**Statistical Methods for Data Science**
**Mini Project #3**

**Group Members**: *Arpita Kumane, Harshali Dube*

**Contribution**: Both the team members collaborated to work on the project equally. We discussed with each other the approaches to come up with the solutions

## Question 1

### 1a. Explain how you will compute the mean squared error of an estimator using Monte Carlo simulation.

**Ans.**

We will compute MSE of an estimator by generating n uniformly distributed random samples with range [0, θ] and then we can compute Maximum Likelihood Estimator (max sample) and Method of Moments Estimator (2 * sample mean). After obtaining θ̂ from MLE and MME we calculate squared difference between estimated θ̂ and actual parameter θ i.e. (_θ̂−θ )^2. We repeat these steps large number of times and record squared errors. Finally, we take mean of all squared errors recorded that gives us MSE.

### 1b. For a given combination of (n, θ), compute the mean squared errors of both θ1 and θ2 using Monte Carlo simulation with N = 1000 replications. Be sure to compute both estimates from the same data.

**Ans.**

Let us consider n = 1 and θ = 1

```
> set.seed(1)
> n_rep = 1000
> n = 1
> theta = 1
> mle_vector = numeric()
> mme_vector = numeric()
> for (i in 1 : n_rep){  z = runif(n,0,theta)
+ mle_vector = append(mle_vector,(max(z)-theta)**2)
+ mme_vector = append(mme_vector, (2*mean(z)-theta)**2)}
> mean(mle_vector)
[1] 0.3333923
> mean(mme_vector)
[1] 0.332336
>
```

θ 1 = 0.3333923 and θ 2 = 0.332336

**1c. Repeat (b) for the remaining combination of (n, θ). Summarize your results graphically.**
**Ans:**

```
> set.seed(1)
> n_rep = 1000
> n_vector = c(1,2,3,5,10,30)
> theta_vector = c(1,5,50,100)
> mle_mean = numeric()
> mme_mean = numeric()
> mse = function(n,theta,n_rep){
+     mle_vector = numeric()
+     mme_vector = numeric()
+     for(k in 1: n_rep){
+         z = runif(n,0,theta)
+         mle_vector = c(mle_vector, (max(z)-theta)**2)
+         mme_vector = c(mme_vector, (2*mean(z)-theta)**2)
+     }
+     cat(sprintf("n = %d \t theta = %d \t MLE = %f \t MME = %f \n",
+                 n, theta,mean(mle_vector),mean(mme_vector)))
+ }
> #MSE fucntion for various values of n ans theta
> for(i in 1:length(n_vector)){
+     for(j in 1:length(theta_vector)){
+         mse(n_vector[i],theta_vector[j],n_rep)
+     }
+ }
```

```
n = 1    theta = 1      MLE = 0.333392          MME = 0.332336
n = 1    theta = 5      MLE = 8.688112          MME = 8.782065
n = 1    theta = 50     MLE = 862.796359        MME = 848.157016
n = 1    theta = 100    MLE = 3421.668982       MME = 3628.807995
n = 2    theta = 1      MLE = 0.157267          MME = 0.162851
n = 2    theta = 5      MLE = 4.138238          MME = 4.219905
n = 2    theta = 50     MLE = 417.346383        MME = 409.833477
n = 2    theta = 100    MLE = 1615.113650       MME = 1629.510551
n = 3    theta = 1      MLE = 0.102622          MME = 0.112104
n = 3    theta = 5      MLE = 2.651796          MME = 2.777197
n = 3    theta = 50     MLE = 242.566840        MME = 273.426158
n = 3    theta = 100    MLE = 977.098805        MME = 1075.178685
n = 5    theta = 1      MLE = 0.049822          MME = 0.064475
n = 5    theta = 5      MLE = 1.180158          MME = 1.660749
n = 5    theta = 50     MLE = 117.997355        MME = 155.353895
n = 5    theta = 100    MLE = 443.174989        MME = 638.188186
n = 10   theta = 1      MLE = 0.014816          MME = 0.033988
n = 10   theta = 5      MLE = 0.347641          MME = 0.869284
n = 10   theta = 50     MLE = 37.286499         MME = 87.322392
n = 10   theta = 100    MLE = 151.473447        MME = 326.181707
n = 30   theta = 1      MLE = 0.002134          MME = 0.011031
n = 30   theta = 5      MLE = 0.047096          MME = 0.267574
n = 30   theta = 50     MLE = 5.292310          MME = 26.723381
n = 30   theta = 100    MLE = 20.579199         MME = 110.844742
```

- par () func@on is used to change the layout of the plot output. Hence, the sub-sequent four graphs are placed in matrix of size 3x2.

  Graph 1: n fixed variable vs theta

```
> #MSE for all combinations
>
> mse1_1 = mse(1,1)
> mse1_5 = mse(1,5)
> mse1_50 = mse(1,50)
> mse1_100 = mse(1,100)
> mse2_1 = mse(2,1)
> mse2_5 = mse(2,5)
> mse2_50 = mse(2,50)
> mse2_100 = mse(2,100)
> mse3_1 = mse(3,1)
> mse3_5 = mse(3,5)
> mse3_50 = mse(3,50)
> mse3_100 = mse(3,100)
> mse5_1 = mse(5,1)
> mse5_5 = mse(5,5)
> mse5_50 = mse(5,50)
> mse5_100 = mse(5,100)
> mse10_1 = mse(10,1)
> mse10_5 = mse(10,5)
> mse10_50 = mse(10,50)
> mse10_100 = mse(10,100)
>
>
> mse50_1 = mse(50,1)
> mse50_5 = mse(50,5)
> mse50_50 = mse(50,50)
> mse50_100 = mse(50,100)
```

```
> plot(c(1,5,50,100),c(mse1_1[1],mse1_5[1],mse1_50[1],mse1_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=1")
> lines(c(1,5,50,100),c(mse1_1[2],mse1_5[2],mse1_50[2],mse1_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
>
```

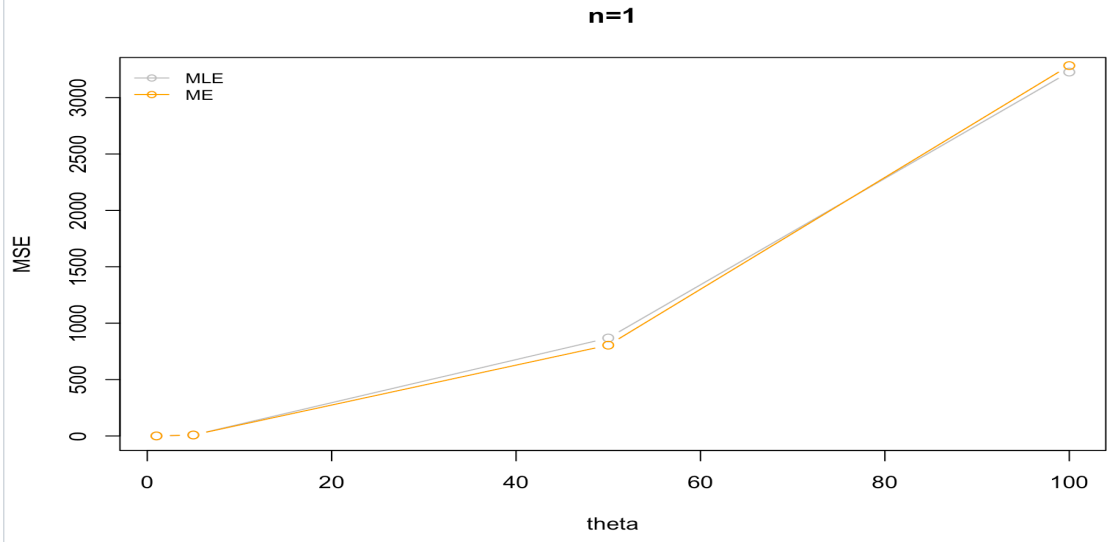
```
> plot(c(1,5,50,100),c(mse2_1[1],mse2_5[1],mse2_50[1],mse2_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=2")
> lines(c(1,5,50,100),c(mse2_1[2],mse2_5[2],mse2_50[2],mse2_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
>
```

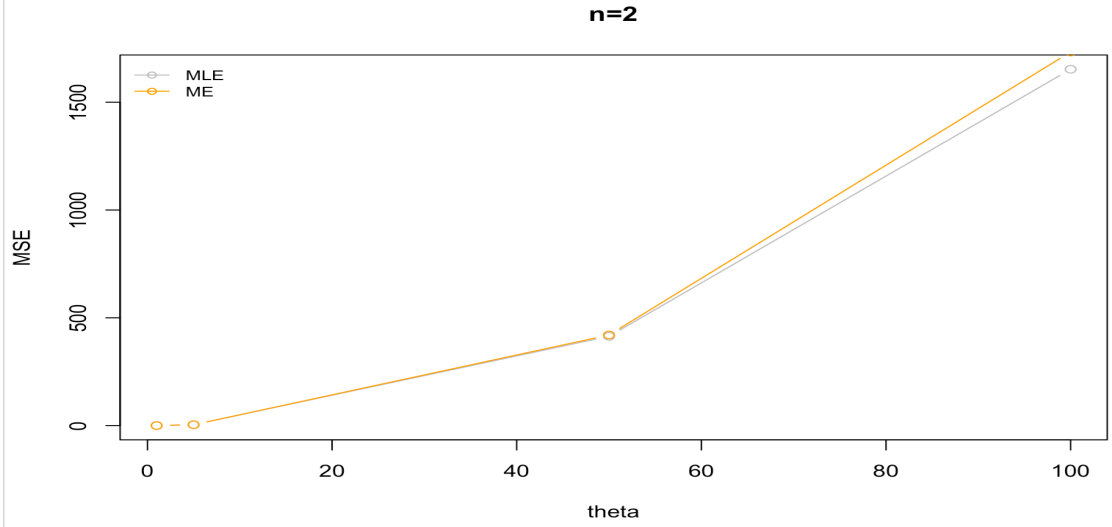
```
> plot(c(1,5,50,100),c(mse3_1[1],mse3_5[1],mse3_50[1],mse3_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=3")
> lines(c(1,5,50,100),c(mse3_1[2],mse3_5[2],mse3_50[2],mse3_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
>
```

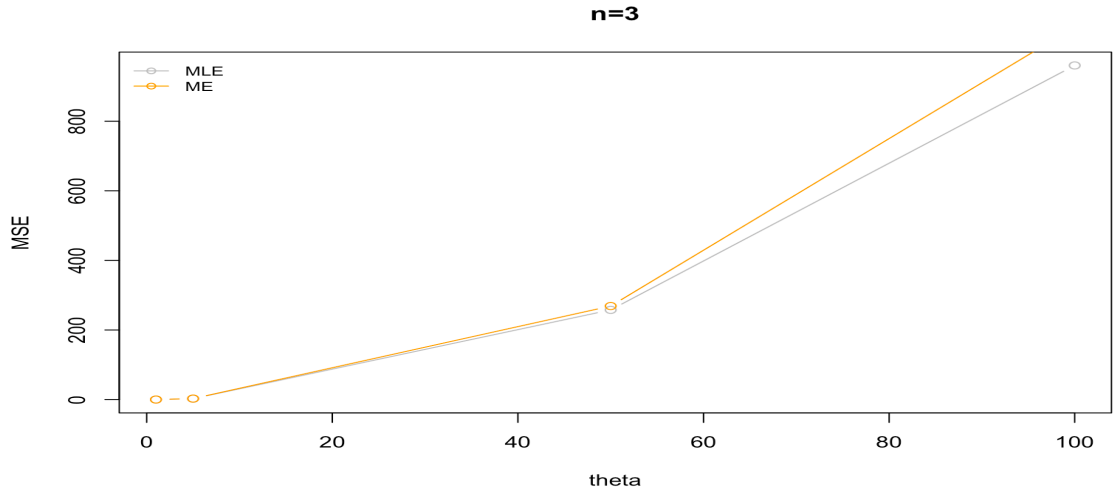
```
> plot(c(1,5,50,100),c(mse5_1[1],mse5_5[1],mse5_50[1],mse5_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=5")
> lines(c(1,5,50,100),c(mse5_1[2],mse5_5[2],mse5_50[2],mse5_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
>
```

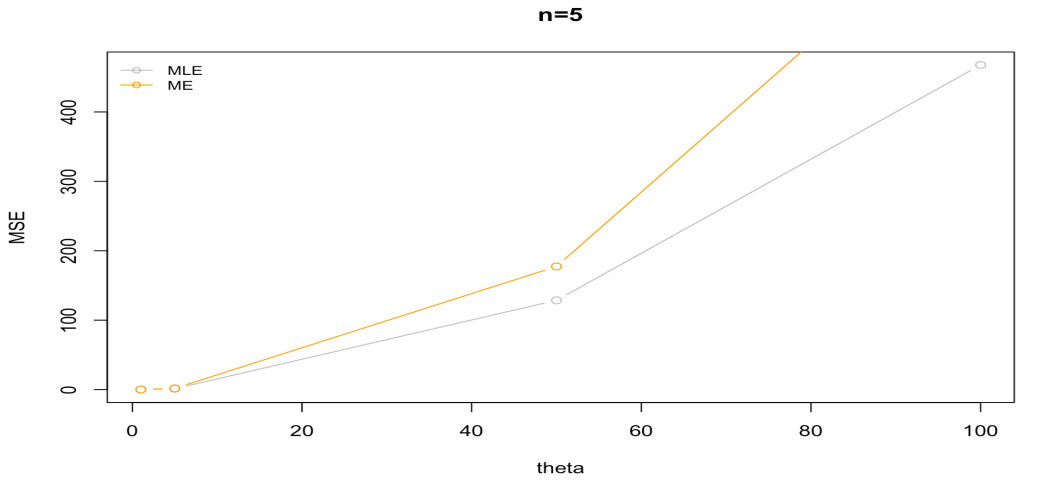
```
> plot(c(1,5,50,100),c(mse10_1[1],mse10_5[1],mse10_50[1],mse10_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=10")
> lines(c(1,5,50,100),c(mse10_1[2],mse10_5[2],mse10_50[2],mse10_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
```

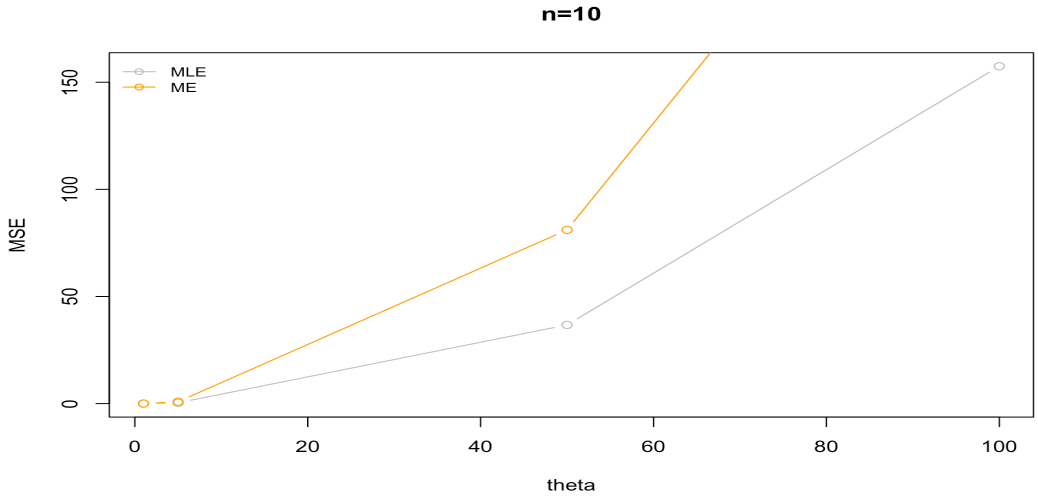
```
> plot(c(1,5,50,100),c(mse50_1[1],mse50_5[1],mse50_50[1],mse50_100[1]),
+      type="b",xlab='theta',ylab='MSE',col='grey',main="n=50")
> lines(c(1,5,50,100),c(mse50_1[2],mse50_5[2],mse50_50[2],mse50_100[2]),
+       type="b",col='orange')
> legend("topleft",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,
+        ncol=1,cex=0.8,bty='n')
>
```

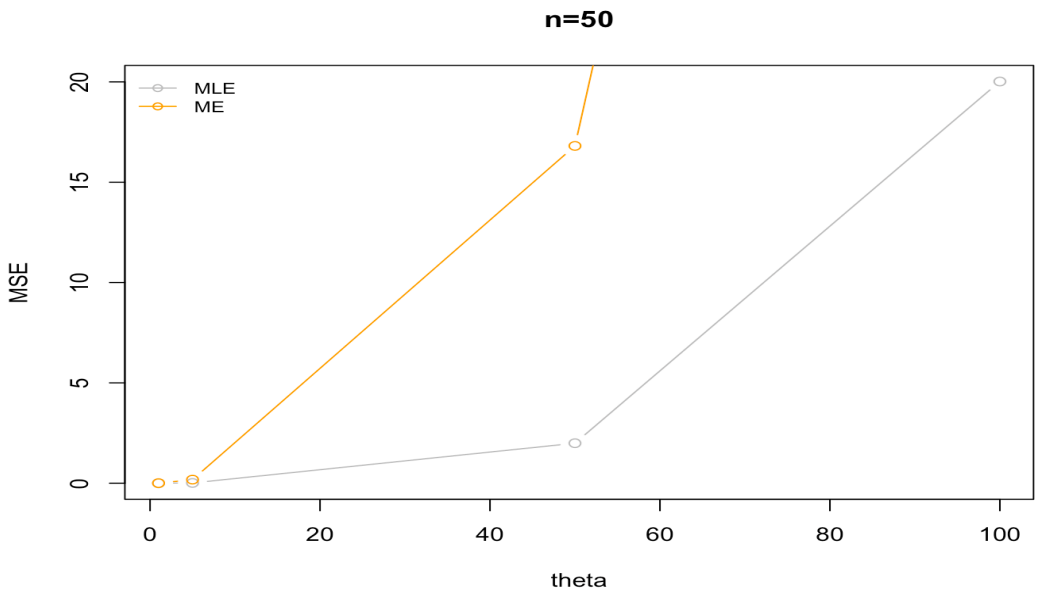

n=1
MLE
ME
MSE
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100
theta


n=2
MLE
ME
MSE
0
500
1000
1500
0
20
40
60
80
100
theta


n=3
MLE
ME
MSE
0
200
400
600
800
0
20
40
60
80
100
theta


n=5
MLE
ME
MSE
0
100
200
300
400
0
20
40
60
80
100
theta


n=10
MLE
ME
MSE
0
50
100
150
0
20
40
60
80
100
theta

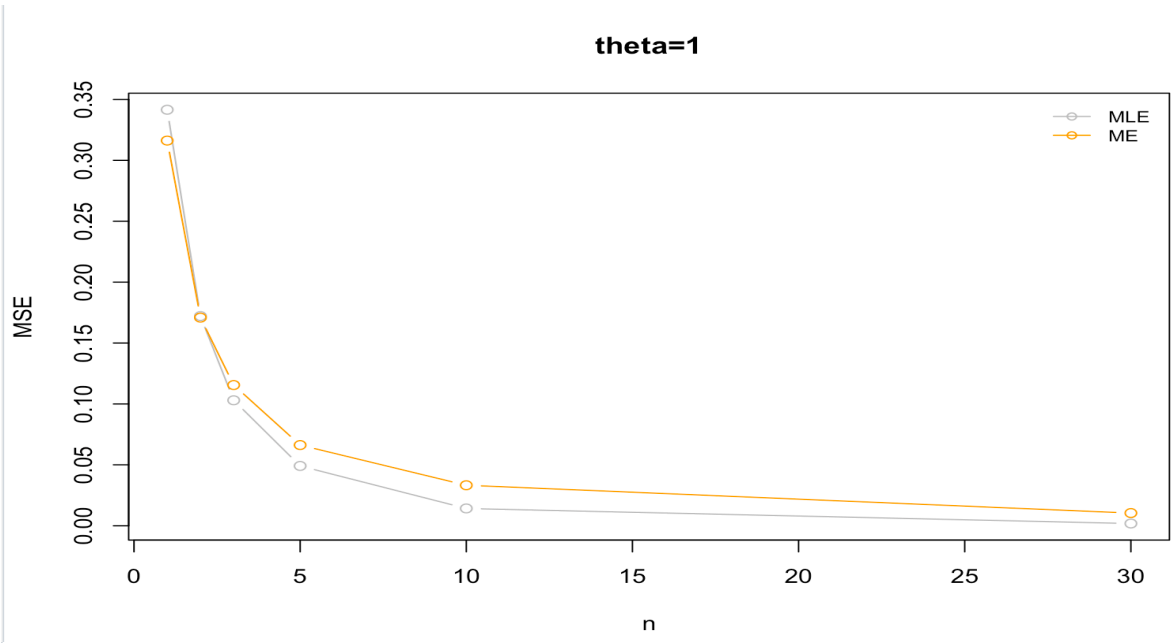


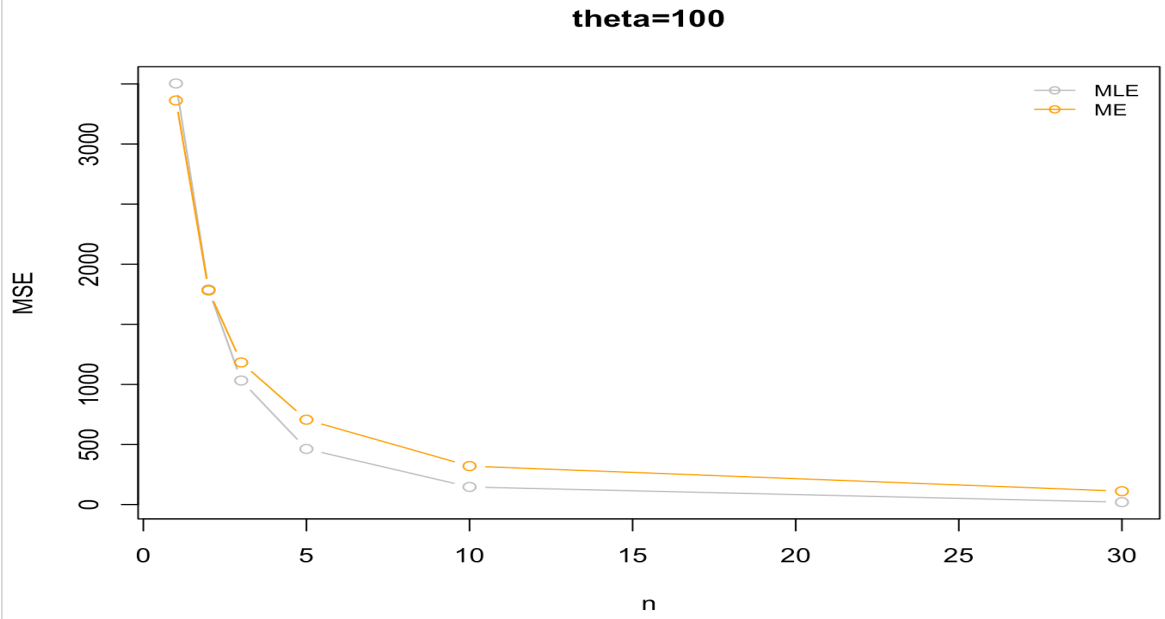

Graph 2: theta fixed variable vs n,

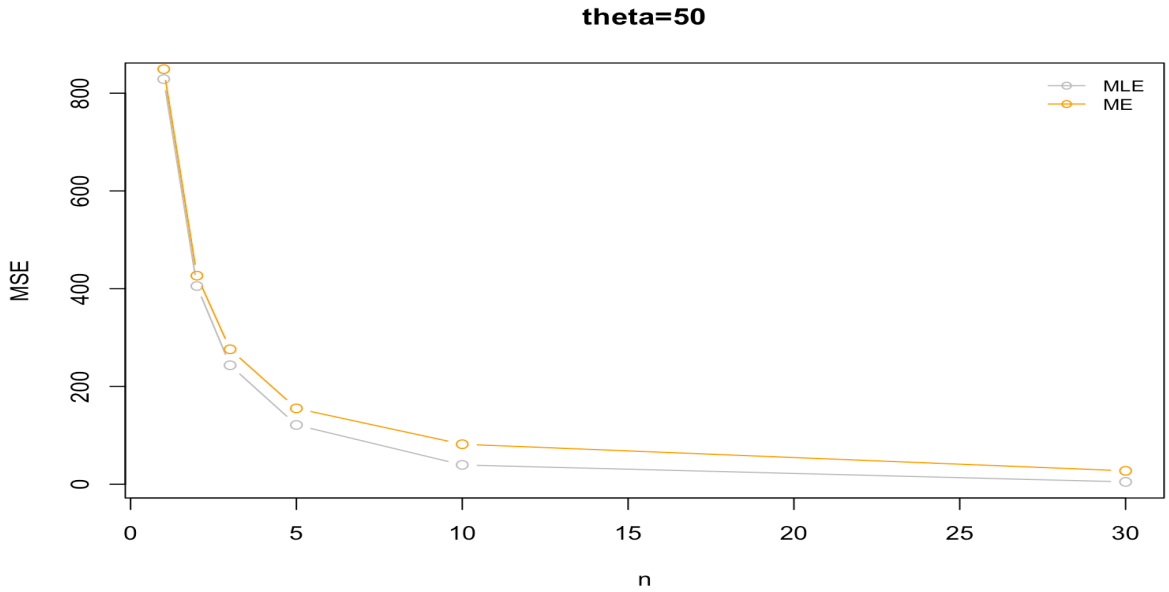
```
> plot(c(1,2,3,5,10,30),c(mse1_1[1],mse2_1[1],mse3_1[1],mse5_1[1],mse10_1[1],
+                         mse30_1[1]),type="b",ylab='MSE',xlab='n',
+                         col='grey',main="theta=1")
> lines(c(1,2,3,5,10,30),c(mse1_1[2],mse2_1[2],mse3_1[2],mse5_1[2],mse10_1[2],
+                          mse30_1[2]),type="b",col='orange')
> legend("topright",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,
+        cex=0.8,bty='n')
```

```
> plot(c(1,2,3,5,10,30),c(mse1_100[1],mse2_100[1],mse3_100[1],mse5_100[1],mse10_100[1],
+                         mse30_100[1]),type="b",ylab='MSE',xlab='n',
+                         col='grey',main="theta=100")
> lines(c(1,2,3,5,10,30),c(mse1_100[2],mse2_100[2],mse3_100[2],mse5_100[2],mse10_100[2],
+                          mse30_100[2]),type="b",col='orange')
> legend("topright",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,
+        cex=0.8,bty='n')
>
```

```
> plot(c(1,2,3,5,10,30),c(mse1_50[1],mse2_50[1],mse3_50[1],mse5_50[1],mse10_50[1],
+                         mse30_50[1]),type="b",ylab='MSE',xlab='n',
+                         col='grey',main="theta=50")
> lines(c(1,2,3,5,10,30),c(mse1_50[2],mse2_50[2],mse3_50[2],mse5_50[2],mse10_50[2],
+                          mse30_50[2]),type="b",col='orange')
> legend("topright",legend=c("MLE","ME"),col=c('grey','orange'),
+        text.col=c('black','black'),lty=1,pch=1,inset=0.01,ncol=1,
+        cex=0.8,bty='n')
>
```


theta=1
MSE
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
5
10
15
20
25
30
n
MLE
ME


theta=100
MSE
0
500
1000
2000
3000
0
5
10
15
20
25
30
n
MLE
ME



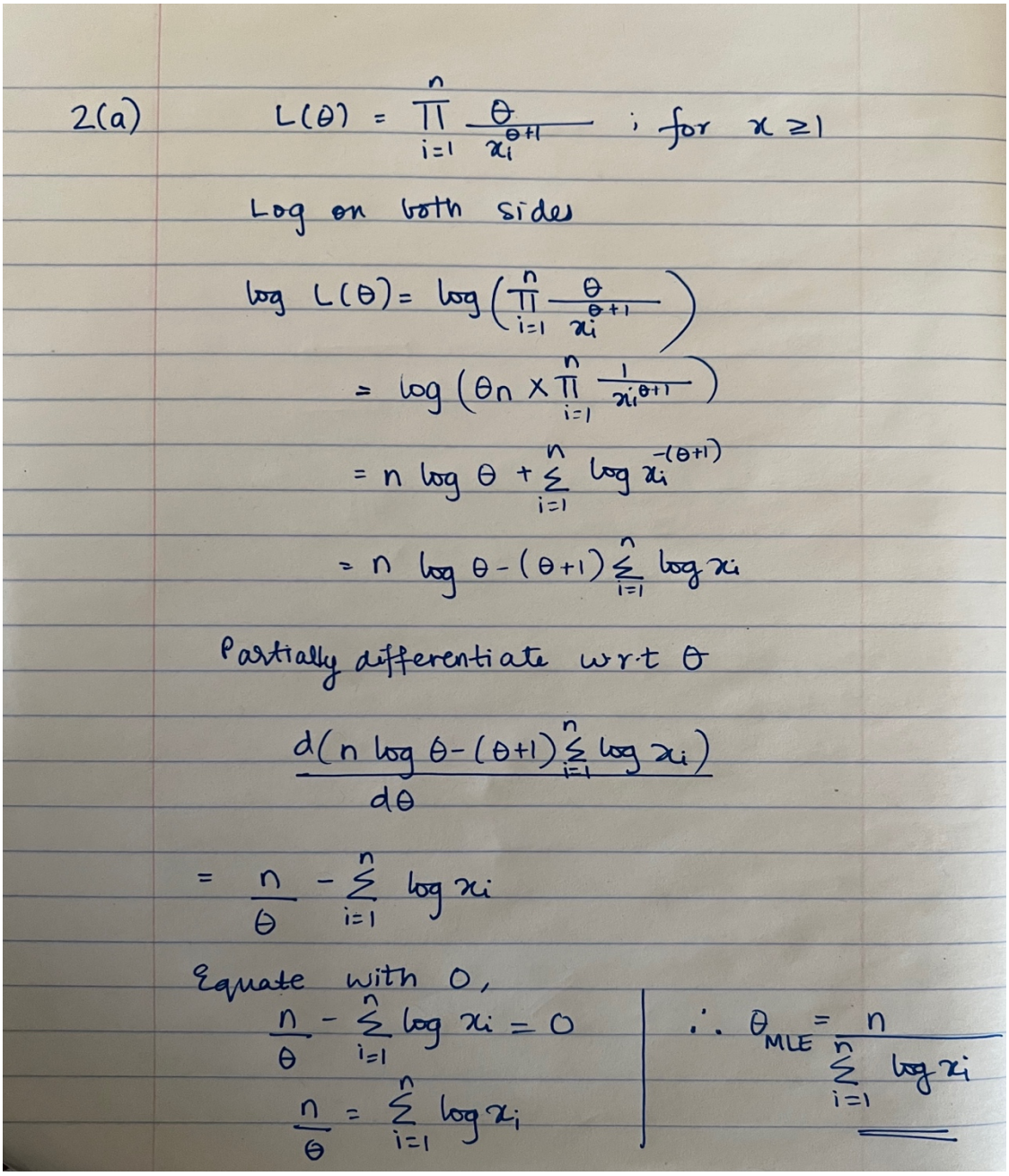

**(d) Based on (c), which estimator is better? Does the answer depend on n or θ? Explain. Provide justification for all your conclusions.**

**Ans:** By observing graph 2 it is clearly concluded that no matter what value of theta is fixed, the resulting graphs are similar. So, it is depicted that the estimator is not depended on the value of theta. By observing Graph 1 it is concluded that for fixed n, small values of n we can use method of moment estimator. But as the value of n increases Maximum likelihood estimator is better. So, for the mean squared error Maximum likelihood is preferred.
Also, for smaller sample sizes like n = 1,2 both MLE and MME estimators gave similar Mean Square Error, but for larger sample sizes like n = 3,5,10,30 MLE estimator is having less MSE than MME estimators. Thus, MLE is better estimator when compared to MME.
It is evident from the graph that for larger values of sample size (n) we have lesser MSE i.e., $\boldsymbol{n \propto 1 / MSE}$

**Question 2**

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

**Ans:**



2(a) $L(\theta) = \prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}$ ; for $x \geq 1$

Log on both sides

$$\log L(\theta) = \log\left(\prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}\right)$$

$$= \log\left(\theta^n \times \prod_{i=1}^{n} \frac{1}{x_i^{\theta+1}}\right)$$

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

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

Partially differentiate w.r.t θ

$$\frac{d\left(n \log \theta - (\theta+1) \sum_{i=1}^{n} \log x_i\right)}{d\theta}$$

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

Equate with 0,

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

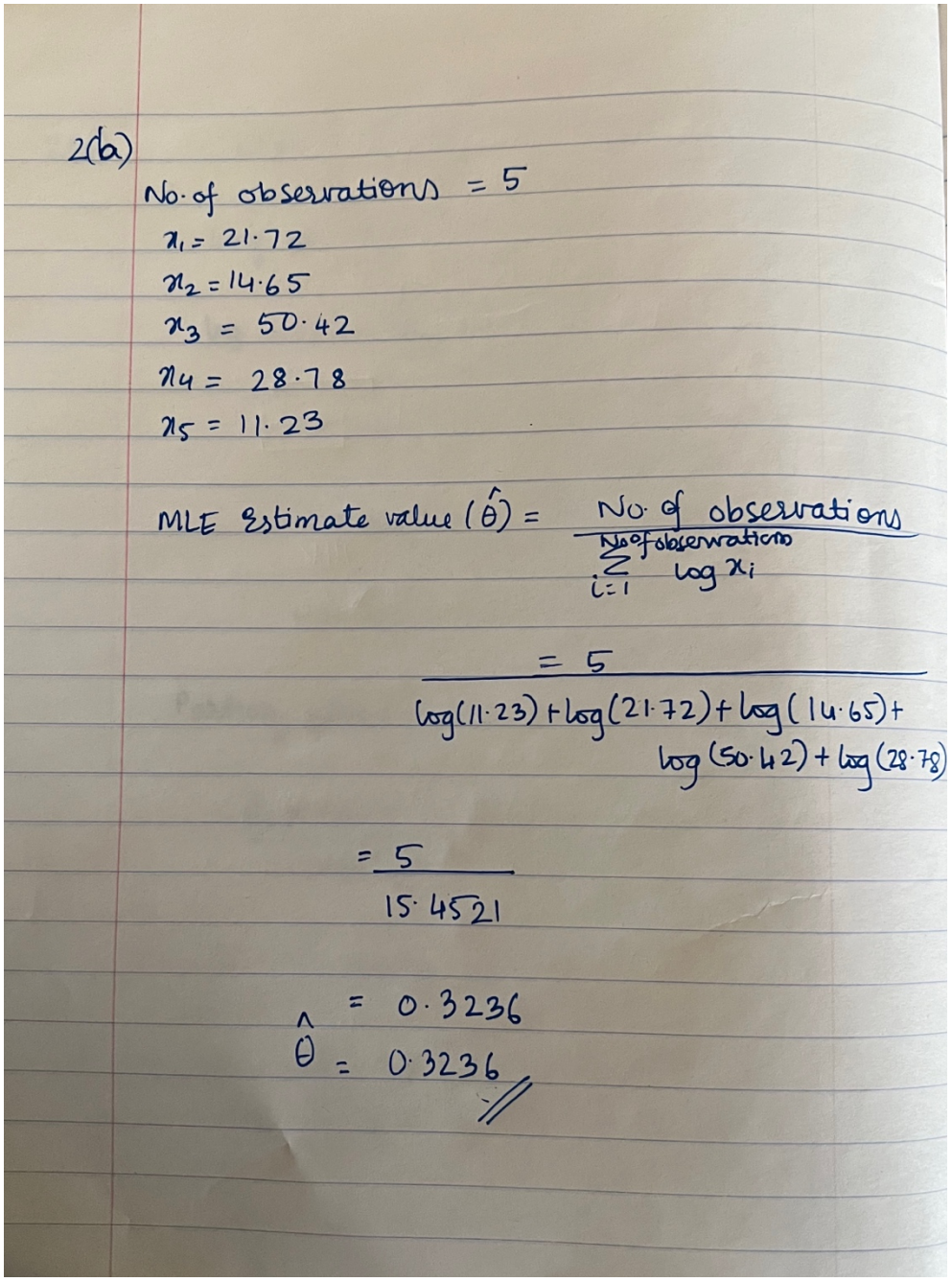
$$\frac{n}{\theta} = \sum_{i=1}^{n} \log x_i$$

$$\therefore \theta_{MLE} = \frac{n}{\sum_{i=1}^{n} \log x_i}$$

**b)**

**Suppose n = 5 and the sample values are $x_1 = 21.72$, $x_2 = 14.65$, $x_3 = 50.42$, $x_4 = 28.78$, $x_5 = 11.23$. Use the expression in (a) to provide the maximum likelihood estimate for θ based on these data.**

**Ans:**



2(b)

No. of observations = 5

$x_1 = 21.72$

$x_2 = 14.65$

$x_3 = 50.42$

$x_4 = 28.78$

$x_5 = 11.23$

MLE Estimate value $(\hat{\theta}) = \frac{\text{No. of observations}}{\sum_{i=1}^{\text{No of observations}} \log x_i}$

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

$= \frac{5}{15.4521}$

$= 0.3236$

$\hat{\theta} = 0.3236$

**c)**

**Even though we know the maximum likelihood estimate from (b), use the data in (b) to obtain the estimate by numerically maximizing the log-likelihood function using optim function in R. Do your answers match?**

**Code:**

```
mle = function (theta, value) {              #negative log function
   -sum(log(theta) - (theta+1)*log(value))
}
x = c(21.72,14.65,50.42,28.78,11.23)         #data
#mle by optim
mleo = optim(par = theta<-0, fn = mle, value=x, method="Brent", hessian = TRUE,lower=0,upper=10)
print(mleo$par)
```

**Output:**

```
> mle = function (theta, value) {              #negative log function
+    -sum(log(theta) - (theta+1)*log(value))
+ }
> x = c(21.72,14.65,50.42,28.78,11.23)         #data
> #mle by optim
> mleo = optim(par = theta<-0, fn = mle, value=x, method="Brent", hessian = TRUE,lower=0,upper=10)
> print(mleo$par)
[1] 0.3233874
>
```

As computed numerically the maximum likelihood in 2b, the output of 2c matches it which is approximately 0.323

**d)**

**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. Are these approximations going to be good? Justify your answer.**

**Code:**

```
sd = (1/mleo$hessian)^0.5  #standard deviation
mean = mean(x)             #mean
ci = function(mean, sd){   #confidence interval
  aci=mean+c(-1,1)*c(sd)*qnorm(0.975)
}
ACI = ci(mean,sd)   #actual confidence interval
print(sd)
print(mean)
print(ACI)
upperBound = mleo$par + qnorm(0.975) * sd
print(upperBound)
lowerBound = mleo$par - qnorm(0.975) * sd
print(lowerBound)
```

**Output:**

```
> sd = (1/mleo$hessian)^0.5  #standard deviation
> mean = mean(x)             #mean
> ci = function(mean, sd){   #confidence interval
+   aci=mean+c(-1,1)*c(sd)*qnorm(0.975)
+ }
> ACI = ci(mean,sd)   #actual confidence interval
> print(sd)
          [,1]
[1,] 0.1446219
> print(mean)
[1] 25.36
> print(ACI)
[1] 25.07655 25.64345
> upperBound = mleo$par + qnorm(0.975) * sd
> print(upperBound)
          [,1]
[1,] 0.6068411
> lowerBound = mleo$par - qnorm(0.975) * sd
>
```

The values obtained from b and c are within the range of 0.0323 and 0.606, hence the obtained interval helps in finding the correct estimate.