**Mini Project 3**

**Name :**

**Maitri Dharmendrakumar Shah (mxs172030)**

**Nayana Thomas(nxt170630)**

**Contribution of team members:**

Maitri:

Wrote equation and calculation in word

Drew a graph about simulation and found the way to distribute it

Learned R Coding

Tried different R codes

Wrote narration for codes

Derived conclusions from calculations

Nayana:

Did calculation on paper

Explained the simulation

Learned R coding

Did debugging for R codes

Derived conclusions from graphs

Wrote explanation for section1

Que1. (a)

The mean square error of an estimator is defined as

MSE(θˆ) = E[(θ̂ − θ)2]

= Var(θ̂ ) + E[(θ̂)- θ

= Var( θ̂ ) + [Bias(θ̂)]2

where θ̂ is an estimator of θ, an unknown population parameter.

The bias of an estimator θ̂ of a parameter θ is the difference between the expected value of θ̂ and θ; that is,

Bias(θ̂) = E(θ̂) − θ.

If E(θ̂) = θ for all θ then the estimator is unbiased.

Thus, MSE has two components, one measures the variability of the estimator (precision) and the other measures the its bias (accuracy).

An estimator that has good MSE properties has small combined variance and bias. To find an estimator with good MSE properties we need to find estimators that control both variance and bias.

For an unbiased estimator θ̂, we have

MSE(θ̂) =E(θ̂ −θ)2 =Var(θ̂)

and so, if an estimator is unbiased, its MSE is equal to its variance.

If E(θ̂) != θ then the estimator has either a positive or negative bias. That is, on average the estimator tends to over (or under) estimate the population parameter.

Given the maximum likelihood estimator θ1̂=X(n) and the method of moments estimator, θ2̂= 2x̅, where x̅ is the sample mean.

To compare these two estimators, by Monte Carlo simulation for a specific n and θ:

1. Generate X1, ..., Xn ∼ Uniform(0, θ)

2. Calculate θ̂1 and θ2̂

3. Save (θ1̂ − θ)2 and (θ2̂ − θ)2

4. Repeat step 1-3 N times

5. Then the means of the (θ1̂ −θ)2 and (θ2̂ −θ)2, over the N replicates, are the monte carlo estimators of the MSEs of θ1̂ and θ2̂.

Que 1. (b)

Section 1:

Let N= 1000

n=1, θ=1 ,5,10,50,10,15

First, we will generate numbers from 0 to θ and then we will calculate values of θ̂1 and θ2̂ and then mean square error for both the estimators are counted. This can be done for N times and this is how Monte Carlo simulation is done.

Section2:

theta <- c(1,5,10,50,100)

#Matrix to store the MSEs of both the estimators

MSE <- matrix(0, length(theta), 2)

#Loop through values in theta

for(i in 1:length(theta))

{

dataset <- matrix(runif(1000\*1,0,theta[i]),1000,1)

dataset

# Calculate theta\_hat1 (mle) for each data set

thetaHat\_1 <- apply(dataset, 1, max)

thetaHat\_1

# Calculate theta\_hat2 (mom) for each data set

thetaHat\_2 <- 2\*apply(dataset, 1, mean)

thetaHat\_2

# Save the MSEs

MSE[i,1] <- mean((thetaHat\_1 - theta[i])^2)

MSE[i,2] <- mean((thetaHat\_2 - theta[i])^2)

}

# Plot the results on the same axes

plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",

main=expression(paste("MSE for each value of ", theta)),

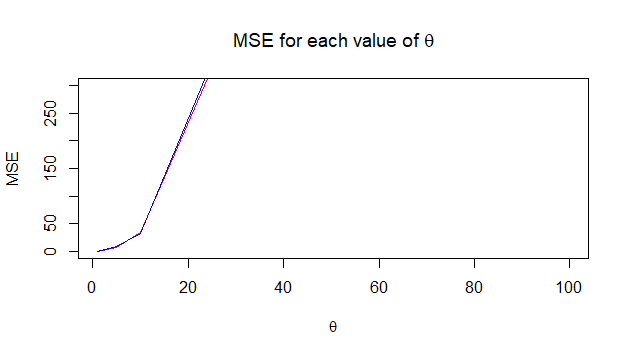
type="l", col="red",ylim = c(0,300))

lines(theta, MSE[,2], col="blue")

Narration of code:

Here, n=1 the code is written. First of all, MSE matrix is generated to store estimated values of θ̂1 and θ̂2. After generating numbers which are uniformly distributed, through apply method for both estimators their functions are implemented. After that mean square error is calculated for both estimators by applying the formula. At the last, through plot function, graph is plotted for both. Blue line is for θ̂1 and red line is for θ̂2.

Output:



Que 1. (c)

Section 1:

For different values of n, n\*N values are generated from uniform distribution and the results are summarized below.

For n=5

theta <- c(1,5,10,50,100)

MSE <- matrix(0, length(theta), 2)

for(i in 1:length(theta))

{

dataset <- matrix(runif(1000\*5,0,theta[i]),1000,5)

thetaHat\_1 <- apply(dataset, 1, max)

thetaHat\_2 <- 2\*apply(dataset, 1, mean)

MSE[i,1] <- mean((thetaHat\_1 - theta[i])^2)

MSE[i,2] <- mean((thetaHat\_2 - theta[i])^2)

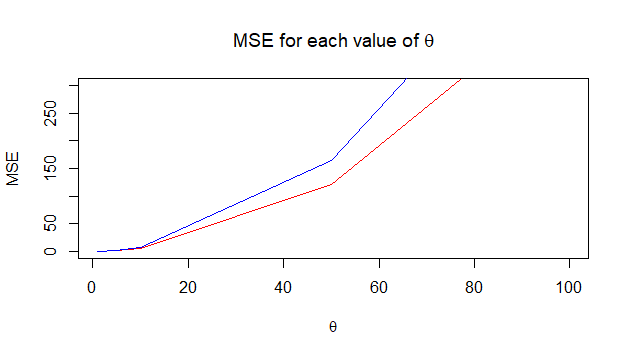
}

plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",

main=expression(paste("MSE for each value of ", theta)),

type="l", col="red",ylim = c(0,300))

lines(theta, MSE[,2], col="blue")



For n=10

theta <- c(1,5,10,50,100)

MSE <- matrix(0, length(theta), 2)

for(i in 1:length(theta))

{

dataset <- matrix(runif(1000\*10,0,theta[i]),1000,10)

thetaHat\_1 <- apply(dataset, 1, max)

thetaHat\_2 <- 2\*apply(dataset, 1, mean)

MSE[i,1] <- mean((thetaHat\_1 - theta[i])^2)

MSE[i,2] <- mean((thetaHat\_2 - theta[i])^2)

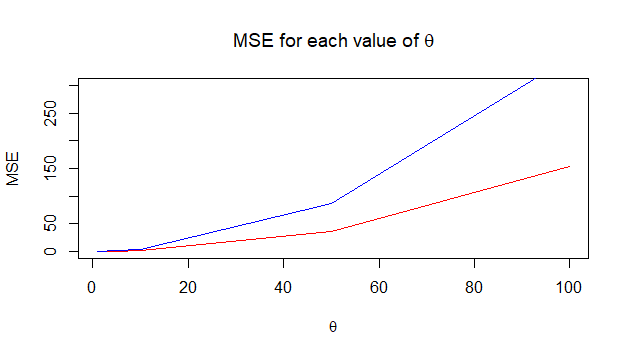
}

plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",

main=expression(paste("MSE for each value of ", theta)),

type="l", col="red",ylim = c(0,300))

lines(theta, MSE[,2], col="blue")



For n=50

theta <- c(1,5,10,50,100)

MSE <- matrix(0, length(theta), 2)

for(i in 1:length(theta))

{

dataset <- matrix(runif(1000\*50,0,theta[i]),1000,50)

thetaHat\_1 <- apply(dataset, 1, max)

thetaHat\_2 <- 2\*apply(dataset, 1, mean)

MSE[i,1] <- mean((thetaHat\_1 - theta[i])^2)

MSE[i,2] <- mean((thetaHat\_2 - theta[i])^2)

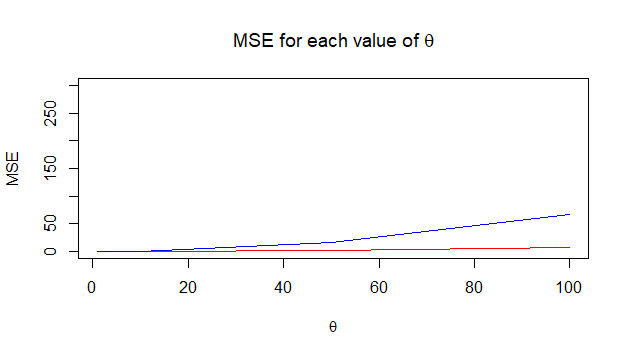
}

plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",

main=expression(paste("MSE for each value of ", theta)),

type="l", col="red",ylim = c(0,300))

lines(theta, MSE[,2], col="blue")



For n=100

theta <- c(1,5,10,50,100)

MSE <- matrix(0, length(theta), 2)

for(i in 1:length(theta))

{

dataset <- matrix(runif(1000\*100,0,theta[i]),1000,100)

thetaHat\_1 <- apply(dataset, 1, max)

thetaHat\_2 <- 2\*apply(dataset, 1, mean)

MSE[i,1] <- mean((thetaHat\_1 - theta[i])^2)

MSE[i,2] <- mean((thetaHat\_2 - theta[i])^2)

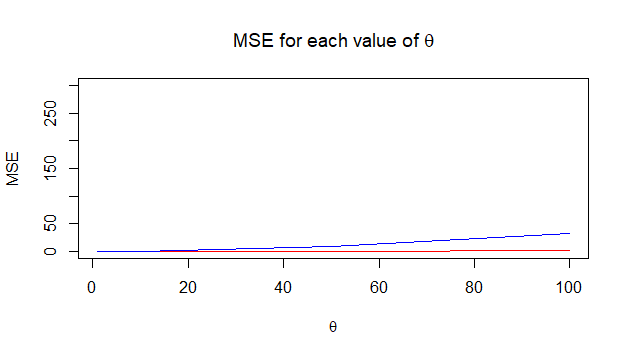
}

plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",

main=expression(paste("MSE for each value of ", theta)),

type="l", col="red",ylim = c(0,300))

lines(theta, MSE[,2], col="blue")



Section 2:

For different values of n, only one line is changed in the 1.(b) code which is :

dataset <- matrix(runif(1000\*n,0,theta[i]),1000,n)

Narration of code:

To change n, runif is done 1000\*n is done, and to store these value rows are increased.

Que 1.(d)

From 1.(c) we can see that for n=1 and for θ = 1, the MSE values of both θ1̂ and θ̂2 are lower compared to the θ1̂ and θ̂2 values when θ = 100. So we can see that as θ increases for a fixed n,ie here n=1, the MSE values of both the estimators are increasing with MSE of θ̂2 greater than MSE of θ̂1. So we can see that for small value of n and θ, the efficiency of both the maxium likelihood estimator and the method of moments estimator are the same. As θ increases MLE becomes more efficient than Method Of Moments estimator.

Similarly, we can see that for n=5 and for θ = 1, the MSE values of both θ1̂ and θ̂2 are lower compared to the θ1̂ and θ̂2 values when θ = 100. So we can see that as θ increases for a fixed n,ie here n=10, the MSE values of both the estimators are increasing with MSE of θ̂2 greater than MSE of θ̂1. So we can see that for small value of n and θ, the efficiency of both the maxium likelihood estimator and the method of moments estimator are the same. As θ increases MLE becomes more efficient than Method Of Moments estimator. Also when compared with n=1, MSE of both θ1̂ and θ̂2 are less in the case of n = 5. This shows that as n increases, MSE decreases and also for larger values of n, θ1̂ and θ̂2 gives more accurate estimation than for smaller values of n.

Similarly, we can see that for n=100 and for θ = 1, the MSE values of both θ1̂ and θ̂2 are lower compared to the θ1̂ and θ̂2 values when θ = 100. So we can see that as θ increases for a fixed n,ie here n=10, the MSE values of both the estimators are increasing with MSE of θ̂2 greater than MSE of θ̂1. So we can see that for small value of n and θ, the efficiency of both the maxium likelihood estimator and the method of moments estimator are the same. As θ increases MLE becomes more efficient than Method Of Moments estimator. Also when compared with n=1,5, 10 and 50 MSE of both θ1̂ and θ̂2 are less in the case of n = 100. This shows that as n increases, MSE decreases and also for larger values of n, θ1̂ and θ̂2 gives more accurate estimation than for smaller values of n.

In conclusion from the graph, we can say that θ̂1 has low MSE than θ2̂ for higher values of n and θ. So, θ1̂ is better estimator than θ̂2.

For smaller values of n and θ both the maximum likelihood estimator and the method of moments estimator are equally efficient.

As θ is increasing, MSE is increasing for both θ̂1 and θ̂2. But for higher θ, MSE for θ̂1 is less than MSE of θ̂2. If n is small, with respect to θ both estimators have higher MSE.

As n increases, for both estimators MSE is decreasing in different proportion with respect to θ.

But for any value of n, MSE of θ2̂ is more compared to MSE of θ̂1.