating the same task, or reduce complexity."

Code      Output

```
## [1] 9.16
```

# Structure of R function



# Round Function

---

Code

Output

---

```
round(x = 564.56743, digits = 2)
```

# Round Function

Code

Output

```
## [1] 564.57
```

# Round Function

---

Code

Output

---

```
round(x = 564.56743, digits = 1)
```

# Round Function

Code

Output

```
## [1] 564.6
```

# Square Root Function

---

Code

Output

---

```
sqrt(x = 9)
```

# Square Root Function

Code

Output

```
## [1] 3
```

# Sequence Function

---

Code

Output

---

```
seq.int(from = 10, to = 30, by = 5)
```

# Sequence Function

Code

Output

```
## [1] 10 15 20 25 30
```



Comments

# R Comment:

- "Humans will be able to read the comments, but your computer will pass over them."<sup>1</sup>
- In R, `#` is used as a commenting symbol

# How to add comment?

Code

Output

```
# secret code of happiness  
(1 + 4) - (3 * 5) / 10  
## r does not evaluate this  
# all these are comments  
# comments are very important
```

# How to add comment?

Code

Output

```
## [1] 3.5
```



# *That's okay but you promise to...*



- combine plot, text, tables and images in a single file.
- publish my work online or convert into a word, pdf or html file.
- work efficiently with my different projects and save, share and track them.

WE NEED A SUPERHERO ...



Studio<sup>®</sup>

The word "Studio" is written in a large, bold, dark gray sans-serif font. A registered trademark symbol (®) is positioned in the top right corner of the letter 'o'.



# About RStudio:

- 2009, Joseph J. Allaire founded RStudio.
- 2011, RStudio IDE for R was launched.
- "RStudio is dedicated to sustainable investment in free and open-source software for data science." (16 Jan 2022)



[Source](#)

# R \rightleftharpoons RStudio

Imagine RStudio as a  
stylish car ...

and R as its powerful  
engine.



# RStudio IDE

RSTUDIO \(\rightarrow\) TOOLS \(\rightarrow\) GLOBAL OPTIONS



RSTUDIO \(\rightarrow\) TOOLS \(\rightarrow\) GLOBAL OPTIONS





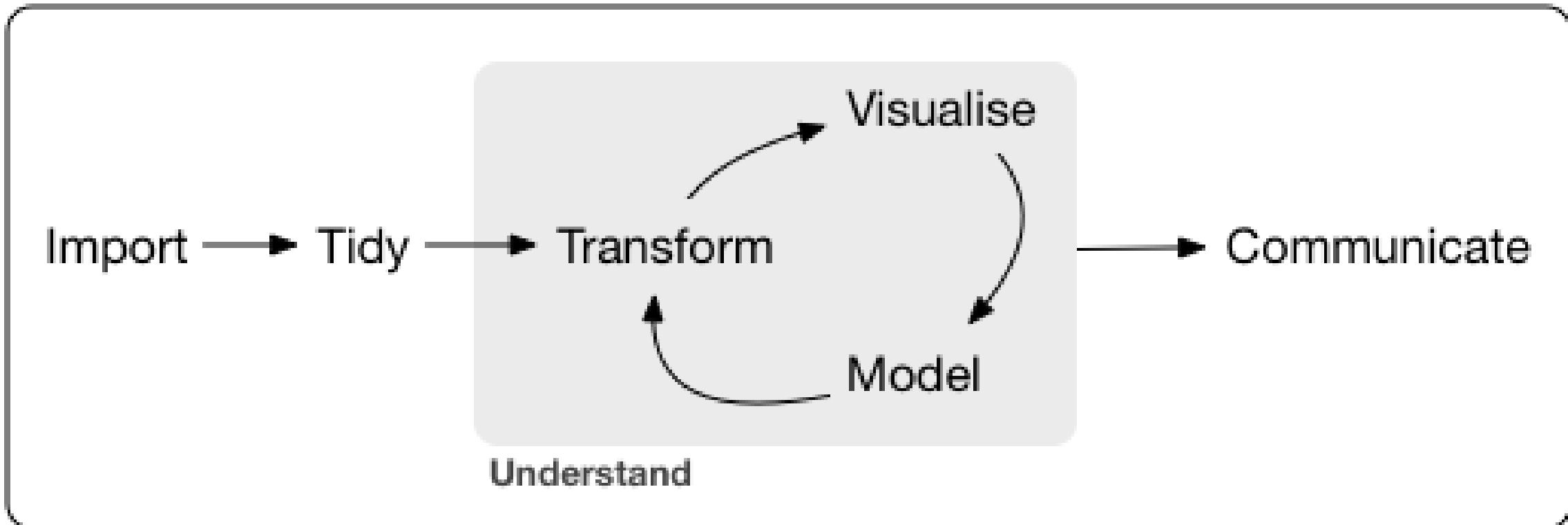
## We all want to learn.

No one is born a data scientist. Every person who works with R today was once a complete beginner. No matter how much you know about the R ecosystem already, you'll always have more to learn.

At RStudio, we know that everyone is at a different stage in learning the vast ecosystem of R, ranging from rank beginner to seasoned data scientists to professional educators. Choose a starting point below, or catch up with us on our [blog](#).

[Source](#)

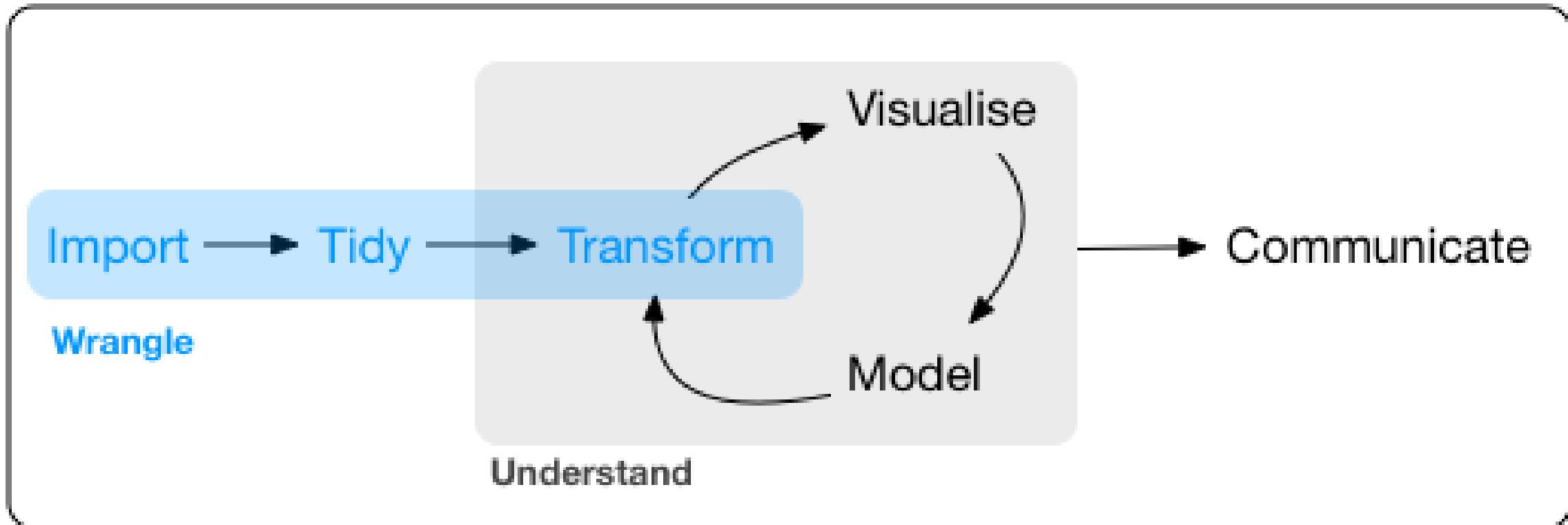
# R Program



Program

[Source](#)

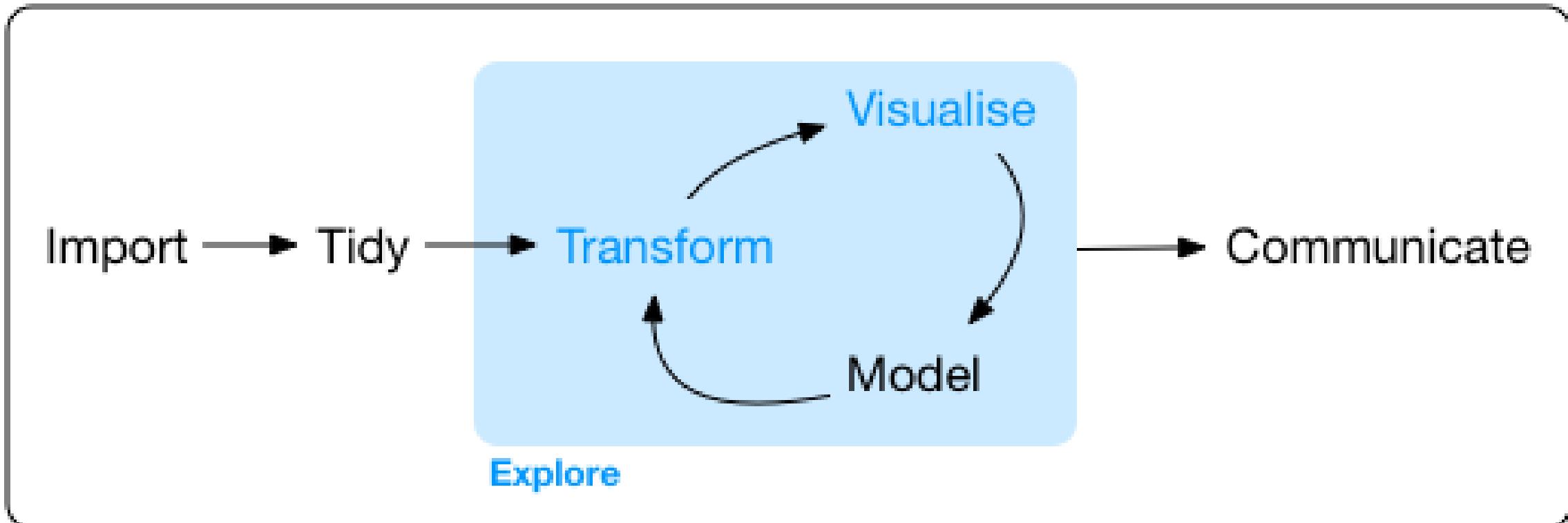
# Data Wrangling



Program

[Source](#)

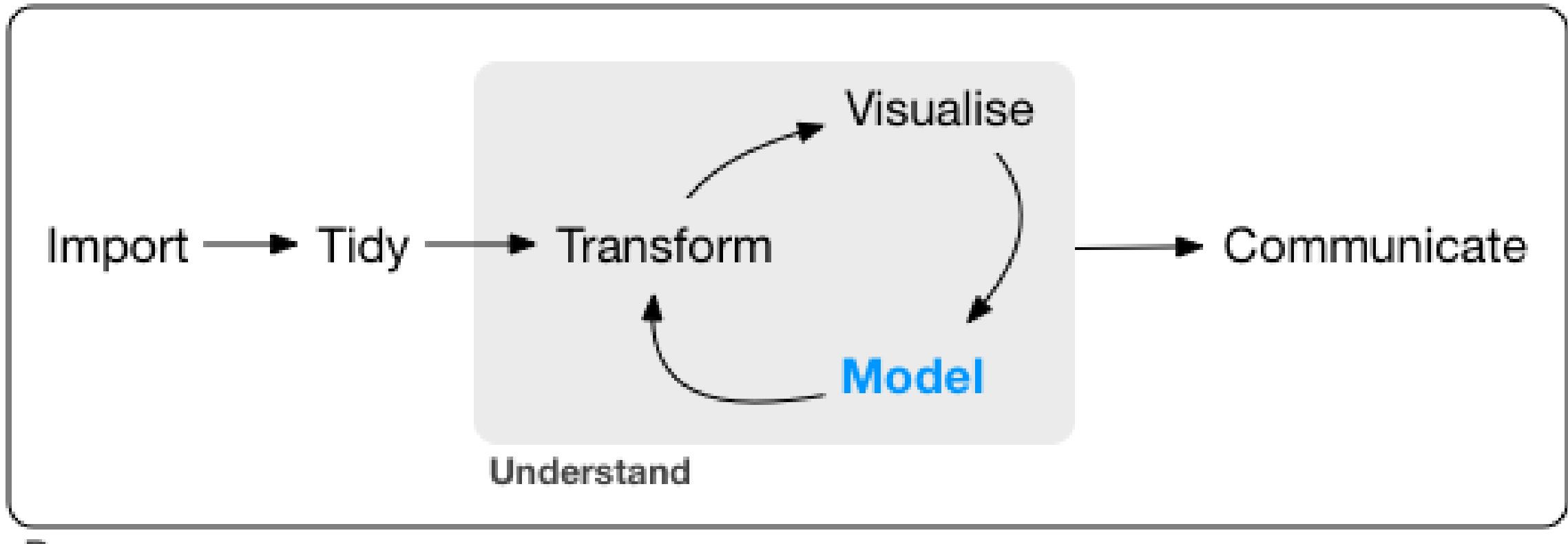
# Exploratory Data Analysis



Program

[Source](#)

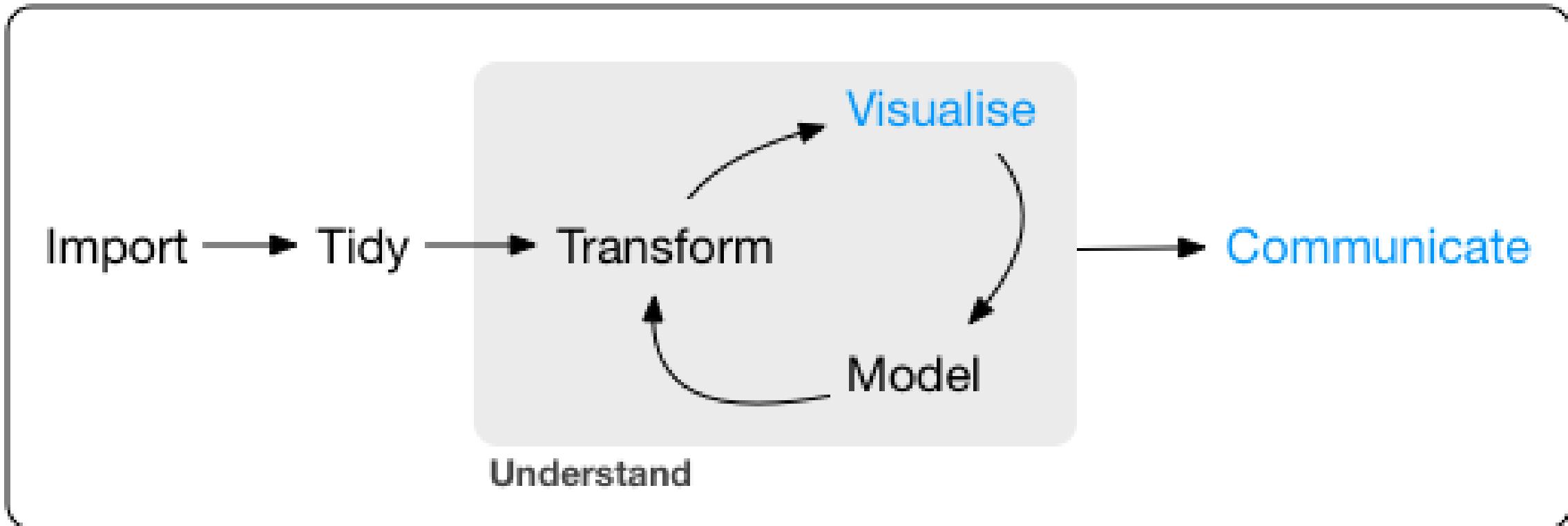
# Modeling



Program

[Source](#)

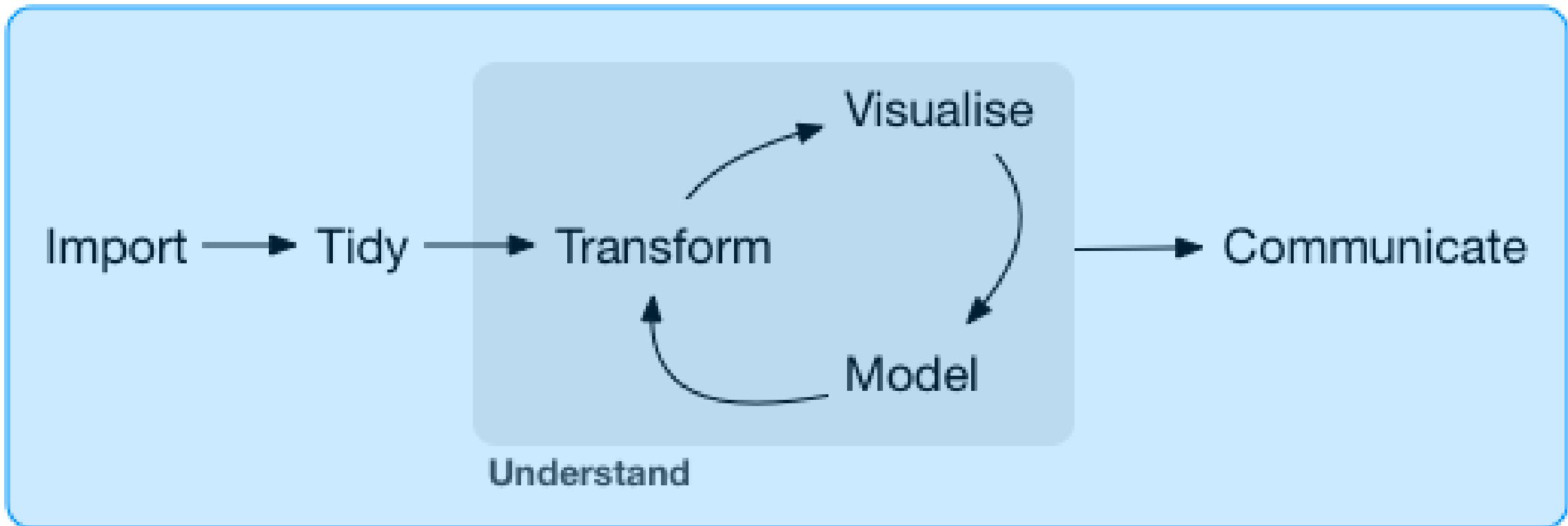
# Data Visualisation



Program

Source

# R Program



Program

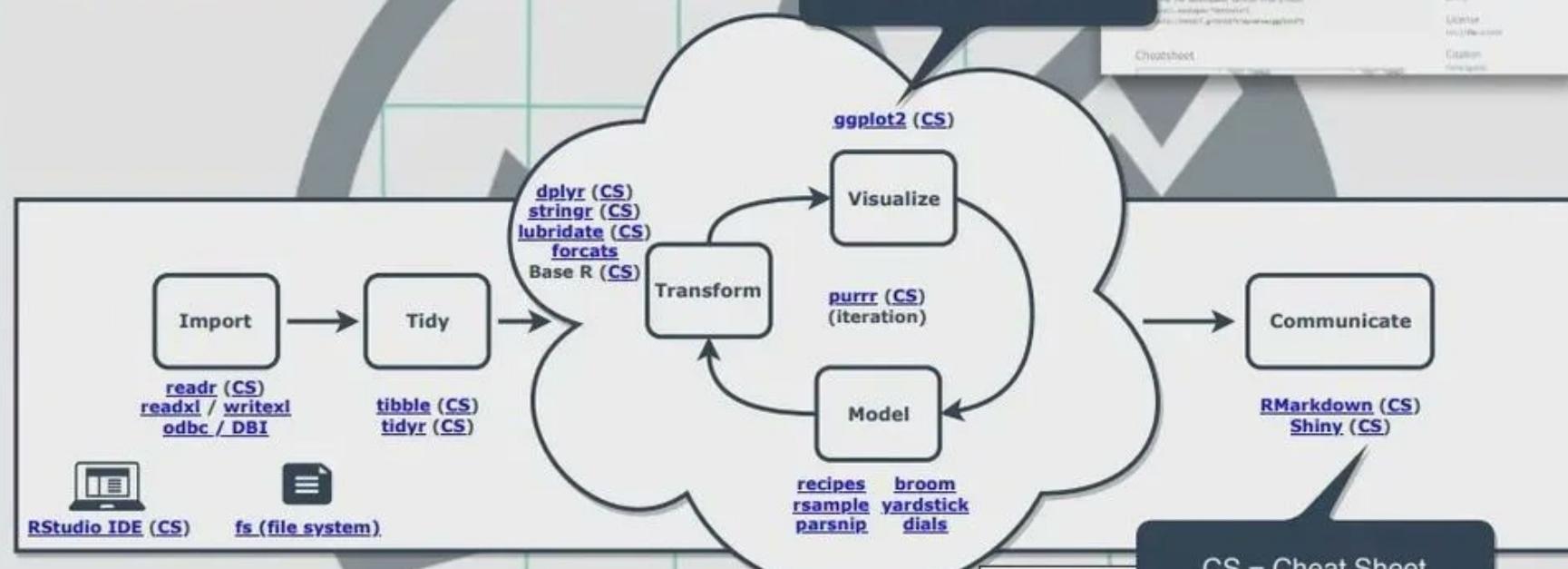
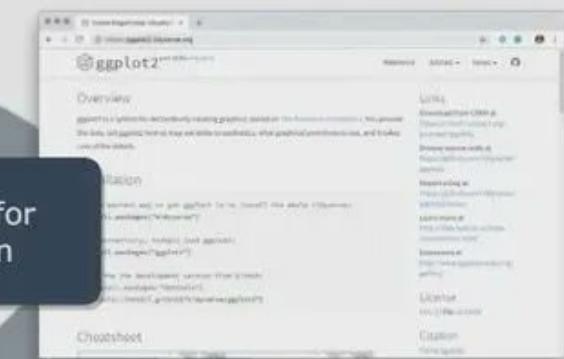
Source

# Data Science with R Workflow

The Data Science With R Workflow is available in the book: [R For Data Science](#). If you want to learn R and this workflow **for business**, take the [R For Business Analysis \(DS4B 101-R\) course](#) through Business Science University.



Click the links for Documentation



## Important Resources

- R For Data Science Book: <http://r4ds.had.co.nz/>
- Rmarkdown Book: <https://bookdown.org/yihui/rmarkdown/>
- Data Visualization Book: <https://rkbacoff.github.io/datavis/>
- More Cheatsheets: <https://www.rstudio.com/resources/cheatsheets/>
- tidyverse packages: <https://www.tidyverse.org/>
- Connecting to databases: <https://db.rstudio.com/>
- RMarkdown website: <https://rmarkdown.rstudio.com/>
- Shiny web applications website: <http://shiny.rstudio.com/>
- Jenny Bryan's purrr tutorial: <https://jennybryan.org/>

[Source](#)



# R Packages



# R Packages:

- On 12 Jan 2022, 18698 R packages were available at [CRAN](#)
- "An R package is a collection of functions, data, and documentation that extends the capabilities of base R. Using packages is key to the successful use of R."
- Top downloaded packages [source](#)

[Source](#)

# To Download pkgs

# Name of the R package(s)



# Installed R package(s)

# R Function to Download Package

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

# R Function to use Package

```
library(tidyverse)
```

# About R Packages:

- You need to install package only once like
  -  We buy books once and use them again and again
  -  Fix the bulb once and use it again and again
- In every R document you need to **call** once the package using function **library()**, for example `library(ggplot2)`.
- Once in a while, you need to update the installed packages as well.
- If you un-install R or RStudio, you will lose all installed packages.

# Tools \(\rightarrow\) Check Package Updates



# Select Package(s) to Update

# Click Install Updates

# To Remove Package(s)

Name	Description	Version	Action
<b>System Library</b>			
abind	Combine Multidimensional Arrays	1.4-5	
afex	Analysis of Factorial Experiments	0.28-1	
airports	Data on Airports	0.1.0	
amap	Another Multidimensional Analysis Package	0.8-18	
AsioHeaders	'Asio' C++ Header Files	1.16.1-1	
askpass	Safe Password Entry for R, Git, and SSH	1.1	
assertthat	Easy Pre and Post Assertions	0.2.1	
av	Working with Audio and Video in R	0.5.1	
babynames	US Baby Names 1880-2017	1.0.0	
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.2.1	
<input checked="" type="checkbox"/> base	The R Base Package	4.0.2	
base64enc	Tools for base64 encoding	0.1-3	
BayesFactor	Computation of Bayes Factors for Common Designs	0.9.12-4.2	
bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.8.2	
bbmle	Tools for General Maximum Likelihood Estimation	1.0.23.1	
BDgraph	Bayesian Structure Learning in Graphical Models using Birth-Death MCMC	2.63	
bdsmatrix	Routines for Block Diagonal Symmetric Matrices	1.3-4	



# YOUR TURN

05 : 00

Task

Answer

- Install R packages `xaringan` & `palmerpenguins`
- Call R packages using console



# YOUR TURN

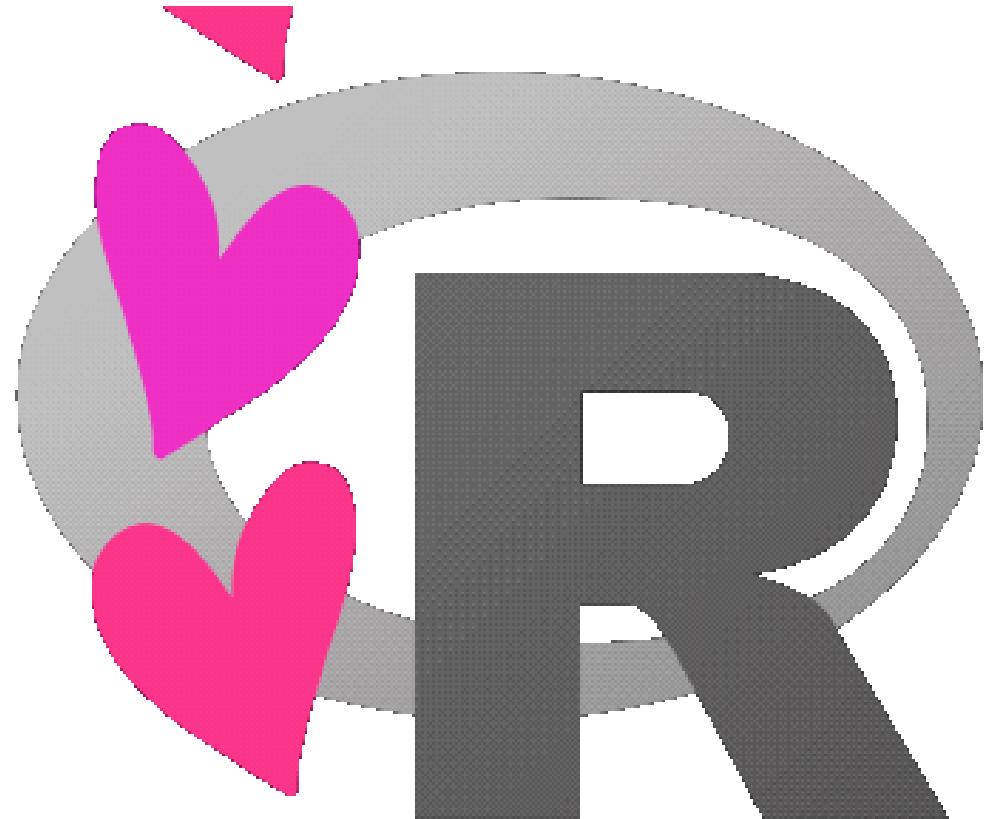
05 : 00

Task

Answer

```
install.packages(c("xaringan", "palmerpenguins"))

library(xaringan)
library(palmerpenguins)
```



# Objects

# R Object

- "Just a name that you can use to call up stored data"

# Create Object

---

Code

Output

---

```
salary <- c(20, 30, 40, 50, -10)  
salary
```

# Create Object

Code

Output

```
## [1] 20 30 40 50 -10
```

# Create Object

---

Code

Output

---

```
name <- c("Ram", "Rani", "Ali", "Preeti", "John")
name
```

# Create Object

Code

Output

```
## [1] "Ram"     "Rani"    "Ali"     "Preeti"   "John"
```

# Create Object

---

Code

Output

---

```
age <- c(34, 54, 23, 65, 2 )  
age
```

# Create Object

Code

Output

```
## [1] 34 54 23 65 2
```

# Create Object

---

Code

Output

---

```
place <- c("ny", "ber", "dhl", "tko", "lon")
place
```

# Create Object

Code

Output

```
## [1] "ny"  "ber" "dhl" "tko" "lon"
```

# Create Object

---

Code

---

Output

```
books <- c(4, 0, 3, 24, 5)
books
```

# Create Object

Code

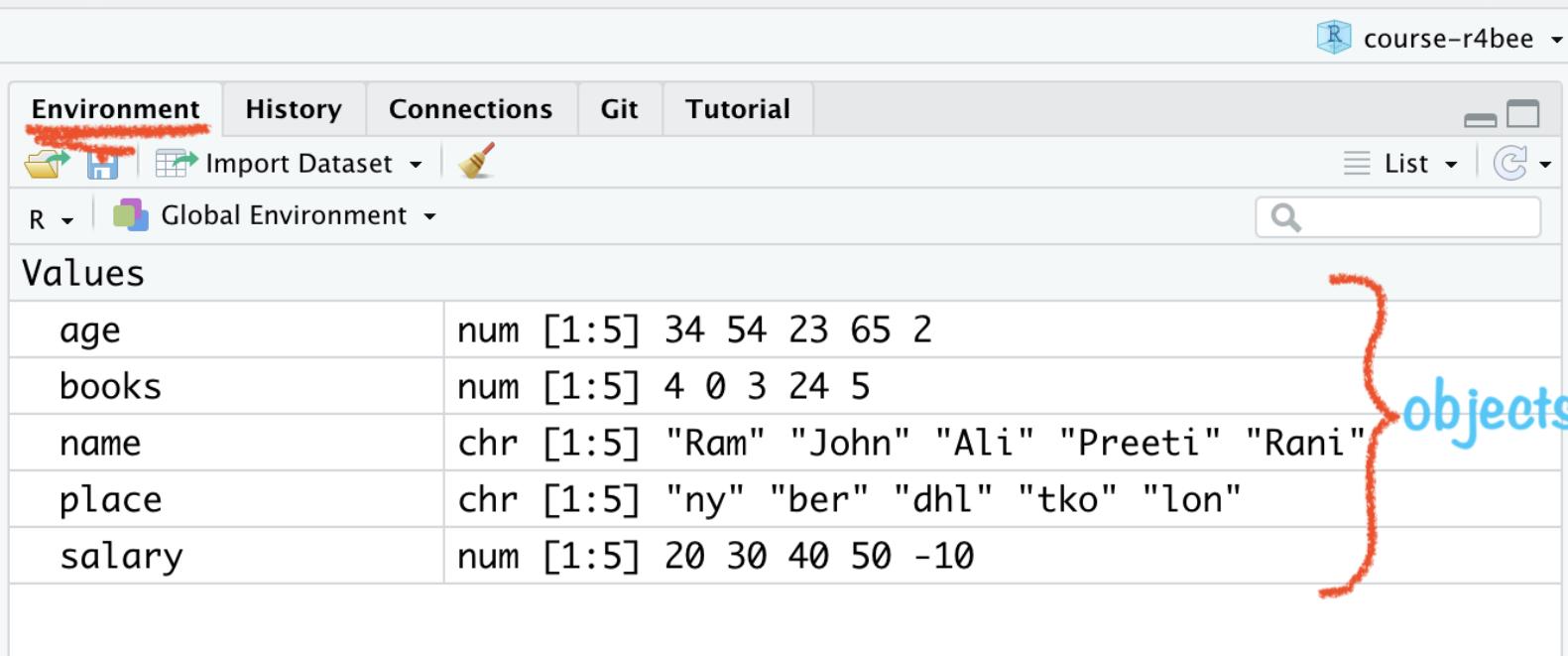
Output

```
## [1] 4 0 3 24 5
```

# Guidelines to name objects in R:

- a name cannot start with a number
- a name cannot use some special symbols, like ^, !, \$, @, +, -, /, or \*:
- avoid caps
- avoid space
- use dash (like na-me) or underscore (like na\_me)
- if chronology matters then add date (2020-09-05-file-name)

# RStudio Environment Window



The screenshot shows the RStudio Environment window with the title bar "course-r4bee". The "Environment" tab is selected. Below it, there are buttons for "Import Dataset" and "Global Environment". The main area displays a table titled "Values" with five rows:

	Values
age	num [1:5] 34 54 23 65 2
books	num [1:5] 4 0 3 24 5
name	chr [1:5] "Ram" "John" "Ali" "Preeti" "Rani"
place	chr [1:5] "ny" "ber" "dhl" "tko" "lon"
salary	num [1:5] 20 30 40 50 -10

A red curly brace is drawn around the last four columns of the table, labeled "objects" in blue.

🤔 *how to combine these  
objects/variables into a data or say tidy data*

# Tidy data



```
##   age books   name place salary
## 1 34     4    Ram   ny     20
## 2 54     0   Rani  ber     30
## 3 23     3    Ali  dhl     40
## 4 65    24 Preeti tko     50
## 5  2     5   John lon    -10
```

variables

country	year	cases	population
Afghanistan	1990	745	187071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	1272915272
China	2000	216766	1280426583

observations

country	year	cases	population
Afghanistan	1990	745	187071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	1272915272
China	2000	216766	1280426583

values

Afghanistan	1990	745	187071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	1272915272
China	2000	216766	1280426583

[Source](#)

# How to create a data object?

---

Code

Output

---

```
social <- data.frame(age, books, name, place, salary)  
social
```

# How to create a data object?

Code

Output

```
##   age books    name place salary
## 1  34     4      Ram   ny    20
## 2  54     0     Rani  ber    30
## 3  23     3      Ali   dhl    40
## 4  65    24  Preeti  tko    50
## 5   2     5     John  lon   -10
```

# How to export data as a csv file?

---

Code

Output

---

```
library(readr)
# to save this data set as a csv file
write_csv(social, "data/social.csv")
```

# How to export data as a csv file?

Code

Output



# YOUR TURN

10:00

Write codes for below dataframe

---

Task	Answer
------	--------

##		state	pop	capital	foundation
## 1	Germany	20	Berlin	1870-12-10	
## 2	France	19	Paris	1789-07-14	
## 3	India	50	Delhi	1947-08-15	
## 4	Russia	25	Moscow	1990-06-12	
## 5	USA	30	Washington	1776-07-04	
## 6	New Zealand	5	Wellington	1840-02-06	



# YOUR TURN

10:00

Write codes for below dataframe

Task      Answer

```
state <- c("Germany", "France", "India", "Russia", "USA", "New Zealand")
pop <- c(20, 19, 50, 25, 30, 5)
capital <- c("Berlin", "Paris", "Delhi", "Moscow", "Washington", "Wellington")
foundation <- c("1870-12-10", "1789-07-14", "1947-08-15", "1990-06-12", "1776-07-04", "1854-11-15")
world <- data.frame(state, pop, capital, foundation)
world
```

# Get a List of all Objects

---

Code

Output

---

```
# names of created objects  
objects()
```

# Get a List of all Objects

Code

Output

```
## [1] "age"          "books"        "capital"      "foundation"  
## [5] "input"        "name"         "output_file" "place"  
## [9] "pop"          "salary"       "soc"          "social"  
## [13] "state"        "world"
```

Need  
Help!



# Using Console >

in console type ?your query

for example ?ggplot

# RStudio: pkg Help Docs

# RStudio Community

stackoverflow

# GitHub Issues

# Twitter #rstats



# Q&A

# Dynamic Documents Using R Markdown

NEXT MODULE - 2

# Modules:

1. Basics of R & RStudio
2. Dynamic Documents using R Markdown
3. Data Visualisation using ggplot2
4. Data Wrangling using dplyr
5. Slide Crafting using xaringan

# Dynamic Documents

SESSION - 2

Image credits:[John Anvik](#)





# Open RStudio



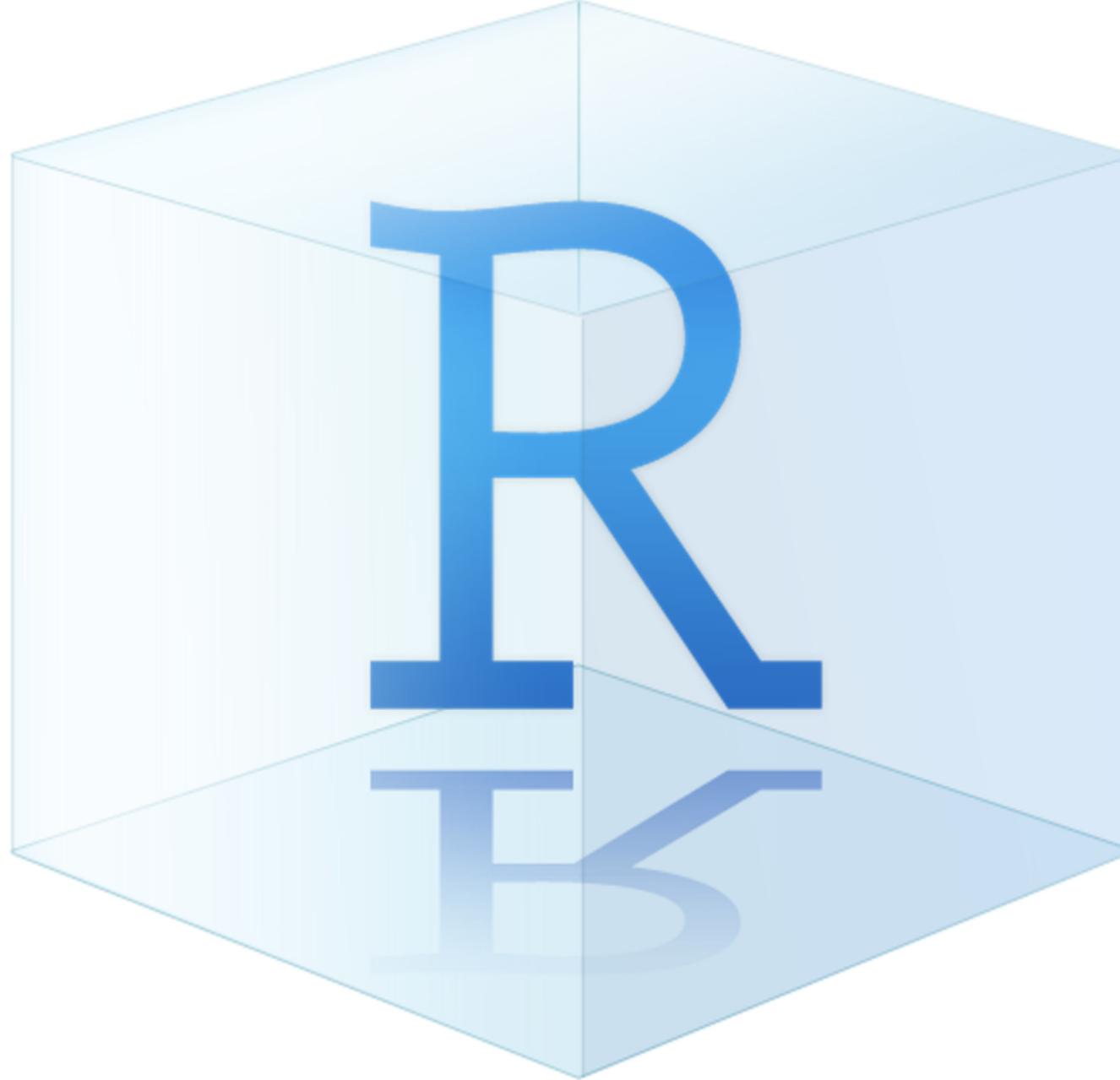
# Open RStudio



# Open RStudio

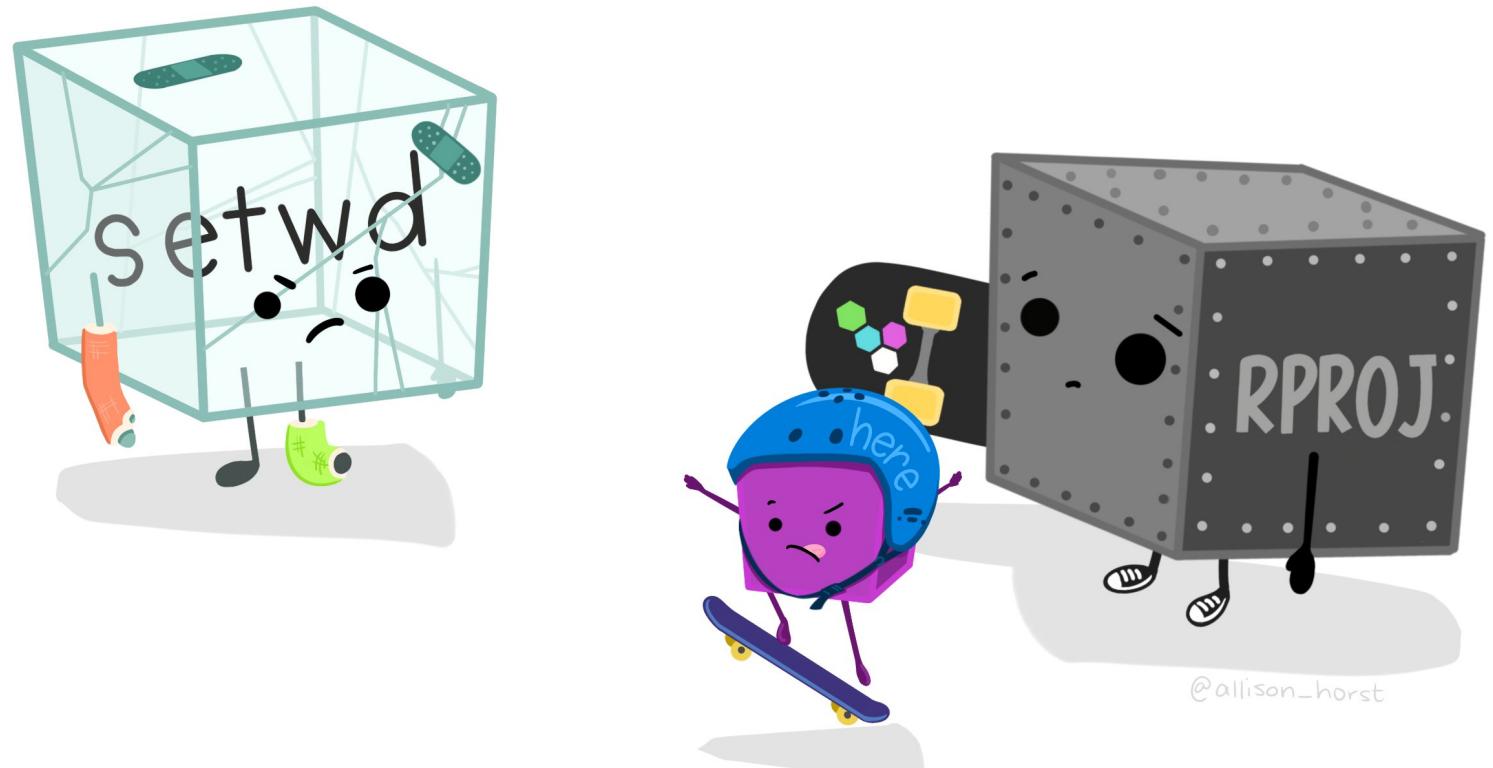


# RStudio Project



# About RStudio Projects

- "to divide your work into multiple contexts, each with their own:
  - working directory,
  - workspace,
  - history, and
  - source documents."



[Source](#) & Artwork [Source](#)

# 🔥 Create RStudio Project in 4 Steps 🔥

# Create RStudio Project in 4 Steps

# Open RStudio Project



# Open RStudio Project



# Open RStudio Project



# R Package



# About R Markdown:

- "You bring your data, code, and ideas, and R Markdown renders your content into a polished document that can be used to:
  - Do data science interactively within the RStudio IDE,
  - Reproduce your analyses,
  - Collaborate and share code with others, and
  - Communicate your results with others."



## What is R Markdown?

from RStudio, Inc.

# what is rmarkdown ?



01:12

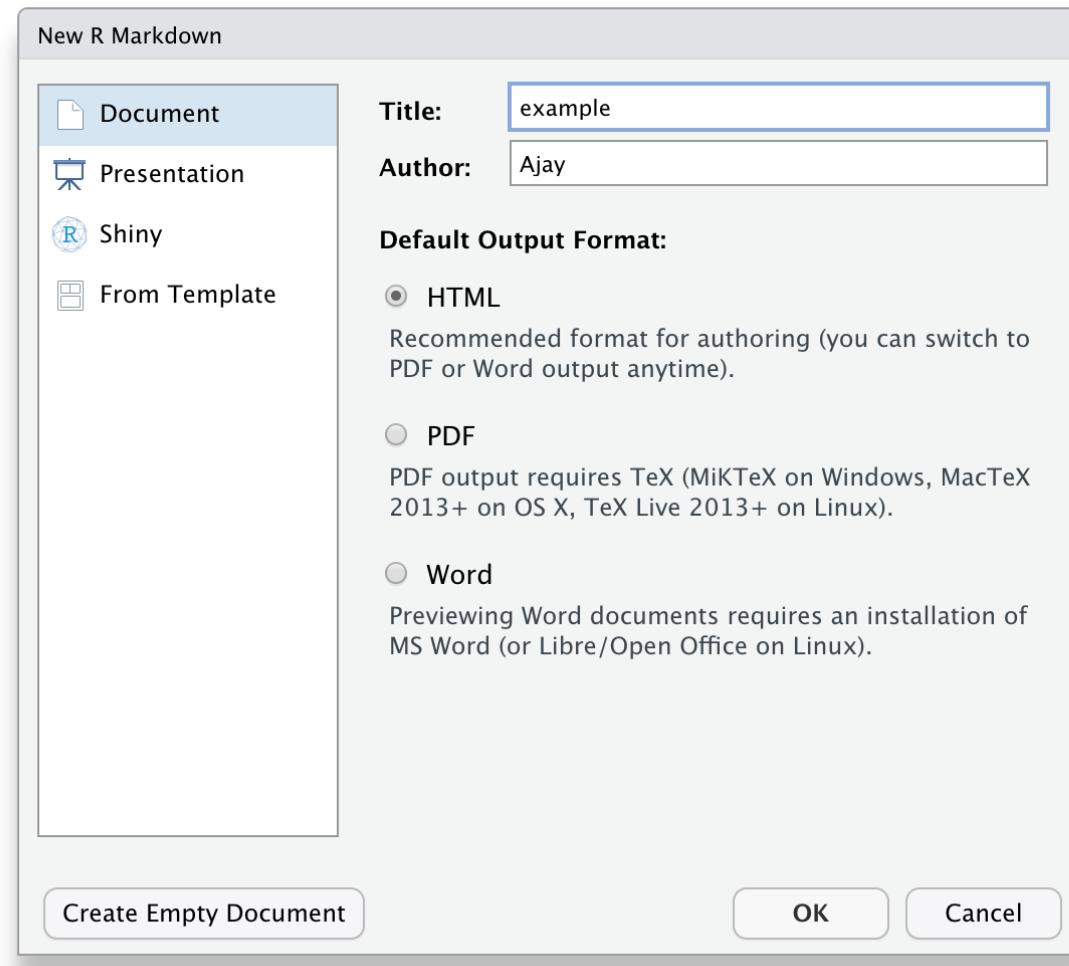


[What is R Markdown?](#) from [RStudio, Inc.](#) on [Vimeo](#).

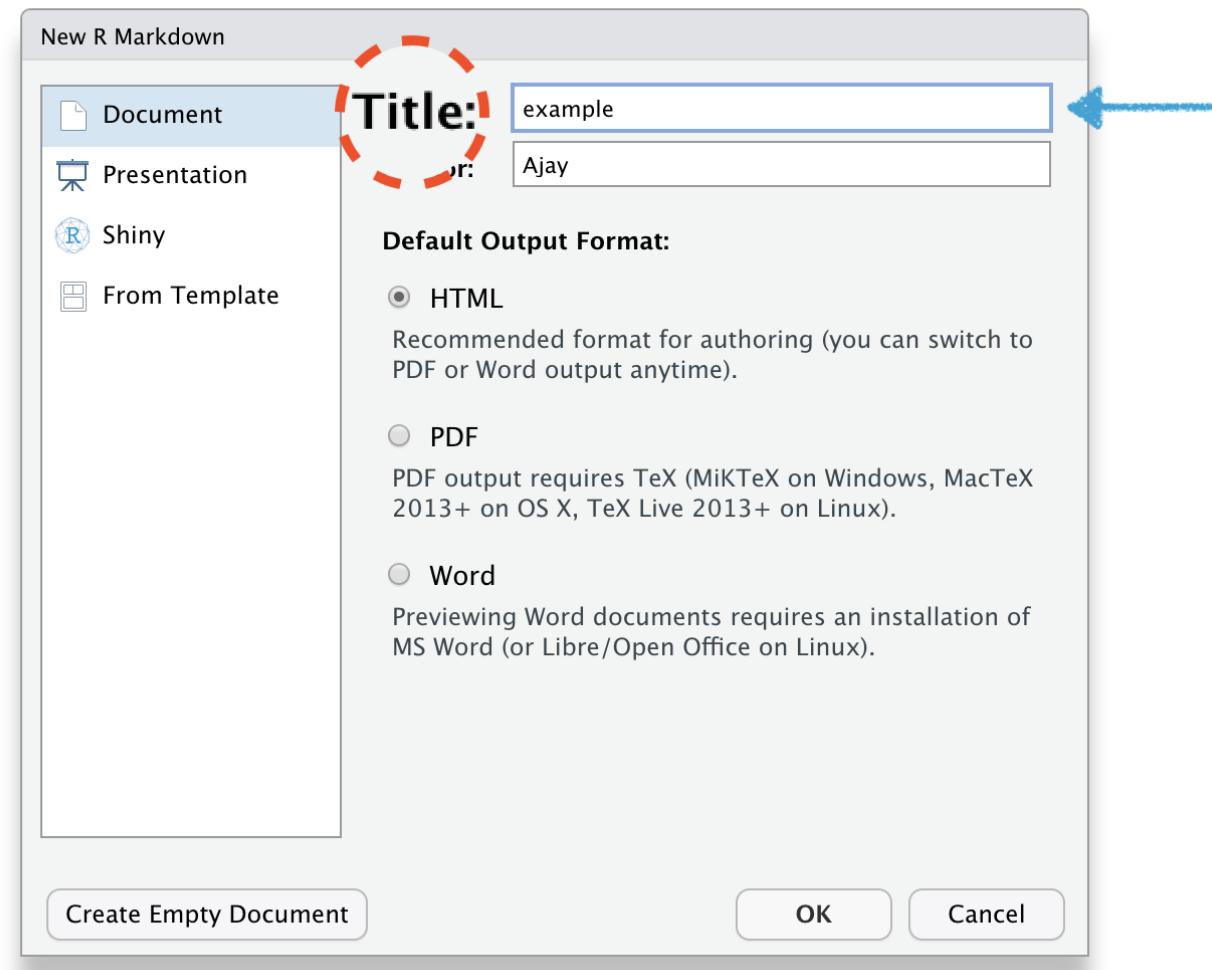
🔥 How to create R Markdown file? 🔥

File \(\rightarrow\) New File \(\rightarrow\) R Markdown

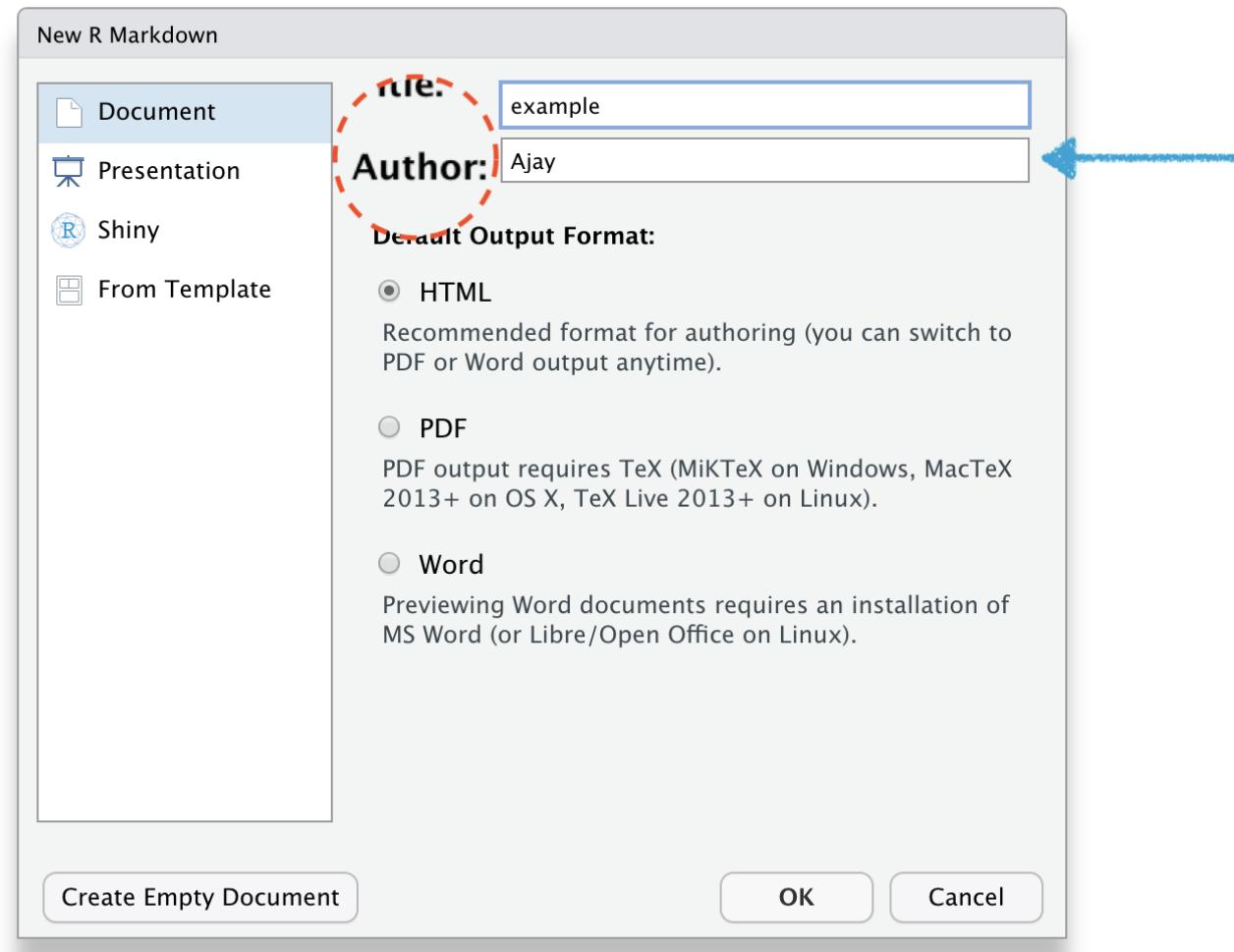
# R Markdown



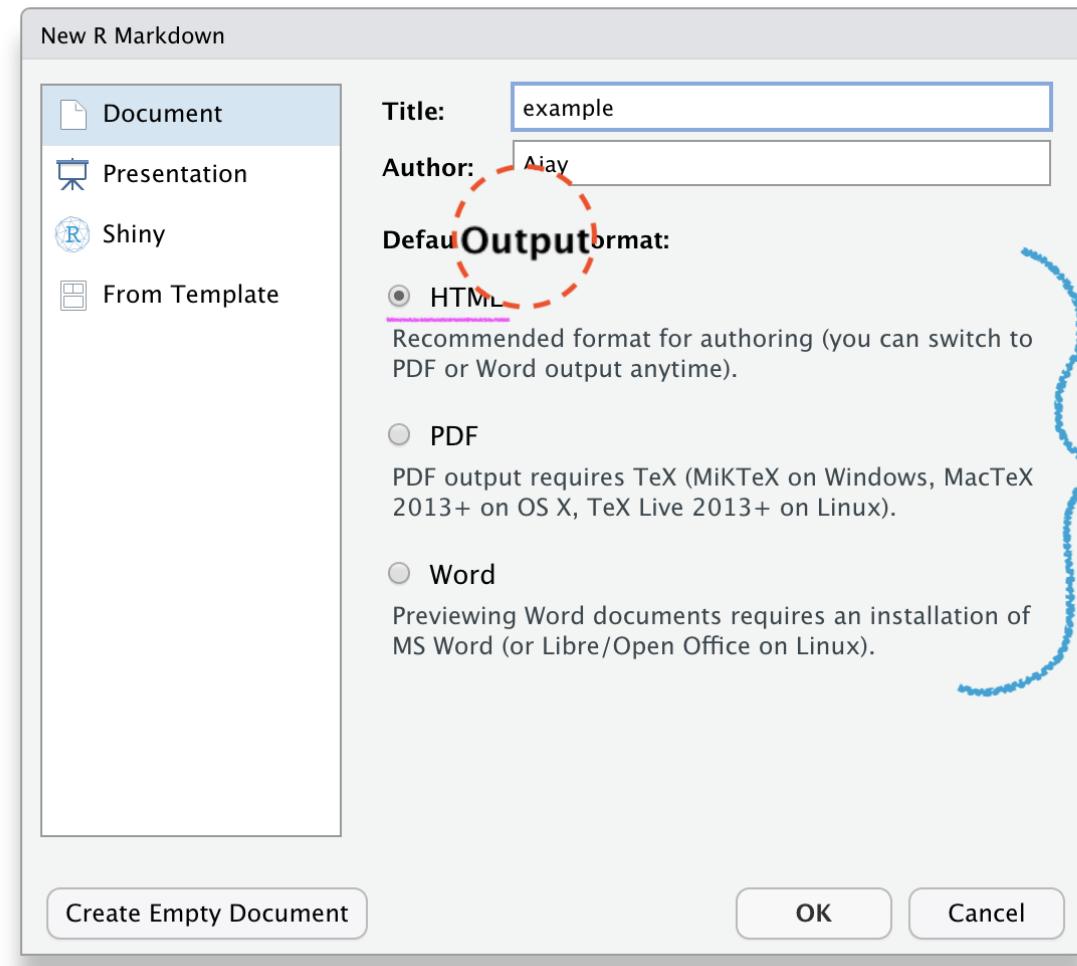
# R Markdown



# R Markdown

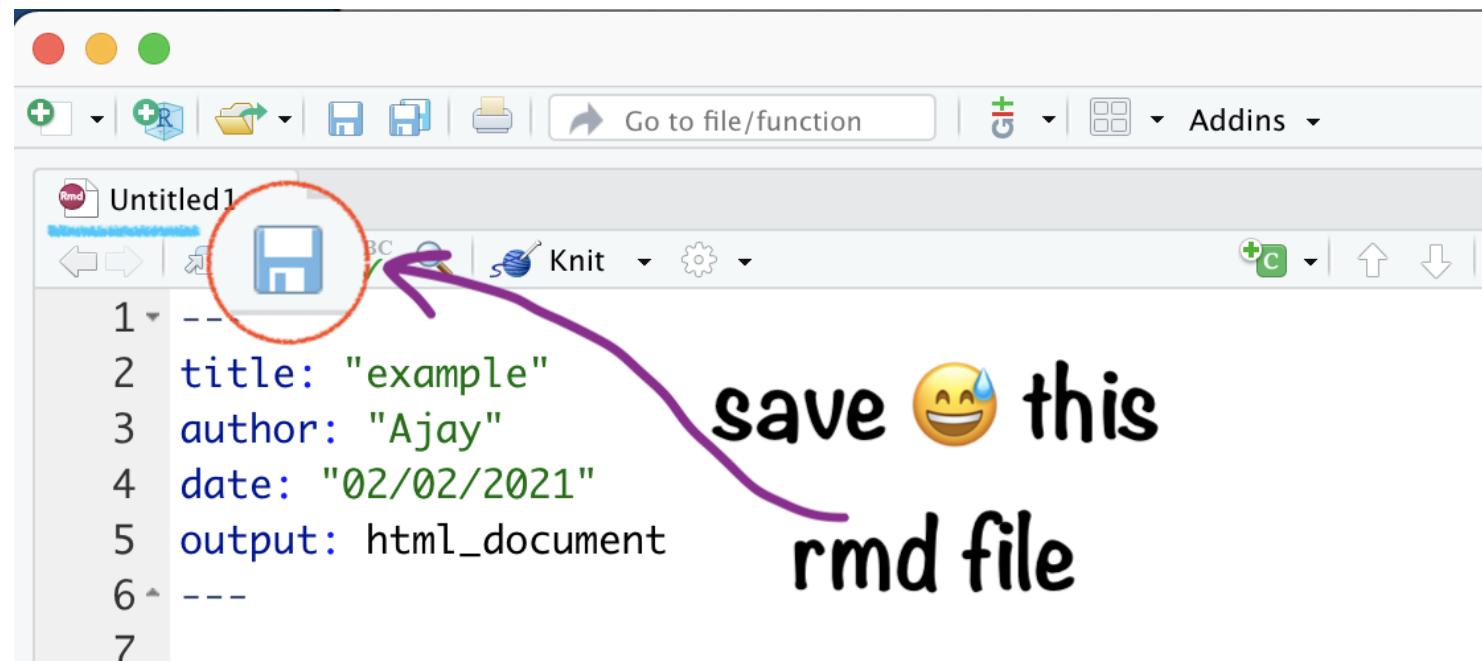


# R Markdown



start with  
HTML

# Save your .Rmd file



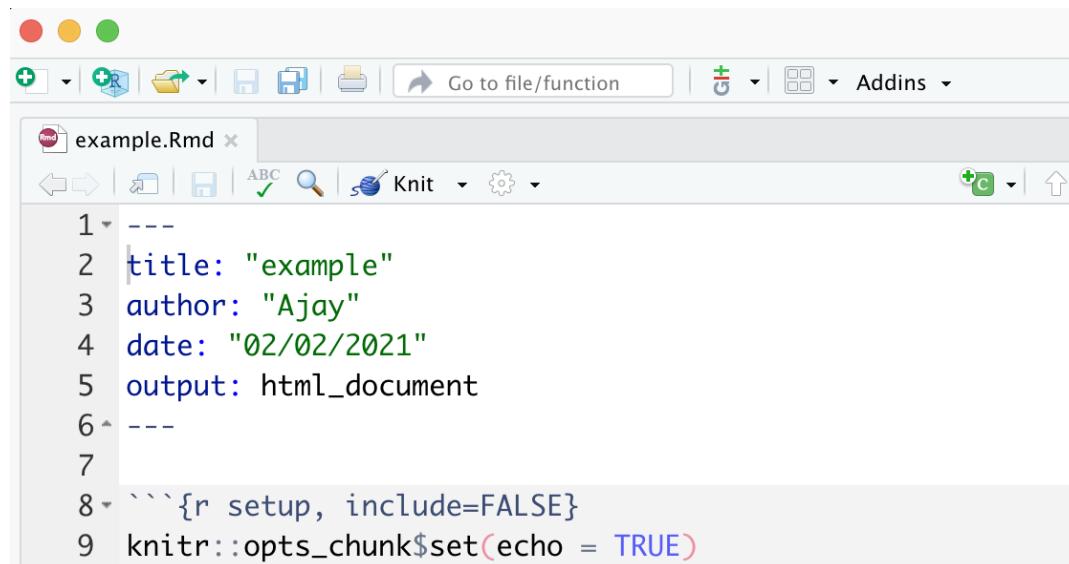
# Name your .Rmd file

# Name your .Rmd file

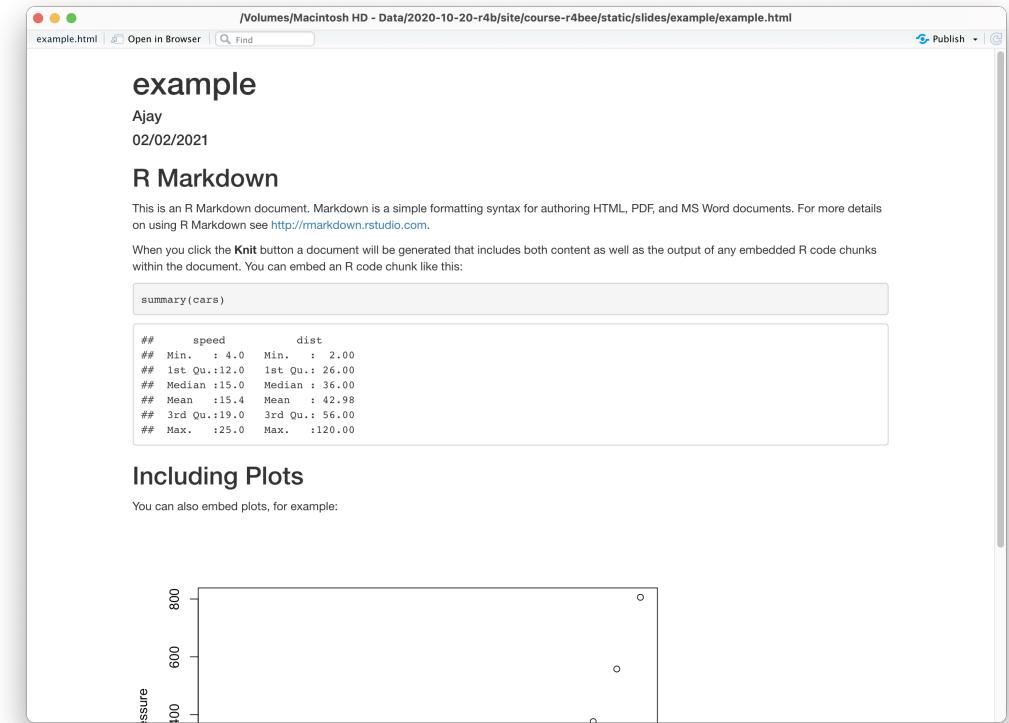
# Save your .Rmd file

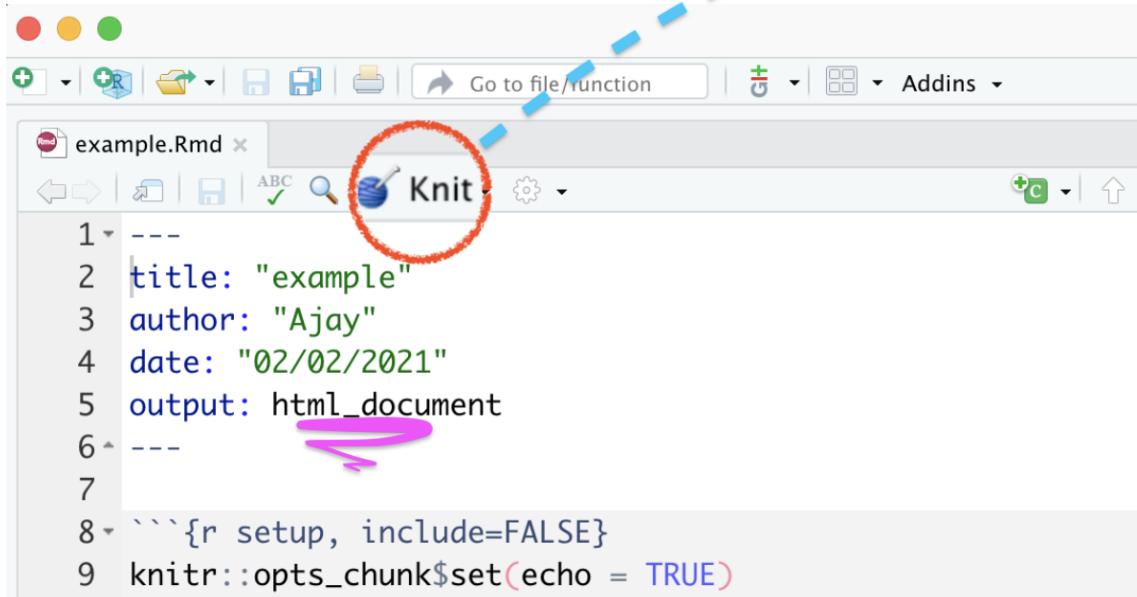
# Saved .Rmd file \(\rightarrow\) in RStudio Project





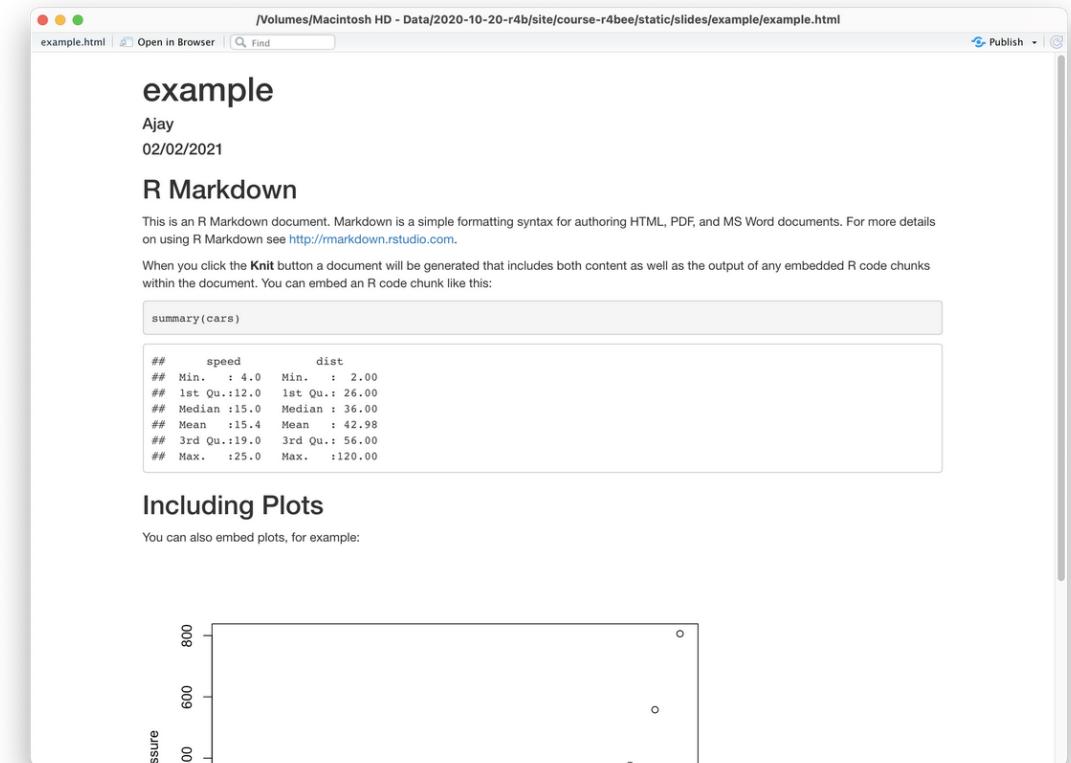
```
1 ---  
2 title: "example"  
3 author: "Ajay"  
4 date: "02/02/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)
```





```
1 ---  
2 title: "example"  
3 author: "Ajay"  
4 date: "02/02/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)
```

R markdown file



HTML output

# R Markdown has 3 important parts:

- YAML
- Code chunk
- Text

Untitled1 x

ABC Knit

Insert Run

```
1 ---  
2 title: "Untitled"  
3 author: "Ajay"  
4 date: "9/3/2020"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax  
for authoring HTML, PDF, and MS Word documents. For more details on using  
R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that  
includes both content as well as the output of any embedded R code chunks
```

22:19 # Including Plots R Markdown



Untitled1 x

ABC Knit Insert Run

```
1 ---  
2 title: "Untitled"  
3 author: "Ajay"  
4 date: "9/3/2020"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax  
for authoring HTML, PDF, and MS Word documents. For more details on using  
R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that  
includes both content as well as the output of any embedded R code chunks
```

22:19 # Including Plots R Markdown

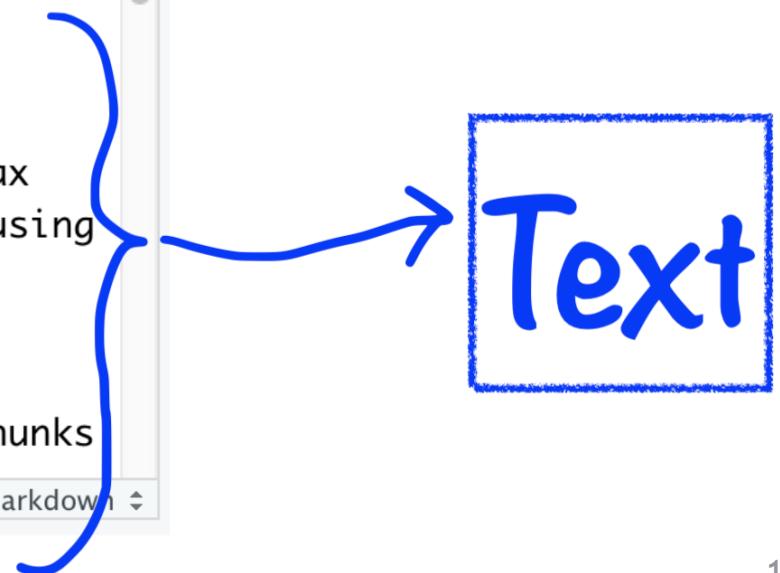
Code  
Chunk

Untitled1 x

ABC Knit

Insert Run

```
1 ---  
2 title: "Untitled"  
3 author: "Ajay"  
4 date: "9/3/2020"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax  
for authoring HTML, PDF, and MS Word documents. For more details on using  
R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that  
includes both content as well as the output of any embedded R code chunks
```



Text



# Knit a R Markdown File



- YAML options
- Headings, subheadings, text & maths equations
- Code Chunk
  - Include images
  - Include tables
  - Include plot
- Themes
- Multiple Output Formats
- eBook



# Q&A

# Dynamic Visualisation Using ggplot2

NEXT MODULE - 3

# Modules:

1. [Basics of R & RStudio](#)
2. [Dynamic Documents using R Markdown](#)
3. [Data Visualisation using ggplot2](#)
4. [Data Wrangling using dplyr](#)
5. [Slide Crafting using xaringan](#)



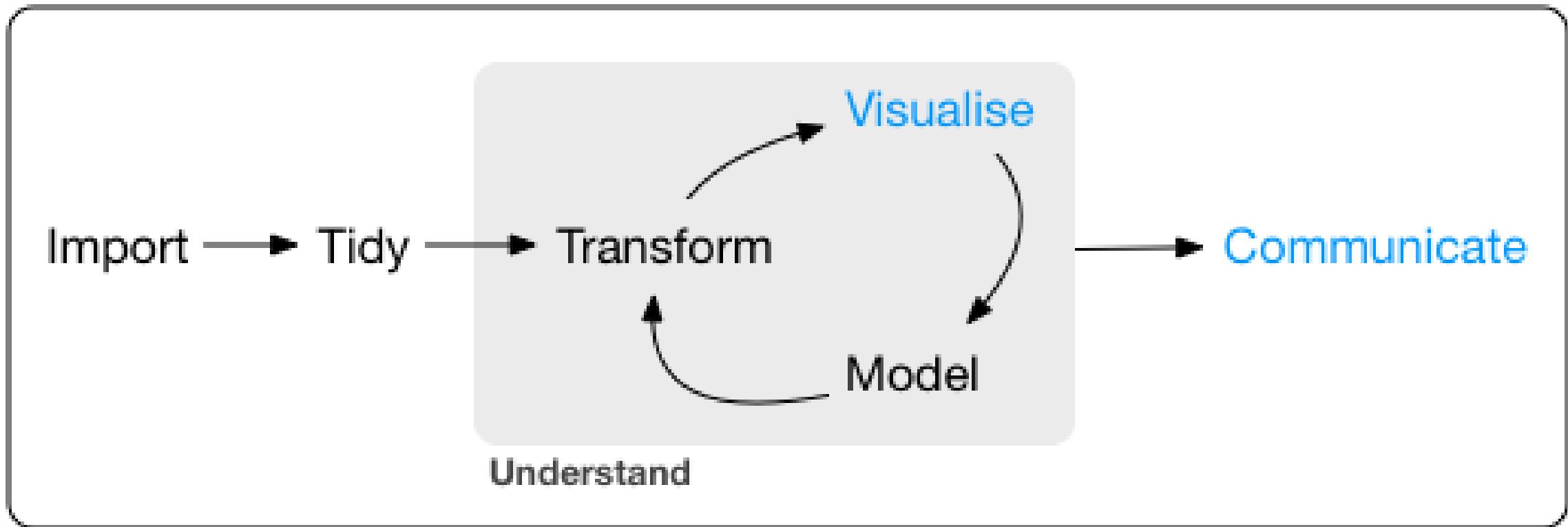
# Data Visualisation

SESSION - 3

Image credits:[Kayvan Mazhar](#)

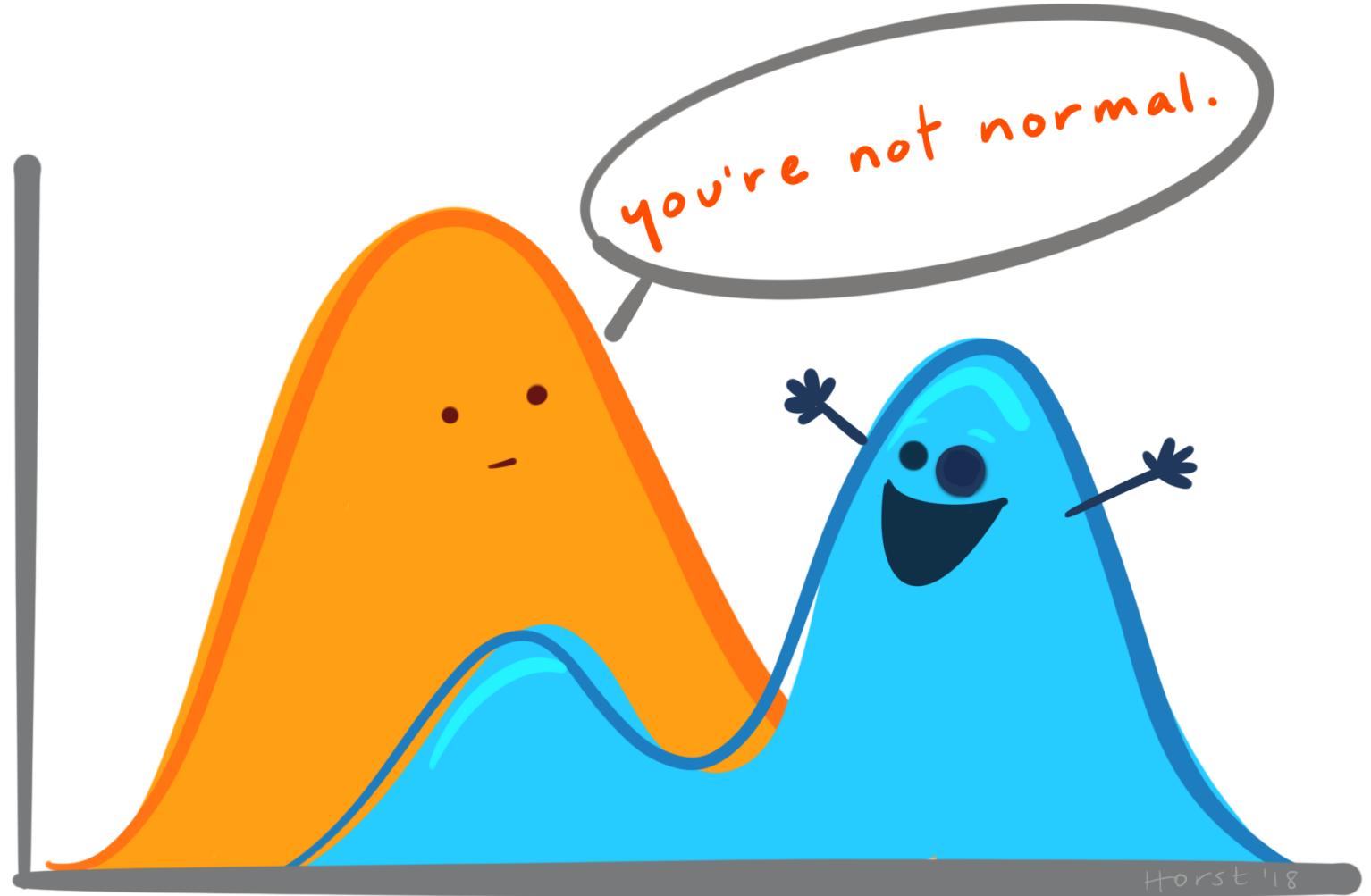


# Course Progress



Program

# Data



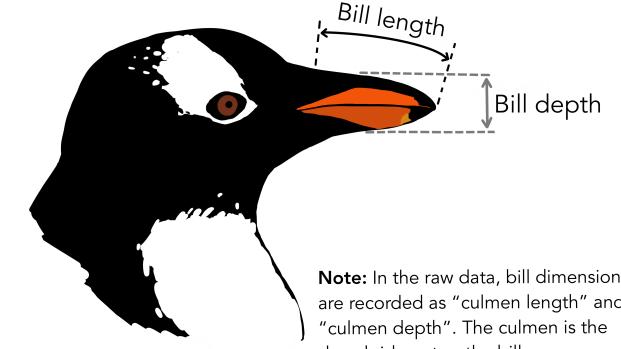
[Artwork Source](#)

# Variable types in R:

- `int` stands for integers, like 4, 55, 300.
- `dbl` stands for doubles, or real numbers like 3, 7.45, 1.565, 12.
- `chr` stands for character vectors, or strings like names.
- `dttm` stands for date-times (a date + a time).
- `lgl` stands for logical, vectors that contain only TRUE or FALSE.
- `fct` stands for factors, which R uses to represent **categorical variables** with fixed possible values like occupation: student, professional, government, business.
- `date` stands for dates.

# Data of Palmer Penguins

- It comes with R package `palmerpenguins`
- Name of the data is `penguins`
- To know more about the data `?penguins`
- Included variables are:
  - `species`, `island`, `bill_length_mm`, `bill_depth_mm`, `flipper_length_mm`, `body_mass_g`, `sex`, `year`



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

# An Overview of Data

---

Codes

Output

---

```
glimpse(penguins)
```

# An Overview of Data

Codes

Output

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie...
## $ island             <fct> Torgersen, Torgersen, Torgersen, Torg...
## $ bill_length_mm     <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9...
## $ bill_depth_mm      <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8...
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195,...
## $ body_mass_g        <int> 3750, 3800, 3250, NA, 3450, 3650, 3625...
## $ sex                <fct> male, female, female, NA, female, male...
## $ year               <int> 2007, 2007, 2007, 2007, 2007, 2007, 20...
```

Codes

Output

```
summary(penguins)
```

Codes

Output

```
##          species      island   bill_length_mm   bill_depth_mm
##  Adelie     :152    Biscoe     :168     Min.    :32.10     Min.    :13.10
##  Chinstrap: 68    Dream      :124    1st Qu.:39.23    1st Qu.:15.60
##  Gentoo    :124   Torgersen: 52   Median   :44.45    Median   :17.30
##                                         Mean     :43.92    Mean     :17.15
##                                         3rd Qu.:48.50    3rd Qu.:18.70
##                                         Max.     :59.60    Max.     :21.50
##                                         NA's     :2       NA's     :2
##          flipper_length_mm body_mass_g      sex        year
##  Min.    :172.0       Min.    :2700 female:165     Min.    :2007
##  1st Qu.:190.0       1st Qu.:3550 male   :168     1st Qu.:2007
##  Median  :197.0       Median  :4050 NA's   : 11     Median  :2008
##  Mean    :200.9       Mean    :4202                   Mean    :2008
##  3rd Qu.:213.0       3rd Qu.:4750                   3rd Qu.:2009
##  Max.    :231.0       Max.    :6300                   Max.    :2009
##  NA's    :2           NA's    :2
```

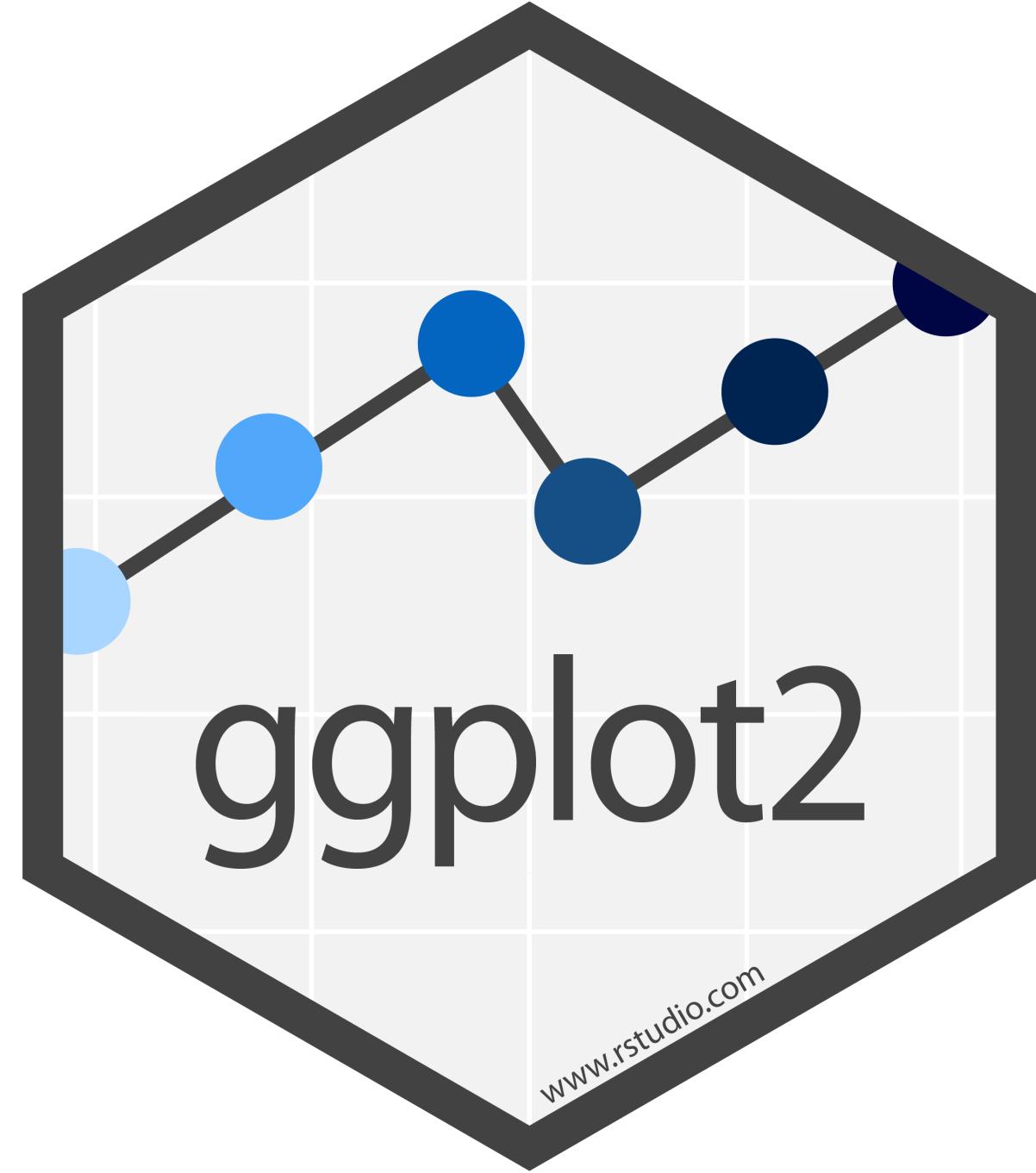
# Packages required:

```
library(palmerpenguins) # to access penguin data  
library(tidyverse) # to use ggplot2 pkg
```

- Packages recommended:

```
install.packages(c(  
  "directlabels", "dplyr", "gameofthrones", "ggforce", "gghighlight",  
  "ggnewscale", "ggplot2", "ggraph", "ggrepel", "ggtext", "ggthemes",  
  "hexbin", "mapproj", "maps", "munsell", "ozmaps", "paletteer",  
  "patchwork", "rmapshaper", "scico", "seriation", "sf", "stars",  
  "tidygraph", "tidyrr", "wesanderson"  
)
```

# R Package



# ggplot2 by Hadley Wickham

- "is a system for declaratively creating graphics, based on The Grammar of Graphics" (book by Late Leland Wilkinson)

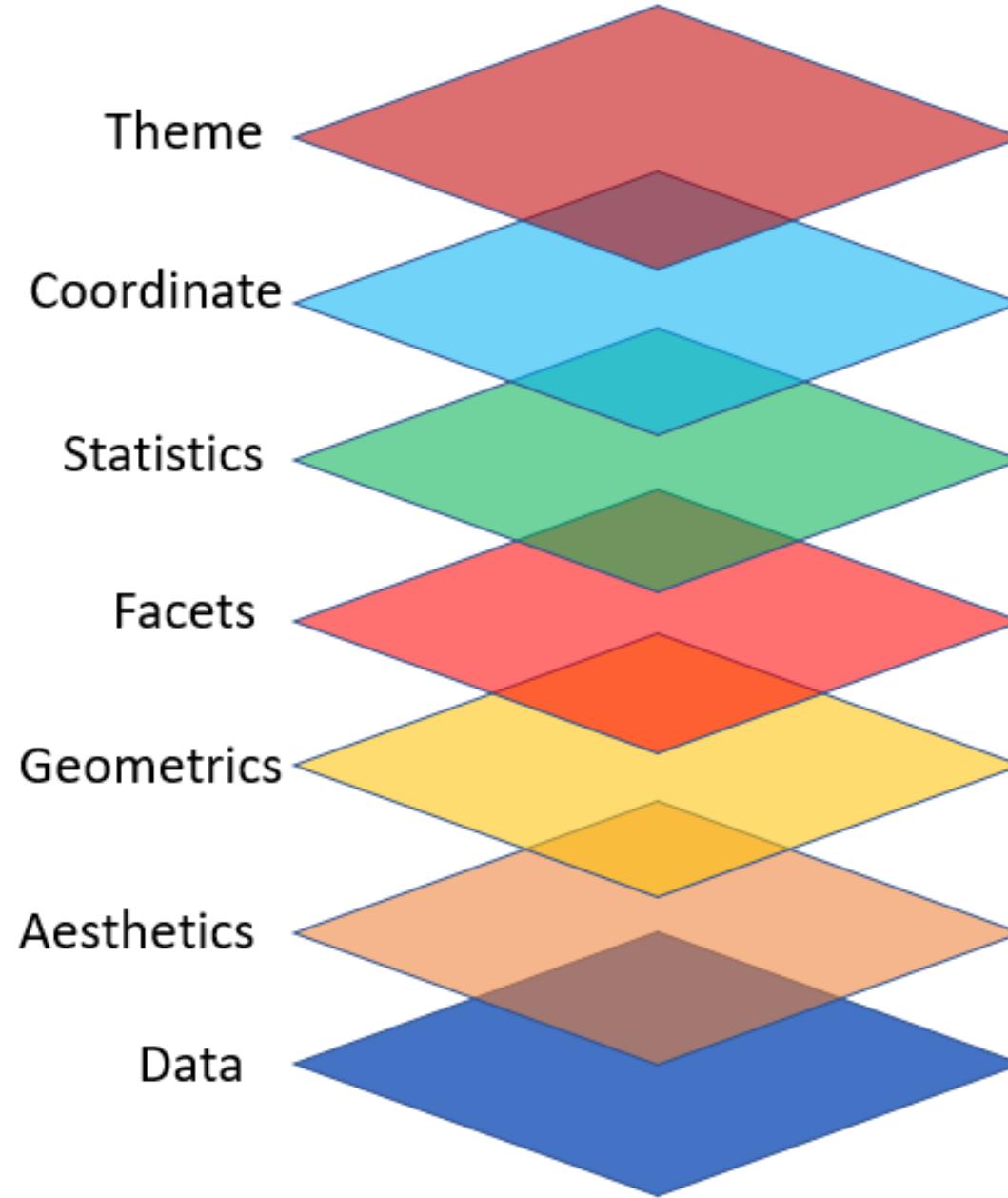


Late Leland Wilkinson



Hadley Wickham

Source



[Source](#)

# Key Components for ggplot2 Plot

1. data,
2. aesthetic mapping
3. at least one layer of geom function

---

Task

Codes

Output

---



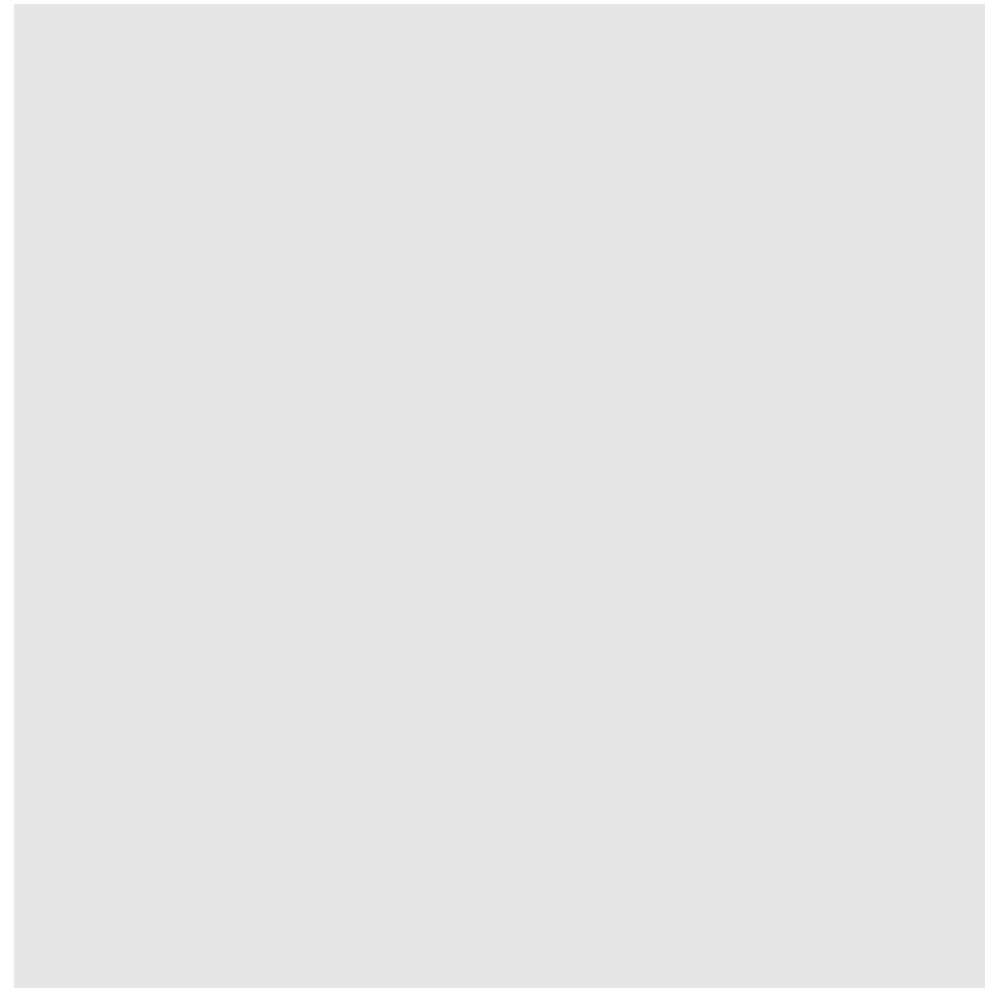
```
ggplot(data = penguins)
```

Task

Codes

Output

---



---

Task

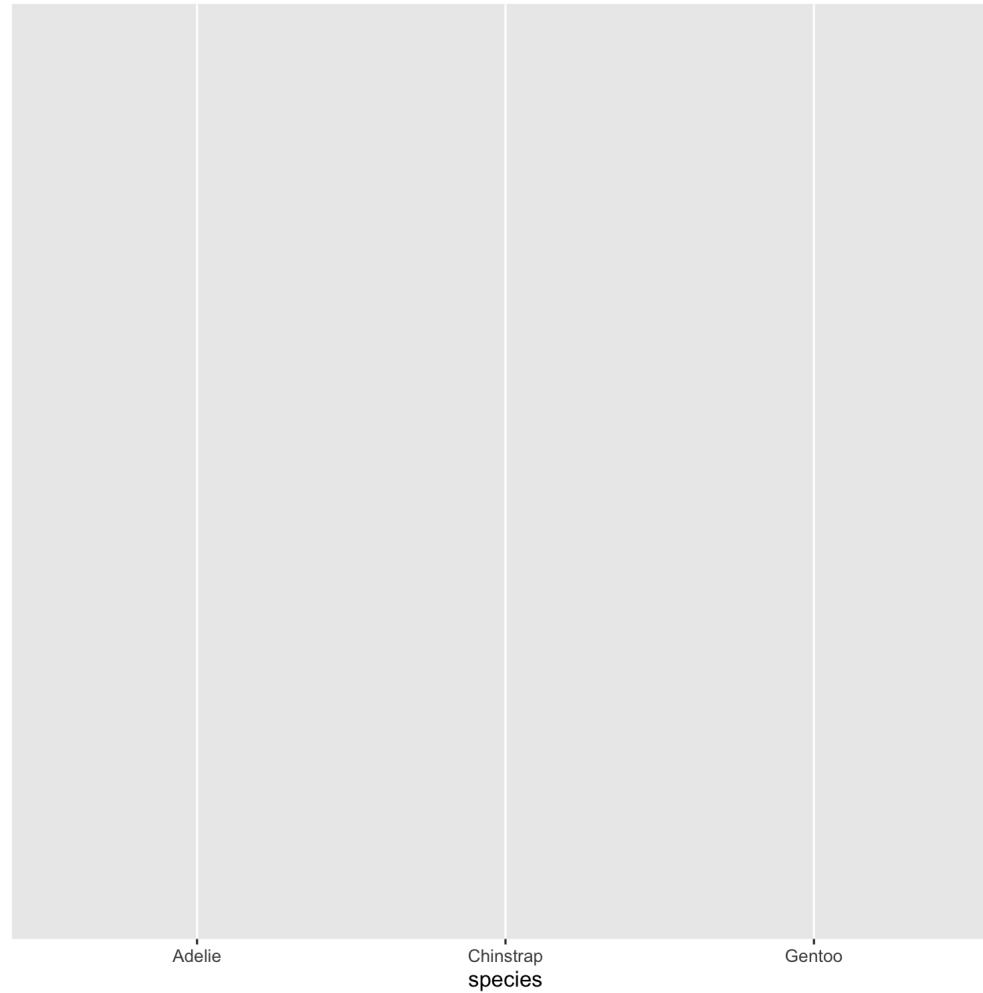
Codes

Output

---



```
ggplot(data = penguins, mapping = aes(x = species))
```



---

Task

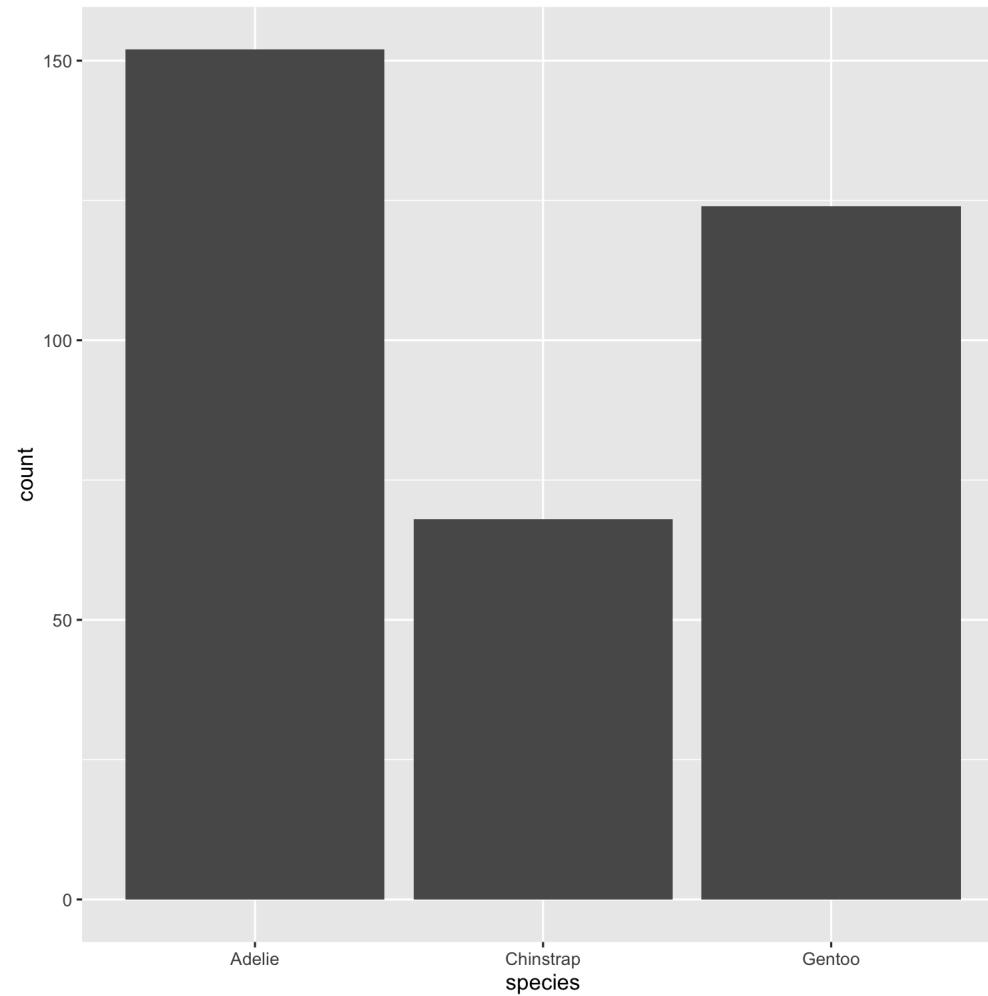
Codes

Output

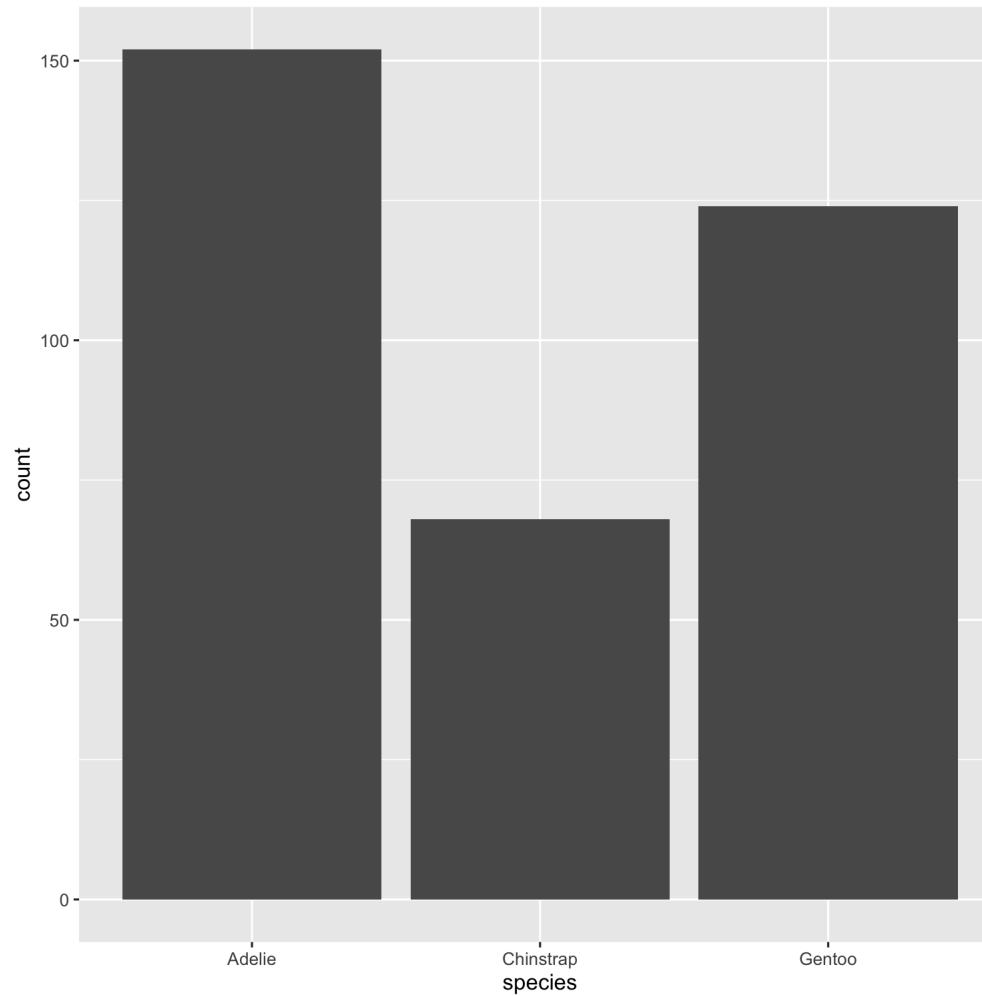
---



```
ggplot(data = penguins, mapping = aes(x = species)) +  
  geom_bar()
```



```
ggplot(penguins, aes(x = species)) +  
  geom_bar()
```





# YOUR TURN

05 : 00

Output

Codes



# YOUR TURN

05:00

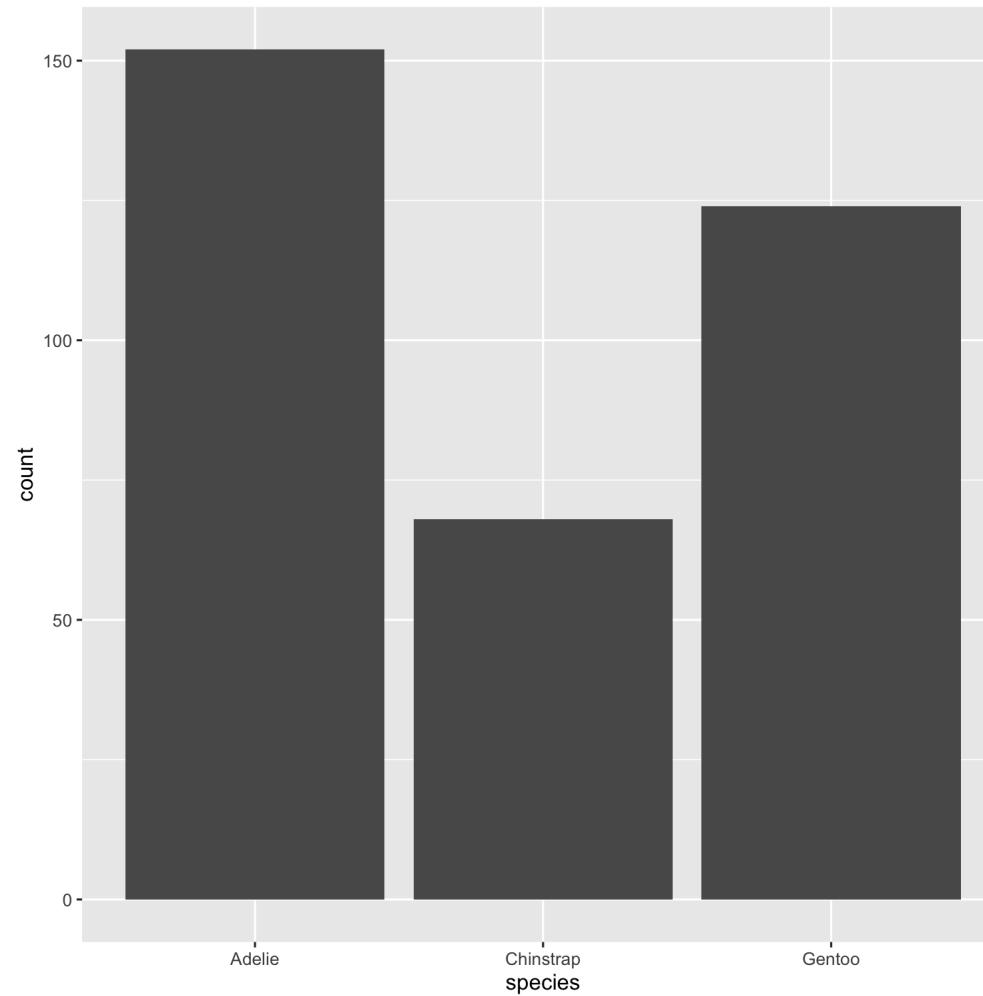
Output

Codes

```
ggplot(data = penguins, mapping = aes(x = island)) +  
  geom_bar()
```

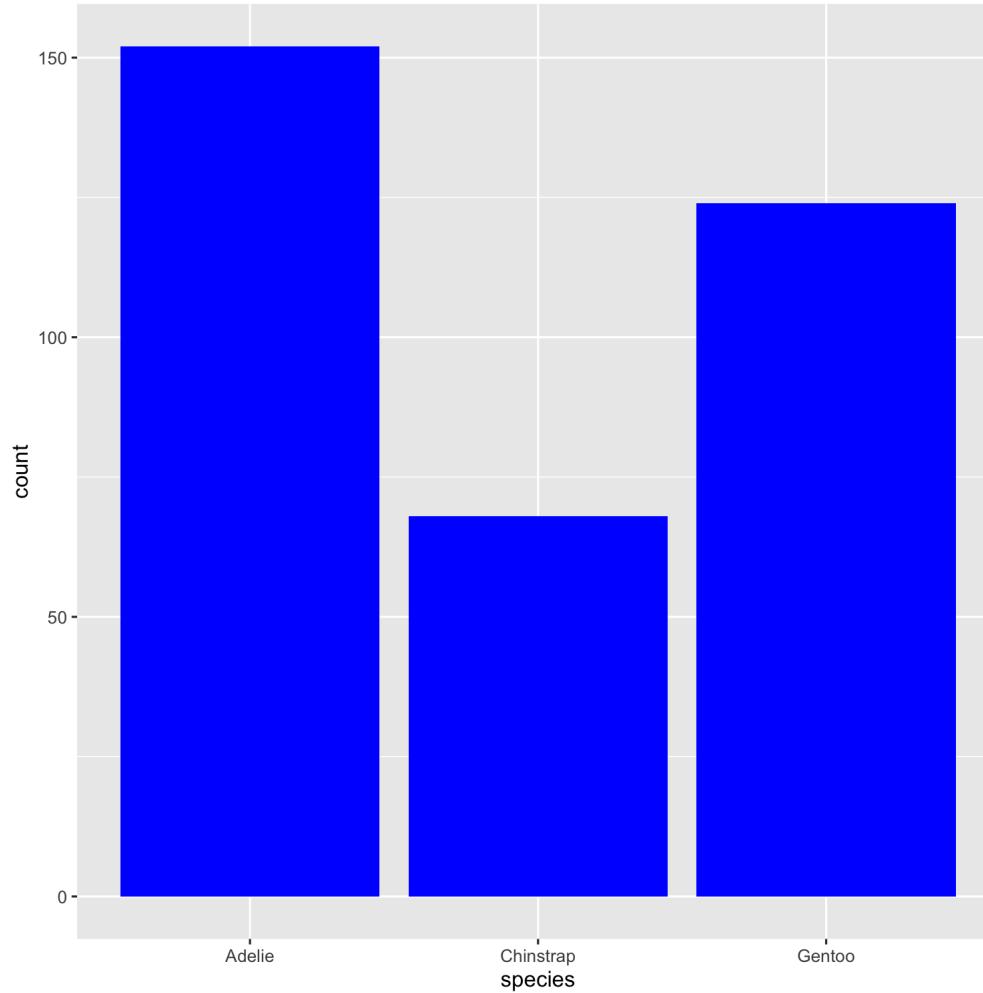
# How to export plot to your computer?

```
ggplot(data = penguins, mapping = aes(x = species)) +  
  geom_bar()  
  
ggsave("peng-species.pdf") # also try jpg/jpeg/png
```

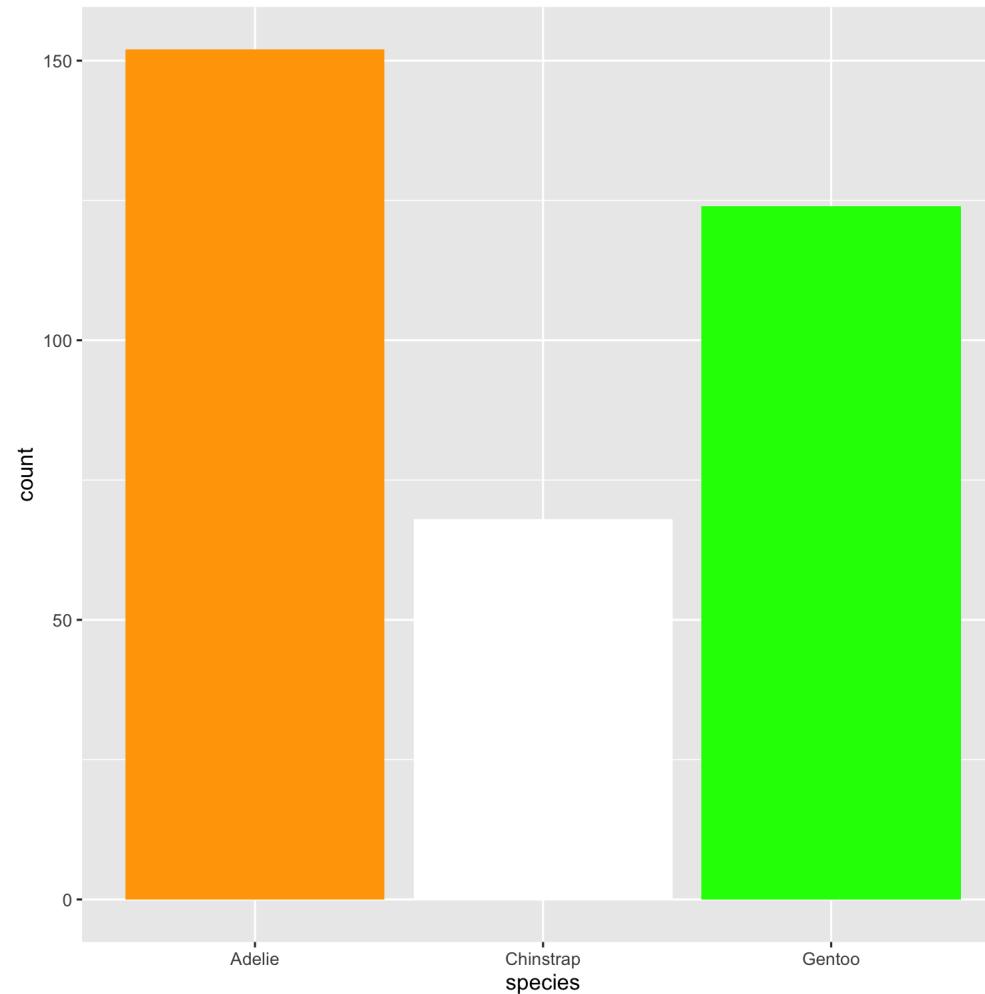


# How to add color to bars?

```
ggplot(data = penguins, mapping = aes(x = species)) +  
  geom_bar(fill = "blue")
```



```
ggplot(data = penguins, mapping = aes(x = species)) +  
  geom_bar(fill = c("orange", "white", "green"))  
  
# color names should be equal to the factor levels  
# in case of factor species levels are three  
# Adele, Chinstrap & Gentoo
```



How to add color using palette? 



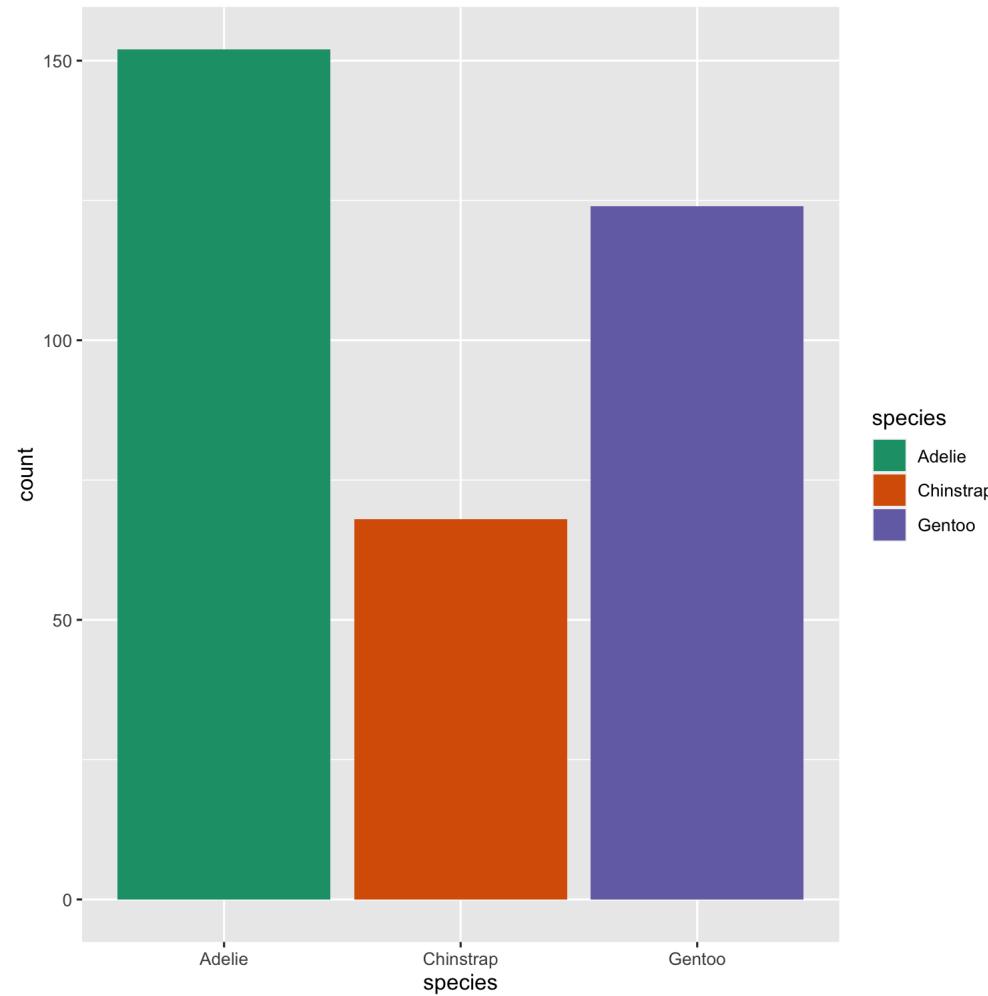
# Color Palette



- R package `RColorBrewer` & `wesanderson`

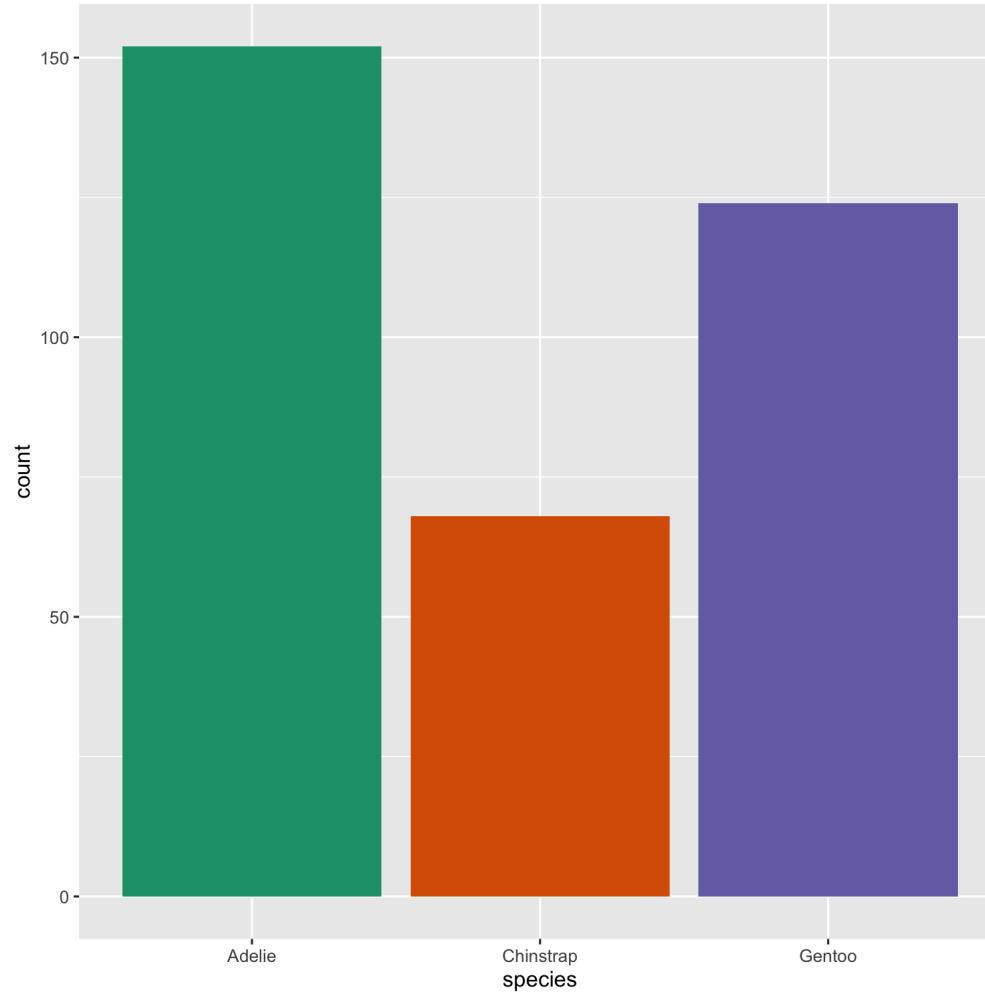
```
library(RColorBrewer)

ggplot(data = penguins,
       mapping = aes(x = species,
                      fill = species)) +
  geom_bar() +
  scale_fill_brewer(palette = "Dark2")
```



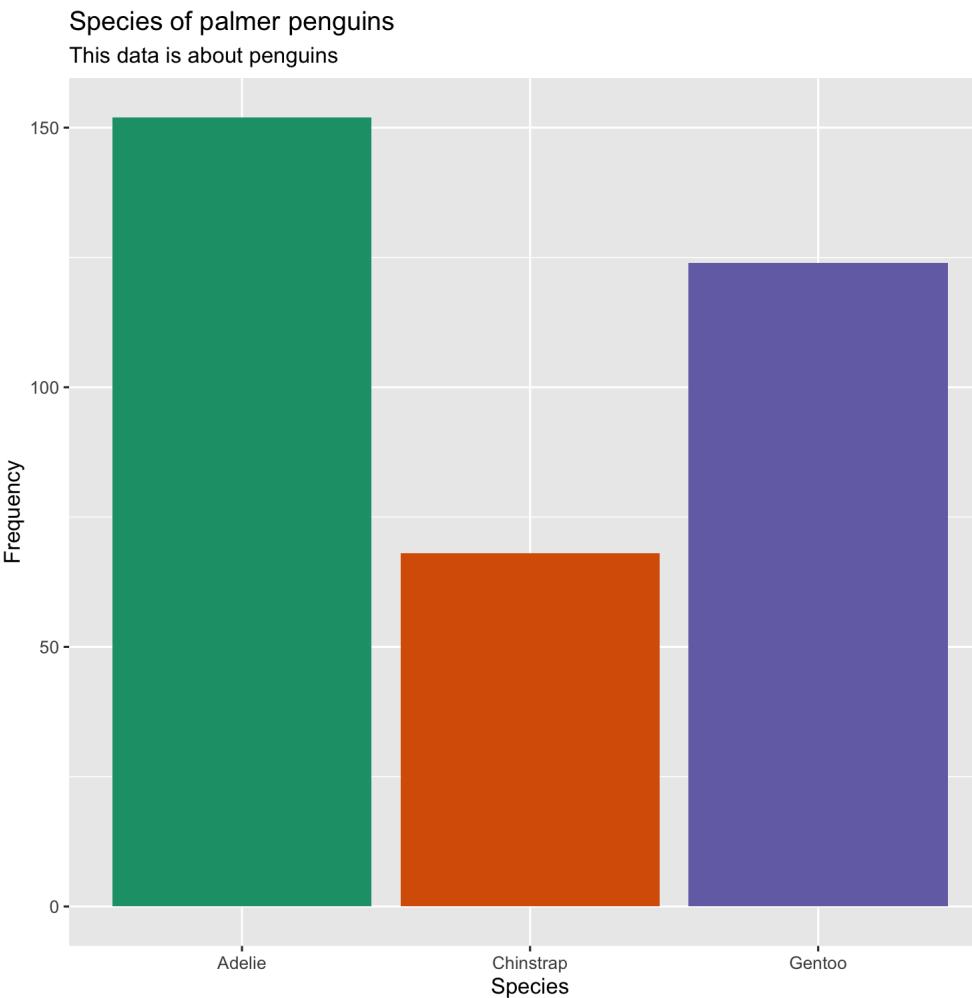
How to remove legend or change its position?

```
ggplot(data = penguins,  
       mapping = aes(x = species,  
                      fill = species)) +  
  geom_bar() +  
  scale_fill_brewer(palette = "Dark2") +  
  theme(legend.position = "none") # top, bottom, left
```



# How to plot title and axis titles?

```
ggplot(data = penguins,
       mapping = aes(x = species,
                      fill = species)) +
  geom_bar() +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none") +
  labs(
    title = "Species of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Species",
    y = "Frequency"
  )
```

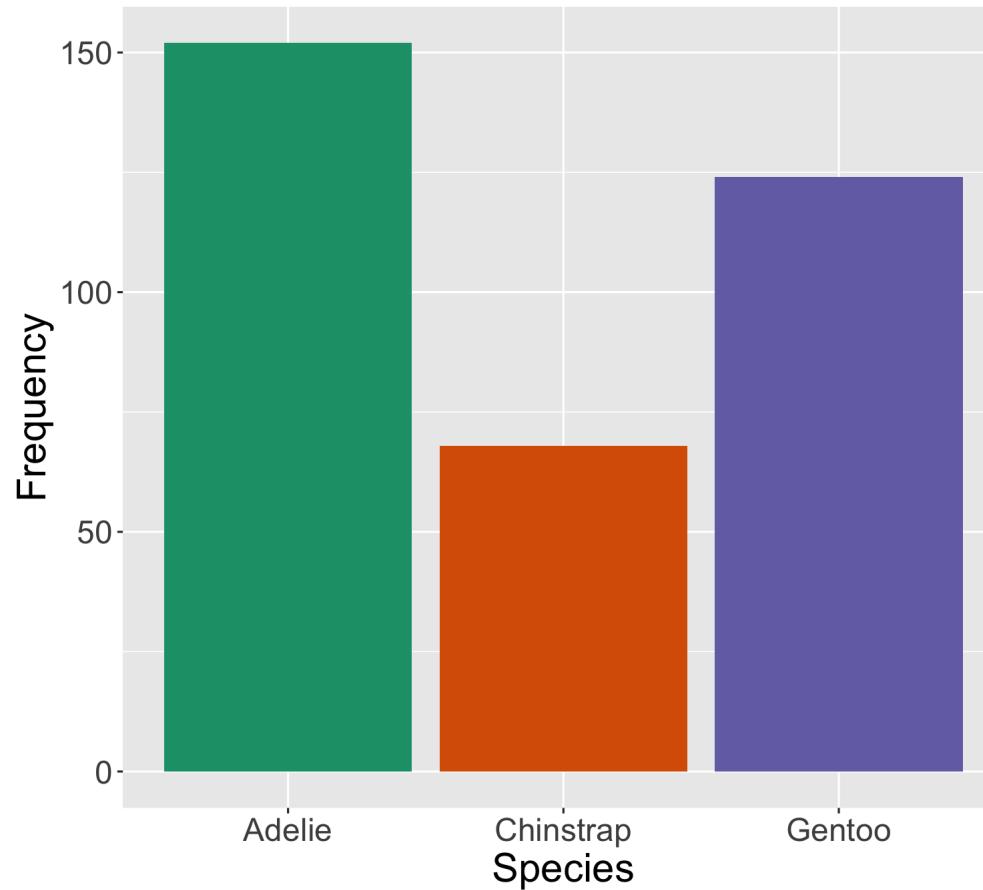


# How to control size of text?

```
ggplot(data = penguins,
       mapping = aes(x = species,
                      fill = species)) +
  geom_bar() +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none",
        text = element_text(size = 20)) +
  labs(
    title = "Species of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Species",
    y = "Frequency"
  )
```

## Species of palmer penguins

This data is about penguins

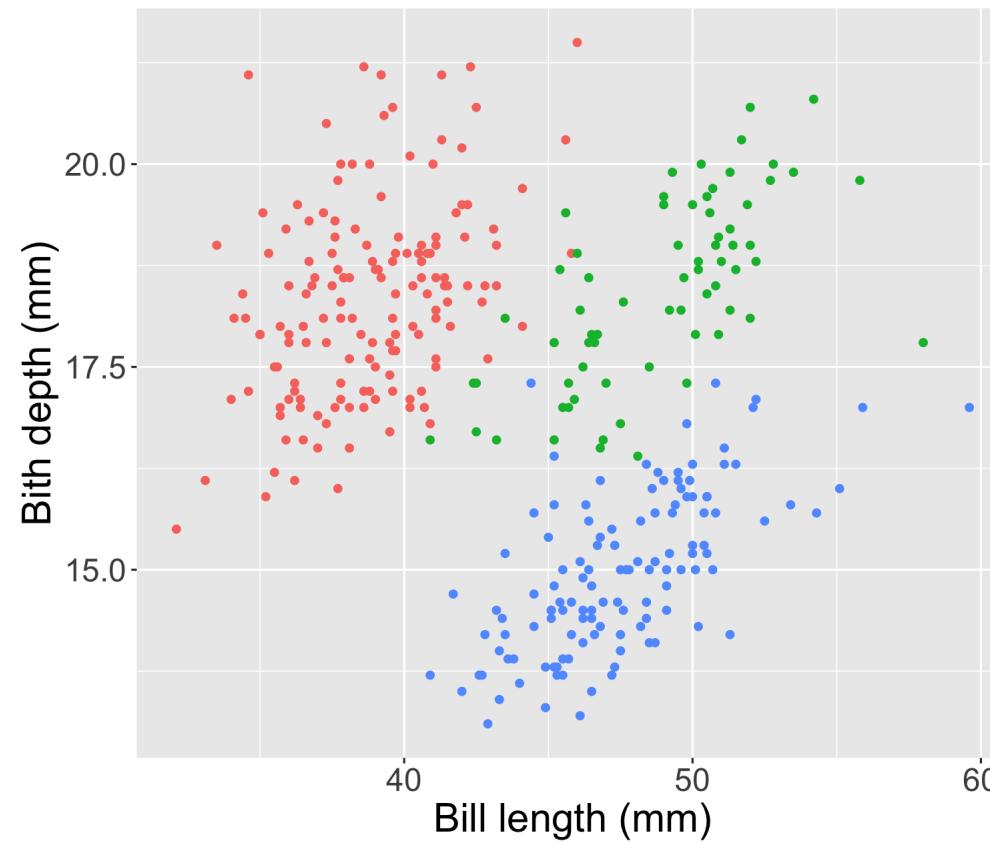


# How to plot two numeric variables?

```
ggplot(data = penguins,
       mapping = aes(x = bill_length_mm,
                      y = bill_depth_mm,
                      color = species)) +
  geom_point() +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none",
        text = element_text(size = 20)) +
  labs(
    title = "Relationship between bill length \n& depth of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Bill length (mm)",
    y = "Bith depth (mm)"
  )
```

## Relationship between bill length & depth of palmer penguins

This data is about penguins



# How to add themes to ggplot?

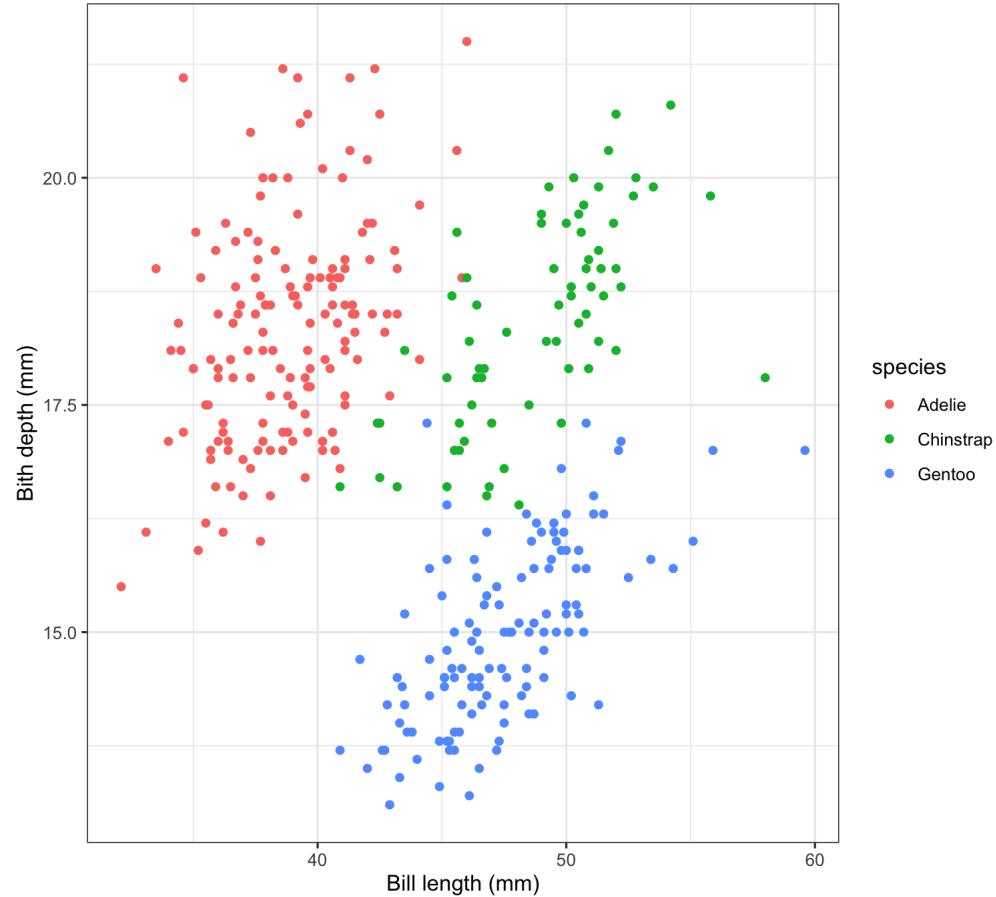
# ggplot2 themes

<https://ggplot2.tidyverse.org/reference/ggtheme.html>

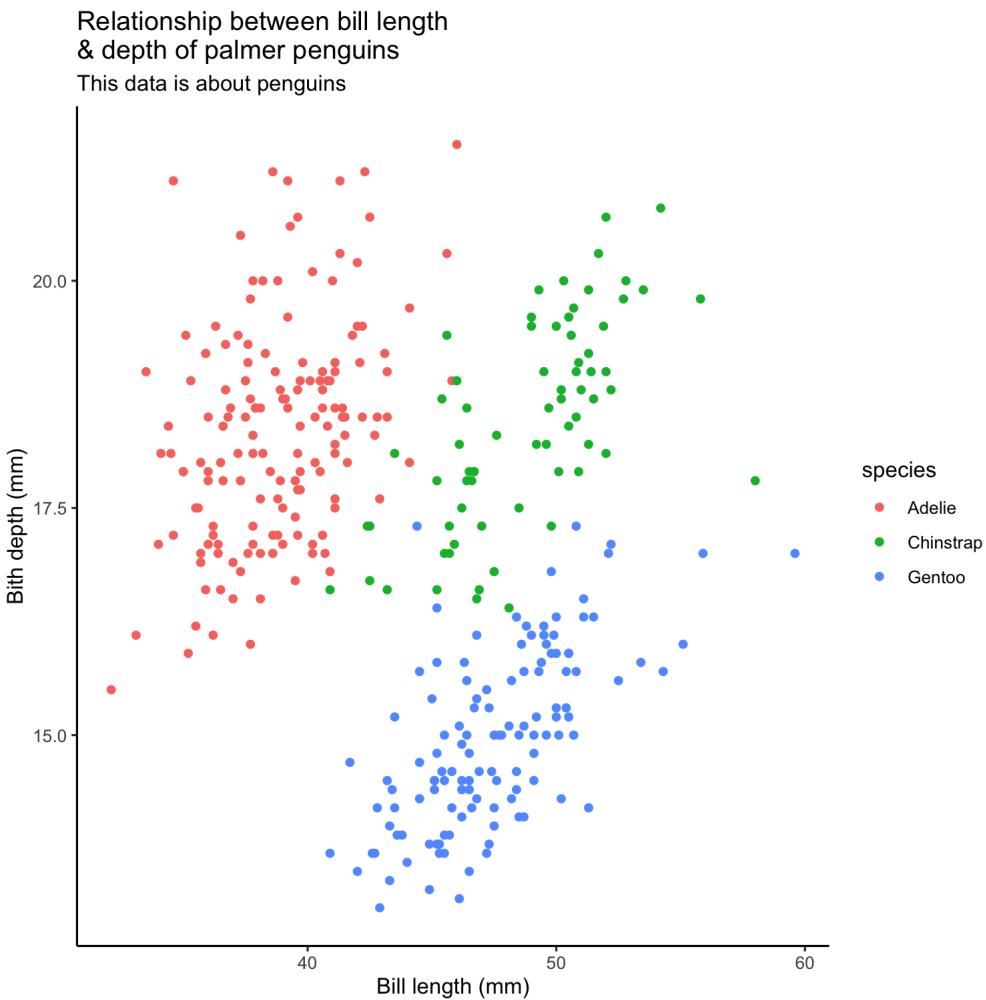
- theme\_gray()
- theme\_bw()
- theme\_linedraw()
- theme\_light()
- theme\_dark()
- theme\_minimal()
- theme\_classic()
- theme\_void()
- theme\_test()

```
ggplot(data = penguins,
       mapping = aes(x = bill_length_mm,
                      y = bill_depth_mm,
                      color = species)) +
  geom_point() +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none",
        text = element_text(size = 20)) +
  labs(
    title = "Relationship between bill length \n& depth of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Bill length (mm)",
    y = "Bith depth (mm)"
  ) +
  theme_bw()
```

Relationship between bill length  
& depth of palmer penguins  
This data is about penguins

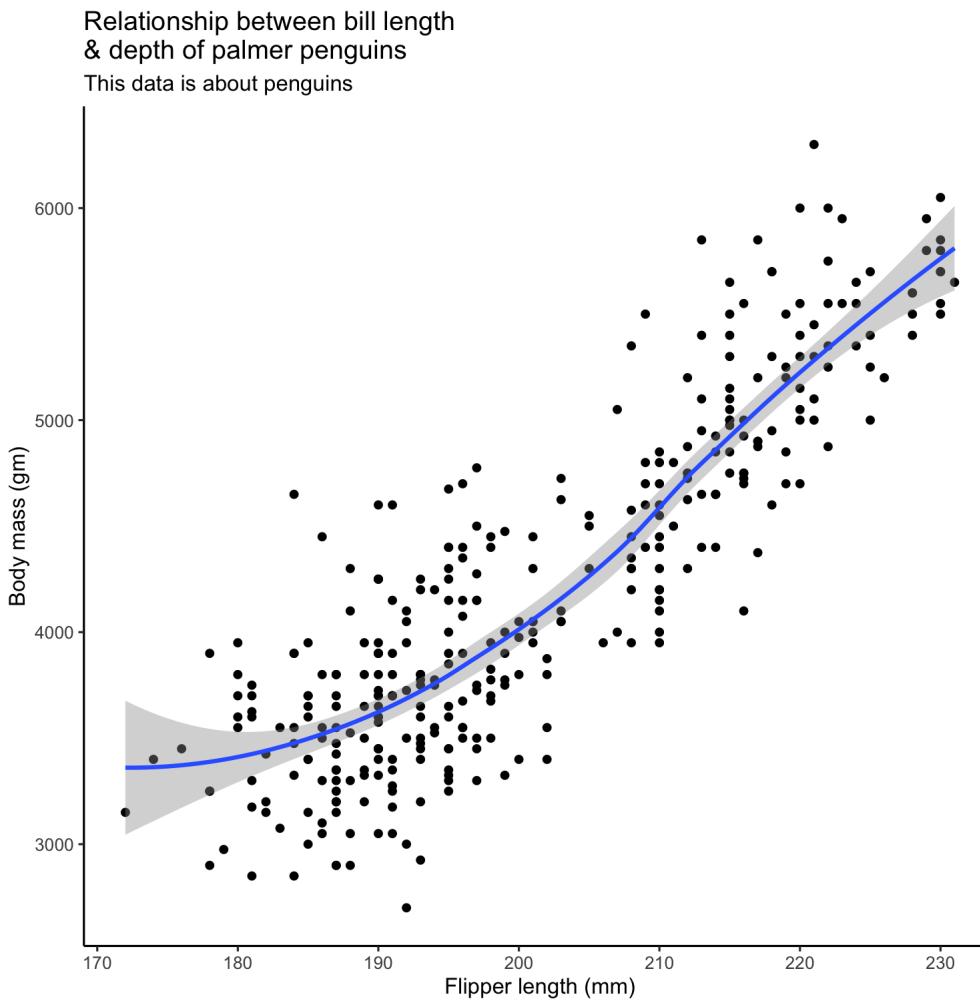


```
ggplot(data = penguins,
       mapping = aes(x = bill_length_mm,
                      y = bill_depth_mm,
                      color = species)) +
  geom_point() +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none",
        text = element_text(size = 20)) +
  labs(
    title = "Relationship between bill length \n& depth of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Bill length (mm)",
    y = "Bith depth (mm)"
  ) +
  theme_classic()
```



# How to add regression line to ggplot?

```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm,
                      y = body_mass_g)) +
  geom_point() +
  theme(legend.position = "none",
        text = element_text(size = 24)) +
  labs(
    title = "Relationship between bill length \n& depth of palmer penguins",
    subtitle = "This data is about penguins",
    x = "Flipper length (mm)",
    y = "Body mass (gm)"
  ) +
  theme_classic() +
  geom_smooth()
```



# More resources

- ggplot2 book <https://ggplot2-book.org/>
- CÉDRIC SCHERER <https://www.cedricscherer.com/>
- ggplot2 cook book <http://www.cookbook-r.com/>



# Q&A

# Dynamic Wrangling Using dplyr

NEXT MODULE - 4

# Modules:

1. Basics of R & RStudio
2. Dynamic Documents using R Markdown
3. Data Visualisation using ggplot2
4. Data Wrangling using dplyr
5. Slide Crafting using xaringan

A photograph showing several skeins of yarn in various colors (yellow, purple, white, orange, blue) arranged next to a circular, textured rug made of pom-poms in shades of blue, white, and yellow.

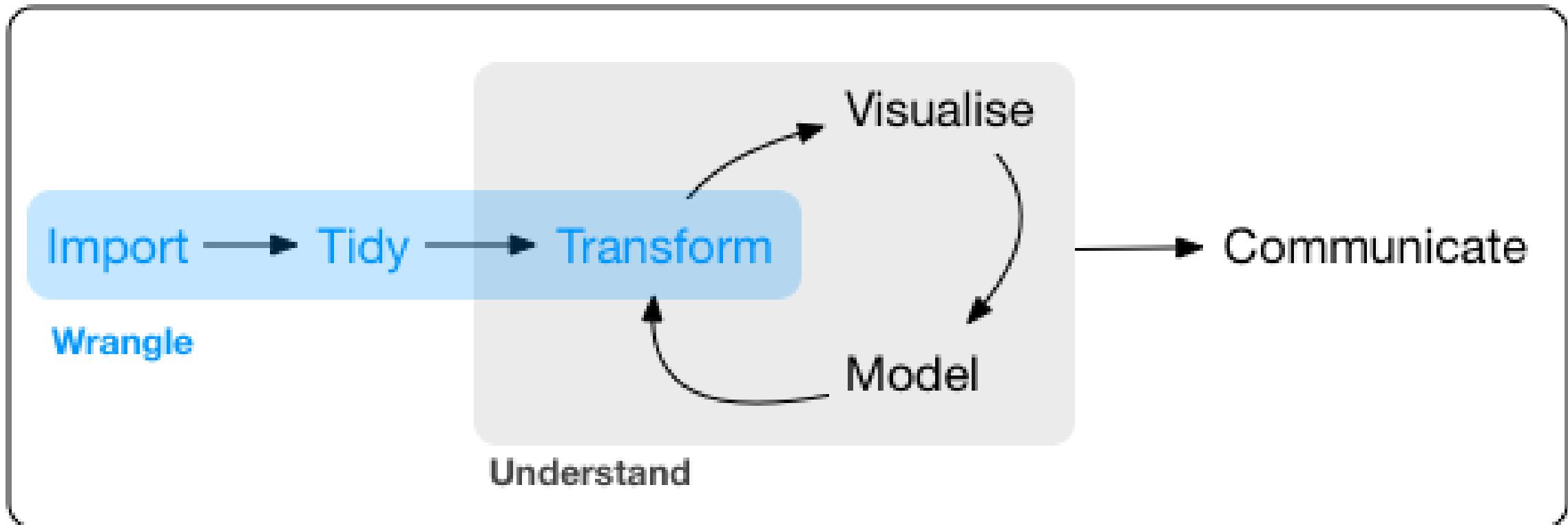
# Data Wrangling

SESSION - 4

Image credits:[Karina L](#)



# Course Progress



Program

# What is Data wrangling?

- "data exploration and data manipulation" ([Jesse Mostipak](#)).
- "tidying and transforming" ([Hadley & Garrett](#)).

country	year	cases	population
Afghanistan	1990	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

variables



country	year	cases	population
Afghanistan	1990	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

observations



country	year	cases	population
Afghanistan	1990	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

values



# "Transforming" data means:

- "narrowing in on observations of interest ..."
- creating new variables that are functions of existing variables ... and
- calculating a set of summary statistics."

# R Package



# dplyr package

- "dplyr is a grammar of data manipulation"
- "providing a consistent set of verbs that help you solve the most common data manipulation challenges:"
- Few important functions:
  - `filter()`
  - `select()`
  - `mutate()`
  - `arrange()`
  - `summarise()`

# filter() function:

- Picks cases based on their values.

# How to have a data of only Gentoo penguins?

```
# there are three species: Chinstrap, Gentoo, Adelie  
penguins %>%  
  filter(species == "Gentoo")
```

```
## # A tibble: 124 × 8
##   species island bill_length_mm bill_depth_mm flipper_length_mm
##   <fct>    <fct>        <dbl>        <dbl>            <int>
## 1 Gentoo  Biscoe        46.1        13.2            211
## 2 Gentoo  Biscoe        50          16.3            230
## 3 Gentoo  Biscoe        48.7        14.1            210
## 4 Gentoo  Biscoe        50          15.2            218
## 5 Gentoo  Biscoe        47.6        14.5            215
## 6 Gentoo  Biscoe        46.5        13.5            210
## 7 Gentoo  Biscoe        45.4        14.6            211
## 8 Gentoo  Biscoe        46.7        15.3            219
## 9 Gentoo  Biscoe        43.3        13.4            209
## 10 Gentoo Biscoe       46.8        15.4            215
## # ... with 114 more rows, and 3 more variables: body_mass_g <int>,
## #   sex <fct>, year <int>
```

```
# there are three species: Chinstrap, Gentoo, Adelie  
  
praw <- read_csv("data/gentoo-penguins1.csv")  
  
praw %>%  
  filter(species == "Gentoo") %>%  
  summary() %>%  
  kableExtra::kable()
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Length:124	Length:124	Min. :40.90	Min. :13.10	Min. :203.0	Min. :3950	Length:124	Min. :2007
Class :character	Class :character	1st Qu.:45.30	1st Qu.:14.20	1st Qu.:212.0	1st Qu.:4500	Class :character	1st Qu.:2007
Mode :character	Mode :character	Median :47.30	Median :15.00	Median :216.0	Median :4925	Mode :character	Median :2008
NA	NA	Mean :47.50	Mean :14.98	Mean :217.2	Mean :4985	NA	Mean :2008
NA	NA	3rd Qu.:49.55	3rd Qu.:15.70	3rd Qu.:221.0	3rd Qu.:5400	NA	3rd Qu.:2009
NA	NA	Max. :59.60	Max. :17.30	Max. :231.0	Max. :6050	NA	Max. :2009
NA	NA	NA's :1	NA's :1	NA's :1	NA's :1	NA	NA

# How to export data file to your computer?

```
# three species are Chinstrap, Gentoo, Adelie  
penguins %>%  
  filter(species == "Gentoo") %>%  
  write_csv("data/gentoo-penguins.csv")
```





# WAIT! What is %>%

- this is called **pipe** ( `%>%` = control + shift + m)
- "a powerful tool for clearly expressing a sequence of **multiple operations**"
- interpret/read it as **then**.

```
penguins %>%
  filter(species == "Gentoo") %>%
  summary() %>%
  kableExtra::kable()
```

# Comparison: Relational Operators

x < y

x > y

x <= y

x >= y

x == y (equal)

x != y (not equal)

How to have a data of penguins with bill length more than 43 mm?

```
penguins %>%
  filter(bill_length_mm > 43)
```

```
## # A tibble: 188 × 8
##   species island    bill_length_mm bill_depth_mm flipper_length...
##   <fct>   <fct>        <dbl>          <dbl>            <int>
## 1 Adelie  Torgersen      46            21.5            194
## 2 Adelie  Dream         44.1           19.7            196
## 3 Adelie  Torgersen      45.8           18.9            197
## 4 Adelie  Dream         43.2           18.5            192
## 5 Adelie  Biscoe         43.2            19              197
## 6 Adelie  Biscoe         45.6           20.3            191
## 7 Adelie  Torgersen      44.1            18              210
## 8 Adelie  Torgersen      43.1           19.2            197
## 9 Gentoo  Biscoe         46.1           13.2            211
## 10 Gentoo Biscoe          50             16.3            230
## # ... with 178 more rows, and 3 more variables: body_mass_g <int>,
## #       sex <fct>, year <int>
```

How to have a data of Gentoo penguins  
with bill length more than 50 mm?

```
penguins %>%
  filter(species == "Gentoo",
    bill_length_mm > 55)
```

```
## # A tibble: 3 × 8
##   species island bill_length_mm bill_depth_mm flipper_length_mm
##   <fct>    <fct>        <dbl>          <dbl>            <int>
## 1 Gentoo  Biscoe       59.6           17             230
## 2 Gentoo  Biscoe       55.9           17             228
## 3 Gentoo  Biscoe       55.1           16             230
## # ... with 3 more variables: body_mass_g <int>, sex <fct>,
## #     year <int>
```

How to have data of non-Gentoo penguins  
with bill length more than 45 mm and  
weight more than 4 kg?

```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000)
```

```
## # A tibble: 18 × 8
##   species     island bill_length_mm bill_depth_mm flipper_length...
##   <fct>      <fct>        <dbl>          <dbl>            <int>
## 1 Adelie     Torg...        46             21.5           194
## 2 Adelie     Torg...        45.8            18.9           197
## 3 Adelie     Biscoe        45.6            20.3           191
## 4 Chinstrap  Dream         46             18.9           195
## 5 Chinstrap  Dream         52             18.1           201
## 6 Chinstrap  Dream         50.5            19.6           201
## 7 Chinstrap  Dream         49.2            18.2           195
## 8 Chinstrap  Dream         52              19             197
## 9 Chinstrap  Dream         52.8            20             205
## 10 Chinstrap Dream         54.2            20.8           201
## 11 Chinstrap Dream         51              18.8           203
## 12 Chinstrap Dream         52              20.7           210
## 13 Chinstrap Dream         53.5            19.9           205
## 14 Chinstrap Dream         50.8            18.5           201
## 15 Chinstrap Dream         49              19.6           212
```

How to have only top or bottom rows from data?

```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000) %>%
head()
```

```
## # A tibble: 6 × 8
##   species     island bill_length_mm bill_depth_mm flipper_length...
##   <fct>      <fct>        <dbl>          <dbl>            <int>
## 1 Adelie    Torgers...       46             21.5           194
## 2 Adelie    Torgers...      45.8            18.9           197
## 3 Adelie    Biscoe         45.6            20.3           191
## 4 Chinstrap Dream          46             18.9           195
## 5 Chinstrap Dream          52             18.1           201
## 6 Chinstrap Dream          50.5            19.6           201
## # ... with 3 more variables: body_mass_g <int>, sex <fct>,
## #   year <int>
```

```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000) %>%
  tail(3)
```

```
## # A tibble: 3 × 8
##   species    island bill_length_mm bill_depth_mm flipper_length_mm
##   <fct>      <fct>          <dbl>           <dbl>             <int>
## 1 Chinstrap  Dream          50.7            19.7              203
## 2 Chinstrap  Dream          49.3            19.9              203
## 3 Chinstrap  Dream          50.8            19                210
## # ... with 3 more variables: body_mass_g <int>, sex <fct>,
## #   year <int>
```



# YOUR TURN

10:00

---

Task

Codes

Output

---

How many Chinstrap penguins are with bill length more than 45 mm and weight more than 4 kg?



# YOUR TURN

10:00

Task    Codes    Output

```
penguins %>%
  filter(species == "Chinstrap",
         bill_length_mm > 45,
         body_mass_g > 4000) %>%
head()
```

10:00



# YOUR TURN

Task    Codes    Output

```
## # A tibble: 6 × 8
##   species    island bill_length_mm bill_depth_mm flipper_length_mm
##   <fct>     <fct>        <dbl>          <dbl>            <int>
## 1 Chinstrap Dream         46             18.9            195
## 2 Chinstrap Dream         52             18.1            201
## 3 Chinstrap Dream        50.5            19.6            201
## 4 Chinstrap Dream        49.2            18.2            195
## 5 Chinstrap Dream         52              19              197
## 6 Chinstrap Dream        52.8            20              205
## # ... with 3 more variables: body_mass_g <int>, sex <fct>,
## #   year <int>
```

`select()` function: Chooses rows based  
on column values.



How to have only **species** variable in  
data?

```
penguins %>%  
  select(species)
```

```
## # A tibble: 344 × 1
##   species
##   <fct>
## 1 Adelie
## 2 Adelie
## 3 Adelie
## 4 Adelie
## 5 Adelie
## 6 Adelie
## 7 Adelie
## 8 Adelie
## 9 Adelie
## 10 Adelie
## # ... with 334 more rows
```

How to have a specific range of variables  
in data?

```
penguins %>%
  select(species : bill_depth_mm)
```

```
## # A tibble: 344 × 4
##   species island    bill_length_mm bill_depth_mm
##   <fct>   <fct>          <dbl>           <dbl>
## 1 Adelie  Torgersen      39.1            18.7
## 2 Adelie  Torgersen      39.5            17.4
## 3 Adelie  Torgersen      40.3            18
## 4 Adelie  Torgersen       NA              NA
## 5 Adelie  Torgersen      36.7            19.3
## 6 Adelie  Torgersen      39.3            20.6
## 7 Adelie  Torgersen      38.9            17.8
## 8 Adelie  Torgersen      39.2            19.6
## 9 Adelie  Torgersen      34.1            18.1
## 10 Adelie Torgersen       42              20.2
## # ... with 334 more rows
```

How to have variables based upon their location in data?

```
penguins %>%  
  select(4:8)
```

```
## # A tibble: 344 × 5
##   bill_depth_mm flipper_length_mm body_mass_g sex     year
##       <dbl>           <int>        <int> <fct>    <int>
## 1     18.7            181         3750 male    2007
## 2     17.4            186         3800 female  2007
## 3     18              195         3250 female  2007
## 4      NA             NA          NA    <NA>    2007
## 5     19.3            193         3450 female  2007
## 6     20.6            190         3650 male    2007
## 7     17.8            181         3625 female  2007
## 8     19.6            195         4675 male    2007
## 9     18.1            193         3475 <NA>    2007
## 10    20.2            190         4250 <NA>    2007
## # ... with 334 more rows
```

# How to have specific variables in data?

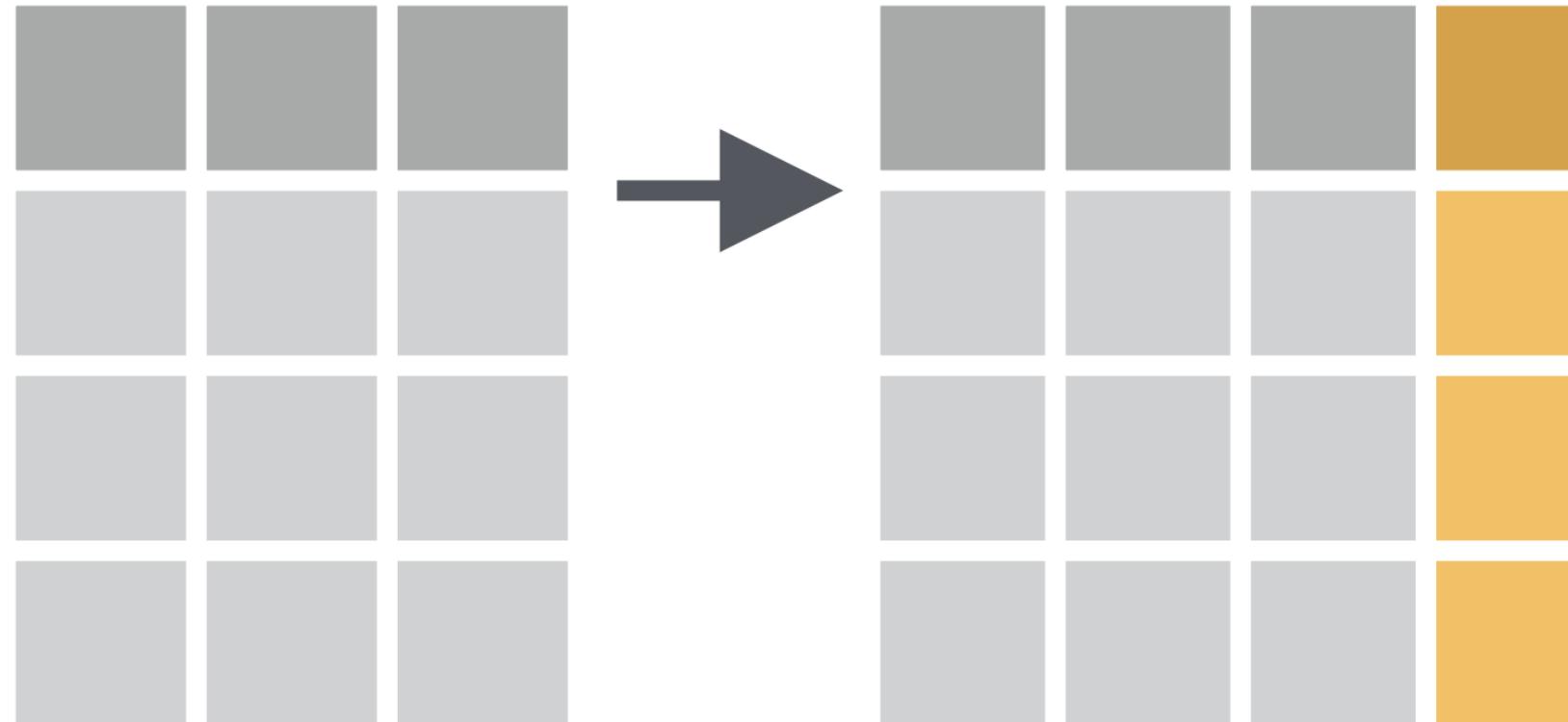
```
penguins %>%
  select(species, body_mass_g, year)
```

```
## # A tibble: 344 × 3
##   species body_mass_g year
##   <fct>     <int> <int>
## 1 Adelie      3750  2007
## 2 Adelie      3800  2007
## 3 Adelie      3250  2007
## 4 Adelie        NA  2007
## 5 Adelie      3450  2007
## 6 Adelie      3650  2007
## 7 Adelie      3625  2007
## 8 Adelie      4675  2007
## 9 Adelie      3475  2007
## 10 Adelie     4250  2007
## # ... with 334 more rows
```

```
penguins %>%
  select(-c(species, body_mass_g, year))
```

```
## # A tibble: 344 × 5
##   island    bill_length_mm bill_depth_mm flipper_length_mm sex
##   <fct>        <dbl>          <dbl>            <dbl>   <int> <fct>
## 1 Torgersen     39.1           18.7             181   male
## 2 Torgersen     39.5           17.4             186   female
## 3 Torgersen     40.3           18               195   female
## 4 Torgersen      NA             NA               NA   <NA>
## 5 Torgersen     36.7           19.3             193   female
## 6 Torgersen     39.3           20.6             190   male
## 7 Torgersen     38.9           17.8             181   female
## 8 Torgersen     39.2           19.6             195   male
## 9 Torgersen     34.1           18.1             193   <NA>
## 10 Torgersen    42              20.2             190  <NA>
## # ... with 334 more rows
```

**mutate()** function: Adds new variables  
that are functions of existing variables



How to convert penguin body mass from grams to kilograms?

```
penguins %>%  
  mutate(body_mass_kg = body_mass_g / 1000)
```

```
## # A tibble: 344 × 9
##   species island   bill_length_mm bill_depth_mm flipper_length...
##   <fct>   <fct>        <dbl>          <dbl>            <int>
## 1 Adelie Torgersen      39.1          18.7            181
## 2 Adelie Torgersen      39.5          17.4            186
## 3 Adelie Torgersen      40.3           18              195
## 4 Adelie Torgersen       NA             NA              NA
## 5 Adelie Torgersen      36.7          19.3            193
## 6 Adelie Torgersen      39.3          20.6            190
## 7 Adelie Torgersen      38.9          17.8            181
## 8 Adelie Torgersen      39.2          19.6            195
## 9 Adelie Torgersen      34.1          18.1            193
## 10 Adelie Torgersen      42             20.2            190
## # ... with 334 more rows, and 4 more variables: body_mass_g <int>,
## #   sex <fct>, year <int>, body_mass_kg <dbl>
```

```
penguins %>%
  select(body_mass_g) %>%
  mutate(body_mass_kg = body_mass_g / 1000)
```

```
## # A tibble: 344 × 2
##   body_mass_g body_mass_kg
##   <int>      <dbl>
## 1     3750      3.75
## 2     3800      3.8
## 3     3250      3.25
## 4       NA        NA
## 5     3450      3.45
## 6     3650      3.65
## 7     3625      3.62
## 8     4675      4.68
## 9     3475      3.48
## 10    4250      4.25
## # ... with 334 more rows
```

```
penguins %>%
  mutate(body_mass_kg = body_mass_g / 1000,
    bill = bill_length_mm * bill_depth_mm)
```

```
## # A tibble: 344 × 10
##   species island    bill_length_mm bill_depth_mm flipper_length...
##   <fct>   <fct>        <dbl>          <dbl>            <int>
## 1 Adelie  Torgersen     39.1           18.7            181
## 2 Adelie  Torgersen     39.5           17.4            186
## 3 Adelie  Torgersen     40.3           18              195
## 4 Adelie  Torgersen     NA              NA              NA
## 5 Adelie  Torgersen     36.7           19.3            193
## 6 Adelie  Torgersen     39.3           20.6            190
## 7 Adelie  Torgersen     38.9           17.8            181
## 8 Adelie  Torgersen     39.2           19.6            195
## 9 Adelie  Torgersen     34.1           18.1            193
## 10 Adelie Torgersen      42              20.2            190
## # ... with 334 more rows, and 5 more variables: body_mass_g <int>,
## #       sex <fct>, year <int>, body_mass_kg <dbl>, bill <dbl>
```

```
penguins %>%
  mutate(body_mass_kg = body_mass_g / 1000,
        bill = bill_length_mm * bill_depth_mm) %>%
  select(body_mass_kg,
        bill)
```

```
## # A tibble: 344 × 2
##   body_mass_kg bill
##       <dbl> <dbl>
## 1         3.75  731.
## 2         3.8    687.
## 3         3.25  725.
## 4           NA     NA
## 5         3.45  708.
## 6         3.65  810.
## 7         3.62  692.
## 8         4.68  768.
## 9         3.48  617.
## 10        4.25  848.
## # ... with 334 more rows
```

arrange() function: Changes the order of the rows.

How to have data arranged by the  
ascending order of bill length of  
penguins?

```
penguins %>%  
  arrange(bill_length_mm)
```

```
## # A tibble: 344 × 8
##   species island    bill_length_mm bill_depth_mm flipper_length...
##   <fct>   <fct>        <dbl>          <dbl>            <int>
## 1 Adelie  Dream       32.1           15.5            188
## 2 Adelie  Dream       33.1           16.1            178
## 3 Adelie  Torgersen   33.5           19               190
## 4 Adelie  Dream       34               17.1            185
## 5 Adelie  Torgersen   34.1           18.1            193
## 6 Adelie  Torgersen   34.4           18.4            184
## 7 Adelie  Biscoe      34.5           18.1            187
## 8 Adelie  Torgersen   34.6           21.1            198
## 9 Adelie  Torgersen   34.6           17.2            189
## 10 Adelie  Biscoe     35               17.9            190
## # ... with 334 more rows, and 3 more variables: body_mass_g <int>,
## #   sex <fct>, year <int>
```

```
penguins %>%  
  arrange(desc(bill_length_mm))
```

```
## # A tibble: 344 × 8
##   species    island bill_length_mm bill_depth_mm flipper_length_mm
##   <fct>     <fct>          <dbl>           <dbl>              <int>
## 1 Gentoo    Biscoe        59.6            17                230
## 2 Chinstrap Dream         58               17.8              181
## 3 Gentoo    Biscoe        55.9            17                228
## 4 Chinstrap Dream         55.8            19.8              207
## 5 Gentoo    Biscoe        55.1            16                230
## 6 Gentoo    Biscoe        54.3            15.7              231
## 7 Chinstrap Dream         54.2            20.8              201
## 8 Chinstrap Dream         53.5            19.9              205
## 9 Gentoo    Biscoe        53.4            15.8              219
## 10 Chinstrap Dream        52.8             20                205
## # ... with 334 more rows, and 3 more variables: body_mass_g <int>,
## #       sex <fct>, year <int>
```

```
penguins %>%  
  arrange(species)
```

```
## # A tibble: 344 × 8
##   species island    bill_length_mm bill_depth_mm flipper_length...
##   <fct>   <fct>        <dbl>          <dbl>            <int>
## 1 Adelie  Torgersen     39.1           18.7            181
## 2 Adelie  Torgersen     39.5           17.4            186
## 3 Adelie  Torgersen     40.3           18              195
## 4 Adelie  Torgersen      NA             NA              NA
## 5 Adelie  Torgersen     36.7           19.3            193
## 6 Adelie  Torgersen     39.3           20.6            190
## 7 Adelie  Torgersen     38.9           17.8            181
## 8 Adelie  Torgersen     39.2           19.6            195
## 9 Adelie  Torgersen     34.1           18.1            193
## 10 Adelie Torgersen      42              20.2            190
## # ... with 334 more rows, and 3 more variables: body_mass_g <int>,
## #       sex <fct>, year <int>
```

# summarise() function

**summarise()** function: Chooses rows  
based on column values.



How to find mean bill length of all penguins?

```
penguins %>%  
  drop_na() %>%  
  summarise(mean_bill_length_mm = mean(bill_length_mm))
```

```
## # A tibble: 1 × 1
##   mean_bill_length_mm
##   <dbl>
## 1 44.0
```

# How to find species-wise mean bill length of penguins?

```
penguins %>%  
  drop_na() %>%  
  group_by(species) %>%  
  summarise(mean_bill_length_mm = mean(bill_length_mm))
```

```
## # A tibble: 3 × 2
##   species    mean_bill_length_mm
##   <fct>          <dbl>
## 1 Adelie         38.8
## 2 Chinstrap      48.8
## 3 Gentoo         47.6
```

How to find species-wise mean bill length  
of penguins and total number of penguins  
in each species?

```
penguins %>%  
  drop_na() %>%  
  group_by(species) %>%  
  summarise(mean_bill_length_mm = mean(bill_length_mm), n = n())
```

```
## # A tibble: 3 × 3
##   species    mean_bill_length_mm     n
##   <fct>          <dbl> <int>
## 1 Adelie         38.8   146
## 2 Chinstrap      48.8    68
## 3 Gentoo         47.6   119
```



# Q&A

# Slide Crafting using xaringan

NEXT MODULE - 5

# Modules:

1. Basics of R & RStudio
2. Dynamic Documents using R Markdown
3. Data Visualisation using ggplot2
4. Data Wrangling using dplyr
5. Slide Crafting using xaringan

# Slide Crafting

SESSION - 5

Image credits:[Markus Spiske](#)





R  
Package



# xaringan

- xaringan package to be a Presentation Ninja 🥊
- "for creating slideshows with remark.js through R Markdown"
- [Xie Yihui](#)



[Source](#)

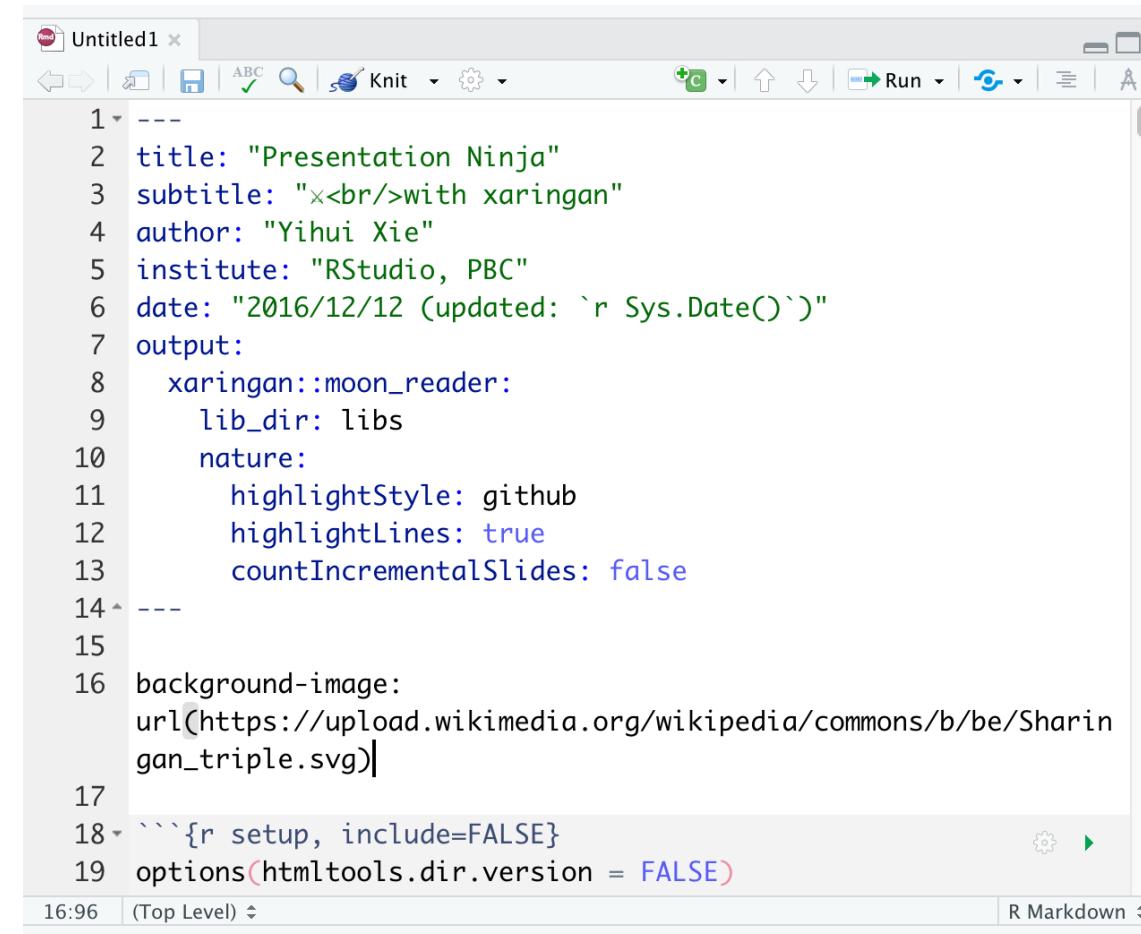
# Packages required:

```
library(palmerpenguins) # to access penguin data  
  
library(xaringan)  
library(xaringanthemer)  
library(xaringanExtra)
```

File \(\rightarrow\) New File \(\rightarrow\) R Markdown

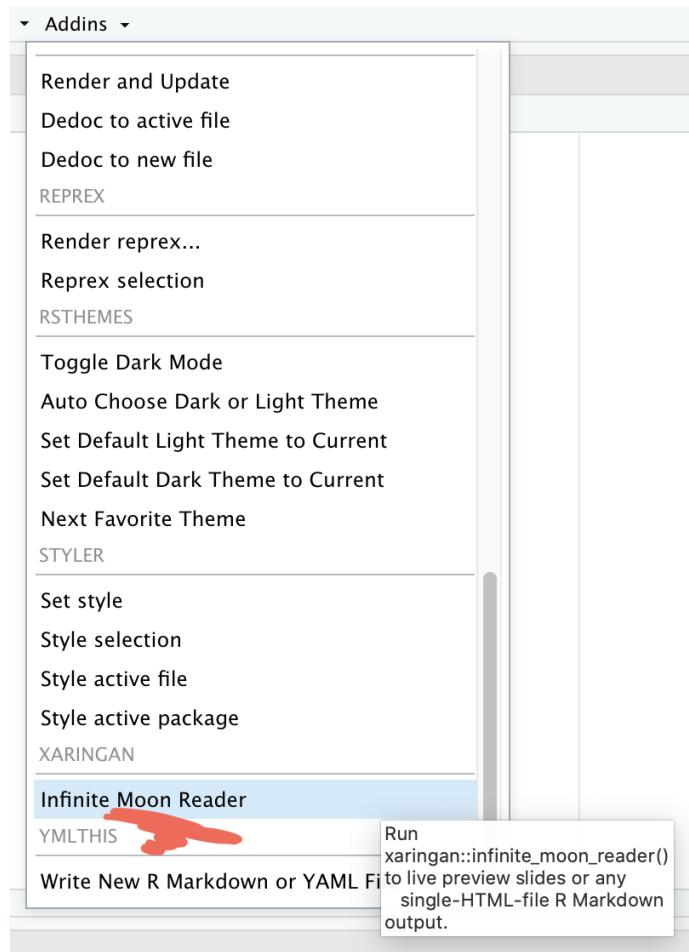
# Template \(\rightarrow\) Ninja Presentation

# Save this Rmd file



```
1 ---  
2 title: "Presentation Ninja"  
3 subtitle: "<br/>with xaringan"  
4 author: "Yihui Xie"  
5 institute: "RStudio, PBC"  
6 date: "2016/12/12 (updated: `r Sys.Date()`)"  
7 output:  
8   xaringan::moon_reader:  
9     lib_dir: libs  
10    nature:  
11      highlightStyle: github  
12      highlightLines: true  
13      countIncrementalSlides: false  
14 ---  
15  
16 background-image:  
17 url("https://upload.wikimedia.org/wikipedia/commons/b/be/Sharin  
gan_triple.svg")  
18 ````{r setup, include=FALSE}  
19 options(htmltools.dir.version = FALSE)
```

# ADDINS \(\rightarrow\) INFINITE MOON READER



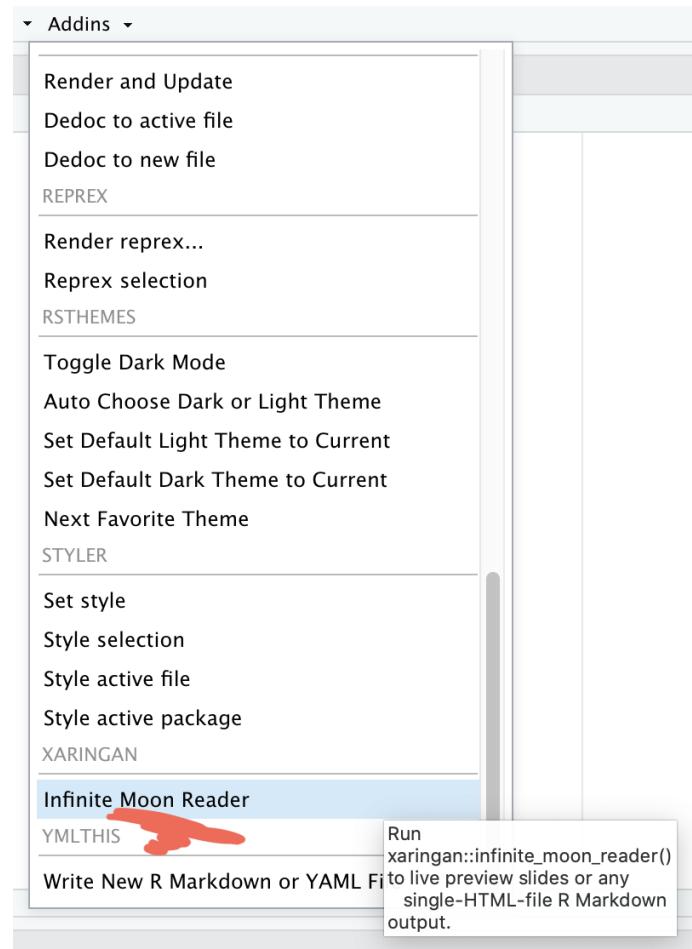
# XARINGAN OUTPUT



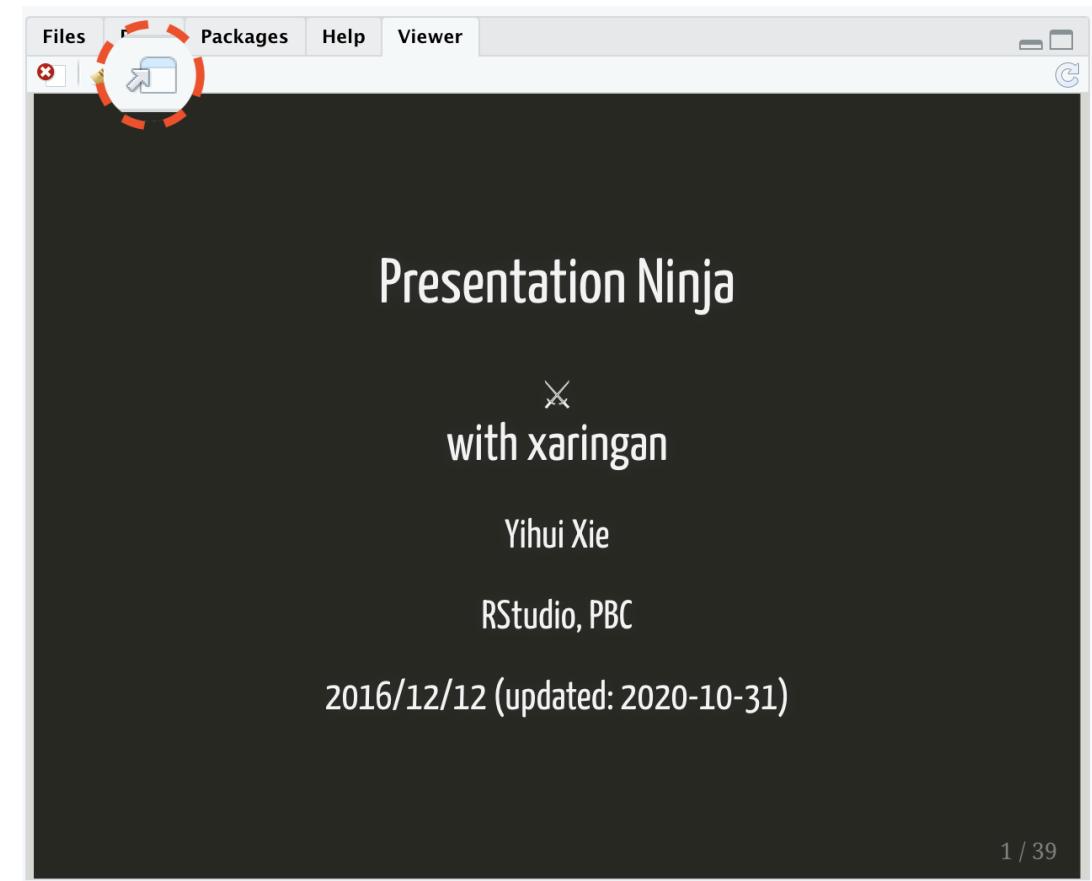
The screenshot shows a presentation slide with the following content:

- Header: Presentation Ninja
- Image: A small icon of a person with a sword.
- Title: with xaringan
- Text: Yihui Xie
- Text: RStudio, PBC
- Date: 2016/12/12 (updated: 2020-10-31)
- Page number: 1 / 39

## ADDINS \(\rightarrow\) INFINITE MOON READER



## XARINGAN SLIDE \(\rightarrow\) BROWSER



- We need to click **Inifinite Moon Reader** only to start the slideshow. To see the

# Using xaringan how to:

1. create a new slide
2. hide an existing slide
3. heading, subheadings, points and normal text
4. include images
  - as background
  - as part of slide
5. make plots
6. include tables
7. in-text R output
8. create columns

1. Use `---` to create a new slide
2. `exclude:true` To hide an existing slide
3. Slide text sizes:
  - `#` for main heading
  - `##` for sub-heading
  - `###` for sub-sub-heading
  - indented `*` for sub-point1
  - indented `*` for sub-point2
  - indented `*` for sub-point3
- `-` for normal text size

# To include images using:

CSS BACKGROUND OPTION:

- `background-image: url("path of the image")` = path of the image
- `background-size: contain, cover, 50%, 70%` = size of the image
- `background-position: left top` = position of the image

# To include images using:

KNITR CHUNK OPTION:

```
knitr::include_graphics("path of the image")
```

# To include plots

```
library(palmerpenguins)
ggplot(penguins, aes(x = species)) +
  geom_bar()
```

# To include tables

```
library(kableExtra)
library(tidyverse)

penguins %>%
  drop_na() %>%
  head() %>%
  kable()
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007

# In-text R output

- penguins data have a sample of  $(n) = 344$  on total 8 variables.
- math expressions

$$(a + b = \sigma - \sum{x}_2^2)$$

# Column division of slide

- left column

- a
  - b
  - c

- right column

- apple
- boy
- cat

# Slide class

- class can be assigned to each slide
- it decides how all elements of one particular slide will look like

# Slide class

- class can be assigned to each slide
- it decides how all elements of one particular slide will look like
- `class: center`

# Slide class

- class can be assigned to each slide
- it decides how all elements of one particular slide will look like
- `class: center, middle, inverse, right`

# Extend the power of xaringan:

- using R packages like [xaringanExtra](#)
- learn little about [CSS](#)
- use [cheatsheets](#)



Congratulations!!!



Thank you