

Statistical Programming for the Social Sciences Using R

Wesley Stubenbord

2024-01-28

Contents

1	About	5
2	An Introduction to R	7
2.1	Installing R	7
2.2	Installing RStudio	8
2.3	Using the Console	9
2.4	Variables	12
2.5	Saving Your Work	12
2.6	Creating and Saving R Script Files	13
2.7	Interacting with an R Script	15

Chapter 1

About

Welcome! This is the companion website for the Statistical Programming for the Social Sciences Using R course taught at Sciences Po on the 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 2

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.

2.1 Installing R

To install R, go to <https://cran.irsu.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, in most cases, 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:

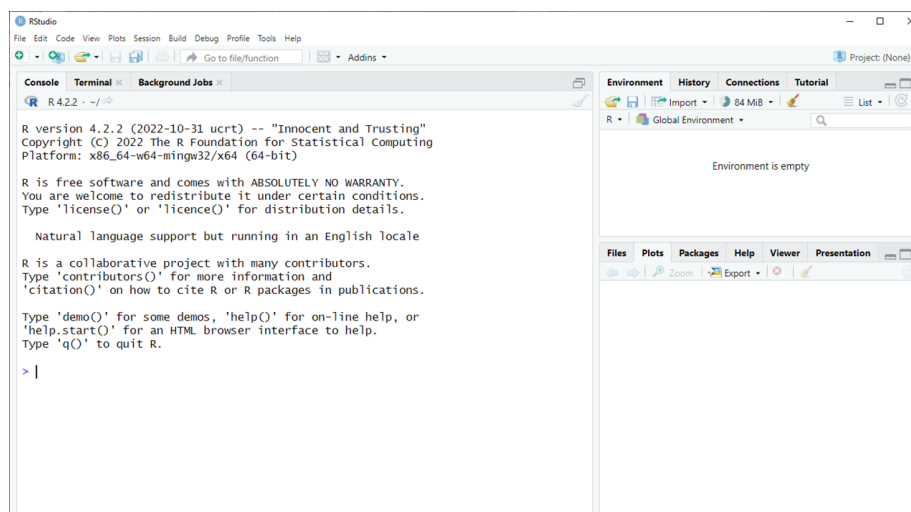
1. If you are using an older laptop, you may need to download a different version of R or RStudio. When in doubt, read the installation pages and refer to your operating system version.
2. If you have very little hard drive space on your computer, you may need to clear some space before you go to install RStudio. The latest RStudio version requires 215 MB and you'll need some additional space for other software and data later on in the course (around 2 GB should suffice).

2.2 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. They also have a cloud-based version of the software (Posit Cloud) which has a free tier and paid tiers. If you have trouble using RStudio Desktop in this course, you may consider using a Posit Cloud account as discussed in the course syllabus.

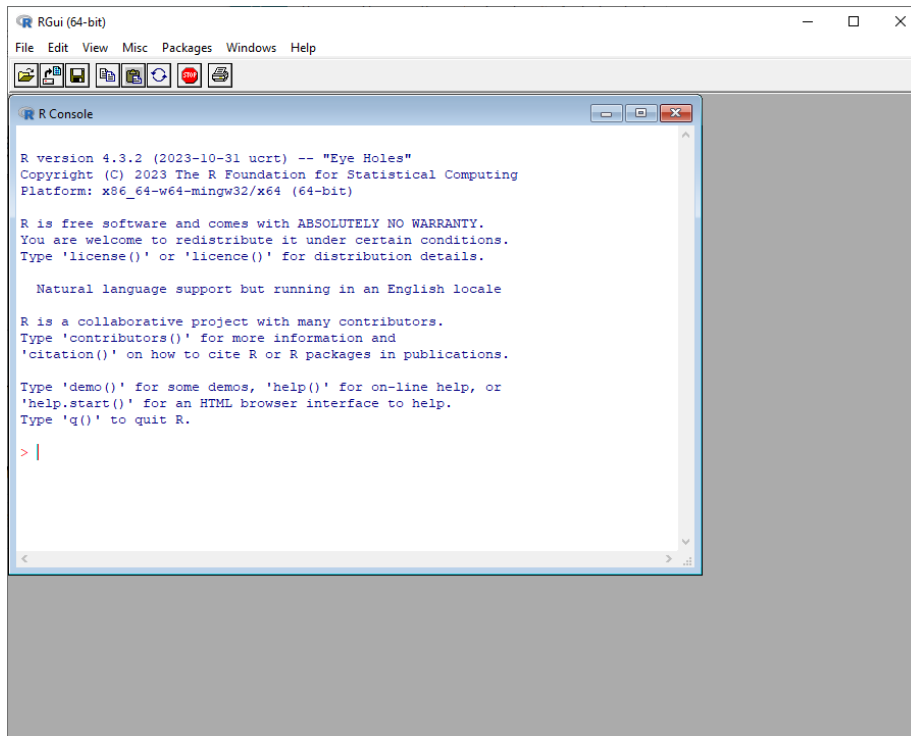
Step 1 is complete - you've already installed R. On the landing page linked above, you'll find different versions of RStudio according to your computer's operating system. Select the operating system that applies to your case, download the installer, and then, once it has downloaded, run the installer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



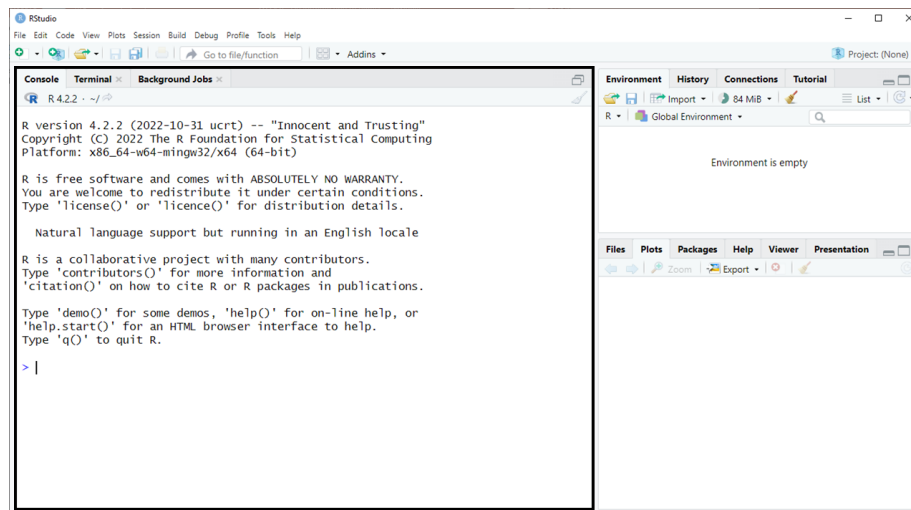
On the other hand, if your screen looks like the image below, it means that you've opened the base R GUI and not RStudio. We're always going to be

working in RStudio for this class, so you can close out of R GUI and open RStudio instead.



2.3 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 window (highlighted below).



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

```
1+1
```

What happens when you press enter? You get something like this:

```
1+1
```

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

In other words, you've provided an *input*, `1+1`, and received an *output*, `2`. The first number `[1]` in brackets indicates the number of the line of output. This is particularly helpful when you are running a complex bit of code, but it's important not to confuse it with the output.

Try entering a few more inputs, such as:

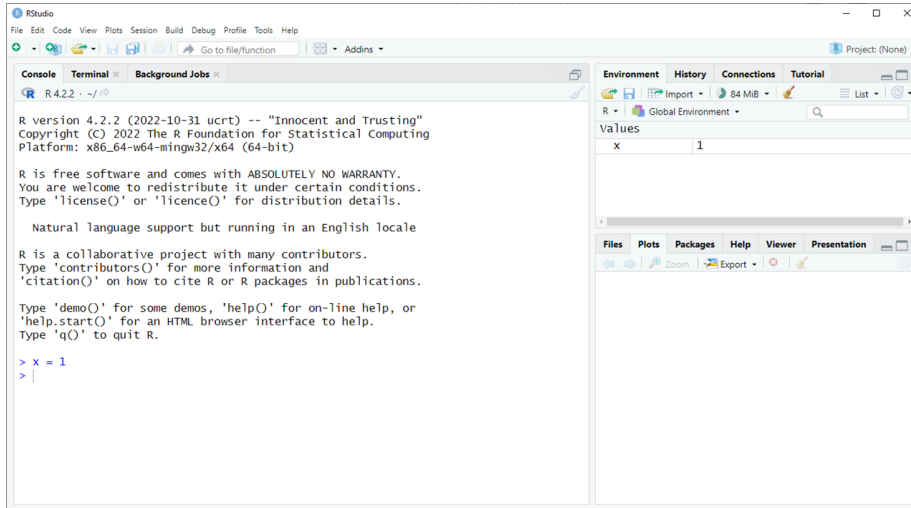
1. `10/3`
2. `(10/3) + 1`
3. `(10/3) + 1 + (5.111)^2`

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?



The trick here 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 might call the environment. RStudio shows you this in the Environment window 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”?

2.4 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 (=) and 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?
2. What happens when you multiply a vector?
3. Can you create a new vector which consists only of letters?

2.5 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.

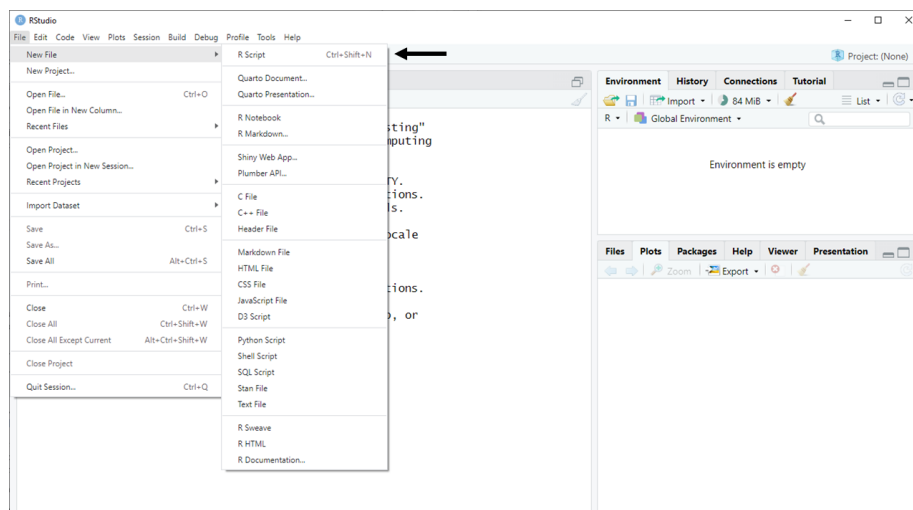
There are a couple of different ways of saving your code:

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

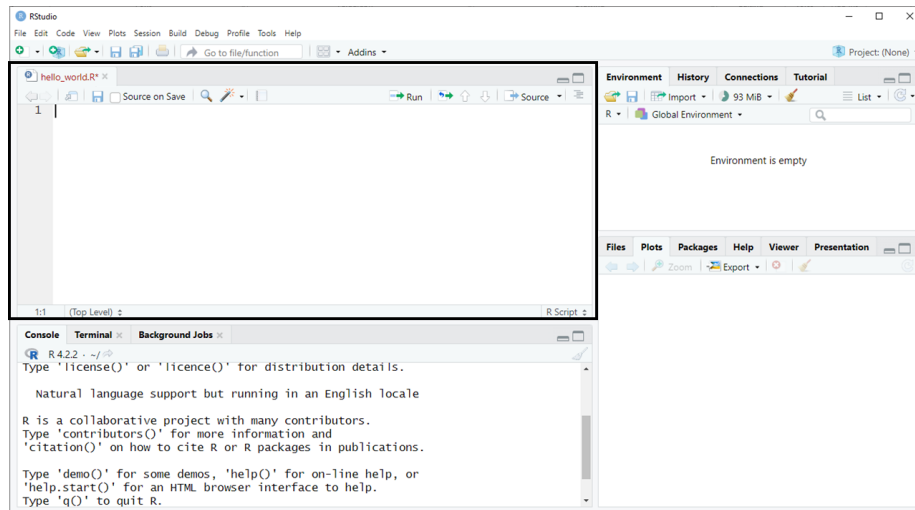
We're going to start with an R Script file and try out R Markdown later on.

2.6 Creating and Saving R Script Files

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

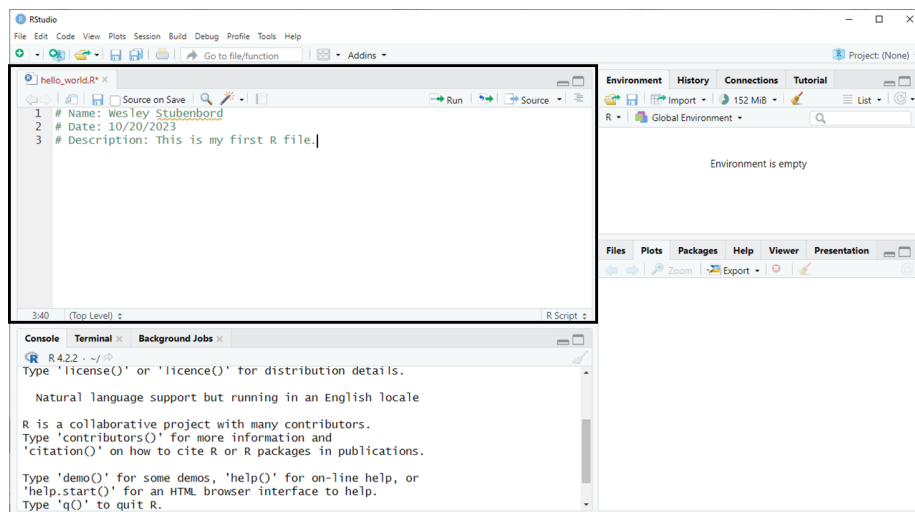


You should now have a window open in RStudio which looks like this:



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.



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”.

2.7 Interacting with an R Script

Interacting with an R Script is slightly different from interacting in 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 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 window

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 window below your R Script file when run successfully.

After you finish modifying your R Script file, you can save it and close out of R Studio. 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.