Lab 1

Name: Yi Chen, UNI: yc3356

September 19, 2017

Instructions

Before you leave lab today make sure that you upload a .pdf file to the canvas page (this should have a .pdf extension). This should be the PDF output after you have knitted the file, we don't need the .Rmd file (don't upload the one with the .Rmd extension). Note that since you have already knitted this file, you should see both a **Lab1_UNI.pdf** and a **Lab1_UNI.Rmd** file in your GR5206 folder. Click on the **Files** tab to the right to see this. The file 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. Note, however, in the file you upload you should the above header to have the date, your name, and your UNI. Similarly, when you save the file you should replace **UNI** with your actualy UNI.

Please feel free to work together in groups to get this done. The actual work itself is not really the important part, but rather I want to make sure that you are confortable with the process. At the end of lab you should:

- Have `R` and `R` Studio downloaded.
- Understand how to create and knit to PDF an `R` Markdown document.
- Be able to upload your solution to the Courseworks page.

Background: The Normal Distribution

Recall from your probability class that a random variable X has a normal distribution 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 function. For example,

```
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.

```
normal100 <- rnorm(n = 100, mean = 0, sd = 1) # create 100 random draws from the standard normal distribution
mean_of_normal100 <- mean(normal100) # calculate the mean of normal100
standard_deviation_of_normal100 <- sd(normal100) #calculate the standard deviation of normal100
result <- list(mean_of_normal100,standard_deviation_of_normal100)
whether_equal <- list(mean_of_normal100==0,standard_deviation_of_normal100==1)
result
```

```
## [[1]]
## [1] 0.08256659
##
## [[2]]
## [1] 0.8891336
```

```
whether_equal
```

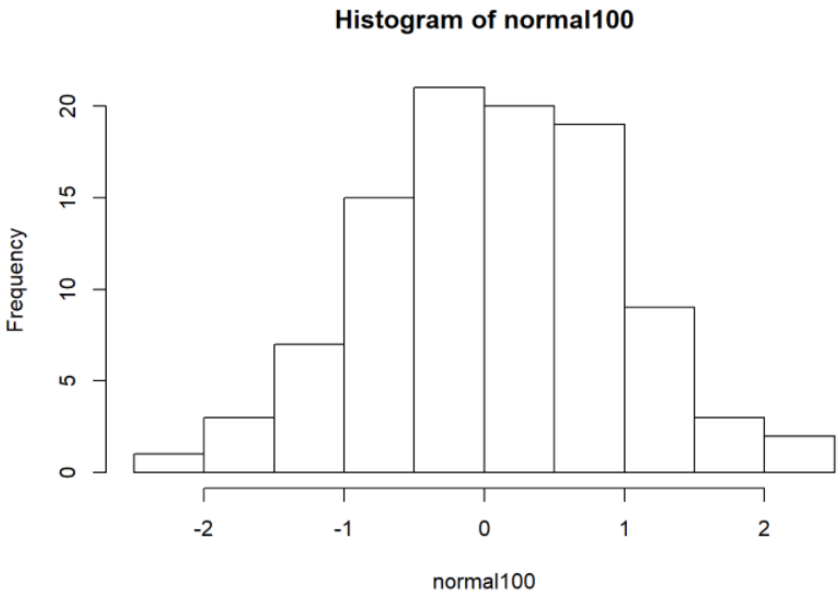
```
## [[1]]
## [1] FALSE
##
## [[2]]
## [1] FALSE
```

the reason:

- clearly the nornam100 is a list of random variable, we generate these numbers randomly. It is likely that the mean of these numbers would have some small difference between the true mean(0) and standard deviation(1) due to the randomness of the sample.
- if we generate more variables in the sample, say 100000000 rather than just 100. the mean and statndard deviation are more likely to be 0 and 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)
```



- as you can see these numbers follow basically the standard normal distribution, even through there have small difference. It is because of the randomness of the sample and relatively small size of sample.

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**.

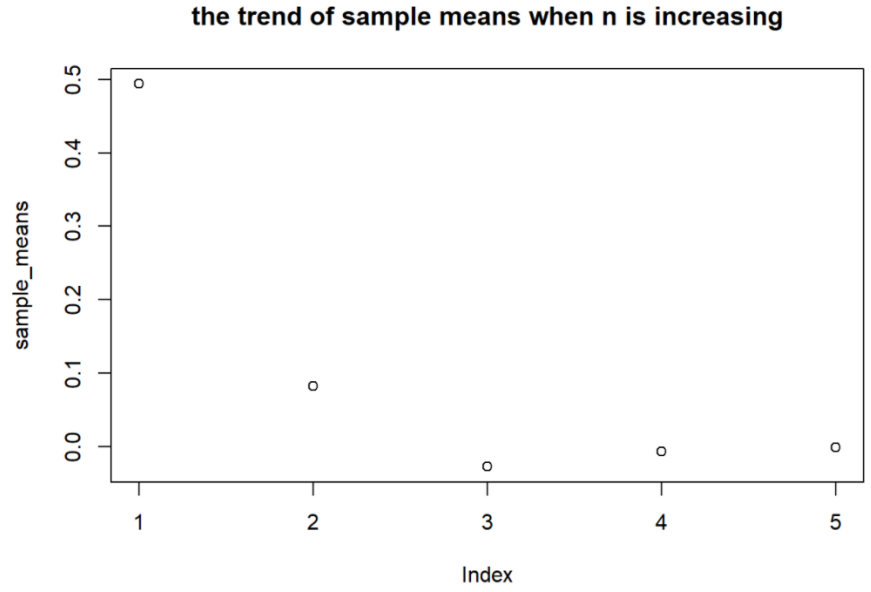
```
normal10 <- rnorm(n=10,mean = 0,sd = 1)
normal1000 <- rnorm(n=1000,mean = 0,sd=1)
normal10000 <- rnorm(n=10000,mean = 0,sd=1)
normal100000 <- rnorm(n=100000,mean=0,sd=1)
```

4. We want to compare the means of our 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))
sample_means
```

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

```
plot(sample_means,main = 'the trend of sample means when n is increasing')
```



the reason:

- according to the law of large number: when the sample size is increasing, the distribution is more likely to follow the normal distribution. And as a result, the mean is closer to 0 and standard deviation is closer to 1.