Course Project for Statistical Inference by Hsin-Hua Lai

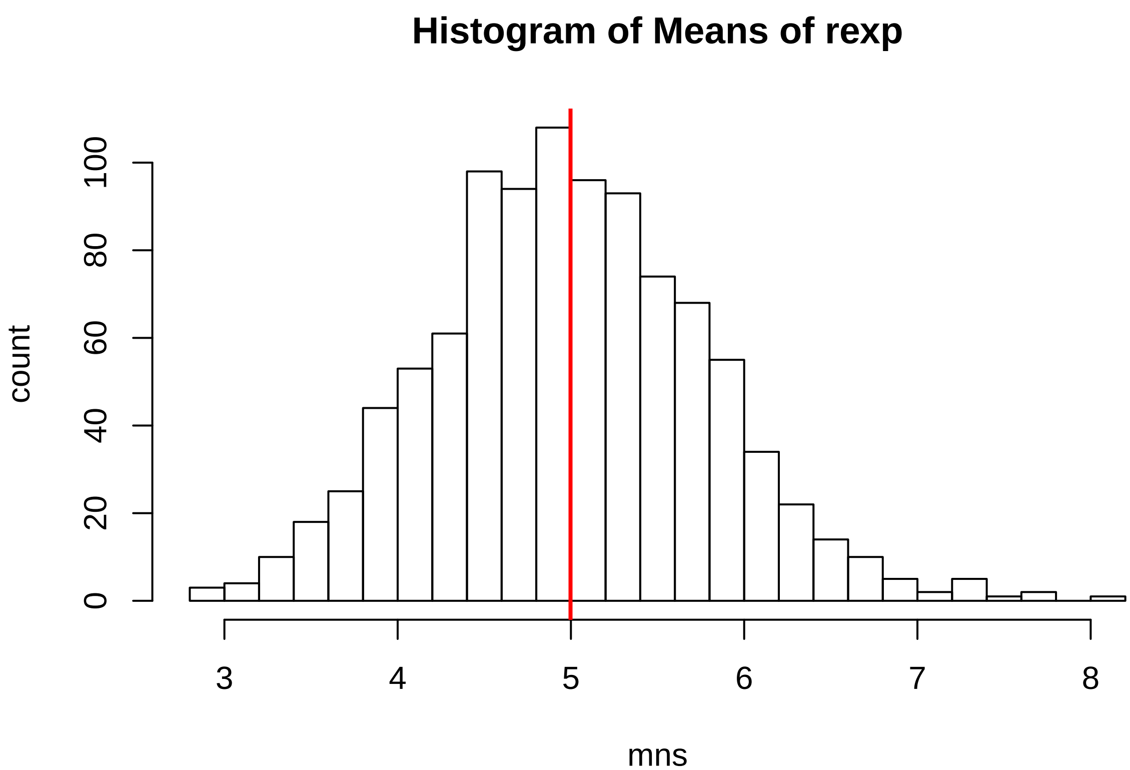
Overview

In this Course Project we will investigate the distribution of average 40 exponentials generated by a thousnad simulations and compare it with the Central Limit Theorem.

Simulation Logic

We follow the logic of the example code shown on the assignment website to first create a NULL vector called mns. We then generate 40 random exponentials with lambda = 0.2 and calculate its mean and append it to the NULL vector mns. We keep doing this for 10000 times and plot the histogram of the resulting data to check the Central Limit Theorem.

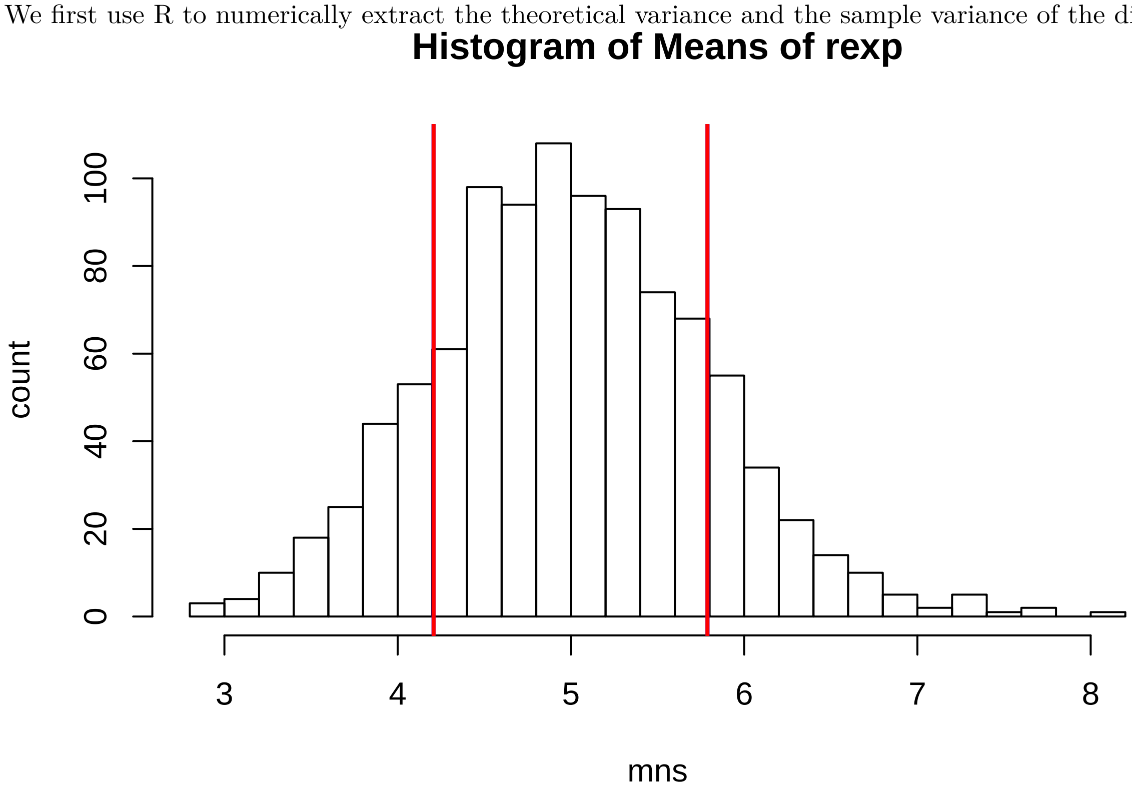
Simulation and Results



Therefore we can see that the Sample Mean is 4.9976676 pointed by the red vertical line, which is roughly equal to the Theoretical Mean 0.5.

Sample Variance versus Theoretical Variance

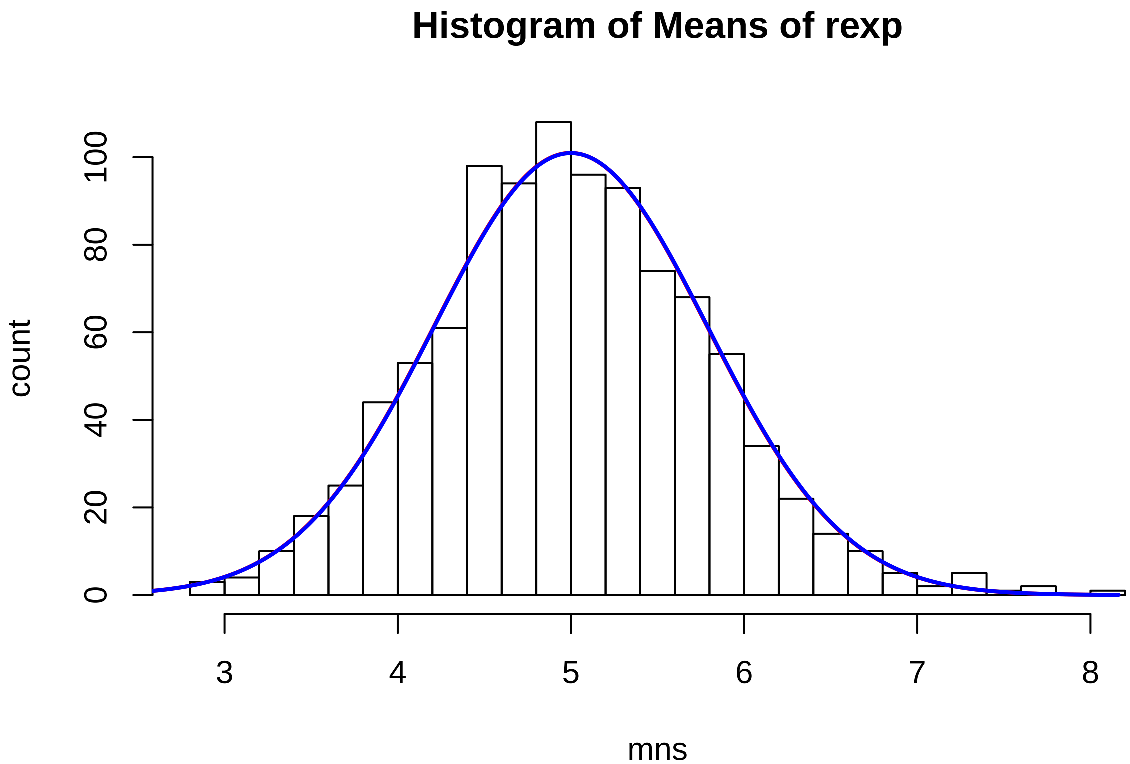
We first use R to numerically extract the theoretical variance and the sample variance of the distribution



Again we can see that the Sample Variance 0.6246197 is nearly equal to the Theoretical Variance 0.625. The above histogram also include red lines and blue lines giving the locations of the sample mean standard deviation and the theoretical mean standard deviation, which almost overlap with each other.

Is the distribution approximately normal?

As discussed previously, we have obtained the sample mean and the sample variance (which gives sample standard deviation(Ssd)). We, therefore, first use the sample mean and the Ssd to plot the normal distribtion on top of the histogram and compare it with the normal distribution obtained using theoretical mean and theoretical sd (Tsd).



The red curve is obtained based on the sample mean data, including sample mean and sample variance or sample standard deviation. The blue curve is obtained purely based on the theory with theoretical mean and theoretical variance. We can see that for 1000 times of simulations, they match each other quite well, which confirms the central limit theorem.

Appendix: Codes

```
### Codes for Sample Mean

## We first fix the value of lambda
# lambda <- 0.2

## We create a Null vector called mns
# mns <- c()

## For each i, we generate 40 data using exponential distribution with
## given lambda and calculate the mean of these 40 data and append it
## to the vector mns. We iterate it for 1000 times to generate a distribution.
# for (i in 1:1000) mns <- c(mns, mean(rexp(40,lambda)))

## Let's first show the distribution and calculate the sample mean
# hist(mns, xlab = "mns", ylab = "count", main = "Histogram of Means of rexp", breaks = 20)

## We add a verticle line to illustrate the sample mean
# abline( v = mean(mns), col=2, lwd =2)

## The mean of the distribution gives the sample mean
#samplemean <- mean(mns)
```

```
### Codes for Sample Variance versus Theoretical Variance
## The theoretical var can be obtained from the exponetial distribution
# Tvar <- 1/(lambda^2)/40

## The sample var is
# Svar <- var(mns)

## In order to compare Tvar and Svar using some plots, we can look at the
## standard deviation of the sample mean distribution which is
## basically the square root of the sample variance.
# hist(mns, xlab = "mns", ylab = "count", main = "Histogram of Means of rexp",
# breaks = 20)

## illustrate the theoretical sample sd
# abline( v = samplemean + sqrt(Tvar), col=4, lwd = 2)

## the sample sd
# abline( v = samplemean + sqrt(Svar), col=2, lwd = 2)

## illustrate the theoretical sample sd
# abline( v = samplemean - sqrt(Tvar), col=4, lwd = 2)

## the sample sd
# abline( v = samplemean - sqrt(Svar), col=2, lwd = 2)



### Codes for checking the consistence of normal distribution
## Let's first replot the histogram
# ht <- hist(mns, xlab = "mns", ylab = "count", main = "Histogram of Means of rexp",
# breaks = 20)

## The sample sd is as follows
# Ssd <- sqrt(Svar)

## The theoretical sd is
# Tsd <- sqrt(Tvar)

## Nowe we generate the normal distribution data using sample mean and Ssd
## We first generate sequence of data between samplemean +- 4 Ssd which
## should cover >99% data
# xsample<- seq(samplemean - 4*Ssd, samplemean + 4*Ssd, length = 1000)

## generate the normal distribution
# ysample <- dnorm(xsample, mean = samplemean, sd = Ssd)

## Rescale the y value
## Below ht$mids gives the mid points of the bars and diff(ht$mids[1:2])
## gives the difference between the first two mid points which should be
## universal distance between nearby mid points
## We then multiply it by length(xsample) which tells that how many points
```

```
## are contained withing each difference between mid points
# ysample <- ysample*diff(ht$mids[1:2])*length(xsample)

## Now we generate the normal distribution data using theoretical mean and Tsd
## Note that theoretical mean is 1/lambda = 5
# xtheo <- seq(1/lambda - 4 *Tsd, 1/lambda + 4 * Tsd, length = 1000)

## general the theoretical normal distribution
# ytheo <- dnorm(xtheo, mean = 1/lambda, sd = Tsd)

## Rescale
# ytheo <- ytheo*diff(ht$mids[1:2])*length(xtheo)

## Now we plot both normal distributions based on sample mean data and theoretical analysis
# lines(xsample,ysample, col =2, lwd = 2) ## sample normal
# lines(xtheo, ytheo, col = 4, lwd = 2) ## theoretical normal
```