Lab 1

Shreya Rao sr3843

May 6, 2021

Instructions

Before you leave lab today, try to upload an **RMarkdown** file to the canvas page (this should have a .Rmd extension) as well as the HTML or PDF output after you have knitted the file (this will have a .html or .pdf extension). Note that since you have already knitted this file, you should see both a **Lab1__UNI.html** and a **Lab1__UNI.Rmd** file in your GU4206/GR5206 folder. Click on the **Files** tab to the right to see this. The files you upload to the Canvas page should be updated with commands you provide to answer each of the questions below. You can edit this file directly to produce your final solutions. If you do not finish the lab during the scheduled session, then make sure to submit the file by Tuesday, September 22nd by 11:59pm.

Background: The Normal Distribution

Recall from your probability class that a random variable X is normally-distributed with mean μ and variance σ^2 (denoted $X \sim N(\mu, \sigma^2)$) if it has a probability density function, or *pdf*, equal to

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

In *R* we can simulate $N(\mu, \sigma^2)$ random variables using the `rnorm()` function. For example,

```r
rnorm(n = 5, mean = 10, sd = 3)
```

```
## [1]  8.120639 10.550930  7.493114 14.785842 10.988523
```

outputs 5 normally-distributed random variables with mean equal to 10 and standard deviation (this is σ) equal to 3. If the second and third arguments are ommited the default rates are **mean = 0** and **sd = 1**, which is referred to as the "standard normal distribution".

Tasks

Sample means as sample size increases

1) Generate 100 random draws from the standard normal distribution and save them in a vector named **normal100**. Calculate the mean and standard deviation of **normal100**. In words explain why these values aren't exactly equal to 0 and 1.

```r
normal100 <- rnorm(n = 100)
mean(normal100)
```

```
## [1] 0.08256659
```

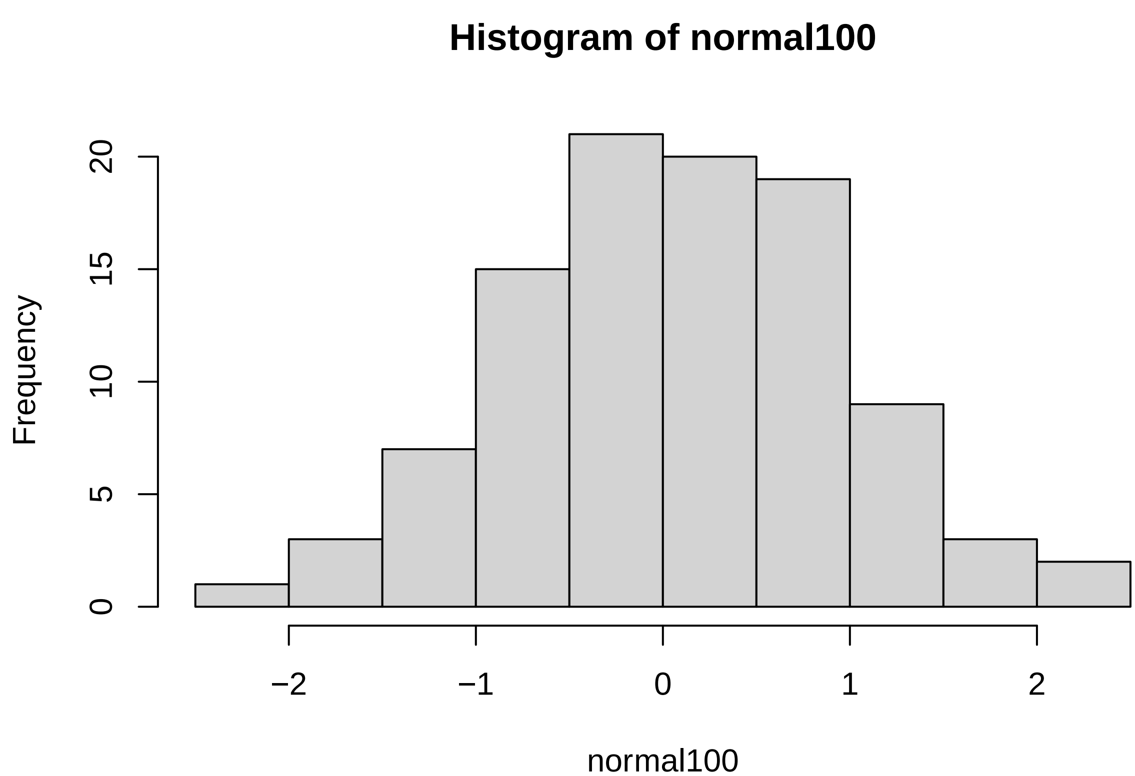
```r
sd(normal100)
```

```
## [1] 0.8891336
```

The values aren't exactly equal to 0 and 1 because the sample size isn't large enough for the values to form a perfect normal distribution. There is also going to be more variability for smaller datasets.

2) The function **hist()** is a base *R* graphing function that plots a histogram of its input. Use **hist()** with your vector of standard normal random variables from question (1) to produce a histogram of the standard normal distribution. Remember that typing **?hist** in your console will provide help documents for the **hist()** function. If coded properly, these plots will be automatically embedded in your output file.

```r
hist(normal100)
```



3) Repeat question (1) except change the number of draws to 10, 1000, 10,000, and 100,000 storing the results in vectors called **normal10**, **normal1000**, **normal10000**, **normal100000**.

```r
normal10 <- rnorm(n = 100)
normal1000 <- rnorm(n = 10^3)
normal10000 <- rnorm(n = 10^4)
normal100000 <- rnorm(n = 10^5)
```

4) We want to compare the means of our four random draws. Create a vector called **sample_means** that has as its first element the mean of **normal10**, its second element the mean of **normal100**, its third element the mean of **normal1000**, its fourth element the mean of **normal10000**, and its fifth element the mean of **normal100000**. After you have created the **sample_means** vector, print the contents of the vector and use the **length()** function to find the length of this vector. (it should be five). There are, of course, multiple ways to create this vector. Finally, explain in words the pattern we are seeing with the means in the **sample_means** vector.

```r
sample_means <- c(mean(normal10), mean(normal100), mean(normal1000), mean(normal10000),
    mean(normal100000))
sample_means
```

```
## [1] -0.007263591  0.082566589 -0.038948508 -0.004870671 -0.001012609
```

```r
length(sample_means)
```

```
## [1] 5
```

As the sample size increases, the mean values gets closer to 0 (except in the case of normal10).

Sample distribution of the sample mean

5) Let's push this a little farther. Generate 1 million random draws from a normal distribution with $\mu = 3$ and $\sigma^2 = 4$ and save them in a vector named **normal1mil**. Calculate the mean and standard deviation of **normal1mil**.

```r
normal1mil <- rnorm(n = 10^6, mean = 3, sd = 2)
mean(normal1mil)
```

```
## [1] 3.000214
```

```r
sd(normal1mil)
```

```
## [1] 1.999953
```

6) Find the mean of all the entries in **normal1mil** that are greater than 3. You may want to generate a new vector first which identifies the elements that fit the criteria.

```r
mean(normal1mil[normal1mil > 3])
```

```
## [1] 4.596143
```

7) Create a matrix **normal1mil_mat** from the vector **normal1mil** that has 10,000 columns (and therefore should have 100 rows).

```r
normal1mil_mat <- matrix(normal1mil, ncol = 10000, nrow = 100)
```

8) Calculate the mean of the 1234^{th} column.

```r
mean(normal1mil_mat[, 1234])
```

```
## [1] 2.735641
```

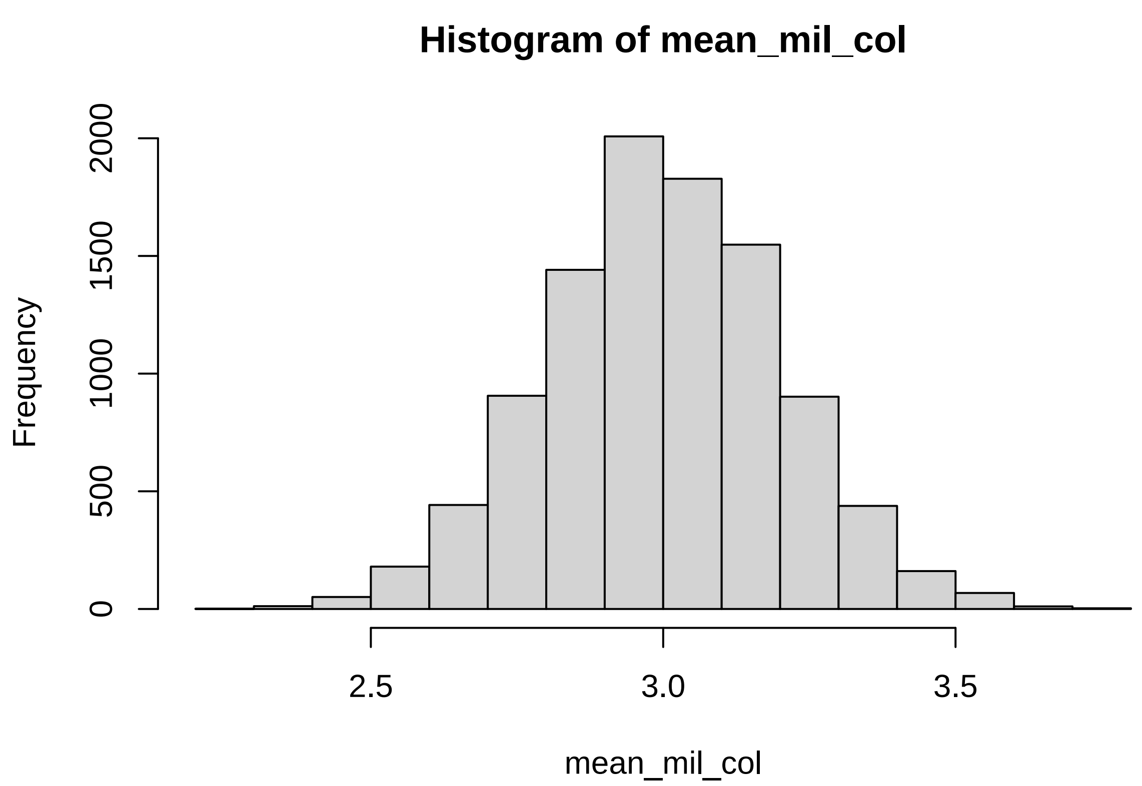
9) Use the **colSums()** functions to calculate the *means* of each column of **normal1mil_mat**. Remember, **?colSums** will give you help documents about this function. Save the vector of column means with an appropriate name as it will be used in the next task.

```r
mean_mil_col <- colSums(normal1mil_mat)/nrow(normal1mil_mat)
head(mean_mil_col)
```

```
## [1] 2.814381 3.045045 2.867702 3.163020 2.900292 3.238833
```

10) Finally, produce a histogram of the column means you calculated in task (9). What is the distribution that this histogram approximates (i.e. what is the distribution of the sample mean in this case)?

```r
hist(mean_mil_col)
```



The histogram approximates a normal distribution.

11) Let's push this even farther. Generate 10 million random draws from an exponential distribution with rate parameter $\lambda = 3$ (**Hint: ?rexp**). Save the simulated draws in a vector named **exp_10mil**. Calculate the mean and standard deviation of **exp_10mil**. How do these numbers compare to $E(X) = 1/3$ and $sd(X) = 1/3$?

```
exp_10mil <- rexp(10^7, rate = 3)
mean(exp_10mil)
```

```
## [1] 0.3333224
```

```
sd(exp_10mil)
```

```
## [1] 0.3332437
```

The mean and sd of the simulated draws almost equal to 1/3 each (or 1/lambda).

12) Create a matrix **exp10mil_mat** from the vector **exp_10mil** that has 10,000 columns (and therefore should have 100 rows). Use the **colMeans()** function to calculate the *means* of each column of **exp_mil_mat**. Show the first 10 computed means.

```
exp_10mil_mat <- matrix(exp_10mil, ncol = 10000)
temp <- colSums(exp_10mil_mat)/nrow(exp_10mil_mat)
mean_exp_10mil <- colMeans(exp_10mil_mat)
head(mean_exp_10mil)
```

```
## [1] 0.3387260 0.3303376 0.3253542 0.3250035 0.3314085 0.3302435
```

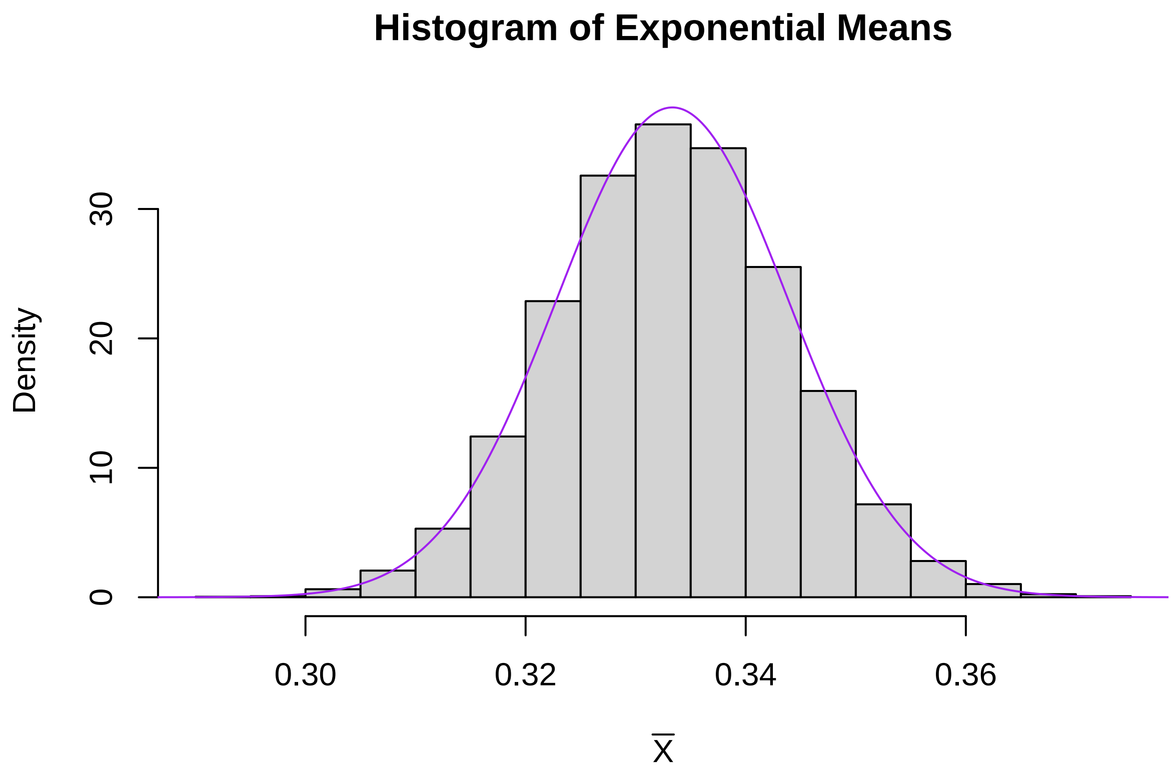
13) Finally, produce a histogram of the column means you calculated in task (12). What is the approximate distribution that this histogram displays (i.e. what is the distribution of the sample mean in this case)? Overlay the true approximate density function over the histogram. **Note:** the correct code is displayed below.

```
# hist(exp_means, main='Histogram of Exponential Means', xlab=expression(bar(X)),
# prob = T, breaks=20) n <- nrow(exp10mil_mat) mean_exp <- 1/3 mean_exp sd_exp <-
# 1/(3*sqrt(n)) sd_exp x <- seq(0,1,by=.0001) my_density <-
# dnorm(x,mean=mean_exp,sd=sd_exp) lines(x,my_density,col='purple')

hist(mean_exp_10mil, main = "Histogram of Exponential Means", xlab = expression(bar(X)),
    prob = T, breaks = 20)

n <- nrow(exp_10mil_mat)
mean_exp <- 1/3
sd_exp <- 1/(3 * sqrt(n))
x <- seq(0, 1, by = 1e-04)

my_density <- dnorm(x, mean = mean_exp, sd = sd_exp)
lines(x, my_density, col = "purple")
```


Histogram of Exponential Means
Density
0
10
20
30
0.30
0.32
0.34
0.36
$\overline{X}$