Mini Project # 3

Names of group members: (Group-18)

Deepika Mamidipelly

Preethi Pasunuri

Contribution of each group member

Both group members contributed equally to the inputs for both questions and best is chosen among them to solve these problems. Collaboratively learned R, ran the scripts, and assessed the results. Some of the scripts written by Deepika were analysed and finalized by Preethi and similarly scripts written by Preethi were assessed and finalized by Deepika. Both group members distribute equal amounts of report documentation, which is then integrated into a single final document. Members of the group worked diligently to meet all the project criteria.

Question 1)

Solution

a)

Mean squared error of an estimator θ^ of a parameter θ is defined as $E\{(\theta\text{^} - \theta)^2\}$.

The mean square error (MSE) of an estimator is the difference squared between the values estimated by an estimator (i.e., parameter estimates) and the population parameter.

We must first set the first population parameter and then calculate the estimator value by simulating the sample values.

b)

The maximum likelihood estimator, ˆθ1 = X(n), where X(n) is the maximum of the sample

Therefore, **MLE =maximum(sample)**

The method of moments estimator, ˆθ2 = 2X, where X is the sample mean

Therefore, **MOM= 2*mean(sample)**

The functions **methodOfMoment(n,theta)** and **maxLikelihood (n,theta)** simulates samples from a uniform distribution and then calculates method of moments and maximum likelihood and returns the values.

The function **meanSquareError (n,theta)** calls the **methodOfMoment(n,theta)** and **maxLikelihood(n,theta)** functions 1000 times and calculates the mean squared error using the formula $E\{(\theta\text{^} - \theta)^2\}$ and returns the mean squared errors for both estimators.

We take combination of n and theta as (1,1) and (2,5)

And calculate the meanSquareErrors:

MSE(θ_1) = 0.3356 MSE(θ_2) = 0.3349

MSE(θ_1) = 4.226 MSE(θ_2) = 4.010

Input:

```
methodOfMoment <- function(n,theta) {
  sample = runif(n, min = 0, max = theta);
  return(2*mean(sample));

}

maxLikelihoodEvent <-function(n,theta){
  sample = runif(n, min = 0, max = theta);
  return(max(sample));
}

meanSquareError <- function(n,theta){
  estimateValues=
   replicate(1000,c(methodOfMoment(n,theta),maxLikelihoodEvent(n,theta)));
  estimateValues=  (estimateValues - theta)^2;
  estimateValues.methodOfMoments = estimateValues[c(TRUE,FALSE)];
  estimates.maxLikelihoodEvents = estimateValues[c(FALSE,TRUE)]
   return(c(mean(estimates.maxLikelihoodEvents), mean( estimateValues.methodOfMoments)))
}

meanSquareError(1,1);
meanSquareError(2,5);
```

Output:

```
>
> meanSquareError(1,1);
[1] 0.3327007 0.3349893
> meanSquareError(2,5);
[1] 4.226379 4.010391
```

Rcode:

```
methodOfMoment <- function(n,theta) {
  sample = runif(n, min = 0, max = theta);
  return(2*mean(sample));
}


maxLikelihoodEvent <-function(n,theta){
  sample = runif(n, min = 0, max = theta);
  return(max(sample));
}
meanSquareError <- function(n,theta){
  estimateValues=
   replicate(1000,c(methodOfMoment(n,theta),maxLikelihoodEvent(n,theta)));
  estimateValues=  (estimateValues - theta)^2;
```

```
 estimateValues.methodOfMoments = estimateValues[c(TRUE,FALSE)];
 estimates.maxLikelihoodEvents = estimateValues[c(FALSE,TRUE)]
  return(c(mean(estimates.maxLikelihoodEvents), mean( estimateValues.methodOfMoments)))
}
```

c) Repeating the (b) for remaining combinations of theta and n and graphically plotting them

Input:

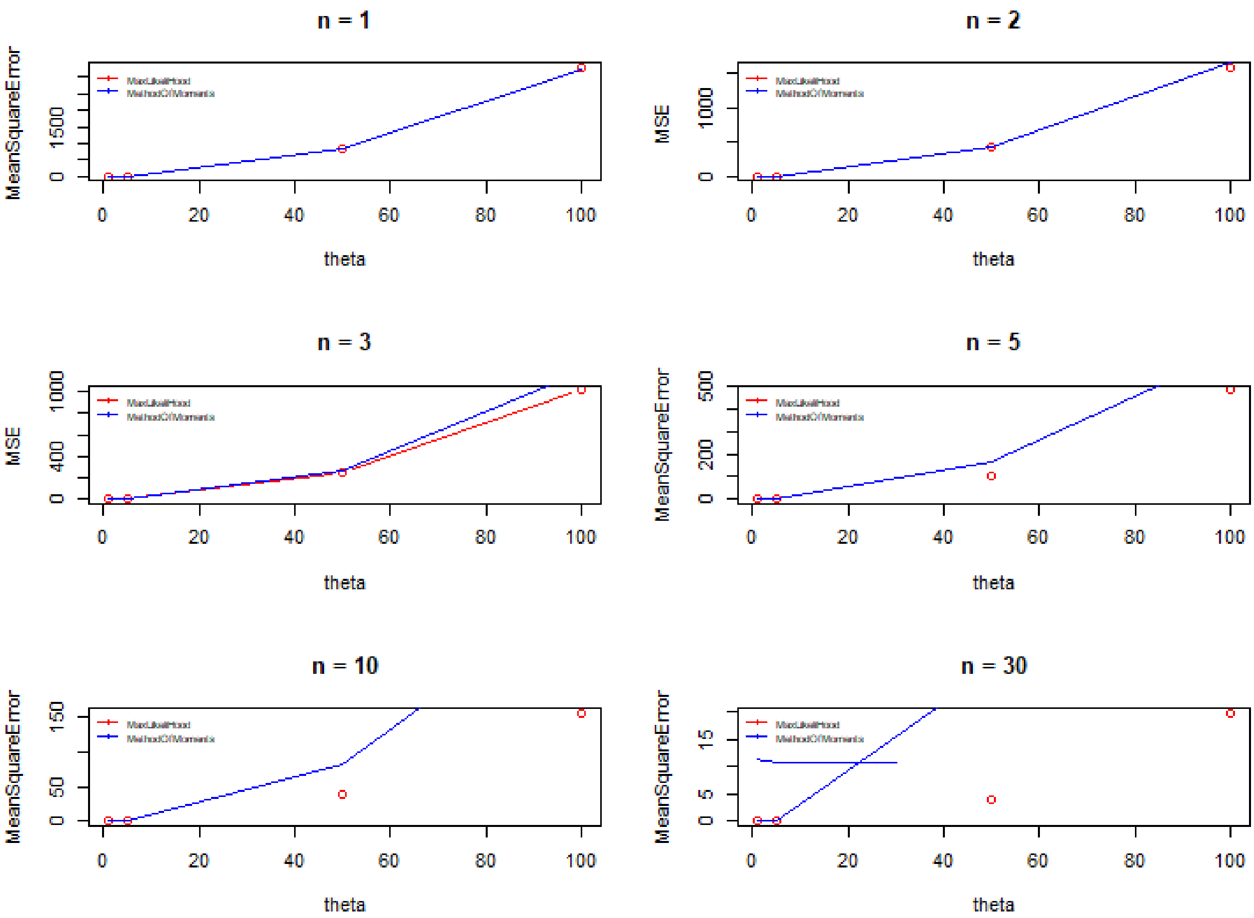
```
#  fixed n value and varying  θ -graph slots
 par(mfrow=c(3,2))
plot(c(1,5,50,100), c(meanSquareError1_1[1],meanSquareError1_5[1], meanSquareError1_50[1],
                      meanSquareError1_100[1]),
       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 1")
lines(c(1,5,50,100), c(meanSquareError1_1[2],meanSquareError1_5[2], meanSquareError1_50[2],
                      meanSquareError1_100[2]),
        col = 'blue')
legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError2_1[1],meanSquareError2_5[1], meanSquareError2_50[1], meanSquareError2_100[1]),
       xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 2")
lines(c(1,5,50,100), c(meanSquareError2_1[2],meanSquareError2_5[2], meanSquareError2_50[2],
                      meanSquareError2_100[2]),
         col = 'blue')
 legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =

                     c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError3_1[1],meanSquareError3_5[1], meanSquareError3_50[1],
                      meanSquareError3_100[1]), type="b",
       xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 3")
lines(c(1,5,50,100), c(meanSquareError3_1[2],meanSquareError3_5[2], meanSquareError3_50[2],
                      meanSquareError3_100[2]),
         col = 'blue')
legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')



plot(c(1,5,50,100), c(meanSquareError5_1[1],meanSquareError5_5[1], meanSquareError5_50[1],
                      meanSquareError5_100[1]),
       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 5")
lines(c(1,5,50,100), c(meanSquareError5_1[2],meanSquareError5_5[2], meanSquareError5_50[2],
                      meanSquareError5_100[2]),
       col = 'blue')
legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError10_1[1],meanSquareError10_5[1], meanSquareError10_50[1],
                      meanSquareError10_100[1]),
       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 10")
lines(c(1,5,50,100), c(meanSquareError10_1[2],meanSquareError10_5[2], meanSquareError10_50[2],
                      meanSquareError10_100[2]),
         col = 'blue')
```

47:44 (Top Level)

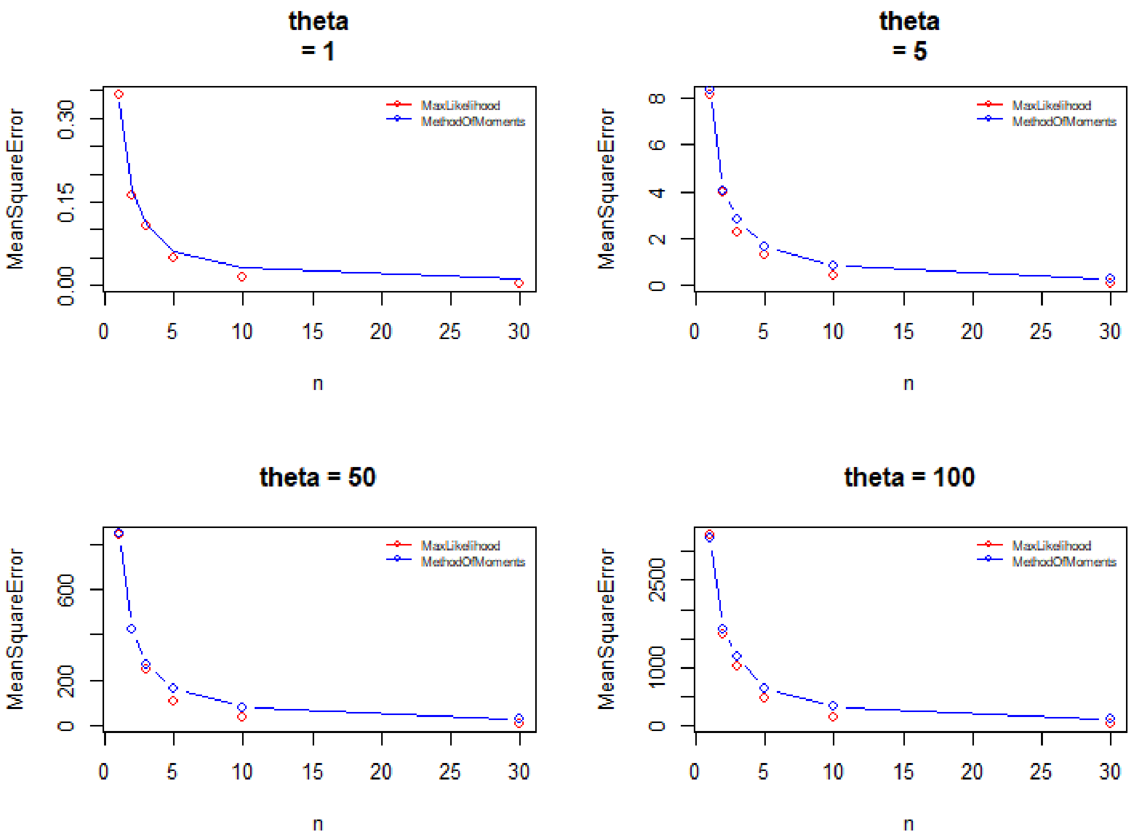
```r
#  fixed θ value and varying n values-graph slots
par(mfrow=c(2,2))
plot(c(1,2,3,5,10,30), c(meanSquareError1_1[1],meanSquareError2_1[1], meanSquareError3_1[1], meanSquareError5_1[1],
                         meanSquareError10_1[1], meanSquareError30_1[1]), ylab = 'MeanSquareError', xlab = 'n', col = 'red', main = "thet
= 1")
lines(c(1,2,3,5,10,30), c(meanSquareError1_1[2],meanSquareError2_1[2], meanSquareError3_1[2], meanSquareError5_1[2],
                          meanSquareError10_1[2], meanSquareError30_1[2]),  col = 'blue')
legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')

plot(c(1,2,3,5,10,30), c(meanSquareError1_5[1],meanSquareError2_5[1], meanSquareError3_5[1], meanSquareError5_5[1],
                         meanSquareError10_5[1], meanSquareError30_5[1]), ylab = 'MeanSquareError', xlab = 'n', col = 'red', main = "thet
= 5")
 lines(c(1,2,3,5,10,30), c(meanSquareError1_5[2],meanSquareError2_5[2], meanSquareError3_5[2], meanSquareError5_5[2],
                           meanSquareError10_5[2], meanSquareError30_5[2]), type="b", col = 'blue')
legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,2,3,5,10,30), c(meanSquareError1_50[1],meanSquareError2_50[1], meanSquareError3_50[1], meanSquareError5_50[1],
                         meanSquareError10_50[1], meanSquareError30_50[1]), ylab = 'MeanSquareError', xlab = 'n', col = 'red', main =
         "theta = 50")
lines(c(1,2,3,5,10,30), c(meanSquareError1_50[2],meanSquareError2_50[2], meanSquareError3_50[2], meanSquareError5_50[2],
                          meanSquareError10_50[2], meanSquareError30_50[2]), type="b", col = 'blue')
legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,2,3,5,10,30), c(meanSquareError1_100[1],meanSquareError2_100[1], meanSquareError3_100[1], meanSquareError5_100[1],
                         meanSquareError10_100[1], meanSquareError30_100[1]),  ylab = 'MeanSquareError', xlab = 'n', col = 'red', main =
         "theta = 100")
lines(c(1,2,3,5,10,30), c(meanSquareError1_100[2],meanSquareError2_100[2], meanSquareError3_100[2], meanSquareError5_100[2],
                          meanSquareError10_100[2], meanSquareError30_100[2]), type="b", col = 'blue')
legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')
```

Output:

Graph1:**Mean Squared Errors with MLE and MOM with fixed theta and varying n**



Graph2:**Mean Squared Errors with MLE and MOM with fixed n and varying theta**



Rcode:

```
meanSquareError(1,1);
meanSquareError(2,5);
meanSquareError1_5=meanSquareError(1,5);
meanSquareError1_50=meanSquareError(1,50);
meanSquareError1_100=meanSquareError(1,100);
meanSquareError2_1=meanSquareError(2,1);
meanSquareError2_5=meanSquareError(2,5);
meanSquareError2_50=meanSquareError(2,50);
meanSquareError2_100=meanSquareError(2,100);
meanSquareError3_1=meanSquareError(3,1);
meanSquareError3_5=meanSquareError(3,5);
meanSquareError3_50=meanSquareError(3,50);
meanSquareError3_100=meanSquareError(3,100);
```

```
meanSquareError5_1=meanSquareError(5,1);

meanSquareError5_5=meanSquareError(5,5);

meanSquareError5_50=meanSquareError(5,50);

meanSquareError5_100=meanSquareError(5,100);

meanSquareError10_1=meanSquareError(10,1);

meanSquareError10_5=meanSquareError(10,5);

meanSquareError10_50=meanSquareError(10,50);

meanSquareError10_100=meanSquareError(10,100);

meanSquareError30_1=meanSquareError(30,1);

meanSquareError30_5=meanSquareError(30,5);

meanSquareError30_50=meanSquareError(30,50);

meanSquareError30_100=meanSquareError(30,100);


#  fixed n value and varying θ-graph slots

 par(mfrow=c(3,2))

plot(c(1,5,50,100), c(meanSquareError1_1[1],meanSquareError1_5[1], meanSquareError1_50[1],

                    meanSquareError1_100[1]),

       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 1")

lines(c(1,5,50,100), c(meanSquareError1_1[2],meanSquareError1_5[2],
meanSquareError1_50[2],

                     meanSquareError1_100[2]),

        col = 'blue')

legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError2_1[1],meanSquareError2_5[1], meanSquareError2_50[1],
meanSquareError2_100[1]),

       xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 2")

lines(c(1,5,50,100), c(meanSquareError2_1[2],meanSquareError2_5[2],
meanSquareError2_50[2],

                     meanSquareError2_100[2]),

         col = 'blue')
```

```
 legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =    c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError3_1[1],meanSquareError3_5[1], meanSquareError3_50[1],

                     meanSquareError3_100[1]), type="b",

       xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 3")

lines(c(1,5,50,100), c(meanSquareError3_1[2],meanSquareError3_5[2],
meanSquareError3_50[2],

                      meanSquareError3_100[2]),

         col = 'blue')

legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError5_1[1],meanSquareError5_5[1], meanSquareError5_50[1],

                     meanSquareError5_100[1]),

       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 5")

lines(c(1,5,50,100), c(meanSquareError5_1[2],meanSquareError5_5[2],
meanSquareError5_50[2],

                      meanSquareError5_100[2]),

       col = 'blue')

legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

           c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError10_1[1],meanSquareError10_5[1],
meanSquareError10_50[1],

                     meanSquareError10_100[1]),

       xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 10")

lines(c(1,5,50,100), c(meanSquareError10_1[2],meanSquareError10_5[2],
meanSquareError10_50[2],

                      meanSquareError10_100[2]),

         col = 'blue')
```

```
legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
      c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,5,50,100), c(meanSquareError30_1[1],meanSquareError30_5[1], meanSquareError30_50[1],
              meanSquareError30_100[1]),
    xlab = 'theta', ylab = 'MeanSquareError', col = 'red', main = "n = 30")
lines(c(1,5,50,100), c(meanSquareError30_1[2],meanSquareError30_5[2],meanSquareError30_50[2], meanSquareError30_100[2]),
     col = 'blue')
legend("topleft", legend = c("MaxLikeliHood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
      c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


#  fixed θ value and varying n values-graph slots
par(mfrow=c(2,2))
plot(c(1,2,3,5,10,30), c(meanSquareError1_1[1],meanSquareError2_1[1], meanSquareError3_1[1], meanSquareError5_1[1],
                meanSquareError10_1[1], meanSquareError30_1[1]), ylab = 'MeanSquareError', xlab = 'n', col = 'red', main = "theta
= 1")
lines(c(1,2,3,5,10,30), c(meanSquareError1_1[2],meanSquareError2_1[2], meanSquareError3_1[2], meanSquareError5_1[2],
                 meanSquareError10_1[2], meanSquareError30_1[2]),  col = 'blue')
legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'), text.col =
      c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,2,3,5,10,30), c(meanSquareError1_5[1],meanSquareError2_5[1], meanSquareError3_5[1], meanSquareError5_5[1],
                meanSquareError10_5[1], meanSquareError30_5[1]), ylab = 'MeanSquareError', xlab = 'n', col = 'red', main = "theta
= 5")
```

```
 lines(c(1,2,3,5,10,30), c(meanSquareError1_5[2],meanSquareError2_5[2],
meanSquareError3_5[2], meanSquareError5_5[2],

                    meanSquareError10_5[2], meanSquareError30_5[2]), type="b", col = 'blue')

legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

        c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')



plot(c(1,2,3,5,10,30), c(meanSquareError1_50[1],meanSquareError2_50[1],
meanSquareError3_50[1], meanSquareError5_50[1],

                   meanSquareError10_50[1], meanSquareError30_50[1]), ylab =
'MeanSquareError', xlab = 'n', col = 'red', main =

       "theta = 50")

lines(c(1,2,3,5,10,30), c(meanSquareError1_50[2],meanSquareError2_50[2],
meanSquareError3_50[2], meanSquareError5_50[2],

                   meanSquareError10_50[2], meanSquareError30_50[2]), type="b", col = 'blue')

legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

        c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')


plot(c(1,2,3,5,10,30), c(meanSquareError1_100[1],meanSquareError2_100[1],
meanSquareError3_100[1], meanSquareError5_100[1],

                   meanSquareError10_100[1], meanSquareError30_100[1]),  ylab =
'MeanSquareError', xlab = 'n', col = 'red', main =

       "theta = 100")

lines(c(1,2,3,5,10,30), c(meanSquareError1_100[2],meanSquareError2_100[2],
meanSquareError3_100[2], meanSquareError5_100[2],

                   meanSquareError10_100[2], meanSquareError30_100[2]), type="b", col = 'blue')

legend("topright", legend = c("MaxLikelihood", "MethodOfMoments"), col = c('red', 'blue'),
text.col =

        c('black','black'),lty = 1, pch = 1, inset =0.01, ncol = 1, cex = 0.6, bty = 'n')
```

d) The second graph shows that regardless of what value of theta is fixed, the resulting graphs are extremely similar. As a result, it's possible to derive that the estimator isn't dependent on the value of theta. The plotted values of mean squared errors varying with theta with fixed n are shown in Graph 1. It is clear that the Method of Moments estimator can be used for small values of n (=1, 2, 3). The Maximum Likelihood Estimator is better when the n (=5, 10, 30) number grows. MLE is better for bigger n numbers, and as n grows larger, MLE becomes the preferred option. In compared to MOM, MLE is recommended since the mean squared error is lower for the same value of n.

Question 2)

<u>Solution:</u>

a) Take likelihood function as

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

Taking log on both sides and the solving then we would get it,

$$\begin{aligned}\log(L(\theta)) &= log\left(\prod_{i=1}^{n}\left(\frac{\theta}{x_i^{\theta+1}}\right)\right)\\ &= log\left(\theta^n \times \prod_{i=1}^{n}\frac{1}{x_i^{\theta+1}}\right)\\ &= nlog\theta\ + \sum_{i=1}^{n} log\left(x_i^{-\theta-1}\right)\\ &= nlog\theta - (\theta+1)\sum_{i=1}^{n} logx_i\\ &= nlog\theta - \theta\sum_{i=1}^{n} logx_i - \sum_{i=1}^{n} logx_i\end{aligned}$$

Now partially differentiate the above obtained equation with θ

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

Equate the above obtained equation to 0, then the result obtained is:

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

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

$$\hat{\theta}_{MLE} = \frac{n}{\sum_{i=1}^{n} logx_i}$$

b) Given n = 5 ;

Sample values are x1 = 21.72, x2 = 14.65,

x3 = 50.42, x4 = 28.78, x5 = 11.23

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

Substituting the sample values in the above equation, we get:

$$\hat{\theta}_{MLE} = \frac{5}{\sum_{i=1}^{5} \log(xi)}$$

$$\hat{\theta}_{MLE} = \frac{5}{\log(x1)+\log(x2)+\log(x3)+\log(x4)+\log(x5)}$$

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

log(a)+log(b) = log(ab)

$$\hat{\theta}_{MLE} = \frac{5}{\log(21.72*14.65*50.42*28.78*11.23)}$$

$$\hat{\theta}_{MLE} = \frac{5}{\log(5137517.08)}$$

$$\hat{\theta}_{MLE} = \frac{5}{15.45}$$

$\hat{\theta}_{MLE}$ =0.3236

c) In R, there is a feature that allows you to minimize any function.

We need to estimate by numerically maximizing the log-likelihood function.

We need maximizing to happen in this scenario.

As a result, the function's negative must be maximized. As a result, in R, the function is negated and then minimized.

The syntax of the function used in R is as follows*: optim (par, fn, ...)*

This function minimizes the function given to it.

par : initial values for the parameters to be optimized over fn = the function to be minimized

The estimated value is 0.3236, as shown by the R code output.

Input:

```
negetive.loglikelihood.fn <- function(par, data) {
  result = -(length(data) * log(par) - (par + 1) * sum(log(data)) )
}
x <- c(21.42, 14.65, 50.42, 28.78, 11.23)
mle <- optim(par=0.926,fn=negetive.loglikelihood.fn,method="L-BFGS-B",hessian=TRUE, lower=0.01, data=x)
print(mle)

se <- (1/mle$hessian)^0.5
se
```

Output:

```
R 4.1.1 · ~/
+   result = -(length(data) * log(par) - (par + 1) * sum(log(data)) )
+ }
> x <- c(21.42, 14.65, 50.42, 28.78, 11.23)
> mle <- optim(par=0.926,fn=negetive.loglikelihood.fn,method="L-BFGS-B",hessian=TRUE, lower=0.01, data=x)
> print(mle)
$par
[1] 0.3236796
```

Rcode:

```
negetive.loglikelihood.fn <- function(par, data) {
result = -(length(data) * log(par) - (par + 1) * sum(log(data)) )
}
x <- c(21.42, 14.65, 50.42, 28.78, 11.23)
mle <- optim(par=0.926,fn=negetive.loglikelihood.fn,method="L-BFGS-B",hessian=TRUE,
lower=0.01, data=x)
print(mle)
```

d) According to large sample properties of MLE $\hat{\theta}$ of θ

If θ is scalar. Then, $SE(\hat{\theta}) \approx \sqrt{\hat{I}} - 1$

Where $\hat{I}$ is a hessian function.

From R code, we get value of standard error as 0.1447525

Given $1\text{-}\alpha = 0.95$

α = 1-0.95

α=0.05

and $\frac{\alpha}{2} = \frac{0.05}{2} = 0.025$

so 1- α=0.975

The confidence interval formula is given by:θˆ =$\boldsymbol{z_{\frac{\alpha}{2}}}$***SE(θˆ)**

In R, we use **qnorm** function to get the $\boldsymbol{z_{\frac{\alpha}{2}}}$ value.

From R we get confidence intervals as (0.03996463,0.60739461).

So, we know that out of the 100 trials to get the population estimated, the true estimate value lies in the within the interval 95% of the times.

Input:

```
> se <- (1/mle$hessian)^0.5
> se
          [,1]
[1,] 0.1447525
> 
```

Output:

```
> confidence_interval
[1] 0.03996463 0.60739461
> 
```

Rcode:

```
se <- (1/mle$hessian)^0.5
se

confidence_interval<-mle$par+c(-1,1)*1.96*se
confidence_interval
```