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:

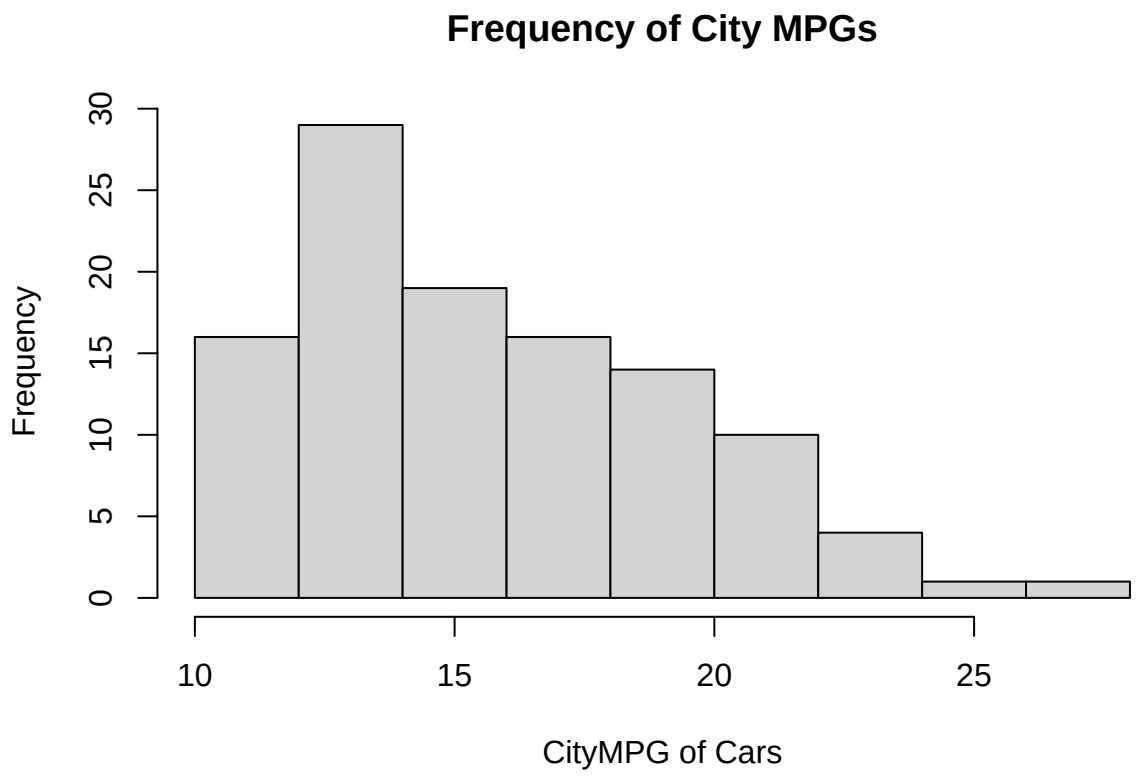
```
Cars2020 <- read.csv("../datasets/Cars2020.csv")
str(Cars2020)
```

```
## 'data.frame':    110 obs. of  21 variables:
##  $ Make     : chr  "Acura" "Acura" "Audi" "Audi" ...
##  $ Model    : chr  "MDX" "RLX" "A3" "A4" ...
##  $ Type     : chr  "SUV" "Sedan" "Sedan" "Sporty" ...
##  $ LowPrice : num  44.4 54.9 33.3 37.4 54.9 ...
##  $ HighPrice: num  60.1 61 43 45.7 73.9 ...
##  $ CityMPG  : int  14 15 18 18 17 20 15 18 19 16 ...
##  $ HwyMPG   : int  31 36 40 40 39 27 33 35 44 40 ...
##  $ Seating  : int  7 5 5 5 5 5 5 4 5 5 ...
##  $ Drive    : chr  "AWD" "AWD" "AWD" "AWD" ...
```

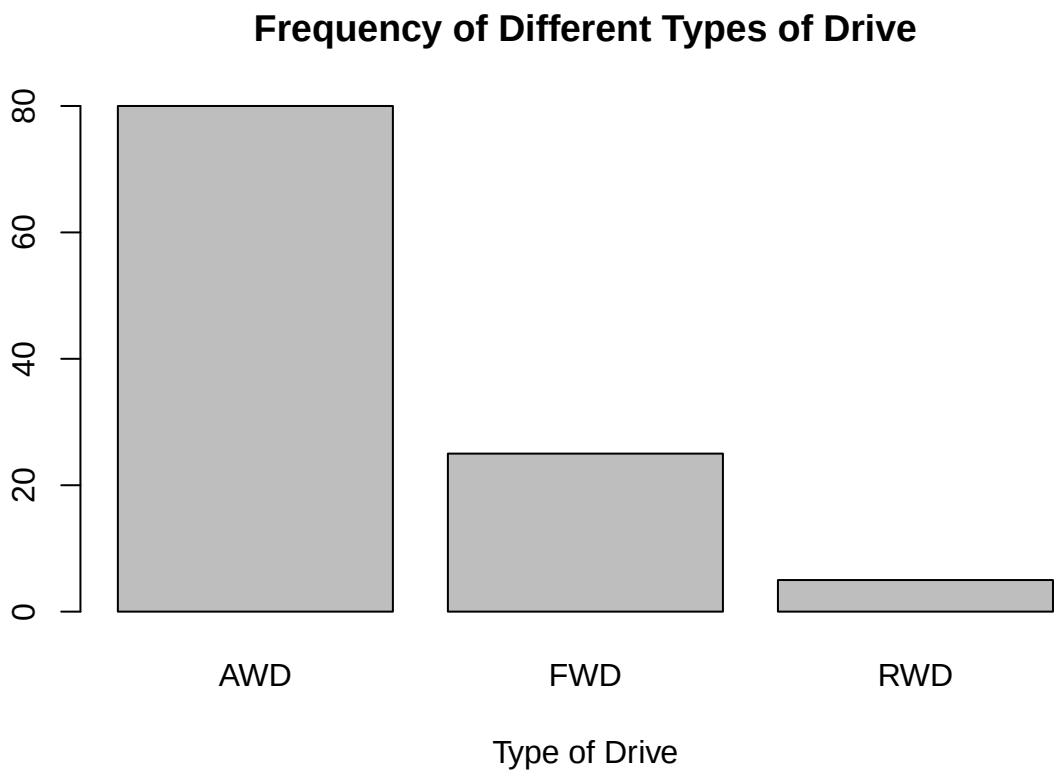
```
##  $ Acc030   : num  2.8 2.7 3.2 2.7 2.8 2.4 3.2 2.5 2.6 2.9 ...
##  $ Acc060   : num  6.8 6.5 8.3 6.3 6.8 6.1 7.8 6.7 6.4 7.2 ...
##  $ QtrMile  : num  15.3 15 16.4 14.9 15.3 14.5 16.1 14.8 14.8 15.5 ...
##  $ Braking  : int  135 128 124 135 129 133 126 113 129 130 ...
##  $ FuelCap  : num  19.5 18.5 13.2 15.3 19.3 21.7 15.9 14.5 15.6 17.9 ...
##  $ Length   : int  196 198 175 186 195 209 177 165 186 195 ...
##  $ Width    : int  77 74 70 73 74 77 73 72 72 74 ...
##  $ Height   : int  67 58 56 56 57 59 63 53 57 58 ...
##  $ Wheelbase: int  111 112 104 111 115 123 106 99 112 117 ...
##  $ UTurn    : int  40 40 37 40 38 43 40 36 41 42 ...
##  $ Weight   : int  4200 3930 3135 3630 4015 4810 3880 3140 3640 3950 ...
##  $ Size     : chr  "Midsized" "Midsized" "Small" "Small" ...
```

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.

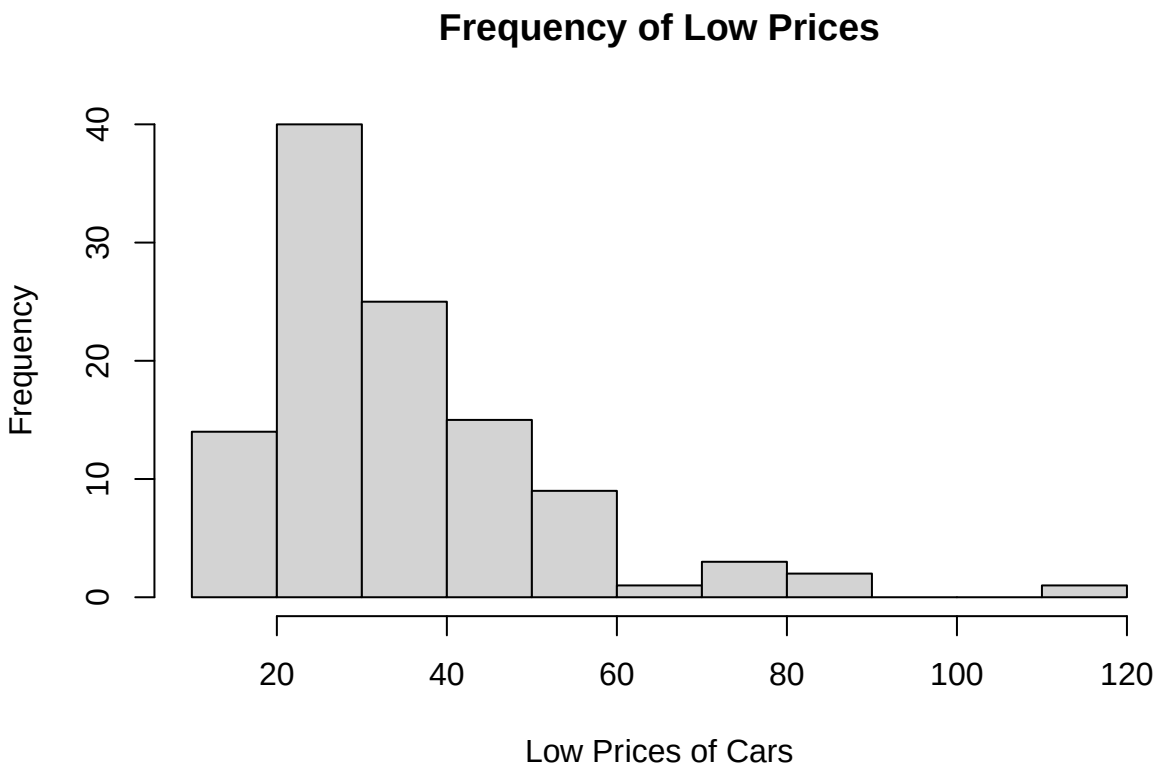
```r
# CityMPG
hist(Cars2020$CityMPG, xlab = "CityMPG of Cars",
     main = "Frequency of City MPGs")
```



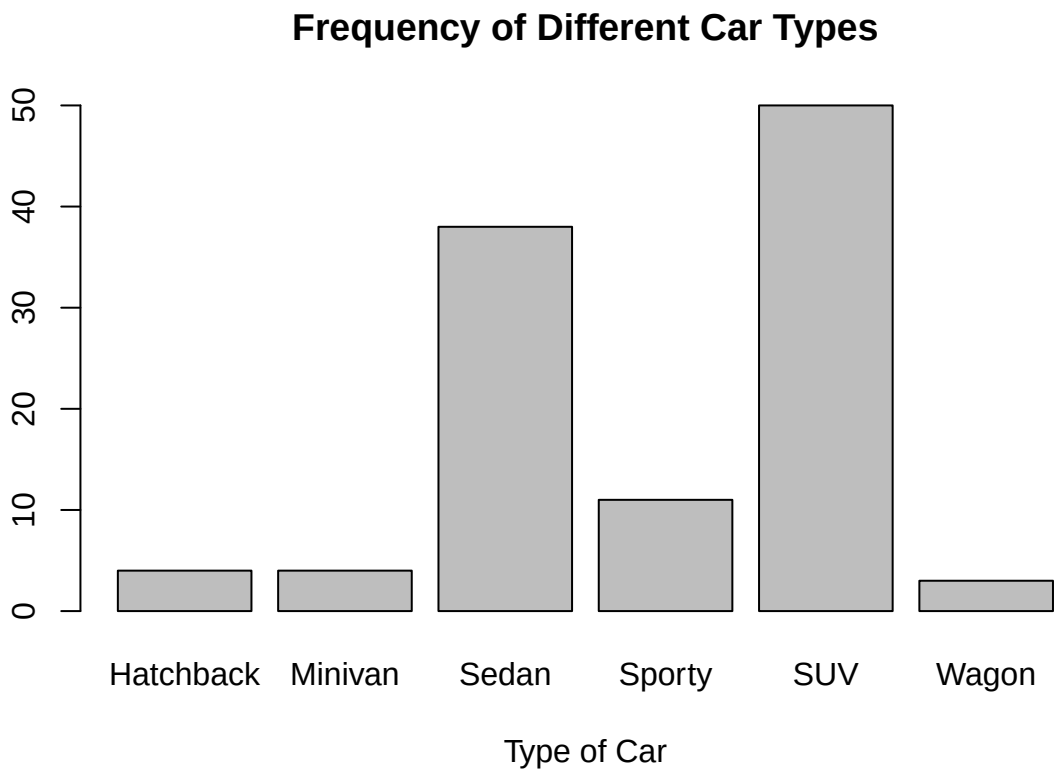
```r
# Drive
drives <- table(Cars2020$Drive)
barplot(drives, xlab = "Type of Drive",
        main = "Frequency of Different Types of Drive")
```




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




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



b) For the CityMPG variable, calculate the mean, median, standard deviation, IQR, range, 5^{th} percentile, and 95^{th} 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^{th} percentile: 11

95^{th} 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:

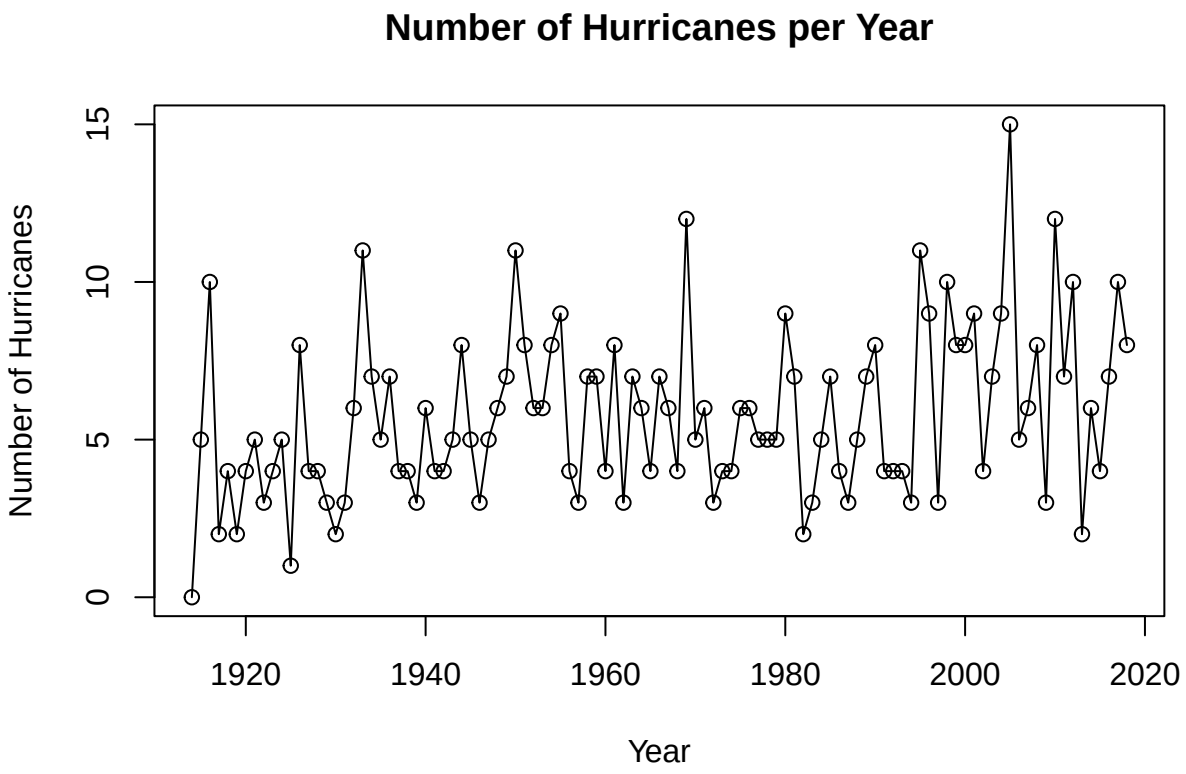
- Frequency = 3
- Percentage = 2.73%

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.

```
Hurricanes2018 <- read.csv("../datasets/Hurricanes2018.csv")
str(Hurricanes2018)
```

```
## 'data.frame':    105 obs. of  2 variables:
##  $ Year      : int  1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 ...
##  $ Hurricanes: int  0 5 10 2 4 2 4 5 3 4 ...
```

```
plot(Hurricanes2018$Year, Hurricanes2018$Hurricanes,
     xlab = "Year",
     ylab = "Number of Hurricanes",
     main = "Number of Hurricanes per Year")
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.

```
HollywoodMovies <- read.csv("../datasets/HollywoodMovies.csv",
                            na.strings = "")
```

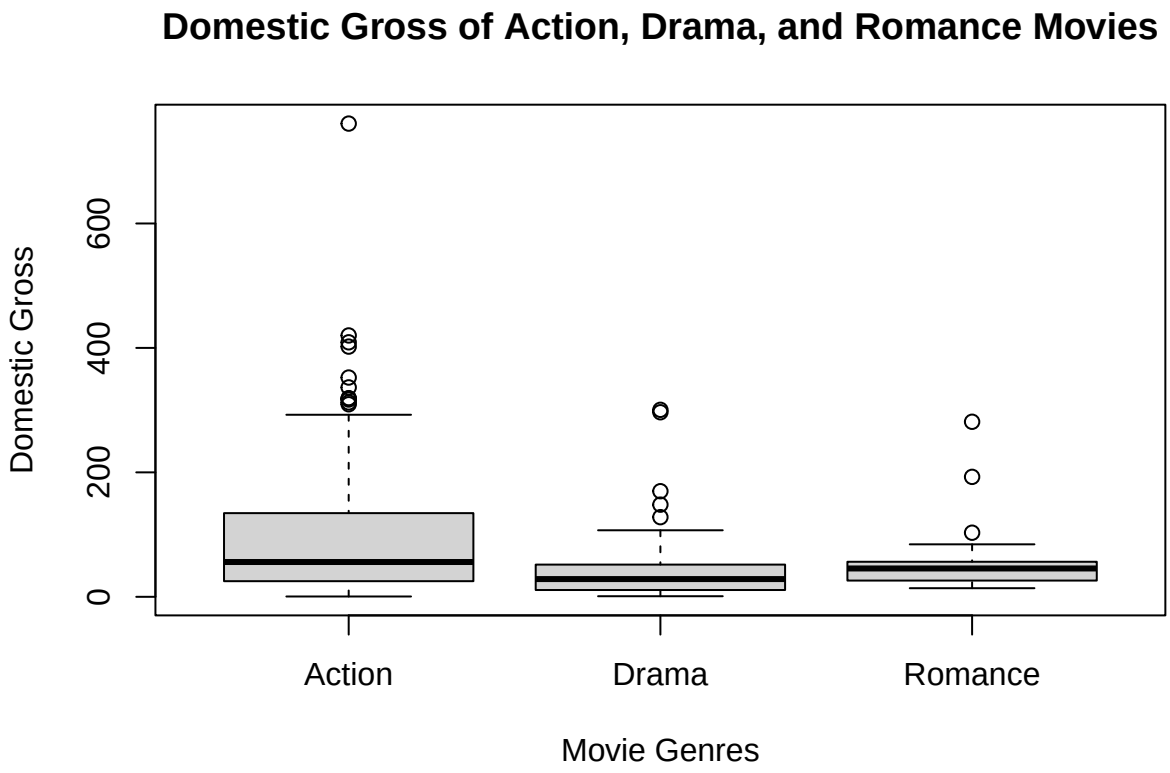
```
unique(HollywoodMovies$Genre)
```

```
##  [1] "Action"      "Animation"   "Adventure"   "Thriller"    "Comedy"
##  [6] "Musical"     "Drama"       "Biography"   "Horror"      "Romance"
## [11] "Fantasy"     "Documentary" "Crime"       NA            "Mystery"
```

```
genre <- HollywoodMovies$Genre
dgross <- HollywoodMovies$DomesticGross

boxplot(dgross[genre == "Action"],
        dgross[genre == "Drama"],
```

```
        dgross[genre == "Romance"],
        ylab = "Domestic Gross",
        xlab = "Movie Genres",
        main = "Domestic Gross of Action, Drama, and Romance Movies",
        names = c("Action", "Drama", "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).

```
mlb2011 <- read.csv("../datasets/mlb2011.csv")
```

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.

```
league <- mlb2011$LG
gamesStarted <- mlb2011$GS
NLPitchers <- mlb2011[league == 'NL' & gamesStarted >= 10,]
ALPitchers <- mlb2011[league == 'AL' & gamesStarted >= 10,]

plot(NLPitchers$SOA, NLPitchers$W,
     col = 'red',
     main = 'Number of Strikeouts vs. Wins for
     Pitchers in the MLB National and American Leagues in 2011',
     xlab = 'Number of Strikeouts',
     ylab = 'Number of Wins')
points(ALPitchers$SOA, ALPitchers$W, col='blue')
abline(a = 1.966, b = 0.059, col='red')
abline(a = 0.945, b = 0.073, col='blue')
```