Statistical Methods for Data Science
Mini-Project #3

Group-46
Anish Joshi(UTD ID-2021591978), Aneena Manoj(UTD ID-2021623362)

Contributions – The mini-project was discussed in detail by both of us. Further, Anish and Aneena studied, discussed, and wrote the R code/script for both the questions and came up with the output, and wrote the report together. Both the partners worked efficiently to complete the mini-project.

SECTION - 1

Question 1

1a)

The following are the steps to compute the mean squared error of an estimator using monte carlo simulation.

1. The mean squared error is nothing but the estimated value of the squared difference between the estimator and the parameter.
2. Firstly, we need to find the required n and Theta.
3. Then, we need to calculate the method of moment estimator value by multiplying 2 to the mean of our x variable which contains the n and the Theta value.
4. Then, we can calculate the maximum likelihood estimator by taking the max of the x variable.
5. Now, we have to simulate the mean squared error function using monte carlo simulation.

1b)

In the second part, we need to compute the mean squared error of a combination (n, Theta) using Monte Carlo Simulation.

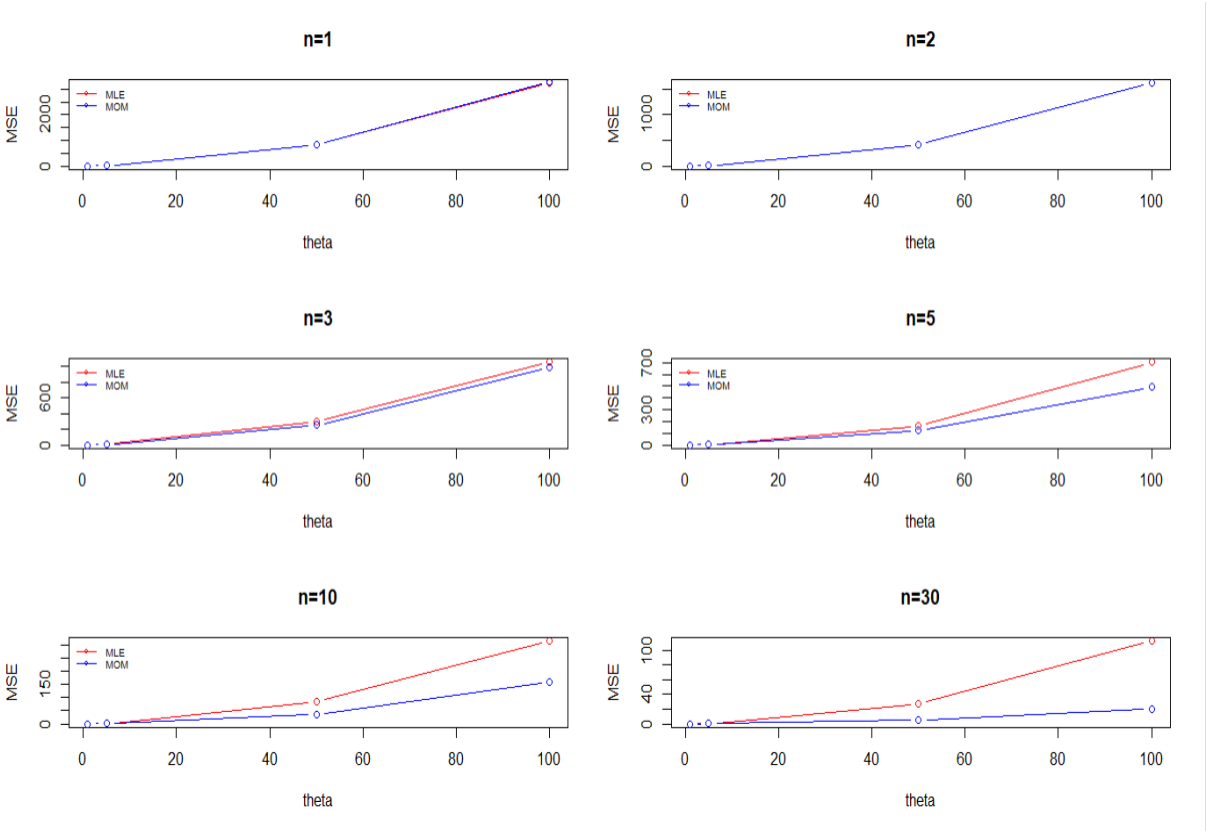
The function f_mse(n, theta) will take the sample from the uniform distribution and then calculate the MLE and MOM of the sample. The above function will be replicated 1000 times as mentioned in the question and it will calculate the MSE by using the $E\{(\hat{\theta} - \theta)/2\}$.

First, we will find the MSE for the combination of (1,1).

MSE(theta 1)=0.3468872
MSE(theta 2)=0.3395115

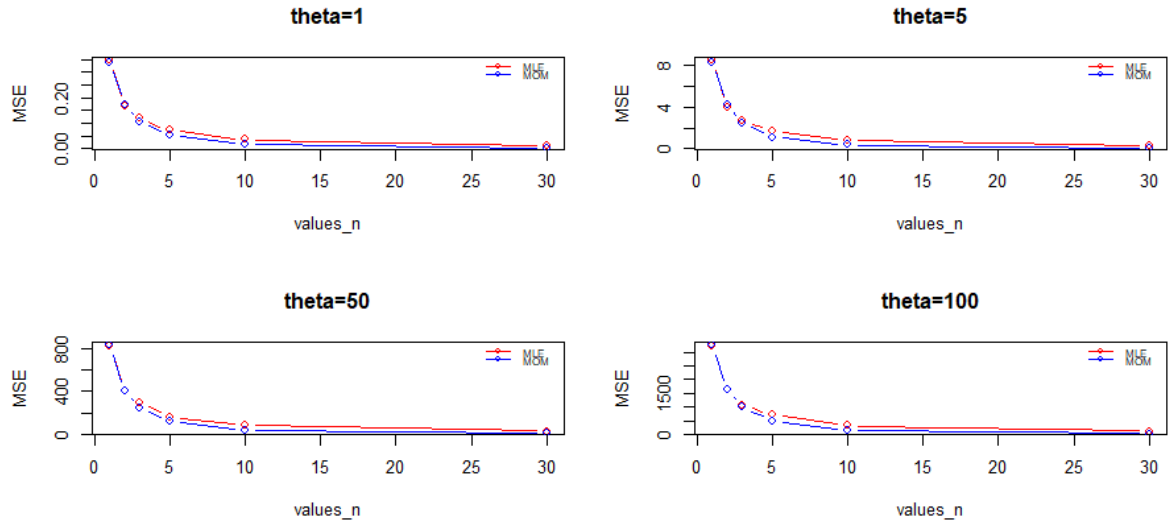
1c)

In this question, we need to plot graphs for the MSE function with different combinations. We will be initialising two variables to store different n values(values_n) and theta values(values_theta).
The required code for different combinations is in Section - 2.
The graphs are plotted in two ways to take into account every combination.
The first one is “Graphs with fixed n value and varying theta value”.
The R code is in SECTION-2.
Output–



The second one is “Graphs with fixed theta value and varying n value”.
The R code is in SECTION-2.

Output–



1d)

Comparing both the plots, we can see that the graphs with a fixed theta value are very similar so we can draw a conclusion that the estimator doesn't depend on the value of theta. From graph 1, it is evident that for the values of n being 1,2 and 3, MOM can be used as an estimator but for the values of n being 5,10 and 30, MLE seems more pragmatic to use as it has a lesser MSE value. Hence, we can conclude that the MLE (Maximum Likelihood Estimator) is the preferred choice since as the value of n increases, the MLE value doesn't increase that much as compared to MOM.

Question 2:

2a) Derive an expression for the maximum likelihood estimator of θ.

$$f(x) = \theta / X^{\theta+1} \quad where\ x \geq 1$$

Log likelihood of a sample

$$ln\, f(x) = \sum_{i=1}^{n} ln\, (\theta / X_i^{\theta+1}) = \sum_{i=1}^{n} (ln\, \theta - ln\, X_i^{\theta+1})$$

$$= n\, ln\, \theta - \sum_{1}^{n} (\, ln\, X_i^{\theta+1})$$

$$\partial / \partial \theta\ ln f(x) = n/\theta - \sum_{i}^{\theta+1} (\, ln\, X_i^{\theta+1}) \,/\, \sum_{i}^{\theta+1} (X_i^{\theta+1}) = 0$$

$$= n/\theta - \partial / \partial \theta\ (\theta + 1) \sum_{i}^{n} (X_i)$$

$$= n/\theta - \partial / \partial \theta\ (\theta . \sum_{i}^{n} (X_i) + \sum_{i}^{n} (X_i))$$

$$= n/\theta - \sum_{i}^{n} ln\, X_i$$

$$n/\theta - \sum_{i}^{n} ln\, X_i = 0$$

$$n/\theta = \sum_{i}^{n} ln\, X_i$$

$$\hat{\theta} = n / \sum_{i}^{n} ln\, X_i$$

2b) Use the expression in (a) to provide the maximum likelihood estimate for θ based on these data.

n = 5

x1 = 21.72, x2 = 14.65, x3 = 50.42, x4 = 28.78, x5 = 11.23

$$\hat{\theta} = n / \sum_{i}^{n} ln\, X_i$$

= 5/ (3.078 + 2.684 + 3.920 + 3.350 + 2.419)

= 5/15.461

= 0.3234

2c) Use the data in (b) to obtain the estimate by numerically maximizing the log-likelihood function using the optim function in R.

Using negative of log-likelihood function because, optim function by default give the minimum, but we want maximum.

```
x <- c(21.72, 14.65, 50.42, 28.78, 11.23)

# Negative of log-likelihood function

neg.loglik.fun <- function(theta, dat){
  result <- length(dat)*log(theta)-(theta+1)*sum(log(dat))
  return(-result)
}

# Minimize -log(L), i.e., maximize log(L)

ml.est <- optim(par=0.1, fn=neg.loglik.fun, method="L-BFGS-B", lower=0.1,
                hessian=TRUE, dat=x)
print(ml.est)
ml.est$par
```

```
> print(ml.est)
$par
[1] 0.3233885

$value
[1] 26.10585

$counts
function gradient
       9        9

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

$hessian
          [,1]
[1,] 47.81116

> ml.est$par
[1] 0.3233885
```

We got the theta hat value using calculations as 0.3234 and we got the theta hat value as 0.3233885 Using the R optim function. Hence our answers match.

2d) Use the output of numerical maximization in (c) to provide an approximate standard error of the maximum likelihood estimate and an approximate 95% confidence interval for θ.

To find the approximate 95% confidence interval for **θ**, we can use the following formula:

$$1 - \alpha = 0.95$$
$$\alpha = 1 - 0.95$$
$$\alpha = 0.05$$

$$\alpha/3 = 0.05/2 = 0.025$$
$$1 - \alpha/2 = 0.975$$

```
> #Standard error calculation
> SE <- sqrt(solve(ml.est$hessian))[1]
> print(SE)
[1] 0.1446223
> # The confidence interval
> ml.est$par + c(-1,1)*SE*qnorm(0.975)
[1] 0.03993389 0.60684301
```

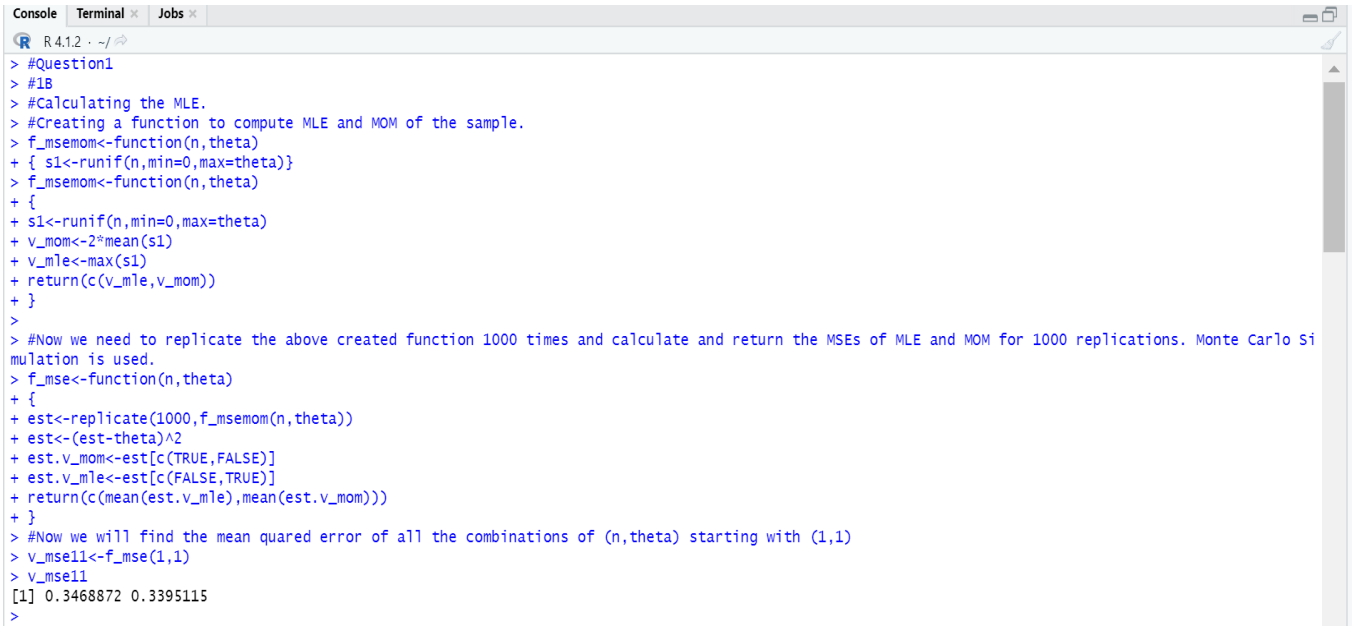
The approximate 95% confidence interval for **θ** is [0.03993389, 0.60684301].
This means that if we repeat a large number of times to estimate **θ** from randomly selected samples from the population, then the true estimate value lies in the interval [0.03993389, 0.60684301] 95% of the time.

If the sample size is large then the MLE follows an asymptotically normal distribution. But here the sample size is 5 which is not large. Thus, the confidence interval maybe not be very accurate.

SECTION - 2

Question - 1

1B]



```
Console  Terminal  Jobs
R 4.1.2 · ~/
> #Question1
> #1B
> #Calculating the MLE.
> #Creating a function to compute MLE and MOM of the sample.
> f_msemom<-function(n,theta)
+ { s1<-runif(n,min=0,max=theta)}
> f_msemom<-function(n,theta)
+ {
+ s1<-runif(n,min=0,max=theta)
+ v_mom<-2*mean(s1)
+ v_mle<-max(s1)
+ return(c(v_mle,v_mom))
+ }
>
> #Now we need to replicate the above created function 1000 times and calculate and return the MSEs of MLE and MOM for 1000 replications. Monte Carlo Si
mulation is used.
> f_mse<-function(n,theta)
+ {
+ est<-replicate(1000,f_msemom(n,theta))
+ est<-(est-theta)^2
+ est.v_mom<-est[c(TRUE,FALSE)]
+ est.v_mle<-est[c(FALSE,TRUE)]
+ return(c(mean(est.v_mle),mean(est.v_mom)))
+ }
> #Now we will find the mean quared error of all the combinations of (n,theta) starting with (1,1)
> v_mse11<-f_mse(1,1)
> v_mse11
[1] 0.3468872 0.3395115
>
```

1C]

```
>
> #1C
> #Repeating the same process as above for all the other combinations of n and theta
> #n=1,2,3,5,10,30
> #theta=1,5,50,100
> values_n<-c(1,2,3,5,10,30)
> values_theta<-c(1,5,50,100)
> v_mse15<-f_mse(1,5)
> v_mse150<-f_mse(1,50)
> v_mse1100<-f_mse(1,100)
> v_mse21<-f_mse(2,1)
> v_mse25<-f_mse(2,5)
> v_mse250<-f_mse(2,50)
> v_mse2100<-f_mse(2,100)
> v_mse31<-f_mse(3,1)
> v_mse35<-f_mse(3,5)
> v_mse350<-f_mse(3,50)
> v_mse3100<-f_mse(3,100)
> v_mse51<-f_mse(5,1)
> v_mse55<-f_mse(5,5)
> v_mse550<-f_mse(5,50)
> v_mse5100<-f_mse(5,100)
> v_mse101<-f_mse(10,1)
> v_mse105<-f_mse(10,5)
> v_mse1050<-f_mse(10,50)
> v_mse10100<-f_mse(10,100)
> v_mse301<-f_mse(30,1)
> v_mse305<-f_mse(30,5)
> v_mse3050<-f_mse(30,50)
> v_mse30100<-f_mse(30,100)
```

```
> #As per the question, we need to summarize the abpve values graphically.
> #Plotting graphs with a fixed n value and varying theta value
> par(mfrow=c(3,2))
> plot(values_theta,c(v_mse11[1],v_mse15[1],v_mse150[1],v_mse1100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=1")
Error in plot.new() : figure margins too large
> plot(values_theta,c(v_mse11[1],v_mse15[1],v_mse150[1],v_mse1100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=1")
Error in plot.new() : figure margins too large
> plot(values_theta,c(v_mse11[1],v_mse15[1],v_mse150[1],v_mse1100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=1")
> lines(values_theta,c(v_mse11[2],v_mse15[2],v_mse150[2],v_mse[1100]))
Error in xy.coords(x, y) : object 'v_mse' not found
> lines(values_theta,c(v_mse11[2],v_mse15[2],v_mse150[2],v_mse1100[2]),type="b",col='blue')
> legend("topleft",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_theta,c(v_mse21[1],v_mse25[1],v_mse250[1],v_mse2100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=2")
> lines(values_theta,c(v_mse21[2],v_mse25[2],v_mse250[2],v_mse2100[2]),type="b",col='blue')
> legend("topleft",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_theta,c(v_mse31[1],v_mse35[1],v_mse350[1],v_mse3100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=3")
> lines(values_theta,c(v_mse31[2],v_mse35[2],v_mse350[2],v_mse3100[2]),type="b",col='blue')
> legend("topleft",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_theta,c(v_mse51[1],v_mse55[1],v_mse550[1],v_mse5100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=5")
> lines(values_theta,c(v_mse51[2],v_mse55[2],v_mse550[2],v_mse5100[2]),type="b",col='blue')
> legend("topleft",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_theta,c(v_mse101[1],v_mse105[1],v_mse1050[1],v_mse10100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=10")
> lines(values_theta,c(v_mse101[2],v_mse105[2],v_mse1050[2],v_mse10100[2]),type="b",col='blue')
> legend("topleft",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_theta,c(v_mse301[1],v_mse305[1],v_mse3050[1],v_mse30100[1]),type="b",xlab="theta",ylab="MSE",col='red',main="n=30")
> lines(values_theta,c(v_mse301[2],v_mse305[2],v_mse3050[2],v_mse30100[2]),type="b",col='blue')
>
>


>
> #drawing graphs with fixed theta value and varying n value.
> par(mfrow=c(2,2))
> plot(values_n,c(v_mse11[1],v_mse21[1],v_mse31[1],v_mse51[1],v_mse101[1]),type="b",ylab="MSE",col='red',main="theta=1")
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
> plot(values_n,c(v_mse11[1],v_mse21[1],v_mse31[1],v_mse51[1],v_mse101[1],v_mse301[1]),type="b",ylab="MSE",col='red',main="theta=1")
> lines(values_theta,c(v_mse11[2],v_mse21[2],v_mse51[2],v_mse101[2],v_mse301[2]),type="b",col='blue')
Error in xy.coords(x, y) : 'x' and 'y' lengths differ
> lines(values_n,c(v_mse11[2],v_mse21[2],v_mse51[2],v_mse101[2],v_mse301[2]),type="b",col='blue')
Error in xy.coords(x, y) : 'x' and 'y' lengths differ
> lines(values_n,c(v_mse11[2],v_mse21[2],v_mse31[2],v_mse51[2],v_mse101[2],v_mse301[2]),type="b",col='blue')
> legend("topright",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_n,c(v_mse15[1],v_mse25[1],v_mse35[1],v_mse55[1],v_mse105[1],v_mse305[1]),type="b",ylab="MSE",col='red',main="theta=5")
> lines(values_n,c(v_mse15[2],v_mse25[2],v_mse35[2],v_mse55[2],v_mse105[2],v_mse305[2]),type="b",col='blue')
> legend("topright",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_n,c(v_mse150[1],v_mse250[1],v_mse350[1],v_mse550[1],v_mse1050[1],v_mse3050[1]),type="b",ylab="MSE",col='red',main="theta=50")
> lines(values_n,c(v_mse150[2],v_mse250[2],v_mse350[2],v_mse550[2],v_mse1050[2],v_mse3050[2]),type="b",col='blue')
> legend("topright",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
> plot(values_n,c(v_mse1100[1],v_mse2100[1],v_mse3100[1],v_mse5100[1],v_mse10100[1],v_mse30100[1]),type="b",ylab="MSE",col='red',main="theta=100")
> lines(values_n,c(v_mse1100[2],v_mse2100[2],v_mse3100[2],v_mse5100[2],v_mse10100[2],v_mse30100[2]),type="b",col='blue')
> legend("topright",legend=c("MLE","MOM"),col=c('red','blue'),text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,cex=0.6,bty='n')
>
```

2C]

```
> x <- c(21.72, 14.65, 50.42, 28.78, 11.23)
>
> # Negative of log-likelihood function
>
> neg.loglik.fun <- function(theta, dat){
+   result <- length(dat)*log(theta)-(theta+1)*sum(log(dat))
+   return(-result)
+ }
>
> # Minimize -log(L), i.e., maximize log(L)
>
> ml.est <- optim(par=0.1, fn=neg.loglik.fun, method="L-BFGS-B", lower=0.1,
+                 hessian=TRUE, dat=x)
> print(ml.est)
$par
[1] 0.3233885

$value
[1] 26.10585

$counts
function gradient
       9        9

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

$hessian
         [,1]
[1,] 47.81116

> ml.est$par
[1] 0.3233885
```

2D]

```
> #Standard error calculation
> SE <- sqrt(solve(ml.est$hessian))[1]
> print(SE)
[1] 0.1446223
> # The confidence interval
> ml.est$par + c(-1,1)*SE*qnorm(0.975)
[1] 0.03993389 0.60684301
```