Central limit theorem simulation using exponentially distributed samples

Gurmeet Singh

November 16, 2015

Overview

This report describes a simulation to obtain sample sets from an exponential distribution using R and compare the distribution of the sample means with what is predicted by the Central Limit Theorem.

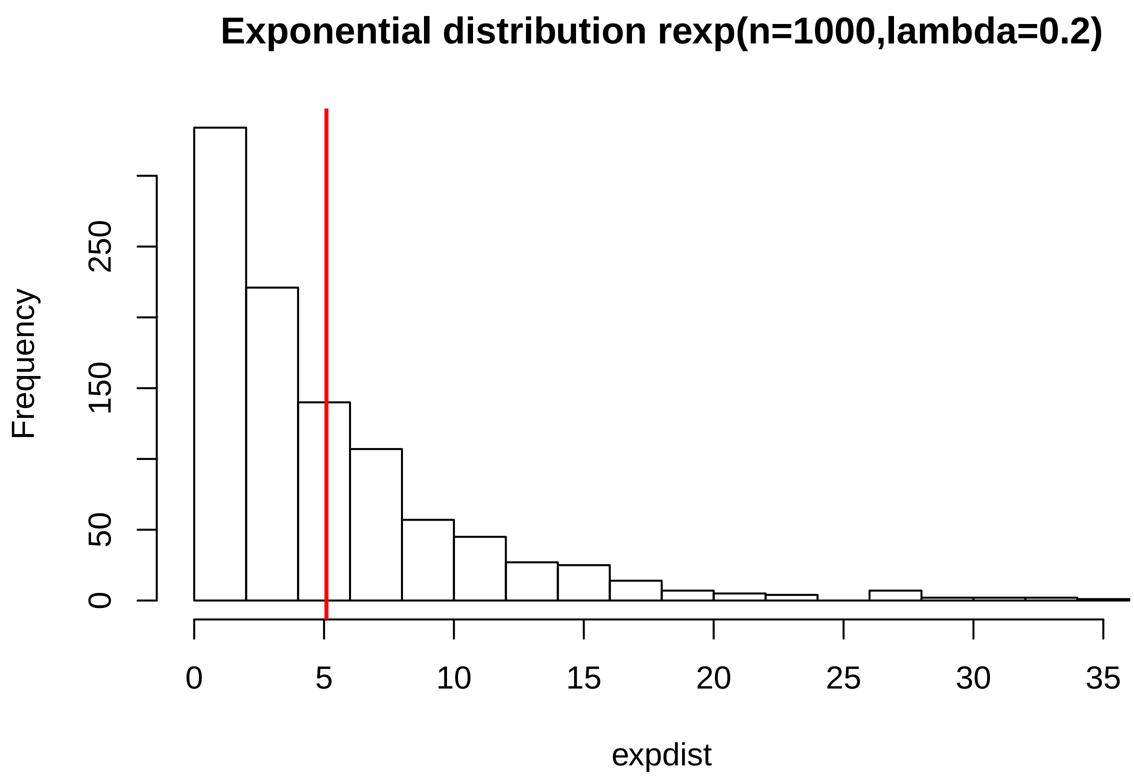
Exponential Distribution The exponential distribution is defined by its cumulative distribution function

$$F(x) = 1 - e^{-\lambda x}$$

The R function `rexp` generates random variables with an exponential distribution.

```
rexp(n=10, rate=0.2)
```

produces 10 exponentially-distributed numbers with rate (λ) of 0.2. If the second argument is omitted, the default rate is 1; this is the "standard exponential distribution". Here's an example of a histogram showing data from an exponential distribution with n = 1000 and $\lambda = 0.2$.



The red vertical line indicates the mean of the generated samples.

The simulation

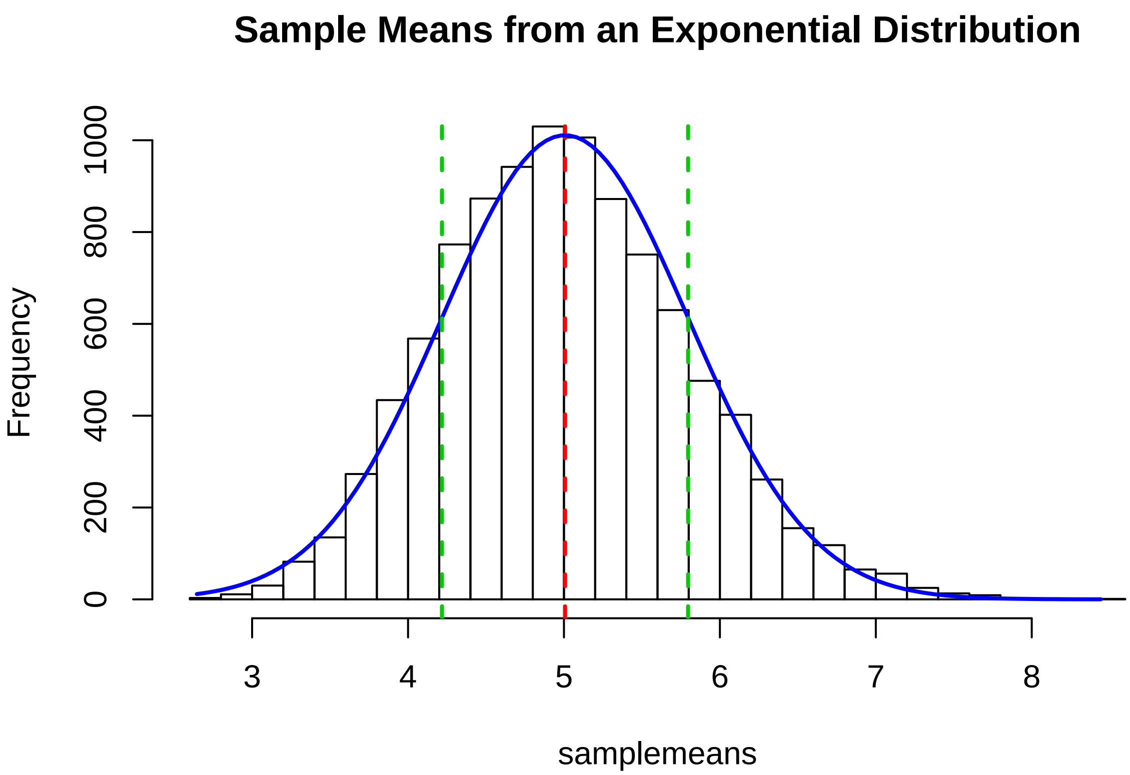
The simulation is done by:

1. Get a large number n of set of data, each of size 40, that is exponentially distributed.
2. Compute the mean of all these distributions resulting in another n means of 40 numbers each
3. This set of n means is a new distribution with its own mean and variance
4. The Central Limit Theorem mandates that this new distribution should be similar to a normal distribution
5. Draw a histogram of the sample means to visualize
6. Draw a normal distribution approximating the data in the histogram and verify if the two match.

The R code chunk to create the relevant data is shown below. See appendix for complete code.

```
n <- 40; k <- 10000; lambda <- 0.2
samplemeans <- NULL
for (i in 1:k) {
  expmean <- mean(rexp(n, lambda))
  samplemeans <- c(samplemeans, expmean)
}
```

The following graph is created from the simulation of taking means of samples of 10000 exponential distributions each with a size of 40 and $\lambda = 0.2$.



The vertical dashed red line indicates the mean of the sample means and the $\mu \pm \sigma$ points are marked by dashed green lines. It should be close to the distribution mean of $1/\lambda$, or 5.0 in our example. The blue line attempts to fit an ideal normal distribution to the histogram of the simulated data.

Conclusion

1. From the graph, it seems clear that the sample means are normally distributed
2. Also, the actual mean of the sample means and the actual variance of the sample means are respectively a good approximation of values predicted by the central limit theorem. This is verified in R code.

```
## Computed Mean of the n sample means = 5.007

## Expected Mean of the n sample means (1/lambda) = 5

## Computed Standard Deviation of the n sample means = 0.789

## Expected Standard Deviation of the n sample means (1/lambda)/sqrt(n) =  0.791
```

Appendix

The content of this report is authored in RStudio using R Markdown format and converted to PDF format using the **knitr** package. The R Markdown file itself can be found on GitHub

Code to generate an example of an exponential distribution.

```
expdist <- rexp(1000, 0.2)
hist(expdist,
     breaks = 20,
     main = "Exponential distribution rexp(n=1000,lambda=0.2)")
abline(v=mean(expdist),col=2,lwd=2)
```

Code for runnning the simulation and displaying results.

```
n <- 40; k <- 10000; lambda <- 0.2
samplemeans <- NULL
for (i in 1:k) {
  expmean <- mean(rexp(n, lambda))
  samplemeans <- c(samplemeans, expmean)
}
histbreaks <- 40
h <- hist(samplemeans,
     breaks = histbreaks,
     main = "Sample Means from an Exponential Distribution")
abline(v=mean(samplemeans),col=2,lty=2,lwd=2)
xfit<-seq(min(samplemeans),max(samplemeans),length=histbreaks*3)
yfit<-dnorm(xfit,mean=mean(samplemeans),sd=sd(samplemeans))
yfit <- yfit*diff(h$mids[1:2])*length(samplemeans)
lines(xfit, yfit, col="blue", lwd=2)
abline(v=mean(samplemeans)-sd(samplemeans),col=3,lty=2,lwd=2)
abline(v=mean(samplemeans)+sd(samplemeans),col=3,lty=2,lwd=2)
```

Code for comparing the theoretical and expected values of mean and variance.

```
meanofmeans <- mean(samplemeans)
sdofmeans <- sd(samplemeans)
cat("Computed Mean of the n sample means =", round(meanofmeans,3) ,"\n")
cat("Expected Mean of the n sample means (1/lambda) =", 1/lambda ,"\n")
cat("Computed Standard Deviation of the n sample means =", round(sdofmeans,3),"\n")
cat("Expected Standard Deviation of the n sample means (1/lambda)/sqrt(n) = ", round((1/lambda)/sqrt(n)
```