STATISTICAL METHODS FOR DATA SCIENCE
MINI PROJECT #3
SPRING 2018

NAMES OF GROUP MEMBERS:

ADRITA DUTTA(axd172930)

NEETHU ANTONY(nxa171330)

Contribution of team members:

ADRITA: Question 2

NEETHU: Question 1

SECTION 1:

1.a) To find the mean squared error using monte carlo simulation, generate n uniform random variables with the parameter θ. ie draw a random sample of size n from a Uniform(0, θ) distribution. Then compute the estimators (In this case MLE and MOM, MLE is the max value and MOM twice the mean). Then compute the square of the difference b/w the estimator and the actual value. Repeat this process a number of times, noting down the squared error we got on each sample. After this is complete take the mean of squared errors for all samples.

Given the maximum likelihood estimator $\hat{\theta_1}$=X(n) and the method of moments estimator, $\hat{\theta_2}$= 2$\bar{x}$, where $\bar{x}$ is the sample mean.

To compare these two estimators, by Monte Carlo simulation for a specific n and θ:
1. Generate X1, ..., Xn ~ Uniform(0, θ)
2. Calculate $\hat{\theta_1}$ and $\hat{\theta_2}$
3. Save $(\hat{\theta_1} - \theta)^2$ and $(\hat{\theta_2} - \theta)^2$
4. Repeat step 1-3 N times
5. Then the means of the $(\hat{\theta_1} - \theta)^2$ and $(\hat{\theta_2} - \theta)^2$, over the N replicates, are the monte carlo estimators of the MSEs of $\hat{\theta_1}$ and $\hat{\theta_2}$.

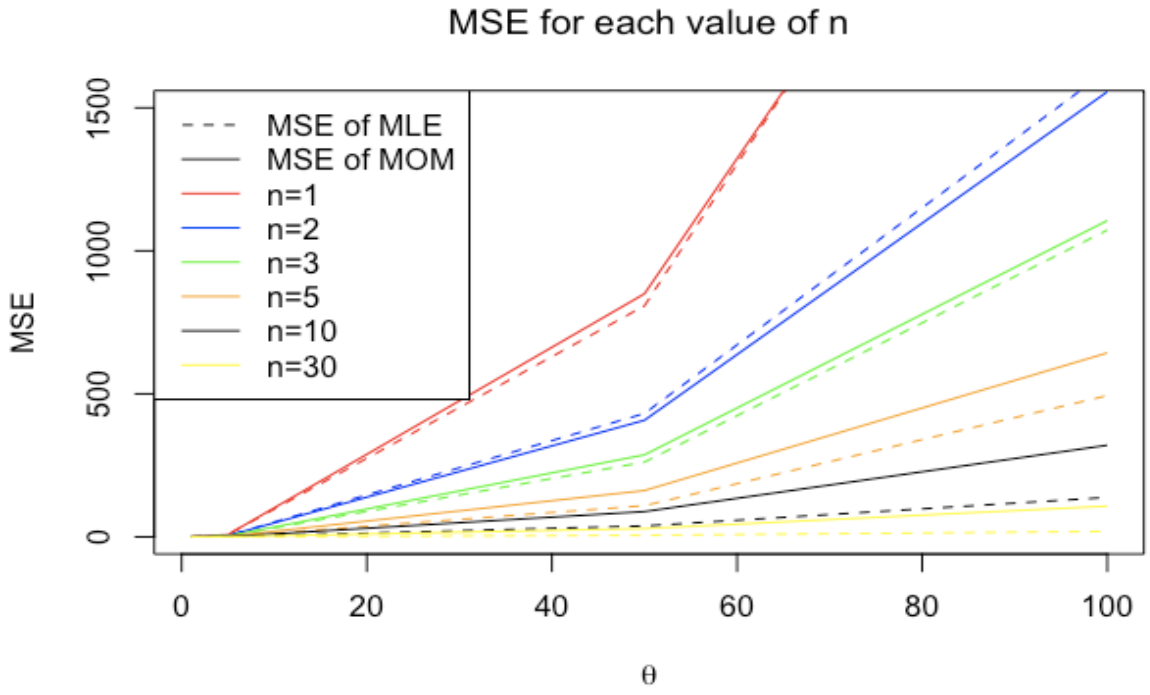
1.b) Mean square erroe of the estimators for n = 1 and θ = 1 using Monte Carlo simulation with N=1000 replications.

MSE(θˆ1) = 0.3472676.
MSE(θˆ2) = 0.3307753.

1.c) The following table has the value of mean squared error for different θ and n combinations:-

S.No	n	Parameter,θ	MSE of MLE	MSE of MOM
1	1	1	0.3472676	0.3307753
2	1	5	8.0448717	7.8424929
3	1	50	807.7511085	849.7993132
4	1	100	3286.9284612	3212.1109498
5	2	1	0.1739032	0.1637819
6	2	5	4.2604971	4.1820538
7	2	50	431.4702714	407.6796556
8	2	100	1629.9929003	1556.0639669
9	3	1	0.1027721	0.1126447
10	3	5	2.5442398	2.8867602
11	3	50	261.3919816	286.7608110
12	3	100	1072.5370244	1105.5048976
13	5	1	0.04737599	0.06859401
14	5	5	1.29660315	1.81587127
15	5	50	109.02106504	161.99301039
16	5	100	493.83066042	643.08428534
17	10	1	0.01646214	0.03285345
18	10	5	0.37533165	0.87976788
19	10	50	38.27817348	88.39233217
20	10	100	137.80997374	320.44634929
21	30	1	0.001870972	0.01123121
22	10	5	0.046695226	0.27596283
23	10	50	5.575015117	28.87742374
24	10	100	18.910617947	107.45399807

The following plot depict the trend in the value of the mean squared error for both MLE AND MOM as n and θ changes:-



In the graph the dotted lines represent mean squared error for MLE estimator and the straight line shows mean squared error for method of moments.

1.d) The following observations can be made by looking at the graphs

- The Mean Squared Error of both estimators increases as the value of θ increases.
- The Mean Squared Error of both estimators decreases as the value of sample size, n increases.
- The Mean Squared Errors of MLE is generally less than that of method of moments estimator. Which implies that, for these sample sizes at least MLE is the better estimator.
- The difference in the mean squared errors of the two estimators go down as the value of n increases. Which suggests convergence for even larger value of n.

These observations can be explained the following way:-

- As the value of θ increases the differences between the parameter and the sample values increase as the range of the random variable goes up. This results in increases mean squared errors for both estimators as the value of θ increases.
- As n increases the MSE of the maximum likelihood estimator decreases as a result of the large sample properties of the MLE. Since MLE is asymptotically unbiased, as the value of n increases the mean squared error decreases.
- The improvement in the performance of the method of moments estimator is a result of the Law of Large numbers. We know that the twice the mean is almost the same as the max for a Uniform(0, θ) distribution. So analytically these two estimators are the same. As the value of n increases the sample mean approaches the population mean due to the Law of Large numbers. And twice the sample mean approaches the population parameter. This causes the mean squared error of the method of moments estimator to go down as n increases.
- The above fact explains the third and fourth observations as well.

- We can conclude that for an arbitrary choice of sample size for a random variable following uniform distribution the MLE is the better estimator. However for larger 'n' the method of moments estimator is as good as the MLE.

2a)

The given function:

$f_\theta(x) = \frac{\theta}{(x_i)^{(\theta+1)}}$ for 0<=x<=1;

$f_\theta(x) = 0$ otherwise

->The likelihood function:

$f_\theta(x) = \theta^n \prod_{i=1}^{n} x_i^{\theta+1}$

->The log likelihood function:

$ln(f_\theta(x)) = nln(\theta) - (\theta + 1)\Sigma ln(x_i)$

->On differentiation of log likelihood function and equating it to 0:

$\frac{d}{d\theta} ln(f_\theta(x)) = \frac{n}{\theta} - (\theta + 1)\Sigma ln(x_i) = 0$

->On evaluating this we get:

$\widehat{\theta} = \frac{n}{\Sigma ln(x_i)}$

b) for n=4, x_1=4.79, x_2=10.89, x_3=6.54, x_4=22.15

$\widehat{\theta} = \frac{4}{\Sigma ln(x_i)} = \frac{4}{ln(x_1)+ln(x_2)+ln(x_3)+ln(x_4)} = \frac{4}{8.91} = 0.4479208071$

c)estimate by numerically maximizing the log-likelihood function using optim function in R:
->The output of the optim function also gives the same output as the maximum likelihood estimate from (b)
MLE(using optim function)= 0.4478906

d)
Standard error(SE($\widehat{\theta}$))= $\frac{1}{\sqrt{\hat{I}}} = \sqrt{1/19.93978}$
=0.2239442(same as value in R)

Confidence Interval= $\widehat{\theta} \pm t(\alpha/2)\, SE(\widehat{\theta})$ where 95%=1- α , α/2 = 0.025 or 27.5th percentile
=[-0.2647998 , 1.1605810]

In this case t distribution is taken as $\boldsymbol{\sigma}$ is unknown and n is small(4)
These approximations are good as the MLE obtained from both numerical calculation and from part a give the same value that is in this interval and the interval is not too large.

SECTION 2:

1.(b)

```
dataset <- matrix(runif(1000*1,0,1),1000,1)
dataset
# Calculate theta_hat1 (mle) for each data set
thetaHat_1 <- apply(dataset, 1, max)
thetaHat_1
# Calculate theta_hat2 (mom) for each data set
thetaHat_2 <- 2*apply(dataset, 1, mean)
thetaHat_2
#Take mean of the squared error of thetaHat_1's
x<-mean((thetaHat_1 - 1)^2)
x
#Take mean of the squared error of thetaHat_2's
y<-mean((thetaHat_2 - 1)^2)
y
```

Narration of code:

First of all, use the runif() function to generate numbers which are uniformly distributed between 0 and 1 and store the generated numbers in the dataset. After generating numbers which are uniformly distributed, use apply method with the corresponding functions for both estimators. After that mean square error is calculated for both estimators by applying the formula. Finally, print the outputs which gives the mean square error of each of the estimators with respect to $\theta=1$(in this case).

1.(c)

Idea of the Code:-

```
#Repeat the below code for different values of n= 1,2,3,5,10,30.
#For different values of n, only one line is changed in the below code which is :
#dataset <- matrix(runif(1000*n,0,theta[i]),1000,n)

theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*n,0,theta[i]),1000,n)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  thetaHat_1
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  thetaHat_2
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
```

```
}
# Plot the results on the same axes
plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",
    main=expression(paste("MSE for each value of ", theta)),
    type="l", col="red",ylim = c(0,300))
lines(theta, MSE[,2], col="blue" lty = 2 )
```

Narration of code:
First of all, MSE matrix is generated to store estimated values of $\hat{\theta}_1$ and $\hat{\theta}_2$. After generating numbers which are uniformly distributed, use apply method with the corresponding functions for both estimators. After that mean square error is calculated for both estimators by applying the formula. At the last, using plot function, graph is plotted for both. Red line is for $\hat{\theta}_1$ and Blue line is for $\hat{\theta}_2$.

To change n, runif is done 1000*n is done, and to store these value number of columns in dataset matrix is changed to new value of n.

Complete Code:-

```
#n= 1,2,3,5,10,30.
#n=1
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*1,0,theta[i]),1000,1)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
plot(theta, MSE[,1], xlab=quote(theta), ylab="MSE",
    main=expression(paste("MSE for each value of n")),
    type="l", col="red",ylim = c(0,1500), lty = 2)
lines(theta, MSE[,2], col="red")

#n=2
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
```

```
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*2,0,theta[i]),1000,2)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
lines(theta, MSE[,1], col="blue", lty = 2)
lines(theta, MSE[,2], col="blue")

#n=3
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*3,0,theta[i]),1000,3)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
lines(theta, MSE[,1], col="green", lty = 2)
lines(theta, MSE[,2], col="green")

#n=5
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*5,0,theta[i]),1000,5)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
```

```
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
lines(theta, MSE[,1], col="orange", lty = 2)
lines(theta, MSE[,2], col="orange")

#n=10
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*10,0,theta[i]),1000,10)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
lines(theta, MSE[,1], col="black", lty = 2)
lines(theta, MSE[,2], col="black")

#n=30
theta <- c(1,5,50,100)
#Matrix to store the MSEs of both the estimators
MSE <- matrix(0, length(theta), 2)
#Loop through values in theta
for(i in 1:length(theta))
{
  dataset <- matrix(runif(1000*30,0,theta[i]),1000,30)
  # Calculate theta_hat1 (mle) for each data set
  thetaHat_1 <- apply(dataset, 1, max)
  # Calculate theta_hat2 (mom) for each data set
  thetaHat_2 <- 2*apply(dataset, 1, mean)
  # Save the MSEs
  MSE[i,1] <- mean((thetaHat_1 - theta[i])^2)
  MSE[i,2] <- mean((thetaHat_2 - theta[i])^2)
}
# Plot the results on the same axes
MSE
lines(theta, MSE[,1], col="yellow", lty = 2)
lines(theta, MSE[,2], col="yellow")

legend ( "topleft",
```

```
    legend = c("MSE of MLE", "MSE of MOM", "n=1", "n=2", "n=3", "n=5", "n=10", "n=30"),
    col = c("BLACK" ,"BLACK" , "RED" , "BLUE" , "GREEN" , "ORANGE" , "BLACK" ,
"YELLOW"),
    lty = c(2, 1, 1, 1, 1, 1, 1, 1))
```

2c)

```
#input array
x <- c(4.79,10.89,6.54,22.15);
#length of input
n=length(x);


#negative of log likelihood function. as log likelihood by default gives minimization result.
neg.loglik.fun <- function(theta)
{

  result <- n*log(theta)-((theta-1)*(sum(log(x))))
  return(-result)
}


#optim function to optimise the log likelihood
optim(theta <- 0,neg.loglik.fun, hessian=TRUE)
```

d)

```
#Standard error of the distribution
SE<- sqrt(diag(solve(ml.est$hessian)));

#confidence interval of the distribution for values of theta
#using qt(0.025, df=n-1) we get the opp of the interval.
alpha<-0.05
CI <- ml.est$par + c(-1, 1) * qt(0.975, df=n-1) *SE
```