Activity 1: Salvin Chowdhury

Due Date: Febraury 5, 2025

The file “Activity1.csv”’ (found in Canvas) contains some data for the 574 players that played at least one game in the NBA during the 2023-2024 season (collected from basketball-reference.com).

If you make sure that the file is in the same folder as this markdown file, you should be able to use it with no problems as long as you run the chunk of code below. I’ve just named it “nba” for ease of use, but you can change that if you want to. I’ve also included some code to take a random sample of size 50, and called it “mysample” (again feel free to change the name if you want). You will need to use the sample in Problem 2.

```r
# converting .csv file into a data frame
nba <- data.frame(read.csv("Activity1.csv"))

# setting the seed
set.seed(77)

# random sampling
mysample <- nba[sample(nrow(nba), 50), ]
```

Problem 1: Describing/Summarizing One Categorical Variable

Part (a)

- Use the table() function to create what is essentially a frequency distribution for the position (Pos) variable.
- Assign this table the name “positions” and then view the table.

```r
# picking out the positions column
Pos = nba$Pos

# converting the column into a table
positions = table(Pos)

# printing out the table
positions
```

```
## Pos
##   C  PF  PG  SF  SG
##  94 115 107 123 135
```

Part (b)

Use your result from part (a) to determine how many and what percent of NBA players are guards (both PG and SG are considered guards). Simply type your answer below, after “ANSWER:” but before the second “**“. If you use R to help in your calculations, include the code you used in the following chunk.

```r
# This chunk is here for any calculations if needed in part (b).

# checking the type of data structure of positions
class(positions)
```

```
## [1] "table"
```

```
# converting from table to a dataframe
positions_df <- as.data.frame(positions)
positions_df
```

```
##   Pos Freq
## 1   C   94
## 2  PF  115
## 3  PG  107
## 4  SF  123
## 5  SG  135
```

```
# grabbing the PG & SG values from the dataframe
pg_guards = positions_df[positions_df$Pos == "PG", "Freq"]
sg_guards = positions_df[positions_df$Pos == "SG", "Freq"]

# finding the total number of guards
total_guards = pg_guards + sg_guards
print(paste("Total Guards: ", total_guards))
```

```
## [1] "Total Guards:  242"
```

```
# finding the total number of NBA players
total_players = sum(positions_df$Freq)
print(paste("Total Players: ", total_players))
```

```
## [1] "Total Players:  574"
```

```
# finding the percentage
percentage = (total_guards / total_players) * 100
print(paste("Percentage: ", percentage))
```

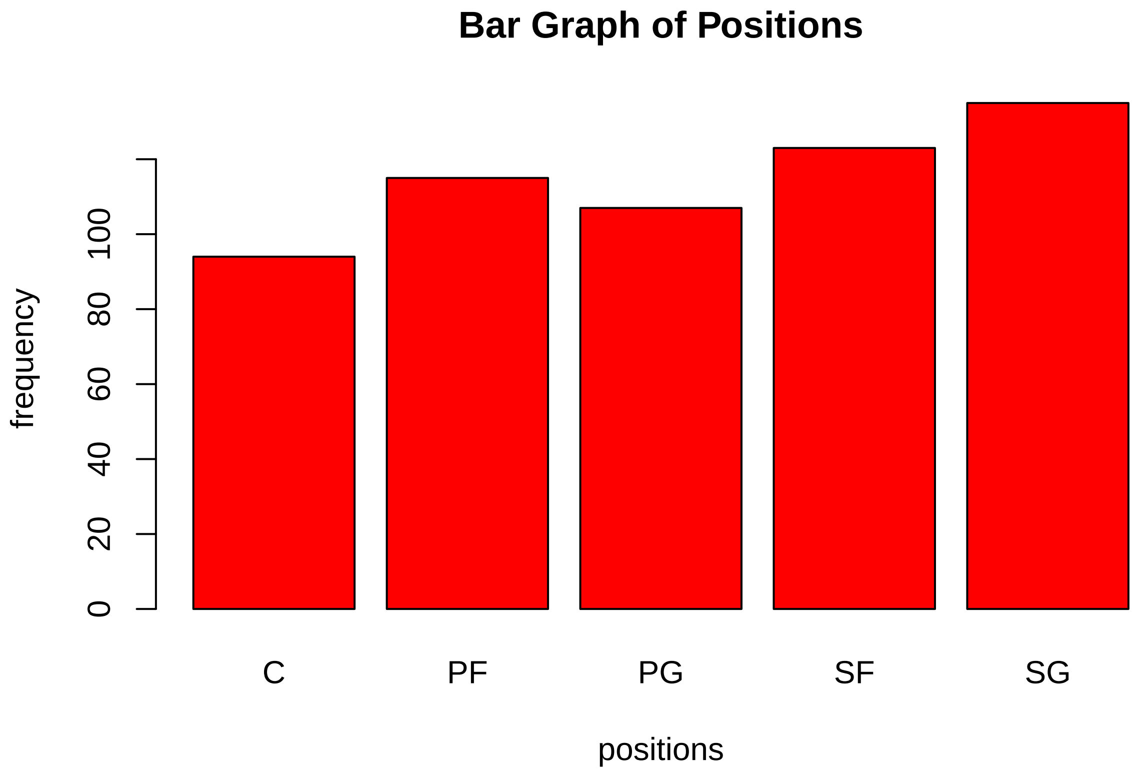
```
## [1] "Percentage:  42.1602787456446"
```

ANSWER: - The number of players that are guards are 242 - The percentage of players that are guards are 42.2%

Part (c)

Use the barplot() function to create a bar graph summarizing the position variable. Include a title and labels for each axis using the arguments like: main = "title", xlab = "x label", and ylab = "y label"

```
# creating a bar plot
barplot(positions, col="red",
        main = "Bar Graph of Positions",
        xlab = "positions", ylab = "frequency",
        border = "black")
```



Problem 2: Summarizing One Quantative Variable

Part (a)

Calculate the mean and standard deviation of the points per game (PTS) variable for the population. Do the same for the sample. Use the code chunk below for calculations, and type your answers as one or more sentences in the space provided, as before.

```
# picking out the PTS data
points_per_game = nba$PTS

# figuring out the class
class(points_per_game)
```

```
## [1] "numeric"
```

```
# count of values of points
cat("Number of Values of PTS: ", length(points_per_game), "\n")
```

```
## Number of Values of PTS:  574
```

```
# count of values of positions
cat("Number of Positions: ", length(Pos), "\n")
```

```
## Number of Positions:  574
```

```
# calculating the mean
cat("The PTS Mean: ", mean(points_per_game), "\n")
```

```
## The PTS Mean:  8.410627
```

```r
# calculating the standard deviation
cat("The PTS Standard Deviation: ", sd(points_per_game), "\n")
```

```
## The PTS Standard Deviation:  6.782938
```

```r
# checking the type of the sample
class(mysample)
```

```
## [1] "data.frame"
```

```r
# finding the mean from the sample
nba_pts_mean <- mean(mysample$PTS, na.rm = TRUE)

# finding the standard deviation from the sample
nba_pts_sd <- sd(mysample$PTS, na.rm = TRUE)

# printing out mean and standard deviation
cat("The Sample Mean: ", nba_pts_mean, "\n")
```

```
## The Sample Mean:  10.196
```

```r
cat("The Sample SD: ", nba_pts_sd, "\n")
```

```
## The Sample SD:  6.598391
```

ANSWER: - The PTS Mean is 8.41 - The PTS Standard Deviation is 6.78 - The Sample Mean is 10.196 - The Sample Standard Deviation is 6.6

Part (b)

Construct the "five number summary" for the points per game for the population, using the fivenum() function. Do the same for the sample. Give the interquartile range for each. Use the code chunk and answer space as before.

```r
# storing the summaries
population_summary <- fivenum(points_per_game)
sample_summary <- fivenum(mysample$PTS)

# printing out the five number summary for the population
cat("The Five Number Summary (Population): ", population_summary, "\n")
```

```
## The Five Number Summary (Population):  0 3.4 6.4 11.7 34.7
```

```r
# printing out the five number summary for the sample
cat("The Five Number Summary (Sample): ", sample_summary, "\n")
```

```
## The Five Number Summary (Sample):  1.5 5 8.4 16 25.7
```

```r
# finding the upper and lower quartile in population summary
population_lower_quartile <- population_summary[2]
population_upper_quartile <- population_summary[4]

# finding the upper and lower quartile in sample summary
sample_lower_quartile <- sample_summary[2]
sample_upper_quartile <- sample_summary[4]

# finding the interquartile range in population
population_interquartile = population_upper_quartile - population_lower_quartile
cat("Interquartile Range of Population: ", population_interquartile, "\n")
```

```
## Interquartile Range of Population:  8.3
```

```
# finding the interquartile range in sample
sample_interquartile = sample_upper_quartile - sample_lower_quartile
cat("Interquartile Range of Sample: ", sample_interquartile, "\n")
```

```
## Interquartile Range of Sample:  11
```

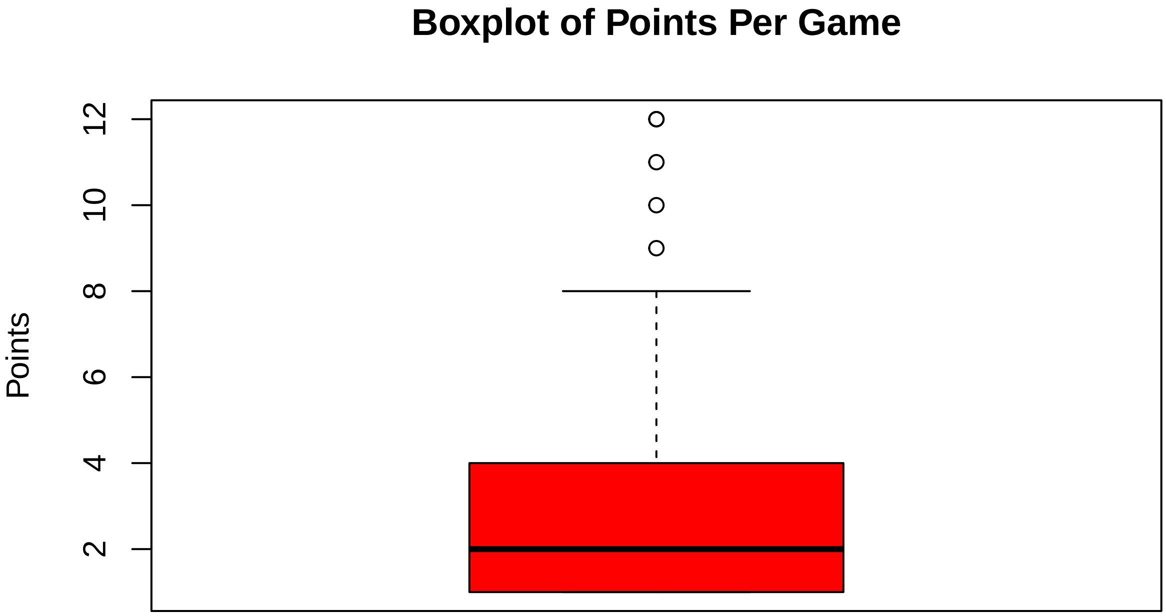
ANSWER: - Interquartile range of Population: 8.3 - Interquartile range of Sample: 11

Part (c)

Use the boxplot() function to construct a boxplot of the points per game variable for the population. Include a title and labels for the appropriate axis (the arguments should be similar to the barplot from earlier).

```
# creating a points table
points_table <- table(points_per_game)

# creating the box plot
boxplot(points_table, main = "Boxplot of Points Per Game",
        ylab = "Points", col="red")
```



Part (f)

Let's use the boxplot.stats() function, to get some additional information. "stats" in the output of this function gives the five values of each part of the box/whiskers: the "bottom whisker" value, Q1, med, Q2, and the "top whisker" value. The "out" in the output is where you can find each of the outliers.

```r
# retrieving the box plot statistics
box_stats <- boxplot.stats(points_table)

# printing out the statistics
print(box_stats)
```

```
## $stats
## points_per_game
##  0.3 21.4 19.6  2.7  4.4
##    1    1    2    4    8
##
## $n
## [1] 202
##
## $conf
## [1] 1.666495 2.333505
##
## $out
## points_per_game
##   0   2   4 4.4   5
##  10  11  12  12   9
```

Now we can answer the following questions (type your answers in the appropriate spaces, as before):

How many outliers are there?

ANSWER: $out points_per_game 0 2 4 4.4 5 10 11 12 12 9

There are 10 outliers in total

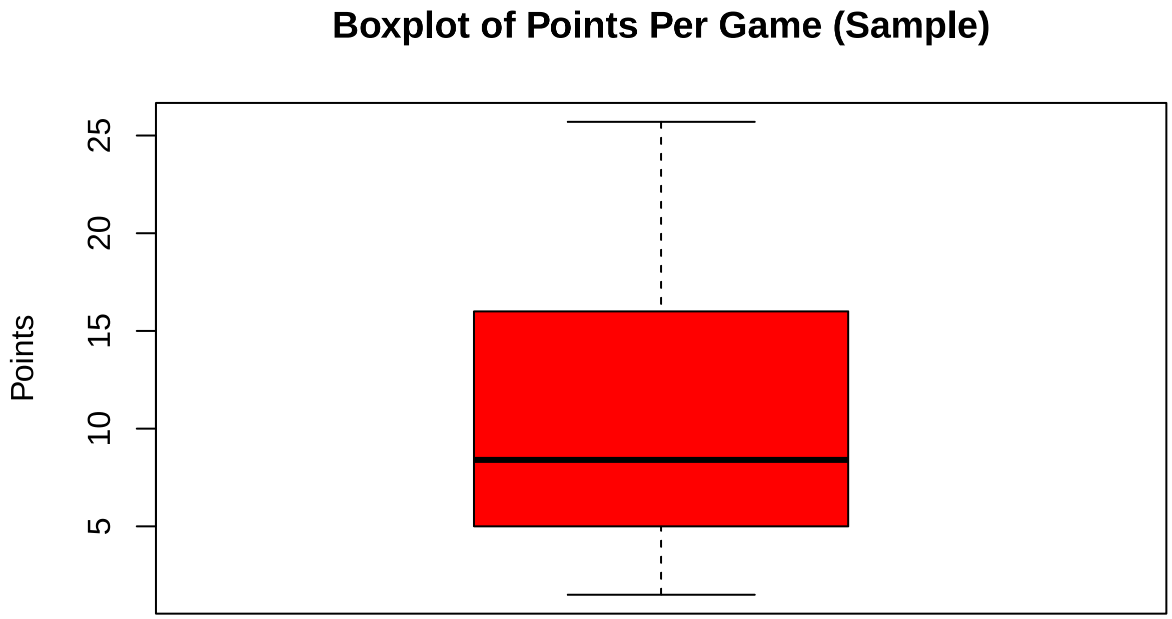
Which player is the lowest scoring player that is considered an outlier?

ANSWER: The player that scored 0 points

Part (g)

Create a boxplot of the points per game variable for the sample. Include a title and axis label, as before.

```r
boxplot(mysample$PTS, main = "Boxplot of Points Per Game (Sample)",
        ylab = "Points", col="red")
```



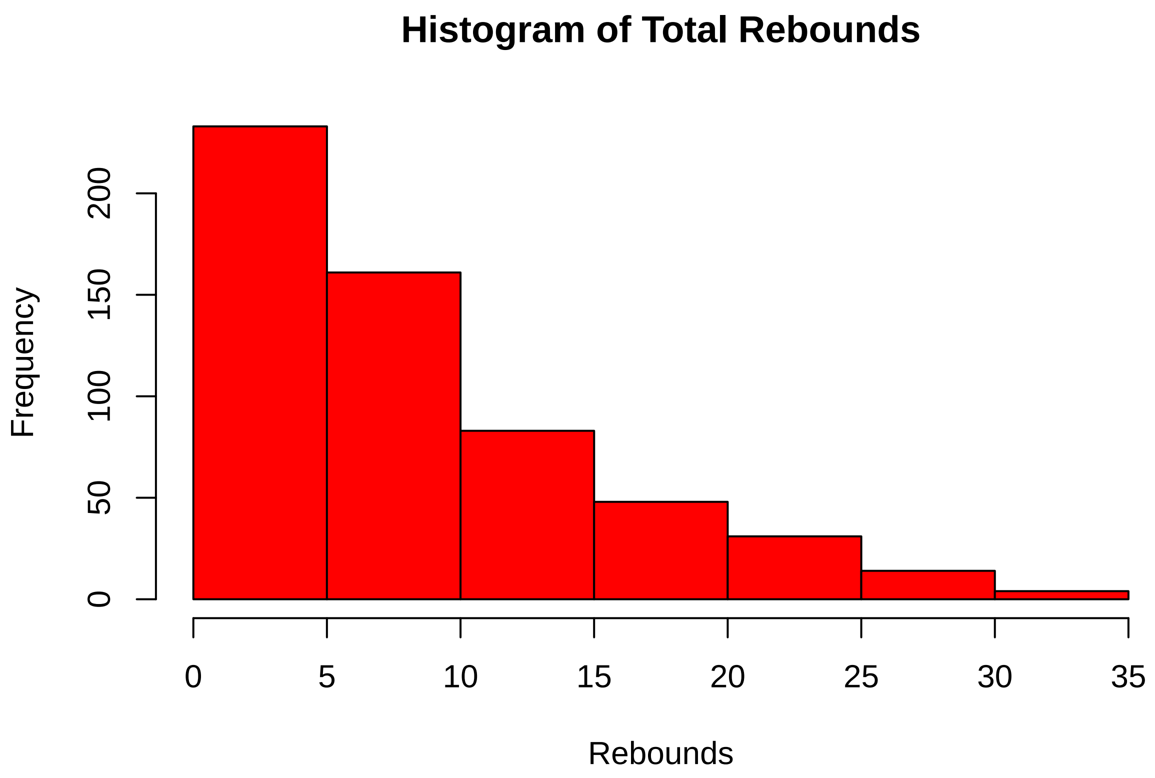
How many outliers are there (if any)?

ANSWER: There are no outliers.

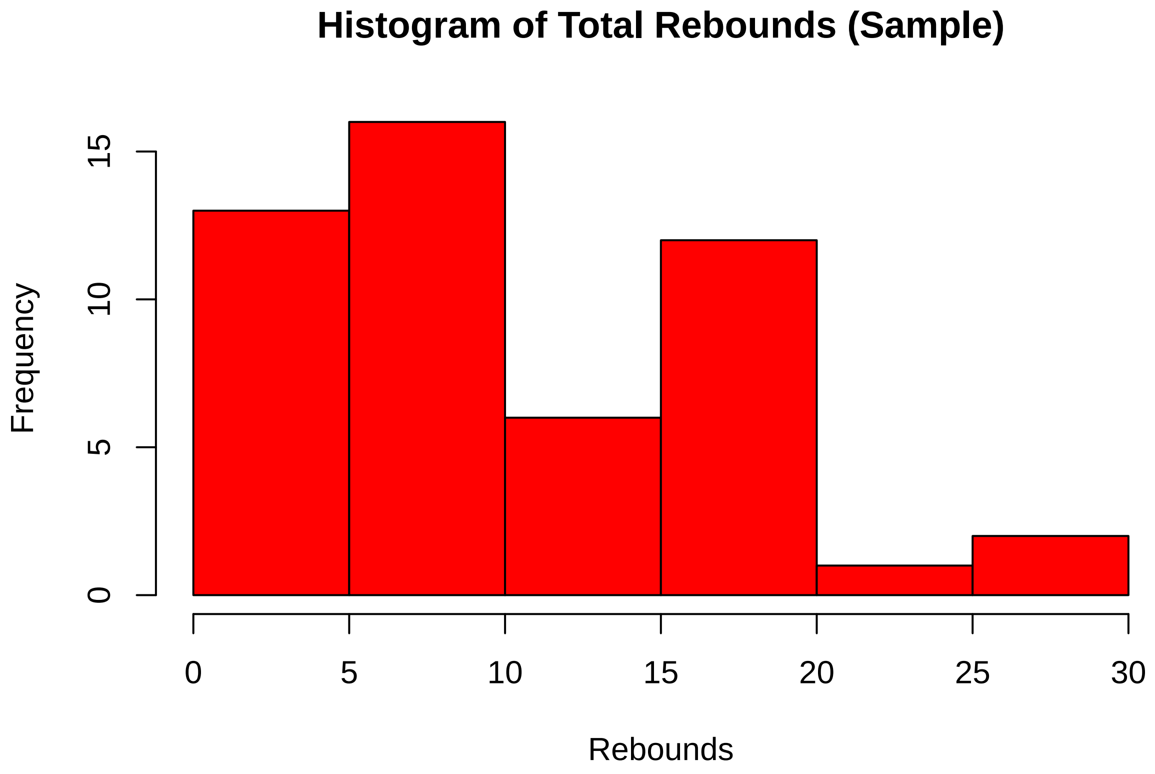
Part (h)

Use the hist() function to create two histograms: one for the variable total rebounds per game (TRB) for the population, and one for the total rebounds per game of the sample. Include titles and axes labels, as before.

```r
# histogram of population
hist(nba$PTS, col="red", main="Histogram of Total Rebounds", xlab="Rebounds",
    ylab="Frequency")
```



```
# histogram of sample
hist(mysample$PTS, col="red", main="Histogram of Total Rebounds (Sample)", xlab="Rebounds",
    ylab="Frequency")
```



Describe the shape of each histogram.

ANSWER: - The shape of the population histogram is right-skewed and unimodal - The shape of the sample histogram is bimodal