Statistical Inference Course Project: Part 1, Simulation Exercise

Amal Haq

June 19, 2015

The instructions to Part 1 ask us to investigate the exponential distribution and compare it to the Central Limit Theorem. We are provided the following parameters:

- A rate parameter *lambda* of **0.2**;
- A mean *1/lambda* of **5**;
- A sample size *n* of **40**; and
- An expected number of simulations or *sim* of **1000**.

Let's set up the environment:

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials.

Refer to Appendix 1 for an explanation and justification on the approach I adopted for this assignment. It explains how to make sure we are comparing our simulated distribution to the Central Limit Theorem (CLT), and not the Law of Large Numbers (LLN).

Let's generate our data and illustrate using the recommended function `rexp()`

```
dataset<-rexp(n*sim, lambda) # generating random variables, or rather, random exponentials
df<- as.data.frame(matrix(dataset, sim)) # I prefer to work with data franmes over matrices
df<- mutate(df, sample_mean= rowMeans(df)) # the dplyr package makes it easy to add new colum
ns.  You can also use the apply() function Brian uses in his lectures.
```

1. Show where the distribution is centered at and compare it to the theoretical center of the distribution.

According to the CLT, the distribution of averages (or means) is centered around (has a mean) the population mean. The population mean is provided to us in the instructions as 1/lambda.

In our dataframe above, we have a distribution of 1000 means, stored in the column titled “sample_mean”. These means come from a 1000 samples (our simulation), all of which are representative of the entire population of exponentials. So our distribution should follow the CLT.

When we talk about the CLT, we are talking about all of these 1000 means, i.e the distribution of these means. This distribution in itself represents a sample population, and has a mean. This mean is the average of the 1000 means.

```
Theoretical_Mean<- 1/lambda # This is the theoretical mean of the distribution we have genera
ted. In theory, it is equal to the mean of the population distribution.
print(Theoretical_Mean)
```

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

```
Calculated_Mean<- mean(df$sample_mean) # This is the calculated mean of the distribution we h
ave generated, and it should be quite close to the mean of the population distribution.
print(Calculated_Mean)
```

```
## [1] 4.98662
```

BTW, if each of the samples in our simulation was made up of >40 exponentials, the calculated mean would be more and more accurate, until our samples were made up of infinity exponentials, at which point the mean of our distribution would ‘limit to’ the population mean, i.e. be equal to it. In theory.

2. Show how variable it is and compare it to the theoretical variance of the distribution.

According to the CLT, the distribution of averages (or means) has a standard deviation that is equal to the standard error of the mean (it is referring to the mean of the distribution but since we just showed that the mean of the distribution and the population mean are thoretically equal, it doesn’t actually matter). ‘Standard Error of the mean’ is the mean divided by the square root of the sample size *n*.

```
Theoretical_SD<- (1/lambda)/sqrt(n)
print(Theoretical_SD)
```

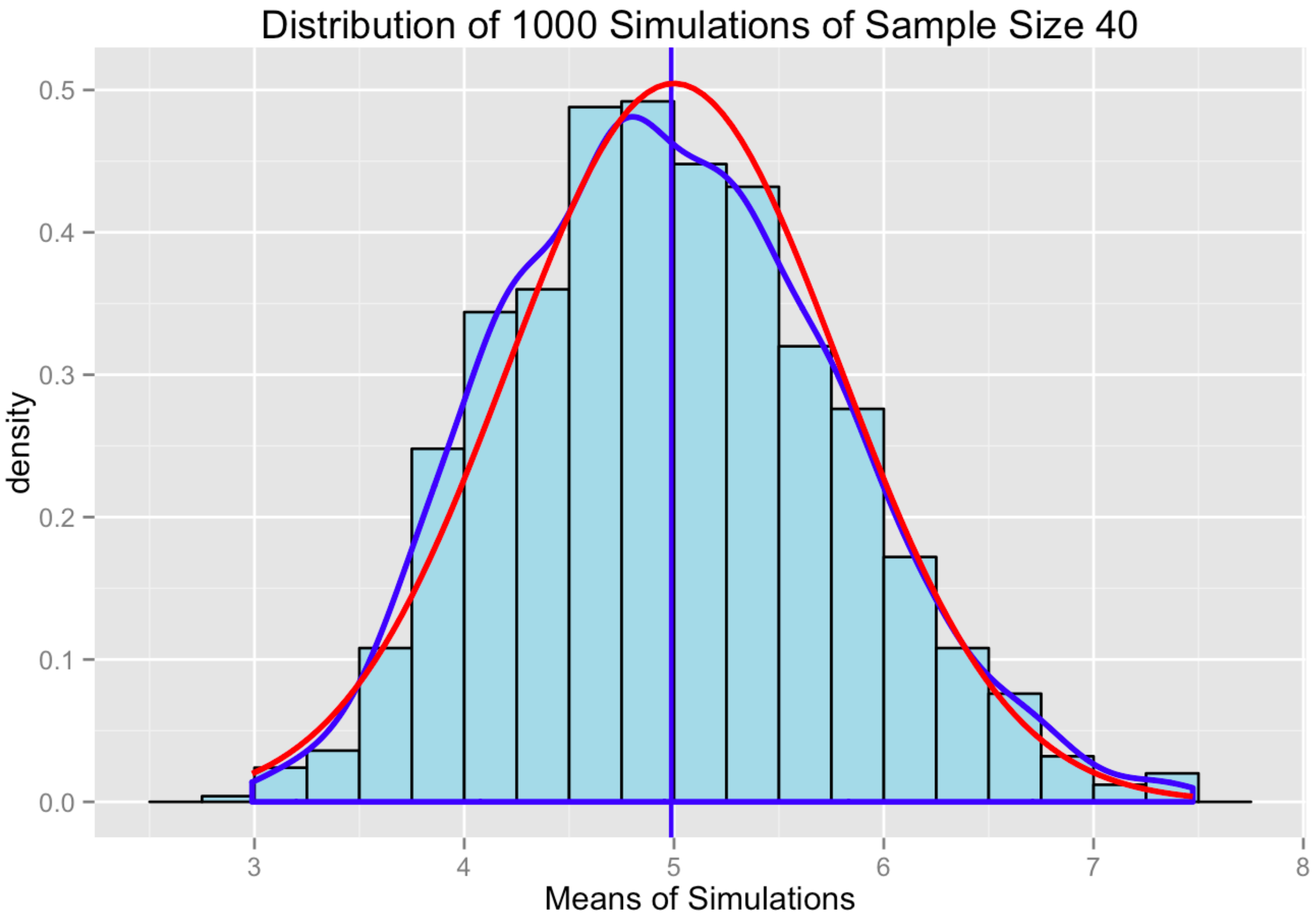
```
## [1] 0.7905694
```

```
Calculated_SD<- sd(df$sample_mean)
print(Calculated_SD)
```

```
## [1] 0.7910484
```

3. Show that the distribution is approximately normal.

For this, we need a visual. We'll plot distribution stored in our sample_mean column. What we should see is something that looks 'Gaussian' or bell-shaped. We already know that it will be centered around 5. *See Appendix 2 for the r code*.



Conclusion

We have shown that our simulated exponential distribution of 1000 means, which we got from 1000 samples of 40 exponentials each, follows the Central Limit Theorem. It has a mean that is approaching the population mean, it has a standard deviation that is equal to the standard error of the mean, and it has a density that is roughly normal.

APPENDIX 1

Let's make sure we understand what is being asked of us.

The code, `rexp(40,0.2)` gives us just a single sample. A single sample of 40 exponentials that has just **one** mean. The code, `rexp(40*1000, 0.2)` *again* gives us just a single sample. A single sample of 40,000 exponentials that has just **one** mean.

This does **not** prove the Central Limit Theorem but the Law of Large Numbers. It says, that the mean of any sample estimates the mean of the population from which that sample was taken. And if the size of that sample is large, then it will be a more accurate estimate. See below:

```
set.seed(1)
mean_of_one_sample_of_40<- mean(rexp(40,0.2))
print(mean_of_one_sample_of_40)
```

```
## [1] 4.860372
```

```
set.seed(2)
mean_of_one_sample_of_40000<- mean(rexp(40*1000,0.2))
print(mean_of_one_sample_of_40000)
```

```
## [1] 5.016356
```

Great. Now that we have that clarified, we can focus on our task

Our Task is to “investigate the distribution of averages of 40 exponentials”.

What does this mean?

- It means that we don’t want *a single sample of 40 exponentials that has just one mean*.
- It means that we want *multiple samples*.
- It means that we want *each sample to be made up of 40 exponentials*.
- It means that because we want multiple samples, we will end up with **multiple means/a collection of means/a distribution of means**.

This is why the instructions tell us to do a 1000 simulations, so that we get a 1000 means.

APPENDIX 2

Below is the code for the histogram plot overlayed with the appropriate densities:

```
Plot_distr<- ggplot(df, aes(x=df$sample_mean)) + #initialize plot
        geom_histogram (binwidth=0.25, color = "black", fill = "light blue", aes(y=..densit
y..)) + # plot a histogram with density on the y axis instead of count
        geom_vline(aes(xintercept=Calculated_Mean), color="blue", size=0.8) + #place a vertic
al line at the mean of the generated distribution
        geom_density (size =1, color = "blue", alpha = 0.3) + # plot the density of generated
distribution
        stat_function(fun=dnorm, args=list(mean=Theoretical_Mean, sd=Theoretical_SD), color =
"red", size = 1.0) + # superimpose the density of the population distribution
        labs(title="Distribution of 1000 Simulations of Sample Size 40", x="Means of Simulati
ons")
```