

Introduction to RStudio

Instructor: **Selina Baldauf**

Freie Universität Berlin - Theoretical Ecology



2021-08-01 (updated: 2022-03-15)

Difference between R and RStudio



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

- Can be use with many different programming environments (But RStudio is the best for R)



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
- ...

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

Difference between R and RStudio

R is like the engine



RStudio is more like the dashboard, etc.



analogy and image from [ModernDive Book](#)

Basic idea

- Write precise instructions telling the computer what to do step by step
 - For this you need to use the language R
- Tell R to execute these instructions
 - R will return the result of your instructions (or an error message)
- R code is basically just text that can be saved with file extension `.R`

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 shows the output of the script execution:

```
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 'rexprex' 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 current environment with the following variables:

Variable	Value
someData	10 obs. of 2 variables
variableA	num [1:3] 1 2 3
variableB	10.5

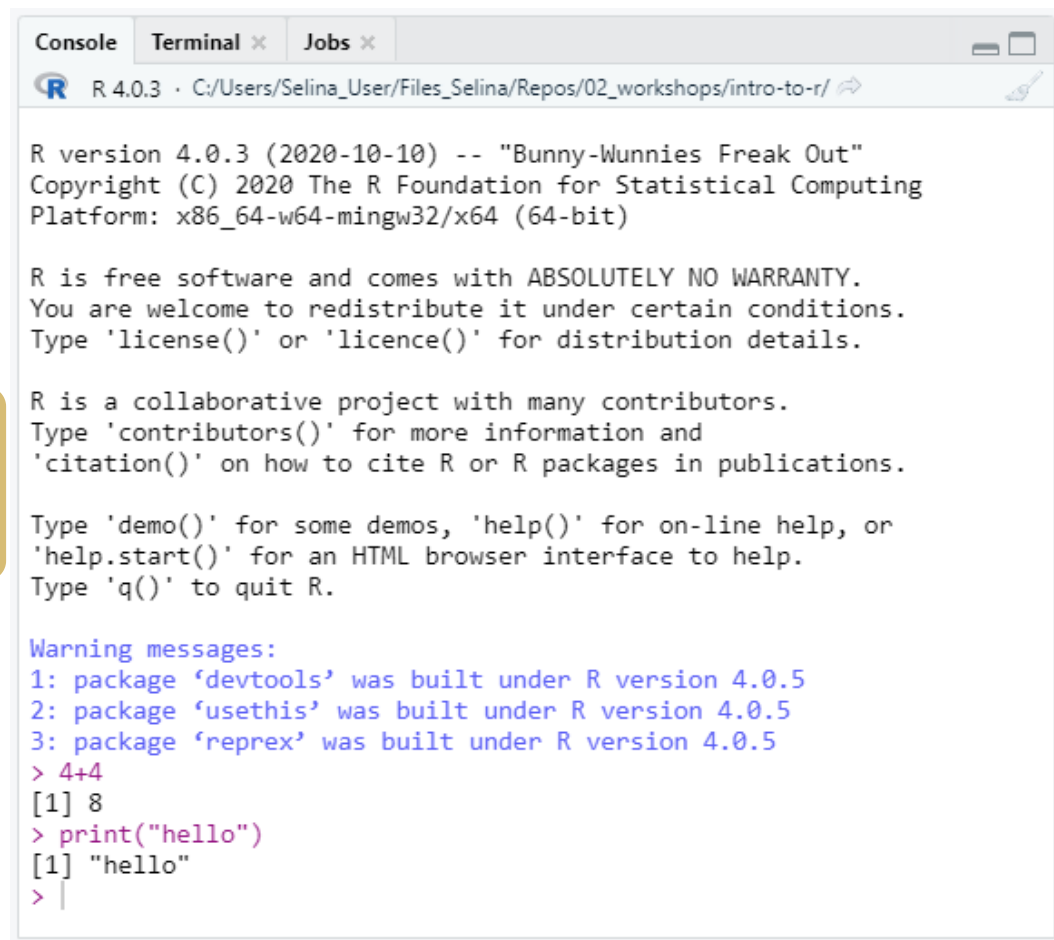
The Files pane at the bottom right shows the project structure:

- ..
- .gitignore
- day_00
- day_01
- day_02
- day_03
- day_04
- example_files
- example.html
- example.Rmd
- intro-to-r.Rproj
- libs

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`

💡 Use arrow keys to bring back last commands



```
Console Terminal x Jobs x
R 4.0.3 · C:/Users/Selina_User/Files_Selina/Repos/02_workshops/intro-to-r/

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.

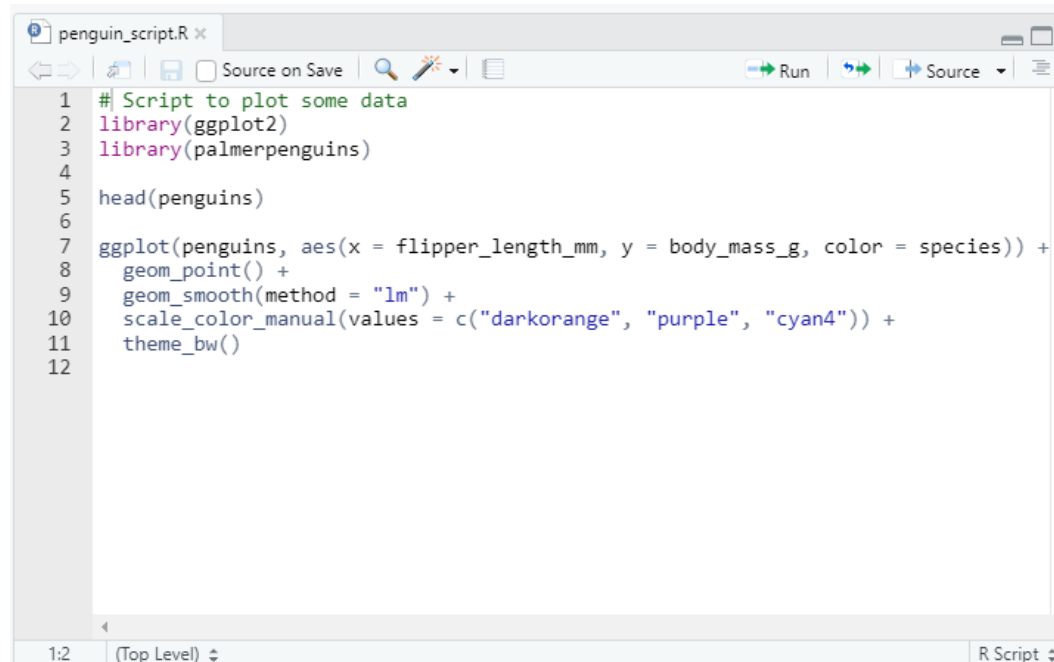
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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



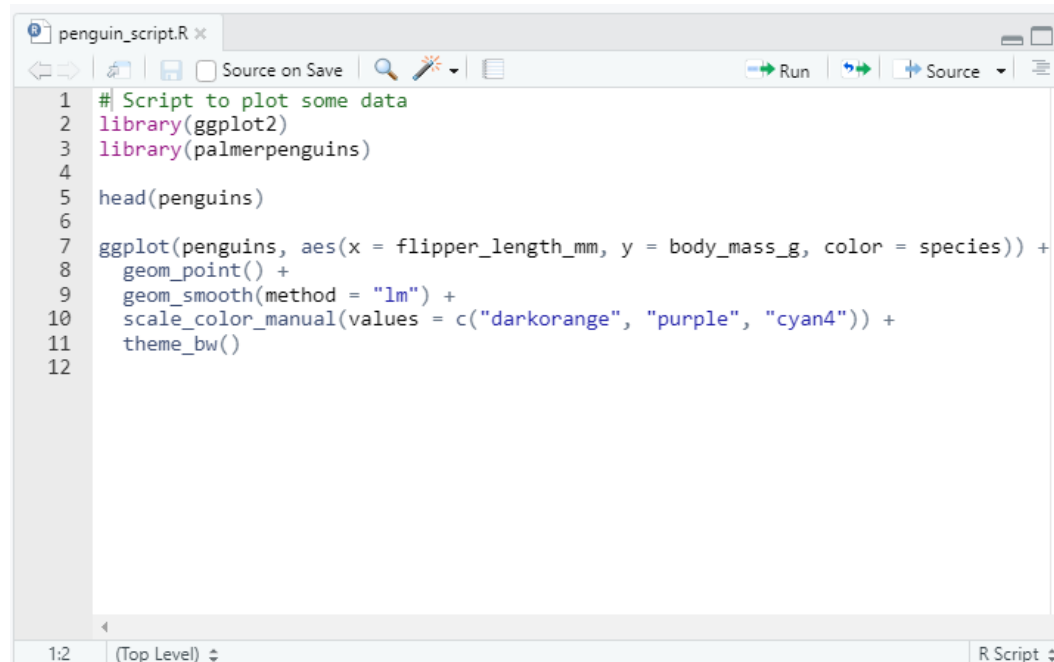
The screenshot shows an R script editor window titled 'penguin_script.R'. The script contains the following R code:

```
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 editor interface includes a toolbar with icons for running, saving, and searching, and a status bar at the bottom indicating the current line (1:2) and the script's context (Top Level).

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 at the same time



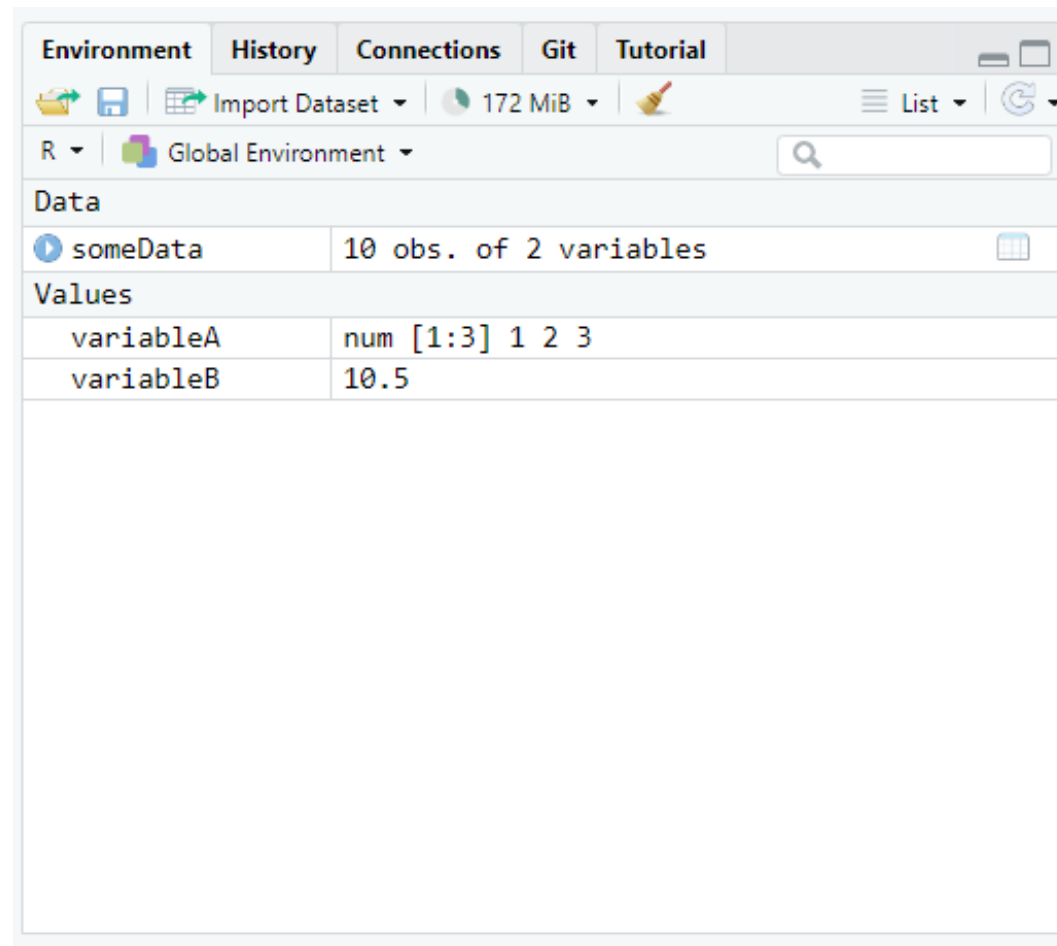
```
1 # Script to plot some data
2 library(ggplot2)
3 library(palmerpenguins)
4
5 head(penguins)
6
7 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g, color = species)) +
8   geom_point() +
9   geom_smooth(method = "lm") +
10  scale_color_manual(values = c("darkorange", "purple", "cyan4"))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

💡 Use **scripts** for all your analysis and for commands that you want to save.

💡 Use **console** for temporary commands, e.g. to test something.

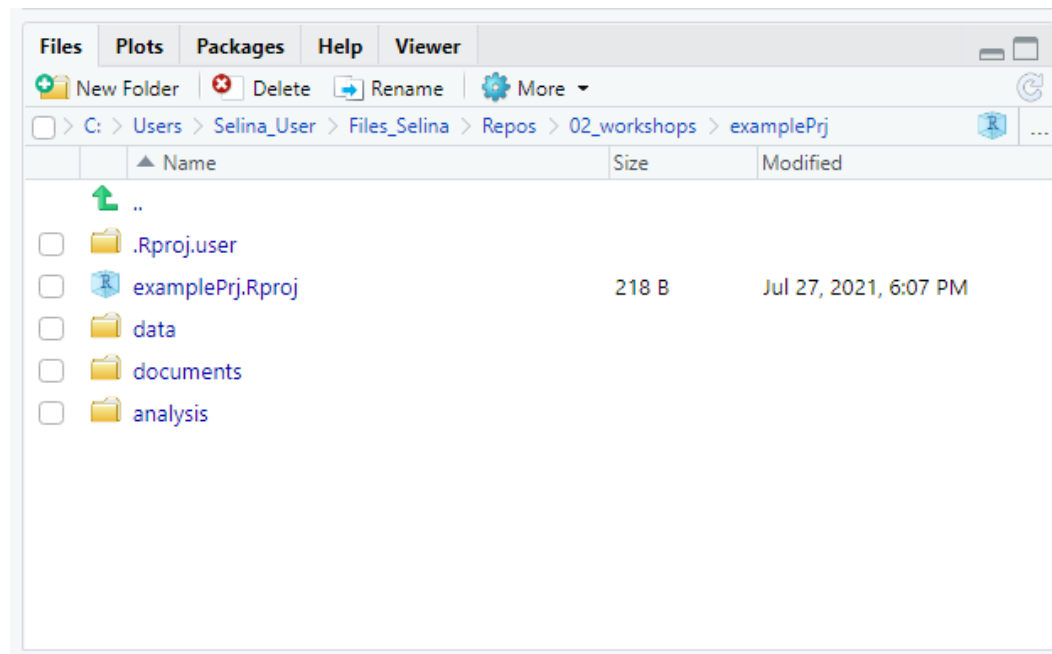
Environment pane

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



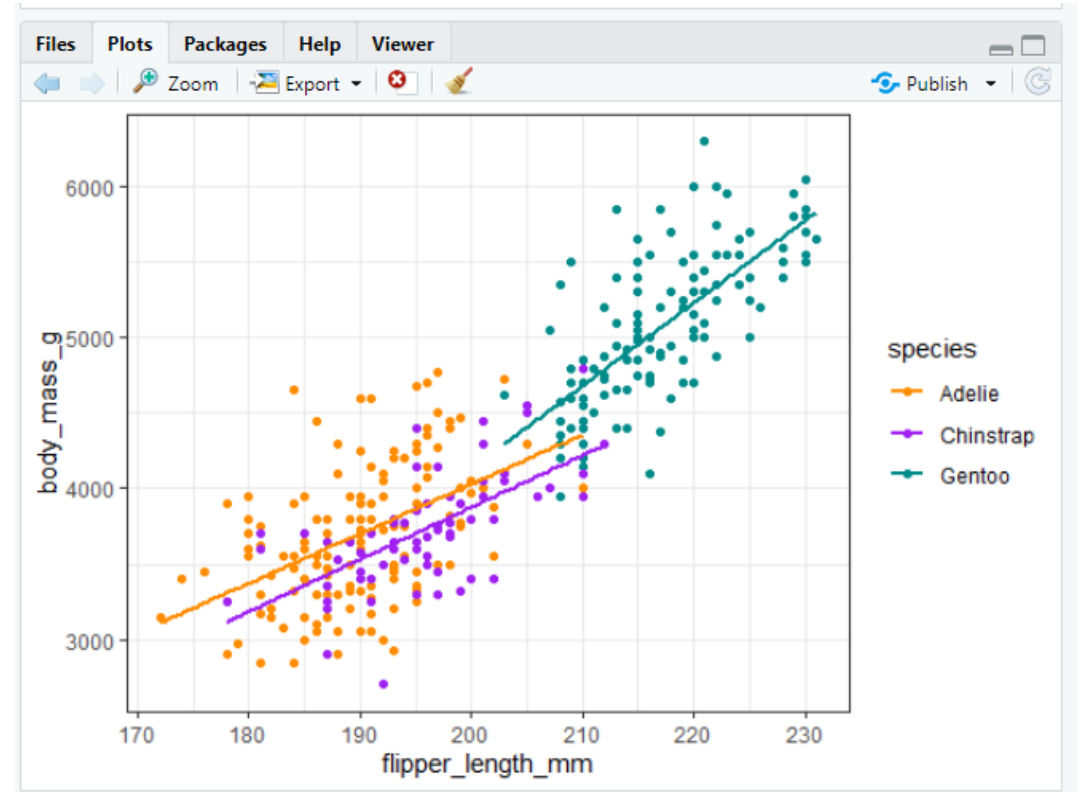
Files pane

- 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
- You can export plots by clicking on export button
 - But better to do it by code



Project oriented workflow with RStudio

- One directory with all files relevant for project
 - Scripts, data, plots, documents, ...
- An RStudio project is just a normal directory with an `*.Rproj` file
- Advantages of using RStudio projects
 - Easy to navigate in R Studio (`File` pane)
 - Easy to find and access scripts in RStudio
 - Project root is working directory
 - Open multiple projects simultaneously in separate RStudio instances

```
Project
|
|- data
|
|- documents
|   |
|   |- notes
|   |- reports
|- analysis
|   |
|   |- clean_data.R
|   |- statistics.R
|- *.RProj
```

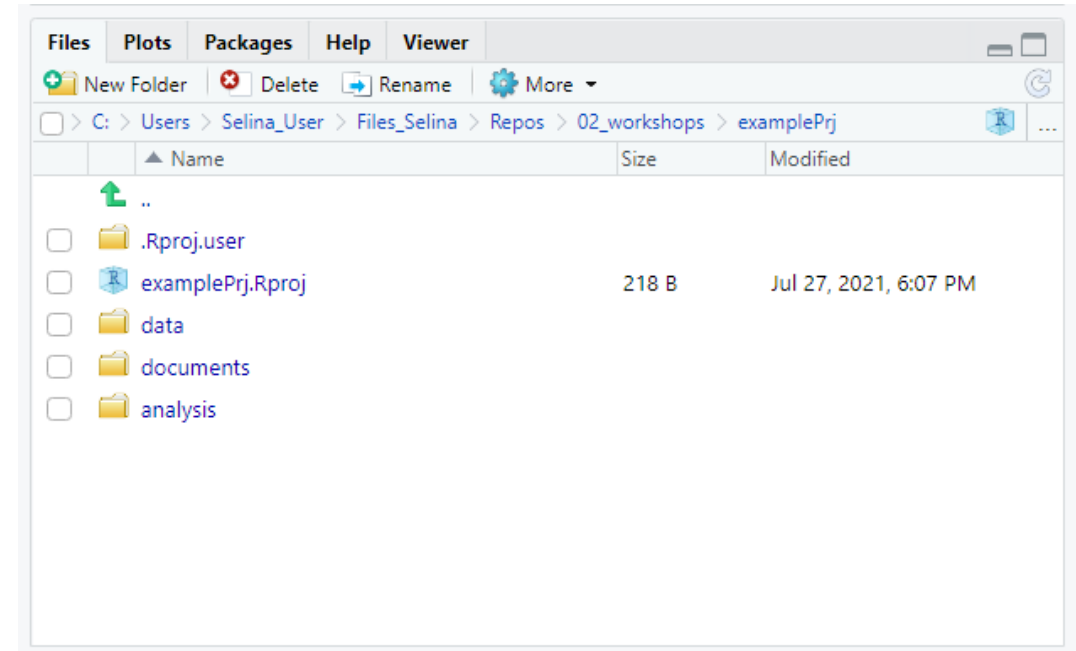
Example project structure

Create an RStudio project

Create a project from scratch:

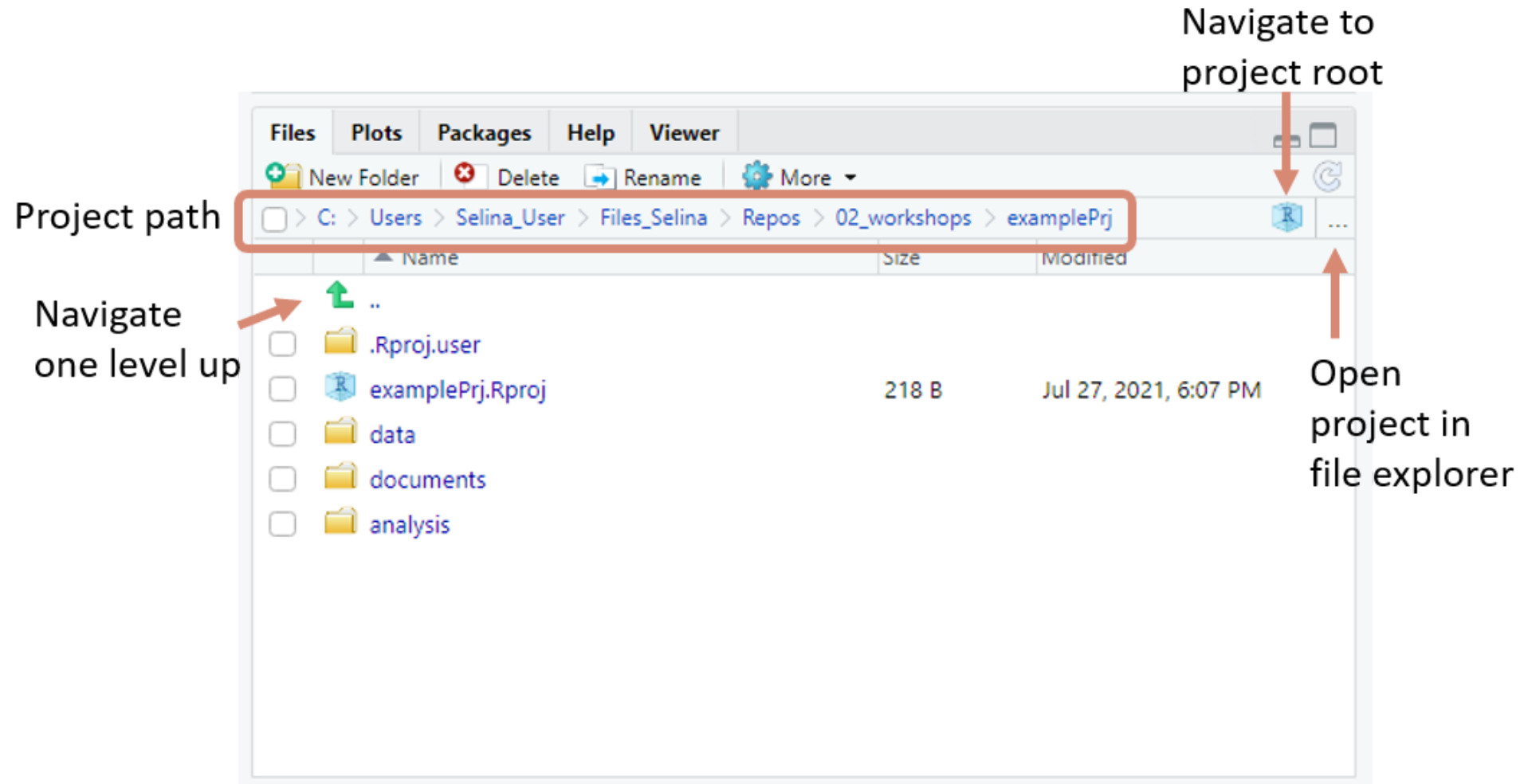
1. File -> New Project -> New Directory -> New Project
2. Enter a directory name (this will be the name of your project)
3. Choose the Directory where the project should be initiated
4. Create Project

RStudio will now create and open the project for you.



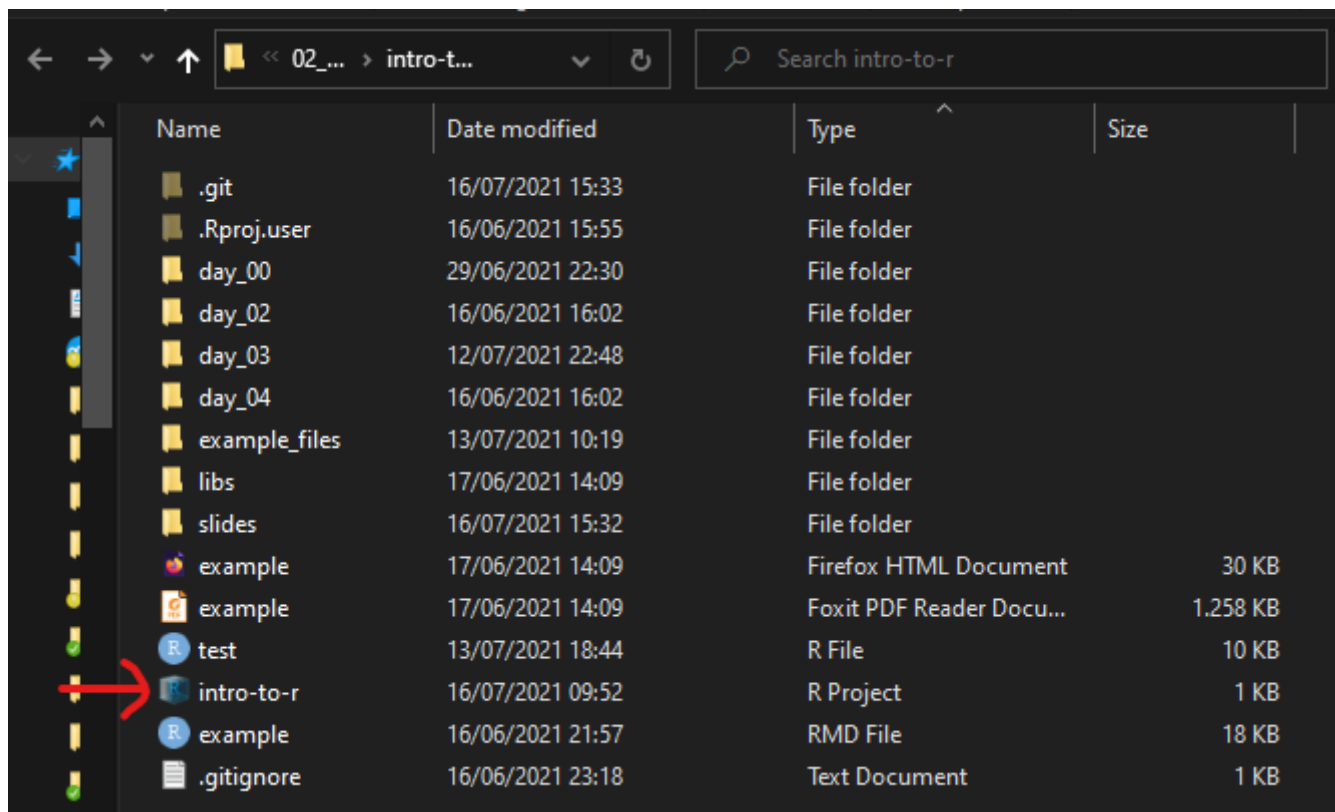
Example project structure in RStudio

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.

Now you

Task 1: Set up your own RStudio project for this workshop (25 min)

Find the task description [here](#)