Homework 02

Christina Lee

09/17/2021

1.

a. Write a function that calculates the mean of any numeric vector you give it, without using the built-in mean() or sum() functions.

To answer this question, I first created a function called "mymean". I then initialized a variable called "nsum" to store the sum of all numbers in the vector. Next, I used "for" loop that adds whatever number or numbers that is put into "i" in the mymean function to nsum. I then used the return() to ask for result of nsum divided by the length (# of numbers) that is in the mymean function, which gives the mean of any vectors you give it.

```
mymean <- function(x){
  nsum <- 0
  for(i in x){
    nsum <- nsum + i
  }
  return(nsum/length(x))
}
```

This is just another way to write the for loop, which it follows similar logic as with the one above. The only difference is that I created another variable called "nlength" that basically tells how many numeric variables are in a given vector by using the length() function which serves to divide against the total sum of the vector. Another difference is the for loop, instead of "x" I wrote 1:nlength, I also added x[i] in the next line. Line 30 and 31 basically helps design a for loop that will keep adding elements from position 1 to the end length of the vector into nsum. At last, we divide nsum of the vector and the nlength of the vector to get the mean.

```
mymean <- function(x){
  nlength <- length(x)
  nsum <- 0
  for(i in 1:nlength){
    nsum <- nsum + x[i]
  }
  return(nsum/nlength)
}
```

b. Write a function that takes as its input a vector with four elements. If the sum of the first two elements is greater than the sum of the second two, the function returns the vector; otherwise it returns 0.

To answer this, I created a function called "fourinvect". I then created two variables, one is called "nsum1" which is used to add the first two elements in the vector, and another called "nsum2" that is used to add the second two elements in the vector. Next, I created an if statement that ask it to return my vector if nsum1 is greater than nsum2, and if else, return 0.

```
fourinvect <- function(x){
  nsum1 <- x[1] + x[2]
  nsum2 <- x[3] + x[4]
  if(nsum1>nsum2){
    return(x)
  }else{
    return(0)
  }
}
```

c. Write a function that calculates the Fibonacci sequence up to the nth element, where n is any number input into your function (its argument). The Fibonacci sequence is: 1, 1, 2, 3, 5, 8, 13, 21. . . , ie, each element is the sum of the previous two elements. One way to do this is to start off with the first two elements, c(1,1) and set an internal variable to this sequence. Then write a loop that counts up to n, where for each new element, you first calculate it by adding the last two elements of the growing sequence, and then stick that new number onto the growing sequence using c(). When the loop is finished, the function should return the final vector of Fibonacci numbers.

To answer this, I created a function called "fiboseq". I then created a vector and assigned first two elements c(1,1) of the fibonacci sequence in it. Next, I used for loop to set up some conditions, which for i from position 3 (since I already assigned the first two elements, I will start with the third) to num (the nth element) – Fn=Fn-1+Fn-2, which I wrote x[i-1]+x[i-2] so that each element in the vector is the sum of the previous two elements.

```
fiboseq <- function(num){
  x <- c(1,1)
  for(i in 3:num){
    x <- c(x, x[i-1] + x[i-2])
  }
   return(x)
}
```

d. Create a 4x4 matrix of the numbers 1 through 16. Use apply to apply your function from (a) to each of the rows in your matrix.

To answer this, I created a matrix called "m1" with numbers 1:16. I then used the apply function, where m1 is the input data, 1 means we apply the function to rows, and "mymean" is the function that I created in 1a.

```
m1 <- matrix(1:16, nrow=4,ncol=4)
m1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
apply(m1,1,mymean)
```

```
## [1]  7  8  9 10
```

2.

a. Using the airquality data set, construct an aggregated data set which shows the maximum wind and ozone by month.

To answer this, I used the aggregate function and cbind wind and ozone by Month using "~", from data =airquality, and max, which is asking for the maximum wind and ozone of the data set.

```
aggregate(cbind(Wind,Ozone) ~ Month, data=airquality,max)
```

```
##   Month Wind Ozone
## 1     5 20.1   115
## 2     6 20.7    71
## 3     7 14.9   135
## 4     8 15.5   168
## 5     9 16.6    96
```

b. Create the authors and books data sets following the example and data in the lecture, and then create a new data set by merging these two data sets by author, preserving all rows.

To answer this, I followed steps on Computational Statistics 2.3:Data Manipulation's 2.2 "Merge by" section, and to include all rows, I added all.x =TRUE and all.y=TRUE to the argument of the merge function.

```
authors <- data.frame(
    surname =c("Tukey", "Venables","Tierney","Ripley","McNeil"),
    nationality =c("US","Australia","US","UK","Australia"),
    stringsAsFactors=FALSE)
books <- data.frame(
    name =c("Tukey","Venables","Tierney","Ripley","Ripley","McNeil","R Core"),
    title =c("Exploratory Data Analysis","Modern Applied Statistics","LISP-STAT",
             "Spatial Statistics","Stochastic Simulation",
             "Interactive Data Analysis","An Introduction to R"),
    stringsAsFactors=FALSE)

# Viewing the datasets
authors
```

```
##    surname nationality
## 1    Tukey          US
## 2 Venables   Australia
## 3  Tierney          US
## 4   Ripley          UK
## 5   McNeil   Australia
```

```
books
```

```
##       name                     title
## 1    Tukey Exploratory Data Analysis
## 2 Venables Modern Applied Statistics
## 3  Tierney                 LISP-STAT
## 4   Ripley        Spatial Statistics
## 5   Ripley     Stochastic Simulation
## 6   McNeil Interactive Data Analysis
## 7   R Core      An Introduction to R
```

```
# New data set that merges two data sets by author while including rows.
bookmerge <- merge(authors,books, by.x="surname",by.y="name",all.x=TRUE,all.y=TRUE)
bookmerge
```

```
##    surname nationality                     title
## 1   McNeil   Australia Interactive Data Analysis
## 2   R Core        <NA>      An Introduction to R
## 3   Ripley          UK        Spatial Statistics
## 4   Ripley          UK     Stochastic Simulation
## 5  Tierney          US                 LISP-STAT
## 6    Tukey          US Exploratory Data Analysis
## 7 Venables   Australia Modern Applied Statistics
```

c. Take the following string and replace every instance of "to" or "To" with "2" :

To be, or not to be – that is the question: Whether ’tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to take arms against a sea of troubles, And by opposing end them. To die – to sleep – No more…

To answer this, I created a vector called “to2to” and used the gsub function to search for “to” and “To” in the first argument, and added “2” to the second argument, which is what I want to replace it with.

```
to2to <- gsub("to|To","2",c("To be, or not to be --
that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them.
To die -- to sleep --
No more..."))

to2to
```

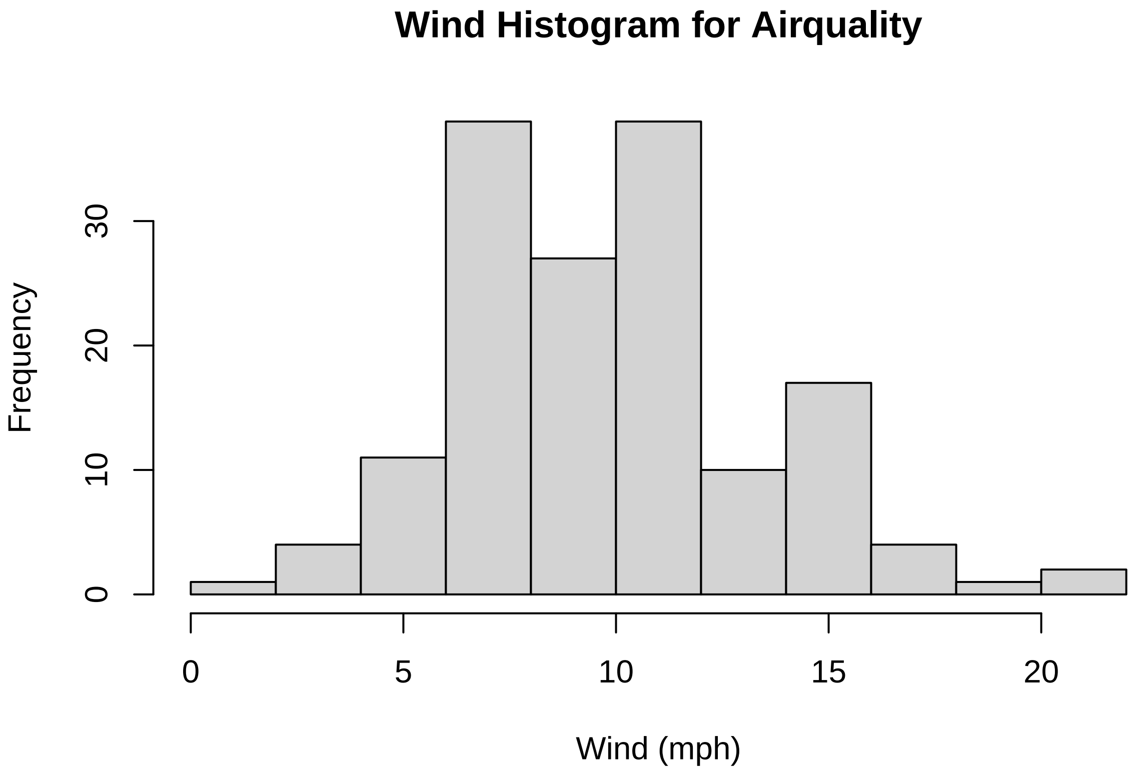
```
## [1] "2 be, or not 2 be --\nthat is the question:\nWhether 'tis nobler in the mind 2 suffer\nThe slin
```

3.

a. Create a histogram using the base R graphics using some dataset or variable other than the one in the lessons. Always make sure your graph has well-labeled x and y axes and an explanatory title.

To answer this, I used the wind variable from the airquality dataset, and followed instructions from Computational Statistics 2.4:Graphics’s 1.1 “Histogram” section.

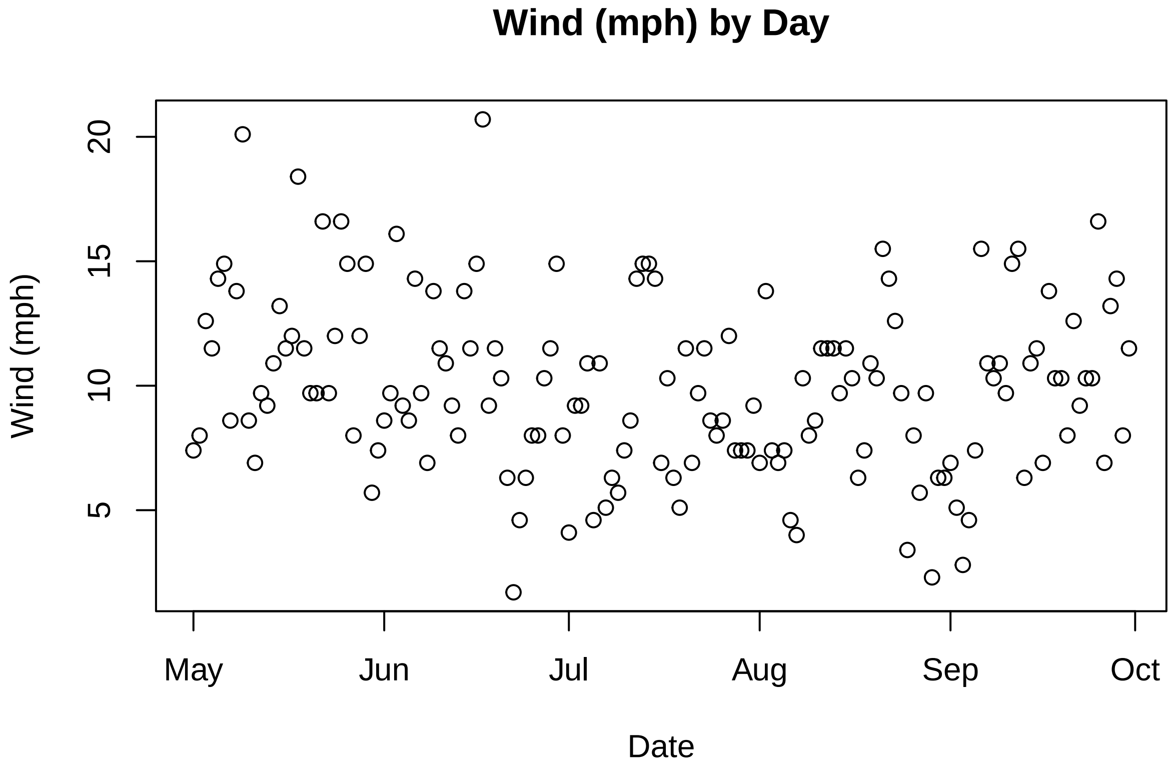
```
hist(airquality$Wind,main="Wind Histogram for Airquality",xlab="Wind (mph)")
```



b. Create a scatter plot using the base R graphics, again with some variable other than the one in the lessons.

To answer this, I created a two variable scatter plot from the airquality data set. I used the wind variable from the airquality data set and examined wind as a function of date (also examined a different year, which I examined 1973 instead of 1972 from the module). Again, I followed instructions provided on the lesson example under the 1.3 "Scatter plot" section.

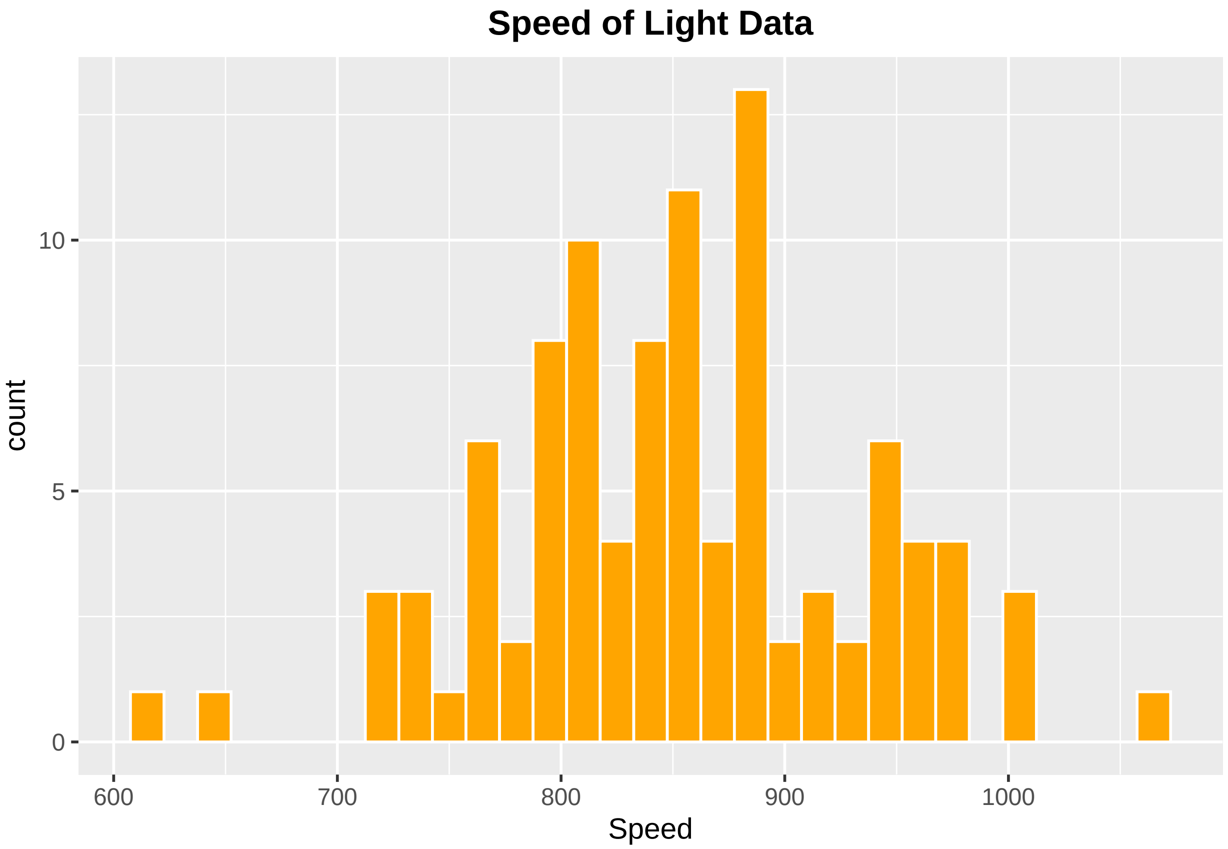
```
airdate <- as.Date(paste("1973","-",airquality$Month,"-",airquality$Day,sep=""))
plot(airdate,airquality$Wind,xlab="Date",ylab="Wind (mph)",main="Wind (mph) by Day")
```



c. Create a histogram using ggplot, using some new data. In this and the later plots, please tinker with the settings using the examples in http://www.cookbook-r.com/Graphs/ to make it prettier.

To answer this, I created a histogram via ggplot2 function using the R built-in data set called "morley" which is a classical data of Michelson on measurements done in 1879 on the speed of light. I used "Speed" (speed-of-light measurements) as my variable. I also added a bolded title that is centered, and color fill ins (went with blue) to this histogram to make it look more colorful and clear.

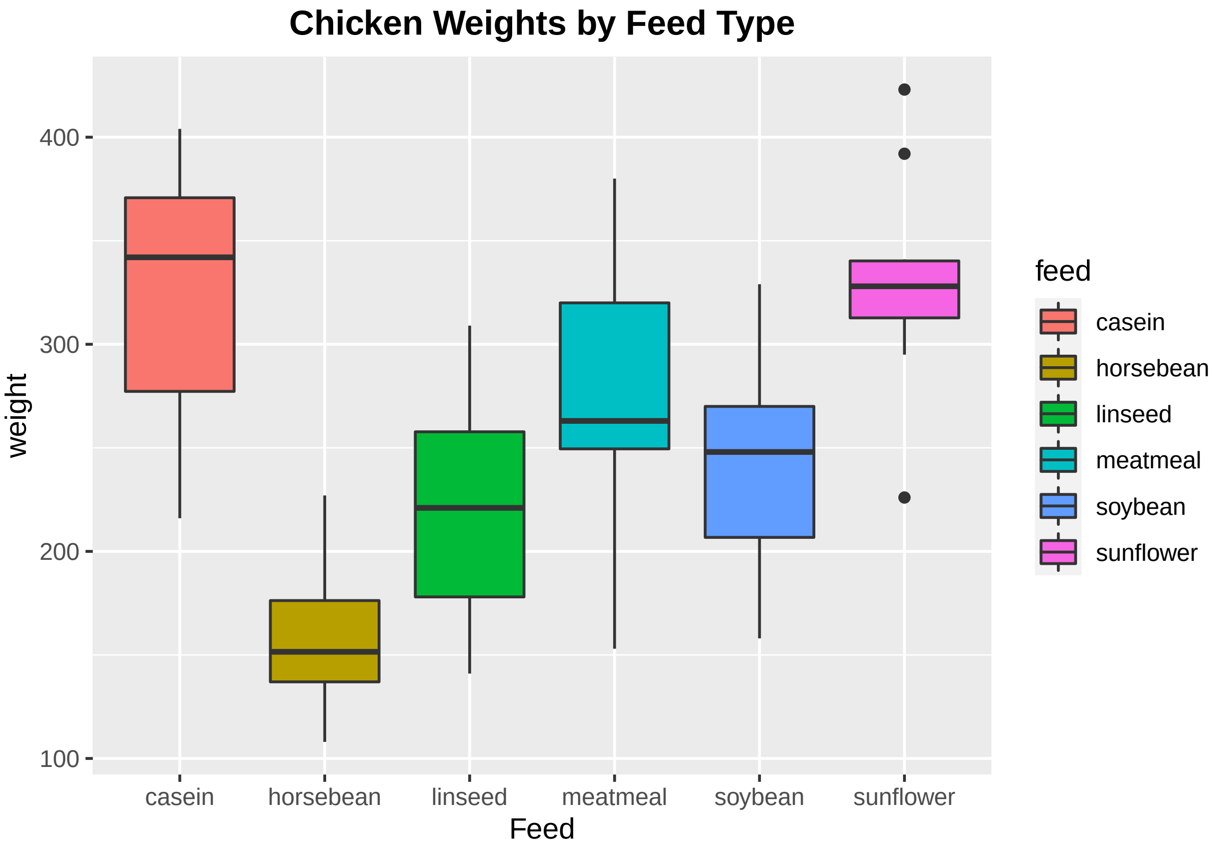
```
library(ggplot2)
MichelsonSpeed <- ggplot(data=morley,aes(x=Speed)) +
  geom_histogram(binwidth=15,color="white",fill="orange")
MichelsonSpeed + ggtitle("Speed of Light Data") +
theme(plot.title = element_text(hjust = 0.5, face="bold"))
```



d. Create a box plot (with multiple categories) using ggplot, using some new data.

To answer this, I created a box plot with a different R built in data set called “chickwts”, which is an experiment conducted to measure and compare the effectiveness of various feed types on the growth rate of chickens (measured in weights). I added a bolded title, and added colors to the different categories on the box plot.

```
chickweights <- ggplot(data=chickwts,aes(x=feed,y=weight, fill=feed)) +
  geom_boxplot() + xlab("Feed")
chickweights + ggtitle("Chicken Weights by Feed Type") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"))
```

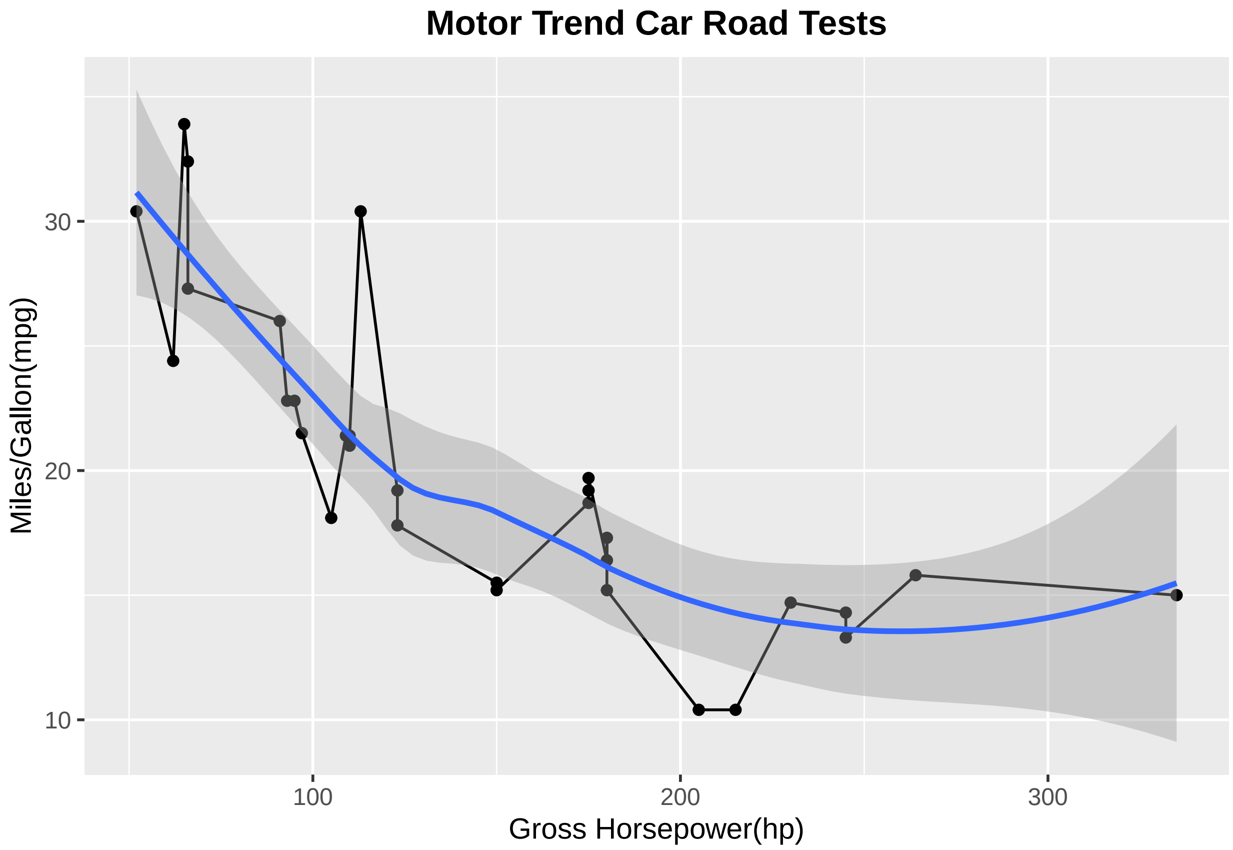


e. Create a scatter plot using ggplot, using some new data.

To answer this, I created a scatter plot with another R built in data set called "mtcars", which the data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (74 car models). For this scatter plot, I plotted mpg against hp , added a bolded title that is centered, lined the scatter points, and added a regression line (in color blue) with a confidence interval of 0.95. The "se" is a logical value, which if TRUE, confidence interval will display around smooth, "fullrange" is also a logical value, which if TRUE, the fit spans the full range of the plot.

```
cartesting <- ggplot(data=mtcars,aes(x=hp,y=mpg)) + geom_point()+
  geom_line() + xlab("Gross Horsepower(hp)") +
  ylab("Miles/Gallon(mpg)") +
  geom_smooth(method="auto",se=TRUE, fullrange=FALSE, level=0.95)
cartesting + ggtitle("Motor Trend Car Road Tests") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


Motor Trend Car Road Tests
Miles/Gallon(mpg)
30
20
10
100
200
300
Gross Horsepower(hp)

“‘