CMDA-3654

Homework 1

Michael La Vance

Due as a .pdf upload

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.

```
#creating matrix
A <- matrix( rep(1:7, c(1:7)), nrow = 4, byrow = TRUE)
print(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
```

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
```

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(1:7, c(1:7)), nrow = 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
```

```
#Matrix B and the transpose of A are the same matrix
```

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

```
A %*% B
```

```
     [,1] [,2] [,3] [,4]
[1,]   52   85  107  126
[2,]   85  148  188  224
[3,]  107  188  241  287
[4,]  126  224  287  343
```

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

```
my_first_df <- as.data.frame(A %*% B)
```

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)`.

```
my_first_df$experiment <- c("+","+","-","-")
my_first_df[,5] <- factor(my_first_df[,5])
my_first_df
```

```
   V1  V2  V3  V4 experiment
1  52  85 107 126          +
2  85 148 188 224          +
3 107 188 241 287          -
4 126 224 287 343          -
```

```
str(my_first_df)
```

```
'data.frame':   4 obs. of  5 variables:
 $ V1        : num  52 85 107 126
 $ V2        : num  85 148 188 224
 $ V3        : num  107 188 241 287
 $ V4        : num  126 224 287 343
 $ experiment: Factor w/ 2 levels "-","+": 2 2 1 1
```

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.

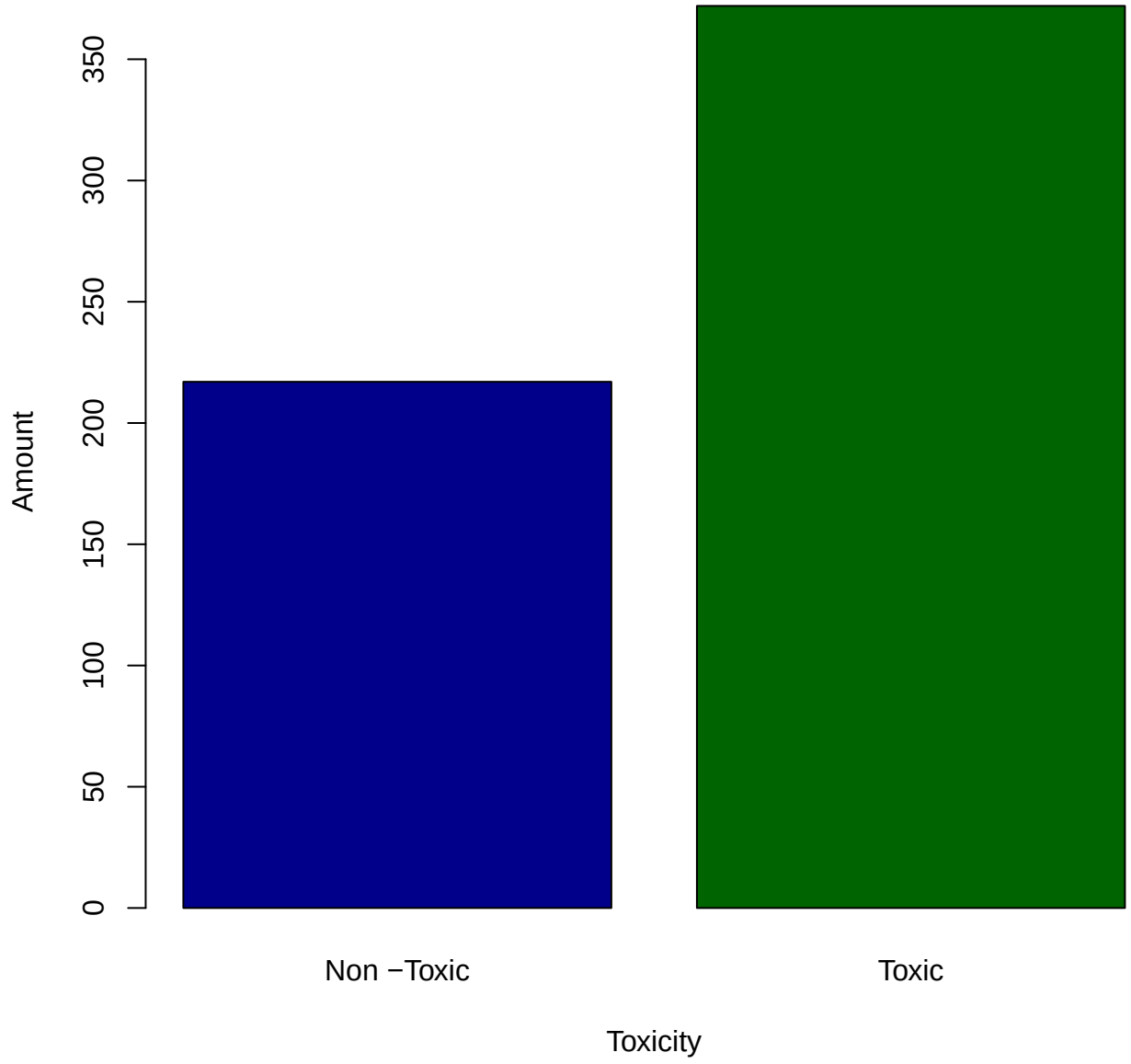
a. Begin by reading in the puso.csv file into your R session, and properly storing it as a dataframe (note is does have a header). Show the first 5 rows of the first 8 columns to demonstrate that you loaded it in correctly.

```
pusodata <- read.csv(file = "puso.csv", header = TRUE)
pusodata[1:5, 1:8]
```

```
  TOXCODE toxic      lars      lcad lchr     lcop    llead     lmerc
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
```

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(pusodata$toxic),
        main="Amount of Toxic and non-Toxic samples",
        names=c("Non -Toxic","Toxic"),ylab="Amount",
        col=c("darkBlue","darkgreen"),
        xlab = "Toxicity")
```



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.

```
nontoxic <- subset(pusodata, pusodata$toxic == 0)
toxic <- subset(pusodata, pusodata$toxic == 1)
print(toxic[1:5,])
```

```
  TOXCODE toxic      lars      lcad lchr     lcop    llead     lmerc    lnick
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
      lsilv    lzinc    lacen    lacpt    lanth     lbaa     lban     lbap
1 -2.813411 3.912023 2.397895 2.397895 2.397895 2.397895 2.397895 2.397895
4 -3.101093 4.465908 2.397895 2.397895 2.397895 2.397895 2.397895 2.397895
5 -2.733368 3.931826 4.787492 2.442347 4.418841 5.828946 4.304065 5.913503
8 -2.302585 4.343805 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
```

```
9 -2.120264 4.143135 2.442347 2.442347 2.442347 3.258097 2.442347 2.442347
     lchry    lflan    lflen    lmeth    lnapt    lphen    lpyre
1 3.555348 3.496508 2.397895 2.397895 2.397895 2.397895 3.951244
4 3.555348 3.496508 2.397895 2.397895 2.397895 2.397895 3.610918
5 6.214608 6.522093 5.075174 3.663562 5.393628 6.579251 6.492240
8 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
9 3.737670 3.988984 2.442347 2.442347 2.442347 3.332205 4.787492
```

```
print(nontoxic[1:5,])
```

```
   TOXCODE toxic      lars      lcad lchr     lcop    llead     lmerc    lnick
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
       lsilv    lzinc    lacen    lacpt    lanth     lbaa     lban     lbap
2  -2.813411 3.806662 2.302585 2.302585 2.302585 2.302585 2.302585 2.302585
3  -2.900422 4.189655 2.397895 2.397895 2.397895 2.397895 2.397895 2.397895
6  -3.101093 4.127134 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
7  -2.995732 4.110874 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
11 -2.733368 4.406719 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
      lchry    lflan    lflen    lmeth    lnapt    lphen    lpyre
2  2.302585 2.302585 2.302585 2.302585 2.302585 2.302585 3.178054
3  2.397895 2.397895 2.397895 2.397895 2.397895 2.397895 2.397895
6  2.251292 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
7  2.251292 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
11 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292 2.251292
```

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()`

```
#creating blank matrices
toxSum <- matrix(,nrow = 5, ncol = 22)
nontoxSum <- matrix(,nrow = 5, ncol = 22)
#adding row names
rownames(toxSum) <- c("Mean","Standard Deviation","Min","Max",
                      "# of Missing data")
rownames(nontoxSum) <- c("Mean","Standard Deviation","Min","Max"
                         ,"# of Missing data")
#copying column names
colnames(toxSum) <- colnames(toxic)[3:24]
colnames(nontoxSum) <- colnames(toxic)[3:24]
#getting summary stats and putting it into the 2 tables
for (i in 3:24) {
  toxSum[1, i-2] <- mean(toxic[,i],na.rm = TRUE)
  toxSum[2, i-2] <- sd(toxic[,i],na.rm = TRUE)
  toxSum[3, i-2] <- range(toxic[,i],na.rm = TRUE)[1]
  toxSum[4, i-2] <- range(toxic[,i],na.rm = TRUE)[2]
  toxSum[5, i-2] <- sum(is.na(toxic[,i]))
  nontoxSum[1, i-2] <- mean(nontoxic[,i],na.rm = TRUE)
  nontoxSum[2, i-2] <- sd(nontoxic[,i],na.rm = TRUE)
  nontoxSum[3, i-2] <- range(nontoxic[,i],na.rm = TRUE)[1]
  nontoxSum[4, i-2] <- range(nontoxic[,i],na.rm = TRUE)[2]
  nontoxSum[5, i-2] <- sum(is.na(nontoxic[,i]))
}
#printing tables
kable(toxSum, digits = 3)
```

	lars	lcad	lchr	lcop	llead	lmerc	lnick	lsilv	lzinc	lacen	lacpt	lanth	lbaa	lban	lbap	lchry	lflan	lflen	lmeth	lnapt	lphen	lpyre
Mean	2.284	-0.340	3.532	4.129	3.388	-1.682	3.255	-0.897	4.626	3.599	3.357	4.104	4.636	3.446	4.557	4.994	5.376	3.697	3.761	4.055	4.972	5.436
Standard Deviation	1.059	1.160	0.669	1.031	1.398	1.257	0.685	1.237	0.842	1.660	1.417	1.768	1.933	1.640	1.893	1.985	1.940	1.695	1.629	1.824	1.874	1.966
Min	-1.187	-3.912	1.686	1.569	0.182	-5.298	1.386	-4.200	2.708	-0.693	-0.693	0.000	0.000	-0.916	0.000	0.182	1.609	-0.693	0.049	1.386	1.386	
Max	7.346	2.845	4.860	7.714	7.098	2.351	4.942	2.501	8.243	10.404	10.519	12.155	12.619	9.393	11.513	12.766	14.078	10.518	8.975	9.852	12.707	13.514
# of Missing data	79.000	0.000	218.000	12.000	79.000	0.000	79.000	0.000	81.000	1.000	0.000	0.000	0.000	6.000	0.000	1.000	6.000	0.000	80.000	0.000	0.000	6.000

```
kable(nontoxSum, digits = 3)
```

	lars	lcad	lchr	lcop	llead	lmerc	lnick	lsilv	lzinc	lacen	lacpt	lanth	lbaa	lban	lbap	lchry	lflan	lflen	lmeth	lnapt	lphen	lpyre
Mean	1.840	-1.472	3.340	3.261	2.600	-2.583	3.189	-2.017	4.035	2.550	2.430	2.966	3.316	2.600	3.241	3.654	4.048	2.624	2.497	2.840	3.816	3.937
Standard Deviation	0.884	1.372	0.633	1.019	1.053	1.101	0.643	1.252	0.635	1.217	1.206	1.601	1.726	1.336	1.699	1.860	1.816	1.315	1.259	1.531	1.688	1.864
Min	-0.301	-4.605	2.028	0.000	-2.996	-5.298	1.589	-4.605	2.639	0.693	-0.223	0.000	0.693	0.000	0.000	1.099	0.470	-1.204	0.405	0.693	1.099	
Max	4.500	1.281	4.836	5.991	5.323	0.615	5.094	1.461	5.591	7.244	7.208	8.389	8.537	7.550	8.132	9.127	10.086	6.908	8.086	7.496	8.594	9.680
# of Missing data	46.000	0.000	78.000	17.000	46.000	0.000	46.000	0.000	47.000	4.000	0.000	1.000	0.000	1.000	0.000	4.000	1.000	0.000	18.000	1.000	0.000	1.000

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.

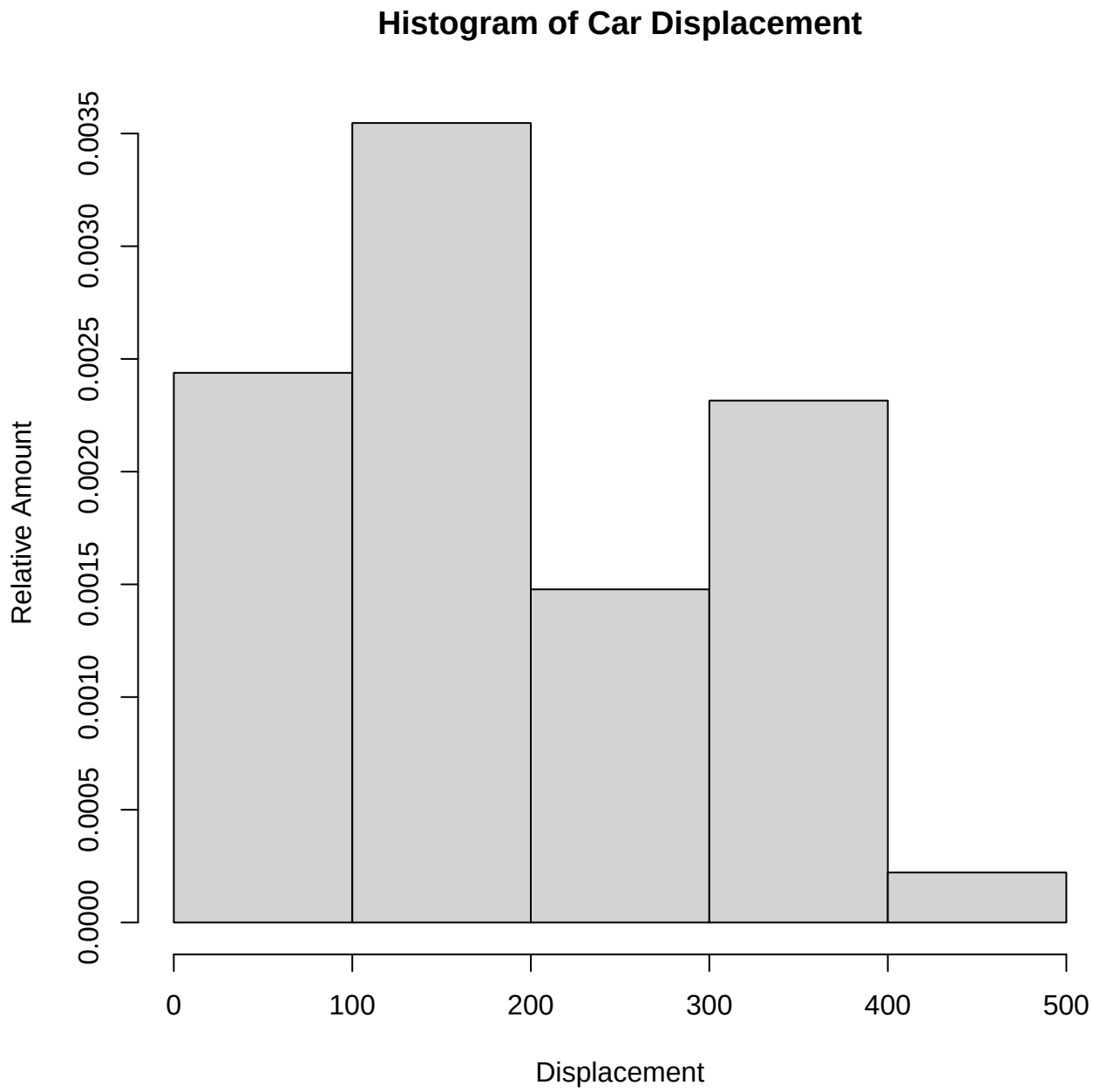
Answer the following questions based on this dataset.

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).

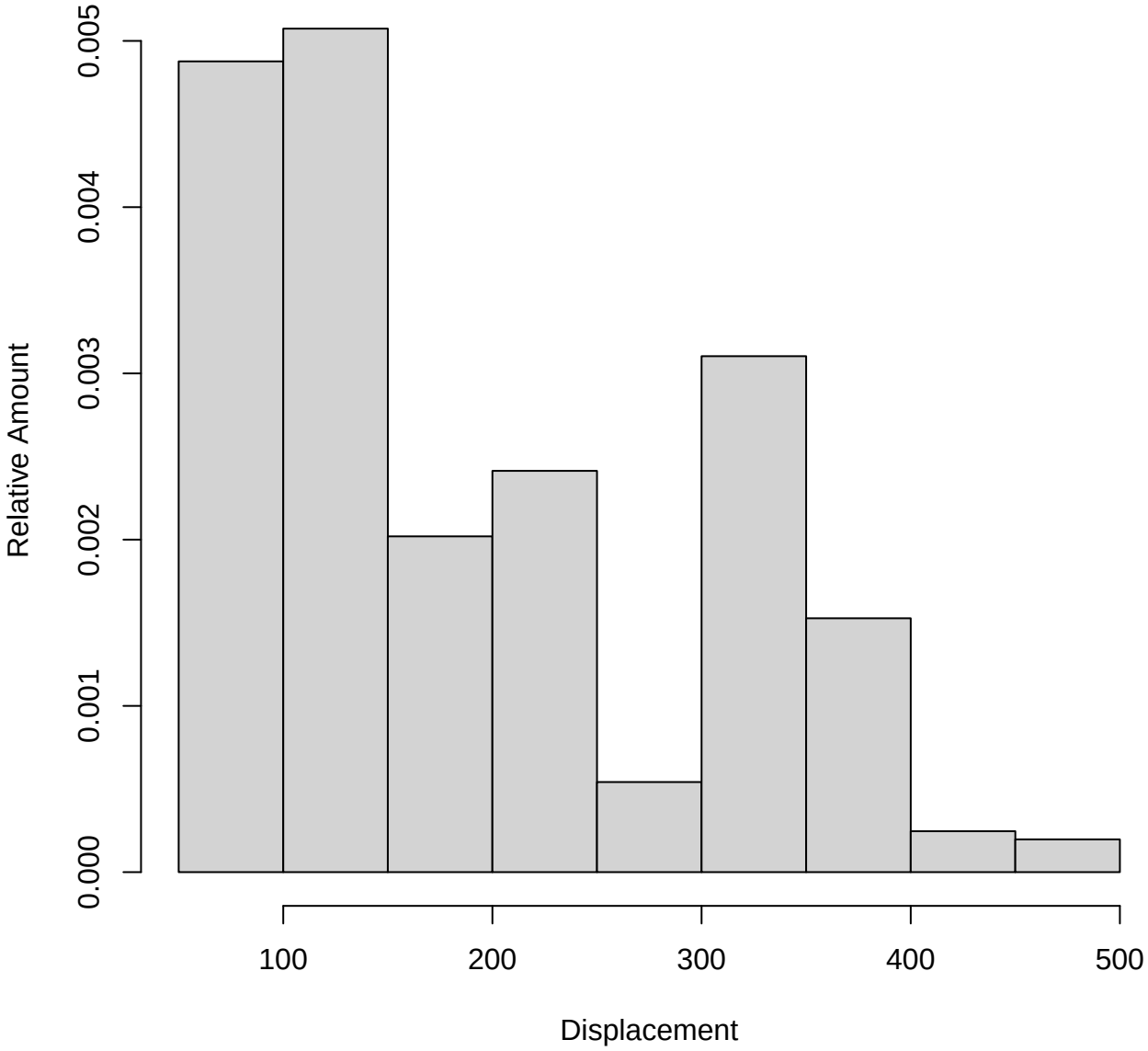
```
#reading file
cardata <- read.csv(file = "cars.csv", header = TRUE)
#Car: nominal
#MPG: continuous
#Cylinders: discrete
#Displacement: continuous
#Horsepower: discrete
#Weight: continuous
#Acceleration: continuous
#Model:Ordinal
#Origin: Nominal
```

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.

```
#creating histograms
hist(cardata$Displacement, breaks = 5, xlab = "Displacement",
     ylab = "Relative Amount", main = "Histogram of Car Displacement", freq = FALSE)
```



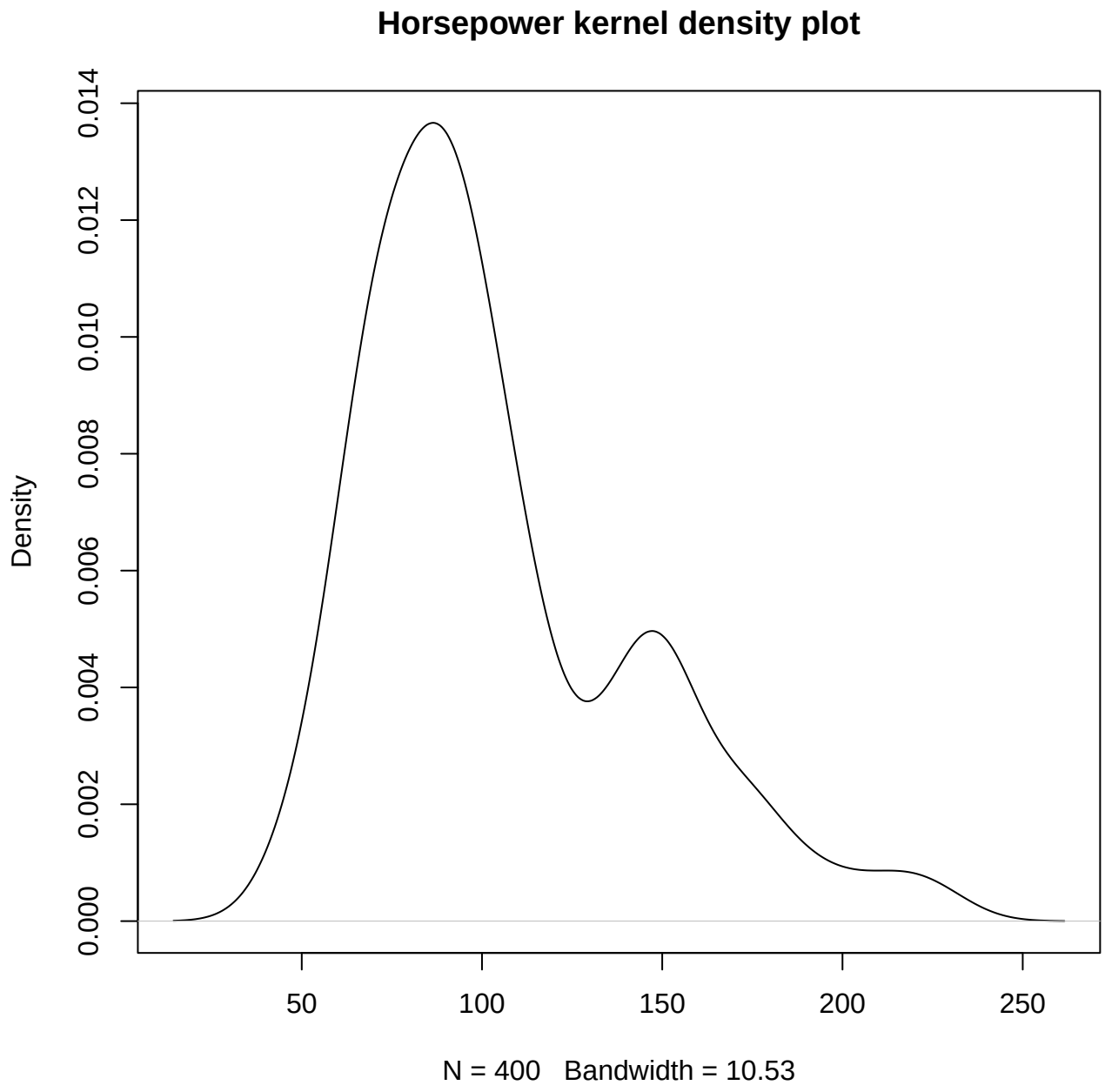

```
hist(cardata$Displacement, breaks = 10, xlab = "Displacement",
     ylab = "Relative Amount", main = "Histogram of Car Displacement", freq = FALSE)
```



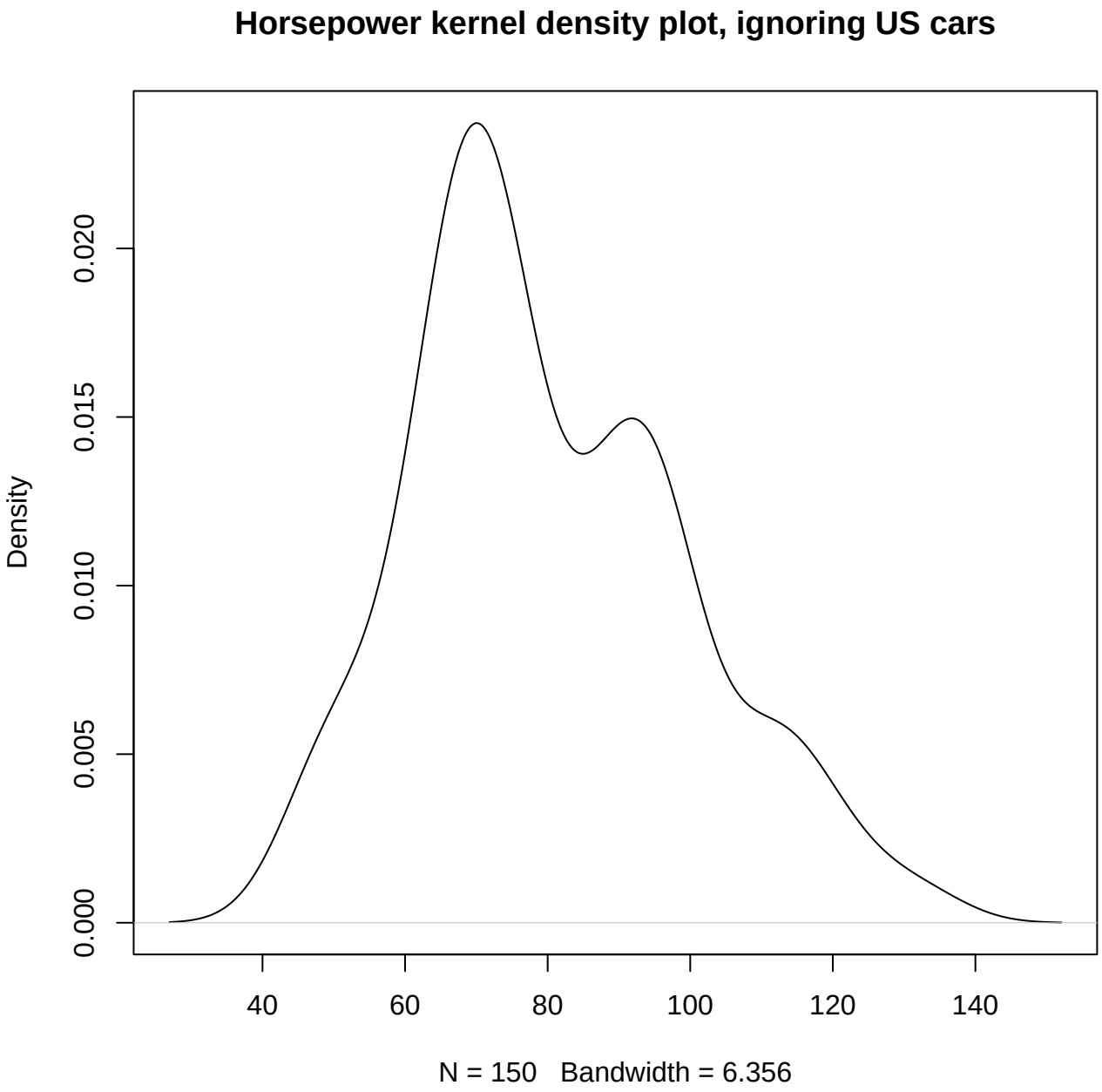

```
#Break 5: Skew Right and mode of 100-200
#Break 10: skew Right and mode of 100-150
```

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.

```
#removing 0s from data and plotting
plot(density(cardata$Horsepower[cardata$Horsepower != 0]), main = "Horsepower kernel density plot")
```

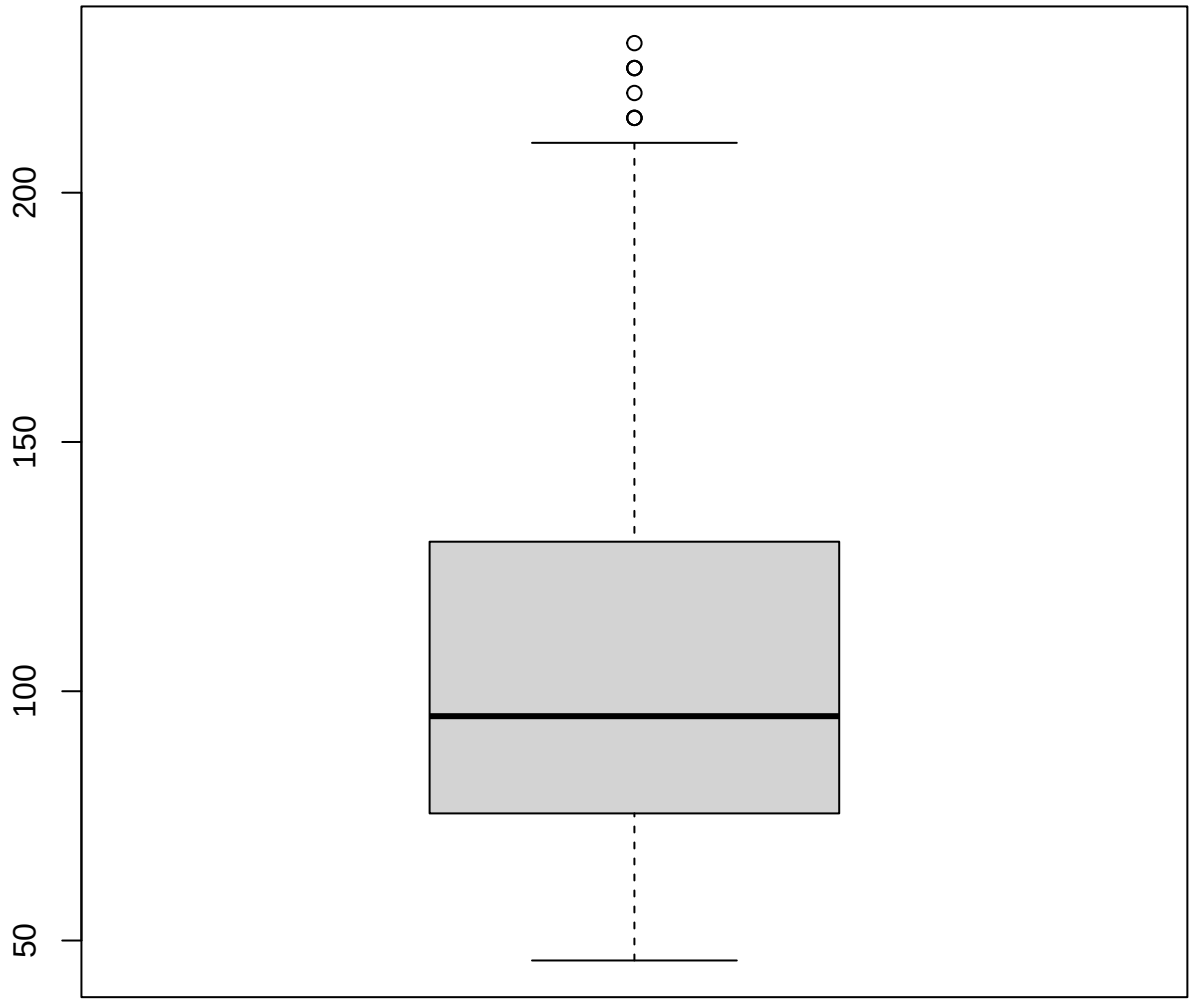



```
plot(density(cardata$Horsepower[cardata$Origin!="US" & cardata$Horsepower != 0]),
     main = "Horsepower kernel density plot, ignoring US cars")
```

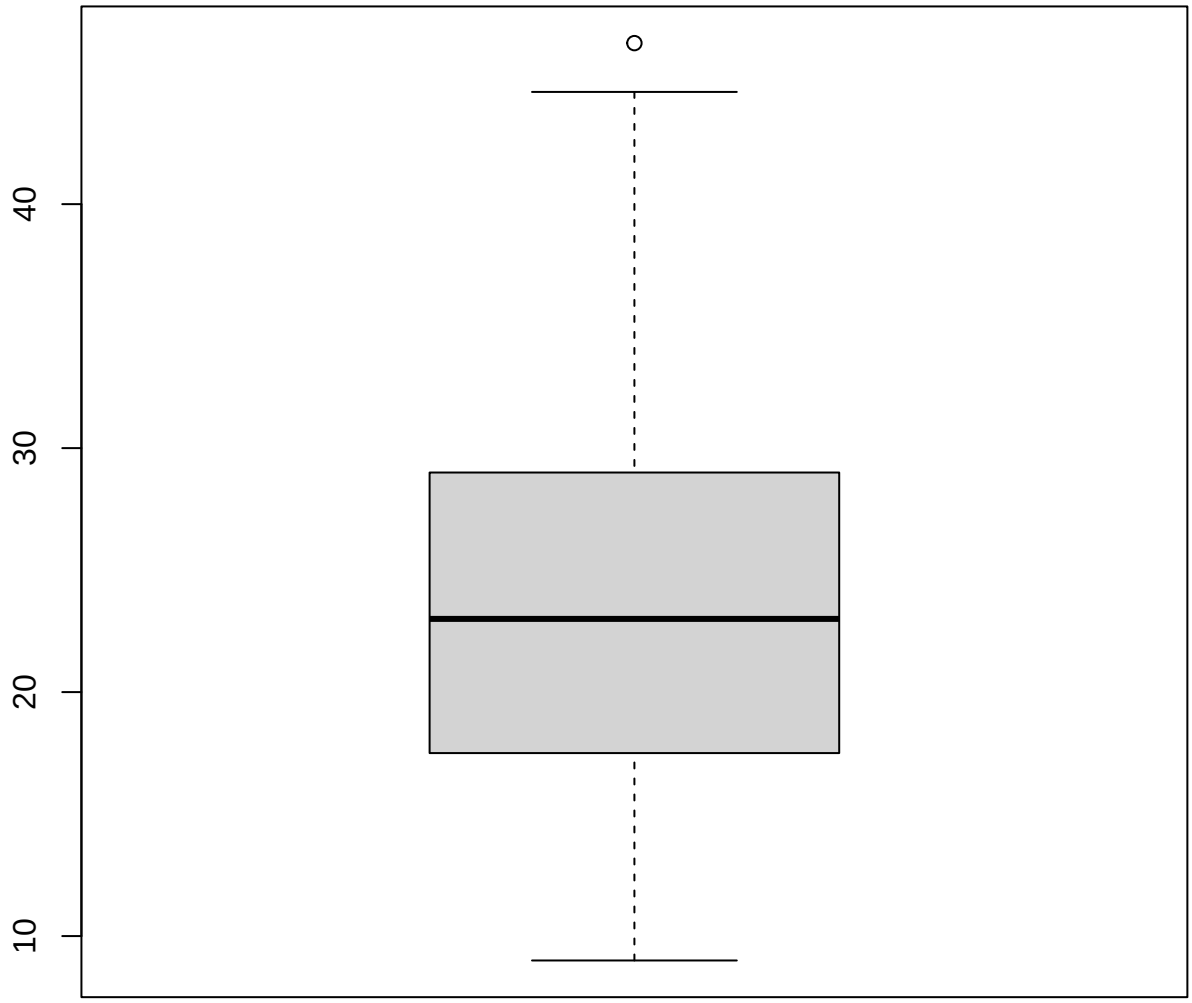


d. Generate a boxplot for the Horsepower variable. Discuss briefly what the boxplot indicates about the horsepower of the cars in the dataset. Generate a boxplot for the MPG variable. Do you notice any suspicious observations or outliers for MPG? Explain.

```
#data in boxplot with 0s ommited
boxplot(cardata$Horsepower[cardata$Horsepower != 0])
```



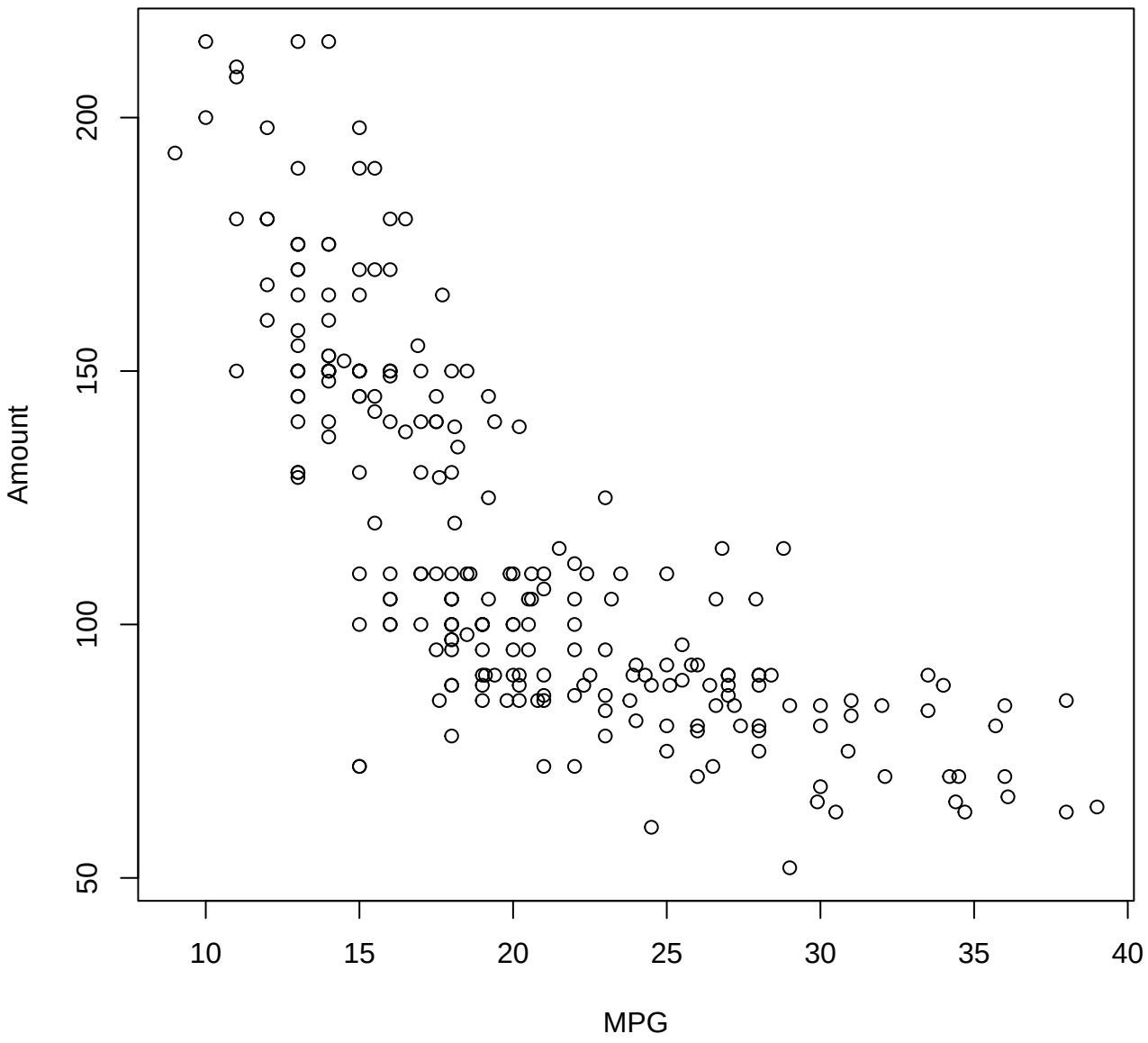

```
boxplot(cardata$MPG[cardata$MPG != 0])
```



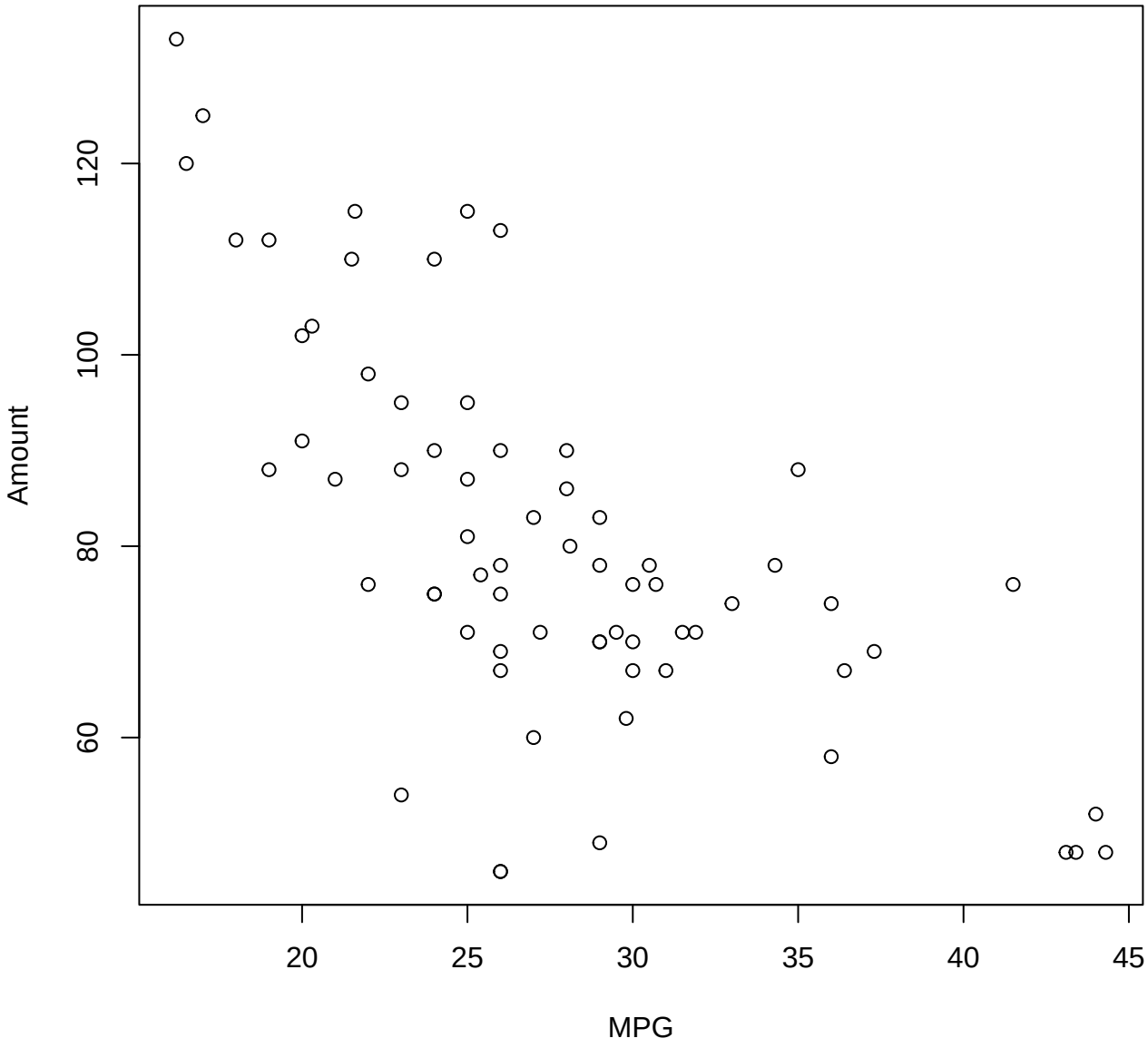

```
#While there are some outliers in the horsepower boxplot, there appears to be
# only one outlier in the MPG
```

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.

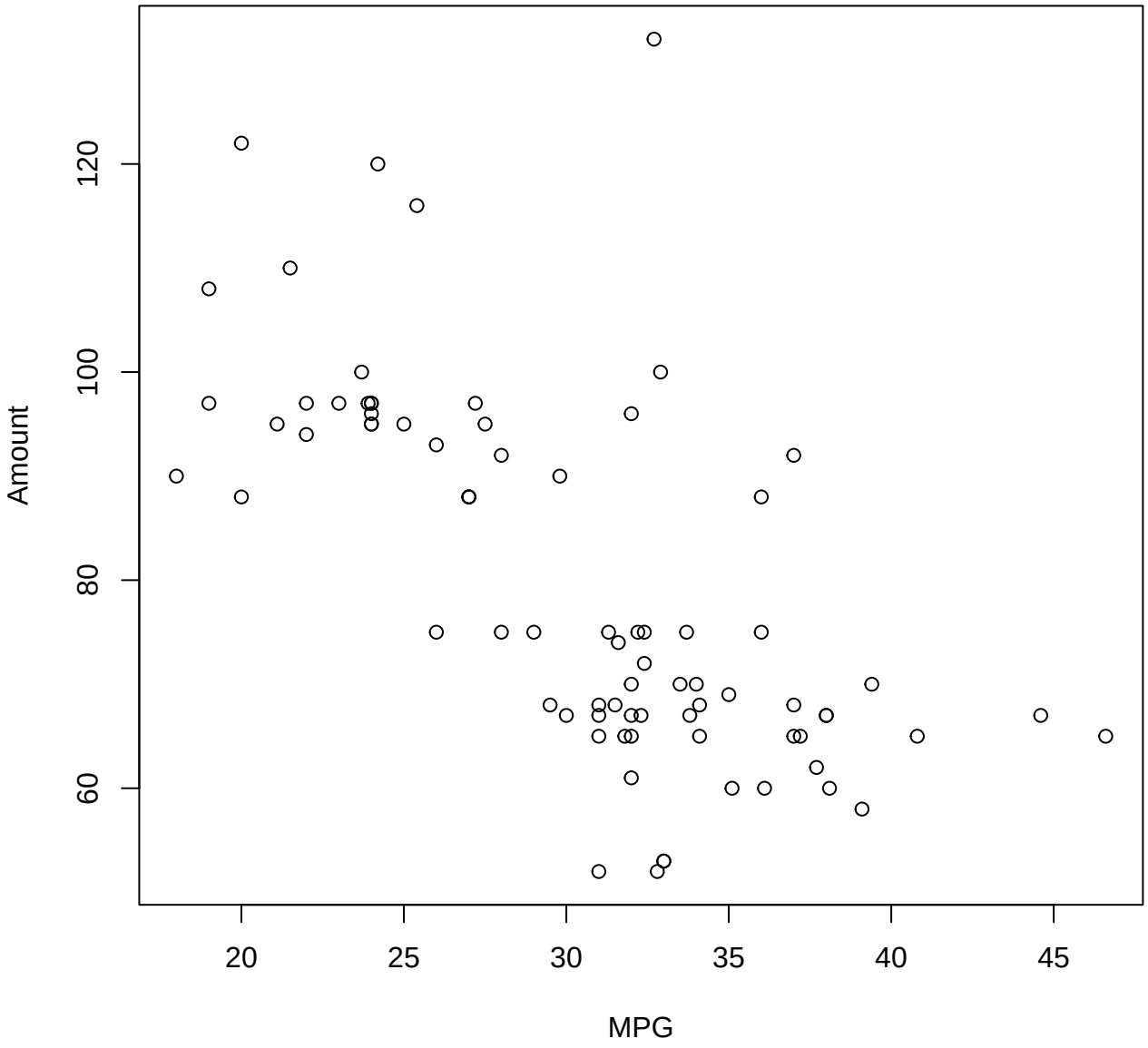
```
#removing outliers and 0s from data
carOut <- cardata[order(-cardata$Horsepower),]
carOut <- carOut[-1:-4,]
invisible(carOut[order(-carOut$MPG),])
carOut <- carOut[-1,]
carOut$MPG[carOut$MPG==0] <- NA
carOut$Horsepower[carOut$Horsepower==0] <- NA
carOut <- carOut[complete.cases(carOut[,c(2,5)]),]
#plotting based on specifications
plot(carOut$MPG[carOut$Origin == "US"],carOut$Horsepower[carOut$Origin == "US"],
     xlab = "MPG", ylab = "Amount", main = "MPG vs Horsepower US cars only")
```




```
plot(carOut$MPG[carOut$Origin == "Europe"],
     carOut$Horsepower[carOut$Origin == "Europe"],
     xlab = "MPG", ylab = "Amount", main = "MPG vs Horsepower European cars only")
```




```
plot(carOut$MPG[carOut$Origin == "Japan"],
     carOut$Horsepower[carOut$Origin == "Japan"],
     xlab = "MPG", ylab = "Amount", main = "MPG vs Horsepower Japanese cars only")
```


MPG vs Horsepower Japanese cars only
Amount
MPG
60
80
100
120
20
25
30
35
40
45

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?

```
#It is a list of the amount, frequency, sex, and year of baby names.
#The list has 5 columns and 1825433 rows
```

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

```
#there are 93889 unique names in the list, the names are by year so there is a
#lot of overlap
```

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?

```
#Male
for (i in 0:4) {
  print((1900 + 25 *  i ))
  print(babynames$name[which(babynames$year ==
                               (1900+25*i) & babynames$sex == "M")[1]])
}
```

```
[1] 1900
[1] "John"
[1] 1925
[1] "Robert"
[1] 1950
[1] "James"
[1] 1975
[1] "Michael"
[1] 2000
[1] "Jacob"
```

```
#Female
for (i in 0:4) {
  print((2010 + i ))
  print(babynames$name[which(babynames$year ==
                               (2010 + i) & babynames$sex == "F")[1]])
}
```

```
[1] 2010
[1] "Isabella"
[1] 2011
[1] "Sophia"
[1] 2012
[1] "Sophia"
[1] 2013
[1] "Sophia"
[1] 2014
[1] "Emma"
```

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

```
temp <- babynames[order(-babynames$prop),]
mnames <- subset(temp, temp$sex == "M")
print("Top 10 most used male names")
```

```
[1] "Top 10 most used male names"
```

```
unique(mnames$name)[1:10]
```

```
 [1] "John"    "William" "Robert"  "James"   "Michael" "Charles" "George"
 [8] "David"   "Richard" "Jason"
```

```
fnames <- subset(temp, temp$sex == "F")
print("Top 10 most used female names")
```

```
[1] "Top 10 most used female names"
```

```
unique(fnames$name)[1:10]
```

```
 [1] "Mary"     "Linda"    "Jennifer" "Shirley"  "Barbara"  "Lisa"
 [7] "Betty"    "Dorothy"  "Patricia" "Helen"
```
