

Sampling and Replication

Surabhi Chouhan

December 20, 2016

```
library("moments")

printVecInfo <- function(vec) #creating a function as asked in the question
{
  #using the cat() function, displayed various values in the format asked
for, the second argument is the separator which makes the values separated.
"\n" gives new line and " " gives a space as separator
  cat("mean:",mean(vec),"\n")
  cat("median:",median(vec),"\n")
  cat("min:",min(vec)," ")
  cat("max:",max(vec),"\n")
  cat("sd:",sd(vec),"\n")
  cat("quantile (0.05 - 0.95):",quantile(vec,probs=c(0.05,0.95)),"\n")
  cat("skewness:",skewness(vec))
}

test <- c(1,2,3,4,5,6,7,8,9,10,50) #creates the vector test
printVecInfo(test) #calling the function by passing test vector as argument

## mean: 9.545455
## median: 6
## min: 1 max: 50
## sd: 13.72125
## quantile (0.05 - 0.95): 1.5 30
## skewness: 2.620396

jar <- c() #creates the jar vector
jar[1:50] <- "Red" #the first 50 elements have value "Red"
jar[51:100] <- "Blue" #the last 50 elements have value "Blue"

#Reference taken from http://stackoverflow.com/questions/1923273/counting-the-number-of-elements-with-the-values-of-x-in-a-vector
BlueRed <- table(jar) #to count the number of red we use the function
table(), which creates a tabular view of the elements in the vector jar
BlueRed[names(BlueRed)=="Red"] #when we compare the names(BlueRed) with
"Red", we get the count of elements with value "Red" in table BlueRed (values
of vector jar that have value "Red" stored in table)

## Red
## 50

set.seed(10) #setting seed of sampling to get the same result each time
sampling is done
```

```

sample(jar, size=10, replace=TRUE) #sample jar 10 times and count the
occurrences of "Red"

## [1] "Blue" "Red" "Red" "Blue" "Red" "Red" "Red" "Red" "Blue" "Red"

#Simple print statements
print("Got 7 red marbles")

## [1] "Got 7 red marbles"

print("Percentage of red marbles:70%")

## [1] "Percentage of red marbles:70%"

#sampling jar 10times and as jar contains strings, so table() makes it into a
tabular format to get the numeric values, give the count of each red and blue
sample, then replicate it 10 times
sampling10 <-
replicate(10,table(sample(jar,size=10,replace=TRUE)),simplify=TRUE)
printVecInfo(sampling10[1,]) #calling function for row with values of Blue

## mean: 4.3
## median: 4
## min: 2 max: 7
## sd: 1.636392
## quantile (0.05 - 0.95): 2.45 6.55
## skewness: 0.2950825

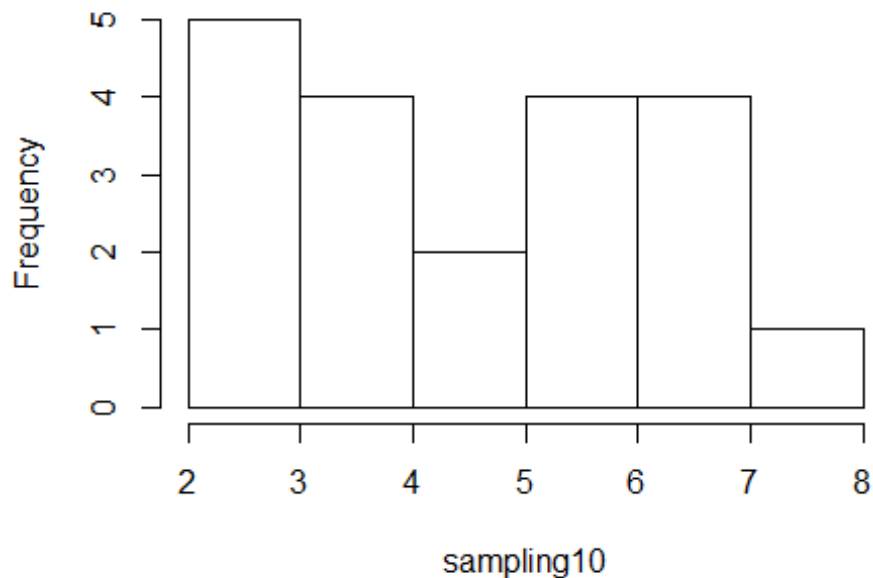
printVecInfo(sampling10[1,]) #calling function for row with values of Red

## mean: 4.3
## median: 4
## min: 2 max: 7
## sd: 1.636392
## quantile (0.05 - 0.95): 2.45 6.55
## skewness: 0.2950825

hist(sampling10) #generating the histogram for the sample created

```

Histogram of sampling10



#sampling jar 100times and as jar contains strings, so table() makes it into a tabular format to get the numeric values, give the count of each red and blue sample, then replicate it 10 times

```
sampling100 <-  
replicate(10,table(sample(jar,size=100,replace=TRUE)),simplify=TRUE)  
printVecInfo(sampling100[1,]) #calling function for row with values of Blue
```

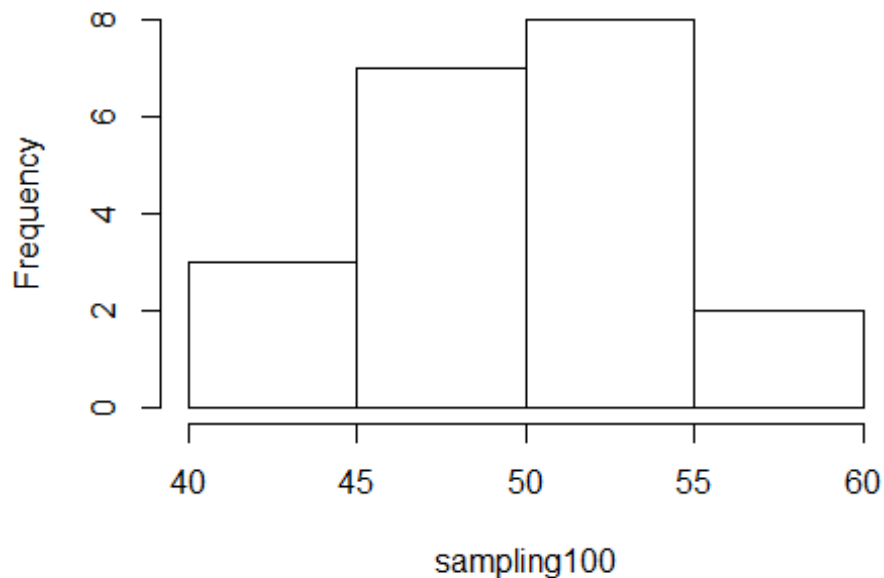
```
## mean: 50.9  
## median: 50  
## min: 45 max: 60  
## sd: 4.748099  
## quantile (0.05 - 0.95): 45.9 58.65  
## skewness: 0.6957482
```

```
printVecInfo(sampling100[2,]) #calling function for row with values of Red
```

```
## mean: 49.1  
## median: 50  
## min: 40 max: 55  
## sd: 4.748099  
## quantile (0.05 - 0.95): 41.35 54.1  
## skewness: -0.6957482
```

```
hist(sampling100) #generating the histogram for the sample created
```

Histogram of sampling100



#sampling jar 100times and as jar contains strings, so table() makes it into a tabular format to get the numeric values, give the count of each red and blue sample, then replicate it 100 times

```
replicate100 <-
```

```
replicate(100,table(sample(jar,size=100,replace=TRUE)),simplify=TRUE)
```

```
printVecInfo(replicate100[1,]) #calling function for row with values of Blue
```

```
## mean: 49.45
```

```
## median: 50
```

```
## min: 35 max: 66
```

```
## sd: 5.67357
```

```
## quantile (0.05 - 0.95): 39.95 59
```

```
## skewness: 0.04934929
```

```
printVecInfo(replicate100[2,]) #calling function for row with values of Red
```

```
## mean: 50.55
```

```
## median: 50
```

```
## min: 34 max: 65
```

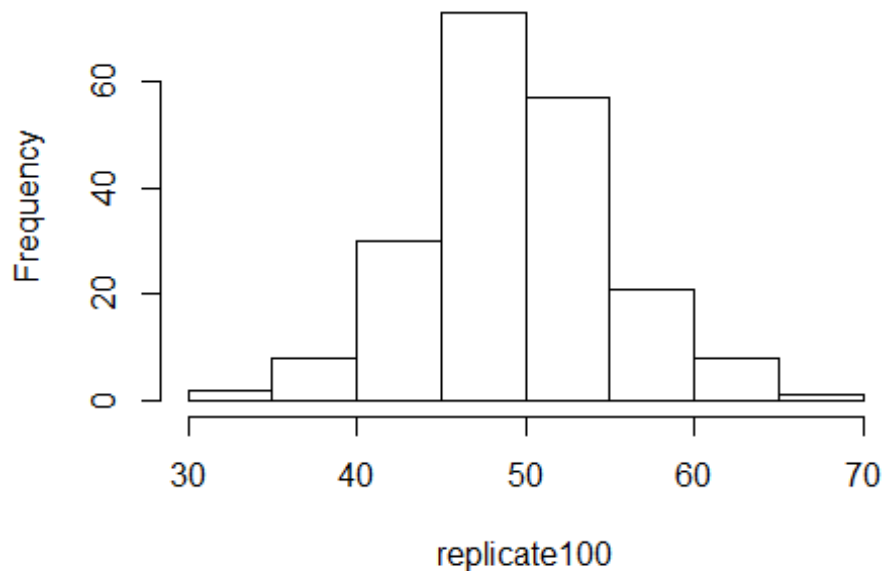
```
## sd: 5.67357
```

```
## quantile (0.05 - 0.95): 41 60.05
```

```
## skewness: -0.04934929
```

```
hist(replicate100)
```

Histogram of replicate100



```
temp_airquality <- airquality #copy the dataframe into a temporary one to be manipulated
```

```
without_na <- na.omit(temp_airquality) #omitting the NA values in dataframe
```

```
ozone <- printVecInfo(without_na[,1]) #calling function for the column Ozone
```

```
## mean: 42.0991
```

```
## median: 31
```

```
## min: 1 max: 168
```

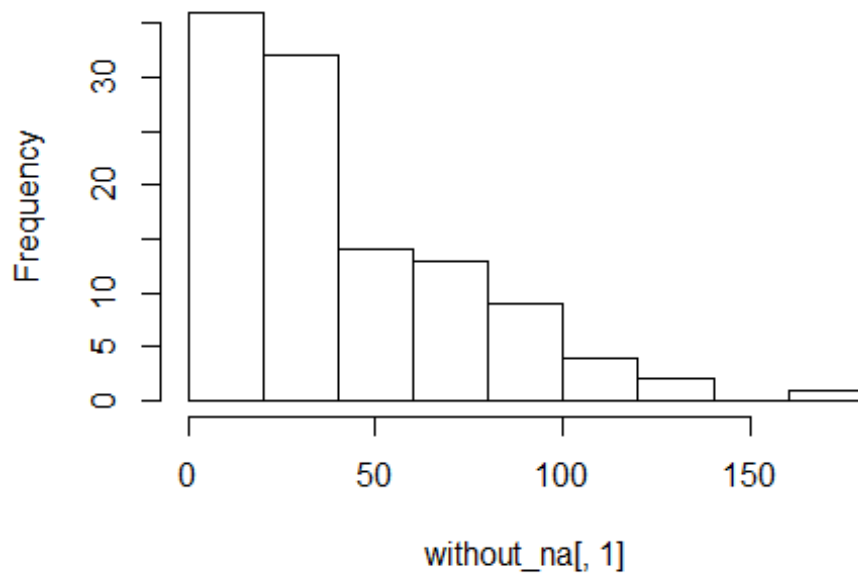
```
## sd: 33.27597
```

```
## quantile (0.05 - 0.95): 8.5 109
```

```
## skewness: 1.248104
```

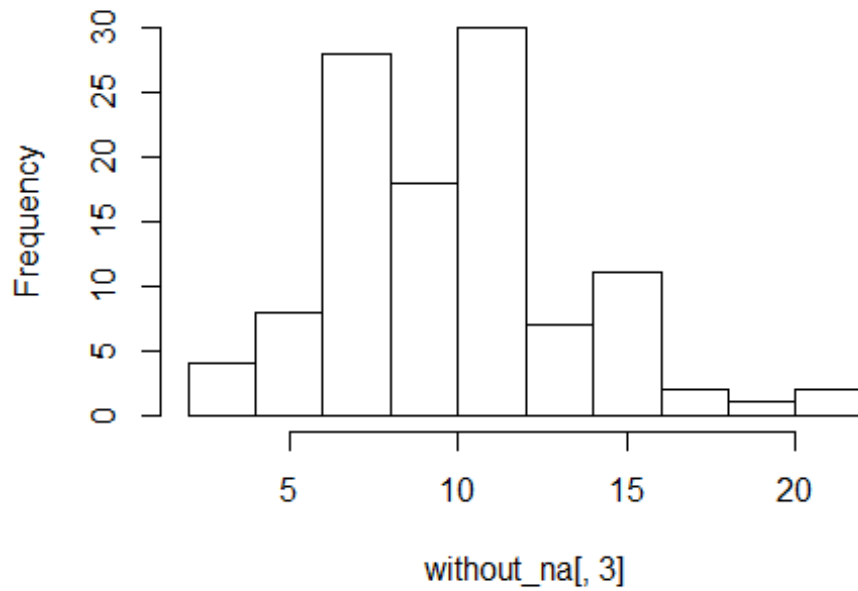
```
hist(without_na[,1]) #generating the histogram for column Ozone
```

Histogram of without_na[, 1]



```
wind <- printVecInfo(without_na[,3]) #calling function for the column Wind  
## mean: 9.93964  
## median: 9.7  
## min: 2.3 max: 20.7  
## sd: 3.557713  
## quantile (0.05 - 0.95): 4.6 15.5  
## skewness: 0.4556414  
hist(without_na[,3]) #generating the histogram for column Wind
```

Histogram of without_na[, 3]



```
temp <- printVecInfo(without_na[,4]) #calling function for the column Temp  
## mean: 77.79279  
## median: 79  
## min: 57 max: 97  
## sd: 9.529969  
## quantile (0.05 - 0.95): 61 92.5  
## skewness: -0.2250959  
  
hist(without_na[,4]) #generating the histogram for column Temp
```

Histogram of without_na[, 4]

