Statistical Inference: Sample Project Report Structure

José Manuel Peñas

Contents

1 Overview **1**

2 Simulations **1**

2.1 Show the sample mean and compare it to the theoretical mean of the distribution. 1

2.2 Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution. 2

2.3 Show that the distribution is approximately normal. 2

3 APENDIX **3**

1 Overview

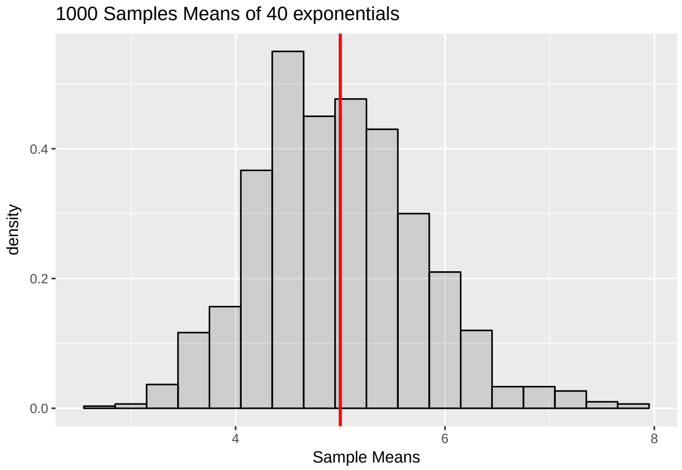
The present document will investigate the exponential distribution in R and compare it with the Central Limit Theorem. It will generate a sample distribution of 40 exponentials and will compare the sample means of 1 thousand simulations and the variance with the theoretical values. Finally, it will show the difference between the distribution of a large number of random exponentials(1000) and the distribution of a large group of averages of 40 exponentials(1000 as well).

2 Simulations

The simulation will work with exponential distributions of 40 elements with λ of 0.2. Therefore the mean and the standard deviation of the distrubtions will be $1/\lambda$: **5** With this initial data, it generates 1000 simulations of 40 random exponentials, calculate its means and stored in a vector that will be the base of the study.

2.1 Show the sample mean and compare it to the theoretical mean of the distribution.

As it was stated earlier the theoretical mean of this distrbution is **5**. The expected result with these number of samples.



As you can see at the plot the means of the simulated samples have a very similar distribution of normal distribution with a mean very close to the theoretical figure of **5**.

```
mean(sampleMeans)
```

```
## [1] 4.983284
```

2.2 Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.

The theoretical variance of an exponential distribution is $1/\lambda^2/n$ with n being 40. In this case:

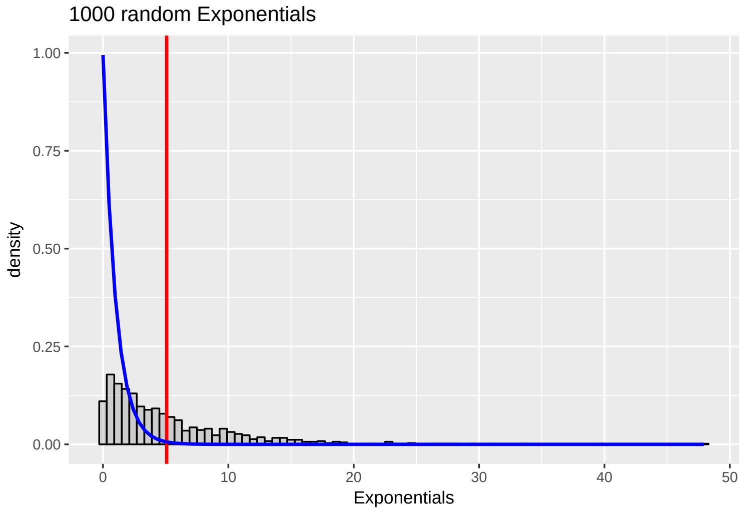
The variance of the samples is:

```
## [1] 0.611492
```

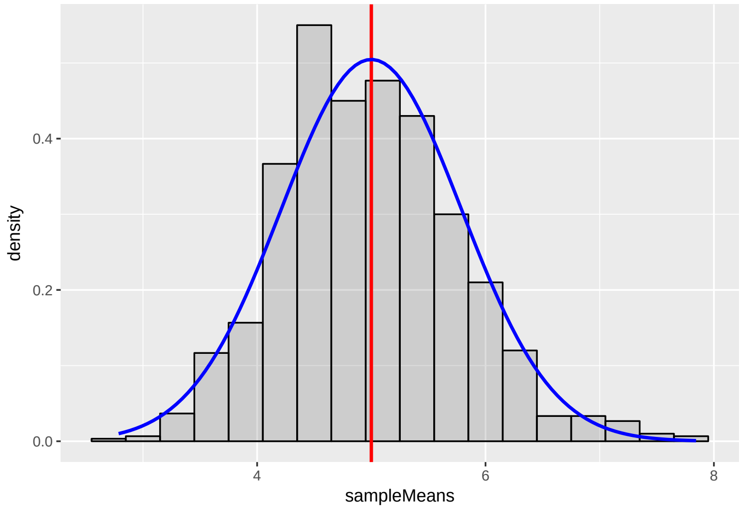
Both figures are very close. If we would increase the number of simulations the results will be closer as CLT states.

2.3 Show that the distribution is approximately normal.

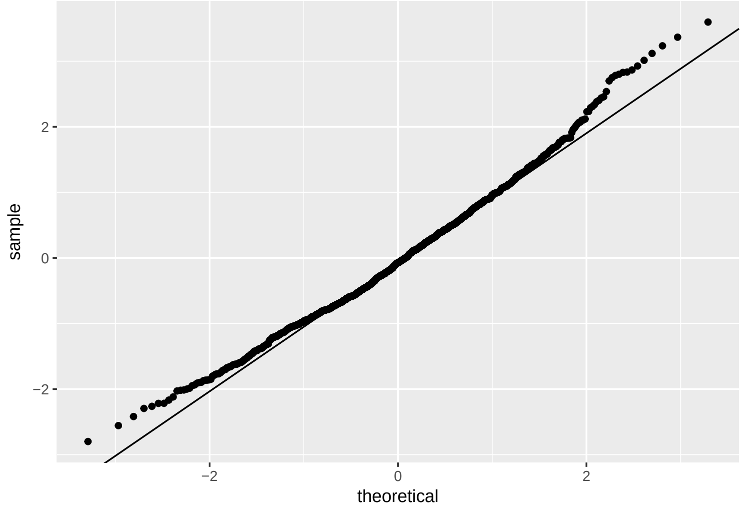
First we will show the histogram of the distribution 1000 exponentials of a distribution of the indicated λ



It shows the expected exponential distribution that is very different from normal.
But if we add the histogram of the simulated sample the distribution curve of a normal distribution of mean $1/\lambda$ and standard deviation of $1/\lambda/sqrt(n)$, being n 40, the Bell curve is very close to the histogram.



Finally if we create a qqplot that compares the actual quantiles of the normalized values with the theoretical ones, the points are really close to the line, so it's not difficult to infer that following to the Central Limit Theorem that the distriubtion after 1000 samples is getting close to a normal one.



3 APENDIX

For reproducible research purposes, here it shows the R code used for this report

```
sampleMeans <- NULL
exp <- 40
lambda <- .2
theoretical <- 1 / lambda
simul <- 1000
set.seed(4169)
for (i in 1:simul)
```

```r
sampleMeans = c(sampleMeans, mean(rexp(exp, lambda)))
#
g <-
ggplot(data.frame(sampleMeans), aes(x = sampleMeans)) + ggtitle("1000 Samples Means of 40 exponentials")
g <-
g + geom_histogram(
alpha = .20,
binwidth = .3,
colour = "black",
aes(y = ..density..)
) + xlab("Sample Means")
g <- g + geom_vline(xintercept = theoretical,
colour = "red",
size = 1)
#g
#
mean(sampleMeans)
#
theoVariance <- 1 / lambda ^ 2 / exp
#
var(sampleMeans)
#
samp <- rexp(simul, lambda)
g <-
ggplot(data.frame(samp), aes(x = samp), main =) + ggtitle("1000 random Exponentials")
g <-
g + geom_histogram(
alpha = .20,
binwidth = .6,
colour = "black",
aes(y = ..density..)
) + xlab("Exponentials")
g <- g + geom_vline(xintercept = mean(samp),
colour = "red",
size = 1)
g <- g + stat_function(fun = dexp, size = 1, colour = "blue")
#
g <- ggplot(data.frame(sampleMeans), aes(x = sampleMeans))
g <-
g + geom_histogram(
alpha = .20,
binwidth = .3,
colour = "black",
aes(y = ..density..)
)
g <- g + geom_vline(xintercept = theoretical,
colour = "red",
size = 1)
g <-
g + stat_function(
fun = dnorm,
args = list(mean = theoretical, sd = theoretical / sqrt(exp)),
size = 1,
colour = "blue"
)
#
normSMean <- (sampleMeans - theoretical) / (theoretical / sqrt(exp))
y1 <- quantile(normSMean[!is.na(normSMean)], c(0.25, 0.75))
x1 <- qnorm(c(0.25, 0.75))
slope <- diff(y1) / diff(x1)
int <- y1[1L] - slope * x1[1L]
g <-
ggplot(data.frame(normSMean), aes(sample = normSMean)) + stat_qq()
g <- g + geom_abline(slope = slope, intercept = int)
```