Statistical Inference Course Project Part 1: Simulation Exercise

Edgar Bahilo Rodríguez

December 12 of 2018

1 Introduction

This R Markdown notebook develops part 1 of the statistical inference project for the Statistical Inferece part of the Data Science Specialization by Coursera and John Hopkins University. As it is described in the instructions of the assignment the purpose of this document is:

1. Investigate the exponential distribution given by the formula $\lambda e^{-\lambda x}$ in R, compare it with the Central Limit Theorem.
2. Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials.
   - Show the sample mean and compare it to the theoretical mean of the distribution.
   - Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
   - Show that the distribution is approximately normal.

2 Simulations settings

The code displayed in the chunk below creates a vector with 4 different simulation sizes in order to study the effect of the CLT in a better way. Also the code defines the parameter of the exponential distribution λ and the sample size n.

```
set.seed(1234)
lambda<-0.2
n<-40
runs<-c(10,100,1000,10000)
```

2.1 Simulations

The code below uses replicate (apply family) to create a matrix of size $n_{samples} * runs[i]$ under the exponential distribution described before. Then, the mean is calculated for each simulation of $n_{samples}$ for the different runs sizes of the simulations $[10, 100, 1000, 1000]$

```
simulations_10<-replicate(runs[1], rexp(n,lambda))
means_simul_10<-apply(simulations_10, 2, mean) #analogue can be done using for loops
simulations_100<-replicate(runs[2], rexp(n,lambda))
means_simul_100<-apply(simulations_100, 2, mean)
simulations_1000 <- replicate(runs[3], rexp(n, lambda))
means_simul_1000<-apply(simulations_1000, 2, mean)
simulations_10000 <- replicate(runs[4], rexp(n, lambda))
means_simul_10000<-apply(simulations_10000, 2, mean)
```

2.2 Comparison: mean, variance, standard deviation and distribution

Now different statistics (mean of all the runs, variance of this mean and standard deviation of this mean) are calculated for each simulation size.

```r
mean_10<-mean(means_simul_10)
mean_100<-mean(means_simul_100)
mean_1000<-mean(means_simul_1000)
mean_10000<-mean(means_simul_10000)
var_10<-var(means_simul_10)
var_100<-var(means_simul_100)
var_1000<-var(means_simul_1000)
var_10000<-var(means_simul_10000)
sd_10<-sd(means_simul_10)
sd_100<-sd(means_simul_100)
sd_1000<-sd(means_simul_1000)
sd_10000<-sd(means_simul_10000)
mean_theoretical<-1/lambda
var_theoretical<-(1/lambda)^2/(n)
sd_theoretical<-1/(lambda * sqrt(n))
```

Then we compare the theoretical statistics (mean, variance and standard deviation) with the real ones:

```r
print(means_all) #code of each cbind is hiden to increase readability
```

```
##    mean_10 mean_100 mean_1000 mean_10000 mean_theoretical
## 1 4.942464 5.052657  4.959047   5.007675                5
```

```r
print(var_all) #code of each cbind is hiden to increase readability
```

```
##     var_10   var_100  var_1000 var_10000 var_theoretical
## 1 1.102861 0.5103012 0.5665642 0.6167534           0.625
```

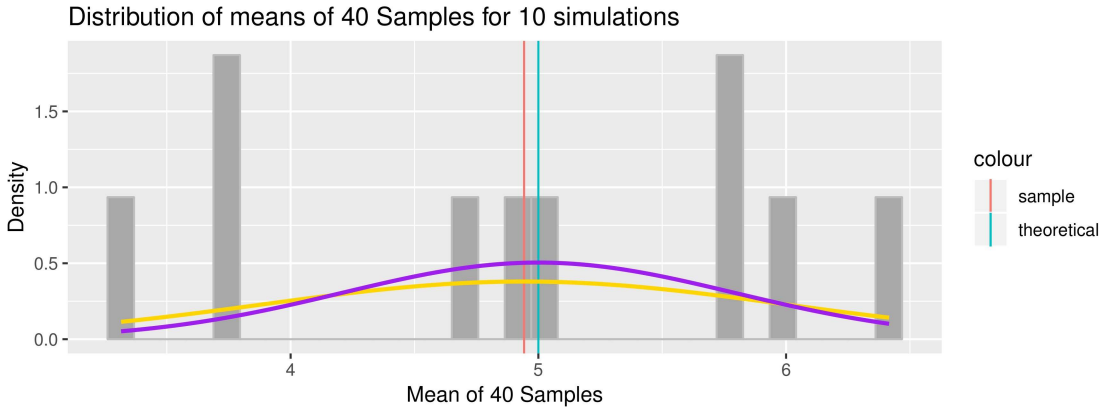
```r
print(sd_all) #code of each cbind is hiden to increase readability
```

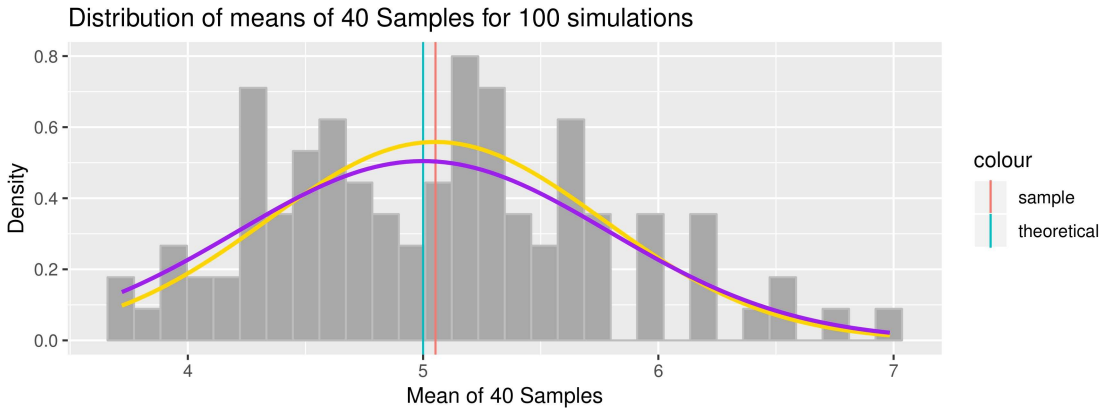
```
##      sd_10    sd_100   sd_1000  sd_10000 sd_theoretical
## 1 1.050172 0.7143537 0.7527046 0.7853365      0.7905694
```

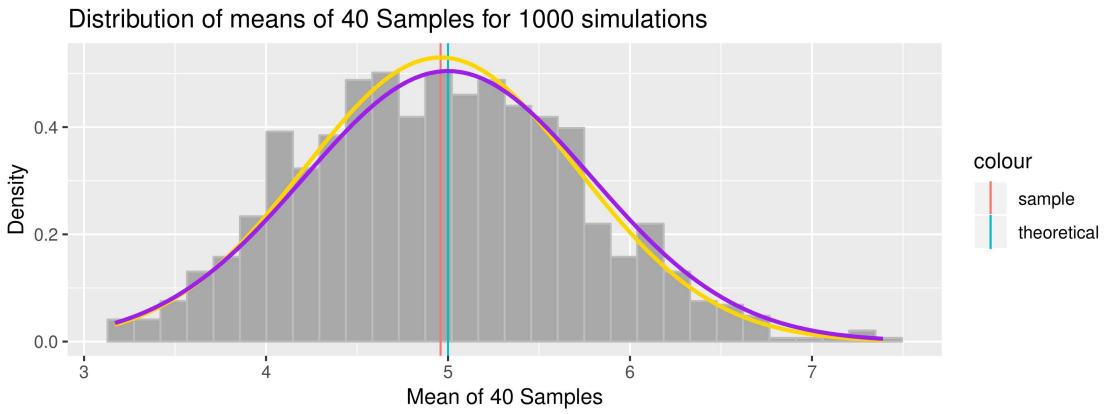
It can be seen how despite the similarities of the means with different simulation sizes, the variance and consequently the standard deviation look quite worse when the simulation runs are smaller.

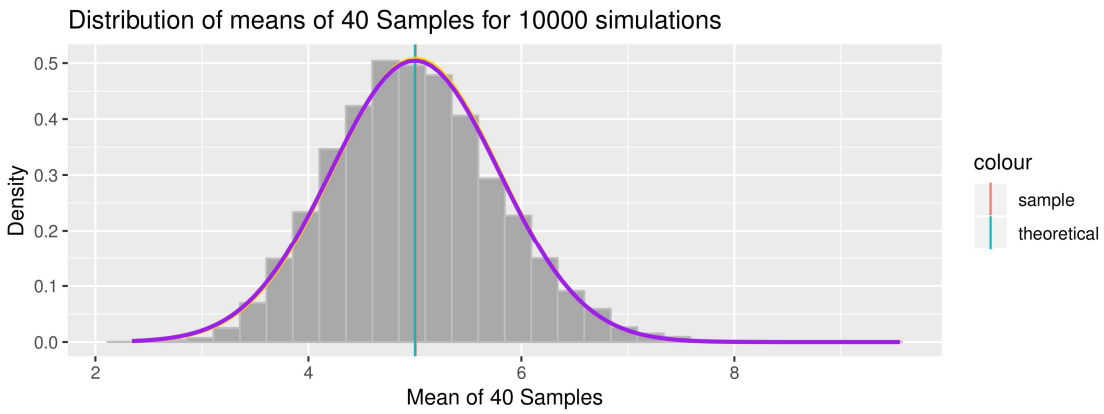
As a matter of clarification, the distribution of different simulations sizes is showed in the plot grid displayed below. It is shown how as the run size increases the distribution looks more normal. At the end what we are doing when we increase the number of simulations is creating more samples, therefore allowing the data to converge according to the CLT.

```r
grid.arrange(p1,p2,p3,p4, ncol=1)
```


Distribution of means of 40 Samples for 10 simulations
Density
1.5
1.0
0.5
0.0
4
5
6
Mean of 40 Samples
colour
sample
theoretical


Distribution of means of 40 Samples for 100 simulations
Density
0.8
0.6
0.4
0.2
0.0
4
5
6
7
Mean of 40 Samples
colour
sample
theoretical


Distribution of means of 40 Samples for 1000 simulations
Density
0.4
0.2
0.0
3
4
5
6
7
Mean of 40 Samples
colour
sample
theoretical


Distribution of means of 40 Samples for 10000 simulations
Density
0.5
0.4
0.3
0.2
0.1
0.0
2
4
6
8
Mean of 40 Samples
colour
sample
theoretical

```
#code of each plot hidden to improve readability
#purple line distribution is a normal distribution
#yellow line is the distribution of the simulated data
```

2.3 Normality tests and QQplot

Finally, we want to ensure that the data is normal (despite of having this hindshight using the distribution plots). We use Shapiro-Wilk test to check the normality of the data obtained after 1000 simulations.

```
shapiro.test(means_simul_1000)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  means_simul_1000
## W = 0.9953, p-value = 0.003568
```

The test shows that we can not reject normality and the data looks normal with a significant p-value. We also plot the Q-Q plot for the data obtained after 10000 as Shapiro-Wilk can only deal with sample sizes lower than 5000.

```
qqnorm(means_simul_10000)
qqline(means_simul_10000, col = "red")
```