Statistical Inference Course Project

Chris van Hasselt

Sunday, May 24, 2015

Overview:

The exponential distribution is simulated in R with *rexp(n, lambda)* where *lambda* is the rate parameter. The mean of the exponential distribution is *1/lambda* and the standard deviation is *1/lambda*. This report examines the distribution of averages of 40 exponentials with rate *lambda = 0.2*, using plots generated with the *ggplot2* library, illustrating important properties of the distribution:

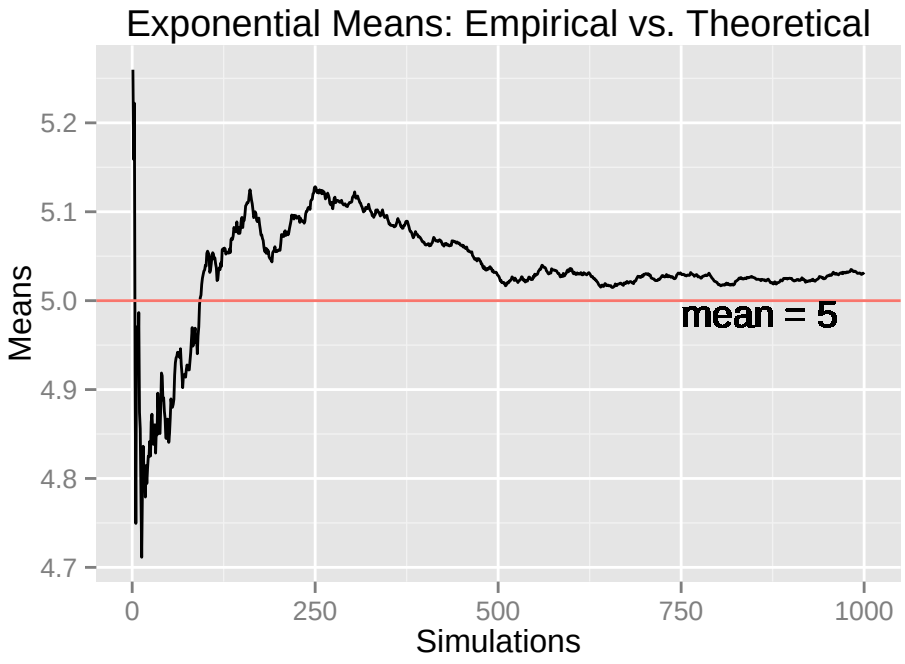
1. The sample mean converges to the theoretical mean of the distribution;
2. The sample variance converges to the theoretical variance of the distribution;
3. The distribution is approximately normal.

Simulations:

To analyze empirical versus theoretical statistics, I create a data.frame named *dfSamples* of random exponentials with rate *lambda = 0.2*. Code available in appendices.

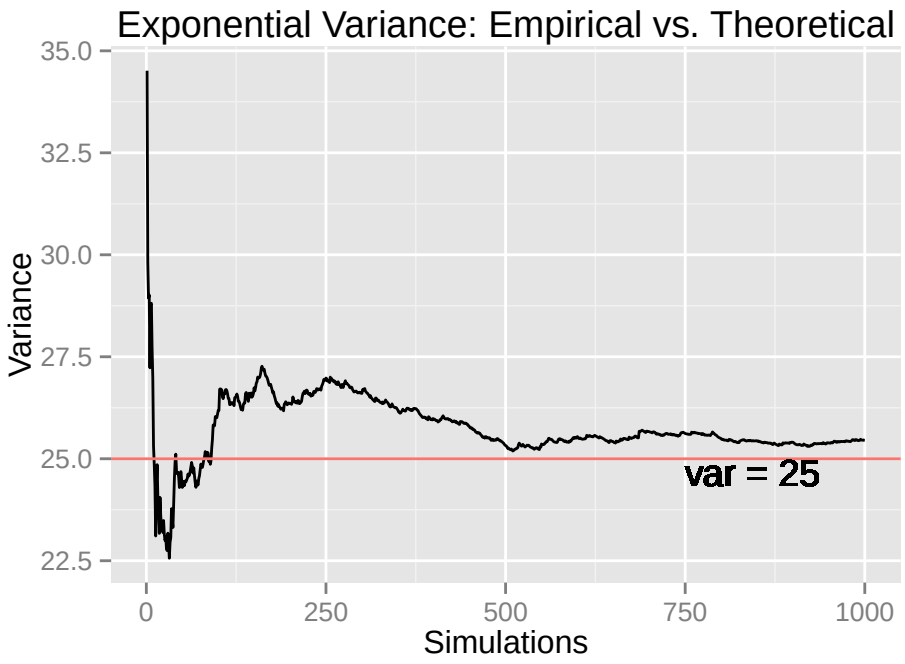
Sample Mean versus Theoretical Mean:

Comparing the empirical mean to the theoretical mean, I calculate row means and cumulative row means for data.frame *dfSamples*. The plot below shows the sample means of 40 exponentials converging to the theoretical mean of *1/lambda = 5*, demonstrating asymptotic convergence.



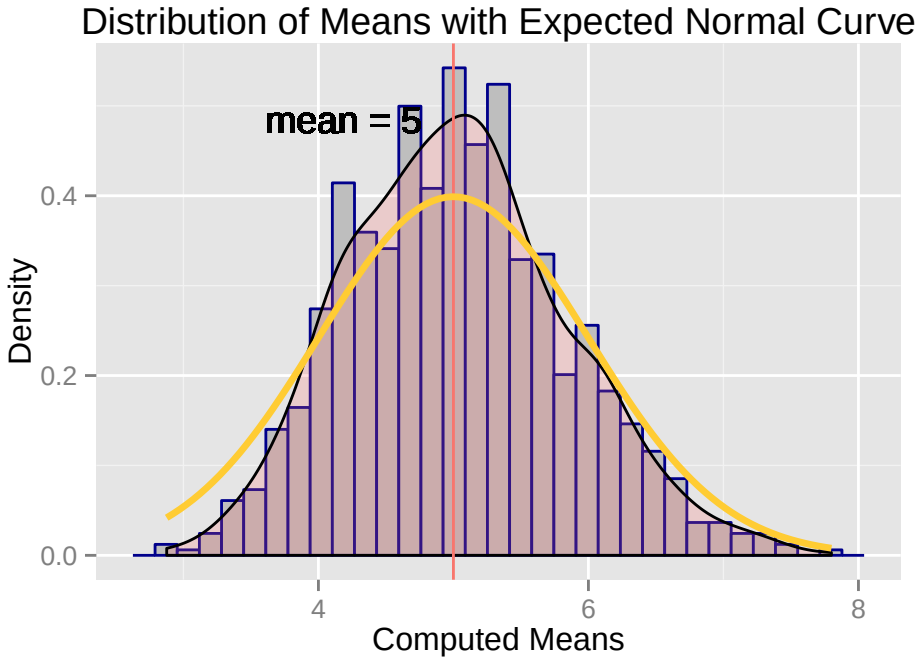
Sample Variance versus Theoretical Variance

Plotting the rolling value of the variance as n increases, we see it converging to the expected theoretical variance *(1/lambda) ^ 2 = 25* (red line), demonstrating asymptotic convergence of the variance.



The Distribution of the Means

Plotting the histogram with its empirical density overlaying (pink) shows the distribution is approximately normal, centered around the theoretical mean. The normal distribution (orange) is also shown.



Appendices: Code Samples & Plots

Libraries

```
# libraries loaded here
library(ggplot2)
```

Generating Data

In the code below R generates a 1000 samples of 40 exponentials with lambda value of 0.2, converting it immediately to a data.frame named *dfSamples*.

```
set.seed(2132)          # set random seed, for repeatability

lambda <- 0.2           # define lambda, the exponential rate

numSimulations <- 1000  # numSimulations = number of simulations (1000)

sampleSize <- 40        # sampleSize = number of exponentials per simulation (40)

# allSamples is the numSimulations (1000) times sampleSize (40) exponentials
allSamples <- rexp(sampleSize*numSimulations, rate=lambda)

# sampleMatrix packages the exponentials as a matrix
sampleMatrix <- matrix(allSamples, numSimulations)

# dfSamples is a data.frame of all the samples
dfSamples   <- as.data.frame(sampleMatrix)
```

Convergence of the Mean

The code below computes the means for each row of 40 samples, and prepares the plot demonstrating asymptotic convergence of the mean.

```
# dfSamples$xbar is a column appended of row means
dfSamples$xbar <- apply(dfSamples, 1, mean)

# dfSamples$xbarCumulative is appended as a column of cumulative means
dfSamples$xbarCumulative <- cumsum(dfSamples$xbar)/(1:numSimulations)

xbarTheoretical <- 1 / lambda  # the theoretical mean

# Prepare plot of cumulative means, demonstrating convergence to the theoretical mean
meanPlot <- ggplot(data.frame(x = 1:numSimulations,
                              y = dfSamples$xbarCumulative),
            aes(x = x, y = y)) + geom_line() +
            geom_hline(aes(yintercept=xbarTheoretical,
                           color="red")) +
                geom_text(aes(750,xbarTheoretical,
                              label = "mean = 5",
```

```
                              vjust = 1,
                              hjust = 0)) +
            labs(title='Exponential Means: Empirical vs. Theoretical',
                 x='Simulations',y='Means')

# display the plot
meanPlot
```

Convergence of the Variance

The code below is computes the variance for each row of 40 samples, and prepares the plot demonstrating asymptotic convergence of the mean.

```
# Compute variance for each sample of 40 exponentials, and append to  data.frame
dfSamples$varEmpirical <- apply(dfSamples[,paste("V",1:40,sep="")],1,var)

# Compute a rolling estimate of variance as the number of samples increases.
dfSamples$varCumulative <- cumsum(dfSamples$varEmpirical)/(1:numSimulations)

# compute the theoretical variance
varTheoretical <- (1/lambda) ^ 2

# Prepate plot
varPlot <- ggplot(data.frame(x = 1:numSimulations,
                             y = dfSamples$varCumulative),
            aes(x = x, y = y)) +
            geom_line() +
            geom_hline(aes(yintercept=varTheoretical,
                           color="red")) +
                geom_text(aes(750,varTheoretical,
                              label = "var = 25",
                              vjust = 1,
                              hjust = 0)) +
            labs(title='Exponential Variance: Empirical vs. Theoretical',
                 x='Simulations',y='Variance')

# diplay plot
varPlot
```

The Distribution of the Means

The code below creates the plot of the empirical mean as a histogram, with a density function overlaying it, and the true normal distribution displayed as well. The plot shows that the empirical distribution is approximately normal.

```
# display a plot of the histogram;
xbarDist <- ggplot(data=dfSamples, aes(dfSamples$xbar)) +
                geom_histogram(aes(y=..density..),
                               col="darkblue",fill="grey",
                               binwidth=diff(range(dfSamples$xbar))/30) +
                geom_density(alpha=.2, fill="#FF6666") +
                geom_vline(aes(xintercept=xbarTheoretical,
```

```
                          color="red")) +
                geom_text(aes(xbarTheoretical,.5,
                              label = "mean = 5",
                              vjust = 1,
                              hjust = 1.2)) +
                stat_function(fun = dnorm, arg=list(mean=xbarTheoretical),
                              colour = "#FFCC33",
                              size=1.2) +
                labs(title='Distribution of Means with Expected Normal Curve',
                     x='Computed Means',
                     y='Density')
# plot the distribution
xbarDist
```