

# Psychological Statistics Lab

# 1

PSYC 2020-A01 / PSYC 6022-A01 | 2025-08-22 | Basic R

Jessica Helmer

# Outline

- Introductions!
- Lab Syllabus
- What is R?
- Today's Dataset
- R's Layout
- R Basics

# Instructor Introduction

Hi! I'm Jessica Helmer. I'm a second-year PhD student in Quantitative Psychology.

My research focuses on the measurement and modeling of decision-making processes.

- Email: [jhelmer3@gatech.edu](mailto:jhelmer3@gatech.edu)
- Office: J.S. Coon Rm 240
- Office Hours: Wednesdays, 10–11am, my office / [Zoom](#)
  - Or by appointment

# What You Can Expect From Me

I am available as a resource! Please feel free to reach out / email anytime if you need help, have questions about the lab, or have any suggestions or feedback for the class.

I encourage helping each other / collaboration during the lab! While working on assignments, I will try to answer everyone's questions as soon as I can. but if someone near you is confused and you can help, go for it!

# Student Introductions

- Name
- Year at Georgia Tech
- Major
- Coding Experience?
- Anything else!

# Lab Assignments

Students are expected to attend and participate in all scheduled labs.

We will have a lab lecture and assignment each week (barring institute holidays).

Students are expected to work on the activity during the scheduled lab time.

Lab assignments are due on Canvas at 11:59pm the same day they are distributed.

We will not be able to accept late lab assignments!

Assignment distributed after lecture (with plenty of time to complete!)

# Lab Assignments

**Always** offer a few words of explanation for questions in the assignment.

- Output alone will not get full credit.

Please submit both the knitted (.html) and the RMarkdown (.rmd) file.

- Without your R code, we will not be able to give partial credit if the result is incorrect.

**Always** provide R output for every analysis carried out.

- If the assignment asks you to make a plot, include that plot in your submission (not just the code).

Graded 0–100, equally weighted to compose 20% of your PSYC 2020 / PSYC 6022 grade

# About AI

AI can be a very helpful tool for learning coding! Feel free to use it to clarify concepts, troubleshoot code, or syntax explanations

However, for the assignments, your code needs to be your own.

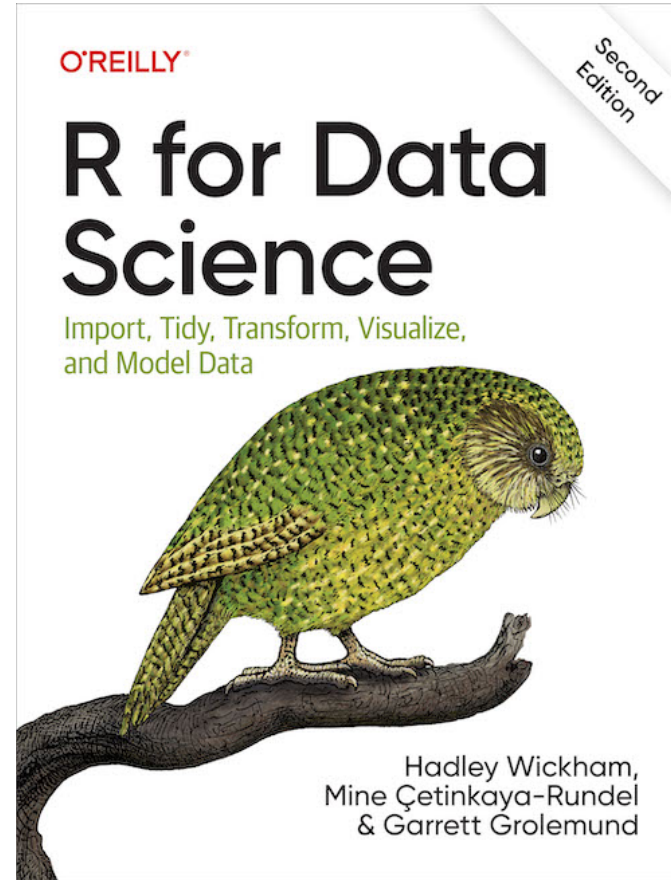
It is important to practice and gain experiencing coding independently.



# Lab Resources

Introduction to Data  
Exploration and  
Analysis with R  
(IDEAR)

R for Data Science  
(2e)



YaRrr! The Pirate's  
Guide to R



All free, open source materials for learning R

**Any questions on syllabus, assignments, materials?**

# What is R?

Open source coding language, primarily used for data analysis and data science

Coding (per the IDEAR book):

“Giving very specific instructions to a very stupid machine.”



```
1 print("Hello, World!")
```

```
[1] "Hello, World!"
```

# Why R?

Because R is a programming language, you can make your own codes / functions as commands

R has many “packages” for statistical functions

R is open source: you can see the exact code that underlies the functions you use

- And if you'd like, you can modify it

# What is RStudio?

An interface for R

Helps keep R files organize

Helps “knit” code, output, and text to create reports

# Installing R

We need to download two things:

R and RStudio

R is the language, RStudio is an Interactive Development Environment (IDE)

We will use R through RStudio

# RStudio Introduction

Files, Plots, Packages

Environment, History

Console



# R Console

```
1 2 + 4
```

```
[1] 6
```

```
1 2 + 4 / 4
```

```
[1] 3
```

## PEMDAS!

```
1 2 ^ 3
```

```
[1] 8
```

```
1 8 %% 3
```

```
[1] 2
```

# Vectors

```
1 c(2, 4, 6)
```

```
[1] 2 4 6
```

```
1 c(1, "hello!", "Georgia Tech")
```

```
[1] "1"          "hello!"      "Georgia Tech"
```

## c() function

- Stands for “combine”

## Matrix algebra!

```
1 c(1, 2, 3) / 3
```

```
[1] 0.3333333 0.6666667 1.0000000
```

```
1 c(1, 2, 3) / c(1, 2, 3)
```

```
[1] 1 1 1
```

# Data Types

```
1 c(1, 2, 3)
```

```
[1] 1 2 3
```

```
1 c(1, "hello!", "Georgia Tech")
```

```
[1] "1"          "hello!"      "Georgia Tech"
```

Quotes around **1** in the first vector, not in the second

Vectors can only hold one type of data!

- Numeric
- Character

```
1 c(1, 2, 3) / 3
```

```
[1] 0.3333333 0.6666667 1.0000000
```

```
1 c(1, "hello!", "Georgia Tech") / 3
```

```
Error in c(1, "hello!", "Georgia Tech")/3: non-numeric argument to binary operator
```

Error because we cannot divide characters

# Data Types

```
1 TRUE
```

```
[1] TRUE
```

```
1 c(1, "hello!", TRUE)
```

```
[1] "1"      "hello!" "TRUE"
```

```
1 c(TRUE, FALSE)
```

```
[1] TRUE FALSE
```

# Data Types: Logical (boolean)

```
1 6 > 4
```

```
[1] TRUE
```

```
1 12 / 2 < 6
```

```
[1] FALSE
```

The `<`, `>`, and `<=` symbols are *logical operators*

Logical data handled like binary numeric data

```
1 TRUE == 1
```

```
[1] TRUE
```

```
1 FALSE + 3
```

```
[1] 3
```

`TRUE` and `FALSE` can be abbreviated as `T` and `F`

```
1 TRUE == T
```

```
[1] TRUE
```

```
1 FALSE == F
```

```
[1] TRUE
```

Harder to tell apart, so be careful!

# Practice Break!

```
1 2 + 2
```

```
[1] 4
```

```
1 2 + 2 / 2
```

```
[1] 3
```

```
1 2 + 2 / 2 ^ 2
```

```
[1] 2.5
```

```
1 c("hello!", 2) / 2
```

```
Error in c("hello!", 2)/2: non-numeric argument to binary operator
```

# Variable Assignment

In R, we can assign objects (including functions) to some variable with the *assignment operator* `<-`

```
1 # assign variable a the value 1
2 a <- 1
3
4 # print a
5 a
```

```
[1] 1
```



# Variable Assignment

```
1 a <- c(1, 2, 3)
2 a
```

```
[1] 1 2 3
```

```
1 a <- c("Hello, world!")
2 a
```

```
[1] "Hello, world!"
```

Every time we reassign a value to `a`, it holds that new value

Overwrites previous value!

# Variable Assignment

Overwriting can mess stuff up!

```
1 T <- F
```

```
1 T
```

```
[1] FALSE
```

```
1 T == F
```

```
[1] TRUE
```

imgflip.com



# Variables in Statistics: Numeric vs. Categorical

## Numeric

- Quantitative
- Values describing a measurable quantity as a number
- “How many” or “how much”

## Categorical

- Distinguish distinct elements or entities
- Name, gender, ethnicity...

# Four Types of Measurement

## Nominal

- Variable that represents categories without any inherent order.
- E.g., types of fruit (“apple,” “orange,” “banana”).

## Ordinal

- Variable that represents categories with a meaningful order.
- For example, levels of satisfaction (“very unsatisfied,” “unsatisfied,” “neutral,” “satisfied,” “very satisfied”)

## Discrete

- Variable that can only take specific values.
- Think of things you can count, like the number of people in a room or the number of apples in a basket.

## Continuous

- Variable that can take any value within a range.
- Think of things you can measure, like height or time.

# Dataframes

Table of data

Can make by hand with the `data.frame()` function:

```
1 data.frame(x = c(1, 2, 3),  
2           y = c("a", "b", "c"),  
3           z = c(TRUE, FALSE, TRUE))
```

	x	y	z
1	1	a	TRUE
2	2	b	FALSE
3	3	c	TRUE

Shift+Enter to go to new line on Console

Most of the time, we import data (and don't manually type it in)

In console, can press up arrow to reload last line of code

# Dataframes

If you already have vector objects saved, you don't have to type them in again

```
1 a <- c(1, 2, 3)
2 b <- c(2, 3, 4)
3 c <- c(3, 4, 5)
```

```
1 data.frame(a = a,
2             b = b,
3             c = c)
```

	a	b	c
1	1	2	3
2	2	3	4
3	3	4	5

```
1 data.frame(a, b, c)
```

	a	b	c
1	1	2	3
2	2	3	4
3	3	4	5

```
1 data.frame(A = a, b, c)
```

	A	b	c
1	1	2	3
2	2	3	4
3	3	4	5

# R Projects

RStudio's way of helping organizing files, scripts, etc.

I strongly recommend this!!

- File > New Project
- If you don't already have a folder associated with this class, "New Directory"
- If you do, "Existing Directory"

All R Scripts under the same project share a *working directory*

- Location of files

`getwd()` tells us the location of our working directory

`setwd("C:/Users/Desktop/R Example")` sets the working directory

Or, `here::here()` lets us do relative directories (my favorite!)

- Just use the command at the top of the file to see where your directory is

# R Scripts

R script files end with .R

## How to Create an R Script File

### 1. Manually

File > New File > R Script

### 2. Shortcut

Ctrl + Shift + N



# RMarkdown

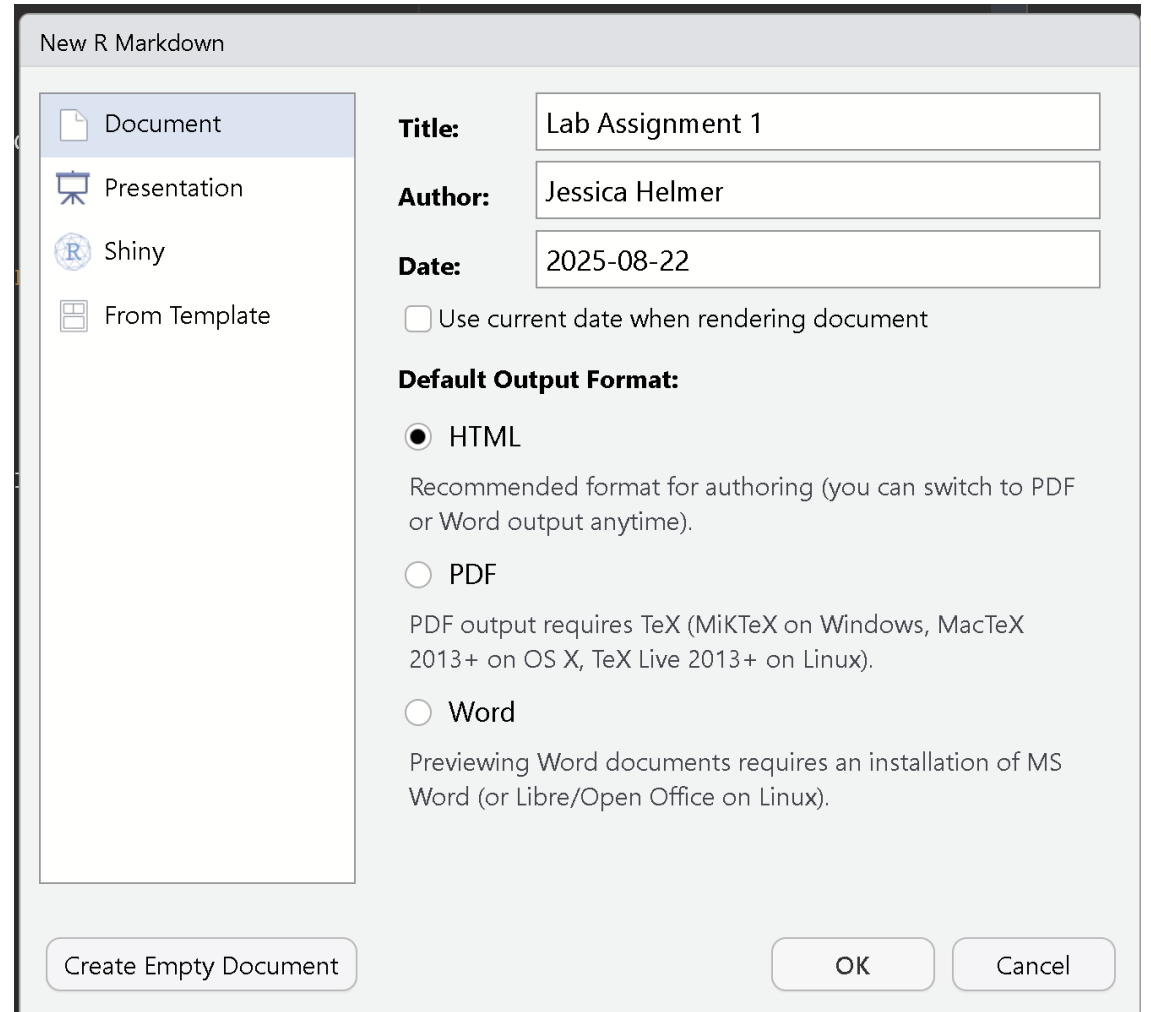
A way of typesetting R code into a document

End in .rmd

File > New File > RMarkdown...

Input Title, Author, Date

How we will complete assignments



The screenshot shows the 'New R Markdown' dialog box. On the left, there is a list of options: 'Document' (selected), 'Presentation', 'Shiny', and 'From Template'. On the right, there are input fields for 'Title' (Lab Assignment 1), 'Author' (Jessica Helmer), and 'Date' (2025-08-22). Below these fields is a checkbox labeled 'Use current date when rendering document' which is unchecked. Under the heading 'Default Output Format:', there are three radio buttons: 'HTML' (selected), 'PDF', and 'Word'. Below the 'PDF' option, there is a note: 'PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux)'. Below the 'Word' option, there is a note: 'Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux)'. At the bottom left is a button labeled 'Create Empty Document'. At the bottom right are 'OK' and 'Cancel' buttons.

# Insert a Chunk

How to insert an RMarkdown chunk in RStudio

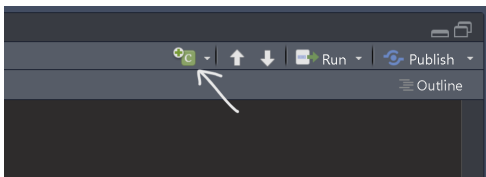
## 1. Manually:

- Type `` `` {r}``.
- Add your R code.
- Close the chunk with `` ```.

## 2. Shortcut:

- Use the shortcut Ctrl + Alt + I (Windows) or Cmd + Option + I.

## 3. Button



# Run a Chunk

How to run RMarkdown code chunks in RStudio

## 1. Shortcut:

- Use the shortcut Ctrl + Enter (Windows) or Ctrl + Return (Mac)



## 2. Button

Can also select a variable and run it to see its output (helpful!)

# Clearing Environment and Restarting R

Important to ensure that your code isn't unintentionally relying on hidden objects in the environment

- E.g., you created an object, deleted the line to create it, but R still sees it there

When you try to knit your file, it will clear, restart, and run all code

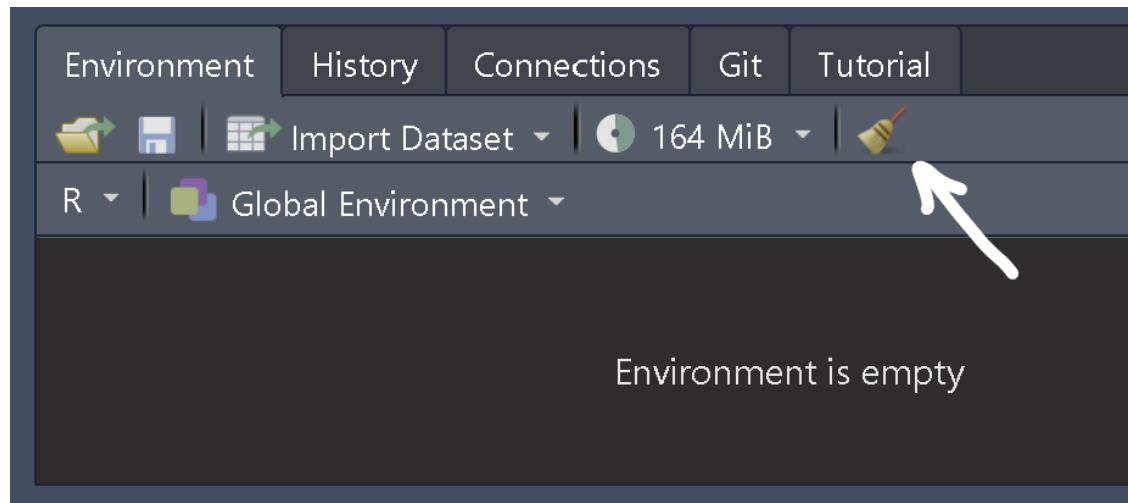
- In the previous situation, your code would error

Clearing your environment and restarting R frequently helps you avoid this

# Clearing Environment and Restarting R

## How to Clear Your Environment

### 1. Button



## How to Restart R

### 1. Manually:

- Session > Restart R.

### 2. Shortcut:

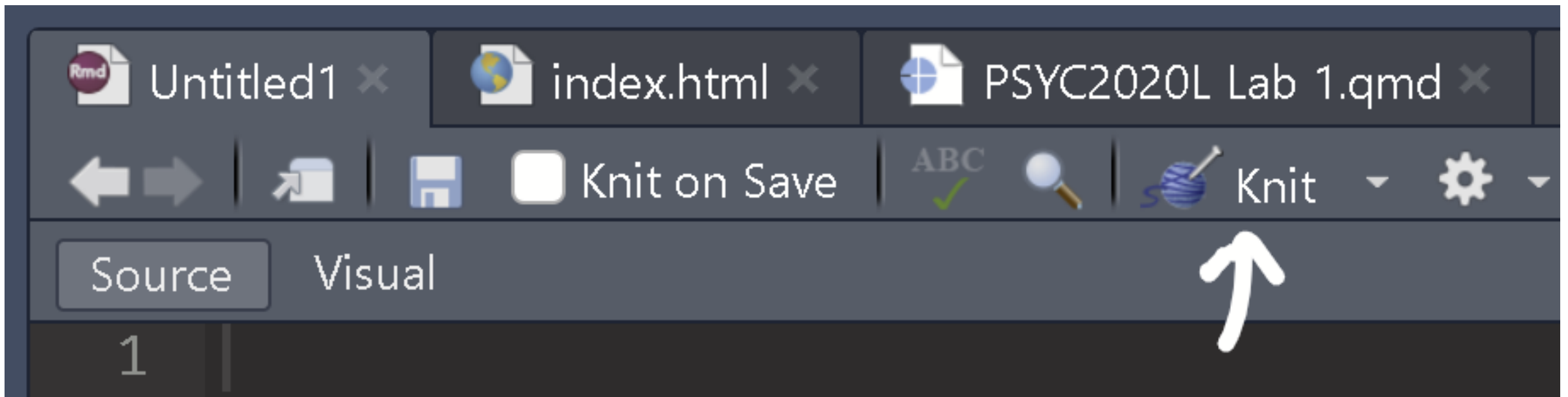
- Use the shortcut Ctrl + Shift + F10 (Windows) or Cmd + Shift + F10.

# Tips for RMarkdown!

Run code often to ensure it's working

Restart R frequently

Knitting will combine text, syntax, results, and output into a Word, PDF, or HTML file



# Today's Dataset

`iris` dataset has measurements of 150 flowers, preloaded from R

`head()` function shows first few rows of a dataframe

```
1 head(iris)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4		
0.2					setosa
2	4.9	3.0	1.4		
0.2					setosa
3	4.7	3.2	1.3		
0.2					setosa
4	4.6	3.1	1.5		
0.2					setosa
5	5.0	3.6	1.4		
0.2					setosa
6	5.4	3.9	1.7		
0.4					setosa

`tail()` shows last few rows

Can specify how many rows you want

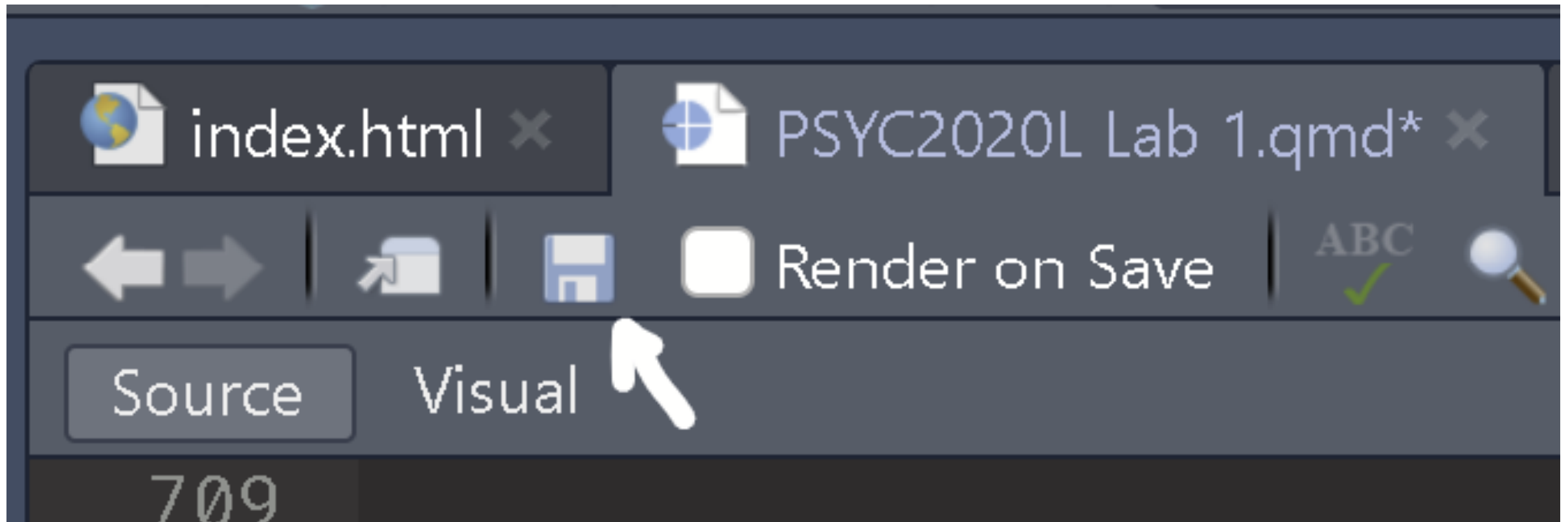
```
1 tail(iris, n = 3)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
148	6.5	3.0	5.2		
2.0					virginica
149	6.2	3.4	5.4		
2.3					virginica
150	5.9	3.0	5.1		
1.8					virginica

# Saving Work

Press Ctrl + S frequently to save your work!!

Can also press the save icon in the top left





# Suggested R Workflow

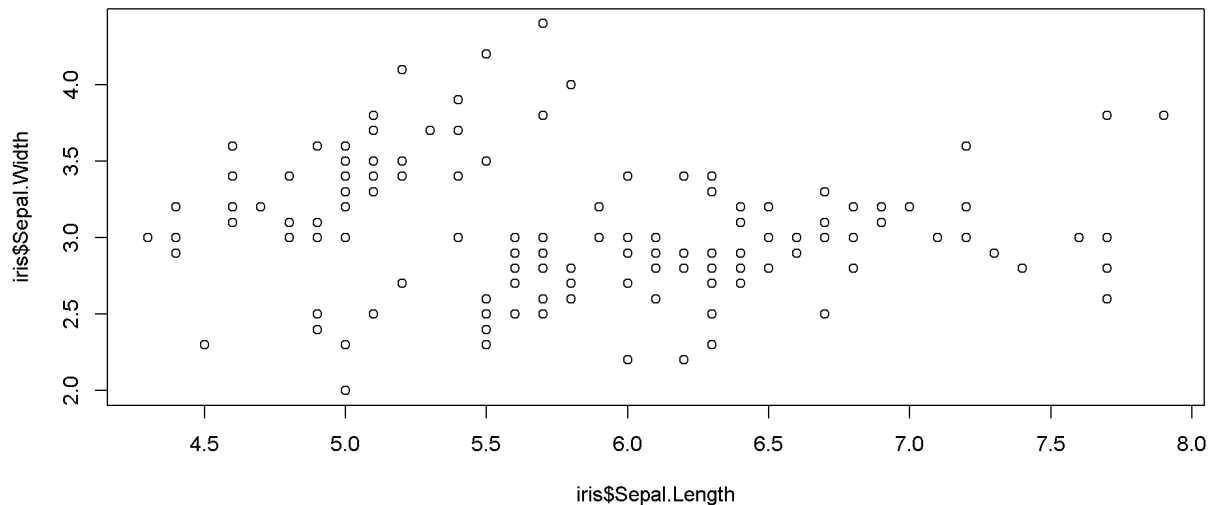
1. Open a new script or RMarkdown file
2. Check (and set if needed) your working directory
3. Load any packages you'll be working with
4. Write your code
5. Knit and save your file

# Data Visualization

Let's make our first plot!

\$ operator references columns within a dataframe

```
1 plot(iris$Sepal.Length, iris$Sepal.Width)
```



Do we like this plot?

# Assignment 1

Objectives:

- Do some basic R!
- Get familiar with RMarkdown