Data Visualization Using an Open Source Statistical Program - RStudio

Author: Shaley Valentine

Grade Level: 10th – 12th grade

Group Size: 1-2 per station; full class

Setting: Computer Lab: Program installed prior to lesson

Time Needed: 2 x 50 minute sessions

Equipment Needed: Computer; Installed version of R/Rstudio (most recent; installation instructions in lesson)

Objectives

Composite Learning Objective: Students will gain a basic understanding of R statistical software and develop basic statistical and graphing skills using R. Students will use data collected by researchers to (a) compute and visualize simple summary statistics comparing groups of Lake Sturgeon; and (b) assess statistical relationships between different variables.

Knowledge Outcomes:

- Students will learn how to install and navigate the R/Rstudio environment.
- Students will learn how to use basic code input to conduct simple data visualization and computation.

Skills Outcomes

- Students will compute and visualize summary statistics for comparing two or more groups by their sample average (μ).
- Students will use R/Rstudio to assess statistical relationships between different variables.

Disposition Outcomes:

- Student will have an adequate understanding of how fisheries professionals use R/Rstudio to visualize different fish population characteristics using a threatened fish species to provide context.
- Students will understand the importance of open source programming to answer basic fisheries questions.

Summary

R is a free, open-sourced statistics program that is widely used by scientists. Statistical analyses, summary statistics, and data visualizations are important tools for students to learn and use. These tools help to simplify relationships or differences between variables. Students will go through a tutorial using program R to learn how to conduct basic summary statistics and analyses and will use graphing functions to compare groups. Students will use data collected by MSU/MiDNR Black Lake Stream-side Facility researchers to evaluate relationships between different variables.

Background

R is free statistical software that is used widely by scientists. It is free to download from the internet and all the code is open-sourced, meaning that you can look at how people created the analyses and replicate that code. A simple way of explaining code is that it is a language. In the case of R, the language the code-writer speaks is interpreted, rather than compiled. Simply put, this means that what you type is interpreted by the computer as a command, rather than as a program. So if you tell the computer to conduct an analysis, that's what it does. Whereas an application on your iPhone is a program, which requires each request to be confirmed by a coded program. As a result, R is intuitive and easy to reproduce. Additionally, other programs like SPSS, SAS, or C++ are extremely costly and generally proprietary. As such it may be difficult to replicate analyses conducted by other researchers.

We can use R to calculate **summary statistics** including the **sample mean**, **median**, **mode**, **quantiles**, and **standard deviation**. These statistics can tell us general trends in data and among groups within data. For example, we could look at average height between males and females to get a general idea of differences between sexes (Figure 1 below). What we see is that females appear to spend less money both on lunch and dinner, and both males and females tend to spend less on lunch than dinner. Graphs like Figure 1 and summary statistics allow researchers to make generalizations about data and compare one or more groups to each other. We can also use data visualization to evaluate other relationships including those which are **predictive**. Figure 2 shows an example of a predictive relationship, where a **scatterplot** is used to show the relationship between height (in) and speed (mi/hour). In this case, we can use data visualization to infer how the value of a predictor variable can affect the magnitude of a **dependent variable**.

Figure 1. Cost of lunch organized by sex and time of day

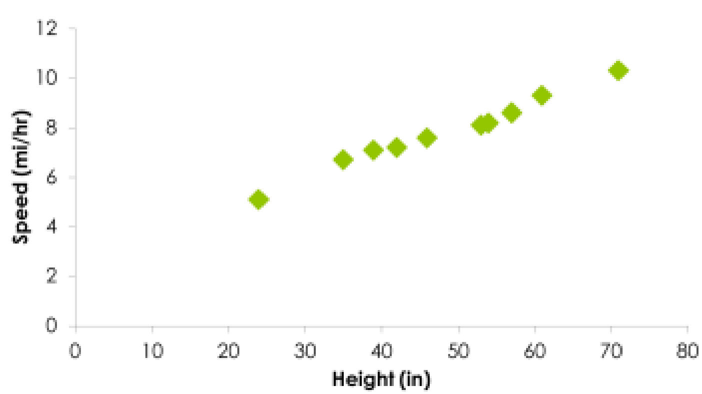


Figure 2. Walking speed (mi/hr) as a function of height (in)

Researchers use **Statistical analyses** to evaluate whether differences in data are “significant.” For example, the mean cost of lunch may appear to be different between males and females in Figure 1, but this may not be meaningfully different based on a statistical analysis. Simple analyses include like correlations (looking at how variables are related to each other) and T-tests (comparing the mean values of two groups). Analyses can be powerful, but they do have downfalls. For example, correlations show how related two variables are to one another. Often, two unrelated variables may be correlated, but that correlation may not be of consequence. For example, the price of pickles may increase as your age increases. While this may be true, it’s not your age that causes the price of pickles to increase, thus it’s important to remember that correlation does not equal causation; and that all statistics must be used with care.

In this lesson, we will use RStudio to visually and analyze data collected by researchers at the Black River Streamside Rearing Facility, where researchers have studied threatened Lake Sturgeon since 2001.

Definitions

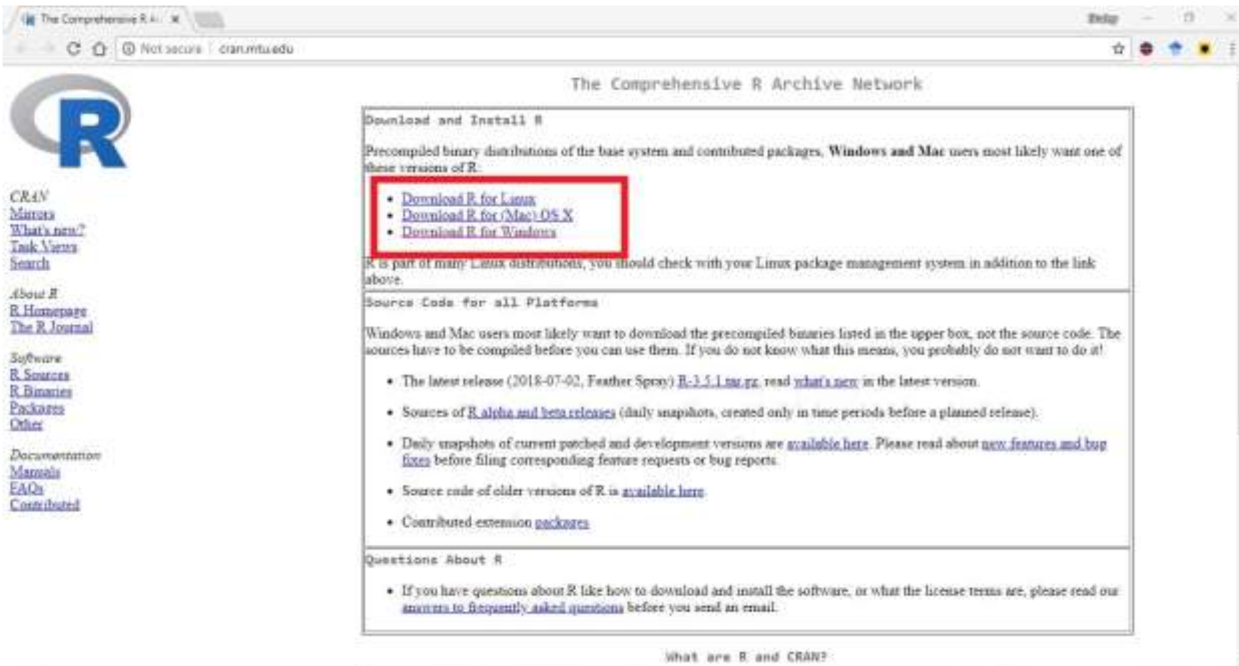
Summary statistics: information that gives a quick and simple description of data. Can include mean, median, mode, minimum value, maximum value, range, standard deviation.
Sample mean: the average of n observations from the sample. The sum of all values in a sample divided by the number of values in a sample.
Dependent variable: a variable (often denoted by y) whose value depends on that of another.
Median: The midpoint value of any sample. Where the number of observations is an even number, the median is the average of the two midpoint values.
Mode: The value in a sample population which appears most frequently.
Quantile: A subset of a sample divided into equal groups. If a population has values of 1, 2, 3, 4, 5, and six, a quantile might represent 2 units; ie: [1,2], [3,4], [4,5].
Predictive (independent variable): The process by which a value can be used to infer the magnitude of another value which directly results from the first. (a variable (often denoted by x) whose variation does not depend on that of another).
Scatter plot: a graph in which the values of two variables are plotted along two axes, the pattern of the resulting points revealing any correlation present.
Standard deviation: A value used to quantify the amount of variation of individual observations from the sample mean. Ie: One might have a sample of fish sizes: 130cm, 135 cm, 140cm and a second sample of fish sizes: 130cm, 130cm, and 145cm. Both groups have a mean of 135cm. Standard deviation allows us to determine how much each individual differs from the sample mean. In this case, group 2 deviates 8.66cm from the mean, where group 1 deviates 5cm from the mean.
Statistical analyses: mathematical comparison between one or more discrete groups and their variability.

Activity

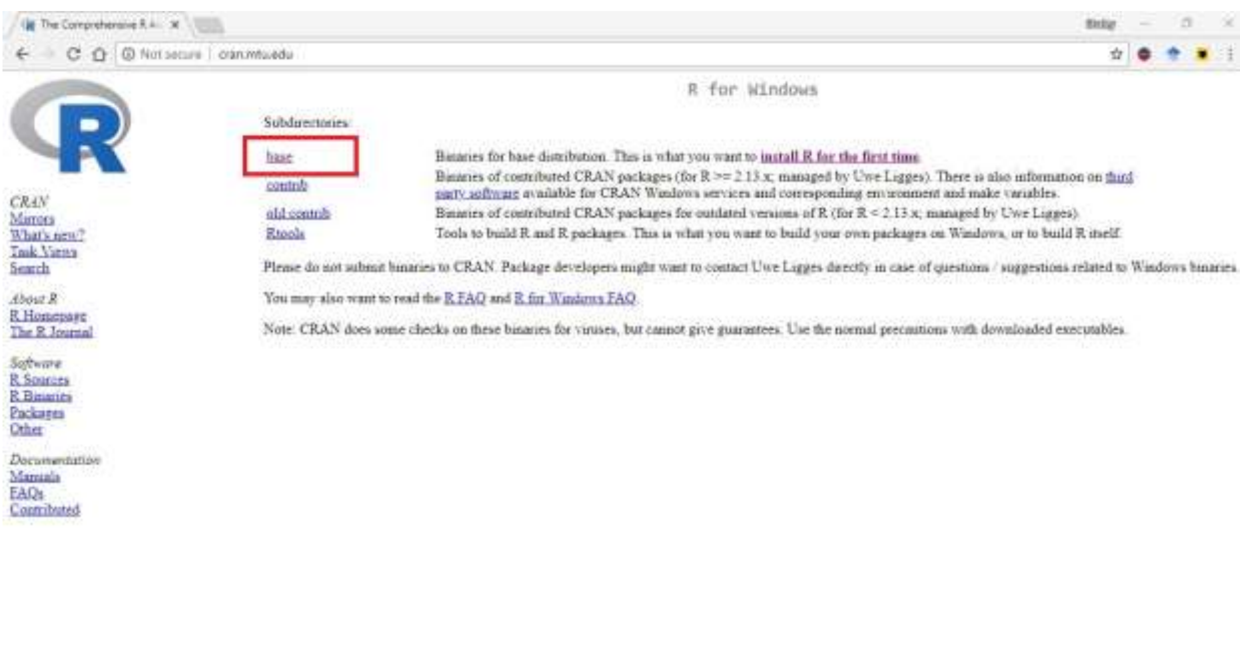
Download R

First, we need to download R. Go to the [cran-R website] (http://cran.mtu.edu/) and under "Download and Install R" click on the link that your computer is. After clicking on the link you will need to click on "base" under "Subdirectories" in the next window. In the next window click "Download R 3.x.x for Computer Type". Follow the images below for help:

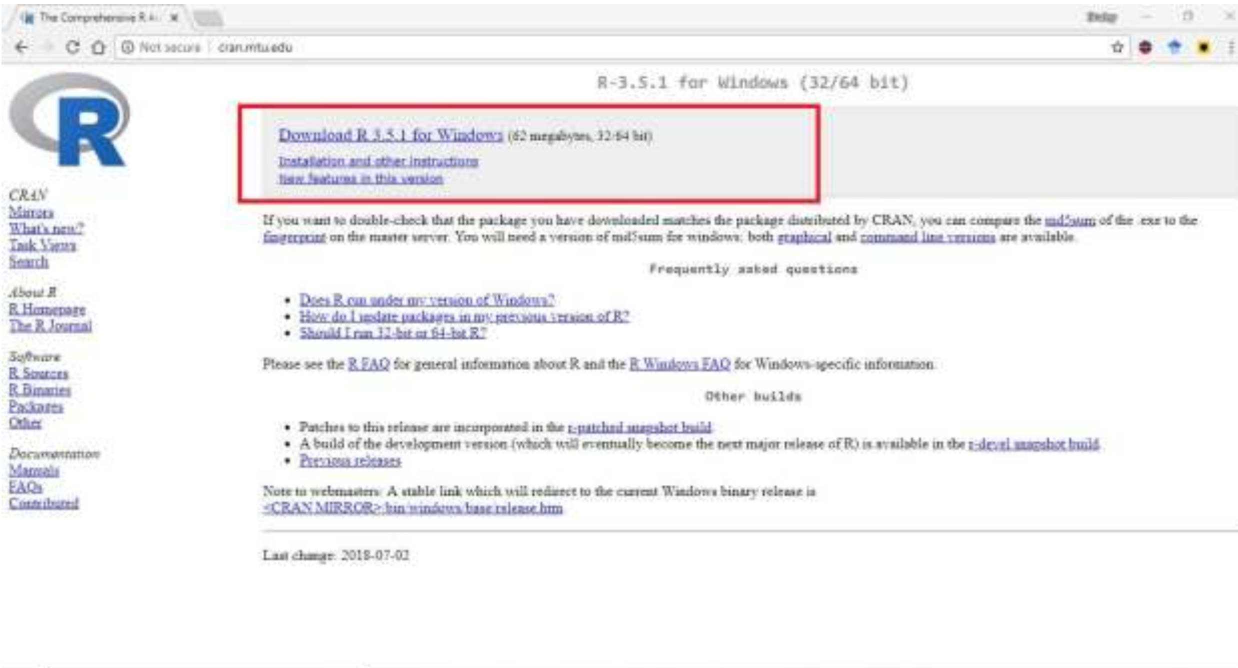
Step 1: Download and Install R



Step 2: Click on “Base”

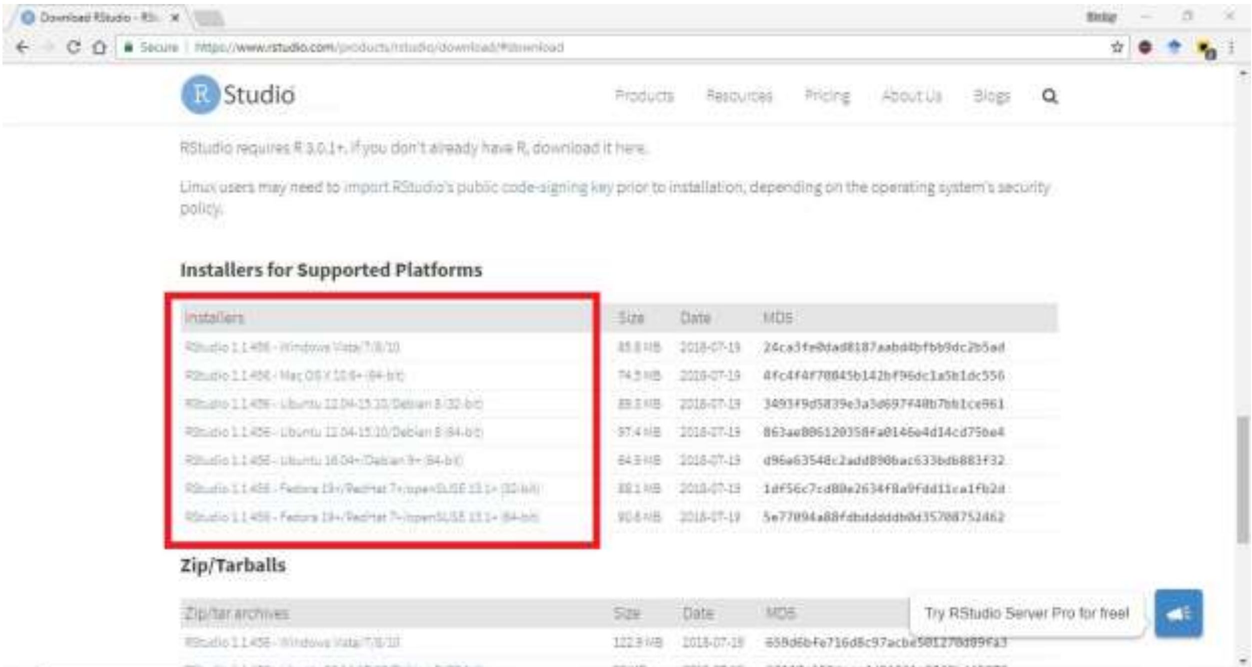


Step 3: Select “Download R 3.x.x for Windows



You will also need to download R studio. Go to the [download site](https://www.rstudio.com/products/rstudio/download/#download) and scroll down to "Installers for Supported Platforms" under "RStudio Desktop 1.x.xxx". Choose the download based on the computer you have.

Step 4: Download “R Studio”

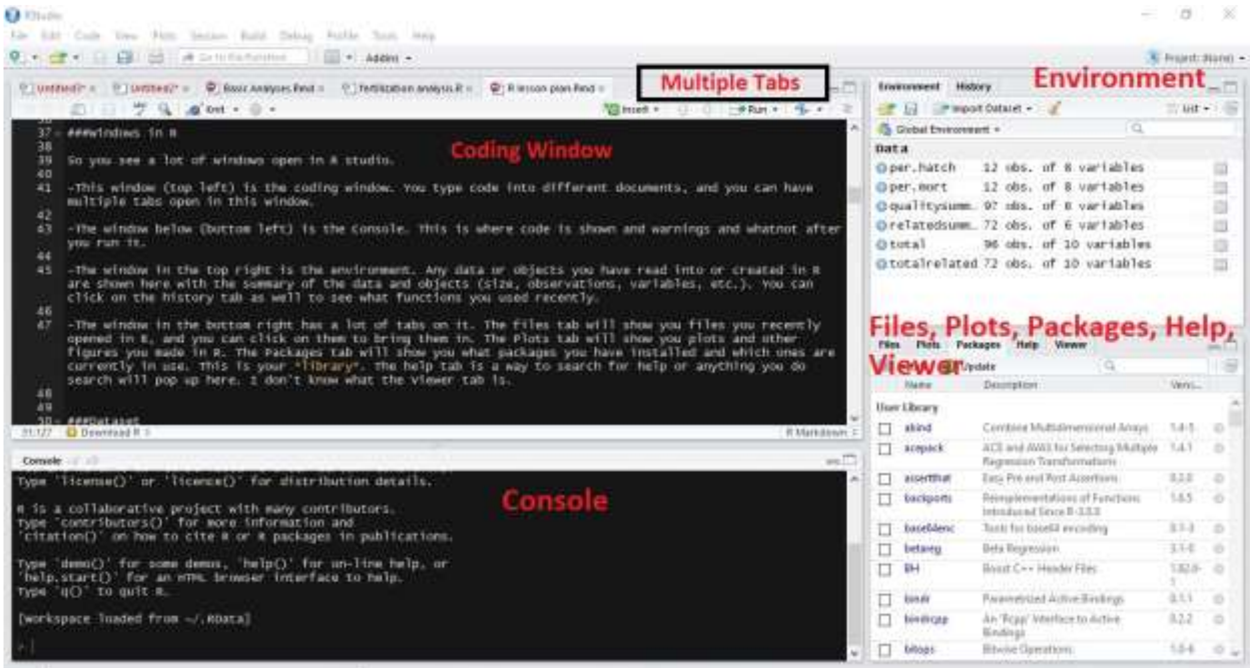


Once both R and R studio are downloaded from the website, follow their prompts to install them on the computer.

NOTE: When you use R, open R studio only. You do not need to open Base R as well. To use a simple analogy, R and RStudio work together like the motor and dashboard of a car. You can certainly get to point A with an engine, but the process is a lot easier if you can use a dashboard user interface. Working in R, one would only enter their code. One error can result in code failure and the loss of hours of work. RStudio lets the user revisit and generate code, is much more intuitive and visually pleasing, and allows for simple modifications to code, rather than an “all or nothing” approach.

###Windows in R

When you open R studio, you will see 4 windows. This section explains the purpose of each.



1. The top left window is the **Coding Window**. You type code into different documents, and you can have multiple tabs open in this window. If you've written code that you think will be useful to revisit for future projects, this window can be saved and reopened. In the above example, the author has five windows, or "**Multiple Tabs**," open. Clearly, this author is working on a few different projects.
2. The bottom left window is the **Console**. This is where RStudio conveys the results of the code written in the **Coding Window.** Here, the user can see specific error messages, and can ask RStudio for help if needed using the command 'help()'.
3. The window in the top right is the **Environment**. Any data or objects you have read into or created in Rstudio are shown here with the summary of the data and objects (size, observations, variables, etc.). You can click on the history tab as well to see what functions you used recently. More often than not, the user does not use this window.
4. The window in the bottom right has a lot of tabs on it. The files tab will show you files you recently opened in R, and you can click on them to bring them in. The Plots tab will show you plots and other figures you made in R. The Packages tab will show you what packages you have installed and which ones are currently in use. This is your *library*. The help tab is a way to search for help or anything you do search will pop up here. I don't know what the Viewer tab is.

###Dataset

We will be working with a data set of adult lake sturgeon captures from 2018. These are lake sturgeon that were handled by MSU/MiDNR researchers in May 2018. Length (centimeters; total and fork; see figure below), mass (kilogram), and girth (centimeters; diameter of fish) measurements along with gender identification are recorded in the data set.

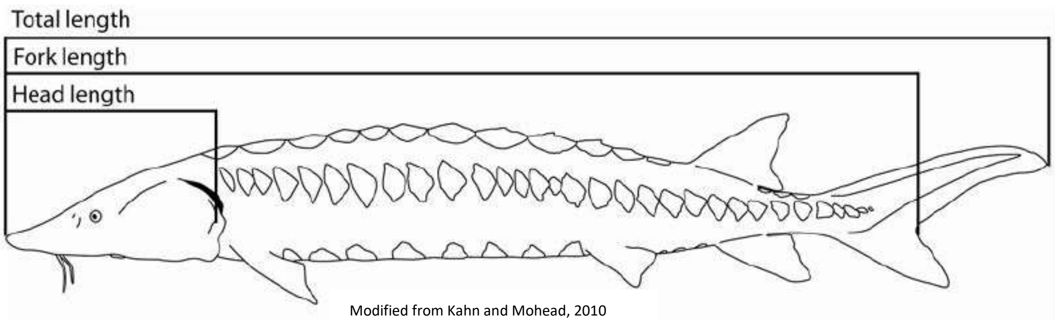
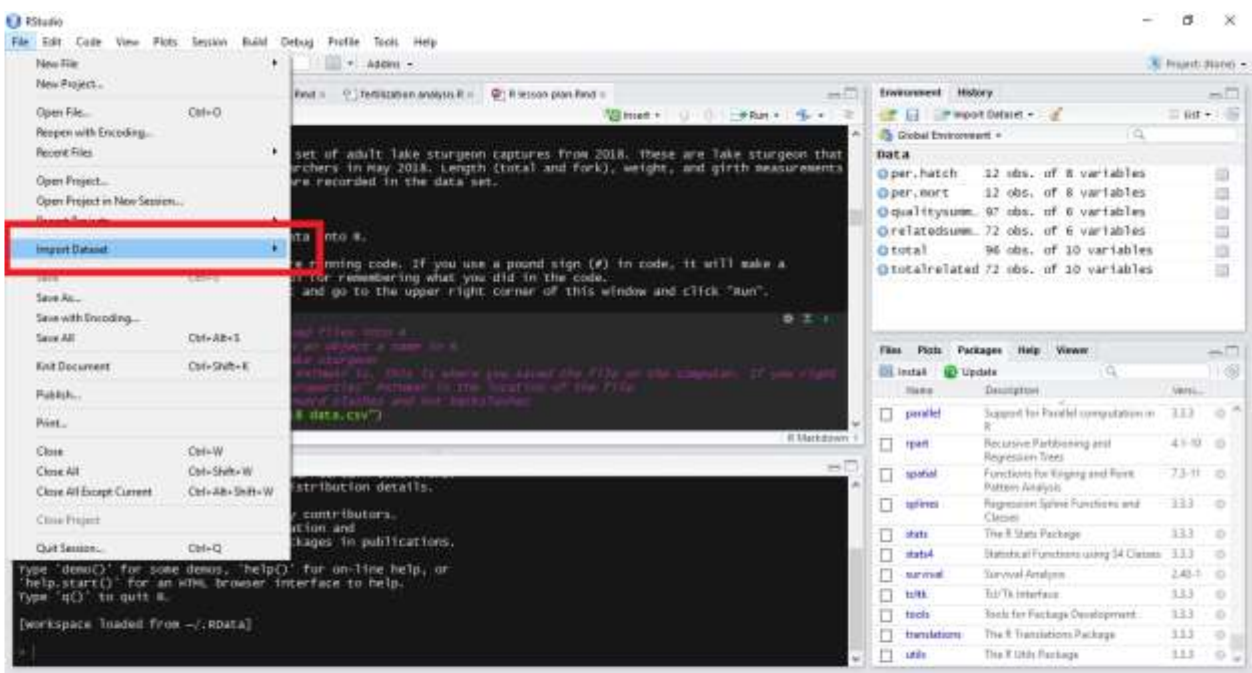


Figure 3. Total and fork length (cm) measurements on an adult Atlantic Sturgeon.

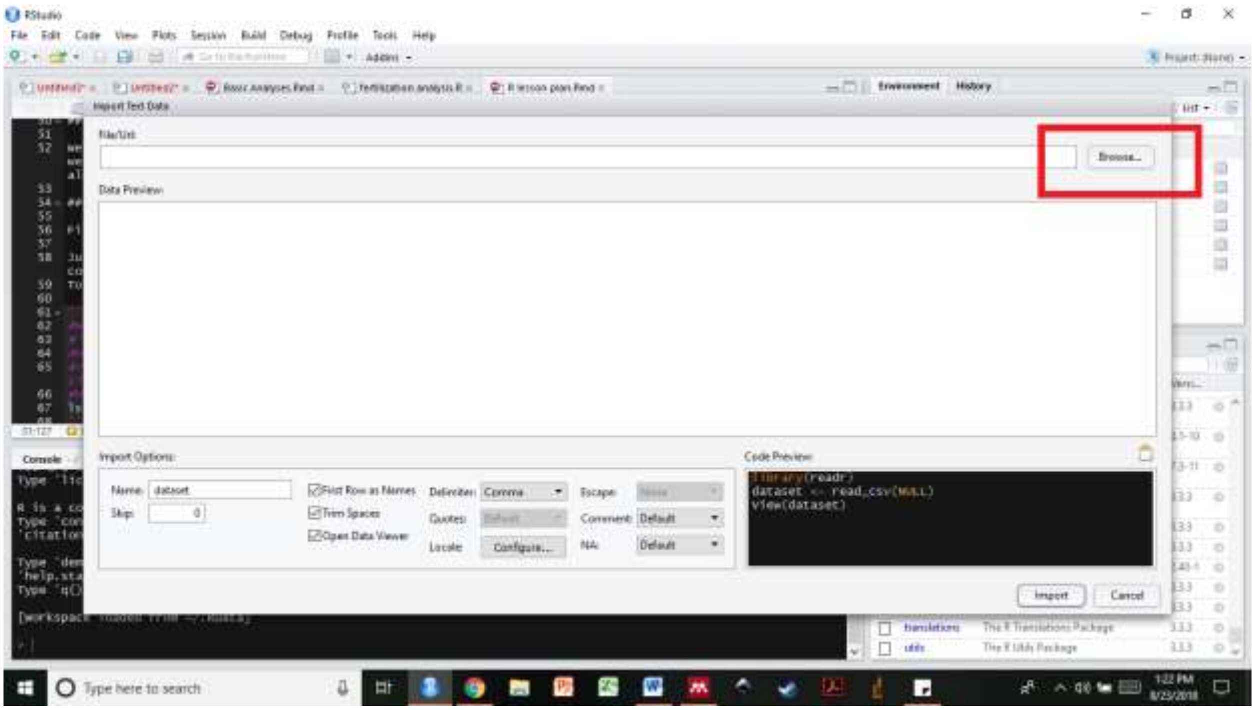
###Importing Data

First, we need to import data into R. Importing data means R will take a file on your computer and read it into the program so it is usable within R. You can find the dataset for this lesson via: https://www.glsturgeon.com/sturgeon/behavioral-ecology/lesson-14-Introduction-to-R/. The data is in an excel file, which you can download at the bottom of the page. Download the data to the desktop of your computer.

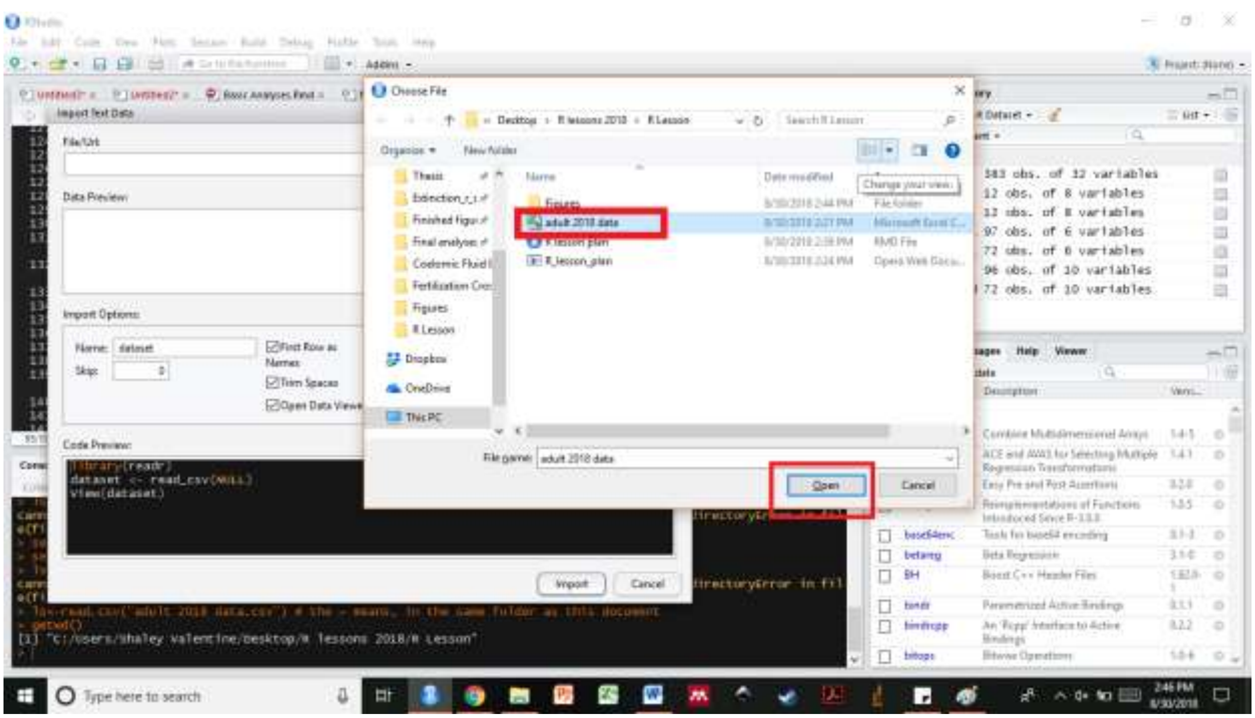
1. We need to import our data into R first. Go to File -> Import Dataset -> From CSV



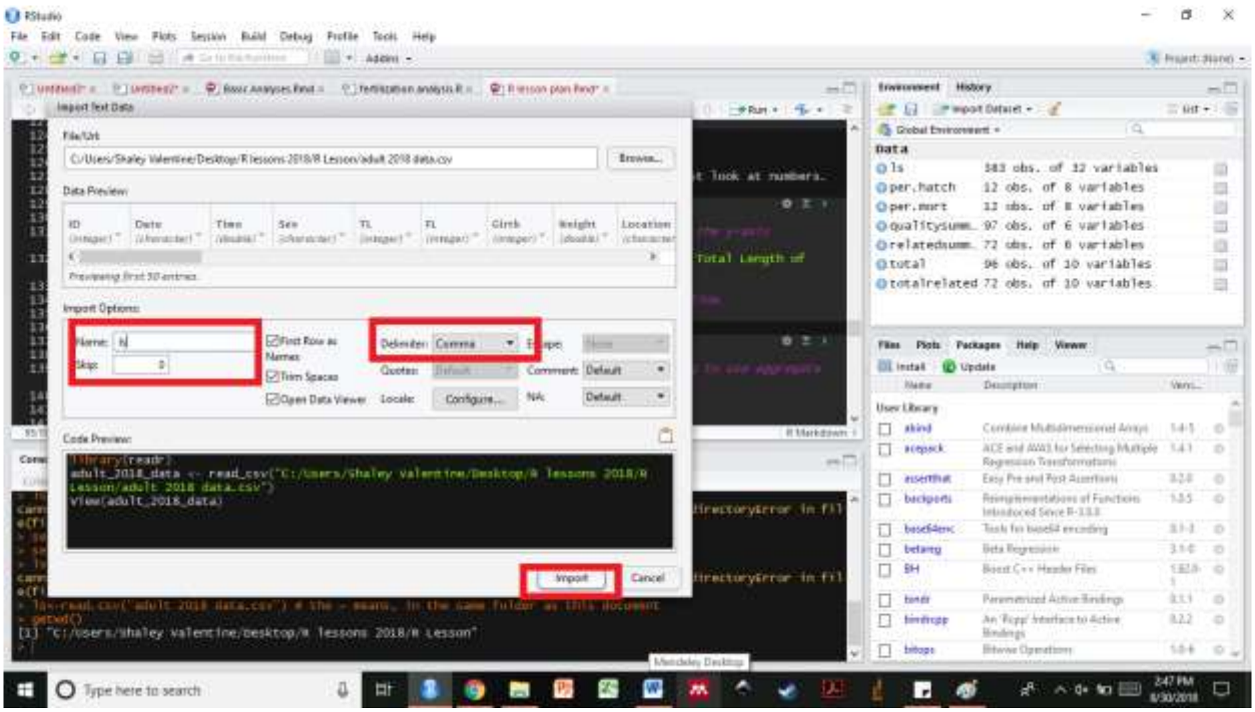
2. This next window will open once you click on Import Data from CSV. Click on the "Browse" Button.



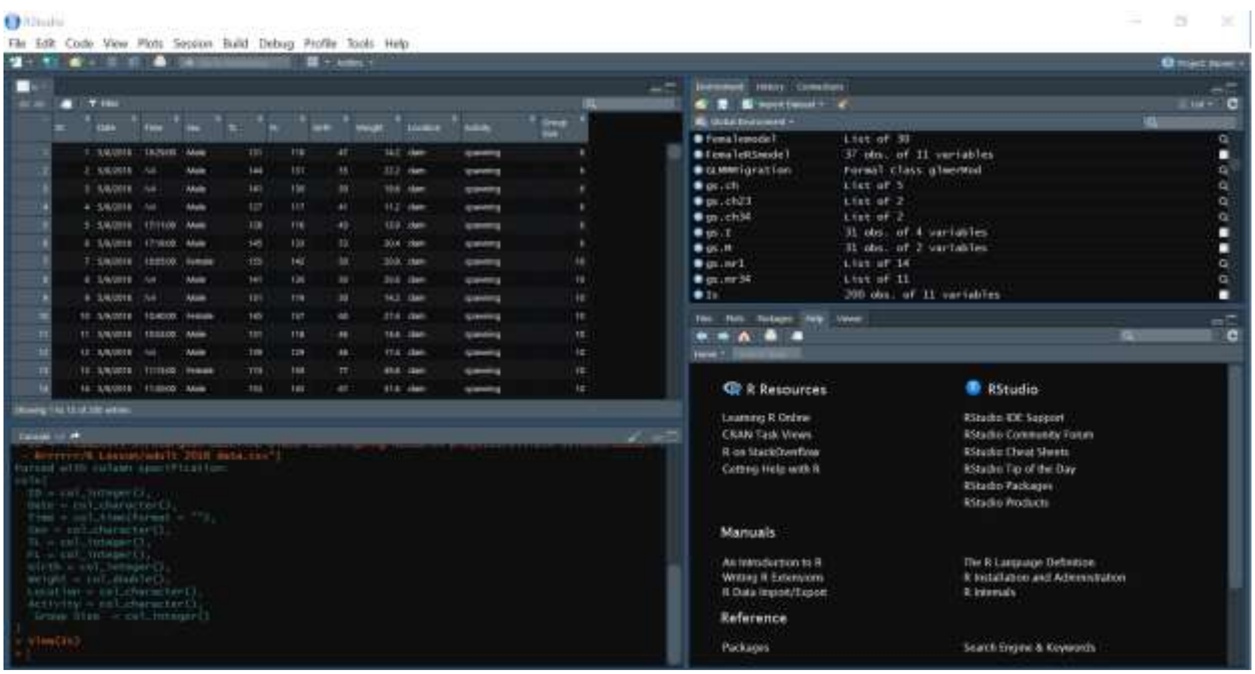
3. The file explorer will open. Find the data wherever you downloaded it, and then open it.



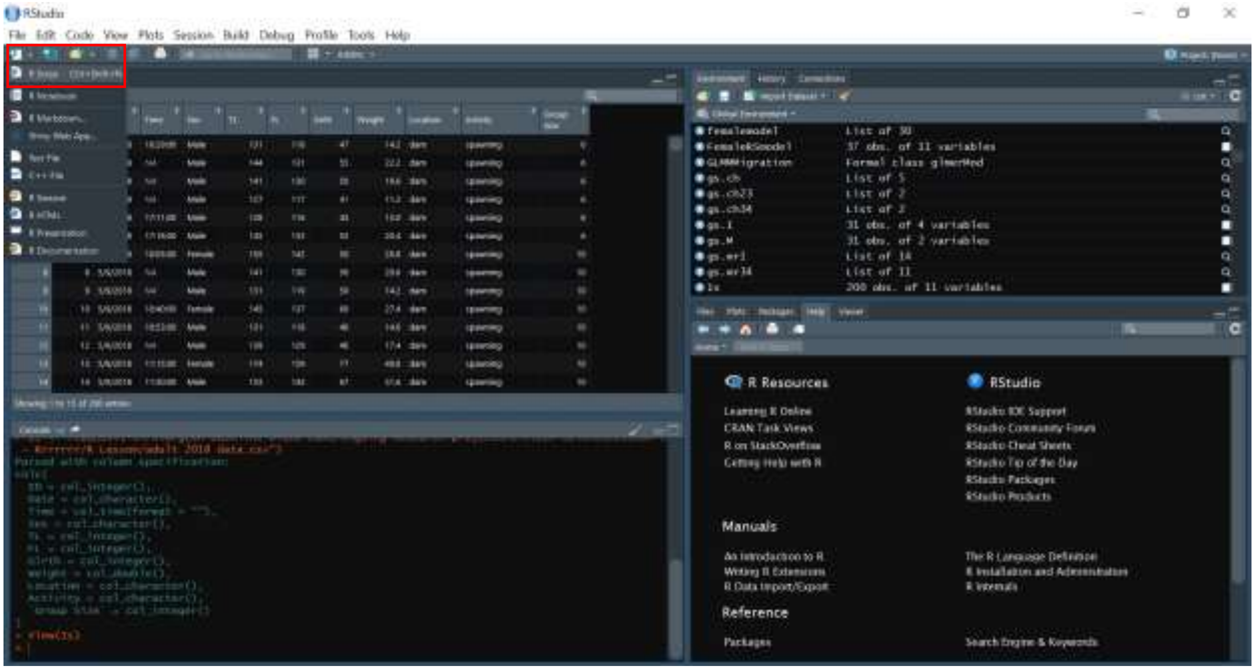
4. Before you hit "Import", make sure the "Import Options" section is filled out exactly as in the image below. Name the file "ls" (lake sturgeon). This step tells RStudio that you are importing a file which is in a comma separated format and that RStudio should read it in with the exact column names.



5. After you import, you should see the file you imported in the **Coding Window.** At any time, you can view the data by clicking on the open tab, or by entering the command View(ls) in the coding window.

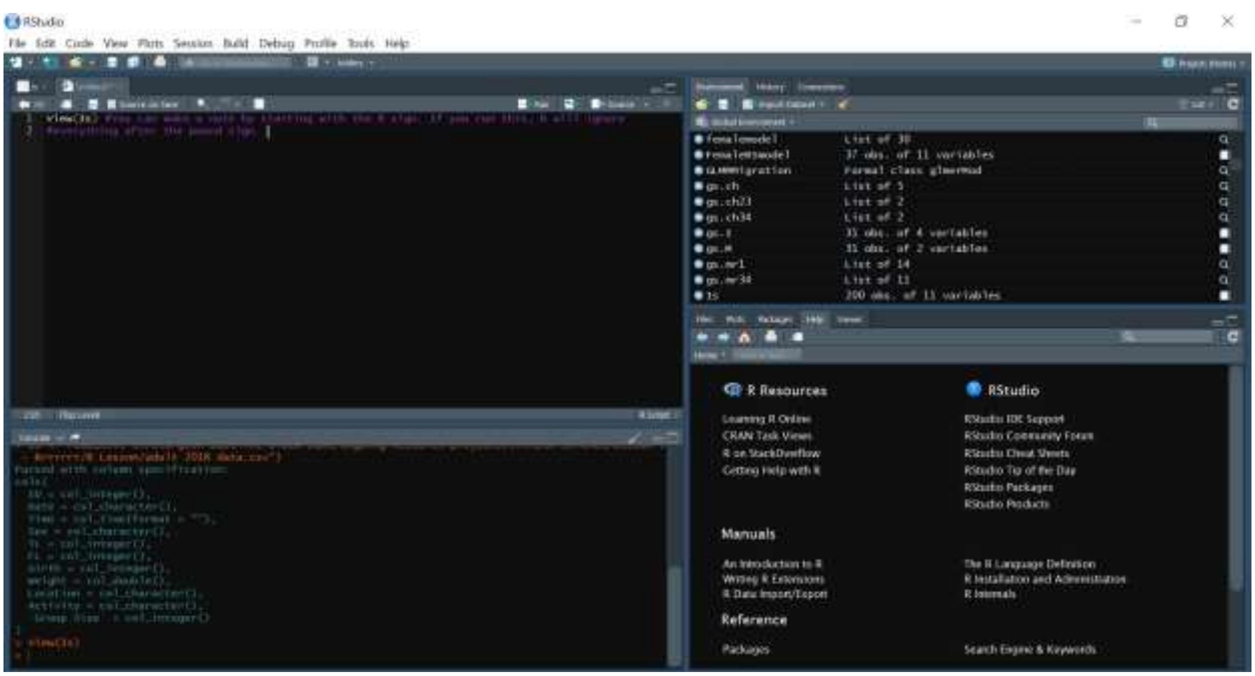


6. Before we can evaluate our data, we need to open a new R Script in the **Coding Window.** You can do this by selecting the paper and plus icon in the upper right corner, then select “R Script”.

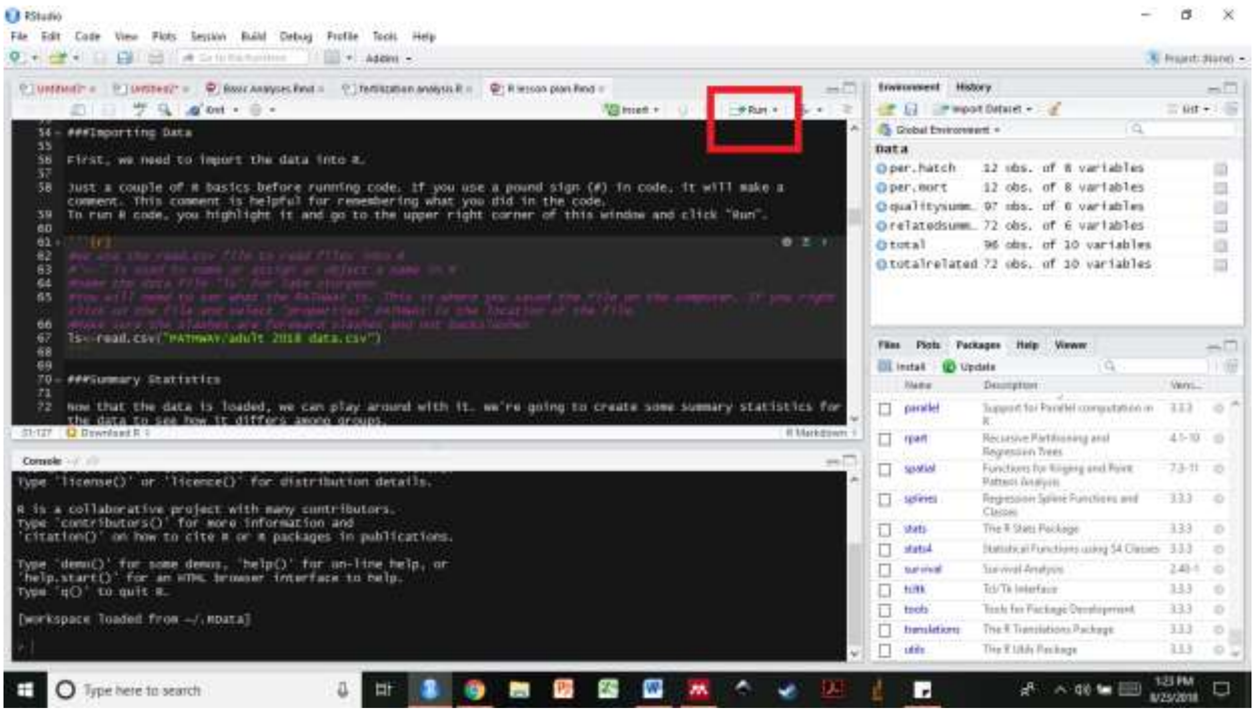


###R Basics and Summary Statistics

1. Now that the data is loaded, we can evaluate it. First, we want to view some of the **summary statistics** for the dataset. Note, if at any time you want to write notes in your code to remind you what you're doing, you can do so by starting the notes with the # sign.



2. You can also tell R to accept, or "Run" your code by highlighting the code you want to use, then select the "Run" button top middle.



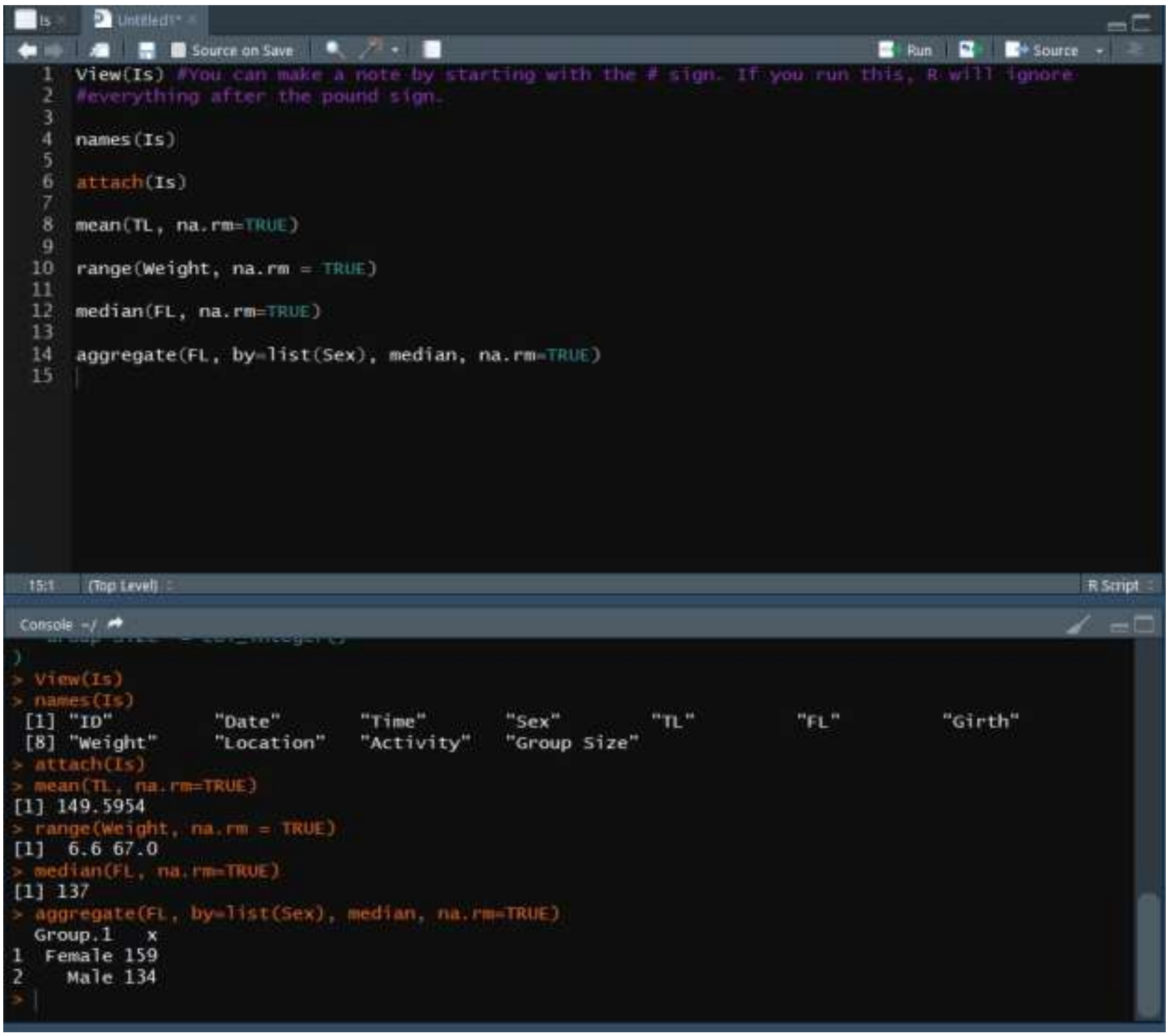
3. names(ls) – Enter the names(Is) function into the coding window, then select "Run". The names() function tells R to retrieve the column names from your dataset. This is a good way to make sure that you have all the data you need. Also, when you're running code later, you can copy and paste the names from this output. R is case sensitive, so capitalizing when you didn't in your dataset can cause code to fail.
    a. The names() function should output the following names in the **Console**: , "ID", "Date", "Time", "Sex", "TL", "FL", "Girth", "Weight", "Location", "Activity", "Group Size"

4. attach(ls) – The attach() function allows us to tell R to consider ONLY the dataset we are working with at the current time. If you are working with multiple datasets, using the attach() function tells R to focus just on one. In the **Coding Window**, enter attach(Is) and select "Run"

5. Calculating the Mean Total Length (cm). Now that the data is attached, we can tell R to calculate the mean Total Length (cm) from all fish in 2018 using the mean() function. Sometimes, your data may be missing a measurement. It happens in the field. We can tell R to not consider cells in which there is no data using the na.rm = TRUE command. To calculate the mean Total Length, enter mean(TL, na.rm=TRUE) in the **Coding Window**, and select "Run". What is the mean TL for Lake Sturgeon in 2018? (149.5954 cm)

6. Following the above format, we can determine the range in weights
    a. range(Weight, na.rm = TRUE) (6.6kg to 67.0kg)
7. And the median FL for male AND female lake sturgeon
    a. median(FL, na.rm=TRUE) (137 cm)

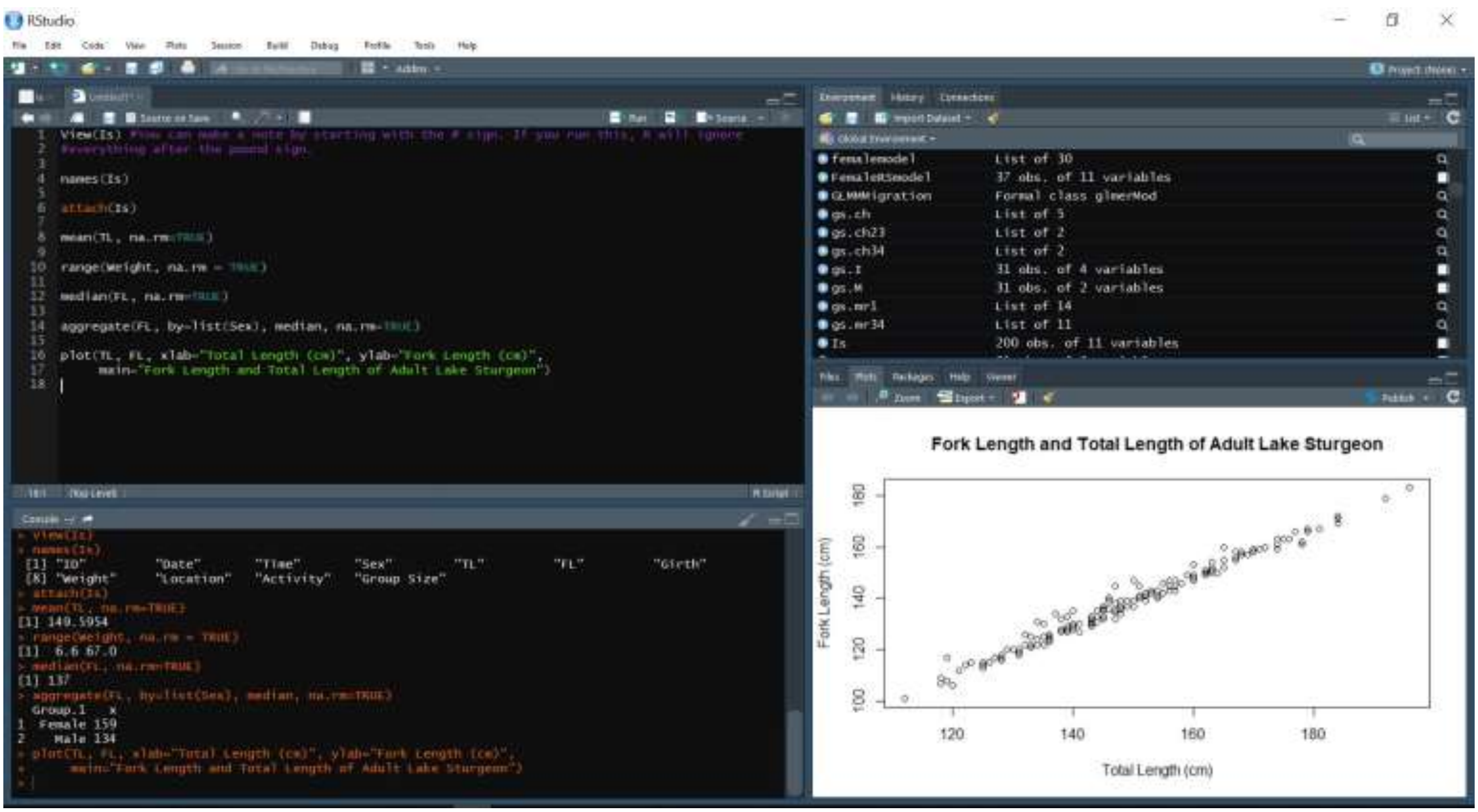
8. If we want to evaluate summary statistics by groups, in our case by considering males and females as their own group, we can do that using the aggregate() function. Here, we're still telling R that we want to get a summary statistic, but we want to list that summary statistic by a certain grouping. In step 7, we calculated the median fork length for all sturgeon. Here, let's calculate the median fork length for male and female lake sturgeon. We are going to use the aggregate function to group by sex (by=list(Sex)), then insert the function we want (median)
    a. Enter aggregate(FL, by=list(Sex), median, na.rm=TRUE) in the coding window, then **select "Run"**.
    b. What are the median fork lengths for male and female lake sturgeon?
        i. Females: 159 cm
        ii. Males 134 cm



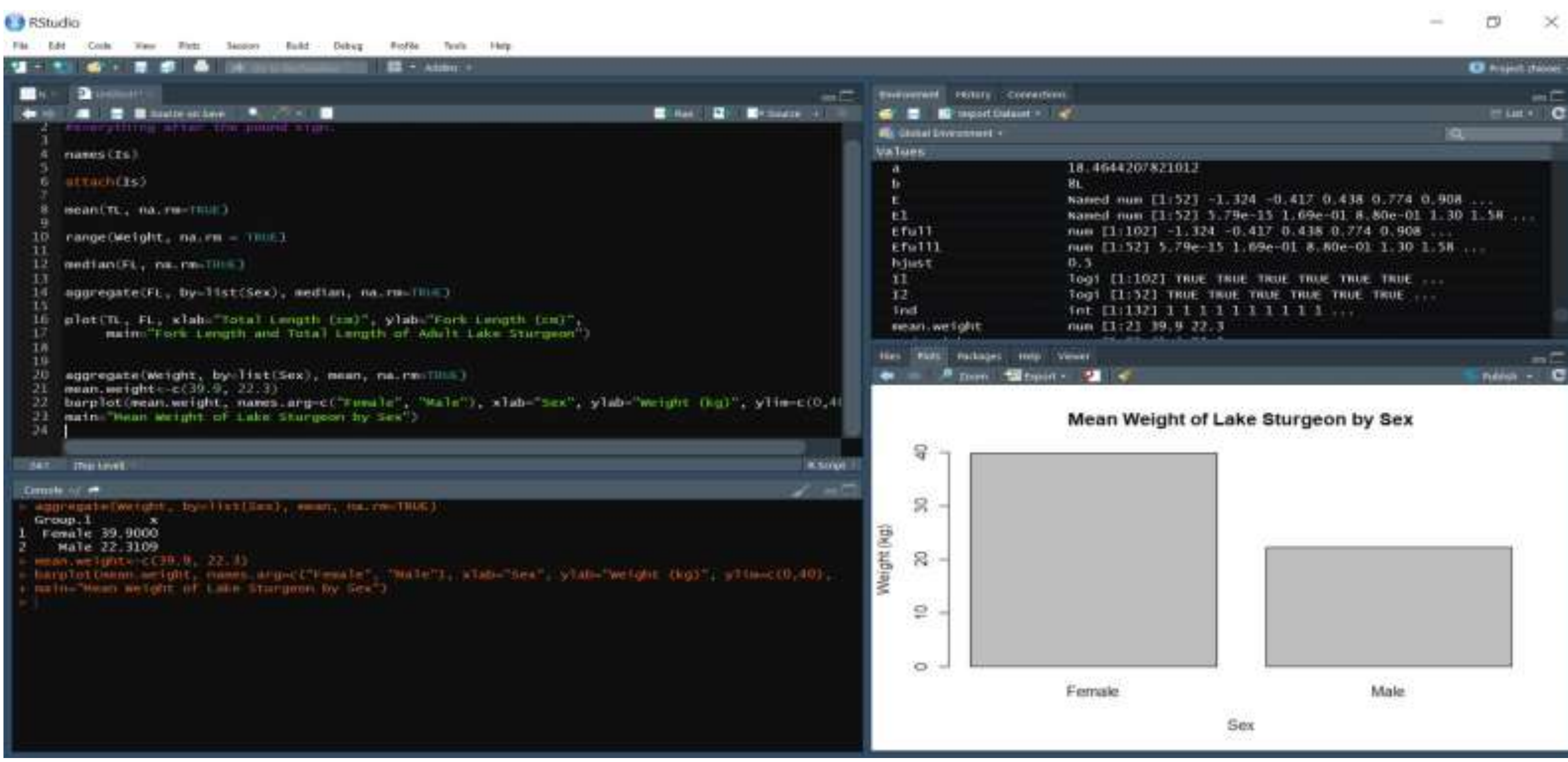
###Fast Visualizations

One of the most useful part of RStudio is the ability to see your data on a graph. Here, we will create some basic graphs which will allow us to see our data in order to help us detect differences. First, we will create a **scatter plot.** A **scatter plot** is a two dimensional graph which allows you to see each "fish" as the intersection of an x-axis variable and a y-axis variable in the form of a data point.

1. Create a scatterplot that shows FL and TL for each individual
    a. We will use the plot() function to plot the Total Length (x-axis) of adult Lake Sturgeon against the Fork Length (y-axis). The plot function is written as follows for a **scatter plot:**
    b. plot(x-axis variable name, y-axis variable name, "x-axis label", "y-axis label", "main graph title")
    c. Since we are graphic Total Length vs Fork Length, our code should read:
        i. plot(TL, FL, xlab="Total Length (cm)", ylab="Fork Length (cm)", main="Fork Length and Total Length of Adult Lake Sturgeon") **select "Run".**
            1. Where xlab= is the command which tells R the label of the x-axis;
            2. ylab= is the command which tells R the label of the y-axis;
            3. and main= is the commain which tells R the main chart label.
    d. When you select "Run," you'll notice that a **scatter plot** appears in the bottom right box in Rstudio. You can download this graph, or simply view it in the window.



2. Another useful graph which allows you to compare two related groups of data to one another is a **bar graph or bar plot**. A **bar plot** allows you to compare a median or mean value of one group (ie: median FL of female sturgeon to male sturgeon) to another. Below, we are going to make a **bar plot** which does just that.
    a. Make a **bar plot** of mean weight by Lake Sturgeon gender.
    b. first we need to make an object that has the mean weight by each sex. We are going to use aggregate again
        i. aggregate(Weight, by=list(Sex), mean, na.rm=TRUE) **select "Run"**
        ii. females have a mean weight of 39.9 kg while males have a median weight of 22.3 kg.
        iii. we're going to save these two values in an object called "mean.weight" using the following command:
            1. mean.weight<-c(39.9, 22.3) **select "Run"**
    c. Now, we are going to use the barplot() function to create the graph. We are graphing the mean.weight object we just made.
        i. barplot(mean.weight, names.arg=c("Female", "Male"), xlab="Sex", ylab="Weight (kg)", ylim=c(0,40), main="Mean Weight of Lake Sturgeon by Sex") **select "Run"**
    d. Where:
        i. Names.arg is the labels that go under the column values;
        ii. xlab= and ylab= are the x and y axes lables;
        iii. ylim= sets the scale of the y-axis from 0 to 40; and
        iv. main= is the title of the graph
    e. Again, the **bar plot** should appear in the lower right box. It appears that there's a pretty noticeable difference between male and female median weight.



###Correlation

Graphs are very useful in detecting difference in data, but how do we know that those relationships can be quantified? In the first example, we looked at the visual relationship between fork length and total length. They look like their related, in when total length increases on the x-axis, fork length increases on the y-axis. But how do we know if that's ACTUALLY what's going on?

We can use a simple **correlation analysis** to determine if two variables are related. A **correlation** analysis simply tells us whether two variables track one another. We have to be careful with correlation. If we consider the variables with which we work, it's great. But we should remember that correlation does NOT equal causation. In other words, we can have a scatter plot which tells us that two variables appear to be related, when in fact, they are obviously not (Figure 4). As long as you are careful when considering the relationship two variables have, correlation can be a powerful investigative tool. A **correlation analysis** outputs a single value between 0 and 1 called the **correlation coefficient**. The closer to 1 a **correlation coefficient** falls; the more correlated two variables are. Conversely, as a correlation coefficient approaches 0, the more poorly correlated variables are to one another.

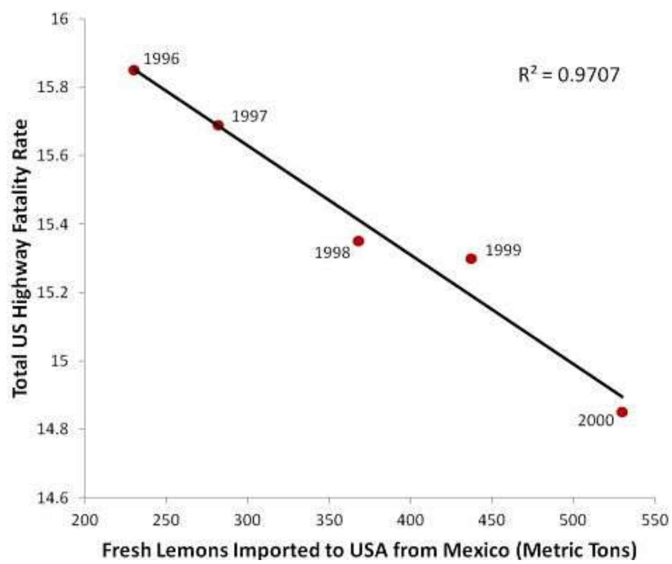


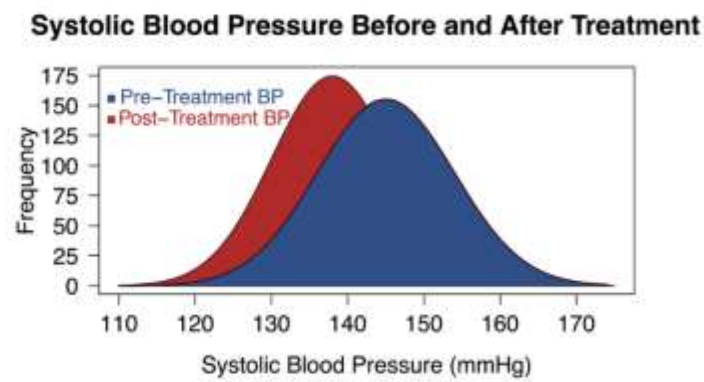
Figure 4. Correlation ≠ causation. Is there a reasonable relationship between the number of lemons imported from Mexico and the number of US Highway Fatalities….?

Here, we will consider how variables from the **scatter plot** example are actually related.

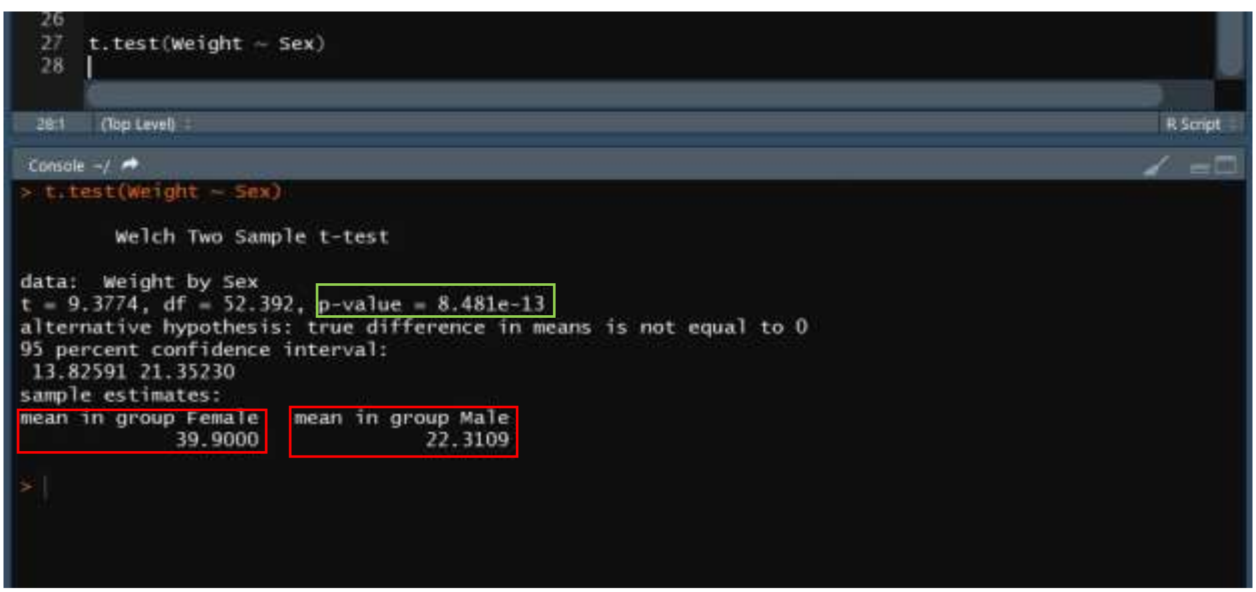
1. Find the correlation factor between TL and FL.
    a. Use the cor() function for correlation. We are correlating FL and TL. In the case where we may be missing data from either total length, or fork length we can use the command use="complete.obs" to tell R to ignore incomplete data.
    b. In the **Coding** Window, enter: cor(FL, TL, use="complete.obs") **select "Run"**
    c. The **correlation coefficient** is 0.9894 which is very close to 1, so these two variables are highly correlated. This makes sense as FL is a very similar measurement to TL.

###T-Test

In the second example, we considered the average weight of male and female Lake Sturgeon. It appeared that females weighed more than males. However, how do we know that? What if the majority of your female population wasn't all that different from the male population, but you happened to catch three female Lake Sturgeon that were **outliers**? An **outlier** is a data point which is distant or well removed from other observations. We need a way to determine if the data we consider are **statistically different.** In other words, can we use basic statistics to determine if two groups are quantifiably different, or just visually different. In the case of our weight example, we can use a **T-Test.** A **T-Test** compares the mean values of a variables between two groups. It does that by calculating a t-statistic and a p-value. Now, rather than explain the math, let's instead consider a visual explanation. Figure 5 shows distributions of blood pressure measurements prior to and after a treatment. We do that by interpreting a **p-value**. For the sake of this lesson, a **p-value** less than 0.05 (P < 0.05) would tell us that our considered groups are statistically different.



1. Use a t-test to statistically compare the mean weights of male and female lake sturgeon caught in 2018.
2. Use the t.test() function to see if Weight differs by (~) sex
3. Enter: t.test(Weight ~ Sex) **select "Run"**
4. We see a lot of information in the output from the t-test code (see below). The important values are in boxes. As above, we can see that the average weight of female Lake Sturgeon is 39.9 kg, which the average weight of males is 22.3 kg. Additionally, we can see in the green box that the calculated p-value in 8.48 x 10^{-13}. This value is far less than the threshold of 0.05 mentioned above. As such, we can interpret that the mean weight of female Lake Sturgeon is different than the mean weight of male Lake Sturgeon. Our graph shows us that female Lake Sturgeon weight more than male Lake Sturgeon.



###Assignment

1. What is the minimum and maxium (range) total length (cm) of male and female lake sturgeon captured in 2018?

Teacher Answer:

Code (Both combined): range(TL, na.rm = TRUE)

Answer: min – 112cm; max 196 cm.

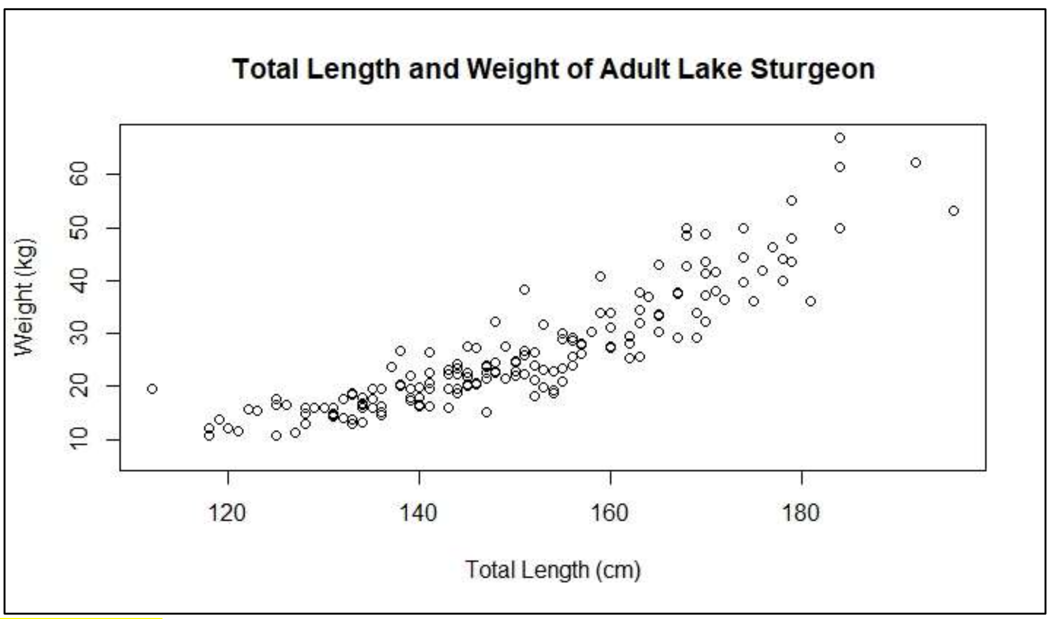
Code (Seperated by group): aggregate(TL, by=list(Sex), range, na.rm=TRUE)

Answer: Females: 138cm – 196cm; Males: 112cm – 178 cm

2. Create a scatter plot and correlation coefficient for Total Length (x-axis) and Weight (y-axis) of adult lake sturgeon captured in 2018. Are Total Length and Weight correlated?

Graph Code: plot(TL, Weight, xlab="Total Length (cm)", ylab="Weight (kg)", main="Total Length and Weight of Adult Lake Sturgeon")

Plot:

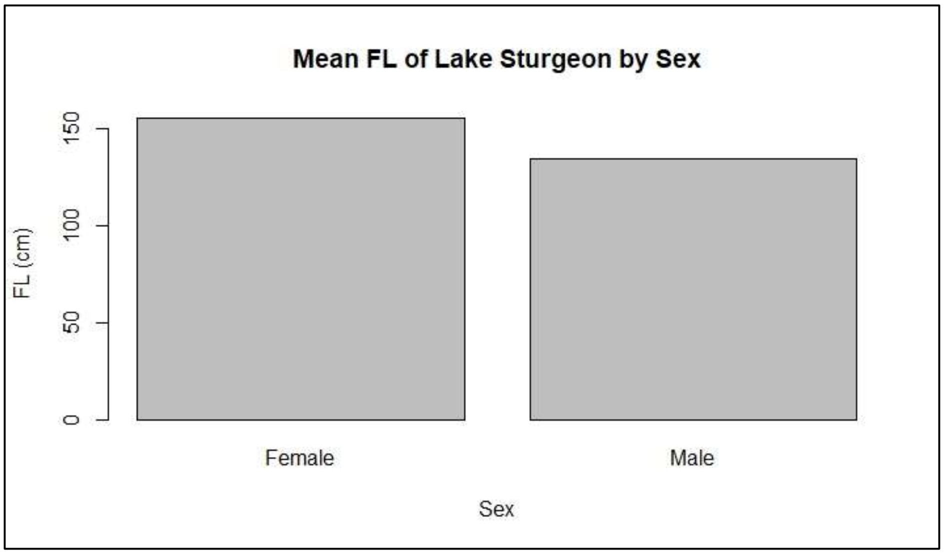


Correlation Code: cor(Weight, TL, use="complete.obs")

Teacher Answer: Correlation coefficient: 0.8947 – yes they are correlated.

3. Calculate the mean fork length for male and female Lake Sturgeon. Create a barplot showing the means of the two groups. Use a T-Test to determine if fork length differs between male and female Lake Sturgeon.

Mean Code:
aggregate(FL, by=list(Sex), mean, na.rm=TRUE)
mean.FL<-c(155.64, 134.23)
Graph Code: barplot(mean.FL, names.arg=c("Female", "Male"), xlab="Sex", ylab="FL (cm)", ylim=c(0,160), main="Mean FL of Lake Sturgeon by Sex")



T-Test Code: t.test(FL ~ Sex)

mean in group Female   mean in group Male

155.6389   134.2263

Teacher Answer: p-value = 4.10 x 10^{-11}; because $p<0.05$, we can determine that Fork Length differs between male and female Lake Sturgeon

4. Come up with a simple question you could answer with the 2018 adult capture data. This question must be different than the analyses performed above. Answer your question using at least 2 of the following: descriptive statistics, visualizations, statistical analyses.

Teacher Answer: Answers will vary. Could consider numerous relationships via correlation/scatter plot or barplot/T-test. As long as the code is similar to above, should be fine.