First Steps with RStudio

Data Analysis and Forecasting Course

Prof. Bee M.

Teaching Assistant: Tesfaye Salarin, Ph.D.

University of Trento

Contacts:

Email: tesfaye.salarin@unitn.it

Office: 159A/1st Floor, Via Inama 5, Trento (Italy) — Old Economics Building

Office Hours: Thursdays, 11:00-13:00

Why R?

- Open-source, free, and community-driven
- Ideal for data analysis, statistics, and visualization
- Widely used in academia, finance, and industry
- Huge ecosystem of packages (`CRAN`, `Bioconductor`, `GitHub`)

Exercise: Go to `https://cran.r-project.org` and check how many CRAN packages exist today.

Step 1: Getting Started - installing

- Download and Install R (`https://cran.r-project.org`)[1]
- Download and Install RStudio Desktop `https://posit.co/download/rstudio-desktop/` (the GUI = Graphical User Interface)
- Scroll to "All installers and tarballs" to find your OS version
- Check installation with:
  - `version` in R
  - `RStudio.Version()` in RStudio

Exercise: Open RStudio → Console pane (bottom-left corner if you've just installed RStudio) → type `1+1`. What happens?

[1] Choose your system: Windows, Mac, or Linux (Debian, Fedora/RedHat, Ubuntu)

Alrighty then...

- → **N.B.** R works fine without RStudio, but it acts like an "engine"/"terminal". For any graphical user interaction with R, you need RStudio.
- → **WARNING:** Always keep your R scripts and datasets in the same working directory (or use projects in RStudio) for your scripts to execute smoothly!
- → **WARNING:** Use projects to manage your work more efficiently!

How to set a common working directory?

Option 1: Beginner (manual paths)

- Create a subfolder in `Documents` (e.g., `Rcourse/ClassX`)

Windows:

- `C:/Users/username/Documents/Rcourse/ClassX`

Mac:

- `/Users/username/Documents/Rcourse/ClassX`

Option 2: Recommended (reproducible)

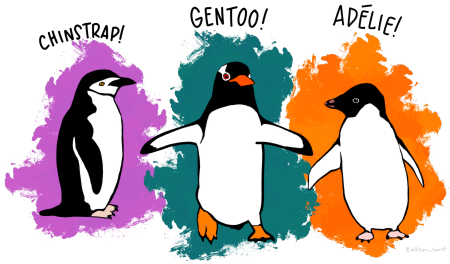
- Use **RStudio Projects**:
  - Create/open a `.Rproj` file in your course folder
  - The working directory is set automatically

⇒ **Use *`getwd()`* to confirm your current working directory!!!** ⇐

Data is everywhere...

Titanic Dataset

Penguins Dataset

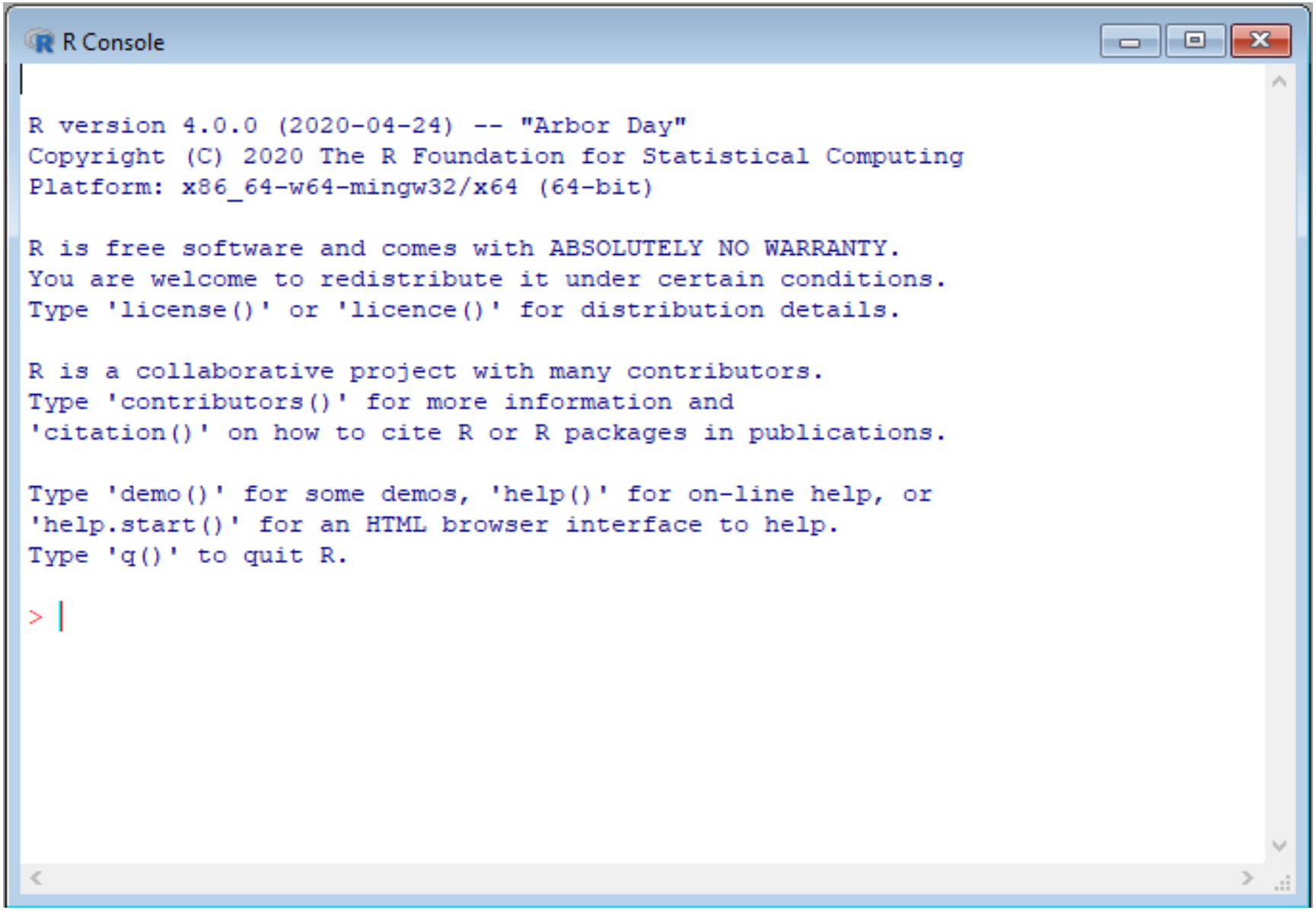


Iris Dataset

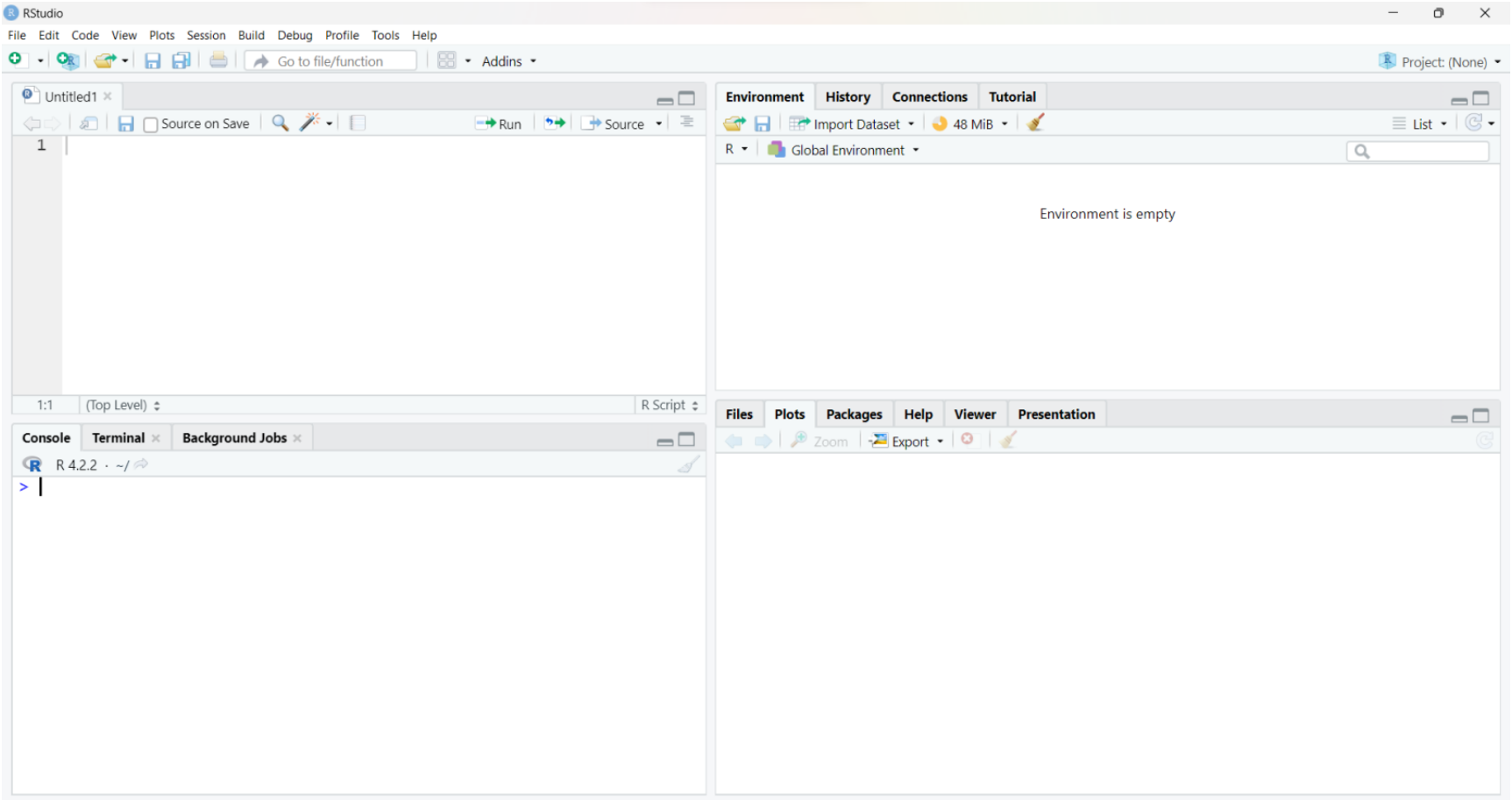
And the list goes on and on...

- *Boston Housing (Kaggle)* - House prices
- *Wine Quality (UCI)* - Predict wine scores
- *MNIST (Kaggle)* - Handwritten digits
- *Heart Disease (UCI)* - Medical classification
- *IMDB Reviews (Kaggle)* - Sentiment analysis
- *Fake Job Postings (Kaggle)* - Text classification

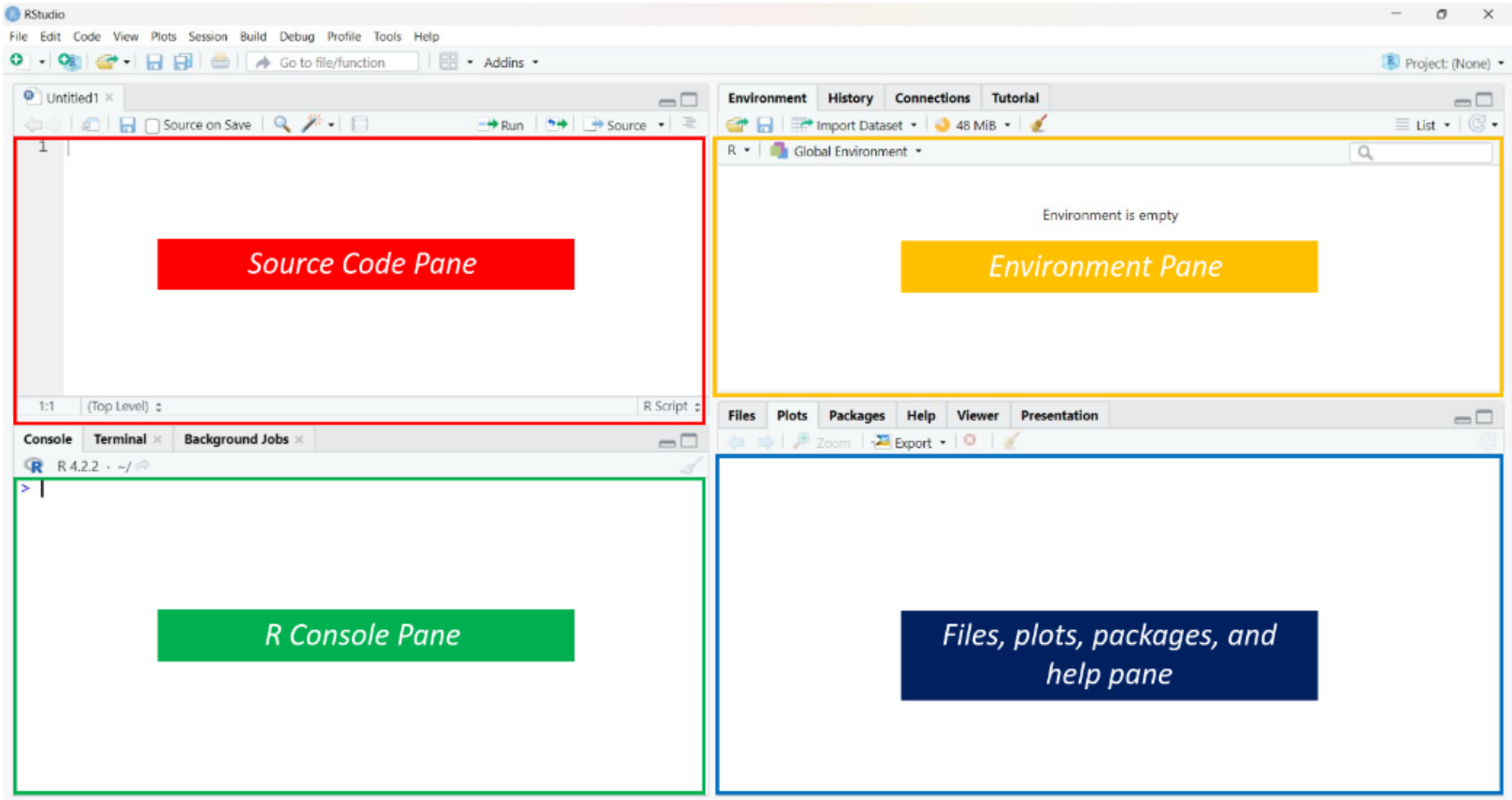
R Terminal


R Console
R version 4.0.0 (2020-04-24) -- "Arbor Day"
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.
>

RStudio Interface



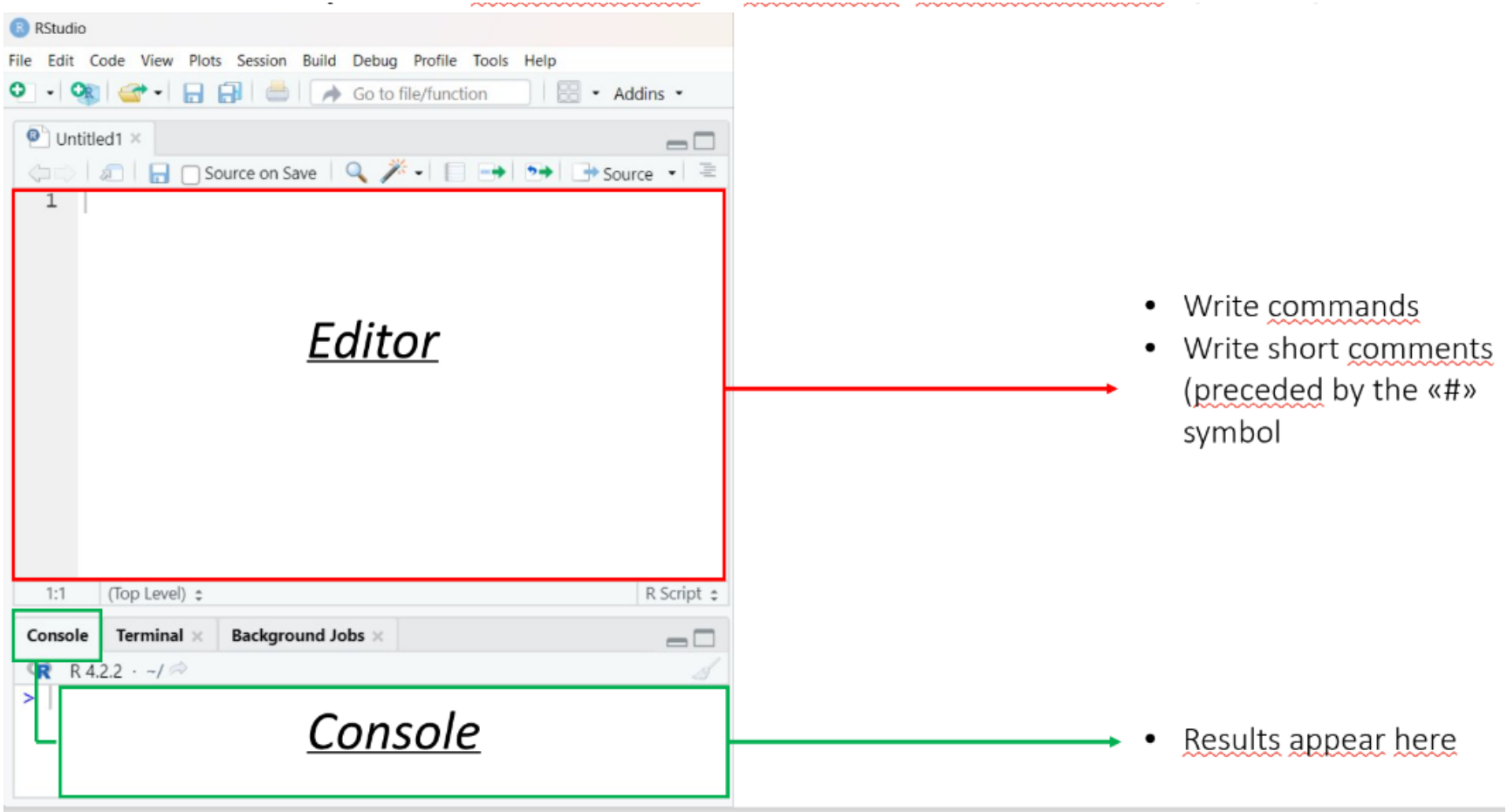
RStudio Panes



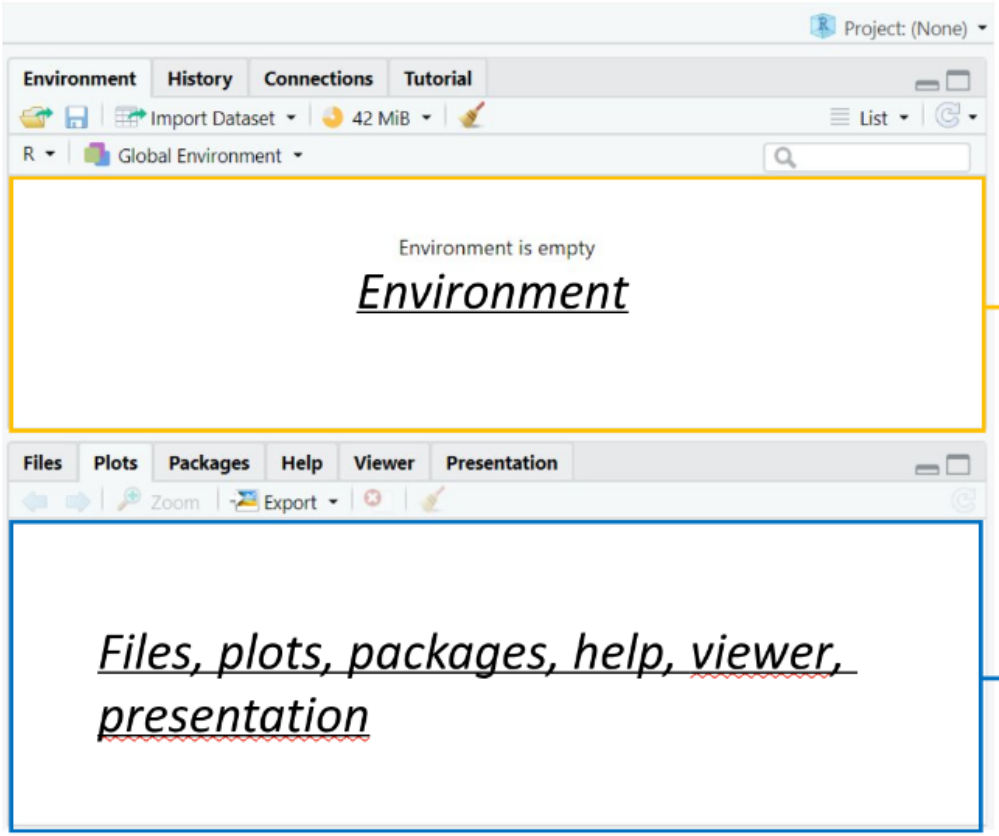
Step 2: RStudio - Main Sections

- **Source Code Pane**: where you write and save scripts
- **Console Pane**: where commands are executed
- **Environment Pane**: lists all active objects
- **Files/Plots/Packages/Help Pane**: for file access, plots, and help

Editor & Console



Environment & Viewer



This is where you see what vectors, data sets, and values you created while running codes in R

This is where you see the list of files from any given directory, the plots you created, your installed packages, and the **Help** tab.

Getting started - updating packages in R

Keeping packages up-to-date is important! (Following on Prof.Bee's advice of last monday

- **Update all installed packages:**
  `update.packages(ask = FALSE)`
- **Update a specific package:**
  `install.packages("dplyr")`
- **Check which are outdated:**
  `old.packages()`
- **Update from GitHub (dev version):**
  `remotes::install_github("tidyverse/ggplot2")`
- **RStudio GUI:**
  Tools → Check for Package Updates

Step 2: R as a Calculator

- Basic operations: `+ - * / ^`
- Math functions: `sqrt()`, `log()`, `exp()`
- Logical tests: ==, !=, <, >=

Exercise: Compute $\sqrt{16}$, log(10), and check whether $5^2 == 25$ in R.

Step 3: Variables and Assignment

- Use <- to assign values
- Example: `x <- 5`
- Inspect type with: `class(x)`, `typeof(x)`
- Remove objects: `rm(x)`

Exercise: Create two variables `a = 10`, `b = 3` Compute: `a+b`, `a/b`, `a^b` Remove the newly created variable from the environment

Step 4: Data Structures (1/2)

One-dimensional:

- **Vectors:** homogeneous elements e.g. `v <- c(1,2,3,4,5)`
- **Lists:** heterogeneous elements e.g. `L <- list("apple", 3.14, TRUE)`

Exercise: Create a vector of numbers 1-10. Use `mean()`, `sum()`, `length()`

Step 4: Data Structures (2/2)

Two-dimensional:

- **Matrix:** homogeneous elements, arranged in rows/columns e.g. `M <- matrix(1:9, nrow=3)`
- **Data frame:** tabular, columns can differ in type e.g. `df <- data.frame(Name=c("A","B"), Score=c(90,85))`

Exercise: Create a data frame with two columns: `Student` and `Grade`. Add 3 rows with your own values.

Debugging Quiz 1: Basic Arithmetic

Find the bug!

```
mean[1,2,3,4,5]
```

Debugging Quiz 1: Basic Arithmetic

Find the bug!

```
mean[1,2,3,4,5]
```

Question: Why does this fail? What should it do instead?

Debugging Quiz 1: Basic Arithmetic

Find the bug!

```
mean[1,2,3,4,5]
```

Question: Why does this fail? What should it do instead?

Solution: Functions in R use parentheses, not square brackets.

```
mean(c(1,2,3,4,5))
```

Step 5: Built-in Datasets

R comes with many datasets for practice.

- `iris` (flowers)
- `mtcars` (cars)
- `airquality` (NYC air quality)

Exercise: Load the `iris` dataset: `head(iris)`. Check column names: `colnames(iris)`. What is the average Sepal.Length?

Step 6: Basic Plotting

- Quick plots: `plot()`, `hist()`, `boxplot()`
- Example: `hist(mtcars$mpg)`

Exercise: Draw a histogram of `iris$Sepal.Length`. Draw a boxplot of `mpg` by `cyl` from `mtcars`

Debugging Quiz 2: Vectors

Find the bug!

```
numbers <- c(1; 2; 3; 4; 5)
```

Debugging Quiz 2: Vectors

Find the bug!

```
numbers <- c(1; 2; 3; 4; 5)
```

Question: What's wrong with this syntax?

Debugging Quiz 2: Vectors

Find the bug!

```
numbers <- c(1; 2; 3; 4; 5)
```

Question: What's wrong with this syntax?

Solution: Use commas `,` to separate elements, not semicolons `;`.

```
numbers <- c(1, 2, 3, 4, 5)
```

Step 7: Installing Packages

- Install once: `install.packages("ggplot2")`
- Load every session: `library(ggplot2)`
- CRAN and *GitHub* are the main sources

Exercise: Install and load the package `dplyr`. Try: `library(dplyr)`; `glimpse(mtcars)`

Mini Project: Bring It All Together

Task: Using the `mtcars` dataset...

1. Find the average horsepower (`hp`)
2. Create a new column `power_to_weight = hp/wt`
3. Plot a histogram of `power_to_weight`

Bonus: Use `ggplot2` for the histogram

Course Objectives Recap

By the end of this course, you will be able to:

1. Gain confidence in using RStudio as a daily working tool
2. Understand the basics of the R programming
3. Manipulate variables, vectors, lists, and data frames using specific packages to analyze and organize data
4. Learn and implement forecasting models
5. Explore and visualzie data to communicate insights

Debugging Quiz 3: Data Frames

Find the bug!

```
df <- data.frame(Name = c("Alice", "Bob"), Age = (25, 30))
```

Debugging Quiz 3: Data Frames

Find the bug!

```
df <- data.frame(Name = c("Alice", "Bob"), Age = (25, 30))
```

Question: Why does this throw an error?

Debugging Quiz 3: Data Frames

Find the bug!

```
df <- data.frame(Name = c("Alice", "Bob"), Age = (25, 30))
```

Question: Why does this throw an error?

Solution: Vectors must be created with `c()`.

```
df <- data.frame(Name = c("Alice", "Bob"), Age = c(25,
30))
```

Next Steps

Eager to know what we'll do next? Try out yourselves in the meantime

- Practice with real datasets (Excel, CSV imports)
- Learn `dplyr` for data manipulation
- Learn `ggplot2` for advanced visualization
- Explore reproducible research with `RMarkdown`

Forecasting in R with `ffp3`

What is forecasting?

- Forecasting = using past data to predict the future
- Widely applied in economics, finance, business, and science
- Common methods: time series models, regression, machine learning

The `ffp3` Package:

- Stands for: *Forecasting with Flexible Probabilities (3rd version)*
- Allows simulation and forecasting under **different probability measures**
- Useful in: risk management, portfolio optimization, and financial modeling
- Part of modern forecasting workflows in R, alongside packages like `fable`, `forecast`, `tsibble`

Exercise: Install and load the package. `install.packages("ffp3")`
`library(ffp3)`