Mini Project 3 (Solutions)

Mini Project Duo Group # 12
Contribution of each group member
Chetan Siddappareddy – 50%
Ankit Sahu – 50%
Both of us have contributed equally to the project. We learnt R through collaboration and then write the R scripts for the corresponding and report all the findings

Section 1: Answers to the specific questions asked

Question 1

a)

Below are the steps to compute the mean squared error of an estimator using Monte Carlo simulation.

1. Set the population parameter θ and simulate the sample values.

2. Compute maximum likelihood estimator ($\hat{\theta}$) from the sample simulated.

3. Repeat the process multiple times for different θ.

4. Compute the square of difference each time between $\hat{\theta}$ and θ.

5. Now compute the average of the squared differences between $\hat{\theta}$ and θ.

b)

R code is included in the end or in the section 2.

The function mlemomFunction(n, θ) generates samples from a uniform distribution and for each sample, it computes MLE and MOM and returns those two values in an array.

mseEst(n, θ) calls mlemomFunction(n, θ) 1000 times and computes the mean squared errors for both MLE and MOM estimators in an array format using the below formula E{(θ - $\hat{\theta}$)2}

Below are the results of the code

```
> #Function that calculates and returns the mean squared errors of MLE/MOM of
 1000 samples
> mseEst = function(n, theta) {
+   est = replicate(1000, mlemomFunction(n, theta))
+   est = (est - theta)^2
+   est.mleEst = est[c(FALSE, TRUE)]
+   est.momEst = est[c(TRUE, FALSE)]
+   return(c(mean(est.mleEst),mean(est.momEst)))
+ }
>
> mseEst(1,1)
[1] 0.3326646 0.3291113
> mseEst(1,5)
[1] 8.542441 8.461265
> mseEst(1,50)
[1] 858.0344 835.5789
> mseEst(1,100)
[1] 3381.136 3281.193
> mseEst(2,1)
[1] 0.1576493 0.1582821
> mseEst(2,5)
[1] 4.536386 4.182274
> mseEst(2,50)
[1] 416.1525 398.3780
> mseEst(2,100)
[1] 1680.122 1752.292
```

```
> mseEst(3,1)
[1] 0.11097943 0.09831679
> mseEst(3,5)
[1] 2.701147 2.490739
> mseEst(3,50)
[1] 266.3447 246.0571
> mseEst(3,100)
[1] 1175.377 1043.010
> mseEst(5,1)
[1] 0.06327386 0.04720110
> mseEst(5,5)
[1] 1.776261 1.184668
> mseEst(5,50)
[1] 179.0886 123.1334
> mseEst(5,100)
[1] 688.4044 413.1258
> mseEst(10,1)
[1] 0.03373666 0.01486866
> mseEst(10,5)
[1] 0.9188146 0.3857225
> mseEst(10,50)
[1] 81.89888 36.59023
> mseEst(10,100)
[1] 343.5163 150.0917
> mseEst(30,1)
[1] 0.011168428 0.001992115
```

```
> mseEst(30,5)
[1] 0.29489926 0.04769021
> mseEst(30,50)
[1] 26.313952  5.086711
> mseEst(30,100)
[1] 115.49496  20.29594
```

c)

R code is included in the end or section 2

Below are the graphs summarized for different combinations of (n, θ)

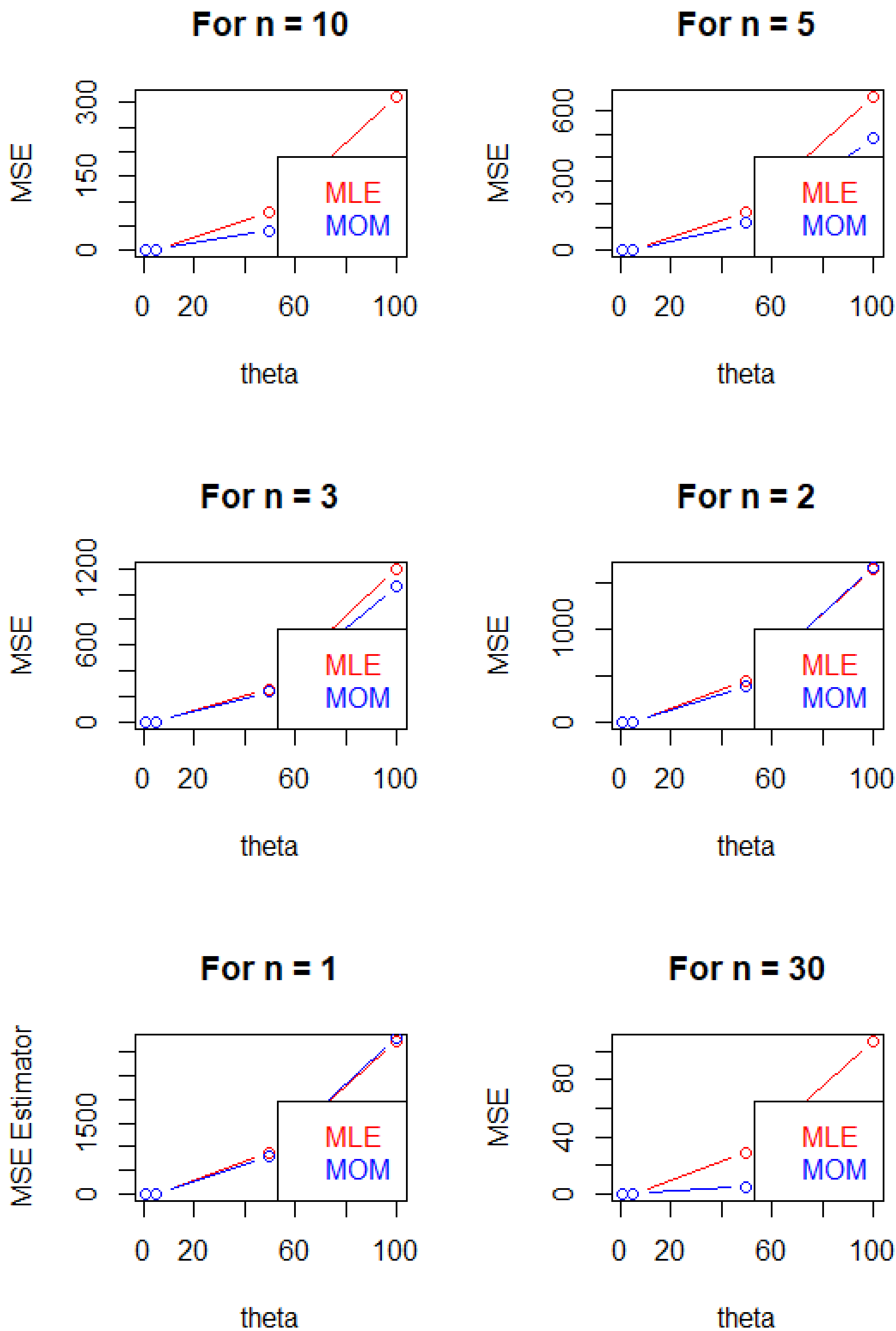

Fig:Graphs for Mean Squared error of MLE and MOM, fixed n varying θ

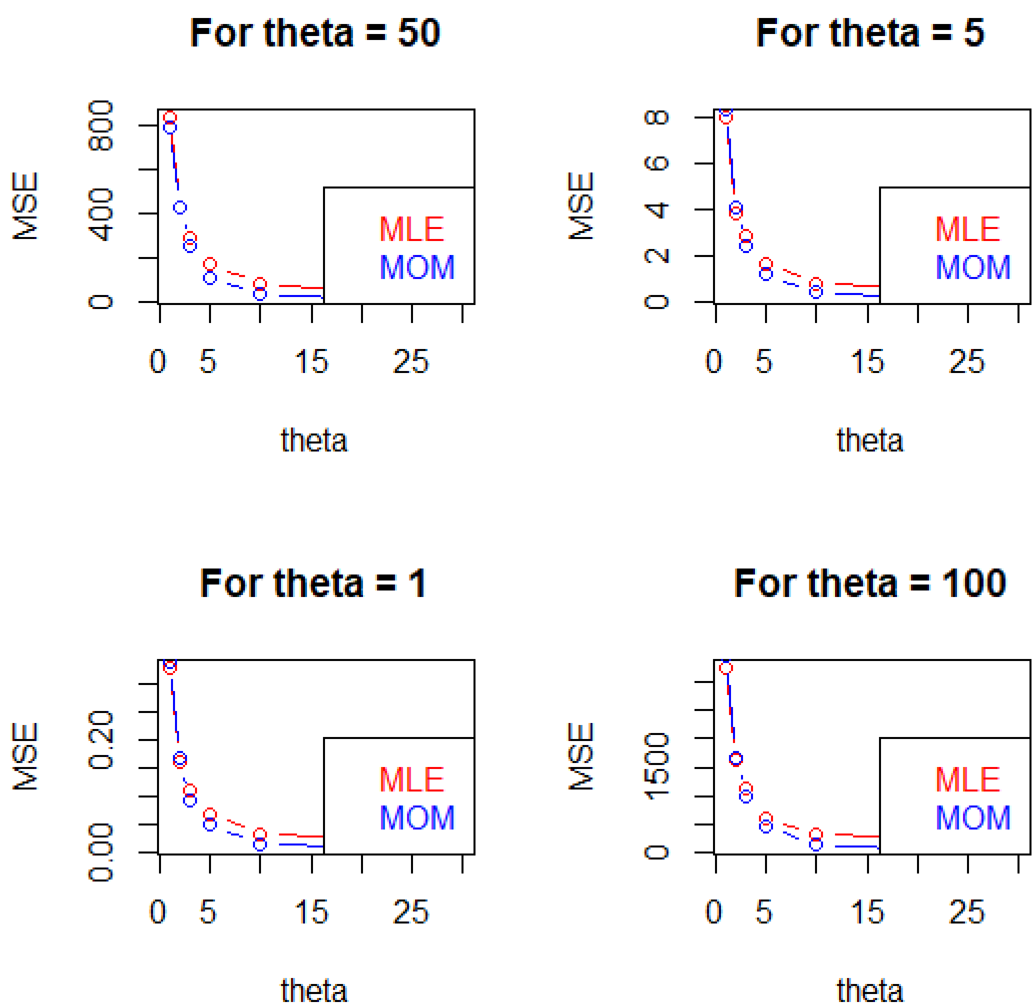

Fig:Graphs for Mean Squared error of MLE and MOM, varying n and fixed θ

d) Below are the observations

Observation 1: Graph 1 shows the plots of Mean Squared Errors varying with θ with fixed n. For small values of n, say, n = 1, 2, 3, it is obvious that Method of Moments estimator is good, while for large values of n, say, n = 5, 10, 30, the maximum likelihood estimator is better as mean squared error is less when compared to MOM for the same value of n.

Observation 2: From the second graph, it is apparent that graphs are exceptionally similar when θ get fixed. Hence it can be inferred that estimator is independent of θ.

Question 2

a) Consider the Likelihood function is $L((\theta) = \prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}$

Taking logarithm on both sides

$$\log (L((\theta)) = \log (\prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}})$$
$$= \log (\theta^n \mathrm{X} \prod_{i=1}^{n} \frac{1}{x_i^{\theta+1}})$$
$$= nlog\theta + \sum_{i=1}^{n} \log (x_i^{-\theta-1})$$
$$= nlog\theta - (\theta+1) \sum_{i=1}^{n} logx_i$$
$$= nlog\theta - \theta \sum_{i=1}^{n} logx_i - \sum_{i=1}^{n} logx_i$$

Partially differentiation of the above equation gives,

$$\frac{n}{\theta} - \sum_{i=1}^{n} logx_i$$

Equating to 0, to find the maximum

$$\frac{n}{\theta} - \sum_{i=1}^{n} logx_i = 0$$
$$\frac{n}{\theta} = \sum_{i=1}^{n} logx_i$$
$$\theta_{MLE} = \frac{n}{\sum_{i=1}^{n} logx_i}$$

b) Given values are plugged into the equation as follows:

$$\theta_{MLE} = \frac{5}{\log(21.72) + \log(14.65) + \log(50.42) + \log(28.78) + \log(11.23)}$$

Since log(a*b) = log(a) + log(b)

- ⇨ $\theta_{MLE} = \frac{5}{\log (21.72*14.65*50.42*28.78*11.23)}$
- ⇨ $\theta_{MLE} = \frac{5}{\log (5185263.5231)}$
- ⇨ $\theta_{MLE} = \frac{5}{15.4613}$
- ⇨ $\theta_{MLE} = 0.3233$

c) R code is included in the section 2. From the results of R Code as shown below, the output is 0.3236796

F

```
> #Function to return negative log-likelihood value
> logLikelihoodFunction <- function(param, data) {
+   logLikelihood = length(data)*log(param)-(param+1)*sum(log(data))
+   return(-logLikelihood)
+ }
>
> #Optimum function to minimize the negative log likelihood value
> optim(par=1, fn=logLikelihoodFunction,method = "L-BFGS-B", hessian=TRUE, low
er=0.01, data=c(21.42,14.65,50.42,28.78,11.23))[1]
$par
[1] 0.3236796
```

d) The standard error is:

SE($\hat{\theta}$) $\approx \sqrt{\hat{I}^{-1}}$

Where $\hat{I}$ is hessian function

1- α = 0.95

α = 0.05

⇨ 1-(α/2) = 0.975

The confidence interval is given by $\hat{\theta} \pm Z_{\alpha/2}$X SE($\hat{\theta}$)

qnorm is used in R to get $Z_{\alpha/2}$ value

As shown below, the confidence obtained is (0.03996984 0.60738939)

```
> #Standard Error
> x<- optim(par=1, fn=logLikelihoodFunction,method = "L-BFGS-B", hessian=TRUE,
 lower=0.01, data=c(21.42,14.65,50.42,28.78,11.23))
> stdError <- (1/x$hessian)^(1/2)
>
> #Confidence interval
> x$par + c(-1,1)*stdError*qnorm(0.975)
[1] 0.03996984 0.60738939
```

Section 2: R Code

#############

#R code for Question 1

#############

#**1b)** Function to calculate MLE/MOME for the sample

mlemomFunction <- function(n, theta) {

```
  simSample = runif(n, min=0, max=theta)
  momEst = 2*mean(simSample)
  mleEst = max(simSample)
  return(c(mleEst, momEst))
}

#Function that calculates and returns the mean squared errors of MLE/MOM of 1000 samples
mseEst = function(n, theta) {
  est = replicate(1000, mlemomFunction(n, theta))
  est = (est - theta)^2
  est.mleEst = est[c(FALSE, TRUE)]
  est.momEst = est[c(TRUE, FALSE)]
  return(c(mean(est.mleEst),mean(est.momEst)))
}

mseEst(1,1)
mseEst(1,5)
mseEst(1,50)
mseEst(1,100)
mseEst(2,1)
mseEst(2,5)
mseEst(2,50)
mseEst(2,100)
mseEst(3,1)
mseEst(3,5)
mseEst(3,50)
mseEst(3,100)
mseEst(5,1)
mseEst(5,5)
```

```
mseEst(5,50)

mseEst(5,100)

mseEst(10,1)

mseEst(10,5)

mseEst(10,50)

mseEst(10,100)

mseEst(30,1)

mseEst(30,5)

mseEst(30,50)

mseEst(30,100)

#1c)

#graphs for fixed n and varying theta

par(mfrow = c(5,5))

#For n = 1

plot(c(1, 5, 50, 100),c(mseEst(1,1)[1], mseEst(1,5)[1], mseEst(1,50)[1], mseEst(1,100)[1]), type="b", col="red", main="For n = 1", xlab="theta", ylab="MSE Estimator")

lines(c(1, 5, 50, 100), c(mseEst(1,1)[2], mseEst(1,5)[2], mseEst(1,50)[2], mseEst(1,100)[2]), type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For n = 2

plot(c(1, 5, 50, 100),c(mseEst(2,1)[1], mseEst(2,5)[1], mseEst(2,50)[1], mseEst(2,100)[1]), type="b", col="red", main="For n = 2", xlab="theta", ylab="MSE")

lines(c(1, 5, 50, 100), c(mseEst(2,1)[2], mseEst(2,5)[2], mseEst(2,50)[2], mseEst(2,100)[2]), type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For n = 3

plot(c(1, 5, 50, 100),c(mseEst(3,1)[1], mseEst(3,5)[1], mseEst(3,50)[1], mseEst(3,100)[1]), type="b", col="red", main="For n = 3", xlab="theta", ylab="MSE")
```

```
lines(c(1, 5, 50, 100), c(mseEst(3,1)[2], mseEst(3,5)[2], mseEst(3,50)[2], mseEst(3,100)[2]), type="b",
col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For n = 5

plot(c(1, 5, 50, 100),c(mseEst(5,1)[1], mseEst(5,5)[1], mseEst(5,50)[1], mseEst(5,100)[1]), type="b",
col="red", main="For n = 5", xlab="theta", ylab="MSE")

lines(c(1, 5, 50, 100), c(mseEst(5,1)[2], mseEst(5,5)[2], mseEst(5,50)[2], mseEst(5,100)[2]), type="b",
col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For n = 10

plot(c(1, 5, 50, 100),c(mseEst(10,1)[1], mseEst(10,5)[1], mseEst(10,50)[1], mseEst(10,100)[1]), type="b",
col="red", main="For n = 10", xlab="theta", ylab="MSE")

lines(c(1, 5, 50, 100), c(mseEst(10,1)[2], mseEst(10,5)[2], mseEst(10,50)[2], mseEst(10,100)[2]),
type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For n = 30

plot(c(1, 5, 50, 100),c(mseEst(30,1)[1], mseEst(30,5)[1], mseEst(30,50)[1], mseEst(30,100)[1]), type="b",
col="red", main="For n = 30", xlab="theta", ylab="MSE")

lines(c(1, 5, 50, 100), c(mseEst(30,1)[2], mseEst(30,5)[2], mseEst(30,50)[2], mseEst(30,100)[2]),
type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#graphs for varying n and fixed theta

par(mfrow = c(3,2))

#For theta = 1

plot(c(1, 2, 3, 5, 10, 30),c(mseEst(1,1)[1], mseEst(2,1)[1], mseEst(3,1)[1], mseEst(5,1)[1], mseEst(10,1)[1],
mseEst(30,1)[1]), type="b", col="red", main="For theta = 1", xlab="theta", ylab="MSE")

lines(c(1, 2, 3, 5, 10, 30), c(mseEst(1,1)[2], mseEst(2,1)[2], mseEst(3,1)[2], mseEst(5,1)[2],
mseEst(10,1)[2], mseEst(30,1)[2]), type="b", col="blue")
```

```
legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For theta = 5

plot(c(1, 2, 3, 5, 10, 30),c(mseEst(1,5)[1], mseEst(2,5)[1], mseEst(3,5)[1], mseEst(5,5)[1], mseEst(10,5)[1], mseEst(30,5)[1]), type="b", col="red", main="For theta = 5", xlab="theta", ylab="MSE")

lines(c(1, 2, 3, 5, 10, 30), c(mseEst(1,5)[2], mseEst(2,5)[2], mseEst(3,5)[2], mseEst(5,5)[2], mseEst(10,5)[2], mseEst(30,5)[2]), type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For theta = 50

plot(c(1, 2, 3, 5, 10, 30),c(mseEst(1,50)[1], mseEst(2,50)[1], mseEst(3,50)[1], mseEst(5,50)[1], mseEst(10,50)[1], mseEst(30,50)[1]), type="b", col="red", main="For theta = 50", xlab="theta", ylab="MSE")

lines(c(1, 2, 3, 5, 10, 30), c(mseEst(1,50)[2], mseEst(2,50)[2], mseEst(3,50)[2], mseEst(5,50)[2], mseEst(10,50)[2], mseEst(30,50)[2]), type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))


#For theta = 100

plot(c(1, 2, 3, 5, 10, 30),c(mseEst(1,100)[1], mseEst(2,100)[1], mseEst(3,100)[1], mseEst(5,100)[1], mseEst(10,100)[1], mseEst(30,100)[1]), type="b", col="red", main="For theta = 100", xlab="theta", ylab="MSE")

lines(c(1, 2, 3, 5, 10, 30), c(mseEst(1,100)[2], mseEst(2,100)[2], mseEst(3,100)[2], mseEst(5,100)[2], mseEst(10,100)[2], mseEst(30,100)[2]), type="b", col="blue")

legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red", "blue"))
```

#############

#R code for Question 2

#############

#2c) logLikelihoodFunction is a function that returns negative log-likelihood value

```
logLikelihoodFunction <- function(par, data) {

  logLikelihood = length(data)*log(par)-(par+1)*sum(log(data))
```

```
  return(-logLikelihood)

}
```

#2d) Optimum function to minimize the obtained negative log likelihood value

```
optim(par=1, fn=logLikelihoodFunction,method = "L-BFGS-B", hessian=TRUE, lower=0.01, data=c(21.42,14.65,50.42,28.78,11.23))[1]

#Standard Error

x<- optim(par=1, fn=logLikelihoodFunction,method = "L-BFGS-B", hessian=TRUE, lower=0.01, data=c(21.42,14.65,50.42,28.78,11.23))

stdError <- (1/x$hessian)^(1/2)


#Confidence interval

x$par + c(-1,1)*stdError*qnorm(0.975)
```