**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 X1, . . . ,Xn 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 ϴMME(hat):**

The first sample moment M1 = X(bar) = ϴ/2

Hence, ϴMME(hat) = 2X(bar).

**Method of Maximum Likelihood estimator ϴMLE(hat):**

ϴMLE(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 ˆ*θ* of a parameter *θ* is defined as

*E{*(ˆ*θ −θ*)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)

1. 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)

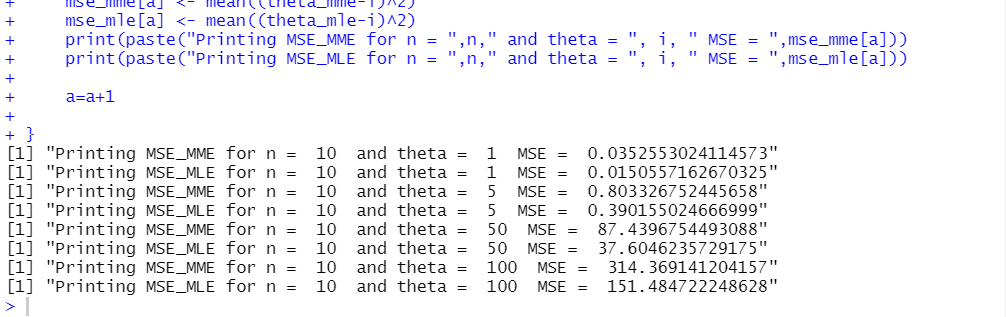
theta\_mle[k] <- max(X)

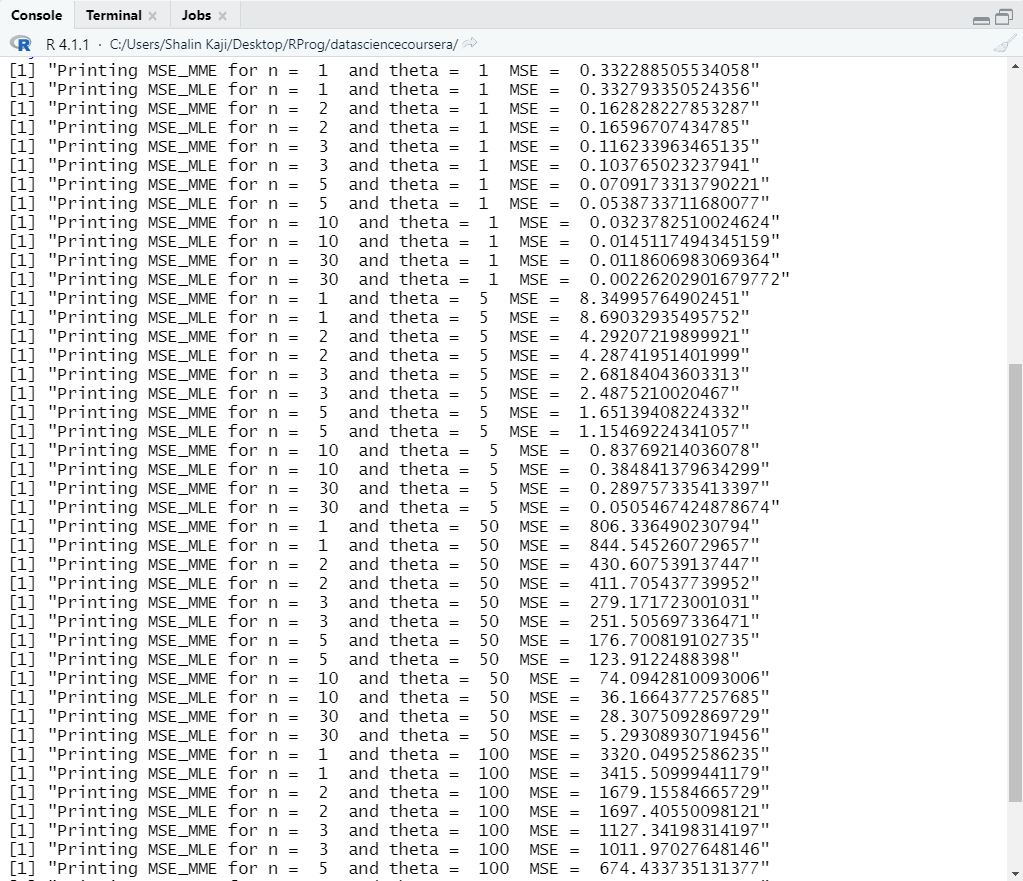
}

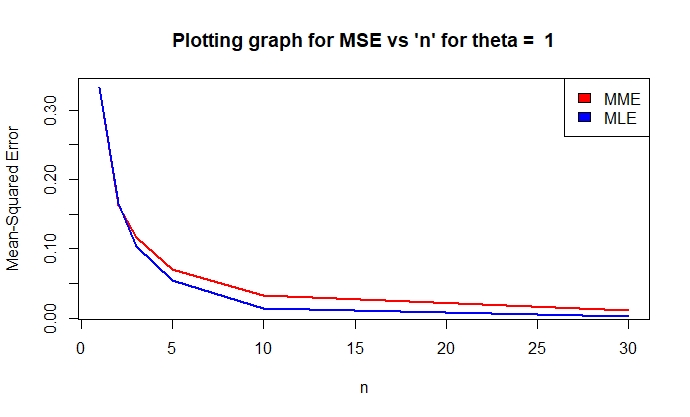
Calculate the difference between the estimate and the true value of the parameter: For each random sample, we calculate the difference between the estimate of the parameter and the true value of the parameter. Then, square the difference. The MSE will be obtained by calculating the average of the squared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

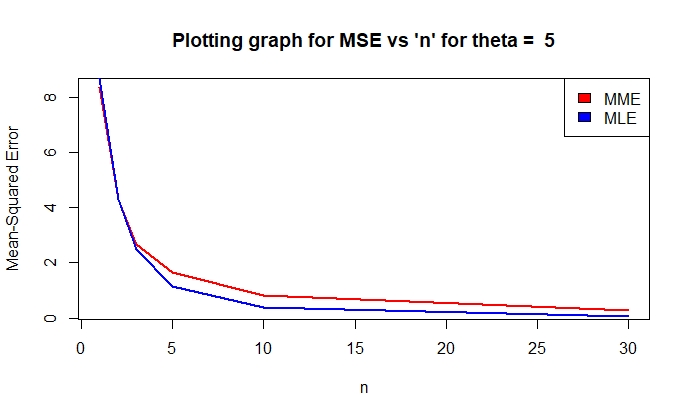
> mse\_mme[a] <- mean((theta\_mme-i)^2)

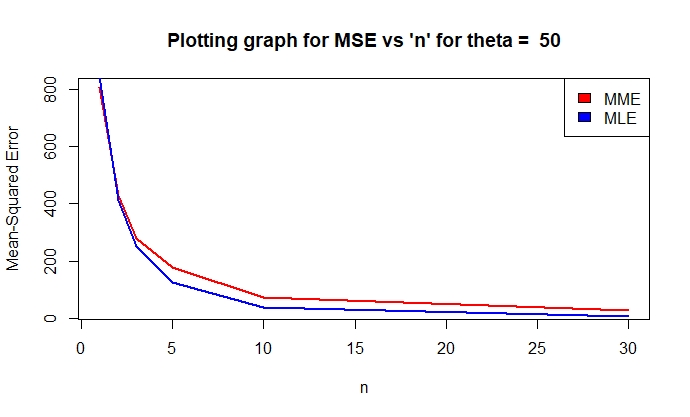
> mse\_mle[a] <- mean((theta\_mle-i)^2)

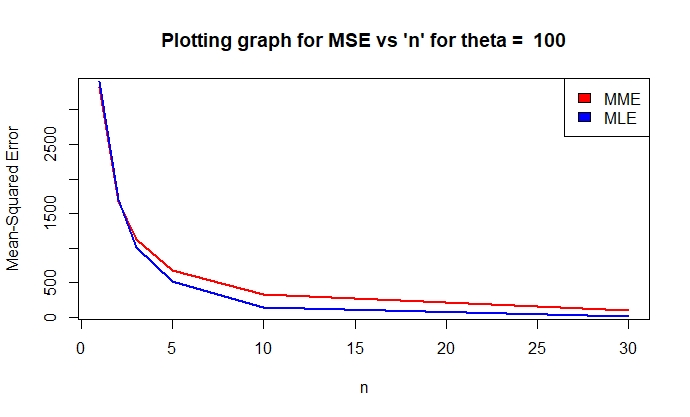
1. Computing the mean squared errors of both ϴMME(hat) and ϴMLE(hat) using Monte-Carlo simulation using N=1000 replications for (n=10, ϴ):
2. Repeating the process in (b) for the other combinations of (n,ϴ) and summarising the results graphically:











1. 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 4th 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')

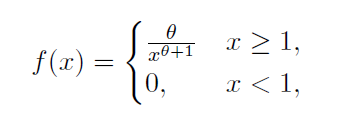
lines(n,mse\_mle,type = 'l', col="blue",lwd=2)

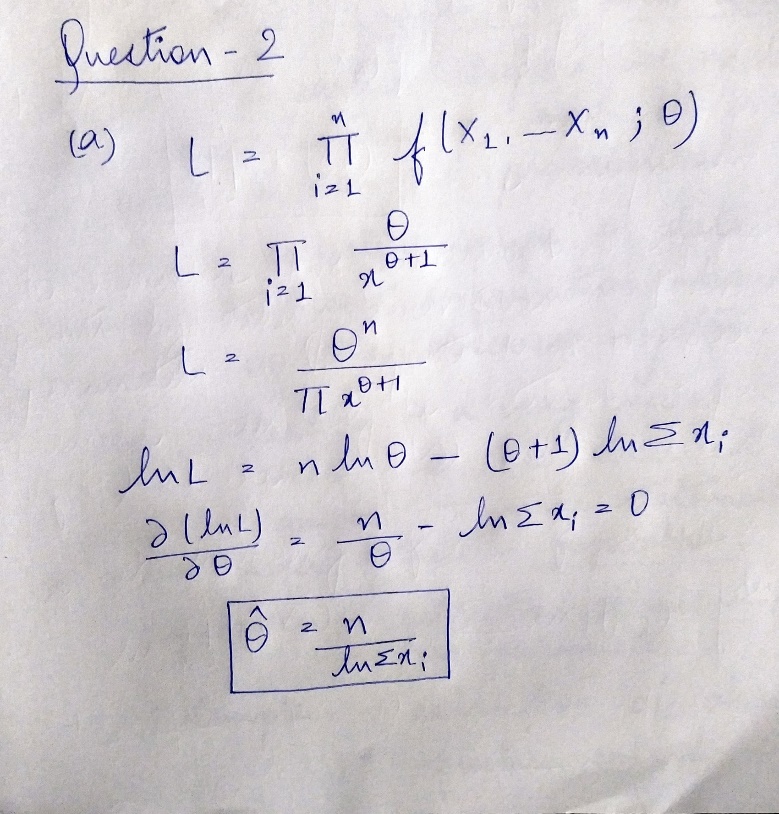
legend("topright", legend = c('MME','MLE'), fill = c('red','blue')) }

}

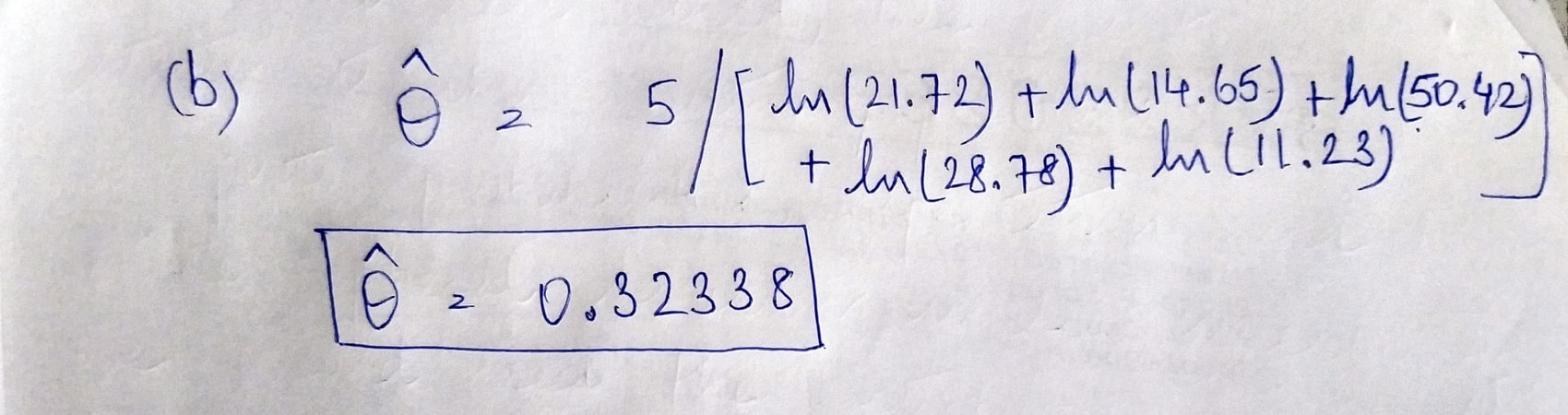
**Question 2:**

1. 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:



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)* .

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



1. 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){

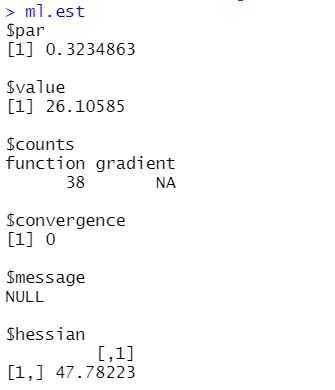
total<- total + (log(par)-(par+1)\*log(i) )

}

return(-total)

}

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





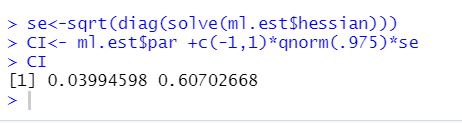
> ml.est <- optim(par=5, fn=neg.loglik.fun, hessian = T, dat=dat)

> ml.est

> se<-sqrt(diag(solve(ml.est$hessian)))

> CI<- ml.est$par +c(-1,1)\*qnorm(.975)\*se

> CI



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