Mini Project #3

Group 2.

Group Members:

Manav Gupta [MXG220027]

Shalin Ronakkumar Kaji [SXK220263]

Contribution:

Manav : Solved Question 1.

Shalin : Solved Question 2 and prepared the final draft of the documentation.

Question 1:

We would like to estimate the parameter θ (> 0) of a Uniform (0, θ)

population based on a random sample $X_1, . . . ,X_n$ from the population.

Now for a Uniform Distribution the density function is given by:

$f(x) = 1/(b – a),$ $a < x < b,$

thus, it has a constant density function to give equal preference to all values.

The expectation of X is given by:

E(X) = (a + b)/2 ------------> **E(X) = (0+Θ)/2 = Θ/2**

Method of Moments estimator Θ_{MME}(hat):

The first sample moment M_1 = X(bar) = Θ/2

Hence, Θ_{MME}(hat) = 2X(bar).

Method of Maximum Likelihood estimator Θ_{MLE}(hat):

Θ_{MLE}(hat) = $X_{(n)}$, $X_{(n)}$ is the maximum of the sample.

Goal : To compare the Mean Squared Errors [MSE] of both the estimators to determine which estimator is better.

The *mean squared error* of an estimator $\hat{\vartheta}$ of a parameter ϑ is defined as

$E\{(\hat{\vartheta} - \vartheta)^2\}$.

We are storing the different values of 'n' and ϴ in two vectors and will use them to calculate MSE of both estimators in form of different cases.

```
> n <- c(1,2,3,5,10,30)
> theta <- c(1,5,50,100)
> theta_mme=numeric(0)
> theta_mle= numeric(0)
```

a) For computing MSE of an estimator using Monte-Carlo simulation:

   We will generate a large number of random samples from the population using the **runif()** function using a given value of 'n' and ϴ as parameters.

```
> for(k in 1:1000){
    X <- runif(n,0,ϴ)
    theta_mme[k] <- 2*mean(X)
    theta_mle[k] <- max(X)
    }
```

   Calculate the difference between the estimate and the true value of the parameter: For each random sample, we calculate the difference between the estimate of the parameter and the true value of the parameter. Then, square the difference. The MSE will be obtained by calculating the average of the squared differences.

```
> mse_mme[a] <- mean((theta_mme-i)^2)
>  mse_mle[a] <- mean((theta_mle-i)^2)
```

b) Computing the mean squared errors of both Θ_{MME}(hat) and Θ_{MLE}(hat) using Monte-Carlo simulation using N=1000 replications for (n=10, Θ):

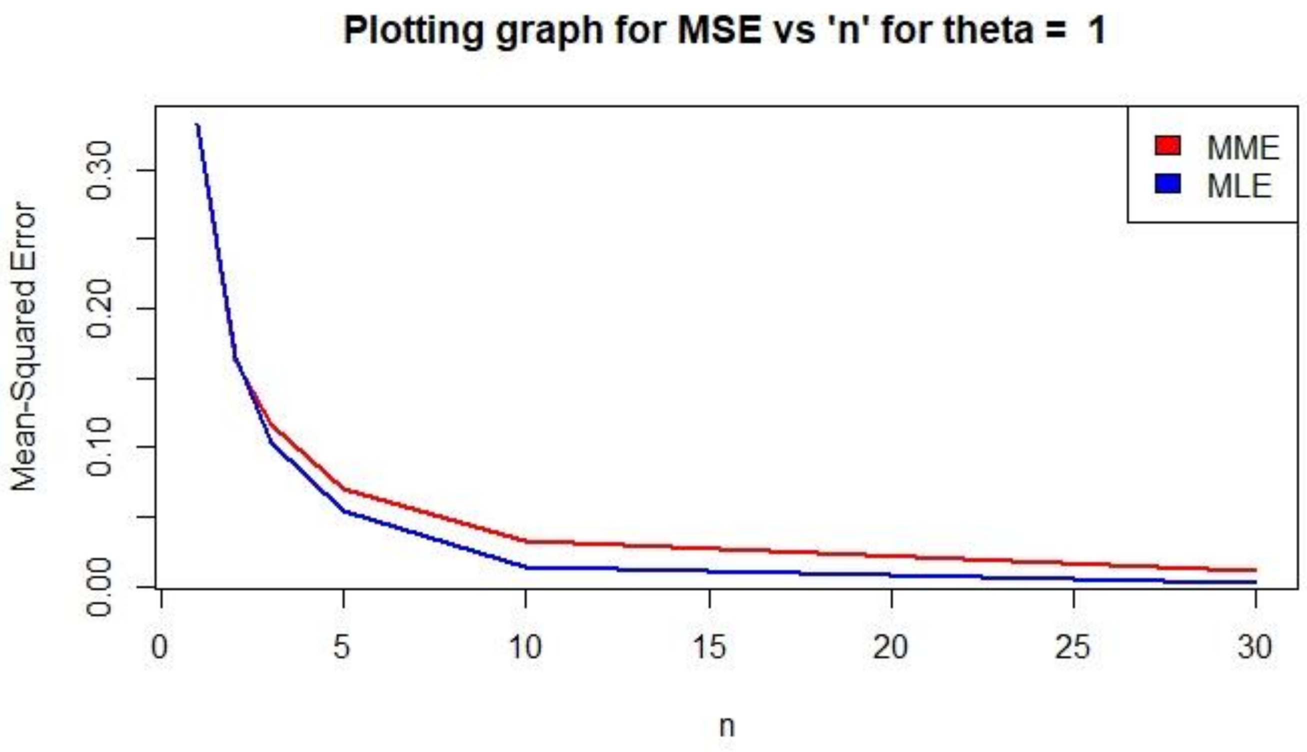
```
+     mse_mme[a] <- mean((theta_mme-i)^2)
+     mse_mle[a] <- mean((theta_mle-i)^2)
+     print(paste("Printing MSE_MME for n = ",n," and theta = ", i, " MSE = ",mse_mme[a]))
+     print(paste("Printing MSE_MLE for n = ",n," and theta = ", i, " MSE = ",mse_mle[a]))
+
+     a=a+1
+
+ }
[1] "Printing MSE_MME for n =  10  and theta =  1  MSE =  0.0352553024114573"
[1] "Printing MSE_MLE for n =  10  and theta =  1  MSE =  0.0150557162670325"
[1] "Printing MSE_MME for n =  10  and theta =  5  MSE =  0.803326752445658"
[1] "Printing MSE_MLE for n =  10  and theta =  5  MSE =  0.390155024666999"
[1] "Printing MSE_MME for n =  10  and theta =  50  MSE =  87.4396754493088"
[1] "Printing MSE_MLE for n =  10  and theta =  50  MSE =  37.6046235729175"
[1] "Printing MSE_MME for n =  10  and theta =  100  MSE =  314.369141204157"
[1] "Printing MSE_MLE for n =  10  and theta =  100  MSE =  151.484722248628"
>
```

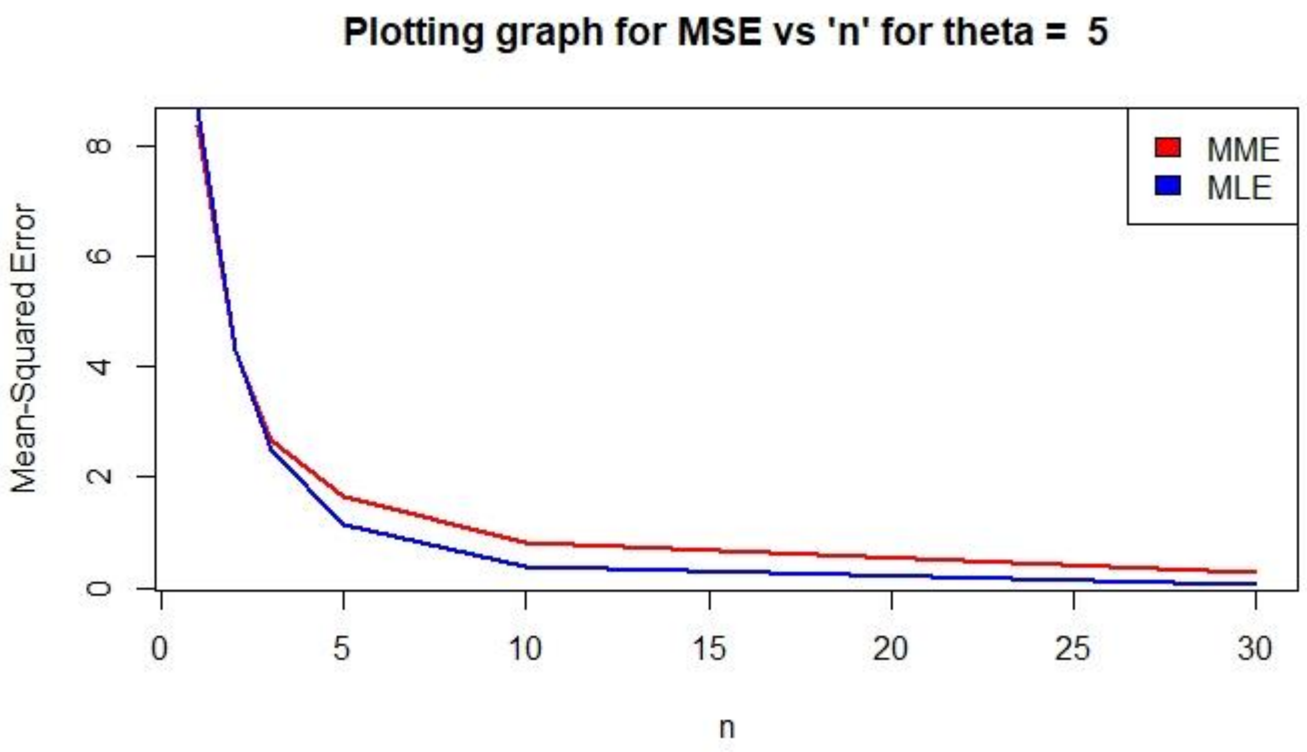
c) Repeating the process in (b) for the other combinations of (n,Θ) and summarising the results graphically:

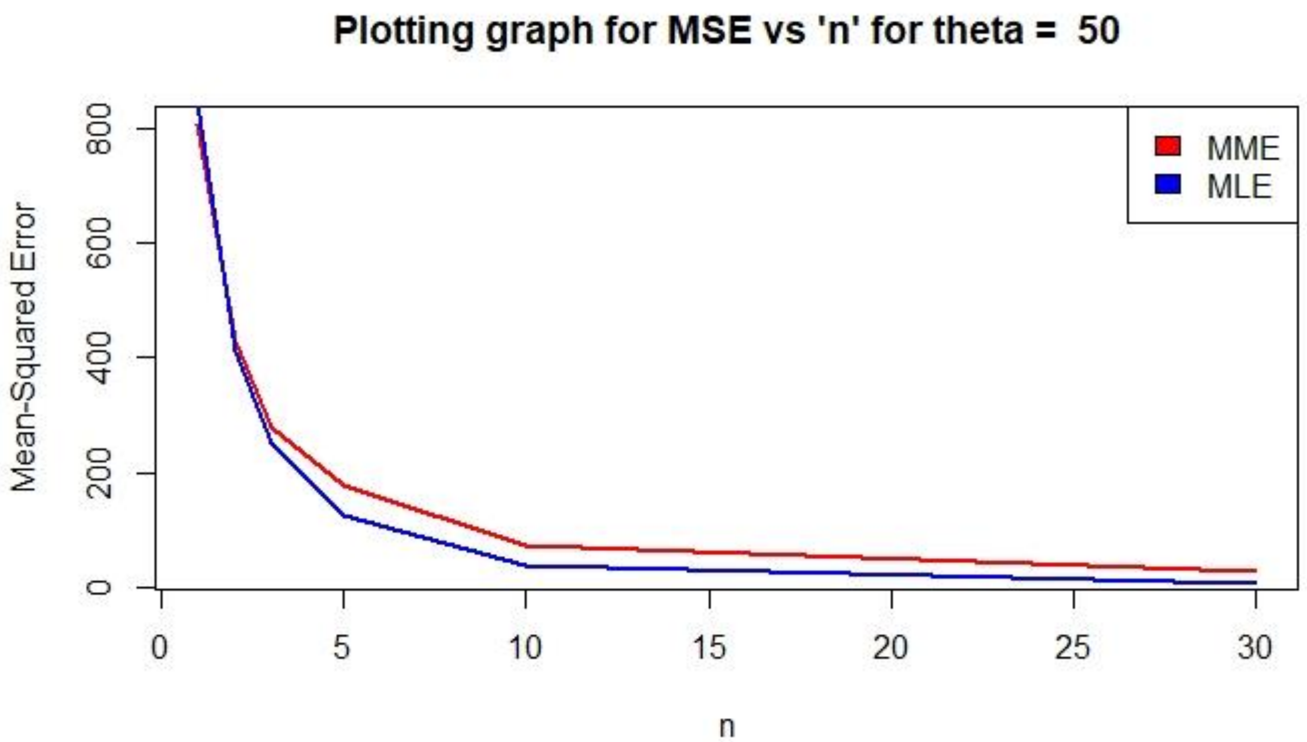
Console Terminal Jobs

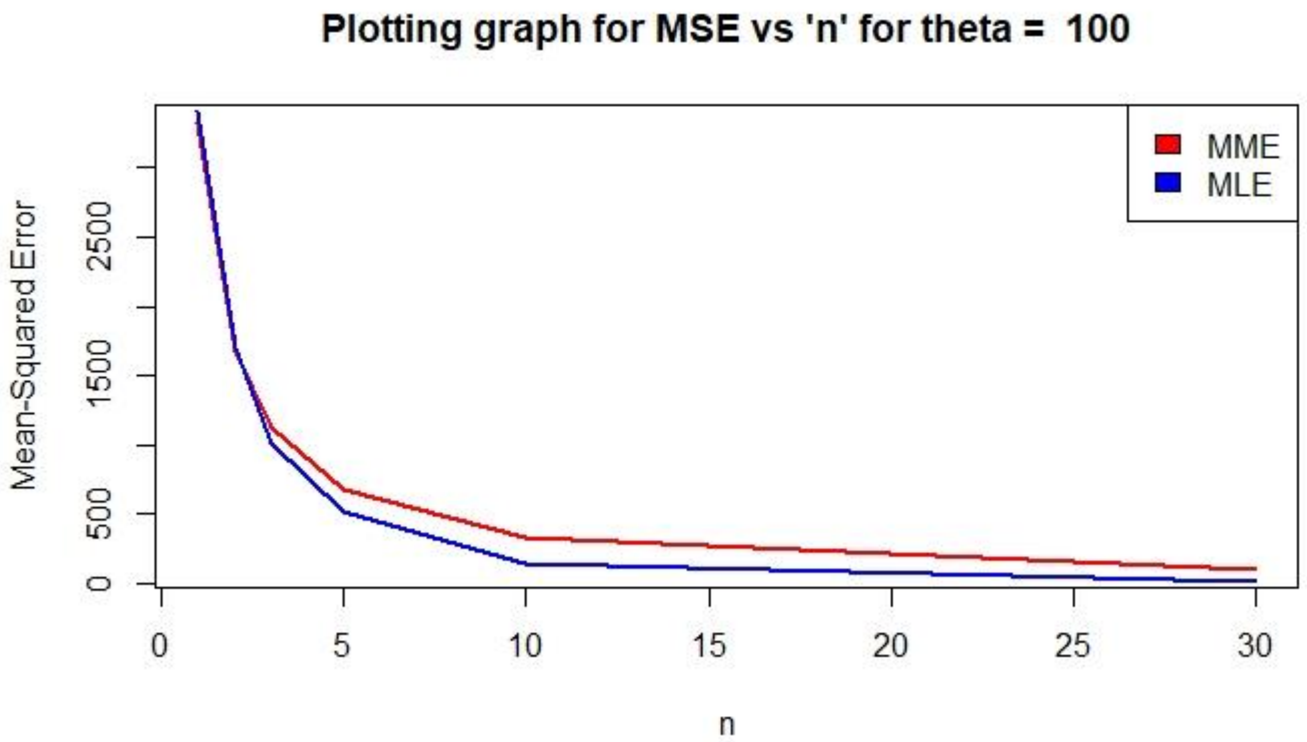
R 4.1.1 · C:/Users/Shalin Kaji/Desktop/RProg/datasciencecoursera/

```
[1] "Printing MSE_MME for n =  1  and theta =  1  MSE =  0.332288505534058"
[1] "Printing MSE_MLE for n =  1  and theta =  1  MSE =  0.332793350524356"
[1] "Printing MSE_MME for n =  2  and theta =  1  MSE =  0.16282822785328 7"
[1] "Printing MSE_MLE for n =  2  and theta =  1  MSE =  0.1659670743478 5"
[1] "Printing MSE_MME for n =  3  and theta =  1  MSE =  0.116233963465135"
[1] "Printing MSE_MLE for n =  3  and theta =  1  MSE =  0.103765023237941"
[1] "Printing MSE_MME for n =  5  and theta =  1  MSE =  0.0709173313790221"
[1] "Printing MSE_MLE for n =  5  and theta =  1  MSE =  0.0538733711680077"
[1] "Printing MSE_MME for n =  10  and theta =  1  MSE =  0.0323782510024624"
[1] "Printing MSE_MLE for n =  10  and theta =  1  MSE =  0.0145117494345159"
[1] "Printing MSE_MME for n =  30  and theta =  1  MSE =  0.0118606983069364"
[1] "Printing MSE_MLE for n =  30  and theta =  1  MSE =  0.00226202901679772"
[1] "Printing MSE_MME for n =  1  and theta =  5  MSE =  8.34995764902451"
[1] "Printing MSE_MLE for n =  1  and theta =  5  MSE =  8.69032935495752"
[1] "Printing MSE_MME for n =  2  and theta =  5  MSE =  4.29207219899921"
[1] "Printing MSE_MLE for n =  2  and theta =  5  MSE =  4.28741951401999"
[1] "Printing MSE_MME for n =  3  and theta =  5  MSE =  2.68184043603313"
[1] "Printing MSE_MLE for n =  3  and theta =  5  MSE =  2.4875210020467"
[1] "Printing MSE_MME for n =  5  and theta =  5  MSE =  1.65139408224332"
[1] "Printing MSE_MLE for n =  5  and theta =  5  MSE =  1.15469224341057"
[1] "Printing MSE_MME for n =  10  and theta =  5  MSE =  0.83769214036078"
[1] "Printing MSE_MLE for n =  10  and theta =  5  MSE =  0.384841379634299"
[1] "Printing MSE_MME for n =  30  and theta =  5  MSE =  0.289757335413397"
[1] "Printing MSE_MLE for n =  30  and theta =  5  MSE =  0.0505467424878674"
[1] "Printing MSE_MME for n =  1  and theta =  50  MSE =  806.336490230794"
[1] "Printing MSE_MLE for n =  1  and theta =  50  MSE =  844.545260729657"
[1] "Printing MSE_MME for n =  2  and theta =  50  MSE =  430.607539137447"
[1] "Printing MSE_MLE for n =  2  and theta =  50  MSE =  411.705437739952"
[1] "Printing MSE_MME for n =  3  and theta =  50  MSE =  279.171723001031"
[1] "Printing MSE_MLE for n =  3  and theta =  50  MSE =  251.505697336471"
[1] "Printing MSE_MME for n =  5  and theta =  50  MSE =  176.700819102735"
[1] "Printing MSE_MLE for n =  5  and theta =  50  MSE =  123.9122488398"
[1] "Printing MSE_MME for n =  10  and theta =  50  MSE =  74.0942810093006"
[1] "Printing MSE_MLE for n =  10  and theta =  50  MSE =  36.1664377257685"
[1] "Printing MSE_MME for n =  30  and theta =  50  MSE =  28.3075092869729"
[1] "Printing MSE_MLE for n =  30  and theta =  50  MSE =  5.29308930719456"
[1] "Printing MSE_MME for n =  1  and theta =  100  MSE =  3320.04952586235"
[1] "Printing MSE_MLE for n =  1  and theta =  100  MSE =  3415.50999441179"
[1] "Printing MSE_MME for n =  2  and theta =  100  MSE =  1679.15584665729"
[1] "Printing MSE_MLE for n =  2  and theta =  100  MSE =  1697.40550098121"
[1] "Printing MSE_MME for n =  3  and theta =  100  MSE =  1127.34198314197"
[1] "Printing MSE_MLE for n =  3  and theta =  100  MSE =  1011.97027648146"
[1] "Printing MSE_MME for n =  5  and theta =  100  MSE =  674.43373513137 7"
```


Plotting graph for MSE vs 'n' for theta = 1
Mean-Squared Error
0.00
0.10
0.20
0.30
0
5
10
15
20
25
30
n
MME
MLE


Plotting graph for MSE vs 'n' for theta = 5
Mean-Squared Error
0
2
4
6
8
0
5
10
15
20
25
30
n
MME
MLE


Plotting graph for MSE vs 'n' for theta = 50
MME
MLE
Mean-Squared Error
0
200
400
600
800
0
5
10
15
20
25
30
n


Plotting graph for MSE vs 'n' for theta = 100
MME
MLE
Mean-Squared Error
0
500
1500
2500
0
5
10
15
20
25
30
n

d) Based on the graphs plotted in (c) for the MSE of both estimators vs ‘n’ we can conclude that:

Maximum Likelihood Estimator [MLE] is better compared to Method of Moment Estimator [MME].

This is fairly evident from the 4^{th} graph of (c) where MLE is significantly lesser (tending to zero) than MME for any given value of ϴ when large values of ‘n’ are taken.

When choosing the better of the two estimators, the values of n and ϴ do not matter. However, for a constant ϴ, the MLE decreases drastically with increase in ‘n’ (>50).

R Code:

```
n <- c(1,2,3,5,10,30)
theta <- c(1,5,50,100)
theta_mme = numeric(0)
theta_mle = numeric(0)
mse_mme = numeric(0)
mse_mle = numeric(0)

for (i in theta){
 a=1
 for (j in n){

  for(k in 1:1000){
   X <- runif(j,0,i)
   theta_mme[k] <- 2*mean(X)
   theta_mle[k] <- max(X)
   }
  mse_mme[a] <- mean((theta_mme-i)^2)
  mse_mle[a] <- mean((theta_mle-i)^2)
  print(paste("Printing MSE_MME for n = ",j," and theta = ", i, " MSE = ",mse_mme[a]))
  print(paste("Printing MSE_MLE for n = ",j," and theta = ", i, " MSE = ",mse_mle[a]))
  a=a+1
 }

 plot(n,mse_mme,type = 'l', col="red",lwd=2, ylab="")
 title(main = paste("Plotting graph for MSE vs 'n' for theta = ",i), ylab = 'Mean-Squared Error')
 lines(n,mse_mle,type = 'l', col="blue",lwd=2)
 legend("topright", legend = c('MME','MLE'), fill = c('red','blue')) }
```

Question 2:

a) Here the lifetime of an electric component is modelled by a continuous random variable which follows the Pareto Distribution, and its density function is given by:

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

To find the maximum likelihood estimator, we need to maximize the probability of observing a value close to *x* (as it is proportional to the density *f(x)* .

Question - 2

(a) $L = \prod_{i=1}^{n} f(x_1, \ldots x_n ; \theta)$

$$L = \prod_{i=1} \frac{\theta}{x^{\theta+1}}$$

$$L = \frac{\theta^n}{\prod x^{\theta+1}}$$

$$\ln L = n \ln \theta - (\theta+1) \ln \Sigma x_i$$

$$\frac{\partial(\ln L)}{\partial \theta} = \frac{n}{\theta} - \ln \Sigma x_i = 0$$

$$\boxed{\hat{\theta} = \frac{n}{\ln \Sigma x_i}}$$

b) x1 = 21.72, x2 = 14.65, x3 = 50.42, x4 = 28.78, x5 = 11.23.

(b) $\hat{\theta} = 5 / \left[\ln(21.72) + \ln(14.65) + \ln(50.42) + \ln(28.78) + \ln(11.23) \right]$

$$\boxed{\hat{\theta} = 0.32338}$$

c) Obtaining the maximum-likelihood estimate by numerically maximizing the log-likelihood function using optim function in R, we get the same answer as computed in (b).

```
> neg.loglik.fun<-function(par,dat){
 total=0
 for(i in dat){
  total<- total + (log(par)-(par+1)*log(i) )
 }
 return(-total)
}

> ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T, dat=dat)
```

```
> ml.est
$par
[1] 0.3234863

$value
[1] 26.10585

$counts
function gradient
      38       NA

$convergence
[1] 0

$message
NULL

$hessian
         [,1]
[1,] 47.78223
```

d)

```
> ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T, dat=dat)
> ml.est
> se<-sqrt(diag(solve(ml.est$hessian)))
> CI<- ml.est$par +c(-1,1)*qnorm(.975)*se
> CI
```

```
> se<-sqrt(diag(solve(ml.est$hessian)))
> CI<- ml.est$par +c(-1,1)*qnorm(.975)*se
> CI
[1] 0.03994598 0.60702668
> |
```

From the confidence interval which we calculated above we can say that the range (0.0399, 0.607) will capture the value of θ for 95% of the times.

In the part (b) and (c) the calculated value of θ lies within this range. So, these approximations will be good in determining the value of θ.

R Code:

```
# maximizing the log-likelihood function of Pareto distribution by using Optim function.
dat <- c(21.72,14.65,50.42,28.78,11.23)

neg.loglik.fun<-function(par,dat){
 total=0
 for(i in dat){
  total<- total + (log(par)-(par+1)*log(i) )
 }
 return(-total)
}

ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T, dat=dat)
ml.est
se<-sqrt(diag(solve(ml.est$hessian)))
CI<- ml.est$par +c(-1,1)*qnorm(.975)*se
CI
```