

Kickstart with R:Your Gateway to Data Science

IOC-R Week1

Table of contents

Unlock the Power of Your Data with	2
Welcome to R!	2
The R Console	3
Rstudio	3
Rstudio Login	3
Rstudio IDE	4
Getting Started in R	5
Some Configuration	6
Some Configuration	7
Some Configuration	8
Setting Up an R Project	10
Create an R Project	10
Understanding File Paths	10
A Well Organized Folder	11
Where Are You?	11
Import Dataset	12
Automated Reporting	14
Quarto	14
An Example	14
Need Some Help?	15
Let's Practice !	16
Today's Goals	16

Unlock the Power of Your Data with



Welcome to R!

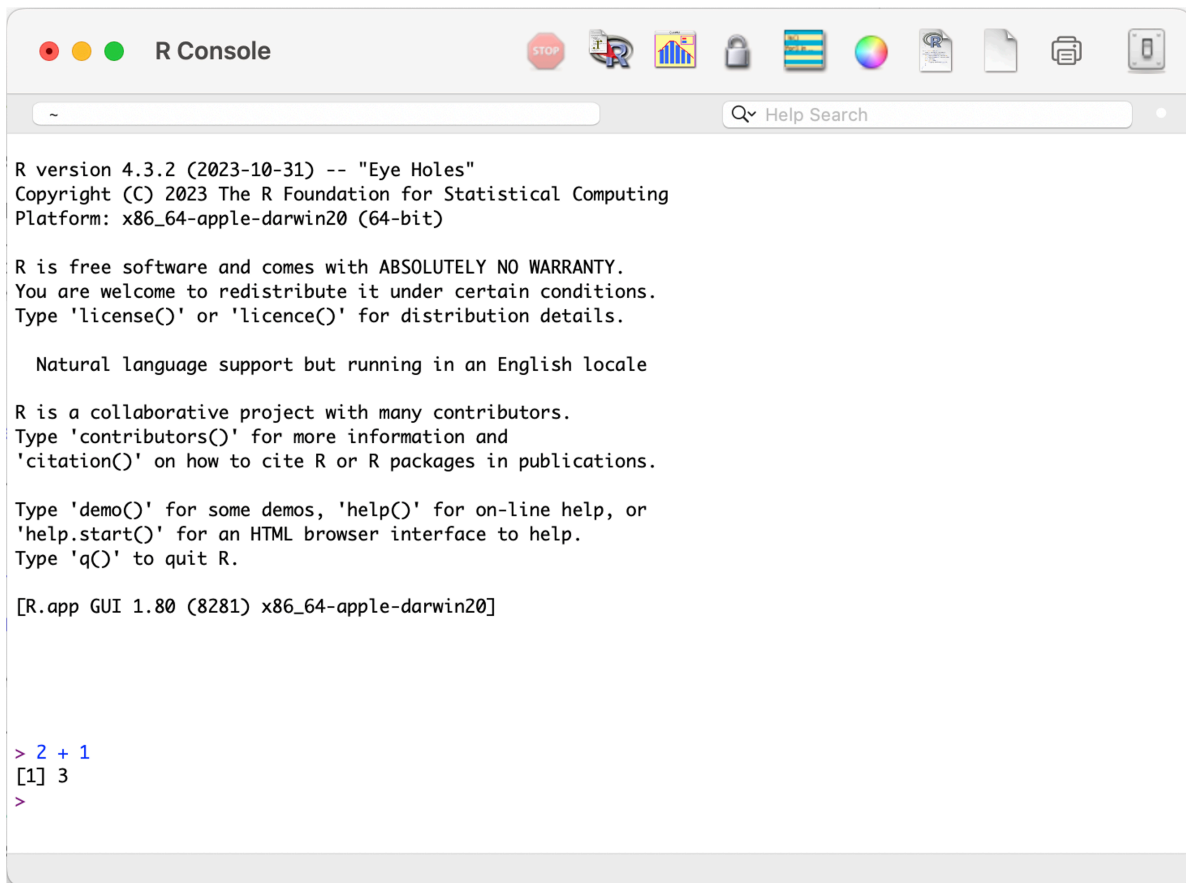
Why R?

- Biologist friendly!
- Free and open-source
- Supported by a huge community

What you can do with R?

- Analyze your data
- Visualize your findings
- Automate your workflow
- Share reproducible research

The R Console



The screenshot shows the R Console window with a title bar that says "R Console". The window contains the following text:

```
R version 4.3.2 (2023-10-31) -- "Eye Holes"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin20 (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.

Natural language support but running in an English locale

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.

[R.app GUI 1.80 (8281) x86_64-apple-darwin20]

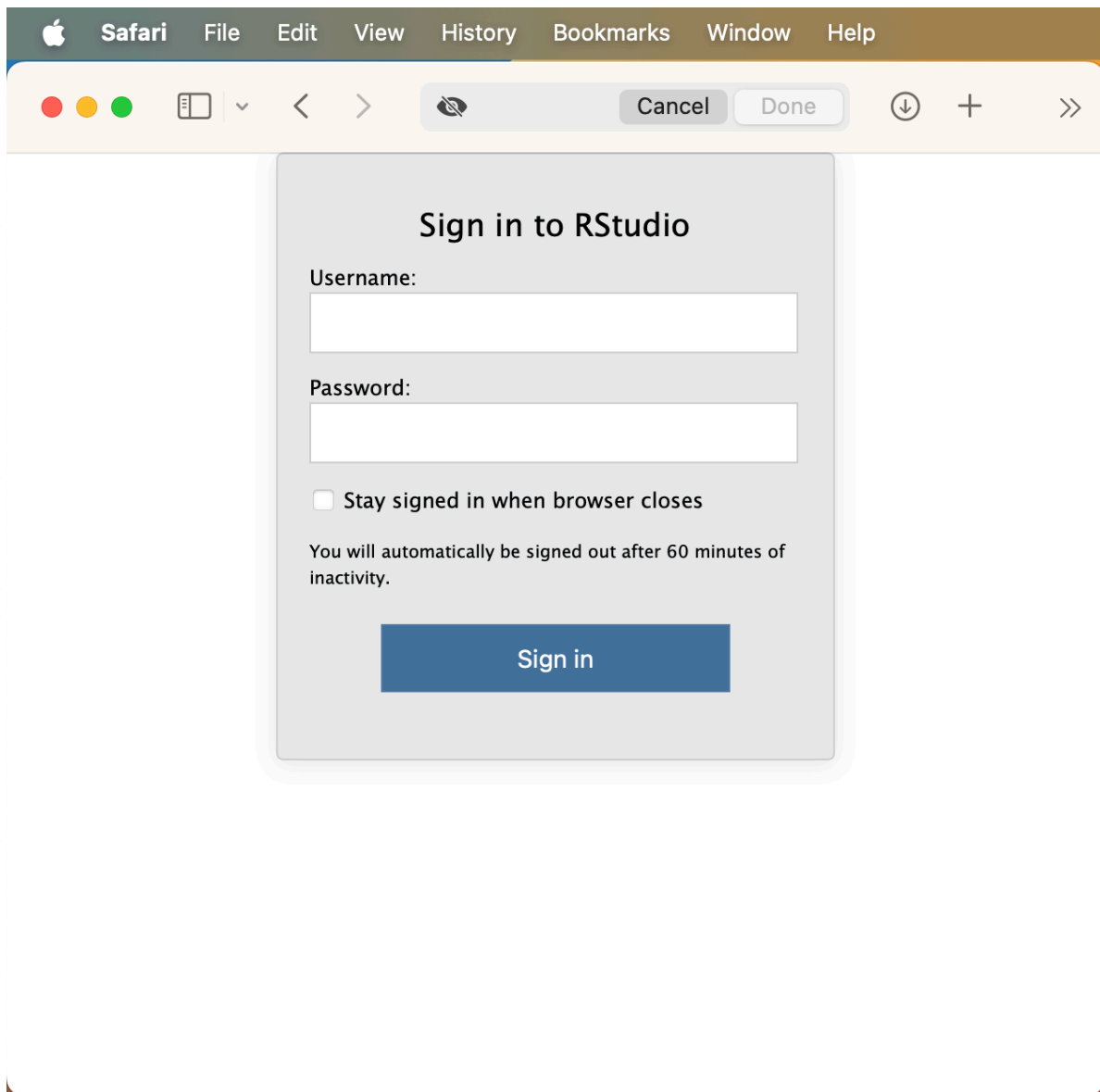
> 2 + 1
[1] 3
>
```

Type a command after the prompt “>” and type “Enter” to execute it.

Rstudio

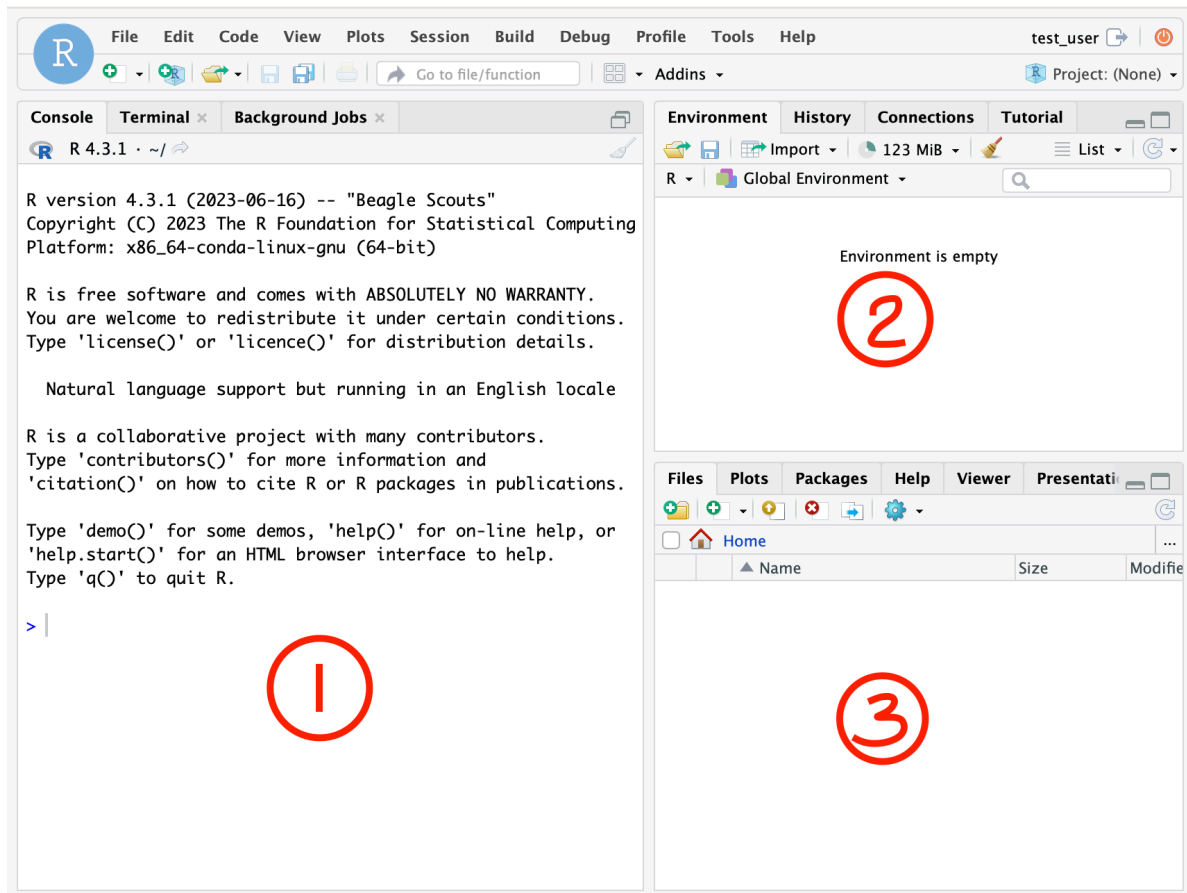
Rstudio Login

Connect to your RStudio server account.



Rstudio IDE

Integrated Development Environment (IDE)



1. Console: where R “talks back” to you.
2. Environment/History: tracks your variables and past commands.
3. Files/Plots/Packages: managing your files, viewing graphs, or installing tools.

Rstudio [cheat sheet](#) (and [French version](#))

Everything you need in one organized window. Save your work as a script, RStudio includes many useful features which let us code easily. notebook, show results

Getting Started in R

- Create a new file¹: *File -> New File -> R Script*
- Basic commands²:

¹Shortcut to create new file: press *Shift + Ctrl + N* (Windows) or *Shift + Cmd + N* (Mac)

²Shortcut to run code: highlight it and press *Ctrl + Enter* (Windows) or *Cmd + Enter* (Mac)

```

1 # Math with R:
2 2 + 2      # Yes, R can do math!
3 sqrt(16)   # Square root.
4
5 # Create a variable:
6 x <- 5     # Now x is 5. You can reuse it!
7
8 # Inspect your environment:
9 ## Look at the "Environment" pane, do you see your x?
10 ls()
11
12 # Load a package:
13 library(stats)
14
15 # Get help of a function (documentation):
16 ?mean
17
18 # Quit RStudio properly
19 q()

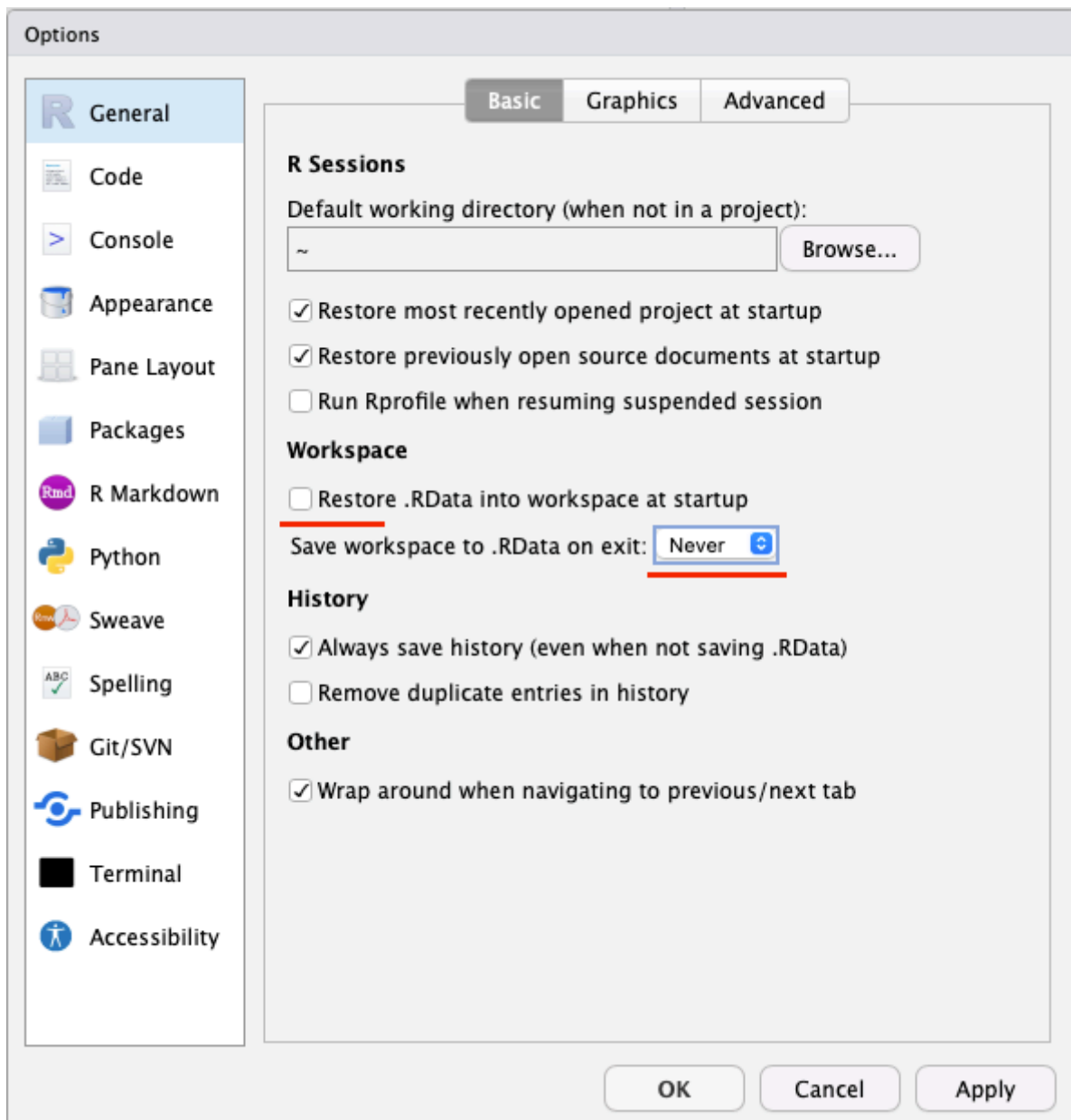
```

Some Configuration

Go to *Tools -> Global Options...*

Under the *General* tab:

- Uncheck the box “Restore .RData...”
- Set “Save workspace to .RData on exit” to **Never**.

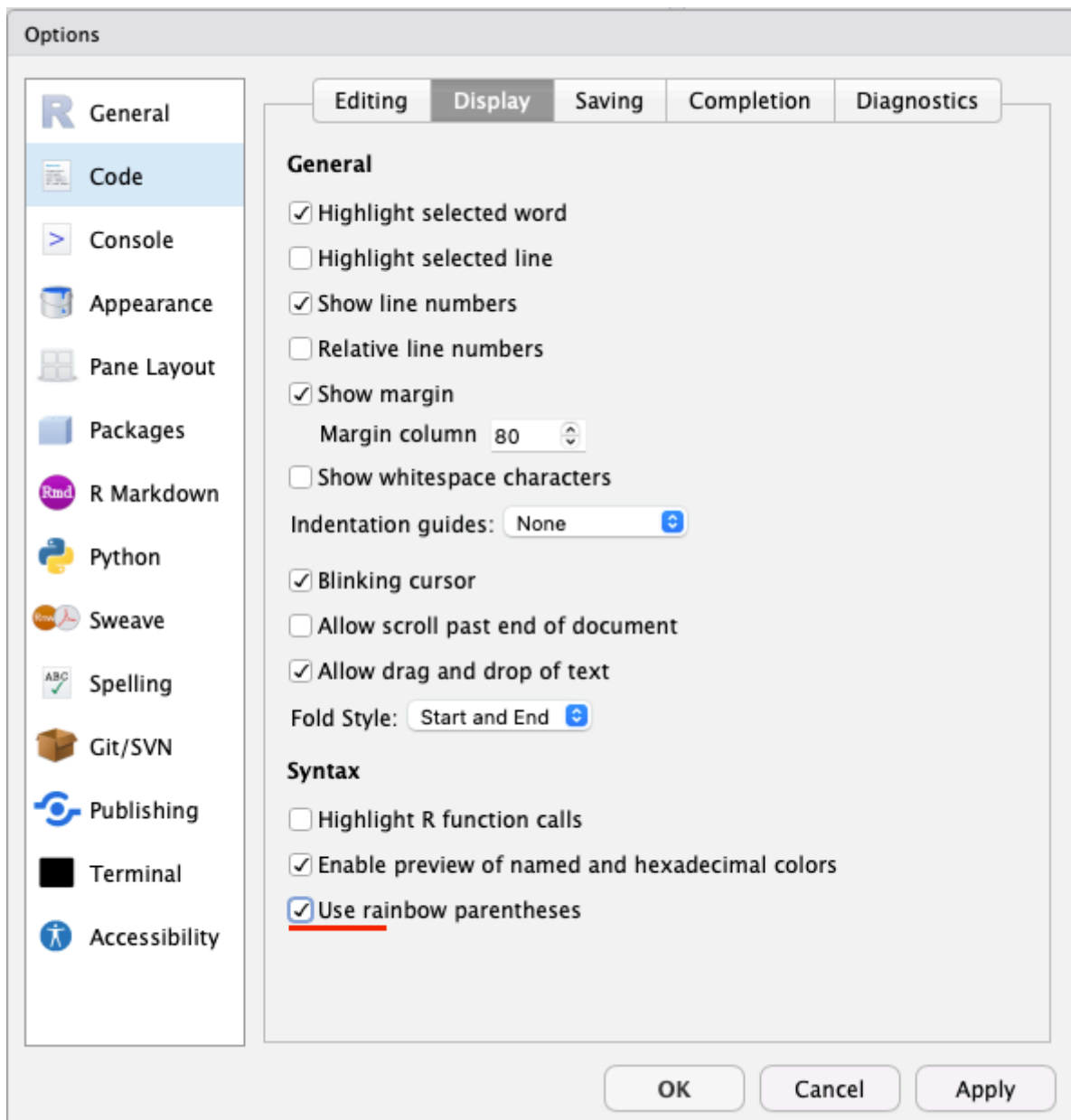


Some Configuration

Go to *Tools -> Global Options...*

Under the *Code* tab:

In the *Display* panel, check the box “Use rainbow parentheses”



Some Configuration

Go to *Tools -> Global Options...*

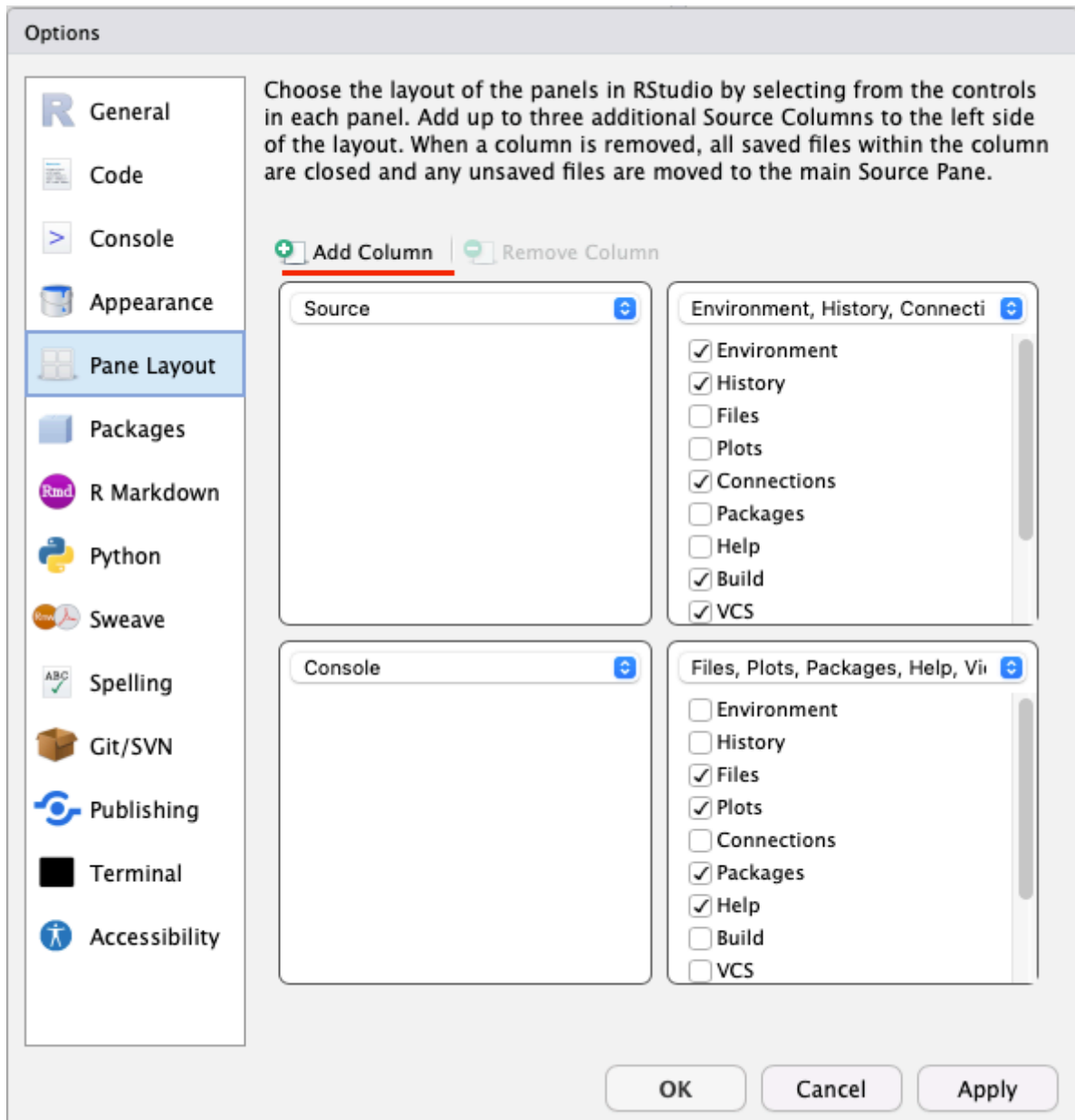
- Under the *Appearance* tab:

You can select a theme for you RStudio, e.g., the “Cobalt” for a dark theme.

- Under the *Pane Layout* tab:

You can arrange the four panels as you prefer.

In addition, click “Add Column” lets you open an extra “Source” code column, allowing you to view your scripts side by side!



Setting Up an R Project

Create an R Project

An R project keeps everything for your project in **one working directory** and helps R to know where to look for your files.

. . .

1. In RStudio, go to *File -> New Project*
2. Choose *New Directory* and click *New Project*
3. Set a folder name (e.g., **my_project**) and select where to save it.
4. Click *Create Project*

. . .

You'll notice RStudio restarts, and now your project is set up!



An R Project is a way to keep everything for your work (scripts, data, outputs) in one place. Think of it like a special folder where RStudio automatically knows where to look for your files. It helps avoid problems with messy file paths.

Understanding File Paths

A **path** tells your computer where to find files.

. . .

- **Absolute path**, starts from the very top of your computer's folder
 - E.g.:
 - * In real-life: France, Paris, 9 Quai St Bernard, Building B, office 725
 - * On Linux/Mac: `/home/test_user/my_project/data/myfile.csv`
 - * On Windows: `C:\Users\test_user\my_project\data\myfile.csv`

. . .

- **Relative path**, starts from your working directory (folder)

- E.g.:
 - * Building B, office 725
 - * data/myfile.csv or data\myfile.csv (Windows)

computer's file system -> a country files -> offices in a building

A Well Organized Folder

Tree structure:

```
my_project          # Your working directory
  my_project.Rproj  # The R project configuration file
  README.md         # A short introduction of the project
  data              # Raw data files
    raw_count.tsv
  scripts           # Your R code files
    01-analysis.R
    02-figures.R
  outputs           # Results, e.g., graphs, reports
```

example of bad home organisation: pillow in kitchen and microoven in the bathroom The trunk is the main folder (your R Project). Branches are subfolders for scripts, data, and output, ect. By building a clear structure, you'll: Always know where to find things. Save time when coding in R or sharing your project with others. Avoid messy files scattered everywhere.

Where Are You?

```
my_project          # Your working directory
  my_project.Rproj  # The R project configuration file
  README.md         # A short introduction of the project
  data              # Raw data files
    raw_count.tsv
  scripts           # Your R code files
    01-analysis.R
    02-figures.R
  outputs           # Results, e.g., graphs, reports
```

```
getwd() # Get working directory
[1] "/home/test_user/my_project"
```

...

What are the absolute and relative paths of the script “02-figures.R”?

...

absolute path: /home/test_user/my_project/scripts/02-figures.R

relative path: scripts/02-figures.R

...

```
file.exists("/home/test_user/my_project/scripts/02-figures.R")  
[1] TRUE  
file.exists("scripts/02-figures.R")  
[1] TRUE
```

Import Dataset

- Supported formats:
 - text files (.txt, .csv, *etc.*)
 - Excel files
 - data from other software (e.g., SAS)
- Click-button methods:
 - From menu bar: *File* -> *Import Dataset* -> Select the format -> Choose your file
 - From the “Environment” pane: click “Import Dataset” -> Select the format -> Choose your file
 - From the “Files” pane: Navigate to your file location -> Click the file and select “Import Dataset”

Import Text Data

File/URL:

~/Documents/my_project/data/shiny_sample_data.csv

Update

Data Preview:

...1 (character) ▾	sample1 (double) ▾	sample2 (double) ▾	sample3 (double) ▾	sample4 (double) ▾	sample5 (double) ▾	san
gene_1	89	4664	274	126	1267	
gene_2	568	3456	201	9	139	
gene_3	275	4307	449	187	7909	
gene_4	3913	5912	4082	4318	8892	
gene_5	3382	155	2897	2626	2069	
gene_6	4167	5194	4404	4757	4765	
gene_7	4831	227	5009	4554	479	
gene_8	2532	3395	3000	3441	4575	
gene_9	3274	1658	3074	3189	2293	
gene_10	3272	4169	3454	3192	7403	
gene_11	465	3645	125	234	246	
qene_12	3432	3704	3455	3770	3196	

Previewing first 50 entries.

Import Options:

Name:

shiny_sample_data

Skip:

0

☒ First Row
as Names
☒ Trim
Spaces
☒ Open
Data Viewer

Delimiter:

Comma ▾

Quotes:

Default ▾

Locale:

Configure...

Escape:

None ▾

Comment:

Default ▾

NA:

Default ▾

Code Preview:

```
library(readr)
shiny_sample_data <- read_csv("data/shiny_sample_data.csv")
View(shiny_sample_data)
```

? Reading rectangular data using readr

Import

Cancel

Automated Reporting

Quarto

- Open-source tools for creating dynamic and shareable document.
- Combines narrative text and code (multi-language) in one script (.qmd).
- Supports multiple output formats: .html, .docx, .pptx, .pdf, *etc.*

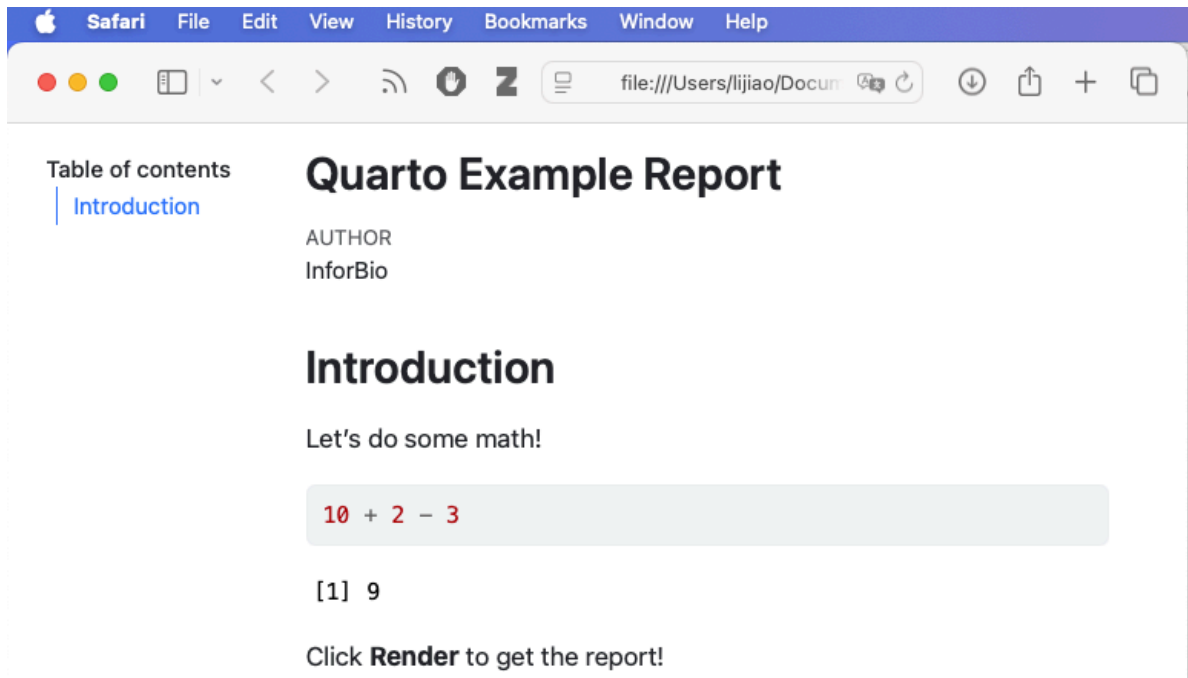
An Example

The screenshot shows the Quarto editor interface with a file named `quarto_example.qmd`. The editor has a toolbar with icons for navigation, saving, and rendering. Below the toolbar are tabs for 'Source' and 'Visual', and an 'Outline' button. The 'Source' tab is active, showing the following content:

```
1 ---
2 title: "Quarto Example Report"
3 author: InforBio
4 format:
5   html:
6     toc: true
7     toc-location: left
8 ---
9
10 # Introduction
11
12 Let's do some math!
13
14 ```{r}
15 #| echo: true
16
17 10 + 2 - 3
18 ```
19
20 Click Render to get the report!
```

Annotations in the image highlight different parts of the code:

- An orange box around lines 2-7 is labeled "YAML header".
- A blue box around lines 10-12 is labeled "Markdown text".
- A green box around lines 14-18 is labeled "Code chunk".



Markdown syntax [cheat sheet](#)

Need Some Help?

- Google it!
- Ask/discuss your question via Slack.
- AI (like ChatGPT) is your friend , but be careful of false answers! (More details in session 6)

Key points:

- Explain your question step-by-step.
- Share the context.
- Use keywords.
- Try to rephrase the question.

google/forum stack overFlow - what are you trying to do / what did you try? - the warning or error message / your code / your data structure - language name / function or package name / a specific problem - try and iterate

Let's Practice !

Today's Goals

- Get familiar with the RStudio
- Create an R project in RStudio
- Upload and import data file into RStudio
- Generate your first Quarto report