Statistical Programming for the Social Sciences Using R

Wesley Stubenbord

2024-01-30

Contents

Introduction

Welcome! This is the companion website for *Statistical Programming for the Social Sciences Using R*, taught at the Sciences Po Reims campus for the Spring 2024 term.

This website will contain the relevant tutorials for each week's lesson as well as other resources that you may find helpful throughout the course. The syllabus, assignment submission portals, and other files can be found on the course Moodle site.

Chapter 1

An Introduction to `R`

To get started, you will need to install two things:

1. `R`, a programming language
2. RStudio, a software program that helps you program in `R`
    - This type of software program is called an IDE, an Integrated Development Environment

You don't necessarily need RStudio to program in `R`, but it makes life a lot easier and it's what we'll be using throughout the course.

1.1 Installing `R`

To install `R`, go to https://cran.irsn.fr/index.html, select the appropriate operating system, and follow the instructions.

For example, if you have a Mac, you will click on "Download R for macOS," followed by clicking on the "R-4.3.2-arm64.pkg" link beneath the "Latest release" header.

If you have a PC running Windows, you will click on "Download R for Windows" followed by "install R for the first time" and "Download R-4.3.2 for Windows."

In either case, your browser should start downloading an executable file which you will then need to run to install `R`.

CAUTION - A couple of things you may need to watch out for:

1. If you are using an older laptop, you may need to download a different version of R or RStudio. If in doubt, read the installation pages and refer to your operating system version to find the right version for your needs.

2. If you have very little hard drive space on your computer, you may need to clear some space before you install RStudio. The latest RStudio version requires 215 MB and you'll likely need some additional space for other software and data we will be using in the course later on (around 2 GB should suffice).

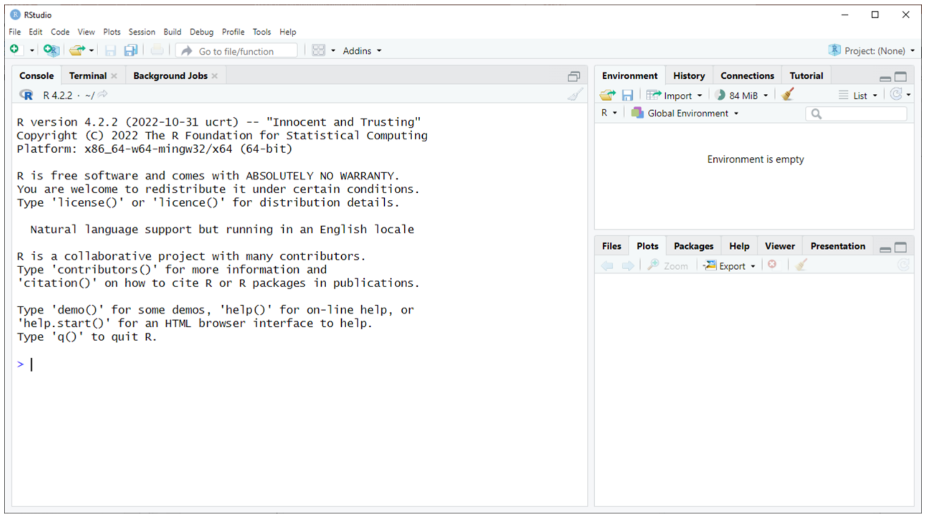
1.2 Installing RStudio

Once you've installed R, go to https://posit.co/download/rstudio-desktop/.

Posit (a company formerly known as RStudio) offers RStudio Desktop free of charge. Posit also offers a cloud-hosted version of the software (called Posit Cloud) which has both free and paid tiers. If you have trouble running RStudio Desktop on your computer, you may wish to consider using a Posit Cloud account, as described in the course syllabus.

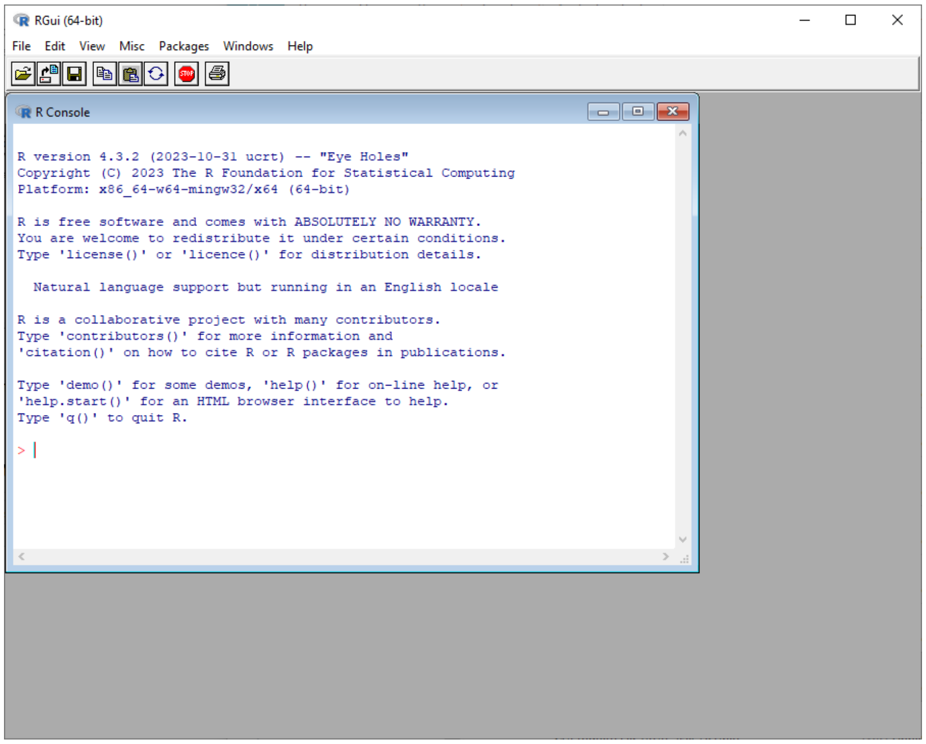
Step 1 is complete, you've already installed R. On the landing page linked above, you'll find different versions of RStudio according to your computer's operating system. Select the operating system that applies to your particular case, download the installer, and once it has downloaded, run the installation file from your computer and follow the on-screen steps.

If all goes well, your RStudio screen should look something like this once it is correctly installed and running:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console Terminal Background Jobs
R 4.2.2 · ~/
Environment History Connections Tutorial
Import
84 MiB
List
R
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
Zoom
Export

```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 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.

  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.

> |
```

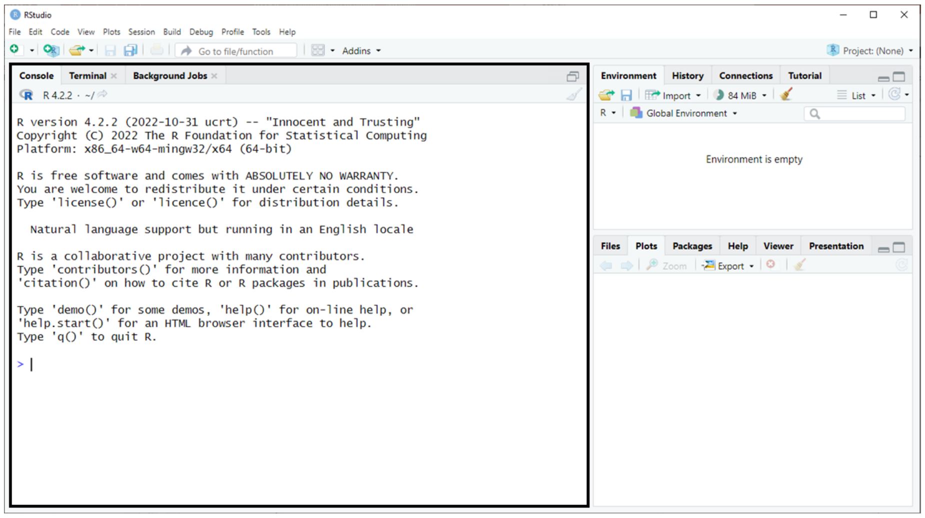
On the other hand, if your screen looks like the image below, it means that you've accidentally opened RGui (a basic graphical user interface included with `R`) and not RStudio. We're always going to be working in RStudio for this class, so you can close out of RGui and open RStudio instead.



```
R version 4.3.2 (2023-10-31 ucrt) -- "Eye Holes"
Copyright (C) 2023 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.

  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.

> |
```

1.3 Using the Console

Now the fun begins. The RStudio window you've opened consists of a few different parts. The most important of these right now is the console pane (highlighted below).


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console Terminal Background Jobs
R 4.2.2 · ~/
Environment History Connections Tutorial
Import
84 MiB
List
R
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
Zoom
Export

```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 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.

  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.

> |
```

The console allows you to interact with your computer using R. So, for example, if I want to use R as a calculator, I can type in the following code in the console and obtain a result:

```
1+1
```

What happens when you press Enter on your keyboard? You get something like this:

```
1+1
```

```
## [1] 2
```

You've provided an **input**, 1+1, and received an **output**, 2. The first number in brackets, [1], indicates the number of the line of output. This is helpful when you are running more complex code that generates multiple lines of output. For now, it's important not to confuse this number with the output itself, 2.

Try entering a few more inputs, such as:

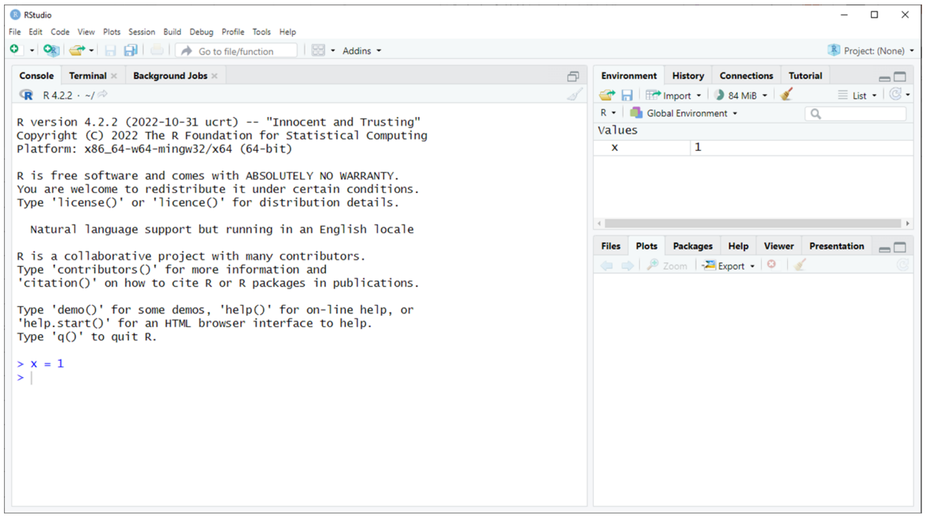
1. 10/3
2. (10/3) + 1
3. (10/3) + 1 + (5.111)^2

You can see that R is able to handle basic math operations with ease. What about other operations? Can R accept variables?

Try this:

```
x = 1
```

What happens when you press `Enter`? What happened to the output?


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console Terminal Background Jobs
R 4.2.2 · ~/
Environment History Connections Tutorial
Import 84 MiB List
R Global Environment
Values
x 1
Files Plots Packages Help Viewer Presentation
Zoom Export

```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 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.

  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.

> x = 1
> 
```

The trick is that there is no output. You've stored a value, `1`, in a variable, `x`, somewhere in your computer's memory or in what we call the environment. RStudio shows you this in the Environment pane in the top right.

We can recall the value we input into our variable, `x`, by entering just the variable name in the console.

```
x
```

```
## [1] 1
```

See! `R` remembers what you stored in your environment.

Try the following:

1. Can you assign a new value to your variable x?
2. Can you perform math operations on a variable (e.g., x*2)?
3. Can you create a new variable, y, and use it in math operations with x (e.g., x * y)? What might you have to do with y first?
4. Can you change the type of variable? What if, for example, I don't want x = 1, but I want x to be equal to the word "apple"?

1.4 Calculations with Variables

If you've made it this far, congrats! Here's another thing you can try. Enter the following in the console:

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

You'll notice, we're using a different operator here. It's a less than symbol (<) followed by a dash (-). This is called an **assignment operator**. It has the same function as the equals sign (=). You can use either, I just so happen to prefer the way this one looks.

What happens when you press `Enter`? You have created a **vector**. Like other types of variables in `R`, a vector is an object. A vector holds a set of values of the same type. In this case, the vector x is holding a set of numbers.

We can do all sorts of things with vectors and other objects in `R`. We can, for example, find the sum of a vector.

```
sum(x)
```

```
## [1] 15
```

How did we get an output of 15? 1+2+3+4+5 = 15. We can also find the mean of a vector.

```
mean(x)
```

```
## [1] 3
```

And, we can perform other operations. Try the following:

1. Can you find the median of the vector x? What about the mode?[1]

2. What happens when you multiply a vector?

3. Can you create a new vector which consists only of letters?

[1]As you will have found, `mode()` doesn't calculate a statistic here (although even if it had, you still wouldn't have gotten a meaningful statistic for this particular vector). Some functions are easy to guess, like `median()` , but others can be false cognates just like in a spoken language. We'll talk more about finding the purpose of a function in the next chapter.

1.5 Saving Your Work

As you've begun to see, working with a scripting language like `R` is quite different from working with software like Microsoft Excel or Google Sheets. You work interactively with data using code rather than by changing values directly in a user interface.

One of the great benefits of interacting in this way, particularly for the social sciences, is that it allows others to fully reproduce your work. If they have your code, they can repeat the exact same steps you took.

Think about it this way: when you change a value in a cell in an Excel spreadsheet, will someone else know what was in that cell before you changed it? No.

In `R`, however, we can leave the source data alone and save the code we use to manipulate it. Now others can see exactly what you've done to each value and reproduce your analysis.

There are a couple of different ways of saving your code:

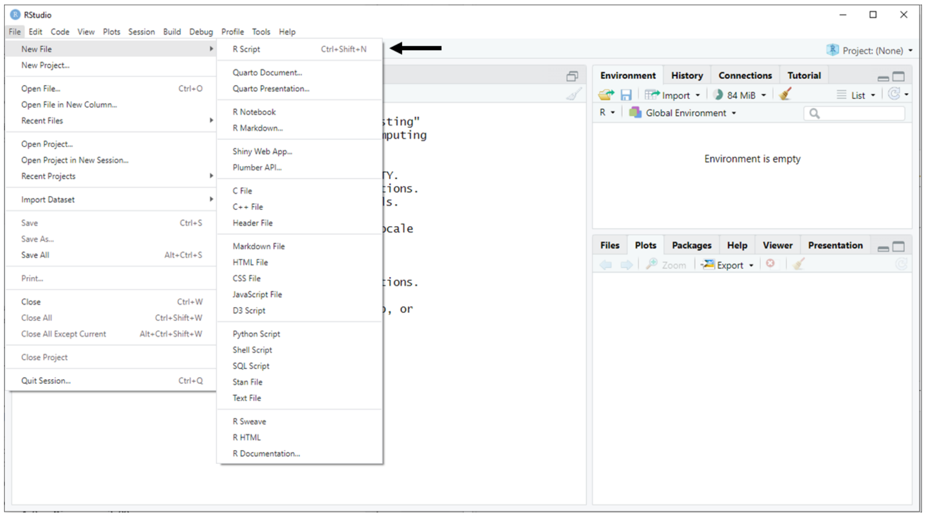
1. In an R Script file, which is a simple text file containing only your code

2. In an R Markdown file, which is basically an interactive notebook that allows you to see your code and the results next to each other

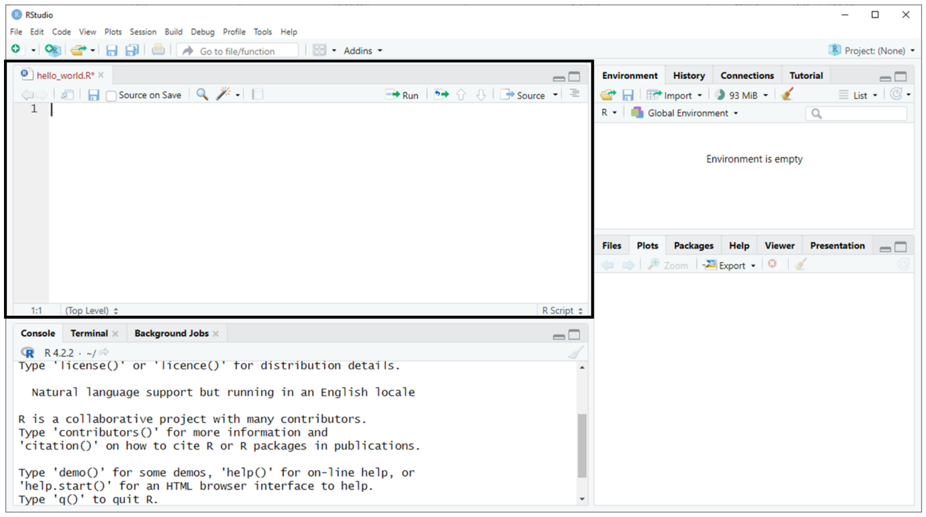
We're going to start with an R Script file and try out R Markdown later on.

1.6 Creating and Saving R Script Files

To create an R Script file in RStudio, go to File > New File > R Script.



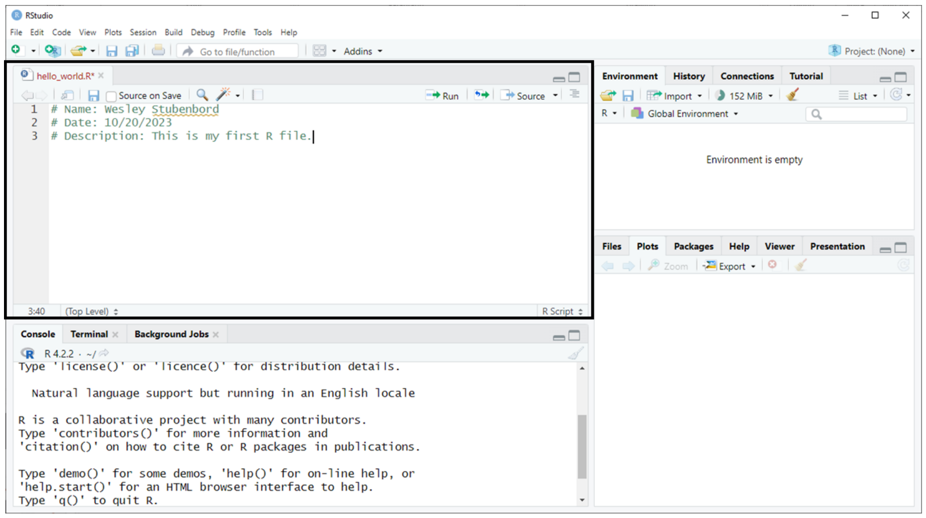
You should now have a window open in RStudio which looks like this:



```
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.
```

You can enter comments in your R Script file using a hash tag (#) at the beginning of each comment line. A hash tag lets R know that this line should not be run as code.

I like to add my name, the date, and a description to each file I use. I'll ask that you use a header for each R file you submit for this class as well.



```
# Name: Wesley Stubenbord
# Date: 10/20/2023
# Description: This is my first R file.
```

```
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.
```

Now, save your R Script somewhere on your computer. Go to File > Save As, choose a safe place on your computer to store it (I recommend creating a folder for this course), give it a name, and then press save. I called mine "hello_world".

1.7 Interacting in an R Script

Interacting in an R Script is slightly different from interacting with the console. Now when you type in code and hit `Enter`, it will not execute the code, it just creates a new line in your file.

To run code in a script in RStudio, you can either:

1. Select the lines you wish to run with your cursor and then press `Ctrl` + `Enter`
2. Or, put your cursor on the line you wish to run and click the **`Run`** button in the top right of the R Script file pane

The first option allows you to run multiple lines at a time and the second runs only the line you are currently on. The results of your code will appear in the console pane below your R Script file when run successfully.

After you finish modifying your R Script file, you can save it and close out of RStudio. The next time you wish to access your saved code, you can open your R Script file and it will be exactly as you left it.