Exam Question 6

Nicole Baker

2023-03-05

# Question 4: ggplot2 Concepts

##this code is so that the figures display in this markdown file without species names overlapping on the x axis.  
library(knitr)  
knitr::opts\_chunk$set(fig.width=12, fig.height=9)

### layering

#### Layering is how you are able to add iterative detail and complexity to figures in ggplot. Each layer controls a specific aesthetic or data component of the figure, like color of plots, x and y axes titles, where data points sit relative to other chart elements, etc.

* First, I installed and loaded ggplot2.

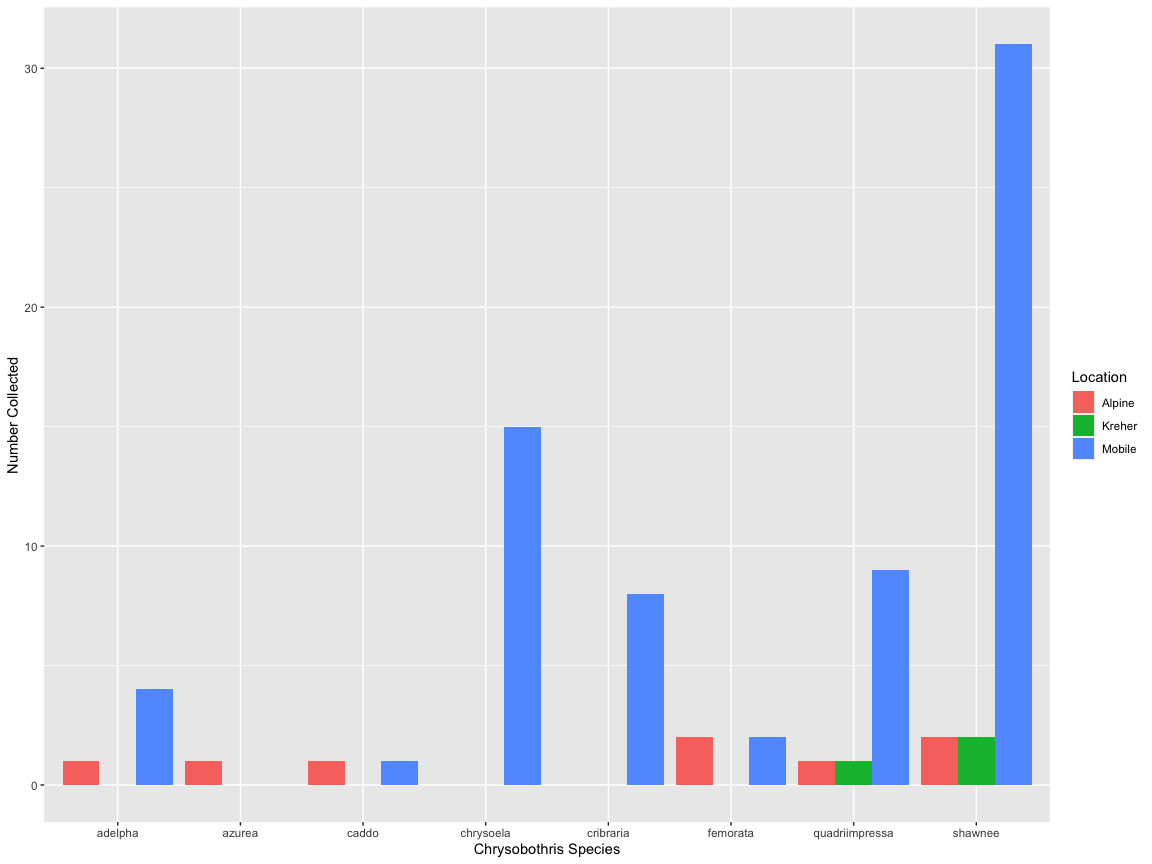
#install.packages("ggplot2")  
library(ggplot2)

* Next, I loaded in my data. For this question, I chose to use some of my trapping data from the Spring/Summer 2022 season. This data includes 8 wood boring beetle species in the genus *Chrysobothris* collected across 3 locations in Alabama.
* *in order for the data to be used by functions, I made my data an object called “TrapData”.*

TrapData<-read.csv("TrapData.csv")

* Now that I have my data, I can use the ggplot() function and a few layers to make a basic bar graph:

ggplot(TrapData, aes(x=Species,y=Collected,fill=Location))+  
 stat\_summary(fun=mean,geom="bar",position="dodge")+  
 ylab("Number Collected")+  
 xlab("Chrysobothris Species")



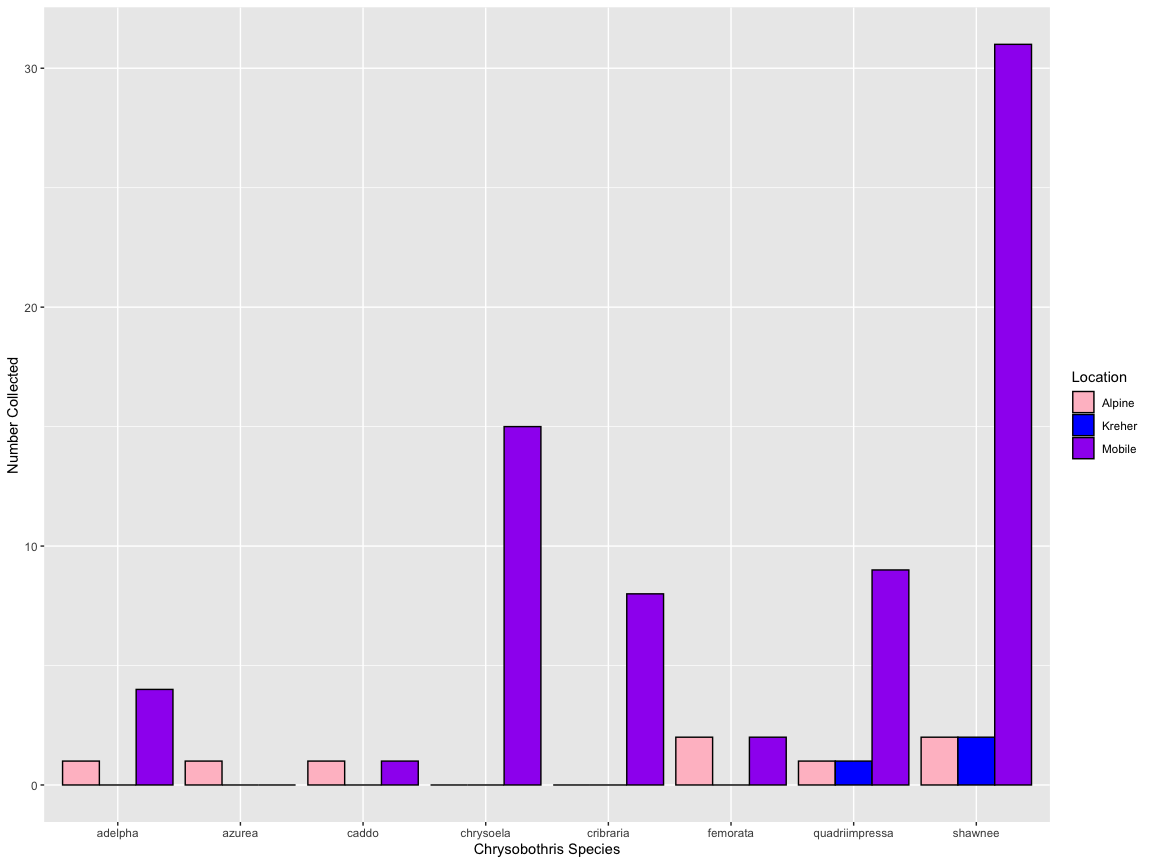
* The stat\_summary, ylab, and xlab lines of code are examples of layers, since they control the bar appearance and position, and the labels of the axes.

### Scales

#### Scales are ways to map data information as aesthetic information: aspects of figure elements such as color, size, and shape are all things that can be changed manually with scales.

* In order to use a scale function to manually change color, I will add it to my previous ggplot code from the layering example.
* Additionally, by adding color=“black” to the stat\_summary line of code, I am able to outline all bars with black.

ggplot(TrapData, aes(x=Species,y=Collected,fill=Location))+  
 stat\_summary(fun=mean,geom="bar",position="dodge",color="black")+  
 ylab("Number Collected")+  
 xlab("Chrysobothris Species")+  
 scale\_fill\_manual(values=c("pink","blue","purple"))



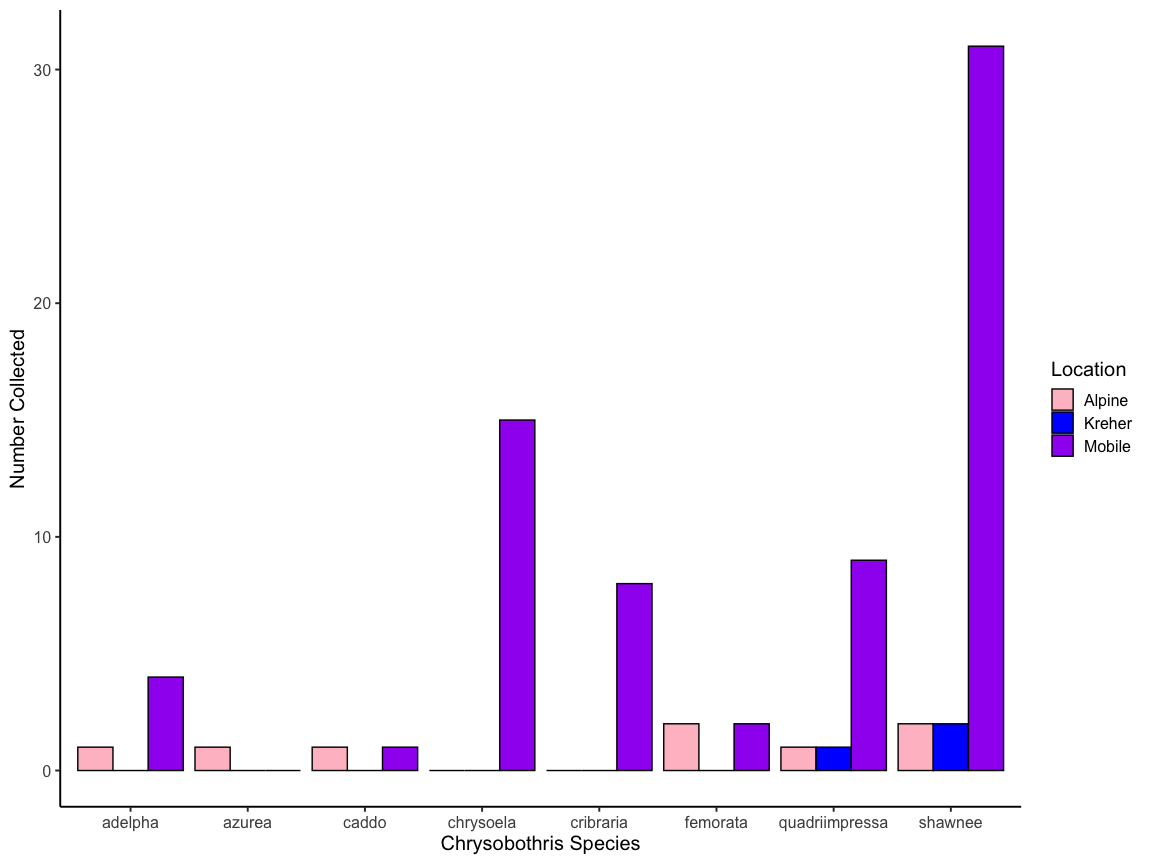
* The scale\_fill\_manual() function allowed me to change the colors of the bars to colors of my choice. Specific color numbers from color palettes can be used as well.

### Themes

#### Themes in ggplot can be used to change the basic layout, font, size of labels and legends, gridlines, and background aspects of a figure. Themes are an easy way to give a figure a consistent, neat appearance.

* To add a theme, I can include it into the code from the previous figures:

ggplot(TrapData, aes(x=Species,y=Collected,fill=Location))+  
 stat\_summary(fun=mean,geom="bar",position="dodge",color="black")+  
 ylab("Number Collected")+  
 xlab("Chrysobothris Species")+  
 scale\_fill\_manual(values=c("pink","blue","purple"))+  
 theme\_classic(base\_size = 15)



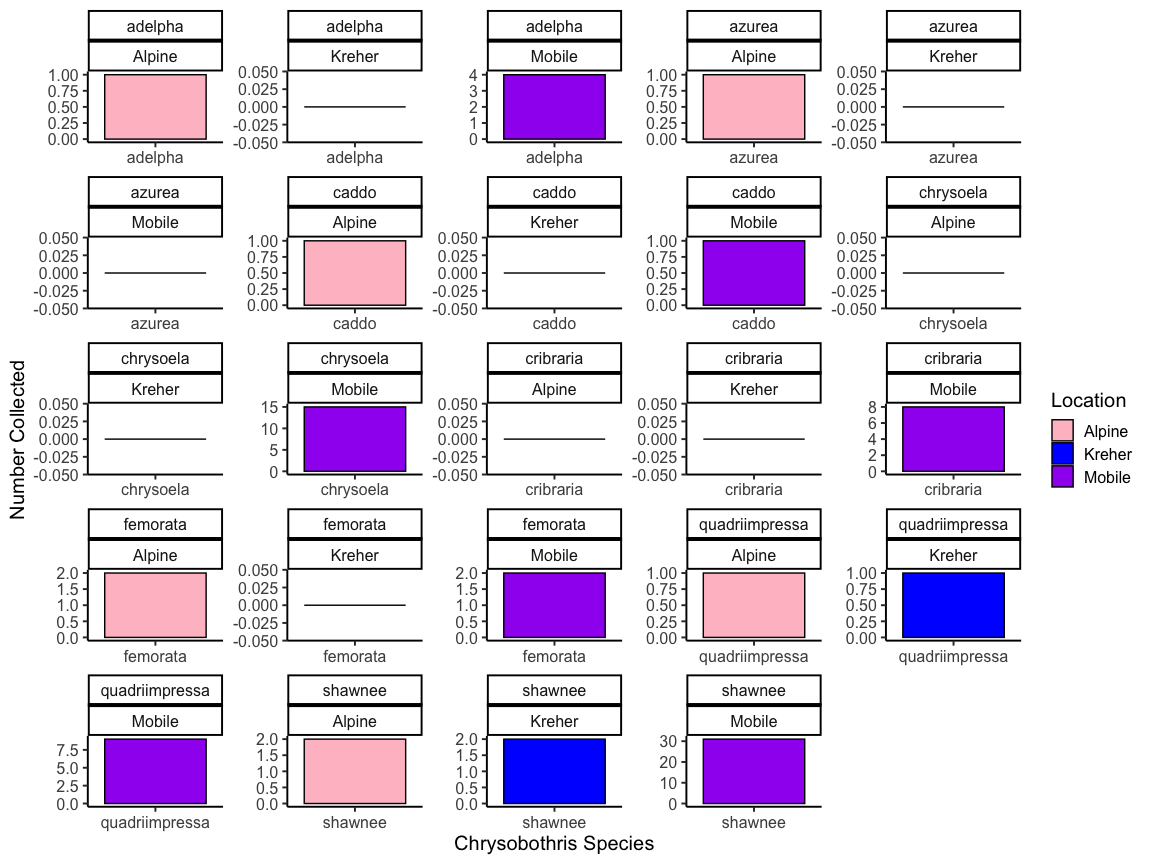
* By adding theme\_classic() to my code, the figure becomes a bit more streamlined, with no gridlines or grey shading behind the bars. Other themes include theme\_dark() or theme\_bw(). Since the theme controls the font of the figure, I set the base size to 15 so my figure is more readable.

### Facets

#### Facets split a plot of data into a set or block of figures, which can be compared or inspected individually. This is useful if you want to block by specific variables, or look at various subsets of data.

* To facet my data, I just need to add the facet\_wrap() function to the previously generated code:

ggplot(TrapData, aes(x=Species,y=Collected,fill=Location))+  
 stat\_summary(fun=mean,geom="bar",position="dodge",color="black")+  
 ylab("Number Collected")+  
 xlab("Chrysobothris Species")+  
 scale\_fill\_manual(values=c("pink","blue","purple"))+  
 theme\_classic(base\_size = 15)+  
 facet\_wrap(~Species\*Location,scales = "free")

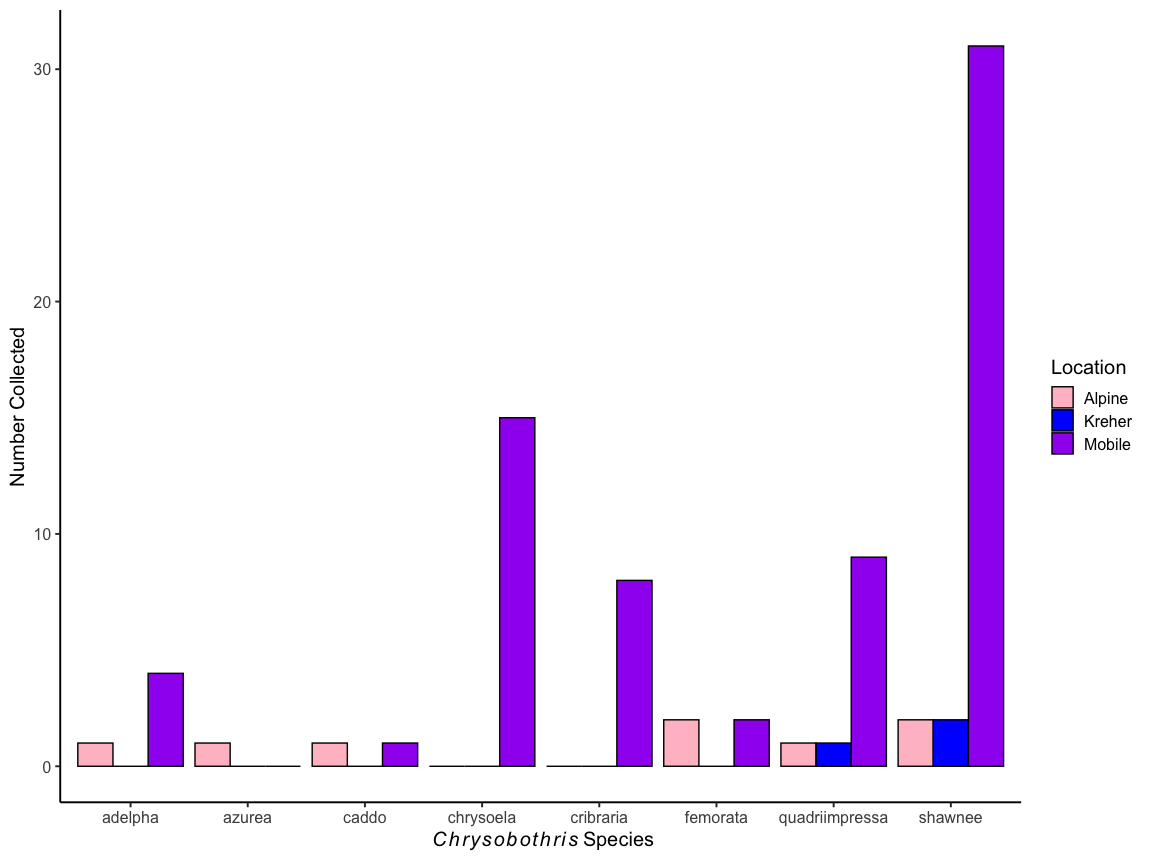


* By using the asterisk in between the variables I wanted to subset, I was able to block the *Chrysobothris* species by the 3 locations I collected from. The scales=“free” argument of the facet code allows for the x and y arguments to exist independently in each facet.

### Extra: Italicizing the x axis title

* I just wanted to see that I could do it:

ggplot(TrapData, aes(x=Species,y=Collected,fill=Location))+  
 stat\_summary(fun=mean,geom="bar",position="dodge",color="black")+  
 labs(y="Number Collected",  
 x=expression(~italic(Chrysobothris)~ 'Species'))+  
 scale\_fill\_manual(values=c("pink","blue","purple"))+  
 theme\_classic(base\_size = 15)

 - Now *Chrysobothris* is italicized on the figure!

# Question 5: Vectors, Matrices, and Dataframes: Subsetting

### Vectors

* A vector is a set of scalar elements, like a list of numbers, that are grouped together in a single object.
  + Example:

vec<-c(1:100)  
vec

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36  
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54  
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72  
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
## [91] 91 92 93 94 95 96 97 98 99 100

### Matrices

* A matrix is an object that is made up of only one type of data: only numbers, or only character values, but not both.
  + Example:

Matrix<-matrix(data=c("Nicole","Jordan","Kayla"),nrow=3,ncol=4)  
Matrix

## [,1] [,2] [,3] [,4]   
## [1,] "Nicole" "Nicole" "Nicole" "Nicole"  
## [2,] "Jordan" "Jordan" "Jordan" "Jordan"  
## [3,] "Kayla" "Kayla" "Kayla" "Kayla"

### Dataframes

* A dataframe is an object that can contain multiple different kinds of data, making them more useful.
  + Example:

df<-data.frame(vec[1:3],Matrix[,2])  
df

## vec.1.3. Matrix...2.  
## 1 1 Nicole  
## 2 2 Jordan  
## 3 3 Kayla

#### While all three of these are objects, vectors are the simplest. Both matrices and dataframes can have columns and rows, but dataframes can have more than one kind of data, whereas matrices can not. All of these objects can be subsetted and indexed in similar ways.

### Subsetting A Dataframe

* First, import the tooth growth data:

ToothGrowth

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5  
## 11 16.5 VC 1.0  
## 12 16.5 VC 1.0  
## 13 15.2 VC 1.0  
## 14 17.3 VC 1.0  
## 15 22.5 VC 1.0  
## 16 17.3 VC 1.0  
## 17 13.6 VC 1.0  
## 18 14.5 VC 1.0  
## 19 18.8 VC 1.0  
## 20 15.5 VC 1.0  
## 21 23.6 VC 2.0  
## 22 18.5 VC 2.0  
## 23 33.9 VC 2.0  
## 24 25.5 VC 2.0  
## 25 26.4 VC 2.0  
## 26 32.5 VC 2.0  
## 27 26.7 VC 2.0  
## 28 21.5 VC 2.0  
## 29 23.3 VC 2.0  
## 30 29.5 VC 2.0  
## 31 15.2 OJ 0.5  
## 32 21.5 OJ 0.5  
## 33 17.6 OJ 0.5  
## 34 9.7 OJ 0.5  
## 35 14.5 OJ 0.5  
## 36 10.0 OJ 0.5  
## 37 8.2 OJ 0.5  
## 38 9.4 OJ 0.5  
## 39 16.5 OJ 0.5  
## 40 9.7 OJ 0.5  
## 41 19.7 OJ 1.0  
## 42 23.3 OJ 1.0  
## 43 23.6 OJ 1.0  
## 44 26.4 OJ 1.0  
## 45 20.0 OJ 1.0  
## 46 25.2 OJ 1.0  
## 47 25.8 OJ 1.0  
## 48 21.2 OJ 1.0  
## 49 14.5 OJ 1.0  
## 50 27.3 OJ 1.0  
## 51 25.5 OJ 2.0  
## 52 26.4 OJ 2.0  
## 53 22.4 OJ 2.0  
## 54 24.5 OJ 2.0  
## 55 24.8 OJ 2.0  
## 56 30.9 OJ 2.0  
## 57 26.4 OJ 2.0  
## 58 27.3 OJ 2.0  
## 59 29.4 OJ 2.0  
## 60 23.0 OJ 2.0

* In order to include rows where supp is equal to VC, you will subset like this:

subset(ToothGrowth,supp=="VC")

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5  
## 11 16.5 VC 1.0  
## 12 16.5 VC 1.0  
## 13 15.2 VC 1.0  
## 14 17.3 VC 1.0  
## 15 22.5 VC 1.0  
## 16 17.3 VC 1.0  
## 17 13.6 VC 1.0  
## 18 14.5 VC 1.0  
## 19 18.8 VC 1.0  
## 20 15.5 VC 1.0  
## 21 23.6 VC 2.0  
## 22 18.5 VC 2.0  
## 23 33.9 VC 2.0  
## 24 25.5 VC 2.0  
## 25 26.4 VC 2.0  
## 26 32.5 VC 2.0  
## 27 26.7 VC 2.0  
## 28 21.5 VC 2.0  
## 29 23.3 VC 2.0  
## 30 29.5 VC 2.0

this restricts the data set to rows 1-30 only.

* In order to include rows where supp is equal to VC *and* dose is equal to 0.5, you will subset like this:

subset(ToothGrowth,supp=="VC" & dose=="0.5")

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5

this restricts the data to rows 1-10 only.

* In order to get the values of len where supp is equal to VC and dose is equal to 0.5, you can do this:

ToothGrowth$len[1:10]

## [1] 4.2 11.5 7.3 5.8 6.4 10.0 11.2 11.2 5.2 7.0

this outputs the values of len for rows 1-10 only, which is where supp=VC and dose=0.5 (as determined above). This is a vector.

### This markdown file, as well as the script files for questions 4 and 5, are located in my github repository:

### <https://github.com/njbaker61/Reproducibility-Class>