Lab 2

Salim M'jahad msm2243

January 25, 2018

Instructions

Before you leave lab today make sure that you upload an RMarkdown file to the canvas page (this should have a .Rmd extension) as well as the PDF or HTML output after you have knitted the file (this will have a .pdf or .html extension). Note that since you have already knitted this file, you should see both a **Lab2_UNI.pdf** and a **Lab2_UNI.Rmd** file in your UN2102 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.

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
# You'll want to type your response here.  Your response should look like:
normal100 <- rnorm(n = 100, mean = 0, sd = 1)
mean(normal100)
```

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

```r
sd(normal100)
```

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

Because we have not taken a sample big enough to perfectly represent the distribution. As n tends to infinity, the mean will tend to 0 and the sd will tend to 1.

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.

```
hist(normal100)
```
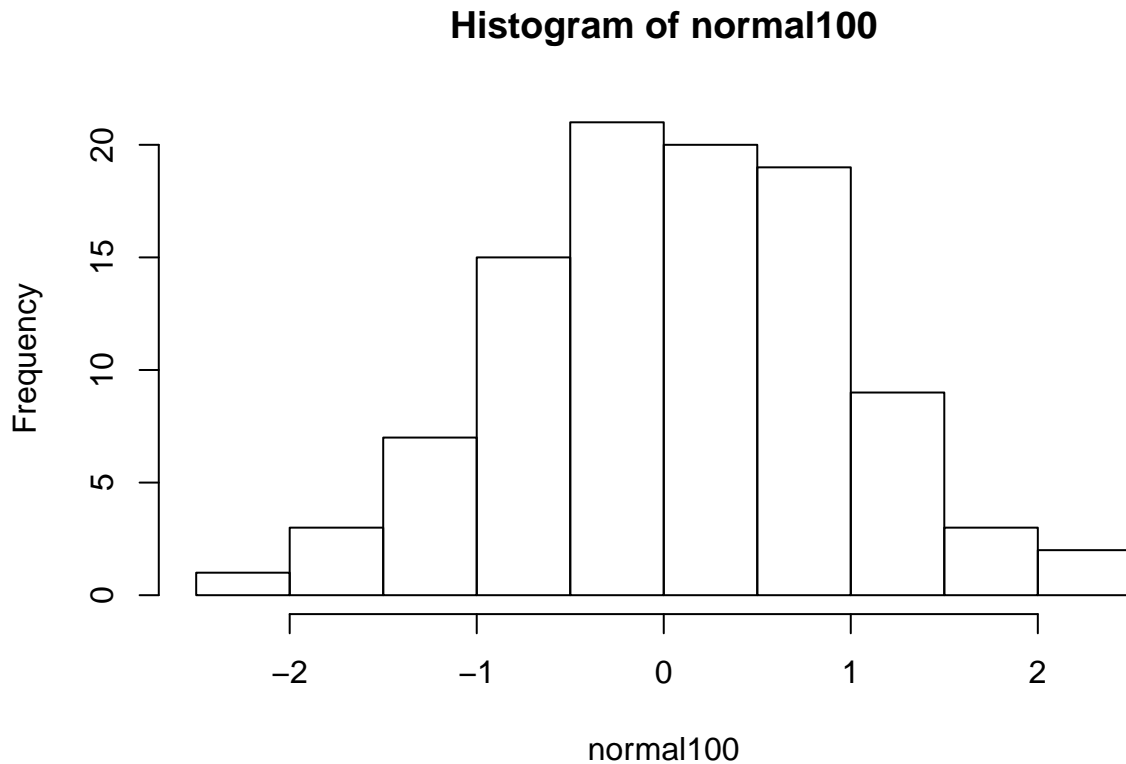

3) Repeat question (1) except change the number of draws to 10, 1000, 10,000, and 100,000 storing the results in vectors called **normal10**, **normal1000**, **normal10000**, **normal100000**.

```
normal10 <- rnorm(n = 10, mean = 0, sd = 1)
normal1000 <- rnorm(n = 1000, mean = 0, sd = 1)
normal10000 <- rnorm(n = 10000, mean = 0, sd = 1)
normal100000 <- rnorm(n = 100000, mean = 0, sd = 1)
```

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.

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

```
## [1]  0.493735437  0.082566589 -0.026875723 -0.006719807 -0.001114476
```

```
length(sample_means)
```

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

The pattern is the bigger the sample the closer the mean is to 0.

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**.

```
normal1mil <- rnorm(n = 1000000, mean = 3, sd = 2)
mean(normal1mil)
```

```
## [1] 3.000241
```

```
sd(normal1mil)
```

```
## [1] 1.999961
```

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.

```
greaterthan3 <- normal1mil[normal1mil>3]
mean(greaterthan3)
```

```
## [1] 4.596158
```

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

```
normal1mil_mat <- matrix(normal1mil ,ncol = 10000)
dim(normal1mil_mat)
```

```
## [1]   100 10000
```

8) Calculate the mean of the 1234th column.

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

```
## [1] 3.056545
```

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.
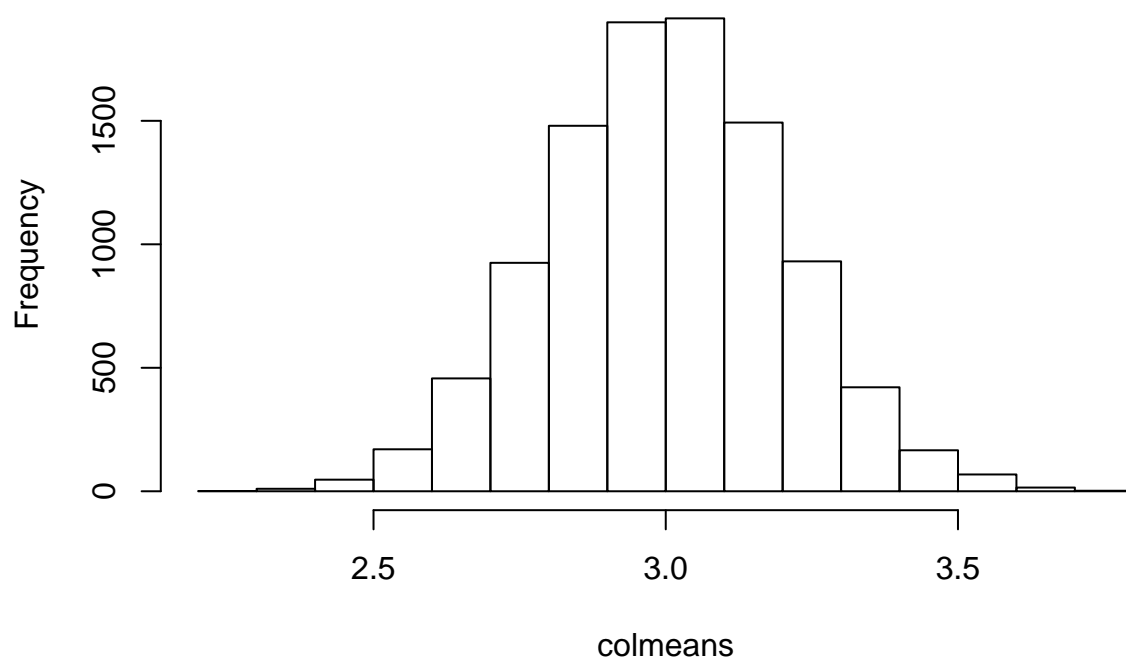
```
colmeans <- colMeans(normal1mil_mat)
summary(colmeans)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.251   2.865   3.001   3.000   3.136   3.746
```

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)?

```
hist(colmeans)
```

Histogram of colmeans


```r
sd(colmeans)
```

```
## [1] 0.2011195
```

```r
mean(colmeans)
```

```
## [1] 3.000241
```

As one would expect (from distribution of mean of distribution formula), we get the same mean (3) and a SD of 2/sqrt(100)=0.2