Mini Project 03
Mini Project Group 16
Group Members
Dhairya Pinakin Desai: DPD140130
Jaya Venkata Sai Krishna Kosana: JXK220036
Krishna Sai Gandi: KXG220038

Contribution: The three team members collaborated seamlessly, each putting in an equal amount of effort to complete the assigned mini project.

Q.1:

As per the question we would like to estimate the parameter θ (> 0) of a Uniform (0, θ) population based on a random sample $X_1, \ldots, X_n$ from the population.

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

$$f(x) = \frac{1}{b - a}, \qquad \text{for} \;\; a < x < b,$$

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

The expectation of x is given by:

$$\boldsymbol{E}(X) = \frac{a + b}{2}$$

$$\boldsymbol{E}(X) = \frac{\boldsymbol{0 + \theta}}{\boldsymbol{2}} = \frac{\boldsymbol{\theta}}{\boldsymbol{2}}$$

Method of Moments estimator $\hat{\theta}_{MME}$:
The first sample moment M1 = $\bar{X} = \frac{\theta}{2}$
Hence, $\hat{\theta}_{MME} = 2 * \bar{X}$

Method of Maximum Likelihood estimator $\hat{\theta}_{MLE}$:
$\hat{\theta}_{MLE}$= 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 $\hat{\theta}$ of a parameter θ is defined as
E $\{(\hat{\theta} - \theta)^2\}$

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)
> thetas <- c(1,5,50,100)
> theta_mme = numeric(0)
> theta_mle = numeric(0)
```

(a):

To compute 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){
+       data <- runif(i,0,j)
+       theta_mme[k] <- 2*mean(data)
+       theta_mle[k] <- max(data)
+     }
```

Note: In above code, which is a part of a bigger code, i and j represent n and thetas respectively.

Now to 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-j)^2)
+     mse_mle[a] <- mean((theta_mle-j)^2)
```

(b):

Computing the mean squared errors of both $\hat{\theta}_{MME}$ and $\hat{\theta}_{MLE}$ using Monte-Carlo simulation using N=1000 replications for n and θ:

```
> for (i in n){
+   a=1
+   for (j in thetas){
+     for(k in 1:1000){
+       data <- runif(i,0,j)
+       theta_mme[k] <- 2*mean(data)
```

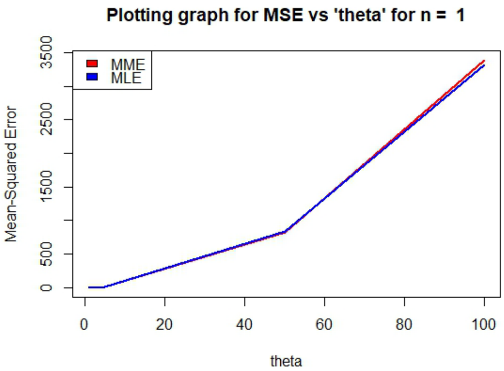
```
+       theta_mle[k] <- max(data)
+     }
+     mse_mme[a] <- mean((theta_mme-j)^2)
+     mse_mle[a] <- mean((theta_mle-j)^2)
+     print(paste("Printing MSE_MME for theta = ",j," and n =
", i, " MSE = ",mse_mme[a]))
+     print(paste("Printing MSE_MLE for theta = ",j," and n =
", i, " MSE = ",mse_mle[a]))
+     a=a+1
+   }
```

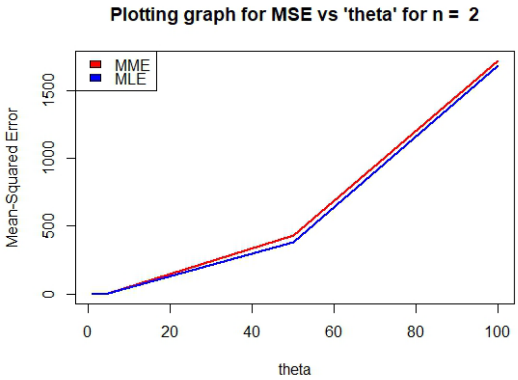
Result:

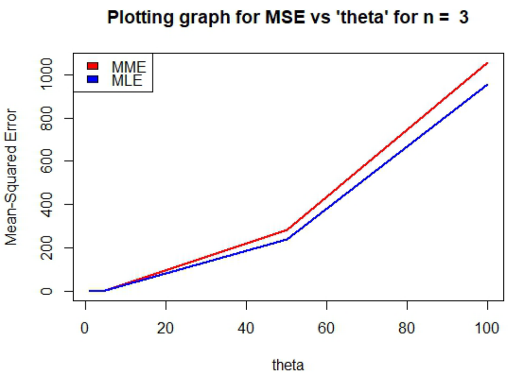
```
[1] "Printing MSE_MME for theta =  1  and n =  1  MSE =  0.331153007710925"
[1] "Printing MSE_MLE for theta =  1  and n =  1  MSE =  0.332681779244825"
[1] "Printing MSE_MME for theta =  5  and n =  1  MSE =  8.46721392786986"
[1] "Printing MSE_MLE for theta =  5  and n =  1  MSE =  8.53474694894026"
[1] "Printing MSE_MME for theta =  50  and n =  1  MSE =  825.620165002546"
[1] "Printing MSE_MLE for theta =  50  and n =  1  MSE =  823.676510331807"
[1] "Printing MSE_MME for theta =  100  and n =  1  MSE =  3490.57462663835"
[1] "Printing MSE_MLE for theta =  100  and n =  1  MSE =  3196.19782803692"
[1] "Printing MSE_MME for theta =  1  and n =  2  MSE =  0.172210263370391"
[1] "Printing MSE_MLE for theta =  1  and n =  2  MSE =  0.165092296803794"
[1] "Printing MSE_MME for theta =  5  and n =  2  MSE =  3.88987215723434"
[1] "Printing MSE_MLE for theta =  5  and n =  2  MSE =  3.82787117973797"
[1] "Printing MSE_MME for theta =  50  and n =  2  MSE =  412.149917776478"
[1] "Printing MSE_MLE for theta =  50  and n =  2  MSE =  408.096169423815"
[1] "Printing MSE_MME for theta =  100  and n =  2  MSE =  1708.57113066997"
[1] "Printing MSE_MLE for theta =  100  and n =  2  MSE =  1676.68498326964"
[1] "Printing MSE_MME for theta =  1  and n =  3  MSE =  0.107110610254249"
[1] "Printing MSE_MLE for theta =  1  and n =  3  MSE =  0.0985192953282392"
[1] "Printing MSE_MME for theta =  5  and n =  3  MSE =  2.59802914675163"
[1] "Printing MSE_MLE for theta =  5  and n =  3  MSE =  2.27219639861795"
[1] "Printing MSE_MME for theta =  50  and n =  3  MSE =  284.348306357839"
[1] "Printing MSE_MLE for theta =  50  and n =  3  MSE =  254.53787317292"
[1] "Printing MSE_MME for theta =  100  and n =  3  MSE =  1076.94718987417"
[1] "Printing MSE_MLE for theta =  100  and n =  3  MSE =  971.668350630441"
[1] "Printing MSE_MME for theta =  1  and n =  5  MSE =  0.0650705966500358"
[1] "Printing MSE_MLE for theta =  1  and n =  5  MSE =  0.0483789451334599"
[1] "Printing MSE_MME for theta =  5  and n =  5  MSE =  1.58591028362936"
[1] "Printing MSE_MLE for theta =  5  and n =  5  MSE =  1.11859074230106"
[1] "Printing MSE_MME for theta =  50  and n =  5  MSE =  175.270304320806"
[1] "Printing MSE_MLE for theta =  50  and n =  5  MSE =  121.436298877823"
[1] "Printing MSE_MME for theta =  100  and n =  5  MSE =  743.872221558526"
[1] "Printing MSE_MLE for theta =  100  and n =  5  MSE =  461.752386201887"
[1] "Printing MSE_MME for theta =  1  and n =  10  MSE =  0.0312678327671693"
[1] "Printing MSE_MLE for theta =  1  and n =  10  MSE =  0.0149653287328979"
[1] "Printing MSE_MME for theta =  5  and n =  10  MSE =  0.834466216656068"
[1] "Printing MSE_MLE for theta =  5  and n =  10  MSE =  0.39591568450704"
[1] "Printing MSE_MME for theta =  50  and n =  10  MSE =  84.2656913977662"
[1] "Printing MSE_MLE for theta =  50  and n =  10  MSE =  34.9285681143642"
[1] "Printing MSE_MME for theta =  100  and n =  10  MSE =  327.726951311369"
[1] "Printing MSE_MLE for theta =  100  and n =  10  MSE =  135.839827377665"
[1] "Printing MSE_MME for theta =  1  and n =  30  MSE =  0.0112562795610284"
[1] "Printing MSE_MLE for theta =  1  and n =  30  MSE =  0.00210697756891632"
[1] "Printing MSE_MME for theta =  5  and n =  30  MSE =  0.264864211850186"
[1] "Printing MSE_MLE for theta =  5  and n =  30  MSE =  0.0494868148317671"
[1] "Printing MSE_MME for theta =  50  and n =  30  MSE =  26.0062080559792"
[1] "Printing MSE_MLE for theta =  50  and n =  30  MSE =  4.72302897416884"
[1] "Printing MSE_MME for theta =  100  and n =  30  MSE =  118.979294205354"
[1] "Printing MSE_MLE for theta =  100  and n =  30  MSE =  21.0407406958756"
```

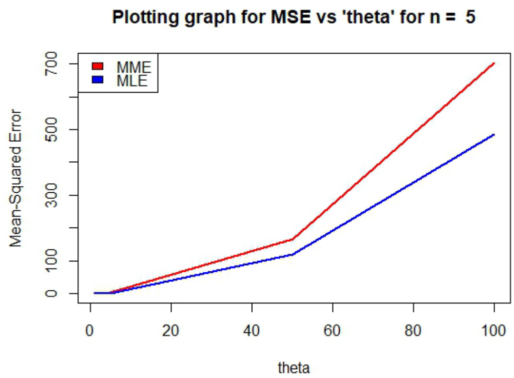
(c):

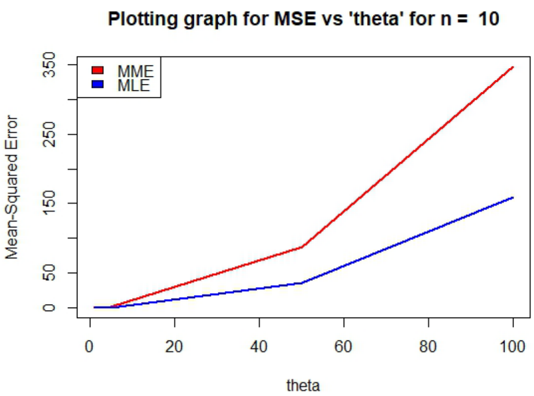
Repeating the process in (b) for the other combinations of (n, θ) and summarizing the results graphically:

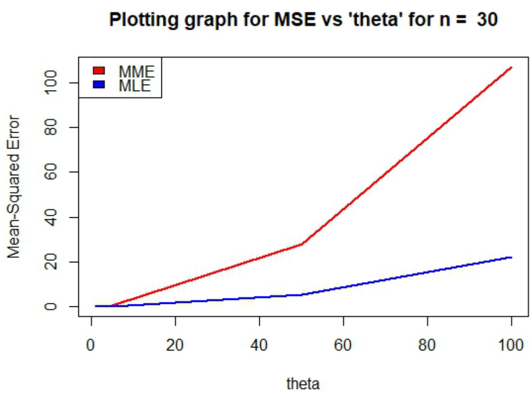












(d):

From the graphs shown above, we can easily observe that for small values of n like 1 and 3, the performance of both the estimators (MME, MLE) is almost the same. As soon as the value of n increases, we can notice a difference in the performance of these estimators.

The Mean Squared Error of MLE is more as compared to that of MME, so we can conclude that MME is the better estimator from the two estimators.
We can also see that the estimation is closer to the actual values for higher values of n.

R Code:

```
n <- c(1,2,3,5,10,30)
thetas <- c(1,5,50,100)
theta_mme <- numeric(0)
theta_mle <- numeric(0)
mse_mme <- numeric(0)
mse_mle <- numeric(0)
for (i in n){
  a=1
  for (j in thetas){
    for(k in 1:1000){
      data <- runif(i,0,j)
      theta_mme[k] <- 2*mean(data)
      theta_mle[k] <- max(data)
    }
    mse_mme[a] <- mean((theta_mme-j)^2)
    mse_mle[a] <- mean((theta_mle-j)^2)
    print(paste("Printing MSE_MME for theta = ",j," and n =
", i, " MSE = ",mse_mme[a]))
    print(paste("Printing MSE_MLE for theta = ",j," and n =
", i, " MSE = ",mse_mle[a]))
    a=a+1
  }
  plot(thetas,mse_mme,type = 'l', col="red",lwd=2, ylab="")
  title(main = paste("Plotting graph for MSE vs 'theta' for
n = ",i), ylab = 'Mean-Squared Error')
  lines(thetas,mse_mle,type = 'l', col="blue",lwd=2)
  legend("topleft", legend = c('MME','MLE'), fill =
c('red','blue'))
}
```

Q.2:

(a):

The likelihood function considered is L(θ) = $\prod_{i=1}^{n}\left(\frac{\theta}{xi^{\theta+1}}\right)$

The differentiation technique is used to solve the likelihood function. 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)* . For that 'log' is taken on both sides of the equation

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

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

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

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

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

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

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

$$\hat{\boldsymbol{\theta}} = \frac{\boldsymbol{n}}{\sum_{i=1}^{n}\mathbf{log}(x_i))}$$ **is the resultant estimator of 'θ'**

(b):

In the question, we have been given the sample size n as 5 and we are also given the sample values.

X1 = 21.72
X2 = 14.65
X3 = 50.42
X4 = 28.78
X5 = 11.23

Now we will use the expression derived in the part a to calculate the Maximum Likehood Estimate for θ based on the data given.

```
> data=log(21.7)+log(14.65)+log(50.42)+log(28.78)+log(11.23)
> data
[1] 15.46133
> theta = 5 / data
#5 is the sample size, which is 5 in this question
> theta
[1] 0.3233874
```

If we solve the function derived, we get value for θ:-
MLE = 0.3233874

(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,data){
+   result=0
+   for(i in data){
+     result<- result + (log(par)-(par+1)*log(i) )
+   }
+   return(-result)
+ }
> ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T,
dat=data)
> 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):

The estimated standard error of the maximum likelihood estimates is given by:

```
> # Standard error of the estimation
> std_err<-sqrt(diag(solve(ml.est$hessian)))
> std_err
[1] 0.1446661
```

To calculate the 95% confidence interval, we use the following formula:

```
> #Confidence interval for the estimation
> conf_intr<- ml.est$par +c(-1,1)*qnorm(.975)*std_err
> conf_intr
[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.
data <- c(21.72,14.65,50.42,28.78,11.23)
neg.loglik.fun<-function(par,data){
  result=0
  for(i in data){
    result<- result + (log(par)-(par+1)*log(i) )
  }
  return(-result)
}
ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T,
dat=data)
#get estimated value and other associated results.
ml.est

# Standard error of the estimation
std_err<-sqrt(diag(solve(ml.est$hessian)))
std_err

#Confidence interval for the estimation
conf_intr<- ml.est$par +c(-1,1)*qnorm(.975)*std_err
conf_intr
```