

# Introduction to RStudio

Day 1 - Introduction to Data Analysis with R

Selina Baldauf

Freie Universität Berlin - Theoretical Ecology

February 28, 2025

# Difference between R and RStudio



R is the **programming language** and the **program** that does the actual work

- Can be used with many different programming environments



RStudio is the **integrated development environment (IDE)**

- Provides an interface to R
- Specifically built around R code
- Execute code
- Syntax highlighting
- File and project management
- ...

# Difference between R and RStudio



Analogy and image from [ModernDive Book](#)

## Summary

You can use R without RStudio but RStudio without R would be of little use

# A quick tour around RStudio

# A quick tour around RStudio

The screenshot displays the RStudio IDE interface. The main editor window shows a script named 'penguin\_script.R' with the following R code:

```
1 # Script to plot some data
2 library(ggplot2)
3 library(palmerpenguins)
4
5 head(penguins)
6
7 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g, color = species)) +
8   geom_point() +
9   geom_smooth(method = "lm") +
10  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
11  theme_bw()
12
```

The console window at the bottom left shows the output of the script, including the R license notice, version information, and the execution of the code:

```
R 4.0.3 · C:/Users/Selina_User/Files_Selina/Repos/02_workshops/intro-to-r/
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.

Warning messages:
1: package 'devtools' was built under R version 4.0.5
2: package 'usethis' was built under R version 4.0.5
3: package 'reprex' was built under R version 4.0.5
> variableA <- c(1,2,3)
> variableB <- 10.5
> someData <- data.frame(a=1:10, b=1:10)
> 2+2
[1] 4
> print("hello")
[1] "hello"
>
```

The environment pane on the right shows the 'Global Environment' with a dataset 'someData' containing 10 observations of 2 variables. The 'Values' section displays the first three rows of the data frame:

variableA	num
1	1
2	2
3	3

# Console pane

- Execute R code
- Output from R code in scripts is printed there
- Type a command into the console and execute with **Enter/Return**



Tip

Use arrow keys to bring back last commands

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 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.

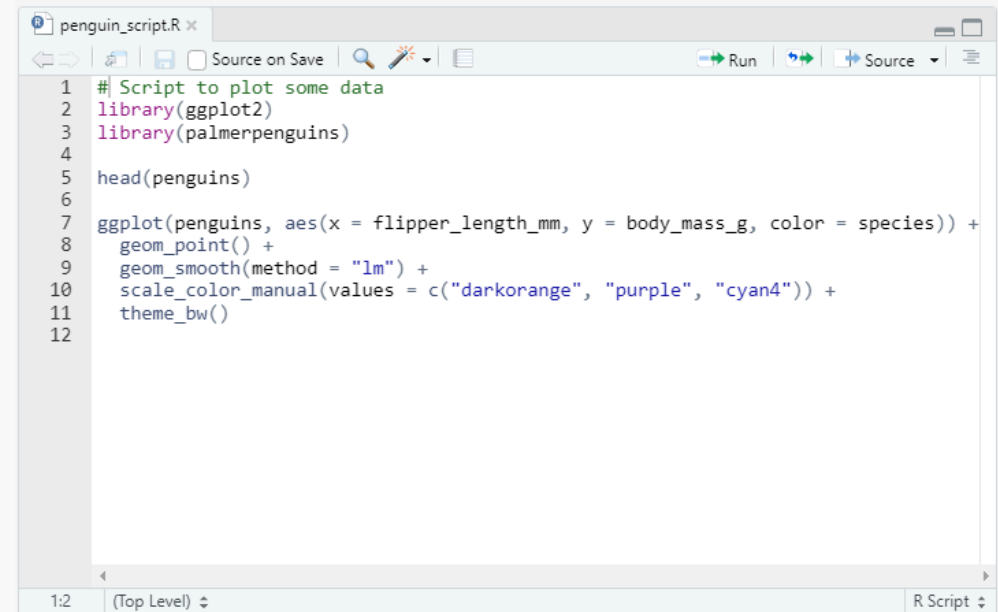
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

Warning messages:
1: package 'devtools' was built under R version 4.0.5
2: package 'usethis' was built under R version 4.0.5
3: package 'reprex' was built under R version 4.0.5
> 4+4
[1] 8
> print("hello")
[1] "hello"
> |
```

# Script pane

- Write scripts with R code
  - Scripts are text files with R commands (file ending `.R`)
  - Use scripts to save commands for reuse



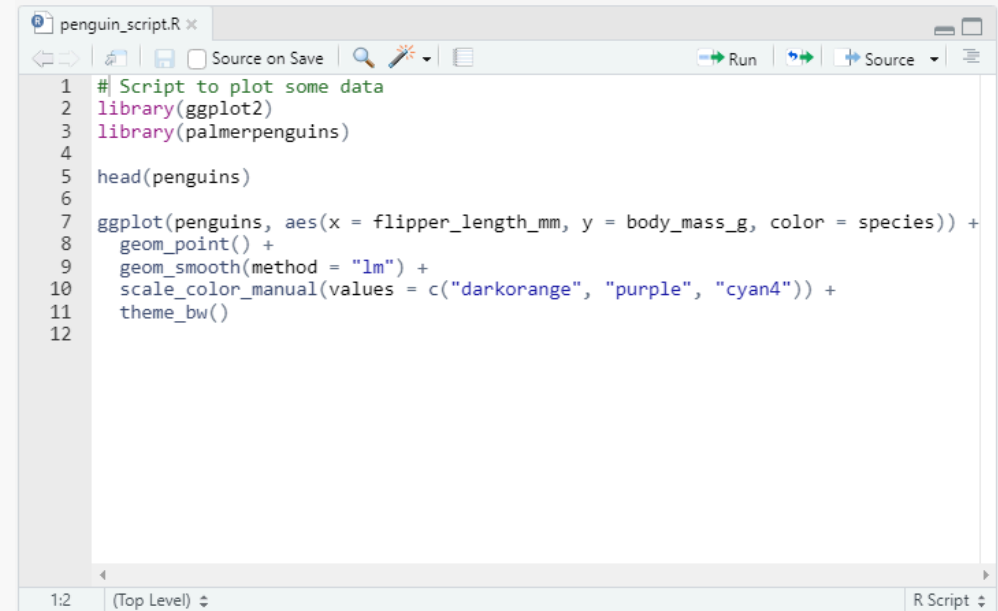
```
penguin_script.R x
Source on Save
Run
Source

1 # Script to plot some data
2 library(ggplot2)
3 library(palmerpenguins)
4
5 head(penguins)
6
7 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g, color = species)) +
8   geom_point() +
9   geom_smooth(method = "lm") +
10  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
11  theme_bw()
12
```

1:2 (Top Level) ↕ R Script ↕

# Script pane

- Create a new R script:  
File -> New File -> R Script
- Save an R script:  
File->Save (Ctrl/Cmd + S)
- Run code line by line with **Run button**  
(Ctrl+Enter/Cmd+Return)
- You can open multiple scripts



```
penguin_script.R x
1 # Script to plot some data
2 library(ggplot2)
3 library(palmerpenguins)
4
5 head(penguins)
6
7 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g, color = species)) +
8   geom_point() +
9   geom_smooth(method = "lm") +
10  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
11  theme_bw()
12
```

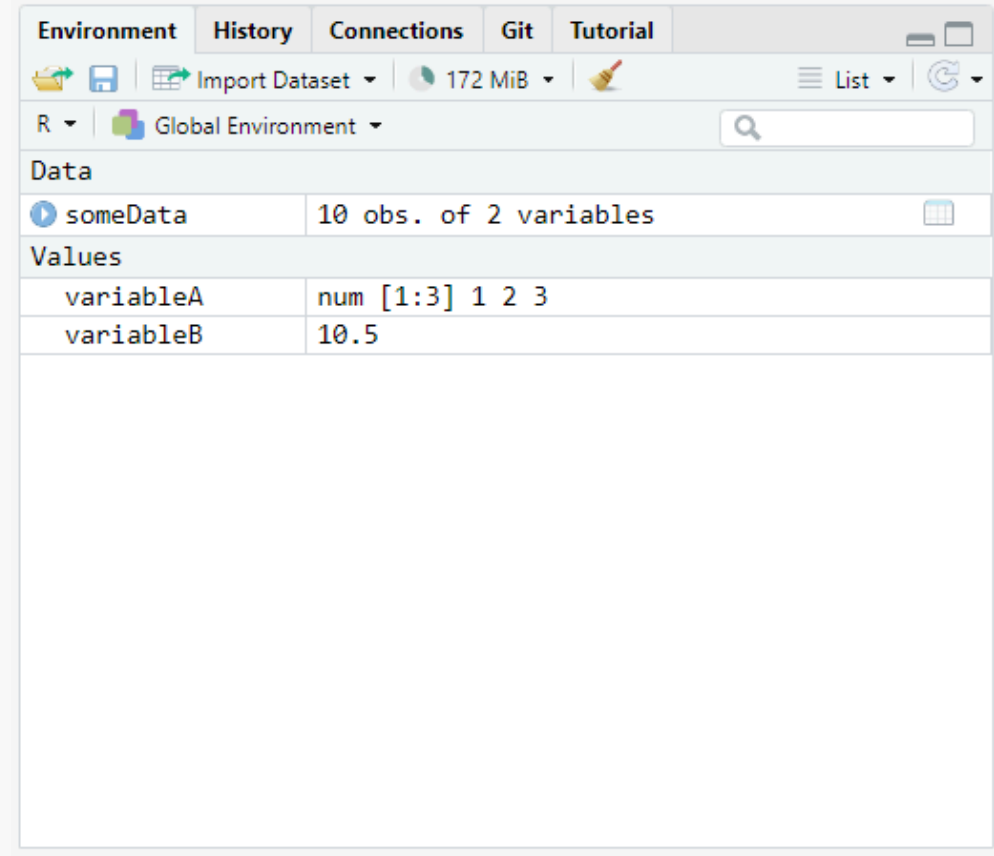
## Summary

Use **scripts** for all your analysis and for commands that you want to save.  
Use **console** for temporary commands, e.g. to test something.



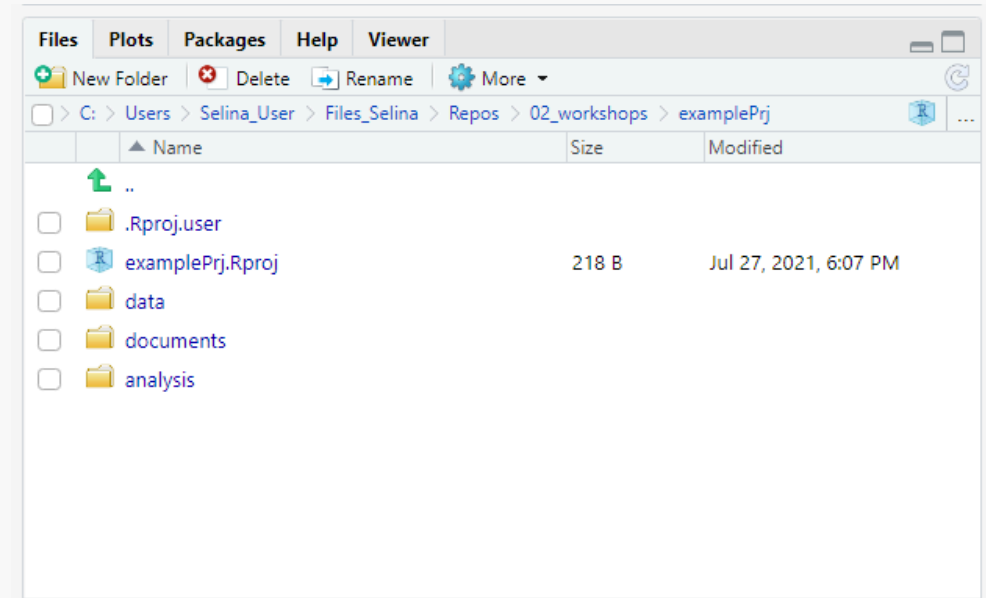
# Environment pane

- Shows objects currently present in the R session
- Is empty if you start R



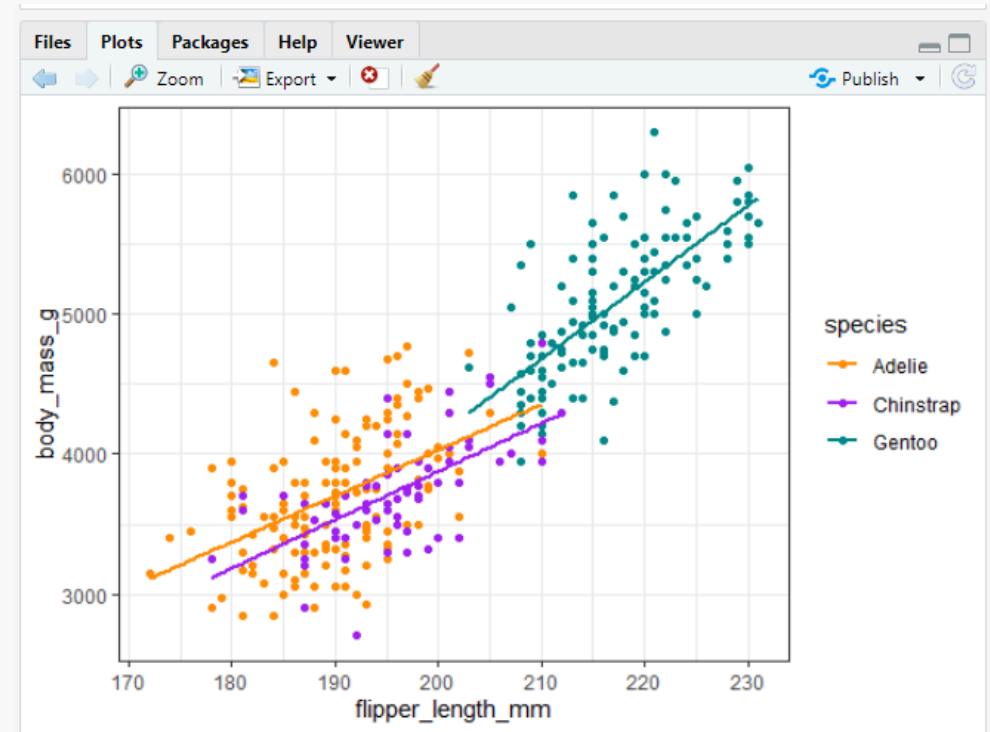
# Files pane

- Similar to Explorer/Finder
- Browse project structure and files
  - Find and open files
  - Create new folders
  - Delete files
  - Rename files
  - ...
- Practical if you don't want to switch between File Explorer and RStudio all the time



# Plot pane

- Plots that are created with R will be shown here



# Project oriented workflow

How to use RStudio to organize your projects

# Project oriented workflow

- One directory with all files relevant for project
  - Scripts, data, plots, documents, ...

```
MyProject
|
|- data
|
|- docs
|   |
|   |- notes
|   |
|   |- reports
|
|- R
|   |
|   |- clean_data.R
|   |
|   |- statistics.R
|
```

Example project structure

# Project oriented workflow

- One directory with all files relevant for project
  - Scripts, data, plots, documents, ...
- An RStudio project is just a normal directory with an **.Rproj** file

```
Project
|
|- data
|
|- docs
|   |
|   |- notes
|   |
|   |- reports
|
|- R
|   |
|   |- clean_data.R
|   |
|   |- statistics.R
|
|- MyProject.RProj
```

Example RStudio project structure

# Project oriented workflow

## Advantages of using RStudio projects

- Easy to navigate in R Studio (**File pane**)
- Easy to find and access scripts and data in RStudio
- Project root is working directory
- Open multiple projects simultaneously in separate RStudio instances

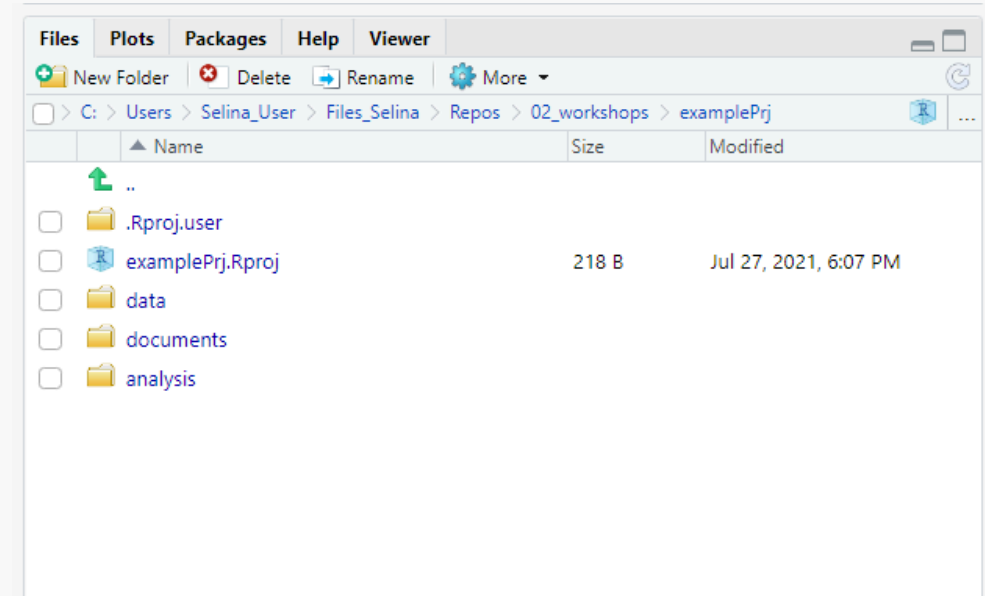
```
Project
|
|- data
|
|- docs
|   |
|   |- notes
|   |- reports
|
|- R
|   |
|   |- clean_data.R
|   |- statistics.R
|
|- *.Rproj
```

Example RStudio project structure

# Create an RStudio project

Create a project from scratch:

1. **File -> New Project -> New Directory -> New Project**
2. Enter a directory name (this will be the name of your project)
3. Choose the Directory where the project should be initiated
4. **Create Project**

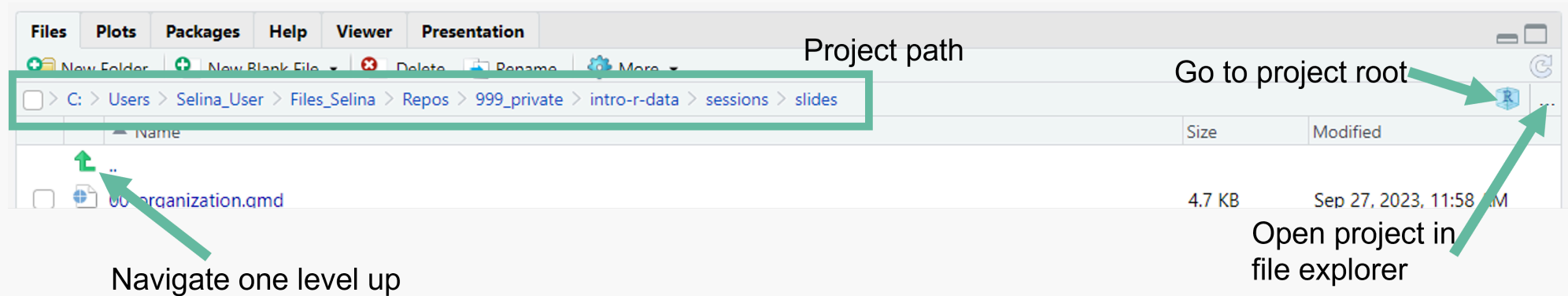


Example RStudio project structure in the Files pane

RStudio will now create and open the project for you.

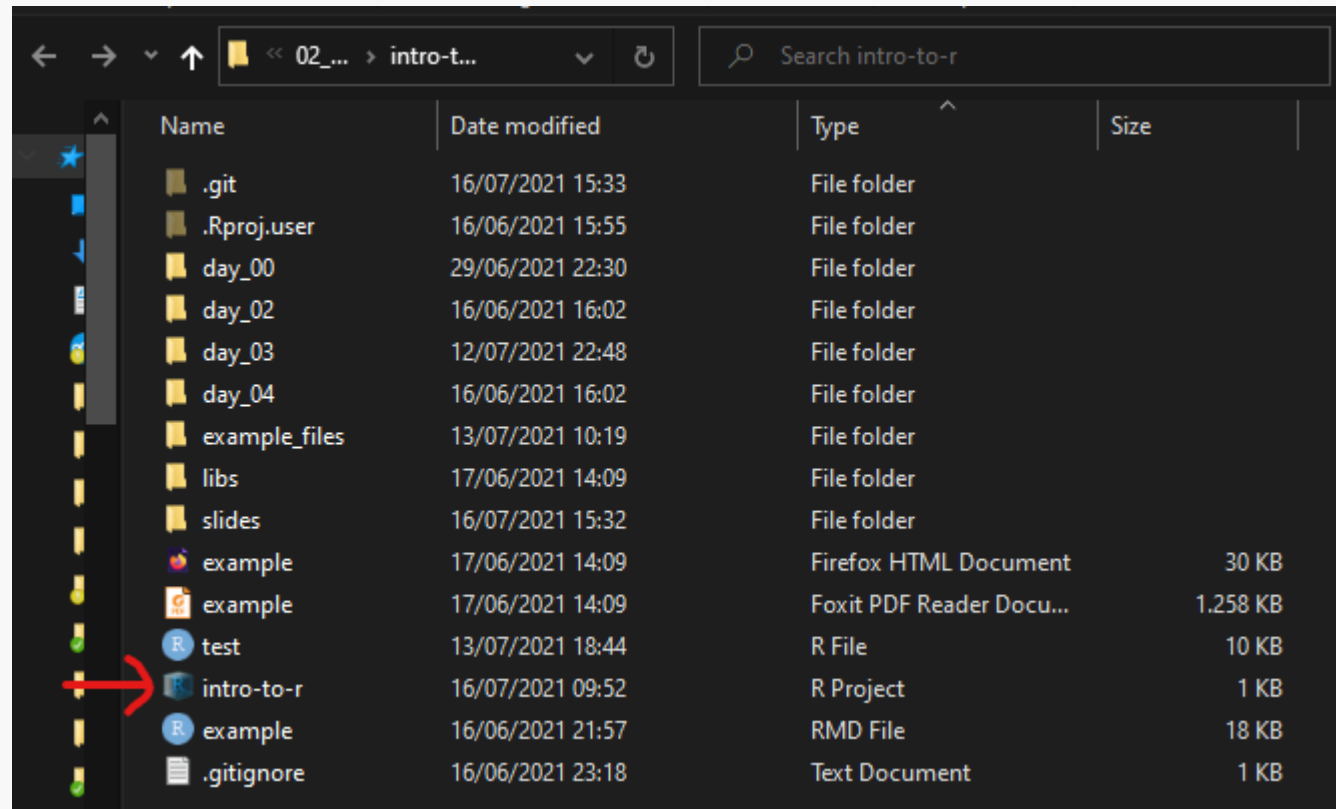


# Navigate an RStudio project



# Open a project from outside RStudio

To open an RStudio project from your file explorer/finder, just double click on the .Rproj file



# Open a project inside RStudio

To open an RStudio project from RStudio, click on the project symbol on the top right of R Studio and select the project from the list.



# A tip before we get started

Learn the most important keyboard shortcuts of R Studio.

Find all shortcuts under **Tools -> Keyboard Shortcuts Help**

- Save active file: Ctrl/Cmd + S
- Run current line: Ctrl/Cmd + Enter
- Create new R Script: Ctrl/Cmd + N
- Undo: Ctrl/Cmd + Z
- Redo: Ctrl/Cmd + Y
- Copy/Paste: Ctrl/Cmd + C/V

# Now you

## Task 1 (20 min)

Set up your own RStudio project for this workshop

Find the task description [here](#)