

# STAT 2200: Problem Set 3

Jack Ambery

Due: Monday, 2/26 at the beginning of class

- You may discuss this assignment with other students in the class, but you may not sit down and type it up with them or show them your code. You also may not discuss the assignment with anyone who isn't in our class, nor may you look up anything online.
- You must type up your homework using R Markdown. I want to see all of your code and output, and any answers you provide that aren't code must be typed above the respective R code chunk.
- Make sure none of your code runs off the page, otherwise you will lose points.
- You must print and turn in a PDF of your homework. I won't accept anything else (e.g., a Word document). All pages must be stapled together.
- This assignment is worth 100 points.

In problems 1 and 2 you will work with datasets from the website for the Lock, Lock, Lock, Lock, and Lock textbook *Statistics: Unlocking the Power of Data* (3rd ed.). Go to the following URL to access the two datasets: <https://www.lock5stat.com/datapage3e.html>.

1. Download the CSV file that contains the *Cars2020* dataset and read it into R. Use `str()` to examine the structure of the data set. Then:

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.3.2
```

```
Cars2020 <- read_csv("../datasets/Cars2020.csv")
```

```
## Rows: 110 Columns: 21
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (5): Make, Model, Type, Drive, Size
```

```
## dbl (16): LowPrice, HighPrice, CityMPG, HwyMPG, Seating, Acc030, Acc060, Qtr...
```

```
##
```

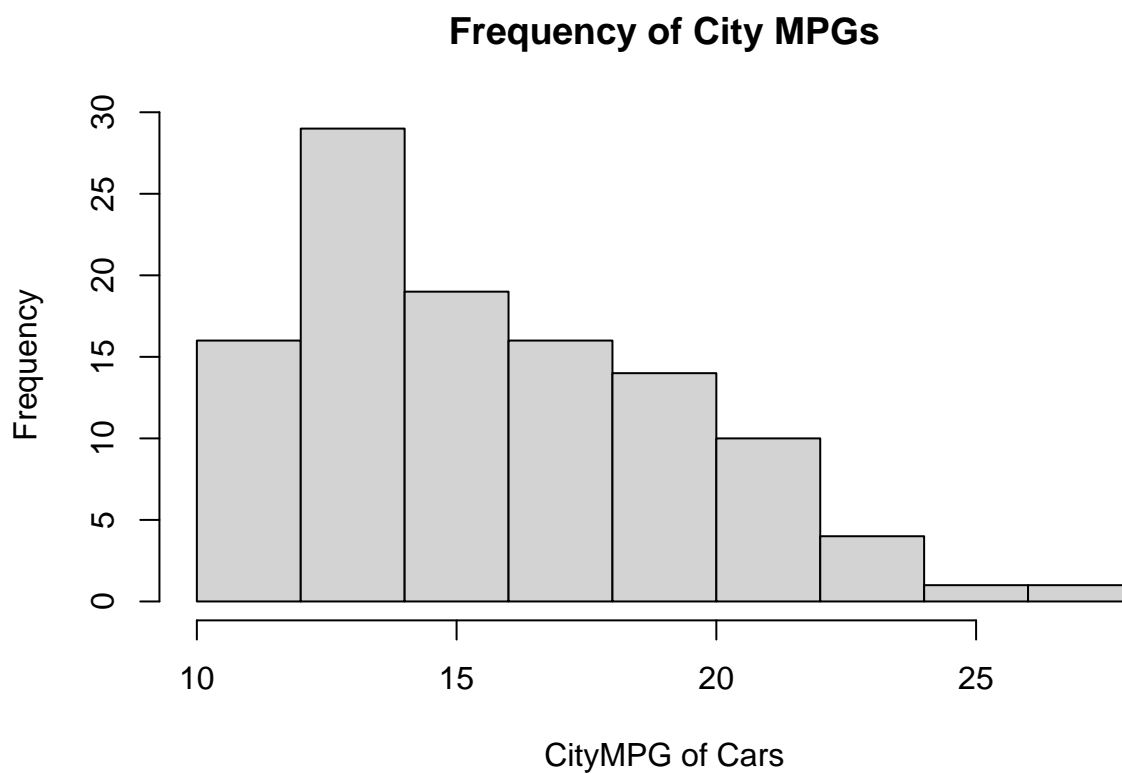
```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

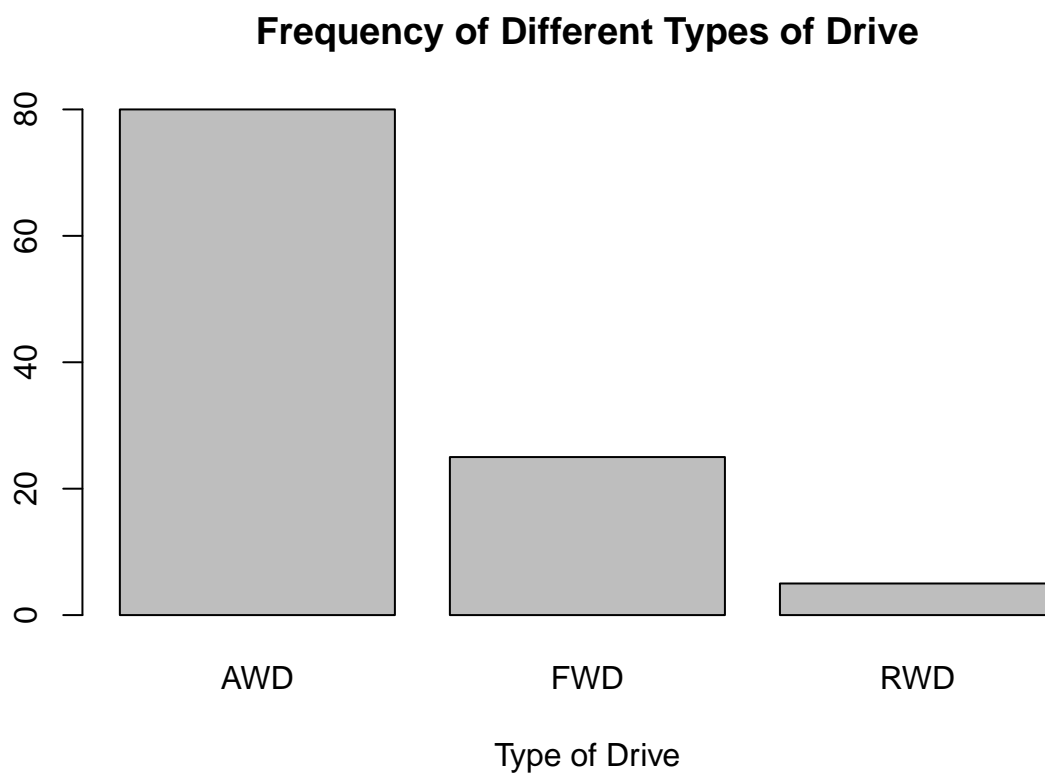
```
# str() prints out a page worth of text
# str(Cars2020)
```

- a) Make four plots – one for each of the following variables: CityMPG, Drive, LowPrice, and Type. Make sure you're using an appropriate type of plot for each. Note: please don't make a pie chart for any categorical variable. You can earn a bonus point if you add appropriate labels to each plot.

```
# CityMPG
hist(Cars2020$CityMPG, xlab = "CityMPG of Cars",
     main = "Frequency of City MPGs")
```

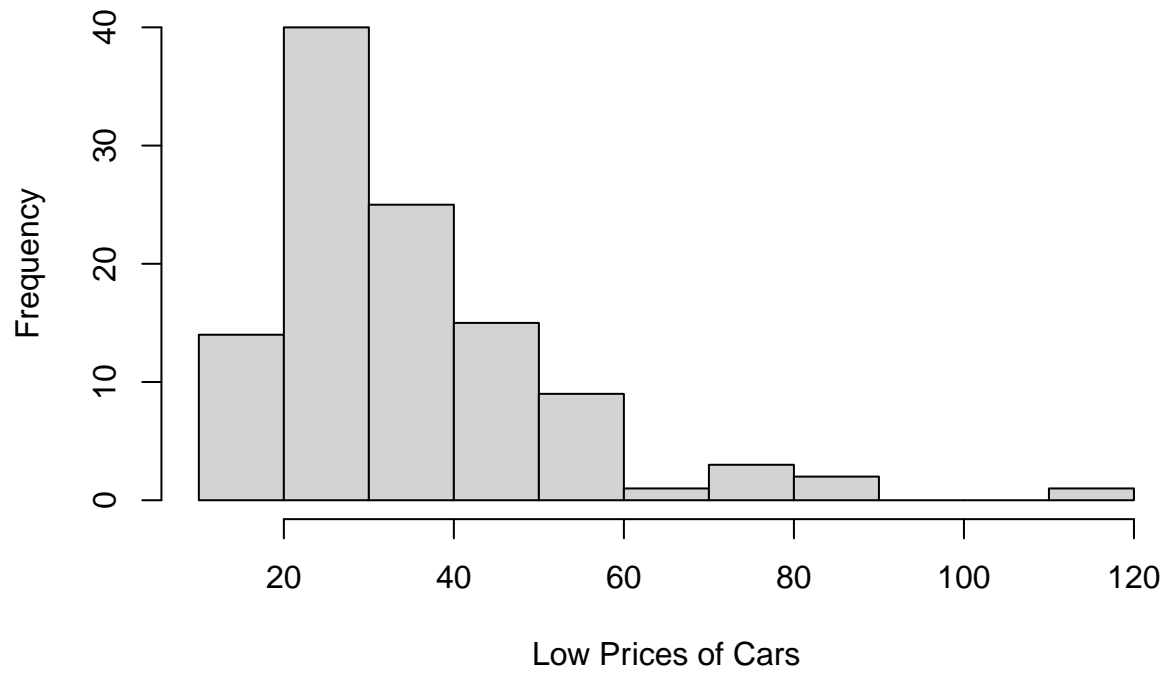


```
# Drive
drives <- table(Cars2020$Drive)
barplot(drives, xlab = "Type of Drive",
       main = "Frequency of Different Types of Drive")
```



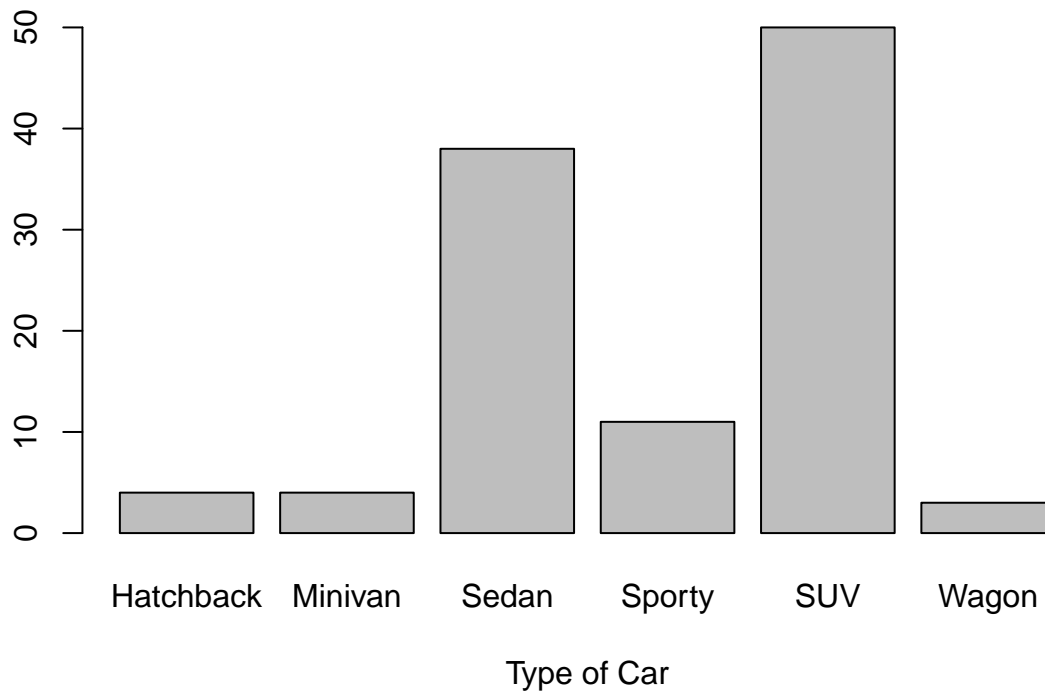
```
# LowPrice  
hist(Cars2020$LowPrice, xlab = "Low Prices of Cars",  
      main = "Frequency of Low Prices")
```

## Frequency of Low Prices



```
# Type
types <- table(Cars2020$Type)
barplot(types, xlab = "Type of Car",
        main = "Frequency of Different Car Types")
```

### Frequency of Different Car Types



b) For the CityMPG variable, calculate the mean, median, standard deviation, IQR, range, 5<sup>th</sup> percentile, and 95<sup>th</sup> percentile. Then type these values outside of the R code chunk.

```
mpgs <- Cars2020$CityMPG  
summary(mpgs)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
##   10.00   13.25   15.50   16.19   19.00   28.00
```

```
sd(mpgs)
```

```
## [1] 3.74042
```

```
iqr <- 19.0 - 13.25  
iqr
```

```
## [1] 5.75
```

```
range <- 28 - 10  
range
```

```
## [1] 18
```

```
quantile(mpgs, 0.05)
```

```
## 5%
## 11
```

```
quantile(mpgs, 0.95)
```

```
## 95%
## 22.55
```

**For the CityMPG variable**

Mean: 16.19

Median: 15.5

Standard Deviation: 3.74

IQR: 5.75

Range: 18

5<sup>th</sup> percentile: 11

95<sup>th</sup> percentile: 22.55

- c) For the Type variable, calculate (i) the frequency of cars that fall into each category, and (ii) the percentage of cars that fall into each category. Then type these values outside of the R code chunk.

```
anyNA(Cars2020$Type) #FALSE
```

```
## [1] FALSE
```

```
table(Cars2020$Type)
```

```
##
## Hatchback   Minivan   Sedan   Sporty   SUV   Wagon
##           4         4       38       11     50      3
```

```
round(((table(Cars2020$Type)/nrow(Cars2020)) * 100), 2) #percentages
```

```
##
## Hatchback   Minivan   Sedan   Sporty   SUV   Wagon
##       3.64      3.64    34.55    10.00    45.45    2.73
```

**Frequencies of Car Types**

Hatchback:

- Frequency = 4
- Percentage = 3.64%

Minivan:

- Frequency = 4

- Percentage = 3.64%

Sedan:

- Frequency = 38
- Percentage = 34.55%

Sporty:

- Frequency = 11
- Percentage = 10%

SUV:

- Frequency = 50
- Percentage = 45.45%

Wagon:

- Frequency = 3
- Percentage = 2.73%

2. Download the CSV file that contains the *Hurricanes2018* dataset and read it into R. Run `str()` to examine the structure of the dataset. Next, make a timeplot that plots the number of hurricanes over the past century or so. Make sure the timeplot has points at each year and lines connecting the points. You can earn a bonus point if you add a title and appropriate labels to both axes.

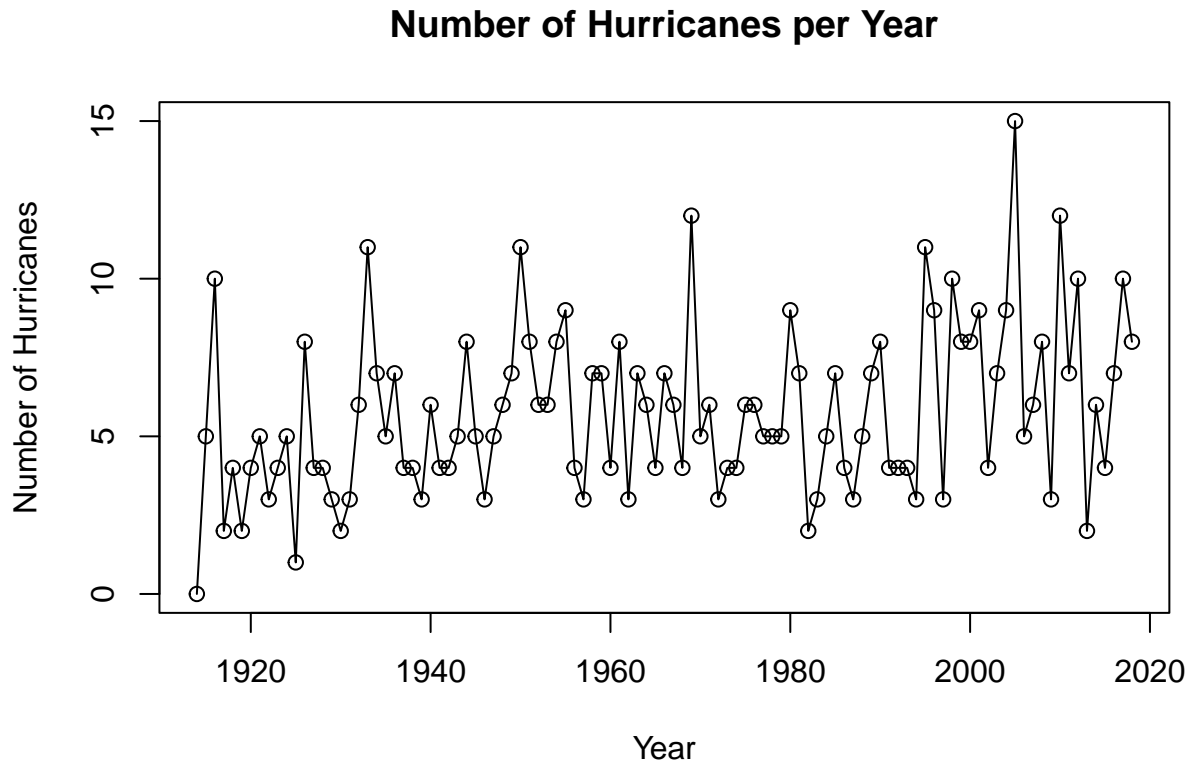
```
library(readr)
Hurricanes2018 <- read_csv("../datasets/Hurricanes2018.csv")

## Rows: 105 Columns: 2
## -- Column specification -----
## Delimiter: ","
## dbl (2): Year, Hurricanes
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
str(Hurricanes2018)

## spc_tbl_ [105 x 2] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ Year      : num [1:105] 1914 1915 1916 1917 1918 ...
## $ Hurricanes: num [1:105] 0 5 10 2 4 2 4 5 3 4 ...
## - attr(*, "spec")=
## .. cols(
## ..   Year = col_double(),
## ..   Hurricanes = col_double()
```

```
## .. )
## - attr(*, "problems")=<externalptr>

plot(Hurricanes2018$Year, Hurricanes2018$Hurricanes, xlab = "Year", ylab = "Number of Hu
lines(Hurricanes2018$Year, Hurricanes2018$Hurricanes)
```



3. Download the *HollywoodMovies* dataset from Blackboard. As you look at the dataset, you should notice missing values. Read the file into R and run the `unique()` function on the variable `Genre`. This function outputs all of the unique/different entries that exist in an object. Do any entries stand out as possibly having been read improperly? If so, reimport your data properly. Next, use an appropriate type of plot to plot the domestic revenue of the movies (`DomesticGross`) based on genre (`Genre`), but only considering the following three genres: action, drama, and romance.

4. Download the *mlb2011* dataset from Blackboard. This file was originally found on BaseballGuru.com (located at <https://BaseballGuru.com>) and contains data for Major League Baseball (MLB) pitchers in 2011. In this problem you are going to look at the relationship between the number of wins (variable “W”) and the number of strikeouts (variable “SOA”) for pitchers with at least 10 games started (variable “GS”) across the two different leagues (variable “LG”). The two leagues are the National League (NL) and the American League (AL).



After reading the data into R, you first want to separate pitchers into two groups: those who had at least 10 starts and played in the National League, and those who had at least 10 starts and played in the American League. Then create a scatterplot with the number of wins on the  $y$ -axis and the number of strikeouts on the  $x$ -axis. You should appropriately distinguish between the National League and American League pitchers using points with different colors. Next, add two regression lines (aka lines of best fit) to the plot. The first line, which has a  $y$ -intercept of 0.945 and a slope of 0.073, corresponds to American League pitchers. The second line, which has a  $y$ -intercept of 1.966 and a slope of 0.059, corresponds to National League pitchers. Color the lines appropriately. You can earn a bonus point if you correctly add axis labels and a legend to your plot.