

Machine learning and data science with R



Getting started with R and RStudio

- R is open source and freely available.
- R is available for Windows, Mac and Linux operating systems.
- R has an extensive and coherent set of tools for statistical analysis.
- R has an extensive and highly flexible graphical facility capable of producing publication quality figures.
- R has an expanding set of freely available ‘packages’ to extend R’s capabilities.
- R has an extensive support network with numerous online and freely available documents.

- A collection of R functions that can be shared among users is called a **package**. Free packages exist for each of the machine learning algorithms
- If you are interested in the breadth of R packages, you can view a list at **Comprehensive R Archive Network (CRAN)**, a collection of web sites located around the world to provide the most up-to-date versions of R software and packages
- The **Packages** a total 6,779 packages were available

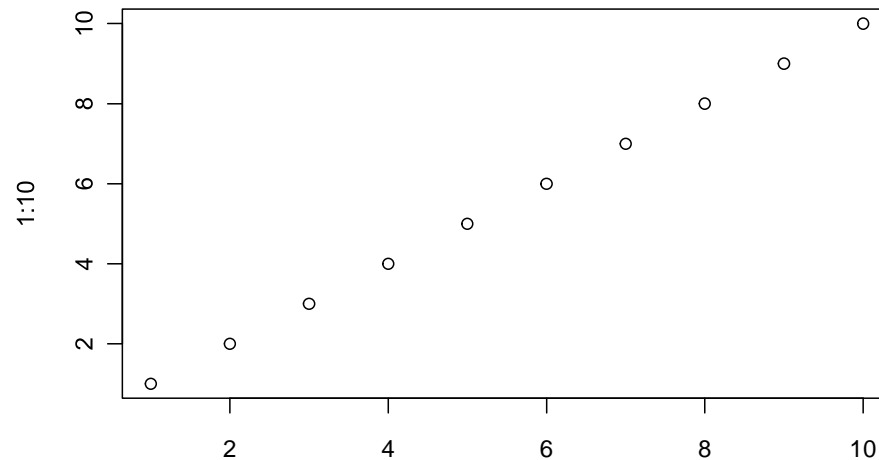
Installing R

- To get up and running the first thing you need to do is install R. R is freely available for Windows, Mac and Linux operating systems from the [Comprehensive R Archive Network \(CRAN\) website](#). For Windows and Mac users we suggest you download and install the pre-compiled binary versions.

Testing R

• Whichever operating system you're using, once you have installed R you need to check its working properly. The easiest way to do this is to start R by double clicking on the R icon (Windows or Mac) or by typing `R` into the Console (Linux). You should see the R Console and you should be able to type R commands into the Console after the command prompt `>`. Try typing the following R code and then press enter (don't worry if you don't understand this - we're just checking if R works)

```
> plot(1:10)
```



Installing RStudio

- Whilst it's eminently possible to just use the base installation of R (many people do), we will be using a popular **I**ntegrated **D**evelopment **E**nvironment (IDE) called RStudio.
- RStudio can be thought of as an add-on to R which provides a more user-friendly interface, incorporating the R Console, a script editor and other useful functionality (like R markdown and Git Hub integration). You can find more information about [RStudio here](#)

Testing RStudio

The screenshot displays the RStudio desktop environment. The top toolbar includes icons for file operations and a search bar. The main interface is divided into several panes:

- Console:** Shows the R startup message: "R version 3.6.2 (2019-12-12) -- 'Dark and Stormy Night' Copyright (C) 2019 The R Foundation for Statistical Computing Platform: x86_64-apple-darwin15.6.0 (64-bit)". It also displays the R license notice and help instructions.
- Environment:** A table showing the current environment, which is empty.
- Files:** A file browser showing the contents of the user's home directory.

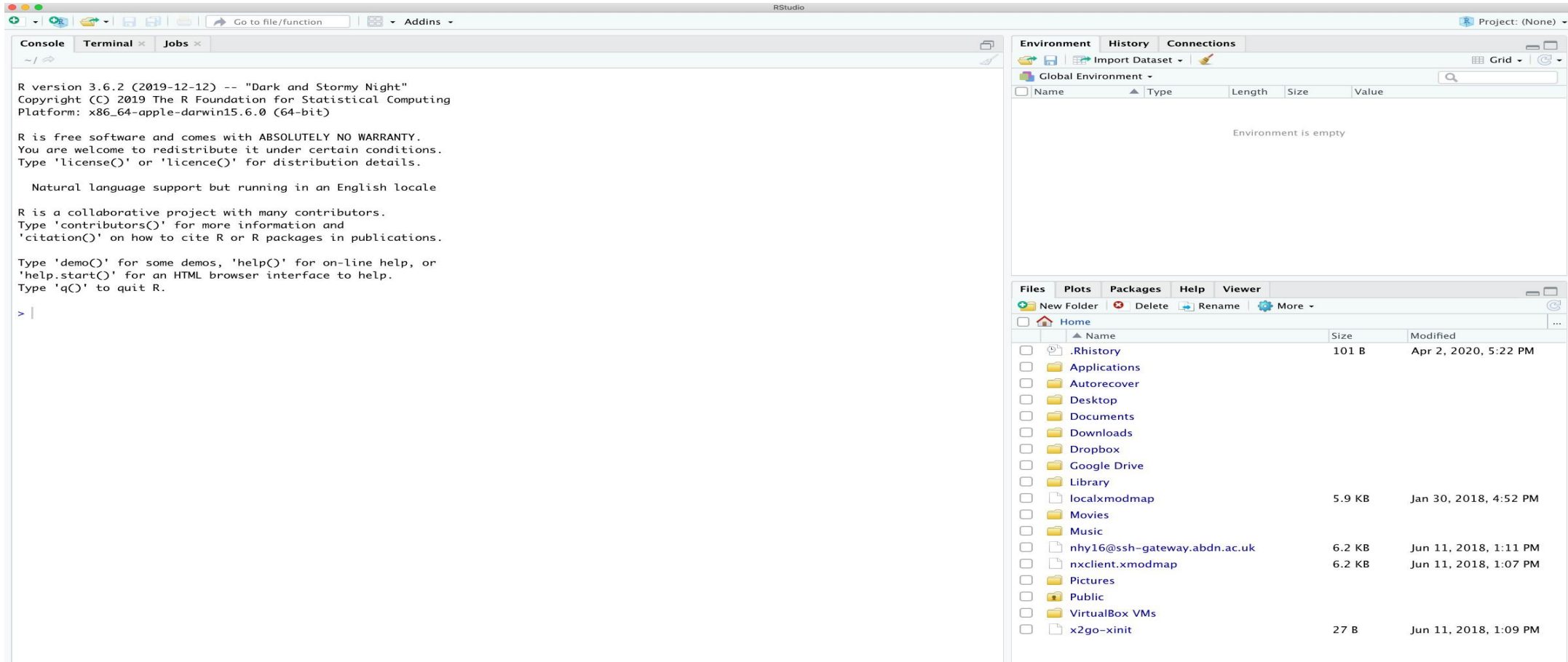
Environment Pane:

Name	Type	Length	Size	Value
Environment is empty				

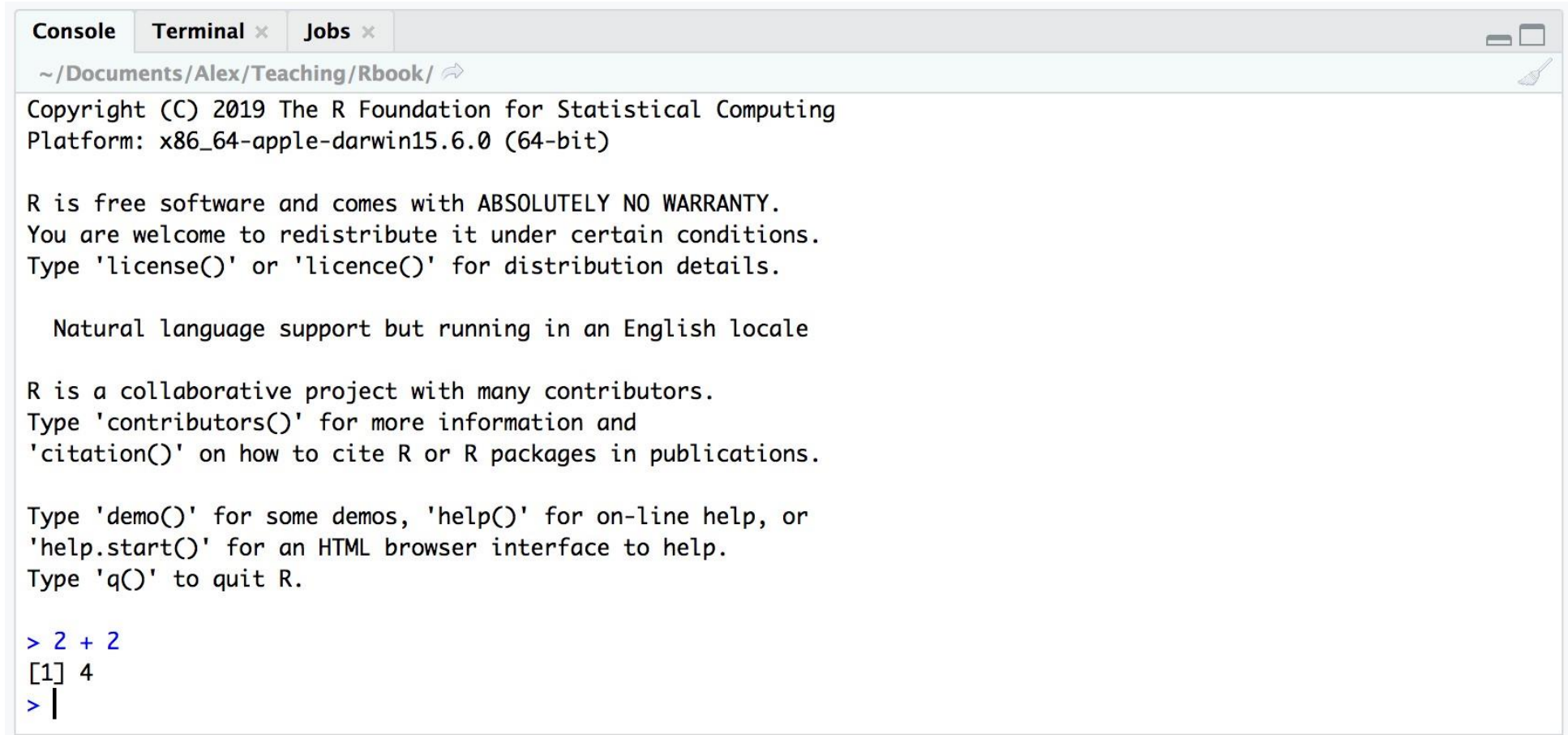
Files Pane:

Name	Size	Modified
.Rhistory	101 B	Apr 2, 2020, 5:22 PM
Applications		
Autorecover		
Desktop		
Documents		
Downloads		
Dropbox		
Google Drive		
Library		
localxmodmap	5.9 KB	Jan 30, 2018, 4:52 PM
Movies		
Music		
nhy16@ssh-gateway.abdn.ac.uk	6.2 KB	Jun 11, 2018, 1:11 PM
nxclient.xmodmap	6.2 KB	Jun 11, 2018, 1:07 PM
Pictures		
Public		
VirtualBox VMs		
x2go-xinit	27 B	Jun 11, 2018, 1:09 PM

The large window (aka pane) on the left is the **Console** window. The window on the top right is the **Environment / History / Connections** pane and the bottom right window is the **Files / Plots / Packages / Help / Viewer** window



Console



```
Console Terminal x Jobs x
~/Documents/Alex/Teaching/Rbook/
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

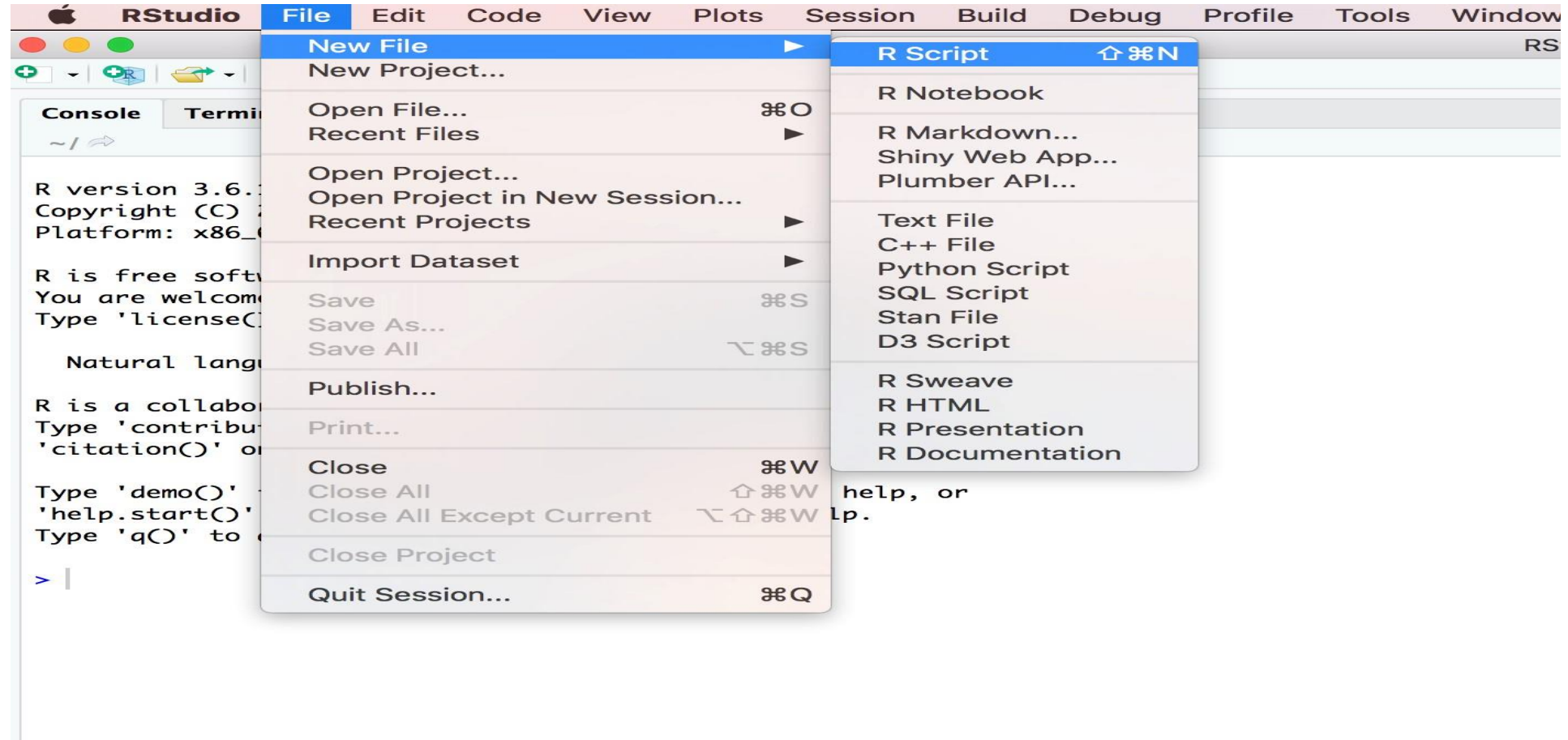
Natural language support but running in an English locale

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.

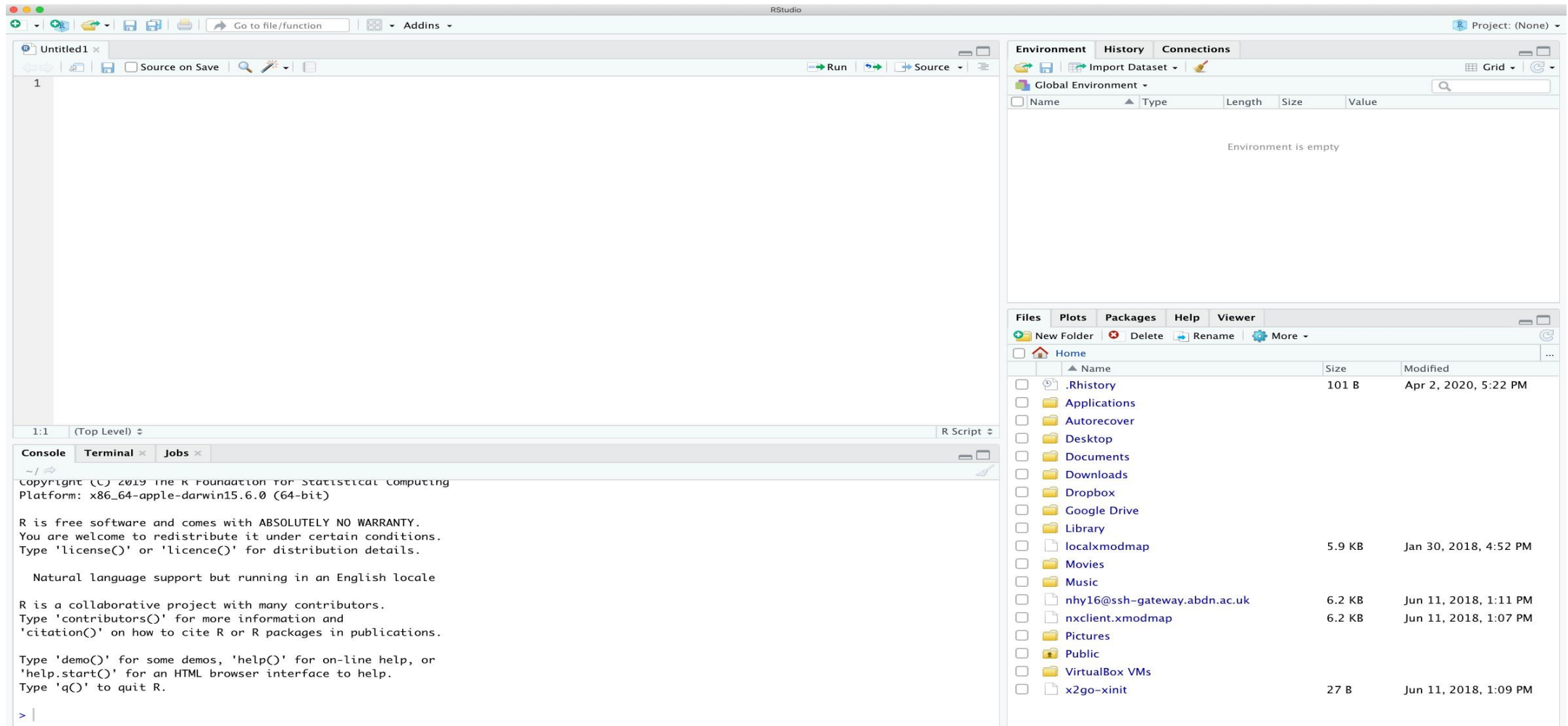
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

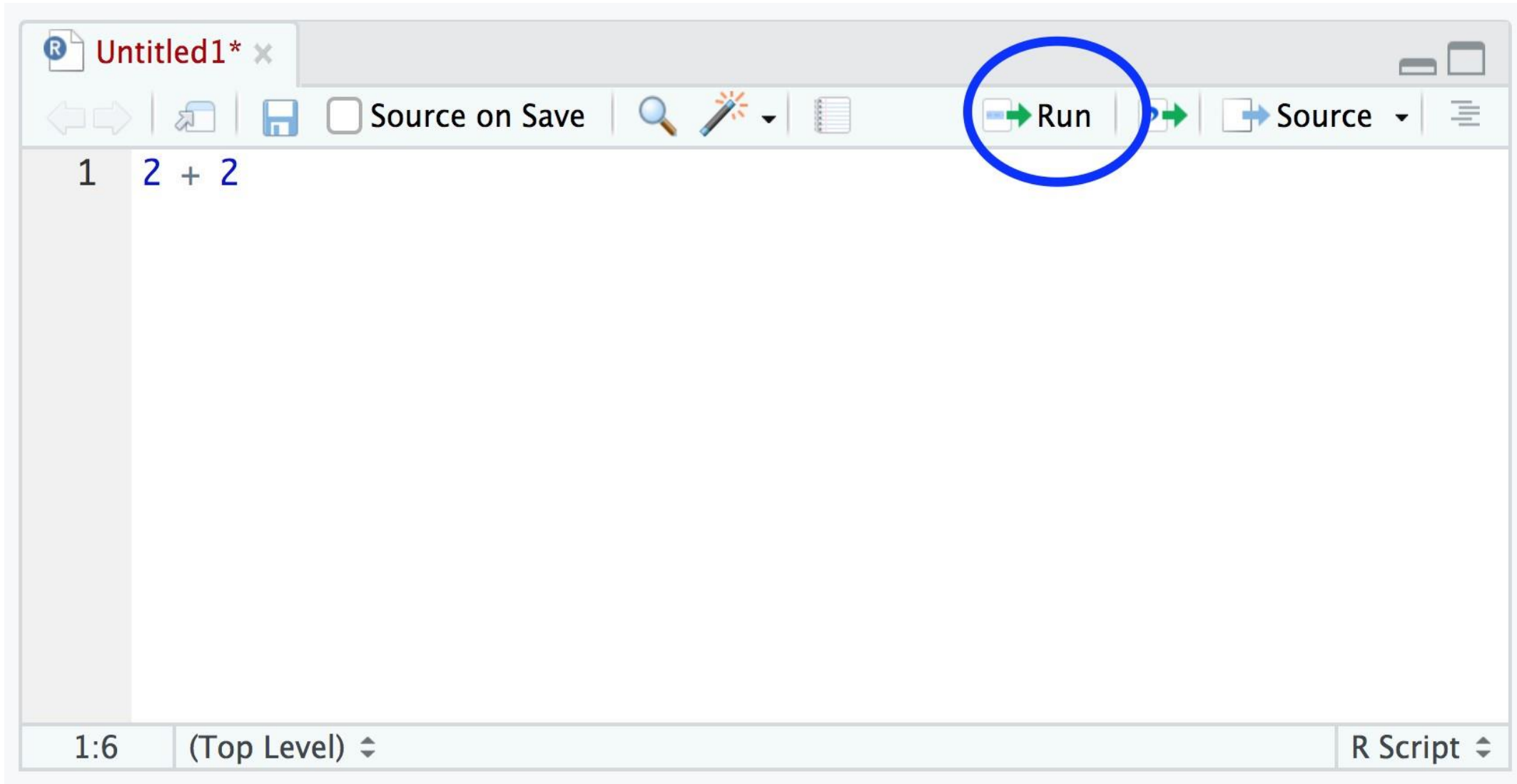
> 2 + 2
[1] 4
> |
```

An R script is just a plain text file with a **R** file extension which contains your lines of R code. These lines of code are then sourced into the R Console line by line. To create a new R script click on the 'File' menu then select New File → R Script.



Notice that you have a new window (called the Source pane) in the top left of RStudio and the Console is now in the bottom left position. The new window is a script editor and where you will write your code.





Environment/History/Connections

- **The ‘Environment’** tab displays all the objects you have created in the current (global) environment. These objects can be things like data you have imported or functions you have written. Objects can be displayed as a List
- **The ‘History’** tab contains a list of all the commands you have entered into the R Console. You can search back through your history for the line of code you have forgotten, send selected code back to the Console or Source window. We usually never use this as we always refer back to our R script.
- **The ‘Connections’** tab allows you to connect to various data sources such as external databases.

Files/Plots/Packages/Help/Viewer

- **The ‘Files’** tab lists all external files and directories in the current working directory on your computer. It works like file explorer (Windows) or Finder (Mac). You can open, copy, rename, move and delete files listed in the window.
- **The ‘Plots’** tab is where all the plots you create in R are displayed (unless you tell R otherwise). You can ‘zoom’ into the plots to make them larger using the magnifying glass button, and scroll back through previously created plots using the arrow buttons. There is also the option of exporting plots to an external file using the ‘Export’ drop down menu. Plots can be exported in various file formats such as jpeg, png, pdf, tiff or copied to the clipboard .
- **The ‘Packages’** tab lists all of the packages that you have installed on your computer. You can also install new packages and update existing packages by clicking on the ‘Install’ and ‘Update’ buttons respectively.
- **The ‘Help’** tab displays the R help documentation for any function.
- **The ‘Viewer’** tab displays local web content such as web graphics generated by some packages.

CRAN packages

- To install a package from CRAN you can use the **install.packages()** function. For example if you want to install the remotes package enter the following code into the Console window of RStudio (note: you will need a working internet connection to do this)

- install.packages('remotes', dependencies = TRUE)**

- You may be asked to select a CRAN mirror, just select '0-cloud' or a mirror near to your location. The *dependencies = TRUE* argument ensures that additional packages that are required will also be installed.

- It's good practice to occasionally update your previously installed packages to get access to new functionality and bug fixes. To update CRAN packages you can use the **update.packages()** function (you will need a working internet connection for this)

- update.packages(ask = FALSE)**

- The **ask = FALSE** argument avoids having to confirm every package download which can be a pain if you have many packages installed.

Using packages

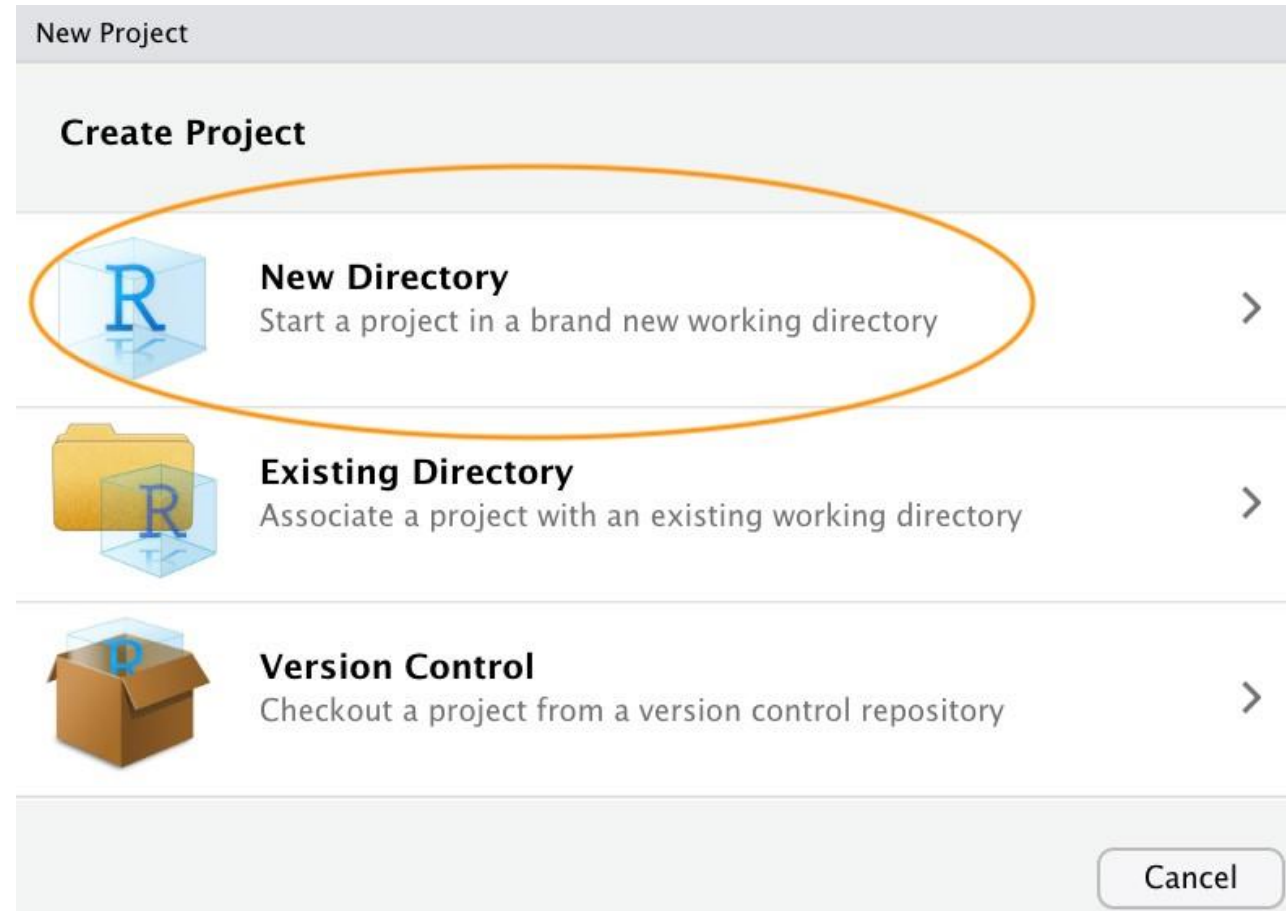
- Once you have installed a package onto your computer it is not immediately available for you to use. To use a package you first need to load the package by using the **library()** function. For example, to load the **remotes** package you previously installed

- **library(remotes)**

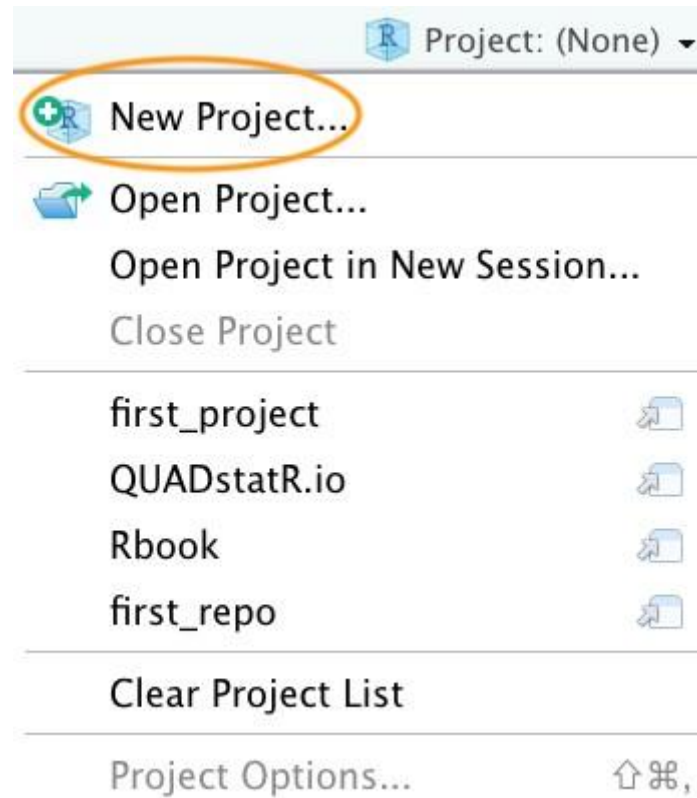
- The **library()** function will also load any additional packages required and may print out additional package information. It is important to realise that every time you start a new R session (or restore a previously saved session) you need to load the packages you will be using. We tend to put all our **library()** statements required for our analysis near the top of our R scripts to make them easily accessible and easy to add to as our code develops.

Projects in Rstudio

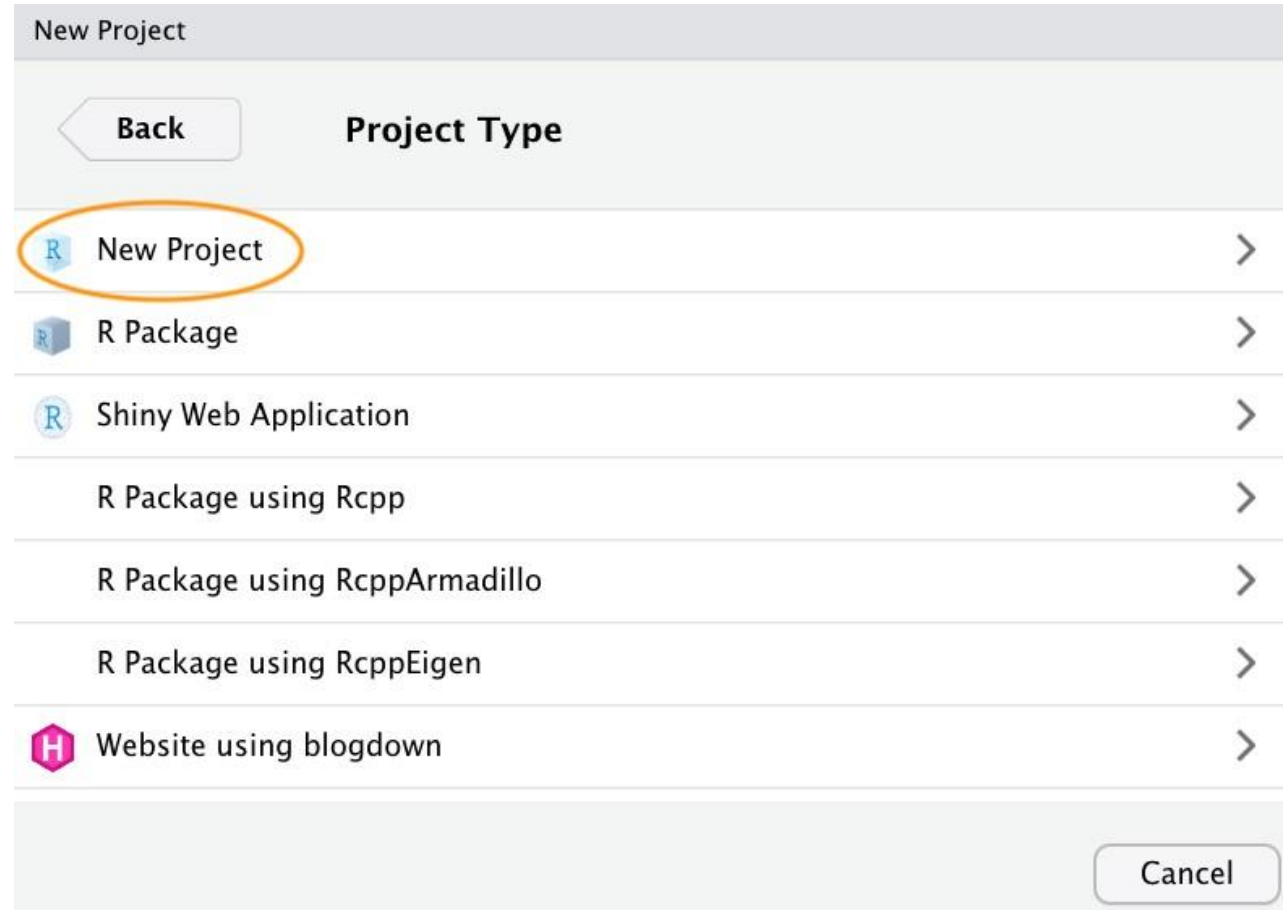
To create a project, open RStudio and select **File -> New Project...** from the menu. You can create either an entirely new project, a project from an existing directory or a version controlled project (see the GitHub Chapter for further details about this). In this Chapter we will create a project in a new directory.



You can also create a new project by clicking on the 'Project' button in the top right of RStudio and selecting 'New Project...'



In the next window select 'New Project'.



Now enter the name of the directory you want to create in the ‘Directory name:’ field (we’ll call it **first_project** for this Chapter). If you want to change the location of the directory on your computer click the ‘Browse...’ button and navigate to where you would like to create the directory. We always tick the ‘Open in new session’ box as well. Finally, hit the ‘Create Project’ to create the new project.

New Project

Back Create New Project

Directory name:
first_project

Create project as subdirectory of:
~/Documents/Alex/Teaching Browse...

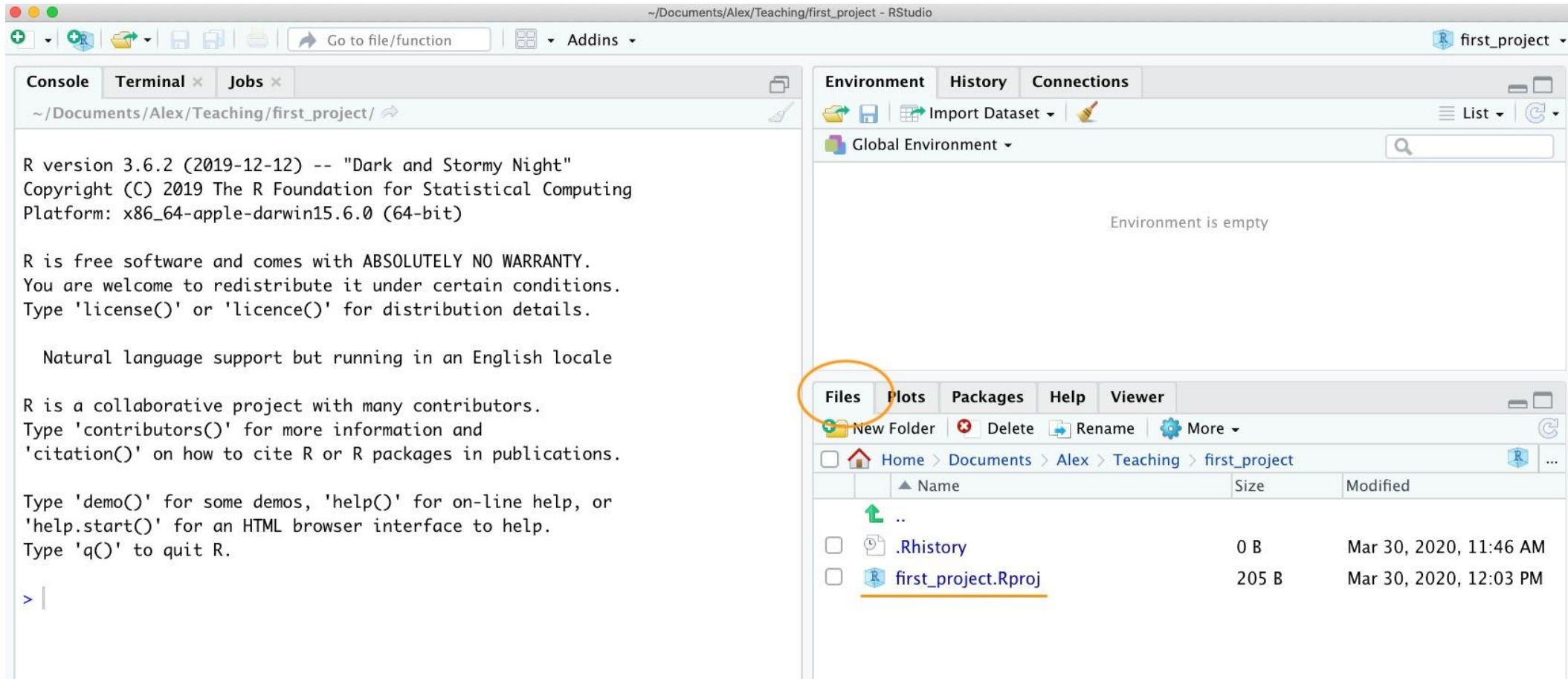
Create a git repository

Use packrat with this project

Open in new session

Create Project Cancel

Once your new project has been created you will now have a new folder on your computer that contains an RStudio project file called `first_project.Rproj`. This `.Rproj` file contains various project options (but you shouldn't really interact with it) and can also be used as a shortcut for opening the project directly from the file system (just double click on it). You can check this out in the 'Files' tab in RStudio (or in Finder if you're on a Mac or File Explorer in Windows).

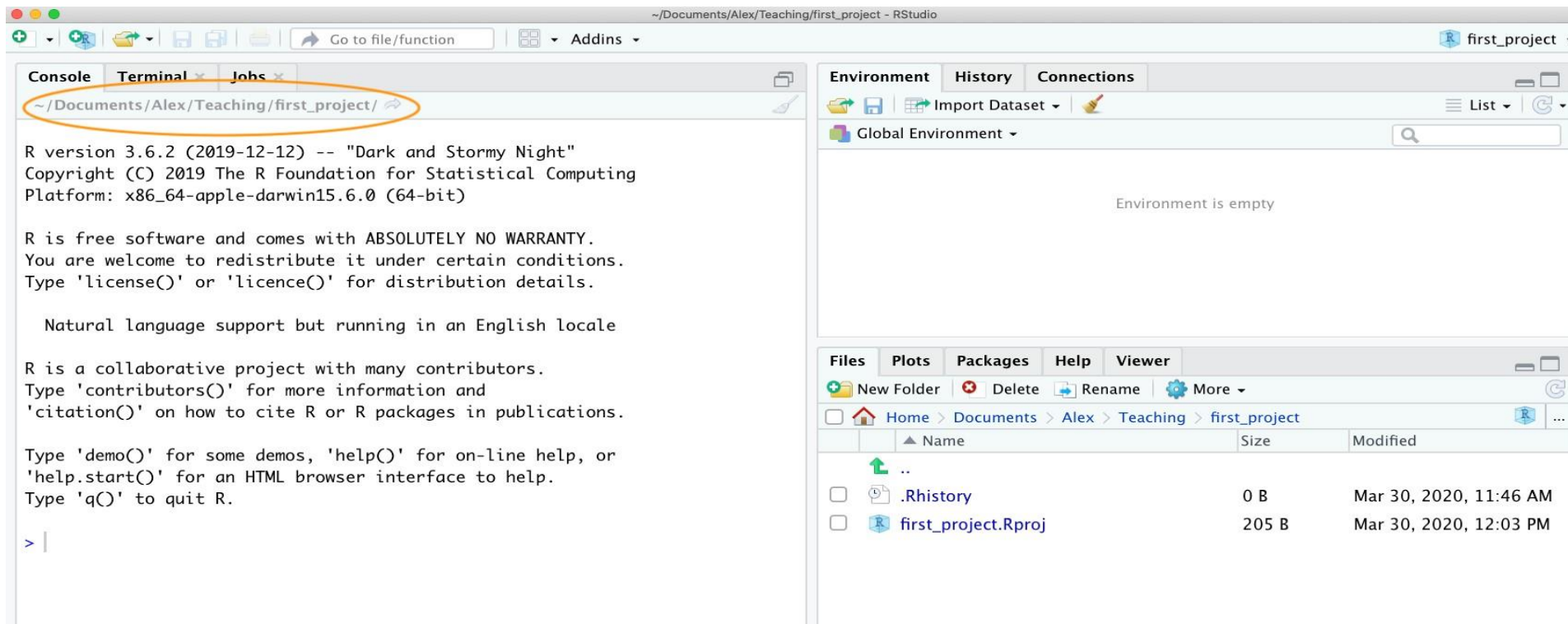


The last thing we suggest you do is select Tools -> Project Options... from the menu. Click on the 'General' tab on the left hand side and then change the values for 'Restore .RData into workspace at startup' and 'Save workspace to .RData on exit' from 'Default' to 'No'. This ensures that every time you open your project you start with a clean R session. You don't have to do this (many people don't) but we prefer to start with a completely clean workspace whenever we open our projects to avoid any potential conflicts with things we have done in previous sessions. The downside to this is that you will need to rerun your R code every time you open you project.

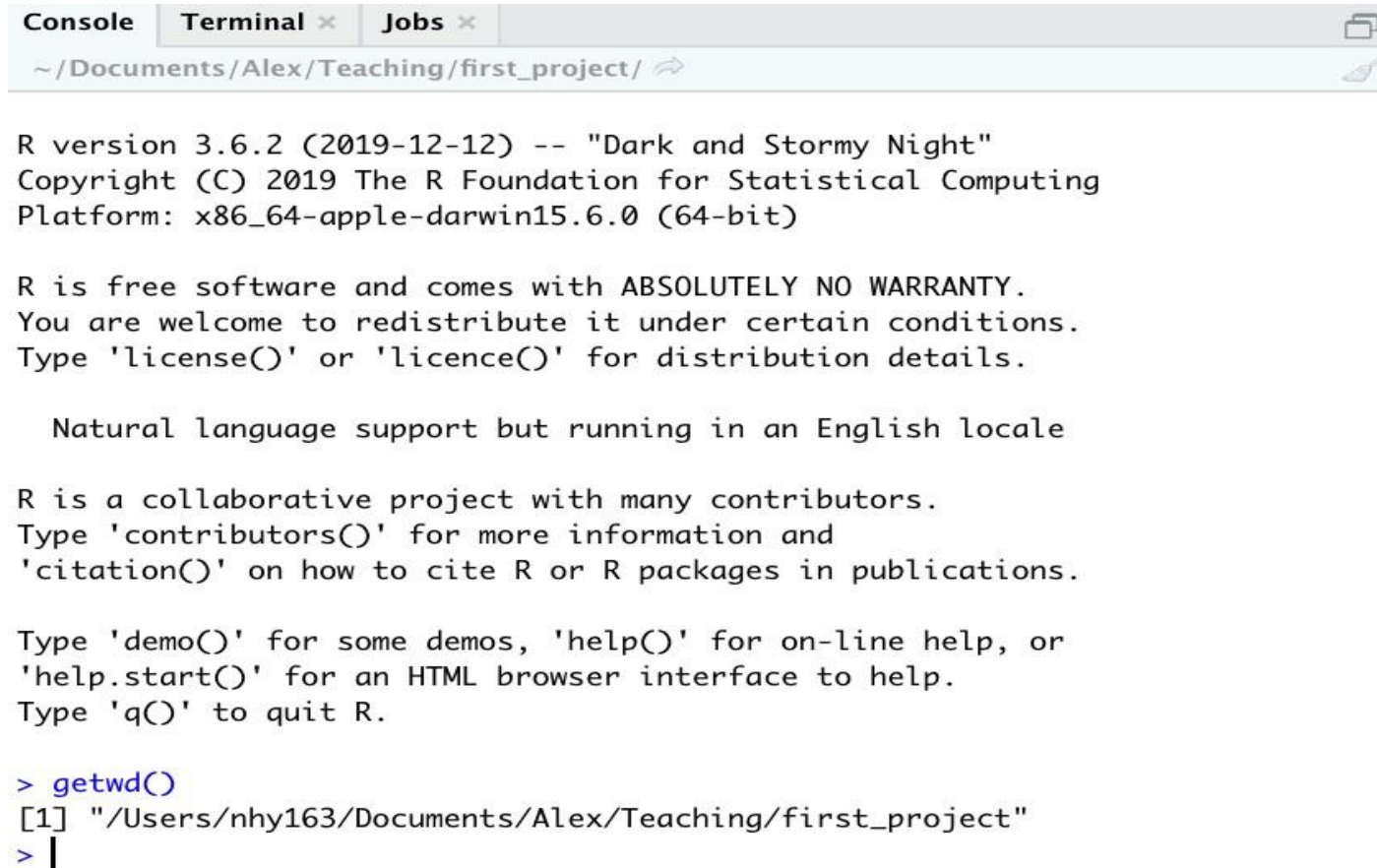


Working directories

The working directory is the default location where R will look for files you want to load and where it will put any files you save. One of the great things about using RStudio Projects is that when you open a project it will automatically set your working directory to the appropriate location. You can check the file path of your working directory by looking at bar at the top of the Console pane.



You can also use the `getwd()` function in the Console which returns the file path of the current working directory.



```
Console Terminal x Jobs x
~/Documents/Alex/Teaching/first_project/

R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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

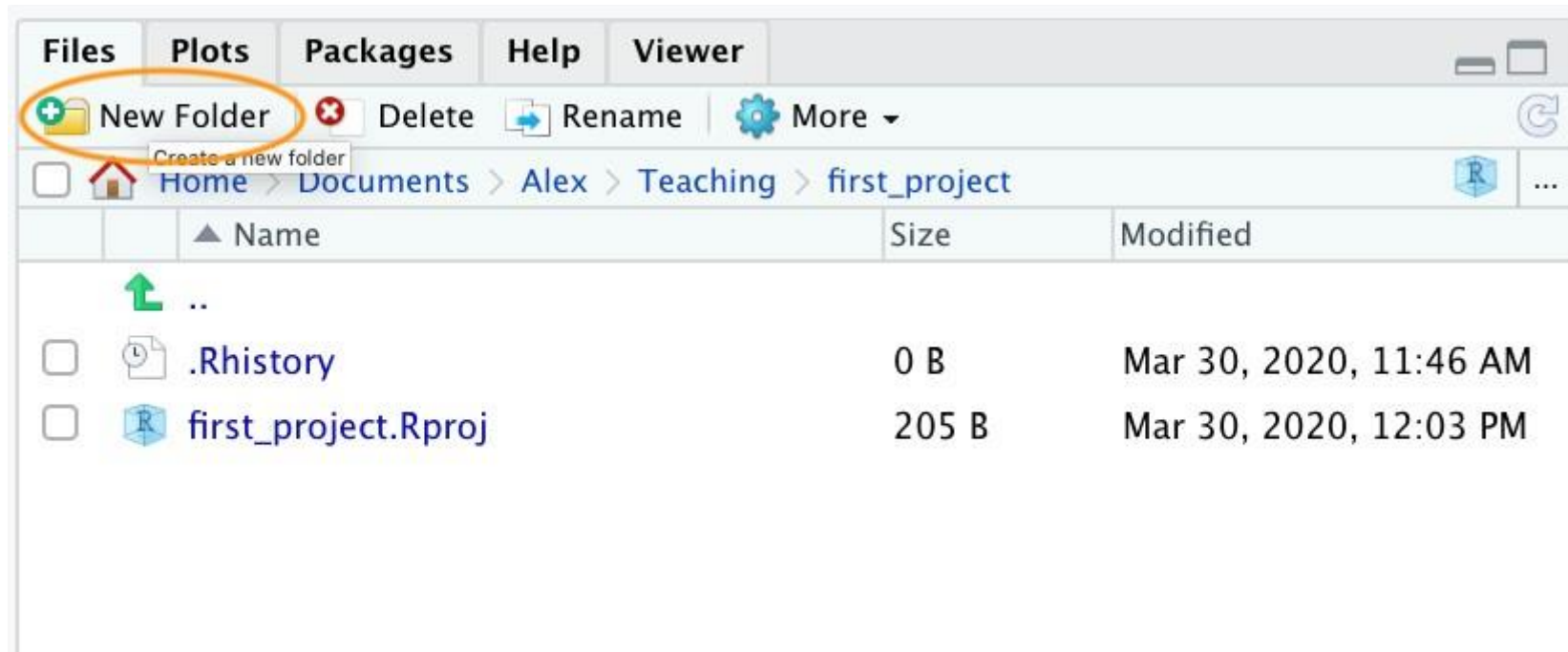
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> getwd()
[1] "/Users/nhy163/Documents/Alex/Teaching/first_project"
> |
```


FILE NAMES

You can create your directory structure using Windows Explorer (or Finder on a Mac) or within RStudio by clicking on the 'New folder' button in the 'Files' pane.



Thank you