# Introduction to RStudio

## Day 1 - Introduction to Data Analysis with R

Selina Baldauf

Freie Universität Berlin - Theoretical Ecology

October 7, 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

# 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

```
Console  Terminal ×  Jobs ×
R  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"
>
```
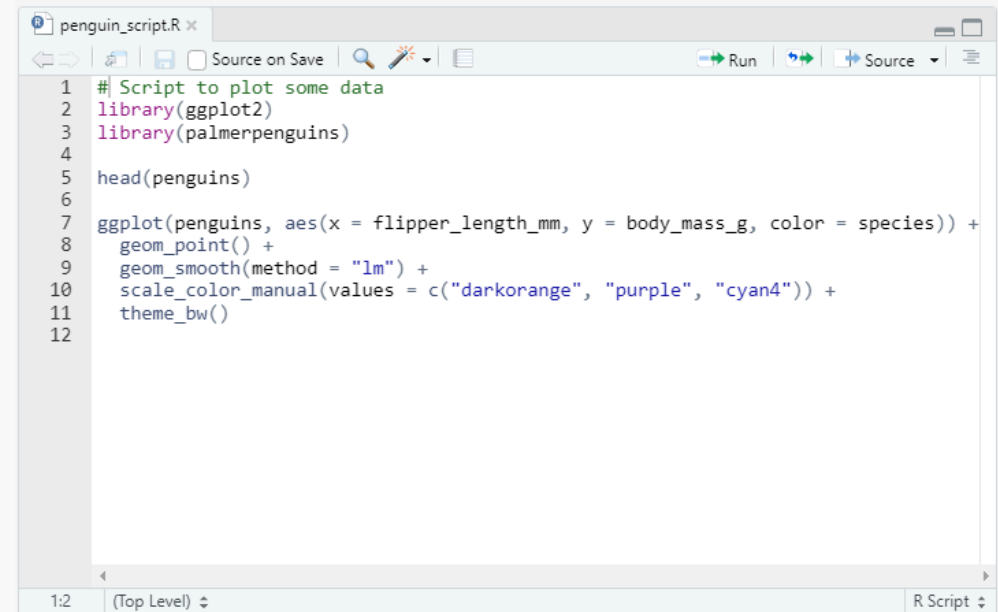
# Script pane

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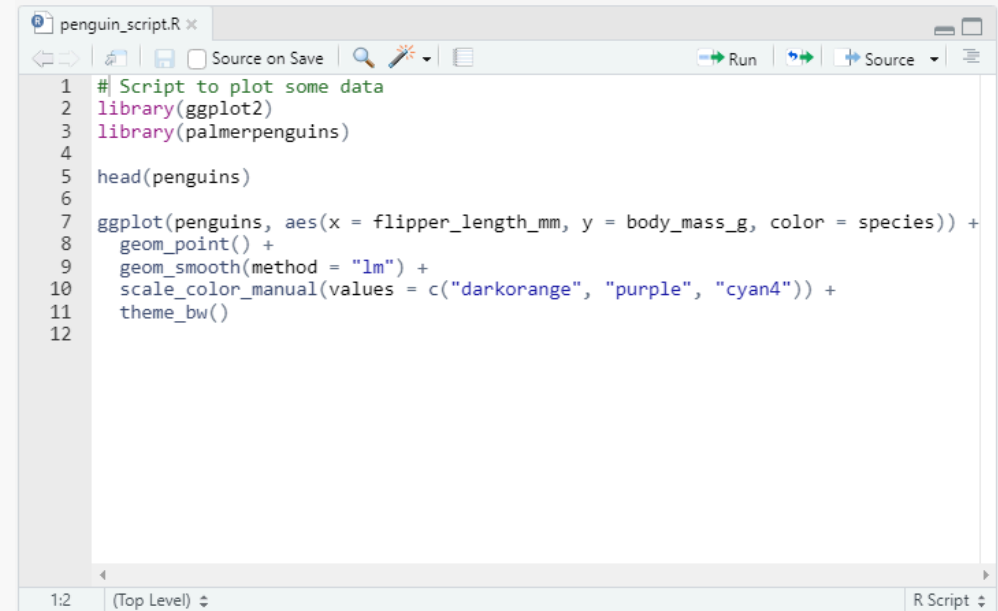  - Use scripts to save commands for reuse

# 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)**
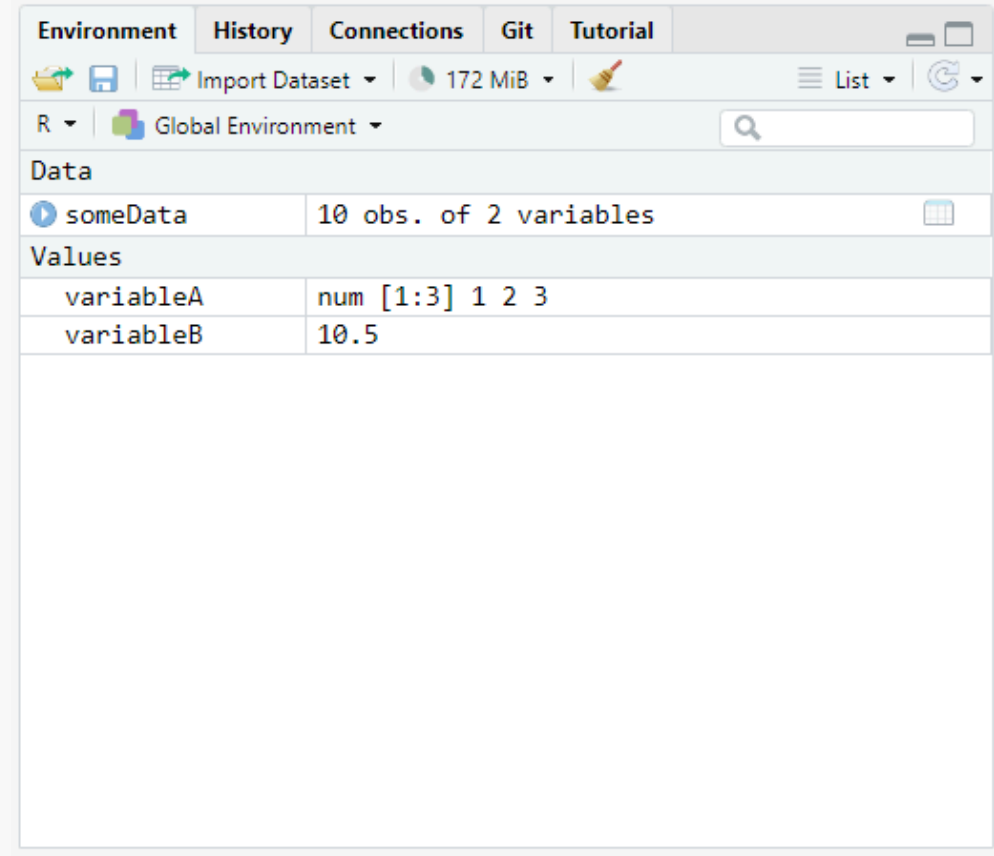
- You can open multiple scripts



---

ⓘ 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

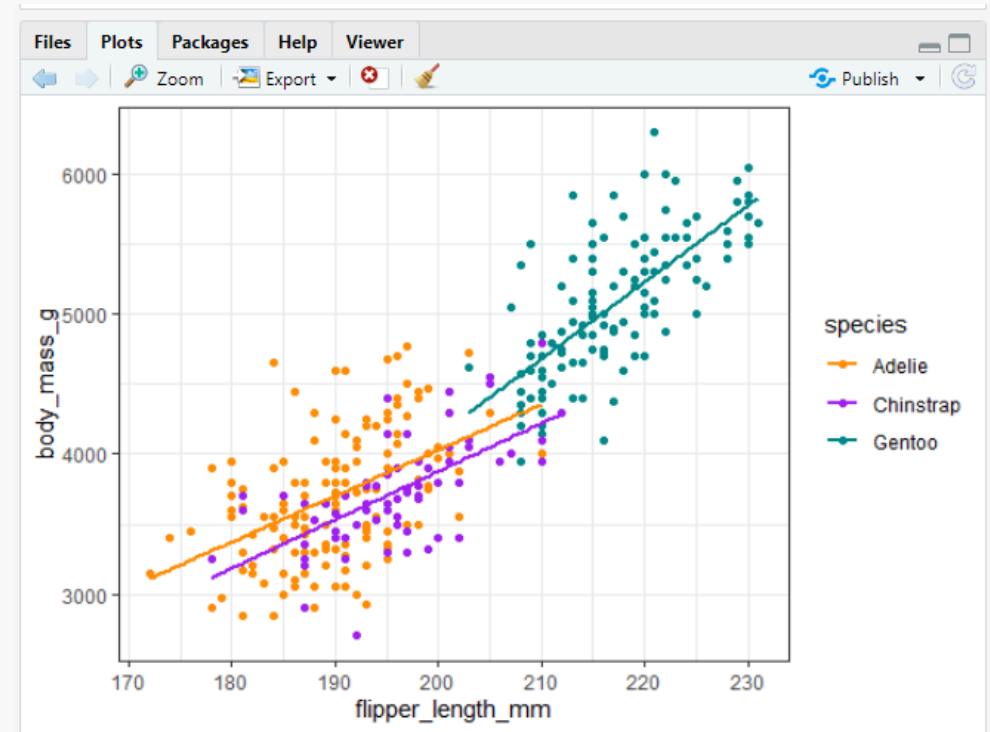# Files pane

- 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
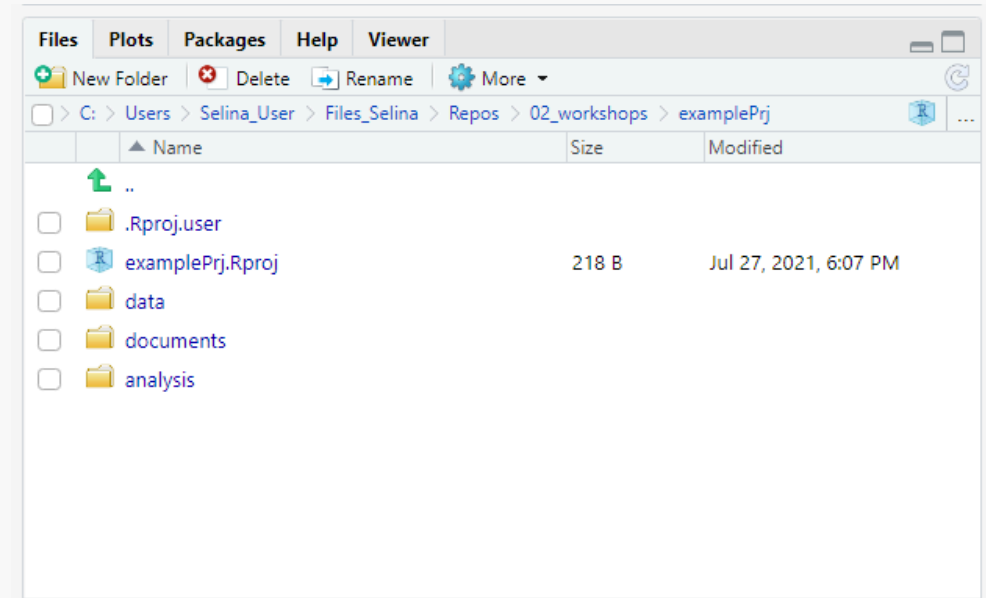
# Plot pane

- 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

```
MyProject
|
|- 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

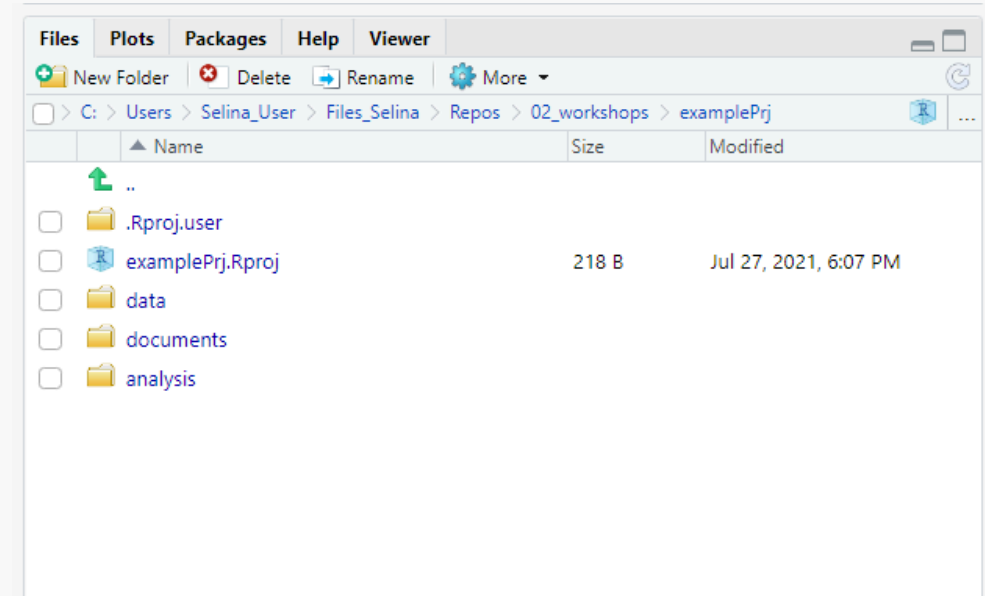- Open multiple projects simultaneously in separate RStudio instances

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

Example RStudio project structure

# Create an RStudio project
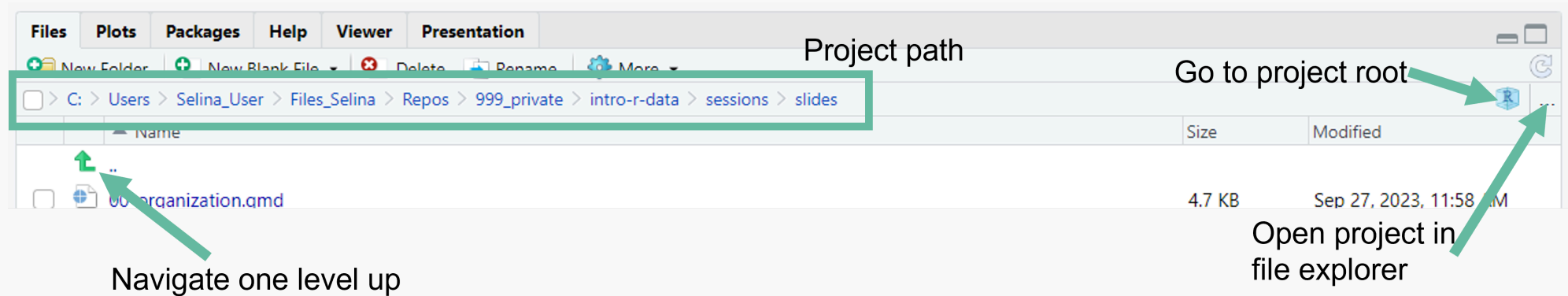
Create a project from scratch:

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**

RStudio will now create and open the project for you.



Example RStudio project structure in the Files pane

# Navigate an RStudio project



Project path

Go to project root

C: > Users > Selina_User > Files_Selina > Repos > 999_private > intro-r-data > sessions > slides

| Name | Size | Modified |
|---|---|---|
| .. | | |
| 00_organization.qmd | 4.7 KB | Sep 27, 2023, 11:58 M |

Navigate one level up

Open project in file explorer

# Open a project from outside RStudio

To open an RStudio project from your file explorer/finder, just double click on the
**.Rproj** file
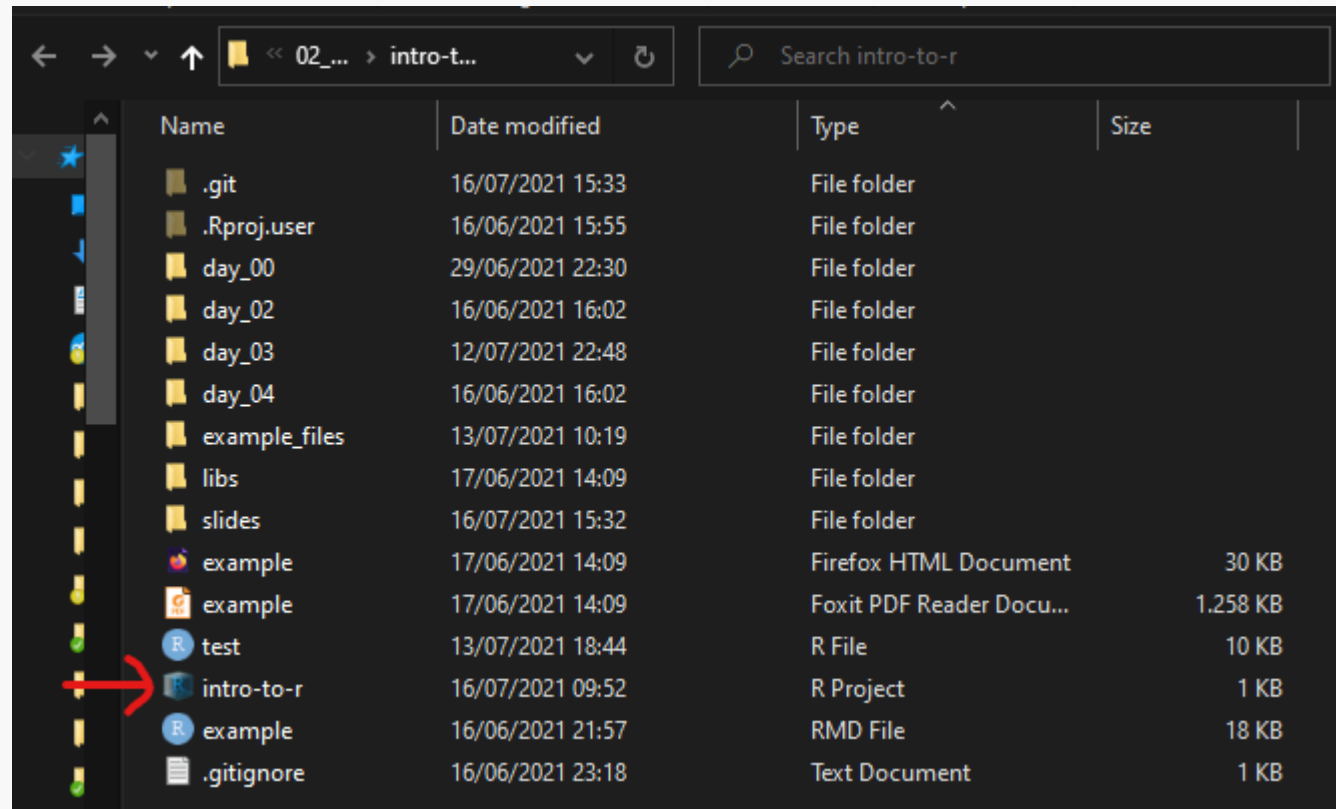
# Open a project inside RStudio

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

<span style="background-color:#a8c8e8">Task 1 (15 min)</span>

Set up your own RStudio project for this workshop

**Find the task description** here