Exponential Distribution in R Compared to Central Limit Theorem

By Aiman D.

Overview

Part 1 of the project 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, λ) where λ is the rate parameter. The mean of exponential distribution is 1/λ and the standard deviation is 1/λ. We will set λ = 0.2 for all of the simulations and investigate the distribution of averages of 40 exponentials.

Simulations

To compare, we will execute 1000 simulations that has 40 exponentials with λ = 0.2 and and apply the function rexp(n, λ).

The following code was executed:

```r
NumSims = 1000;
n = 40;
Lambda = 0.2

Means <- vector("numeric")
Sum_of_Means <- vector("numeric")
Cum_of_Means <- vector("numeric")

for (i in 1:NumSims)
{
  Means[i] <- mean(rexp(n, Lambda))
}
Sum_of_Means <- Means[1]

for (i in 2:NumSims)
{
  Sum_of_Means[i] <- Sum_of_Means[i-1] + Means[i]
}

for (i in 1:NumSims)
{
  Cum_of_Means[i] <- Sum_of_Means[i]/i
}

print(sprintf("The Means of the sample equal: %f", Cum_of_Means[NumSims]))
print(sprintf("The theoretical mean is equal to: %f", 1/Lambda))
```

This was the output:

```
[1] "The Means of the sample equal: 5.003542"
```

```
[1] "The theoretical mean is equal to: 5.000000"
```

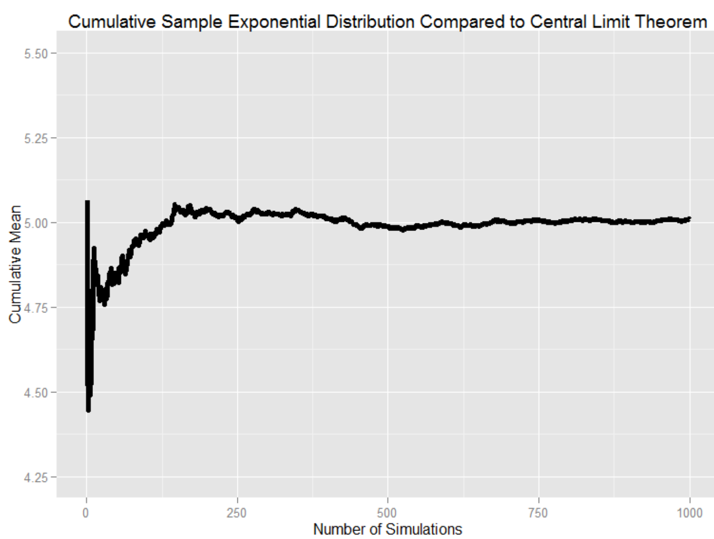
The end of the simulations was very close to the theoretical mean as the plot will also show after running the following code:

```r
library(ggplot2)
NumSims = 1000;
n = 40;
Lambda = 0.2

Means <- vector("numeric")
Sum_of_Means <- vector("numeric")
Cum_of_Means <- vector("numeric")

g<-ggplot(data.frame(x = 1:NumSims, y = Cum_of_Means), aes(x = x, y = y))
g<-g+geom_hline(yintercept = 0) + geom_line(size = 1)
g<-g+scale_y_continuous(breaks=c(4.25, 4.50, 4.75, 5.00, 5.25, 5.50), limits =c(4.25, 5.5))
g<-g+theme(plot.title=element_text(size=12, face="bold", vjust=2, hjust=0. 5))
g<-g+labs(title="Cumulative Sample Exponential Distribution Compared to Central Limit Theorem")
g<-g+labs(x="Number of Simulations", y="Cumulative Mean")
print(g)
```

The code produced the following plot:



Sample variance compared to the theoretical variance of the distribution

```r
NumSims = 1000;
n = 40;
Lambda = 0.2

Means <- vector("numeric")

for (i in 1:NumSims)
{
  Means[i] <- mean(rexp(n, Lambda))
}

print(sprintf("The Sample Variance Means: %f", var(Means)*n))
print(sprintf("The theoretical variance is: %f", (1/Lambda)^ 2))
```

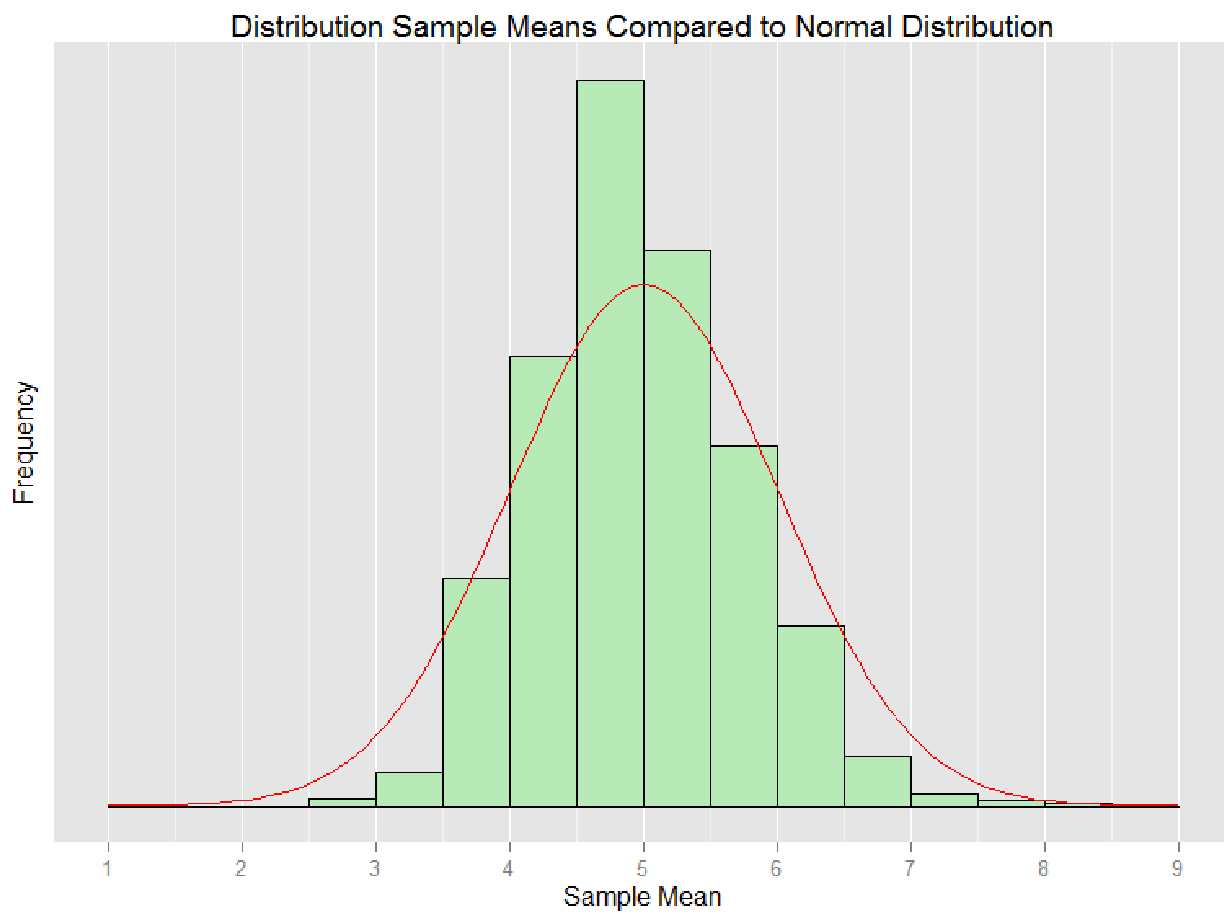
The output was also very close:

```
> print(sprintf("The Sample Variance Means: %f", var(Means)*n))
[1] "The Sample Variance Means: 23.834909"
> print(sprintf("The theoretical variance is: %f", (1/Lambda)^ 2))
[1] "The theoretical variance is: 25.000000"
```

Sample Distribution Means Compared to Normal Distribution

```
library(ggplot2)
Means <- vector("numeric")
for (i in 1:NumSims)
{
  Means[i] <- mean(rexp(n, Lambda))
}
g<-ggplot(data.frame(x = Means), aes(x = x ))
g<-g+geom_histogram(position="identity", fill="green", color="black", alpha= 0.2, binwidth=0.5, aes(y=..density..))
g<-g+stat_function(fun=dnorm, colour="red", args=list(mean=5))
g<-g+scale_x_continuous(breaks=c(1, 2, 3, 4, 5, 6, 7, 8, 9), limits=c(1, 9))
g<-g+scale_y_continuous(breaks=c())
g<-g+theme(plot.title=element_text(size=14, face="bold", vjust=2, hjust=0. 5))
g<-g+labs(title="Distribution Sample Means Compared to Normal Distribution")
g<-g+labs(x="Sample Mean", y="Frequency")
print(g)
```

The code produced the following plot:



The sample shows normal distribution around the mean of 5.