

Week 1: Getting Started with R

POP88162 Introduction to Quantitative Research Methods

Installation

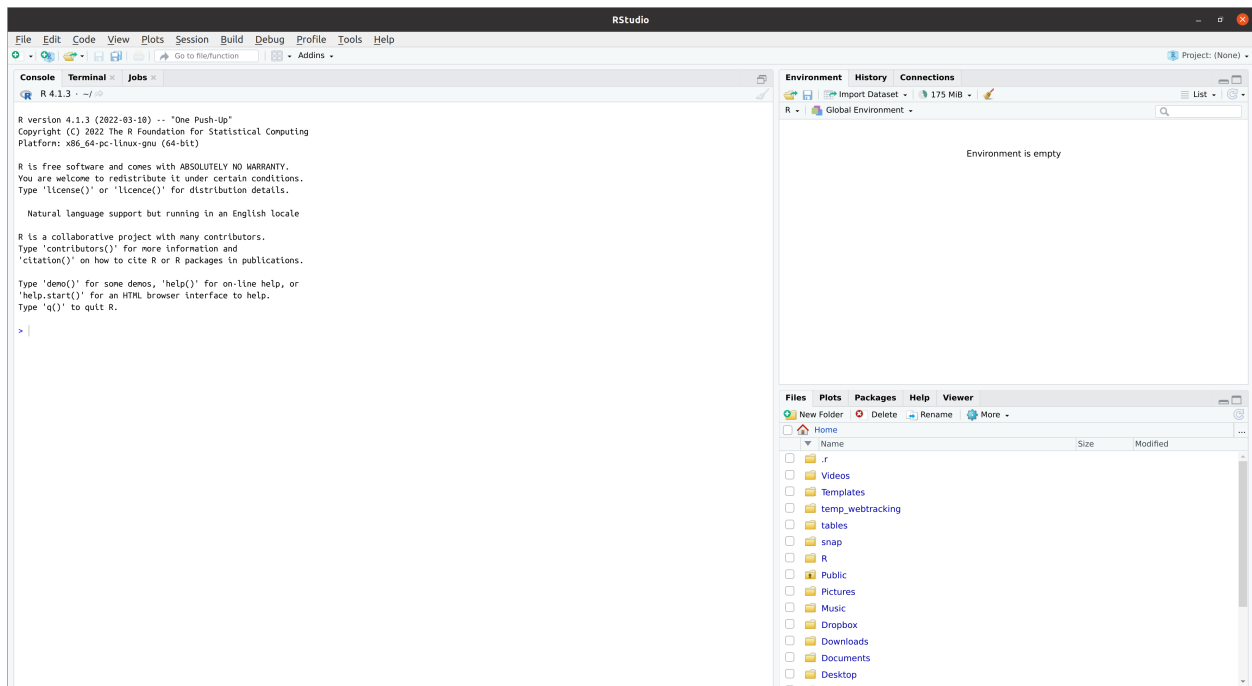
R is an interpretable programming language developed with a focus on data manipulation and statistical analysis. R is free and available for many platforms, including Windows, Mac and Linux.

First step for using R is installing it on your machine. R can be downloaded from [CRAN \(Comprehensive R Archive Network\)](https://cran.r-project.org/). Follow the directions on CRAN website for installing R on your specific platform.

Once you have your R installation complete, it's time to look at the tools that can help us write code more efficiently. While it is possible to use R from command-line or simple GUI (graphical user interface) that comes with basic installation, I recommend using a more advanced IDE (integrated development environment). There are different options available, but in recent years [RStudio](https://www.rstudio.com/) has become the de-facto standard IDE for programming in R. To install RStudio go to the website's [Downloads](https://www.rstudio.com/downloads/) page, download and install RStudio Desktop free version.

Interface

Locate installed RStudio on your machine and launch it. You should see a window like this:



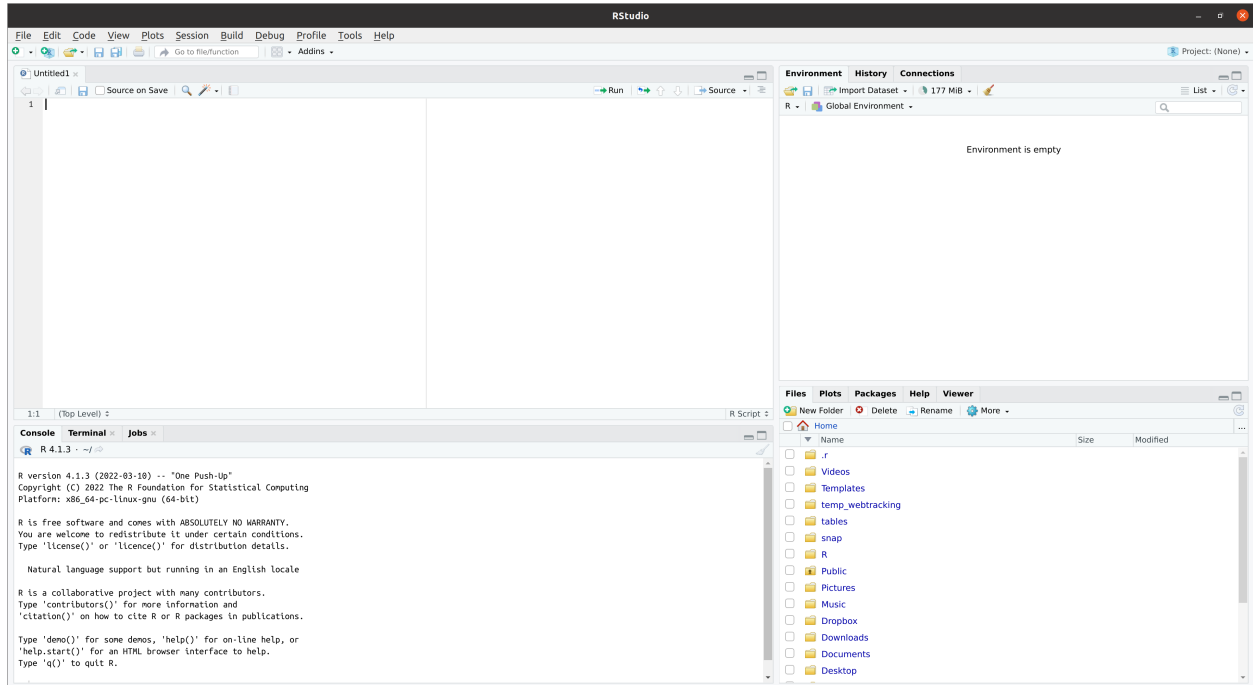
R Script

1. Let's start by creating a simple R script that would contain just one line:

3.5 + 2

Create a new R script by selecting **File -> New File -> R Script** (or by pressing **Ctrl/CMD + Shift + N**).

A new tab should appear:



Write the mathematical expression from above and save the file as **test.R**.

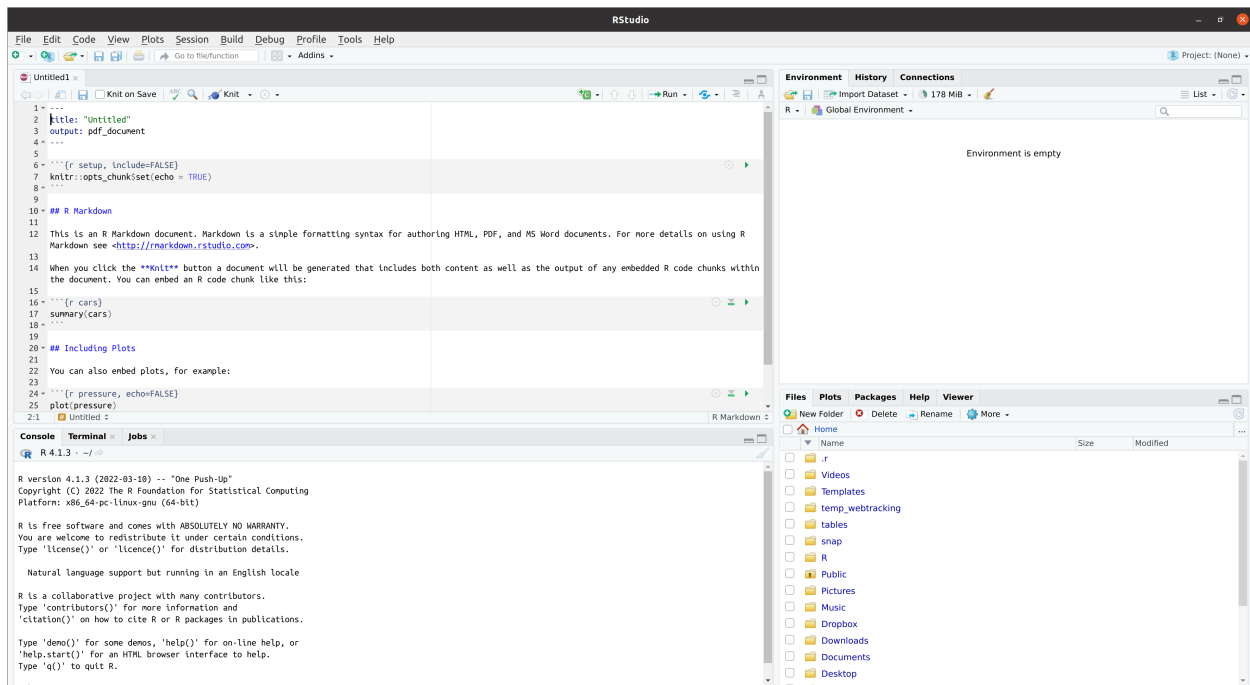
To run this code (by sending it to the console pane below) put cursor on the line of code and press **Ctrl/CMD + Enter**

R Markdown

2. Now let's embed this calculation into R Markdown.

First, create a new R Markdown file by selecting **File -> New File -> R Markdown**.

Fill in the title of the document and your name in the pop-up window (or leave the defaults). Now you should get a default R Markdown document with some initial text and a couple of R code chunks that show its basic functionality:



Try knitting the R Markdown document (i.e. compiling into specified format). You can do that by clicking on **Knit** button in the menu above the editor pane.

Go through the examples provided in the default R Markdown document and compare with the output generated. Do of them make sense? What does not?

Remove all code under the header and leave just one code chunk, which would contain a simple arithmetic calculation from above.

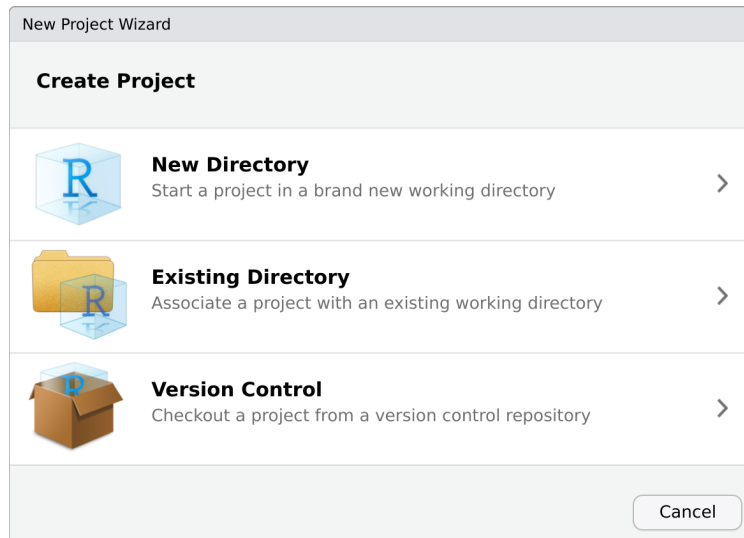
Re-knit the document. Did the compiled document include the output of the arithmetic operation?

RStudio Projects

Individual R scripts and R Markdown files are useful for some specific tasks and documents, but what if we have a project that relies on several scripts that pre-process different datasets or we would like to break-up a larger problem into smaller tasks. RStudio projects provide a way to do just that!

As a good exercise practice I recommend creating a dedicated RStudio project for this module.

New project can be created by selecting **File -> New Project**. We would then be asked either to create a new folder, associate an existing folder or clone a project from a version control repository (ignore this option):



If you have not created a module-specific folder already, you can now create one and initialize it as an RStudio project.

Within your project create a folder titled `code` and put the R script you created above into it.