CMDA-3654

Homework 1

Eduardo Salvador

Due as a .pdf upload

setwd("~/Desktop/CMDA /Assignments/HW1")

Instructions:

I have given you this assignment as an .Rmd (R Markdown) file.

- Change the name of the file to: `Lastname_Firstname_CMDA_3654_HW1.Rmd` , and your output should therefore match but with a `.pdf` extension.
- You need to edit the R Markdown file by filling in the chunks appropriately with your code. Output will be generated automatically when you compile the document.
- You also need to add your own text before and after the chunks to explain what you are doing or to interpret the output.
- Feel free to add additional chunks if needed. I **will not** be providing assignments to you like this for the entire semester, just long enough for you to learn how to do it for yourself.

Required: The final product that you turn in must be a .pdf file.

- You can Knit this document directly to a PDF if you have LaTeX installed (which is preferred).
- If you absolutely can't get LaTeX installed and/or working, then you can compile to a .html first, by clicking on the arrow button next to knit and selecting Knit to HTML.
- You must then print you .html file to a .pdf by using first opening it in a web browser and then printing to a .pdf

---

Problem 1: (30 pts) Learning about new `R` functions and matrix multiplication.

a. Do the following using only a single line of code. First, learn how to use the `rep()` function. Using `rep()` create the following vector $\mathbf{x}$:

$$\mathbf{x} = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7]^T$$

then convert this vector into a 4×7 matrix, called `A` formed by filling it by the rows. In an additional line, please print `A` to verify your result.

#rep is a funciton to replicate and used times to say the number of times I want to replicte each number

```
A<-matrix(rep(c(1,2,3,4,5,6,7),times=c(1,2,3,4,5,6,7))^(T),nrow=4,ncol=7, byrow=T)
A
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    1    2    2    3    3    3    4
[2,]    4    4    4    5    5    5    5
[3,]    5    6    6    6    6    6    6
[4,]    7    7    7    7    7    7    7
```

#Repeat each number, this many times with number of rows being 4 and number of columns being 7 byrow

b. Print out the entry $a_{1,4}$, that is, the from the first row and fourth column of matrix A.

```
A[1,4]
```

```
[1] 3
```

#Select number in 1st row of 4th column

c. Using a single line, convert $\mathbf{x}$ into a 7×4 matrix called `B` by filling in by rows first. For comparison, take the transpose of `A` and comment on the difference.

```
B<-matrix(rep(c(1,2,3,4,5,6,7),times=c(1,2,3,4,5,6,7))^(T), nrow=7,ncol=4, byrow=T)
B
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    2    3
[2,]    3    3    4    4
[3,]    4    4    5    5
[4,]    5    5    5    6
[5,]    6    6    6    6
[6,]    6    7    7    7
[7,]    7    7    7    7
```

```
t(A)
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    5    7
[2,]    2    4    6    7
[3,]    2    4    6    7
[4,]    3    5    6    7
[5,]    3    5    6    7
[6,]    3    5    6    7
[7,]    4    5    6    7
```

#Recreate matrix A but this time number of rows is 7 and number of columns is 3 ordered by rows #The difference can be seen in the arrangement of both matrixces, B being by row and the transpose of A being by column

d. Learn how to perform matrix multiplications in R. Then perform the matrix multiplication AB, and report the result.

```
AB<-A %*% B
AB
```

```
     [,1] [,2] [,3] [,4]
[1,]   94   98  102  106
[2,]  152  161  169  178
[3,]  191  202  214  225
[4,]  224  238  252  266
```

With %% one can perform multiplications

#The matrix got reduced to a 4 by 4 instead of a 4 by 7 or 7 by 4

e. Convert matrix AB to a data frame, and save it as `my_first_df` .

```
my_first_df<-data.frame(AB)
my_first_df
```

X1 <dbl>	X2 <dbl>	X3 <dbl>	X4 <dbl>
94	98	102	106
152	161	169	178
191	202	214	225
224	238	252	266

4 rows

#data.frame function converges any type of data into a dataframe

f. Add a column named `experiment` to `my_first_df` , where the first two observations are the string `"+"` , and the last two observations are the string `"-"` , and print the resulting data frame. Convert this column to a factor. Print out your final data frame along with the output from `str(my_first_df)` .

```
experiment<-c("+","+","-","-")
experiment
```

```
[1] "+" "+" "-" "-"
```

```
newdata<-cbind(my_first_df,experiment)
newdata
```

X1 <dbl>	X2 <dbl>	X3 <dbl>	X4 <dbl>	experiment <chr>
94	98	102	106	+

X1	X2	X3	X4	experiment
<dbl>	<dbl>	<dbl>	<dbl>	<chr>
152	161	169	178	+
191	202	214	225	-
224	238	252	266	-

4 rows

```
newdata$experiment=factor(experiment)
str(newdata)
```

```
'data.frame':   4 obs. of  5 variables:
 $ X1        : num  94 152 191 224
 $ X2        : num  98 161 202 238
 $ X3        : num  102 169 214 252
 $ X4        : num  106 178 225 266
 $ experiment: Factor w/ 2 levels "-","+": 2 2 1 1
```

#The c function combines values into vector or list desired and by using cbind I was able to combine objects by rows or columns

Problem 2: (20 pts) Loading in and exploring data with `R`.

The `puso` dataset contains information from NOAA concerning sediment contents of soil samples, along with a label discerning whether the soil is considered toxic or not.

a. Begin by reading in the `puso.csv` file into your `R` session, and properly storing it as a dataframe (note is does have a header). Show the first 5 rows of the first 8 columns to demonstrate that you loaded it in correctly.

```
puso <- read.csv(file = "/Users/eduardosalvador/Desktop/CMDA /Assignments/HW1/puso.csv", header =T)
puso[1:5,1:8]
```

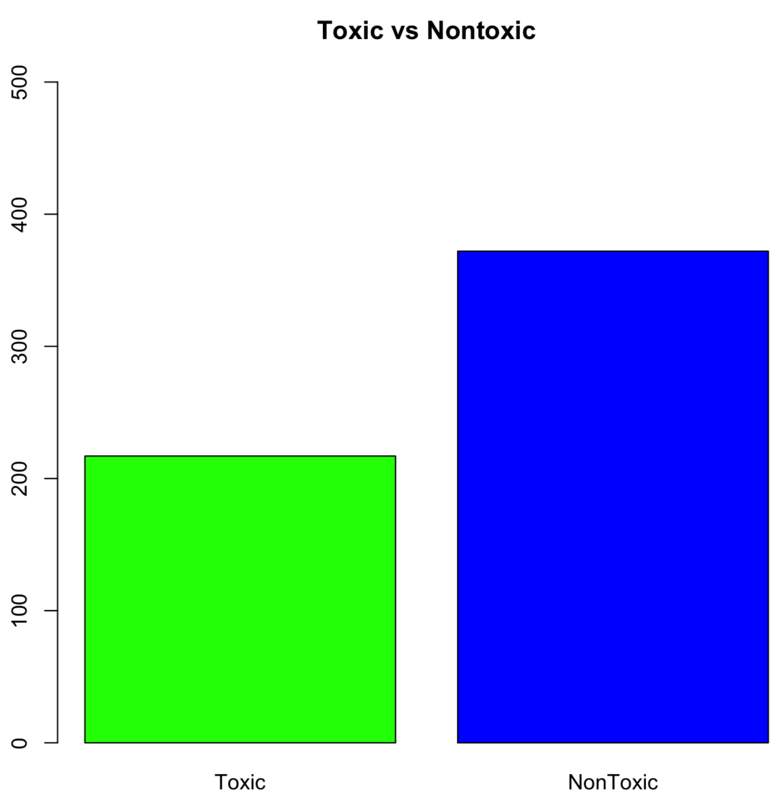
	TOXCODE <lgl>	toxic <int>	lars <dbl>	lcad <dbl>	lchr <dbl>	lcop <dbl>	llead <dbl>	lmerc <dbl>
1	TRUE	1	0.5596158	-2.040221	NA	2.397895	2.944439	-3.506558
2	FALSE	0	0.5596158	-2.659260	NA	1.871802	2.639057	-2.659260
3	FALSE	0	0.4700036	-2.407946	NA	3.178054	3.258097	-2.813411
4	TRUE	1	0.3364722	-2.207275	NA	3.295837	3.135494	-2.813411
5	TRUE	1	0.6418539	-2.995732	NA	2.397895	2.639057	-2.525729

5 rows

#For this mac use /Users/eduardosalvador/Desktop/CMDA / and if dataframe has header, equal it to true

b. Create a barplot depicting the proportion of toxic samples and non-toxic samples. Be sure to create appropriate axis labels, make the bars *distinct* colors, give the binary values descriptive names (1 = Toxic, 0 = Non-Toxic) and create a descriptive main title for your plot. There are a number of different ways to accomplish this task, so don't feel like there is **only** one solution.

```
barplot( table( puso$toxic ) , names.arg = c("Toxic", "NonToxic"),
        ylim = c(0, 500), col=c("green","blue"),
        main = "Toxic vs Nontoxic" )
```



Convertaded dataframe into table them used boxplot function and detailed it, making the toxic green and nontoxic blue

c. Seperate the dataset into two seperate datasets: one containing samples classified as toxic, and those that are not. Report the first 5 rows of each data set.

```
Toxic <- subset(puso, TOXCODE == T)
Toxic[1:5,]
```

	TOXCODE <lgl>	toxic <int>	lars <dbl>	lcad <dbl>	lchr <dbl>	lcop <dbl>	llead <dbl>	lmerc <dbl>	lnick <dbl>
1	TRUE	1	0.5596158	-2.040221	NA	2.397895	2.944439	-3.506558	2.397895
4	TRUE	1	0.3364722	-2.207275	NA	3.295837	3.135494	-2.813411	3.044522
5	TRUE	1	0.6418539	-2.995732	NA	2.397895	2.639057	-2.525729	2.174752
8	TRUE	1	0.3715636	-2.525729	NA	3.218876	2.890372	-3.218876	2.995732
9	TRUE	1	0.5596158	-1.966113	NA	3.091042	3.583519	-2.995732	2.772589

5 rows | 1-10 of 25 columns

```
Nontoxic<-subset(puso,TOXCODE==F)
Nontoxic[1:5,]
```

	TOXCODE <lgl>	toxic <int>	lars <dbl>	lcad <dbl>	lchr <dbl>	lcop <dbl>	llead <dbl>	lmerc <dbl>	lnick <dbl>
2	FALSE	0	0.5596158	-2.659260	NA	1.871802	2.639057	-2.659260	2.197225
3	FALSE	0	0.4700036	-2.407946	NA	3.178054	3.258097	-2.813411	2.944439
6	FALSE	0	0.3001046	-2.525729	NA	2.995732	2.708050	-2.995732	3.091042
7	FALSE	0	0.3715636	-2.525729	NA	2.944439	2.890372	-3.912023	3.135494
11	FALSE	0	0.6931472	-2.995732	NA	3.044522	2.944439	-3.912023	3.218876

5 rows | 1-10 of 25 columns

#Used subset function to separate datasets and where TAXCODE when True is Toxic and when False is Nontoxic

d. For each dataset, create a summary table for each variable in the data set. The descriptive statistics should include the mean, standard deviation, range, and number of missing values for that given variable. *Hint:* A very simple way to do this is to create an empty matrix, fill it with the needed values, and to name the rows and columns appropriately. Print your table nicely using `kable()` or `pandoc.table()`

```
MeanT<-(apply(Toxic,2,mean,na.rm=T))
SDT<-(apply(Toxic,2,sd,na.rm=T))
RangeT<-(apply(Toxic,2,max,na.rm=T))-(apply(Toxic,2,min,na.rm=T))
MissingValuesT<-(apply(Toxic,2,function(x) sum(is.na(x))))

MeanNT<-(apply(Nontoxic,2,mean,na.rm=T))
SDNT<-(apply(Nontoxic,2,sd,na.rm=T))
RangeNT<-(apply(Nontoxic,2,max,na.rm=T))-(apply(Nontoxic,2,min,na.rm=T))
MissingValuesNT<-(apply(Nontoxic,2,function(x) sum(is.na(x))))

Tmatrix<-rbind(MeanT, SDT, RangeT, MissingValuesT)[,3:24]
Tmatrix
```

```
                   lars      lcad        lchr      lcop     llead     lmerc
MeanT          2.283596 -0.340354   3.5320143  4.128842  3.388027 -1.682065
SDT            1.059227  1.160208   0.6692169  1.031039  1.397571  1.257340
RangeT         8.533454  6.756932   3.1734135  6.145615  6.916054  7.649693
MissingValuesT 79.000000  0.000000 218.0000000 12.000000 79.000000  0.000000
                   lnick       lsilv       lzinc      lacen      lacpt      lanth
MeanT          3.2553517 -0.8968391  4.6263286  3.598841  3.357118  4.103581
SDT            0.6852677  1.2366288  0.8416856  1.660251  1.416576  1.767530
RangeT         3.5553481  6.7011410  5.5347061 11.097410 11.211820 12.154779
MissingValuesT 79.0000000  0.0000000 81.0000000  1.000000  0.000000  0.000000
                   lbaa      lban      lbap     lchry     lflan     lflen
MeanT          4.635919  3.446150  4.557458  4.993998  5.375908  3.697498
SDT            1.932893  1.639532  1.893056  1.984656  1.940302  1.694946
RangeT        12.611538 10.308953 11.512925 12.583367 12.468437 11.211820
MissingValuesT  0.000000  6.000000  0.000000  1.000000  6.000000  0.000000
                  lmeth    lnapt     lphen     lpyre
MeanT          3.760638 4.054879  4.971757  5.435742
SDT            1.629392 1.823850  1.874068  1.965622
RangeT         8.281471 9.803404 11.320554 12.128111
MissingValuesT 80.000000 0.000000  0.000000  6.000000
```

```
NTmatrix<-rbind(MeanNT, SDNT, RangeNT, MissingValuesNT)[,3:24]
NTmatrix
```

```
                    lars      lcad       lchr      lcop     llead     lmerc
MeanNT          1.8400845 -1.472469  3.3401377  3.260988  2.599807 -2.583319
SDNT            0.8838987  1.371818  0.6332274  1.018562  1.053191  1.100811
RangeNT         4.8009148  5.886104  2.8081337  5.991465  8.318742  5.913503
MissingValuesNT 46.0000000  0.000000 78.0000000 17.000000 46.000000  0.000000
                   lnick     lsilv      lzinc    lacen    lacpt    lanth
MeanNT          3.1890359 -2.016522  4.0348399 2.550112 2.430355 2.966341
SDNT            0.6431917  1.251645  0.6352055 1.216577 1.206211 1.601188
RangeNT         3.5045150  6.066108  2.9519297 6.551080 7.431003 8.389360
MissingValuesNT 46.0000000  0.000000 47.0000000 4.000000 0.000000 1.000000
                    lbaa     lban     lbap    lchry    lflan    lflen     lmeth
MeanNT          3.315819 2.599771 3.241363 3.654486 4.048441 2.623760  2.496656
SDNT            1.725771 1.335641 1.699215 1.859554 1.816015 1.314582  1.258656
RangeNT         8.536996 8.242756 8.131531 9.126959 8.987197 6.437752  9.290383
MissingValuesNT 0.000000 1.000000 0.000000 4.000000 1.000000 0.000000 18.000000
                   lnapt    lphen    lpyre
MeanNT          2.839548 3.816098 3.937206
SDNT            1.530545 1.688103 1.863607
RangeNT         7.090077 7.901007 8.581732
MissingValuesNT 1.000000 0.000000 1.000000
```

```
kable(Tmatrix)
```

	lars	lcad	lchr	lcop	llead	lmerc	lnick	lsilv	lzinc	lacen	lacpt	
MeanT	2.283596	-0.340354	3.5320143	4.128842	3.388027	-1.682065	3.2553517	-0.8968391	4.6263286	3.598841	3.357118	4.1
SDT	1.059227	1.160208	0.6692169	1.031039	1.397571	1.257340	0.6852677	1.2366288	0.8416856	1.660251	1.416576	1.7
RangeT	8.533454	6.756932	3.1734135	6.145615	6.916054	7.649693	3.5553481	6.7011410	5.5347061	11.097410	11.211820	12.1
MissingValuesT	79.000000	0.000000	218.0000000	12.000000	79.000000	0.000000	79.0000000	0.0000000	81.0000000	1.000000	0.000000	0.0

```
kable(NTmatrix)
```

	lars	lcad	lchr	lcop	llead	lmerc	lnick	lsilv	lzinc	lacen	lacpt	la
MeanNT	1.8400845	-1.472469	3.3401377	3.260988	2.599807	-2.583319	3.1890359	-2.016522	4.0348399	2.550112	2.430355	2.966
SDNT	0.8838987	1.371818	0.6332274	1.018562	1.053191	1.100811	0.6431917	1.251645	0.6352055	1.216577	1.206211	1.601
RangeNT	4.8009148	5.886104	2.8081337	5.991465	8.318742	5.913503	3.5045150	6.066108	2.9519297	6.551080	7.431003	8.389
MissingValuesNT	46.0000000	0.000000	78.0000000	17.000000	46.000000	0.000000	46.0000000	0.000000	47.0000000	4.000000	0.000000	1.000

#Found on google that apply function can find any metric applied to all columns(2) or rows(1) #Used na.rm to as a logical value that strips any NA value from the dataset #For the range I looked for the max value and subtracted the min of each column #For the missing values I looked up a function in google called sum(is.na) which calculate the amount of NA for every column using apply #Combine each calculated field by row starting from column 3 to 24 since the first 2 are not to be considered #Used kable to be able to create tables

Problem 3: (25 pts) Common Plots in `Base R.`

Consider the dataset `cars.csv` . It contains information about 406 cars (in 407 rows - the first row is the names of the variables). Information on car name, mileage (MPG), number of cylinders, displacement, horsepower, weight, acceleration, model, and country of origin are available.

Answer the following questions based on this dataset.

a. Identify the types of each variable available in the dataset. Be as specific as you possibly can (Quantitative variables can be either Continuous vs discrete, Categorical can be either Nominal vs Ordinal etc).

```
#For this mac use /Users/eduardosalvador/Desktop/CMDA / and if dataframe has header, equal it to true
cars <- read.csv(file = "/Users/eduardosalvador/Desktop/CMDA /Assignments/HW1/cars.csv", header =T)

typeof(cars$Car)
```

```
[1] "character"
```

```
#Car is a Nominal type of vairable which falls in the Categorical group
typeof(cars$MPG)
```

```
[1] "double"
```

```
#MPG is a Discrete type of variable which falls in the Quantitative group
typeof(cars$Cylinders)
```

```
[1] "integer"
```

```
#Cylinders is a Discrete type of variable which falls in the Quantitative group
typeof(cars$Displacement)
```

```
[1] "double"
```

```
#Displacement is a Contineous type of variable which falls in the Quantitative group
typeof(cars$Horsepower)
```

```
[1] "double"
```

```
#Horsepower is a Discrete type of variable which falls in the Quantitative group
typeof(cars$Weight)
```

```
[1] "double"
```

```
#Weight is a Continuous type of variable which falls in the Quantitative group
typeof(cars$Acceleration)
```

```
[1] "double"
```

```
#Acceleration is a Discrete type of variable which falls in the Quantitative group
typeof(cars$Model)
```

```
[1] "integer"
```

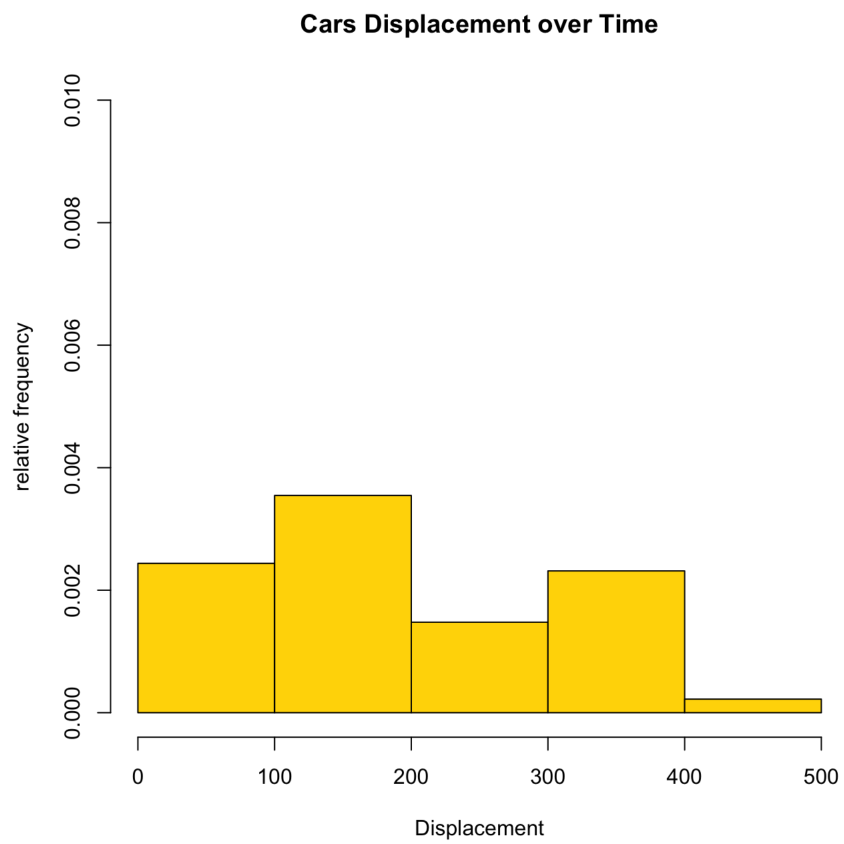
```
#Model is a Ordinal type of variable which falls in the Categorical group
typeof(cars$Origin)
```

```
[1] "character"
```

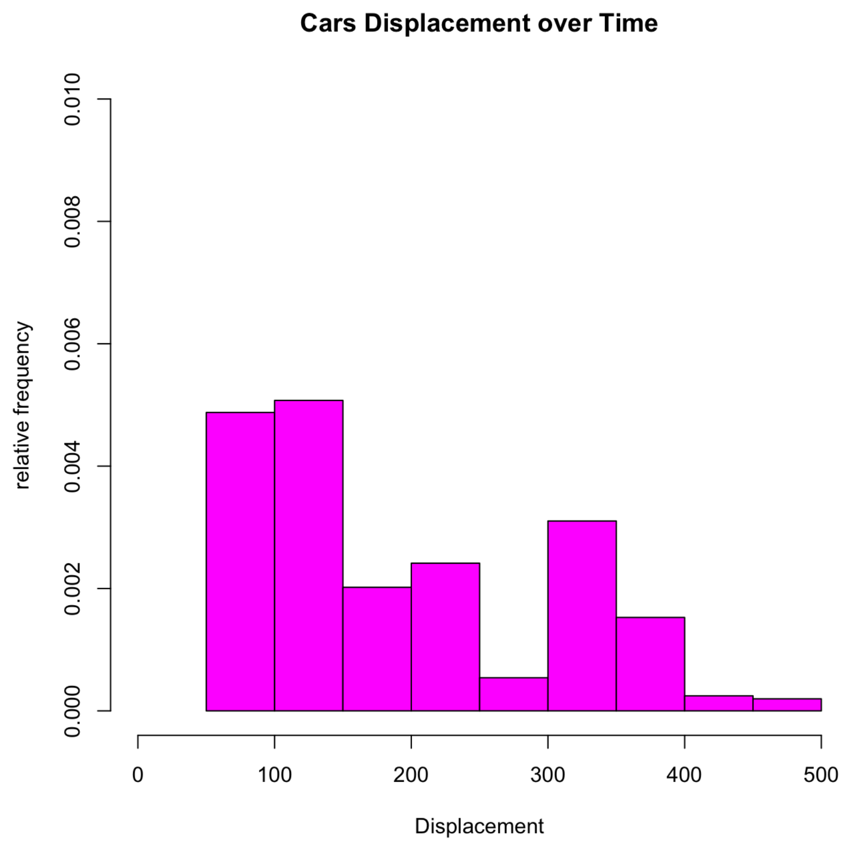
```
#Origin is a Nominal type of variable which falls in the Categorical group
```

b. Make a histogram for the displacement variable first using `breaks = 5` and again with `breaks = 10` . Use relative frequencies (or densities). Label all the axes properly. Identify the skew of the histogram and the mode of the data.

```
hist(cars$Displacement, probability = T, xlim = c(0, 500) , ylim = c(0, 0.010),
     xlab = "Displacement", ylab = "relative frequency", main = "Cars Displacement over Time",
     breaks = 5, col="gold")
```




```
hist(cars$Displacement, probability = T, xlim = c(0, 500) , ylim = c(0, 0.010),
     xlab = "Displacement", ylab = "relative frequency", main = "Cars Displacement over Time",
     breaks = 10, col="magenta")
```




```
library(e1071)
skewness(cars$Displacement)
```

```
[1] 0.6890094
```

```
#For mode created a table of cars displacement, coverted it into a data.frame, then to a character to be able to
 convert it gagin into numeric so that I could figure out which has the most frequencies. Then output this Var wh
ich has the most frequency
mode_1<-as.data.frame(table(cars$Displacement))
mode_1
```

Var1 <fct>	Freq <int>
68	1
70	3
71	2
72	1
76	1
78	1
79	6
80	1
81	1
83	1

1-10 of 83 rows Previous **1** 2 3 4 5 6 ... 9 Next

```
as.numeric(as.character(mode_1$Var1))
```

```
 [1]  68.0  70.0  71.0  72.0  76.0  78.0  79.0  80.0  81.0  83.0  85.0  86.0
[13]  88.0  89.0  90.0  91.0  96.0  97.0  97.5  98.0 100.0 101.0 104.0 105.0
[25] 107.0 108.0 110.0 111.0 112.0 113.0 114.0 115.0 116.0 119.0 120.0 121.0
[37] 122.0 130.0 131.0 133.0 134.0 135.0 140.0 141.0 144.0 145.0 146.0 151.0
[49] 155.0 156.0 163.0 168.0 171.0 173.0 181.0 183.0 198.0 199.0 200.0 225.0
[61] 231.0 232.0 250.0 258.0 260.0 262.0 267.0 302.0 304.0 305.0 307.0 318.0
[73] 340.0 350.0 351.0 360.0 383.0 390.0 400.0 429.0 440.0 454.0 455.0
```

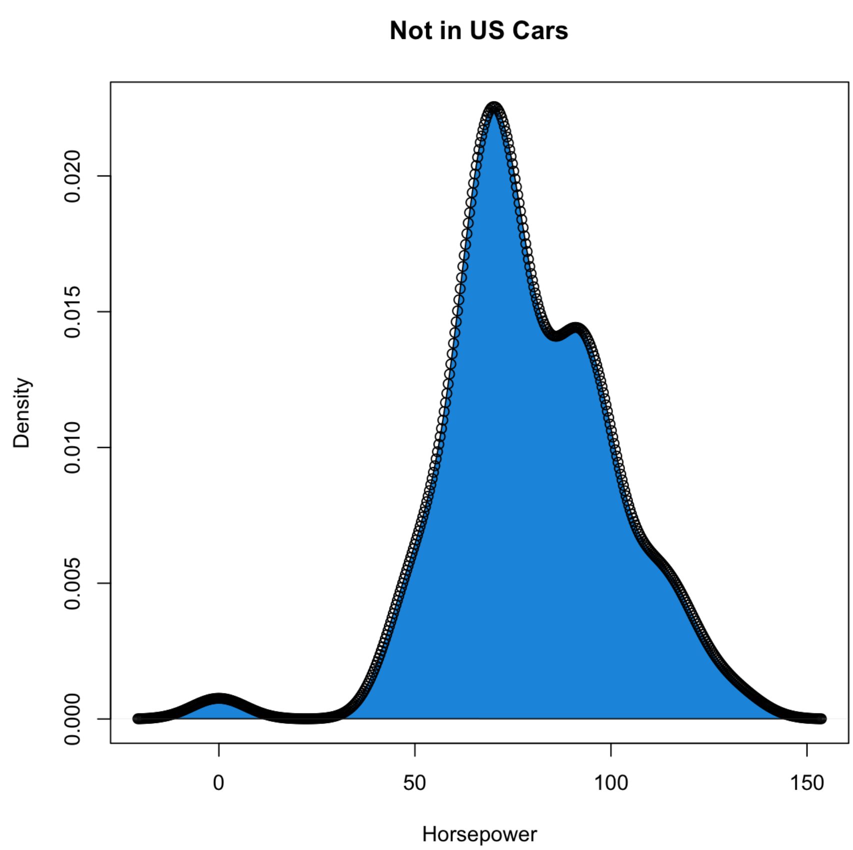
```
max_mode<-max(mode_1$Freq)
mode_1[mode_1$Freq==max_mode,1]
```

```
[1] 97
83 Levels: 68 70 71 72 76 78 79 80 81 83 85 86 88 89 90 91 96 97 97.5 ... 455
```

#Created histogram with details and for the skeweness downloaded package e1071 to get function skewness. #For mode created a table of cars displacement, coverted it into a data.frame, then to a character to be able to convert it gagin into numeric so that I could figure out which has the most frequencies. Then output this Var which has the most frequency

c. Make a kernel density estimation plot for the horsepower variable. Make a kernel density estimation plot for the horsepower variable, but this time exclude all vehicles that originate in the US.

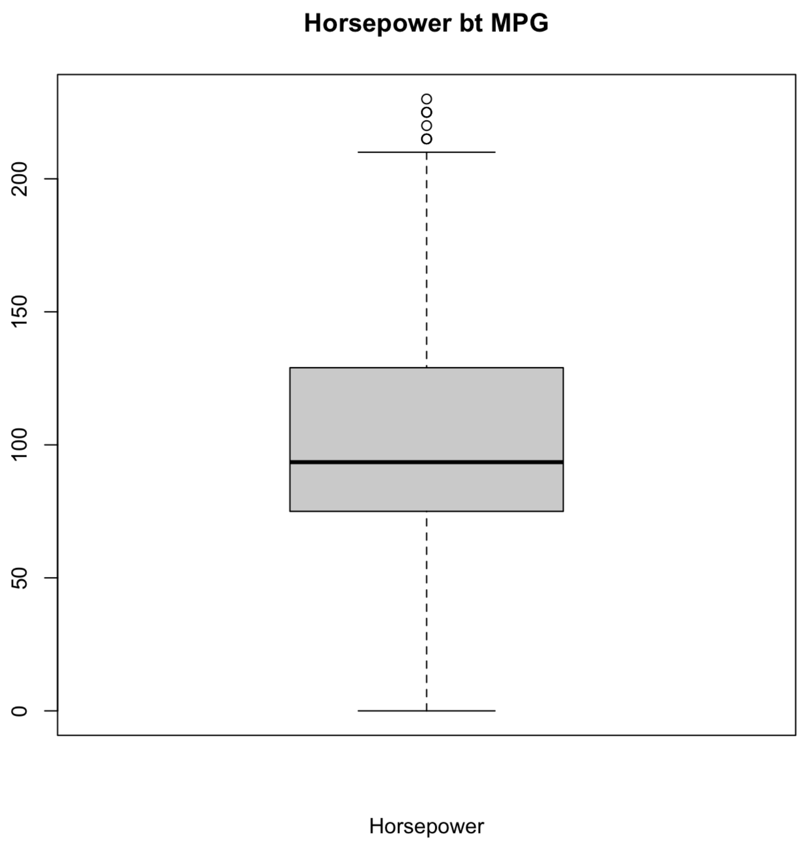
```
Not_US<-subset(cars, Origin!="US")
plot(density(Not_US$Horsepower),
     main = "Not in US Cars",
     xlab = "Horsepower",
     ylab = "Density", polygon(density(Not_US$Horsepower), col = "#1b98e0", ))
```



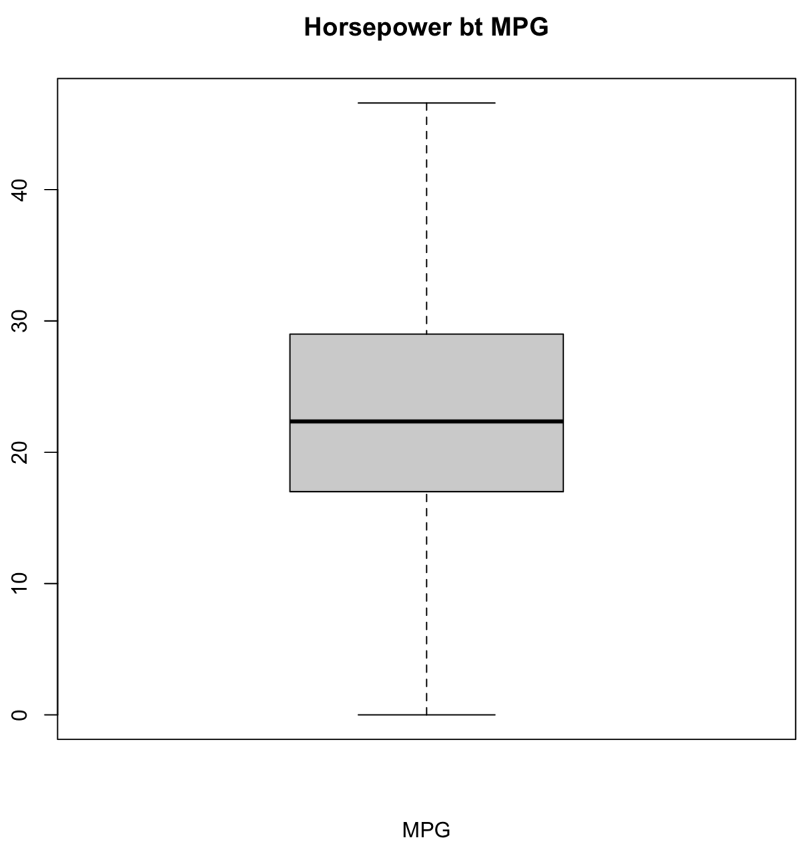
#Created subset of cars data and use != as not equal to, to get rid of cars from the US. #Then created a density plot adding colors and a line on the edges with polygon and color function.

d. Generate a boxplot for the Horsepower variable. Discuss briefly what the boxplot indicates about the horsepower of the cars in the dataset. Generate a boxplot for the MPG variable. Do you notice any suspicious observations or outliers for MPG? Explain.

```
boxplot( cars$Horsepower ,
        xlab = "Horsepower", ylab = "", main="Horsepower bt MPG")
```



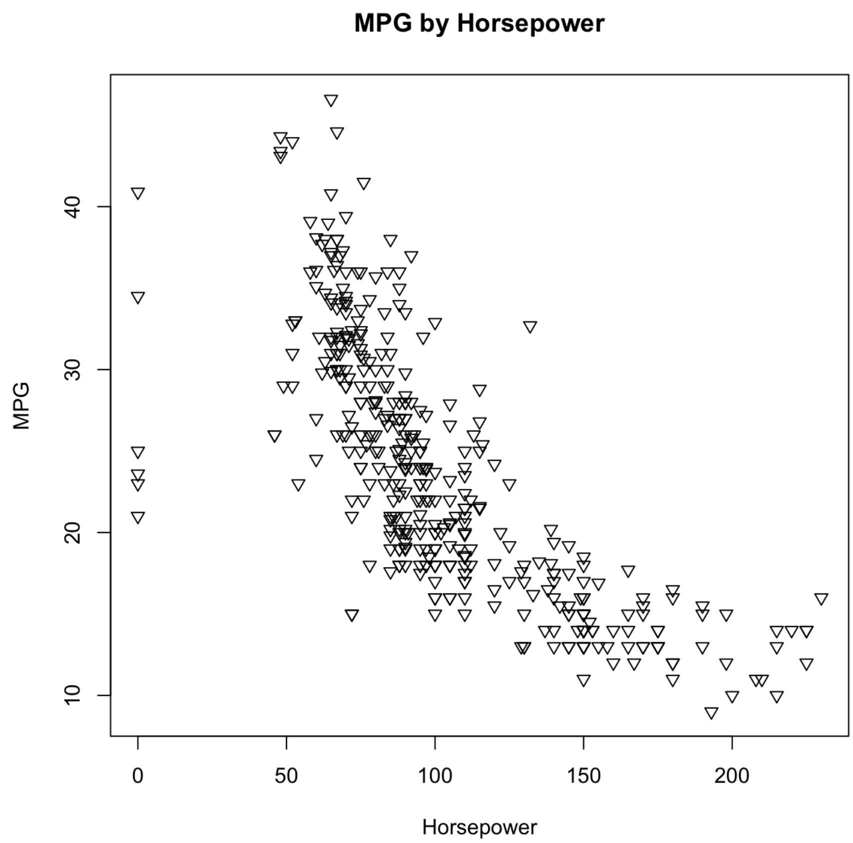

```
boxplot( cars$MPG ,
         xlab = "MPG", ylab = "", main="Horsepower bt MPG")
```



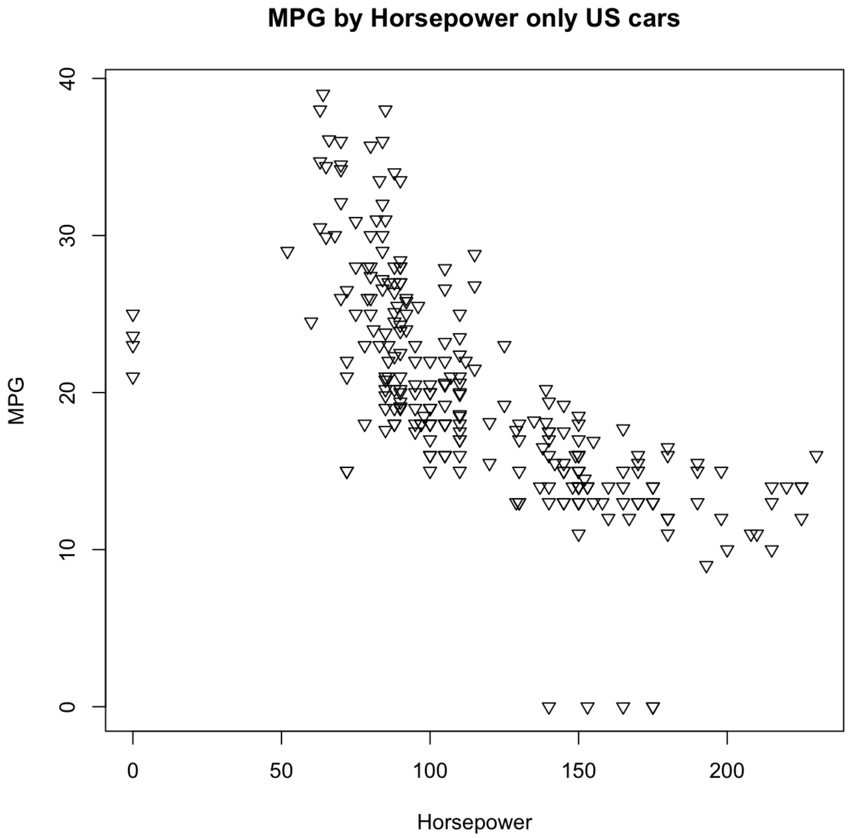
#MPG outliers are every value of MPG which results in 0 since it is unreasonale to have a car with MPG at 0

e. For the cars that do not have suspicious observations for MPG, plot the MPG versus Horsepower. Repeat the above, but this time make three scatter plots. One for US cars, one for European Cars, and finally one for Japanese Cars.

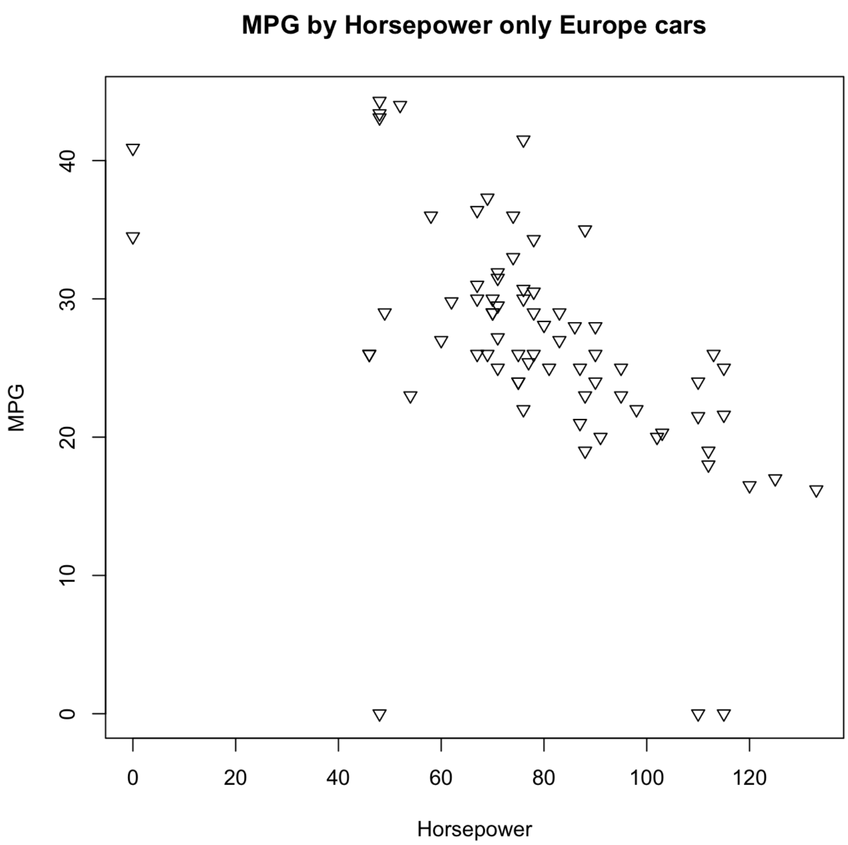
```
Disregard_NMPG<-subset(cars, MPG!="0")
plot( Disregard_NMPG$MPG ~ Disregard_NMPG$Horsepower,
        xlab = "Horsepower", ylab = "MPG", main="MPG by Horsepower", pch=25)
```



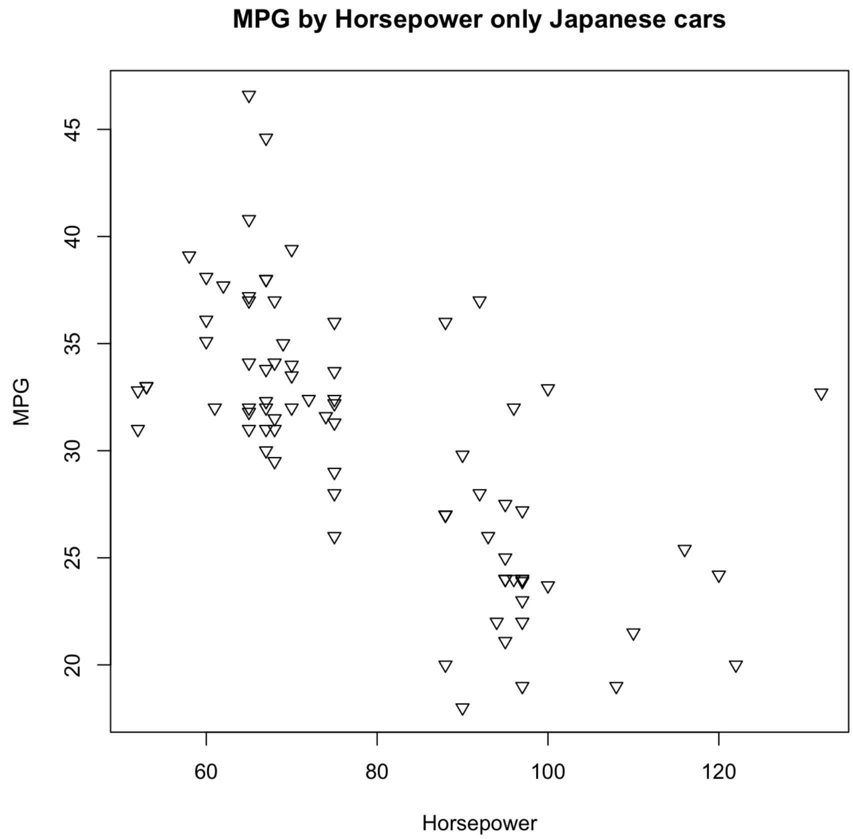

```
US_cars<-subset(cars, Origin=="US", MPG!="0")
plot( US_cars$MPG ~ US_cars$Horsepower,
        xlab = "Horsepower", ylab = "MPG", main="MPG by Horsepower only US cars", pch=25)
```




```
Europe_cars<-subset(cars, Origin=="Europe", MPG!="0")
plot( Europe_cars$MPG ~ Europe_cars$Horsepower,
         xlab = "Horsepower", ylab = "MPG", main="MPG by Horsepower only Europe cars", pch=25)
```



```
Japanese_cars<-subset(cars, Origin=="Japan", MPG!="0")
plot( Japanese_cars$MPG ~ Japanese_cars$Horsepower,
        xlab = "Horsepower", ylab = "MPG", main="MPG by Horsepower only Japanese cars", pch=25)
```



#Created it to disregard MPG’s outlier and then created a boxplot of MPG versus Horsepower with ~

Problem 4: [25 pts]

Install the R package `babynames` . Load the `babynames` data and answer the following questions. Report R code and answers.

a. Describe the dataset in two sentences. How many rows and columns does the dataset have?

```
library(babynames)
```

The babynames dataset has 5 columns and 1,924,665 rows containing several entires of babies from 1880 to 2017, sex, Name of the baby, prop which is the variable prop represents the proportion of all applicants of that sex in that year that had that name and n which represents the number of applications in that year for that name and sex.

b. How many unique names are there in the dataset? Why is this number different from the number of rows in (a)?

```
length(unique(babynames$name))
```

```
[1] 97310
```

##97310, it is different from the number of rows since some names repeat itself

c. What were the most popular male names for the years 1900, 1925, 1950, 1975, 2000? What were the most popular female names for the years 2010, 2011, 2012, 2013, 2014?

```
Names_1900<-subset(subset(babynames, sex=="M"), year=="1900" )
Maxquant_1900=max(Names_1900$n)
Names_1900[Names_1900[,4]==Maxquant_1900, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
1900	M	John	9829	0.06062307

1 row

```
Names_1925<-subset(subset(babynames, sex=="M"), year=="1925")
Maxquant_1925=max(Names_1925$n)
Names_1925[Names_1925[,4]==Maxquant_1925, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
1925	M	Robert	60896	0.05288659

1 row

```
Names_1950<-subset(subset(babynames, sex=="M"), year=="1950")
Maxquant_1950=max(Names_1950$n)
Names_1950[Names_1950[,4]==Maxquant_1950, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
1950	M	James	86239	0.04740837

1 row

```
Names_1975<-subset(subset(babynames, sex=="M"), year=="1975")
Maxquant_1975=max(Names_1975$n)
Names_1975[Names_1975[,4]==Maxquant_1975, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
1975	M	Michael	68454	0.0421767

1 row

```
Names_2010<-subset(subset(babynames, sex=="F"), year=="2010")
Maxquant_2010=max(Names_2010$n)
Names_2010[Names_2010[,4]==Maxquant_2010, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
2010	F	Isabella	22905	0.01169646

1 row

```
Names_2011<-subset(subset(babynames, sex=="F"), year=="2011")
Maxquant_2011=max(Names_2011$n)
Names_2011[Names_2011[,4]==Maxquant_2011, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
2011	F	Sophia	21837	0.011285

1 row

```
Names_2012<-subset(subset(babynames, sex=="F"), year=="2012")
Maxquant_2012=max(Names_2012$n)
Names_2012[Names_2012[,4]==Maxquant_2012, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
2012	F	Sophia	22304	0.01151924

1 row

```
Names_2013<-subset(subset(babynames, sex=="F"), year=="2013")
Maxquant_2013=max(Names_2013$n)
Names_2013[Names_2013[,4]==Maxquant_2013, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
2013	F	Sophia	21213	0.01102629

1 row

```
Names_2014<-subset(subset(babynames, sex=="F"), year=="2014")
Maxquant_2014=max(Names_2014$n)
Names_2014[Names_2014[,4]==Maxquant_2014, ]
```

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
2014	F	Emma	20924	0.01072117

1 row

#Created a subset with all male or female babynames of a specific year then looked for the max repeats and output the max quantity for every single year

d. What are the 10 most popular male baby names across years? What are the 10 most popular female baby names across years?

```
PopularMaleNames<-subset(babynames, sex=="M")
DifferenciationM<-PopularMaleNames[PopularMaleNames[,3]==unique(PopularMaleNames$name),]

MaleNamesbyOrder<-DifferenciationM[order(DifferenciationM$n, decreasing=TRUE),]
MaleNamesbyOrder[1:10,3]
```

name
<chr>
John

name <chr>
William
James
Charles
George
Frank
Ricardo
Joseph
Thomas
Henry

1-10 of 10 rows

```
PopularFemaleNames<-subset(babynames, sex=="F")
DifferenciationF<-PopularFemaleNames[PopularFemaleNames[,3]==unique(PopularFemaleNames$name),]

FemaleNamesbyOrder<-DifferenciationF[order(DifferenciationF$n, decreasing=TRUE),]
FemaleNamesbyOrder[1:10,3]
```

name <chr>
Stephanie
Mary
Anna
Emma
Elizabeth
Minnie
Margaret
Ida
Alice
Bertha

1-10 of 10 rows

#Created a subset of babies only males and only females #Made the names unique so that it doesn't repeat themselves in the list #Ordered the unique names from most to least and outputed the solution