RStudio, Quarto, and LaTeX

Why use RStudio and LaTeX?

> **Note**
>
> **RStudio** enables the combination of plain text, R code, statistical results, and various coding languages that enable nice formatting into a single coherent document. In doing so, we ensure that our work is reproducible. The following document provides a high level introduction to RStudio, Quarto (.qmd files), and the process of rendering .qmd files into a single document.

> **Note**
>
> **LaTeX** is required to allow RStudio to render .qmd files, which represent an amalgamation of R code, plain text, and other coding languages, into a .pdf document. **In this class, you will be expected to turn in all documents as .pdf files.**

RStudio architecture

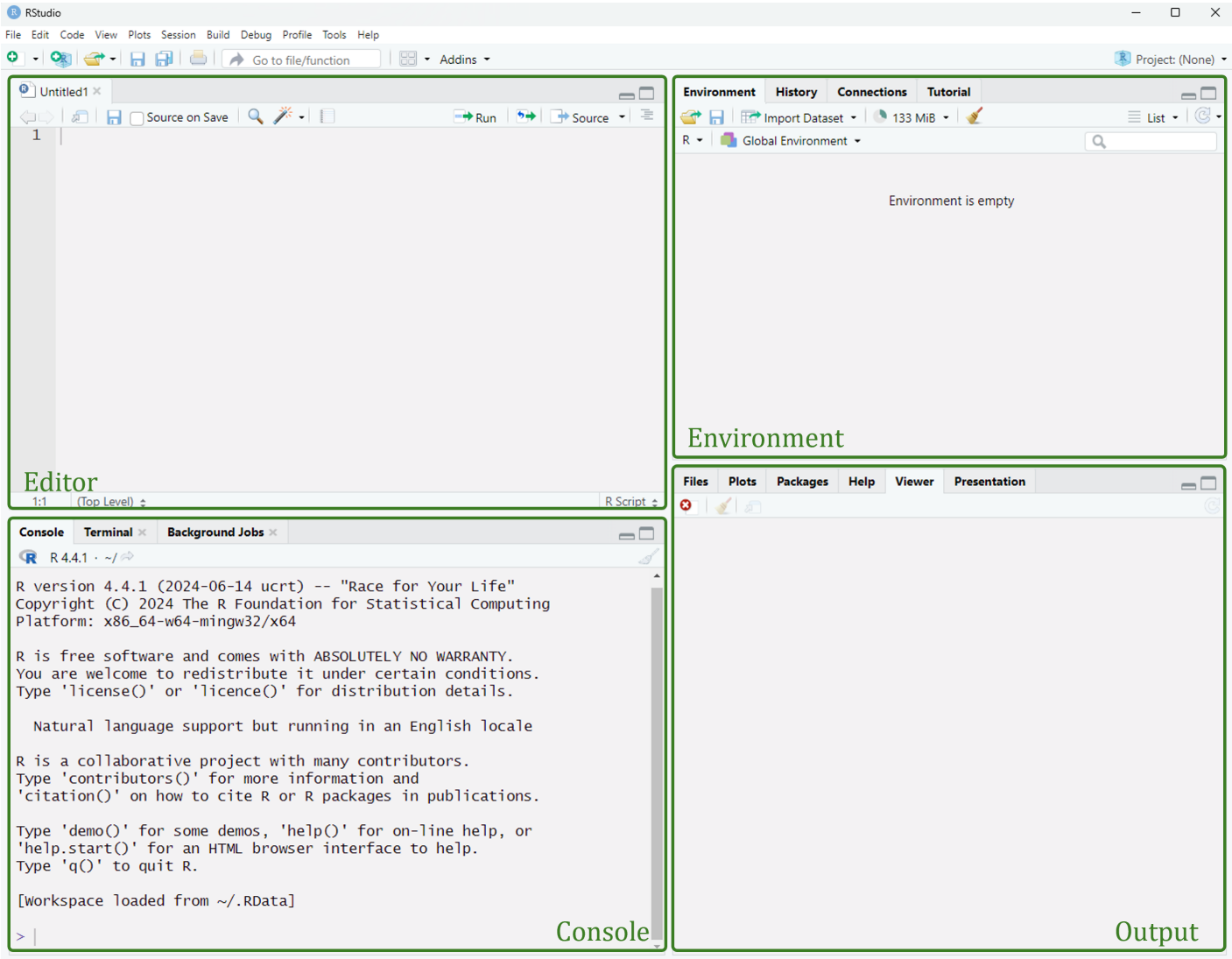
> **Note**
>
> RStudio is an integrated development environment (IDE) that we will use to call R to run our code; very rarely (if ever) will we open R directly.

The **editor** pane is in the top left corner. This is where you will write and edit code, and where your .qmd files will be displayed.

The **environment** is in the top right corner. This is where you can view the objects and data you have in RStudio.

The **console** is in the bottom left corner. This is where code output is displayed.

The **output** pane is in the bottom right corner, this is where figures are displayed, where you can view compiled documents, and where you can find package help.

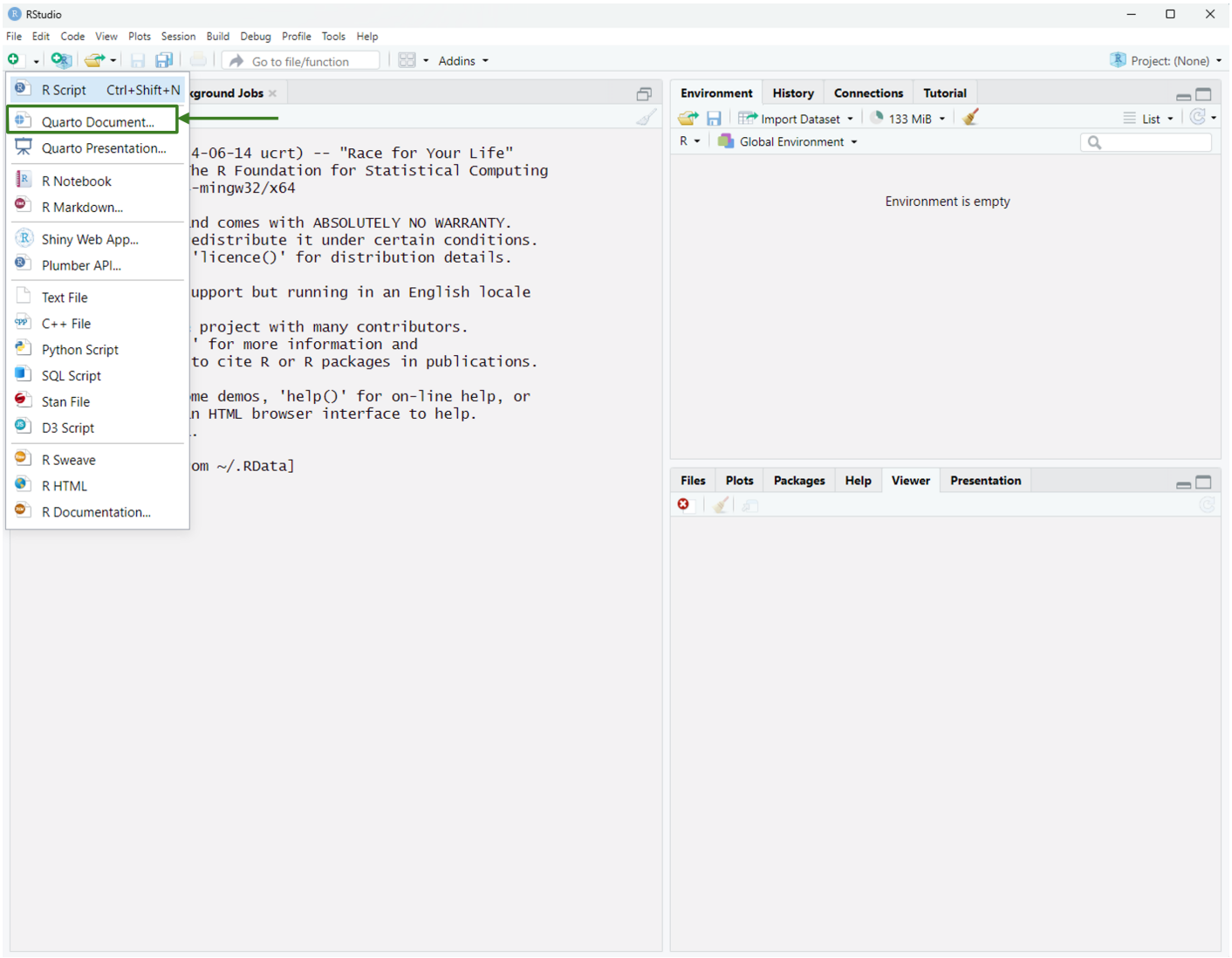


What are .qmd files?

For notes, homeworks, labs, and exams, I will provide you with a Quarto document (.qmd file) used to create a pdf. RStudio (but really, Pandoc) allows us to render a combination of plain text, code, and output into a variety of file types using a single reproducible document known as a Quarto document (.qmd file). While RStudio is capable of rendering documents into numerous file types, we will generally focus on .pdf in this class.

A .qmd file is generally composed of R code chunks, which contain R code that is then evaluated to produce output. Code can be run directly from an R code chunk, without pasting the code into the console, by selecting the line of code and pressing **Ctrl + Enter** on Windows and **Cmd + Return** on Mac.

You can create a new Quarto document in RStudio as shown below:



RStudio workflow

Tip

1. Download the .zip folder containing the .qmd file from Canvas.
2. Extract the .zip folder into an uncompressed folder.
3. Open the .qmd file in RStudio, and set your working directory to the source file location. To do this, select the following options, beginning with **session** in the taskbar at the top of the screen: **Session** → **Set Working Directory** → **To Source File Location**.
4. Edit the document with responses and R code.
5. Render the document as you go. My advice would be to *render early and often*.

A quick note on formatting

Generally speaking, .qmd files will be sensitive to indentation when formatting. For example, code chunks and callout boxes that are contained within a number list must be indented to render properly.

1. Here is an example (see the .qmd file for the indenting - notice the indent option in the code chunk).

```
round(pi, 5)
```

```
[1] 3.14159
```

> **Note**
>
> Callout boxes must also be indented (three spaces under a single digit number and four spaces under a two digit number).

Example functionality in RStudio

R as a calculator

Among many things, R is a calculator. For example,

```
1 + sqrt(4^2)
```

```
[1] 5
```

Note that you can either run the above code by putting your cursor on line 96 and pressing **Ctrl + Enter** on windows or **Cmd + Return** on Mac, or by typing `1 + sqrt(4^2)` into the console and pressing **Enter/Return**.

Manipulating objects

To do more useful things in R, we need to assign values to an object. To create an object, we tell `R` the objects name, followed by an assignment arrow (`<-`), and finally the value of the object. This should look something like this:

```
object <- 1
```

In general, please use informative object names. For example, `bat_data <- ...` would be preferred to `a <- ...` if loading a data set containing information on bats. All objects in R are stored in the environment. For example, running the following code creates a **vector** of numbers called `num_vec`, which can be seen in the environment.

```
num_vec <- c(8, 6, 7, 5, 3, 0, 9)
num_vec
```

```
[1] 8 6 7 5 3 0 9
```

A things to note when creating objects:

- `R` is case sensitive, so if you name your variable `cat`, and then try to run the code `Cat + 2`, you will get an error saying that `Cat` was not found.
- You want your object name to be explanatory but not long, think `current_temp` versus `current_temperature`
- Object names cannot begin with a number, but can end with a number.
- Object names should generally only include periods or underscores as punctuation, and spaces should be avoided.
- You should not name your object the same as any common functions you may use (`mean`, `sd`, etc.)
- Objects can have a variety of classes, including character vectors, numeric vectors, data objects, and many more.

Objects can then be manipulated for a variety of purposes, including arithmetic and statistical operations.

```
num_vec^2
```

```
[1] 64 36 49 25  9  0 81
```

```
mean(num_vec)
```

```
[1] 5.428571
```

Importing data sets

> **Note**
>
> Data sets can be imported into R using the Import Dataset wizard found in the environment tab. This wizard creates a line of code that can be used to import a data set into R (you may use either the base R version or readr version).

Once you have obtained that line of code from the wizard, **you must paste it into the .qmd file.** An example is provided below. Note that when you click render, RStudio assumes that the data set is in the same directory as the source .qmd file. Therefore, **it is highly recommended to set the working directory to the source file location** when working in R, and to save all required files (images and data sets) in the same folder as the .qmd file.

```
library(readr)
barre_weather <- read_csv("barre_weather.csv")
```

Now that the data are in R, we can manipulate the data object

```
# print first six rows
head(barre_weather)
```

```
# A tibble: 6 x 11
  Date       Perci~1 Snow ~2 Snowf~3 Max T~4 Min T~5 Ave W~6 FOG   Sleet Smoke~7
  <chr>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <lgl> <lgl> <lgl>
1 06/01/1948    0          0       0    80.1    43.0      NA FALSE FALSE FALSE
2 06/02/1948    0          0       0    79.0    48.0      NA FALSE FALSE FALSE
3 06/03/1948    0          0       0    82.0    48.0      NA FALSE FALSE FALSE
4 06/04/1948    0          0       0    84.0    50        NA FALSE FALSE FALSE
5 06/05/1948    0.15       0       0    62.1    48.9      NA FALSE FALSE FALSE
6 06/06/1948    0          0       0    69.1    34.0      NA FALSE FALSE FALSE
# ... with 1 more variable: Thunder <lgl>, and abbreviated variable names
#   1: `Percipitation in.`, 2: `Snow Depth in.`, 3: `Snowfall in.`,
#   4: `Max Temp F`, 5: `Min Temp F`, 6: `Ave Wind Spd mph`, 7: `Smoke/Haze`
```

```
# calculate the mean of the max temperature column
mean(barre_weather$`Max Temp F`)
```

```
[1] 52.65269
```

```
# calculate the standard deviation of the max temp column
sd(barre_weather$`Max Temp F`)
```

```
[1] 21.29714
```