Bing-Je_Wu_HW4

Bing-Je Wu

4/25/2019

Step1. Write a summarizing function to understand a distribution of a vector

```r
printVecInfo <- function(inputVector){
  # set mean, median, min ,max, and sd
  Vmean <- mean(inputVector)
  Vmedian <- median(inputVector)
  Vmin <- min(inputVector)
  Vmax <- max(inputVector)
  Vsd <- sd(inputVector)
  # set quantile on 0.05% and 0.95%
  Q005<- quantile(inputVector,c(0.05,0.95))[1]
  names(Q005)<-NULL
  Q005
  Q095<- quantile(inputVector,c(0.05,0.95))[2]
  names(Q095)<-NULL
  Q095
  # set Skewness
  library(moments)
  Vskewness <- skewness(inputVector)
  # print all values
  print(paste("mean:", Vmean))
  print(paste("median:", Vmedian))
  print(paste("min:", Vmin, "max:", Vmax))
  print(paste("sd:", Vsd))
  print(paste("quantile (0.05 - 0.95):",Q005,"--",Q095))
  print(paste("Skewness:", Vskewness))
}
```

Step 2. Creating samples in a Jar

4. Create a variable 'jar' that has 50 red and 50 blue marbles

```r
jar <- c(replicate(50,"red"), replicate(50, "blue"))
```

5. Confirm it has 50 reds by summing the samples that are red

```r
length(jar[jar == "red"])
```

```
## [1] 50
```

6. Sample 10 'marbles' from the jar. How many are red? What was the percentage of red marbles?

```r
sample10<- sample(x = jar, size = 10, replace = TRUE)
length(sample10[sample10 == "red"])
```

```
## [1] 2
```

```
length(sample10[sample10 == "red"])/length(sample10)
```

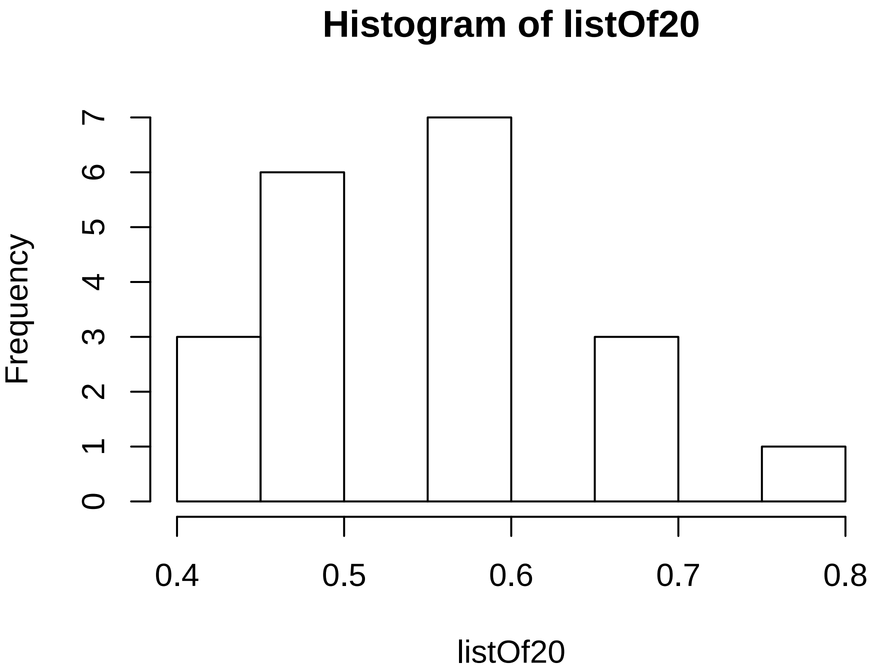
```
## [1] 0.2
```

7. Do the sampling 20 times, using 'replicate' command. This should generate a list of 20 numbers. Each number is the mean of how many reds there were in 10 samples. Use your printVecInfo to see information of the samples. Also generate a histogram of the samples.

```
listOf20<-
  replicate(20,length(sample(jar,10,replace = TRUE)[
  sample(jar,10, replace = TRUE) == "red"])/length(
    sample(jar,10, replace = TRUE))
  )
printVecInfo(listOf20)
```

```
## [1] "mean: 0.565"
## [1] "median: 0.6"
## [1] "min: 0.4 max: 0.8"
## [1] "sd: 0.108942283125661"
## [1] "quantile (0.05 - 0.95): 0.4 -- 0.705"
## [1] "Skewness: 0.231160062770844"
```

```
hist(listOf20)
```

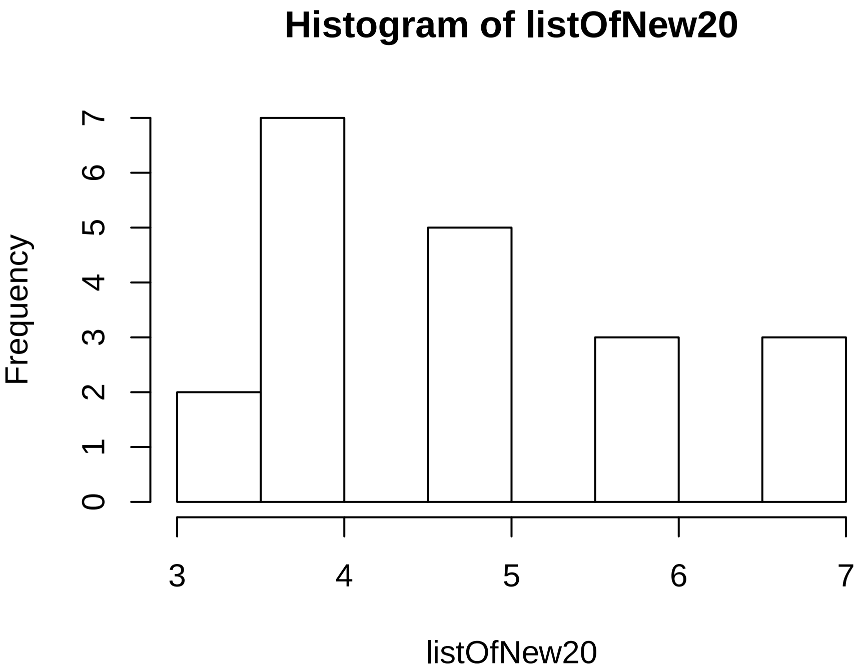


8. Repeat #7, but this time, sample the jar 100 times.You should get 20 numbers, this time each numebr represents the mean of how many reds were in the 100 samples. Use your printVecInfo to see information of the samples.Also generate a histogram of the samples.

```
listOfNew20 <-
  replicate(20,length(sample(jar,100,replace = TRUE)[
  sample(jar,10, replace = TRUE) == "red"])/length(
    sample(jar,10, replace = TRUE))
)
printVecInfo(listOfNew20)
```

```
## [1] "mean: 4.9"
## [1] "median: 5"
## [1] "min: 3 max: 7"
## [1] "sd: 1.25236618152662"
## [1] "quantile (0.05 - 0.95): 3 -- 7"
## [1] "Skewness: 0.356283412413649"
```

```
hist(listOfNew20)
```

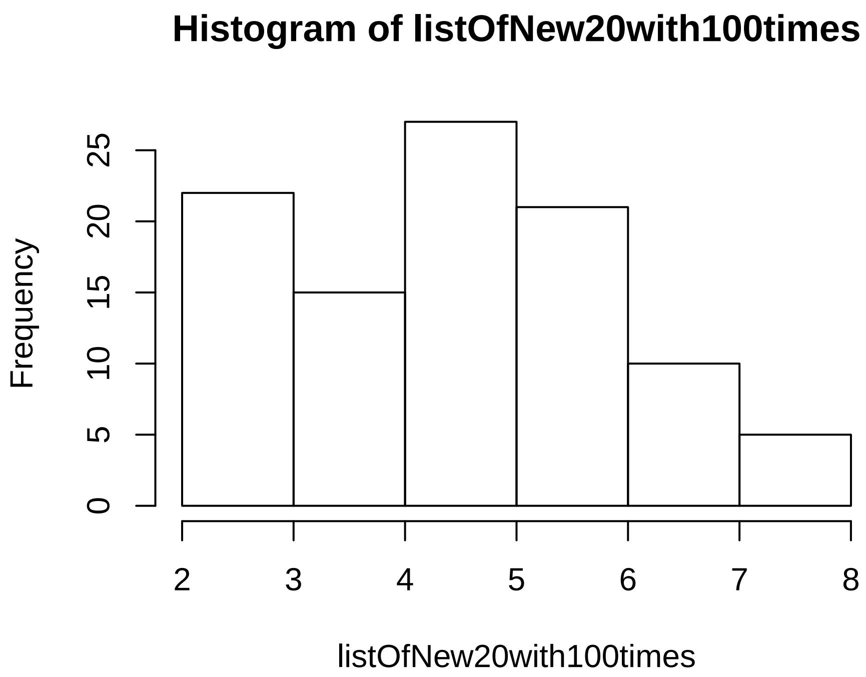


9. Repeat #8, but this time, replicate the sampling 100 times.You should get 20 numbers, this time each numebr represents the mean of how many reds were in the 100 samples. Use your printVecInfo to see information of the samples. Also generate a histogram of the samples.

```
listOfNew20with100times <-
  replicate(100,length(sample(jar,100,replace = TRUE)[
    sample(jar,10, replace = TRUE) == "red"])/length(
      sample(jar,10, replace = TRUE))
  )
printVecInfo(listOfNew20with100times)
```

```
## [1] "mean: 4.91"
## [1] "median: 5"
## [1] "min: 2 max: 8"
## [1] "sd: 1.55111559722659"
## [1] "quantile (0.05 - 0.95): 2 -- 7.05"
## [1] "Skewness: -0.0125563003674659"
```

```
hist(listOfNew20with100times)
```



Step 3. Explore the airquality dataset

10. Store the 'airquality' dataset into a tempary variable

```
T_airquality <- airquality
```

11. Clean the dataset (remove the NAs)

```
summary(T_airquality)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

```
colSums(is.na(T_airquality))
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

```
 # Remove the rows with NA
T_airquality_clean <- na.omit(T_airquality)
```

```
  # Check NAs again
rownames(T_airquality_clean)<-NULL
any(is.na(T_airquality_clean))
```

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

```
colSums(is.na(T_airquality_clean))
```

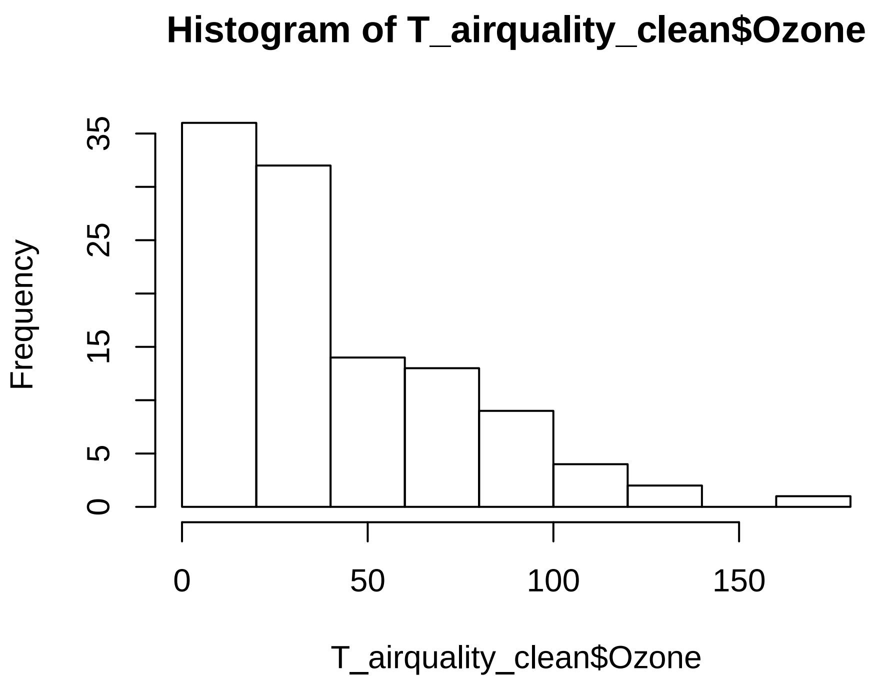
```
##   Ozone Solar.R    Wind    Temp   Month     Day
##       0       0       0       0       0       0
```

12. Explore Ozone, Wind and Temp by doing a 'printVecInfo' on each as well as generating a histogram for each

```
printVecInfo(T_airquality_clean$Ozone)
```

```
## [1] "mean: 42.0990990990991"
## [1] "median: 31"
## [1] "min: 1 max: 168"
## [1] "sd: 33.2759686574274"
## [1] "quantile (0.05 - 0.95): 8.5 -- 109"
## [1] "Skewness: 1.24810370040404"
```

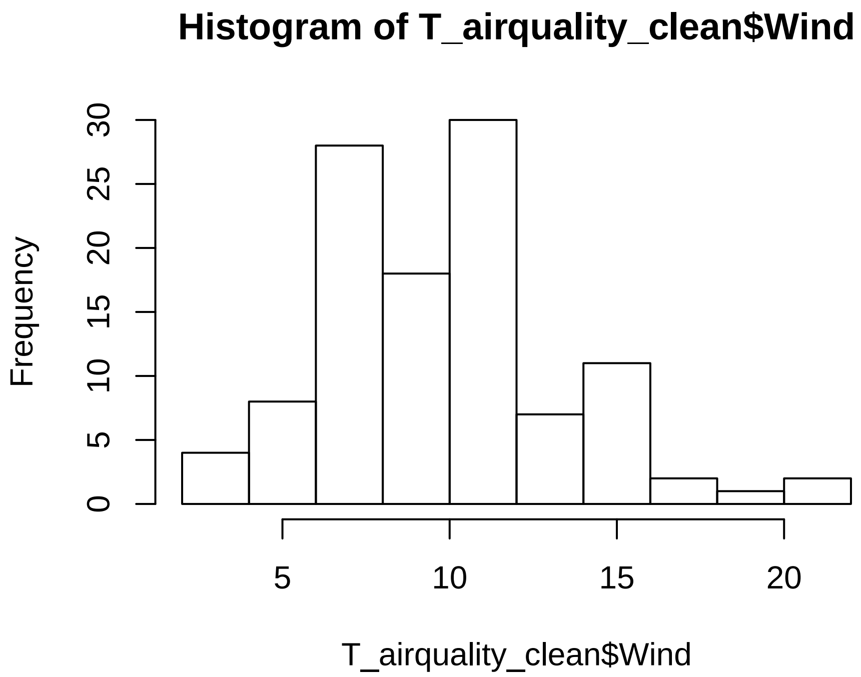
```
hist(T_airquality_clean$Ozone)
```



```
printVecInfo(T_airquality_clean$Wind)
```

```
## [1] "mean: 9.93963963963964"
## [1] "median: 9.7"
## [1] "min: 2.3 max: 20.7"
## [1] "sd: 3.55771324101922"
## [1] "quantile (0.05 - 0.95): 4.6 -- 15.5"
## [1] "Skewness: 0.455641432036776"
```

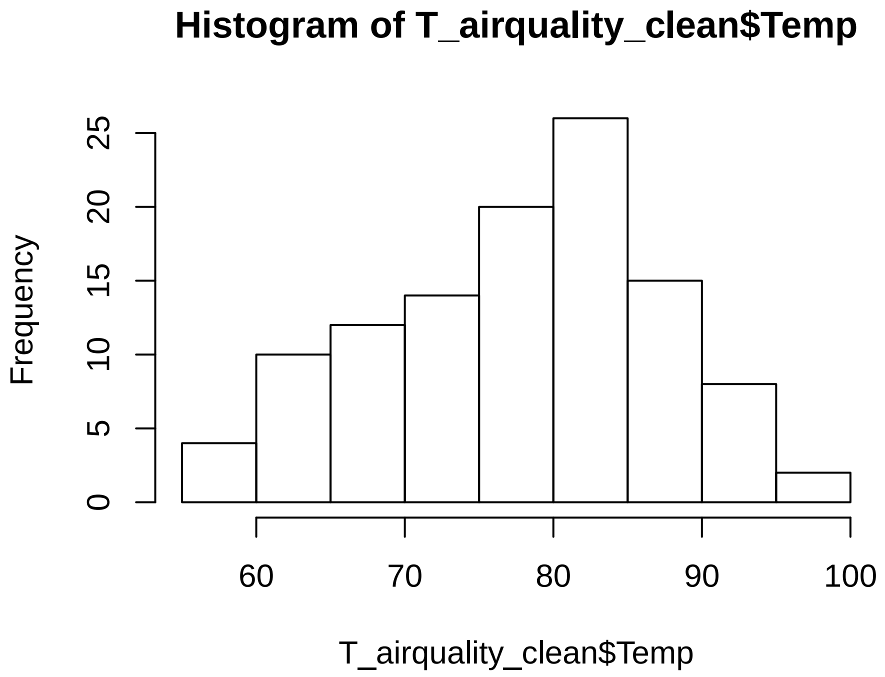
```
hist(T_airquality_clean$Wind)
```



```r
printVecInfo(T_airquality_clean$Temp)
```

```
## [1] "mean: 77.7927927927928"
## [1] "median: 79"
## [1] "min: 57 max: 97"
## [1] "sd: 9.52996910909533"
## [1] "quantile (0.05 - 0.95): 61 -- 92.5"
## [1] "Skewness: -0.225095889347339"
```

```r
hist(T_airquality_clean$Temp)
```


Histogram of T_airquality_clean$Temp
Frequency
T_airquality_clean$Temp