

# Lab 0: Getting Started

Justin Baumann

## Table of contents

<b>1</b>	<b>Learning Objectives</b>	<b>2</b>
<b>2</b>	<b>Additional Tutorials and Resources</b>	<b>2</b>
<b>3</b>	<b>1.) Install R and RStudio</b>	<b>2</b>
3.1	Install R . . . . .	3
3.2	Install RStudio . . . . .	3
<b>4</b>	<b>2.) RStudio Layout</b>	<b>3</b>
4.1	Top Left: Script . . . . .	3
4.2	Bottom Left: Console/terminal . . . . .	3
4.3	Top Right: Environment, History, etc... . . . . .	3
4.4	Bottom Right: Files, Plots, Packages, etc... . . . . .	4
4.5	Top Bar: File, Edit, Etc... . . . . .	4
		<b>4</b>
<b>5</b>	<b>3.) The Working Directory - What is it, how to check it, and how to set it!</b>	<b>5</b>
5.1	How to SET the working directory . . . . .	6
<b>6</b>	<b>3.) RStudio Projects!</b>	<b>6</b>
<b>7</b>	<b>Using Quarto</b>	<b>7</b>
7.1	To make a Quarto document . . . . .	8
7.2	Formatting text . . . . .	8
7.3	Making a code chunk . . . . .	10
7.4	Rendering your report . . . . .	10

# 1 Learning Objectives

IN THIS TUTORIAL YOU WILL LEARN:

- 1.) How to access and/or install R and RStudio
- 2.) How to navigate RStudio
- 3.) How to set and change the working directory
- 4.) How to setup an RStudio Project

## 2 Additional Tutorials and Resources

[Datacamp Tidyverse tutorial](#)

[Books and workshops for learning tidyverse](#)

[A nice step by step walkthoug of Tidyverse functions](#)

[Video Tidyverse tutorial](#)

[Want to TRY some stuff on your own? Use the RStudio.cloud primers](#)

The best way to learn is to GOOGLE IT and try stuff

---

## 3 1.) Install R and RStudio

In this course we will learn how to program in R (a coding language) using RStudio (a coding environment). RStudio makes using R easier and more user friendly!

We will also learn how to make pdf and html output files that include code and outputs (tables and graphs).

These are handy tools for reporting data and even for writing papers! We will use Quarto to do this (a new tool from the folks who designed RStudio).

Your lab reports will all be built using Quarto.

**You have options:**

1. Install R and RStudio on your device(s) and use it locally
2. Sign up for a posit cloud account here: <https://posit.cloud/plans/free>. Posit Cloud is a way to access RStudio online without downloading and installing anything.

### 3.1 Install R

To install R, we will use this link: [install R](#)

- 1.) Choose the operating system you use (macosx or windows)
- 2.) Click the blue .pkg link that aligns with your computer and operating system (ask questions if needed)
- 3.) Follow instructions

### 3.2 Install RStudio

- 1.) Click [this link](#) and follow instructions
- 2.) OPEN RStudio (not R). Click on the logo that is a white R inside a blue circle (RStudio). We never need to open R, we can use RStudio.

---

## 4 2.) RStudio Layout

### 4.1 Top Left: Script

Where you will write your script(s). This is where we should be writing our code! It can be run, commented, and saved here.

### 4.2 Bottom Left: Console/terminal

Here you can run single lines of code and/or see error messages, warnings, and other outputs. Code should not be written here unless it is simple / for testing! Anything worth keeping should go in the script at the top left!

### 4.3 Top Right: Environment, History, etc...

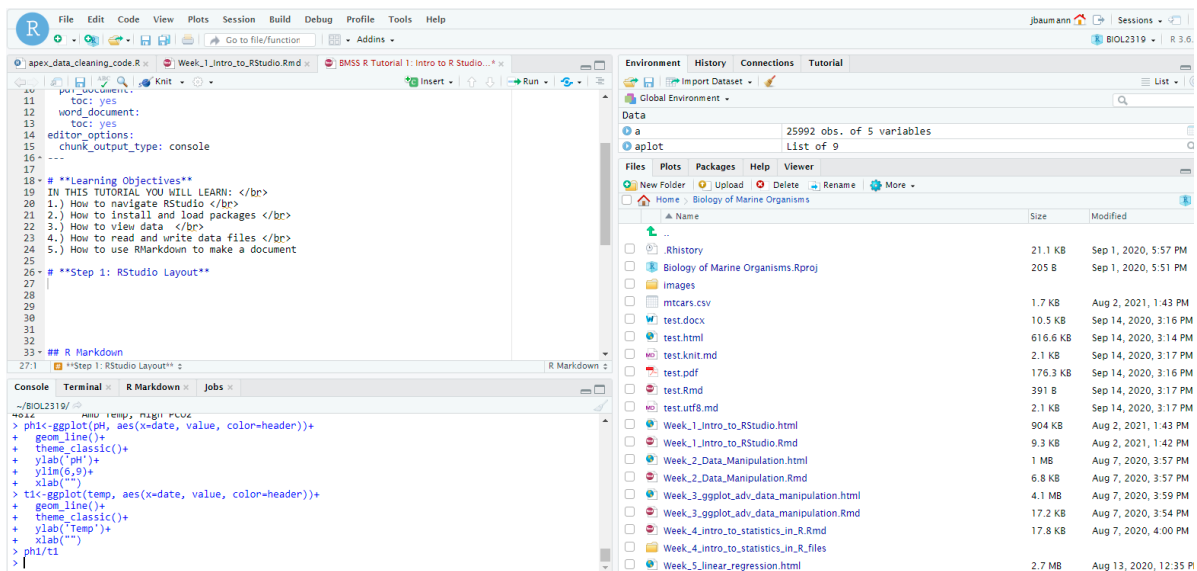
Here you will be able to see the dataframes you have read into R or created (using the “Environment tab”). The other tabs are less useful for us at this stage, but feel free to explore them! Note: The Broom icon can be used to clear dataframes from your environment. You can minimize or maximize this and each other quadrant using the symbols at the top right of the quadrant (a collapsed page next to a full page)

## 4.4 Bottom Right: Files, Plots, Packages, etc...

This is the second most important quadrant (behind top left) and we can change the working directory here very easily. Here we can see the files in our present working directory (we will learn about that next!) We can also see any plots we make in the plots tab. VERY importantly, we can see the packages we have loaded or installed in the packages tab. This will be useful to you! You can also use this tab to search the internal Help dictionary, though I will note that the internet is often more helpful!

## 4.5 Top Bar: File, Edit, Etc...

You can use the top bar in RStudio much like in any other program. I'll let you explore that on your own. Notably, in the top right corner of the top bar you will see an 'R' in a blue box. This is where you can set the project you are working form. Using projects is great because it allows you an easy way to compartmentalize your code, data, figures, and working directory for a single project all in one place! We will get to this shortly.



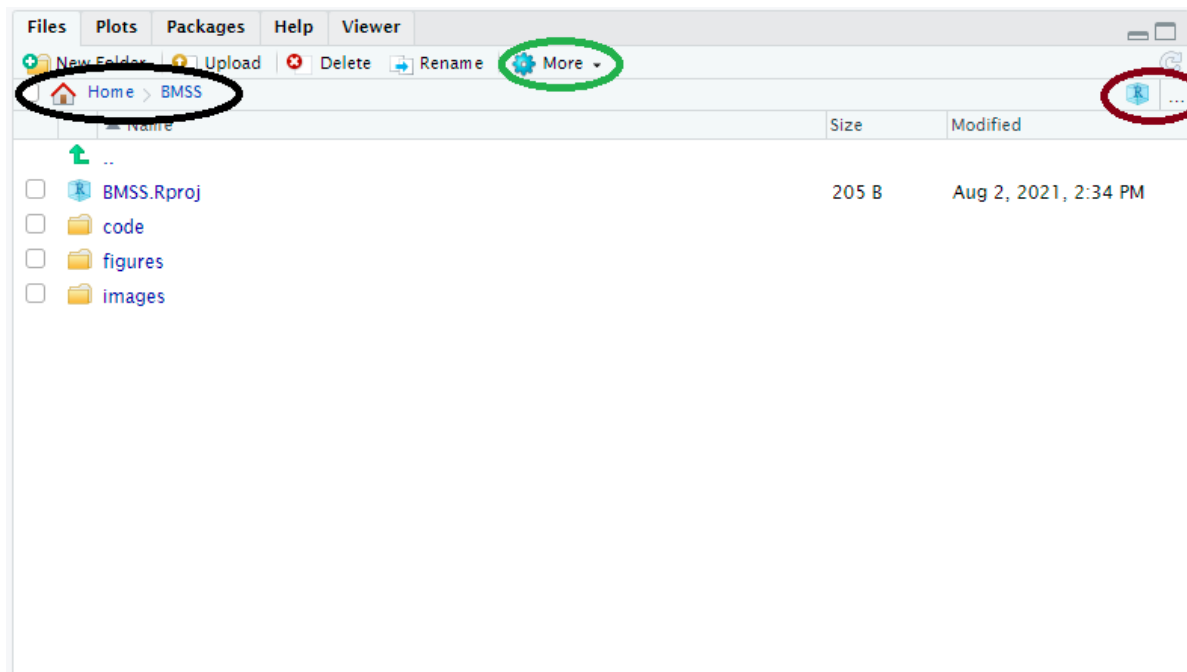
### 5 3.) The Working Directory - What is it, how to check it, and how to set it!

1.) We can use the `getwd()` command!

```
getwd()
```

```
[1] "E:/Google Drive/Teaching/Bates College/ENV 282 - Research Design in Env Sci/ENVR282_web"
```

2.) We can also use the Bottom Right “Files” tab



Here our working directory (and its file path) can be located in the black circle. We can manually change the working directory by using the ‘...’ in the brown circle to find any folder on our computer (or attached cloud folders), navigating to it, and then using the ‘More’ Cog in the green circle to “Set as working directory”

3.) An alternative approach to finding the working directory in the “Files” tab. Using the “More” cog, we can select “Go to working directory”

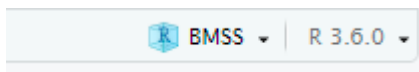
## 5.1 How to SET the working directory

1.) Using the “Files” tab to set manually: a.) Using the ‘...’ in the ‘Files’ tab you can select any directory (folder) on your computer. You can also set a google drive, box, dropbox, or other shared folder as your working directory if you’d like (as long as you are syncing a folder between the cloud and your computer – ASK me if you have questions about this!) b.) Once you navigate to a directory you still need to **SET IT** as your working directory. You do this in the “More” cog– select “Set as working directory”

2.) Set working directory with code: We use the ‘setwd()’ function for this. Below is an example. You will need to replace the path details with your own!

```
setwd("C:/Users/Justin Baumann/Teaching/Bates College/ENV 282 - Research Design in Env Sci
```

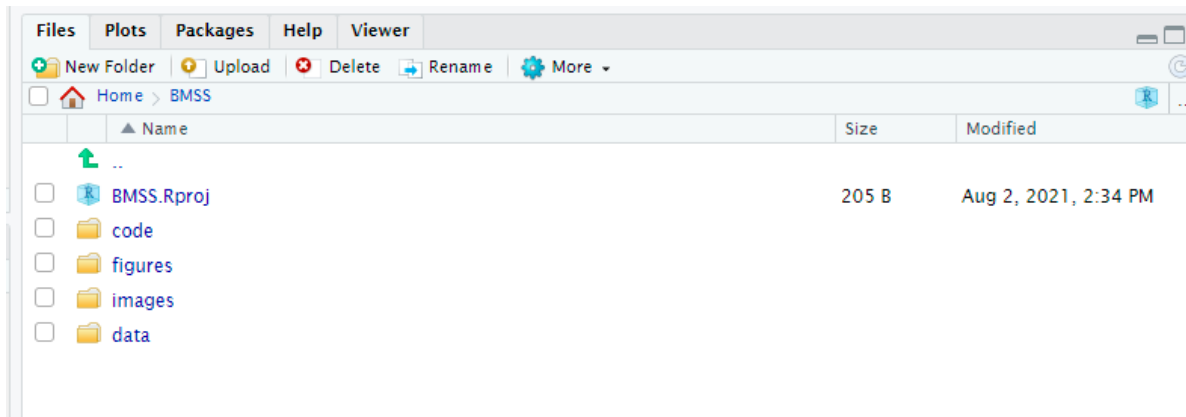
## 6 3.) RStudio Projects!



RStudio Projects are a great way to compartmentalize your coding work! You can store your code, outputs, input files, figures, etc all in one directory (with subdirectories). When you load your R Project, R will automatically load the last scripts you were working with on that project as well as the dataframes and items you have read in (your environment will be ready to go!). It will also navigate to the correct working directory automatically :) This will make your life easier!

*To make an RStudio Project*

1.) Create a folder on your computer (or cloud storage) that will serve as the MAIN directory for your project. Within that folder I recommend you make subdirectories for all of your R related inputs and outputs. Maybe something like I have here:



2.) Once you have a MAIN directory folder created (whether you've made subdirectories or not) you can create a project! Set your main folder as your working directory. Next, navigate to the TOP RIGHT of your screen and select the down arrow next to the Big "R" in a blue box. NOW, select "New Project" -> Existing directory -> Name the project and hit done! At this point you will see a .Rproj file appear in your MAIN directory. This means you did it right :) This .Rproj file is how you save all of your project info. It autosaves and when you select your project (Again, TOP RIGHT of your screen, select the down arrow next to the R in the blue box and then select your project name) it will load up your scripts, environment, and set your working directory as the MAIN folder. You can navigate VERY easily from here :)

## 7 Using Quarto

Quarto is a report building software that is integrated into RStudio. It replaces RMarkdown, if you have used that in the past, and is usable with python, julia, and R. Thus, learning it is a transferable skill. Quarto is designed to allow you to easily write documents that integrate text, hyperlinks, code, and images into a one neat file. This website, for example, is made entirely in Quarto! Quarto documents, or markdown documents as they are more generically known, are common in data science. these documents are great for courses, as you can do your programming, share your code, results, figures, stats, and explanations all in one document. Instead of the instructor downloading your code and running it line by line, we can see the results of the code you write just below the code itself! Super nice for assessing work. Plus, Quarto documents will not render unless the code is error free, so this is a nice way for students to check their own work.

Beyond course use, Quarto and markdown is excellent for creating professional looking data driven reports as well as online resources (like this website :) ). Learning Quarto is a great

skill for anyone interested in programming, data, or the sciences! So, let's learn how to use it!

## 7.1 To make a Quarto document

click file -> new file -> Quarto document / Complete the pop up prompts and then wait for the document to load. / We want to replace the top bit (our YAML header, everything between the two lines that contains just — at the top) with the following (use your name and title!)

```
---
title: "Lab 1: Intro to R, RStudio, and Quarto"
author: "Justin Baumann"
format:
  html:
    toc: true
  pdf:
    toc: true
    number-sections: true
    colorlinks: true
editor: visual
---
```

## 7.2 Formatting text

Unlike in a regular R script, using the '#' at the start of a line will not comment that line out. Instead, you can type as you would normally in an R Markdown (Rmd) document. We can format our text in the following ways:



## syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
##### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi * r^{2}$
image: 

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
  + sub-item 1
  + sub-item 2

1. ordered list
2. item 2
  + sub-item 1
  + sub-item 2

Table Header | Second Header
-----|-----
Table Cell   | Cell 2
Cell 3       | Cell 4
```

## becomes

Plain text  
End a line with two spaces to start a new paragraph.  
*italics* and *italics*  
**bold** and **bold**  
<sup>superscript<sup>2</sup></sup>  
~~strikethrough~~  
[link](#)

# Header 1

## Header 2


### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: –  
emdash: —  
ellipsis: …  
inline equation:  $A = \pi * r^2$

image: 

horizontal rule (or slide break):

---

block quote

- unordered list
  - item 2
    - sub-item 1
    - sub-item 2
1. ordered list
  2. item 2
    - sub-item 1
    - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

Bold: “ on either end of a word, phrase, or line will make it bold! this is in bold\*\*  
=’**this is in bold**” without the quotes around the \*\*

Line breaks: DO you want text to be on different lines? Insert a ‘’ at the end of a line to make a line break!

### 7.3 Making a code chunk

Since qmd documents are text based, we need to tell RStudio when we want to actually include code. To do this, we will insert a code chunk. To insert a code chunk:

- 1.) Use the keyboard shortcut ‘ctrl’+‘alt’+‘i’ (PC) or ‘cmd’+‘alt’+‘i’ (Mac) to insert a code chunk.
- 2.) Navigate to the top bar (of the top left quadrant of RStudio), find “+c” at the right of the bar to insert an R code chunk.

Once you have a code chunk inserted you will notice that the background of the chunk is gray instead of your default background color (white or black if you are in dark mode)

```
#this is an example code chunk

# Using '#' at the start of a line indicates a comment, which is not runnable code!
```

### 7.4 Rendering your report

To Visualize what your report will look like, click the ‘visual’ tab in the top bar (on the left). Note that if you do this, it CAN change your code—so be careful. You can also use the GUI to alter your report in the visual tab. This provides a nice alternative to the code based formatting options in the ‘source’ tab.

To actually render into an html or pdf document, you must click “Render”. You can use the arrow to the right of “Render” to choose render to html or render to pdf. I suggest using HTML most of the time but you can use pdf if you prefer. You will need to successfully Render your quarto document into an html or pdf report in order to turn in your labs!