PHP2560 Pset #1 (Week 02 Assignment)

Part 1:

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.

```
#Generate 200 random values from the standard exponential dist.
#standard exponential distribution has lambda = 1.
exp.draws.1 = rexp(200)
mean_1 = mean(exp.draws.1)
sd_1 = sd(exp.draws.1)
```

The mean and standard deviation are 0.9925945 and 0.9034966 respectively.

2. Repeat, but change the rate to 0.2, 5, 7.3 and 10, storing the results in vectors called exp.draws.0.2, exp.draws.5, exp.draws.7.3 and exp.draws.10.

```
#Repeat for rates lambda = 0.2
exp.draws.0.2 = rexp(200, rate = 0.2)
mean_0.2 = mean(exp.draws.0.2)
sd_0.2 = sd(exp.draws.0.2)

#Lambda = 5
exp.draws.5 = rexp(200, rate = 5)
mean_5 = mean(exp.draws.5)
sd_5 = sd(exp.draws.5)

#Lambda = 7.3
exp.draws.7.3 = rexp(200, rate = 7.3)
mean_7.3 = mean(exp.draws.7.3)
sd_7.3 = sd(exp.draws.7.3)

#Lambda = 10
exp.draws.10 = rexp(200, rate = 10)
mean_10 = mean(exp.draws.10)
sd_10 = sd(exp.draws.10)
```

The mean and standard deviation for λ = 0.2 are 5.1235395 and 5.4512669 respectively. The mean and standard deviation for λ = 5 are 0.1857091 and 0.1885386 respectively. The mean and standard deviation for λ = 7.3 are 0.1342679 and 0.119585 respectively. The mean and standard deviation for λ = 10 are 0.1048845 and 0.0965023 respectively.

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

```
#Histogram of standard exponential distribution
hist(exp.draws.1)

#Plot values generated by the standard exponential dist. in the order they were genera
ted
plot(exp.draws.1)

#Scatter plot between the standard distribution and the exponential distribution with
lambda = 10
plot(exp.draws.1, exp.draws.10)
```

4. 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 and explain in words what is going on: a. The five means versus the five rates used to generate the distribution. b. The standard deviations versus the rates. c. The means versus the standard deviations.

For each plot, explain in words what's going on.

```
#Scatterplots from mean vector of previous distribution
mean_vector = c(mean_1, mean_0.2, mean_5, mean_7.3, mean_10)
sd_vector = c(sd_1, sd_0.2, sd_5, sd_7.3, sd_10)
rate_vector = c(1, 0.2, 5, 7.3, 10)
plot(rate_vector, mean_vector) #part a
plot(rate_vector, sd_vector) #part b
plot(sd_vector, mean_vector) #part c
```

Part 2:

5. R's capacity for data and computation is large to what was available 10 years ago. 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. b. Plot a histogram of big.exp.draws.1. Does it match the function (1-e^{-x})? Should it? 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. 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? e. Calculate the mean of the 371st column of big.exp.draws.1.mat. 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).

```
#Part A:
big.exp.draws.1 = rexp(1100000)
big_mean = mean(big.exp.draws.1)
big_sd = sd(big.exp.draws.1)
```

```
#Part B:
hist(big.exp.draws.1)
x = seq(0, 15, by = 0.1)
plot(x, 1 - exp(-x))
```

```
#Part C:
greater_than_1_mean = mean(big.exp.draws.1[which(big.exp.draws.1 > 1)])
```

```
#Part D:
big.exp.draws.1.mat = matrix(data = big.exp.draws.1, nrow = 1100, ncol = 1000)
```

```
big_hist = hist(big.exp.draws.1.mat)
```

```
#Part E:
col_371_mean = mean(big.exp.draws.1.mat[, 371])
```

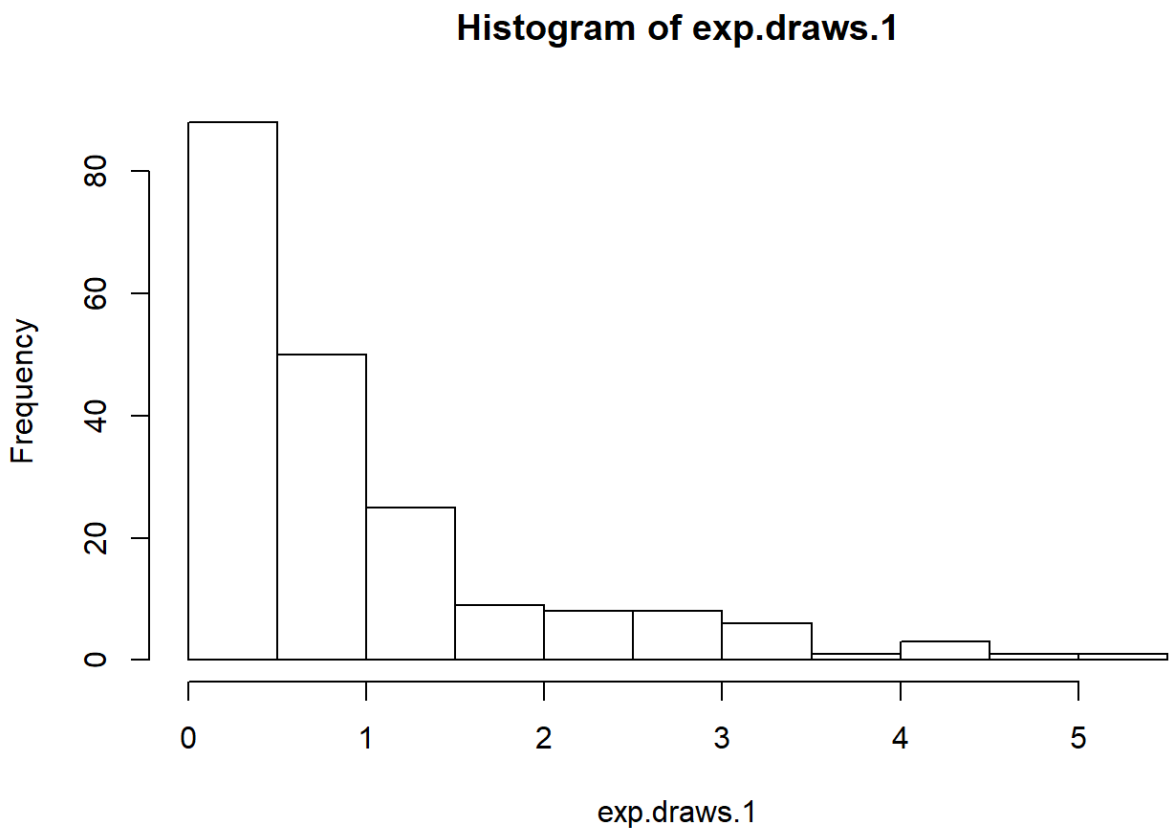
```
#Part F:
col_means = colMeans(big.exp.draws.1.mat)
hist(col_means)
```

The mean of the entire dataset is 1.0001942 and the standard deviation is 0.9988307 of the big dataset. The mean of the entries in the dataset greater than 1 is 1.9988989. The mean of the 371st column is 0.9937327.

Plots and Corresponding Explanations

Part 1:

3a

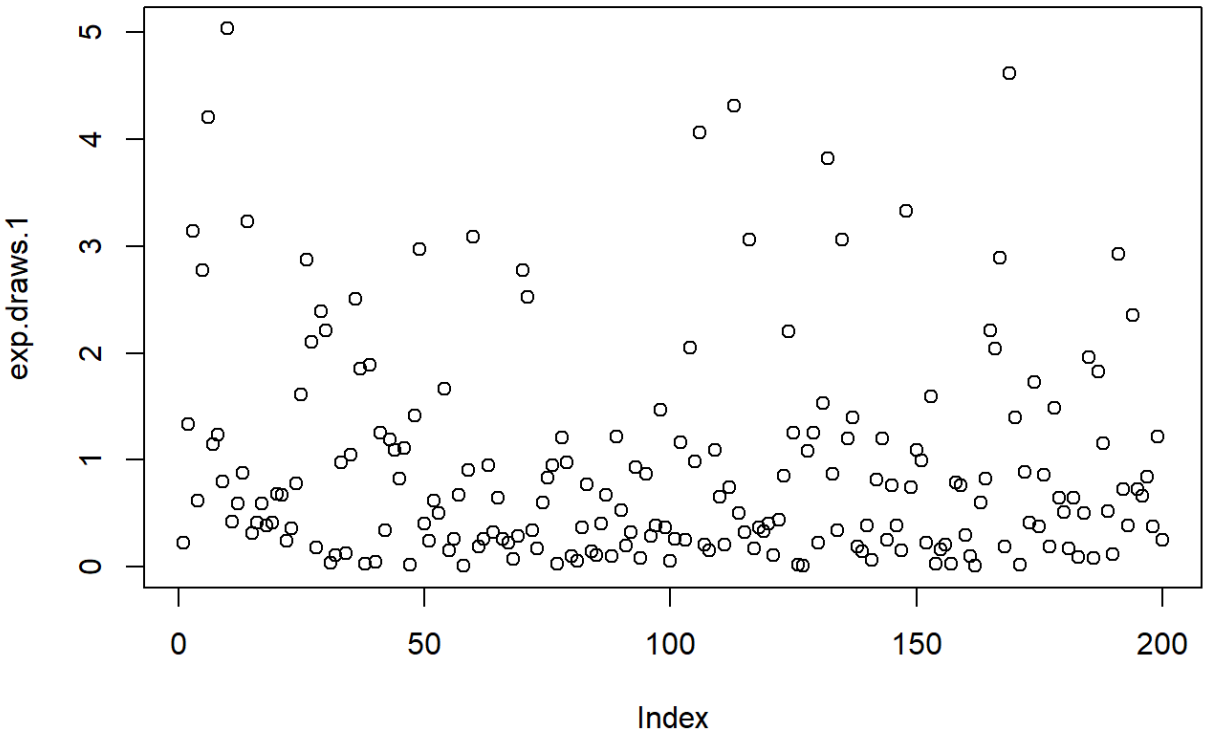


For the histogram of the standard exponential distribution we can see that the the majority of the data is clustered between 0 and 1. This makes sense if we look at the summary of the data. We can see in the summary of the data that the median is 0.6332553. This means that 50% of the data is below 0.6332553 and 50% of the data is greater than 0.6332553. Therefore, in the histogram, it is sensible that 88 out of 200 (44%) randomly generated values would be in the bins between 0 and 1.

I would also like to note that this histogram mirrors what the PDF would look like. This is obvious since the PDF by definition focuses on density, which is exactly what a histogram highlights.

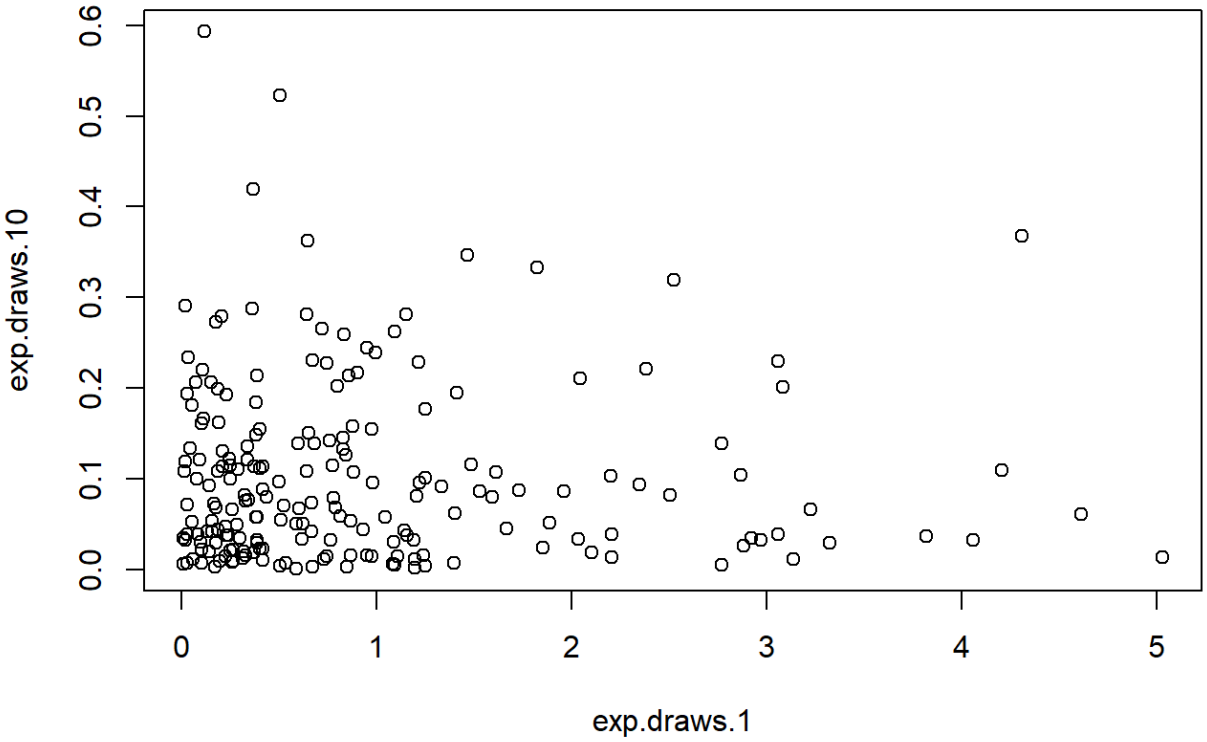
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01046 0.24550 0.63326 0.93737 1.19989 5.02958
```

3b



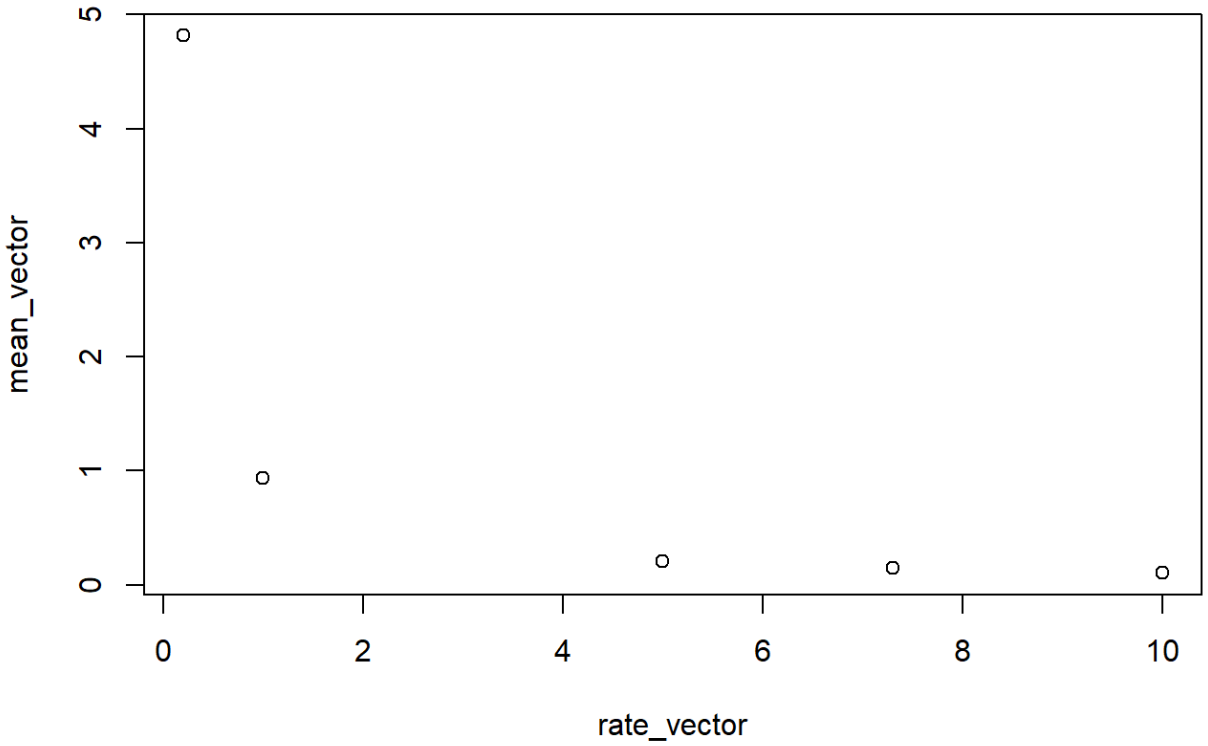
This plot shows that these points were random (rather pseudo-random, but this is a fine point.) If they were not random then there would be a pattern with respect to the index number. This plot also does display some notion of the frequency/density because we can see that there are many points on the lower end of the scale on the y-axis, but few are on the higher end of the scale. To put more simply the graph is very dense on the bottom, which is exactly what is reflected in the histogram.

3c



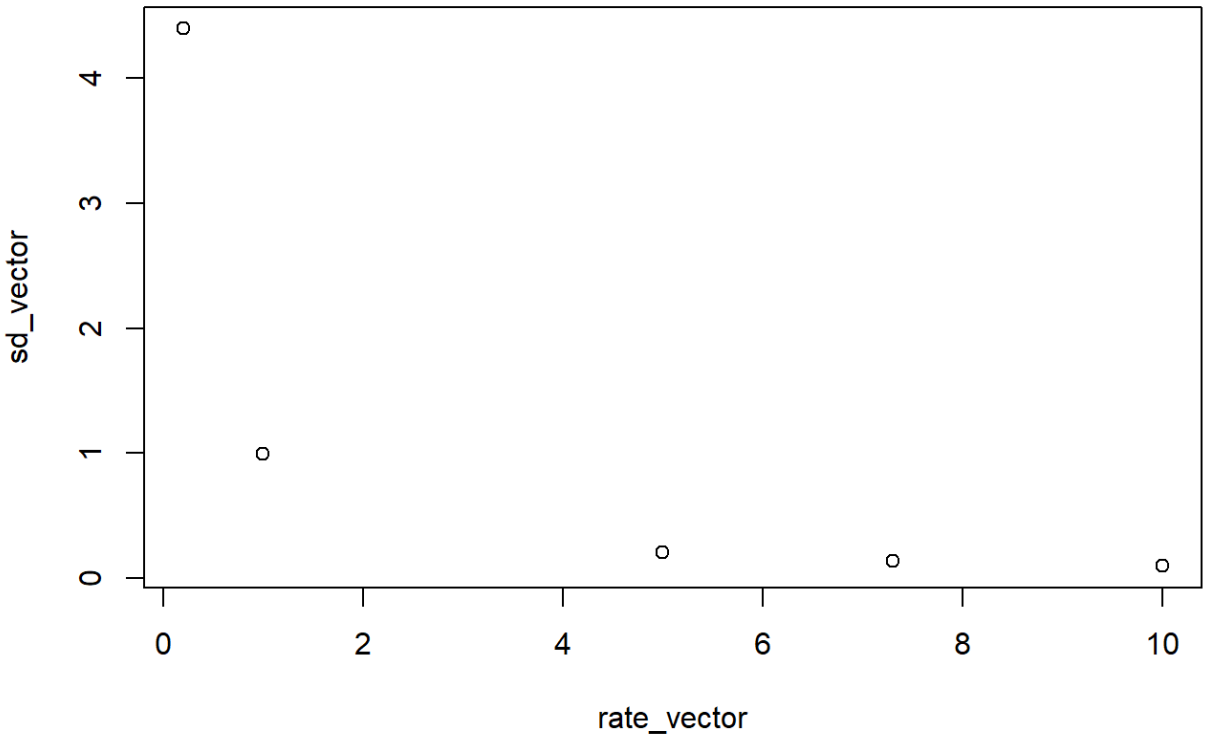
As we can see from the graph, the bulk of the values are concentrated in the bottom left-hand corner. This is because the random values for the standard exponential distribution are likely to be higher than the values for an exponential distribution with $\lambda = 10$ because the probabililty density function for the latter has a much steeper drop than the pdf of the former. See this wiki photo if this is confusing. https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_pdf.svg (https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_pdf.svg)

4a



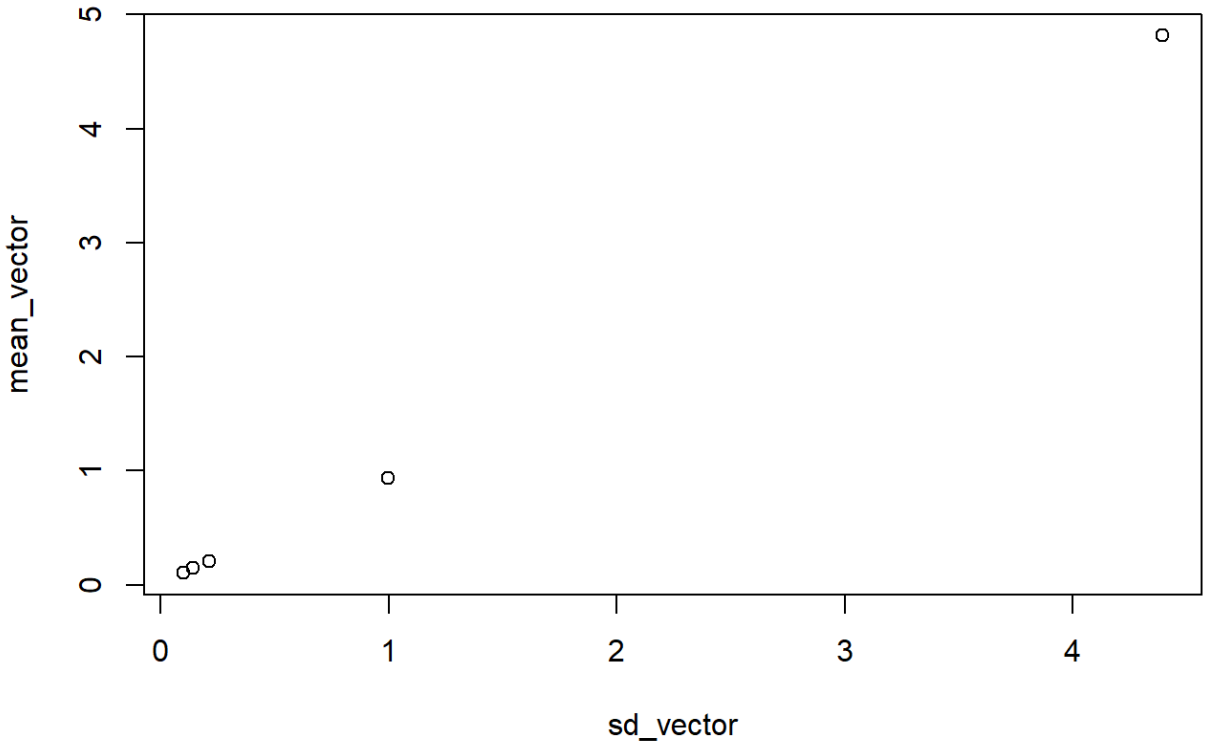
This graph illustrates the relationship between the rate and the mean. As we can see, as the rate goes up the mean goes down. Based on the shape of the pdf of the exponential distribution this is expected since the curve becomes very steep as the rate goes up, which means the frequency of higher x values goes down, thereby dragging the mean down.

4b



This graph shows the relationship between the standard deviation and the rate. As we can see, as the rate increases, the standard deviation decrease. This follows a similar line of logic to the previous graph. Since the pdf has such a steep curve for higher λ values, the range of values it is likely to take is smaller. Therefore, the standard deviation of the pseudo-random numbers that are exponentially distributed will be smaller as the rate goes up.

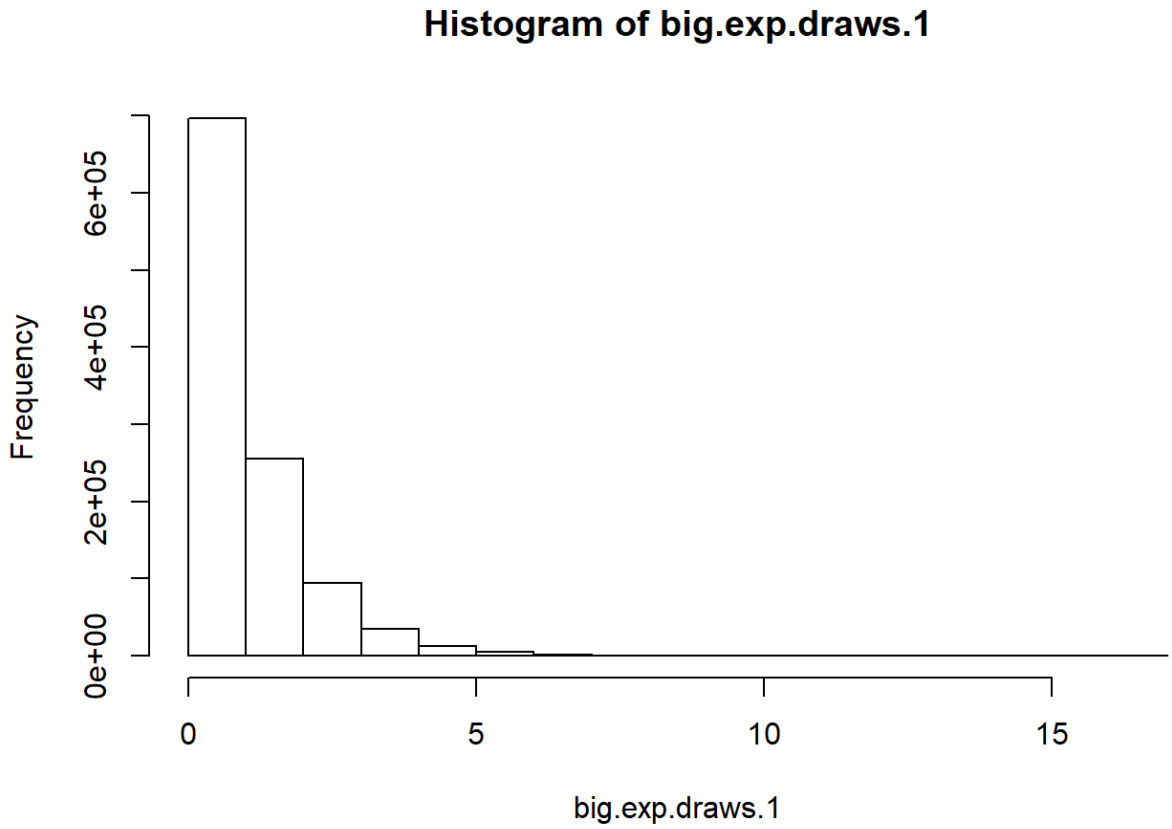
4c



The graph shows the relationship between the standard deviation and the mean for different λ . There is a linear, 1:1 relationship between the mean and the standard deviation. This means that the mean and standard deviation increase proportionally to the inverse of λ . Theoretically the mean is equal to the standard deviation in the exponential distributon, which is why the graph looks this way.

Part 2:

Part 5b

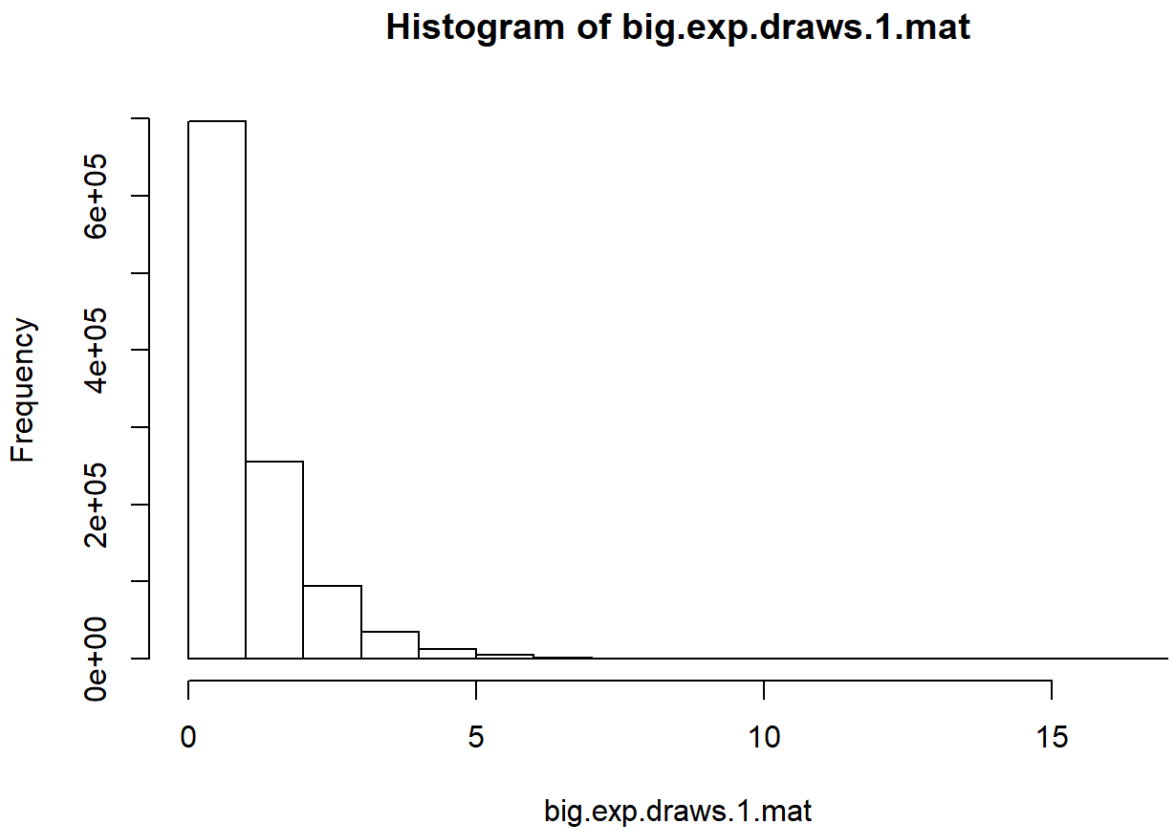


Although, the graph from part b does not look like the graph of the CDF (1- e^{-x}), it communicates the same information. To show this I will give the values of the CDF for x $\in$ [1,10] with λ = 1. If we sum up the the frequencies of all the values up to 2 on the histogram and divide it by 1.1 million, we will get the probability in the third entry in the output below. For example, in this case there are 951050 numbers between 0 and 2. Out of 1.1 million that gives us a percentage of 86.4590909% This gives us almost exactly what we expected from the table of the CDF.

The second part of the question asks if *should* look like the CDF. The answer is no. It should look like the pdf and it in fact does.

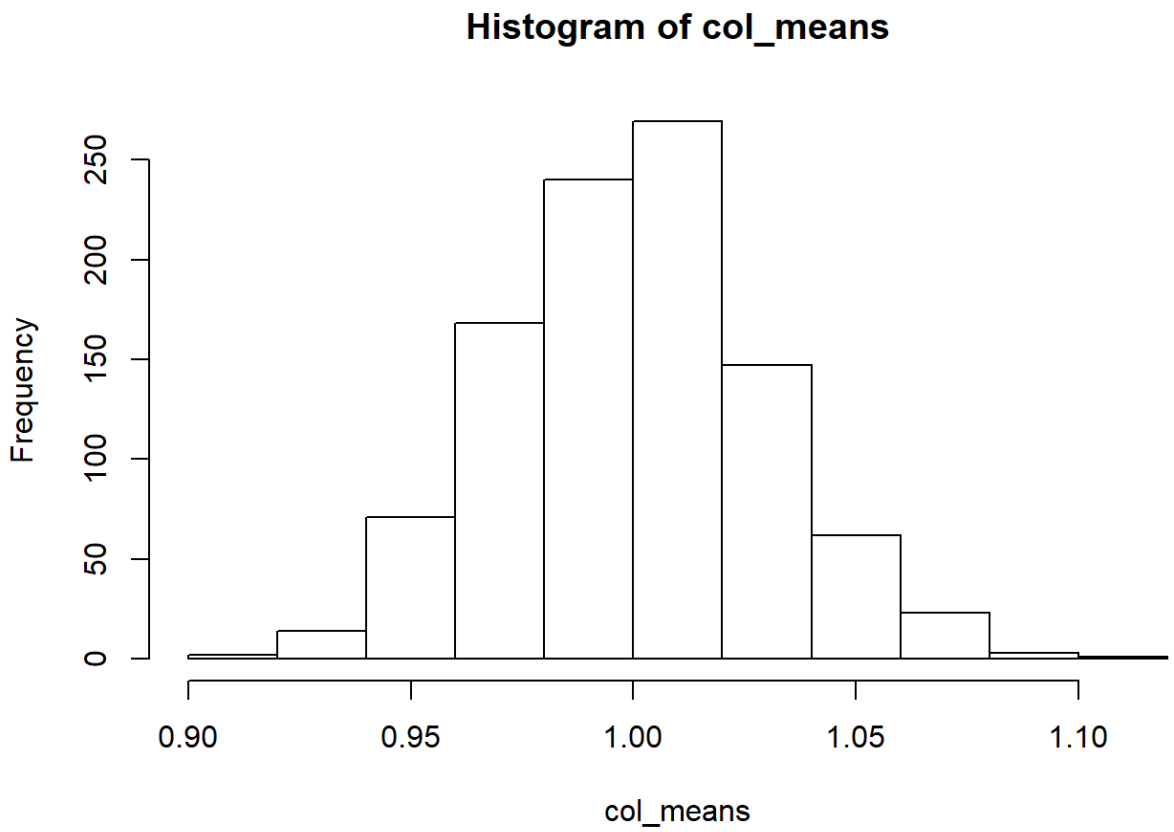
```
##  [1] 0.0000000 0.6321206 0.8646647 0.9502129 0.9816844 0.9932621 0.9975212
##  [8] 0.9990881 0.9996645 0.9998766 0.9999546
```

Part 5d:



When we save the histogram as a variable, in this case big_hist, it becomes a list of 6 different variables: “breaks”, “counts”, “density” “mids”, “xname”, “equidist.”

Part 5f:



The mean of the columns do not necessarily have to follow an exponential distribution because they aren't being sampled from an exponential distribution. The mean of the columns follow a normal distribution due to the central limit theorem. There are 1000 columns and therefore 1000 means. This is sufficiently large to satisfy the conditions for the central limit theorem to apply which would give the distribution of the column means an approximately normal distribution regardless of the fact that the mean was calculated from samples that were *specifically constructed* to be from an exponential distribution.