Mini Project 3

Names of group members: Karan Risbud(KSR190005), Shubham Vartak(SXV200115)

Contribution of each group member: Both the Project group members worked together on the project. Collaborated to solve the problem and implementation of R programming.

Q1. (8 points) Suppose 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. In the class, we have discussed two estimators for θ — the maximum likelihood estimator, ˆθ1 = X(n), where X(n) is the maximum of the sample, and the method of moments estimator, ˆθ2 = 2X, where X is the sample mean. The goal of this exercise is to compare the mean squared errors of the two estimators to determine which estimator is better. Recall that the mean squared error of an estimator ˆθ of a parameter θ is defined as E{(ˆθ − θ) 2}. For the comparison, we will focus on n = 1, 2, 3, 5, 10, 30 and θ = 1, 5, 50, 100.

Q1a) Explain how you will compute the mean squared error of an estimator using Monte Carlo simulation.

1. First, we set the population parameter theta.
2. Now we generate samples from the population and estimate theta hat
3. Mean squared error is the square of the difference between theta and theta hat.

Q1b) For a given combination of (n, θ), compute the mean squared errors of both ˆθ1 and ˆθ2 using Monte Carlo simulation with N = 1000 replications. Be sure to compute both estimates from the same data.

Explanation:

Here function "estimator" simulates n uniform samples using runif function. Further, it calculates ˆθ1 which is maximum likelihood estimator and ˆθ2 which is method of moments estimator. "estimator" function returns an array of ˆθ1 and ˆθ2.

The function "simulations" is used to call "estimator" function 1000 times and computes mean squared errors for both ˆθ1 and ˆθ2 and returns it in an array.

R-code:

```
estimator = function(n,theta)
{
  generator = runif(n,min=0,max=theta)
  mle = max(generator)
  mme = 2 * mean(generator)
  return (c(mle,mme))
}

simulations = function(n,theta)
{
  theta.hat = replicate(1000,estimator(n,theta))
  mse = (theta.hat - theta)^2
  theta.hat1.mse= mean(mse[1,])   #mle
```

```
  theta.hat2.mse = mean(mse[2,])  #mme
  return (c(theta.hat1.mse,theta.hat2.mse))

}
n=1
theta = 1
estimators = simulations(n,theta)
estimators
```

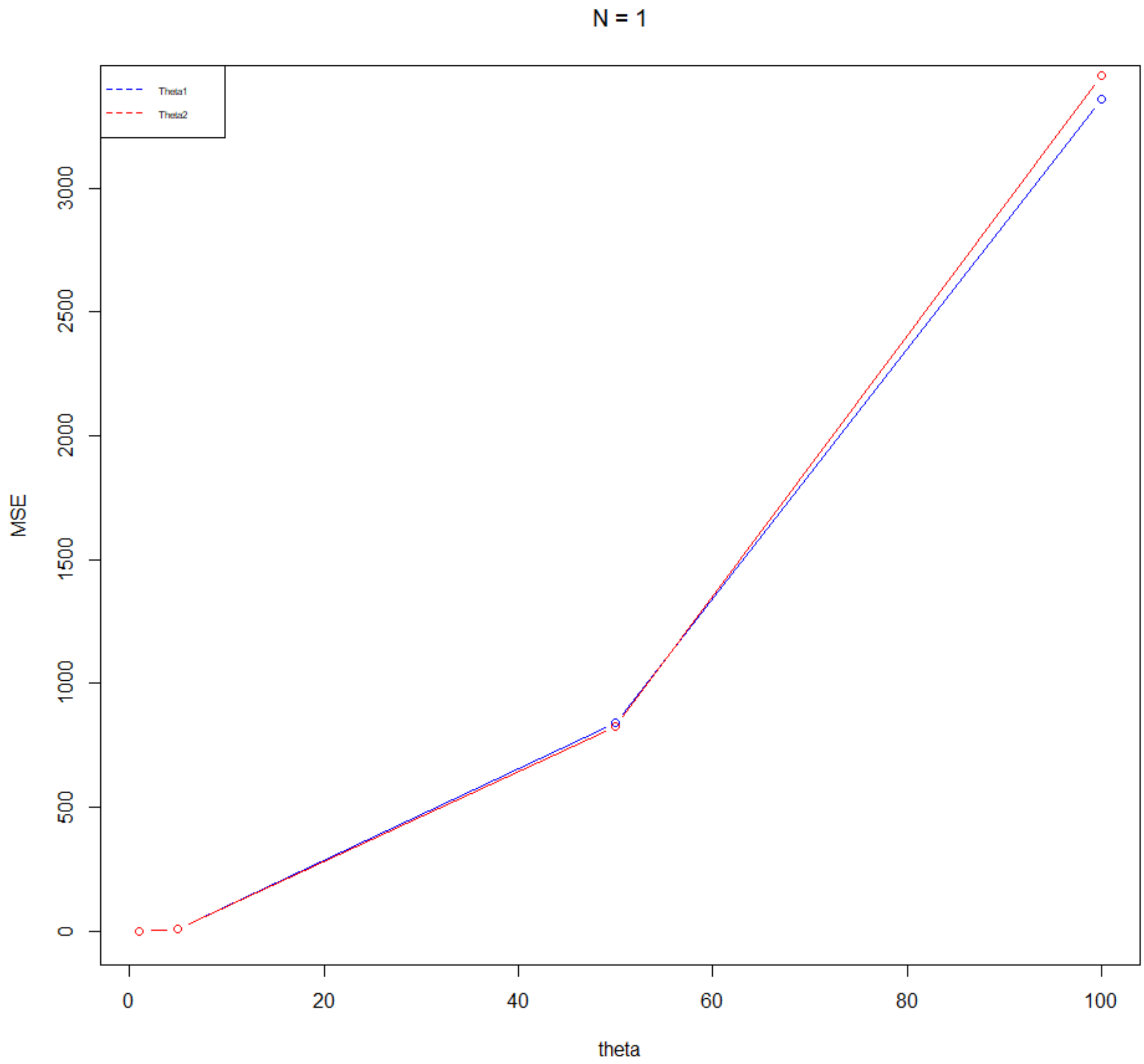
Output:

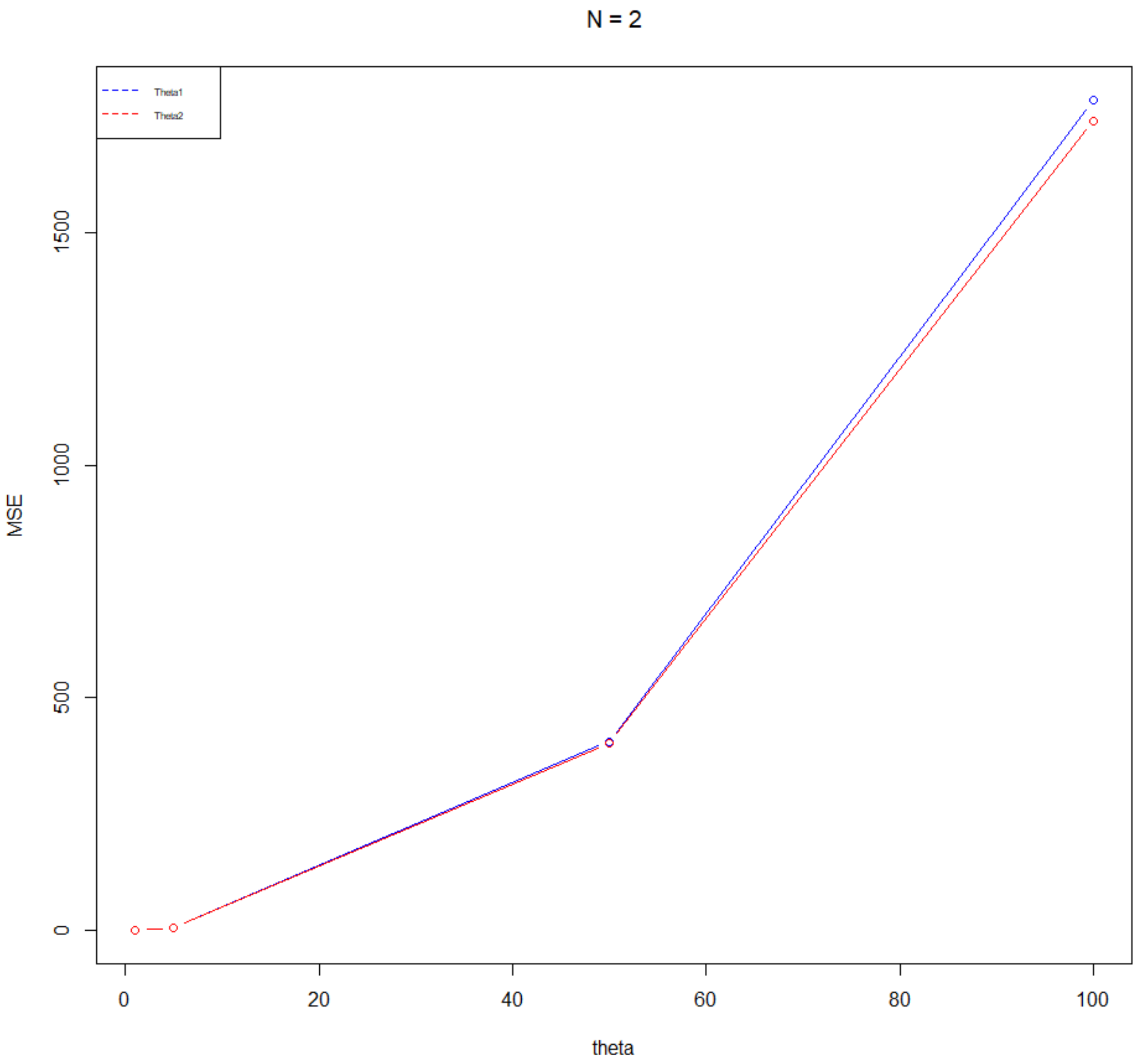
```
> estimators
[1] 0.3379286 0.3395652
>
```

1c) Repeat (b) for the remaining combinations of (n, θ). Summarize your results graphically.

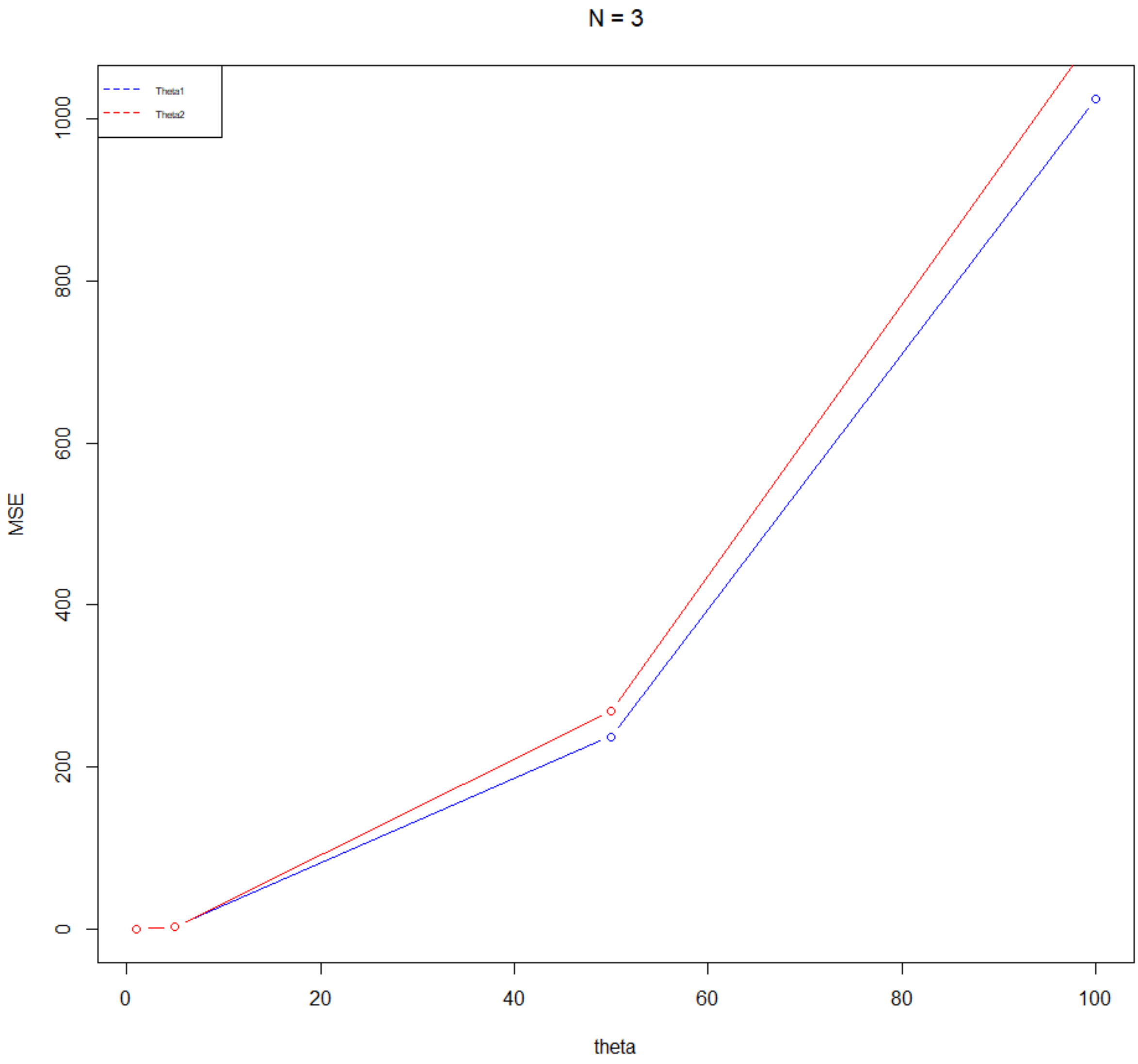
R code:

```
n.values = c(1,2,3,5,10,30)
theta.values = c(1,5,50,100)
k=0
mse.theta1 = c(0,0,0,0)
mse.theta2 = c(0,0,0,0)
for(i in n.values)
{
  k=1
  for(j in theta.values)
  {
    estimators = simulations(i,j)
    mse.theta1[k] = estimators[1]
    mse.theta2[k] = estimators[2]
    k=k+1
  }
  plot(theta.values,mse.theta1,xlab = 'theta',ylab = 'MSE',main=bquote(paste("N = ", .(i))),
      type = 'b',col='blue')
  lines(theta.values,mse.theta2,col='red',type = 'b')
  legend("topleft",legend=c("Theta1","Theta2"),col=c('blue','red'),cex = 0.5,lty=c(2,2),merge = TRUE)
}
```

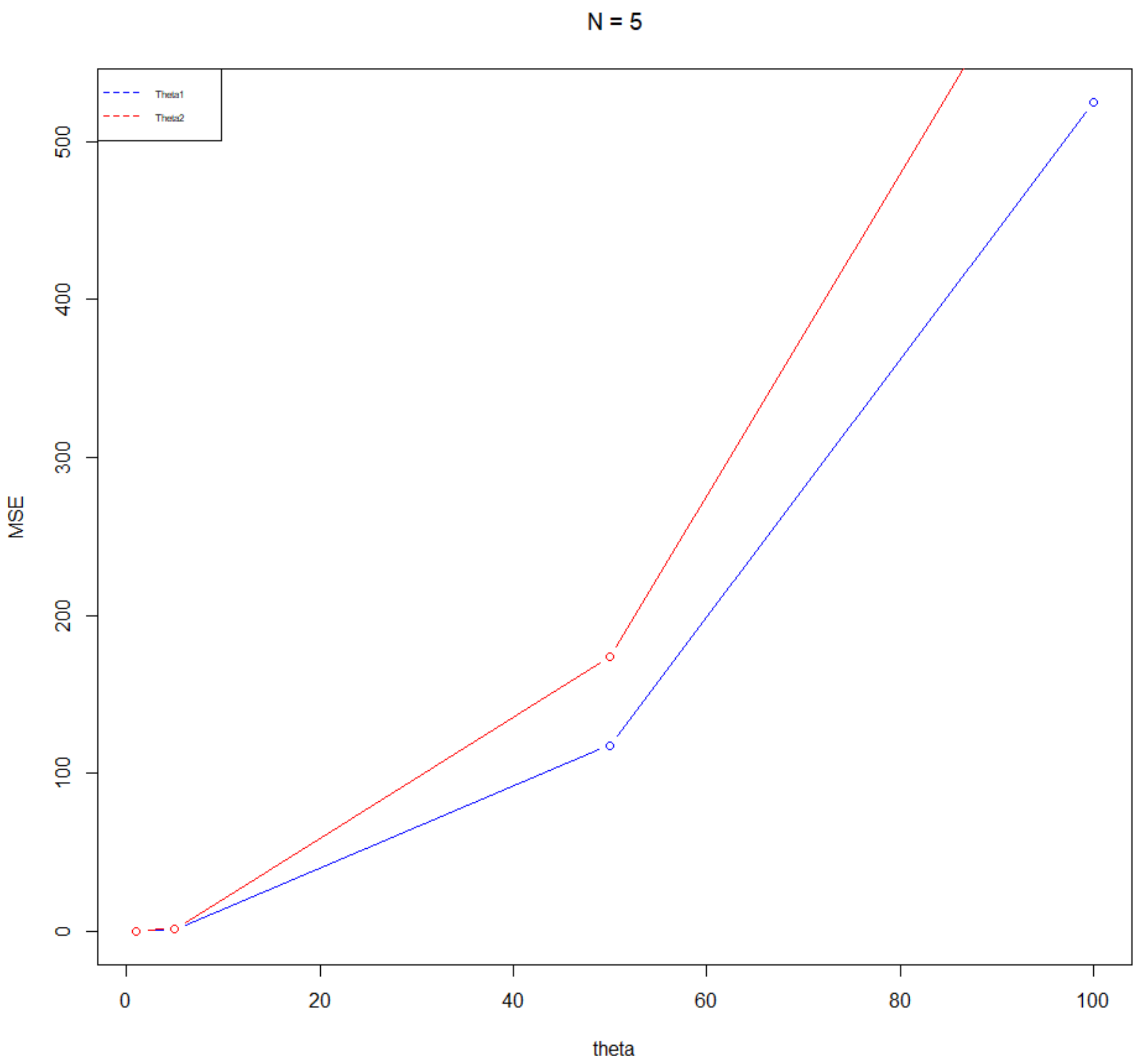

N = 1
Theta1
Theta2
MSE
theta
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100



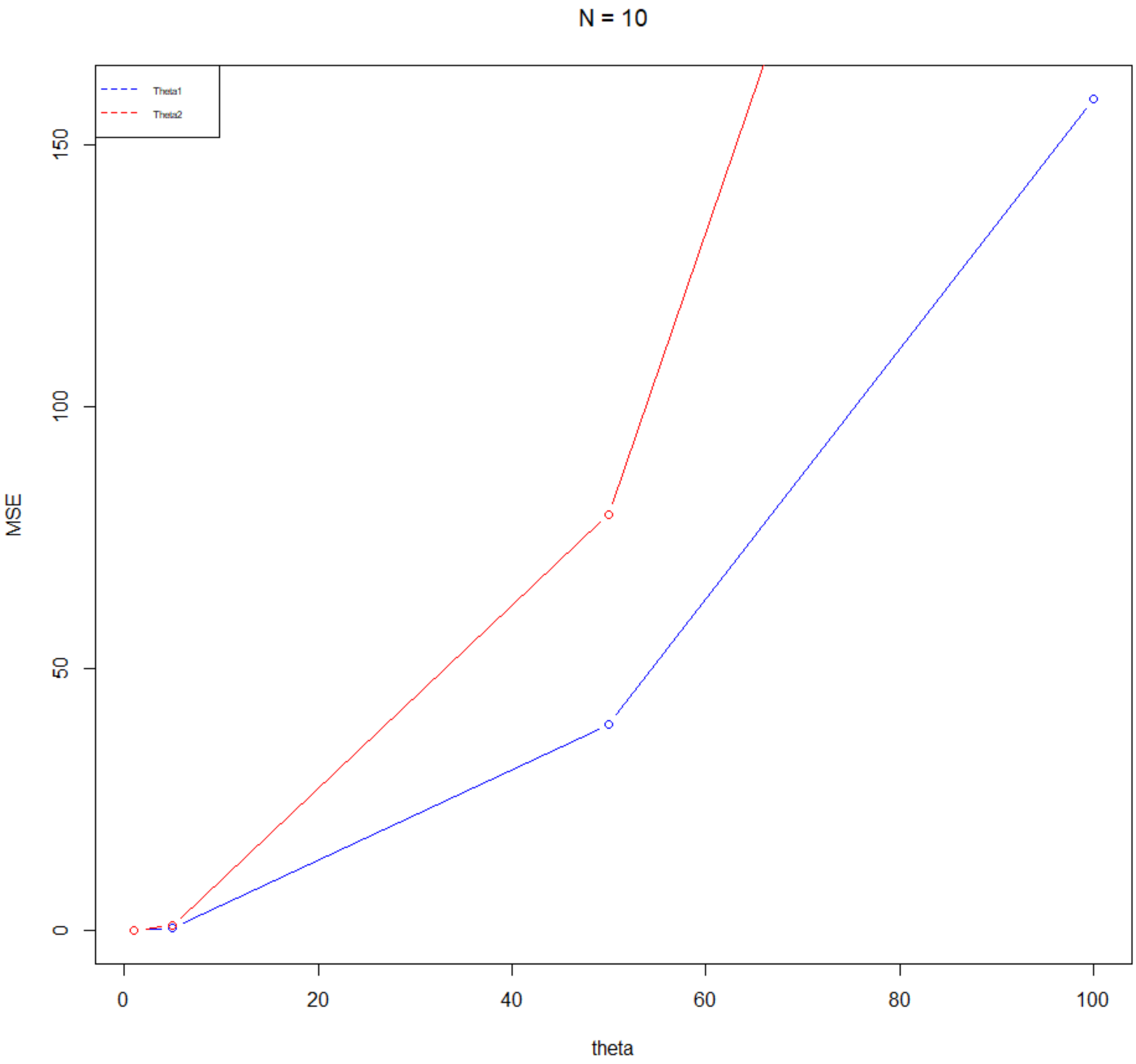
N = 2
Theta1
Theta2
MSE
0
500
1000
1500
0
20
40
60
80
100
theta

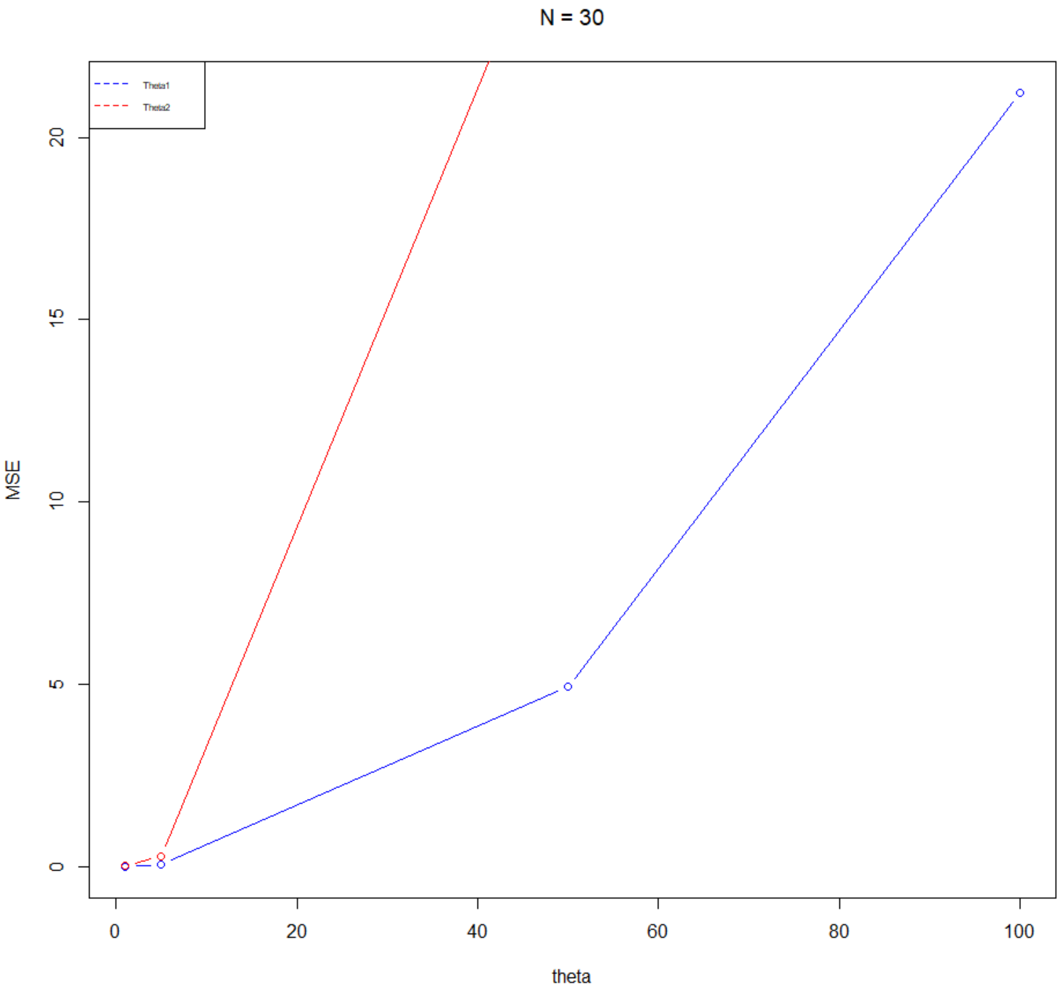


N = 3
Theta1
Theta2
MSE
theta
0
200
400
600
800
1000
0
20
40
60
80
100



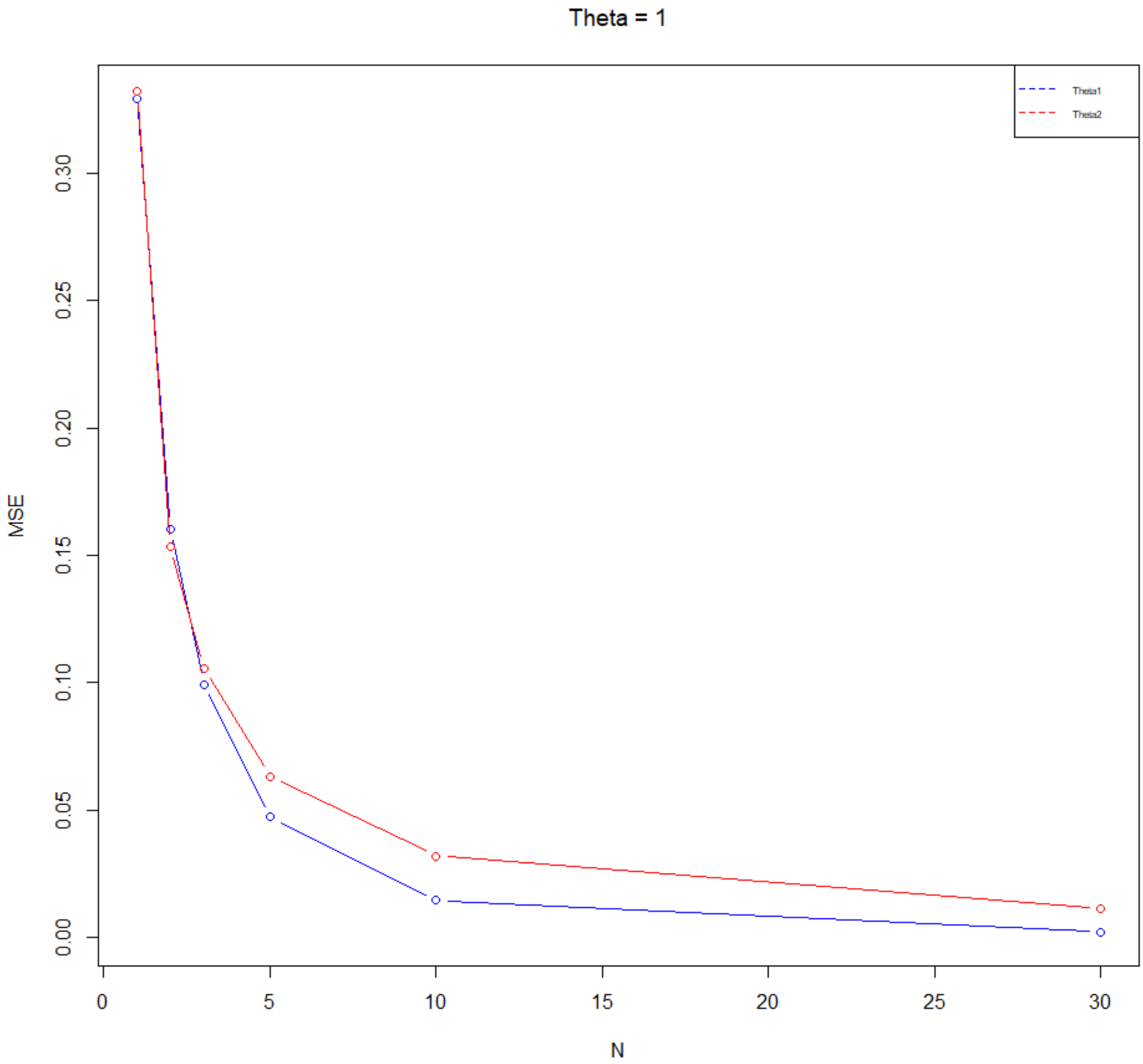
N = 5
Theta1
Theta2
MSE
theta
0
100
200
300
400
500
0
20
40
60
80
100


N = 10
Theta1
Theta2
MSE
0
50
100
150
0
20
40
60
80
100
theta

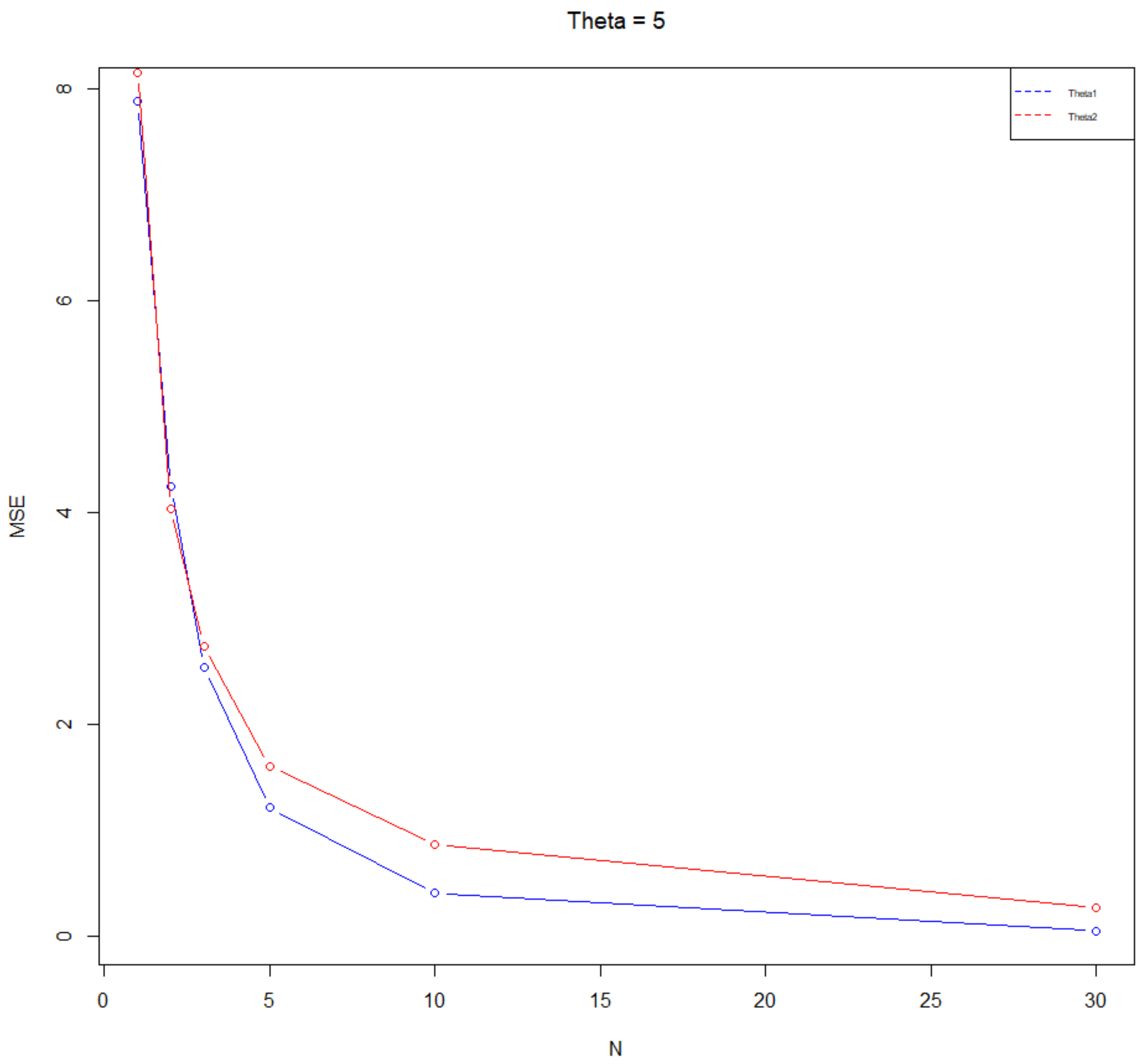


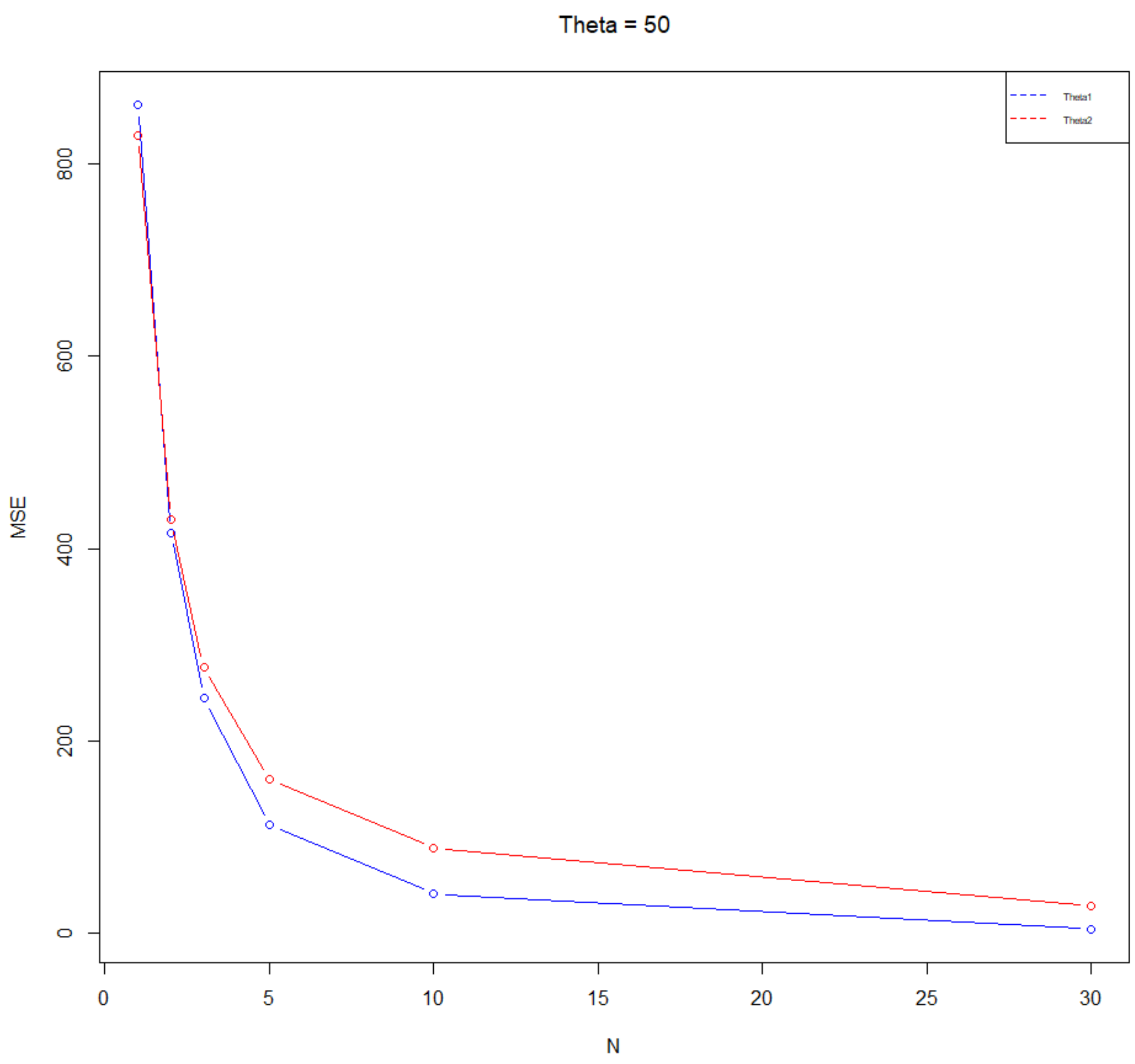

```
#keeping theta constant.
n.values = c(1,2,3,5,10,30)
theta.values = c(1,5,50,100)
k=0
mse.theta1 = c(0,0,0,0,0,0)
mse.theta2 = c(0,0,0,0,0,0)
for(i in theta.values)
{
 k=1
 for(j in n.values)
 {
  estimators = simulations(j,i)
  mse.theta1[k] = estimators[1]
  mse.theta2[k] = estimators[2]
  k=k+1
 }

 plot(n.values,mse.theta1,xlab = 'theta',ylab = 'MSE',main=bquote(paste("Theta = ", .(i))),
    type = 'b',col='blue')
 lines(n.values,mse.theta2,col='red',type="b")
 legend("topright",legend=c("Theta1","Theta2"),col=c('blue','red'),cex = 0.5,lty=c(2,2),merge =
TRUE)
}
```

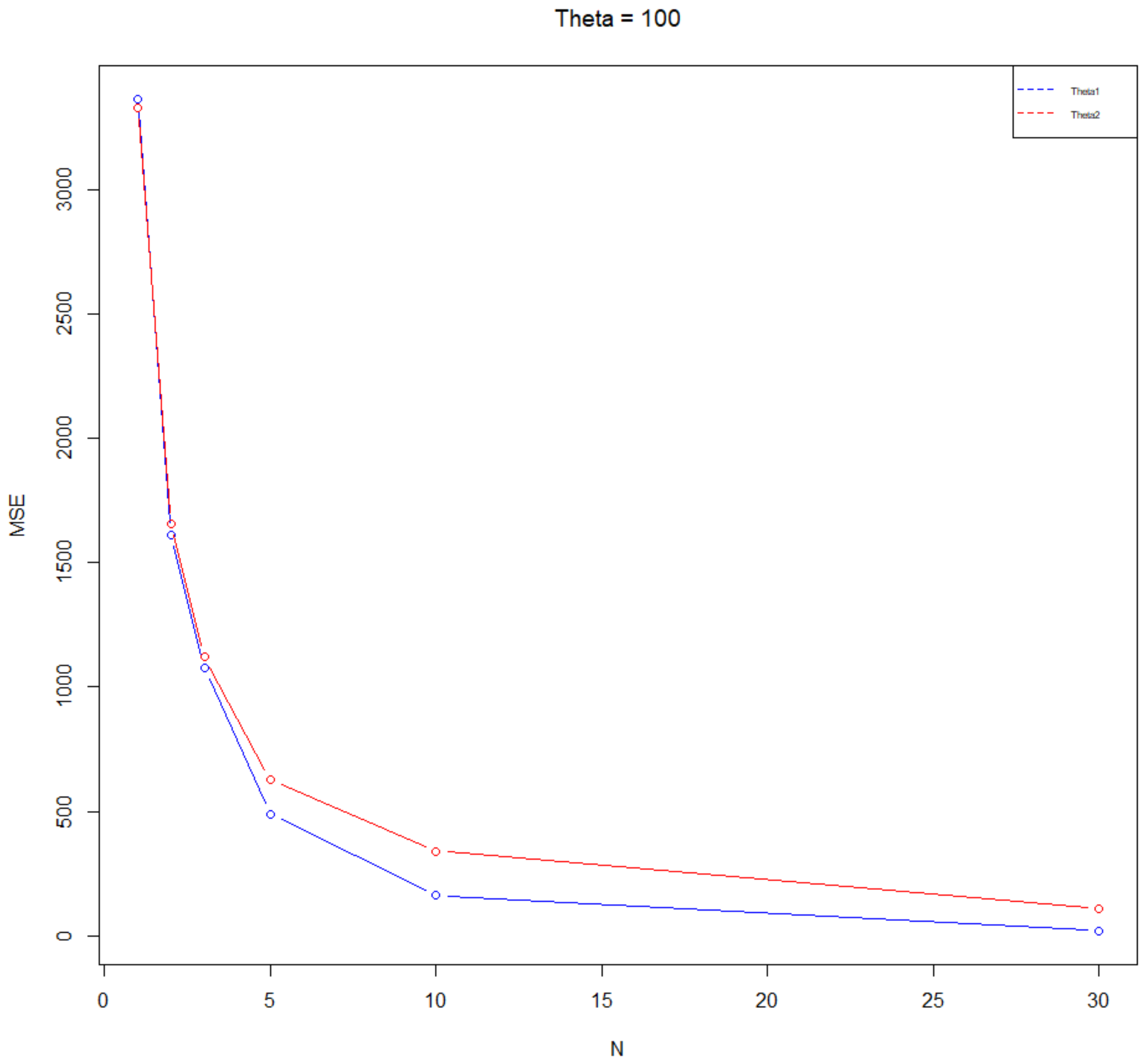


Theta = 1
Theta1
Theta2
MSE
0.00
0.05
0.10
0.15
0.20
0.25
0.30
N
0
5
10
15
20
25
30


Theta = 5
Theta1
Theta2
MSE
N
0
2
4
6
8
0
5
10
15
20
25
30



Theta = 50
Theta1
Theta2
MSE
N
0
200
400
600
800
0
5
10
15
20
25
30



1d) Based on (c), which estimator is better? Does the answer depend on n or θ? Explain. Provide justification for all your conclusions.

1. We can observe that as the value of samples(n) increases the mean squared error decreases. But as sample size increases, we can observe from plot that MSE for MLE is less than MME. So, we can say MLE is better.
2. Thus, as n increases MLE becomes better.
3. It can be seen by comparing the graphs for keeping n as constant and θ as constant that no matter what value of θ gets fixed, the resulting graphs are extremely similar. So, it can be interpreted that the estimator wouldn't depend on the value of θ.

Q2. (12 points) Suppose the lifetime, in years, of an electronic component can be modeled by a continuous random variable with probability density function f(x) = (θ xθ+1 x ≥ 1, 0, x < 1, where θ > 0 is an unknown parameter. Let X1, . . . , Xn be a random sample of size n from this population.

2a) Derive an expression for maximum likelihood estimator of θ.

To calculate MLE, we take likelihood function.

L(θ) =for i=1 to n { π} (θ/xi^ θ+1)

We take Log on both sides,
Log L(θ) = Log(i=1 to n { π} (θ/xi^ θ+1))

= Log(θ^n * (for i=1 to n {π} (1/ xi^ θ+1)))

= n Log θ + i=1 to n ∑ log(xi^(- θ-1))

= nlog θ – (θ+1) {i=1 to n ∑ log xi}

Log L(θ) = nlog θ - θ{i=1 to n ∑ log xi} - {i=1 to n ∑ log xi}

We take partial derivative which gives results,

n/ θ - {i=1 to n ∑ log xi}

Now, we equate to 0 to get MLE.

n/ θ - {i=1 to n ∑ log xi} = 0

n/ θ = i=1 to n ∑ log xi

θ hat (mle) = n/{i=1 to n ∑ log xi}

2b) Suppose n = 5 and the sample values are x1 = 21.72, x2 = 14.65, x3 = 50.42, x4 = 28.78, x5 = 11.23. Use the expression in (a) to provide the maximum likelihood estimate for θ based on these data.

We estimate θ hat (mle) by substituting these values.

θ hat (mle) = 5/ log(21.72) + log(14.65) + log(50.42) + log(28.78) + log(11.23)

= 5/ log(21.72*14.65*50.62*28.78*11.23)

= 5/ log(5185263.523)

= 0.323387

2c) 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?

```
x = c(21.42,14.65,50.42,28.78,11.23)
neg.loglik.fn <- function(par,dat)
{
  result = length(dat)*log(par)-(par+1)*sum(log(dat))
  return(-result)
}
mle = optim(par = 0.5, fn=neg.loglik.fn, method="L-BFGS-B", hessian = TRUE, lower = 0, dat=x)
mle
```

Output:

```
L-BFGS-B needs finite values of 'fn'
> mle
$par
[1] 0.323679

$value
[1] 26.08744

$counts
function gradient 
      20       20 

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

$hessian
         [,1]
[1,] 47.72538
```

Observation :

We got the theta hat value using calculations as 0.323 and we got the theta hat value as 0.323 Using the R optim function. Hence our answers match.

2d) 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.

R Code:

```
## Standard error calculation
SE= sqrt( diag(solve(mle$hessian)))
SE
## alpha 0.05 because 95% confidence interval
alpha <- 0.05
# confidence interval is mean +/- z-score*standard error
CI = mle$par + c(-1, 1)*qnorm(1-(alpha/2))*SE
CI
```

Console Terminal Jobs

R 4.1.1 · C:/Users/risbu/Downloads/

```
[1] 0.1447522
> ## alpha 0.05 because 95% confidence interval
> alpha <- 0.05
> # confidence interval is mean +/- z-score*standard error
> CI = mle$par + c(-1, 1)*qnorm(1-(alpha/2))*SE
> CI
[1] 0.03996976 0.60738814
```

Observation:

The obtained output seems appropriate approximation because 0.3 is the peak value i.e. theta hat and Confidence interval is between 0.03 and 0.6.

For 100 trials the CI indicates that estimated theta will lie within the range 95% of the time.