CS-6313 – Statistical Methods for Data Science

Mini Project #3

Group No - 5
Manan Dalal (MUD200000)
Lipi Patel (LDP210000)

Contribution of Team Members:

Worked together with each other to solve both the problems. Manan worked on the scripts and made them efficient and readable. Lipi worked on analytic solutions. Both worked together on documenting and giving the final touch to the project.

Question 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.**

   Solution:

   - ⇨ For computing the Mean squared error, we must first set the population parameter.
   - ⇨ This will enable us to simulate sample values which would allow us to calculate the estimated value.
   - ⇨ The estimated value of the difference between estimator and the parameter squared will give us the Mean Squared Error.

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.

Solution

- ⇨ To compute MSE's for both estimators for 1000 replications, we will first create a function getMLEandMOM which will give the the values of MLE and MOM for a single sample.
- ⇨ Then, we would create a function called getMSE which will compute the MSE for both the estimators for a given value of n and theta.
- ⇨ For understanding purposes, we then get the MSE for n = 1 and theta = 1.

R Code

```r
#  Returns MLE and MOM for a sample
getMLEandMOM <- function(n, theta) {
  sample = runif(n, min = 0, max = theta)
  mom = 2 * mean(sample)
  mle = max(sample)
  return(c(mle, mom))
}

# Returns Mean squared Error of MLE and MOM for 1000 replications
getMSE <- function(n, theta) {
  values = replicate(1000, getMLEandMOM(n, theta))
  values = (values - theta)^2
  values.mom = values[c(TRUE, FALSE)]
  values.mle = values[c(FALSE, TRUE)]

  return(c(mean(values.mle), mean(values.mom)))
}

mse_1_1 = getMSE(1,1)
```

```
> mse_1_1
[1] 0.3266865 0.3393726
```

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

Solution

- ⇨ In order to fit all plots in one graph, we use the function par().
- ⇨ Generating graphs for MSE for both MLE and MOM, theta with fixed n.
- ⇨ Generating graphs for MSE for both MLE and MOM, n with fixed theta.

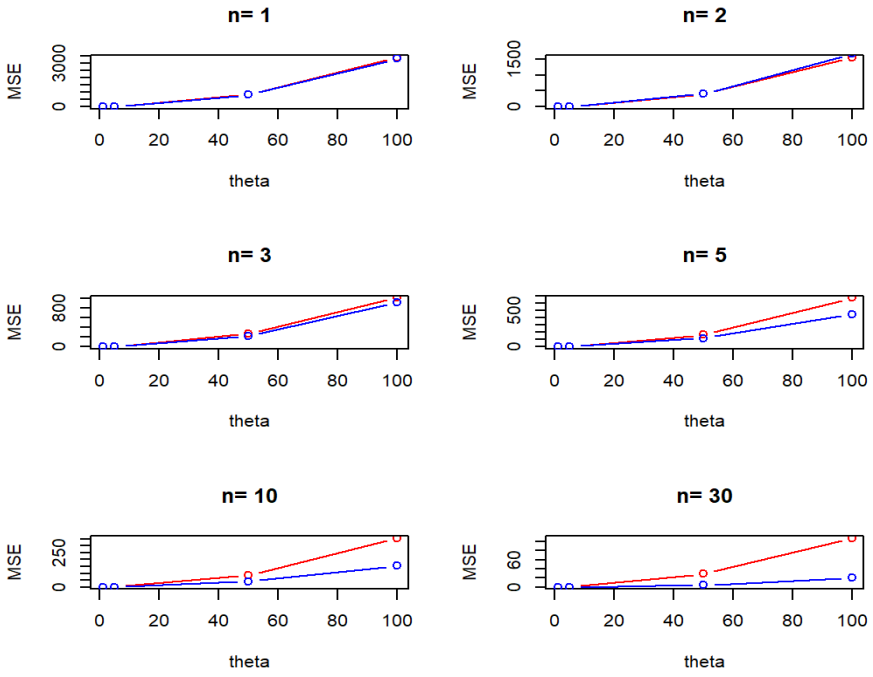
R Code

```
# Question 1 - c

n = c(1,2,3,5,10,30)
theta = c(1,5,50,100)

mse <- list()
for(x in n) {
  for(y in theta){
    mse = append(mse, list(getMSE(x,y)))
  }
}
```

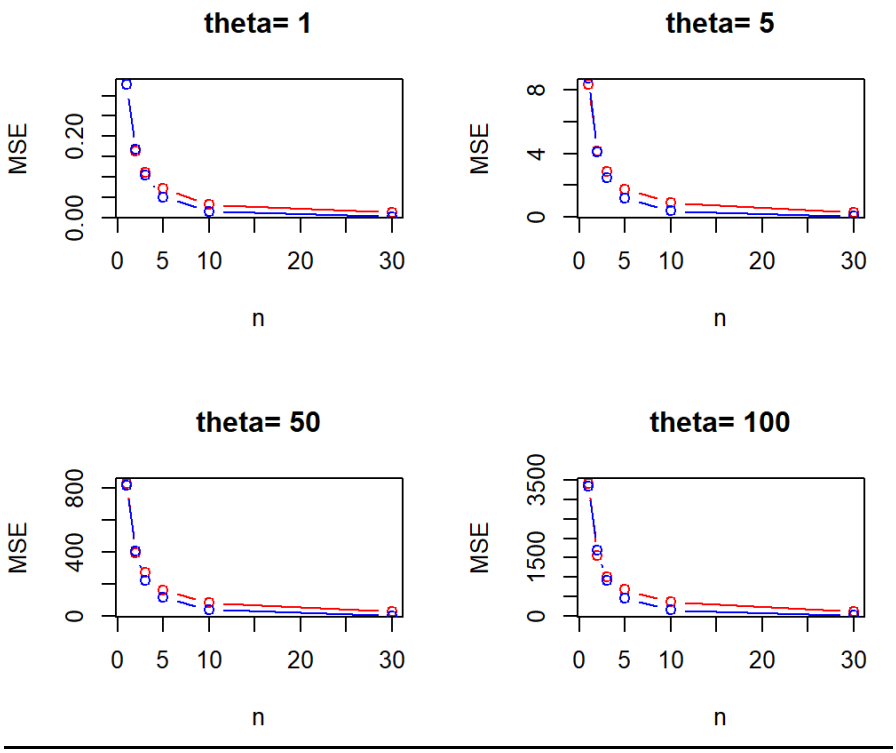
```
# Drawing graphs for fixed 'n' and varying 'theta'
par(mfrow = c(3,2))
t = 1

for(x in n) {
  msel1 = list()
  msel2 = list()
  for(y in 0:3){
    msel1 = append(msel1, unlist(mse[t + y])[1])
    msel2 = append(msel2, unlist(mse[t + y])[2])
  }

  plot(theta, c(unlist(msel1)), type="b", xlab="theta", ylab="MSE",
       col="red", main= paste("n=", x))
  lines(theta, c(unlist(msel2)), 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')
  t = t + 4
}
```

```r
# Drawing graphs for fixed 'theta' and varying 'n'
par(mfrow = c(2,2))
t = 1

for(x in theta) {
  msel1 = list()
  msel2 = list()
  for(y in 0:5){
    msel1 = append(msel1, unlist(mse[t+4*y])[1])
    msel2 = append(msel2, unlist(mse[t+4*y])[2])
  }

  plot(n, c(unlist(msel1)), type="b", xlab="n", ylab="MSE",
       col="red", main= paste("theta=", x))
  lines(n, c(unlist(msel2)), 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')
  t = t + 1
}
```

Outputs:



Graph 1 : Graph with fixed value of n



Graph 2 : Graph with fixed value of theta

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

Solution

- ⇨ The second graph shows us that the value of theta does not affect the graph as they are quite similar for all values of theta.
- ⇨ Thus, we can infer that the estimator would not depend on the value of theta.
- ⇨ By looking at Graph-1, we can evidently say that the Method of Moments(MOM) gives a very good estimate for smaller values of n (i.e. n = 1, 2, 3).
- ⇨ But, as the value of n increases (i.e. n = 5, 10, 30), the Maximum Likelihood Estimator performs better.
- ⇨ Thus, for larger values of n, MLE is better and would be the preferred choice of estimation when choosing among the two estimators.

Question-2:

Suppose the lifetime, in years, of an electronic component can be modeled by a continuous random variable with probability density function

$$f(x) = \begin{cases} \theta / x^{\theta+1}, & x \geq 1, \\ 0, & x < 1 \end{cases}$$

where $\theta > 0$ is an unknown parameter. Let X1, . . . , Xn be a random sample of size n from this population.

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

Solution:

Let the likelihood function be

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

Taking log on both sides:

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

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

$$= n \log\theta + \sum_{i=1}^{n} \log\left(x_i^{-\theta-1} \right)$$

$$= n \log\theta - (\theta+1) \sum_{i=1}^{n} \log x_i$$

$$= n \log\theta - \theta \sum_{i=1}^{n} \log x_i - \sum_{i=1}^{n} \log x_i$$

$$\frac{\partial (L(\theta))}{\partial \theta} = \frac{n}{\theta} - \sum_{i=1}^{n} \log x_i$$

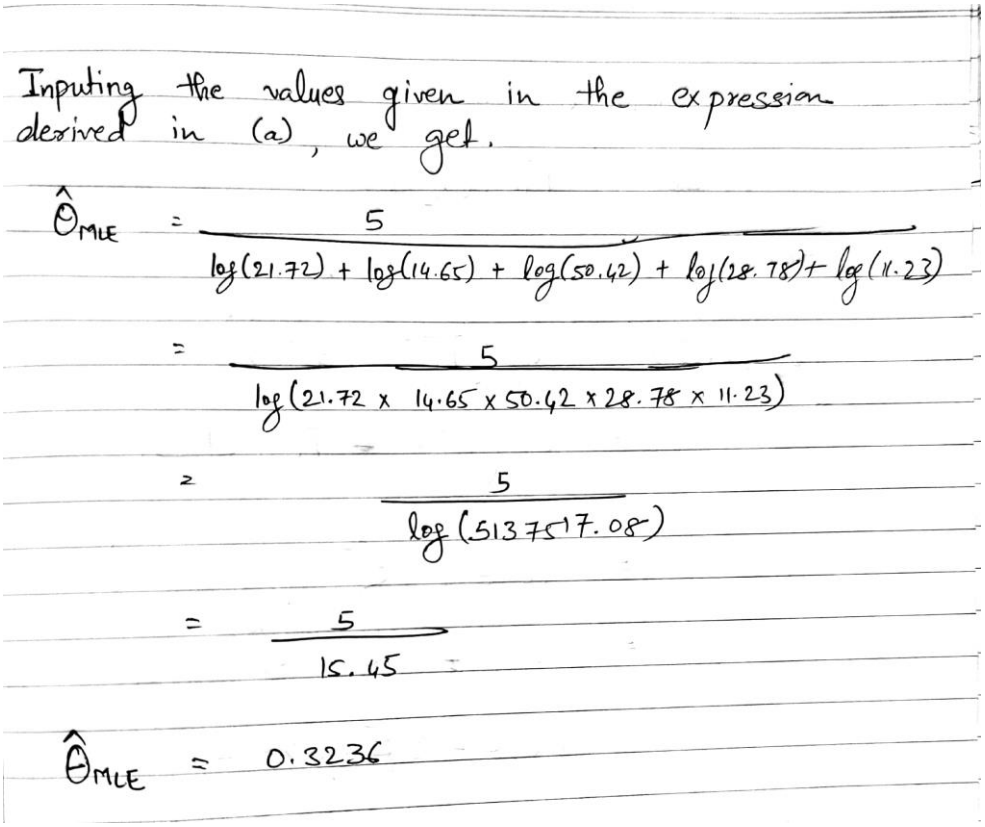
$$\therefore \frac{n}{\theta} - \sum_{i=1}^{n} \log x_i = 0$$

$$\therefore \frac{n}{\theta} = \sum_{i=1}^{n} \log x_i$$

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

b) 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.

Solution



c) 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?

Solution

- ⇨ In R, in order to minimize a function, we can maximize the negative of that function.
- ⇨ We use optim() to perform this task.

- Optim function takes various arguments such as:
  - par – initial value of the parameters that need to be minimized.
  - fn – minimization function
  - method – optimization method to use
  - dat – data available.
- When, we run the code below, we get the value of theta as .3064 which is close to the analytically derived value of 0.3236.

R Code

```
# Question-2 c

# Returns neg value of the derived function
negativeMLE <- function(par, dat) {
  print(par,dat)
  result = length(dat) * log(par) - (par+1) * sum(log(dat))
  return(-result)
}

x <- c(21.42, 14,65, 50.42, 28.78, 11.23)

mle <- optim(par = 2, fn = negativeMLE, method = "L-BFGS-B",
             hessian = TRUE, lower = 0.01, dat = x)
```

Output

```
> mle$par
[1] 0.3064923
```

d) 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.

Solution

- ⇨ The formula for Standard Error is given by:
  - $SE(\theta)^2 = 1/h$ => h is the hessian function
- ⇨ From the following code we get the value for SE as 0.1251236.
- ⇨ A 95% confidence interval, means
  - $(1 - \alpha) = 0.95$
  - $\alpha = 0.05$
  - $\alpha/2 = 0.025$
  - $1 - \alpha/2 = 0.975$
- ⇨ The confidence interval formula is given by:
  - $\theta \perp Z_{\alpha/2} * SE(\theta)$
- ⇨ We can get $Z_{\alpha/2}$ by using the qnorm function.
- ⇨ The confidence interval we get from our calculations is:
  - (0.06125447, 0.55172985)
- ⇨ This lets us infer that out of 100 trails for population estimation, the true estimate value would lie within the interval 95 times.

R Code

```
90  # d
91
92  # Finding Standard Error
93  se <- (1/mle$hessian)^0.5
94
95  # Finding Confidence Interval
96  conf = mle$par + c(-1,1)*se*qnorm(0.975)
```

Output

```
> se
          [,1]
[1,] 0.1251236
> conf
[1] 0.06125447 0.55172985
```