Data cleaning and exploration 1

BIOL90041

Week 3

Dr. Kristoffer Wild

kristofferw@unimelb.edu.au

Todays Learning objectives

- Finding your data…
  - How do we do it?
  - Then how do we save it?

- Viewing and inspecting your data
  - functions to look at data once imported
  - Protocol for data exploration from from Zuur et al., 2010

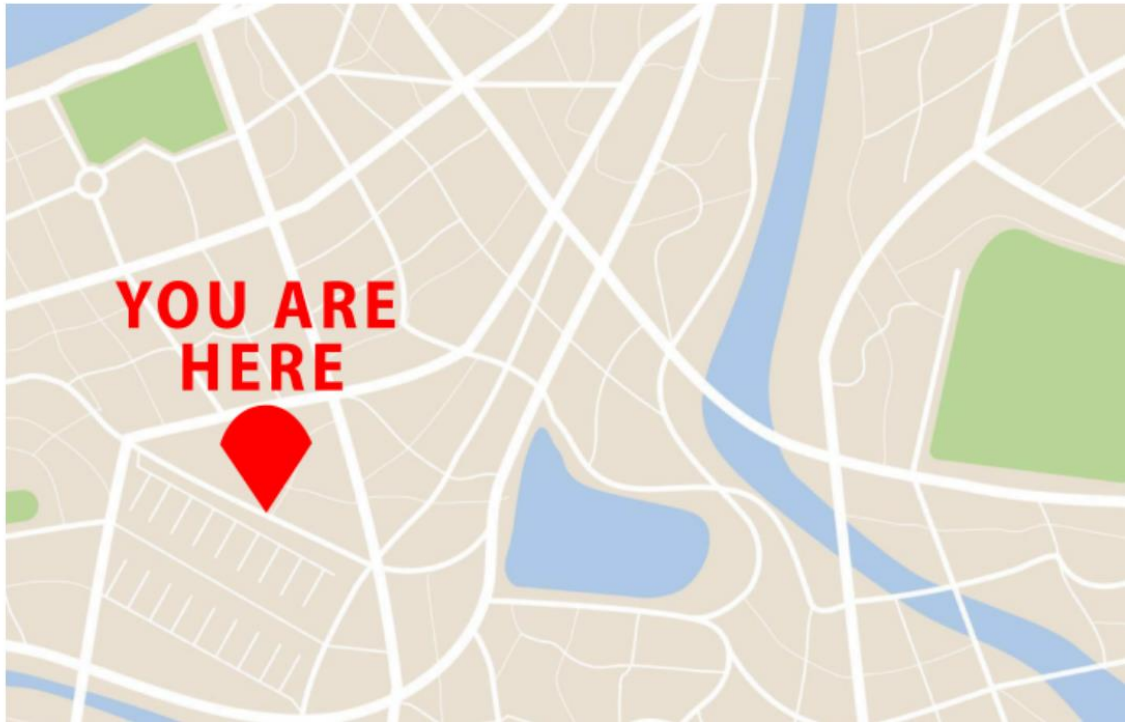
- What is quarto and it’s functionality

So you want to be a data scientist…

How will you spend your time?

Working Directory

- Most of the time, you won't generate data in R — you'll import it from external files
- Before importing data, you need to understand where R is "looking" from
- This location is called your **working directory**
- The working directory is your R console's current vantage point on your files



Working Directory

- You can check your working directory using the *getwd()* function

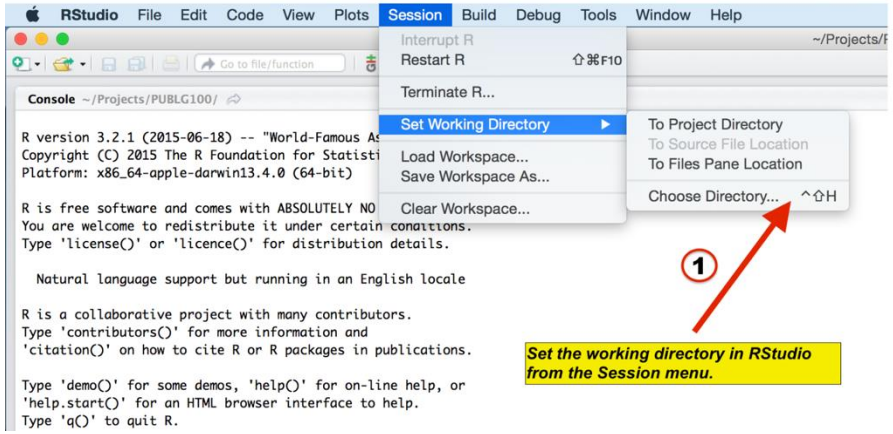
- In a R project it will be the root folder!

R Console

```
> getwd()
[1] "C:/Windows/system32"
```

Setting the working directory

- Your current working directory is usually your **HOME** directory
- For projects, it's best to keep everything in **one folder**:
  - input files
  - output files
  - scripts
- Choose a working directory that's easy to access
- Let's do this on the desktop:
  - Create a new folder called **lecture_3**
  - Open a new R script
  - Save the working directory to this folder



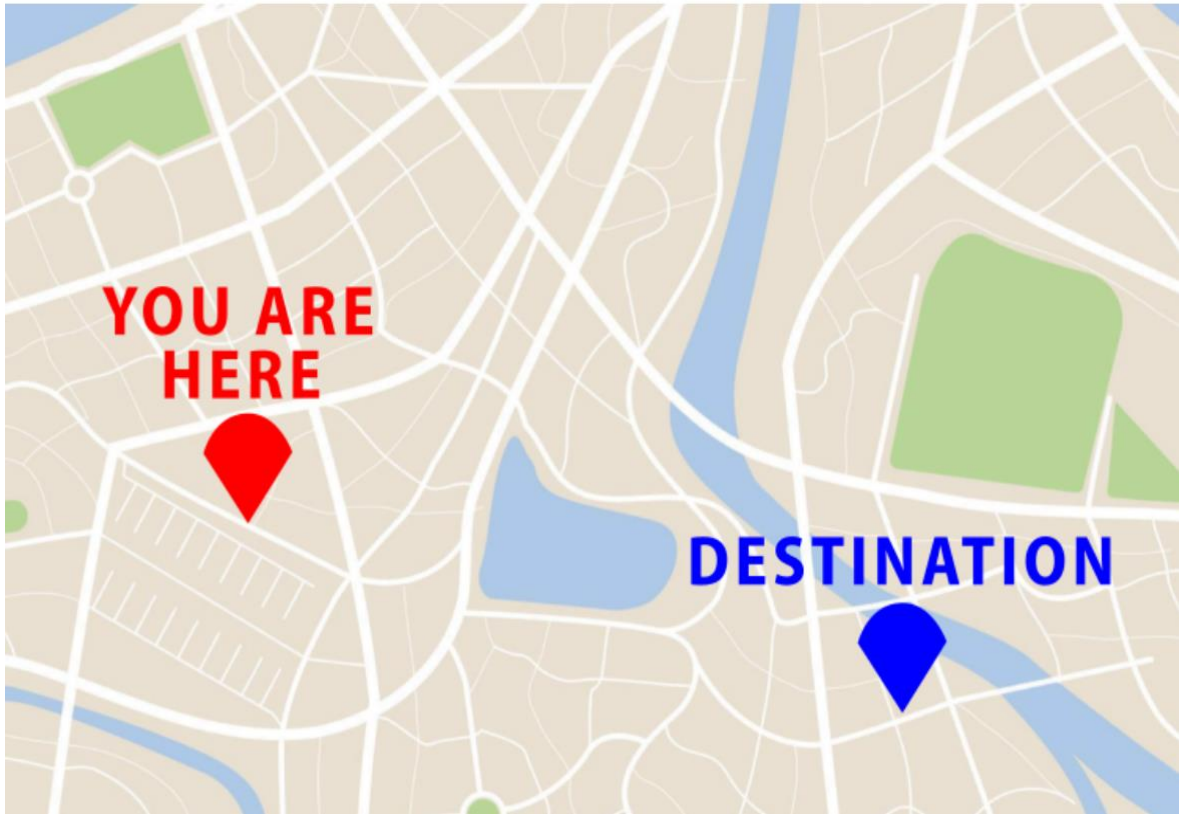
Session -> Set Working Directory -> Choose Directory

OR IN THE CONSOL

```
setwd("~/Desktop/lecture_3")
```

Paths

- When you give a path to R it can either be relative or absolute



Paths

To use our map analogy:

- Relative path are like directions i.e. take a left, go straight then take a right etc. The context of where you START is essential.

- Absolute paths are like an address. They give you the final location in absence to any other external information

Both have their benefits

Paths in use

The command we used before was using an short hand '~' of absolute path. The '~' points to /Users/yourname

```
setwd("~/Desktop/lecture_3")
```

Absolute paths usually start with a '/' to get to the top level of your computers file structure

```
setwd("/Users/yourname/Desktop/lecture_3")
```

Or use a **relative path** from your current working directory (e.g., if you're already in /Users/yourname):

```
setwd("Desktop/lecture_3")
```

Data from external sources

Now we have set our location we can start thinking about reading in data.
A standard format for data is '.csv' for tables like this:

Gene_Name	Sample_1.hi	Sample_2.hi	Sample_3.hi
Gene_a	4.809882	4.462476	3.539506
Gene_b	4.665849	5.597727	3.545790
Gene_c	3.675578	3.444438	3.468934
Gene_d	5.132539	4.839964	3.800771
Gene_e	8.882293	9.913441	9.441016
Gene_f	9.308390	10.614094	11.440200
Gene_g	8.203687	11.334060	10.600226
Gene_h	11.012924	10.731669	8.011221

Go to week 3 lecture material and download the zipped folder 'Path_example' and put it on your desktop

Data from text file with read.csv()

Tables from text files can also be read using the read.csv() function. This is a wrapper for read.table() with sep = "," already set.

```
Table <- read.csv("Path_example/Raw data/gene_data.csv", header = TRUE)
Table[1:4, 1:3]
```

- read.csv() automatically uses sep = ",", ideal for comma-separated files
- header = TRUE tells R the first row contains column names

Console Output

```
  Gene_Name Sample_1.hi Sample_2.hi
1    Gene_a    4.570237    3.230467
2    Gene_b    3.561733    3.632285
3    Gene_c    3.797274    2.874462
4    Gene_d    3.398242    4.415202
```

Row names in read.csv()

read.csv() accepts the row.names argument to use a column as row names.

```
Table <- read.csv("Path_example/Raw data/gene_data.csv", header = TRUE, row.names = 1)
Table[1:4, 1:3]
```

- row.names = 1 uses the first column (Gene_Name) as the row names

Setting factors from read.csv()

Like read.table(), read.csv() lets you control whether character data should be read in as factors using stringsAsFactors.

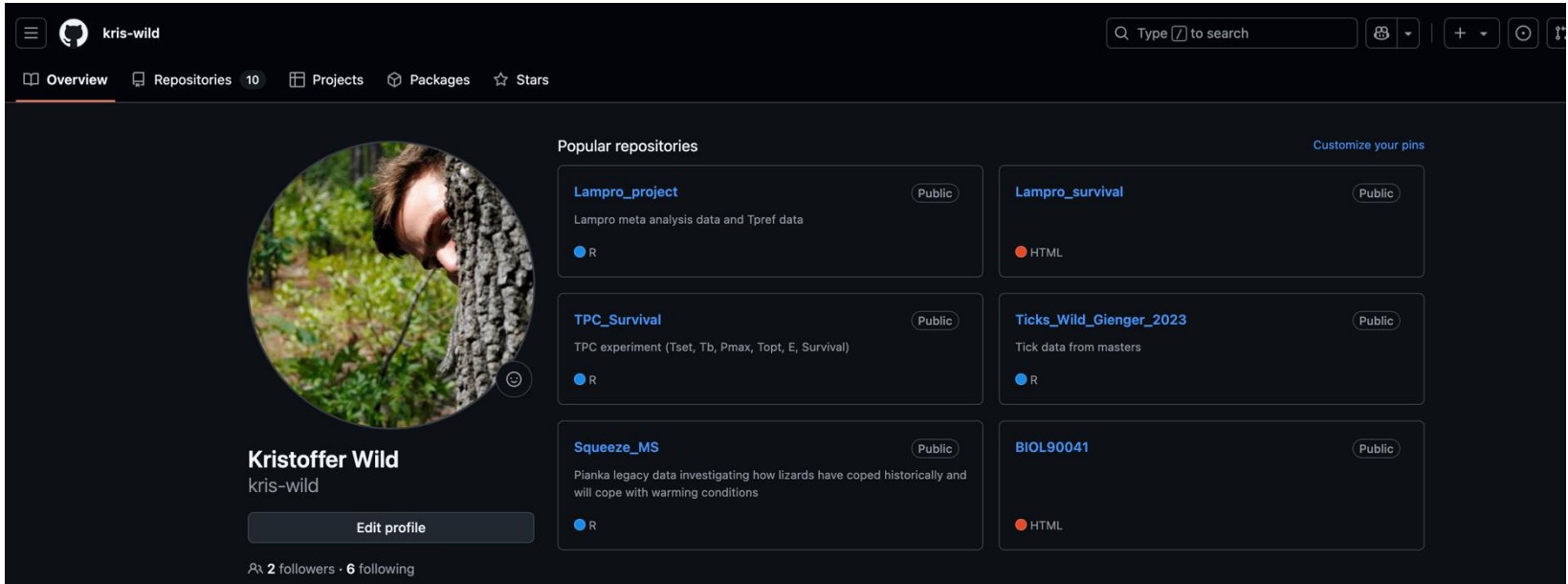
```
Table <- read.csv("Path_example/Raw data/gene_data.csv", header = TRUE, stringsAsFactors = FALSE)
Table[1:4, 1:3]
```

- stringsAsFactors = FALSE is the default in R 4.0 and above
- Prevents automatic conversion of text columns to factor variables
- Other useful arguments:
  - skip — skip a number of lines when reading
  - comment.char — ignore lines starting with a given character

Reading data from outher sources

You can load data directly from GitHub using read.csv() — just make sure you link to the **raw** version of the file:

```
URL <- "https://raw.githubusercontent.com/kris-wild/TPC_Survival/main/R/Final.Analysis/Final.Figure.data/Field_Tb_vs_Tb_predict_SI.csv"
Table <- read.csv(URL, header = TRUE)
head(Table)
```



Class assignment (15 min)

JUST using using the ‘Path_example’ I want you to tell me…

- how many .csv files are in there?
- for each csv file tell me how many columns are character and numeric for each file

Now what if I want to back out of a file or go to a folder ‘behind’ my current directory

- Demo quick demo with Kris

Reviewing your data

It is always important to know what your data is. Especially when you are reading it in for the first time. There are three functions to quickly use your data.

head()

```
> head(Table)
  X    POVI       Date HR Tb_predict Tb_obs
1 1 POVI_16 2019-01-23 10       38.3   36.0
2 2 POVI_16 2019-01-23 11       38.3   36.0
3 3 POVI_16 2019-01-23 12       37.8   36.0
4 4 POVI_16 2019-01-23 13       38.0   35.5
5 5 POVI_16 2019-01-23 14       37.8   35.5
6 6 POVI_16 2019-01-23 15       37.9   35.5
```

Reviewing your data

It is always important to know what your data is. Especially when you are reading it in for the first time. There are three functions to quickly use your data.

head()

tail()

```
> tail(Table)
        X    POVI       Date HR Tb_predict Tb_obs
7144 7144 POVI_45 2019-03-04 14       40.6   36.5
7145 7145 POVI_45 2019-03-04 15       40.6   37.0
7146 7146 POVI_45 2019-03-04 16       40.3   37.0
7147 7147 POVI_45 2019-03-04 17       39.7   37.0
7148 7148 POVI_45 2019-03-04 18       38.8     NA
7149 7149 POVI_45 2019-03-04 19       37.2     NA
```

Reviewing your data

It is always important to know what your data is. Especially when you are reading it in for the first time. There are three functions to quickly use your data.

head()

tail()

str()

```
> str(Table)
'data.frame':   7149 obs. of  6 variables:
 $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ POVI      : chr  "POVI_16" "POVI_16" "POVI_16" "POVI_16" ...
 $ Date      : chr  "2019-01-23" "2019-01-23" "2019-01-23" "2019-01-23" ...
 $ HR        : int  10 11 12 13 14 15 16 17 18 19 ...
 $ Tb_predict: num  38.3 38.3 37.8 38 37.8 37.9 38.7 36.4 35.5 35.1 ...
 $ Tb_obs    : num  36 36 36 35.5 35.5 35.5 35.5 38 35 34.5 ...
```

Saving Data with write.csv()

Save your data in a standard format (CSV) using write.csv()

```
Table <- read.csv('Path_example/Final data/not final yet/lizard.csv', header = TRUE)
write.csv(Table, file = ”data/my_data.csv", row.names = FALSE)
```

- Useful for:
  - Sharing data with others
  - Opening in Excel or other programs
  - Backing up cleaned datasets
- Keep in mind:
  - Data types like dates and factors may lose formatting when reloaded
  - All columns become character or numeric upon re-import

Saving R objects with saveRDS()

Save an entire R object exactly as-is using saveRDS()

```
saveRDS(Table, file = "output/my_data.rds")
```

To reload it later:

```
Table <- readRDS("output/my_data.rds")
```

- RDS files are great because:
  - They **preserve object structure** (e.g., factors, dates, models)
  - Compact and efficient for saving any R object
  - Perfect for saving data between script runs or analyses

Note: RDS files can **only be read back into R**, not into Excel or other programs

Now you have your data!

“I spend more than half of my time integrating, cleansing and transforming data without doing any actual analysis. Most of the time I’m lucky if I get to do any ‘analysis’ at all…

… Most of the time once you transform the data ... the insights can be scarily obvious.”

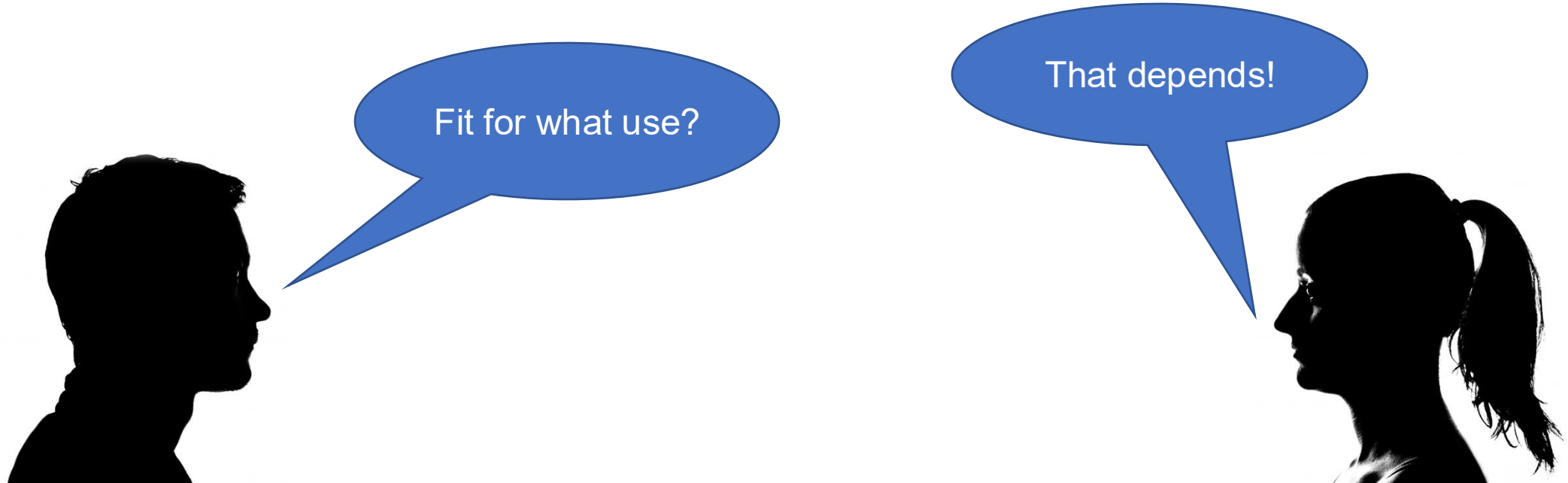
“Once you play with the data you realize you made an assumption that is completely wrong. It’s really useful, it’s not just a waste of time, even though you may be banging your head.”

“In practice it tends not to be just data prep, you are learning about the data at the same time, you are learning about what assumptions you can make.”

Data Wrangling

Aka Data Prep, Data Munging, Data Transformation

*Assessing and transforming raw data to make it **fit for use***



Data Wrangling

Aka Data Prep, Data Munging, Data Transformation

*Assessing and transforming raw data to make it **fit for use***

This is how you “get your head in the game”

- Understand what you have
- Assess strengths and weaknesses of your data
- Hypothesise about what to do with your data
- Get it ready

Nobody will know your data as well as you do while wrangling

- Not even the “you” of a few days later

Discussion

When is data “dirty”?

How does that happen?

BRITISH ECOLOGICAL SOCIETY
Methods in Ecology and Evolution

Free Access

A protocol for data exploration to avoid common statistical problems

Alain F. Zuur, Elena N. Ieno, Chris S. Elphick

First published: 23 February 2010 | https://doi.org/10.1111/j.2041-210X.2009.00001.x | Citations: 6,352

Correspondence site: http://www.respond2articles.com/MEE/

https://besjournals.onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2009.00001.x

Step 1 – Are There Outliers in Y or X?

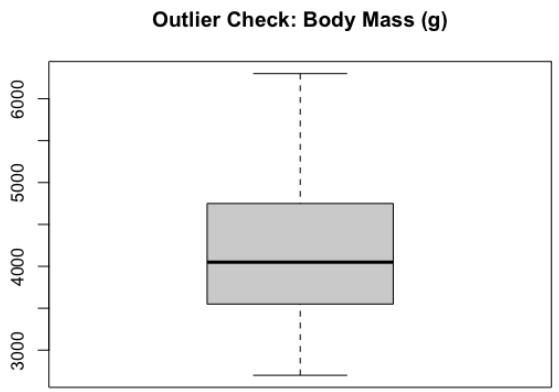
- Outliers are extreme values that differ from the rest of the data
- They may arise from data entry errors, measurement errors, or real biological extremes
- Outliers can skew visualisations, violate model assumptions, and bias results
- *"Outliers can dominate model outcomes or visualizations. Identifying them is essential before analysis."* - **Zuur et al. (2010)**

Step 1 – Are There Outliers in Y or X?

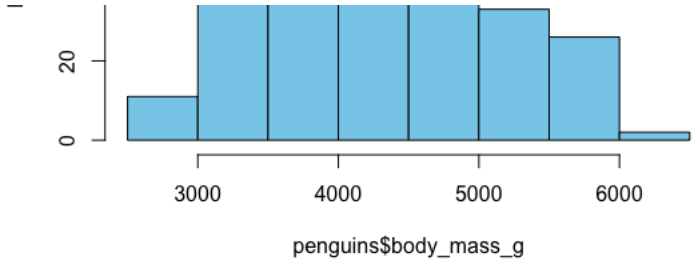
How would we look for outliers?

```
pacman::p_load(palmerpenguins, dplyr)
# Boxplot to detect outliers
boxplot(penguins$body_mass_g,
        main = "Outlier Check: Body Mass (g)",
        col = "lightgrey")

# Histogram to see distribution
hist(penguins$body_mass_g,
     main = "Histogram of Body Mass (g)",
     col = "skyblue")
```



Tip: Combine with summary() or sort() to see extremes numerically.



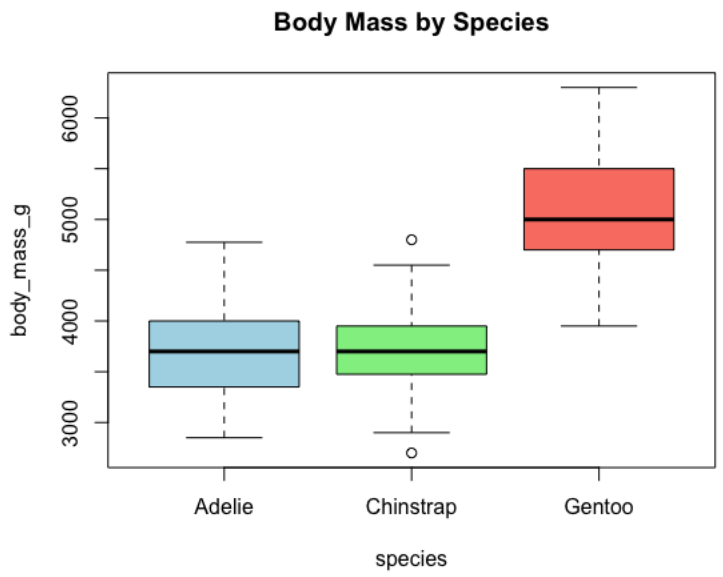
Step 2 – Is the Variance Similar Across Groups?

- Homogeneity means groups have a similar spread (variance).
- Unequal variance can affect models like ANOVA or regression.
- Can be visualized by comparing group distributions.
- *"Even moderate variance differences can degrade model reliability."*
  — **Zuur et al. (2010)**

Step 2 in R – Check Variance (penguins)

How would we compare variance?

```
# Compare body mass variance across species
boxplot(body_mass_g ~ species, data = penguins,
        main = "Body Mass by Species",
        col = c("lightblue", "lightgreen", "salmon"))
```




```
# Numerical check of variances
penguins %>%
  group_by(species) %>%
  summarise(var_mass = var(body_mass_g, na.rm = TRUE))
```

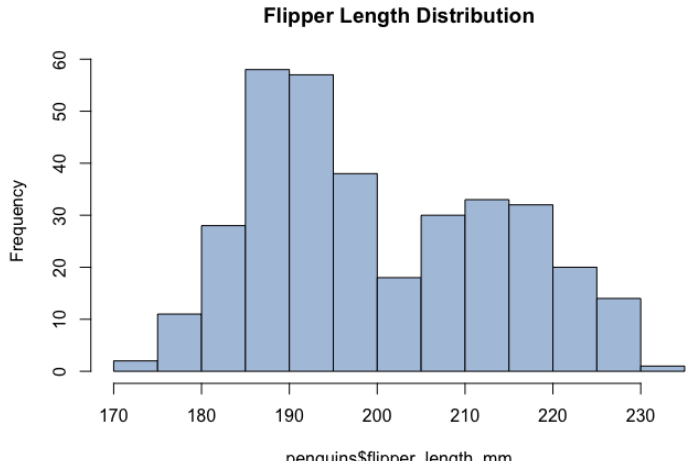
```
# A tibble: 3 × 2
  species   var_mass
  <fct>        <dbl>
1 Adelie     210283.
2 Chinstrap  147713.
3 Gentoo     254133.
```

Step 3 – Are the Data Normally Distributed?

- Normality refers to the bell-shaped curve in data distribution.
- Histograms and QQ plots are used to inspect this.
- Not all methods require normality, but it's important to check.
- 🗣 *"Many methods are robust, but checking normality helps you choose the right approach."*
  — **Zuur et al. (2010)**

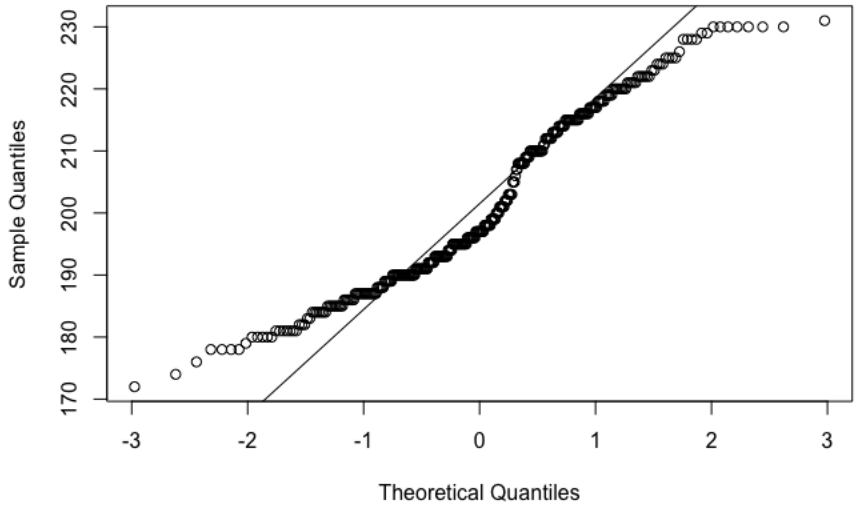
Step 3 – Check Normality (penguins)

```
# Histogram of flipper length
hist(penguins$flipper_length_mm,
    main = "Flipper Length Distribution",
    col = "lightsteelblue")
```



Tip: Look for symmetry and smooth tails in histograms

```
# QQ plot (optional)
qqnorm(penguins$flipper_length_mm)
qqline(penguins$flipper_length_mm)
```



Step 4 – Are There Lots of Zeros?

- Excess zeros can distort models for count or binary data.
- May require special modeling approaches.
- Check both counts and joint absences (multivariate).
- 🗣 *"Zero inflation requires special models or interpretation techniques."*
  — **Zuur et al. (2010)**

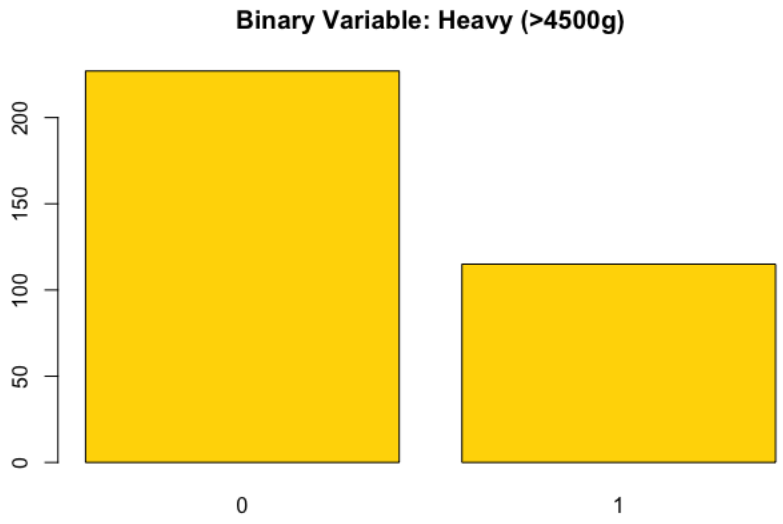
Step 4 –Check Zeros (penguins)

```
# Create a binary variable
penguins$heavy <- ifelse(penguins$body_mass_g > 4500, 1, 0)
# now check frequency of 0s and 1s
table(penguins$heavy)
```

```
> table(penguins$heavy)

  0   1
227 115
```

```
# Visualise
barplot(table(penguins$heavy),
        main = "Binary Variable: Heavy (>4500g)",
        col = "gold")
```

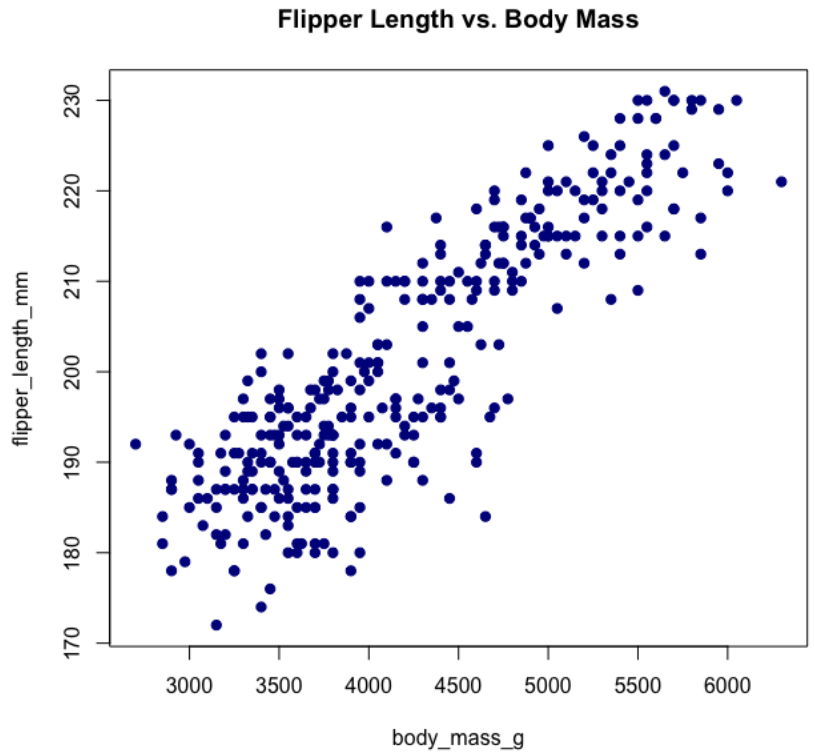


Step 5 – What Is the Relationship Between Y & X?

- Helps identify if a linear model is appropriate, or if relationships are nonlinear.
- Can reveal thresholds, curvatures, or outliers.
- Always plot your response variable vs. each predictor.
- *"Scatterplots can reveal patterns that inform model design."*— **Zuur et al. (2010)**

Step 5 in R – Plot Y vs. X (penguins)

```
# Base R scatterplot
plot(flipper_length_mm ~ body_mass_g,
    data = penguins,
    main = "Flipper Length vs. Body Mass",
    col = "darkblue", pch = 19)
```



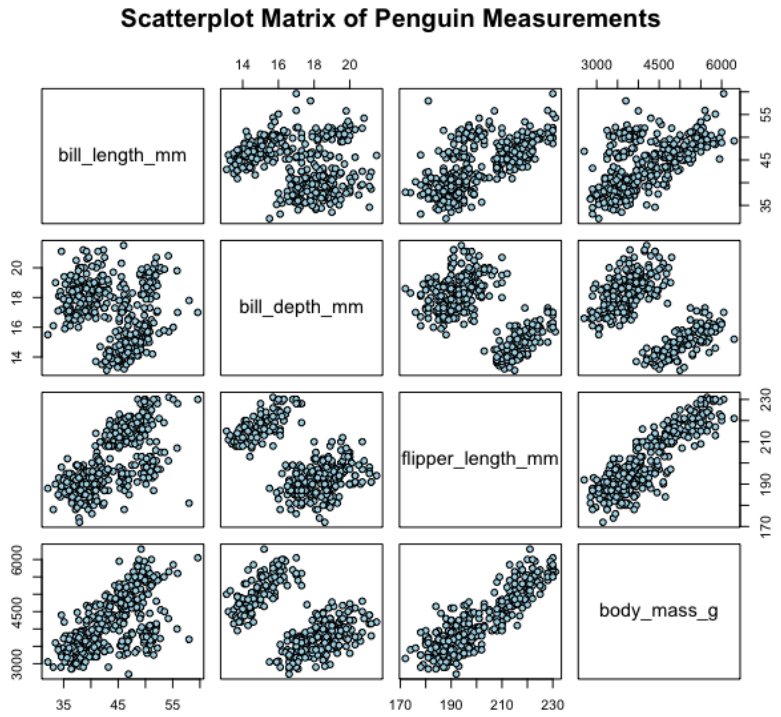
Step 6 – Are the Predictors Too Similar?

- Collinearity occurs when two or more predictors are strongly correlated.
- This can inflate standard errors and mask variable importance
- Use correlation matrices or visual tools to detect it.
- *"Collinearity reduces model interpretability and power."* — **Zuur et al. (2010)**

Step 6 in R – Check Collinearity (penguins)

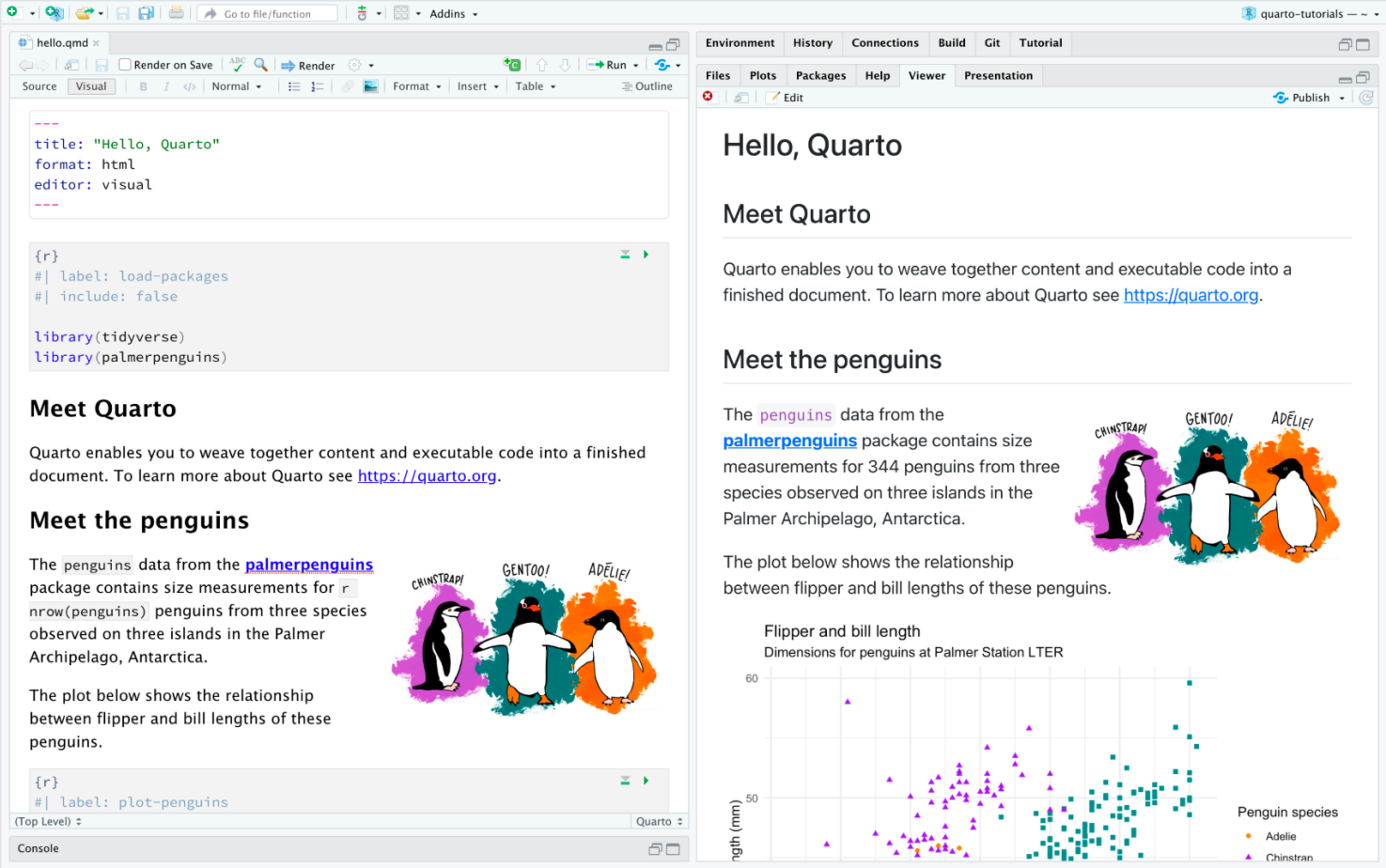
```
# Prepare numerical data
num_vars <- penguins %>%
  select(bill_length_mm, bill_depth_mm,
         flipper_length_mm, body_mass_g) %>%
  filter(if_all(everything(), ~ !is.na(.)))
# Now use correlation matrix to look at 4 variables
cor(num_vars)

# Visualise correlations by a pairs plot
pairs(num_vars,
      main = "Scatterplot Matrix of Penguin Measurements",
      pch = 21,
      bg = "lightblue")
```

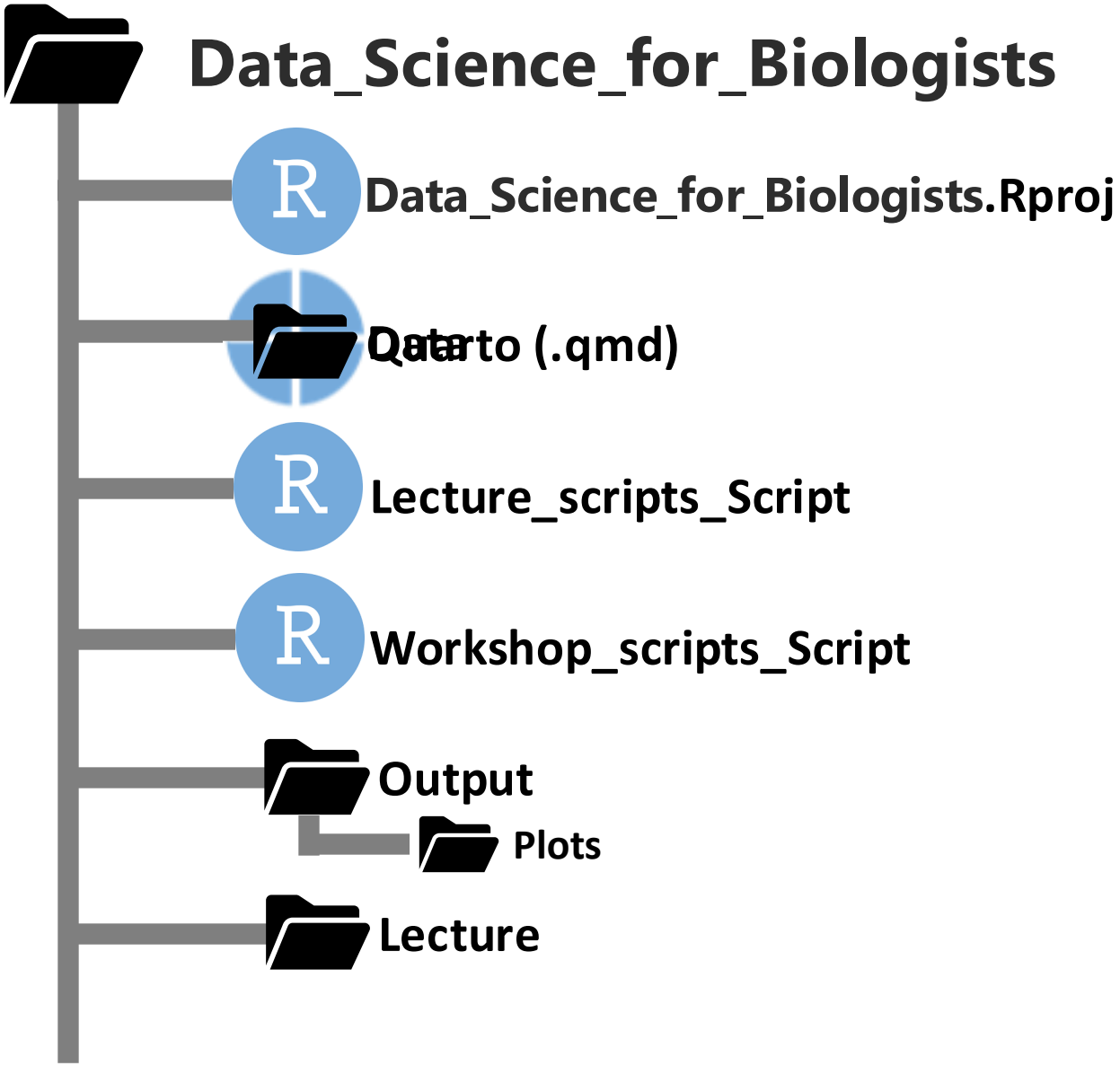


Get Started

quarto®

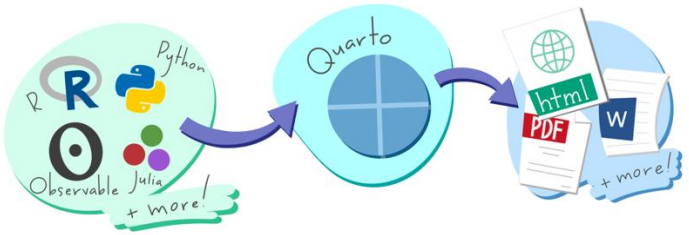


A basic R project set up



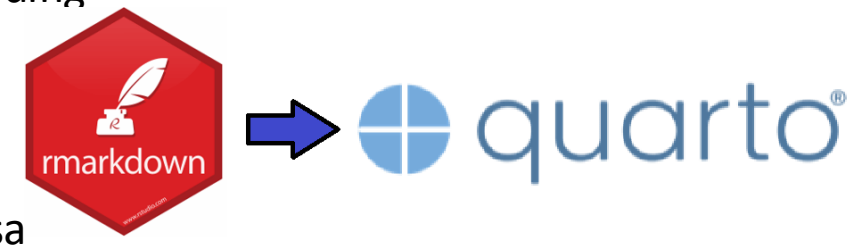
What is Quarto?

An open-source tool that seamlessly integrates code, results, and narrative text into a single document



Key Features:

- Multi-language Support: Works with R, Python, Julia, and more.
- Versatile Outputs: Generate reports, presentations, websites, and dashboards in formats like HTML, PDF, and Word.
- Reproducibility: Ensures that analyses are transparent and reproducible by embedding code and its output directly within the document

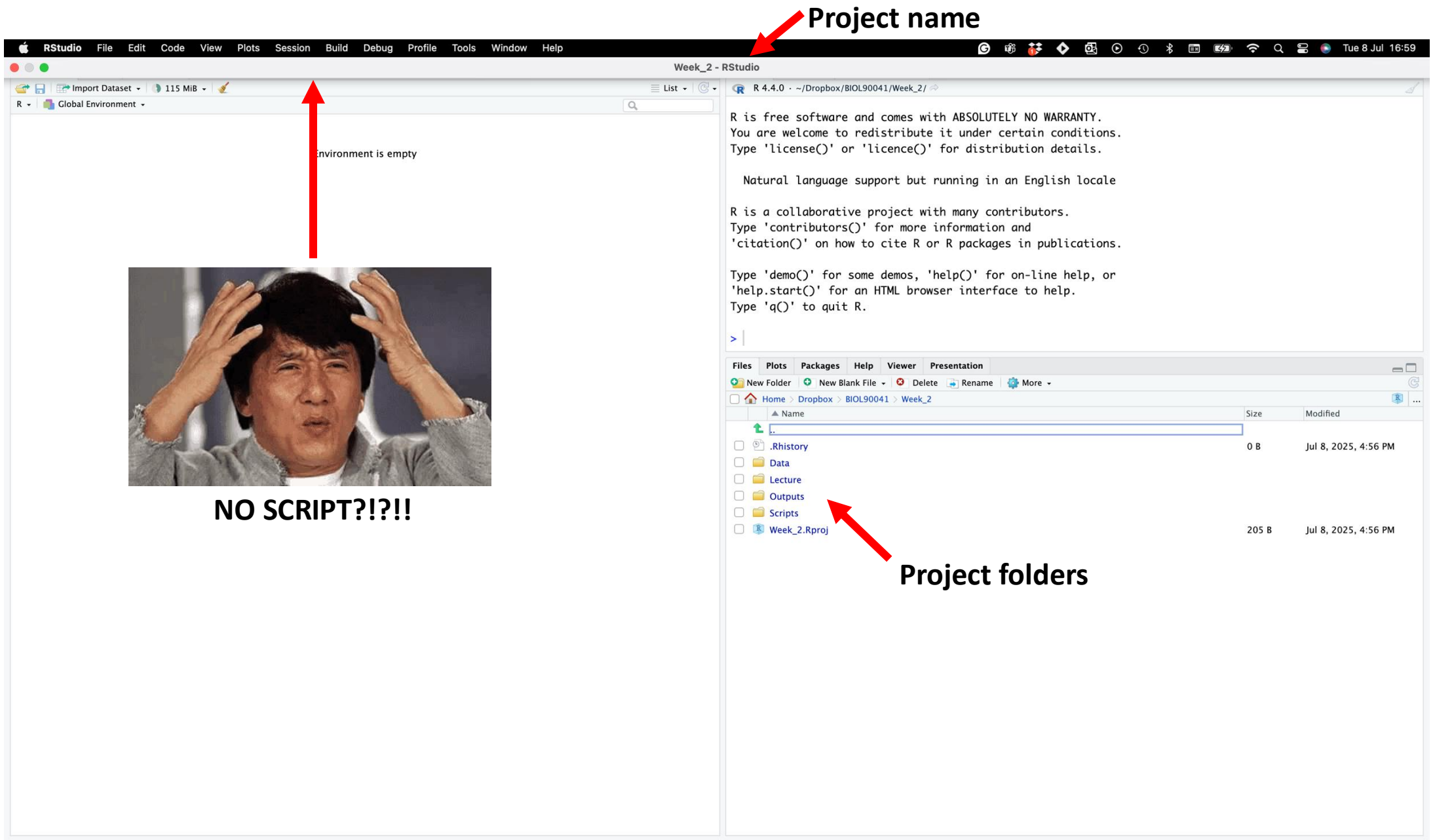


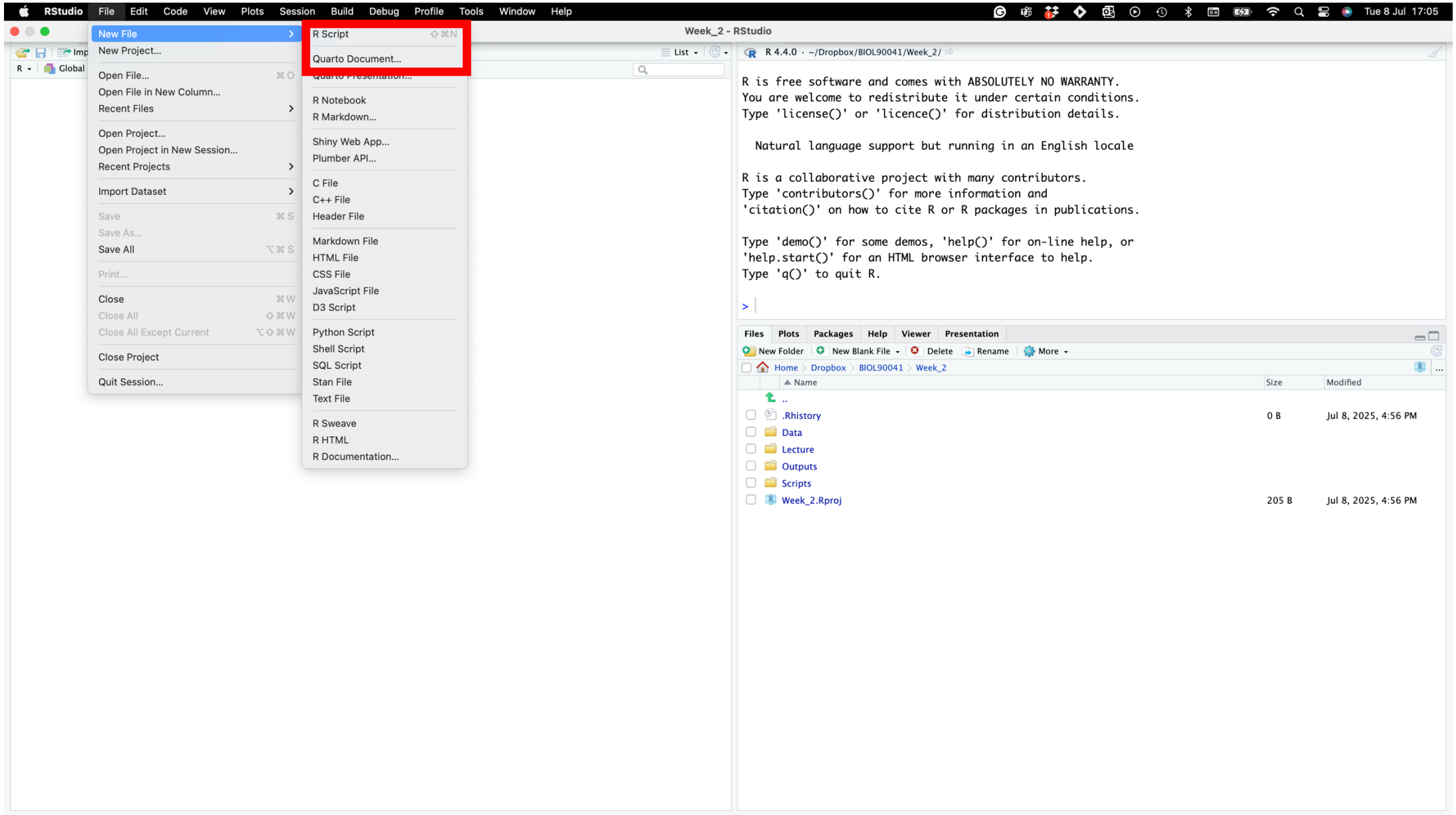
Use Cases:

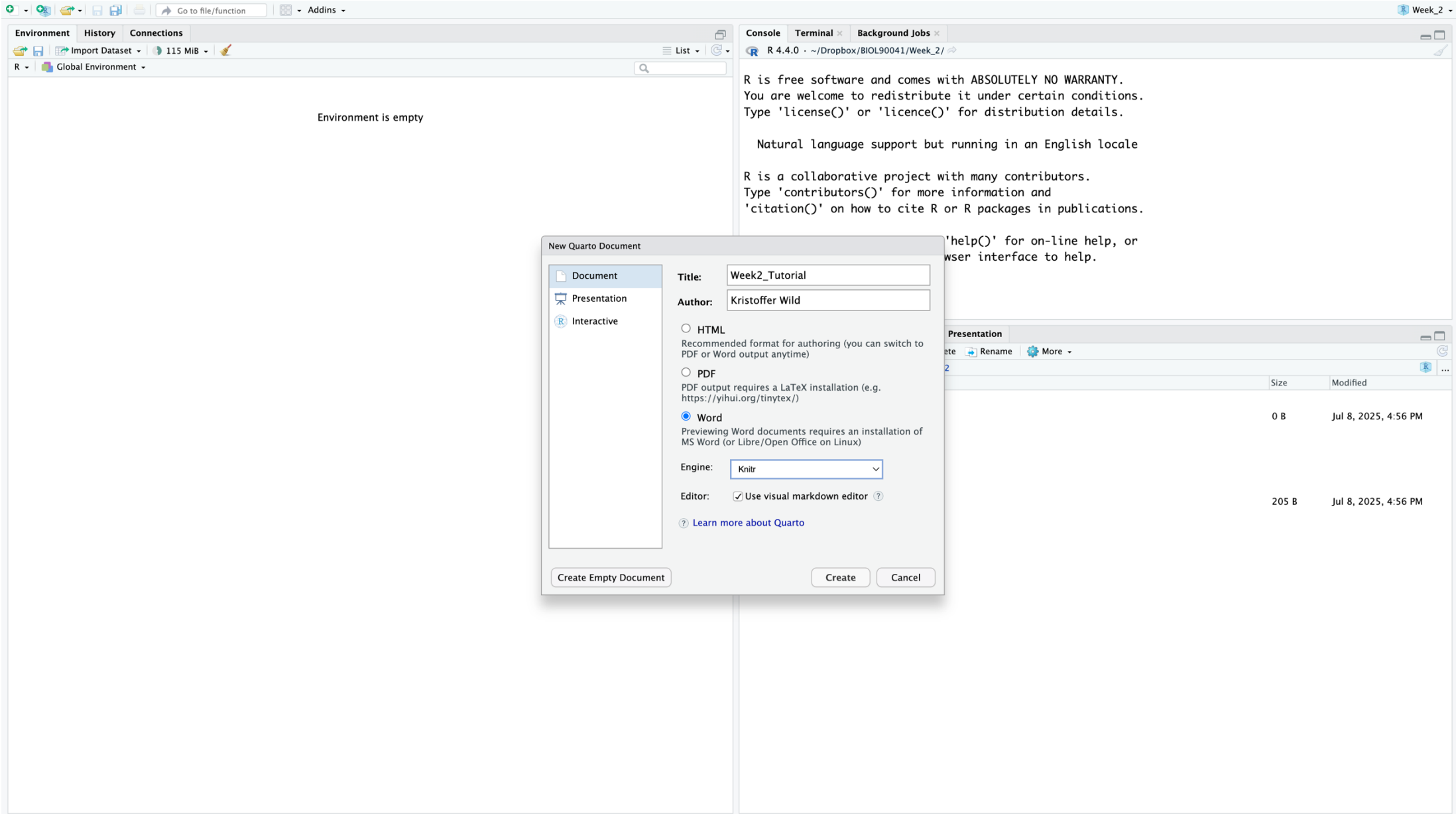
- Communication: Present findings to stakeholders with clear narratives and visualisa
- Collaboration: Share comprehensive analyses with peers, including code and results.
- Documentation: Maintain a detailed record of data analysis processes and decisions.

Integration:

- Fully compatible with RStudio and other IDEs, providing a user-friendly interface for creating and rendering documents.


Project name
RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Tue 8 Jul 16:59
Week_2 - RStudio
Import Dataset
115 MiB
List
R
Global Environment
Environment is empty
NO SCRIPT?!?!!
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/
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.
>
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Delete Rename More
Home > Dropbox > BIOL90041 > Week_2
Name Size Modified
..
.Rhistory 0 B Jul 8, 2025, 4:56 PM
Data
Lecture
Outputs
Scripts
Week_2.Rproj 205 B Jul 8, 2025, 4:56 PM
Project folders


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Tue 8 Jul 17:05
Week_2 - RStudio
New File
New Project...
Open File... ⌘O
Open File in New Column...
Recent Files
Open Project...
Open Project in New Session...
Recent Projects
Import Dataset
Save ⌘S
Save As...
Save All ⌥⌘S
Print...
Close ⌘W
Close All ⇧⌘W
Close All Except Current ⌥⇧⌘W
Close Project
Quit Session...
R Script ⇧⌘N
Quarto Document...
R Notebook
R Markdown...
Shiny Web App...
Plumber API...
C File
C++ File
Header File
Markdown File
HTML File
CSS File
JavaScript File
D3 Script
Python Script
Shell Script
SQL Script
Stan File
Text File
R Sweave
R HTML
R Documentation...
List
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/
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.
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Delete Rename More
Home Dropbox BIOL90041 Week_2
Name Size Modified
..
.Rhistory 0 B Jul 8, 2025, 4:56 PM
Data
Lecture
Outputs
Scripts
Week_2.Rproj 205 B Jul 8, 2025, 4:56 PM


Go to file/function
Addins
Week_2
Environment
History
Connections
Import Dataset
115 MiB
List
R
Global Environment
Environment is empty
Console
Terminal
Background Jobs
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/
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.
'help()' for on-line help, or
wser interface to help.
New Quarto Document
Document
Presentation
Interactive
Title:
Week2_Tutorial
Author:
Kristoffer Wild
HTML
Recommended format for authoring (you can switch to PDF or Word output anytime)
PDF
PDF output requires a LaTeX installation (e.g. https://yihui.org/tinytex/)
Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux)
Engine:
Knitr
Editor:
Use visual markdown editor
Learn more about Quarto
Create Empty Document
Create
Cancel
Presentation
Rename
More
Size
Modified
0 B
Jul 8, 2025, 4:56 PM
205 B
Jul 8, 2025, 4:56 PM

LINE #'s

Metadata

Week2_Tutorial.qmd*

```
1  ---
2  title: "Week2_Tutorial"
3  author: "Kristoffer Wild"
4  format: docx
5  editor: visual
6  ---
7
8  ## Quarto
9
10 Quarto enables you to weave together content and executable code into a finished
   document. To learn more about Quarto see <https://quarto.org>.
11
12 ## Running Code
13
14 When you click the **Render** button a document will be generated that includes both
   content and the output of embedded code. You can embed code like this:
15
16 ```{r}
17 1 + 1
18 ```
19
20 You can add options to executable code like this
21
22 ```{r}
23 #| echo: false
24 2 * 2
25 ```
26
```

Console

```
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/

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.

>
```

Home > Dropbox > BIOL90041 > Week_2

Name	Size	Modified
..		
.Rhistory	0 B	Jul 8, 2025, 4:56 PM
Data		
Lecture		
Outputs		
Scripts		
[illegible]	205 B	Jul 8, 2025, 4:56 PM
Week2_Tutorial.qmd	601 B	Jul 8, 2025, 5:09 PM

Environment is empty

Week2_Tutorial.qmd*

Source | Visual

```
---
title: "Week2_Tutorial"
author: "Kristoffer Wild"
format: docx
editor: visual
---

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
```

You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```
```

Text headings

Text

Code 'Chunks'

```
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/

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.

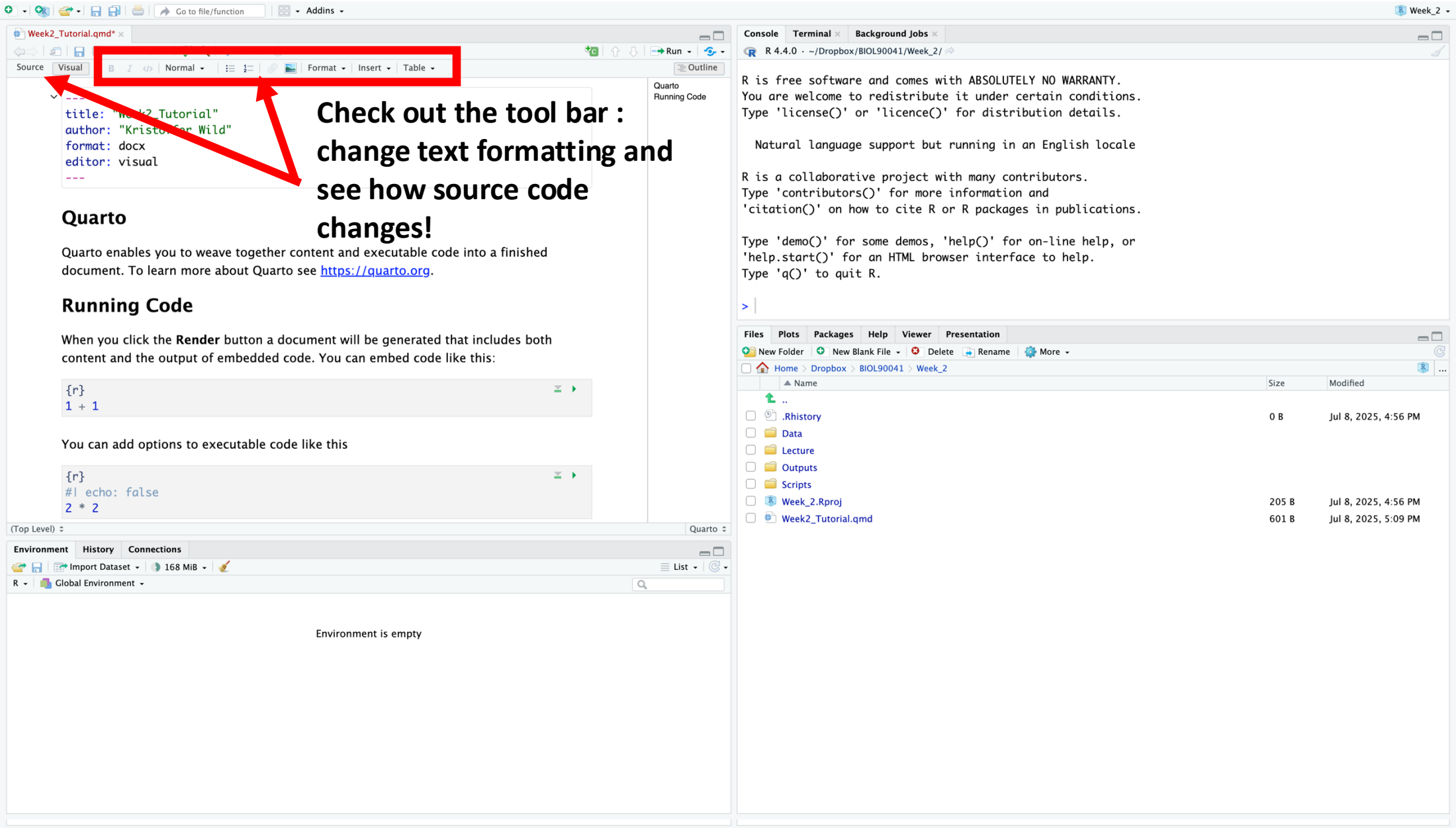
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Home > Dropbox > BIOL90041 > Week_2

Name	Size	Modified
..		
.Rhistory	0 B	Jul 8, 2025, 4:56 PM
Data		
Lecture		
Outputs		
Scripts		
Week_2.Rproj	205 B	Jul 8, 2025, 4:56 PM
Week2_Tutorial.qmd	601 B	Jul 8, 2025, 5:09 PM

Environment is empty


Week2_Tutorial.qmd*
Source
Visual
Normal
Format
Insert
Table
Run
Outline
Quarto
Running Code
title: "Week2_Tutorial"
author: "Kristofer Wild"
format: docx
editor: visual
Quarto
Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.
Running Code
When you click the Render button a document will be generated that includes both content and the output of embedded code. You can embed code like this:
{r}
1 + 1
You can add options to executable code like this
{r}
#| echo: false
2 * 2
(Top Level)
Quarto
Check out the tool bar : change text formatting and see how source code changes!
Environment
History
Connections
Import Dataset
168 MiB
List
R
Global Environment
Environment is empty
Console
Terminal
Background Jobs
R 4.4.0 · ~/Dropbox/BIOL90041/Week_2/
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.
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Delete
Rename
More
Home
Dropbox
BIOL90041
Week_2
Name
Size
Modified
.Rhistory
0 B
Jul 8, 2025, 4:56 PM
Data
Lecture
Outputs
Scripts
Week_2.Rproj
205 B
Jul 8, 2025, 4:56 PM
Week2_Tutorial.qmd
601 B
Jul 8, 2025, 5:09 PM

Quarto live DEMO with Kris

- Here we will go over…
  - setting up metadata to render into a word document
  - formatting code chunks
  - how code should be in code chunks
  - practicing writing results under code chunks
  - rendering document with appropriate code, plots, and write up

Next week…

But first…reminder we have a workshop this week!

- Wednesday, 11-1 at PAR125-L1-Room 127 (WEBS)
    - Cleaning your data and coding assignment rubric!

- Next week we will cover:
    - summarise
    - gsub
    - stringr
    - merging
    - joining