JHU Statistical Inference - course project

Harm Lammers

22 september 2016

Contents

Overview

This report describes the outcome of my search in R and the course material to submit a solution to the course project as described above.

Part 1: Simulation Exercise Instructions

In this project you will investigate the exponential distribution in R and compare it with the Central Limit Theorem. The exponential distribution can be simulated in R with rexp(n, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. Set lambda = 0.2 for all of the simulations.

You will investigate the distribution of averages of 40 exponentials. Note that you will need to do a thousand simulations.

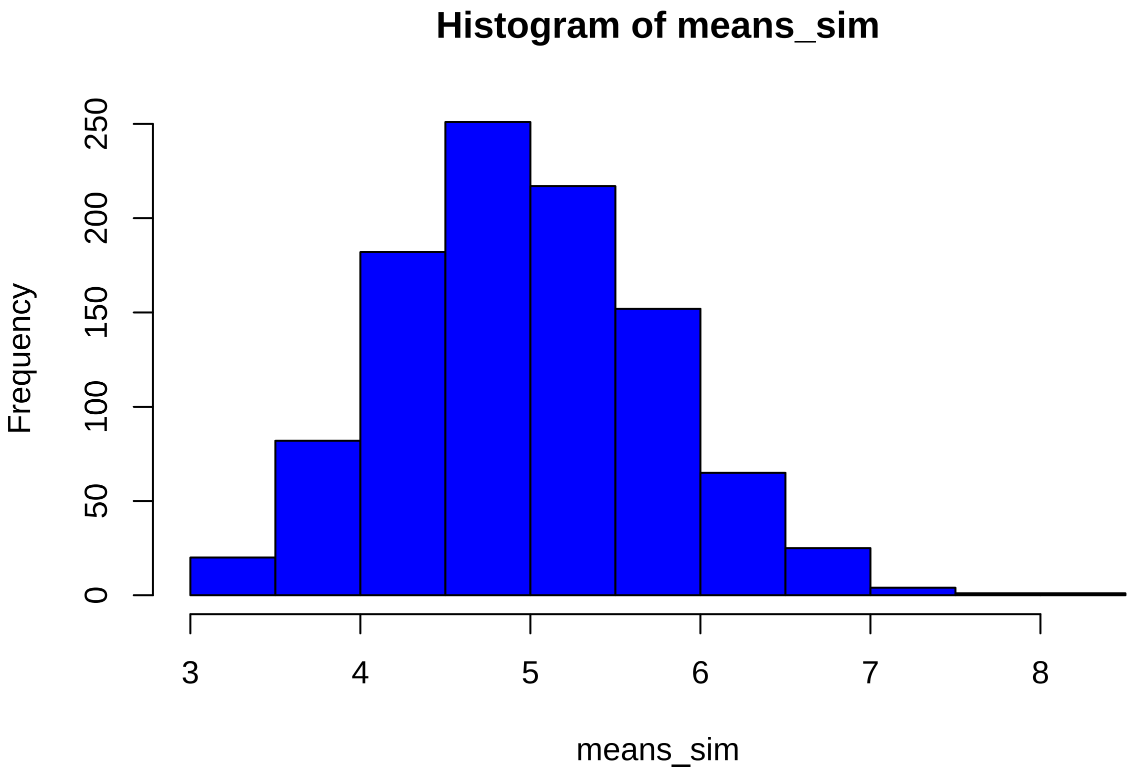
Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials. You should - Show the sample mean and compare it to the theoretical mean of the distribution.
- Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
- Show that the distribution is approximately normal.

In point 3, focus on the difference between the distribution of a large collection of random exponentials and the distribution of a large collection of averages of 40 exponentials.

Simulations and the R-code

We need to do 1000 simulations in which we derive a sample mean out of 40 simulated values for an exponential distribution with rate 0.2 (lambda). Then we plot the means in a Histogram "means_sim".

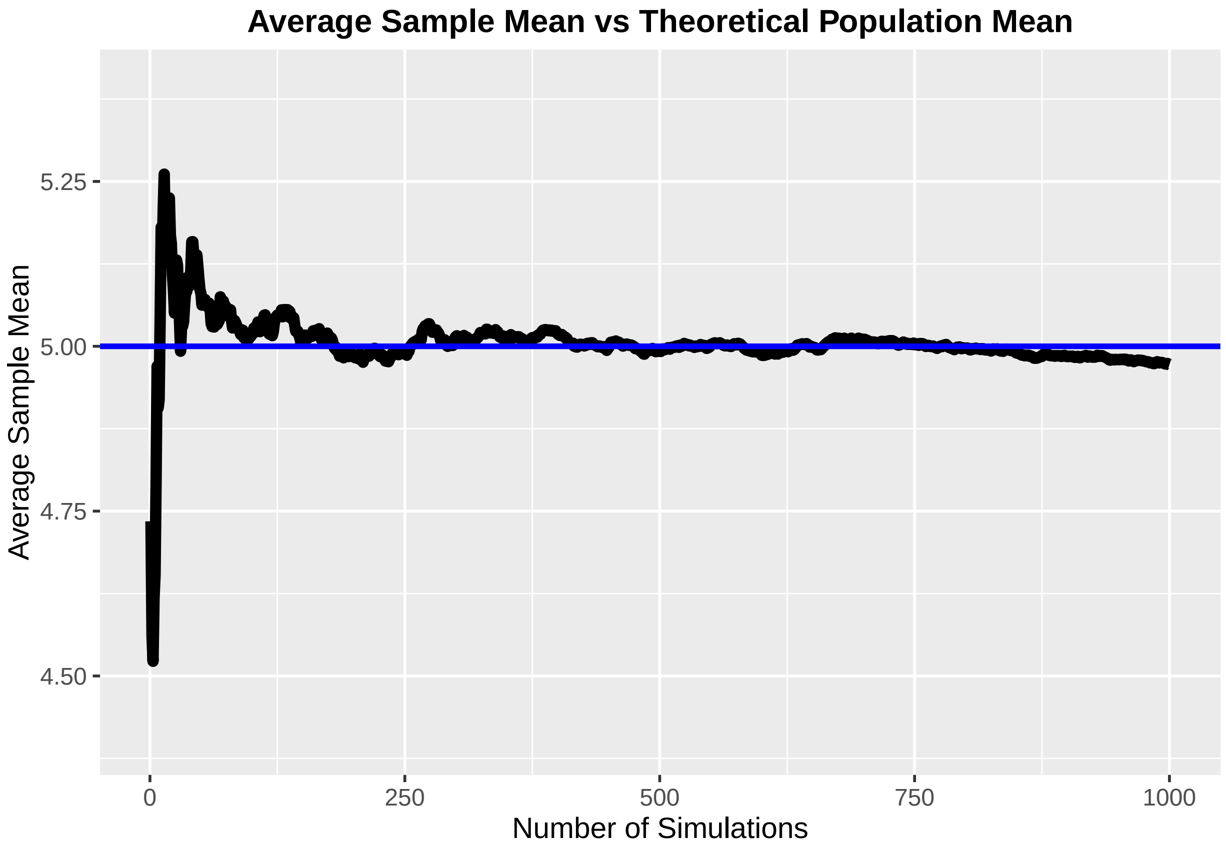
```r
#Definition of variables
n_sim    <- 1000
n        <- 40
lambda   <- 0.2
set.seed(12345)

#Derived distribution parameters
mu       <- 1/lambda
sd       <- 1/lambda

#Create a matrix with n_sim rows and n columns corresponding to random simulation n times
matrix_sim  <- matrix(rexp(n_sim * n, rate=lambda), n_sim, n)

#Create a vector with rowmeans
means_sim   <- rowMeans(matrix_sim)

#Plot the means in a histogram
library(ggplot2)
hist(means_sim, col="blue")
```



Sample Mean versus Theoretical Mean

In order to show what happens we can plot the average sample mean for i iterations, where i = 1 … 1000. The average sample mean should converge to the distribution mean according to the Central Limit Theorem.

```
#Define variables
means_sum    <- vector("numeric")
means_avg    <- vector("numeric")

#For any iteration i compute the average mean of the i sample means
means_sum[1] <- means_sim[1]
for (i in 2:n_sim) { means_sum[i] <- means_sum[i-1] + means_sim[i] }
for (i in 1:n_sim) { means_avg[i] <- means_sum[i]/i }

#Plot the computed average_iteration_mean against the theoretical mean
library(ggplot2)

g <- ggplot(data.frame(x = 1:n_sim, y = means_avg), aes(x = x, y = y))
g <- g + geom_hline(yintercept = 0) + geom_line(size = 2)
g <- g + geom_abline(intercept = 1 / lambda, slope = 0, color = "blue", size = 1)
g <- g + scale_y_continuous(breaks=c(4.50, 4.75, 5.00, 5.25), limits=c(4.4, 5.4))
g <- g + theme(plot.title = element_text(size=12, face="bold", vjust=2, hjust=0.5))
g <- g + labs(title="Average Sample Mean vs Theoretical Population Mean")
g <- g + labs(x = "Number of Simulations", y = "Average Sample Mean")
print(g)
```



The sample means converges with increasing iterations 4.971972.
Which gets close to the theoretical mean of the exponential distribution (1/lambda with lambda = 0.2): 5.

Sample Variance versus Theoretical Variance

According to the Central Limit Theorem the variance of the mean is sigma / squareroot(n).
So the variance of the distribution can be estimated by n * variance(means).

The theoretical variance of the exponential distribution is (1/lambda)^2: 25.
The estimated variance of the distribution is n * variance(means) : 24.6317049.

The reported values are quite close; suggesting support for the CLT :-).

Distribution

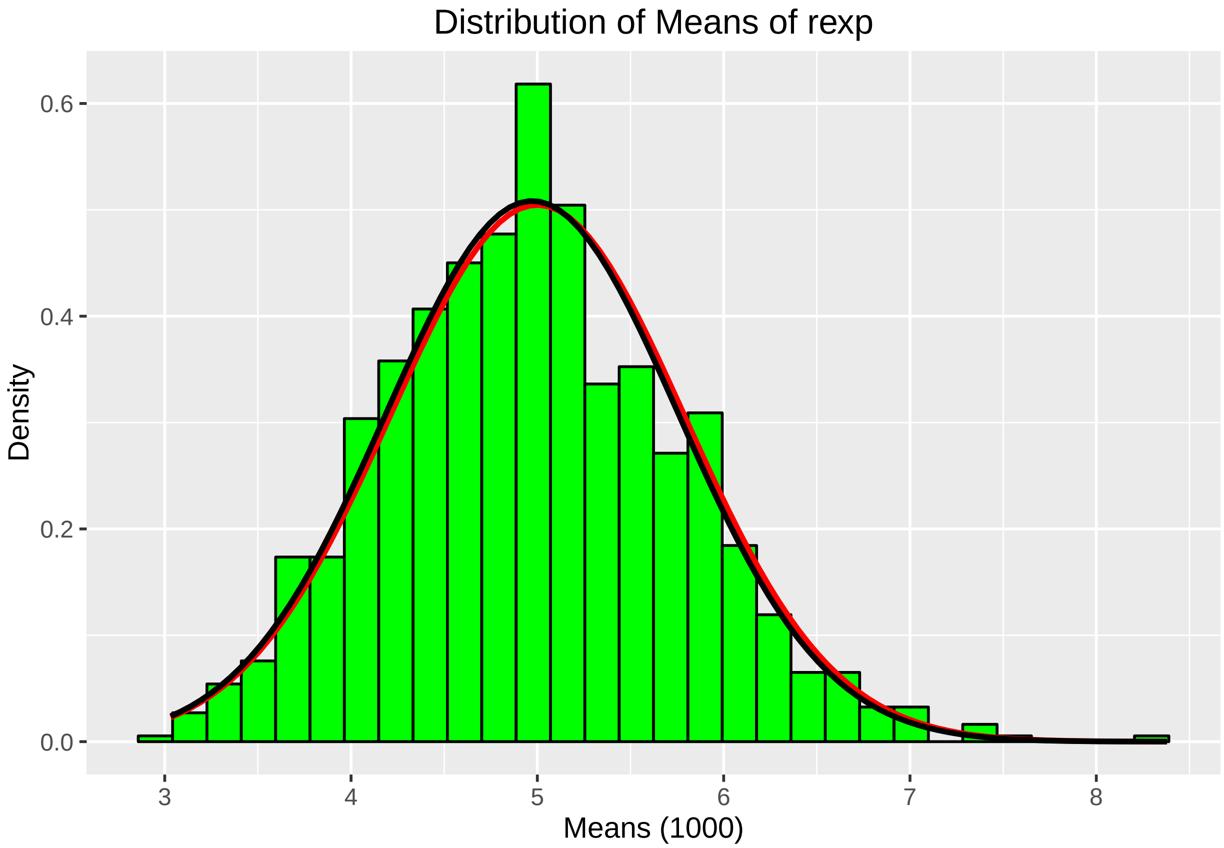
The Law of Large Numbers states that averages of iid samples converge to population means that they are estimating.
The Central Limit Theorem states that averages are aproximately normal, with distributions

- centered at the population mean
- with standard deviation equal to the standard error of the mean
- CLT gives no guarantee that n is large enough.

We can illustrate this with plots of the standard normal distribution (red) overlayed by the distribution of the means with increasing number of simulations. This report only contains the plots for the original value (n_sim0 = 1000) and (n_sim3 = 30.000).

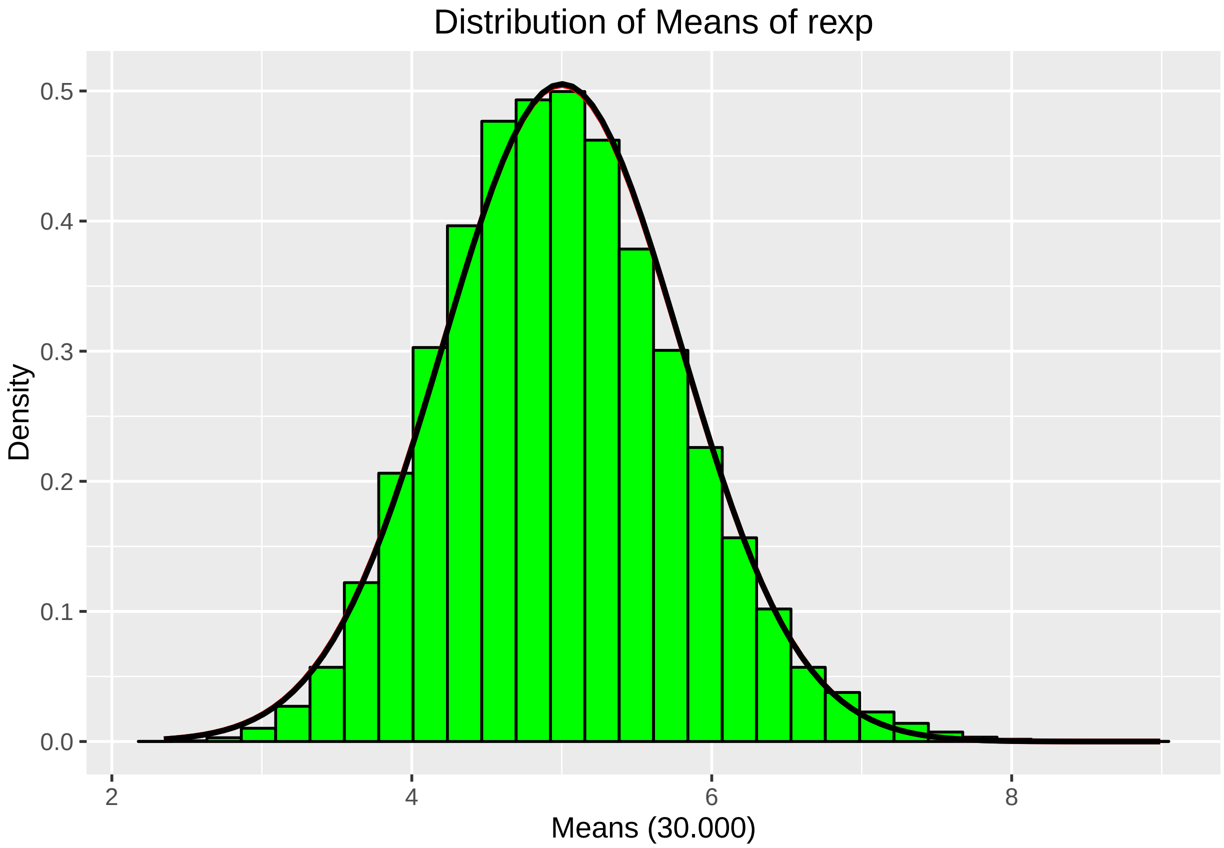
```
#Definition of variables
n_sim0   <- 1000
n_sim3   <- 30000

n        <- 40
lambda   <- 0.2

set.seed(12345)

#Derived distribution parameters
mu       <- 1/lambda
sd       <- 1/lambda

#Create a matrix with n_sim rows and nx columns corresponding to random simulation nx times
matrix_sim0 <- matrix(rexp(n_sim0 * n, rate=lambda), n_sim0, n)
matrix_sim3 <- matrix(rexp(n_sim3 * n, rate=lambda), n_sim3, n)

#Create a vector with rowmeans
means_sim0  <- rowMeans(matrix_sim0)
means_sim3  <- rowMeans(matrix_sim3)

library(ggplot2)
X = means_sim0

plotdata <- data.frame(X)
plot1 <- ggplot(plotdata,aes(x = X))
plot1 <- plot1 +geom_histogram(aes(y=..density..), colour="black",fill="green")
plot1 <- plot1+labs(title="Distribution of Means of rexp", x="Means (1000)", y="Density")
plot1 <- plot1 +stat_function(fun=dnorm,args=list( mean=1/lambda, sd=sqrt((1/lambda)^2/n)), color="red"
plot1 <- plot1 +stat_function(fun=dnorm,args=list( mean=mean(X), sd=sqrt(var(X))),color="black", size=1
print(plot1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#Similar set of statements for X=means_sim3.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As you can see, when the number of simulations increases, the fit seems to be better and the estimated distribution follows the normal distribution more closely.