

A Step-by-Step Guide to Writing R Scripts

1 Start a new R script

The first step whenever you are working on a new project is to start a new R Script. This ensures that you are not confusing any of your work and gives you a clean slate with which to perform your analysis. Once you open a new file, immediately save it.

1.1 To open a new R script

- File → New File → R Script

1.2 To save that R Script:

- File → Save As → (select the location to save your file and give it a name)

2 Describe the script and identify yourself

The second step that you should always take is to write a little about the R Script and yourself at the top of the R Script. Let the reader (and yourself) know who you are and what you are performing in this R Script. Of course, no one would ever see this R Script unless you sent it to them – but there will be times when you share your analysis with others (and certainly with me).

Remember, because you are writing a note in the R Script, you need to use the hashtag (#) before everything you write. You need at least one hashtag on every line that you are writing a note on. More hashtags are fine and sometimes people use more simply in order to better organize the script (like to delineate a new section), but you only need one hashtag per line, no more.

For example:

```
# Name: Rob Williams  
# Title: Introduction to R  
# Course: Poli 281  
# Created: January 10, 2019  
# Updated: January 10, 2019  
# Email: jrw@live.unc.edu
```

3 Clear your workspace

Next, you need to clear out your workspace. R may still have objects, data, or functions that you made in a previous session. Because you are performing a new analysis, you want the workspace completely clear. This will allow you to work with a fresh start. Simply copy and paste the following code into your R Script. This should *always* come before any other coding.

```
rm(list = ls())
```

4 Run your code

Once you type code in (like the code you use to clear your workspace), you will need to ‘run’ it. R does not know, just because you type something in the script, that you want to run it. You need to tell it to run something.

To run your code, you have a couple of options.

1. Place your cursor anywhere in the line of code that you want to run. Before the code, after the code, in the middle - it doesn't matter. Then either:
 - a. Run the code by pressing the ‘run’ button () on the top right corner of your R Script
 - b. Run the code by pressing **Command+Enter** at the same time (if you have a Mac) or **Control+Enter** (PC or a Mac)
2. Highlight the portion of the code that you want to run. This allows you to isolate the code that you want to run instead of running the entire line. Again, run the code in the same way that was described above and is replicated here. Then either:
 - a. Run the code by pressing the ‘run’ button () on the top right corner of your R Script
 - b. Run the code by pressing **Command+Enter** at the same time (if you have a Mac) or **Control+Enter** (PC or a Mac)

5 Set a working directory

Setting a working directory can save you some big headaches. Basically, a working directory tells R where to pull any information from and where to save everything to. It would be a good idea to create a folder for everything you are going to do in R for this class and then set that folder as your working directory. Then, at the beginning of every session, you will use the same code to set your working directory and you won't need to worry about hunting anything down.

Note: this step is specific to you and your computer. *My working directory will not work on your computer.*

- First, figure out what your current working directory is by running the code `getwd()`. This tells you what R thinks your working directory is right now – it will likely be where you saved your R Script. It may look something like this:

```
getwd()
```

```
## [1] "/Users/Rob/Dropbox/UNC/Teaching/281 Spring 2019/In Class Materials"
```

This is telling me that my current working directory is in the In Class Materials folder, which is nested within my 281 Spring 2019, Teaching, and UNC folders, all of which live within my Dropbox. For me, this is perfect. However, if R returned something like this to me

```
## [1] "/Users/Rob/Dropbox/UNC/Teaching"
```

then I would need to make it more specific when I set my working directory.

- Now, set your working directory using the `setwd()` command

```
setwd('/Users/Rob/Dropbox/UNC/Teaching/281 Spring 2019/In Class Materials')
```

However, you should *always* use the `setwd()` command at the start of an R script, with the filepath that leads to the folder where your script lives. If you don't, R won't be able to find

6 Read in data

We frequently use R to analyze data. That is the whole point of this program. If we did not need to analyze data, we could pretty much just use a calculator to do what R does! So – getting data into R is extremely important. There are multiple ways to read in data, but I want you to use the following procedure throughout class. It is the quickest and, once you are accustomed to it, the easiest way to read in data. It is also the most versatile.

Most of the data we use will be in `.csv` (comma separate variable) format, so I am focusing these instructions on that. If you have data in other formats, it is still possible to read it into R. But for now, let's just get our `.csv` skills down pat.

1. Download the data
2. Once the dataset is downloaded, move it to the folder/location that you are using for your working directory. For me, this means I would put the data into my In Class Materials folder.
 - a. This will be specific to your computer. Do not try to look for my folders on your computer because they won't be there. Instead, put the data into the folder that you created for your R work in this class.
 - b. There are a couple of ways to move data (or any files) to the folder you want it in. You can go to your computer (on a PC) or finder (on a Mac) and locate the data in Downloads (if you recently downloaded it). Then, you can drag and drop it to the appropriate folder. You can copy and paste it to the appropriate folder. You can also open it (and it will likely open in excel) and then use Save As to save it to the appropriate folder. This last option will not work if the data is in a format that is more difficult for your computer to open, though.
3. Now that your data is in the folder that you are using for your working directory, reading the data in will be very simple. You are going to use the `read.csv()` command and give R the name of the data that you want to read in, in quotes. Make sure you include the `.csv` at the end of the name of the data.

But, if you do not assign this data to an object, then R will just show you the data without saving it or letting you do anything with it. So, you need to assign it to an object. You can call the object anything, but let's just stick with “data” for now.

```
# read in data  
ACD <- read.csv('ucdp-prio-acd-172.csv')
```

4. If you look at the Global Environment to the right, you should see now that the dataset has been read in.



7 Congratulations!

Now that you've officially started an R session and read in data, the fun begins! We can look at the data, summarize it, analyze it... the possibilities are limitless! **But:** Always remember to save your R Script frequently so you don't lose any of this progress.

7.1 Save your R script

1. Press the floppy disk icon on the top left corner of R Studio.
2. File → Save

If the title of your R Script is **red**, a current version has not been saved. If it is black, then it is saved.