Mini Project # 3

Qingyu Lan, Lakshmi Priyanka Selvaraj

Contribution of each group member :

Both members worked on the questions together.

Section 1. Answers to the specific questions asked

1. Question 1
   a. Simulating sample values by using population parameters. Use simulated sample values to calculate estimator value. Then, find the mean squared error by doing difference between the parameter and the estimator value, then square it. For monte carlos we do multiple replications and take the mean of mean squared error.
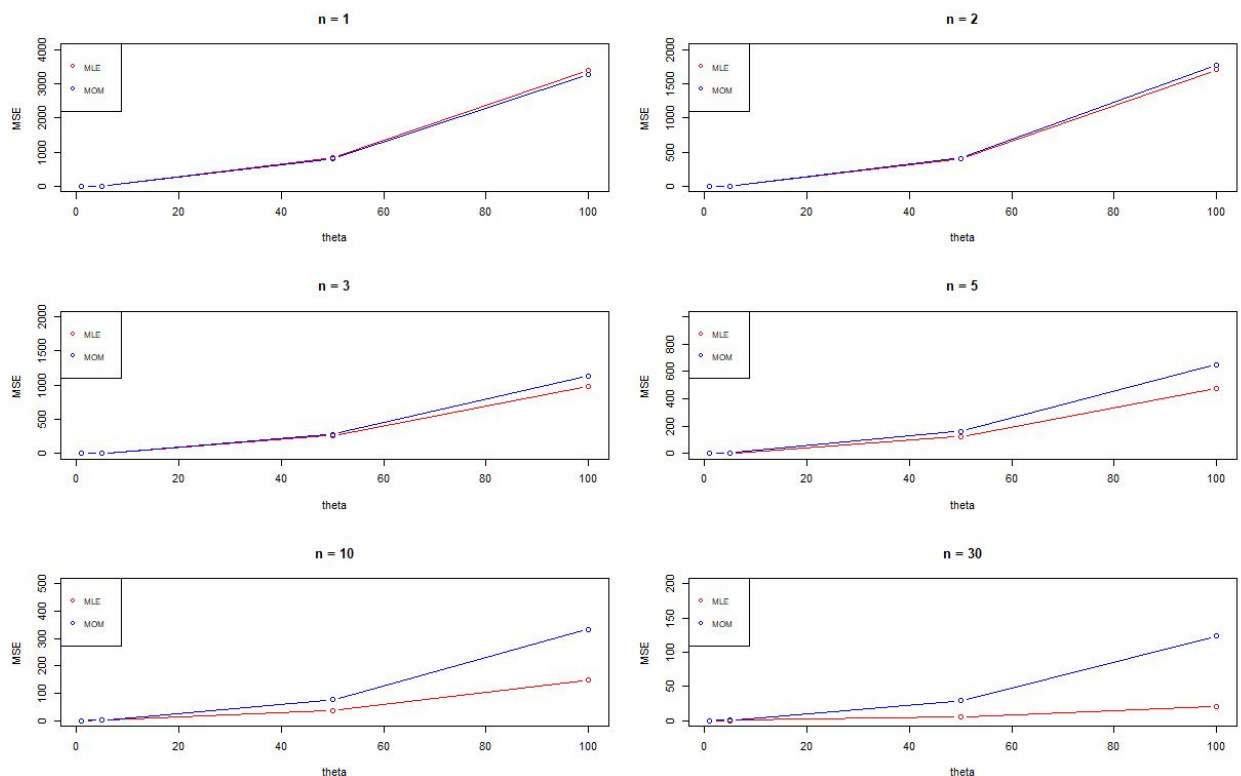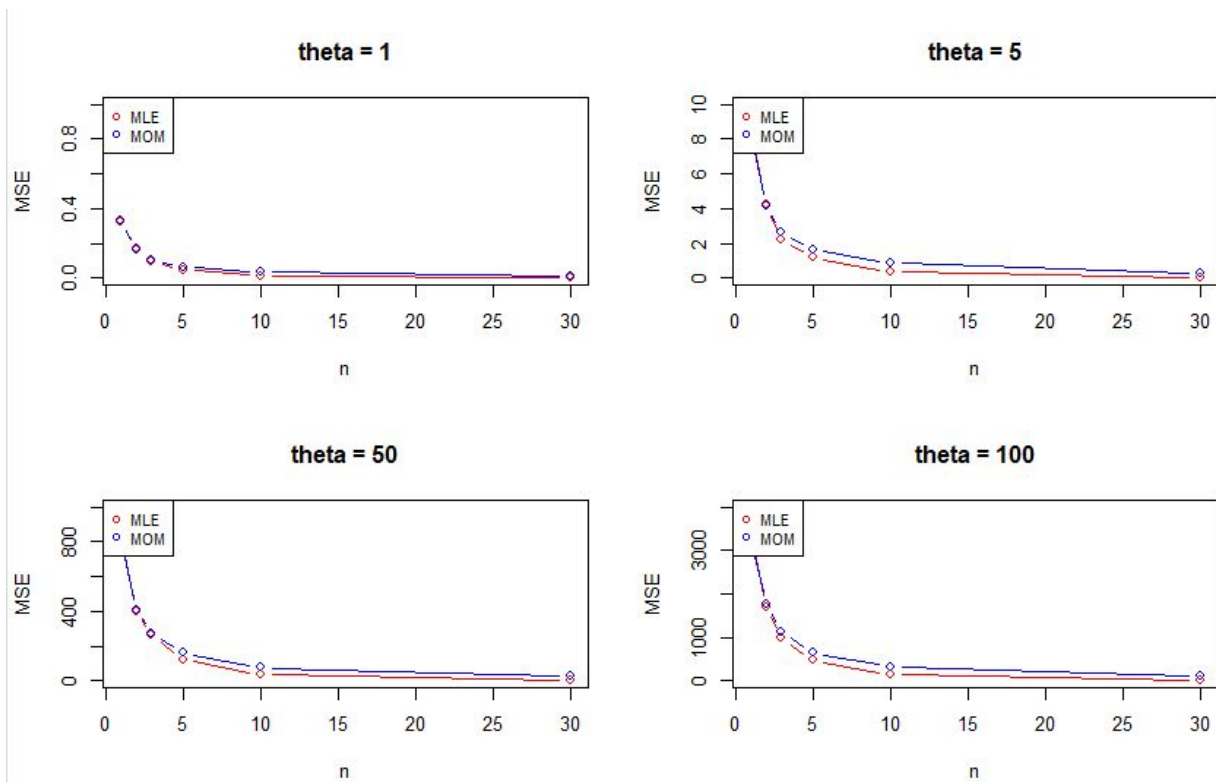
   ```
   > MSE(1,1,1000)
   [1] 0.3297253 0.3202060
   ```
   b.   left estimate with MLE, right estimate with MOM
   c.

Graphs based on n

Graphs based on theta

theta = 1

theta = 5

theta = 50

theta = 100

d. Based on C, MLE estimator is better because it provides a lower MSE in almost every combination of theta and n as seen in the graphs given in C. The answer depends on n because in the graph with fix theta, MSE with MLE and MOM both have very similar results, however, in the graph with fix n, as n increases, you can see that MLE starts to perform better by a significant margin than MOM.

2. Question 2

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

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

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

Taking Natural log:

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

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

Taking derivative & equating to 0.

$$\frac{d}{d\theta} \ln[L(\theta)] = \frac{n}{\theta} - \left[\ln x_1 + \ln x_2 + \cdots + \ln x_n\right] \overset{\Delta}{=} 0$$

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

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

b) $n = 5, \; x_1 = 21.72, \; x_2 = 14.65, \; x_3 = 50.42, \; x_4 = 28.78,$

$\quad x_5 = 11.23$

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

$$= \frac{5}{3.07 + 2.68 + 3.92 + 3.35 + 2.41}$$

$$= \frac{5}{15.46} = 0.323$$

c) By using the R code, in section 2, We can actually observe that the parameter value turns out to be 0.323 which is similar to the value computed above.

```
> se = sqrt(1/ml.est$hessian)
> print(se)
          [,1]
[1,] 0.1446006
>
> upperB = ml.est$par + qnorm(0.975) * se
> lowerB = ml.est$par - qnorm(0.975) * se
>
> #upper bound
> print(upperB)
          [,1]
[1,] 0.6067518
> #lower bound
> print(lowerB)
          [,1]
[1,] 0.03992789
d) > |
```

Section 2: R code.

R code for question 1

```
# function to return MEL est MOM est 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)
```

```r
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)

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

# 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
```

```r
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))

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

# 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]),
```

```r
        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))


# 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

```r
lkhd_function <- function(theta,x)
{
  return (theta/((x)^theta))
}


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

```r
ml.est <- 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(ml.est$par)

# part d

se = sqrt(1/ml.est$hessian)
print(se)

upperB = ml.est$par + qnorm(0.975) * se
lowerB = ml.est$par - qnorm(0.975) * se

#upper bound
print(upperB)
#lower bound
print(lowerB)
```