Exponential Distribution and Central Limit Theorem

Vijay Ramanujam

January 5, 2018

Overview

In this project we will investigate the exponential distribution in R and compare it with the Central Limit Theorem. The exponential distribution can be simulated in R with rexp(n, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. Let's set lambda = 0.2 for all of the simulations. We will investigate the distribution of averages of 40 exponentials. Note that we will need to do a thousand simulations.

Our results will illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials. We should be able to see

1. Comparison of the sample mean to the theoretical mean of the distribution.
2. Variablity of the sample compared to the theoretical variance of the distribution.
3. The distribution is approximately normal.

Known values

lambda = 0.2, n = 40, simulations = 1000, and seed = 123. Refer Appendix (i) for R code

Simulation

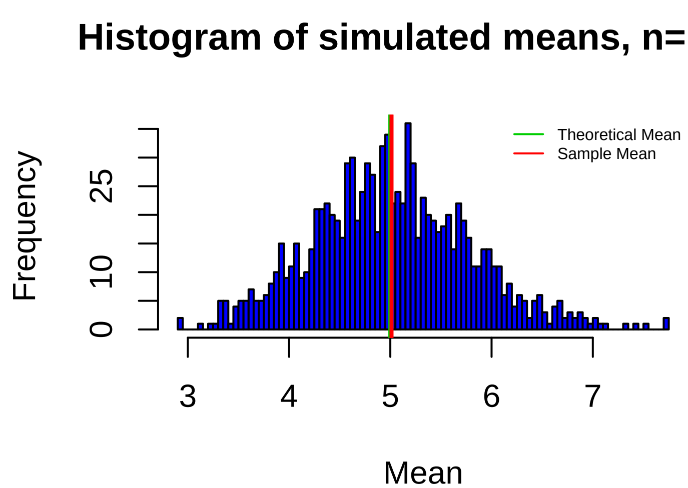
Our next goal is to do 1000 simulations of 40 exponentials and find out the mean of those individual simualtions. Refer Appendix (ii) for R code

Theoretical mean of exponential distribution is 1/lambda

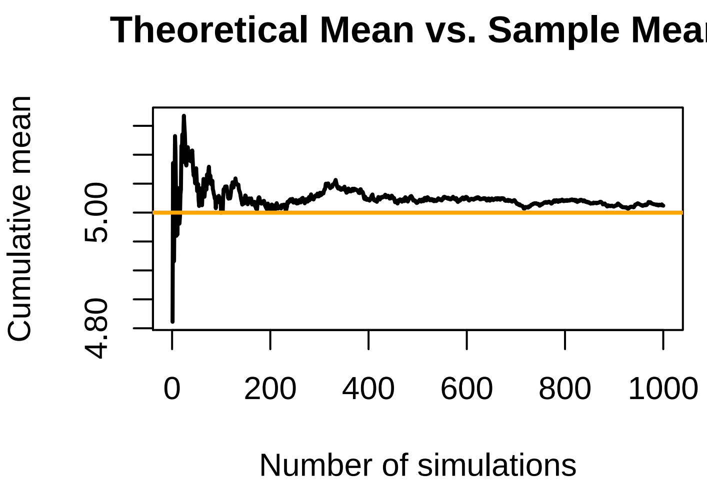
```
## [1] 5
```

Simulation mean, n = 40, simulations = 1000

```
## [1] 5.011911
```



Simulation mean, 5.0119, is very close to theoretical mean of 5. The histogram (means of 40 exponentials of 1000 simulations) above shows the same visually. Refer Appendix (iii) for R code



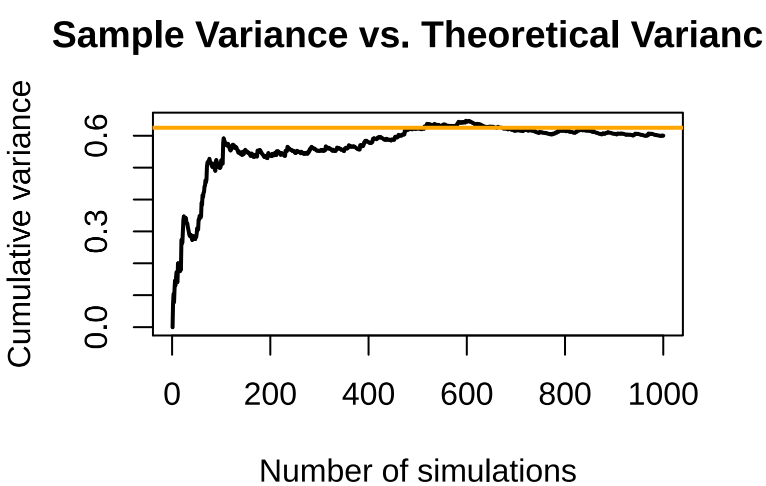
As number of simulations increases, the sample mean converges to the theoretical mean. Refer Appendix (iv) for R code

Theoretical variance is (1/lambda)^2/n

```
## [1] 0.625
```

Sample variance

```
## [1] 0.6004928
```

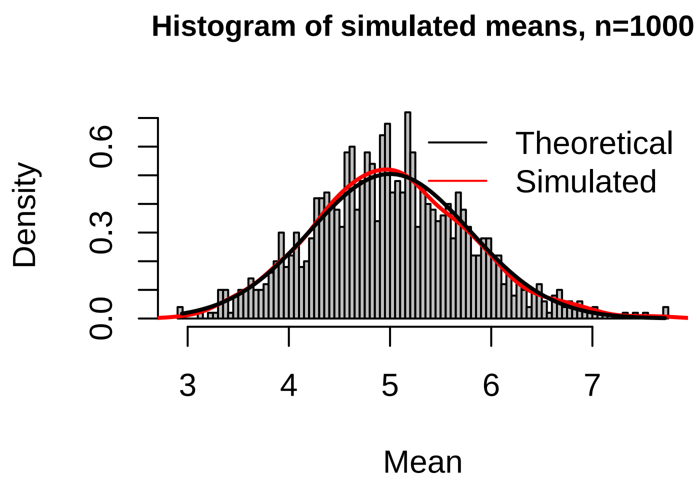


Theoretical variance is 0.625 and sample variance of 1000 simulations of average of 40 exponentials is 0.6004928. As the number of simulations increases, the sample variance converges to the theoretical variance on the plot. Refer Appendix (v) for R code

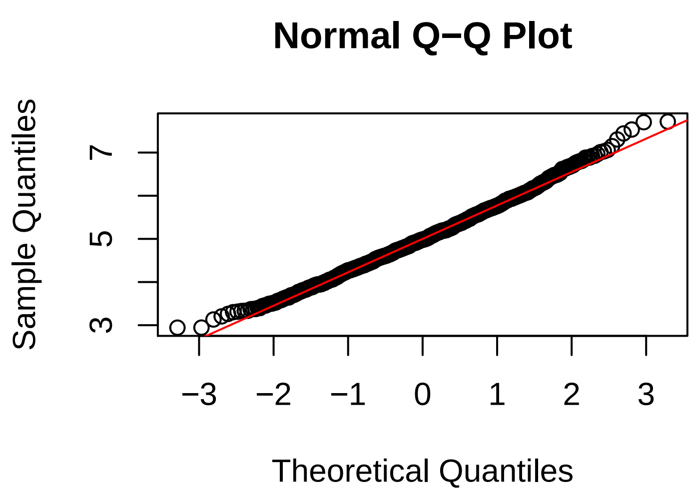
One of the reasons we are comparing the means is to make sure they are not too different. Normally, we will be estimating the population mean based on the sample mean. Fortunately, in this case, we know the population mean, i.e. 5. Let's see whether our sample mean is a good estimator of population mean. One way to do that is to build a confidence interval for the sample mean and see whether the population mean falls in the range. 95% confidence interval of sample mean is

```
## [1] 4.771767 5.252055
```

Since theoretical mean falls in the range, we can say that with 95% certainty, the true population mean will be in between 4.771767 and 5.252055. Refer Appendix (ix) for R code



The above plot shows that the distribution of a large collection of averages of 40 exponentials is approximately normal. Refer Appendix (vii) for R code



q-q plots are another way to show normality. In this plot, the theoretical quantiles match closely with the actual quantiles. Refer Appendix (viii) for R code

Based on all of the above methods of comparison, it is evident that the distribution of the mean of 40 exponentials is approximately normal.

Appendix

(i)

```
nLambda <- 0.2
n <- 40
nSims <- 1000

set.seed(123)
```

(ii)

```
aMeans <- data.frame(Mean = sapply(1:nSims, function(x) {mean(rexp(n, nLambda))}))
```

(iii)

```
#Theoretical mean
ntMean <- 1/nLambda

#Simulation mean
nsMean <- mean(aMeans$Mean)

#Histogram
hist(
    aMeans$Mean
    , col = "blue"
    , xlab="Mean"
    , main="Histogram of simulated means, n=1000"
    , breaks = 100
)
abline(v = ntMean, col= 3, lwd = 2)
abline(v = nsMean, col = 2, lwd = 2)

legend(
    'topright'
    , c("Theoretical Mean", "Sample Mean")
    , bty = "n"
    , lty = c(1,1)
    , col = c(col = 3, col = 2)
)
```

(iv)

```
ncsMeans <- cumsum(aMeans$Mean)/(1:nSims)
plot(ncsMeans, type="l", lwd=2,
     main = "Theoretical Mean vs. Sample Mean",
     xlab = "Number of simulations",
     ylab = "Cumulative mean")
abline(h=1/nLambda, col="orange", lwd=2)
```

(v)

```
#Theoretical variance
ntVar <- ((1/nLambda)^2)/n

#Sample variance
nsVar <- var(aMeans$Mean)
```

```r
#Plot
ncsMeans <- cumsum(aMeans$Mean)/(1:nSims)
ncsVars <- cumsum(aMeans$Mean^2)/(1:nSims)-ncsMeans^2
plot(ncsVars, type="l", lwd=2,
     main = "Sample Variance vs. Theoretical Variance",
     xlab = "Number of simulations",
     ylab = "Cumulative variance")
abline(h=(1/nLambda)^2/n, col="orange", lwd=2)
```

(vii)

```r
hist(
    aMeans$Mean
    , col = "grey"
    , xlab="Mean"
    , ylab="Density"
    , main="Histogram of simulated means, n=1000"
    , breaks = 100
    , prob = TRUE
    , cex.main=0.9
)

lines(density(aMeans$Mean), col = 2, lwd = 2)

x <- seq(min(aMeans$Mean), max(aMeans$Mean), length = 100)

y<- dnorm(x, mean = ntMean, sd = (1/nLambda/sqrt(n)))

lines(x, y, pch = 22, col = 1, lwd = 2)

legend(
    'topright'
    , c("Theoretical", "Simulated")
    , bty = "n"
    , lty = c(1,1)
    , col = c(col = 1, col = 2)
)
```

(viii)

```r
qqnorm(aMeans$Mean)
qqline(aMeans$Mean, col="red")
```

(ix)

```r
sCI <- nsMean + c(-1,1) * qnorm(0.975) * sqrt(nsVar / n)
```