

Week 1: Intro to R and RMarkdown

Billy McLean

8/31/2022

Assignment checklist:

Part 1

- Fill in your name above - Done
- Answer the question below - Done
- Write a comment in the code block below - Done
- Change the title of the graph below - Done

Part 2

- Create a folder somewhere on your laptop where will store all your assignments this semester - Done
- Set 'your working directory' - Done
- Install and load 'readr' - Done
- Import the file 'TrophicStateExample.csv' - Done
- Using base R, create a simple scatterplot with Chlorophyll a on the Y axis and Secchi depth on the X axis. Make sure to include...
 - A graph title - Done
 - X and Y axis labels including units - Done
- Demonstrate commenting out a piece of broken code - Done

Save as an html, convert to pdf and upload to Canvas

PART 1:

What are you hoping to learn/ what are you most excited about:

I'm hoping to learn to better apply my existing R knowledge toward analyzing inland waters. Previous classes have felt more theoretical and big picture so I am looking forward to working more directly on projects

R code instructions.

In part two, we will learn how to import data into R. First, we will just use some 'fake data'.

An important part of writing code is to 'comment' it, both to remind future you of what your code is for, help collaborators understand your code and to keep your code organized. You can write a comment by using a '#'. Writing a note in the code block without using a '#' will give you an error.

In the code block below, **write the name of a book you've enjoyed with a comment**

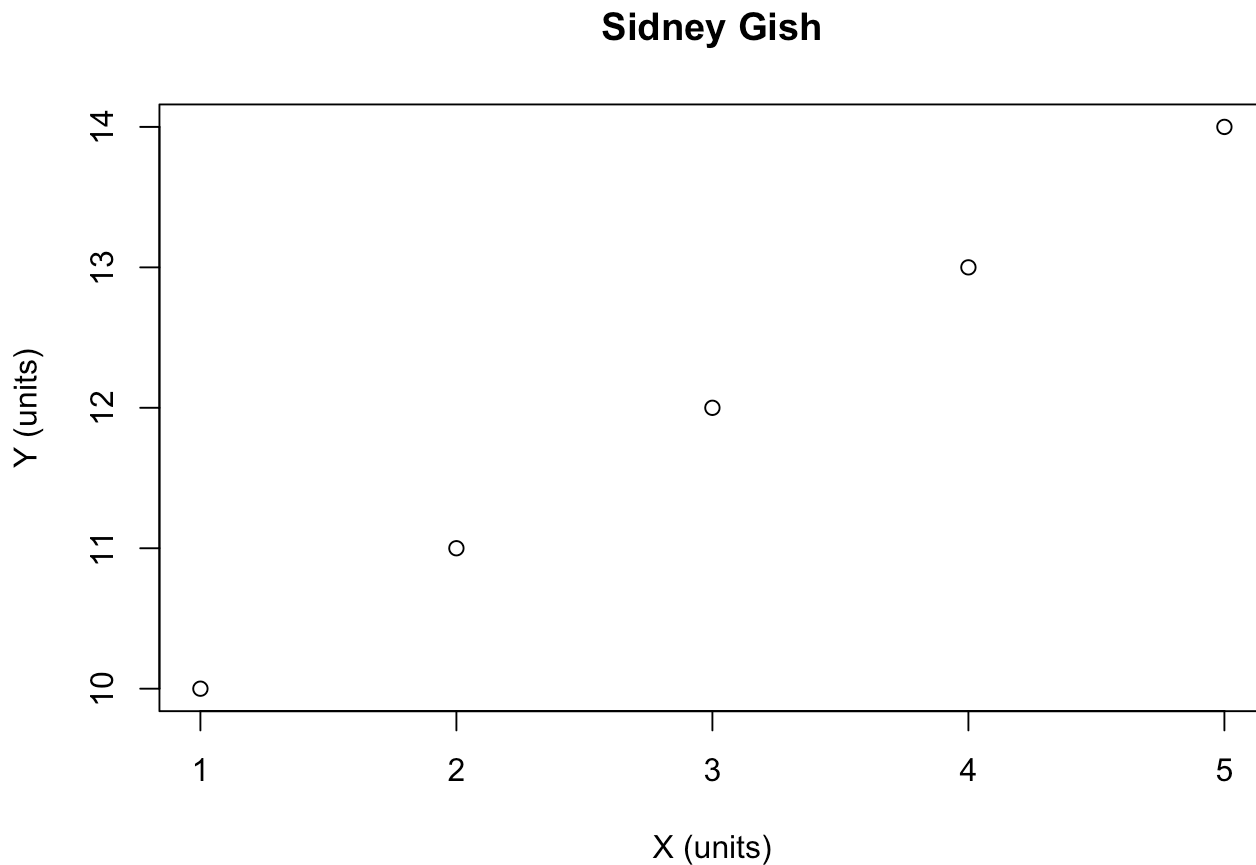
Below there is code for a simple scatterplot. This semester, we will learn how to make more complex and customizable graphs and figures, but we will start with this. Take a look at the code and **change the title to the name of your favorite band or musician**

```
#This a comment
#Below, please write the name of a book you've enjoyed in a comment...
#I enjoyed reading The Sound of A Wild Snail Eating, by Elizabeth Tova Bailey

x<- c(1,2,3,4,5) #You can also write a comment after code.
y<- c(10,11,12,13,14)

example_data<- data.frame(x,y)

plot(y ~ x, data=example_data,
     xlab="X (units)", ylab="Y (units)",
     main="Sidney Gish")
```



PART 2:

R code instructions.

In the example above we used a fake dataset that was generated with R code. Now we will learn to import data into R. The first step is to create your 'working directory'. This is a permanent folder on your computer (usually either in Documents or your Desktop- really anywhere *except* Downloads). If you are working on a WCNR borrowed laptop, make sure to make a folder in your N drive. Make sure to save your R files and all the data files you are using in the same folder.

Make a folder and save this RMarkdown and the TrophicStateExample.csv file there

There are a few ways to set your working directory. Below is some code to set your working directory. You will need to update the file pathway to match your computer.

```
#this is my pathway, edit the pathway and run the line.
setwd("/Users/wamclean/github/LimnologyLabs/Lab1")

#If you ever want to know your current working directory, you can run getwd(). After setting your working directory, run the getwd() in the space below. You should then see your pathway in the console below.
```

Now that you've set your working directory, you are ready to import your data. However, first you will need to install a package. At the top of your screen, go to 'Tools'. Select 'Install Packages...'. In the search bar, type readr. Make sure 'Install dependencies' is checked and click 'Install'. You only need to do this once (unless you download an updated version of R, then you will need to re-install your packages).

While you only need to install packages once, you will need to call the packages every time you use them. This is done with the **library()** command.

Run the code in the block below to call the readr package you just installed and then import TrophicStateExample.csv.

```
library(readr) #Note: this package imports .csv files.
#You will need a package called readxl to import excel sheets.
#I always recommend working with .csv files because they are less prone to errors.

TrophicStateExample <- read_csv("TrophicStateExample.csv")
```

```
## Rows: 14 Columns: 2
## — Column specification —————
## Delimiter: ","
## dbl (2): chl_a_ugL, secchi_depth_m
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Next we want to look at our dataframe. You can either double click the name of your imported dataset in your 'Environment' to the right or typing the name of the dataframe in the code block. Two things to notice about this data...

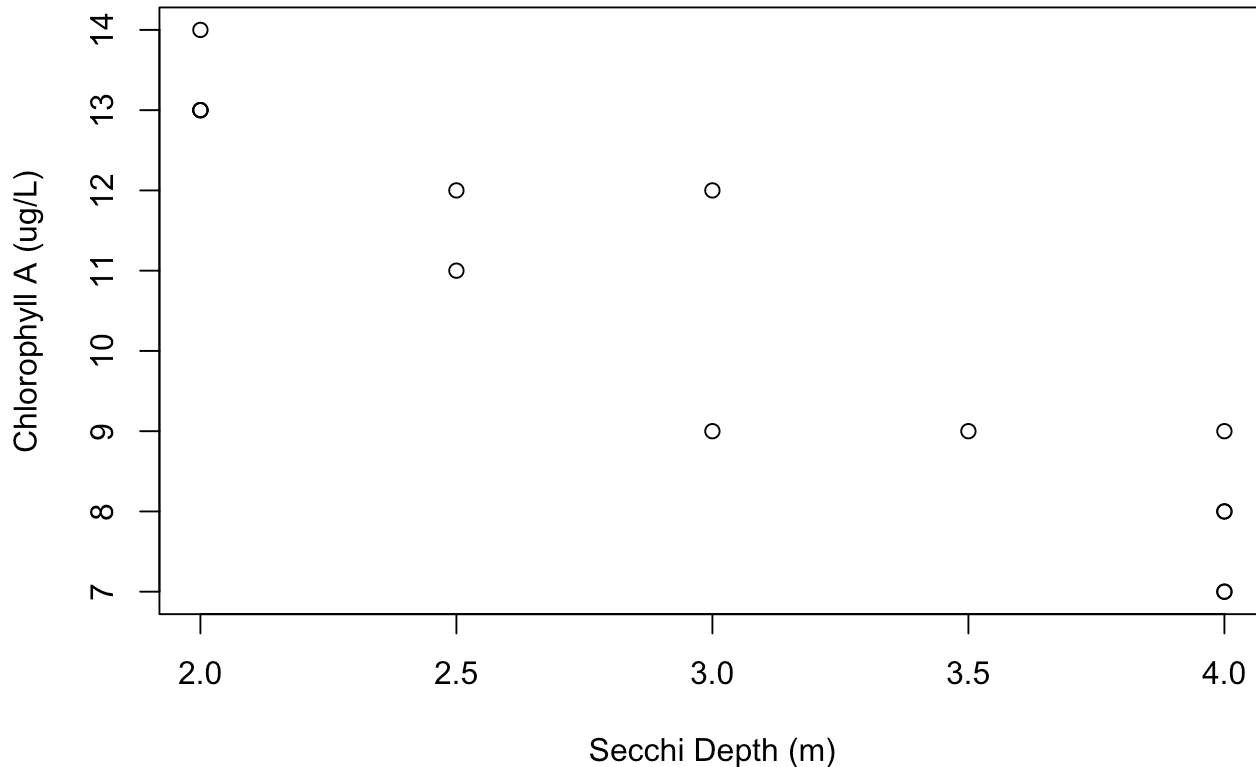
1. The units are included in the column name (I recommend doing this whenever possible to make sure you never lose track of your units). In this dataset, Chlorophyll a is in ug/L and Secchi depth is in m.
2. The column names don't have spaces or any special characters. R sometimes has trouble reading special characters (e.g., :, #) and you will often encounter an error when importing a column name with a space in it. You can fix these things in R but I recommend that you just check column names in excel before importing your data in R and making any necessary changes then.

In the code block below, create a simple scatterplot. You can copy and paste the code from above and edit as needed.

Remember that R is case sensitive

```
#Write your code for your scatterplot here
plot(chl_a_ugL ~ secchi_depth_m, data=TrophicStateExample,
     xlab="Secchi Depth (m)", ylab="Chlorophyll A (ug/L)",
     main="Chlorophyll A vs. Secchi Depth")
```

Chlorophyll A vs. Secchi Depth



Finally, it is important to know how to “comment out” code. Sometimes you will want to comment out code because you want to keep it for later but don’t want to run it now or because you need to troubleshoot something (for example: the code for changing the color or the points in a scatterplot is incorrect and is giving you an error for your whole graph. You can comment out that line but still produce the graph, just without the color edit you wanted).

RMarkdown will not successfully knit with *any* errors in your code. As our assignments get more complex, it’s important to know how to comment out your code so I can see what you tried and you still get credit for the rest of your code. (Similar to showing your work in a math class).

Below is some code that **will not** run because it uses a package that you have not installed yet and references data you don’t have. Instead of deleting the code, comment it out.

```
#plot<- ThisWontRun
```