JHU 06 Course Project part 1 - Central Limit Theorem

Kamran Haroon

June 21, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Overview

This report is part submission for the class project in Statistical Inference - a course in the John Hopkins University Data Science specialization on Coursera. As per the project webpage, this report investigates the exponential distribution in R and compare it with the Central Limit Theorem. The project requirements are to illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials. The project report should:

1. Show the sample mean and compare it to the theoretical mean of the distribution.
2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
3. Show that the distribution is approximately normal.

Building the Simulation

Fortunately, the simulation can be easily run in R using the rexp(n,l) function, where n is the number of exponentials and l or lambda, is a constant set to 0.2 for our report. The following R code is used:

```
# Set random seed and initialize variables
set.seed(123456)
l <- 0.2     # lambda
n <- 40      # number of exponentials
s <- 1000    # number of simulations

# Create a data frame for the simulation size
df <- data.frame(mean=numeric(s))

# Iterate 1 to the number of simulations
for (i in 1:s) {
        sim1 <- rexp(n,l)
        df[i,1] <- mean(sim1)
}
```

Comparing the sample mean and theoretical mean of the distribution

After running our simulation, we can compare the sample mean with the theoretical mean. The sample mean can be defined as the average of the iterated values:

```
sm <- mean(df$mean)      # sample mean
format(round(sm, 2), nsmall=2)

## [1] "5.02"
```

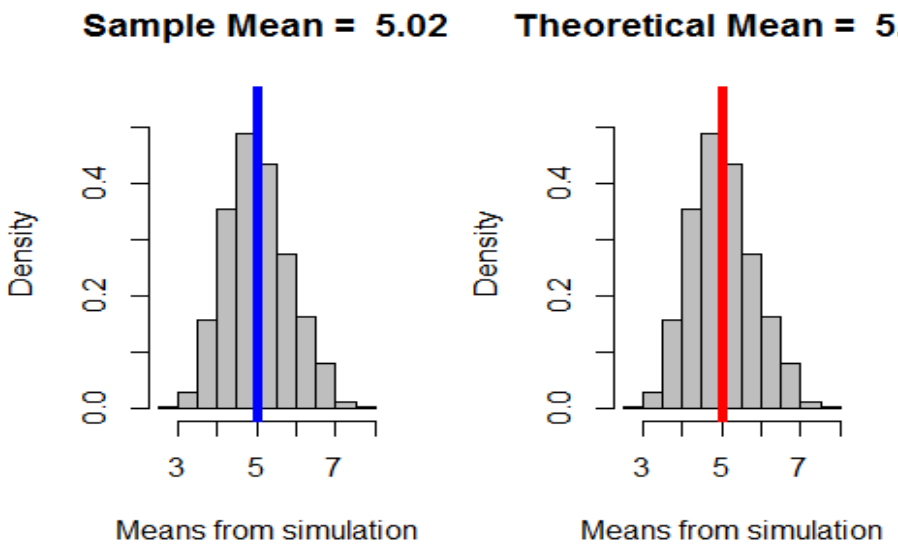
The theoretical mean is given by 1/lambda. For the theoretical mean:

```
tm <- 1/l                # theoretical mean
format(round(tm, 2), nsmall=2)

## [1] "5.00"
```

We can see that the sample mean (5.02) is near identical to theoretical mean (5.0). Plotting the sample observations and the calculated means, the similarly shaped charts seem to confirm that both the means are quite close.

```
par(mfrow=c(1, 2))
hist(df$mean, probability = T, main = paste("Sample Mean = ", format(round(sm,
2), nsmall = 2)), ylim = c(0, 0.55), col = 'gray', xlab = "Means from
simulation")
abline(v = sm, col = 'blue', lwd = 5)
hist(df$mean, probability = T, main = paste("Theoretical Mean = ",
format(round(tm, 2), nsmall = 2)), ylim = c(0, 0.55), col = 'gray', xlab = "Means
from simulation")
abline(v = tm, col = 'red', lwd = 5)
```



Comparing sample variance and theoretical variance of the distribution

We now proceed to compare variance of the sample and theoretical variance of the distribution. The var function in R is used to determine the actual variance of the observations. The sample variance R code is:

```
sv <- var(df$mean)       # sample variance
format(round(sv, 3), nsmall = 3)

## [1] "0.657"
```

We know that the theorical mean is defined as (1/lambda)^2 divided by the number of exponential observations (n). In R, we can calculate the theoretical mean as:

```
tv <- ((1/l)^2)/n       # theoretical variance
format(round(tv, 3), nsmall = 3)

## [1] "0.625"
```

We observe that the sample and theoretical variances are quite close. This is good as the sample data is distributed as expected.

Plotting the distribution curves

Finally, we overlay the actual and normal distribution curves over our sample observation means. It can be clearly observed that our sample set follows a normal distribution.

```
par(mfrow=c(1, 1))
hist(scale(df$mean), probability = T, main = '', ylim = c(0,0.5), xlab = '', col
= 'gray')
curve(dnorm(x, 0, 1), -3, 3, col = 'red', add = T)      # normal distribution
curve
lines(density(scale(df$mean)), col = 'blue')            # actual distribution
curve
legend(2, 0.4, c('Normal', 'Actual'), cex = 0.8, col = c('red', 'blue'), lty = 1)
```