The Exponential Distribution And The CLT

George Papadopoulos pgeorgios8@gmail.com

14 February 2023

abstract

In this small report we will investigate the process of sampling from a population which follows an exponential distribution with parameter λ, compare the estimations of the mean, variance and standard deviation that can be computed from the samples with the theoretical properties of the distribution and how this process can be connected to **the central limit theorem**. We will create one thousand samples of forty observations from an exponential distribution with $\lambda = 0.2$ and will create a simulation in R to answer three questions,

1. Show where the distribution of the samples' means is centered at and compare it to the theoretical center of the distribution.
2. Show how variable it is and compare it to the theoretical variance of the distribution.
3. Show that the samples' means distribution is approximately normal.

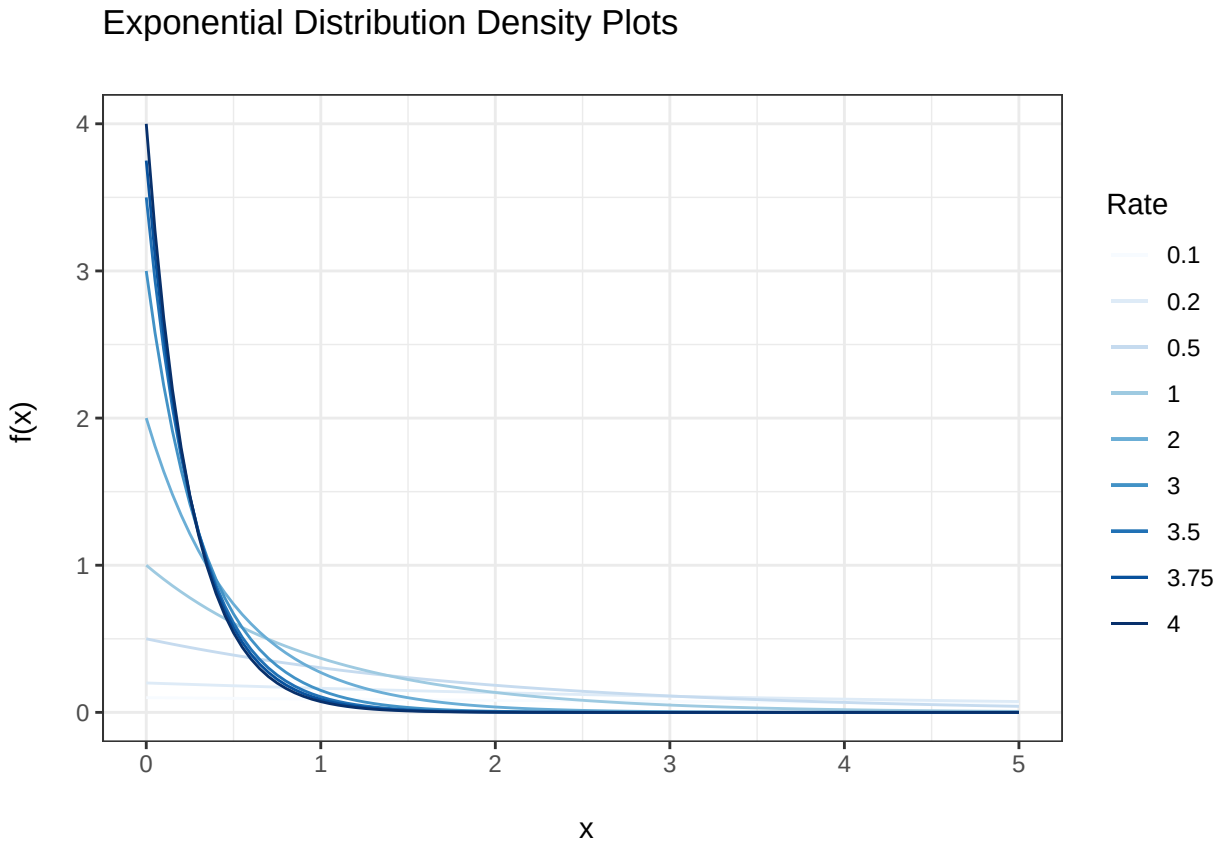
The *exponential* distribution with parameter λ has support in $[0, +\infty]$ with probability density function

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & , x \geq 0 \\ 0 & , \text{otherwise} \end{cases}$$

its mean is λ^{-1} and its variance λ^{-2}. The *exponential* distribution or *negative exponential* distribution is the probability distribution that models the time between the events in a *Poisson* process, a process in which events occur **continuously** and **independently** at a *constant* average rate.

```
library(ggplot2)
library(RColorBrewer)
x_lower <- 0
x_upper <- 5
max_height2 <- max(
  dexp(x_lower:x_upper, rate = 0.1,log = FALSE),
  dexp(x_lower:x_upper, rate = 0.2,log = FALSE),
  dexp(x_lower:x_upper, rate = 0.5,log = FALSE),
  dexp(x_lower:x_upper,rate = 1,log = FALSE),
  dexp(x_lower:x_upper,rate = 2,log = FALSE),
  dexp(x_lower:x_upper,rate = 3,log = FALSE),
  dexp(x_lower:x_upper,rate = 3.5,log = FALSE),
  dexp(x_lower:x_upper,rate = 3.75,log = FALSE),
  dexp(x_lower:x_upper, rate = 4, log = FALSE)
  )
ggplot(data.frame(x = c(x_lower, x_upper)), aes(x = x)) + xlim(x_lower, x_upper) +
     ylim(0, max_height2) +
  stat_function(fun = dexp, args = list(rate = 0.1), aes(colour = "0.1"))    +
```

```
stat_function(fun = dexp, args = list(rate = 0.2), aes(colour = "0.2"))    +
stat_function(fun = dexp, args = list(rate = 0.5), aes(colour = "0.5"))    +
stat_function(fun = dexp, args = list(rate = 1), aes(colour = "1"))        +
stat_function(fun = dexp, args = list(rate = 2), aes(colour = "2"))        +
stat_function(fun = dexp, args = list(rate = 3), aes(colour = "3"))        +
stat_function(fun = dexp, args = list(rate = 3.5),aes(colour = "3.5"))     +
stat_function(fun = dexp, args = list(rate = 3.75), aes(colour = "3.75")) +
stat_function(fun = dexp, args = list(rate = 4), aes(colour = "4"))        +
scale_color_manual("Rate", values =brewer.pal(9,"Blues"))  +
labs(x = "\n x", y = "f(x) \n",
      title = "Exponential Distribution Density Plots \n") +
theme(plot.title = element_text(hjust = 0.5),
      legend.title = element_text(face="bold", size = 10),
      legend.position = "right") +
theme_bw()
```



After the brief introduction to the *exponential* distribution, we are ready to start the simulation in R. We assign the parameter λ to the value 0.2, the parameter "nosim" to the vector with increasing integers from 1 to 1000 (index of each sample) and finally we will create a data frame with the index of each sample in the first column and the mean of the respective sample in the second column. In the following script the command **rexp** generates in our case forty thousand observations from an *exponentially* distributed population with rate $\lambda = 0.2$. With the command **matrix** the program rearranges the random values into a 1000×40 matrix and finally with the command **data.frame** we gather the samples' indexes and means of each sample into a single frame of values.

```
#insert lambda and number of observations
lambda <- 0.2
n <- 40
```

```
# indexes of samples, number of simulations
nosim <- 1:1000
# generate 1000 samples of 40 observations
samples <- matrix(rexp(n*1000,lambda),1000,40)
# create the data frame to gather samples' means
means <- data.frame(x = nosim,y = apply(samples,1,mean))
```

If you want to reproduce the process on your own you can type the commands presented above in a new terminal of a new session in R, but of course you should expect different values since the pseudo-random number generator of R creates different sets each time you insert the commands in a new session. We expect that our results from the experimentation via the simulations should be *consistent* with the population's distribution parameters, meaning that each sample's mean is an *unbiased* estimator of the mean of the population and that the same holds for the S^2 of each sample which estimates the population's variance σ^2.

[1] We know that the mean of the sample's mean is an *unbiased estimator* of the population's mean, so since we have stored the mean values of our one thousand samples in the second column of the means variable, by typing

```
mean(means[,2])
```

```
[1] 5.000546
```

we get the mean of the samples' means and conclude that the population's mean *unbiased estimator* is 5.000546, which is very close to the distribution's theoretical mean $\frac{1}{\lambda} = \frac{1}{0.2} = 5$.

[2] We know that the *unbiased* variance estimation from a sample is derived by the formula

$$S^2 = \frac{\sum_{i=1}^{n} \left(\mathbb{X}_i - \overline{\mathbb{X}}\right)^2}{n-1}$$

where $\overline{\mathbb{X}}$ is the sample's mean $\mathbb{X}_i$ the sample's observations and n the number of the observations in each sample, we will use this formula in R to create a vector with 1000 entries with the *unbiased* estimations of variance from each sample and then divide their sum by the number of samples to get the *unbiased estimation* variance estimate of the population's variance. Before we continue, we need to understand why we use the $n-1$ degrees of freedom in the denominator of the *estimator*. Given that we computed the sample mean, if we wanted to compute S^2 which is the second information we would like to obtain from the sample, we are left with $n-1$ degrees of freedom and this happens because if we were to randomly choose $n-1$ of the sample observations out of n, the last one is always biased since we can find its value by subtracting the sum of the $n-1$ observations from the sample mean we computed in the first step.

```
v <- sum(apply(samples,1,function(x){sum(x-mean(means[,2]))^2/(n-1)}))/1000
sqrt(v)
```

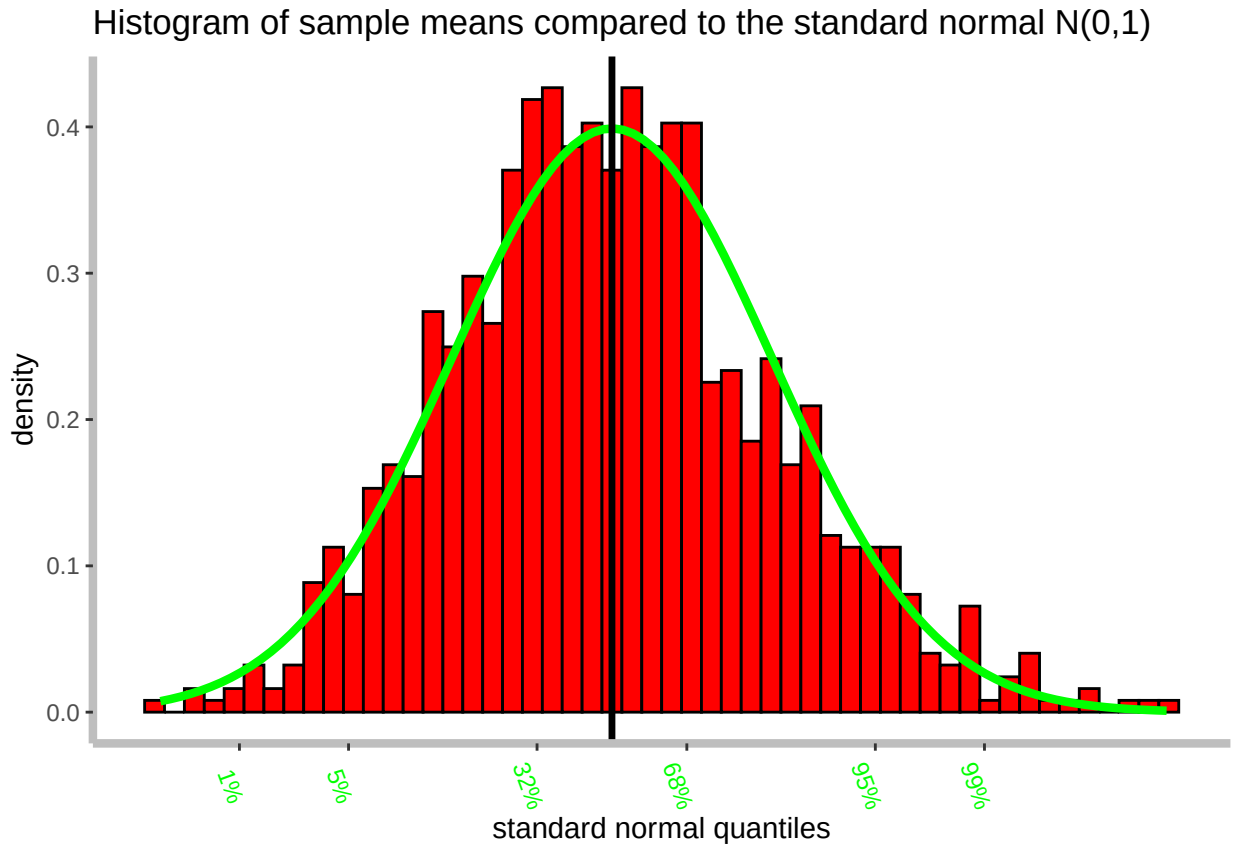
```
[1] 5.066169
```

So we came up with a variance *estimator* of 25.6660682 and a standard deviation of 5.066169, which as expected are very close to the theoretical values they are estimating $\sigma^2 = \frac{1}{\lambda^2} = 25$ and $\sigma = \frac{1}{\lambda} = \frac{1}{0.2} = 5$ respectively, with a small difference between the experimental and the theoretical value.

[3] Briefly the *central limit theorem* (CLT) sates that if we sample from a population in order to estimate a parameter of interest with non infinite variance, even if we don't know anything about the population's distribution and if that population isn't normally distributed, for a large number of observations for each sample and a large number of sampling repetitions, the distribution of the standardized samples' means follows a standard normal distribution. To see that in action the following code creates a plot with the histogram of the one thousand means we sampled in the beginning and the normal distribution centered

at 0 with a standard deviation equal to $\frac{1}{\lambda} = 5$ since the samples are drawn from an exponential distribution with $\lambda = 0.2$.

```
g <- ggplot(data = data.frame(x = (means[,2] - 5)/(5/sqrt(40))),aes(x = x))
g <- g + geom_histogram(aes(y = after_stat(density)),colour = "black",fill = "red",
                        binwidth = diff(range(means[,2]))/40)
g <- g + stat_function(fun = dnorm,args = list(mean = 0,sd = 1),color = "green",linewidth = 1.5)
g <- g + geom_vline(xintercept = 0,linewidth = 1.2) +
  theme(axis.line = element_line(colour = "grey",
                      linewidth = 1.5, linetype = "solid"),panel.grid = element_blank(),
        panel.background = element_rect(fill = "white"),
        axis.text.x = element_text(color = "green",angle = -70)) +
  labs(title = paste("Histogram of sample means compared to the standard normal N(0,1)"))
g <- g + xlab("standard normal quantiles") + scale_x_continuous(
  breaks=c(qnorm(.01),qnorm(.05),
           qnorm(.32),qnorm(.68),
           qnorm(.95),qnorm(.99)),
        label=c("1%","5%","32%","68%","95%","99%"))
g
```



If we used a bigger sample size i.e. 100 and gathered more samples i.e. 2000, what we would expect to see if we created the sample plot is a histogram bounded by the green line perfectly and this complies with the CLT

$$\sqrt{n}\frac{\overline{\mathbb{X}_i} - \mu}{\sigma} \sim \mathcal{N}(0,1)$$

where n is the sample size, $\overline{\mathbb{X}_i}$ is each sample's mean with index i,

$$\mu = \frac{1}{\text{number of samples}} \times \sum_{i=1}^{\text{number of samples}} \frac{\overline{\mathbb{X}_i}}{n}$$

and

$$\sigma = \sqrt{\frac{1}{\text{number of samples}} \times \sum_{i=1}^{\text{number of samples}} \frac{\left(\overline{\mathbb{X}_i} - \mu\right)^2}{n-1}}$$

and this concludes the report.