Mini Project 3

Mohammed Allama Hossain, Lavanya Gopal
We worked on the questions separately, and then we met up and checked our results with each other and verified it. During the same meeting we collaborated and completed the coding part of the assignment. Lavanya then ran the simulations and Allama compiled the results and completed the documentation.

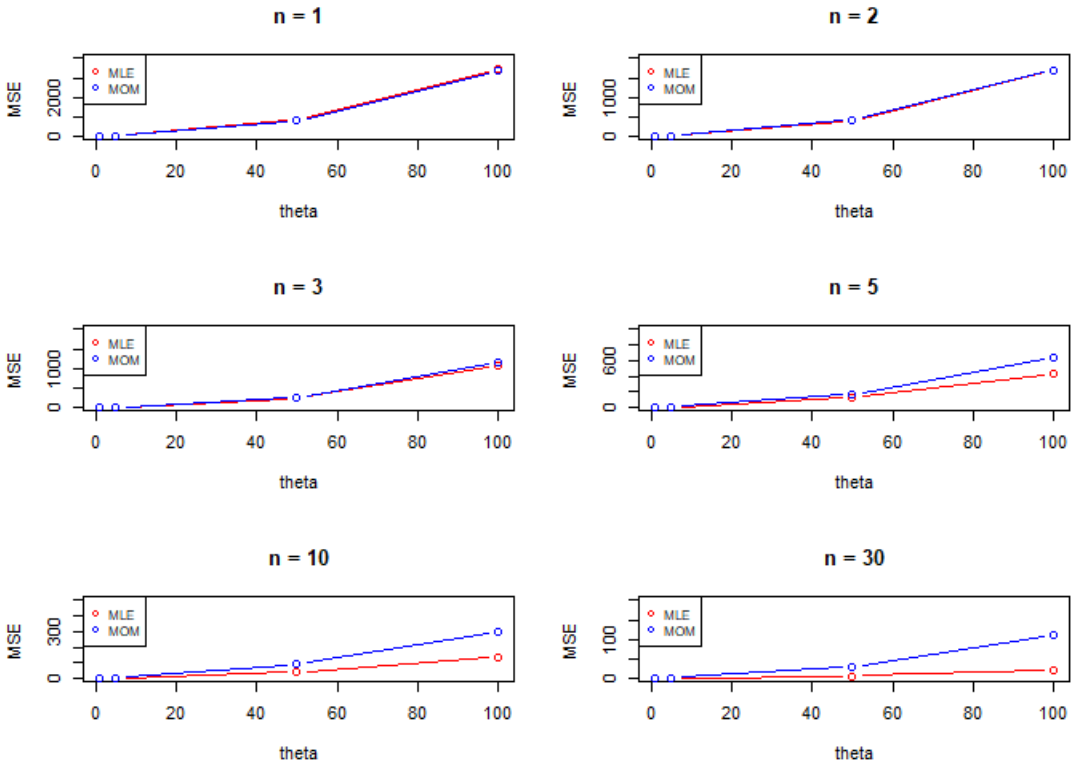
Section 1: Answer to the specific questions asked

Question 1.

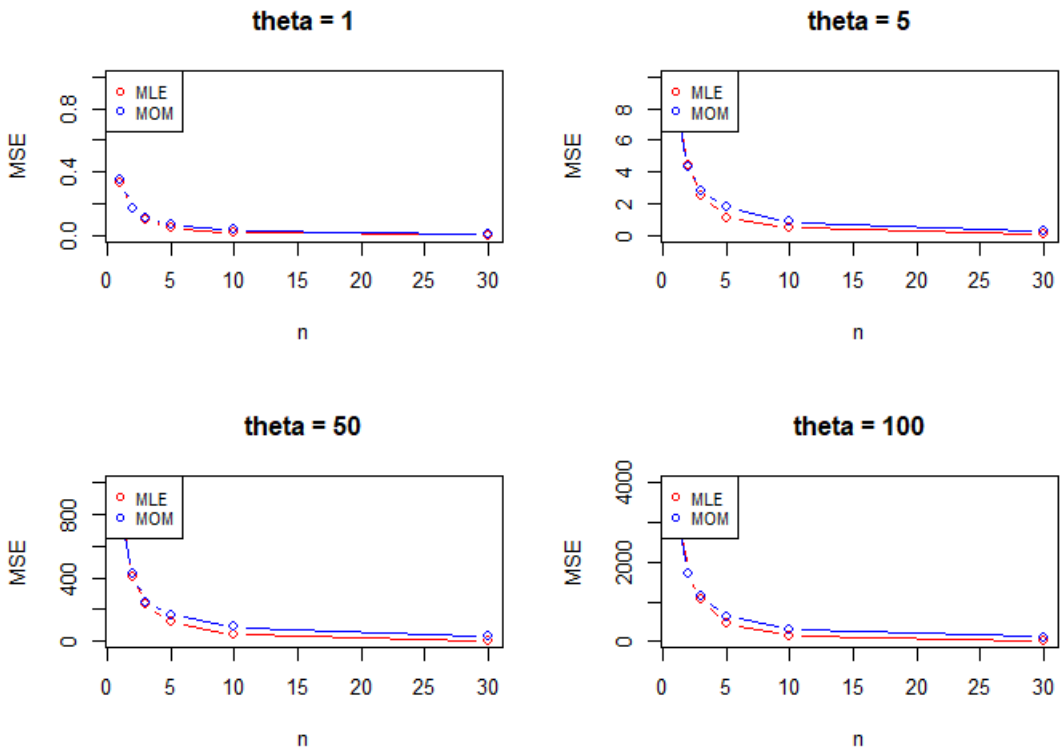
a) In this we have simulated the sample values by using population parameters and then we have used the simulated sample values to calculate the estimator values. Then we found the mean squared error by calculating the difference between the parameter and the estimator values and then we squared it. For Monte Carlo, we do multiple replications and take the mean of mean squared error.

b)
```
> MSE(1,1,1000)
[1] 0.3418188 0.3241167
```
In the above result, the left result is the MLE estimate and the right result is the MoM estimate.

c)



Plot 1.1 Mean Squared Errors of MLE and MoM based on n



Plot 1.2 Mean Squared Errors of MLE and MoM based on ϑ

d) In reference to the results obtained in part 1c. MLE estimator is better than MoM estimator because it gives a lower MSE in almost all combinations of n and ϑ. However, the estimator is more dependent on n because in the graph with fixed ϑ, the mean square error with MLE and MoM have similar results but in the graph with fixed n, as n increases, the MLE estimator performs much better than the MoM estimator.

Question 2.

Question 2

a) $f(x) = \begin{cases} \frac{\theta}{x^{\theta+1}} & x \geq 1 \\ 0 & x < 1 \end{cases}$

$L(\theta; x_1, \ldots, x_n) = \prod_{i=1}^{n} f_\theta(x_i) = \prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}$

$= \frac{\theta}{x_1^{\theta+1}} \cdot \frac{\theta}{x_2^{\theta+1}} \cdots \frac{\theta}{x_n^{\theta+1}} = \frac{\theta^n}{x_1^{\theta+1} \cdot x_2^{\theta+1} \cdots x_n^{\theta+1}}$

Taking Natural log.

$\ln[L(\theta)] = \ln\left[\frac{\theta^n}{x_1^{\theta+1} \cdot x_2^{\theta+1} \cdot x_n^{\theta+1}}\right]$

$= n\ln\theta - [(\theta+1)\ln x_1 + (\theta+1)\ln x_2 + \ldots + (\theta+1)\ln x_n]$

Taking derivative and equating to 0

$\frac{d}{d\theta} \ln[L(\theta)] = \frac{n}{\theta} - [\ln x_1 + \ln x_2 + \ldots + \ln x_n] = 0$

$\frac{n}{\theta} = \ln x_1 + \ln x_2 + \ln x_n$

$\hat{\theta}_{MLE} = \frac{n}{\ln x_1 + \ln x_2 + \ldots + \ln x_n}$

b) Plugging in the values into the equation.

$n = 5$, $x_1 = 21.72$, $x_2 = 14.65$, $x_3 = 50.42$, $x_4 = 28.78$, $x_5 = 11.23$

$\hat{\theta}_{MLE} = \frac{5}{\ln 21.72 + \ln 14.65 + \ln 50.42 + \ln 28.78 + \ln 11.23}$

$= \frac{5}{15.46} = 0.3236$

c)
```
> print(MLEstimator$par)
[1] 0.3233398
```
After numerically maximizing the log-likelihood function using optim function in R, we find out that the computed result above is very similar to the answer obtained in part 2b.

d) The approximate standard error of the maximum likelihood estimates and the approximate 95% confidence interval for ϑ are computed as follows:

```
> # Part 2d
> se = sqrt(1/MLEstimator$hessian)
> print(se)
             [,1]
[1,] 0.1446006
> upperB = MLEstimator$par + qnorm(0.975) * se
> lowerB = MLEstimator$par - qnorm(0.975) * se
> # Upper bound
> print(upperB)
             [,1]
[1,] 0.6067518
> # Lower bound
> print(lowerB)
              [,1]
[1,] 0.03992789
```

The mentioned approximates are going to be good as we know that out of the 100 trials to get the population estimated, the true estimate value will lie within the interval 95% of the times.

Section 2: R Code

```r
# MiniProject 3

# Question 1
# Function to return MLE and MoM of one sample based on n and theta
MLEMOM <- function(n, theta) {
  sample = runif(n, min = 0, max = theta)
  mle = max(sample)
  mom = 2*mean(sample)
  return (c(mle, mom))
}

MLEMOM(1, 1)

MSE <- function(n, theta, rep){
  samples = replicate(rep, MLEMOM(n, theta))
  estimates = (samples - theta)^2
  return (c(mean(estimates[1, ]), mean(estimates[2, ])))
}

# n = 1
MSE_1_1 = MSE(1,1,1000)
```

```
MSE_1_5 = MSE(1,5,1000)
MSE_1_50 = MSE(1,50,1000)
MSE_1_100 = MSE(1,100,1000)

# n = 2
MSE_2_1 = MSE(2,1,1000)
MSE_2_5 = MSE(2,5,1000)
MSE_2_50 = MSE(2,50,1000)
MSE_2_100 = MSE(2,100,1000)

# n = 3
MSE_3_1 = MSE(3,1,1000)
MSE_3_5 = MSE(3,5,1000)
MSE_3_50 = MSE(3,50,1000)
MSE_3_100 = MSE(3,100,1000)

# n = 5
MSE_5_1 = MSE(5,1,1000)
MSE_5_5 = MSE(5,5,1000)
MSE_5_50 = MSE(5,50,1000)
MSE_5_100 = MSE(5,100,1000)

# n = 10
MSE_10_1 = MSE(10,1,1000)
MSE_10_5 = MSE(10,5,1000)
MSE_10_50 = MSE(10,50,1000)
MSE_10_100 = MSE(10,100,1000)

# n = 30
MSE_30_1 = MSE(30,1,1000)
MSE_30_5 = MSE(30,5,1000)
MSE_30_50 = MSE(30,50,1000)
MSE_30_100 = MSE(30,100,1000)

# Align multiple graph on the same page
par(mfrow=c(3,2))

# Graphs with varying n value and fixed theta value
# Plot n = 1
plot(c(1,5,50,100), c(MSE_1_1[1], MSE_1_5[1], MSE_1_50[1], MSE_1_100[1]), type
= "b",
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 1", ylim =
c(0,4000))
lines(c(1,5,50,100), c(MSE_1_1[2], MSE_1_5[2], MSE_1_50[2], MSE_1_100[2]),
type = "b", col
      = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))
```

```
# Plot n = 2
plot(c(1,5,50,100), c(MSE_2_1[1], MSE_2_5[1], MSE_2_50[1], MSE_2_100[1]), type
= "b",
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 2", ylim =
c(0,2000))
lines(c(1,5,50,100), c(MSE_2_1[2], MSE_2_5[2], MSE_2_50[2], MSE_2_100[2]),
type = "b", col
      = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot n = 3
plot(c(1,5,50,100), c(MSE_3_1[1], MSE_3_5[1], MSE_3_50[1], MSE_3_100[1]), type
= "b",
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 3", ylim =
c(0,2000))
lines(c(1,5,50,100), c(MSE_3_1[2], MSE_3_5[2], MSE_3_50[2], MSE_3_100[2]),
type = "b", col
      = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot n = 5
plot(c(1,5,50,100), c(MSE_5_1[1], MSE_5_5[1], MSE_5_50[1], MSE_5_100[1]), type
= "b",
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 5", ylim =
c(0,1000))
lines(c(1,5,50,100), c(MSE_5_1[2], MSE_5_5[2], MSE_5_50[2], MSE_5_100[2]),
type = "b", col
      = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot n = 10
plot(c(1,5,50,100), c(MSE_10_1[1], MSE_10_5[1], MSE_10_50[1], MSE_10_100[1]),
type = "b",
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 10", ylim =
c(0,500))
lines(c(1,5,50,100), c(MSE_10_1[2], MSE_10_5[2], MSE_10_50[2], MSE_10_100[2]),
type =
        "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot n = 30
plot(c(1,5,50,100), c(MSE_30_1[1], MSE_30_5[1], MSE_30_50[1], MSE_30_100[1]),
type = "b",
```

```
     xlab = "theta", ylab = "MSE", col = 'red', main = "n = 30", ylim =
c(0,200))
lines(c(1,5,50,100), c(MSE_30_1[2], MSE_30_5[2], MSE_30_50[2], MSE_30_100[2]),
type =
        "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Align multiple graph on the same page
par(mfrow=c(2,2))

# Graphs with varying theta values and fixed n values.
# Plot theta = 1
plot(c(1,2,3,5,10,30), c(MSE_1_1[1], MSE_2_1[1], MSE_3_1[1], MSE_5_1[1],
MSE_10_1[1],
                         MSE_30_1[1]),
     type = "b", xlab = "n", ylab = "MSE", col = 'red', main = "theta = 1",
ylim = c(0,1))
lines(c(1,2,3,5,10,30), c(MSE_1_1[2], MSE_2_1[2], MSE_3_1[2], MSE_5_1[2],
MSE_10_1[2],
                          MSE_30_1[2]),
      type = "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot theta = 5
plot(c(1,2,3,5,10,30), c(MSE_1_5[1], MSE_2_5[1], MSE_3_5[1], MSE_5_5[1],
MSE_10_5[1],
                         MSE_30_5[1]),
     type = "b", xlab = "n", ylab = "MSE", col = 'red', main = "theta = 5",
ylim = c(0,10))
lines(c(1,2,3,5,10,30), c(MSE_1_5[2], MSE_2_5[2], MSE_3_5[2], MSE_5_5[2],
MSE_10_5[2],
                          MSE_30_5[2]),
      type = "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# Plot theta = 50
plot(c(1,2,3,5,10,30), c(MSE_1_50[1], MSE_2_50[1], MSE_3_50[1], MSE_5_50[1],
                         MSE_10_50[1], MSE_30_50[1]),
     type = "b", xlab = "n", ylab = "MSE", col = 'red', main = "theta = 50",
ylim = c(0,1000))
lines(c(1,2,3,5,10,30), c(MSE_1_50[2], MSE_2_50[2], MSE_3_50[2], MSE_5_50[2],
                          MSE_10_50[2], MSE_30_50[2]),
      type = "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))
```

```r
# Plot theta = 100
plot(c(1,2,3,5,10,30), c(MSE_1_100[1], MSE_2_100[1], MSE_3_100[1],
MSE_5_100[1],
                         MSE_10_100[1], MSE_30_100[1]),
     type = "b", xlab = "n", ylab = "MSE", col = 'red', main = "theta = 100",
ylim = c(0,4000))
lines(c(1,2,3,5,10,30), c(MSE_1_100[2], MSE_2_100[2], MSE_3_100[2],
MSE_5_100[2],
                          MSE_10_100[2], MSE_30_100[2]),
      type = "b", col = 'blue')
legend("topleft", legend = c("MLE", "MOM"), col = c('red', 'blue'), cex=.8,
pch=c(1,1))

# R code for question 2
# Part 2c
# Code for minimizing the function.
lkhd_function <- function(theta,x) {
  return (theta/((x)^theta))
}

neg_loglik_fun <- function(par, data) {
  result <- sum(log(lkhd_function(par,data)))
  return (-result)
}

MLEstimator <- optim(par=0.5, fn=neg_loglik_fun, method = "Nelder-Mead",
                 hessian= TRUE, data= c(21.72,14.65,50.42,28.78,11.23))
print(MLEstimator$par)

# Part 2d
# R code for finding the confidence interval and the standard error
se = sqrt(1/MLEstimator$hessian)
print(se)
upperB = MLEstimator$par + qnorm(0.975) * se
lowerB = MLEstimator$par - qnorm(0.975) * se

# Upper bound
print(upperB)

# Lower bound
print(lowerB)
```