

Computer Project2

Patrick Wong

11/07/2018

MATH 324 Computer Project 2

Algorithm 1: Generate 500 Sample Means from Sample Size n and for a Particular Distribution

```
###Algorithm)

#Sample size = 500
sample_size <- 500
#U(0.5)
a <- 0
b <- 5
#Binomial with number of trials = 15, p = 0.2
number_trials <- 15
p <- 0.2
mu <- 2
#Exp(5)
lambda <- 5

#1)
numList <- runif(sample_size, min = a, max = b)
#NA = numlist

storedData = rep(NA, sample_size)
for (i in 1:sample_size){
  storedData[i] = i
  #Generate random sample of size n for the particular distribution (see I-IV)
  #2)
  rb <- rbinom(numList, size = number_trials, prob = p)
  paste0(rb)
  #3)
  rxp <- rexp(numList, rate = lambda)
  paste0(rxp)
  #4)
  rp <- rpois(numList, mu)
  paste0(rp)
  #####
  #Take the average over the n sampled values.
  #Store the average into your vector from Step 1.
  numListVar <- var(numList)
}
paste0("numListMean: ", sum(numList)/sample_size)

## [1] "numListMean: 2.50822282937355"

paste0("numListVar: ", numListVar)

## [1] "numListVar: 2.10235563929066"
```

Question 1)

```
# if the sample size is 5
n <- 5
simulatedData5 <- matrix(rexp(n*sample_size, lambda), nrow=sample_size, ncol=n)
simMeans5 <- apply(simulatedData5, 1, mean)
sampleMean5 <- round(mean(simMeans5),3);
paste0("sampleMean for n = 5: ", sampleMean5)

## [1] "sampleMean for n = 5: 0.201"
theoreticalMean5 <- round(1/lambda,3);
paste0("theoreticalMean for n = 5: ", theoreticalMean5)

## [1] "theoreticalMean for n = 5: 0.2"

# if the sample size is 50
m <- 50
simulatedData50 <- matrix(rexp(m*sample_size, lambda), nrow=sample_size, ncol=m)
simMeans50 <- apply(simulatedData50, 1, mean)
sampleMean50 <- round(mean(simMeans50),3);
paste0("sampleMean for n = 50: ", sampleMean50)

## [1] "sampleMean for n = 50: 0.2"
theoreticalMean50 <- round(1/lambda,3);
paste0("theoreticalMean for n = 50: ", theoreticalMean50)

## [1] "theoreticalMean for n = 50: 0.2"
```

Question 2)

```
## if the sample size is 5
simulatedData5 <- matrix(rexp(n*sample_size, lambda), nrow=sample_size, ncol=n)
simMeans5 <- apply(simulatedData5, 1, mean)
sampleVar5 <- round(var(simMeans5), 3)
paste0("sampleVariance for n = 5: ", sampleVar5)

## [1] "sampleVariance for n = 5: 0.008"
theoretcalVar5 <- round((1/lambda)^2/n,3);
paste0("theoreticalVariance for n = 5: ", theoretcalVar5)

## [1] "theoreticalVariance for n = 5: 0.008"

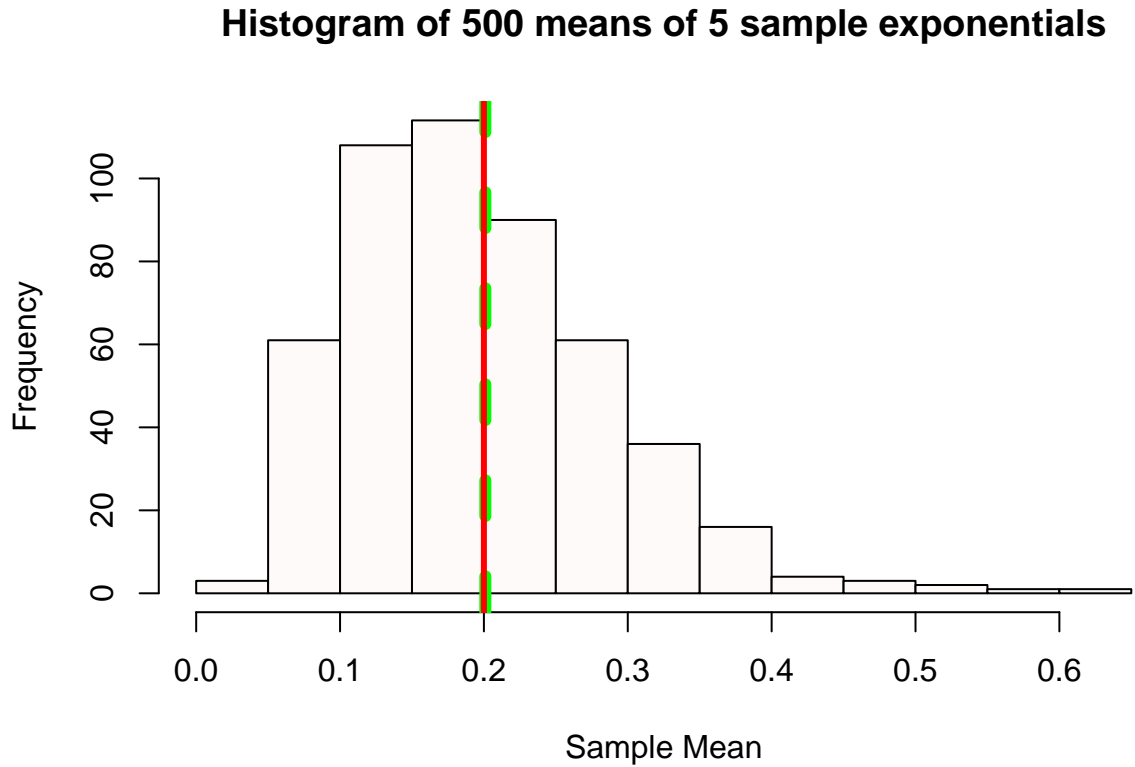
## if the sample size is 50
simulatedData50 <- matrix(rexp(m*sample_size, lambda), nrow=sample_size, ncol=m)
simMeans50 <- apply(simulatedData50, 1, mean)
sampleVar50 <- round(var(simMeans50), 3)
paste0("sampleVariance for n = 50: ", sampleVar50)

## [1] "sampleVariance for n = 50: 0.001"
theoretcalVar50 <- round((1/lambda)^2/n,3);
paste0("theoreticalVariance for n = 50: ", theoretcalVar50)

## [1] "theoreticalVariance for n = 50: 0.008"
```

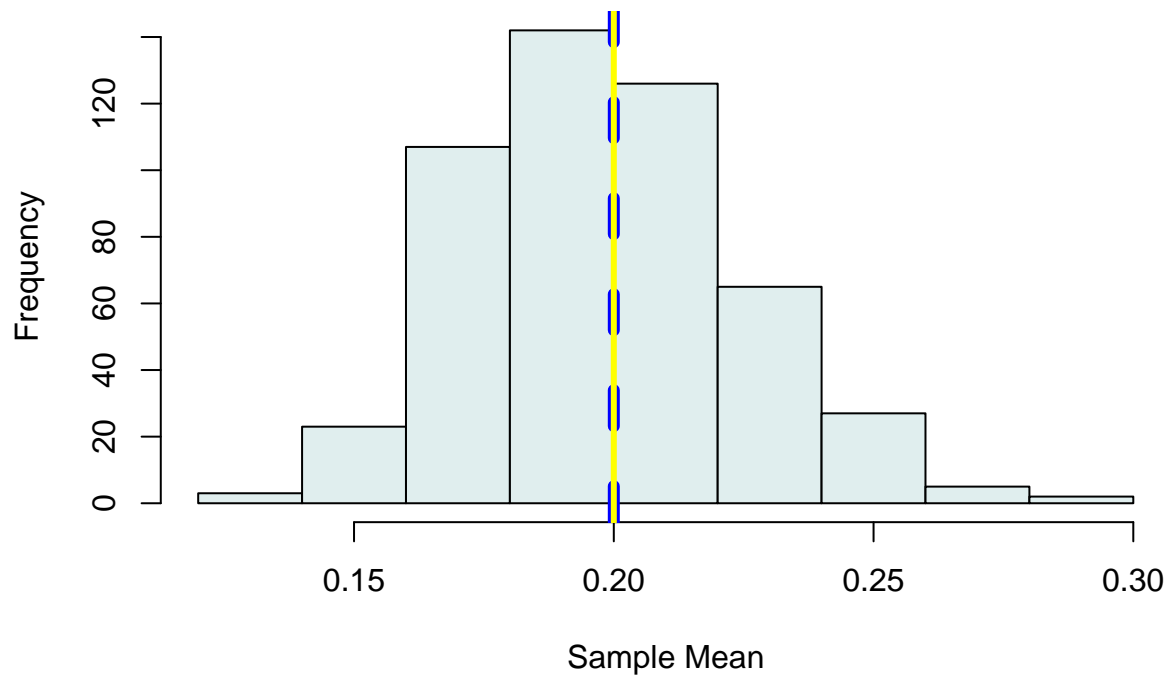
Question 3)

```
hist(simMeans5, col="snow", main="Histogram of 500 means of 5 sample exponentials", xlab="Sample Mean",  
abline(v=sampleMean5, col="green", lwd=6, lty=2)  
abline(v=theoreticalMean5, col="red", lwd=3)
```



```
hist(simMeans50, col="azure2", main="Histogram of 500 means of 50 sample exponentials", xlab="Sample Mean",  
abline(v=sampleMean50, col="blue", lwd=6, lty=2)  
abline(v=theoreticalMean50, col="yellow", lwd=3)
```

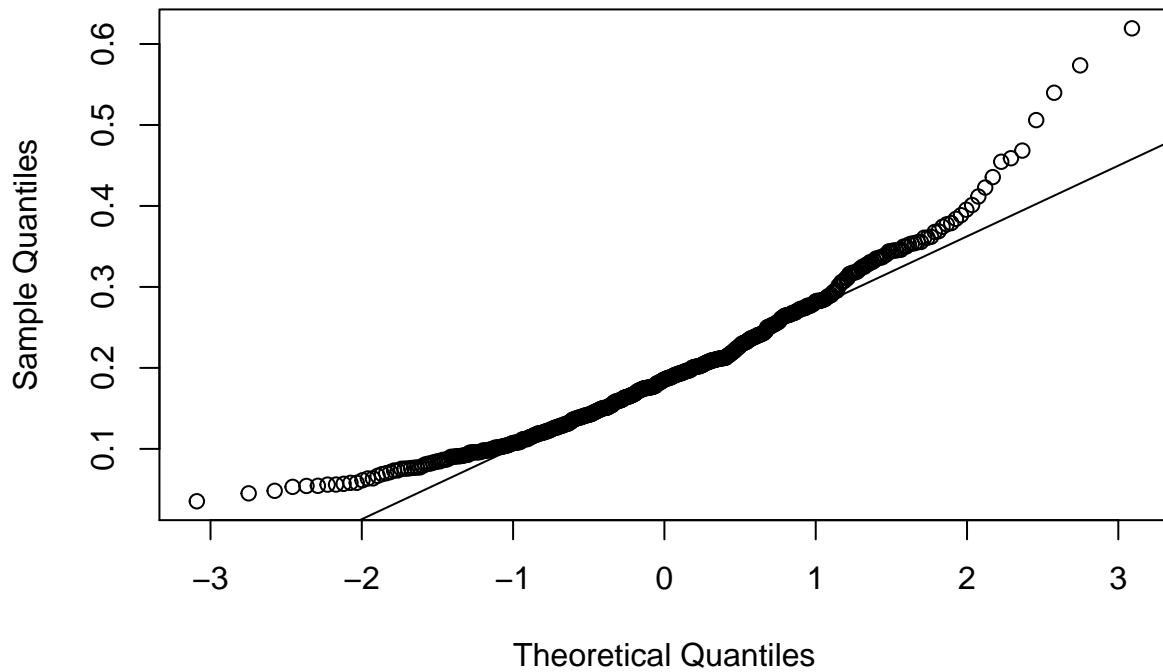
Histogram of 500 means of 50 sample exponentials



Question 4)

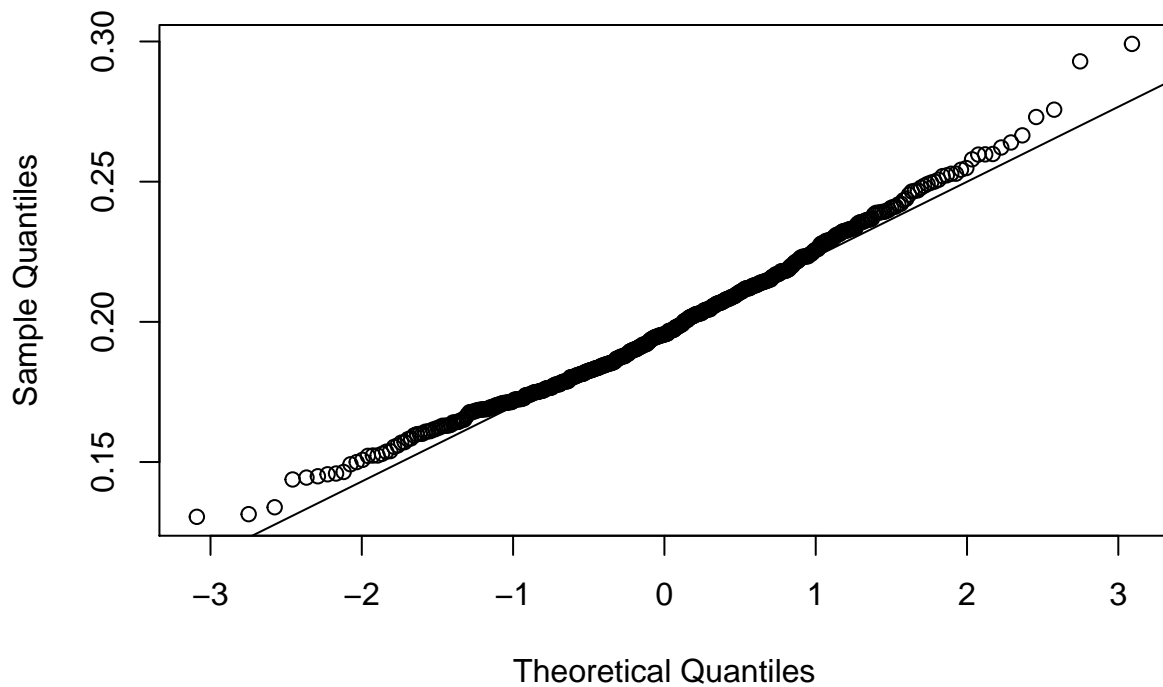
```
qqnorm(simMeans5)
qqline(simMeans5)
```

Normal Q-Q Plot



```
qqnorm(simMeans50)
qqline(simMeans50)
```

Normal Q-Q Plot



Question 5) :Summarize these findings for each of a-d, and use the central limit theorem to explain your findings. ###

It tells us that when the sample sizes get larger, the distribution of means will be calculated through repeating sampling until it reaches normality. It also tells us that the same population will be approximately equal to the mean of the population.

Algorithm 2: Use 40,000 Bootstrapped Means to Estimate Variance of the Mean

```
m2 <- 50
p2 <- 0.2
X <- rbinom(m2,prob = p, size = m2)
bMeans <- 40000
xMeanArray <- array(dim = bMeans)
for(i in 1:bMeans){
  # sample(<your_vector_of_50_binomial_random_numbers>, sample = 50, replace = TRUE)
  xMeanArray[i] <- mean(sample(X, n, replace = TRUE))
}
v <- var(xMeanArray)
```

A)

```
paste0("Bootstrap Variance: " , v)

## [1] "Bootstrap Variance: 1.53890880069502"
```

B)

```
#theoretical = np(1-p)/n
t <- m2*p2*(1-p2)/m2
paste0("Theoretical variance: ", t)

## [1] "Theoretical variance: 0.16"
```