Advances in Biology Laboratory Education
Publication of the Association for Biology Laboratory Education
Volume 41, Article 8, 2020

Learning Basic Data Analysis Skills in Intro Biology Labs Using R

Linda Forrester and Rachel S. Schwartz

University of Rhode Island, Department of Biological Sciences, 120 Flagg Rd, Kingston RI 02881 USA (**LindaForrester@uri.edu; rsschwartz@uri.edu**)

Undergraduate students are increasingly in need of basic skills in data analysis. As a complement, analyzing data provides an opportunity for active learning. Having students utilize data visualization and statistical analysis skills in introductory classes (with their own data) allows students to quickly develop an appreciation for digital skills. Here we introduce these skills to freshmen, giving all students an equal footing in basic data analysis in the context of learning biological concepts. We developed a simple, step-wise introduction beginning with the free, spreadsheet program Google Sheets for storing data and continuing with importing data to the free, data analysis language R in the R-Studio environment. Our approach through the semester provides students with code that they can use to create boxplots, scatterplots, and line graphs in R. Students also use code to produce error bars, *t*-tests, and best-fit linear regressions. Students responded that this approach was effective in making them feel that they can use a coding language program to analyze and graphically display their data. Importantly, students can build on these skills in future classes as R can handle a wide range of analyses. Here we demonstrate how these skills can be taught in the context of learning introductory biology.

Keywords: data analysis, species diversity, coding, R

Introduction

Biological conclusions, such as those presented to students in textbooks, are based on results from data analysis. Recently, data collection has become easier, and data sets continue to get bigger. These data provide a wealth of opportunities for students to do their own analysis and understand the origin of biological concepts through active learning. Here we employ a coding-to-learn model to help students learn code and data analysis as tools to effectively learn biological concepts.

With massive datasets from automatic data logging to DNA sequencing, our ability to "scroll through the data to see what is going on" is outstripped by our ability to collect more data. Most biologists can collect enormous amounts of data; now they need the skills to analyze those large data sets. Furthermore, data analysis skills are prized by employers, biology labs, and graduate students, but not often required by university undergraduate biology programs (Barone et al., 2017; Wilson Sayres et al., 2018).

In a world where employers are looking for people who can utilize data analysis to assess practical data sets, universities need to help students understand how to effectively work with now-common large datasets using code by making it easily accessible and understandable.

Additionally, learning to write code helps develop analytical and reasoning skills. Like learning a new language, learning to write code reveals new ways of expressing abstract ideas, encourages analytical thinking and helps develop creative, new ideas. Understanding how data is organized and analyzed by computers allows an appreciation of how data can be used to express new ideas and theories. Using script-based software, like R, removes the "black box" effect, giving student the ability to experiment with data analysis and data presentation (Berger 2016).

We designed our labs to help our students enter the world of data analysis using code by analyzing their own data in R. Presented here is our series of four labs to allow students to scaffold the data analysis skills in R, beginning with a two-week lab that we run in the first two weeks of the semester. In week 1 of this lab, students collect data examining species diversity indices. In our example, students count plankton samples from Narragansett Bay (near the University of Rhode Island), but the species diversity counts could be calculated from any environments. In week 2, students learn to write basic code in R to (a) examine and their own data and then (b) apply the same code to analyze larger (similar) data sets. We present the code in a stepwise fashion, explaining each portion of the lines of code, allowing students to learn through repeated, building blocks. First, we have the

students use the real data that they have collected themselves. We find this helps students to have an affinity for their data set; they have a better understanding of what they are analyzing, and greater willingness to learn a new approach to analyzing data. Second, the students apply code to larger data sets and students begin to see the applicability and universality of using code for data analysis. These labs are taught as part of the second semester of a year-long introductory biology sequence.

Scaffolding with Code

These short, one-hour, fifty-minute labs are taught by Graduate Teaching Assistants (GTA), who may or may not have much previous experience with coding. Thus, we provided detailed written tutorials to support student learning. Notably, detailed instructions for an analysis are provided once, with students expected to review their prior code and tutorials, altering the variables in their previously written lines of code when a similar analysis is repeated on a new data set. We utilize this same approach within the first lab tutorial, and then with subsequent lab tutorials. This repeated scaffolded approach, showing the repeatability of the code helps students to quickly see patterns and feel increased confidence in their coding skills.

Student Outline

Lab1A: Field Sampling - Measuring Diversity in a Marine Habitat by Assessing Plankton Diversity in Narragansett Bay

Objectives

- Students will learn to sub-sample and count preserved marine plankton.
- Students will use a taxonomic picture key to identify plankton under microscope.
- Students will compare and contrast temporal and tidal regime samples.
- Students will assess species richness and species evenness and estimate species diversity in the plankton samples.

Field Sampling

Why do we need to sample? When we want to learn about specific plants and animals, a good place to begin are to determine if the species under study is present, how many are present in a given area (density) and how that density compares to other species in the same area (evenness). A complete count of all the individuals in a plot (a census) would answer this question. However, it is usually impossible count every individual present unless those individuals are quite large. Even then, counting every individual poses real problems. Instead, scientists usually select representative sample sites in the area they need to survey, and follow rules to obtain unbiased samples, samples truly representative of the area.

How often are samples taken? Researchers sample sites for different periods of time or in different intervals. The URI Graduate School of Oceanography has been sampling the plankton in Narragansett Bay every week since the 1950s. This makes this plankton data set one of the longest running continuous sample sites in existence.

We will be looking at recent weekly samples taken from the long-term plankton sampling site. We will be subsampling and counting these samples, testing hypotheses about temporal and tidal differences in the samples.

How do we identify the plankton species? In the lab, you will have access to a taxonomic picture key. It should help you identify the plankton into major groups. Please remember, your TA does not have "the right answer" to identifying all the organisms. Use the picture keys to identify, placing plants in the group it seems the best fit. Because of limited time and your limited expertise in the field of plankton taxonomy, all the similar looking members of a group we will call a single species. Then we will use our "species numbers" for our data analysis.

How will we sample plankton diversity in the field? We will subsample and count plankton samples collected under four different conditions that will be discussed by your TA. (Samples collected for each class may vary. Your TA will describe your samples in detail.) Using these samples, we will determine A) what plankton species groups live in each environment, B) how many of each live in each environment, and C) how those population numbers compare between species. With these data, we can compare species diversity in the four samples. We will write and test hypotheses about the species found in the environments.

For each sample, working in groups of 2-3 students, you will assess the plankton species groups. Using the Sedgewick Rafter counting cell, you will count all the plankton in THREE of the gridded squares. If you have more than 100 plankton in your squares, then once you complete that square, you can stop counting.

What do we do with our counts of the plankton species groups? Using the numbers of plankton group, we will count the plankton groups to determine three quantifiable parameters that assess the species diversity: species evenness, species richness and species diversity. Species evenness is the relative abundance of each species per unit area. Species richness is the number of different species per unit area. We will use these two counts to make estimates about species diversity. Species diversity is calculated from species richness (the number of different species per unit area) divided by species evenness (the number of each species per unit area).

What do these terms really mean? The following example (from http://www.countrysideinfo.co.uk) illustrates the meaning of these terms. Biological diversity can be quantified in different ways but the main factors that influence diversity are richness and evenness.

SPECIES RICHNESS is easy, the more species present in a sample, the 'richer' the sample. Species richness gives as much weight to those species represented by very few individuals as to those which have many individuals in the sample. Thus, in a field, one daisy has as much influence on the richness of an area as 1000 buttercups.

SPECIES EVENNESS is a measure of the relative abundance of the different species in the sample. So a sample has more evenness if all the species that are in the sample occur in the same number. In the example below, sample 1 is more even, sample 2 is less even.

SPECIES DIVERSITY describes a combination of richness and evenness. Species diversity values give scientists a summative method of comparing diversity in samples.

Look at this following example from www.countrysideinfo.co.uk.

In this example, we sampled two different fields for wildflowers (Table 1). The sample from the first field consists of 300 daisies, 335 dandelions and 365 buttercups. The sample from the second field comprises 20 daisies, 49 dandelions and 931 buttercups (see the table below). Both samples have the same richness (3 species) and the same total number of individuals (1000). However, the first sample has more evenness than the second. This is because the total number of individuals in the sample is quite evenly distributed between the three species. In the second sample, most of the individuals are buttercups, with only a few daisies and dandelions present. Sample 2 is therefore considered to be less diverse than sample 1.

Table 1. Wildflower diversity of two fields.

	Numbers of individuals	
Flower Species	Sample 1	Sample 2
Daisy	300	20
Dandelion	335	49
Buttercup	365	931
Total	1000	1000

A community dominated by one or two species is considered to be less diverse than one in which several different species have a similar abundance.

As species richness and species evenness increase, then species diversity increases. There are formulas designed to measure of diversity which accounts for both richness and evenness. The Simpson's Diversity Indices are used by many scientists to compare species diversity.

We will use the formula:

$$D = \frac{\Sigma\,(n(n-1))}{N(N-1)}$$

where: n = the number of organisms of a single species and
N = the total number of organisms of all species in that sample.

We will then subtract D from 1 to get: Simpson's Index of Diversity = 1 – D

Simpson's Index of Diversity is a value that ranges between 0 and 1; the greater the value, the greater the sample diversity. In this case, the index represents the probability that two individuals randomly selected from a sample will belong to different species.

0← (less diverse) ← —— → (more diverse) **→1**

Procedure

Working in groups of 2-3 people, every group will count four plankton tow samples. Your TA will describe where the plankton tows were collected and what we can compare. Your TA will help you with taking a subsample and setting up your counting cell.

1) The TA will help each group collect a subsample from the four plankton tows.
2) Place the filled Sedgewick-Rafter gridded counting cell onto your microscope. Follow TA procedures for filling the cell.
3) Count at least THREE gridded squares. Count more squares until you have counted 100 plankton. Remember that plankton in chain are INDIVIDUALS, so you need to count EACH INDIVIDUAL in that chain.
4) Using the picture key, identify and record the plankton species groups within your counting cell squares.

5) Once you have counted all four plankton samples, summarize your data into Diversity Index calculation tables..

6) Count the number of different "species" you counted. (Remember, we are using our plankton groups as species.) This is your SPECIES RICHNESS value.

7) List all the plankton "species" and how many times you counted each species (= n).

8) Calculate n (# of a single species) and N (total # organisms of ALL species in sample.)

   The Greek symbol sigma, "∑," in mathematical terms means to find the "sum of the following terms."

   Remember: n = # organisms of a single species N = total # organisms (of all species)

9) Now substitute into the formula:
   Simpson's Index: $D = \frac{\sum (n(n-1))}{N(N-1)}$

10) Lastly solve for Simpson's Index of Diversity = (1-D) = (1 — ____.)
    Take your value of D and subtract it from 1.
    This is your Simpson's Diversity Index; closer to 1 is more diverse, closer to 0 is not diverse.

What does the Simpson's Diversity Index value mean??
"Simpson's Diversity Index" (DI) is a standardized, mathematical way of describing the variety of plankton species groups in your samples. We calculated the D.I.to mathematically compare different samples.
The Simpson's D.I. value is always between 0 and 1.

> THE CLOSER YOUR SIMPSON'S D.I. VALUE IS TO ONE,
> THE GREATER YOUR SAMPLE DIVERSITY.

Data Sheets

Group members: _____

Before beginning the plankton counts, groups should write out their null and alternate hypotheses. The unfinished sentences below can help you.

Once your group has written your testable hypotheses, then you should get your subsamples of the net tows and start counting the plankton.

Null hypothesis
Ho: There is no difference in species richness/ species diversity between

_____ sample and _____ sample.

Alternate hypothesis
HA: There is difference in [species richness/ species diversity] between

_____ sample and _____ sample.

The _____ sample has MORE / LESS species richness/ species diversity than the

_____ sample.

Data (on next page) shows counts of plankton in the net tows, recording the number of gridded cells counted. (All phytoplankton and zooplankton should fit into these groups. IF there are dramatically different plankton in the sample, then the TA will help the class to create a new category which all will use.)

Counts show number of cells in each "plankton group", counting three squares on Sedgewick-Rafter counting cells for each sample.

Table 2. Counts of number of plankton cells at two sites, two tidal flows.

		Four Plankton Net Tow Samples, each sample has 3 SR counting squares reported (under A, B, & C)											
	Plankton groups	GSO Dock Incoming Tide			GSO Dock Outgoing Tide			Fox Island Incoming Tide			Fox Island Outgoing Tide		
		A	B	C	A	B	C	A	B	C	A	B	C
Phytoplankton	Chains WITH spines												
	Chains NO spines												
	Needle-shaped												
	Pill boxes												
	Ceratium												
	Boat shaped												
	Polyhedron												
	Round												
	Pointy diamonds												
Zooplankton	Zooplankton (Copepods)												
	(Other)												

In your group, calculate a Simpson's diversity index for all four samples. After calculating diversity index (1-D), share data with class. Lastly, determine if there is a difference between samples.

Table 3. Simpson Diversity Indices for GSO dock samples**.**

GSO Dock Incoming tide — Species richness: _____

n = # of a single species
N = total # organisms (of ALL species) in sample.

LIST plankton species (we will use plankton groups as species)	Number counted (n) in 1 m	(n – 1)	n(n – 1)
example: Volvox	4	3	12
Total (N) =		∑ (n (n - 1))=	
(N - 1) =			
N(N - 1) =			

Simpson's Index: D = $\frac{\sum n(n-1)}{N(N-1)}$

Simpson's Index: D = _____

Simpson's Index of Diversity
1 - D = _____

@GSO Dock, incoming tide

GSO Dock Outgoing tide — Species richness: _____

n = # of a single species
N = total # organisms (of ALL species) in sample.

LIST plankton species (we will use plankton groups as species)	Number counted (n) in 1 m	(n – 1)	n(n – 1)
Total (N) =		∑ (n (n - 1))	
(N - 1) =			
N(N - 1) =			

Simpson's Index: D = $\frac{\sum n(n-1)}{N(N-1)}$

Simpson's Index: D = _____

Simpson's Index of Diversity
1 - D = _____

@GSO Dock, outgoing tide

In your group, calculate a Simpson's diversity index for all four samples. After calculating diversity index (1-D), share data with class. Lastly, determine if there is a difference between samples.

Table 4. Simpson Diversity Indices for Fox Island samples.

Fox Island Incoming tide	Species richness: _____	Fox Island Outgoing tide	Species richness: _____

n = # of a single species
N = total # organisms (of ALL species) in sample.

n = # of a single species
N = total # organisms (of ALL species) in sample.

LIST plankton species (we will use plankton groups as species)	Number counted (n) in 1 m	(n – 1)	n(n – 1)	LIST plankton species (we will use plankton groups as species)	Number counted (n) in 1 m	(n – 1)	n(n – 1)
example: Volvox	4	3	12				
Total (N) =		∑ (n (n - 1))		Total (N) =		∑ (n (n - 1))	
(N - 1) =				(N - 1) =			
N(N - 1) =				N(N - 1) =			

Simpson's Index: D = $\frac{\sum n(n-1)}{N(N-1)}$

Simpson's Index: D = _____

Simpson's Index of Diversity
1 - D = _____

@Fox Island, incoming tide

Simpson's Index: D = $\frac{\sum n(n-1)}{N(N-1)}$

Simpson's Index: D = _____

Simpson's Index of Diversity
1 - D = _____

@Fox Island, outgoing tide

Table 5. Class Data – Reporting Simpson's Index of Diversity values.

	Species RICHNESS				Species DIVERSITY			
	= How MANY "species groups" in sample				1 = high diversity, 0 = no diversity			
Class groups	GSO Dock		Fox Island		GSO Dock		Fox Island	
	Incoming Tide	Outgoing Tide	Incoming Tide	Outgoing Tide	Incoming Tide	Outgoing Tide	Incoming Tide	Outgoing Tide
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
Average								
StDev								

Questions to Think About:

1. Which sample had the greatest species diversity as reported using the Simpson's Index of Diversity? Why do you think that sample had the highest diversity?

2. Did you expect that sample to have the greatest diversity? Why or why not?

3. List the most abundant species in the data set. Why do you think it was most abundant?

4. (Thought question…) Do you think that the Simpson's Index of Diversity gives a good indication of the species present or not? WHY??

For more information on species diversity, try the following sites that were used to make this lab:
http://evolution.berkeley.edu, and http://www.countrysideinfo.co.uk.

Using Sedgewick-Rafter Counting Cells

A Sedgewick-Rafter(SR) counting cell is a slide made with a frame that holds a 1mL liquid sample, when sealed with the cover glass. The cell allows a count of plankton in a 1mL volume of water. The slide has a grid pattern of 1mmx1mmm, so each square holds 1uL of sample.

Figure 1. Sedgewick-Rafter Counting Cell

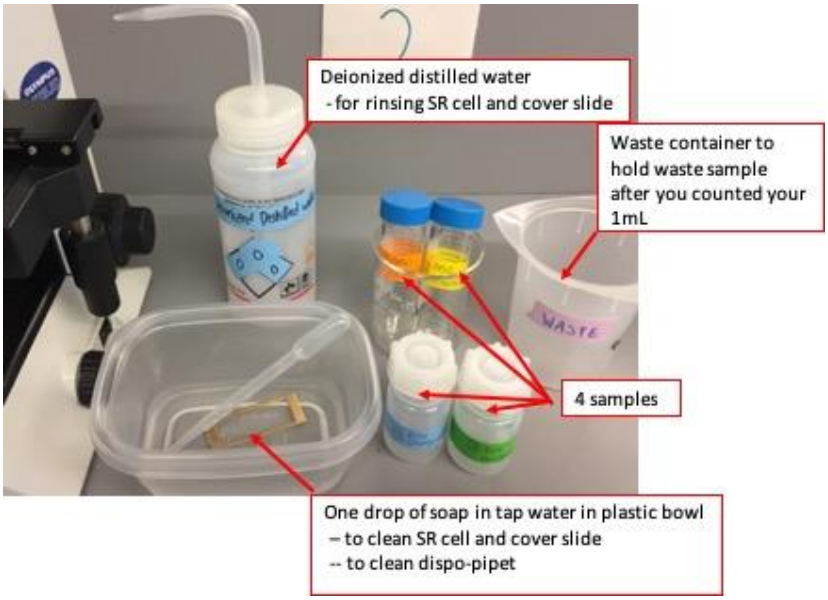


Figure 2. Materials needed for plankton counts on SR slides

To Load Sample

Put SR slide on ¼ piece paper towel. Invert sample bottle 3x to mix. Quickly, pipet out 1mL. With cover slip slightly offset and disposable pipet held at angle, fill the slide with 1mL of sample. Close lid.

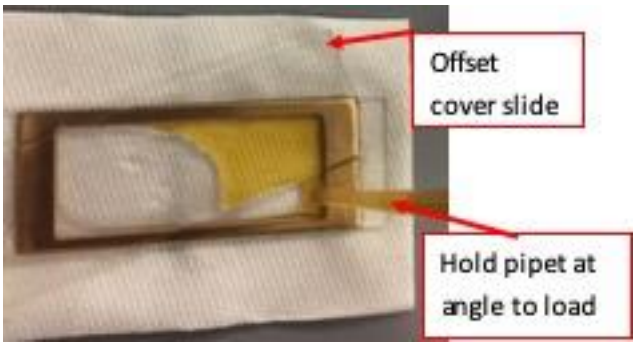


Figure 3. Applying sample to SR slide

To Count Sample

Place slide on microscope stage. Using the lowest power objective focus. You should be able to see the grid on the slide. Do not drop the stage. Change to higher,10- 20x objective.

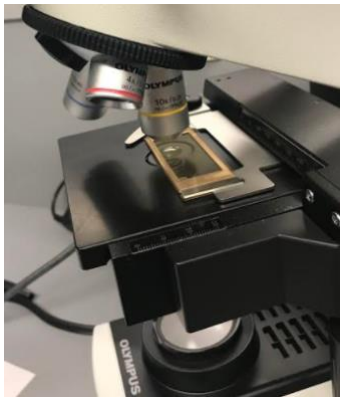


Figure 4. Mounting slide on microscope

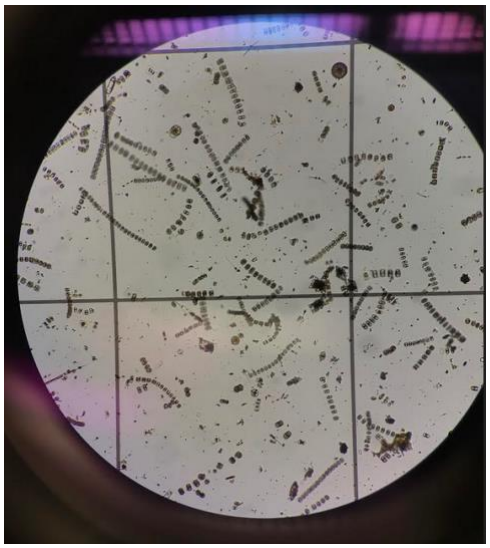


Figure 5. Example of what may be seen under 20X objective

Counting Plankton:

Within a square, identify and count all plankton within the square. Use the picture key provided to help you identify the plankton on the slide.

Count ALL the plankton in the square.
In chains, count each individual cell.

Count TEN complete squares for the sample.

After the Sample Is Counted

Carefully pour your 1mL sample into the waste container. Place SR slide and cover slide into soapy water dish to wash. Rinse SR slide and covers slide with DI water (into soapy water dish).DO NOT EVER touch the gridded portion of the cell, not even with paper towels. It will scratch and become unusable.
Dry the bottom of SR cell and both sides of the cover slide.
Load you next sample and continue counting.

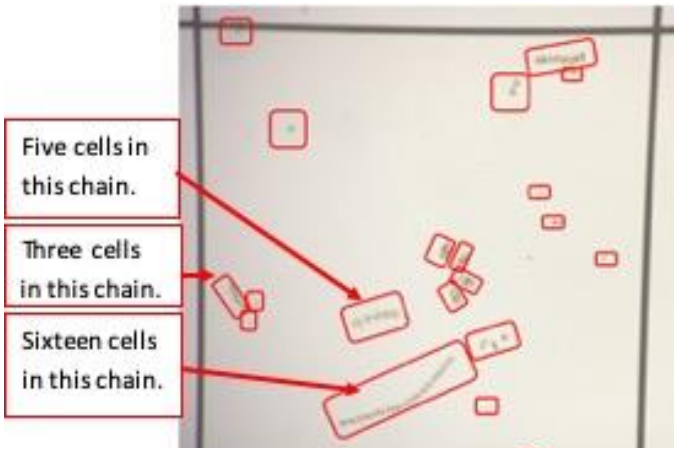


Figure 6. Example of how to count plankton cells on SR slides.

Lab 1B: Graphing Species Diversity Using an Analytical Graphing Program - Learning to Code in R and RStudio

Objectives:

- Learn the basics of data analysis programs R and RStudio
- Use R to create graphs of Narragansett Bay plankton sample data
- Test hypotheses of species richness and species diversity on plankton samples

Goals for Lab:

- Import data into R
- Save data in R as variables
- Plot diversity data as a scatterplot and a boxplot
- Plot certain parts of your data

1. Getting Started

You will be using R for data analysis this semester. R is a versatile, free data handling software widely-used by scientists. R is great for organizing data, analyzing data, and presenting data.

Here is some basic vocabulary you will need when using R.

- R script: the lines of code that you are writing (filename.R)
- R project: your R script, any variables you have created, and your current R
- environment (filename.Rproj)
- Variable: a way to store your data for use in the R script. For example, if you type x=3 into R, then R now stores x (the variable) as the number 3.
- Package: a set of functions/codes that you can load into your script (examples - gsheet, ggplot2)
- Function: an action or calculation that you perform on your variables (examples - gsheet2tbl, subset, mean, library)
- Comment: does not run as code, starts with #. You use comments to keep track of what your code is doing

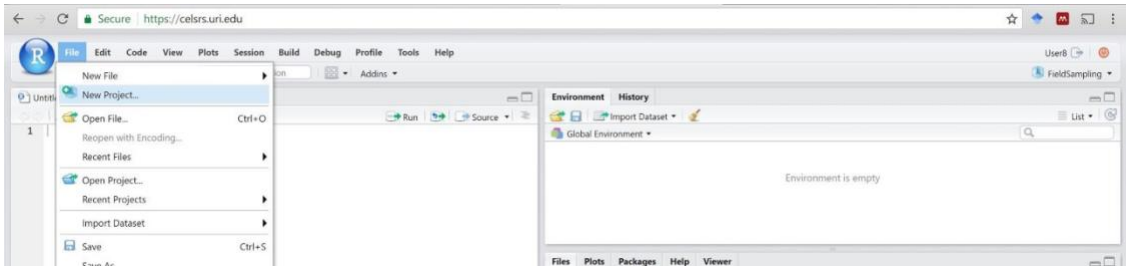
More information on R is at http://www.datacarpentry.org/R-ecology-lesson/00-before- we-start.html#why_learn_r.

TO START: Log in to the R Studio server

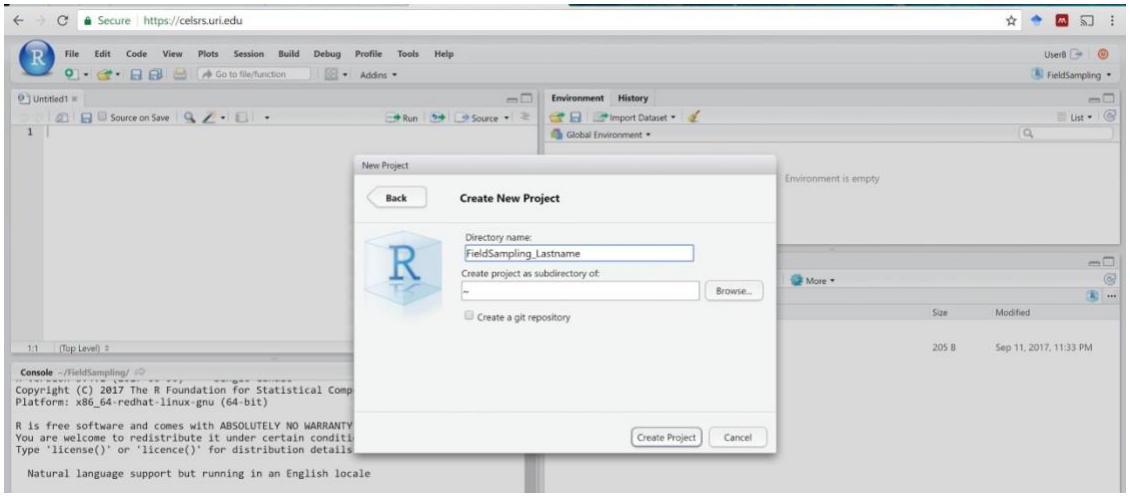
Open RStudio and make a new project.

Select from top left corner:

- File> New Project...
- New Directory and Empty Project

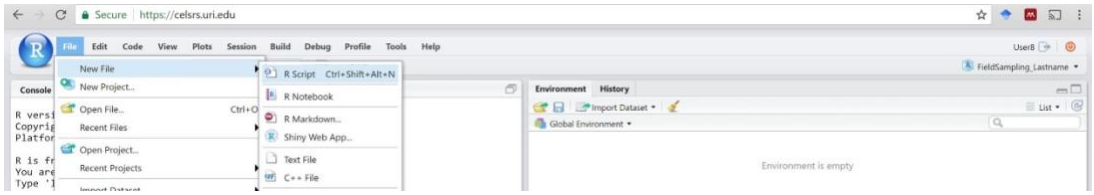


- Directory name: FieldSampling
- Leave the Subdirectory as the default: ~



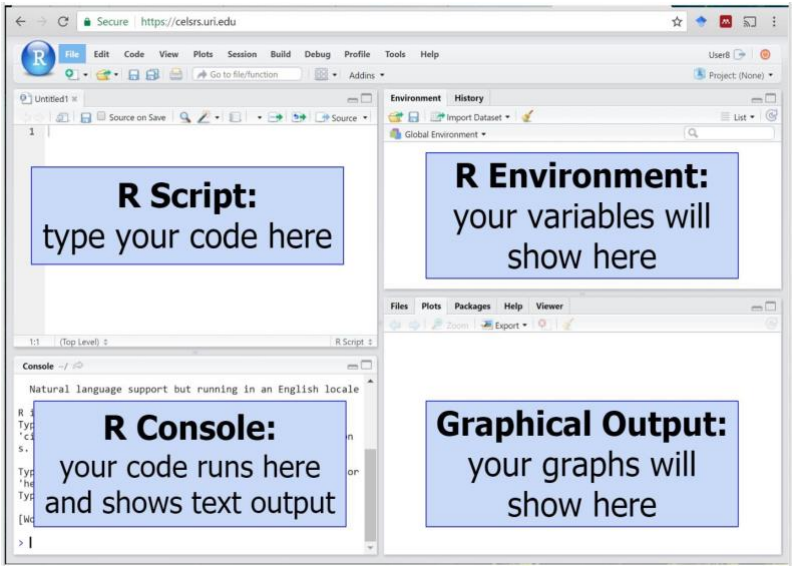
Select from top left corner:

- File> New File> R Script



Creating a new project and script will give you four windows in R:

- R script: where you type your comments and code, and save your work
- R environment: where all your variables show up and are organized
- R console: where the code runs and the text output shows. Errors show here too!
- Graphical output: where your graphs will show.



2. How to Get Your Data into R

Last week you calculated the Simpson diversity index for the plankton samples at each of four site types (GSO outgoing, GSO incoming, FOX outgoing, and FOX incoming). A diversity index allows you to understand the variation in diversity among site types. The following lines of code will import your data (the class data about the diversity of each plot) directly into R from Google Sheets.

a. Load the Packages You Need

The tools you need are found in "packages". Packages are like books from the library OR apps on your phone: packages are pre- written bundles of code that can perform the tasks we want. They contain functions and commands to perform analyses.

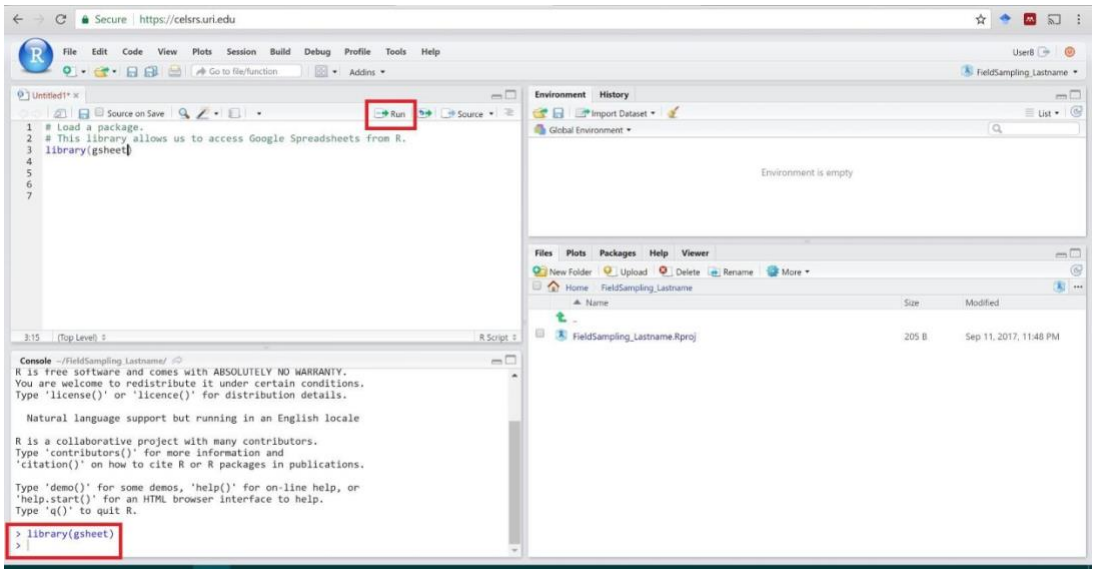
Here, we will load the gsheet package.
Type the code (from the grey boxes) into the SCRIPT window of RStudio (top left). With your cursor on the line of code, click Run (top right of the SCRIPT window in RStudio). The line of code will show up in the R console window once it is run (bottom left).

Use the # symbol to add comments to your script as you enter the commands below.

FIRST line of your code should be:

```
# MyFullName, myTAname  = learning how to make a boxplot in RStudio
# Always include hashtag comments!
# Start code by opening appropriate library packages
```

```
library(gsheet)
```



b. Tell R Where Your Data Is

- Assign the link to your diversity data to the variable "url". Note: “url” stands for Uniform Resource Locator. It is the address of a site on the web.
- You store a number or word or data to a variable using <- .

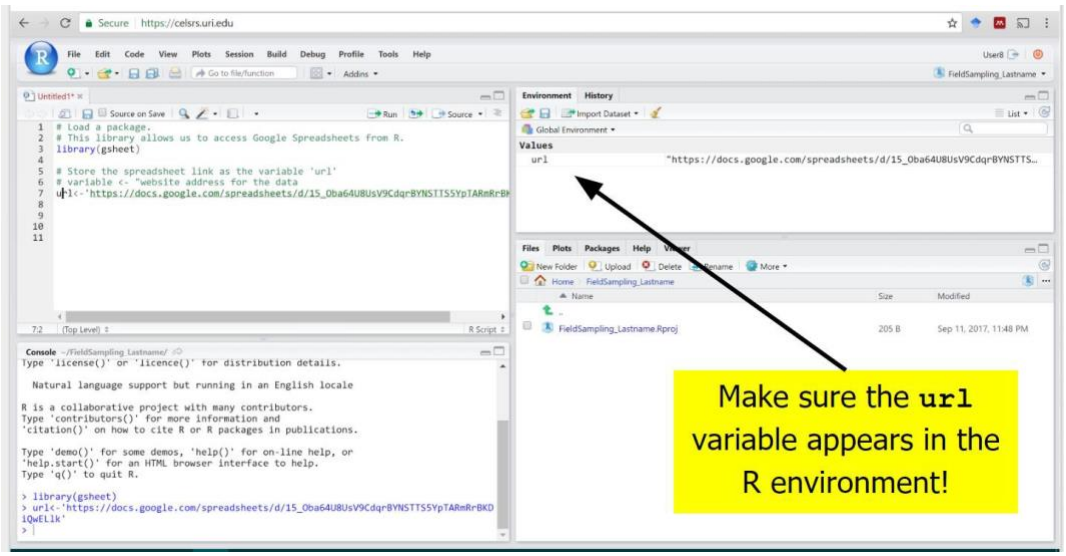
- The variable name goes on the left of the arrow and the information goes on the right.
- Use the link for your section. Find the UPDATED links on the homepage of the BIO104 Sakai site.

EXAMPLE:

Section 21: https://drive.google.com/open?id=1a2gjbssmfXe9rA57gj8pftmQq_XjJh9pDhHebe_bTpI

Replace the link below with your url using following command. Use 'single quotes' !
url <-'https://docs.google.com/spreadsheets/d/1mTQPw3GsqID_ay6YqdNISOOKIMb9HHeBpFu7EWFiGj8/edit#gid=0'

Always click Run with your cursor on the line of code. Once you have Run this command, make sure your new variable shows up in the R environment window (top right).

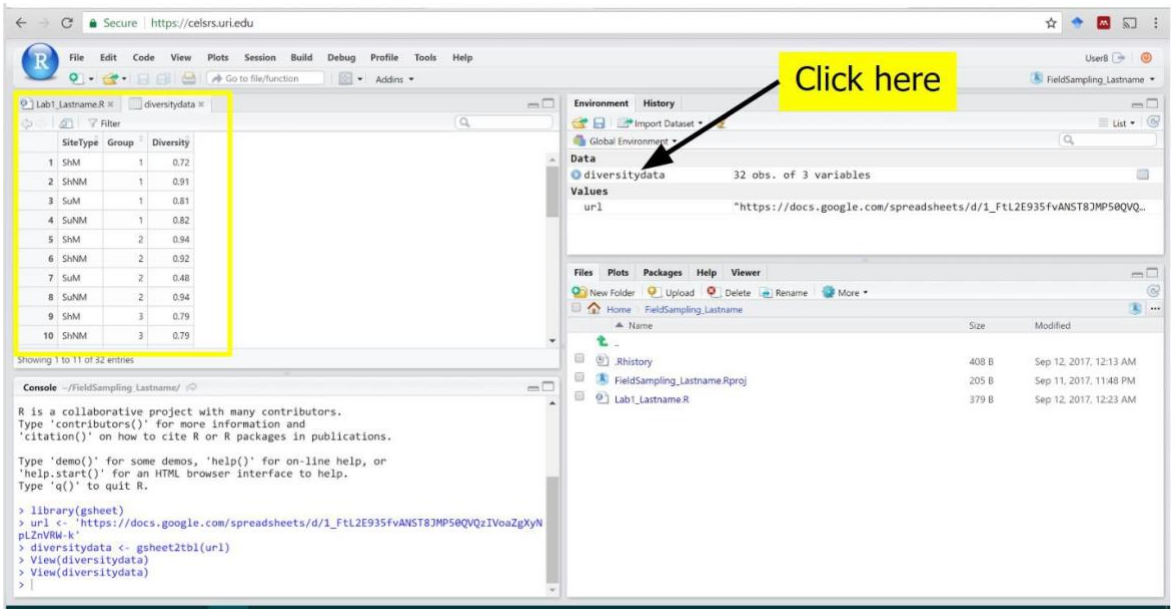


c. Load the Data and Store It as a Variable

- Load the data from the spreadsheet (now stored as variable url).
- Use the function *gsheet2tbl* to get the data from the website for use in R. This function is from the *gsheets* package.
- Store the data as variable diversitydata.
- diversitydata <- gsheet2tbl(url)
- The line was run correctly if you get the prompt (>) in the R console (bottom left)
- Make sure the diversitydata variable shows up in the R environment (top right)

d. Explore your Data

Click on the diversitydatavariable in the R environment. The data stored in diversitydatawill appear in the R script window (top left). It should look just like your spreadsheet data!



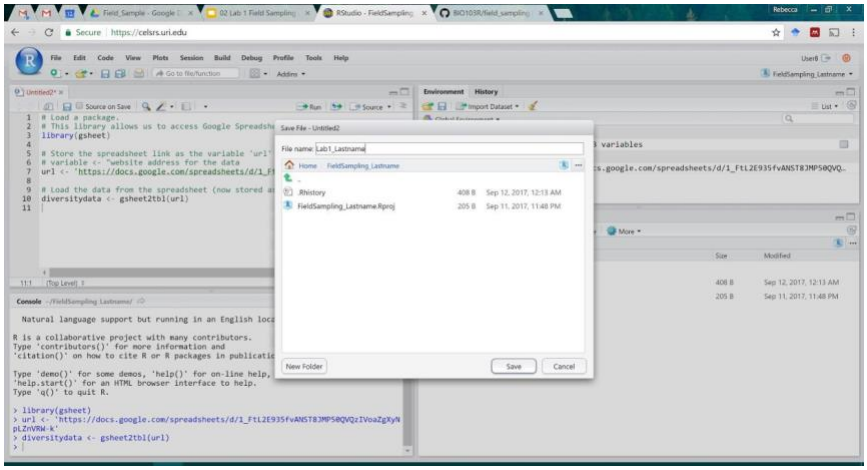
3. Keep Track of What Your Commands Do == use

As you work, you are typing commands that you probably won't remember next week. Use the #symbol to explain your lines of code. Anything that you type after the #will not run in your code.

```
#load the package that allows us to access google sheets
library(gsheet)

#assign the website address for the data to a variable using an arrow <-
# "variable" <- 'website address for data'
url <-'https://docs.google.com/spreadsheets/d/1mTQPw3GsqID_ay6YqdNISOOKIMb9HHeBpFu7EWFiGj8/edit#gid=0'


#load the data from google sheets and store it as the variable diversitydata
# "variable" <- "command to work on"(data from last line)
diversitydata <- gsheet2tbl(url)
```

SAVE your work. Save your code by clicking File > Save.

Name your file: Lab1_YourName.R Remember to put ".R" at the end!
Your work will save to the server that we are using for this lab.
You will be able to access your work on this server from other computers,
by using the web address (https://celsrs.uri.edu/), your username and password.



4. Plot Your Data

a. Load the Plotting Library

```
# Load the package that allows us to plot data
library(ggplot2)
```

Make sure you click Run after typing in each line of code.

b. Create the Base Layer of Your Plot

Use the command ggplot: diversitydata is the name of the variable containing your data

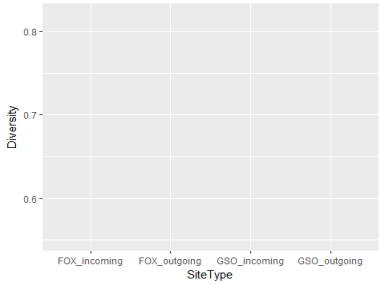
aes tells ggplot what the plot will look like ("aesthetics") including the x and y variables

SiteType is the column name from the data table describing the site locations and is the column of data you want to plot on the x-axis.

Diversity is the column name from the data table listing the Simpson Diversity index values and is the name of the column of data you want to plot on the y-axis.

Run this ggplot command. The graph will show up in the Graphical Output window (bottom right)

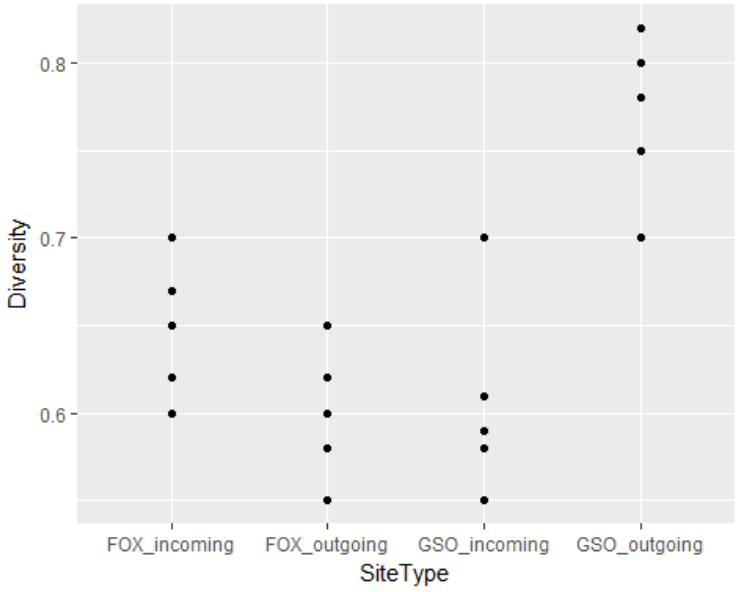
```
# Create a plot
ggplot(diversitydata, aes(x=SiteType,y=Diversity))
```



c. Add Data Points to Your Plot

Use the geom_point() command to add points to your plot.
Put a +between the ggplot command and geom_point()

```
# Add points to the plot using geom_point()
ggplot(diversitydata, aes(x=SiteType,y=Diversity)) + geom_point()
```

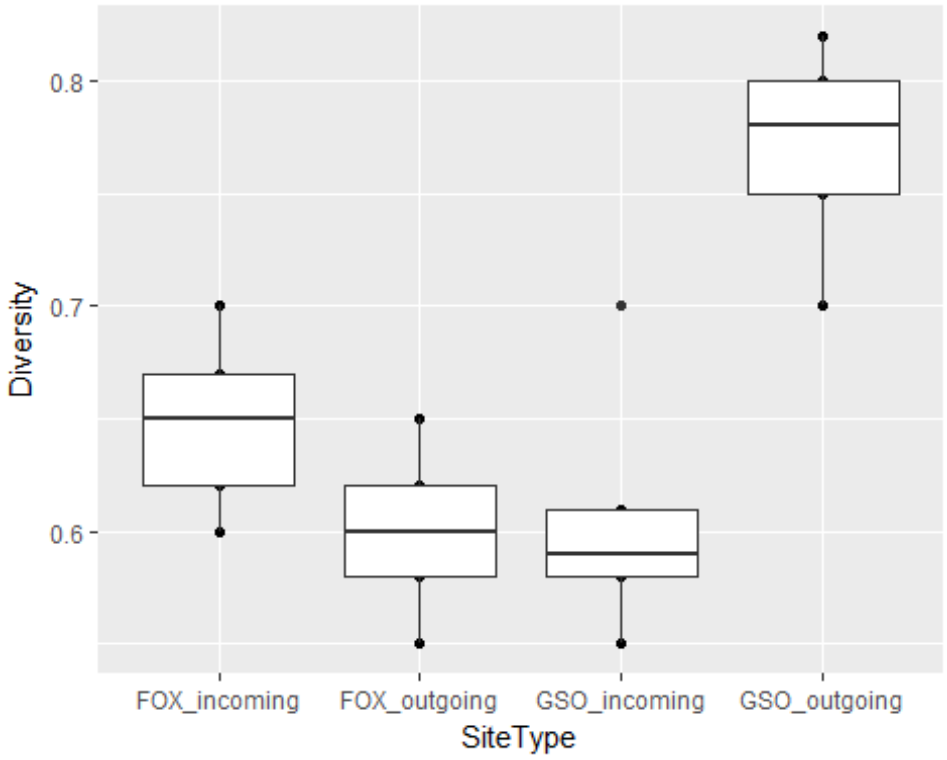


d. Add A Boxplot on Top of Your Plot

- A boxplot summarizes your data by plotting:
  - the median (middle) value as a horizontal line
  - 50% of the data within the box
  - the remaining top and bottom 25% of the data as lines above and below the box.
  - A boxplot based on the numbers 1, 2, 3, 4, 4 and 5 would have the middle line located at 3.5 (the median), the upper part of the box at 4 and the lower part of the box at 3.
- The greater the range of the box plot, the greater the variation of your data!

- Add a boxplot using the geom_boxplot() command
- Put a + between the geom_point()command and geom_boxplot()
- The + must come at the end of a line or the command will end and not plot the layer

```
# Add boxplots to the plot using geom_boxplot()
ggplot(diversitydata, aes(x=SiteType,y=Diversity)) + geom_point() + geom_boxplot()
```



Remember to add descriptive comments to your script!
Note: the order of the plotting instructions tells R to make:

- The base plot
- The data points
- The boxplots on top of the data points

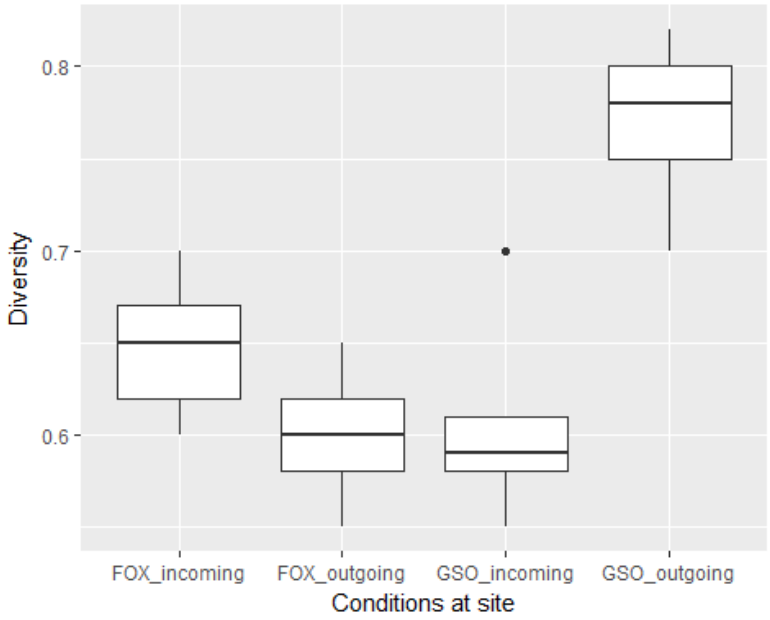
5. Edit Your Boxplots

a. Add Axis Labels to Your Plot

The variable Diversityis an integer (a numerical value), which means it is on a continuous scale; thus, the functions to label your y axis is continuous (scale_y_continuous). Other examples of continuous variables are time, weight, or date since they are numerical integers. We use a discrete scale when plotting variables like Locationor Tide or Color, where each variable is its own category (scale_x_discrete).

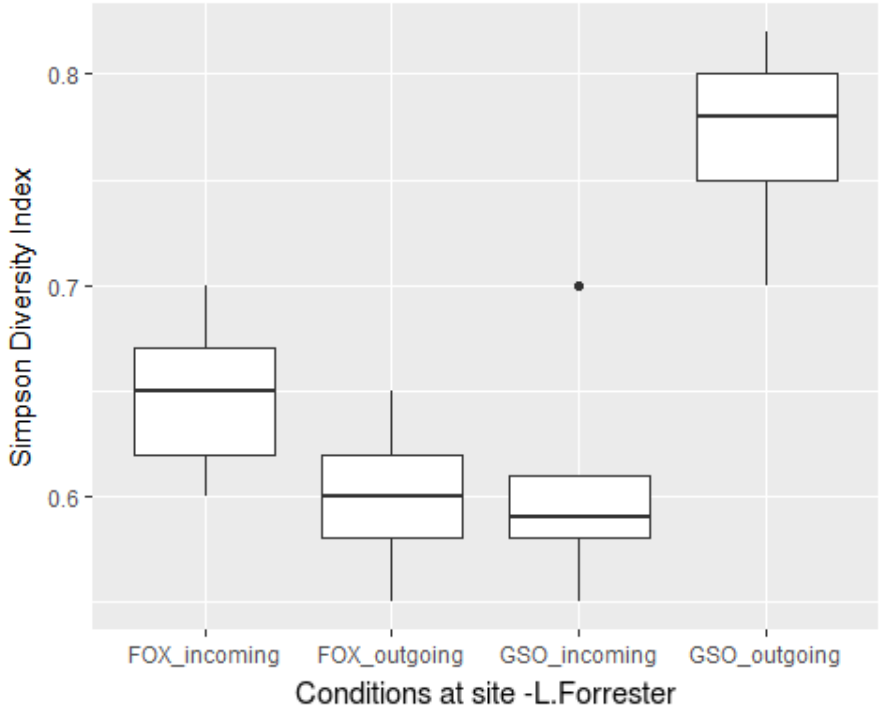
Use scale_x_discrete() function and the name =function to label your x-axis "Conditions at site".

```
# Plot just the boxplots and change the x-axis label
ggplot(diversitydata, aes(x=SiteType, y=Diversity)) + geom_boxplot() + scale_x_discrete(name ="Conditions at site (MyName)")
```



Use scale_y_continuous to label your y-axis correctly.

```
# Plot just the boxplots and change the x-axis and y-axis labels
ggplot(diversitydata, aes(x=SiteType,y=Diversity)) + geom_boxplot() + scale_x_discrete(name ="Conditions at site - MyName") + scale_y_continuous(name="Simpson Diversity Index")
```



b. Save Your Boxplot as an Image

- Click on Plots > Export > Save as Image...
- Name your file (Lab1_Boxplot1) and click Save.
- To view your saved images, go to File > Open Files, then click on the image you want.
- Open the image to copy/paste it into your lab report (written in Word or GoogleDocs).

c. Look at a Subset of Data

If you want to only show data from the two FOX Island sites, then make a new variable containing just these data by filtering your larger dataset.

First load the dplyrlibrary, which contains a filtering function.

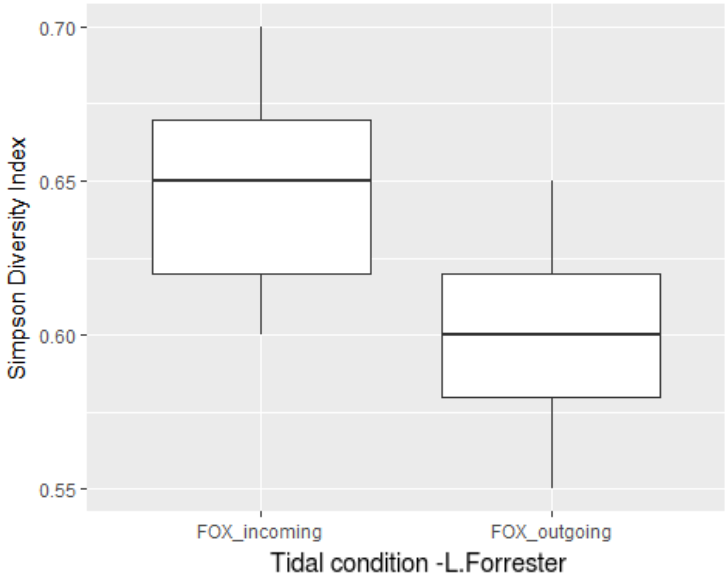
```
# Load the dplyr library
library(dplyr)
```

Then filterthe data to include SiteTypes FOX_outgoing and FOX_incoming (the vertical bar means "or")

```
# Create a new variable named FOXonly with just the FOX_incoming and FOX_outgoing data
FOXonly <- filter(diversitydata, SiteType == "FOX_incoming" |
SiteType=="FOX_outgoing")
```

Now plot this new dataset using boxplots.

```
# Plot the FOXonly data with boxplots and changed axis labels
ggplot(FOXonly, aes(x=SiteType, y=Diversity)) + geom_boxplot() + scale_x_discrete(name ="Tidal condition – MyName") + scale_y_continuous(name = "Simpson Diversity Index")
```



For more explanations of lines of code, read the **R Cookbook.**

Turn in this final boxplot of the data in your lab report. (Click in the Graphical Output pane:

Plots > Export > Save as Image... > Name your file)

6. Homework Assignment

Part 1. Graphing Incoming Vs. Outgoing Diversity at the GSO Sites

1. Subset your diversity data for just the GSO sites.
   Make a boxplot like you did in Step5c. Hint: create a new variable named GSOonly
2. Turn in your boxplots from both sites.
3. Write a summary to compare the incoming and outgoing sites at both locations.
   Which has a higher diversity? Please explain why one tide may have a higher diversity index value than the other. (2-3 sentences)
4. Which has a greater amount of variation in the data? How can you tell?
5. Include your summary table of your data formatted according to the lecture and poster in the BIO 104 lab.

Part 2. Graphing Long-Term Plankton Data

The samples you worked with in class are part of one of the world's oldest datasets of regularly collected plankton samples. These samples have been collected weekly since1957 at the Fox Island Station in Narragansett Bay, but only data from 1999 onward is available to the public. Find more information on this weekly plankton survey at https://web.uri.edu/plankton/data/.

In class, you calculated the diversity values of two samples from the Fox Island Station, and two samples collected from the GSO dock location. All groups in your lab counted and calculated the diversity values for these samples, allowing us to calculated diversity values for these two samples. To more thoroughly and accurately examine plankton diversity at the Fox Island station, you will analyze diversity data from this long-term dataset collected every week since 1999.

1. Make a boxplot comparing plankton diversity during incoming and outgoing tides at the Fox Island Site from all months in the long-term dataset, like you did in Step 4 (Hint: x=Tide).
2. Data are available at https://docs.google.com/spreadsheets/d/1RdEzJqLi1vRcpR80nH____ 0n7nJJ8gPZxoxxu HQqvtbWs/
3. Start with the commands:

   url2 <-
   'https://docs.google.com/spreadsheets/d/1RdEzJqLi1vRcpR80nH_0n7nJJ8gPZxoxxu HQqvtbWs/'
   FOXyearlydiversitydata <- gsheet2tbl(url2)
4. Select three months and make a diversity index boxplot comparing just those 3 months, like you did in Step 5c and Homework Part 1.
5. Hint: create a new variable named threemonthsonly and plot using x=Month.
6. Write a summary describing differences in diversity between tides and months at the Fox Island site (2-3 sentences). Discuss which of the months that you chose had the most variation (boxplot range) in diversity and highest median diversity value.

Lab Assignment 1 Report Submission

Part 1. Done in Lab

1. Individual Summary Table of your class' data
2. Boxplot of FOX Island diversity data (created in Step 5c)
3. Boxplot of GSO diversity data (homework Part 1)
4. Summary of diversity differences (2-3 sentences)

Part 2. Homework

1. Boxplot of all FOX Island long-term diversity data by tide
2. Boxplot of FOX Island long-term diversity data for 3 months
3. Summary of diversity differences in the three months you selected (2-3 sentences)

Lab 2A: Mechanisms of Evolution - Snail Populations

Introduction

The purpose of today's experiments is to examine how populations change over time when exposed to two different environmental situations: natural selection and genetic drift.

1. Experiment 1: Some individuals will survive and have higher reproductive success because of some trait(s) they have (color, size, shape, etc). If these beneficial traits have a genetic basis, they will be inherited by the next generation. This is called natural selection.

2. Experiment 2: Some individuals survive and reproduce better than others due to chance alone, and not because of any particular trait they might have. These surviving parents will pass on their traits to their offspring. This is called genetic drift. In a large population, genetic drift may have very little effect on the populations' traits. However, in a small population, genetic drift can quickly decrease the amount of genetic variation in a population.

   This lab will simulate these two major mechanisms of evolutionary change by comparing how a population responds to the two situations described above. At the end of the lab, you will analyze your results as well as the findings of the entire class.

Procedure Description

You will measure changes that occur within a population of snails. These snails inhabit the rocky ocean shoreline. The snails cling to the rocks, scraping algae (their food) from the rocks. These snails are exposed to <u>two main sources of mortality</u>:

A. Oystercatchers are large seabirds that hunt for snails visually. The color of the snails affects their survival when oystercatchers are present because some colors are more easily seen (and eaten) than others. Your TA may show you short video clips of oystercatchers foraging along a rocky shoreline.

B. Drifting logs are slammed into the rocks by waves and crush snails randomly. Snail color has no effect on whether it is crushed by a log or not.

The Scenario

Each group of two students will be given their own "tide pool" containing an assortment of "empty snail shells" and a number of "live snails." We will run two experiments on the snails in the tide pool, to look at changes in the population over generations.

Experiment 1- Oystercatcher: One person in each group will act as the oystercatcher (seabird) and will feed on the <u>live</u> snails, eating LIVE snails, counting the number consumed, allowing snail reproduction to occur, and then feeding on snails again, for three snail generations. We will examine what changes occur to the color of the snail population over time.

Experiment 2 – Drifting log: One person in each group will act as the drifting log, and will blindly crush the snail shells, of any color, living or dead. After each crushing event, the live snails will be allowed to reproduce. Then, the crushing of snails will occur again, for three snail generations. We will examine what changes occur to the live snail population over this time.

The reality in the lab: While some people might enjoy eating snails or crushing and killing all these snails, we try to run a more humane lab here at URI. Your "tide pool" will be represented by a white plastic bowl. Your "empty snail shells" will be represented by white and red colored beads. Your "live snails" will be represented by white and red colored beads with pale yellow threads (snail bodies) tied through the beads. Both live snails and empty snail shells look very similar on first glance.

Like real birds, you will have to examine the shells (beads) to see if they contain a live snail. With the white live snails (represented by yellow thread in white beads), the white or yellow thread against the white bead may be more difficult to see than yellow thread against the red beads. This difference may make the white live snails more difficult for the oystercatcher (you) to find. This may change the snail population (based on shell color) over time.

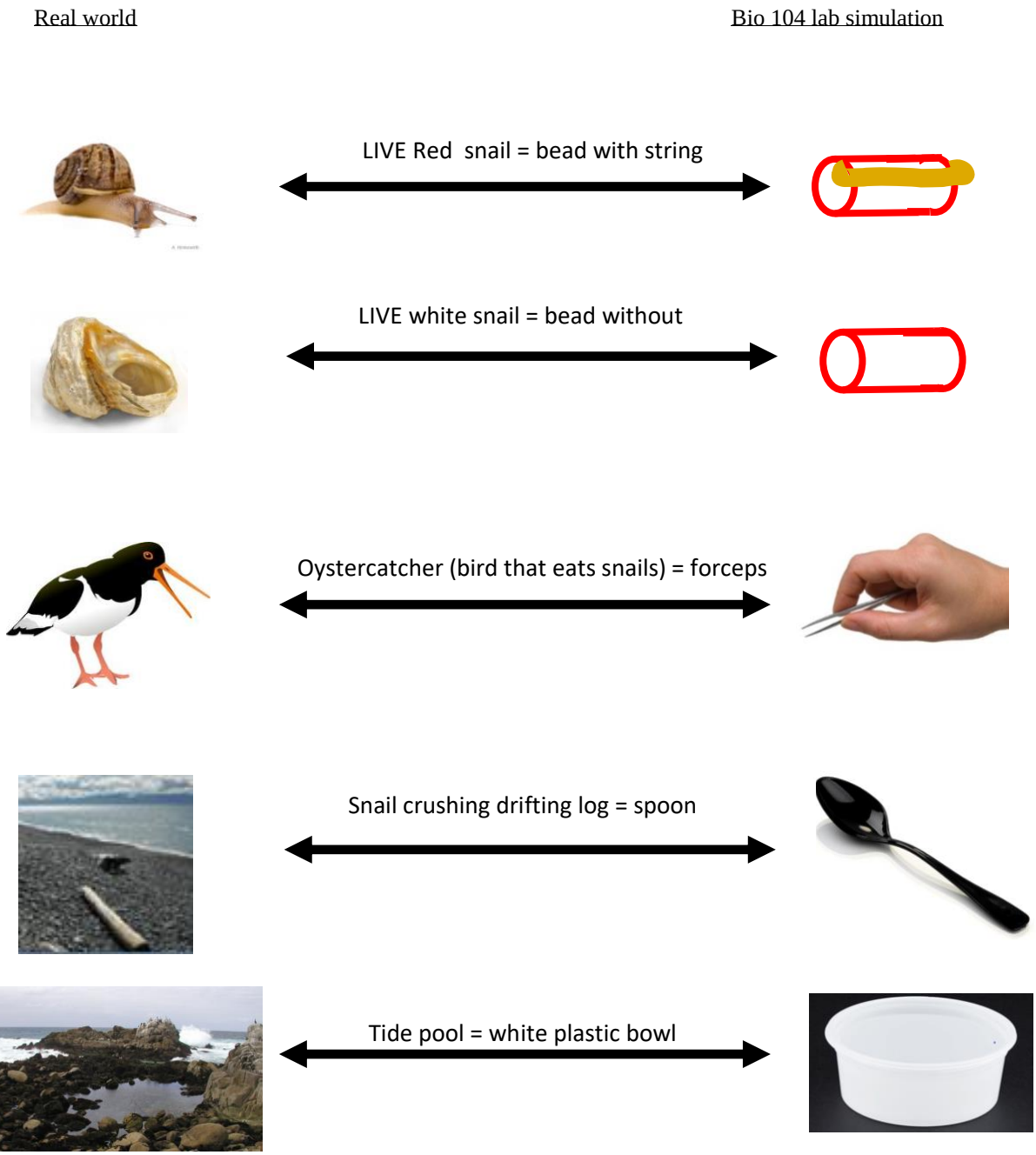


Figure 7. Snail photo: http://www.allwhitebackground.com/snail.htm,
Oystercatcher picture: https://www.onlinelabels.com/clip-art/oyster_catcher-105452.htm
Tidepool picture: https://nmsmontereybay.blob.core.windows.net, other pictures by authors.

Procedure

Start all 3 experiments with tide pool (white plastic bowl) containing:

25 live red snails (red beads, with strings)
25 live white snails (white beads, with strings)
25 red shells (red beads, no string)
25 white shells (white beads, no string)

This is your starting generation (G_0); record the number of live individuals of each color in the tables provided.

Experiment 1: Oystercatcher Experiment

As an oystercatcher bird, you will hunt for live snails visually. Eat snails until you have removed half of the live snails present. Do not remove empty shells.

1. Mortality Episode 1:
   Hunt and remove 25 live snails in your tide pool.
   Remove 25 live snails (beads WITH strings).
   Work as fast as possible.
   Do not remove empty shells from your tide pool; you cannot eat them so they remain constant.

2. Reproduction Episode 1:
   The 25 snails removed from your habitat are eaten, so they are dead.
   They cannot reproduce. The 25 individuals remaining in the tide pool survived the first episode of mortality, and they now can reproduce.
   For every surviving live snail, add 1 new snail of the same color.
   The new population has 50 live snails: 25 survivors and their 25 offspring – this is generation 1 (G_1). Record the number of individuals of each color present.

3. Repeat 2X (for a total of three mortality episodes and three bouts of reproduction).
   After each reproduction event, you will have 50 living snails in your tide pool.

Experiment 1. Prediction:								
Group Data: Oystercatcher experiment -- live snail counts								
	Generation 0		Generation 1		Generation 2		Generation 3	
snail color	Red	White	Red	White	Red	White	Red	White
START count	25	25						
End count								

Experiment 2: Drifting Log Experiment

As a drift log, you will kill half the snails in the tide pool.

1. Mortality Episode 1: With drift log (spoon), remove 25 live snails.
   If you remove empty shells, leave them out.

2. Reproduction Episode 1: The empty shells and live snails crushed and removed from your tide pool cannot reproduce. The live individuals remaining in the tide pool survived the first episode of mortality, and they now can reproduce.
   For every surviving live snail, add 1 new snail of the same color. Do not replace empty shells. This will be your first generation (G_1), Record the number of individuals of each color present in the experiment 2 data table.

3. Repeat 2X (for a total of three mortality episodes and three bouts of reproduction).
After each reproduction event, you will have 50 living snails in your tide pool.

Experiment 2. Prediction:								
Group Data: Drifting log experiment -- live snail counts								
	Generation 0		Generation 1		Generation 2		Generation 3	
snail color	Red	White	Red	White	Red	White	Red	White
START count	25	25						
End count								

Data Analysis

1. Record and share your data with the class. Use the numbers from the "Start" row from the data table above to represent the population size for each generation in your experiment for your group. Enter these numbers in the Google Sheet that is shared with the class.

2. After you have data from the class, you will be able to make a data table to hand in next week. This data table will have your data in a format that is most convenient for graphing.
   Below is an example table for oystercatcher data. You will want to produce something similar to hand in, as well as a second table showing your "Drifting log" data.

   Use computer formulas to calculate standard deviations for the class data set.

Table 6. Example of class data for Experiment 1. Oystercatchers

Experiment 1 Prediction:									
Oystercatcher Exp. Group Name		Generation 0		Generation 1		Generation 2		Generation 3	
		Red	White	Red	White	Red	White	Red	White
Group 1		25	25						
Group 2		25	25						
Group 3		25	25						
Group 4		25	25						
Group 5		25	25						
Group 6		25	25						
Group 7		25	25						
Group 8		25	25						
Group 9		25	25						
Group 10		25	25						
Group 11		25	25						
Group 12		25	25						

3. Graph your data as a line graph (use R) with generation number on the X-axis and 'number of snails' on the Y-axis. Follow the R lab manual (in back of this lab manual) for instructions. Your graphs will show results for the more obvious, red snails and one for the more cryptic, white-shelled snails. Summarize the consequences of mortality acting on your snail population over 3 generations.

4. Present results. List the similarities and differences among your results. Do the data collected by different oystercatcher and drifting log groups all look basically the same? Why or why not? We will use a t-test to determine if the differences among groups are large enough to be significant.

When writing your results, utilize these three main presentation approaches

1. Re-state the biological experiment and statistical test used:
   Example: "We compared red and white snail populations after three generations of predation from oyster catchers using a *t*-test."
2. Discuss whether you have significant differences between conditions:
   Example: "There was a significant (OR not a significant) difference in red snail populations (mean =___) and white snail populations (mean =___) after three generations of predation from oyster catchers; p = ___."
3. Describe your results in simple words:
   Example: "These results suggest that snail color really does have an effect (OR does not have an effect that we could detect) on snail survival."

5. Discussion. Compare the results of oystercatcher mortality with those of drifting logs on color of snail populations. Do these two mechanisms of mortality produce similar results?

Lab 2b: Mechanisms of Evolution - Snail Populations

Lab Goals

- Draw line graphs of evolutionary mechanisms in action.
- Create code that you will understand in the future (i.e. by including comments)
- Calculate mean and standard deviation (in R)
- Plot line graphs (in R)
- Compare two datasets statistically (using a *t*-test in R)

Before Starting with R, Sketch your Expected Results

- Drawing line graphs to represent evolutionary change.
    - Line graphs are useful to represent change in a variable over time. The pace of evolution is not determined by absolute time, but by generations. That why our x-axis for evolutionary change graphs will be 'generation.' We want to compare the population size between red and white snails over generations to illustrate mechanisms of evolution.
- Graph null hypothesis (Ho): Hardy-Weinberg equilibrium.
    - Start with a theoretical population of 100 snails. Assume equal frequency of white and red snails, so draw a graph starting with 50 white and 50 red snails.
      Draw a line graph showing what the frequency of each snail color will likely be over 5 generations if no evolution is occurring. This is our null hypothesis, which we call the Hardy-Weinberg equilibrium.
- Graph evolutionary mechanism in action (H_A.)
    - Start again with a population of 100 snails, again half white and half red snails at generation 0. This time, there is a predator (a bird called an Oystercatcher) that prefers to eat red snails to white snails. What mechanism of evolution is this an example of?
      Draw a line graph, showing your predictions for the frequency of each snail color over 5 generations if this mechanism of evolution were in action every generation.
- Write a 1-2 sentence description of how a mechanism of evolution is acting on color frequencies in the snail population.

1. Get Started – New R Project

Make a new R project in a new directory and open a new R script.

Begin by loading all the libraries you will need. In this case, we will be using ggplot2to make our graphs. We are also entering data into Google Sheets, so we will need gsheet to pull that data into R.

Copy the comment lines into your script

Enter the necessary code based on the previous lab and your R script from Lab A1.
If there are no code instructions then look at your Lab A1 materials

```
# MyFullName, myTAname  = learning linegraph & T-test in RStudio
# Load packages ggplot2, gsheet, and dplyr using the library command
```

SAVE your work. Save your code by clicking File > Save.
Name your file: Lab2LineGraphTtest_YourName.R

Remember to put ".R" at the end! Your work will save to the server that we are using for this lab. You will be able to access your work on this URI server from other computers.

2. Get Your Data into R

Use the same procedure as lab 1 to load data into R.

```
# Get data into R.  Store the Google sheets link as the variable url
# Load the data stored in the variable url
# use gsheet2tbl function and storing it as the variable snail_data
```

Once you have your variables, check to make sure the data imported correctly.
Click on the variables listed in the R Environment (top right) and compare them to the Google Sheet data. OR, look at the first few lines of your data using the head function:

```
head(snail_data)
```

These are the column names of your raw data. Use these to analyze and plot the data.

exp is the experiment number
group is your group number
generation is the number of generations from the beginning (0,1,2,3)
snailcolor is white or red
snails is the number of snails

You can also see the column names (i.e. the variables) by using the colnames function.

```
colnames(snail_data) ## [1] "exp"    "group"  "generation" "snailcolor" "snails"
```

3. Find the Averages and Standard Deviations of the Snail Populations over Time

We want to compare the population size (snails) between red and white snails (snailcolor), over generations (generation), and experiments (exp). To show these differences, we need to graph the mean of the snail populations over time. We do this by calculating the mean for each snail type at each generation for each experiment. To do this, we need to group the data into appropriate categories. Since we are looking for the mean snail population at each generation for each color for both experiments, we will group our data based on experiment / generation / color combination.

We will use the group_byfunction from the dplyrpackage to create a new variable where each subset of data is labeled. Tell group_by the name of the data you want to divide into subsets, followed by the columns you want to include.

```
# Group the snail_data variable by exp, generation, and snailcolor
# using the group_by function
grouped_snail_data <- group_by(snail_data, exp, generation, snailcolor)
```

a. Calculate the Mean for Each Group, and Put this into a New Variable Snail_Data_MeansUsing the Summarise Function.

Here the name of the column containing the mean number of snails (calculated using the function mean) is "mean".

```
# Calculate the mean number of snails for each group using the summarise function
# by giving the name of the data you want to summarize (`grouped_snail_data`)
# and that you want the mean value of the snail counts (`snails`)
snail_data_means <- summarise(grouped_snail_data, mean = mean(snails))
```

Click on the snail_data_means variable in the in the R environment (top right).

- What are your mean values for the data? Check the values make sense before continuing.

b. Calculate the Standard Deviation of the Means

The standard deviation is the average amount the individual data points differ from the overall mean. For example, if we saw red snail population totals of 3, 5, 5 and 6, we would have a low standard deviation since all the data is close to the mean. However, if we had red snail total of 0, 2, 12 and 16 the standard deviation would be high since all of the data is far from the mean.

Standard deviation is calculated in R using the function sd.

```
# Remake your table of means so it includes standard deviation
# and store the new table as the variable snail_data_means_sd
snail_data_means_sd <- summarise(grouped_snail_data,mean = mean(snails), stdev = sd(snails))
(All the above code goes on ONE line in R. It starts a new line on this printed copy only.)
```

Click on the snail_data_means_sd variable in the R environment.

- What are your standard deviation values for the data? Include your mean and SD values in your report.

4. Plot your Data from Experiment 1

a. Subset your Data from Experiment 1

We will use ggplotto graph the mean and standard deviation for Experiment 1. (Remember, ggplot2is a package you load and ggplot is the function you use for graphing).

First, filter your data variable that contains the mean and standard deviation (snail_data_means_sd) for just Experiment 1 (otherwise, you would graph both the data from all experiments, making a very confusing graph to look at).

Assign the filtered data to a new variable named snail_data_means_sd_exp1.

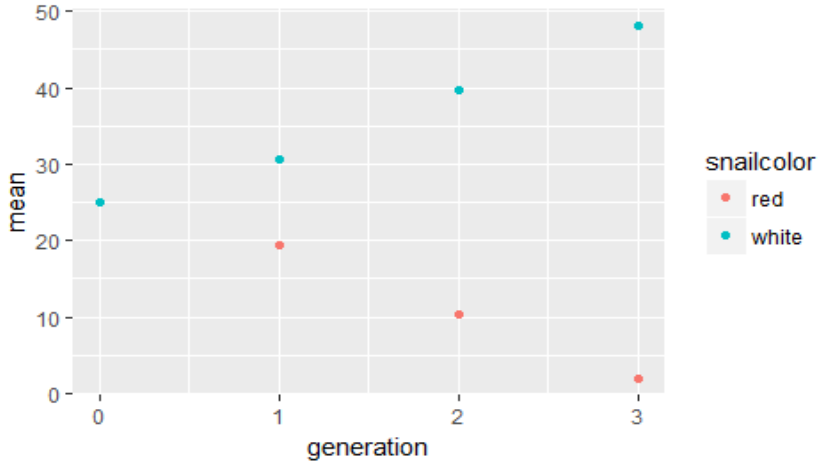
```
# Filter data to include only means for Experiment 1 (exp==1).
# Refer to Lab A1, Section 5c for the commands.
```

b. Create the Base Layer of your Plot and Add Points

Use your new filtered dataset (snail_data_means_sd_exp1)

Add color to the aesfunction. The colorvariable separates the data based on snailcolor and then plots these separately on the same graph (in different colors!).

```
# Make your plot for the snail_data_means_sd_exp1 data and add points
ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean, color=snailcolor))+geom_point()
```



c. Add a Line to your Plot to Connect the Points

The function to add a line is geom_line()and it is added to the ggplot command just like you added geom_point().

```
# Add a line to the plot
```

Note: geom_line()and geom_point()are followed by empty parentheses because they are functions. Functions must be able to accept arguments (e.g. the range of the data you would like to show). These arguments need to go in the parentheses associated with the function.

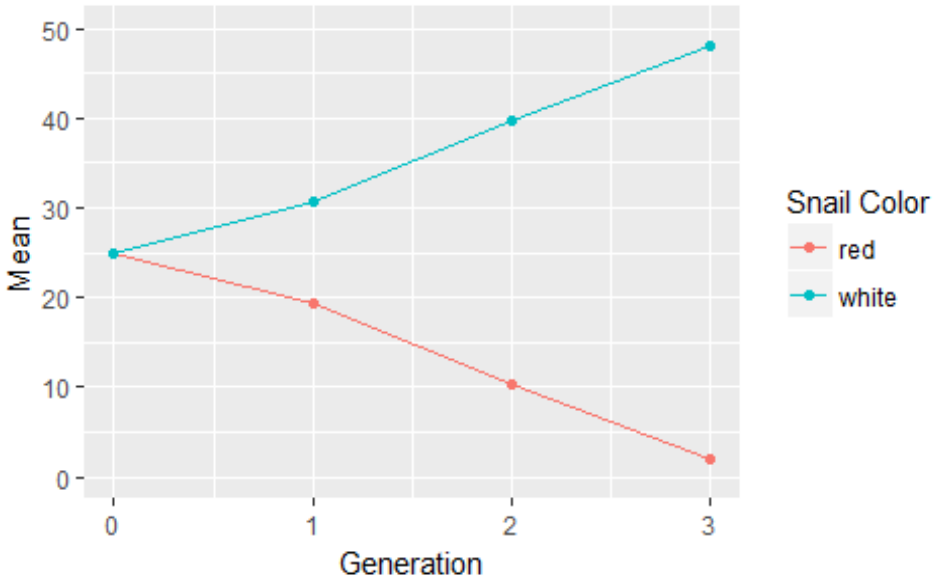
d. Add Labels to your Plot and Change Axis Titles

Label the axes by adding the functions scale_x_continuousand scale_y_continuousafter your code like you did in Lab 1, Section 5c. Use name = " " to name the x and y axes. Label the legend using +labs(color="Snail Color").

To make sure your y-axis starts at 0 and stops at 50, add limits = c(0,50) in the scale_y_continuousfunction. Note, the maximum for your y-axis may not be 50.

#Label your plot, change axis titles, limit range of y axis

#Use scale_y_continuous(name="Mean", limits = c(0,50)) to change limit range and rename y-axis



e. Add Standard Deviation Bars to your Plot

These graphs currently do not show the variance in your class's data like your boxplots in Lab A1. To show variation of our data in our line graphs, we add standard deviation bars by adding a layer using geom_errorbar().

geom_errorbar()draws an error bar that has an upper and lower value. In this case, the upper value is the mean + the standard deviation and the lower value is the mean - the standard deviation.

Add + geom_errorbar(aes(ymin = mean+stdev, ymax = mean-stdev))to your ggplot command.

Add the error bars to the plot using geom_errorbar

Examine your standard deviation bars visually. Do they overlap or are they far apart? Does this indicate that your means are different or similar to each other? Make sure to consider this as you write your lab report.

Save this plot as Lab2_Experiment1 so that you can turn it in with your lab report.

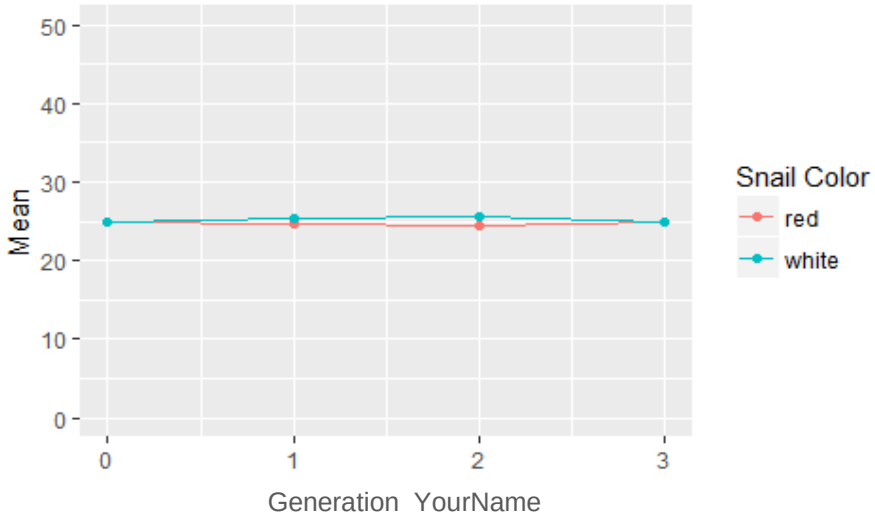
5. Plot your Data from Experiment 2

Repeat everything you did for Experiment 1 for Experiment 2.

Make sure your axis and legend titles are correct, and that error bars are included.

Filter data to include only means for Experiment 2 and name this variable snail_data_means_exp2

Graph Experiment 2 data using the same commands you used in Section 4.



Save this plot as Lab2_Experiment2 so that you can turn it in with your lab report.

6. Compare Number of Snails in Different Groups using a Statistical Test

Now that we have examined our data visually, we are interested in knowing if the red and white snail populations are statistically different at the end of the experiment (generation3). To compare red and white snail populations at generation 3, you will run a t-test. To run a t-test in R, we will use our original data set (snail_data) since the t-test takes into account the means of the data it is comparing, the sample sizes of the data it is comparing and the variation of the data it is comparing.

First, we filter our white and red snails from Generation 3 in both Experiments 1 and 2.
Use the command from Red1for Red2, White1, and White2. Make sure you include quotes around your variable names ("red" or "white").

```
# Subset snail data by generation, experiment and snail color and store it in the Red1 variable.
# Red1 contains just data from experiment 1's, generation 3, red snails
Red1 <- filter(snail_data, exp == 1 & generation == 3 & snailcolor == "red")

# White1 contains just data on experiment 1's, generation 3, white snails
# (write the command for White1 here)

# Red2 contains just data from experiment 2's, generation 3 red snails

# White2 contains just data from exp 2's, generation 3, white snails
```

Now that we have separate variables for each subset of data, we can run our *t*-tests. Input the two groups you want to compare, and then select the type of *t*-test to use. For this class, we will be using the two-sample t-test because we have two samples (a one sample test compares your data to some expected value).

Red1, Red2, White1, and White2 contain much more data than just number of snails, like generation, exp, group, etc. However, we are only interested in comparing the number of snails. To extract the number of snails in Generation 3 from the Red1, Red2, White1, and White2 variables, we use the dollar sign ($). The dollar sign extracts items from a variable based on their names. For example: Red1$snailsextracts the number of red snails in Generation 3 from Experiment 1

What is your null hypothesis for the Experiment 1 data?

```
# My null hypothesis is:
# red and white snail populations are the same at the third generation of experiment1.
# Oystercatcher_ttest
t.test(Red1$snails, White1$snails)
```

```
##
##         Welch Two Sample t-test
##
## data:  Red1$snails and White1$snails
## t = -56.338, df = 4, p-value = 5.943e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##         -48.26696 -43.73304
## sample estimates:
## mean of x mean of y
##       2               48
```

What is your null hypothesis for the Experiment 2 data?

```
# My null hypothesis is:
# Driftlog_ttest
t.test(Red2$snails, White2$snails)
```

```
##
##         Welch Two Sample t-test
##
## data:  Red2$snails and White2$snails
## t = 0, df = 4, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##              -8.173633      8.173633
## sample estimates:
## mean of x mean of y
##     25              25
```

The output from t.test gives you a lot of information. For this class, we are going to focus on the p-value. A p-value is the probability of getting data different from the observed data given that the null hypothesis is true.

Imagine you are willing to accept a 5% probability that you reject the null hypothesis if it is really true. That's like saying for every 20 experiments where the null hypothesis is true, one of them will probably appear as if the null is false (1/20 = 5%).

Because you have some variance in your data you need to allow for some probability of being wrong. If your p-value is less than 0.05 you will reject the null hypothesis. If it is greater than 0.05 you cannot reject the null hypothesis.

Reporting Independent Samples T-Test Data

To report results, include three main ideas:

- Discuss statistical test type and how used. Example: "An independent-samples t-test was conducted to compare red and white snail populations after three generations of predation from oyster catchers."
- Discuss significant differences between conditions. Example: "There was a significant (not a significant) difference in red (mean= __) and white snail (mean= __) populations after three generations of predation from oyster catchers; p = __ ."
- Describe your results in simple words explaining what you found. Example: "These results suggest that snail color really does have an effect on oyster catcher predation.

Make sure to include these three sentences in your results for your oyster catcher and your drift log experiment.

Lab Assign 2 Report Submission

Follow rubric. Submit a hard copy of your lab report that includes the following:

1) Hand-drawn line graphs from pre-lab section 0a & 0b with a description of the mechanism of evolution.
2) Summary table for your section's data (made in excel/sheets/or R)
3) Line graph of Experiment 1 means from Step 4d
4) Line graph of Experiment 2 means from Step 5
5) Results of your t-tests for both experiments from Step 6
6) Explain why you rejected or failed to reject your null hypotheses based on your t-test results and explain the processes that influenced both experiments using the format above.
7) Your code. Code should be organized, include good comments, and include only code that works and does not produce errors.

Lab 3A: Photosynthesis and Absorption Spectra of Algae

Examine Absorption Spectrum of Algae

In this exercise, you will measure the absorption of different wavelengths of light by the pigments from three different algal species using a spectrophotometer. You will identify which wavelengths are effectively absorbed and which are not.

PREDICTIONS

Predict what wavelengths you expect to see peaks and valleys for each sample.

Procedure to Measure the Absorption Spectrum of a Variety of Algal Species

1. Weight out 5g of fresh spinach OR algae. Grind samples into small pieces in a mortar with 10mL of 95% ethanol. Grind thoroughly, extracting the chlorophyll pigments from the chloroplasts.
2. Place a 50mL flask below a funnel lined with filter paper. Pour the ethanol and sample slurry into the funnel lined with filter paper. Once your filtered spinach-ethanol has dripped through, check your pigment concentration again the demo tube your TA has set up for comparison. If necessary, dilute your slurry by adding additional ethanol.
3. Fill one cuvette 3/4 full with 95% ethanol. This is your "blank."
4. Fill one cuvette 3/4 full with filtered sample.
5. Following the instructions next to the spectrophotometer, zero out the spectrophotometer with the "blank" ethanol cuvette, and then measure the absorption of the samples.
6. Measure the absorbance of extracts from three samples from 400 -700nm.
   Measure the absorbance at 10nm intervals. In the peak ranges, determine the highest absorbance peaks with more detailed measurements. Enter data in Table 7.

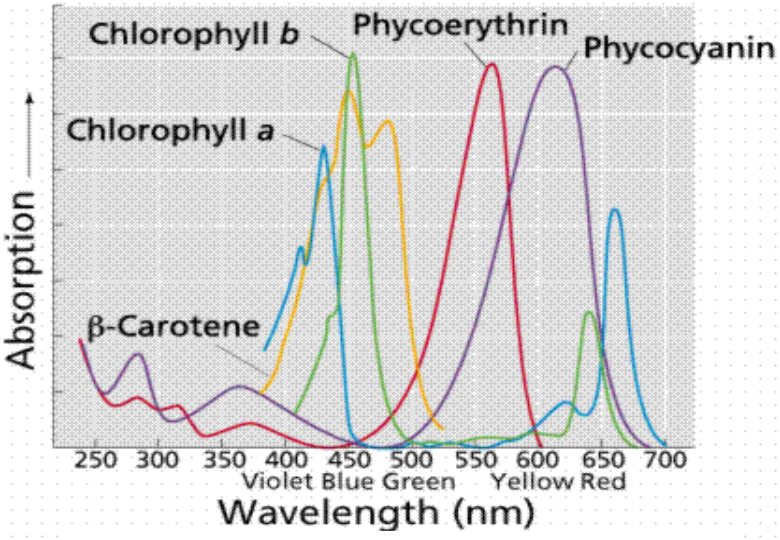


Figure 8. Absorption spectral of major plant pigments

At the end of measuring samples on the spec

1. Put contents of ground sample material and remaining EtOH into trash.
2. Wash mortar and pestle, cuvettes and disposable pipettes in sink and leave to dry on paper towels next to your spectrophotometer:

Assess results in lab report. Follow rubric from TA.

Table 7. Absorbance measures of samples at varying wavelengths.

Wavelength (nm)	Absorbance Spinach	Abs _____	Abs _____
400			
410			
420			
430			
440			
450			
460			
470			
480			
490			
500			
510			
520			
530			
540			
550			
560			
570			
580			
590			
600			
610			
620			
630			
640			
650			
660			
670			
680			
690			
700			

Lab 3B: Photosynthesis and Absorption Spectra of Algae

Lab Goals

- Predict expected absorption spectra
- Put your data into long form in your own Google Sheet.
- Import data and graph your photometric readings as points.
- Compare photometric readings from 3 different algae on one graph using color.

Before Starting with R, Sketch your Expected Results

- Draw line graphs to represent expected spectra for spinach and two macro-algae.
- Line graphs are useful to represent absorption over a range. Draw a line graph showing the expected absorption peaks across the 400-700nm spectrum. Below are the expected peaks of photosynthetic pigments we expect in each sample. Remember, you will graph all the pigments combined as one line per sample.

  Green plants: chlorophyll a (peaks ~ 430 and ~662 nm), chlorophyll b (~453 and ~642 nm)

  Phaeophyceae: chl a & chl b, and carotenoid pigment: fucoxanthin (442 & 472nm)

  Rhodophyta: chl a & chl b, and phycobilins (phycocyanin (605nm) and phycoerythrin (525 and 570nm))

  Chlorophyceae: chl a, chl b, (and carotenoids— 442 and 472 nm, but NOT soluble in ethanol.)

1. Getting Started

a. Put Data in Long Form on Google Sheets and Adjust Share It with your Team

Enter your group's data into a Google Sheet in long form. Set up column names Algae(algae_names or A,B,C), Wavelength(400 to 700, by tens), and Absorption(what you measured). You should have 1 header row and 93 rows of data total.

Change the permission settings to "Anyone with the link can view." Click on the blue Share button in the top right. Click "Get shareable link." Click on the drop-down menu and select "More..." Select "On - Anyone with the link." If you don't change the sharing settings, you will get an error when you try to use the gsheet2tbl() function. This is because RStudio doesn't have the permission to access the Google Sheet data.

Copy the link from your Google Sheet to use in RStudio.

b. Load the Packages You Need

Make a new project in a new directory and open a new script.
Refer to Lab 1, Part 2 if you need help.
Copy the comment lines into your script
Enter the code based on the previous labs and your R scripts from Labs A1 and A2

```
# MyFullName, myTAname  = learning linegraph & T-test in RStudio
# Load the packages ggplot2 and gsheet
```

2. Get your Data into R

```
# Store your Google sheets link as the variable url
# load your data stored in the variable url using the gsheet2tbl function
# and store it as the variable algae_data
```

Remember, once we have our data in, it is always good to check to make sure the data was imported correctly. Use the commands you learned in Lab 2, Part 2.

```
# Check your data to make sure it looks right (first lines only and column names)
```

3. Graphing your Algae Data

Just like last time, you are going to use ggplotto graph your algae data. We want to compare which wavelengths are more absorbed or less absorbed by different species of algae. We are going to graph y data as points. Refer to Lab 2, Parts 4b and 4d if you need help.

a. Create the Base Layer of your Plot with Correct Labels

First identify the variables (by looking at the column names)

- What is the independent variable (x axis)?
- What is the dependent variable (y axis)?
- How will you plot different algae types separately (color)?
- Remember to label your axes and legend (include units!)

```
# Create the base layer of the plot using ggplot
# Add datapoints and labels to the plot
```

Save this plot as Lab3_graph so that you can turn it in with your lab report.

Lab Assignment 3 Report Submission

Turn in a hard copy of your lab report that includes the following:

1. Hand drawn graph with 3 lines:, one line for spinach and one line each for your two algae absorption spectra.
2. Summary table for your group's data (wide form, made in excel/sheets/or R).
3. Graph of absorbance values over the visible spectrum for all 3 samples (one plot, with different colors for spinach and the two different algae).
4. Your code. Code should be organized, include good comments, and include only code that works and does not produce errors.
5. Answers to the following questions (at least one full page, double-spaced):
   - What wavelengths are most strongly (and most poorly) absorbed by which of the algae extracts? Directly reference your data and explain why you chose your answer.
   - Does your data match the known absorption spectra of different pigments found in chloroplasts (e.g. chlorophyll a, chlorophyll b, anthocyanin and carotenoids)? Which pigments have absorption spectra most similar to your results? Refer to your data and give exact values.
   - Would you expect the absorption spectra to be the same for all the spinach and all the algae? Why or why not? Consider where these organisms are found in the wild.
   - Why is it important to have the ethanol blank in this study? What is the purpose of the blank in spectrophotometry?

Lab 4A: Plant Form & Function - Seedling Growth under 2 Conditions

Measure Lettuce Seedling Growth under Two Different Light Conditions

At home over the next week, you will water, grow and measure lettuce seedlings grown under two light conditions: ambient (natural) light and no light. Record the height of the shoots (combining the hypocotyl and epicotyl) to the closest 0.5 cm each day. Record this in the data table provided.

To do this: start two vials, each with a filter on the bottom. Add about 1mL water. (The filter will hold water and seeds in place.) Mark 0.5 cm intervals on the side of each tube. Dry inside of tube, and add 6 lettuce seeds, spread out onto the filter.

Store vials in similar temperatures at home, one in indirect light and one in total dark conditions. (Caution, hot direct sunlight may kill young seedlings). Water as needed. Record the height of your seedlings every other day. Record a measurement for every living seedling. Order of recording is not important. Do not include non-germinating seeds or dying seedlings in your summary data.

Write out a hypothesis about what you believe will happen.

The null hypothesis is:

H_O: Lettuce seedlings will grow to the same height in the light and dark conditions.

Your alternate hypothesis:

H_A: _____

Data shows plant height for each seedling. To calculate the average, only include seedlings that were growing. (Exclude seedlings that did not germinate or desiccated.)

Table 8. Height of seedlings grown in light and dark environments.

	Shoot height after 2 days (cm)		Shoot height after 4 days (cm)		Shoot height after 6 days (cm)	
Seedlings in vial	Light	Dark	Light	Dark	Light	Dark
1						
2						
3						
4						
5						
6						
Avg.						

Record the average plant height for all plants that showed growth, include only growing seeds. Students will compare height of light and dark grown plants and look for differences.

Table 9. Class data table. Average height of plants grown in light and dark environments.

Student names	Average shoot height (cm)					
	Light grown seedlings			Dark grown seedlings		
	2-day	4-day	6-day	2-day	4-day	6-day
Avg.						
St. Dev.						

Lab 4B: Plant Form & Function: Seedling Growth under 2 Conditions

Lab Goals

- Understand where the mass of the plants came from
- Understand variation in data and what that means
- Understand replication and pseudo-replication
- Represent variation in data on graphs
- Compare two treatments statistically, where the data varies among individuals

1. Getting Started

Variation in biology is generally used to describe the differences between individuals (or cells, or groups of organisms) in genetics, in phenotype, or in some combination. In data science, variation has a different meaning, as a measure of spread in data. Measures of variation (spread) describe the distribution of the data and may be calculated in many ways including range, variance, standard deviation, and interquartile range. Variance for example may be an indication of biological variation, but there can also be an additional source of variation that is included, that is chance variation, also known as noise or random error.

Chance variation can be a major challenge in analyzing data because practical considerations limit sample sizes and number of experiments. This random variability/noise appears at multiple levels, either because of changes in time, space, procedural changes, experimental bias, experimenter-bias, or chance events. What can we do to reduce this chance variation? Remember the 3 Rs - Randomization, Rigor, and Replication. Randomization reduces bias, and rigor means following the scientific method and use a methodical approach. Replication of treatments reduces random error, but we aim for true replicates and try to avoid pseudoreplication. How do you know whether you are pseudoreplicating? Think of the unit of your analysis.

Here you will be experimenting to determine the impact of light and dark on plant growth. To test for treatment effects, you must have (independent) replications of your treatments. In your experiment you may have 100 plants in each condition, light and dark, but no matter how many plants, you still only have one true replicate of the treatment of interest (light vs dark).

a. Avoiding Pseudoreplication

Calculate the average height of your group's light and dark seedlings at each interval (2 days, 4 days, and 6 days) using Google Sheets. You should have 6 values total. Put your group's data into a Google Sheet in long form. Set it up with the column names exposure (light or dark), day(2,4,6), and height(what you measured).

By calculating the average of the 6 seedlings you measured, you are getting a good estimate of your group's sample. Each of the seedlings you measured are called pseudoreplicates, while the means for each of your seedling groups are called a true replicate. Calculating a single estimate from multiple plants minimizes the error in your estimate of the height of your seedlings (i.e. measurement error). However, it means you only have one height estimate per treatment/day.

b. Put Class Data in Long Form on Google Sheets

Go to the Lab Site and scroll down like you did in Labs 1 and 2. Select the link for your lab section. Put your group's data into the class' Google Sheet in long form using the same columns you did for your group data. Note that there is another column for group number. Copy the link to this Google Sheet to use in RStudio.

Now you have good estimates of multiple samples to work with. There will be some variation among samples due to the soil, the location the plants were grown, and how different people measured height. Having multiple samples allows you to estimate a mean height despite all this variation.

c. Load the Packages You Need

Make a new project in a new directory and open a new script. Refer to Lab A1, Part 2 if you need help.

- Copy the comment lines into your script
- Enter the code based on the previous labs and your R scripts from Labs A1 and A2.

```
# MyFullName, myTAname  = plotting scatterplot & linear regression
# Load the packages ggplot2, gsheet, and dplyr
```

2. Get your Data into R

```
# Store the Google sheets link as the variable url
# Load the data stored in the variable url using the gsheet2tbl function
# and store it as the variable plant_data
```

Remember, once we have our data in, it is always good to check to make sure the data was imported correctly. Use the commands you learned in Lab A2, Part 2.

```
# Check your data to make sure it looks right (first lines only and column names)
```

3. Graphing the Seedling Data

Because we want to compare the seedlings grown under light and dark conditions, you are going to graph both sets of data together. You will first graph the individual data as points. This way you can see how much variation is in your data.

a. Create the Base Layer of your Plot

What is the independent variable (x axis)?

What is the dependent variable (y axis)?

How will you plot light and dark conditions separately? Refer to Lab A2, Part 4b if you need help.

```
# Create the base layer of your plant_data plot using ggplot
```

b. Add the Data and Labels to the Plot

Note: the variable dayis an integer, which means it is on a continuous scale rather than in discrete categories; thus, the functions to label your axes are continuous (scale_x_continuous, not scale_x_discrete). We use a discrete scale when plotting variables like SiteTypeor Color, where each variable is its own category. Refer to Lab A1, Part 5b if you need help.

```
# Add datapoints and labels to the plot using geom_point
```

Save this plot as Lab4_graph1 so that you can turn it in with your lab report.

Does it look like there might be a difference between the two groups?

Based on just this plot, what can you say about the differences? How confident do you feel?

4. Compare the Height of Seedlings in Different Conditions

We are interested in comparing the growth rates of seedlings in light and dark conditions. To make these comparisons we need to know the mean of each group of exposure data and how much variation there is within each group.

Group the data by exposure and day. Refer to Lab A2, Part 3 if you need help.

```
# Group the plant_data variable by exposure and day using the group_by function
# and store it as the variable grouped_plant_data
```

Calculate the mean heightof each group. Refer to Lab A2, Part 3 if you need help.

Add na.rm=TRUEinto the mean()function. This additional code tells R to ignore areas in the data set that had no values entered. For example, if no seedlings grew under your dark condition, you would not record any data under height for that experiment. By using this code, R can now ignore these blank spaces and calculate the mean correctly.

```
# Calculate the mean height of the seedlings at each interval using the summarise function
# and store it as the variable plant_data_means
# Use na.rm=TRUE to ignore NA values in the data
```

Check the first few lines of your mean data (plant_data_means) using head()

5. Graphing the Mean Seedling Data

Because we want to compare the seedlings grown under light and dark conditions, we are going to graph both sets of data together.

a. Create the Base Layer of your Plot and Add Points

What is the independent variable (x axis)?

What is the dependent variable (y axis)?

How will you plot light and dark conditions separately? Refer to Lab A2, Part 4b if you need help.

```
# Make your plot for the plant_data_means data using the ggplot command
```

b. Calculate the Standard Deviation of the Means

It appears that there are differences in seedling heights between the two different growing conditions. However, if there is a lot of variation in our data those differences may not be statistically significant. We can check if the light and dark grown seedlings are statistically significantly different using a t-test.

The standard deviation is how much the individual data points differ from the overall mean on average. Using the standard deviation gives a single value on how much variation there is in the data. For example, if we saw seedling heights of 3, 2.9, 3, and 3.1, we would have a small standard deviation. However, if we had seedling heights of 0.5, 5.5, 2, and 4, the standard deviation would be high.

Standard deviation is calculated in R using the function sd.

```
# Remake your table of means so it includes std deviation
# and store the additional data as the variable plant_data_means
plant_data_means <- summarise(grouped_plant_data,mean = mean(height, na.rm=TRUE), stdev = sd(height,
na.rm=TRUE))
```

c. Add standard deviation bars to your plot

Next add the standard deviation bars to your graphs by adding a layer using geom_errorbar

geom_errorbardraws an error bar that has an upper and lower value. In this case, the upper value is the mean + the standard deviation and the lower value is the mean - the standard deviation.

Add + geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev))to your ggplot command.

```
# Add the error bars to the plot using geom_errorbar
```

d. Add Labels to your Plot

Just like last class, you'll want to clean up your graph and make it look professional.

- Add labels to the axes and legend

- Change the width of the error bars by adding width = 0.2after ymax in geom_errorbar.

```
# Label your plot and change axes titles
# Edit the width of the error bars
```

e. Draw Lines Representing a Trend That Fits your Data

This model is a linear (straight) best fit line. It is referred to as a regression.

- Add the model lines to your graph by adding a layer using geom_smooth.
- There are two lines: one for each subset of data.
- Add + geom_smooth(method="lm", se = FALSE)to your ggplot command.

```
# Add the regression to your plot using geom_smooth
```

Save this plot as Lab4_graph2 so that you can turn it in with your lab report.

- Under which condition did the seedlings grow better? Light or dark?
- How confident do you feel? More or less than when you plotted all the data?

6. Compare Height under Different Conditions using Statistics (a t-Test)

Just like in Lab A2, we have two groups of data we want to compare to see if there is a statistical difference. In this case, we're interested in whether there is a difference between seedling heights in the 6 day old seedlings.

Filter light and dark grown seedlings for the day 6 time period. Refer to Lab A2. Part 6 for help.

```
# Subset plant data by exposure and growth time
# and store as the variables light6days and dark6days
# Remember to use “quotes” around your variable names.
```

What is your null hypothesis for the 6 day old seedling data?

```
# t test to compare heights of 6 day old seedlings for light and dark conditions
```

Under which condition did the seedlings grow taller? Light or dark?

Lab Assignment 4 — Report Submission

Turn in a hard copy of your lab report that includes the following:

1. Summary table for your section's data (wide form, made in excel/sheets/or R).
   Do not include all the raw data, just means and standard deviations for each group.
2. Scatterplot of all data from Step 3b.
3. Graph with mean data points, error bars, and regression from Step 5e.
4. Your code. Code should be organized, include good comments, and include only code that works and does not produce errors.
5. Answers to the following questions (one full page, double-spaced):
   a. Under which condition did the seedlings grow taller? Light or dark?
   b. Describe the variance you see in the mean data.
      How is this reflected in the error bars?
   c. How does your best fit line (regression) fit the mean data?
      Does it help show the overall trend?
   d. What is your null hypothesis for the 6 day old seedling data?
   e. Explain why you rejected or failed to reject your null hypothesis based on your t-test results and explain the processes that may have influenced the experiment.

More Material To Help You Understand Mean And Standard Deviation:

- Check out this interactive website for a better understanding of mean and standard deviation! http://www.zoology.ubc.ca/~whitlock/Kingfisher/SamplingNormal.htm
- Click on the tutorial button to work though the example. Things to think about:
  When you sample multiple individuals do you see variation?
  When you calculate the mean of your samples what are you estimating?
  Why do you need to calculate the mean of many samples (each of which is the mean of multiple measurements)?

Materials

A computer with Internet access is required for each student. Students may use their own computer or the computer needs to be provided in lab. R and RStudio need to be installed on individual computers or on a server (the latter is preferable). In the latter case, Chromebooks and tablets are sufficient for data analysis.

For the plankton species diversity data collection, the following is necessary:
- Compound microscope shared between 2-3 students.
- Sedgewick-Rafter counting cells (from Cole-Palmer, item #EW-05491-64, ~$120 each.
- Plankton samples from 4 sites, each student needs 1mL to fill counting cell. Samples must be diluted prior to class to a countable density. (see attached student info sheet: "Sedgewick Rafter counting cell").
- Picture key to identify plankton (attached).

Notes for the Instructor

To support successful implementation of these labs we offer the following suggestions:

1. Teach spreadsheets/ data management first. In the first semester of the introductory biology lab sequence (not included here), students are taught spreadsheet/data management skills. Students are taught spreadsheet terminology (columns, rows, pages) and spreadsheet familiarity (writing formulas, creating tables and graphs). We introduce many students to "variation around the mean", hypothesis testing, and simple statistical tests. We utilize Google Sheets so all students have the same spreadsheet version and can use the spreadsheet program without purchasing any software.
2. Use a server. We use a server instead of having students download the free R and R-Studio software. Although it is free and easy to download the software, we found many students were unable to successfully download and access the program on their own computers. To bypass this bottleneck, our university IT staff have set up a server with the software and necessary packages. The following link provides detailed instructions on how to set up a server: https://deanattali.com/2015/05/09/setup-rstudio-shiny-server-digital-ocean/. These instructions can be used to set up a server in the cloud, or adapted by university staff to provide a server on-site. For discussion of the pros and cons of using a server see Wright et al. (2019). We found that we did not have time in our short labs (with inexperienced GTAs) to help every student install R and RStudio on their own computers. Additionally, not every student had a computer that supported RStudio.
3. Be enthusiastic, encouraging and sympathetic. Our GTAs lead their labs with great enthusiasm and empathy. Our GTAs constantly encourage students, emphasizing the challenges of learning a new language, while reinforcing the realistic accessibility. We also have Undergrad Teaching Assistants (UTAs) in the lab. UTAs are students who were successful in the lab in a previous semester, and are excited to help teach fellow students about biology and data analysis. They are great at empathizing with their fellow students, acknowledging the challenges of a new approach to data analysis, while cheering on successes.

Why R?

While getting students to analyze data is likely seen as a general positive, some instructors may wonder at the choice of a coding language over point-and-click software or using Excel for graphs. For one, bioinformatics and code are increasingly a core component of an education in biology (Wilson Sayres et al. 2018). Additionally, when writing code students are able to easily repeat analyses and can do so efficiently for large datasets. Students can reference prior code to repeat an analysis because code generates a record of their work in a way that point-and-click work does not, and writing tutorials in code is straightforward without extensive images of where students need to click. We chose R specifically because students are able to learn sufficient code to produce graphs and manipulate data in a very short period of time. Furthermore, R is widely used in biology, making it a valuable tool in upper-division courses, graduate school, and other jobs. For further discussion see Wright et al. (2019).

How Faculty Can Get Started with R

One hurdle when using R for data analysis is that instructors may not feel comfortable using R themselves (Williams et al. 2017; Cummings and Temple 2010). We recommend Data Carpentry workshops, which are taught regularly around the world. We found that TAs may be sufficiently familiar with R such that a lab coordinator (like author, L. Forrester) may need only limited proficiency. However, with or without TAs, using the pre-written R-lab-manuals (presented here) and the pre-written "R-lab_Answers explaining steps sheets" (also presented here), a novice to coding can quickly become proficient enough to teach these materials.

We encourage our students to learn new code by searching the internet, trying new code they find, and then

sharing useful code with the TAs and lab coordinator. Encouraging students to independently find novel answers increases their interest and excitement in data analysis and learning to write code to solve problems.

Teaching Approach

To support students in multiple ways we have adopted several supporting teaching strategies. First, students are initially taught to code using the live coding method promoted by Data Carpentry (Raj et al. 2018). Tutorials with each new step are provided to students. TAs also hold joint office hours, sharing their instructional load. With a large number of lab sections and TAs at URI, we are able to hold 16 hours per week of "computer help time." These are attended by many students. TAs constantly encourage students to attend, calling the help hours "biology study hall," a concept that first year students find comforting and non-judgmental. Learning to code can be as frustrating as learning a new language, and these open help hours offer students the support they need to feel successful. For further suggestions on teaching strategies, see Wright et al. (2019).

Cited References

Barone L, Williams J, Micklos D. 2017. Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. PLoS Comput Biol. 13(10): e1005755.

Berger, MS. 2016. Utilizing the Software Package "R" to Generate Graphs and Perform Statistical Analyses in Undergraduate Laboratory Courses. Article 2 in Tested Studies for Laboratory Teaching, Volume 37 (K. McMahon, Editor). Proceedings of the 37th Conference of the Association for Biology Laboratory Education (ABLE). http://www.ableweb.org/volumes/vol-37/?art=2

Cummings MP, Temple GG. 2010. Broader incorporation of bioinformatics in education: opportunities and challenges. Brief Bioinform. 11(6): 537–43.

Raj AGS, Patel JM, Halverson R, Halverson, ER. 2018. Role of live-coding in learning introductory programming. In: Proceedings of the 18th Koli Calling International Conference on Computing Education Research ACM. 13.

Williams J, Drew J, Galindo-Gonzalez S, Robic S, Dinsdale E, Morgan W, Triplett EW, Burnette J, Donovan S, Elgin S, Fowlks ER, Goodman AL, Grandgenett NF, Goller C, Hauser C, Jungck JR, Newman JD, Pearson W, Ryder E, Wilson Sayres MA, Sierk M, Smith T, Tosado-Acevedo R, Tapprich W, Tobin TC, Toro A, Welch L, Wright R, Ebenbach D, McWilliams M, Rosenwald AG, Pauley MA. 2017. Barriers to integration of bioinformatics into undergraduate life sciences education. BioRxiv. [Preprint]. 204420.

Wilson Sayres MA, Hauser C, Sierk M, Robic S, Rosenwald AG, Smith TM, Triplett EW, Williams JJ, Dinsdale E, Morgan WR, Burnette JM 3rd, Donovan SS, Drew JC, Elgin SCR, Fowlks ER, Galindo-Gonzalez S, Goodman AL, Grandgenett NF, Goller CC, Jungck JR, Newman JD, Pearson W, Ryder EF, Tosado-Acevedo R, Tapprich W, Tobin TC, Toro-Martínez A, Welch LR, Wright R, Barone L, Ebenbach D, McWilliams M, Olney KC, Pauley MA. 2018. Bioinformatics core competencies for undergraduate life sciences education. PLoS One. 13(6): e0196878.

Wright AM, Schwartz RS, Oaks JR, Newman CE, Flanagan SP. 2019. The why, when, and how of computing in biology classrooms. F1000 Research 2019, 8:1854.

Acknowledgments

Thanks to all who contributed to the development of these materials, including our graduate TAs: Rebecca Stevick, Robert Witkop, and Thomas Privott, and our URI IT guru, Chi Shen.

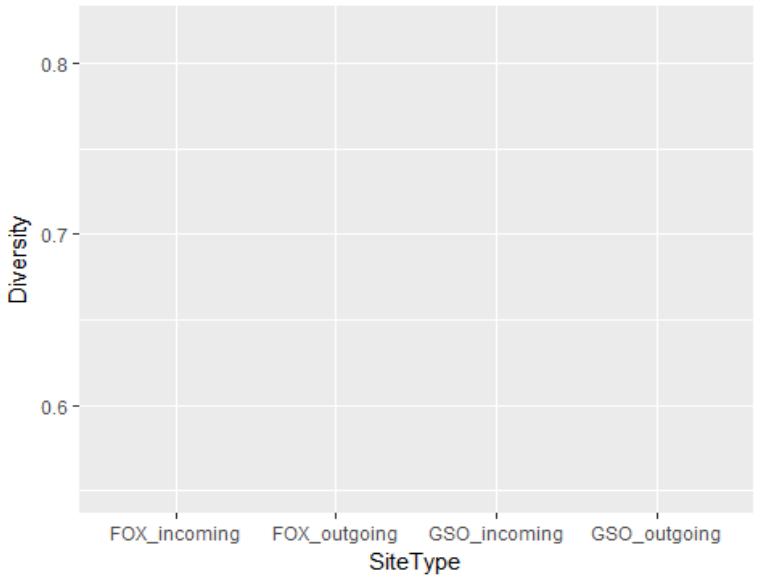
About the Authors

Linda Forrester has been the manager of Introductory Biology labs at the University of Rhode Island since 2006. She is interested in helping high school students transition to being college students and future scientists, by learning to be independent, self-motivated students.

Rachel Schwartz has been an Assistant Professor of Biological Sciences at the University of Rhode Island since 2016. Her research is on using genomic data to understand evolution; she teaches courses in biological data analysis and data science.

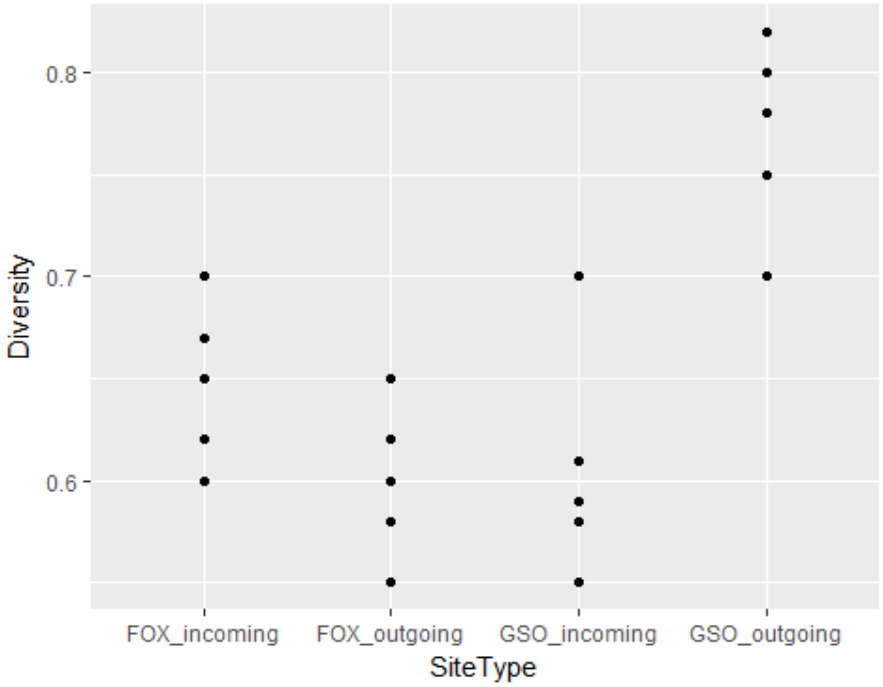
Appendix A
Lab Assignment 1: Answers

```
# MyFullName, myTAName = Learning how to make Boxplot in RStudio

# always start every new R Script with some description of what you are doing
#load the package that allows us to access google sheets
library(gsheet)

#assign the website address for the data to a variable using an arrow <-
# "variable" <- 'website address for data'
url <- 'https://docs.google.com/spreadsheets/d/1mTQPw3GsqID_ay6YqdNISOOKIMb9HHeBpFu7EWFiGj8/edit#gid=0'

#load the data from google sheets and store it as variable diversitydata
diversitydata <- gsheet2tbl(url)

# Load the package that allows us to plot data
library(ggplot2)

# Create a plot base-layer
ggplot(diversitydata, aes(x=SiteType,y=Diversity))
```

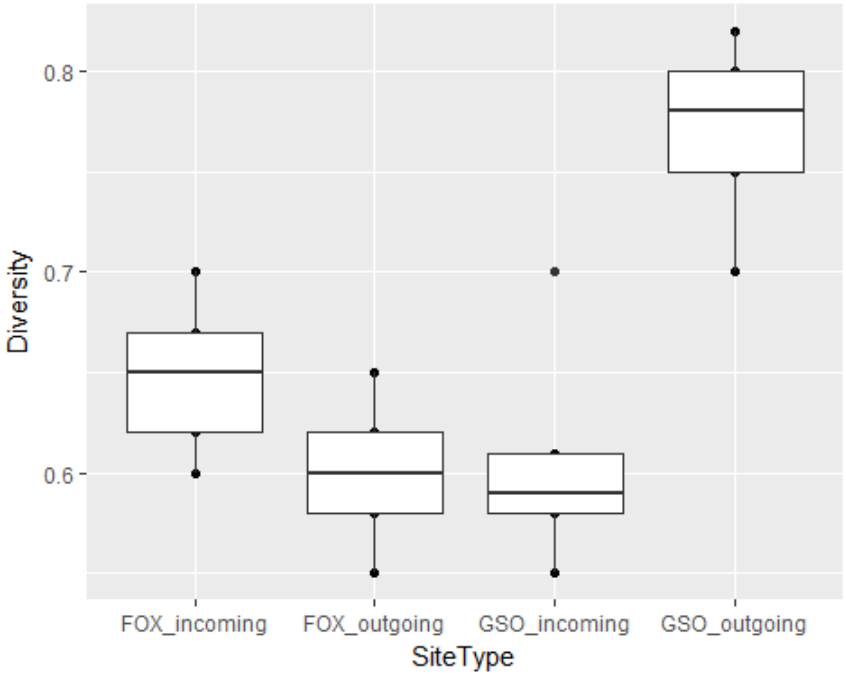


Be sure that your url is pasted here in SINGLE quotation marks.

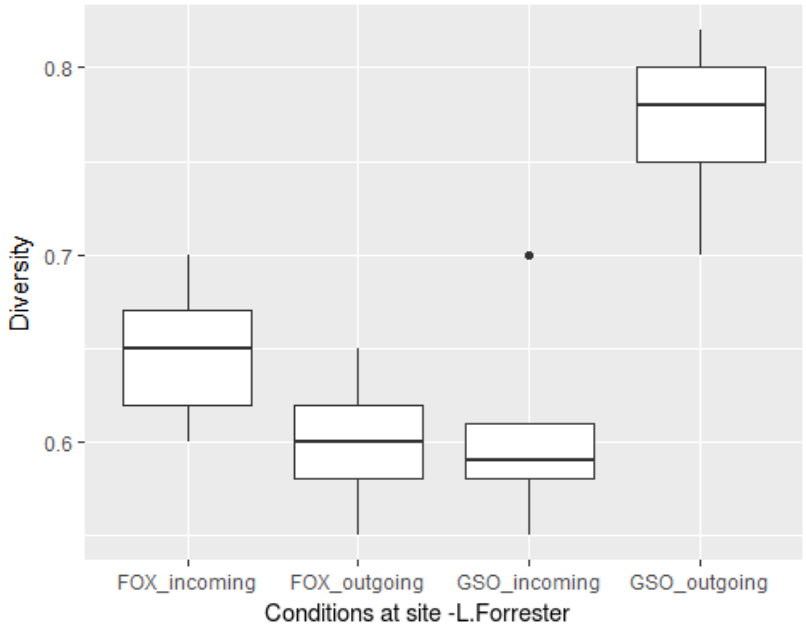
```
# Add points to the plot using geom_point()
ggplot(diversitydata, aes(x=SiteType,y=Diversity)) + geom_point()
```



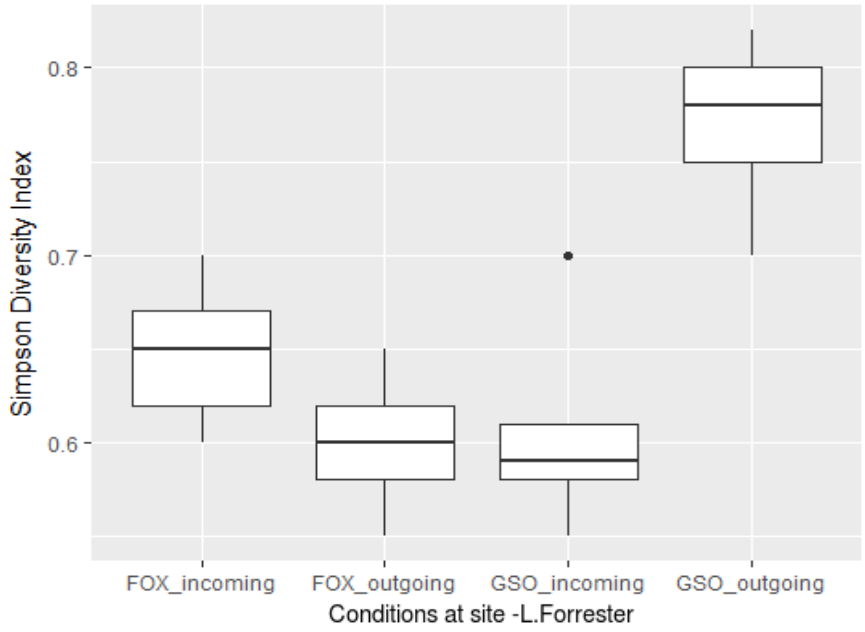

```
# Overlay boxplots to the plot using geom_boxplot()
ggplot(diversitydata, aes(x=SiteType,y=Diversity))+geom_point()+ geom_boxplot()
```




```
# Plot boxplots (w/o geom_points) and change the x-axis label
ggplot(diversitydata, aes(x=SiteType,y=Diversity))+geom_boxplot()+scale_x_discrete(name ="Conditions at site -MyName"
)
```

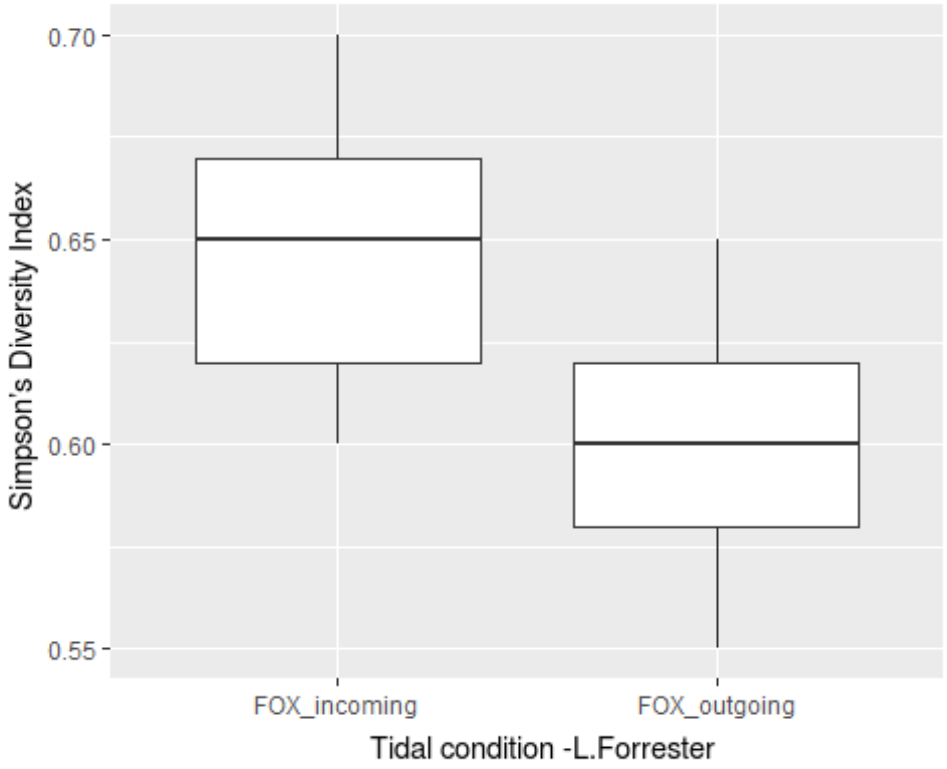



```
# Now also changing the y-axis labels
ggplot(diversitydata, aes(x=SiteType,y=Diversity))+geom_boxplot()+scale_x_discrete(name ="Conditions at site -MyName"
) + scale_y_continuous(name="Simpson Diversity Index")
```

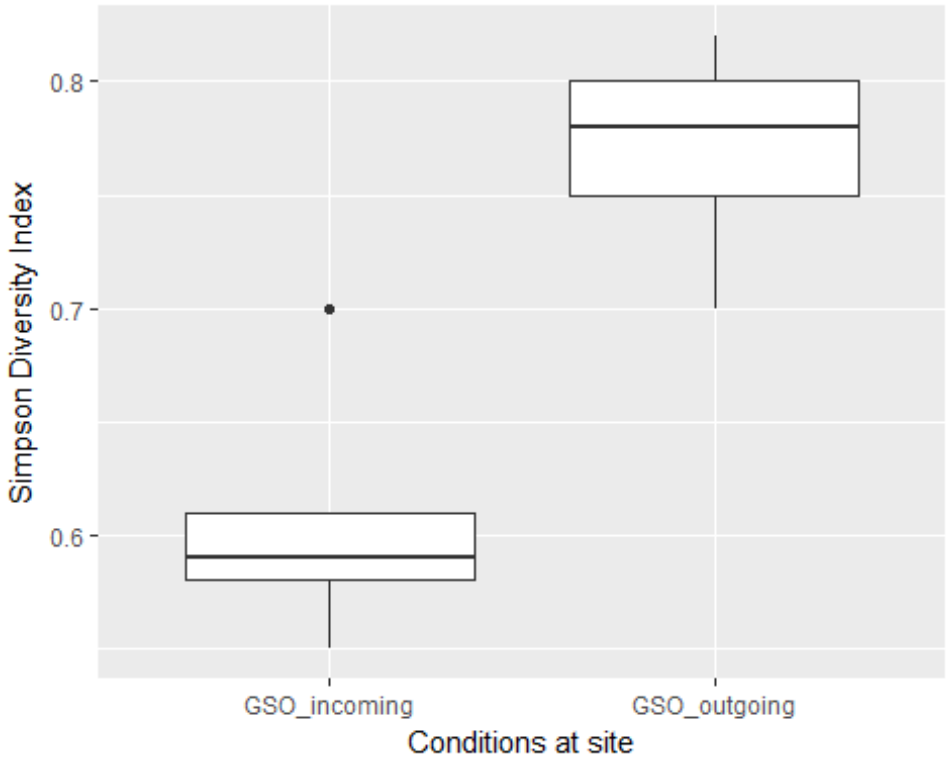



```
# Load the dplyr library
library(dplyr)

# When I only want to plot a subset of data, I filter for what I want

# Create new variable FOXonly with only FOX_incoming & FOX_outgoing data
# To do this: filter variable (diversitydata) using the column SiteType
```

```
# Use double equal (==) to say ONLY “Fox_incoming”
# Use | to show selecting another ONLY “Fox_outgoing”

FOXonly <- filter(diversitydata,SiteType=="FOX_incoming" | SiteType=="FOX_outgoing")

# Plot the FOXonly data with boxplots and changed axis labels
ggplot(FOXonly, aes(x=SiteType, y=Diversity)) +geom_boxplot()+scale_x_discrete(name ="Tidal condition -MyName")+
scale_y_continuous(name="Simpson’s Diversity Index")
```

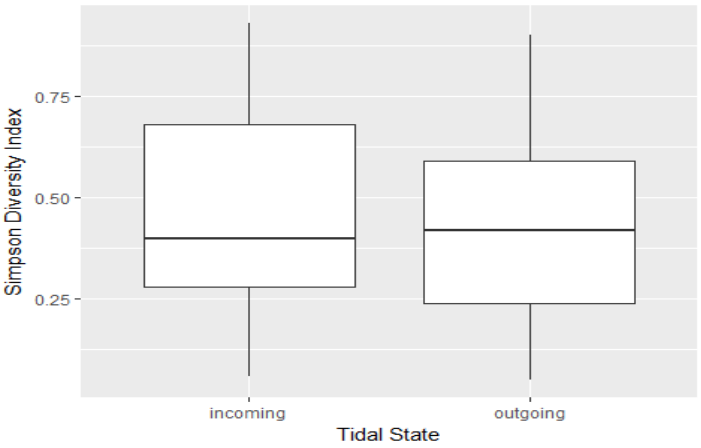



```
## Homework Part 1 ------------------------

# Create a new variable named GSOonly with just the GSO_incoming and GSO_outgoing data
GSOonly <- filter(diversitydata,SiteType=="GSO_incoming" | SiteType=="GSO_outgoing")

# Plot the FOXonly data with boxplots and changed axis labels
ggplot(GSOonly, aes(x=SiteType, y=Diversity)) +geom_boxplot() +scale_x_discrete(name ="Conditions at site") +
scale_y_continuous(name="Simpson Diversity Index")
```

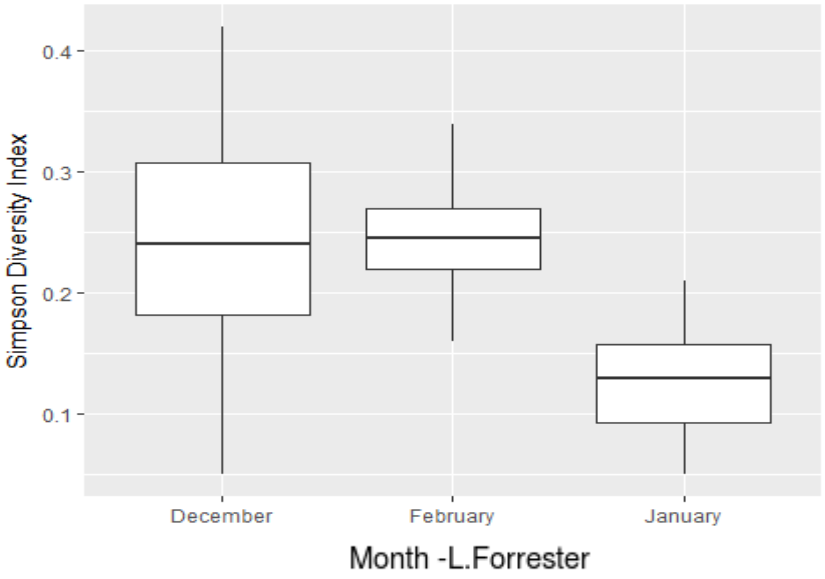


Homework Part 2 ----------

```
#assign the website address for the data to a variable url2
url2 <-'https://docs.google.com/spreadsheets/d/1RdEzJqLi1vRcpR80nH__0n7nJJ8gPZxoxxuHQqvtbWs/'

#load the data from google sheets and store it as the variable FOXyearlydiversitydata
FOXyearlydiversitydata <- gsheet2tbl(url2)

# Plot the FOXyearlydiversitydata with boxplots and changed axis labels
ggplot(FOXyearlydiversitydata, aes(x=Tide, y=Diversity))+geom_boxplot()+scale_y_continuous(name="Simpson Diversity Index")+scale_x_discrete(name="Tidal State")
```



```
# Create a new variable named threemonthsonly with just the January, February, and December data
threemonthsonly <- filter(FOXyearlydiversitydata, Month=="January" | Month == "February" | Month == "December")

# Plot the threemonthsonly data with boxplots and changed axis labels
ggplot(threemonthsonly,aes(x=Month, y=Diversity))+geom_boxplot()+scale_y_continuous(name="Simpson Diversity Index
")+scale_x_discrete(name="Month -MyName")
```



Appendix B
Lab Assignment 2: Snail_populations.R - Answers

```
##  BIO 104 lab Assign 2
## ANSWER KEY  -- RW

#load the libraries.  Rem to RUN the libraries every time you start R.
   Like an app on your phone, they must be “open” or “run” in order to work.

library(ggplot2)
library(gsheet)
library(dplyr)

#assign the website address for the data to a variable using an arrow <-
# "variable" <- 'website address for data'
url <- 'https://docs.google.com/a/my.uri.edu/spreadsheets/d/1IHG_vFP3aocYeKOCah6R8Uy0XU38SvYcgKw8Rol1x28/edit'

#load the data from google sheets and store it as a variable

snail_data <- gsheet2tbl(url)
```

```
#make sure data imported correctly
head(snail_data)

## # A tibble: 6 x 5
##     exp group generation snailcolor snails
##   <int> <int>      <int>      <chr>  <int>
## 1     1     1          0      white     25
## 2     1     1          1      white     26
## 3     1     1          2      white     24
## 4     1     1          3      white     26
## 5     1     1          0        red     25
## 6     1     1          1        red     24
```

```
#examine column names
colnames(snail_data)

## [1] "exp"        "group"      "generation" "snailcolor" "snails"
```

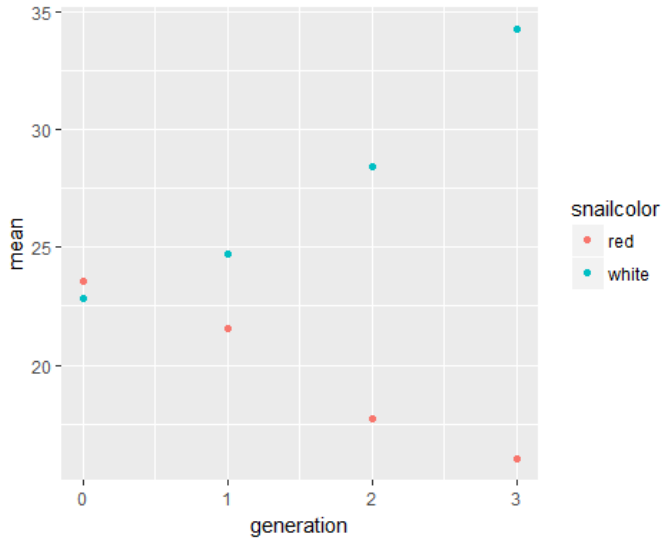
```
# Group the snail_data variable by exp, generation, and snailcolor
# using the group_by function

grouped_snail_data <- group_by(snail_data, exp, generation, snailcolor)
```

```
# Calculate the mean number of snails for each group using the summarise function
# by giving the name of the data you want to summarize (`grouped_snail_data`)
# and that you want the mean value of the snail counts ('snails')

snail_data_means<- summarise(grouped_snail_data, mean=mean(snails))
```

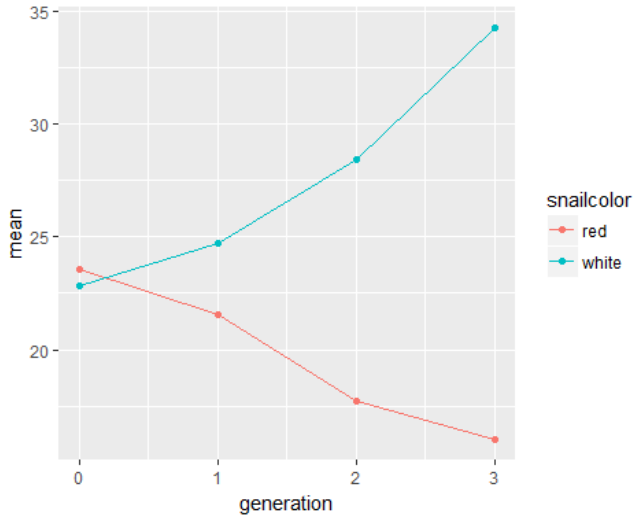
```
# Remake your table of means so it includes standard deviation
#and store the additional data as the variable snail_data_means_sd
snail_data_means_sd <- summarise(grouped_snail_data,mean = mean(snails), stdev = sd(snails))
```

```
# Filter data to include only means for Experiment 1 (exp==1).
# Refer to Lab A1, Section 5d for the commands.
snail_data_means_sd_exp1<-filter(snail_data_means_sd, exp==1)
```

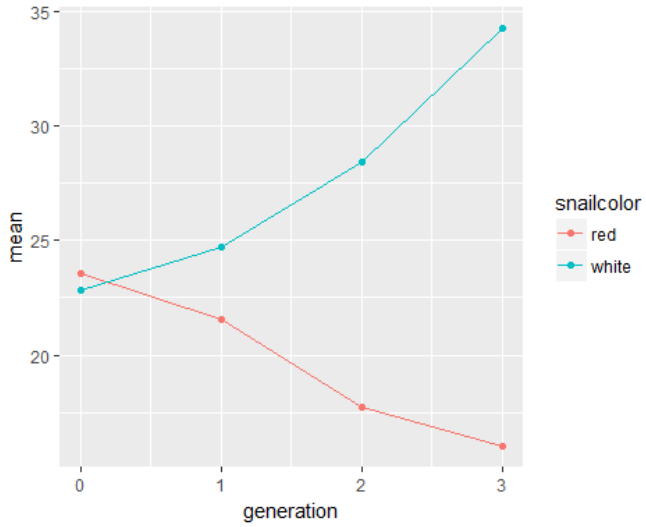
```
# Make your plot for the snail_data_means_exp1 data
ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean, color=snailcolor)) + geom_point()
```



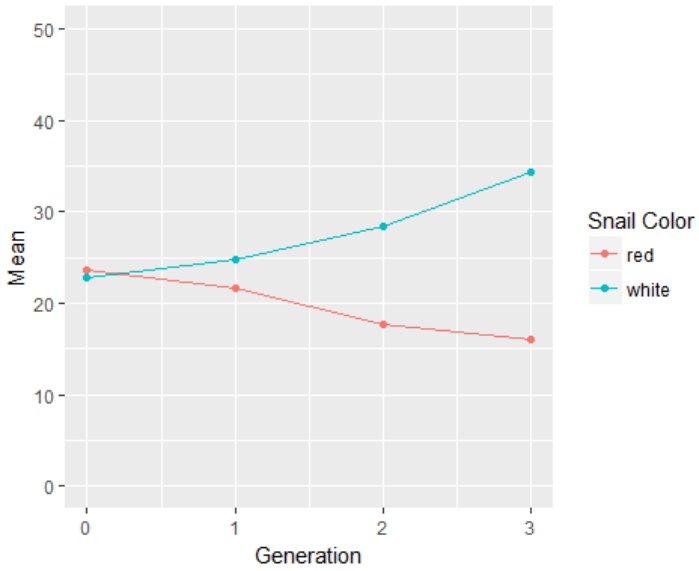

```
# Add the line and points to your plot
ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean, color=snailcolor))+geom_line()+geom_point()
```



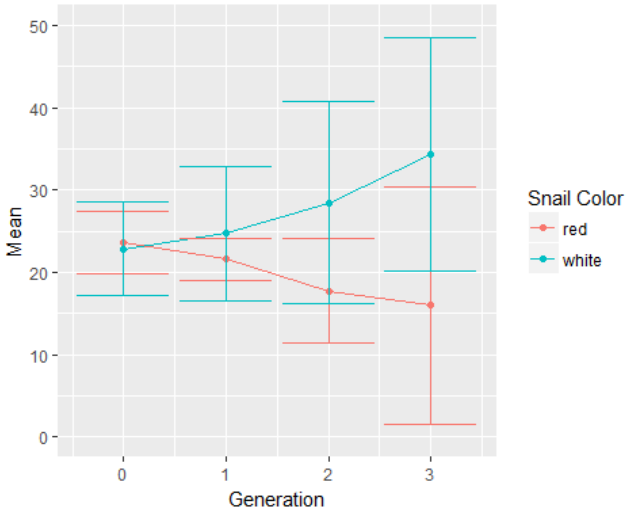
```
# Add the line and points to your plot

ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean, color=snailcolor))+ geom_line()+geom_point()
```




```
#Label your plot, and change axes titles

ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean, color=snailcolor))+geom_line()+geom_point()+scale_x_continuous(name="Generation") + scale_y_continuous(name="Mean", limits = c(0,50))+labs(color="Snail Color")
```

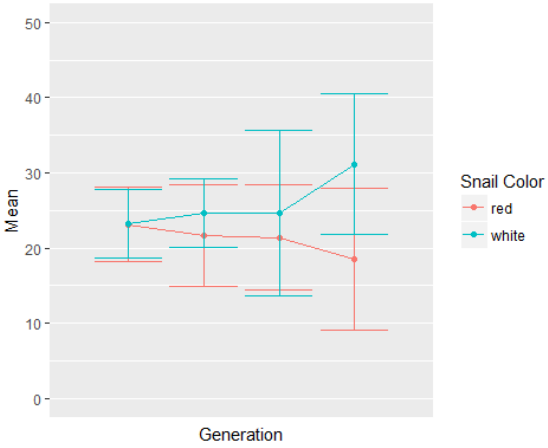



```
# Add error bars to the plot using geom_errorbar
ggplot(snail_data_means_sd_exp1, aes(x=generation, y=mean,
color=snailcolor))+geom_line()+geom_point()+scale_x_continuous(name="Generation")+scale_y_continuous(name="Mean", limits = c(0,50))+labs(color="Snail Color")+geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev))
```



IF error bars for Generation 3 do not appear, then they may need to make the limits larger, eg. limits = c(-2, 52)

```
#Filter for experiment 2
snail_data_means_sd_exp2<-filter(snail_data_means_sd, exp==2)

#Plot experiment 2
ggplot(snail_data_means_sd_exp2, aes(x=generation, y=mean, color=snailcolor))+geom_line()+geom_point()+scale_x_discrete(name="Generation") + scale_y_continuous(name="Mean", limits = c(0,50))+labs(color="Snail Color")+geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev))
```




```
# Subset snail data by generation, experiment and snail color and store it in the Red1 variable.
# Red1 contains just data from exp 1 red snails
Red1 <- filter(snail_data, exp==1 & generation==3 & snailcolor=="red")

# White1 contains just data from exp 1 white snails
White1 <- filter(snail_data, exp==1 & generation==3 & snailcolor=="white")
# Red2 contains just data from exp 2 red snails
Red2 <- filter(snail_data, exp==2 & generation==3 & snailcolor=="red")

# White2 contains just data from exp 2 white snails
White2 <- filter(snail_data, exp==2 & generation==3 & snailcolor=="white")
```

```
# My null hypothesis is that
# red and white snail numbers are the same at the third generation of experiment 1.

# Oystercatcher_ttest
t.test(Red1$snails, White1$snails)

##
##  Welch Two Sample t-test
##
## data:  Red1$snails and White1$snails
## t = -2.3896, df = 11.997, p-value = 0.03417
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -34.959071  -1.612358
## sample estimates:
## mean of x mean of y
##  16.00000  34.28571
```

```
# My null hypothesis is:

# Driftlog_ttest
t.test(Red2$snails, White2$snails)

##
##  Welch Two Sample t-test
##
## data:  Red2$snails and White2$snails
## t = -2.5017, df = 11.999, p-value = 0.02783
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -23.52025  -1.62261
## sample estimates:
## mean of x mean of y
##  18.57143  31.1428
```

Appendix C
Lab A3: Algae Spectrophotometer Data - Answers v. 2183

Load packages ggplot2 and gsheet (load dplyr if you want to use the filter function)

```
library(gsheet)
library(ggplot2)
library(dplyr)
```

Store google sheet link as a variable url

```
url<-'https://docs.google.com/spreadsheets/d/1APc89YPkXCsrhph0W2o2JSMO-5FnvU6NSwFS1wsenvs/edit?ts=5ab002cf#gid=330247896'
```

Load data (stored in the variable url) using gsheet2tbl function, variable name: algae_data

```
algae_data <- gsheet2tbl(url)
```

Check my data (head) and make sure it looks right (column names)

```
head(algae_data)
```

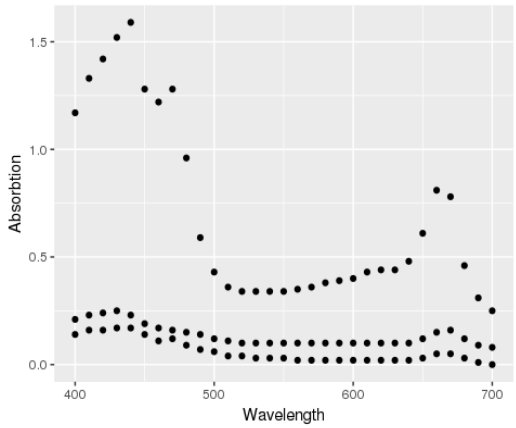
```
## # A tibble: 6 x 3
##   Algae   Wavelength Absorbtion
##   <chr>        <int>     <dbl>
## 1 Spinach        400      1.17
## 2 Spinach        410      1.33
## 3 Spinach        420      1.42
## 4 Spinach        430      1.52
## 5 Spinach        440      1.59
## 6 Spinach        450      1.28
```

```
colnames(algae_data)
```

```
## [1] "Algae"      "Wavelength" "Absorbtion"
```

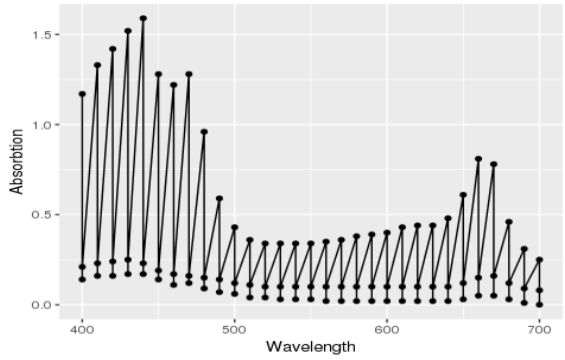
Create a plot for algae_data with points

```
ggplot(algae_data, aes(x=Wavelength, y=Absorbtion)) +geom_point()
```



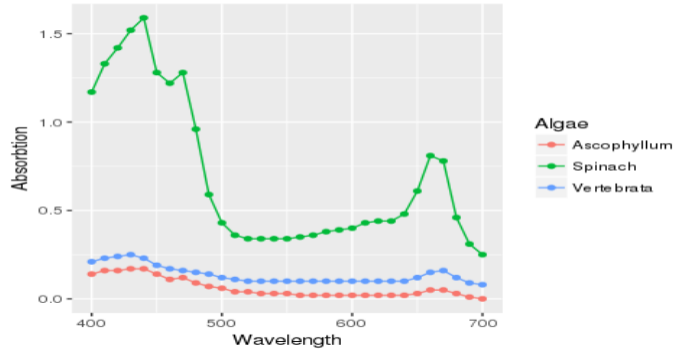
Create a plot for algae_data with points and lines

```
ggplot(algae_data, aes(x=Wavelength, y=Absorbtion)) +geom_point()+ geom_line()
```



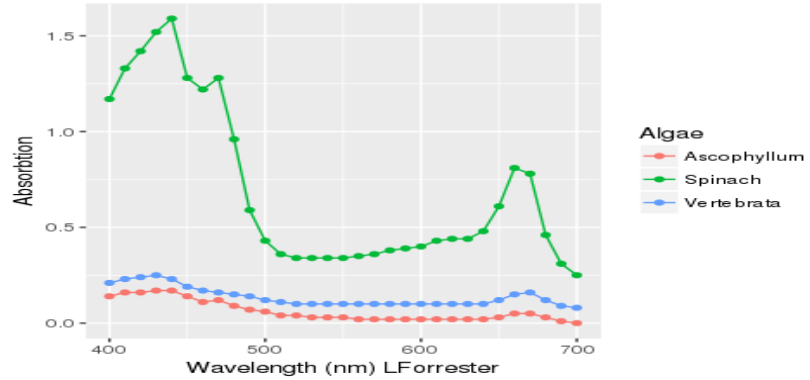
Separate out the algae sp using "color" to group the different algae (spinach vs greenAlgae vs brownAlgae). Create a plot for algae_data with lines

```
ggplot(algae_data, aes(x=Wavelength, y=Absorbtion, color=Algae)) +geom_point()+ geom_line()
```



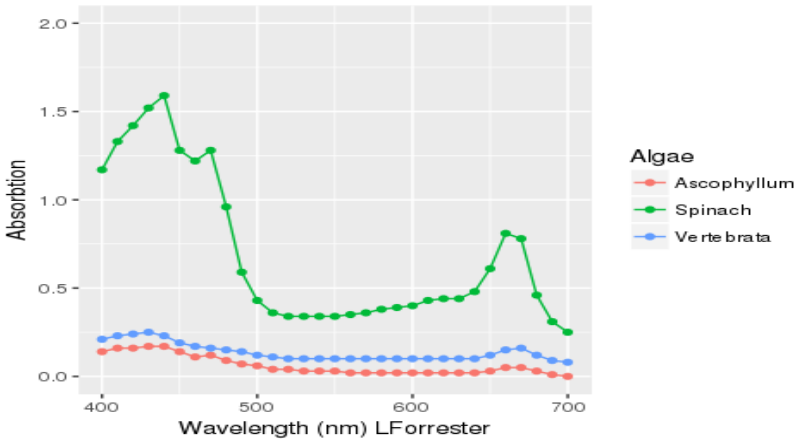
Label my plot axis titles

```
ggplot(algae_data, aes(x=Wavelength, y=Absorbtion, color=Algae)) +geom_point()+geom_line()+ scale_x_continuous(name ="Wavelength (nm) LForrester")+  scale_y_continuous(name="Absorbtion")
```



Change axis to change limit range of y-axis

```
ggplot(algae_data, aes(x=Wavelength, y=Absorbtion, color=Algae)) +geom_point()+geom_line()+ scale_x_continuous(name ="Wavelength (nm) LForrester")+scale_y_continuous(name="Absorbtion", limits = c(0,2.0))
```

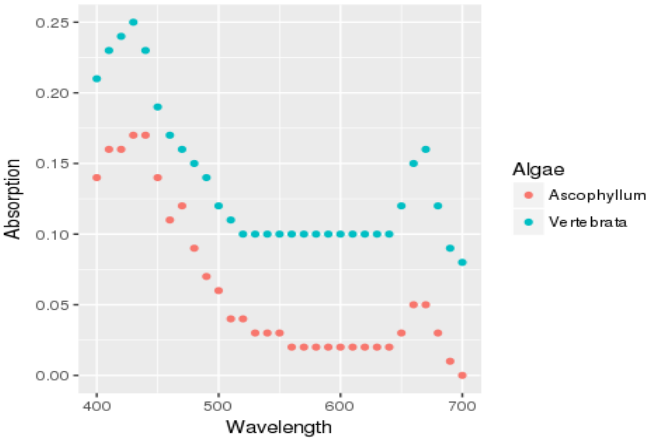


IF the spinach was REALLY HIGH abs and the algae had really low Abs, then we might want to plot the algae separately from the spinach, so that you can see where the peaks are.

We plot the algae separately so we can see the peaks, to write up results.

Use the filter function to plot ONLY the Algae, without the Spinach, so you can see peaks in algae Abs. You must have the 'dplyr' library installed to run the 'filter' function.

```
Algaeonly <- filter(algae_data, Algae=="Vertebrata"|Algae=="Ascophyllum")
ggplot(Algaeonly, aes(x=Wavelength, y=Absorbtion, color=Algae))+ geom_point()+ scale_x_continuous(name="Wavelength")+ scale_y_continuous(name="Absorption")
```



Appendix D

Lab Assignment 4: Seedling Growth - Answers v. 2184

```
---
title: "BIO104_A4_Answers"
output: html_document
---
## 1. Load the packages into R
# Load the packages ggplot2, gsheet, and dplyr
library(ggplot2)
library(gsheet)
library(dplyr)


## 2. Get your data into R

url <-  'https://docs.google.com/spreadsheets/d/1MOkh3SNsuuTixjRFi4ptYmjimUJ9rGXIta1iJDjre9k'

# Load the data stored in the variable url using the gsheet2tbl function
# and store it as the variable   plant_data

plant_data <- gsheet2tbl(url)

Use ‘head’ to check to make sure the data was imported correctly.
# Check your data to make sure it looks right (first lines only and column names)
head(plant_data)
colnames(plant_data)

## 3. Graphing the Seedling Data

Because we want to compare the seedlings grown under light and dark conditions, you are going to graph both sets of data together. You will first graph the individual data as points. This way you can see how much variation is in your data.

### 3a. Create the base layer of your plot
* What is the independent variable (x axis)?
* What is the dependent variable (y axis)?
* How will you plot light and dark conditions separately?

# Create the base layer of your plant_data plot using ggplot

ggplot(plant_data, aes(x=day, y=height, color=exposure))### 3b. Add the data and labels to the plot
# Add datapoints and labels to the plot using geom_point

ggplot(plant_data, aes(x=day, y=height, color=exposure))+
  geom_point()+
  scale_x_continuous(name="Day") + scale_y_continuous(name="Mean Height (mm)") +
  labs(color="Exposure")

## 4. Compare the Height of Seedlings in Different Conditions
* Group the data by exposure and day. _Refer to Lab A2, Part 3 if you need help._
# Group the plant_data variable by exposure and day using the group_by function
# and store it as the variable grouped_plant_data
grouped_plant_data <- group_by(plant_data, exposure, day)
```

Calculate the mean `height` of each group. _Refer to Lab A2, Part 3 if you need help._
* Add `na.rm=TRUE` into the `mean()` function. This additional code tells R to ignore areas in the data set that had no values entered. For example, if no seedlings grew under your dark condition, you would not record any data under height for that experiment. By using this code, R can now ignore these blank spaces and calculate the mean correctly.

```
# Calculate the mean height of the seedlings at each interval using the summarise function and store it as the variable plant_data_means
 Use na.rm=TRUE to ignore NA values in the data

plant_data_means <- summarise(grouped_plant_data, mean = mean(height, na.rm=TRUE))

#Check the first few lines of your mean data (`plant_data_means`) using `head`.

head(plant_data_means)

## 5. Graphing the Mean Seedling Data

### 5a. Create the base layer of your plot and add points
# Make your plot for the plant_data_means data using the ggplot command

ggplot(plant_data_means, aes(x=day, y=mean, color=exposure))+
  geom_point()

### 5b. Calculate the standard deviation of the means
# Remake your table of means so it includes std deviation
# and store the additional data as the variable plant_data_means

plant_data_means <- summarise(grouped_plant_data,
                      mean = mean(height, na.rm=TRUE),
                      stdev = sd(height, na.rm=TRUE))

### 5c. Add standard deviation bars to your plot
# Add the error bars to the plot using geom_errorbar

ggplot(plant_data_means, aes(x=day, y=mean,color=exposure))+
  geom_point()+
  geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev))

### 5d. Add labels to your plot
* Add labels to the axes and legend
* Change the width of the error bars by adding `width = 0.2` after `ymax` in `geom_errorbar`.

# Label your plot and change axes titles
# Edit the width of the error bars
# Add the bars to your plot
ggplot(plant_data_means, aes(x=day, y=mean,color=exposure))+
  geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev, width=.2))+
  geom_point()+
  labs(x="Day", y="Mean height (cm)", color="Exposure")
```

5e. Draw lines representing a trend that fits your data. Linear regression.
* Add the model lines to your graph by adding a layer using `geom_smooth`.
* There are two lines: one for each subset of data.
* Add ` + geom_smooth(method="lm", se = FALSE)` to your ggplot command.

```
# Add the regression to your plot using geom_smooth
# Add the bars to your plot
ggplot(plant_data_means, aes(x=day, y=mean,color=exposure))+
  geom_errorbar(aes(ymin=mean+stdev, ymax=mean-stdev, width=.2))+
  geom_point()+
  labs(x="Day", y="Mean height (cm)", color="Exposure")+
  geom_smooth(method="lm", se = FALSE)
```

6. Compare height under different conditions using statistics (a t-test)

Just like in Lab A2, we have two groups of data we want to compare to see if there is a statistical difference. In this case, we're interested in whether there is a difference between seedling heights in the 6 day old seedlings.

* Filter light and dark grown seedlings for the day 6 time period. _

```
  # Subset plant data by exposure and growth time
 # and store as the variables light6days and dark6days

 Light6Day <- filter(plant_data, exposure=="light" & day=="6")
 Dark6Day <- filter(plant_data, exposure=="dark" & day=="6")
```

 * What is your null hypothesis for the 6 day old seedling data?

```
# T TEST:  Use t test to compare heights of 6 day old seedlings for light and dark conditions

t.test(Light6Day$height, Dark6Day$height)
```

Appendix E
Cookbook for R: Explanations of the R Code used in these Labs

Using R to manipulate and analyze data can be somewhat intimidating to new users. This 'cookbook' aims to help explain the commonly used syntax for the basic data input, analysis, and plotting. Included below are examples from the four labs in this article, as well as a generalized "formula" version of the code needed to complete each task. This document will supplement the instructions and explanations within these Lab Manuals.

For ease of use: all functions are red, all input variables are green, and all output variables are purple.

Table of Contents:

1. Load a Package

Example: library(gsheet)

Formula: library(1)

1 = Name of the package you would like to load into R. Packages greatly expand the functionality of R. Commonly used packages include: gsheets, dplyr, and ggplot2.

2. Load in Data from Google Sheets (Use the Web Address, Called the URL:Uniform Resource Locator)

Example:
url <- 'https://docs.google.com/spreadsheets/d/1HslZUdsqHo9wIoIMywSINfAAw1mElGJrAqkbLzdLw4g/edit#gid=0'

Formula: 2 <- '3'

2 = How you want to refer to the URL (as shorthand) within R. This will only be used once to convert the data into a table within R, so something easy and descriptive like "url" is appropriate. If you are inputting multiple data sets in the same R script, add numbers (e.g. "url2", "url3", etc.) to avoid overwriting any data.
3 = The URL of the googlesheet data being used. Single quotes must surround the full URL. Note: Any googlesheet used must have the share settings so that it is publicly available on the web.

3. Convert Google Sheet to a Variable in R

Example: diversitydata <- gsheet2tbl(url)

Formula: 4 <- gsheet2tbl(5)

4 = The new variable you want to create from the input dataset.
5 = The shorthand reference to the URL with the input data (2 from above example).

4. Filter Data

Example: FOXonly <- filter(diversitydata, SiteType=="FOX_incoming" | SiteType=="FOX_outgoing")

Formula: 6 <- filter(7, 8=="9" | 8=="10")

6 = Variable you want to create that will contain ONLY the specified data
7 = Input dataset you want to filter from
8 = The column within the input dataset (7) you want to filter specific values from
9 = The first specific value you want to include in the new variable (6). The spelling and lettercase must be an exact match to the column name in the input dataset (7).
10 = The second specific value you want to include in the new variable (6). The spelling and lettercase must be an exact match to the column name in the input dataset (7).

Note: This example will allow you to filter two unique values from the same column of the input dataset into a new variable within R.
To filter more than two unique values, add more terms using the vertical bar (|) followed by the next value you would like to filter.

5. Make a Boxplot with GGPlot

```
Example: ggplot(diversitydata, aes(x=SiteType,y=Diversity) +
                geom_boxplot() +
                scale_x_discrete(name="Conditions at Site") +
                scale_y_continuous(name="Simpson Diversity Index")

Formula: ggplot(11, aes(x=12, y=13) +
                geom_boxplot() +
                scale_x_discrete(name="14") +
                scale_y_continuous(name="15")
```

11 = Input dataset you want to graph. The spelling and lettercase must be an exact match to the variable name.
12 = The data within this dataset(11) that you want to graph on the x-axis. The spelling and lettercase must be an exact match to the column name in the input dataset(11).
13 = The data within this dataset(11) that you want to graph on the y-axis. The spelling and lettercase must be an exact match to the column name in the input dataset(11).
14 = The text you want to use to label the x-axis. This text must be in double quotation marks, and it will appear on the graph exactly as it is typed. Note: This part of the code can also be scale_x_continuous(name = "14") if the data being plotted on the x-axis is continuous (e.g. time, etc.)
15 = The text you want to use to label the y-axis. This text must be in double quotation marks, and it will appear on the graph exactly as it is typed.

6. Group Data

Example: grouped_snail_data <- group_by(snail_data, exp, generation, snailcolor)

Formula: 16 <- group_by(17, 18, 19, 20)

16 = Output variable grouped by the specified columns of the input dataset
17 = Input dataset (must be a variable in R—See 3. Convert Google Sheets to a Variable in R).
18 = First column of the input dataset (17) that will be included in the grouped output variable (16).
19 = Second column of the input dataset (17) that will be included in the grouped output variable (16).

20 = Third column of the input dataset (17) that will be included in the grouped output variable (16).

Note: This example includes three columns (18, 19, 20) in the output variable (16) that are from the input dataset (17). It is possible to group using more or fewer columns depending on the dataset and analysis goals.

7. Calculate Means and Standard Deviations of Grouped Data

```
Example: snail_data_means <- summarise(grouped_snail_data,
                                       mean = mean(snails, na.rm=TRUE),
                                       stdev = sd(snails,na.rm=TRUE)

Formula: 21 <- summarise(22,
                         mean = mean (23, na.rm=TRUE),
                         stdev = sd (23, na.rm=TRUE))
```

21 = Output variable that will be created and contain the calculated mean value
22 = Input dataset that must have groups established (see 6. Group Data)
23 = Name of the column within the input dataset (22) that you calculate the mean and standard deviation of

8. Make a Scatterplot with GGPlot

```
Example: ggplot(snail_data_means_sd_exp1, aes(x=generation,y=mean, color=snailcolor)+
                geom_point() +
                scale_x_continuous(name="Generation")+
                scale_y_continuous(limits = c(0,50), name="Mean")

Formula: ggplot(24, aes(x=25,y=26, color=27)+
                geom_point() +
                scale_x_continuous(name="28")+
                scale_y_continuous(name="29", limits = c(30,31))
```

24 = Input dataset you want to graph. The spelling and lettercase must be an exact match to the variable name.
25 = The data within this dataset (24) that you want to graph on the x-axis. The spelling and lettercase must be an exact match to the column name in the input dataset (24).
26 = The data within this dataset (24) that you want to graph on the y-axis. The spelling and lettercase must be an exact match to the column name in the input dataset (24).
27 = The data within this dataset (24) that will be used to change the color the points on the graph. This is often another variable within the data that is important to recognize when we want to compare something (e.g. populations, etc.). The spelling and lettercase must be an exact match to the column name in the input dataset (24).
28 = The text you want to use to label the x-axis. This text must be in double quotation marks, and it will appear on the graph exactly as it is typed.
29 = The text you want to use to label the y-axis. This text must be in double quotation marks, and it will appear on the graph exactly as it is typed.
30 = The desired lower limit of the y-axis. Must be a number .
31 = The desired upper limit of the y-axis. Must be a number.

a. Make a Line Graph with GGPlot

The example above will plot the data points from the input dataset. To connect these points with a line, add the following to your ggplot command: + geom_line()

```
Example: ggplot(snail_data_means_sd_exp1, aes(x=generation,y=mean, color=snailcolor)+
                geom_point() + geom_line()
                scale_x_continuous(name="Generation")+
```

```
scale_y_continuous(limits = c(0,50), name="Mean")
```

b. Add a Regression Line to a Scatterplot with GGPlot

The example above will plot the data points from the input dataset. To plot a line of best fit (linear regression) add the following to your ggplot command: + geom_smooth(method="lm", se = FALSE)

Note: A regression line is very different than making a line graph. They are rarely, if ever, used together

9. Conduct a t-Test

Example: t.test(Red1$snails, White1$snails)

Formula: t.test(32$33, 34$35)

32 = Input dataset for first group being compared. This dataset is most likely one that was filtered from the initial dataset (see 4. Filter Data)
33 = Column within this first dataset (32) that we want to use to compare
34 = Input dataset for second group being compared. This dataset is most likely one that was filtered from the initial dataset (see 4. Filter Data)
35 = Column within this second dataset (34) that we want to use to compare

Note: The $ is used the specify a particular column within the dataset that will be analyzed using the t-test.

Appendix F
The Plankton Community of Narragansett Bay: A Field Guide to Common Species

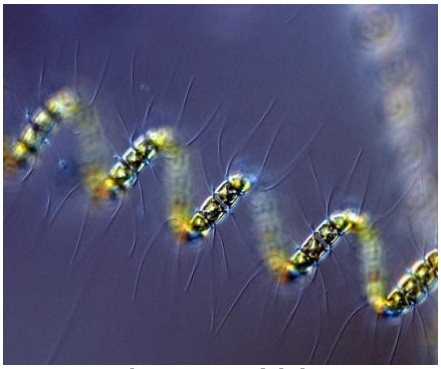
In this lab we will group common species of marine phytoplankton and zooplankton that are found in Narragansett Bay into 7 different groups. Using the microscope, you will count the number of different phytoplankton types in your preserved samples. We will examine the species diversity (using these group types as species).

Phytoplankton Types: Chains WITH spines, Chains WITHOUT spines, Pill box shape, Needle shape, *Ceratium* Boat shape, Polyhedron shape, Round shape, Pointy diamond shape
Mesozooplankton (0.2-20mm): Copepods

Chains **WITH** spines

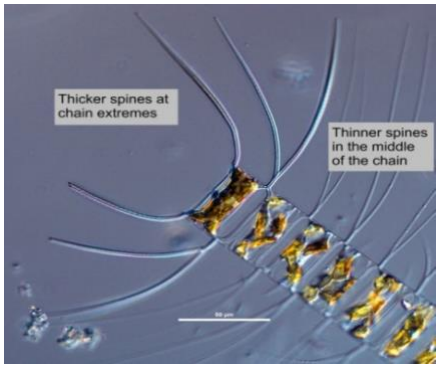
Chaetoceros curvisetus

Diameter: 8 – 50 µm
Long Silica Spines



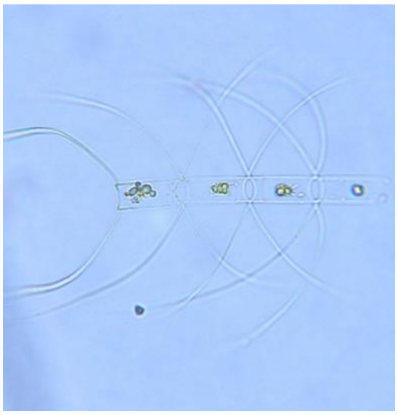
Chaetoceros debilis

Diameter: 8 – 40 µm
Length: 6 – 20 µm
Curved chain and spines



Chaetoceros decipiens

Diameter: 7 – 85 µm
Length: 9 – 35 µm



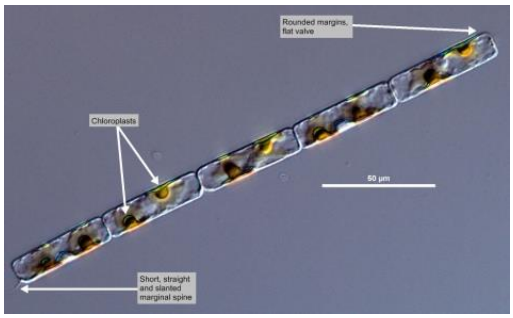
Chaetoceros diversus

Diameter: 8 – 40 µm
Spines pointing in different directions.

Chaetoceros didymus

Diameter: 10 – 40 µm
Differentiated terminal spines.
Bulbous shape at concave valve face.

Chains **WITHOUT** spines



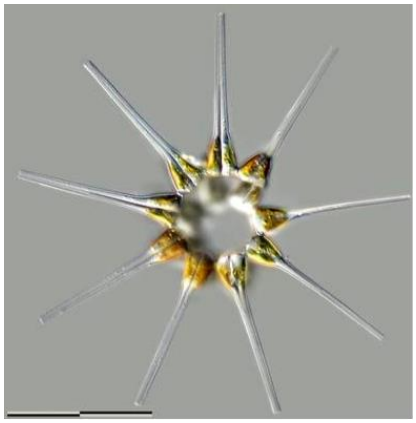
Guinardia delicatula

Diameter: 9– 22 µm
Length: 15 – 120 µm
One short fine spine pointed diagonally from thescorner of the chain end. None others

***Leptocylindrus* sp.**

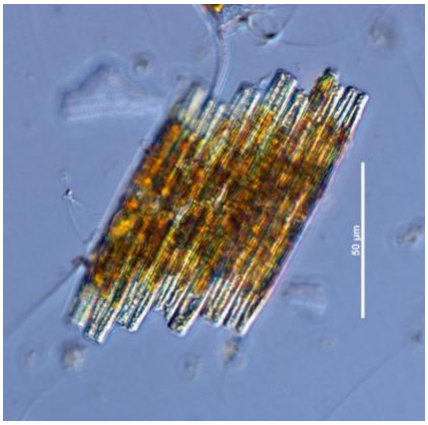
Diameter: 5 – 20 µm
Length: 20-50 µm
Central parts of the valve connections can be slightly concave or convex. No spines.

Needle shape



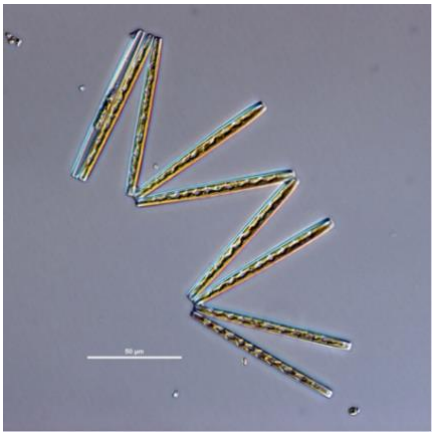
Asterionellopsis glacialis

Length: 30 – 150 µm
Connected by two blunt processes.



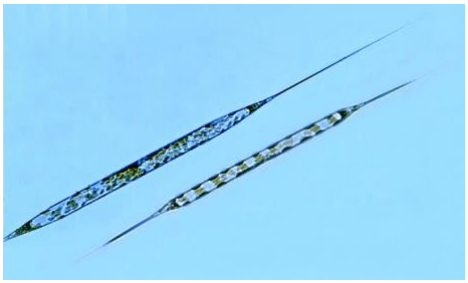
***Bacillaria* spp.**

Width: 5 – 8 µm
Length: 70 – 200 µm
B. paxillifera is pictured.



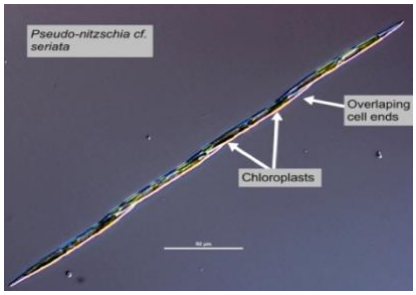
Thalassionema nitzschioides

Length: 10 – 110 µm
Chains may form other shapes like stars.



***Rhizosolenia* spp.**

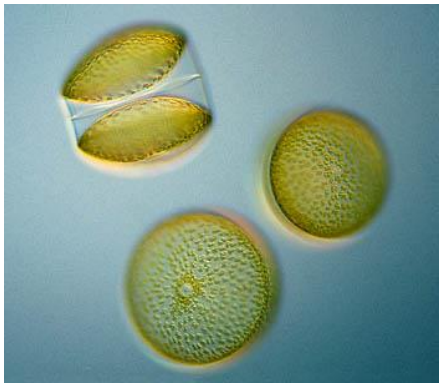
Diameter: 13-230µm
May form short chains.



***Pseudo-nitzschia* spp.**

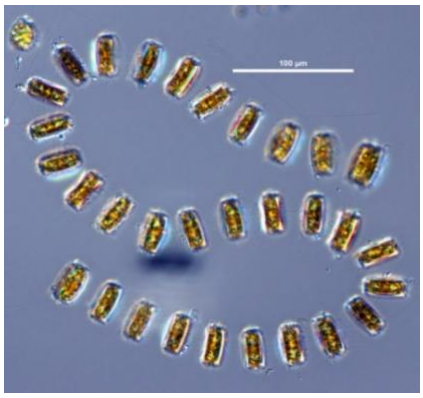
Width: 0.50 – 8 µm, Length: 25 - 160 µm
Chains formed by overlapping cells. Most species produce toxic domoic acid. *P. seriata* is pictured. Motile.

Pill box shapes



***Coscinodiscus* spp.**

Diameter: 40 – 200 µm
Height: 30 – 180 µm
Solitary



***Thalassiosira* spp.**

Size: 2 – 186 µm
Cylindrical to coin shaped cells. Cells connected by single thread from central process. *T. nordenskioeldii* is pictured. Can be unicellular or in short chains.

***Ceratium* shape**

Ceratium tripos

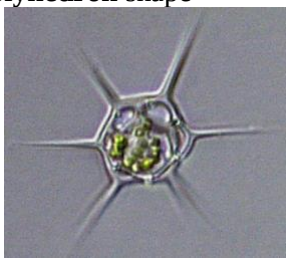
Diameter: 30 – 50 µm
Length: 70 – 200 µm
Two curved horns. Motile.

Boat shape

***Pleurosigma/Gyrosigma* spp.**

Width: 10-30µm, Length: 100-260 µm
Gyrosigma shown, but genera nearly indistinguishable.
Motile.

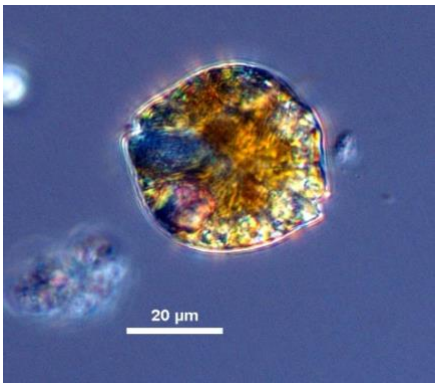
Polyhedron shape



***Dictyocha* spp.**

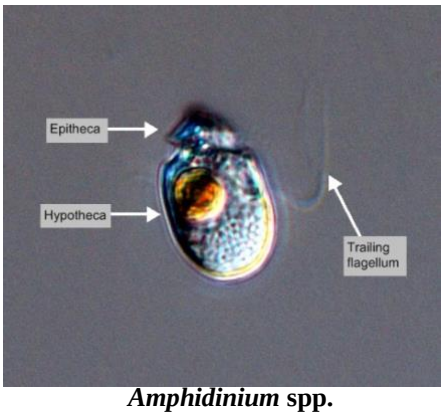
Width: 10-30µm
A silicoflagellate. Structure like the skeleton of a polyhedron with points.

Round shape



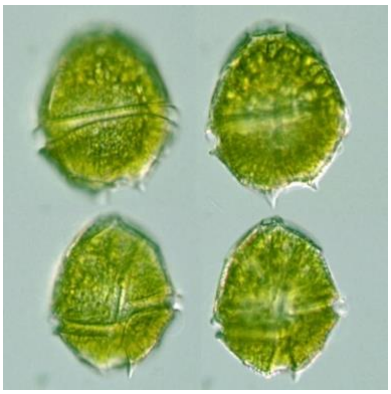
***Alexandrium* spp.**

Length: 20– 48 µm, Width: 18 – 34 µm
Two flagella. *A. catenella* is pictured.
Motile.



***Amphidinium* spp.**

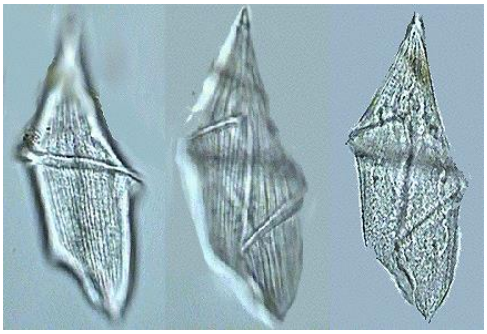
Width: 12 – 17 µm, Length: 7 – 10 µm
A. carterae is pictured. Motile.



***Peridinium* spp.**

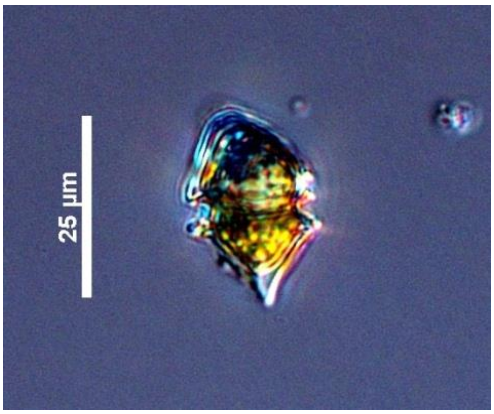
Diameter: 30 – 70 µm
Motile.

Pointy diamond shape



Gyrodinium spirale

Diameter: 20 - 45 µm,Length: 40 – 200 µm
Motile.



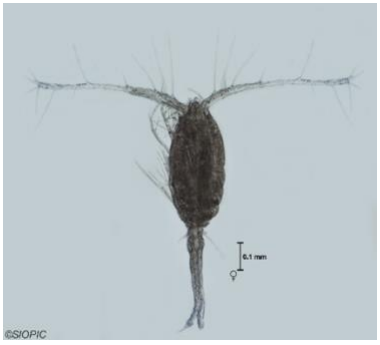
***Heterocapsa* spp.**

Diameter: 9 – 18 µm, Length: 16 - 30 µm
H. triqueta is pictured. Motile.

Mesozooplankton (0.2-20mm)
Copepods

***Calanoida* Copepod**

Length: 0.50 – 1 mm. Biramous second antennae. *Acartia hudsonica* is pictured.



***Cyclopoida* Copepod**

Length: 1.0 – 2.0 mm. First antennae is shorter than length of thorax and head. Uniramous second antennae. *Oithona similis* is pictured.

All photographs used with the permission of Dr. Susanne Menden-Deuer, University of Rhode Island, Oceanographic class, OCG 561.

Mission, Review Process & Disclaimer

The Association for Biology Laboratory Education (ABLE) was founded in 1979 to promote information exchange among university and college educators actively concerned with teaching biology in a laboratory setting. The focus of ABLE is to improve the undergraduate biology laboratory experience by promoting the development and dissemination of interesting, innovative, and reliable laboratory exercises. For more information about ABLE, please visit **http://www.ableweb.org/.**

Advances in Biology Laboratory Education is the peer-reviewed publication of the conference of the Association for Biology Laboratory Education. Published articles and extended abstracts are evaluated and selected by a committee prior to presentation at the conference, peer-reviewed by participants at the conference, and edited by members of the ABLE Editorial Board. Published abstracts are evaluated and selected by a committee prior to presentation at the conference.

Citing This Article

Forrester L, Schwartz RS. 2020. Learning basic data analysis skills in intro biology labs using R. Article 8 In: McMahon K, editor. Advances in biology laboratory education. Volume 41. Publication of the 41st Conference of the Association for Biology Laboratory Education (ABLE). **https://doi.org/10.37590/able.v41.art8**