# Mini Project 3

Name: Tejas Ravi Rao (txr171830)

## Section – 1

Q1(a) Mean Squared Error of an estimator can be calculated using Monte Carlo Simulation. Firstly, we are required to draw random samples from a Uniform distribution. The Maximum Likelihood Estimator is estimated by finding the maximum of the sample and the Method of Moments Estimator is estimated as twice the mean of the sample. Next, we calculate the square of the difference between the actual value and estimator. This step is repeated many times and each time the square of difference is calculated. Finally, mean of all these squared error values is calculated.

Q1(b) Firstly, save the vector of given values. Also, save the vector of given theta values.

```
# given values of n
n = c(1,2,3,5,10,30)

# given values of theta
theta = c(1,5,50,100)
```

Next step is to obtain the values of Maximum Likelihood Estimator and Method of Moments Estimator through defined function. This function takes value of n and theta as input parameters and returns both MLE thetaOne and MOME thetaTwo.

```
# Question 1(b)
# function to obtain Maximum Likelihood Estimator
# and Method of Moments estimator

MC_simulation = function(n, theta){

  result = runif(n, min = 0, max = theta)

  # obtain thetaOne
  thetaOne = max(result)

  # obtain thetaTwo
  thetaTwo = 2 * (mean(result))
```

```
  return (c(thetaOne,thetaTwo))
}
```

Replicate the above method N = 1000 times. Return the mean squared error value.

```
# replicate the simulation 1000 times
MSE_method = function(n,theta){

  thetaEstm = replicate(1000, MC_simulation(n, theta))

  # compute MSE value which is for MLE and MOME
  # for n, theta
  return (rowMeans((thetaEstm - theta) ^ 2))

}
```

Save the lengths of n and theta vectors which are required in the next step.

```
# get the length of n and theta
nLen = length(n)
thetaLen = length(theta)
```

Q1(c) The above (b) step is repeated for all combinations of (n,theta). The MSE values for MLE and MOME are saved in their respective matrices. The rows represent values of n and columns represent values of theta.

```
#Question 1(c)
# create two matrices of dimensions (n x theta)
# these two matrices will store MSE of MLE and MSE of MOME in
# the respective matrices for every combination of n, theta

MSE_MLE = matrix(nrow = nLen, ncol = thetaLen)
MSE_MOME  = matrix(nrow = nLen, ncol = thetaLen)

for(i in 1:nLen){
  for(j in 1:thetaLen){
    finalResult = MSE_method(n[i], theta[j])
    MSE_MLE[i,j] = finalResult[1]
    MSE_MOME[i,j] = finalResult[2]
  }
}
```

**# display the final matrices of MLE and MOME**
**MSE_MLE**
**MSE_MOME**

The following output is obtained for MSE_MLE matrix. (n =1,2,3,5,10,30) (Theta=1,5,50,100)

```
            [,1]       [,2]       [,3]      [,4]
[1,] 0.335293959 8.33231758 841.736755 3323.8538
[2,] 0.174497050 4.25703225 424.110737 1648.6370
[3,] 0.102068384 2.49339638 275.792427  931.1959
[4,] 0.047456769 1.27706736 121.858577  476.3442
[5,] 0.014926774 0.36155216  35.993828  139.2009
[6,] 0.001865776 0.05470539   4.690373   21.3883
```

The following output is obtained for MSE_MOME matrix. (n =1,2,3,5,10,30) (Theta=1,5,50,100)

```
           [,1]      [,2]      [,3]      [,4]
[1,] 0.33028704 8.1549199 852.98331 3454.8227
[2,] 0.17030154 4.1206676 395.96801 1709.9743
[3,] 0.10992293 2.8401453 275.75657 1181.7614
[4,] 0.06731996 1.6098978 160.40724  670.3819
[5,] 0.03572565 0.7970113  78.27285  305.4459
[6,] 0.01106951 0.2798123  27.75344  109.8311
```

Graph plots are obtained for MSE of MLE and MSE of MOME against n for different values of theta. The solid line depicts the behavior of MSE of MLE and the dotted line depicts behavior of MSE of MOME.

**# graph plots for MSE of MLE and MSE of MOME for each value of theta**

**# for theta = 1**
**# solid line for MSE of MLE**
**plot(n, MSE_MLE[,1],main = "theta = 1",ylab = "MSE", type = "l", lty = "solid")**

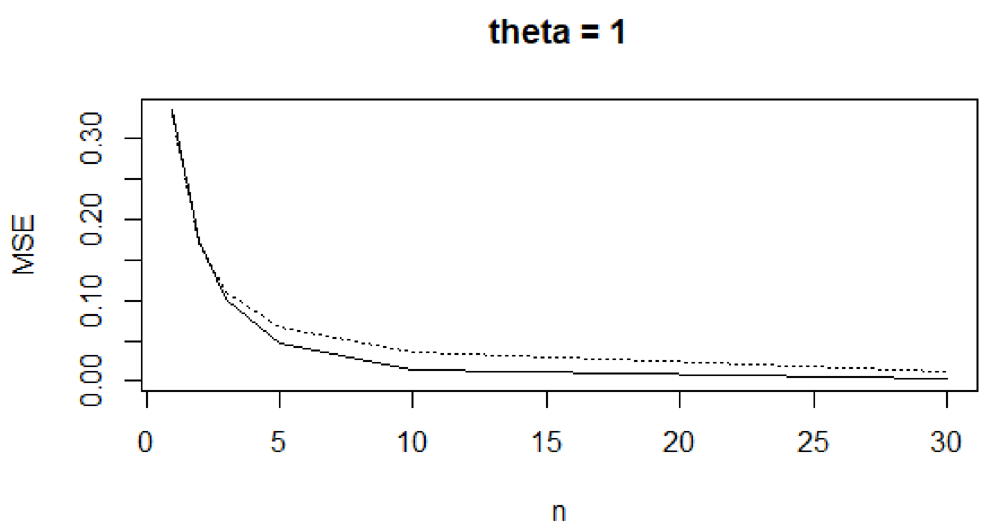
**# dotted line for MSE of MOME**
**lines(n,MSE_MOME[,1], lty = "dotted")**

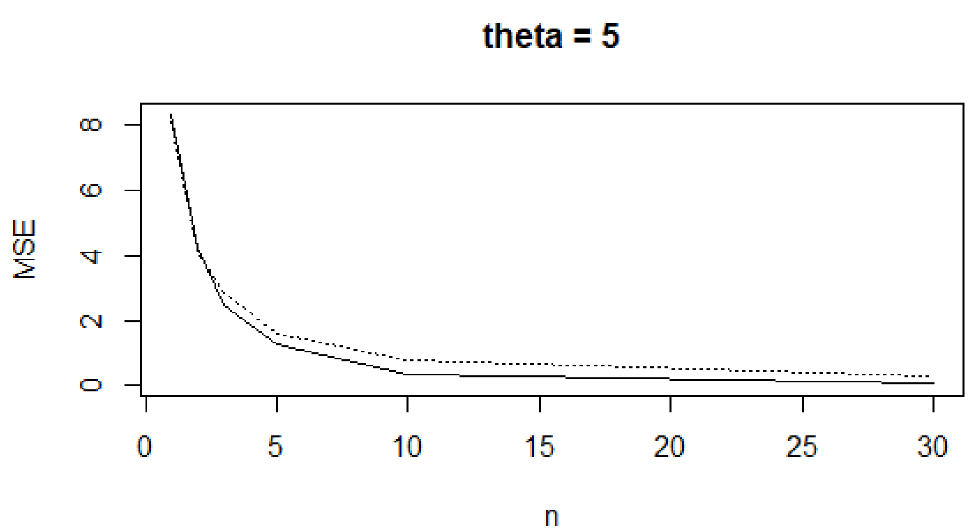
**# for theta = 5**
**# solid line for MSE of MLE**
**plot(n, MSE_MLE[,2],main = "theta = 5",ylab = "MSE", type = "l", lty = "solid")**

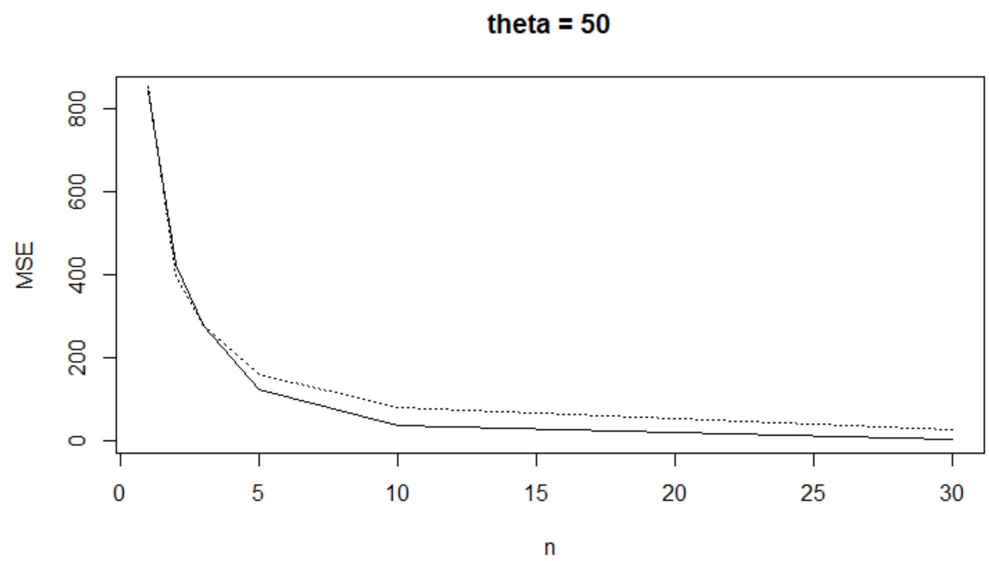
**# dotted line for MSE of MOME**
**lines(n,MSE_MOME[,2], lty = "dotted")**

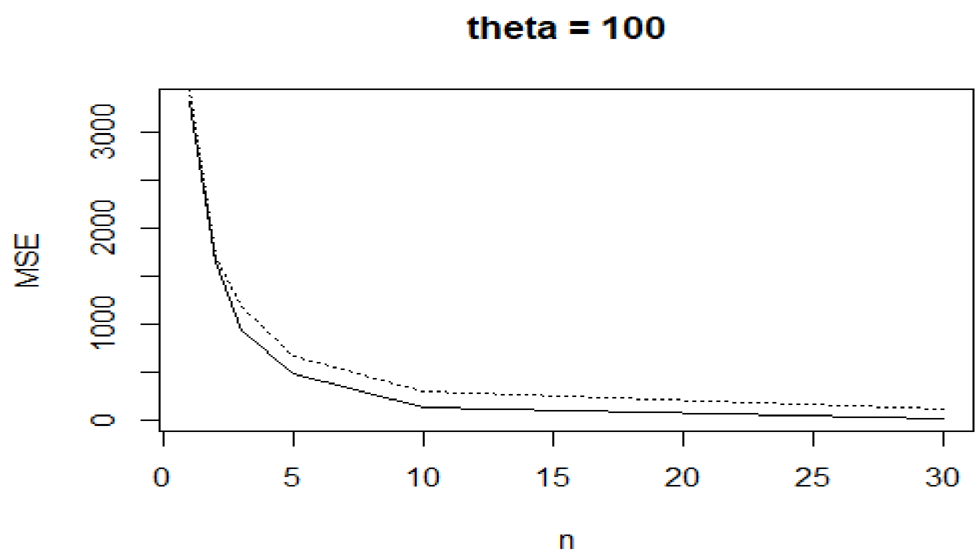
```
# for theta = 50
# solid line for MSE of MLE
plot(n, MSE_MLE[,3],main = "theta = 50",ylab = "MSE", type = "l", lty = "solid")

# dotted line for MSE of MOME
lines(n,MSE_MOME[,3], lty = "dotted")

# for theta = 100
# solid line for MSE of MLE
plot(n, MSE_MLE[,4],main = "theta = 100",ylab = "MSE", type = "l", lty = "solid")

# dotted line for MSE of MOME
lines(n,MSE_MOME[,4], lty = "dotted")
```

The following plots were obtained.






Q1(d) It can be observed from above plots that MSE of both estimators decreases as the value of n increases. The reason being, their accuracy increases as n increases. It can also be observed that as theta increases the MSE also increases. For smaller values of n, both MSE of MLE and MSE of MOME are almost equal. However, as value of n increases MSE of MLE is less when compared to MSE of MOME. We can infer that for n less than 5 MSE is almost equal for both estimators. For n greater than or equal to 5, MSE of MLE is less than MSE of MOME. Therefore, for a given value of n and theta overall, MLE is better than MOME.

Q2(a) Given pdf => $f(x) = \frac{\theta}{x^{\theta+1}}$

$$L(\theta) = \prod_{i=1}^{n} f_\theta(x_i)$$

$$L(\theta) \quad = \quad \prod_{i=1}^{n} \frac{\theta}{x^{\theta+1}}$$

Obtain log likelihood estimate

$$\text{Log}\{L(\theta)\} \quad = \quad \sum_{i=1}^{n} \{\log \frac{\theta}{x^{\theta+1}}\}$$

$$= \quad \sum_{i=1}^{n} \{\log \theta - (\Theta + 1) \log x_i\}$$

$$= \quad \text{n}(\log \theta) - (\Theta + 1)\ log \sum_{i=1}^{n} x_i$$

We can find the maximum likelihood by partially differentiating Log{L(θ)} with respect to θ and equating it to 0.

$$\frac{d}{d\Theta} \log\{L(\theta)\} \quad = \quad 0$$

$$\frac{d}{d\Theta} \text{n}(\log \theta) - (\Theta + 1)\ log \sum_{i=1}^{n} x_i \quad = \quad 0$$

$$\frac{n}{\theta} - \sum_{i=1}^{n} \log(x_i) \quad = \quad 0$$

$$\frac{n}{\theta} \quad = \quad \sum_{i=1}^{n} \log(x_i)$$

$$\theta \quad = \quad \frac{n}{\sum_{i=1}^{n} \log(x_i)}$$

Therefore, we can conclude that

$$\Theta_{MLE} \quad = \quad \frac{n}{\sum_{i=1}^{n} \log(x_i)}$$

Q2(b) Given n = 5,

Sample values are $x_1 = 21.72$, $x_2 = 14.65$, $x_3 = 50.42$, $x_4 = 28.78$, $x_5 = 11.23$

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

$$= \quad 0.32338741$$

Q2(c) Using the above sample values to estimate, by numerically maximizing the log-likelihood function.

```
# n=5, store sample values in vector
sampleVal = c(21.72,14.65,50.42,28.78,11.23)
```

Define PDF in the function

```
# Probability density function
# of the continuous random variable defined in the function
funcTheta = function(t,x){
  return (t/x^(t+1))
}
```

Using the above PDF, we define the Negative of Log-likelihood function. This can then be used in the ***optim*** function.

```
# Negative of log-likelihood function
neg.loglik.fun = function(par,dat){
  logSum = sum(log(funcTheta(par,dat)))
  return (-logSum)
}
```

***Optim*** by default performs minimization, and hence we minimize - log(L) to in turn obtain Maximized log(L).

```
# Minimize -log (L), i.e., maximize log (L)
# obtain parameter estimate
ml.est = optim(par=1, fn = neg.loglik.fun, method = "BFGS", hessian = TRUE, dat = sampleVal)
ml.est$par
```

[1] 0.323387

Therefore, the value obtained for the estimate is 0.323387. This is same as the one obtained in Q2(b).

Q2(d) The standard error of the Maximum Likelihood estimate can be calculated as follows

```
# Question 2(d)
# Standard error of the MLE
(result = sqrt(diag(solve(ml.est$hessian))))
```

[1] 0.1446217

The standard error obtained is 0.1446217

The 95% Confidence Interval for θ is obtained as follows.

```
# Calculate 95% Confidence Interval of estimate
ConfInter = ml.est$par + c(-1,1)*qnorm(0.975)*result
ConfInter

[1] 0.03993372    0.60684034
```

The lower and upper values of the Confidence Interval are 0.03993372 and 0.60684034 respectively. Therefore, Confidence Interval is [0.03993372 , 0.60684034].

The above obtained values of Standard Error and Confidence Interval cannot be good approximations. The reason being, the sample size is very small. Also, it cannot be concluded that θ has normal distribution. Therefore, we cannot conclude that obtained Confidence Interval values may be correct.

## Section – 2
## R - Code

Q1)

```
# Question 1
# given values of n
n = c(1,2,3,5,10,30)

# given values of theta
theta = c(1,5,50,100)

set.seed(123)

# Question 1(b)
# function to obtain Maximum Likelihood Estimator
# and Method of Moments estimator

MC_simulation = function(n, theta){

  result = runif(n, min = 0, max = theta)

  # obtain thetaOne
  thetaOne = max(result)

  # obtain thetaTwo
  thetaTwo = 2 * (mean(result))
```

```
  return (c(thetaOne,thetaTwo))
}

# replicate the simulation 1000 times
MSE_method = function(n,theta){

  thetaEstm = replicate(1000, MC_simulation(n, theta))

  # compute MSE value which is for MLE and MOME
  # for n, theta
  return (rowMeans((thetaEstm - theta) ^ 2))

}

# get the length of n and theta
nLen = length(n)
thetaLen = length(theta)

# Question 1(c)
# create two matrices of dimensions (n x theta)
# these two matrices will store MSE of MLE and MSE of MOME in
# the respective matrices for every combination of n, theta

MSE_MLE = matrix(nrow = nLen, ncol = thetaLen)
MSE_MOME  = matrix(nrow = nLen, ncol = thetaLen)

for(i in 1:nLen){
  for(j in 1:thetaLen){
    finalResult = MSE_method(n[i], theta[j])
    MSE_MLE[i,j] = finalResult[1]
    MSE_MOME[i,j] = finalResult[2]
  }
}

# display the final matrices of MLE and MOME
MSE_MLE
MSE_MOME

# graph plots for MSE of MLE and MSE of MOME for each value of theta
# for theta = 1
# solid line for MSE of MLE
plot(n, MSE_MLE[,1],main = "theta = 1",ylab = "MSE", type = "l", lty = "solid")
```

```
# dotted line for MSE of MOME
lines(n,MSE_MOME[,1], lty = "dotted")

# for theta = 5
# solid line for MSE of MLE
plot(n, MSE_MLE[,2],main = "theta = 5",ylab = "MSE", type = "l", lty = "solid")

# dotted line for MSE of MOME
lines(n,MSE_MOME[,2], lty = "dotted")

# for theta = 50
# solid line for MSE of MLE
plot(n, MSE_MLE[,3],main = "theta = 50",ylab = "MSE", type = "l", lty = "solid")

# dotted line for MSE of MOME
lines(n,MSE_MOME[,3], lty = "dotted")

# for theta = 100
# solid line for MSE of MLE
plot(n, MSE_MLE[,4],main = "theta = 100",ylab = "MSE", type = "l", lty = "solid")

# dotted line for MSE of MOME
lines(n,MSE_MOME[,4], lty = "dotted")
```

Q2)

```
# Question 2(c)

# n=5, store sample values in vector
sampleVal = c(21.72,14.65,50.42,28.78,11.23)

# Probability density function
# of the continuous random variable defined in the function
funcTheta = function(t,x){
  return (t/x^(t+1))
}

# Negative of log-likelihood function
neg.loglik.fun = function(par,dat){
  logSum = sum(log(funcTheta(par,dat)))
  return (-logSum)
}
```

```
# Minimize - log (L), i.e., maximize log (L)
# obtain parameter estimate
ml.est = optim(par=1, fn = neg.loglik.fun, method = "BFGS", hessian = TRUE, dat = sampleVal)
ml.est$par

# Question 2(d)
# Standard error of the MLE
(result = sqrt(diag(solve(ml.est$hessian))))

# Calculate 95% Confidence Interval of estimate
ConfInter = ml.est$par + c(-1,1)*qnorm(0.975)*result
ConfInter
```