Intro to Data Science - HW 4

```
# Enter your name here: Hrishikesh Telang
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

(Chapters 8, 9, and 10 of Introduction to Data Science)

Reminders of things to practice from previous weeks: Descriptive statistics: mean( ) max( ) min( ) Sequence operator: : (For example, 1:4 is shorthand for 1, 2, 3, 4) Create a function: myFunc <- function(myArg) { } ?command: Ask R for help with a command

This module: Sampling is a process of **drawing elements from a larger set**. In data science, when analysts work with data, they often work with a sample of the data, rather than all of the data (which we call the **population**), because of the expense of obtaining all of the data.

One must be careful, however, because **statistics from a sample rarely match the characteristics of the population**. The **goal of this homework** is to **sample from a data set several times and explore the meaning of the results**. Before you get started make sure to read Chapters 8-10 of An Introduction to Data Science. Don't forget your comments!

Part 1: Write a function to compute statistics for a vector of numeric values

A. Create a new function which takes a numeric vector as its input argument and returns a dataframe of statistics about that vector as the output. As a start, the dataframe should have the min, mean, and max of the vector. The function should be called **vectorStats**:

```
vectorStats <- function(inputVec){
  df <- data.frame('Minimum' = min(inputVec),
                   'Mean' = mean(inputVec),
                   'Max' = max(inputVec))
  return (df)
}
```

B. Test your function by calling it with the numbers **one through ten**:

```
vectorStats(1:10)
```

```
##   Minimum Mean Max
## 1       1  5.5  10
```

C. Enhance the vectorStats() function to add the **median** and **standard deviation** to the returned dataframe.

```
vectorStats <- function(inputVec){
  df <- data.frame('Minimum' = min(inputVec),
                   'Mean' = mean(inputVec),
                   'Max' = max(inputVec),
                   'Median' = median(inputVec),
                   'Standard Deviation' = sd(inputVec))
  return (df)
}
```

D. Retest your enhanced function by calling it with the numbers **one through ten**:

```
vectorStats(1:10)
```

```
##   Minimum Mean Max Median Standard.Deviation
## 1       1  5.5  10    5.5            3.02765
```

Part 2: Sample repeatedly from the mtcars built-in dataframe

E. Copy the mtcars dataframe:

```
myCars <- mtcars
```

Use **head(myCars)** and **tail(myCars)** to show the data. Add a comment that describes what each variable in the data set contains. **Hint:** Use the ? or help( ) command with mtcars to get help on this dataset.

```
head(myCars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
tail(myCars)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

F. Sample three observations from **myCars$mpg**.

```
sample1 <- sample(myCars$mpg, size=3, replace=TRUE)
sample1
```

```
## [1] 22.8 19.2 33.9
```

G. Call your vectorStats( ) function with a new sample of three observations from **myCars$mpg**, where the sampling is done inside the **vectorStats** function call. Then use the **mean** function, with another sample done inside the mean function. Is the mean returned from the vectorStats function from the first sample the same as the mean returned from the mean function on the second sample? Why or Why not?

```
vectorStats(sample(myCars$mpg, size=3, replace=TRUE))
```

```
##   Minimum     Mean  Max Median Standard.Deviation
## 1    13.3 15.16667 16.4   15.8           1.644182
```

```
mean(sample(myCars$mpg, size=3, replace=TRUE))
```

```
## [1] 18.6
```

```
#No, the mean returned from the first sample is not the same as that from the second sample because the
```

H. Use the replicate( ) function to repeat your sampling of mtcars ten times, with each sample calling mean() on three observations. The first argument to replicate( ) is the number of repeats you want. The second argument is the little chunk of code you want repeated.

```
replicate(10, mean(sample(myCars$mpg, size=3, replace=TRUE)), simplify = TRUE)
```

```
##  [1] 18.06667 18.43333 20.20000 16.76667 20.03333 19.90000 14.76667 19.76667
##  [9] 20.90000 14.70000
```

I. Write a comment describing why every replication produces a different result.

```
#Every replication produces a different result because the function of replicate() by itself returns di
```

J. Rerun your replication, this time doing 1000 replications and storing the output of replicate() in a variable called **values**.

```
values <- replicate(1000, mean(sample(myCars$mpg, size=3, replace=TRUE)), simplify=TRUE)
values
```

```
##    [1] 16.53333 17.13333 23.20000 23.33333 20.20000 21.50000 21.60000 17.60000
##    [9] 17.16667 22.10000 22.93333 15.83333 25.00000 20.03333 16.06667 15.70000
##   [17] 18.26667 22.70000 22.93333 23.20000 21.26667 23.90000 16.36667 22.56667
##   [25] 20.86667 19.23333 15.90000 16.80000 24.80000 20.03333 19.76667 18.76667
##   [33] 17.96667 22.13333 22.86667 14.70000 20.93333 22.30000 20.23333 20.53333
##   [41] 17.80000 16.10000 22.70000 23.16667 21.10000 17.66667 19.33333 19.06667
##   [49] 18.03333 13.80000 17.96667 16.70000 17.93333 20.10000 16.30000 18.70000
##   [57] 22.26667 22.30000 18.26667 18.60000 24.90000 19.63333 22.33333 18.13333
##   [65] 20.93333 22.33333 16.53333 17.33333 21.33333 19.06667 19.63333 24.26667
##   [73] 17.13333 24.06667 18.43333 23.30000 18.56667 23.20000 24.23333 23.60000
##   [81] 19.36667 13.63333 15.80000 22.30000 18.13333 22.20000 17.76667 17.93333
##   [89] 23.46667 26.53333 23.86667 26.36667 19.53333 14.86667 17.63333 23.00000
##   [97] 25.60000 22.93333 21.86667 17.50000 16.66667 21.46667 22.90000 22.76667
##  [105] 20.93333 21.56667 21.23333 20.83333 22.73333 19.06667 16.10000 24.30000
##  [113] 17.63333 17.63333 20.40000 18.03333 14.93333 19.43333 17.53333 19.40000
##  [121] 18.70000 21.13333 21.26667 17.70000 18.26667 20.13333 22.13333 21.13333
##  [129] 19.46667 20.40000 23.46667 12.10000 20.16667 24.80000 19.00000 25.40000
##  [137] 20.40000 22.93333 18.10000 28.53333 24.03333 22.96667 21.33333 17.36667
##  [145] 17.50000 15.53333 18.96667 21.76667 17.96667 18.53333 22.03333 15.03333
##  [153] 23.86667 15.83333 21.00000 14.66667 15.96667 16.06667 23.16667 19.26667
##  [161] 16.63333 18.36667 17.03333 17.03333 20.10000 25.63333 23.00000 17.10000
##  [169] 15.90000 22.90000 13.83333 17.46667 19.23333 21.96667 23.26667 17.83333
##  [177] 19.93333 16.83333 24.20000 19.93333 16.83333 18.76667 20.36667 20.36667
##  [185] 27.83333 17.10000 16.36667 18.46667 16.60000 21.83333 27.66667 18.76667
##  [193] 22.06667 22.63333 19.06667 20.86667 17.20000 15.70000 21.60000 25.36667
##  [201] 18.10000 26.30000 24.33333 20.53333 17.06667 24.83333 24.80000 14.70000
##  [209] 17.76667 16.86667 29.00000 14.66667 20.03333 27.83333 21.96667 27.26667
##  [217] 20.70000 20.13333 16.63333 22.83333 21.46667 17.66667 21.30000 20.00000
##  [225] 27.76667 20.93333 21.83333 21.60000 15.36667 16.86667 12.86667 22.56667
##  [233] 22.80000 14.40000 19.96667 17.13333 20.20000 19.06667 22.50000 22.20000
##  [241] 17.06667 24.13333 22.00000 16.40000 23.40000 22.30000 23.40000 17.03333
##  [249] 22.46667 19.66667 19.63333 22.06667 15.86667 22.80000 20.06667 17.70000
##  [257] 17.83333 21.76667 22.36667 18.63333 20.96667 19.46667 22.73333 19.50000
##  [265] 13.30000 18.40000 22.03333 21.13333 23.10000 23.06667 17.06667 16.06667
##  [273] 16.30000 20.36667 19.06667 17.60000 18.80000 16.30000 15.36667 14.76667
##  [281] 24.40000 18.46667 18.76667 21.63333 22.93333 18.00000 14.30000 26.90000
##  [289] 22.93333 20.56667 20.33333 18.60000 25.73333 17.20000 18.36667 18.23333
##  [297] 19.80000 28.60000 25.56667 19.86667 23.13333 19.16667 13.93333 26.60000
##  [305] 27.26667 17.16667 14.96667 15.23333 17.40000 19.13333 19.90000 16.06667
##  [313] 21.13333 22.30000 22.86667 15.10000 19.03333 15.13333 16.63333 17.20000
##  [321] 15.33333 17.13333 17.56667 21.70000 23.03333 18.56667 16.00000 20.20000
##  [329] 19.16667 25.33333 15.80000 20.70000 26.96667 18.23333 24.06667 27.36667
##  [337] 24.03333 17.53333 23.46667 20.13333 18.20000 19.53333 17.46667 18.20000
##  [345] 16.40000 26.60000 23.30000 26.83333 22.60000 22.40000 21.26667 18.06667
##  [353] 21.66667 20.56667 18.16667 20.96667 20.53333 17.53333 16.56667 16.80000
##  [361] 23.00000 17.40000 21.96667 21.13333 14.26667 21.36667 23.16667 17.93333
##  [369] 26.66667 22.13333 19.36667 15.93333 22.60000 27.60000 22.46667 22.06667
##  [377] 16.06667 26.33333 23.76667 21.26667 17.66667 22.10000 17.40000 19.50000
##  [385] 22.83333 14.06667 30.26667 25.53333 15.96667 19.80000 22.03333 16.86667
##  [393] 17.03333 17.50000 25.60000 20.60000 16.23333 21.30000 19.30000 24.73333
##  [401] 20.86667 20.26667 17.66667 20.96667 21.93333 19.76667 15.53333 21.96667
##  [409] 14.70000 18.06667 16.86667 21.83333 20.66667 16.26667 19.43333 15.16667
##  [417] 16.86667 22.63333 21.36667 13.50000 18.63333 23.73333 23.26667 15.13333
##  [425] 19.93333 22.16667 22.03333 20.10000 21.86667 22.06667 19.60000 22.13333
##  [433] 22.36667 22.46667 20.76667 15.50000 26.20000 24.66667 15.03333 25.86667
##  [441] 19.43333 18.46667 18.73333 19.13333 19.33333 18.36667 17.70000 19.93333
##  [449] 19.30000 21.13333 23.80000 18.06667 20.66667 27.36667 30.76667 22.83333
##  [457] 16.40000 19.20000 22.66667 22.40000 18.80000 20.66667 19.36667 17.00000
##  [465] 21.20000 20.06667 15.76667 21.26667 20.70000 18.36667 24.73333 31.56667
##  [473] 21.03333 21.76667 20.26667 19.20000 21.06667 19.73333 22.43333 21.46667
##  [481] 15.60000 23.43333 17.16667 18.56667 22.33333 16.66667 17.20000 17.80000
##   [489] 22.63333 19.86667 19.73333 17.73333 19.43333 19.30000 20.43333 22.13333
##   [497] 19.06667 20.16667 20.40000 19.53333 16.23333 18.73333 15.26667 24.83333
##   [505] 19.60000 18.93333 16.10000 22.33333 24.26667 27.03333 18.73333 27.50000
##   [513] 19.83333 14.40000 21.26667 25.66667 19.16667 23.16667 19.56667 21.13333
##   [521] 16.26667 16.30000 15.63333 14.23333 21.90000 21.13333 20.46667 18.13333
##   [529] 23.56667 22.40000 19.60000 18.03333 19.26667 23.66667 17.53333 19.53333
##   [537] 20.80000 22.86667 15.03333 22.73333 14.50000 18.90000 18.13333 21.90000
##   [545] 21.26667 18.90000 16.83333 24.30000 19.76667 13.70000 21.60000 15.23333
##   [553] 19.60000 22.13333 19.53333 22.20000 23.40000 27.10000 23.53333 17.63333
##   [561] 16.46667 16.36667 18.23333 21.53333 18.30000 21.26667 19.80000 20.83333
##   [569] 22.36667 19.03333 23.13333 19.20000 23.16667 22.36667 19.80000 20.93333
##   [577] 17.60000 16.96667 18.86667 19.60000 18.10000 21.36667 17.46667 20.16667
##   [585] 23.96667 19.20000 23.33333 27.63333 23.06667 17.03333 24.53333 17.63333
##   [593] 27.93333 17.46667 22.66667 17.40000 19.00000 24.73333 14.80000 25.20000
##   [601] 18.96667 26.03333 14.96667 19.20000 17.00000 20.73333 24.36667 24.60000
##   [609] 20.16667 18.80000 19.60000 24.26667 20.56667 16.00000 19.43333 19.26667
##   [617] 20.50000 17.93333 17.06667 23.00000 22.63333 26.90000 18.63333 18.53333
##   [625] 19.43333 18.30000 22.50000 15.66667 26.03333 11.70000 20.70000 14.63333
##   [633] 15.93333 15.10000 21.30000 16.80000 24.40000 18.90000 19.36667 20.53333
##   [641] 22.96667 18.23333 13.76667 22.06667 21.83333 18.73333 21.90000 23.33333
##   [649] 18.03333 20.20000 24.83333 26.20000 18.60000 17.50000 23.03333 19.46667
##   [657] 20.66667 16.83333 21.83333 23.36667 18.16667 18.36667 21.00000 24.43333
##   [665] 22.50000 22.96667 19.06667 17.83333 22.86667 21.76667 20.73333 22.40000
##   [673] 18.56667 26.36667 16.63333 20.73333 23.96667 16.46667 24.66667 22.63333
##   [681] 16.90000 21.80000 17.36667 20.66667 18.16667 27.83333 16.96667 22.23333
##   [689] 24.20000 17.40000 20.96667 25.46667 17.00000 18.33333 22.60000 20.23333
##   [697] 17.83333 18.30000 24.53333 25.80000 20.56667 20.50000 23.06667 17.30000
##   [705] 20.20000 22.70000 23.03333 24.33333 14.66667 16.53333 17.90000 16.06667
##   [713] 21.63333 16.93333 13.80000 16.63333 20.56667 20.23333 25.50000 23.13333
##   [721] 21.83333 15.60000 17.23333 18.76667 16.30000 22.30000 23.70000 21.13333
##   [729] 21.93333 16.06667 20.63333 16.76667 25.33333 18.10000 22.26667 15.70000
##   [737] 18.40000 18.83333 20.33333 23.63333 23.06667 19.20000 20.20000 17.86667
##   [745] 18.46667 17.10000 18.76667 22.70000 15.60000 22.93333 14.96667 26.03333
##   [753] 23.90000 22.13333 14.40000 19.06667 19.76667 17.16667 21.36667 22.80000
##   [761] 26.60000 16.26667 19.70000 23.30000 14.06667 15.03333 28.40000 22.16667
##   [769] 15.80000 19.16667 24.36667 22.03333 17.56667 18.70000 27.50000 19.03333
##   [777] 24.83333 18.86667 16.93333 23.90000 21.86667 17.80000 21.10000 22.40000
##   [785] 15.46667 20.76667 15.23333 16.33333 22.63333 24.73333 17.06667 17.93333
##   [793] 23.23333 13.23333 20.13333 15.93333 20.30000 23.70000 16.83333 20.53333
##   [801] 22.26667 18.16667 21.03333 20.20000 15.96667 22.43333 17.86667 16.76667
##   [809] 18.56667 16.20000 24.73333 24.30000 21.63333 22.20000 23.63333 25.30000
##   [817] 14.83333 23.83333 23.26667 23.23333 20.46667 14.13333 19.13333 22.56667
##   [825] 17.50000 19.06667 21.80000 13.36667 19.20000 16.46667 20.73333 22.80000
##   [833] 23.16667 16.53333 17.50000 21.50000 21.13333 21.90000 16.10000 17.20000
##   [841] 21.90000 14.46667 14.86667 19.66667 19.63333 19.66667 18.60000 16.63333
##   [849] 17.96667 15.80000 24.66667 17.73333 20.06667 15.50000 22.93333 18.20000
##   [857] 25.86667 19.26667 16.40000 21.90000 15.90000 17.80000 25.60000 24.23333
##   [865] 16.83333 21.43333 22.83333 16.76667 20.76667 19.20000 18.63333 20.96667
##   [873] 13.16667 13.83333 16.56667 21.16667 20.20000 22.83333 26.60000 21.40000
##   [881] 19.20000 19.63333 15.96667 17.46667 22.43333 22.76667 19.80000 26.93333
##   [889] 19.53333 18.66667 20.40000 17.06667 25.00000 21.73333 15.13333 21.40000
##   [897] 19.43333 27.10000 22.40000 17.43333 16.00000 28.53333 17.00000 18.93333
##   [905] 25.83333 25.40000 26.33333 16.93333 22.93333 32.23333 23.50000 19.50000
##   [913] 25.20000 21.20000 21.56667 18.93333 18.80000 18.26667 17.20000 19.50000
##  [921] 24.93333 22.66667 23.43333 22.33333 16.13333 23.66667 19.40000 29.00000
##  [929] 15.80000 17.33333 18.06667 21.06667 22.96667 17.06667 15.50000 22.06667
##  [937] 13.66667 16.70000 17.16667 20.26667 15.03333 17.50000 19.10000 21.36667
##  [945] 23.56667 21.46667 21.16667 22.36667 17.76667 15.53333 18.60000 16.06667
##  [953] 26.20000 16.56667 16.36667 19.46667 17.10000 17.13333 24.00000 18.46667
##  [961] 16.83333 23.23333 16.86667 19.16667 26.03333 27.16667 23.96667 19.16667
##  [969] 24.36667 15.30000 17.03333 17.86667 19.00000 18.90000 22.26667 19.63333
##  [977] 20.43333 21.70000 32.23333 16.36667 16.43333 20.60000 14.46667 17.03333
##  [985] 18.56667 16.76667 22.30000 21.70000 21.30000 23.20000 14.46667 21.43333
##  [993] 16.20000 18.90000 23.20000 17.63333 17.96667 25.30000 18.60000 18.60000
```

K. Generate a **histogram** of the means stored in **values**.

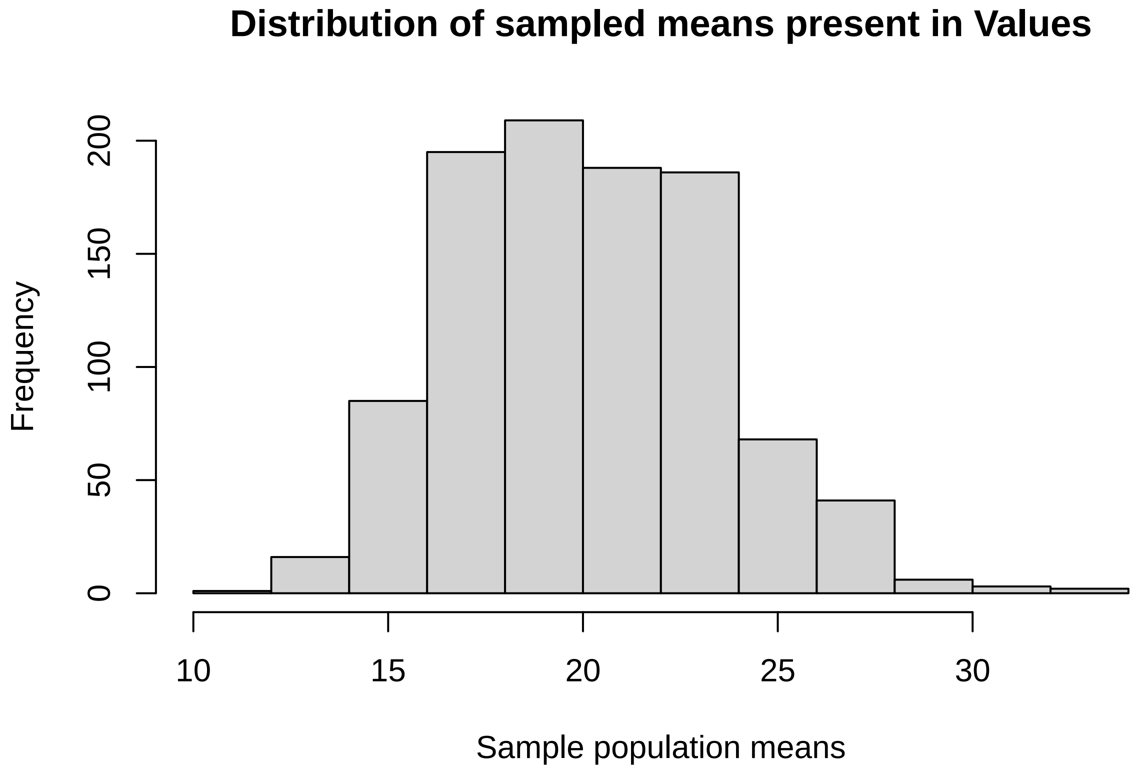
```
hist(values, simplify=TRUE, main='Distribution of sampled means present in Values', xlab='Sample popula
```

```
## Warning in plot.window(xlim, ylim, "", ...): "simplify" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "simplify" is not a graphical parameter
```

```
## Warning in axis(1, ...): "simplify" is not a graphical parameter
```

```
## Warning in axis(2, ...): "simplify" is not a graphical parameter
```



L. Repeat the replicated sampling, but this time, raise your sample size from **3 to 22**.

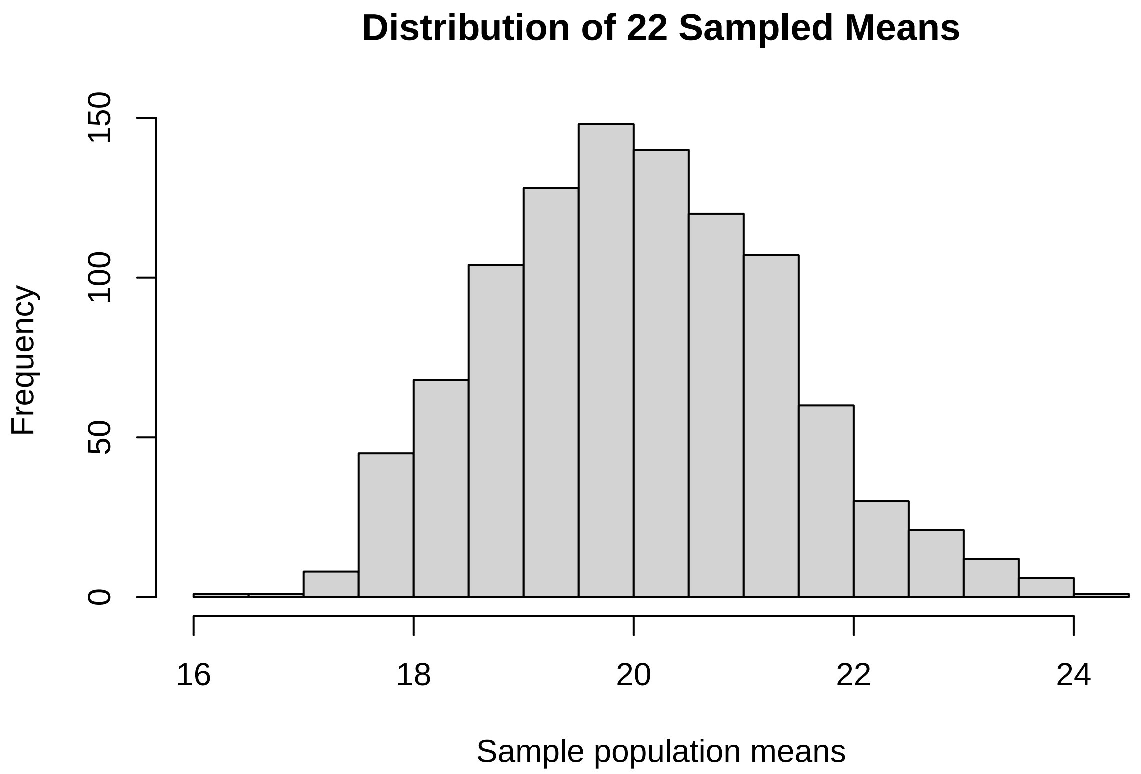
```
hist(replicate(1000, mean(sample(myCars$mpg, size=22, replace=TRUE)), simplify=TRUE),
     simplify=TRUE, main='Distribution of 22 Sampled Means', xlab='Sample population means')
```

```
## Warning in plot.window(xlim, ylim, "", ...): "simplify" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "simplify" is not a graphical parameter

## Warning in axis(1, ...): "simplify" is not a graphical parameter

## Warning in axis(2, ...): "simplify" is not a graphical parameter
```



M. Compare the two histograms - why are they different? Explain in a comment.

```
#The two histograms are different because there are of two different sample sizes. The histogram adjust
```