Introduction to R

Allison White

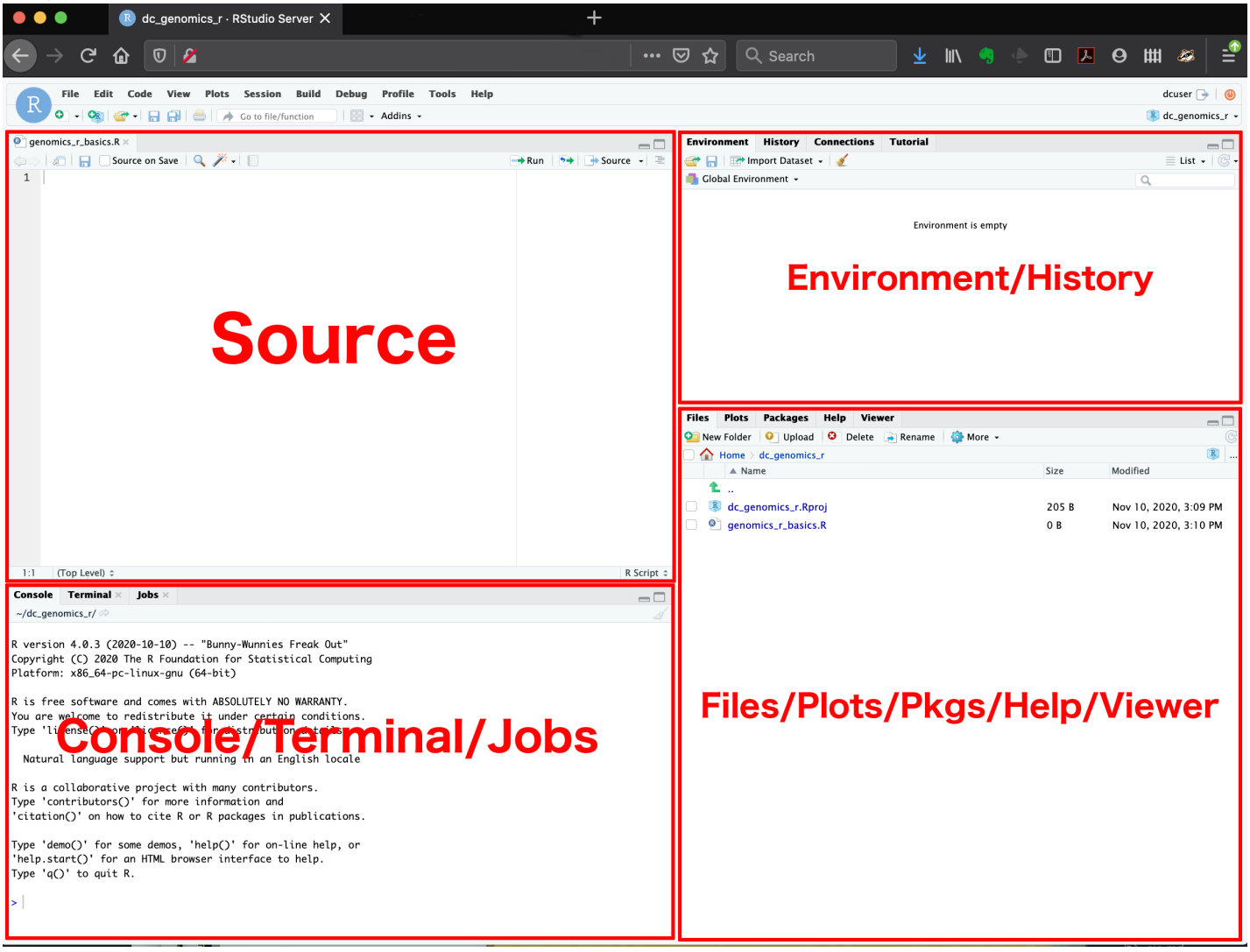
2021-08-13

Welcome to R!

R is an extremely powerful and robust statistical software that is free and regularly updated. RStudio is a more user friendly interface for the R software.

If you have opened this script, then there should be 4 windows in RStudio:

image:



1. The Script or Source Window (upper left) is where you can create, open, and save scripts of code which you can save for later use. To open a new or existing script, use the icons in the upper left of your RStudio directly above where the source window is shown in the image above.
2. The Console Window (lower left) is where you will run the code.
3. The Environment Window (upper right) will show data and objects that you create in RStudio.
4. The bottom right window includes three important tabs:

- the Help Tab which can be used to search for help pages for any R functions
- the Packages Tab can be used to install new packages into R, and
- the Plots tab will display all plots which are run in the Console Window.

R uses a programming language written mostly in C, Fortran, and R. Programming languages are formal languages comprised of a set of strings that produce machine code outputs. This means that they are very sensitive to typos! Misspellings, capitalization, punctuation marks, and spaces all can cause the software to misinterpret a command. Be mindful of this as you begin to write your own code, as the most common source of error in R are typos.

R language is “object” oriented: users define objects of a specific class which are applicable to different “functions” or applications.

3 common object classes:

- “numeric” = a list of numerical values;
- “factor” = a list of categorical values;
- “matrix” or “dataframe” = storage of multiple lists (either numeric or factor) of the same length (rows) as columns

If you are interested in learning to write your own code in R, Google is your best friend! Many people have likely already attempted what you are trying to do, and there are many tutorials and forums available online for help in R coding. Visit stackoverflow to see a commonly used forum for R codes: https://stackoverflow.com/ (https://stackoverflow.com/)

To get started writing your own code, check out the **swirl package**. The swirl package offers step by step demonstrations of how to do many simple commands in R. You can install packages using the Package Tab in the bottom right window of RStudio or by running the following command:

- install.packages(‘swirl’)

```
library(swirl) #opens the package 'swirl' in your R workspace
```

```
## Warning: package 'swirl' was built under R version 4.2.3
```

```
##
## | Hi! Type swirl() when you are ready to begin.
```

As you read through scripts and begin to write your own, keep in mind that **any text that comes after # is not run**. This is a great way to add comments about what your code is performing.

Run code from the script into the Console by placing your cursor somewhere in the line of the script you wish to run (do not highlight!) and hit “Run” in the top right of the Script Window or Ctrl + Enter

R is an object-based language: command outputs are stored under user-specified names and referenced in **functions** which store specific sets of commands. Groups of functions with similar purposes are stored in **packages** which users can download into their RStudio. Many commonly used packages are downloaded into RStudio by default. For example, you can us the Packages Tab in the lower right window to view all functions in the “base” package, which includes many simple commands. To find information on how to run a specific function, you can click on the link for the function or type it into your console after “?”

```
?plot
```

```
## starting httpd help server ... done
```

Example: Do Sharks Hate Bubbles?

To show how everything above is applied in R, the following example analyzes the data in “sharks.csv” using a t-test, ANOVA, and linear model. The sharks data represents an imaginary experiment where SCUBA divers recorded shark presence at five different dive sites. The divers visited each dive site twice: once using an open-circuit (OC) diving apparatus in which their exhaled breaths are released into the water as bubbles and once using a closed-circuit rebreather (CCR) which re-routes their exhaled breaths through a loop and does not produce any bubbles.

Open and Explore Data

First, you’ll need to download the file sharks.csv. To import or export data into R, it’s helpful to set a working directory where all data you want to import is stored and where your exports will be saved to.

```
#Define your working directory
getwd() #displays the current working directory
```

```
## [1] "C:/Users/allwhite/OneDrive - Florida International University (1)/Documents - MEAL/Code/
Intro to R"
```

```
setwd('C:/Users/allwhite/OneDrive - Florida International University (1)/Documents/R') #sets the
working directory to specified folder. Be sure to update the file path to a folder on your machi
ne.

#Import data into R
data<-read.csv('sharks.csv') #open a .csv file with your data. This must be in the folder that y
ou set as your working directory
```

A .csv file is a commonly used file type similar to an Excel spreadsheet. In the command above which imports data into R, the read.csv() function calls to the argument “sharks.csv” and is stored as an object called “data” of class “dataframe”. This object can now be referenced as an argument in other functions.

```
head(data) #prints the first six rows of your data
```

```
##   dive site system sharks
## 1    1    1     OC      0
## 2    2    2     OC      2
## 3    3    3     OC      1
## 4    4    4     OC      5
## 5    5    5     OC      3
## 6    6    1    CCR      4
```

```
attach(data) #identifies each column heading as a variable object in R
```

For small datasets or simple lists, you can also manually enter data.

```
dive<-c(1:10)
site<-c(1:5,1:5)
sharks<-c(0,2,1,5,3,4,7,10,8,5) #Creates list of values named 'sharks'
system<-c('OC','OC','OC','OC','OC','CCR','CCR','CCR','CCR','CCR') #Creates list of values named
'CCR'
data<-data.frame(sharks,system,dive,site) #Creates a table with columns 'sharks' and 'system'
head(data) #prints the first six rows of your data
```

```
##   sharks system dive site
## 1      0     OC    1    1
## 2      2     OC    2    2
## 3      1     OC    3    3
## 4      5     OC    4    4
## 5      3     OC    5    5
## 6      4    CCR    6    1
```

As seen in both methods above, the sharks dataset has four variables: dive, site, sharks, and system. The order of each dive is recorded as “dive”. The location of the dive is recorded as “site”. “Sharks” indicates the number of sharks counted by the divers and “system” indicates which breathing apparatus (open-circuit or closed-circuit rebreather) the diver was using at the time.

It’s often useful to summarize new datasets to better understand the information stored in them.

```
summary(data) #prints minimum, maximum, and mean of each numeric variable in dataframe "data"
```

```
##      sharks         system               dive            site
##  Min.   : 0.00   Length:10          Min.   : 1.00   Min.   :1
##  1st Qu.: 2.25   Class :character   1st Qu.: 3.25   1st Qu.:2
##  Median : 4.50   Mode  :character   Median : 5.50   Median :3
##  Mean   : 4.50                      Mean   : 5.50   Mean   :3
##  3rd Qu.: 6.50                      3rd Qu.: 7.75   3rd Qu.:4
##  Max.   :10.00                      Max.   :10.00   Max.   :5
```

```
mean(data$sharks) #calculates mean number of sharks observed in "data"
```

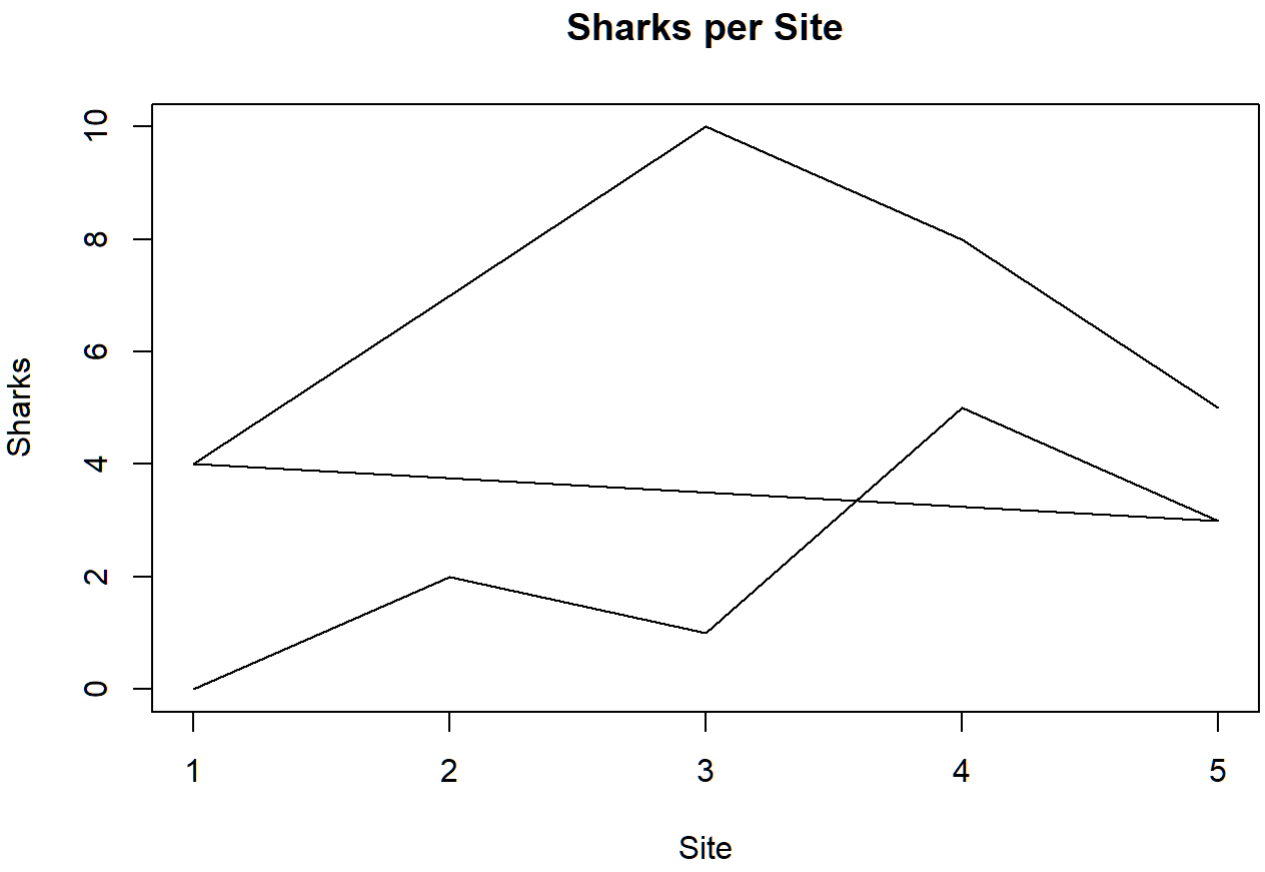
```
## [1] 4.5
```

```
sd(data$sharks) #calculates standard deviation of sharks observed in "data"
```

```
## [1] 3.17105
```

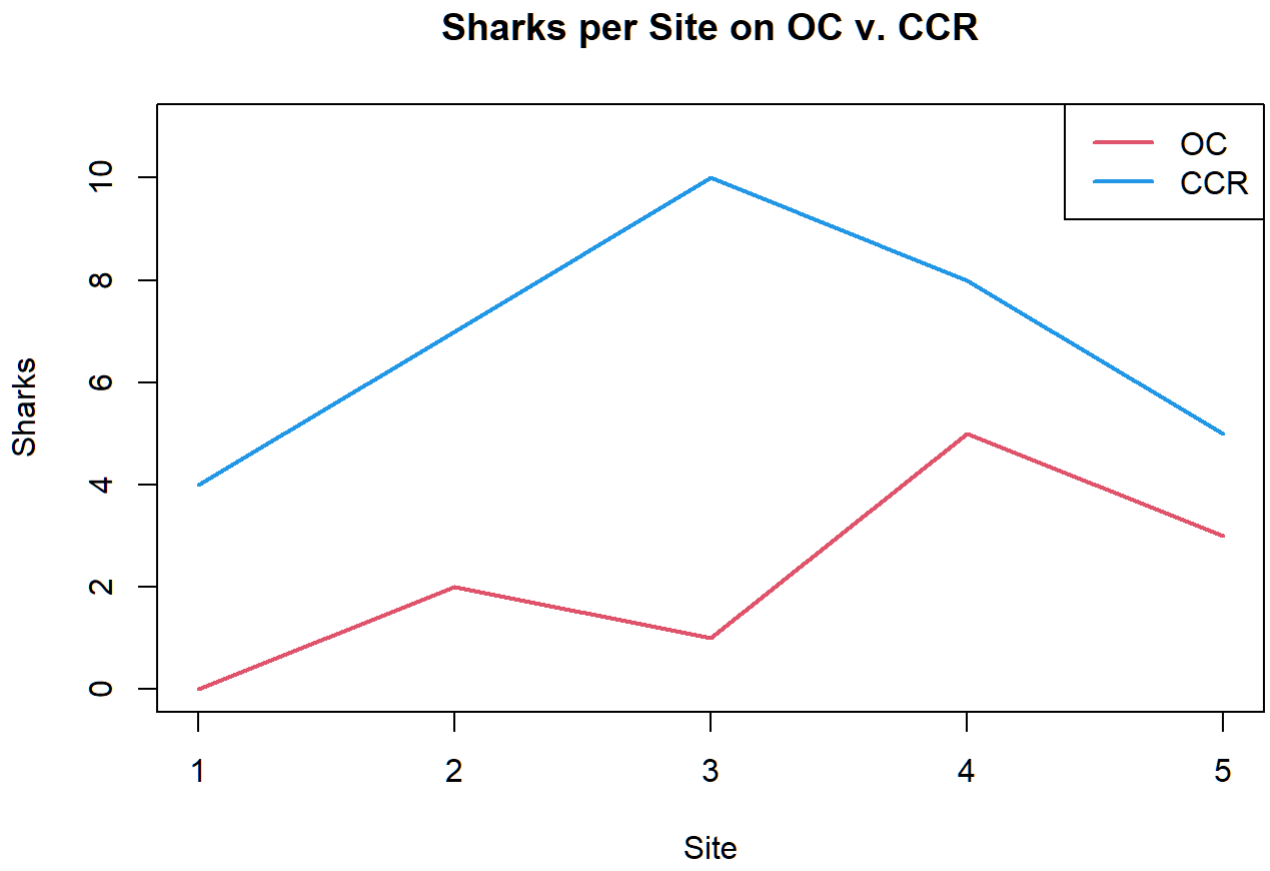
You can also explore datasets by plotting variables.

```
plot(data$site, data$sharks,main="Sharks per Site",xlab='Site',ylab='Sharks',type='l',col=1) #a simple scatterplot of the number of sharks seen at each site with title and axis labels
```



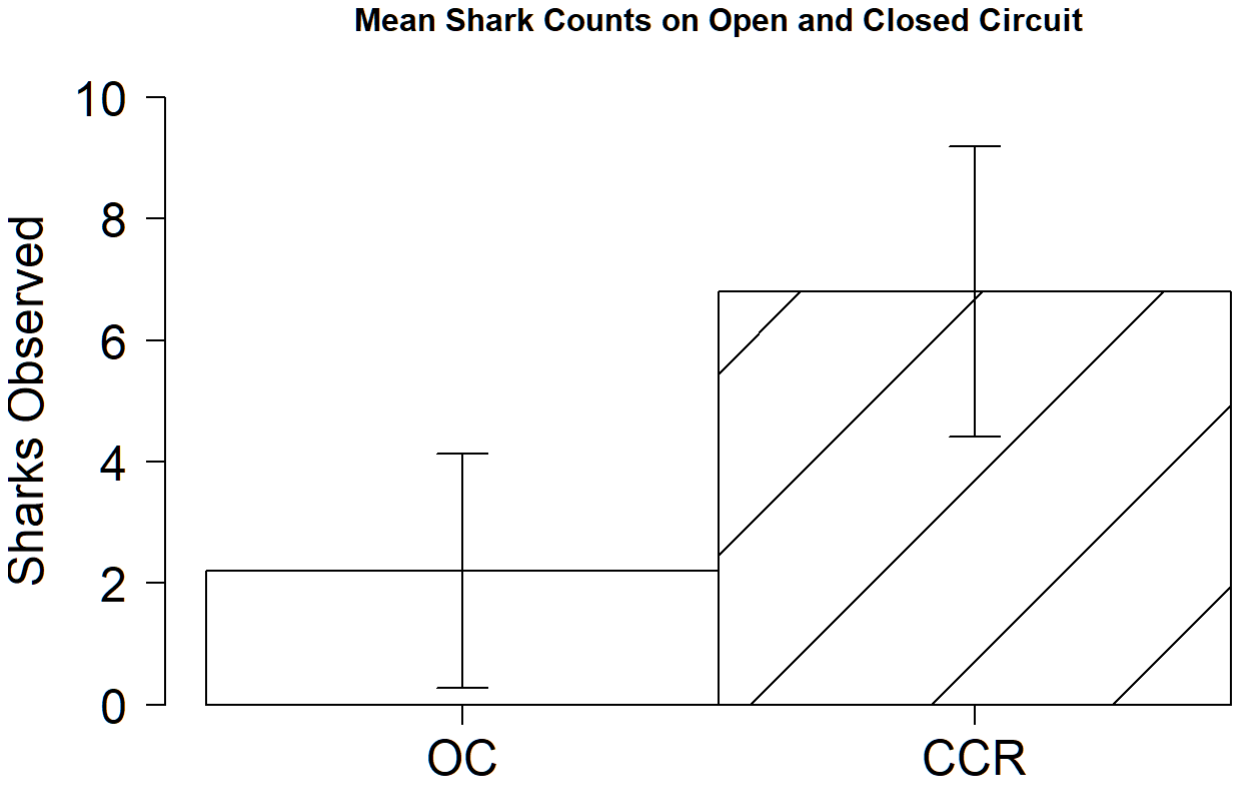
The above plot is messy, we want a separate line for OC v. CCR. So we should subset the data first, then plot it.

```
#subset data
OC<-subset(data,data$system=='OC') #groups all rows where the value in column headed "system" is 'OC'
CCR<-subset(data,data$system=='CCR') #groups all rows where the value in column headed "system" is 'CCR'


#plot OC and CCR shark sightings as separate lines
plot(OC$site,OC$sharks,type='l',col=2,ylim=c(0,11),xlab='Site',ylab='Sharks',main='Sharks per Site on OC v. CCR',lwd=2,xlim=c(1,5)) #Creates a red line plot of sharks per site on OC and creates y space to add CCR
lines(CCR$site,CCR$sharks,type='l',col=4,lwd=2) #adds a blue line of sharks per site on CCR to the plot above
legend('topright',col=c(2,4),lty=c(1,1),lwd=c(2,2),c('OC','CCR')) #creates a legend for the graph in the top right corner
```



Besides scatterplots, barplots are useful for categorical data which is divided into clear groups.

```
#Make a barplot with error bars
barplot(height = c(mean(OC$sharks),mean(CCR$sharks)),cex.axis=1.5,cex.lab=1.5,cex.main=1,ylab='S
harks Observed',main='Mean Shark Counts on Open and Closed Circuit',space=0,ylim = c(0,10),angle
=c(0,45),density=c(0,1.5),col=1,xaxt='n',las=2) #makes a bar plot with custom axis text size and
fill
axis(side=1,at=c(0.50,1.50),labels=c("OC","CCR"),cex.axis=1.5) #creates a custom x-axis
segments(x0=0.50,y0=mean(OC$sharks),x1=0.50,y1=(mean(OC$sharks)+sd(OC$sharks))) #draws an error
bar from the mean to the positive standard deviation
segments(x0=1.50,y0=mean(CCR$sharks),x1=1.50,y1=(mean(CCR$sharks)+sd(CCR$sharks)))
segments(x0=0.50,y0=mean(OC$sharks),x1=0.50,y1=(mean(OC$sharks)-sd(OC$sharks))) #draws an error
bar from the mean to the negative standard deviation
segments(x0=1.50,y0=mean(CCR$sharks),x1=1.50,y1=(mean(CCR$sharks)-sd(CCR$sharks)))
segments(x0=0.45,y0=(mean(OC$sharks)+sd(OC$sharks)),x1=0.55) #caps the error bar with a short ho
rizontal line
segments(x0=1.45,y0=(mean(CCR$sharks)+sd(CCR$sharks)),x1=1.55)
segments(x0=0.45,y0=(mean(OC$sharks)-sd(OC$sharks)),x1=0.55) #caps the error bar with a short ho
rizontal line
segments(x0=1.45,y0=(mean(CCR$sharks)-sd(CCR$sharks)),x1=1.55)
```



Analyze Data

There are several approaches we could use to analyze our dataset depending on the question we want to ask about this information. Below, three commonly used analyses are used to answer three possible questions that we could test using the sharks dataset.

Question 1: Is the number of sharks sighted affected by the system used by the diver?

In this case, we want to compare shark sightings between two categories: dives completed using open-circuit and dives completed using closed-circuit rebreathers. For comparing the means of two groups, we can utilize a two-sample t-test.

```
shark.t.test<-t.test(sharks~system,data=data) #performs a two-sample t-test to compare the mean number of sharks sighted when using each system in the dataset "data"
shark.t.test #prints the significant statistics of the t-test
```

```
##
##  Welch Two Sample t-test
##
## data:  sharks by system
## t = 3.3549, df = 7.6535, p-value = 0.01067
## alternative hypothesis: true difference in means between group CCR and group OC is not equal
to 0
## 95 percent confidence interval:
##  1.413078 7.786922
## sample estimates:
## mean in group CCR  mean in group OC
##               6.8               2.2
```

To test our null hypothesis that sharks are not affected by the system used by divers, we need to observe the p-value. If this value is less than 0.05, we reject the null hypothesis. If it's larger, we fail to reject and state that there was a significant difference in the number of sharks sighted using OC v. CCR.

Now lets say we went diving a third time at each site and added those observations using the code below.

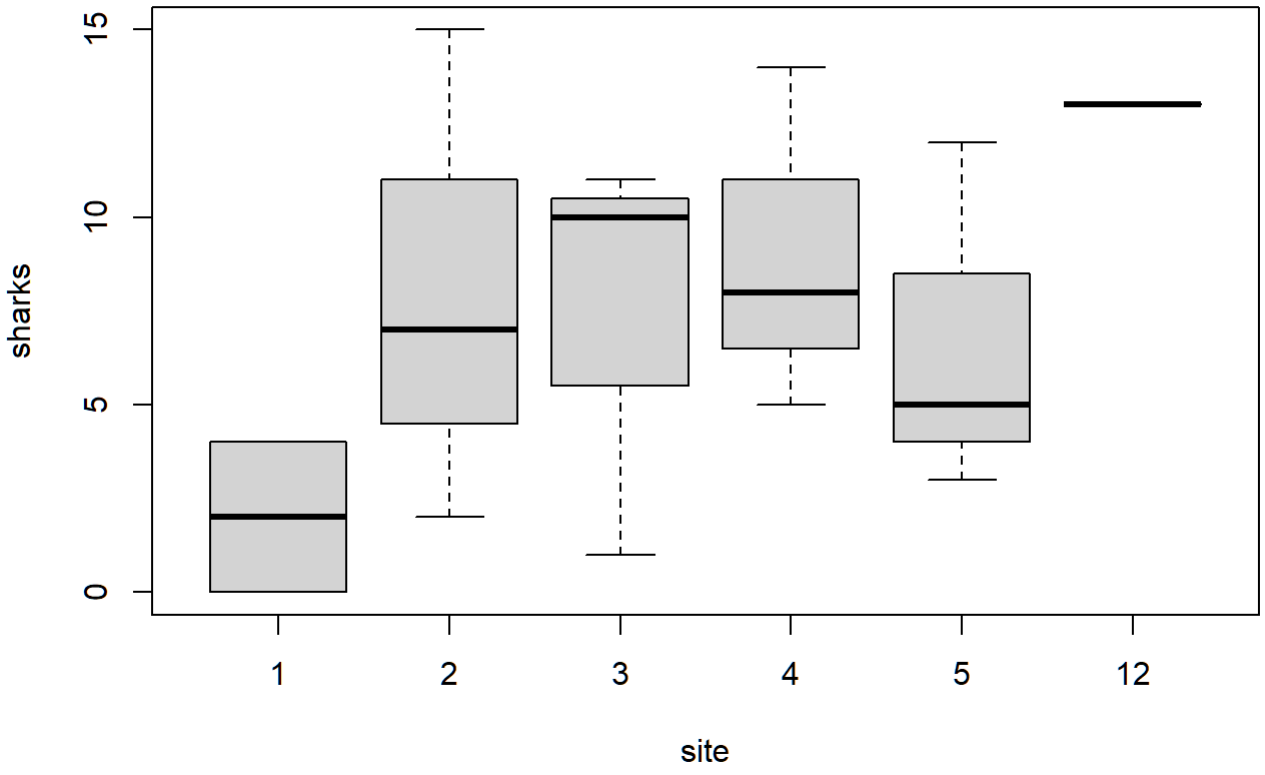
```
data[c(11:15),1]<-c(11:15)
data[c(11:15),2]<-c(1:5)
data[c(11:15),3]<-'OC'
data[c(11:15),4]<-c(3,5,12,4,2)

data
```

```
##    sharks system dive site
## 1       0     OC    1    1
## 2       2     OC    2    2
## 3       1     OC    3    3
## 4       5     OC    4    4
## 5       3     OC    5    5
## 6       4    CCR    6    1
## 7       7    CCR    7    2
## 8      10    CCR    8    3
## 9       8    CCR    9    4
## 10      5    CCR   10    5
## 11     11      1   OC    3
## 12     12      2   OC    5
## 13     13      3   OC   12
## 14     14      4   OC    4
## 15     15      5   OC    2
```

Question 2: Is the number of sharks sighted different among the sites surveyed by the divers?

Now, we want to compare shark sightings between more than two categories. For comparing the means of three or more groups, we can use an analysis of variance (ANOVA).

```
#First, plot the mean, min, max, 1st and 3rd quartiles of number of sharks seen at each site
boxplot(sharks~site,data=data)
```




```
shark.anova<-aov(sharks~site,data=data) #performs an ANOVA of sharks by site in the dataset "dat
a"
summary(shark.anova) #prints the significant statistics of the ANOVA.
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## site         1  51.43   51.43   2.306  0.153
## Residuals   13 289.90   22.30
```

In summary.aov, Pr(>F) is our p-value, which we interpret the same as we would a t-test. Since this p-value is >0.05, we fail to reject our null hypothesis and state that there is no significant difference in the number of sharks sighted at each location.

Now we want to know if the presence of sharks was affected by differences in the length of each dive. We'll add the durations of each dive using the code below.

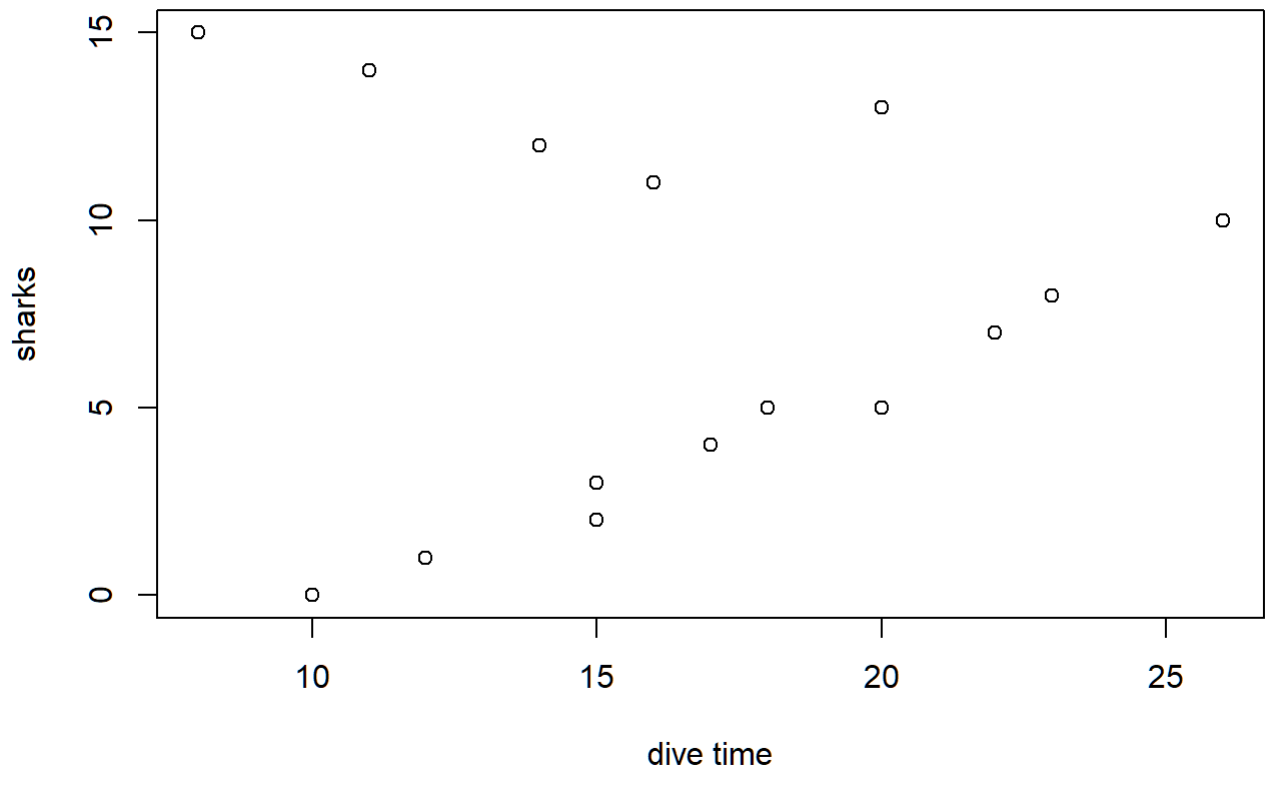
```
data$time<-c(10,15,12,18,15,17,22,26,23,20,16,14,20,11,8)
data
```

```
##    sharks system dive site time
## 1       0     OC    1    1   10
## 2       2     OC    2    2   15
## 3       1     OC    3    3   12
## 4       5     OC    4    4   18
## 5       3     OC    5    5   15
## 6       4    CCR    6    1   17
## 7       7    CCR    7    2   22
## 8      10    CCR    8    3   26
## 9       8    CCR    9    4   23
## 10      5    CCR   10    5   20
## 11     11      1   OC    3   16
## 12     12      2   OC    5   14
## 13     13      3   OC   12   20
## 14     14      4   OC    4   11
## 15     15      5   OC    2    8
```

Question 3: Is there a relationship between the number of sharks observed and the time spent on each dive?

This time, we want to observe the relationship between two continuous variables instead of categorical as we did in the t-test and ANOVA. We can compare the relationship between two continuous variables using a linear model.

```
#First, plot the dive time v. number of sharks seen
plot(sharks~time,data=data,xlab='dive time')
```

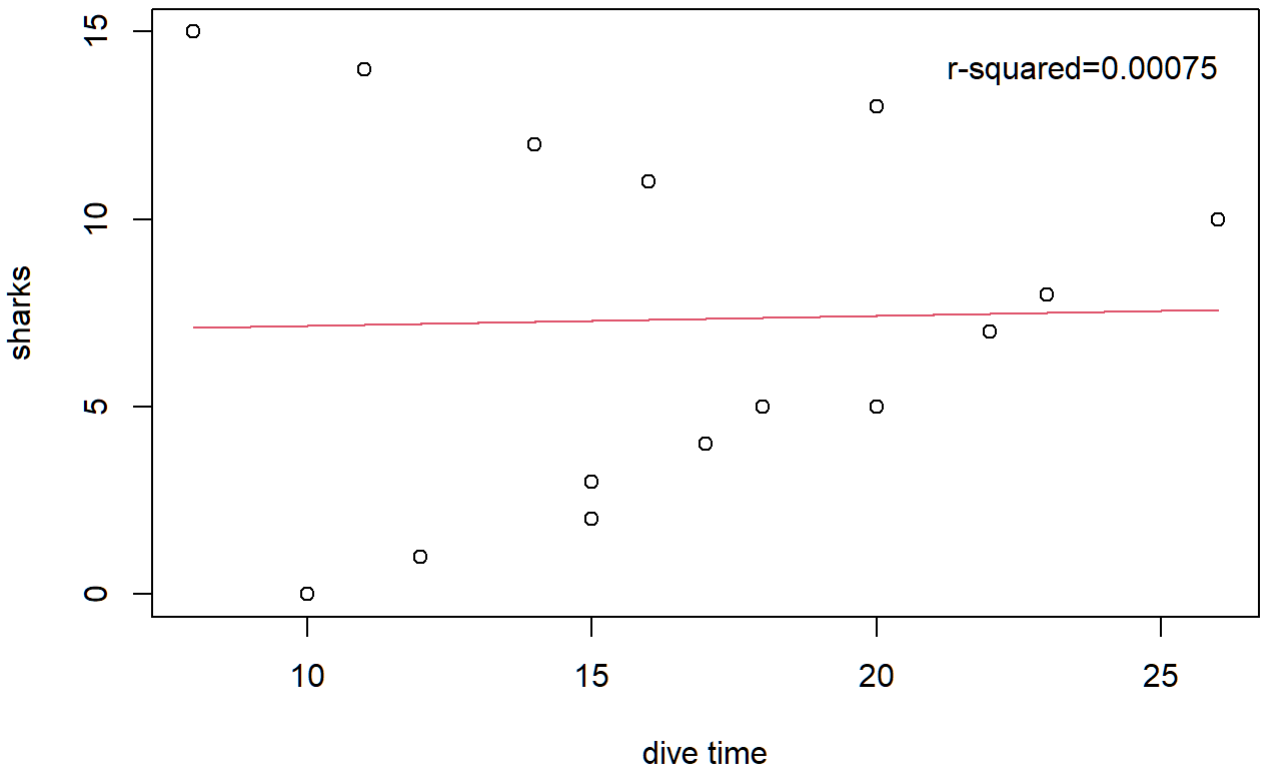


```
shark.lm<-lm(sharks~time,data=data) #fits a linear model to the relationship between sharks and dive time in the dataset "data"
summary(shark.lm) #pints the result of the linear model
```

```
##
## Call:
## lm(formula = sharks ~ time, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1624 -3.8210 -0.4796  4.2054  7.8904
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.89810    4.60443   1.498    0.158
## time         0.02643    0.26784   0.099    0.923
##
## Residual standard error: 5.122 on 13 degrees of freedom
## Multiple R-squared:  0.0007485,  Adjusted R-squared:  -0.07612
## F-statistic: 0.009738 on 1 and 13 DF,  p-value: 0.9229
```

Like the t-test and ANOVA, the p-value will tell us whether there is a significant relationship between dive time and number of sharks spotted. Since the p-value of this linear model was > 0.05, we fail to reject the null hypothesis. There is no significant linear relationship between dive duration and number of sharks seen. In linear models we also need to report r-squared. The adjusted r-squared value represents the fraction of variance in the data that's explained by the fitted linear model. An r-squared of 1 would mean that 100% of the variance is explained, whereas an r-squared of 0 would mean that none of the variance is explained.

```
shark.pred<-predict(shark.lm) #predict the trend between the number of sharks sighted and dive time
plot(sharks~time,data=data,xlab='dive time')
lines(data$time,shark.pred,col=2) #add a red line of the linear model prediction to your plot
text(26,14,paste0('r-squared=',round(summary(shark.lm)$r.squared,5)),adj=1) #add the r-squared value as text in the upper right location of the plot
```



In this case, almost none of the variance is explained by the linear model, which further supports our conclusion that there is no linear relationship between dive duration and number of shark sightings.