ThulasiRam_RuppaKrishnan_HW4

```
# Step 1: Write a summarizing function to understand the distribution of a vect
or
# 1. The function, call it 'printVecInfo' should take a vector as input
# 2. The function will print the following information:
#     a. Mean
#     b. Median
#     c. Min & max
#     d. Standard deviation
#     e. Quantiles (at 0.05 and 0.95)
#     f. Skewness

printVecInfo <- function(vc)
{
  library(moments)
  print(paste("Mean :" ,mean(vc)))
  print(paste("Median :" ,median(vc)))
  print(paste("Min :" ,min(vc)," Max :" ,max(vc)))
  print(paste("sd :" ,sd(vc)))
  print(paste("quantile (0.05-0.95) :" ,paste(quantile(vc,c(0.05,0.95)),collapse = " - ")))
  print(paste("Skewness :" ,skewness(vc)))
}
```

```
#3. Test the function with a vector that has (1,2,3,4,5,6,7,8,9,10,50). You shou
ld see
#       something such as:
#       [1] "mean: 9.54545454545454"
#       [1] "median:    6"
#       [1] "min:   1       max:    50"
#       [1] "sd:    13.7212509368762"
#       [1] "quantile   (0.05   - 0.95):    1.5 -- 30"
#       [1] "skewness:  2.62039633563579"

printVecInfo(c(1,2,3,4,5,6,7,8,9,10,50))
```

```
## [1] "Mean : 9.54545454545454"
## [1] "Median : 6"
## [1] "Min : 1  Max : 50"
## [1] "sd : 13.7212509368762"
## [1] "quantile (0.05-0.95) : 1.5 - 30"
## [1] "Skewness : 2.62039633563579"
```

```
# Step  2:  Creating    Samples in  a   Jar
#4. Create  a   variable    'jar'   that    has 50  red and 50  blue    marbles
#(hint: the jar can have    strings as  objects,    with    some    of  the strings being   're
d'   and
#some   of  the strings being   'blue'

jar <-as.character(replicate(50,sample(c("red","blue"),2,replace = FALSE),simplify = "FALSE"))

#5. Confirm there   are 50  reds    by  summing the samples that    are red
sum(as.numeric(jar=="red"))
```

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

```
#6. Sample  10  'marbles'   (really strings)    from    the jar.    How many    are red?    What
    was the
#percentage of  red marbles?

# Method 1
s1 <- sample(jar,10,replace = TRUE)
# Number of Red
length(which(s1=="red"))
```

```
## [1] 4
```

```
# Percentage of Red
(length(which(s1=="red"))/10)*100
```

```
## [1] 40
```

```
# Method 2
s2 <- as.numeric(sample(jar,10,replace = TRUE)=="red")
# Number of Red
sum(s2)
```

```
## [1] 6
```

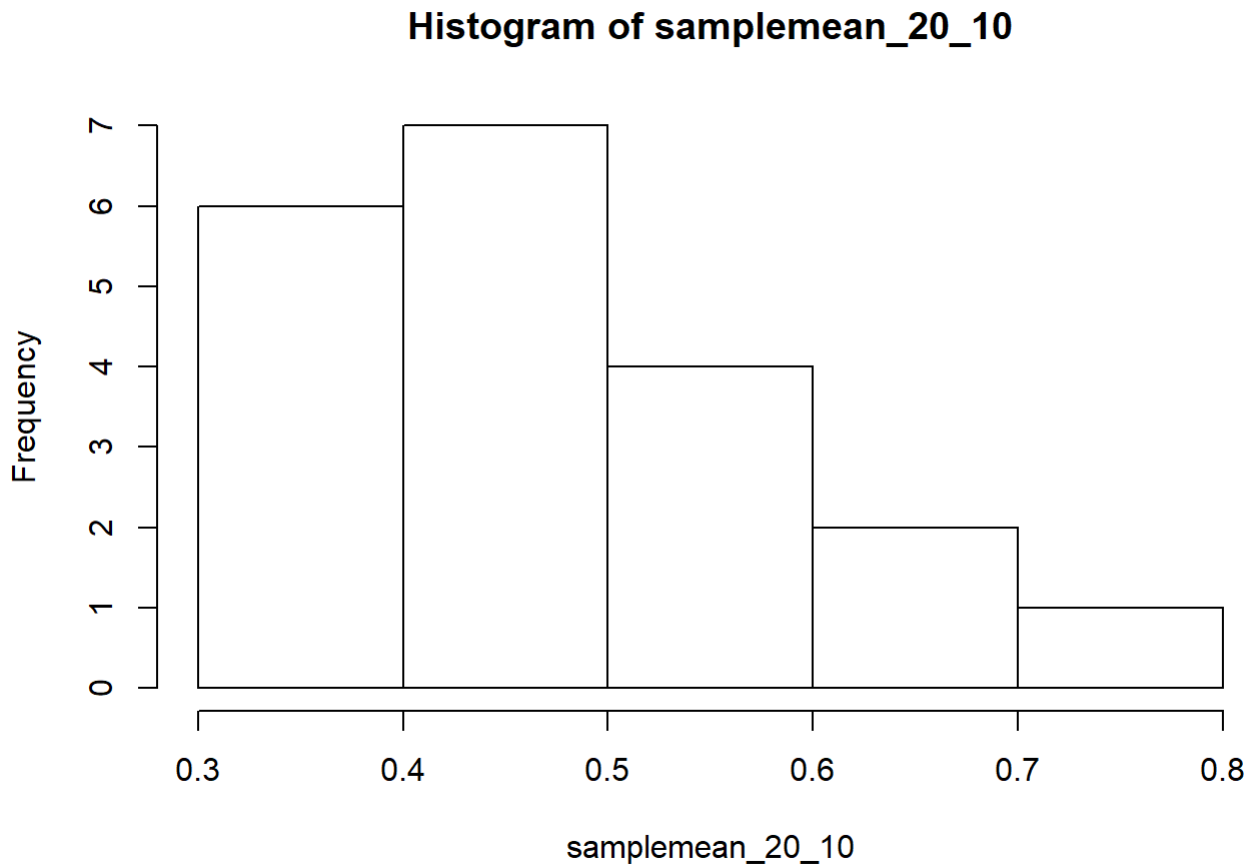
```
# Percentage of Red
(sum(s2)/length(s2))*100
```

```
## [1] 60
```

```
#7. Do the sampling 20 times, using the '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

samplemean_20_10<-replicate(20,mean(as.numeric(sample(jar,10,replace = TRUE)=="red")),simplify =
 "FALSE")
printVecInfo(samplemean_20_10)
```

```
## [1] "Mean : 0.52"
## [1] "Median : 0.5"
## [1] "Min : 0.3  Max : 0.8"
## [1] "sd : 0.123969435961577"
## [1] "quantile (0.05-0.95) : 0.395 - 0.705"
## [1] "Skewness : 0.462552332259953"
```

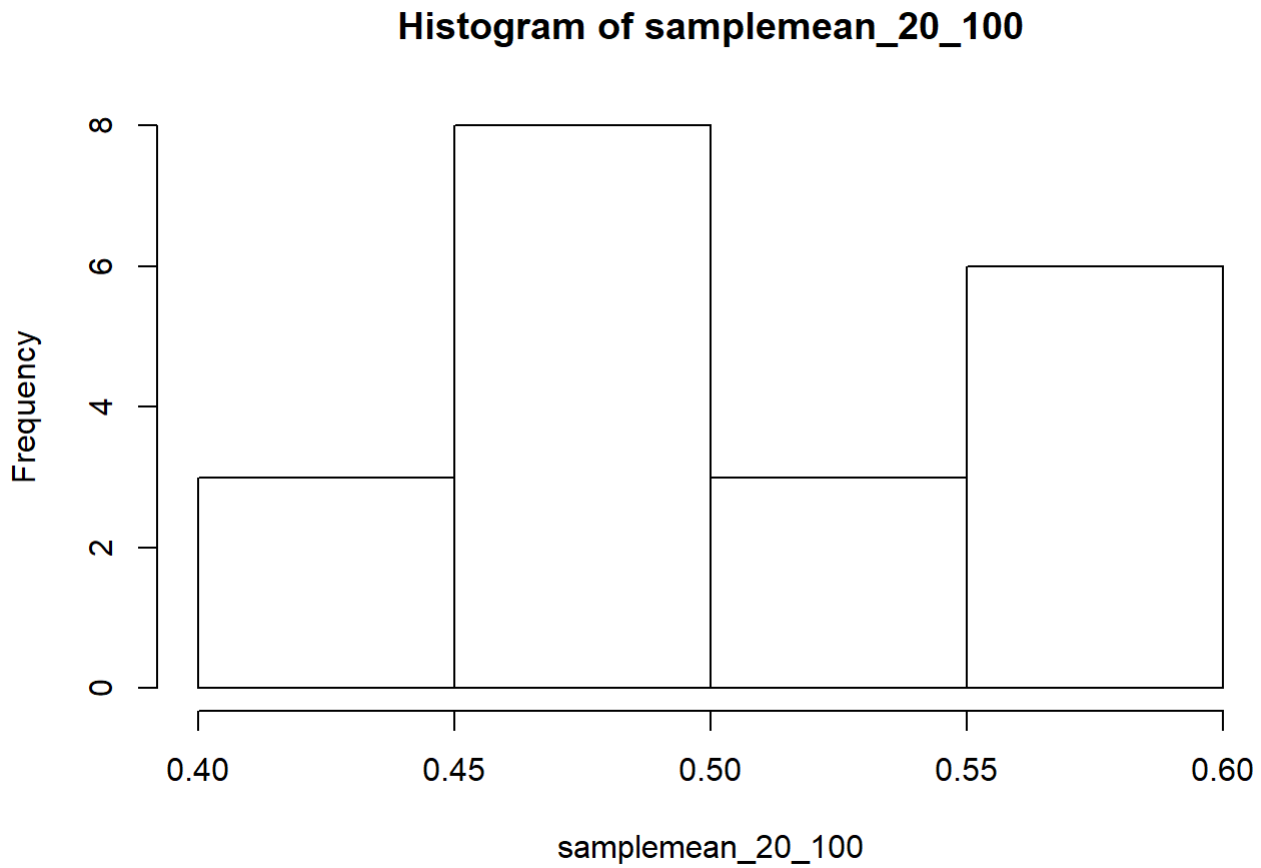
```
hist(samplemean_20_10)
```



```
#8. Repeat  #7, but this    time,   sample  the jar 100 times.  You should  get 20 numbers, this

#time   each    number  represents  the mean    of  how many    reds    there   were    in  the
 100
#samples.   Use your    printVecInfo    to  see information of  the samples.    Also    generate
    a
#histogram  of  the samples

samplemean_20_100<-replicate(20,mean(as.numeric(sample(jar,100,replace = TRUE)=="red")),simplify
 = "FALSE")
printVecInfo(samplemean_20_100)
```

```
## [1] "Mean : 0.514"
## [1] "Median : 0.5"
## [1] "Min : 0.41  Max : 0.6"
## [1] "sd : 0.0547145703677395"
## [1] "quantile (0.05-0.95) : 0.429 - 0.6"
## [1] "Skewness : 0.0185141046987806"
```

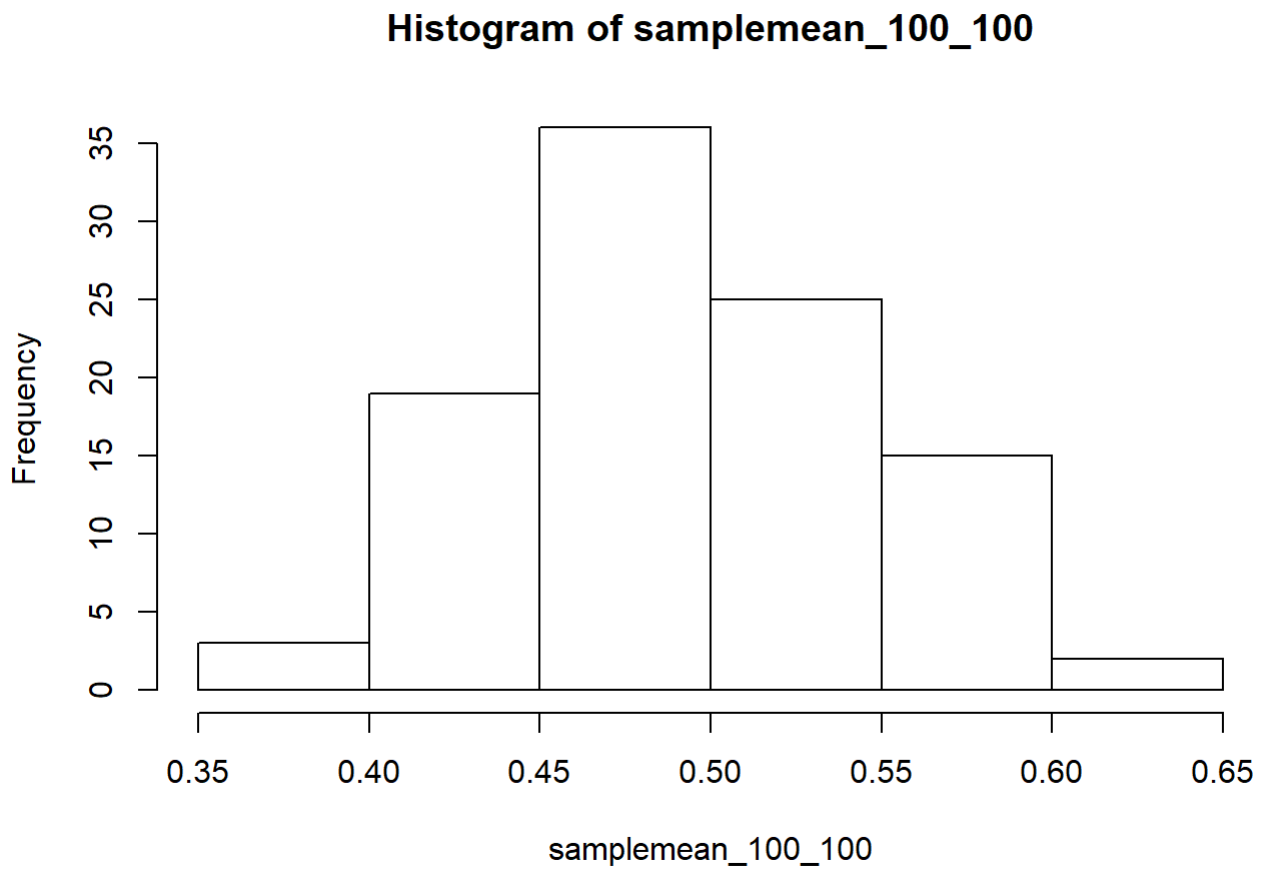
```
hist(samplemean_20_100)
```



```
#9. Repeat #8, but this time, replicate the sampling 100 times. You should get 100
#numbers, this time each number represents the mean of how many reds ther
e were
#in the 100 samples. Use your printVecInfo to see information of the samples. Also

#generate a histogram of the samples

samplemean_100_100<-replicate(100,mean(as.numeric(sample(jar,100,replace = TRUE)=="red")),simpli
fy = "FALSE")
printVecInfo(samplemean_100_100)
```

```
## [1] "Mean : 0.4993"
## [1] "Median : 0.495"
## [1] "Min : 0.37  Max : 0.63"
## [1] "sd : 0.0527286746591751"
## [1] "quantile (0.05-0.95) : 0.4295 - 0.59"
## [1] "Skewness : 0.219035376777496"
```

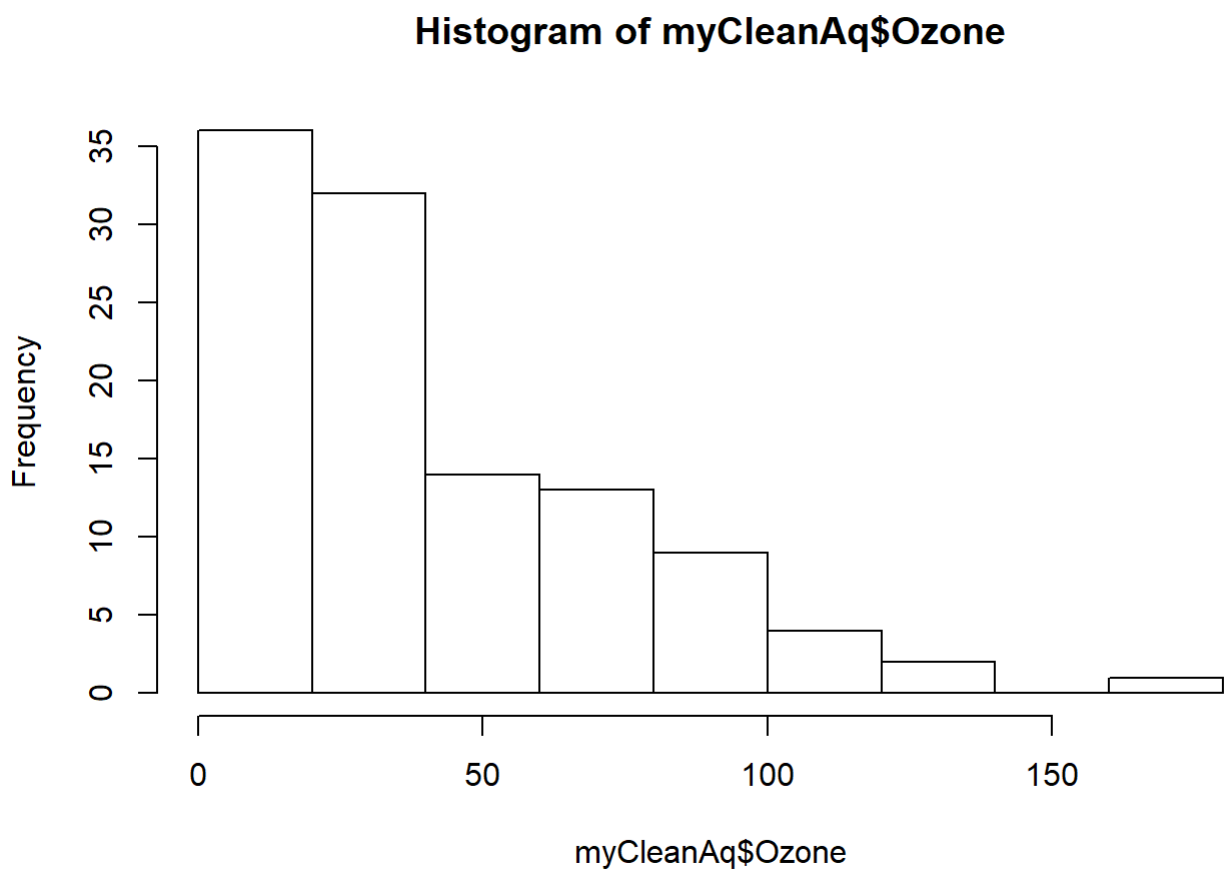
```
hist(samplemean_100_100)
```



```
#Step   3:  Explore the airquality  dataset

#10. Store  the 'airquality'    dataset into    a   temporary   variable

myAq<-airquality

#11. Clean  the dataset (i.e.   remove  the NAs)

myCleanAq<-myAq[which((myAq$Ozone=="NA" | myAq$Solar.R=="NA")=="FALSE"),]

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

printVecInfo(myCleanAq$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"
```

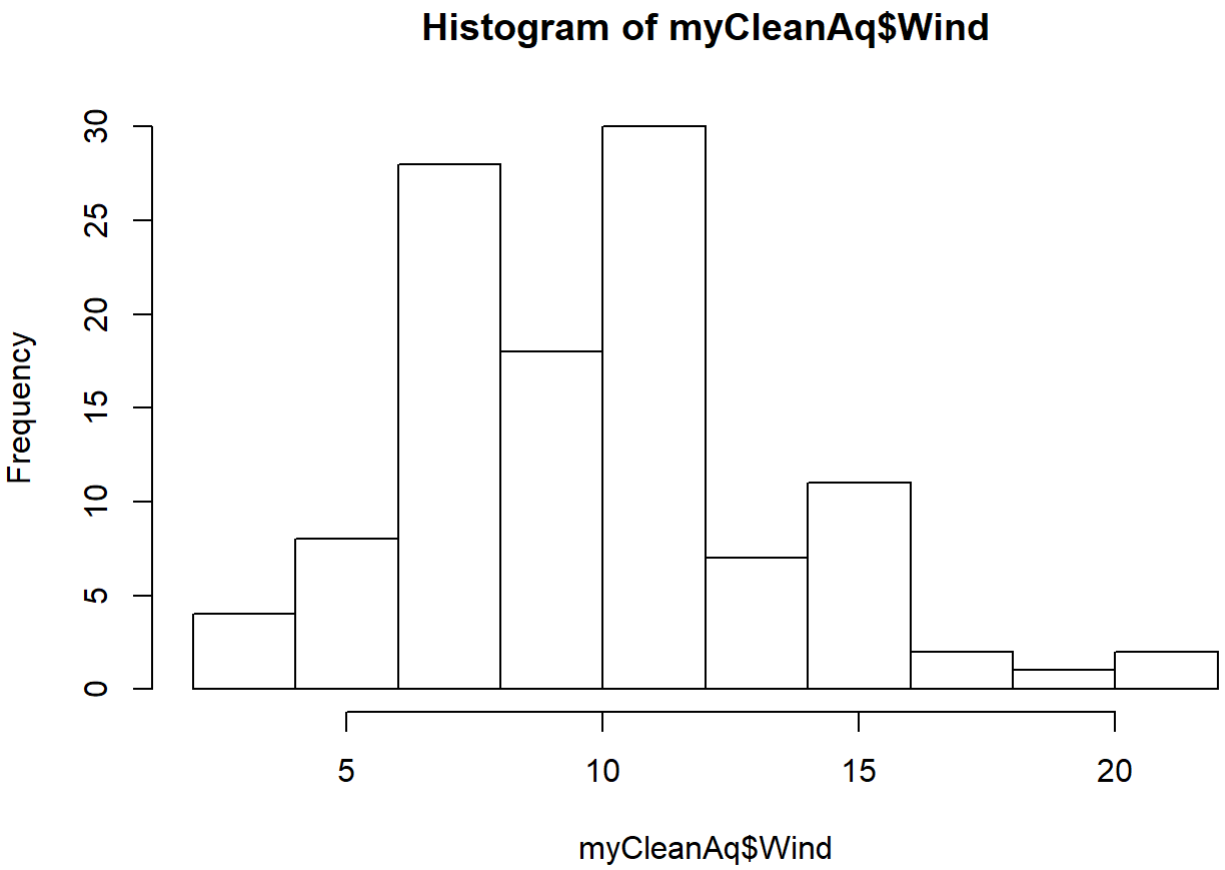
```
hist(myCleanAq$Ozone)
```



```
printVecInfo(myCleanAq$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"
```

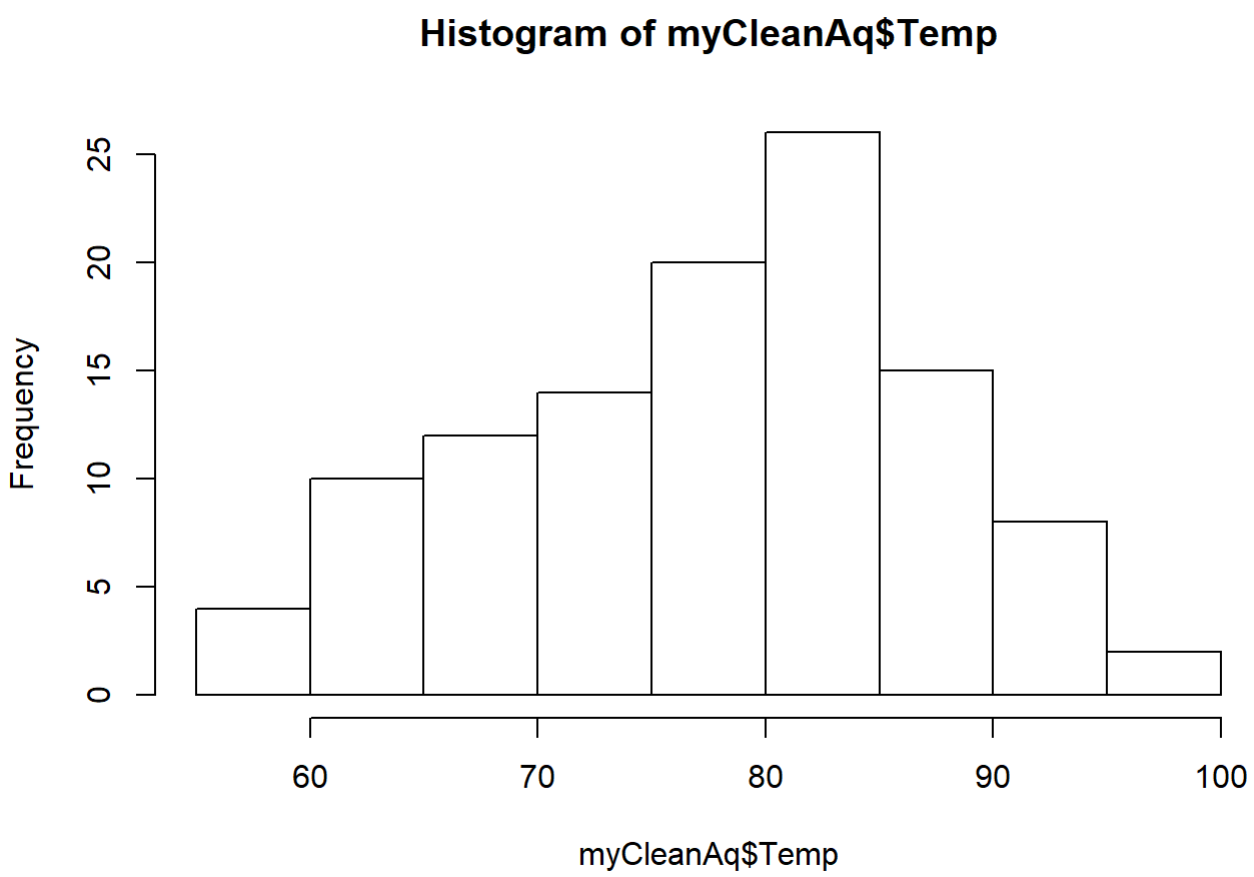
```
hist(myCleanAq$Wind)
```




```
printVecInfo(myCleanAq$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"
```

```
hist(myCleanAq$Temp)
```


Histogram of myCleanAq$Temp
Frequency
0
5
10
15
20
25
60
70
80
90
100
myCleanAq$Temp