Simulating the Central Limit Theorem from an Exponential distribution

Victoria Mestre Runge

11 setembre de 2020

Overview

The Central Limit Theorem (CLT) states that if we repeatedly take independent and identically distributed (iid) random samples, the distribution of sample means approaches a normal distribution as the sample size increases. This occurs regardless of the shape of the distribution of the population or original variables. The present analysis examines this theory through the simulation that projects the population mean of the exponential distribution compared to the sampling distribution.

Simulations

In R programming language, the exponential distribution can be simulated with the $rexp(n, \lambda)$ function, where λ is the rate parameter. The expected mean (theoretical mean) of an exponential distribution is $\mu = 1/\lambda$ and its standard deviation is $\sigma = 1/\lambda$.

To understand the CLT, we will display a simulation that generates a couple of random simulations of an exponential distribution. In particular, we will consider **thousand simulations** and we will examine the distribution of the **averages of 40 exponentials** setting $\lambda = 0.2$ for all simulations.

Let's then first display the simulated data that generates a 1000 random samples of the exponential distribution.

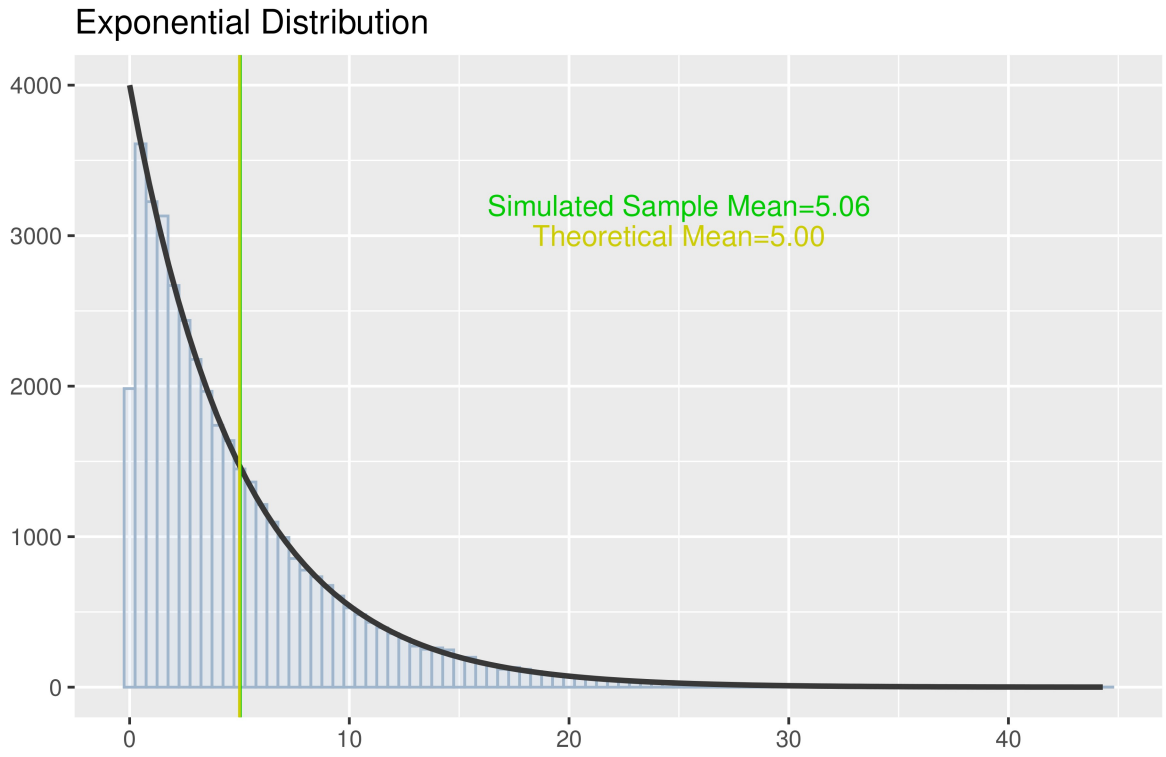


Figure 1. Simulating 1000 random samples overlaiding with the theoretical exponential distribution

The simulation of 1000 exponential random distributions makes an exponential shape, i.e. the Probability Distribution Function (PDF) exponentially declines from $x = 0$ to $x = \infty$. We also observe that the simulated mean of the Exponential distribtuion has a close match to the theoretical population mean.

Showing up next, we will simulate the average distribution of 1000 simulations of 40 exponentials random variables to demostrate the CLT, i.e. we will test these estatement against the exponential distribution ($\lambda = 0.2$) by taking 1000 samples of sample size 40.

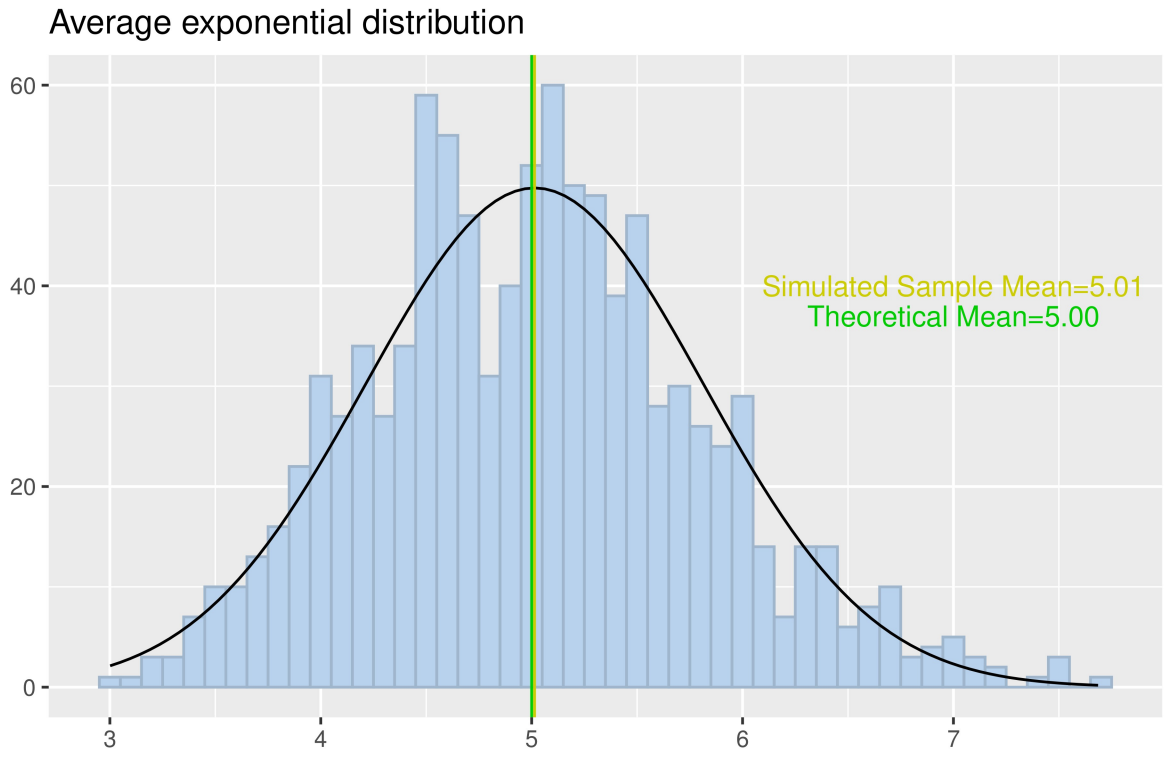
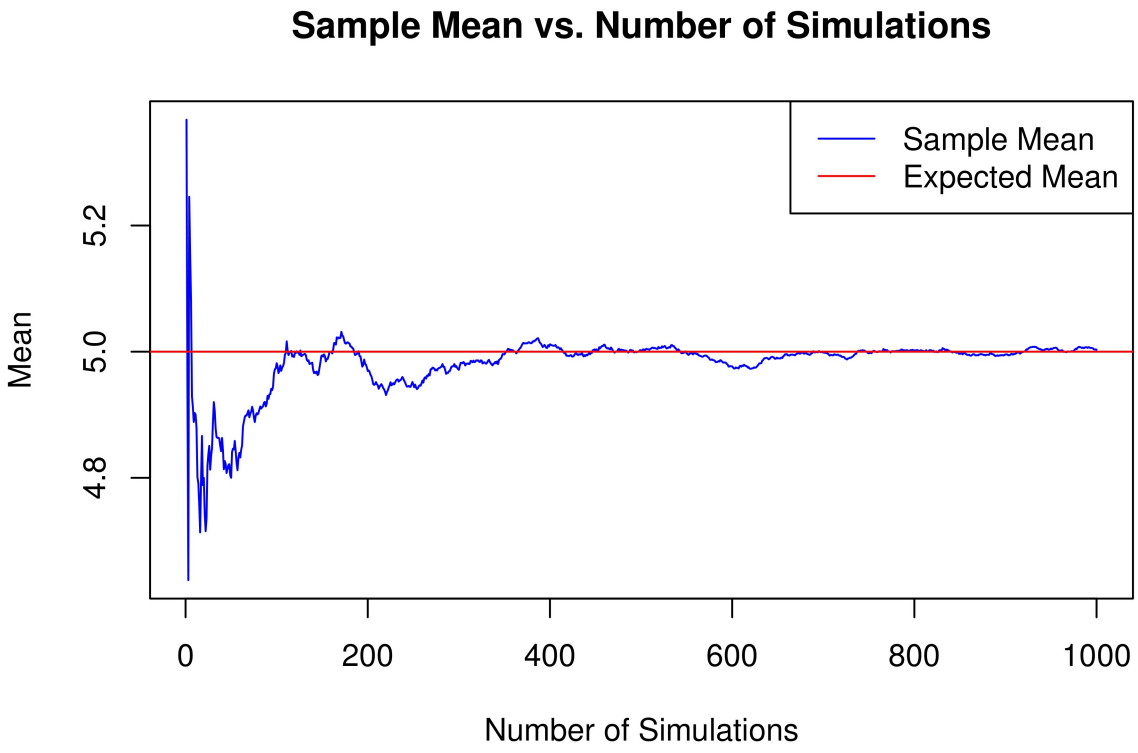


Figure 2. The exponential distribution of 40 average samples is close to a normal distribution

In comparison, when we simulate the average distribution of 1000 simulations of 40 exponential random variables, we see that is close to a normal distribution, i.e. is symmetric around the mean with a bell-shape curve. We also observe that the simulated mean of the averaged Exponential distribtuion has a close match to the theoretical population mean.

Sampling Distribution Mean vs Theoretical Mean

Figure 3 illustrate how the CLT works, i.e. as the number of simulations increases, the sample mean approaches the expected mean.

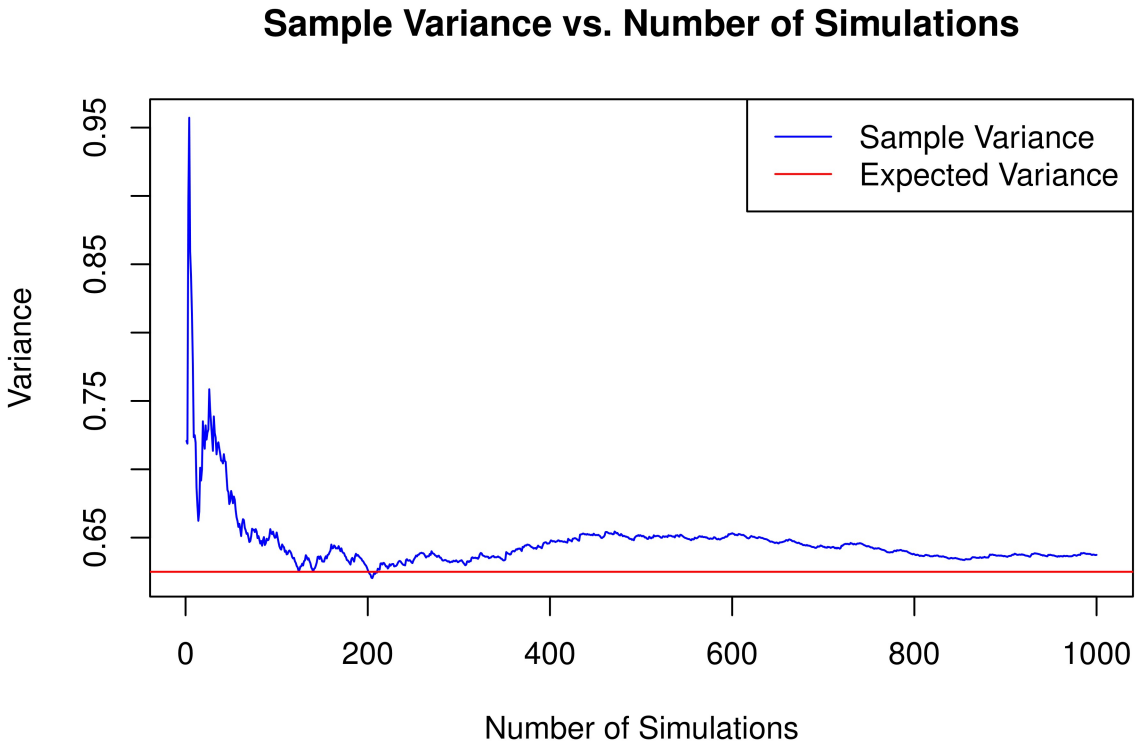


And as we saw in figure 2, the simulated mean of the exponential distribution has a close match to the theoretical population mean.

	Mean
Sampling Distribution Mean	5.01
Theoretical Mean	5.00

Sampling Distribution Variance vs theoretical Variance

Figure 4 illustrate that as the number of simulations increases, the sample variance approaches the expected variance.



The theoretical value for the variance of the distribution of averages is given by the variance of the original population σ^2 divided by the number of samples n used to compute the averages. And again, considering the distribution of the average of 40 exponentials, the sampliing variance of the averaged exponential distribution also converges to the population variance.

	Variance
Sampling Distribution variance	0.64
Theoretical variance	0.62

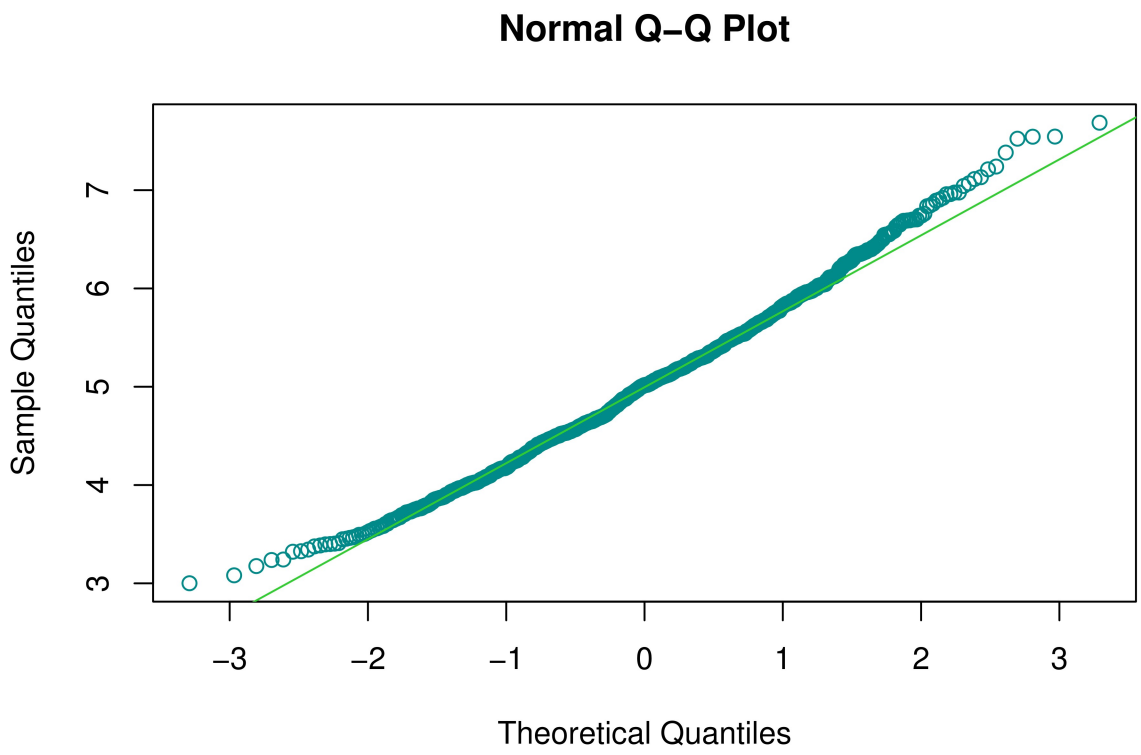
Normality of Sampling Distribution

We have already seen in figure 2 that the sampling distribution looks approximately normal as it approaches a bell-shape curve. Another 'back of the envolope' test is to compare distribution quantiles to those from a normal population with the same mean and variance.

```
##
##  Welch Two Sample t-test
##
## data:  meanExpData and normal_dist
## t = 0.52626, df = 1000.9, p-value = 0.5988
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.03642829  0.06312685
## sample estimates:
## mean of x mean of y
##  5.013745  5.000396
```

Due to the central limit theorem, the averages of samples follow normal distribution, as we saw in figure 2.

The q-q plot below also suggests the normality. The theoretical quantiles again match closely with the actual quantiles. These three methods of comparison prove that the distribution is approximately normal.



Conclusions

We can conclude that the distribution of means of our sampled exponential distributions appear to follow a normal distribution, due to the Central Limit Theorem. If we increased our number of samples (currently 1000), the distribution would be even closer to the standard normal distribution.

Appendix

1. Code for generating samples and calculate the averages of each sample

```
# Setting seed for reproducibility
set.seed(1984)

# Parameters known
nosim <- 1000   # Number of simulated random variables, i.e. number of trials
lambda <- 0.2   # Rate parameter for success
n <- 40         # Sample size, i.e. each sample contains 40 random variables

# Generating the samples, i.e. simulating 1000 runs of 40 random exponential distributions
expData = NULL
for (i in 1 : nosim) expData = c(expData, rexp(n, lambda))
expDat <- data.frame( value = expData)
```

```r
# Generating the average of each sample, i.e. simulating 1000 averages
# of 40 random exponential distributions
meanExpData = NULL
for (i in 1 : nosim) meanExpData = c(meanExpData, mean(rexp(n, lambda)))
meanDat <- data.frame(value = meanExpData)

varExpData = NULL
for ( i in 1 : nosim) varExpData = c(varExpData, var(rexp(n, lambda)))
varDat <- data.frame(value = varExpData)
```

2. Code for generating the exponential distribution histogram

```r
# Plot exponential distribution
figure1 <- ggplot(expDat, aes(x=value)) +
        geom_histogram(alpha=0.2, binwidth=0.5, color="slategray3", fill="slategray2")+
        stat_function(fun=function(x, rate, n) {n*dexp(x, rate)},
                      args=c(rate=lambda, n=n*nosim*0.5),
                      geom="line", color="grey22", size=1) +
        geom_vline(aes(xintercept=mean(rexp(n*nosim, lambda))), color="green3",
                   size=0.5)+
        geom_vline(aes(xintercept=1/lambda), color="yellow3", size=0.5) +
        annotate("text", x=30, y=3200, label=sprintf("Simulated Sample Mean=%03.2f",
                mean(rexp(nosim, lambda))), color="green3")+
        annotate("text", x=30, y=3000,
                 label=sprintf("Theoretical Mean=%03.2f", 1/lambda), color="yellow3")+
        labs(title="Exponential Distribution", x="", y="",
             caption="Figure 1. Simulating 1000 random samples overlaiding
             with the theoretical exponential distribution")
```

3. Code for generating the average exponential distribution histogram

```r
figure2 <- ggplot(meanDat, aes(x=value)) +
        geom_histogram(binwidth=0.1, color="slategray3", fill="slategray2")+
        stat_function(fun=function(x, mean, sd, n) {n*dnorm(x=x, mean=mean, sd=sd)},
                      args=c(mean=mean(meanExpData), sd=sd(meanExpData), n=nosim/5/2))+
        geom_vline(aes(xintercept=mean(meanExpData)), color="yellow3", size=0.5) +
        geom_vline(aes(xintercept=1/lambda), color="green3", size=0.5) +
        annotate("text", x=7, y=40,
                 label=sprintf("Simulated Sample Mean=%03.2f",
                               mean(mean(meanExpData))), color="yellow3")+
        annotate("text", x=7, y=37,
                 label=sprintf("Theoretical Mean=%03.2f", 1/lambda), color="green3")+
        labs(title="Average exponential distribution", x="", y="",
             caption="Figure 2. The exponential distribution of 40 average samples is
             close to a normal distribution")
```

4. Code for comparing sampling distributed mean vs theoretical mean

```r
theoretical_mean <- 1/lambda
sampling_dist_mean <- mean(meanExpData)

meanResults <- data.frame("Mean"=c(sampling_dist_mean, theoretical_mean),
```

```
                    row.names = c("Sampling Distribution Mean", "Theoretical Mean"))

kable(x = round(meanResults,2), align = 'c')

#######

means <- cumsum(sample(meanExpData, nosim, replace = TRUE))/(1:nosim) # cumulative frequency

figure3 <- plot(means, type = "l", col = "blue",
                main = "Sample Mean vs. Number of Simulations",
                xlab = "Number of Simulations", ylab = "Mean")
abline(h = 1/lambda, col = "red")
legend("topright", legend = c("Sample Mean","Expected Mean"),
       col = c("blue","red"), lty = 1)
```

5. Code for comparing sampling distributed variance vs theoretical variance

```
theoretical_sd <- 1/lambda
theoretical_variance <- theoretical_sd^2/n
sampling_dist_variance <- var(meanExpData)

varianceResults <- data.frame("Variance"=c(sampling_dist_variance,
                                           theoretical_variance),
                    row.names = c("Sampling Distribution variance",
                                  "Theoretical variance"))

kable(x = round(varianceResults,2), align = 'c')

#######

variance <- cumsum(sample(varExpData/n, nosim, replace = TRUE))/(1:nosim)

figure4 <- plot(variance, type = "l", col = "blue",
                main = "Sample Variance vs. Number of Simulations",
                xlab = "Number of Simulations", ylab = "Variance")
abline(h = (1/lambda)^2/n, col = "red")
legend("topright", legend = c("Sample Variance","Expected Variance"),
       col = c("blue","red"), lty = 1)
```

6. Code for t-test

```
normal_dist <- rnorm(nosim, theoretical_mean, sqrt(theoretical_variance/nosim))
ttestresults <- t.test(meanExpData, normal_dist)
print(ttestresults)
```

7. Code for quantiles

```
qqnorm(meanDat$value, col="cyan4")
qqline(meanDat$value, col="limegreen")
```