Statistical Inference Course Project: Exploring the Central Limit Theorem

By Matthew Sedlar

Overview

This report explores the Central Limit Theorem using a set of randomly generated exponentials. It will look at the sample mean and how it relates to the theoretical mean, the sample variance and how it relates to the theoretical variance, and whether the resulting distribution approximates normal.

Simulations

I'm going to run 1,000 simulations on 40 randomly generated exponentials with a rate of 0.2 using R's replicate function. Once I have the simulation results in the form of a matrix, I will run the mean function on the matrix using apply, then store the results in a data frame along with the number of simulations. (See Appendix A for the full code).

Sample Mean versus Theoretical Mean

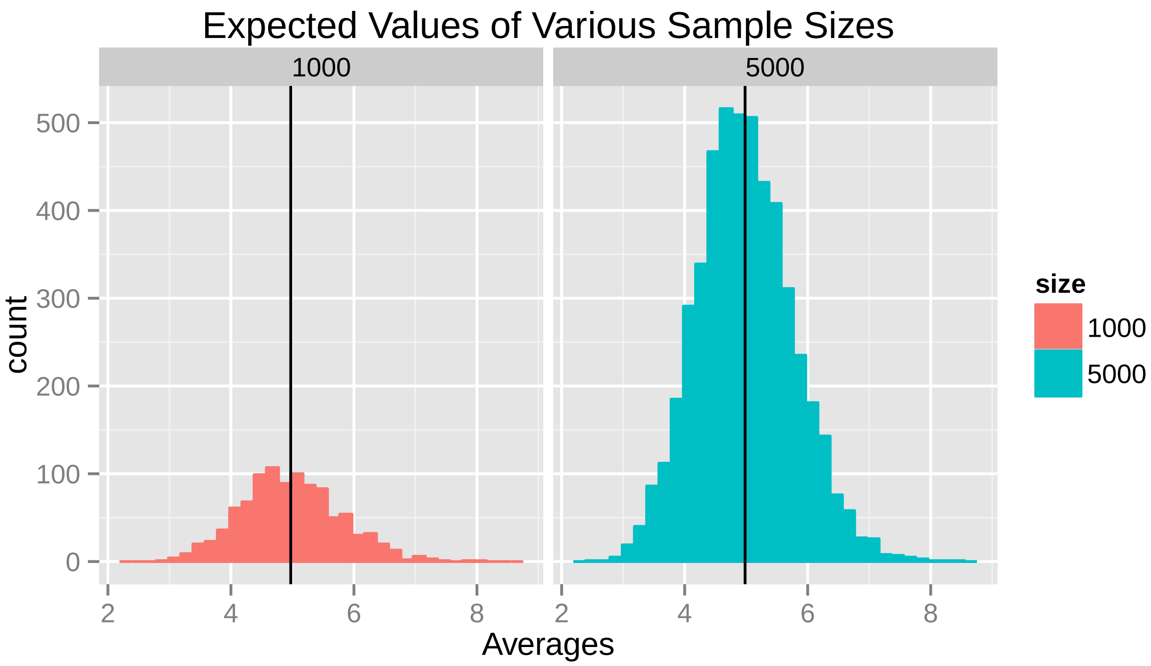
Let's examine the distribution for our simulation (See Appendix B for a plot of the distribution). Our sample mean is:

```
## [1] 4.972512
```

We know given our rate (0.2) that the theoretical mean is 5 based on the following formula:

$$E[X] = \frac{1}{\lambda} = \beta$$

We know from the Central Limit Theorem that the sample mean from unbiased data approximates the population mean. To further illustrate this, I'm going to run another 5,000 simulations and then compare the two distributions (see Appendix C for code).



No matter the sample size, the sample mean approximates the theoretical mean.

Sample Variance versus Theoretical Variance

Next, let's look at our sample variance. Since our mean and our standard deviation are the same, we know our theoretical variance is 0.625 based on the following formula:

$$Var[X] = \frac{1}{\lambda^2}/n$$

We can use that formula or R's var() function to determine our sample variance.

```
## [1] 0.6231669
```

The sample variance shares a similar spread with the theoretical variance.

Distribution

There are a couple of ways to tell if our distribution is normal. Two of them are:

- mean = median = mode
- The distribution is symmetric

These are easy to test.

mean = median = mode.

In R, this is a simple logical comparison. Since we're trying to prove if it is "approximately" normal, let's use the signif function on our values to the first decimal point. (See Appendix D for code on generating the mode.)

```
signif(samplemean,1) == signif(median(originalsim$averages),1)
```

```
## [1] TRUE
```

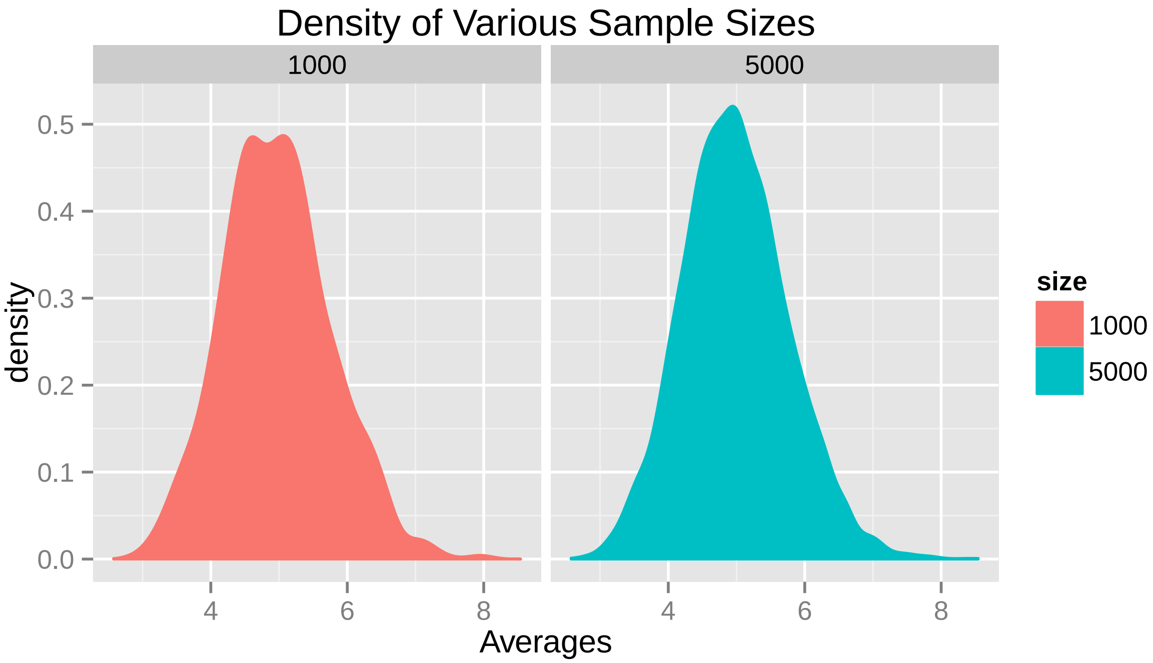
```
signif(samplemean,1) == samplemode
```

```
## [1] TRUE
```

By the transitive property of equality, if mean = median and mean = mode, then median= mode. In this case, that is true.

Symmetry

The plot below shows the 1,000 simulations and the 5,000 simulations I conducted. While the density of the first distribution looks lumpy and a bit asymmetric, it's obvious from the increasing number of simulations that the distribution starts to look more like a bell curve.



If I were to do an infinite number of simulations, I expect the distribution would look normal.

Appendix

Appendix A

Simulation Script

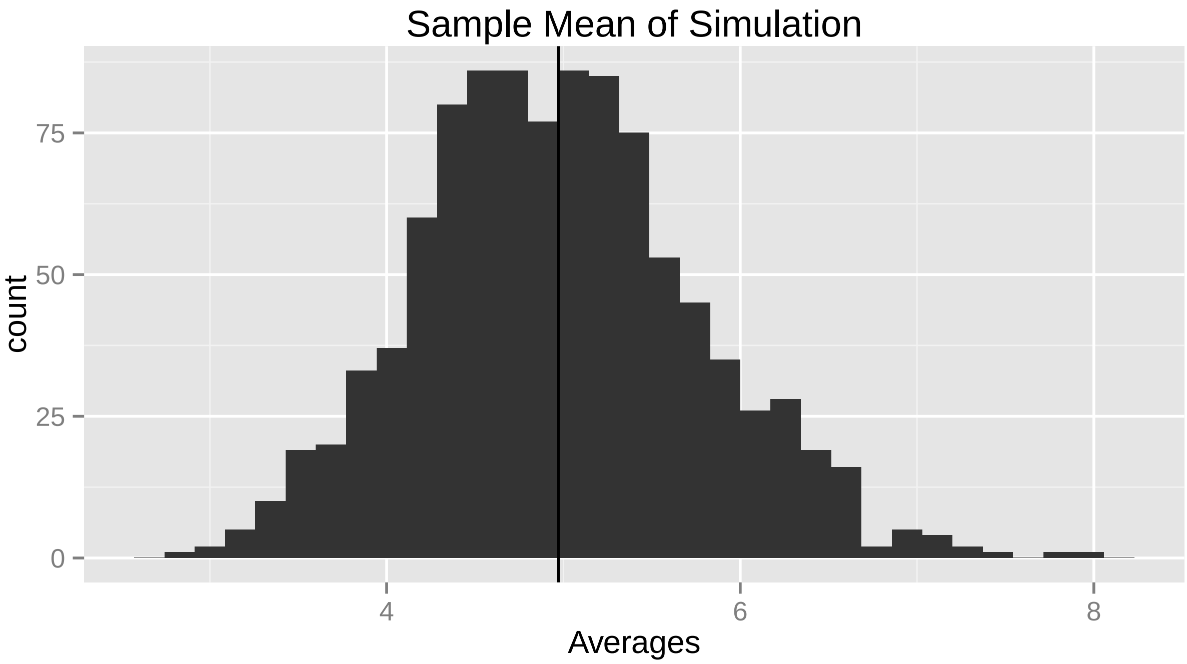
```
nosims <- 1000
nexps <- 40
rate <- 0.2

# using set.seed so the results can be replicated
set.seed(13)

# 1000 simulations on 40 random exponentials
sims <- replicate(nosims,rexp(nexps,rate))

# storing the clt of those simulations in a data frame
# with the size of the simulations
simsdf <- data.frame(averages = apply(sims,2, mean),
                     size = factor("1000"))
```

Appendix B



Appendix C

Running Additional Simulations and Adding Results to the Sims Data Frame

```r
# simulate 5000 random exponentials
set.seed(13)
sims2 <- replicate(nosims*5,rexp(nexps,rate))

# creating a new data frame and then binding those rows to the original
sampledf <- data.frame(averages=c(
  apply(sims2,2, mean)),
  size=factor("5000"))
simsdf <- rbind(simsdf,sampledf)

sample2mean <- mean(simsdf$averages[simsdf$size == "5000"])

simsdf$mean[simsdf$size=="1000"] <- samplemean
simsdf$mean[simsdf$size=="5000"] <- sample2mean
```

Appendix D

Calculating the Mode

```r
# subsetting the original simulation from my data frame
originalsim <- simsdf %>% filter(size=="1000")

# table of original simulation to find max
table_data <- table(signif(originalsim$averages,1))

# mode of table_data
samplemode <- subset(table_data, table_data==max(table_data))
samplemode <- as.numeric(names(samplemode))
```