Shahab Geravesh Statistical Computing 206 Lab1

1. Generate 200 random values from the standard exponential distribution and store them in a vector exp.draws.1. Find the mean and standard deviation of exp.draws.1.

```
exp.draws.1 <-rexp(n=200)
```

Generated 200 random values from the standard exponential distribution and stored them in exp.drawa.1 as a vector

```
mean(exp.draws.1)
```

```
## [1] 0.920945
```

mean is 0.9427413 The mean is 0.9427413

```
sd(exp.draws.1)
```

```
## [1] 0.9501598
```

Standard Deviation is .8935911

2.Repeat, but change the rate to 0.1, 0.5, 5 and 10, storing the results in vectors called exp.draws.0.1, exp.draws.0.5, exp.draws.5 and exp.draws.10.

```
exp.draws.0.1 <-rexp(n=200, rate=0.1)
```

```
exp.draws.0.5 <-rexp(n=200, rate=0.5)
```

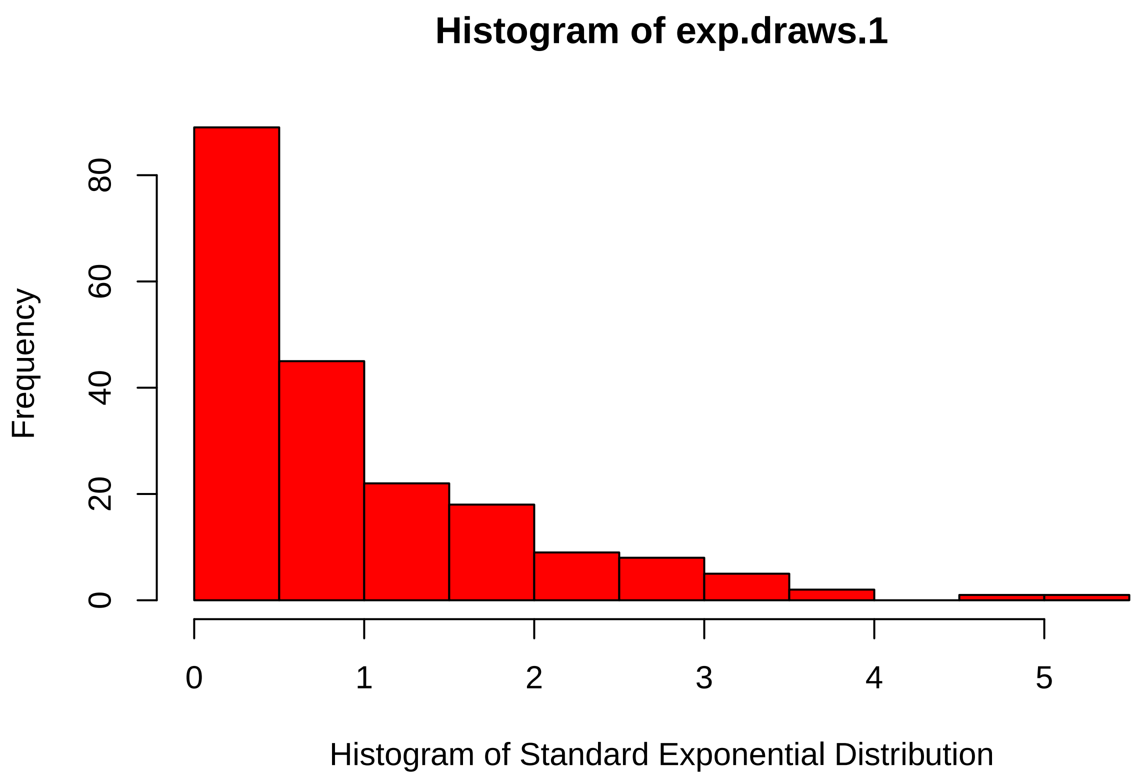
```
exp.draws.5 <-rexp(n=200, rate=5)
```

```
exp.draws.10 <-rexp(n=200, rate=10)
```

3. The function plot() is the generic function in R for the visual display of data. hist() is a function that takes in and bins data as a side effect. To use this function, we must first specify what we'd like to plot.

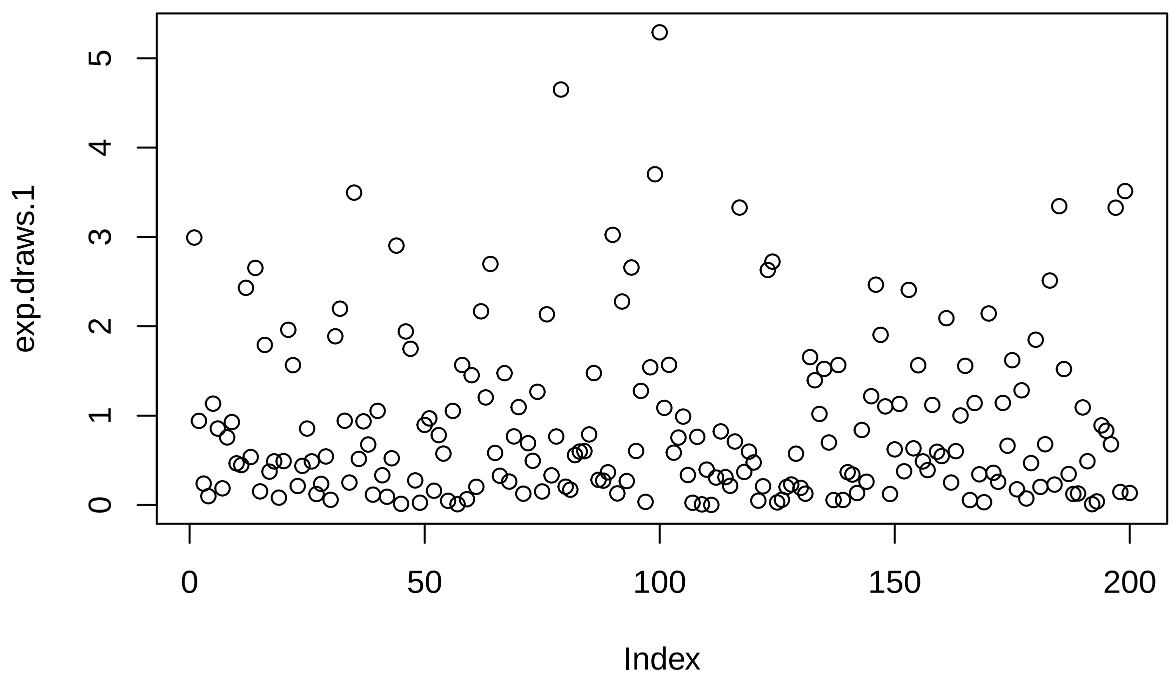
a. Use the hist() function to produce a histogram of your standard exponential distribution

```
hist(exp.draws.1, col='red', border='black', xlab='Histogram of Standard Exponential Distribution', ylal
```



3.b. Use plot() with this vector to display the random values from your standard distribution in order.

```
plot(exp.draws.1)
```

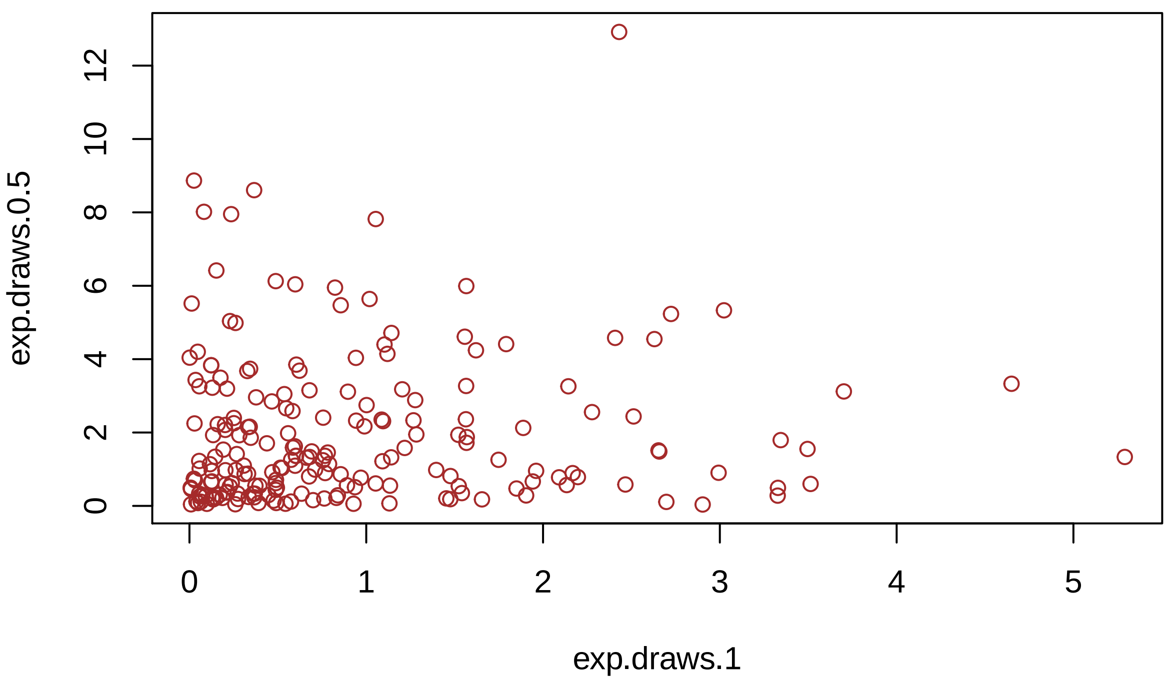


Now, use plot() with two arguments any two of your other stored random value vectors – to create a scatterplot of the two vectors against each other.

```
plot(exp.draws.1, exp.draws.0.5, xlabel='exp.draws.1', ylabel='exp.draws.0.5', col='brown')
```

```
## Warning in plot.window(...): "xlabel" is not a graphical parameter
```

```
## Warning in plot.window(...): "ylabel" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlabel" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ylabel" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlabel" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylabel" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlabel" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ylabel" is not a
## graphical parameter

## Warning in box(...): "xlabel" is not a graphical parameter

## Warning in box(...): "ylabel" is not a graphical parameter

## Warning in title(...): "xlabel" is not a graphical parameter

## Warning in title(...): "ylabel" is not a graphical parameter
```

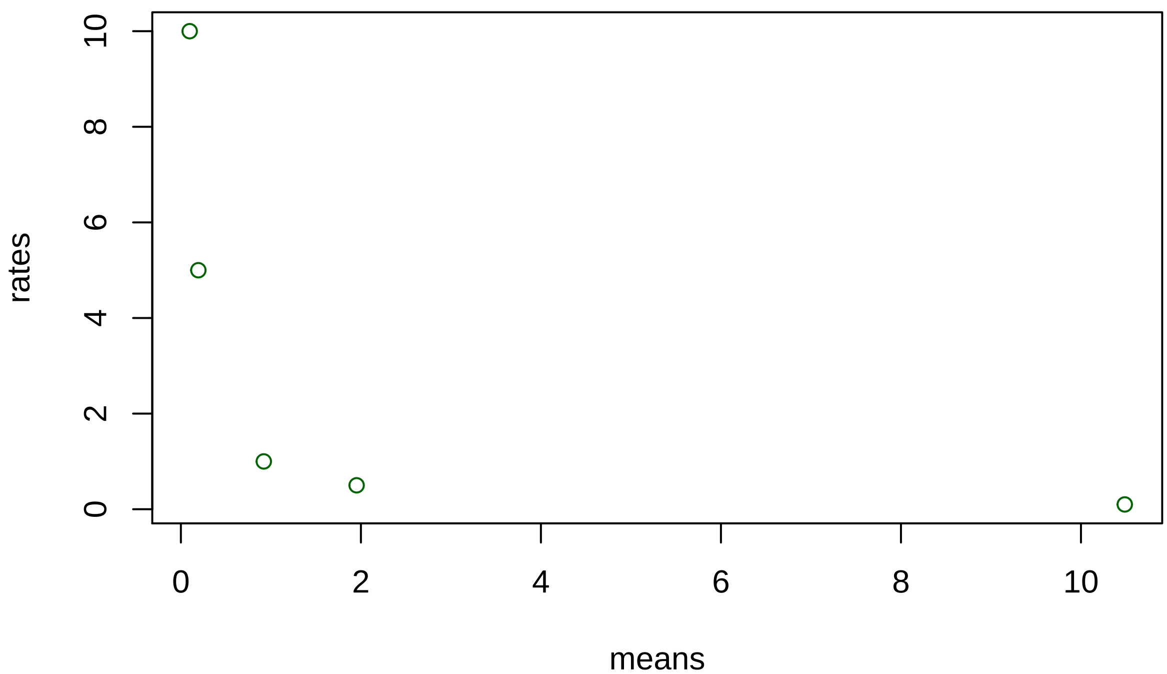


We'd now like to compare the properties of each of our vectors. Begin by creating a vector of the means of each of our five distributions in the order we created them and saving this to a variable name of your choice. Using this and other similar vectors, create the following scatterplots: a. The five means versus the five rates used to generate the distribution.

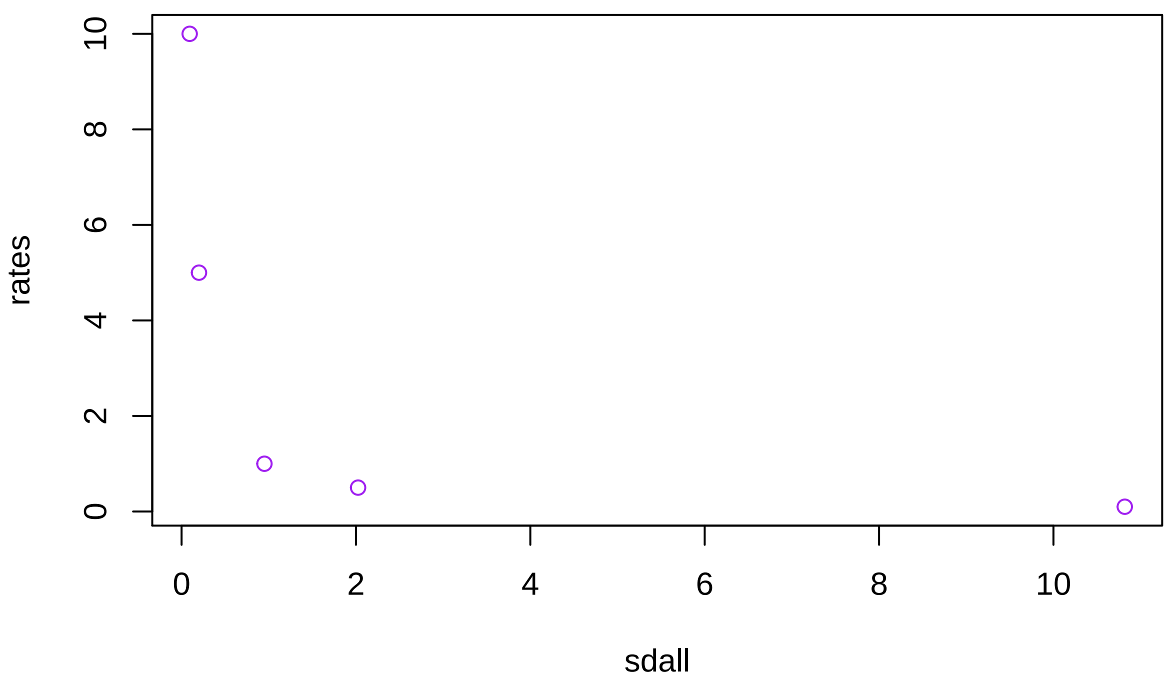
```r
A<-mean(exp.draws.0.1)
B<-mean(exp.draws.0.5)
C<-mean(exp.draws.1)
D<-mean(exp.draws.5)
E<-mean(exp.draws.10)
means<-c(A,B,C,D,E)
```

```r
rates<-c(0.1,0.5,1,5,10)
sdall<-c(sd(exp.draws.0.1), sd(exp.draws.0.5), sd(exp.draws.1), sd(exp.draws.5), sd(exp.draws.10))
```

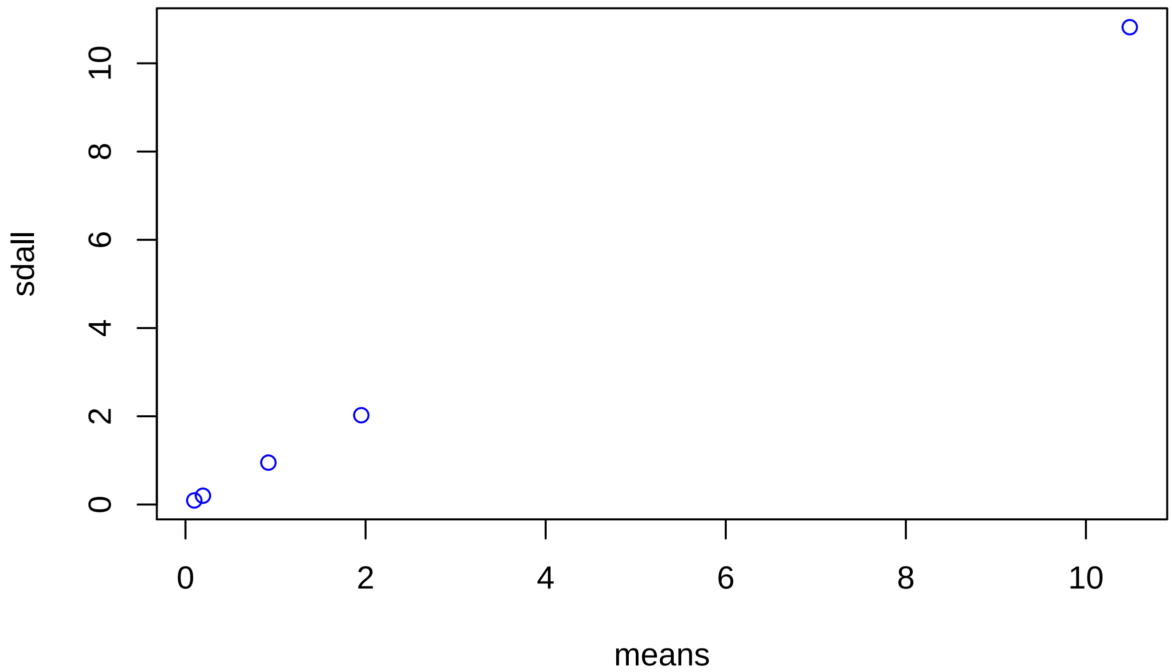
```r
plot(means, rates, col='Dark Green')
```




```r
plot(sdall,rates,col='purple')
```




```r
plot(means, sdall, col='Blue')
```



Part 2

a. To show this, generate 1.1 million numbers from the standard exponential distribution and store them in a vector called big.exp.draws.1. Calculate the mean and standard deviation.

```
draws <- rexp(n=1100000)
mean(draws)
```

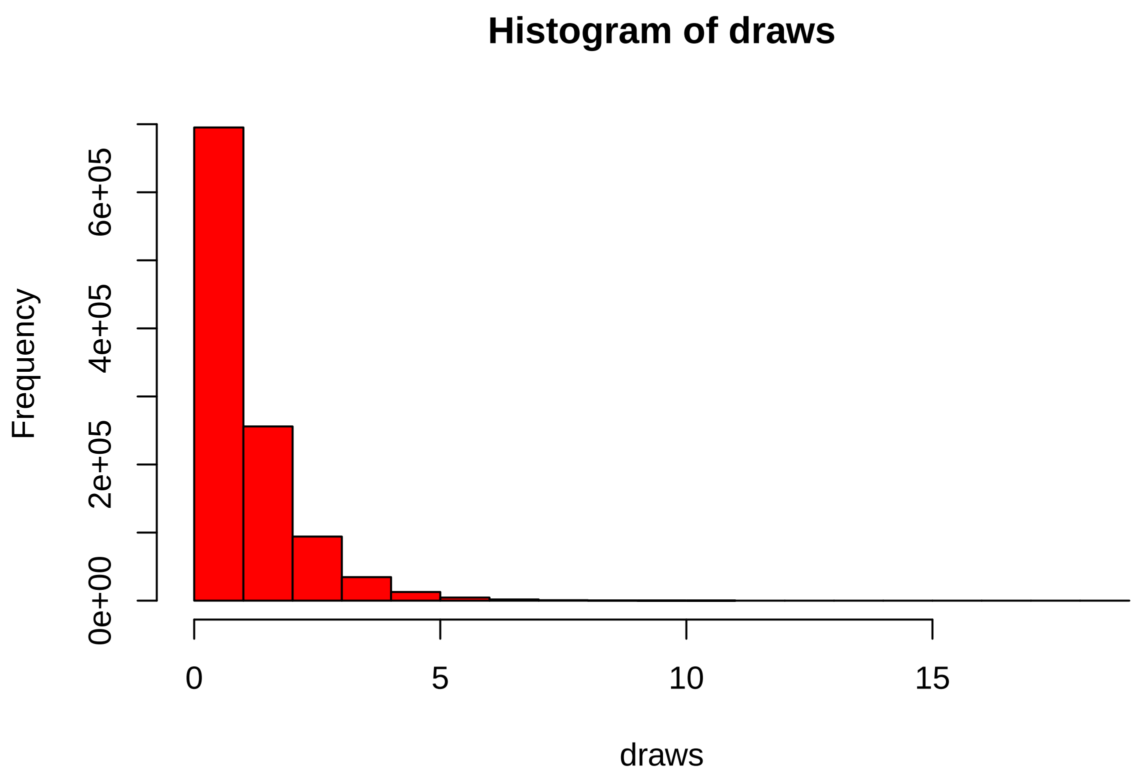
```
## [1] 0.9996841
```

```
sd(draws)
```

```
## [1] 0.9988622
```

b. Plot a histogram of big.exp.draws.1. Does it match the function? Should it?

```
hist(draws,col="red")
```



c. Find the mean of all of the entries in big.exp.draws.1 which are strictly greater than 1. You may need to first create a new vector to identify which elements satisfy this.

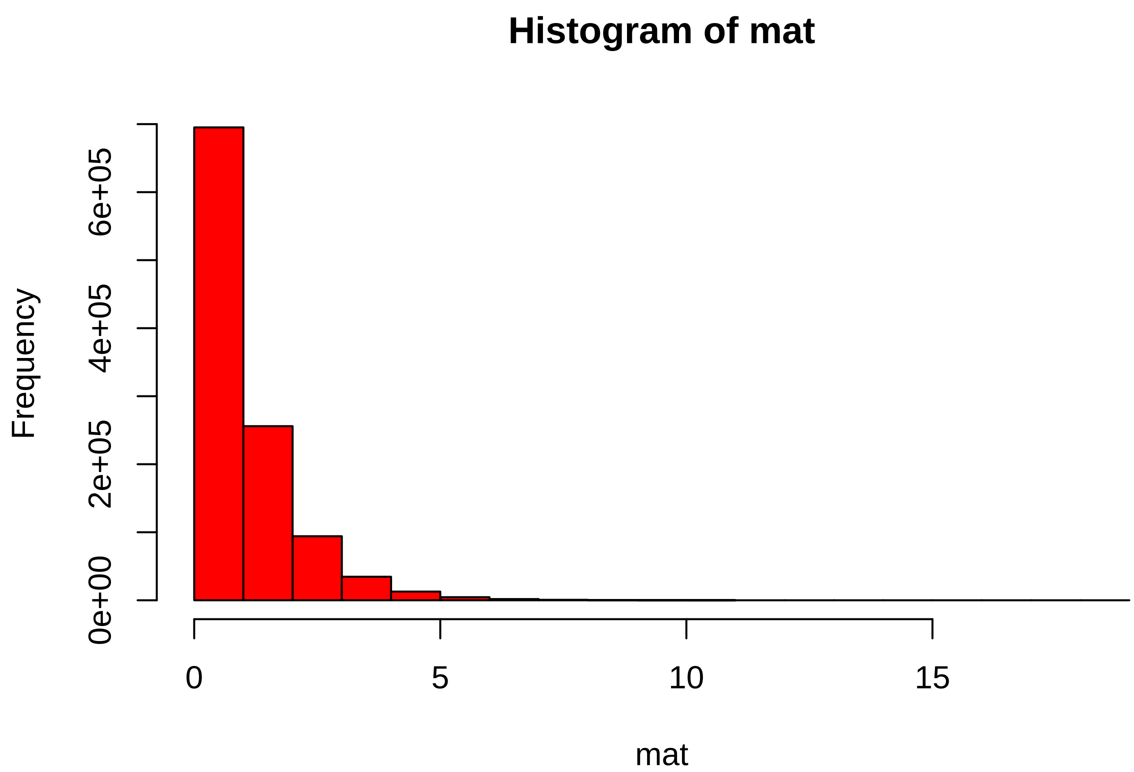
```
mean(which(draws > 1))
```

```
## [1] 549629.2
```

d. Create a matrix, big.exp.draws.1.mat, containing the the values in big.exp.draws.1, with 1100 rows and 1000 columns.Use this matrix as the input to the hist() function and save the result to a variable of your choice. What happens to your data?

```
mat <- matrix(draws, nrow=1100)
```

```
L<-hist(mat, col='red')
```



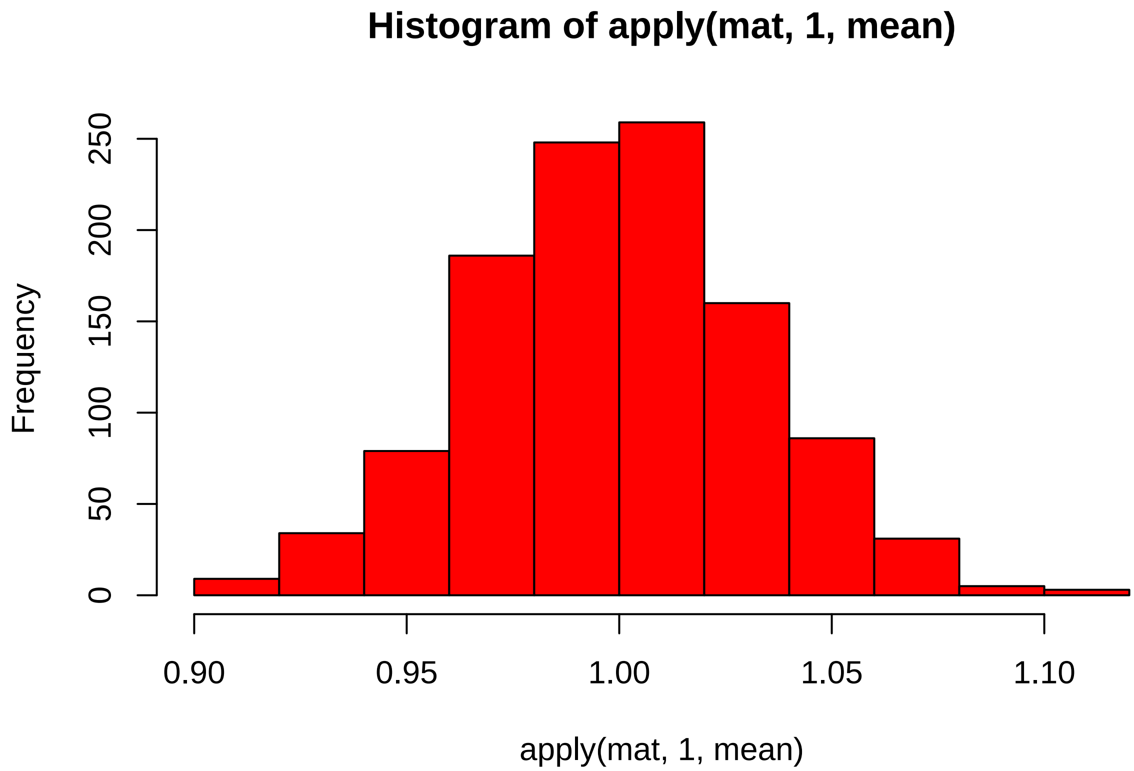
e. Calculate the mean of the 371st column of big.exp.draws.1.mat.

```r
mean(mat[,371])
```

```
## [1] 1.009871
```

f. Now, find the means of all 1000 columns of big.exp.draws.1.mat simultaneously. Plot the histogram of column means. Explain why its shape does not match the histogram in problem 5b).

```r
hist(apply(mat, 1, mean), col='red')
```



g. Take the square of each number in big.exp.draws.1, and find the mean of this new vector. Explain this in terms of the mean and standard deviation of big.exp.draws.1. Hint: think carefully about the formula R uses to calculate the standard deviation.

```
mean(draws*draws)
```

```
## [1] 1.997093
```