Lab 1

Fan Yang

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 `rnorm()` 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.

```r
normal100 <- rnorm(n = 100, mean = 0, sd = 1)
mean(normal100)
```

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

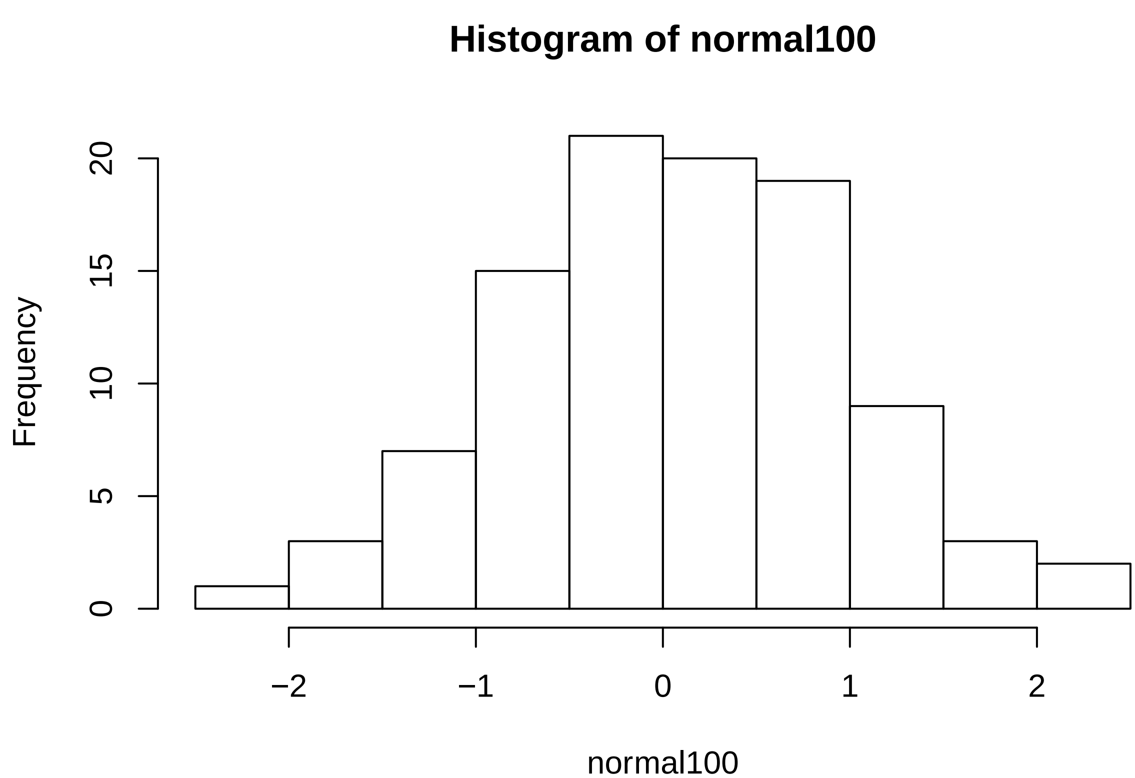
```r
sd(normal100)
```

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

Because the sample are selected randomly, and these practical data may have small deviation with the theoretical mean and standard deviation.

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 = 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(normal10000
length(sample_means)
```

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

the elements in **sample_means** are decreasing with index, which means that the sample mean is approching 0 as sample size increasing. This also proves The Law of Large Numbers in practical.