Assignment #1: Statistical Inference Course

B Mahoney

May 16, 2016

Some Statistical Inference Analyses - Coursera Data Science Specialization

This file encompasses the first of two separate data investigations required under the Project Assignment for the course, Statistical Inference, offered through Coursera in May, 2016.

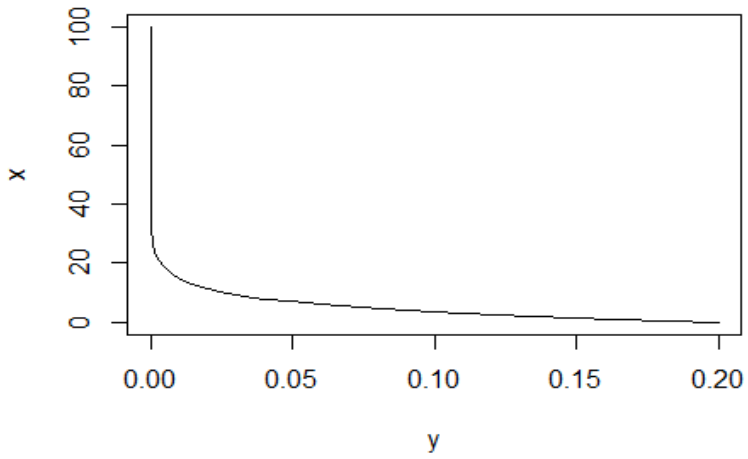
Part 1 - Exponential Distribution Overview

Assignment: In this project we investigate the exponential distribution in R and compare it with the Central Limit Theorem.

The Exponential Distribution is a continuous distribution used to model the waiting time for an event to occur. It is (from Statistics and Data Analysis, Ajit C. Tamhane, Dorothy D. Dunlop),"...the simplest distribution used to model the times to failure (lifetimes) of items or survival times of patients."

The probability density function of the exponential distribution is $f(x) = \lambda * e^{-\lambda * x}$, defined on the interval $[0, \infty]$, where λ is the failure rate (expressed as the average number of failures per unit time).

We can plot a segment of the p.d.f. for x in the interval [0,100] and with λ equal to 0.2.



The theoretical mean of the exponential distribution (equal to the area under this curve (on the $[0, \infty]$ interval) is equal to $\frac{1}{\lambda}$; the theoretical standard deviation of this distribution is also equal to $\frac{1}{\lambda}$.

The objective of this exploration is to "Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials." To do this we will perform 1000 simulations to investigate the distribution of the averages of 40 exponentials, randomly generated using R. The selected λ is set equal to 0.2.

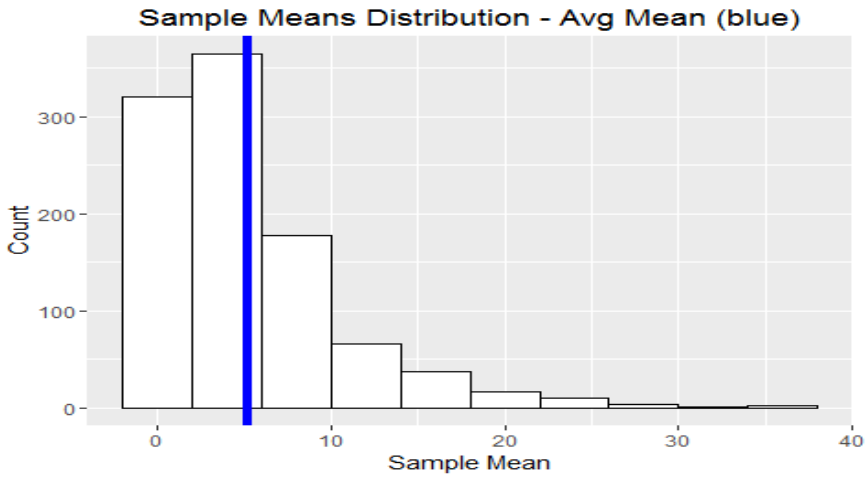
The theoretical mean, for λ of 0.2, is $\frac{1}{\lambda}$, that is, 5. As we have seen, the standard deviation of the theoretical population is equal to the mean, 5.

Simulations

The exponential distribution in R can be simulated with the function, rexp(n,lambda).

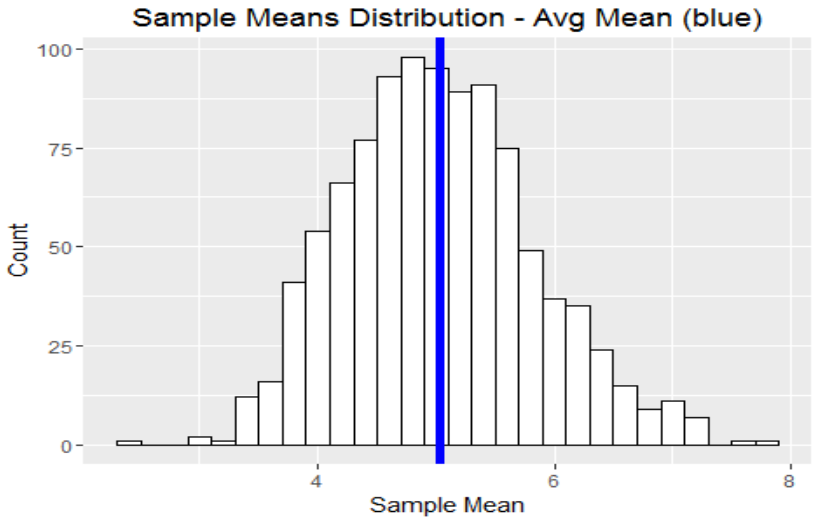
Create a sample dataset, xp, containing 1000 randomly generated exponentials, with the λ in question, 0.2.

Here is the histogram of those 1000 exponentials, roughly following the shape of the line plot of the function, shown above.



Create 1000 sets of 40 randomly generated exponentials and calculate the mean for each set.

We used the template provided in the project assignment based on the uniform distribution, replacing the randomizing function "runif" with "rexp".



Here we see that the distribution of the sample means dataset, mns, does not resemble the exponential curve, but instead appears normally distributed, as the Central Limit Theorem (CLT) would imply.

1. How does the mean of the means dataset compare to the theoretical mean, $\frac{1}{\lambda}$, or 5?

The mean of the simulation dataset, mns, rounded to 3 decimal places, is equal to 5.025, quite close to the theoretical mean, 5, as expected.

2. How does the variance of the mns dataset compare to the theoretical variance of the exponential distribution, for λ of 0.2?

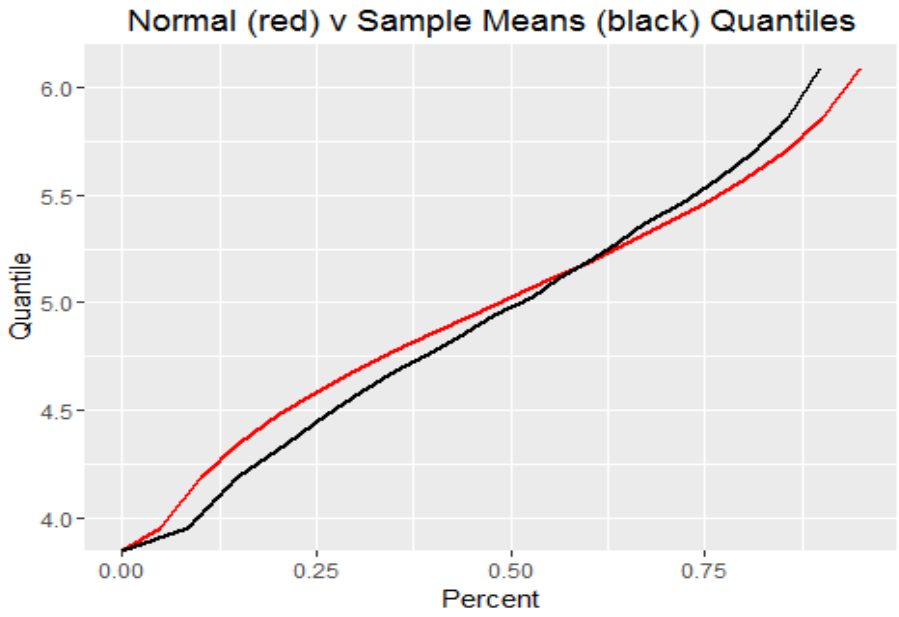
We would expect it to be close to the CLT theoretical variance of sample means, $\frac{\sigma^2}{n}$, where n is 40 in this case and σ^2 is 25, the population variance for the exponential.

The sample variance of the simulation dataset, mns, rounded to 3 decimal places, is 0.652. The variance of the simulation dataset should be close to $\frac{\sigma^2}{n}$, or 0.625, where σ^2 equals 25.

3. Compare selected probabilities for the normal distribution to the percent of means below the respective theoretical quantiles.

If mns approximates the normal distribution, we would expect that the percentage of means in mns below a quantile would correspond to percentages from the normal distribution.

For a selected group of percentages we calculated the corresponding quantiles using the one-sided interval for a normal distribution: the mean of means plus the standard normal quantile times the standard deviation of the means.



The match is quite close in the middle quantiles, less so on the tails. But this evidence, together with the graph of the data, confirms that the exponential means simulation is approximately normally distributed.

Appendix - R code underlying Statistical Inference Project Part 1

This code was used to prepare a segment of the exponential p.d.f. for x in the interval [0,100] and with λ equal to 0.2 for illustrative graph.

```
#plot the pdf of the exponential
lambda=0.2
x=0:100
y=lambda*exp({-lambda*x})
plot(y,x,type="l")
```

Here is the code used to create the sample of 1000 exponentials, and display the histogram of the sample, similar in shape to the line plot of the exponential pdf, shown above.

```
library(ggplot2)
#Establish the seed in order to ensure reproducible results from the random functi
on
set.seed(4)
#First generate 1000 exponentials and plot the histogram.  We will compare this da
taset to the dataset containing averages of 1000 simulations of 40 exponentials.
lambda=.2
nsim=1000
xp=rexp(nsim,lambda)
#Organize as data frame for plotting
xdf=data.frame(xp)

avg=round(mean(xp),3)

#Plot histogram and vertical line to show sample Mean

ggplot(xdf,aes(x=xp))+geom_histogram(color="black",fill="white",binwidth=4)+geom_v
line(aes(xintercept=mean(xdf$xp)),color="blue",size=2)+ggtitle("Sample Means Distr
ibution - Avg Mean (blue)")+labs(x="Sample Mean",y="Count")
```

Below is the code used to execute a simulation of 1000 samples of size 40 from the exponential distribution, and calculate the resulting sample means analyzed subsequently.

```
#Simulate the samples from which means are calculated

#Again, establish the seed to ensure reproducibility
set.seed(4)

library(ggplot2)

#Parameters are identical to the previous simulation, but with the addition of the
n variable, setting the size of each sample set

lambda=.2
n=40
nsim=1000
#Calculate the means, put in mns
mns=NULL
for (i in 1:nsim) mns = c(mns,mean(rexp(n,lambda)))
```

```
#Calculate the mean of mns, and also calculate 1/lambda, the theoretical mean

mmean=round(mean(mns),3)
tmean=1/lambda

#Make data.frame for using ggplot

mnsf=data.frame(mns)

#Plot the histogram and vertical line to show Mean of Means

ggplot(mnsf,aes(x=mns))+geom_histogram(color="black",fill="white",binwidth=.2)+geo
m_vline(aes(xintercept=mean(mnsf$mns)),color="blue",size=2)+ggtitle("Sample Means
Distribution - Avg Mean (blue)")+labs(x="Sample Mean",y="Count")
```

Here is code used to compare normal percentiles with the sample percentiles.

```
library(knitr)

#For a selected group of percentages, calculate the quantiles.  Use the endpoint
#to limit mns and then calculate the proportion of means in mns which are
#strictly less than the selected quantile

#percn holds the percentages we will compare.

percn=c(0.95,0.90,0.85,0.80,0.75,0.70,.65,.60,.55,.50,.45,.40,.35,.30,0.25,.20,.15
,.10,.05,0)
ct=length(percn)
#Now calculate the N(0,1) quantiles corresponding to selected percentages

qns=NULL

for (i in percn) qns=c(qns,qnorm(i))

#Here are the corresponding quantiles for the sample, called ends

ends=mmean + qns*vmean

#Now we want the percentage of means below the sample quantiles

perc=NULL

for(i in 1:ct) perc=c(perc,length(mns[mns<ends[i]])/nsim)

ends=round(ends,3)

#Combine in a single table for display

theostats=cbind(ends,percn,perc)

#Organize as data frame for plotting
```

```r
df=data.frame(theostats)

#Construct the plot to include vertical line marking the Mean of Means

ggplot(df,title="Normal v Sample Mean Quantiles")+geom_line(aes(y=df$ends,x=df$per
cn),color="red",size=1)+geom_line(aes(y=df$ends,x=df$perc),color="black",size=1)+g
gtitle("Normal (red) v Sample Means (black) Quantiles")+labs(x="Percent",y="Quanti
le")
```