Application of the CLT on the Exponential distribution

Spiros Baxevanakis

May 06, 2021

Contents

Overview

In this report we are going to apply the Central Limit Theorem on the Exponential distribution. Using thousands of iid random exponential variables we'll the population mean being approximated by the sample mean, we'll investigate its variance as well as explain how the distribution of averages is approximately normal.

Simulations

In this section we will sample thousands of times the exponential distribution.

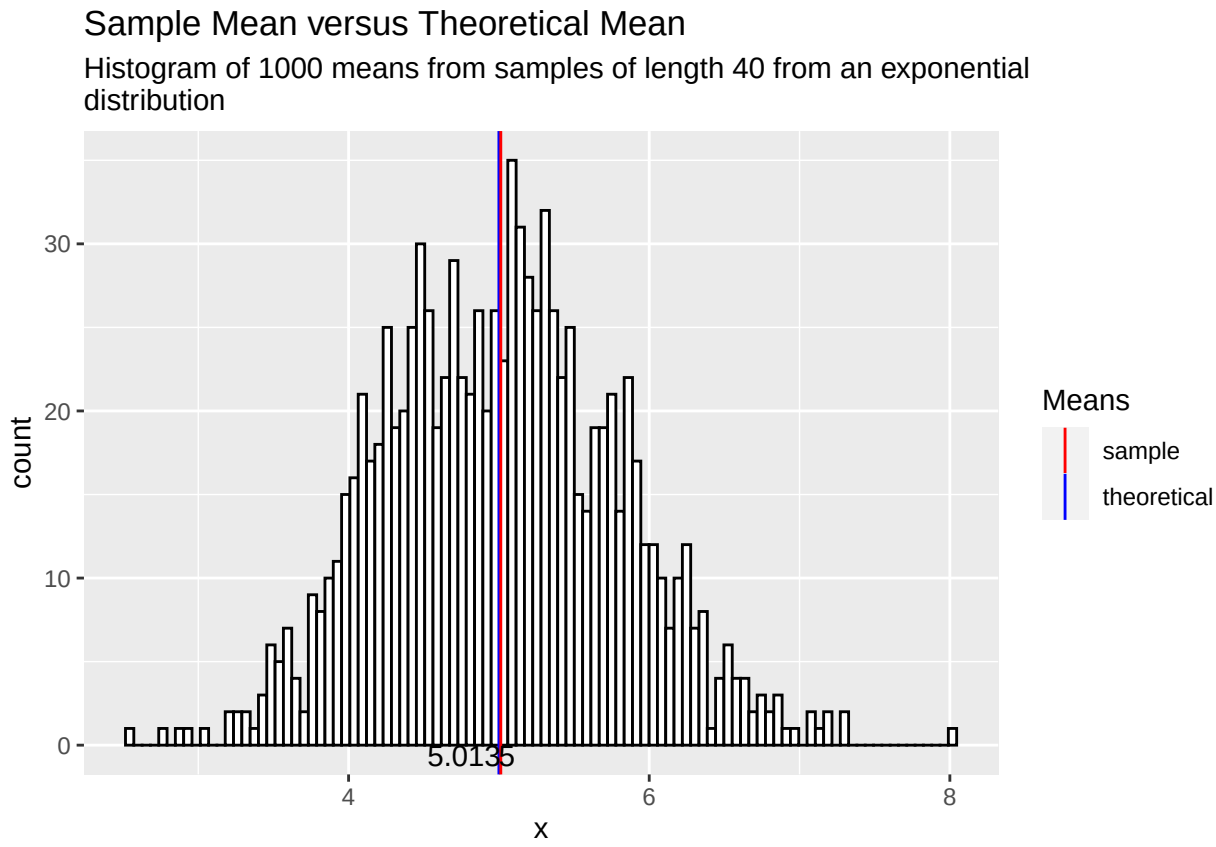
```
lambda = 0.2
tmean = tsd = 1/lambda  # theoretical mean
nexp = 40  # number of random samples from the exponential distribution
B = 1000  # B is the number of samples
tvar = ((1/lambda)/sqrt(nexp))^2  # theoretical varinace
# generate 40 * 1000 random numbers from the exponential distribution with lambda
# = 0.2, then reshape using matrix to (1000,40) matrix. Each row is a sample with
# 40 entries.
runs = matrix(rexp(B * nexp, lambda), B, nexp)
# apply the function mean to each row, this generates 1000 means
runmeans = apply(runs, 1, mean)
```

Sample Mean versus Theoretical Mean

In the figure produced by the code below is a histogram of 1000 means from samples of length 40 from an exponential distribution. The red line is the sample mean 4.980357 while the blue line is the theoretical mean 5.

```
# use ggplot to plot the histogram of the distribution of means contained
# in the variable runmeans. Then plot a vertical line on the theoretical
# mean.
vlines = rbind(
```

```
    data.frame(Ref='theoretical', vals=c(tmean)),
    data.frame(Ref='sample', vals=c(mean(runmeans))))
ggplot(data.frame(x=runmeans), aes(x)) +
    geom_histogram(aes(x), color = "black", fill = "white", bins=100) +
    geom_vline(data = vlines, aes(xintercept = vals, color=Ref)) +
    labs(title = "Sample Mean versus Theoretical Mean",
         subtitle = "Histogram of 1000 means from samples of length 40 from an exponential
distribution") +
    scale_color_manual(name = "Means", values = c(theoretical = "blue",
                                                  sample = "red")) +
    geom_text(data = vlines,
              mapping = aes(x = vals, y = 0,
                            label = c(5,round(mean(runmeans),digits=3)),
                            vjust = c('top','top'),
                            hjust = c('left','right')))
```




```
# print the theoretical and the sample means using a dataframe for pretty printing
data.frame(theoretical_mean = tmean, sample_mean = mean(runmeans)) %>% kable
```

theoretical_mean	sample_mean
5	5.012615

We can see that the theoretical mean is very close to the sample mean, we expect this difference to become smaller as the sample size becomes larger. The Law of Large Numbers guarantees that the sample mean will converge to the theoretical mean as the number of samples n gets close to +infinity.

Sample Variance versus Theoretical Variance

Let's compare the variance of the distribution of averages with the theoretical squared standard error of the mean.

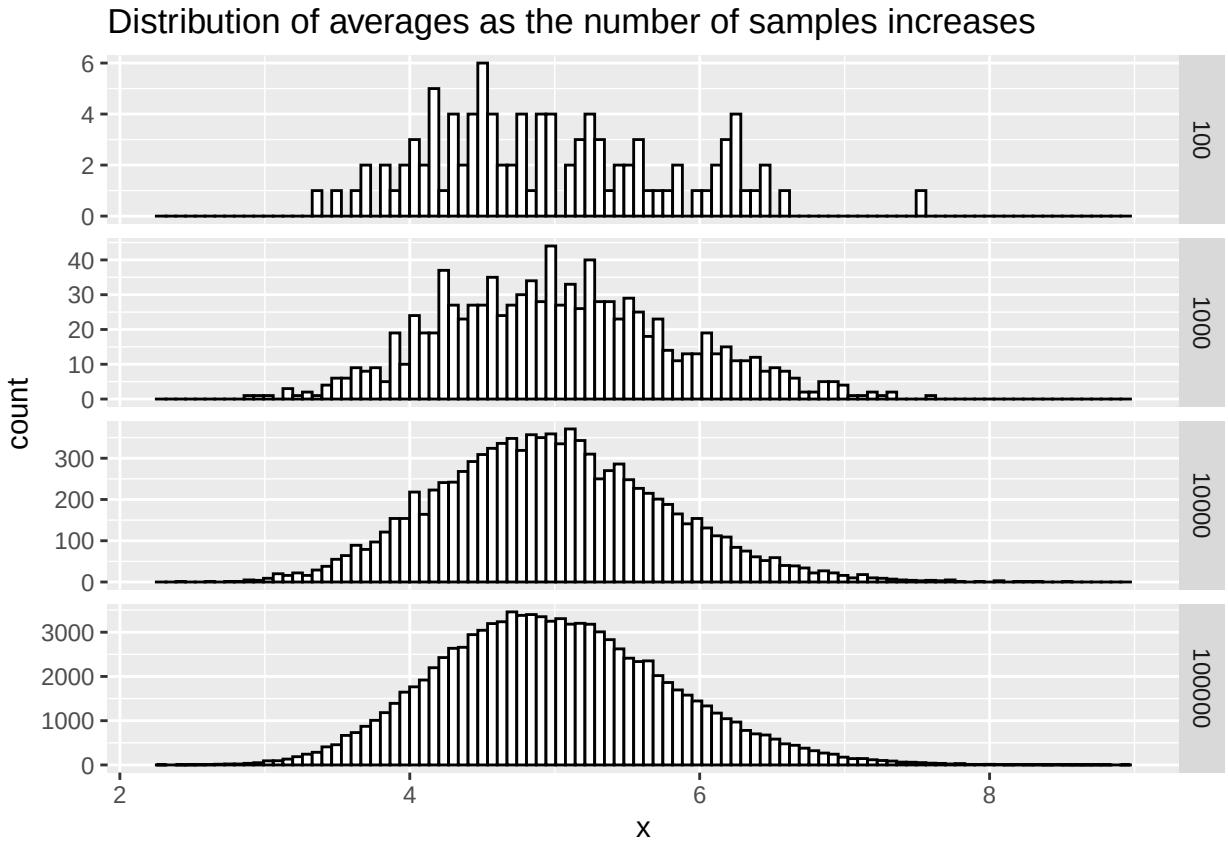
```
# print the variance for the theoretical and distribution of averages
data.frame(theoretical_variance = tvar,
           simulation_variance = var(runmeans)) %>% kable
```

theoretical_variance	simulation_variance
0.625	0.6151721

Distribution

The Central Limit Theorem states that the distribution of averages of iid variables becomes normal as the sample size increases. In the plot produced by the following code, 4 histograms are presented, from top to bottom they represent the distribution of 10^2,10^3,10^4,10^5 averages of 40-length samples for the exponential distribution. It is visually evident that as the number of averages gets larger the distribution look increasingly normal.

```
# the following function samples the exponential distribution B*n times and
# reshapes the samples into a matrix of B,n shape. Then applies the mean function
# to every row.
exp_averages = function(B,n) { apply(matrix(rexp(B*n,lambda),B,n),1,mean) }
# create a dataframe with different number of averages, distringuish them with the
# type column in order to create multiple histograms.
sampleruns = rbind(
    data.frame(x=exp_averages(100,40),type='100'),
    data.frame(x=exp_averages(1000,40),type='1000'),
    data.frame(x=exp_averages(10000,40),type='10000'),
    data.frame(x=exp_averages(100000,40),type='100000')
)
# plot different histograms for each of the type categories
ggplot(data=sampleruns,aes(x)) +
    geom_histogram(aes(x), color='black',fill='white',bins=100) +
    facet_grid(type ~ ., scales = "free_y") +
    labs(title="Distribution of averages as the number of samples increases")
```



If we want the distribution to be a standard normal then we can very easily subtract out the theoretical mean and divide by the standard error of the mean.

```
runmeans = exp_averages(5000, 40)
standard_runmeans = sqrt(length(runmeans)) * ((runmeans - tmean)/tsd)
```

Shapiro-Wilk test of normality

It is very interesting to note that the Shapiro-Wilk test of normality which tests the H0 that the data follow a normal distribution, outputs a pvalue of 1.048e-13. According to this pvalue we have to reject the H0. The same pvalue is given regardless of whether we normalize the data.

```
shapiro.test(runmeans)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  runmeans
## W = 0.99361, p-value = 3.689e-14
```

```
shapiro.test(standard_runmeans)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  standard_runmeans
## W = 0.99361, p-value = 3.689e-14
```

Conclusion about distribution

Although the Shapiro-Wilk test fails, the histogram plots of the data suggest that the data is normally distributed. The **Central Limit Theorem only guarantees normality for an infinite number of samples**. Thus, we can safely conclude that if we were to run the Shapiro-Wilk on averages from an infinite number of samples, it would fail to reject the H0 that the data is normally distributed.

To show that this is indeed the case, consider the following example. We generate 3 5000 averages from 10, 100 and 1000 length samples from the exponential distribution using the function *exp_averages* that we defined earlier. Thus the data frame we create has dimensions 5000x3. Then we run the test on every column and keep only the p values.

```
data.frame(s10 = exp_averages(5000,10),
           s100 = exp_averages(5000,100),
           s1000 = exp_averages(5000,1000)) %>%
    apply(2,function(x) shapiro.test(x)$p.value) %>% kable
```

	x
s10	0.0000000
s100	0.0000044
s1000	0.0291087

It is evident that as the sample size increases the p value becomes less significant which is exactly what we hypothesized and what the CLT predicts.