Expontential Distribution Exploration

Wesley Small (smallwesley)

May 2016

OVERVIEW

In this report, together, we shall investigate the exponential distribution and compare it with the Central Limit Theorem (CLT). With the Central Limit Theorem, the rule states that the distribution of average of IID (Independant & Identically Distributed) variables, (when properly normalized), becomes that of a standard normal distribution. This is more evident as the sample size increases. We will define the exponential formula and investigate the distribution of a large sampling of averages of groups 40 exponentials.

The will be 5 main sections to this report: 1. What do we know? Break down of the known parameters/variables for running this experiment. 2. Simulation & Illustration of the Exponential Distribution. 3. Comparison of the sample mean to the theoretical mean of the distribution. 4. Comparison of variability between the sample variance to the theoretical variance of the distribution. 5. Dicussion about the distribution. Is this it related to a normal distribution?

Reference and Resources:

Please note the following resources for providing clarity in executing these report experiments:

- Course Lecture Video:
- https://www.coursera.org/learn/statistical-inference/lecture/K2IVE/07-02-asymptotics-and-the-clt
- https://www.coursera.org/learn/statistical-inference/lecture/fgsnk/06-02-normal-distribution
- Course Lecture Notes: https://github.com/bcaffo/courses/blob/master/06_StatisticalInference/07_Asymptopia/index.Rmd
- Online Stats Site: Statistics Explained: http://www.animatedsoftware.com/statglos/sgcltheo.htm
- Additional Lectures
- Standard Deviation: https://www.youtube.com/watch?v=dq_D30kyR1A
- CLT: https://www.youtube.com/watch?v=zr-97MVZYb0

Section 1. What do we know?

Parameter	Variable	Value	Notes
Exponential Distribution	getExpDist	function(n, lambda) rexp(n,lambda)	See Param-Note 1 below
Mean	getExpDistMean	function(lambda) 1/lambda	
Standard Deviation	getExpDistStdDev	function(lambda) 1/lambda	
Rate Parameter	lambda	0.2	
Observations Count	n	40	
Simulation Count	nosim	1000	
Theoretical Mean	expDistMean	getExpDistMean(lambda) = 5	

Param-Note 1: The expontential distribution can be simulated with this function *getExpDist*, where *lambda* is a rate parameter. We will pass in the number of exponentials we want to be calculated.

SECTION 2. SIMULATIONS

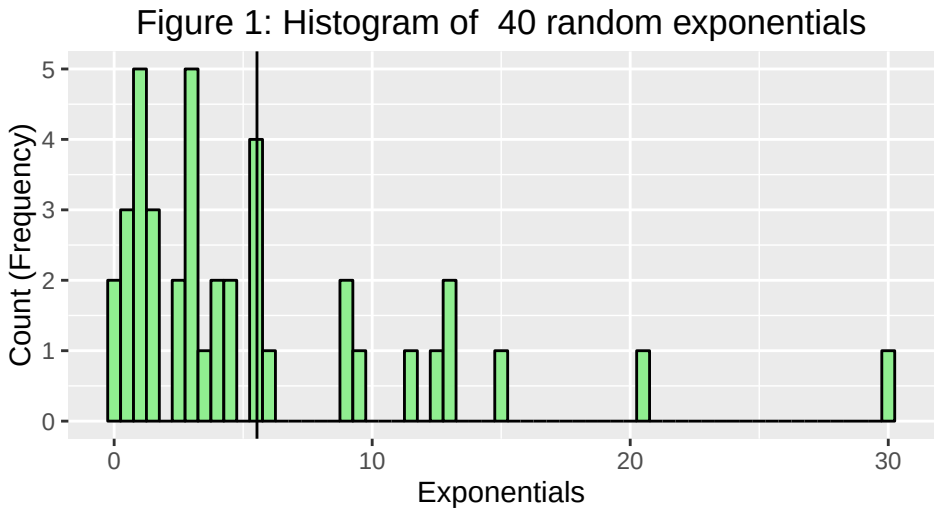
We will start off with an examination of the exponential distribution formula. Well attempt an initial test (n = 40 => number of observations). Please note the other parameters here: *lambda* set to *0.2*.

```
singleSampleDist <- getExpDist(n, lambda)
singleSampleDistMean <- mean(singleSampleDist)
```

The mean of this initial sampling is 5.53

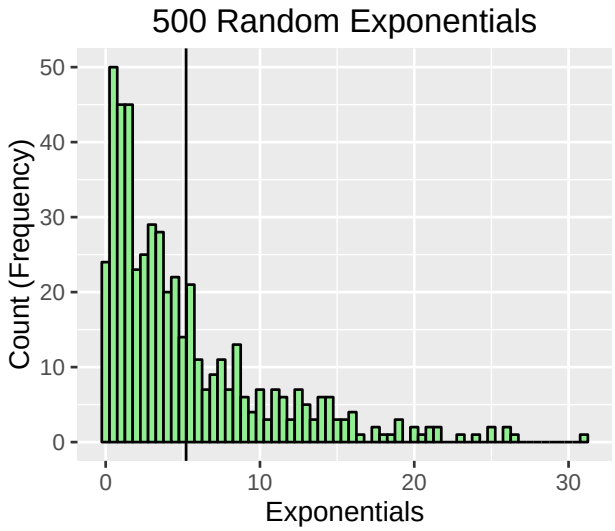
Here, we take a quick peak about how *40* random exponentials are distributed.

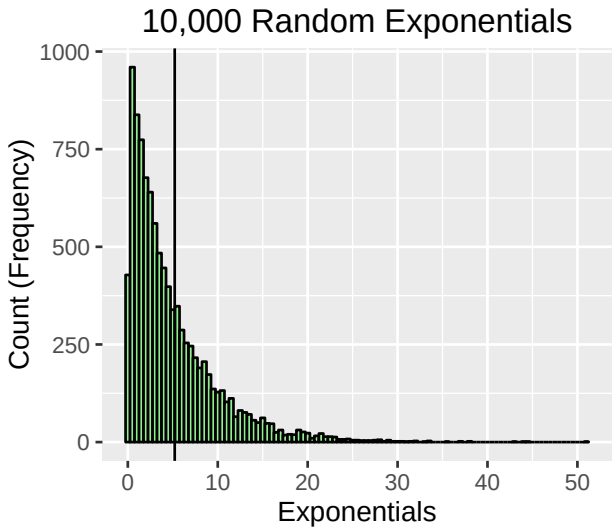
```
qplot( singleSampleDist, geom = "histogram", binwidth = 0.5,
    fill = I("lightgreen"), colour = I("black"), ylab = "Count (Frequency)",
    main = "Figure 1: Histogram of  40 random exponentials", xlab = "Exponentials") +
  geom_vline(xintercept =  mean(singleSampleDist))
```



From this graph, we cannot yet dicern a pattern about if the single average of only 40 IID variables coalese toward the theoretical mean of exponential distribution. Lets try with a large set of random observations, say n = 500 and 10000 respectively

```
grid.arrange( graph500, graph10000, ncol=2)
```





Simulations results:

Observation Count	Mean
500	4.95
10000	4.93

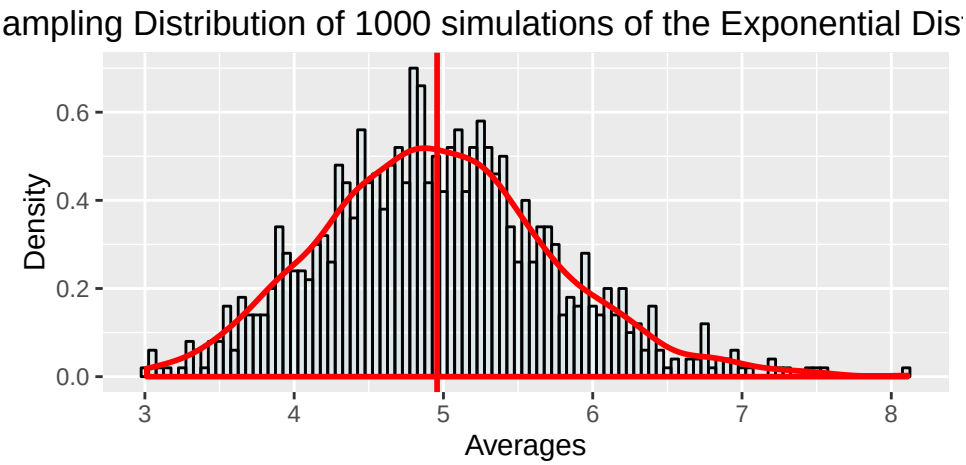
Primary simulation: 1000 Averages of "REXP(n=40, lambda=0.2)"

We will now execute a large number simulations and compare the distribution of the averages per each. This is what the central limit theorem ask us to do.

This function below provides a way to conduct our experiment. It will produce a collection of averages of several simulations (nosim = 1000), groups of *40* random exponentials.

```
getSimulationExpDistMeansDataFrame <- function(n, lambda, nosim) {
  output = NULL
  for (i in 1 : nosim) output = c(output, mean(getExpDist(n=n,lambda=lambda)))
  data.frame(x_axis = output)
}
simulationExpDistMeansDF <- getSimulationExpDistMeansDataFrame(n,lambda,nosim)
simulationExpDistMean <- mean(simulationExpDistMeansDF$x_axis)
```

Lets take a peak at our sample popluation.The vertical redline line is the ploted sample mean of all average (basically the average of averages) at 4.96. The density curve illustrate the shape of the plot results.



3. Sample Mean versus Theoretical Mean

As calculated with our sample population, our sample mean (averages of averages) is:

```
round(mean(simulationExpDistMeansDF$x_axis),2)
```

The theorectical mean has been provided by the problem parameters as 1/lambda. We were given a rate of 0.2, thus out average is:

```
getExpDistMean(lambda)
```

We can see that our sample mean of 4.9572982 is very close to our theoretical mean of 5.

4. Sample Variance versus Theoretical Variance

As calculated with our sample populations, our *sample variance* is calculated to be:

```
round(var(simulationExpDistMeansDF$x_axis),2)
```

```
## [1] 0.61
```

To solve for our sample variance we need to compute our true standard sample deviation for our sample size. *Note:* The standard deviation of sample is to be divided by the square root of your sample size.

```
round(getExpDistStdDev(lambda = lambda) / sqrt(n),2)
```

To obtain our variance, we square this results, as this is the computed square distance of all the averages from the mean.

```
(getExpDistStdDev(lambda = lambda) / sqrt(n))^2
```

We can see that our sample variance of 0.61 is very close to our theoretical variance of 0.625.

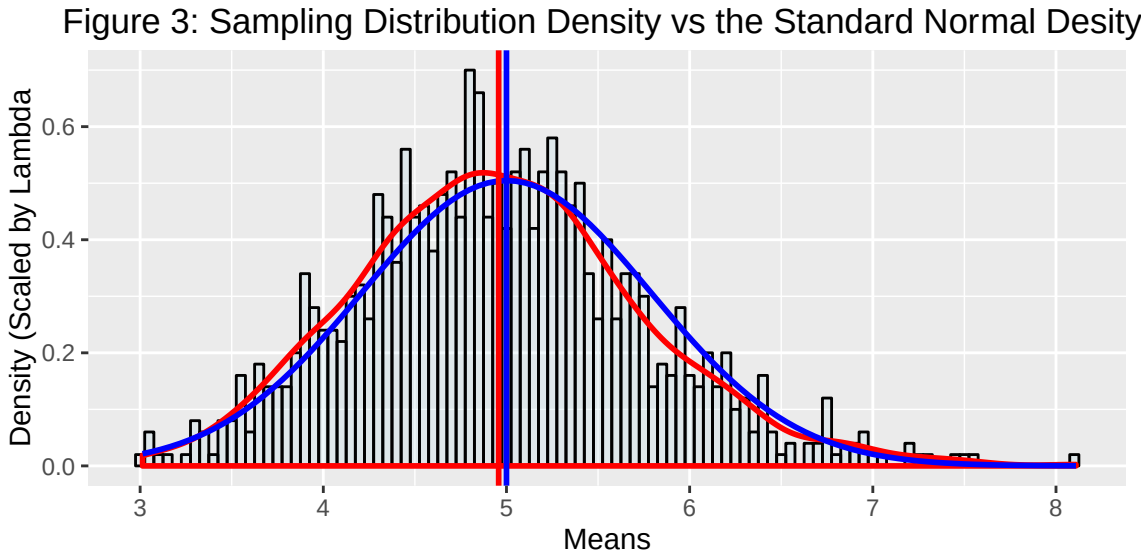
5. Distribution Summary

"The Central Limit Theorem is a statement about the characteristics of the sampling distribution of means of random samples from a given population." (- Quote from Statistic Explained on website : "http://www.animatedsoftware.com/statglos/sgcltheo.htm")

Confirmation A

In other words, after computing sufficiently large number averages of a collection independent random variable sets, *"the sampling distribution"*, the data when plotted will approximately match that of standard normal bell curve, and thus will be normally distributed.

```
ggplot(
  data = simulationExpDistMeansDF) +     # SET DATA
  aes(x = x_axis) +                      # SET X AXIS
  geom_histogram(                        # PLOT HISTOGRAM
    binwidth=0.05, alpha=lambda, aes(y=..density..), colour="black", fill="light blue") +
  labs(                                  # ADD LABELS
       title="Figure 3: Sampling Distribution Density vs the Standard Normal Desity ",
       x="Means", y="Density (Scaled by Lambda") +
  scale_x_continuous(                    # SET X-AXIS SCALE
    breaks=scaleStepBreaks(expDistMean,4)) +
  geom_density(                          # SAMPLE MEANS DENSITY CURVE (RED)
    colour="red", size= 1) +
  stat_function(                         # SET DISTRIBUTED NORMAL DENSITY CURVE (BLUE)
    fun = dnorm,  size = 1, colour = "blue",
    args = list(mean = expDistMean, sd = getExpDistStdDev(lambda = lambda) / sqrt(n))) +
  geom_vline(                            #  SAMPLE MEAN LINE (RED)
    xintercept = simulationExpDistMean, size=1, colour="red", show.legend = TRUE) +
  geom_vline(                            #THEORETICAL MEAN LINE (BLUE)
    xintercept = expDistMean, size=1, colour="blue", show.legend = TRUE) +
  scale_colour_manual("",
    breaks = c("red", "blue"),values = c( "red"="red", "red"="blue"))
```



The overlay of the standard normal bell curve (in Blue) scaled have the mean at 1/lambda = 5. As you can see the distribution shown by the red density curve approximately matches the blue standard normal curve.

Confirmation B:

The variance of the sample distribution is approximately equal to the theoretical variance.

Confirmation C:

Also for a sampling distribution to be normal it should statify attributes of a standard normal:

- Where 68% distribution lies within 1 standard deviation from it's distribution mean.
- Secondly, 97% lies within 2 standard deviations awat from from distribution mean.
- Lastly, 99% lies within 3 standard deviations from the distribution mean.

```
samp <- simulationExpDistMeansDF$x_axis
samp_mu <- mean(samp)
samp_sd <- sd(samp)

# Calculate Standard Deviation Endpoints; Interval based on distance from the mean
getStdDevSampleInterval <- function(g, mu, stddev) c(samp_mu - samp_sd * g, samp_mu + samp_sd * g)

# For a list of results, calculate the precentage of results that lie within the stand
calculateSamplePercentageWithinStdDevInterval <- function(g, list, mu, stddev) {
  sdGroup <- getStdDevSampleInterval(g, mum, stddev)
  validSdGroup <- NULL
  for ( i in 1:length(list) )
    if (list[i] >= sdGroup[1] & list[i] <= sdGroup[2]) validSdGroup <- c(validSdGroup, list[[i]])
  length(validSdGroup)/length(samp)
}
```

Sample mean for sample population = 4.9572982

Standard deviation for sample population = 0.7798874

Standard Deviation	Sampling Distrbution Interval	Coverage Percentage
1	4.1774108, 5.7371857	0.697
2	3.3975233, 6.5170731	0.955
3	2.6176359, 7.2969606	0.996

Additional Info:

You can reference the MarkDown Rmd Source @

https://github.com/smallwesley/statistical-inference-assignement/blob/master/part1.Rmd