Investigating The Central Limit Theorem

Richard Ashley

Sunday, August 23, 2015

Project for the Statistical Inference Coursera Class

Overview

In this project we will investigate the exponential distribution in R and compare it with the Central Limit Theorem. The exponential distribution will be simulated in R with `rexp(n, lambda)` where lambda is the rate parameter. For this project we will set `lambda = 0.2` for all of the simulations and we will investigate the distribution of averages of 40 exponentials over one thousand simulations.

Simulations

To create a simulation that looks at the distribution of the average of 40 exponential values we must use the `rexp(n, lambda)` function to generate the 40 independent draws from the exponential and then take the `mean()` of those 40 and repeat 1000 times. Before we do that lets set up the parameters of the distribution and the simulation as follows:

```
lambda   <- 0.2
sim_runs <- 1000
size     <- 40
```

Now lets create a loop the generates the `mean()` of the each of the 40 samples and store the results in `sim_means` so that we can graph and analyise the results later.

```
sim_mean = NULL
for (i in 1 : sim_runs) sim_mean = c(sim_mean, mean(rexp(size,lambda)))
```

Here you can see that the simulation is run `sim_runs=1000` times and the mean is calculated for `size=40` samples from an exponential distribution with `lambda=0.2`. Each subsequent simulation run is concatinated on to the end of the `sim_mean` array.

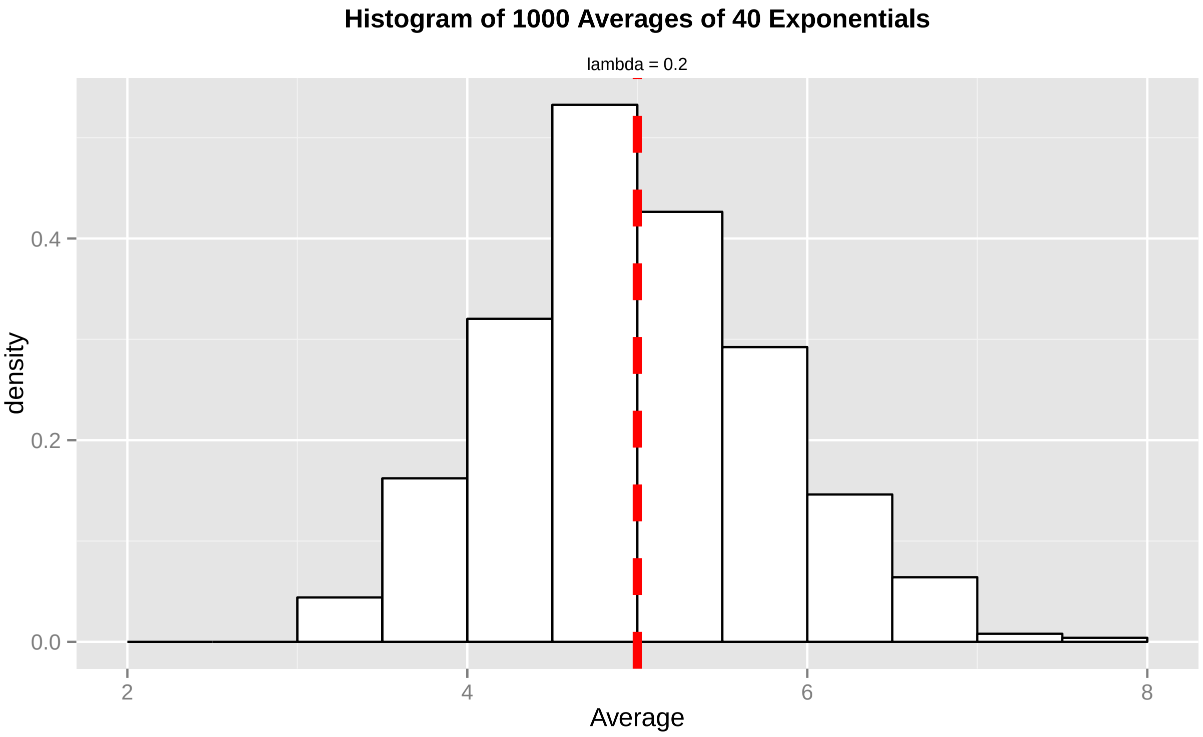
1. Show the sample mean and compare it to the theoretical mean of the distribution

From the Cetral Limit Therom, the theoretical mean of the simulation means should be the same as the mean of the distribution. In this case the mean is $\mu = \frac{1}{\lambda}$.

Value	Mean Result
Thoretical	5.00000
Sample	4.99970

As you can see the Thoretical and Sample mean are very close.

The histogram of the results is a follows and has the sample mean highlited with a red dashed line. Visually, you can see the sample mean is almost exactly 5.



2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution

From the Cetral Limit Therom, the theoretical variance of the simulation means should be the same as the mean of the distribution. In this case the mean is $\mathrm{Var}[X] = \frac{\sigma^2}{\mathrm{N}}$.

Value	Variance Result
Thoretical	0.62500
Sample	0.64324

As you can see the Thoretical and Sample Variance are very close.

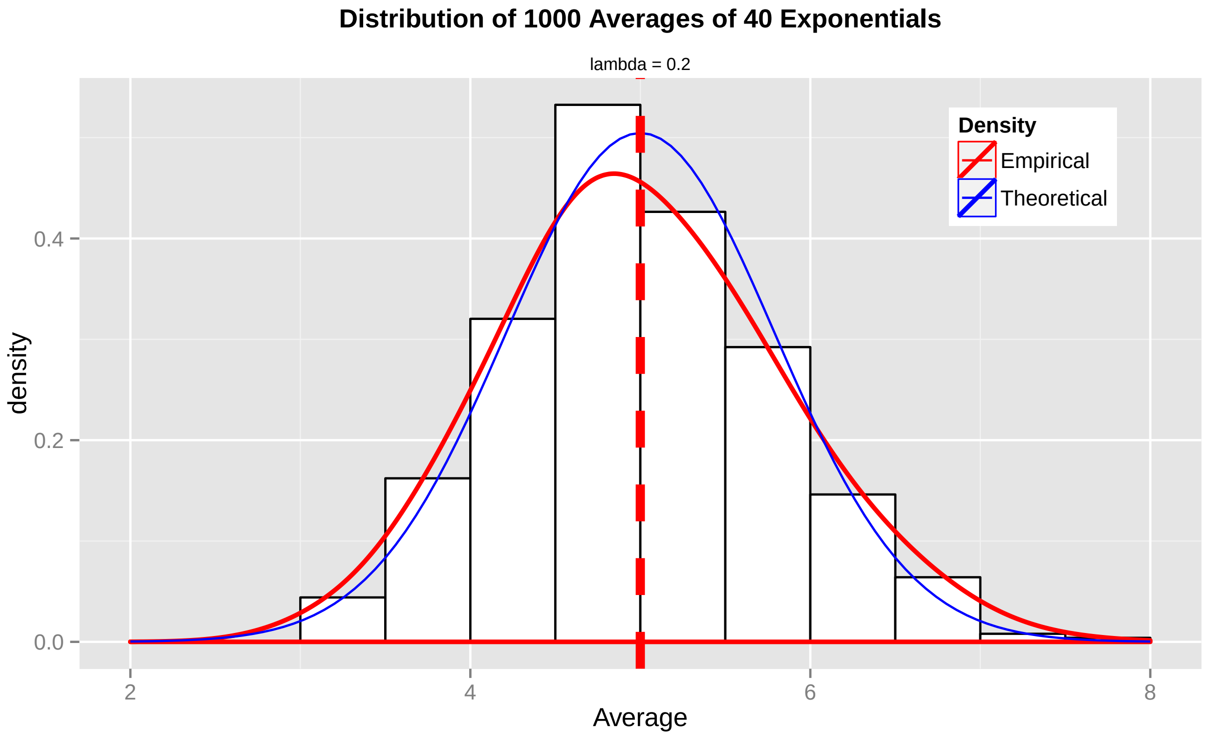
The spread of the sample distribuion can also be seen in the histogram above. The distribution of the mean looks very gausian and we will look at that in the next question.

Distribution

In this section we will look at the shape of the distribution of the 1000 means of the 40 exponential distribution. We will compare the sample results to a normal distribution.

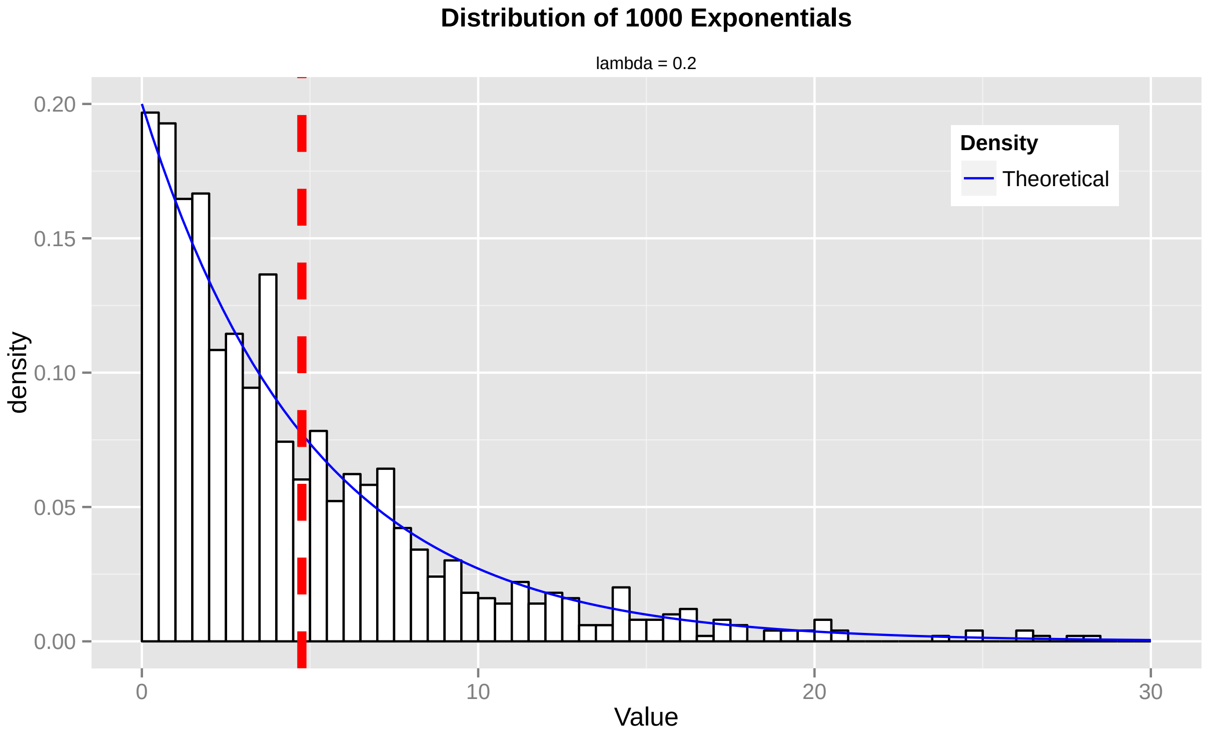
3. Show that the distribution is approximately normal.

To demonstrate this, we will first look at a smoothed density plot of the the simulation results and overlay the thoretical distribution to compare.



You can see that the Theoretical Normal distribution is very simular to the Sampled Emprical results from the simulation.

In contrast, here is the distribuion of 1000 exponentials that are not averaged. The Mean is the same as as above.



Code

```r
library(knitr)
library(markdown)
library(ggplot2)
library(nortest)
dev.off()


set.seed(100)

### Set up parameters of the simulation
lambda   <- 0.2
sim_runs <- 1000
size     <- 40

### Execute the simulation
im_mean = NULL
for (i in 1 : sim_runs) sim_mean = c(sim_mean, mean(rexp(size,lambda)))

### Calculate the Thorectical and Sample statistics
paste("Thoretical Mean =", format(round(1/lambda, 5), nsmall = 5))
paste("Sample Mean =", format(round(mean(sim_mean), 5), nsmall = 5))
paste("Thoretical Variance =" , format(round((1/lambda)^2/size, 5), nsmall = 5))
paste("Sample Variance =", format(round(var(sim_mean), 5), nsmall = 5))


### Generate histogram of the results
m <- ggplot(as.data.frame(sim_mean), aes(x=sim_mean))
m <- m + geom_histogram(aes(y = ..density..), binwidth = 0.5, colour = "black",
                        fill = "white")
m <- m + geom_vline(aes(xintercept=mean(sim_mean)),  color="red",
                        linetype="dashed", size=2)
m <- m + xlim(2,8) + labs(x = "Average")
m + ggtitle(expression(atop(bold("Histogram of 1000 Averages of 40 Exponentials"),
                        scriptstyle("lambda = 0.2")))) +
    theme(plot.title = element_text(size = 12))

### Generate histogram, density, and theoretical distributions
m <- ggplot(as.data.frame(sim_mean), aes(x=sim_mean))
m <- m + geom_histogram(aes(y = ..density..), binwidth = 0.5, colour = "black",
                        fill = "white")
m <- m + geom_density(adjust=2, aes(colour="Empirical"), size=1)
m <- m + stat_function(fun = dnorm, args = list(mean = 1/lambda,
                        sd = (1/lambda)/sqrt(size)),aes(colour = 'Normal'))
m <- m + geom_vline(aes(xintercept=mean(sim_mean)),  color="red",
                        linetype="dashed", size=2)
m <- m + xlim(2,8) + labs(x = "Average")
m <- m +scale_colour_manual(name = 'Density', values = c('red', 'blue'))
m + ggtitle(expression(atop(bold("Distribution of the Averages of 40 Exponentials"),
                        scriptstyle("lambda = 0.2")))) +
    theme(plot.title = element_text(size = 20)) +
    theme(legend.position = c(0.85, 0.85))
```

```
### Generate histogram and Theoretical of exponential
sim <- rexp(1000,lambda)
m <- ggplot(as.data.frame(sim), aes(x=sim))
m <- m + geom_histogram(aes(y = ..density..), binwidth = 0.5, colour = "black",
                        fill = "white")
m <- m + stat_function(fun = dexp, args = list(rate = lambda),
                       aes(colour = 'Theoretical'))
m <- m + geom_vline(aes(xintercept=mean(sim)),  color="red",
                        linetype="dashed", size=2)
m <- m  + labs(x = "Value") +scale_colour_manual(name = 'Density',
                        values = c('blue'))  + xlim(0,30)
m + ggtitle(expression(atop(bold("Distribution of 1000 Exponentials"),
                            scriptstyle("lambda = 0.2")))) +
    theme(plot.title = element_text(size = 20)) +
    theme(legend.position = c(0.85, 0.85))
```