Mini Project : 3

Members :
1. Puru Jaiswal (NetID : PXJ200018)
2. Sara Tabassi (NetID : SXT200083)

Both the team members worked together to finish the project. Collaborated to learn R and then write the code. Both of us worked on each of the problems independently and then reviewed each other's answers to determine which solution was optimal.
Both members worked efficiently to complete the project requirements.

1. 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.

(a) Explain how you will compute the mean squared error of an estimator using Monte Carlo simulation.
   1. Set the population parameter.
   2. Simulate different sample values which will allow for the calculation of estimator value.
   3. MSE := Squared difference between the estimator value and the parameter.

   Sample Code : MSE <- mean((ThetaHat_1 - THETA[j])^2)

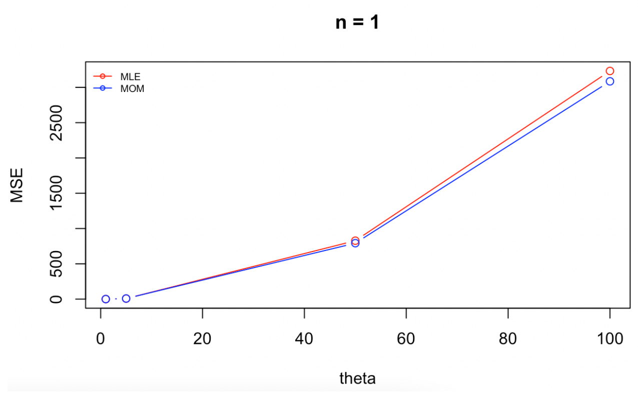
(b) For a given combination of (n, θ), compute the mean squared errors of both ˆθ1 and ˆθ2 using Monte Carlo simulation with N = 1000 replications. Be sure to compute both estimates from the same data.

R Code present at the end.

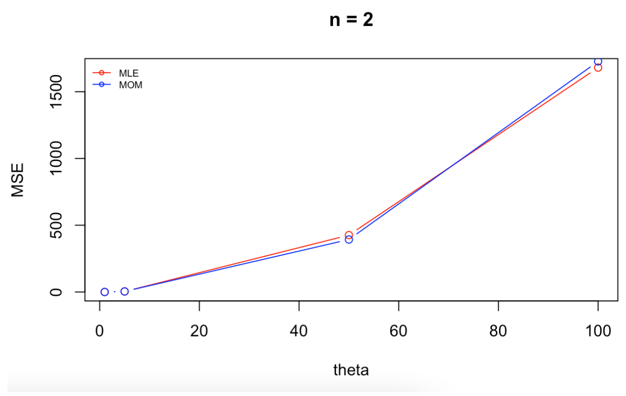
MSE(theta_1) = 0.3276260 MST(theta_2) = 0.3275711

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

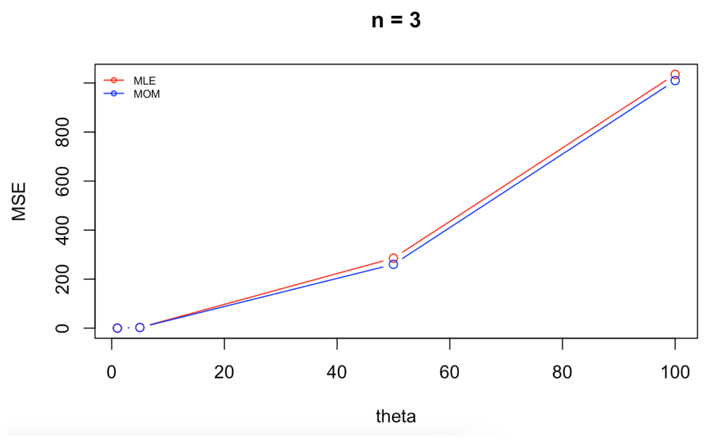
Graph for mean squared error -> MLE and MOM Fixed n varying theta



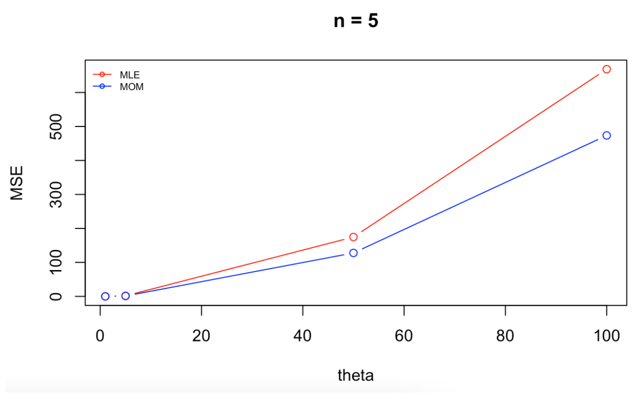
a)



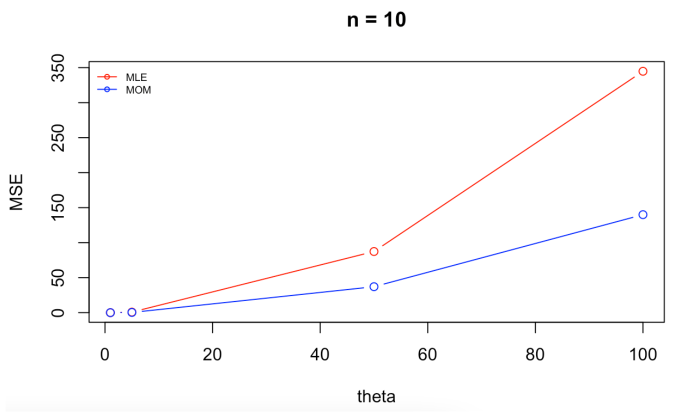
b)



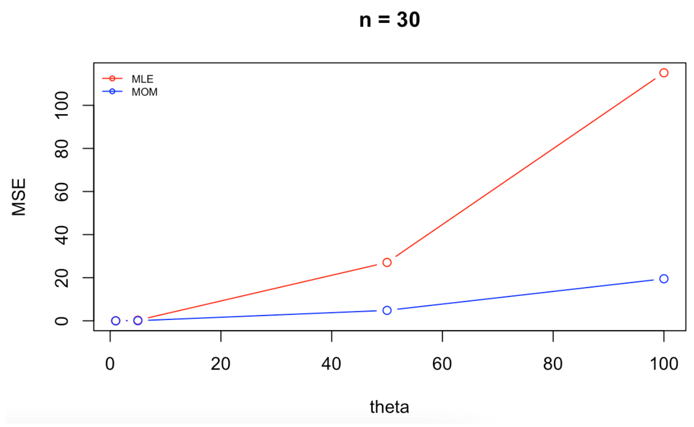
c)


n = 5
MLE
MOM
MSE
0
100
300
500
0
20
40
60
80
100
theta

d)

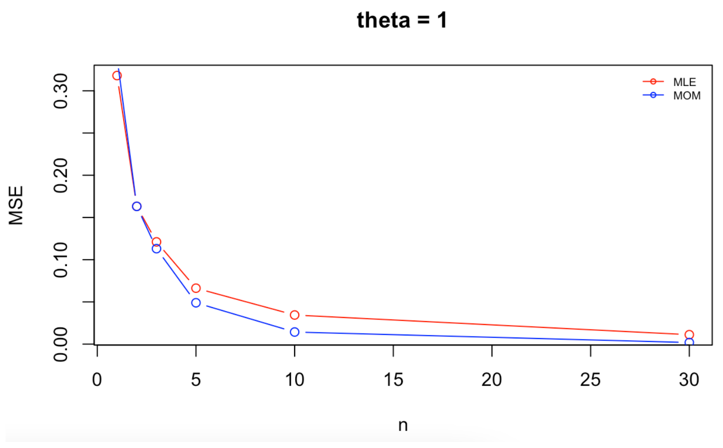

n = 10
MLE
MOM
MSE
0
50
150
250
350
0
20
40
60
80
100
theta

e)

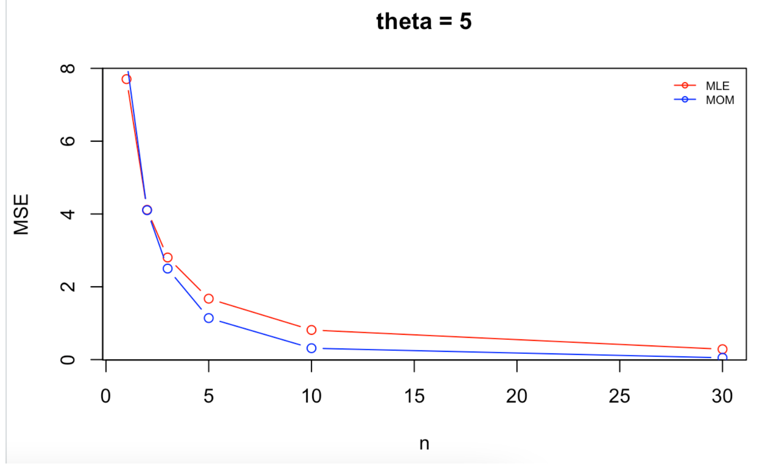

n = 30
MLE
MOM
MSE
0
20
40
60
80
100
0
20
40
60
80
100
theta

f)

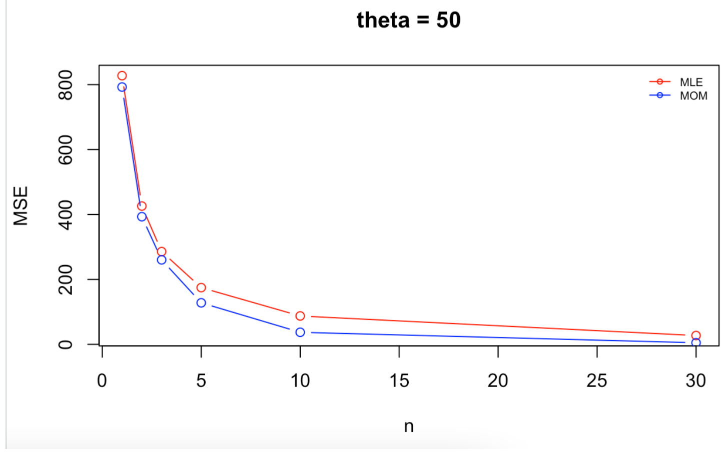
Graph for mean squared error -> MLE and MOM Varying n fixed theta



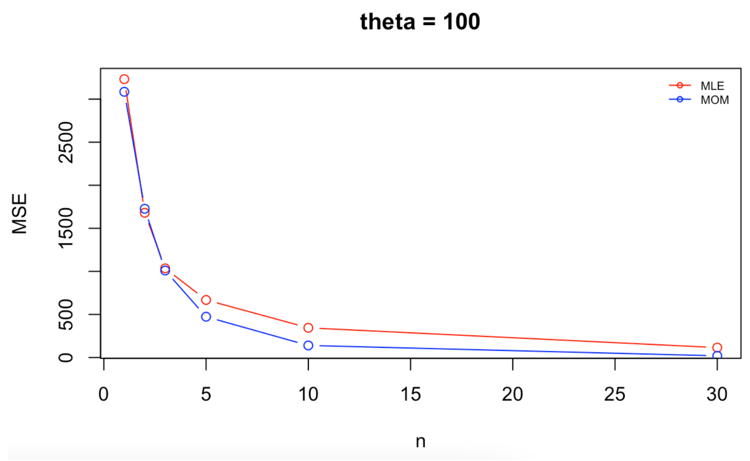
a)



b)



c)



d)

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

(i) In the above graphs where n is fixed and θ is variated, for those with n = 1,2 and 3 we can use method of moments estimator where as n starts increasing i.e n = 5, 10, 30 method of maximum likelihood will be much better. MLE is more preferable than MOM as n starts increasing.

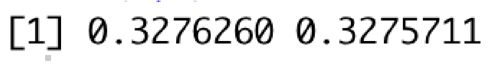
(ii) For graphs where θ is fixed and n is variated, we see that fixing any value of θ there is very minute change in the graph, from this observation we can say that the estimator doesn't depend on θ .

R Code

1b)

```
# returns both mle and mom as these results will passed in replicate function

mle.mom <- function(n, theta){

  sample_data = runif(n, 0, theta)

  method_moments = 2 * mean(sample_data)

  max_likelihood = max(sample_data)

  return (c(max_likelihood, method_moments))

}
```

```
# to calculate the mean squared error

mse <- function (n, theta){

  estimator = replicate(1000, mle.mom(n,theta))

  estimator = (estimator-theta)^2

  estimator.mom = estimator[c(TRUE, FALSE)]

  estimator.mle = estimator[c(FALSE, TRUE)]

  return (c(mean(estimator.mle),mean(estimator.mom)))

}
```

mse(1,1) : 

```
[1] 0.3276260 0.3275711
```

1c)

```
# fixed n varying theta
# n -> 1

plot(c(1,5,50,100),
    c(mse.1.1[1],mse.1.5[1], mse.1.50[1], mse.1.100[1]), type="b",
    xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 1")

lines(c(1,5,50,100), c(mse.1.1[2],mse.1.5[2], mse.1.50[2],
                    mse.1.100[2]), type="b", col = 'blue')

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

```
#n -> 2

plot(c(1,5,50,100),
  c(mse.2.1[1],mse.2.5[1], mse.2.50[1], mse.2.100[1]), type="b",
  xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 2")

lines(c(1,5,50,100), c(mse.2.1[2],mse.2.5[2], mse.2.50[2],
                  mse.2.100[2]), type="b", col = 'blue')

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

#n -> 3

plot(c(1,5,50,100),
  c(mse.3.1[1],mse.3.5[1], mse.3.50[1], mse.3.100[1]), type="b",
   xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 3")

lines(c(1,5,50,100), c(mse.3.1[2],mse.3.5[2], mse.3.50[2],
                  mse.3.100[2]), type="b", col = 'blue')

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




#n -> 5

plot(c(1,5,50,100),
    c(mse.5.1[1],mse.5.5[1], mse.5.50[1], mse.5.100[1]), type="b",
    xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 5")

lines(c(1,5,50,100), c(mse.5.1[2],mse.5.5[2], mse.5.50[2],
                  mse.5.100[2]), type="b", col = 'blue')

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

```
#n -> 10

plot(c(1,5,50,100),
     c(mse.10.1[1],mse.10.5[1], mse.10.50[1], mse.10.100[1]), type="b",
     xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 10")

lines(c(1,5,50,100), c(mse.10.1[2],mse.10.5[2], mse.10.50[2],
                  mse.10.100[2]), type="b", col = 'blue')

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



#n -> 30

plot(c(1,5,50,100),
     c(mse.30.1[1],mse.30.5[1], mse.30.50[1], mse.30.100[1]), type="b",
     xlab = 'theta', ylab = 'MSE', col = 'red', main = "n = 30")

lines(c(1,5,50,100), c(mse.30.1[2],mse.30.5[2], mse.30.50[2],
                  mse.30.100[2]), type="b", col = 'blue')

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



# varying and fixed theta

#theta -> 1

plot(c(1,2,3,5,10,30), c(mse.1.1[1],mse.2.1[1], mse.3.1[1], mse.5.1[1],
     mse.10.1[1], mse.30.1[1]), type="b", ylab = 'MSE',
     xlab = 'n', col = 'red', main = "theta = 1")

lines(c(1,2,3,5,10,30), c(mse.1.1[2],mse.2.1[2], mse.3.1[2], mse.5.1[2],
                   mse.10.1[2], mse.30.1[2]), type="b", col = 'blue')

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

```
#theta -> 5

plot(c(1,2,3,5,10,30), c(mse.1.5[1],mse.2.5[1], mse.3.5[1], mse.5.5[1],
                     mse.10.5[1], mse.30.5[1]), type="b", ylab = 'MSE',
    xlab = 'n', col = 'red', main = "theta = 5")

lines(c(1,2,3,5,10,30), c(mse.1.5[2],mse.2.5[2], mse.3.5[2], mse.5.5[2],
                      mse.10.5[2], mse.30.5[2]), type="b", col = 'blue')

legend("topright", legend = c("MLE", "MOM"), col = c('red', 'blue'),
      text.col = c('black','black'),lty = 1, pch = 1,
      inset =0.01, ncol = 1, cex = 0.6, bty = 'n')



#theta -> 50

plot(c(1,2,3,5,10,30), c(mse.1.50[1],mse.2.50[1], mse.3.50[1], mse.5.50[1],
                     mse.10.50[1], mse.30.50[1]), type="b", ylab = 'MSE',
    xlab = 'n', col = 'red', main = "theta = 50")

lines(c(1,2,3,5,10,30), c(mse.1.50[2],mse.2.50[2], mse.3.50[2], mse.5.50[2],
                      mse.10.50[2], mse.30.50[2]), type="b", col = 'blue')

legend("topright", legend = c("MLE", "MOM"), col = c('red', 'blue'),
      text.col = c('black','black'),lty = 1, pch = 1,
      inset =0.01, ncol = 1, cex = 0.6, bty = 'n')



#theta -> 100

plot(c(1,2,3,5,10,30), c(mse.1.100[1],mse.2.100[1], mse.3.100[1], mse.5.100[1],
                     mse.10.100[1], mse.30.100[1]), type="b", ylab = 'MSE',
    xlab = 'n', col = 'red', main = "theta = 100")

lines(c(1,2,3,5,10,30), c(mse.1.100[2],mse.2.100[2], mse.3.100[2], mse.5.100[2],
                      mse.10.100[2], mse.30.100[2]), type="b", col = 'blue')

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

2.

A.
```
> get.theta.fun<-function(data){
+ theta = length(data)/sum(log(data))
+ return(theta)
+}
```

$$\ln\big(L(\vartheta)\big) = \sum_{i=1}^{n} \ln\left(\frac{\theta}{x_i^{\theta+1}}\right) = n\ln\theta + (-\theta - 1)\sum_{i=1}^{n} \ln(x_i)$$

$$0 = \frac{\vartheta}{\vartheta\theta}\ln\big(L(\vartheta)\big) = \frac{n}{\theta} - \sum_{i=1}^{n} \ln(x_i)$$

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

B.
```
> sampleVals = c(21.72, 14.65, 50.42, 28.78, 11.23)
>
> #Function to calculate theta
> get.theta.fun <- function(data) {
+    theta = length(data)/sum(log(data))
+    return(theta)
+}
> #call above function with sample data
> get.theta.fun(sampleVals)
[1] 0.3233874
```

MLE = 0.3233874

C.

```
> #negative log function
> neg.loglik.fun <- function(theta, dat) {
+ result = sum(log(theta / dat^(theta + 1)))
+ return(-result)
+}
>
> #optimize function to get estimator
> ml.est <- optim(par = 1, fn = neg.loglik.fun, method = "BFGS", hessian = TRUE,
+dat=sampleVals)
> ml.est$par
[1] 0.323387
```

MLE = 0.323387

Answer does match part B

D.

```
> # Standard error of estimate
 > SE = sqrt(1/ml.est$hessian)
> SE
          [,1]
[1,] 0.1446217
> #confidence Interval
> c(ml.est$par-qnorm(.975)*SE, ml.est$par+qnorm(.975)*SE)
```

```
[1] 0.03993372 0.60684034
```

SE = 0.1446217

CI = (0.03993372, 0.60684034)

These approximations will not be good because n is too small (5). This means we have a pretty large error, and our width for the confidence interval will be quite big, since there's a lot of error in our estimate. If our n was bigger (at least 30), our estimates would be much more accurate.