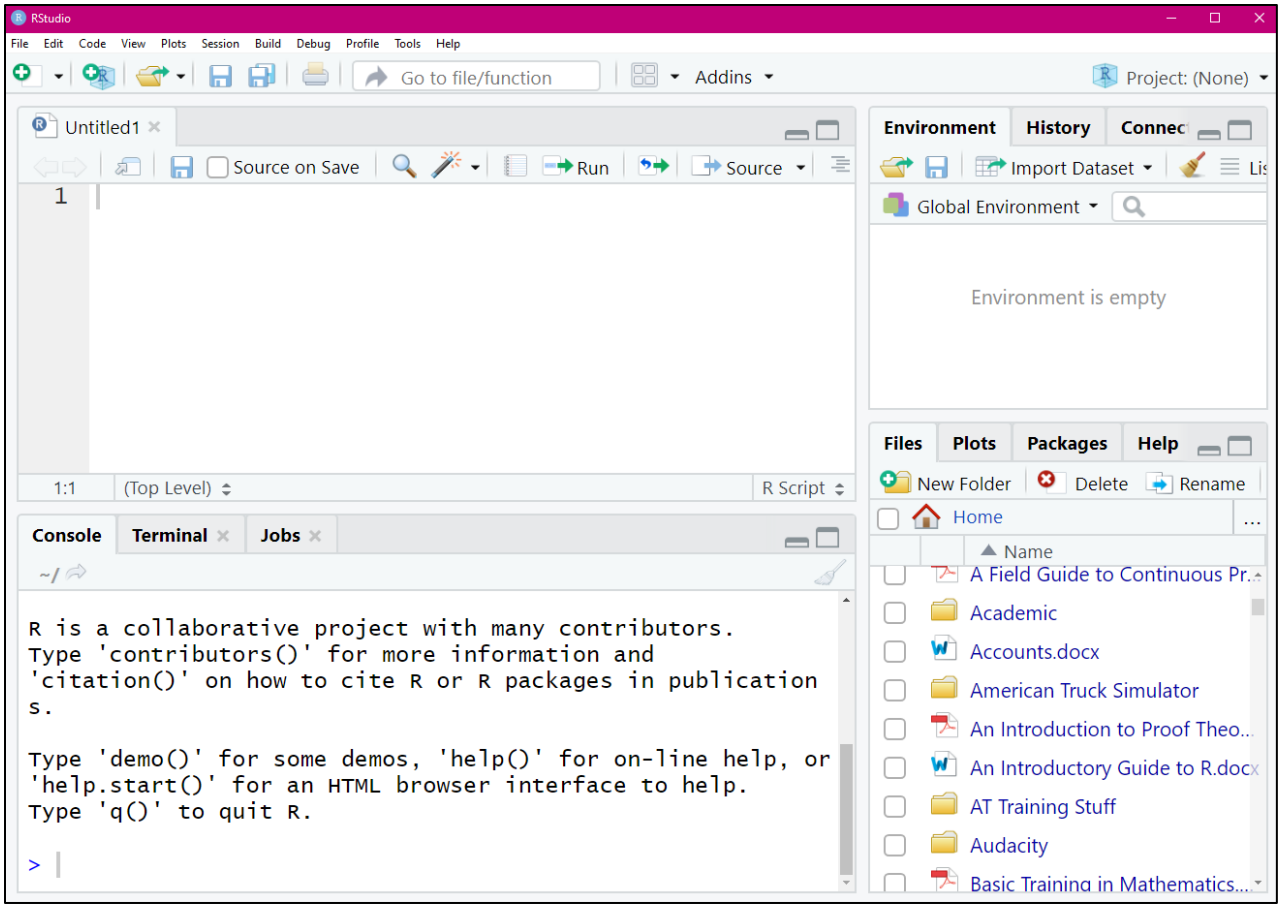
Getting Started with RStudio

This set of instructions will discuss the basics of getting started with RStudio including the interface, panes, and some of the menu options. Please note that this guide is assuming you have already installed RStudio. If you still need to install RStudio and want a guide, check out the "Installing RStudio" pdf.

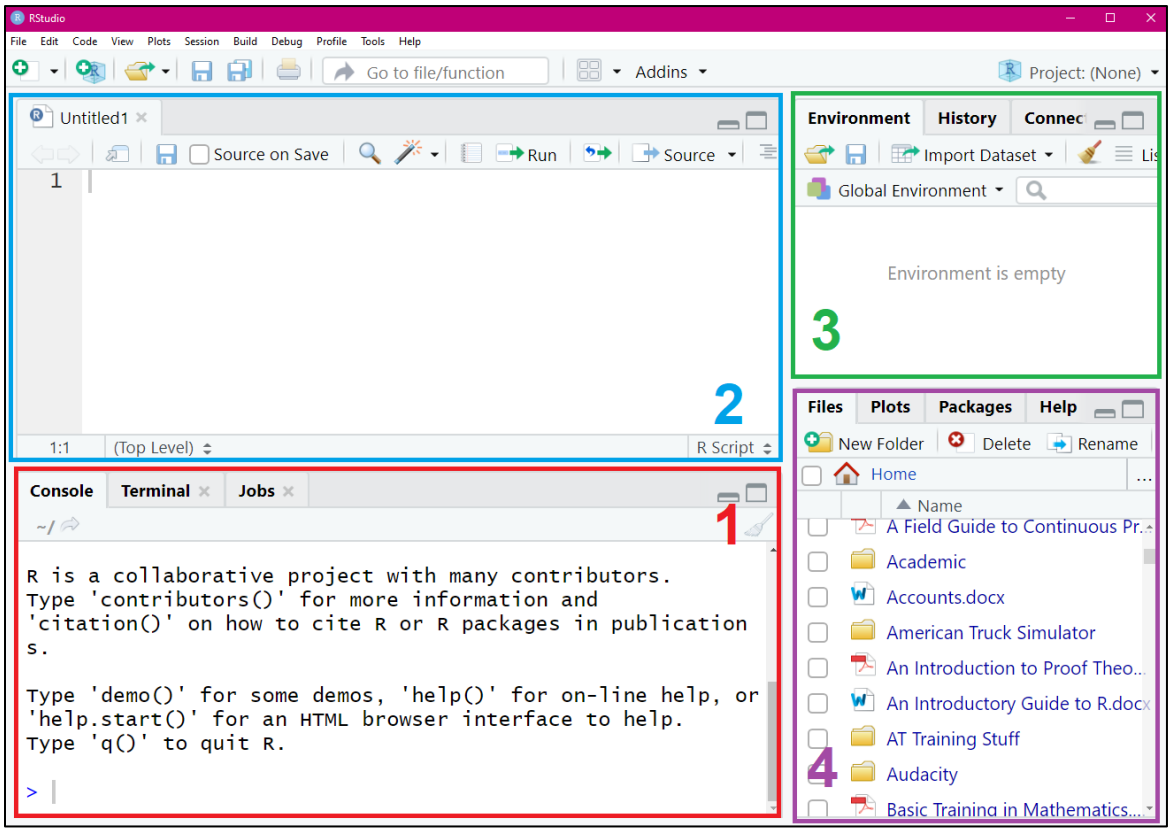
The RStudio Interface

Before we begin, note that you must have R installed before you try to use RStudio. Once you have R installed, RStudio will automatically "pair" with it and open with your R program.

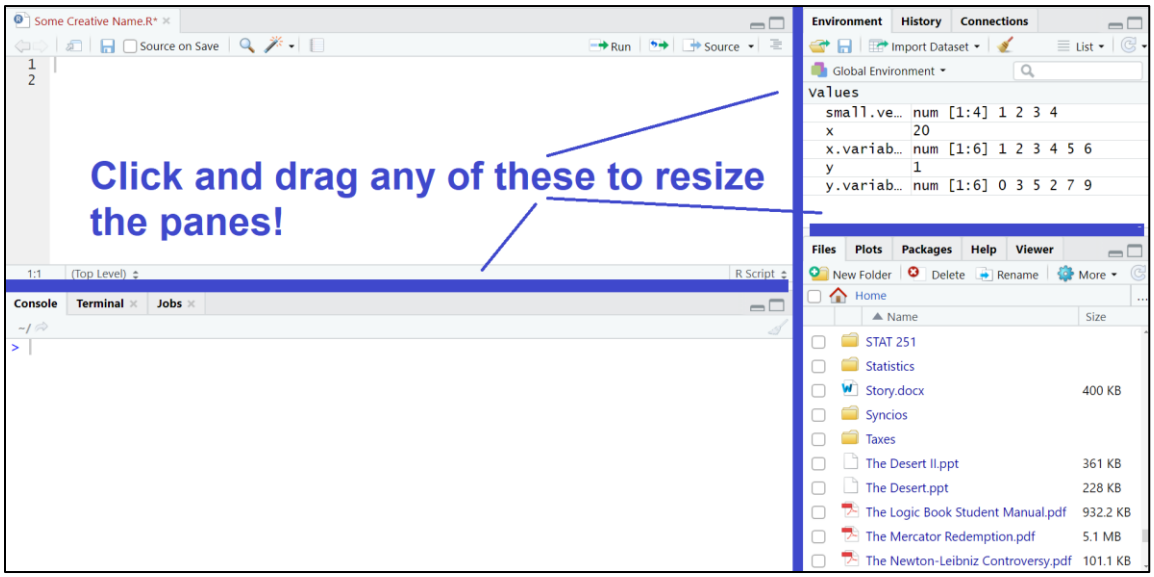
When you open RStudio, this is what you should see (note: I am on a PC; you might see something slightly different on a Mac):



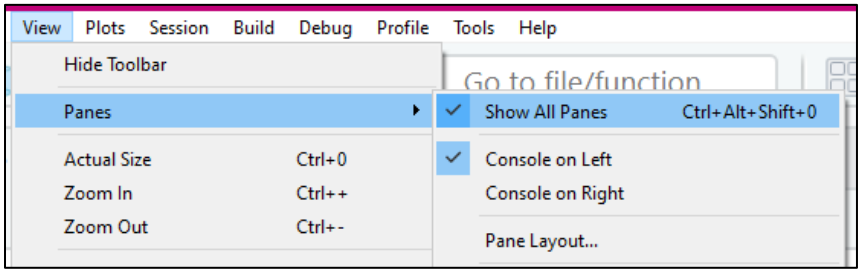
Notice that there are four main “panes” in the RStudio layout:



You can readjust how these panes are displayed. The quickest way to do so is to hover your cursor over any of the borders between the panes. You should be able to click on the border and drag it up/down or left/right to resize the windows.



If you accidentally close a pane or want to reset the layout, go to View → Panes → Show All Panes. This will reset the panes to make them all visible.

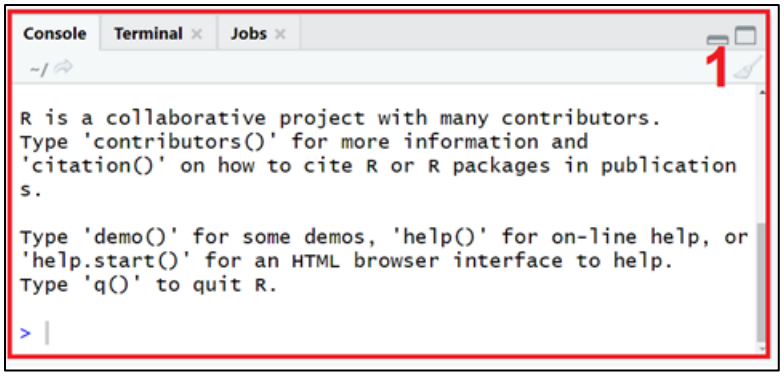


The Four RStudio Panes

Here is more information on the four default panes in RStudio.

Pane 1: The R Console

Pane 1 (boxed in red) is your R Console.



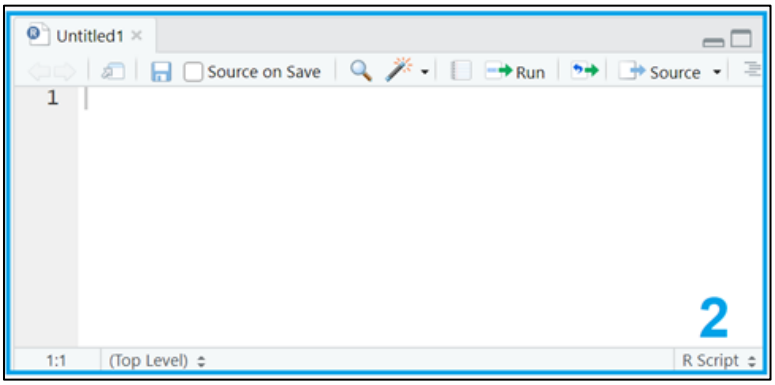
This is the same R Console that you see when you open R.

The menu options can be used to help you open and save R scripts (discussed in further detail below), change your working directory and install packages if necessary (both discussed in the "Installing and Loading R Packages Using R" document).

The console is where all of your R code and resulting R output will be displayed. If you click on the console window to activate it, notice that you can type and the text will show up next to the red cursor. This console works exactly like the R Console in plain old R. If you'd like a guide on all the things you can do in the R Console, please check out the "Getting Started with R" pdf.

Pane 2: R Script

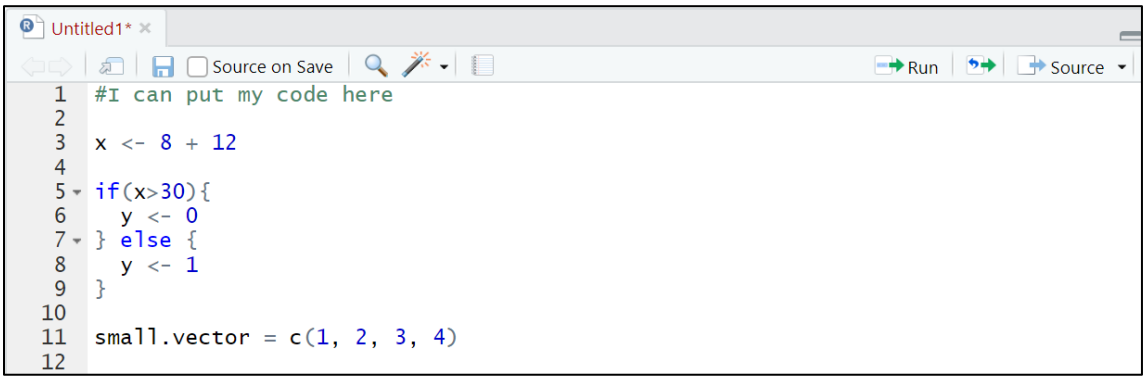
Pane 2 (boxed in blue) is the R script window. An R script is just a plain text file that has R code in it. But it allows you to type your code line by line without having to immediately run it in the R console.



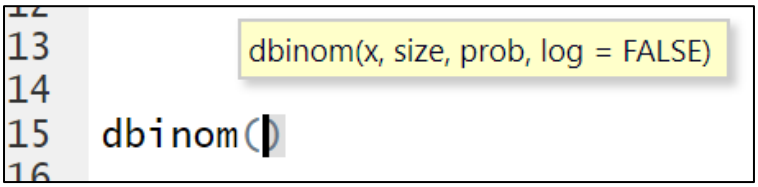
In R, you can open an R script in a separate window as well, but in RStudio, that step is automatically done for you and a blank script (titled "Untitled1") opens for you. You can type in this script the same way you would in an R script using plain old R. However, the script as it is displayed in RStudio has three advantages over how it is displayed in R.

First, RStudio automatically color codes your script. Comments turn green, values and functions turn blue, and parentheses and arithmetic symbols turn gray. It makes the code a bit easier to read.

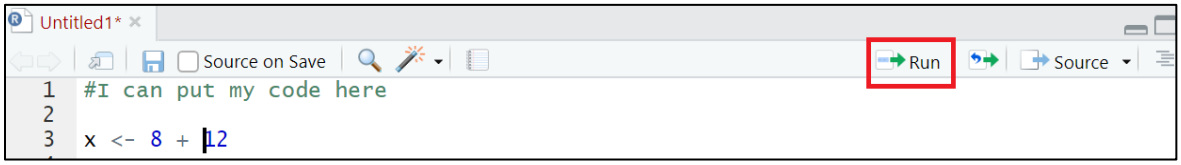
Second, RStudio automatically numbers your lines of code, which may make it easier to troubleshoot longer chunks of code or to refer to a specific line if you're writing up what your code does line by line.



Third, RStudio offers coding "suggestions" based on what you type. Suppose I want to use the dbinom() function. Once I start typing that in the RStudio script pane, it helps me out by telling me the info I need to plug into the function:

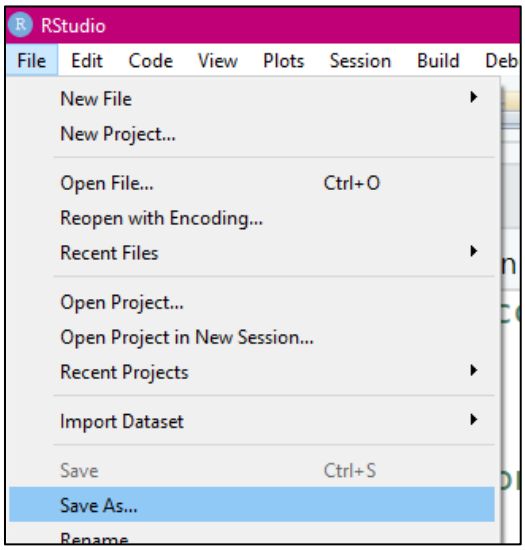


If you want to run a single line of code (that is, send it to the R console), put your cursor somewhere in the line you want to send and click the “Run” button at the top of the script window.

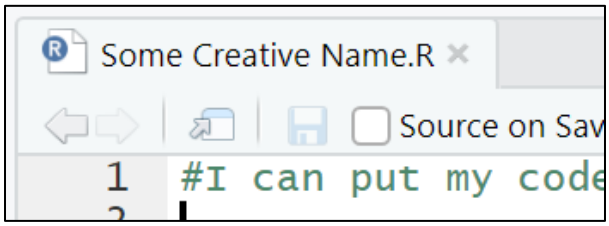


This will send just that line to the R console below the script pane. If you want to run multiple lines, select the lines you want to run and click the “Run” button at the top of the script window.

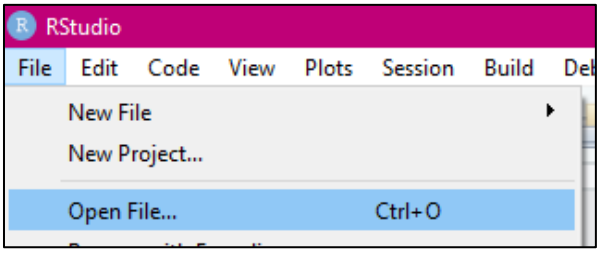
Just like in plain R, you can save your script. To do so, go to File → Safe As…. You can then save the script file anywhere you choose. Notice that it will save as an R file (with a .R extension).



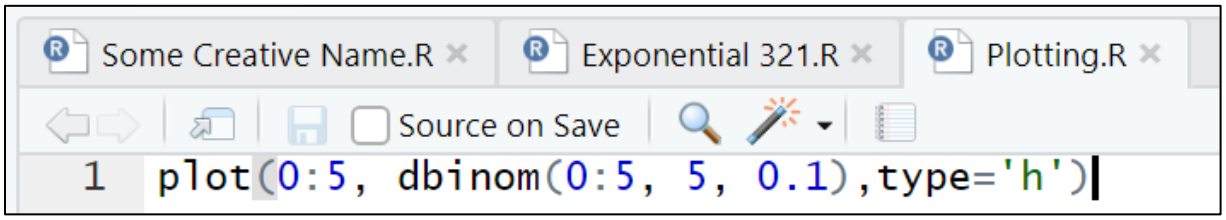
Saved scripts will display with their file name at the top of the script window.



To close/exit a script, click the “x” next to the script name. To open a previously saved script, go to File → Open script… and select the script you wish to open.

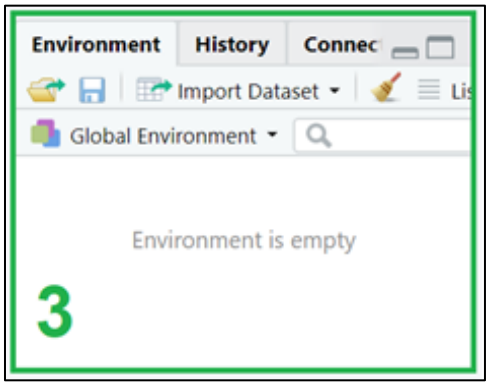


You can have more than one script open at once! If you open multiple scripts at a time, they will show as tabs at the top of the script window. You can toggle through them by clicking the tab with the name of the script you want to work from.

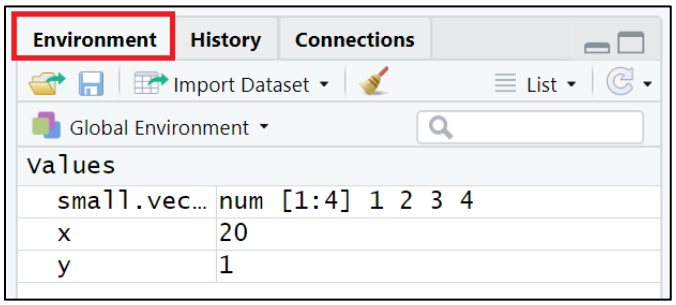


Pane 3: Environment and History

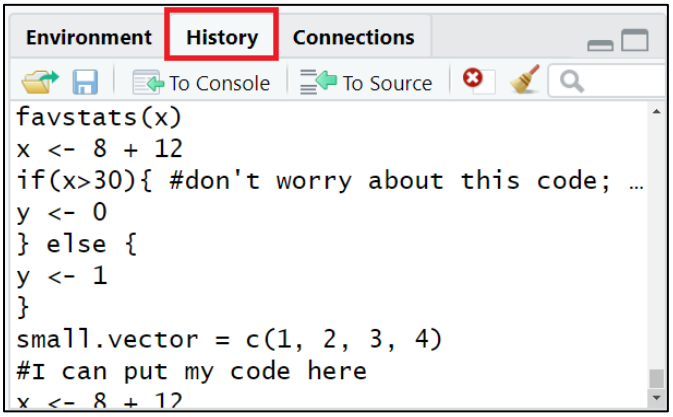
The third pane (boxed in green) contains information on the environment and history of your RStudio sessions.



The “Environment” tab shows all of the declared (named) quantities that are in use in your R session. As an example, if I run the code from my sample script in the above section, I’ve defined the quantities “x”, “y”, and “small.vector.” in that script. You can see those names and their corresponding quantities in the “Environment” tab.



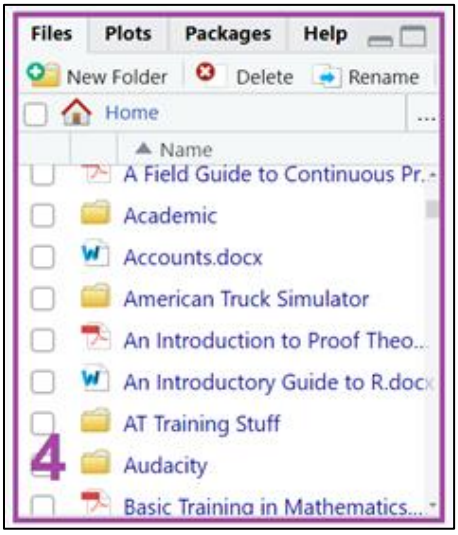
The “History” tab shows the history of the code you’ve sent to the R console.



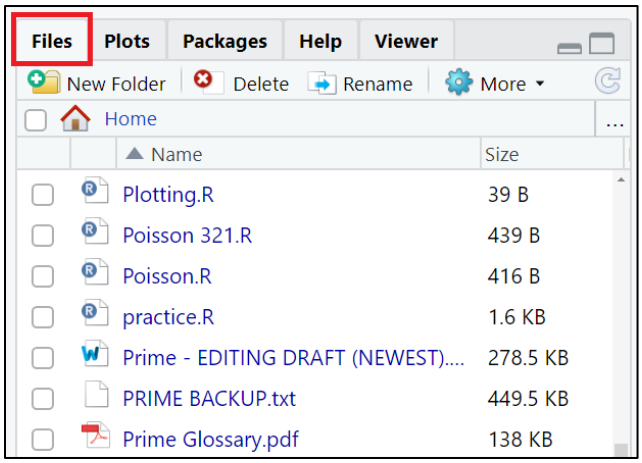
You probably will not need to rely on either of these tabs very much! They are just there to give some more information about your R/RStudio session. However, the “Import Dataset” button under the “Environment” tab can be useful for getting external data files into your RStudio environment. More information on how to import data (if you ever need to do so) is in the “Importing Data Using RStudio” pdf.

Pane 4: Files, Plots, Packages, and Help

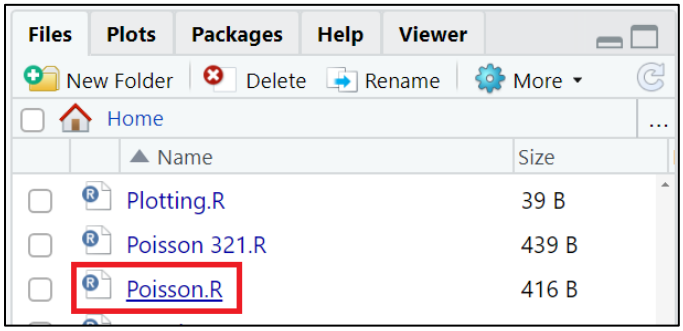
The fourth pane (boxed in purple) contains tabs for files, plots, packages, and getting help in R.



The “Files” tab shows the files in your working directory. My default working directory is my “Documents” folder, so this tab is showing the files in that folder.

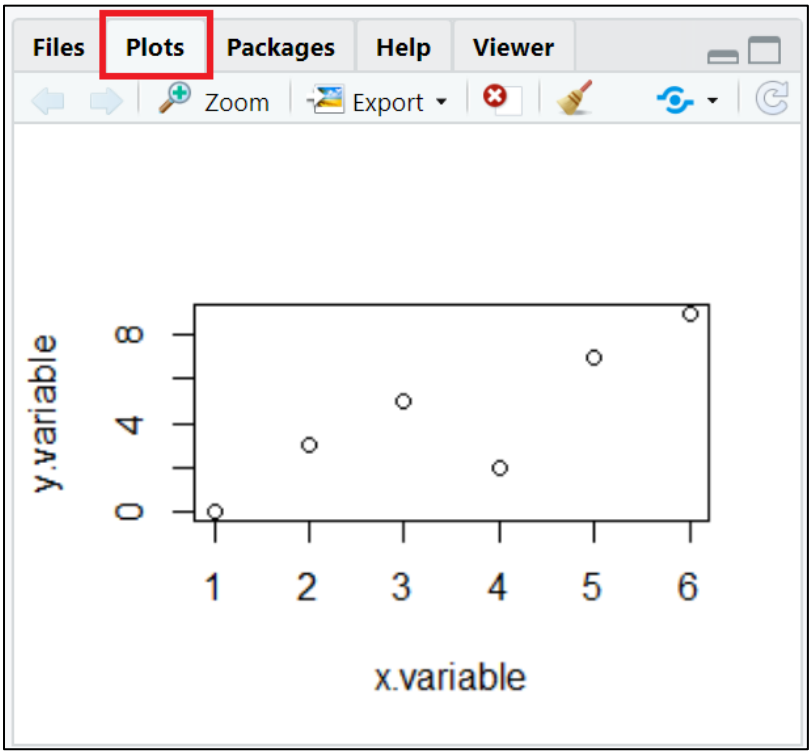


More information about the working directory can be found in the “Importing Data Using RStudio” pdf, but this tab be useful if you are saving your R scripts in your working directory, as you can just click on the name of a script to open it in the script pane. For example, I could open the “Poisson.R” script just by clicking its name.

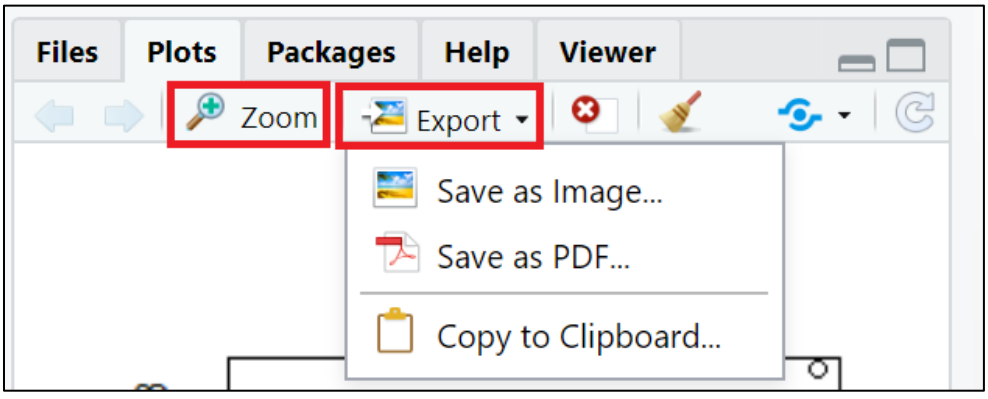


The “Plots” tab will display any plots/graphics that you create in R. It will be blank/empty until a graphic is created. As an example, let’s just see what a scatterplot looks like in this tab:

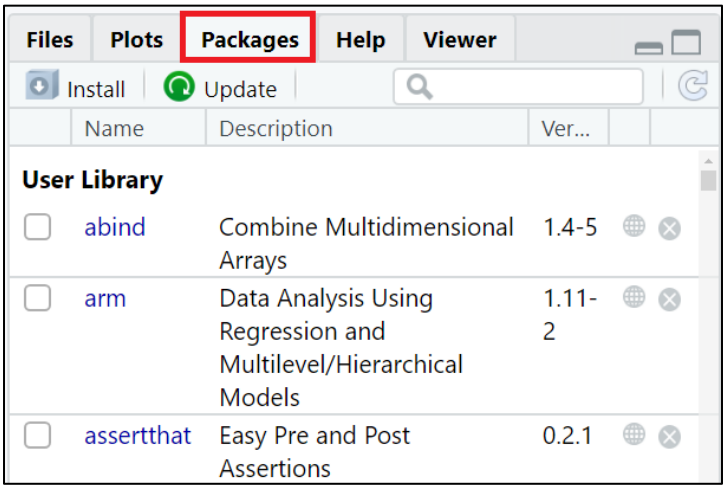
```
x.variable = c(1, 2, 3, 4, 5, 6)
y.variable = c(0, 3, 5, 2, 7, 9)
plot(x.variable,y.variable)
```



It may look a little squished depending on how big your pane is, but you can click the “Zoom” button to open the plot at its “full size” in a different window. You can also save the plot as an image or pdf using the “Export” button.



The “Packages” tab displays all of the installed R packages on your computer. It can be useful if you need to install additional packages or load a previously installed package into your R session. Information on packages can be found in the “Installing and Loading Packages Using RStudio” pdf.



Finally, the “Help” tab is where any additional documentation will be displayed for a given function. There may be times when you need help or documentation on a function you are using in R. To access R’s “help” library, type the following into the R console and hit “enter”:

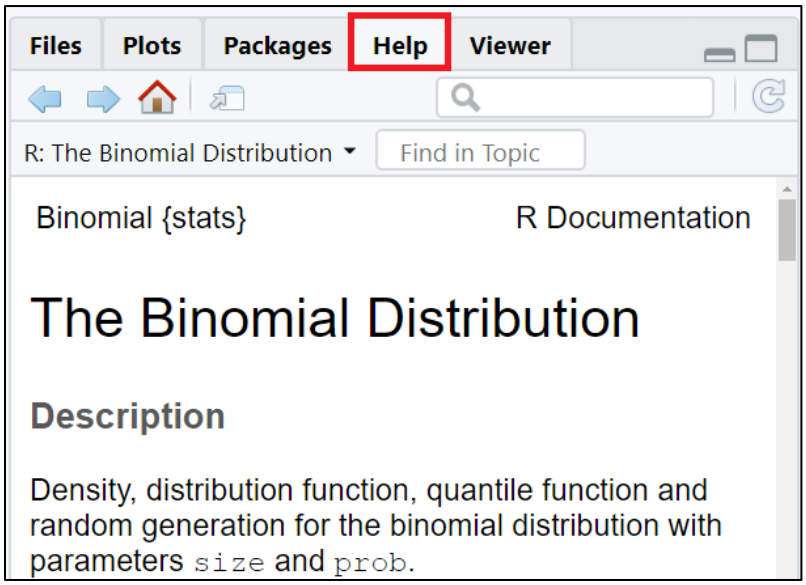
```
?function
```

where function is the name of the function you want to look up. This will open the online help library in this “Help” tab.

As an example, suppose a course I’m taking requires me to use the dbinom() function. If I want some more information on this function or forget what needs to be entered in the function, I can get some help.

```
> ?dbinom
starting httpd help server ... done
>
```

And in the “Help” tab, the following has appeared:



This gives me more info on what the function is, what it does, and what values I need to enter into it to use it correctly.