Sample Estimates vs the Central Limit Theorem

R. Handsfield

February 19, 2016

Abstract

In this project, we compare the exponential distribution to the Central Limit Theorem. Simulated exponential distributions of $n = 40$ are sampled from an exponential distribution with rate parameter $\lambda = 0.2$.

Overview

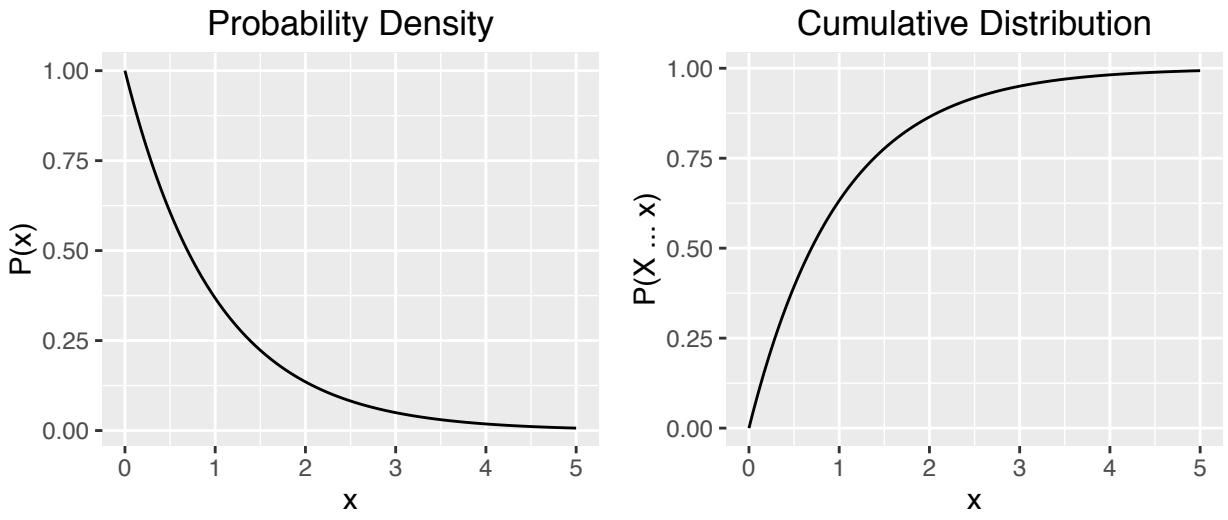
The exponential distribution, also known as the *Poisson* distribution, is often used to measure rates. The probability density and cumulative probability distributions are

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad F(x;\lambda) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The parameter *lambda* is known as the *rate parameter*, and is directly related to the mean and standard deviation.

$$\mu = \frac{1}{\lambda} \qquad \sigma = \frac{1}{\lambda}$$

Plots of the exponential distribution look like this



Simulations

We create each comparison distribution by sampling 40 values randomly from the exponential distribution. We simulate a total of 1000 comparisons; the mean of each simulation ($\bar{X}$) is calculated and appended to the vector `means`.

- Average of simulated means: `mean(means)` = 4.95
- Variance of simulated means: `var(means)` = 0.624

According to the Central Limit Theorem, the average of these 1000 means should estimate the true population mean μ. The means should be normally distributed, with a variance proportional to the number of values in each simulation (40).

Sample Mean vs Population Mean

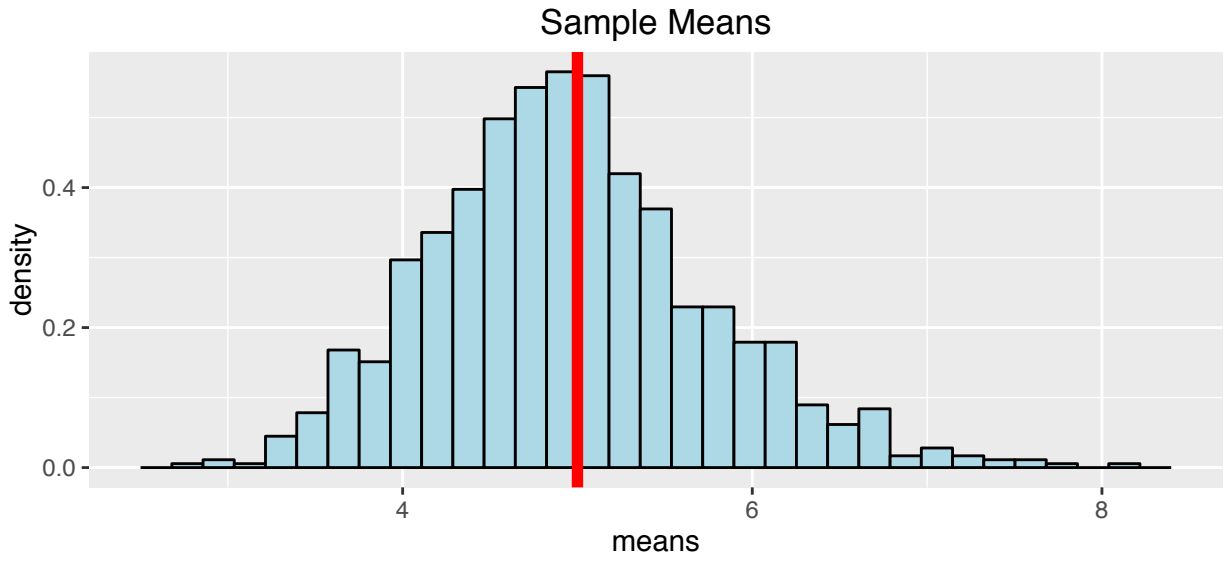
The average of simulated means is the ***Sample Mean:*** $\bar{X}_\mu = 4.95$.

The population mean of the exponential distribution is

$$\mu = \frac{1}{\lambda} = \frac{1}{0.2} = 5.0$$

4.95 seems close to 5.0, but how close of an estimate is this? We can figure it out by looking at the distribution of simulated means.

The histogram of simulated means is:



The 1000 simulated means are normally distributed and centered near the true population mean μ (red line). Because the means are normally distributed, we can use normal statistics to determine how good of an estimte 4.95 actually is.

The probability of the sample means randomly centering at 4.95 is:

```
1 - 2*(pnorm( abs( mean(means)-5 ), mean=0, sd=sd(means), lower.tail = FALSE) )
```

```
## [1] 0.04598328
```

There is a probability of just a few chances out of 100 that our sample mean estimates a non-exponential population mean, but randomly averages to a value near 5.0.

Variance of the Sample Mean

According to the central limit theorem, a sample estimates a statistic, and has variance proportional to the sample size n.

$$Var(\bar{X}) = \left(\frac{\sigma}{\sqrt{n}}\right)^2$$

For the exponential distribution

$$Var(\bar{X}) = \left(\frac{1}{\lambda\sqrt{n}}\right)^2 = \left(0.2\sqrt{40}\right)^{-2} = 0.625$$

The variance of the sample mean is

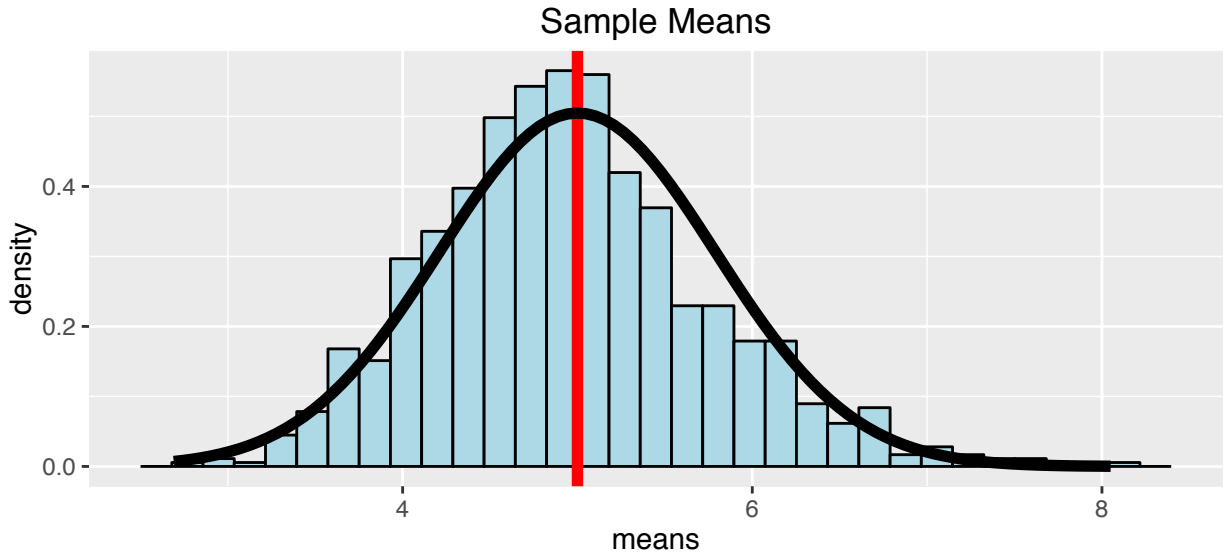
```
var(means);
```

```
## [1] 0.6244729
```

It appears that the sample variance slightly underestimates the population variance.

Distribution of the Sample Mean

Overlaying a normal distribution plot on the the histogram of means shows that the distribution of means is approximately normal.



Conclusion

The simulated distributions yield the following statistic.

- Sample Mean: 4.95
- Variance of Sample Mean: 0.624

Both are in good agreement with the predictions of the central limit theorem.

Distribution of simulated means is approximately normal.

Appendix 1: Simulating the Exponential Distribution

```
means <- NULL;  # initialize list of means

for( i in 1:1000 ){
#   print(i);
    sim <- rexp(40, rate=0.2); # simulate 40 values with lambda = 0.2
    means <- c(means, mean(sim));  # calculate means
}

mean(means)
var(means)
```

Appendix 2: Histogram of Simulations

```
# plot the histogram
gm <- ggplot() + aes(x=means) + ggtitle("Sample Means");
gm <- gm + geom_histogram(aes(y=..density..), color='black', fill='lightblue');
gm <- gm + geom_vline(xintercept = 5, size=2, color='red');

gm
```

```
# add normal distribution plot to histograms
gm <- gm + stat_function(fun=dnorm, args=list(mean=5, sd=sd(means)), size=2)
gm
```