CLT for Sample Means (n=40): Exponential (lambda=0.2)

davegoblue

March 8, 2016

Overview and Synopsis

The Central Limit Theorem (CLT) states that the distribution of means calculated from a large sample of independent draws from a population will be approximately normal. The population distribution may be decidedly non-normal provided the sample mean is derived from a “sufficiently large” sample.

In this analysis, we looked at the distribution of means for 1,000 random samples each of size n=40 taken from the exponential distribution with lambda=0.2. Consistent with the CLT, these sample means follow a normal distribution with mean ~5.0 and variance ~0.63 (standard deviation ~0.79).

Analysis

Simulations

To create data, we took the means of 1,000 random samples of size n=40 from the exponential distribution with lambda (rate) of 0.2. The seed is fixed for reproducibility, and means are stored by sequentially growing vector expMeans.

```
set.seed(0308161239)
expMeans <- NULL
for (intCtr in 1:1000) { expMeans <- c(expMeans,mean(rexp(40,rate=0.2))) }
str(expMeans)
```

```
##  num [1:1000] 7 5.25 5.29 5.36 4.57 ...
```

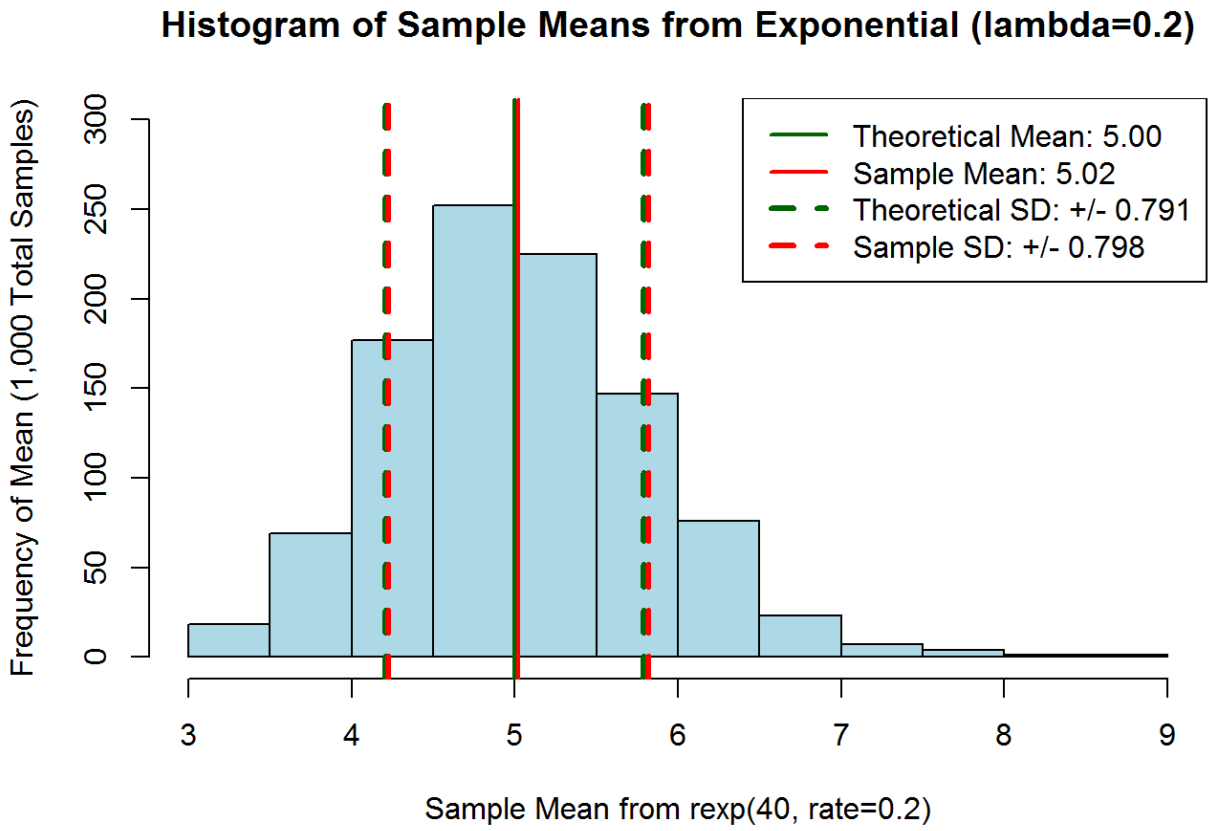
Examination of expMeans shows that we have 1,000 numeric values, each representing one mean of a sample of n=40 random draws taken from the exponential distribution with lambda=0.2.

Sample Mean, Variance, and Standard Deviation

The exponential distribution is well understood to have mean 1/lambda and variance 1/lambda^2. Since we are using lambda=0.2 and sample sizes of 40, we expect:

- Sample Mean = Population Mean = 1/lambda = 5.00
- Variance of Sample Mean = Population Variance / Sample Size = 1/(lambda^2)/40 = 0.625
- Standard Deviation of Sample Mean = sqrt(Variance of Sample Mean) = 0.791

We investigate a histogram of sample means, with vertical lines plotted to show mean as well as +/- 1 standard deviation (theoretical and observed). Standard deviation is used rather than variance as it is easier to visualize on a histogram:

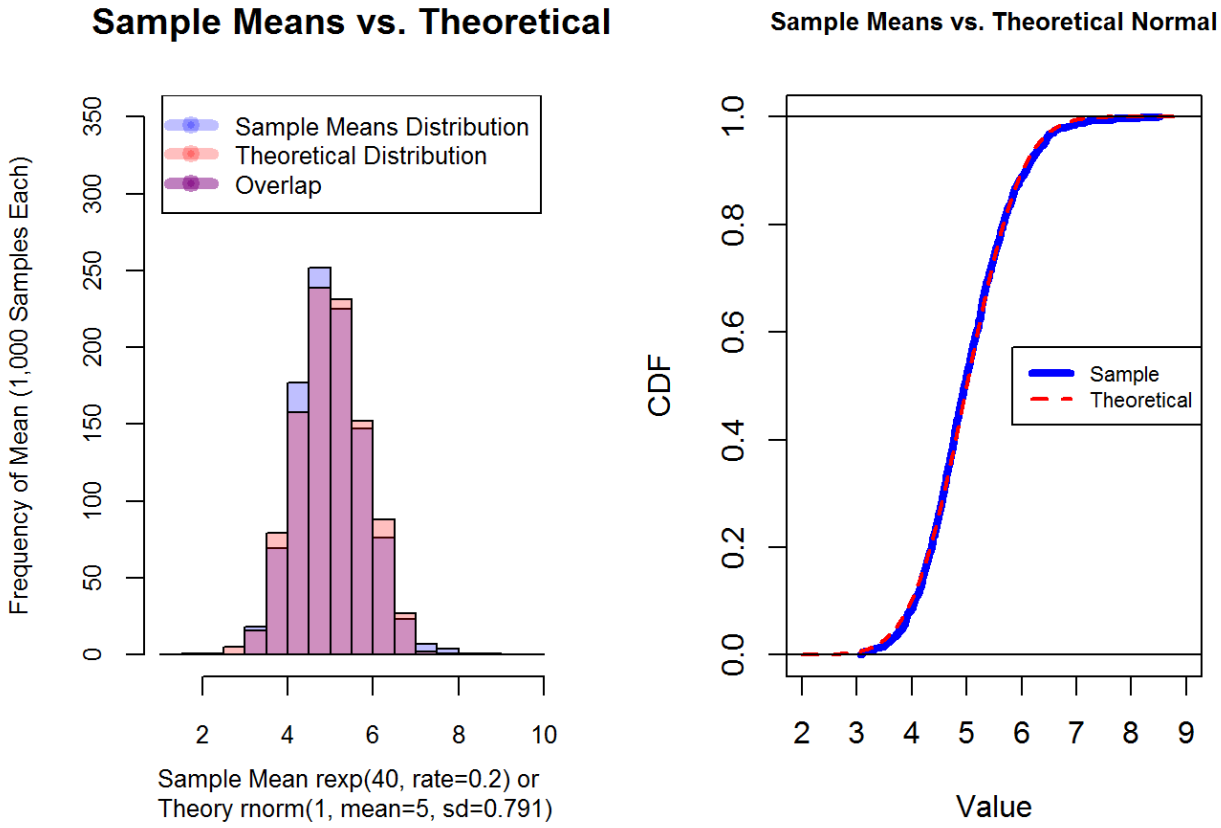


There is such tight overlap of theoretical mean / standard deviation (5.00 +/- 0.79) and observed mean / standard deviation (5.02 +/- 0.80) that lines nearly overlap. This confirms that our 1,000 obervations taken as averages of 40 draws from the exponential with lambda=0.2 are closely aligned with their theoretical mean and standard deviation.

Please see Appendix for the relevant R code (run with echo=FALSE to keep main report to 3 pages)

Validation of Normal Distribution

We also check the degree of fit between our sample means and the normal distribution with mean 5.00 and standard deviation 0.791. We pull 1,000 individual normals and compare them to the histogram of n=40 sample means from the exponential distribution. Further, we plot the CDF for our sample means against the theoretical normal CDF:



While there are modest differences in the histograms and CDF (n=40 and even n=1000 are not n=infinite), our distribution of sample means closely follows the predicted (theoretical) normal distribution with mean=5 and sd=0.791.

Please see Appendix for the relevant R code (run with echo=FALSE to keep main report to 3 pages)

Conclusion

The distribution of n=40 sample means pulled from an exponential with lambda (rate) of 0.2 closely match the predictions made by application of the Central Limit Theorem – a normal distribution with mean 5.0 and standard deviation 0.8.

.

Technical Note: My machine considers the executable required to run knit2pdf to be potential malware. This PDF is instead generated using Acrobat DC to convert the html output of knit2html.

Appendix

R Code for Generating Mean and Standard Deviation Graph

The below code was run using echo=FALSE to create the histogram of sample means, including their mean and standard deviation. It is shown here using eval=FALSE for reference:

```
hist(expMeans,col="light blue",ylim=c(0,300),
     xlab="Sample Mean from rexp(40, rate=0.2)",
     ylab="Frequency of Mean (1,000 Total Samples)",
     main="Histogram of Sample Means from Exponential (lambda=0.2)"
     )
abline(v=c(5 + c(-1,0,1) * 0.791),
       lwd=c(3,2,3), lty=c(2,1,2), col="dark green"
       )
abline(v=c(mean(expMeans) + c(-1,0,1) * sd(expMeans)),
       lwd=c(3,2,3), lty=c(2,1,2), col="red"
       )
legend("topright",lty=c(1,1,2,2),lwd=c(2,2,3,3),
       col=c("dark green","red","dark green","red"),
       legend=c("Theoretical Mean: 5.00",
                paste0("Sample Mean: ",round(mean(expMeans),2)),
                "Theoretical SD: +/- 0.791",
                paste0("Sample SD: +/- ",round(sd(expMeans),3))
                )
       )
```

R Code for Generating Overlapping Histograms and CDF

The below code was run using echo=FALSE to create the overlapping histograms and CDF comparing our sample means to their theoretical normal distribution. It is shown here using eval=FALSE for reference:

```
## Seed was set previously, code is reproducible
normals <- rnorm(n=1000,mean=5,sd=0.791)
par(mfcol=c(1,2))
hist(expMeans,col=rgb(0,0,1,0.25),ylim=c(0,350),
     xlab="Sample Mean rexp(40, rate=0.2) or \n Theory rnorm(1, mean=5, sd=0.79
1)",
     ylab="Frequency of Mean (1,000 Samples Each)",
     main="Sample Means vs. Theoretical",
     breaks=seq(1,10,by=0.5),cex.axis=0.8,cex.lab=0.8
     )
hist(normals,col=rgb(1,0,0,0.25),breaks=seq(1,10,by=0.5),add=TRUE)
legend("top",legend=c("Sample Means Distribution",
                      "Theoretical Distribution",
                      "Overlap"
                      ),
       pch=20,lwd=6,cex=0.8,
       col=c(rgb(0,0,1,0.25),rgb(1,0,0,0.25),rgb(0.5,0,0.5,0.5))
       )

## Create the CDF for each distribution
ordExpMeans <- data.frame(x=expMeans[order(expMeans)],
                          y=seq(0.0005,0.9995,by=0.001)
                          )
plot(ordExpMeans, type="l", col="blue", lwd=4,xlim=c(2,9), cex.main=0.8,
     main="Sample Means vs. Theoretical Normal",xlab="Value",ylab="CDF"
     )
lines(x=seq(2,9,by=.001),
      y=pnorm(seq(2,9,by=.001),mean=5,sd=0.791),
      col="red", lwd=2, lty=2
      )
abline(h=c(0,1))
legend("right",legend=c("Sample","Theoretical"),
       col=c("blue","red"),lty=c(1,2),lwd=c(4,2),cex=0.7
       )
par(mfcol=c(1,1))
```