Mini Project #3

Sudarshana Jagadeeshi

Contribution of each group member: I completed the project in full

**Section 1**

**1.**

**a**) We will calculate MSE by performing (expected\_parameter-observed\_parameter)^2 for each of the N iterations. I will then sum these results up and divide by N, getting the average. This is similar to the standard MSE formula:

(1/N) \* sum\_over\_N[ (expected\_parameter-observed\_parameter)^2 ]

**b)**

[1] 29.66935

[1] 5.169158

The top value is theta 1 MSE, the bottom is theta2 MSE. Parameters chosen

**c)**

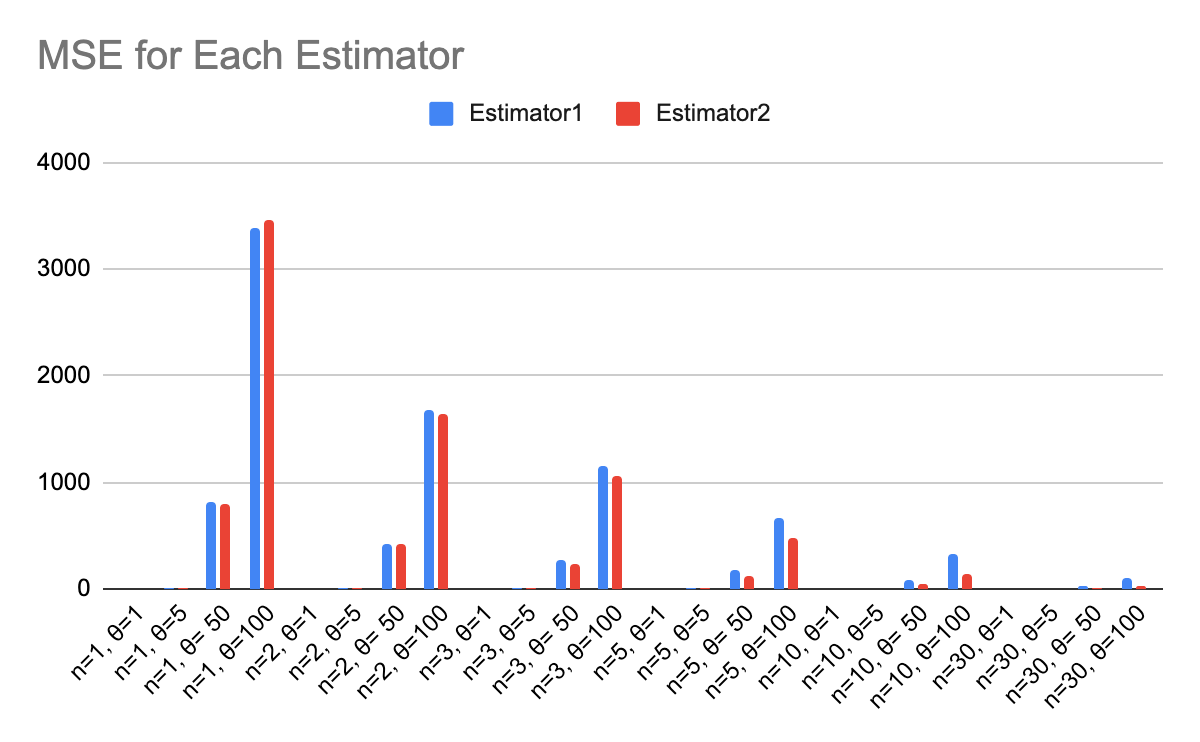
Estimator1

|  | theta=1 | theta=5 | theta=50 | theta=100 |
| --- | --- | --- | --- | --- |
| n=1 | 0.32 | 8.49 | 819 | 3393 |
| n=2 | 0.173 | 4 | 427 | 1688 |
| n=3 | 0.11 | 2.83 | 277 | 1152 |
| n=5 | 0.0656 | 1.631 | 176 | 658 |
| n=10 | 0.033 | 0.86 | 90 | 335 |
| n=30 | 0.01 | 0.28 | 27.37 | 106.02 |

Estimator2

|  | theta=1 | theta=5 | theta=50 | theta=100 |
| --- | --- | --- | --- | --- |
| n=1 | 0.33 | 8.3 | 805 | 3462 |
| n=2 | 0.17 | 4.18 | 423 | 1639 |
| n=3 | 0.106 | 2.49 | 243 | 1069 |
| n=5 | 0.04611 | 1.274 | 125 | 485 |
| n=10 | 0.015 | 0.368 | 38.8 | 147 |
| n=30 | 0.001819 | 0.04 | 4.66 | 20.68 |

Yields the following graph:



**d)**

Note- estimator1 is MoM, estimator2 is MLE

We can see the decreasing values of MSE are correlated to increase in n.

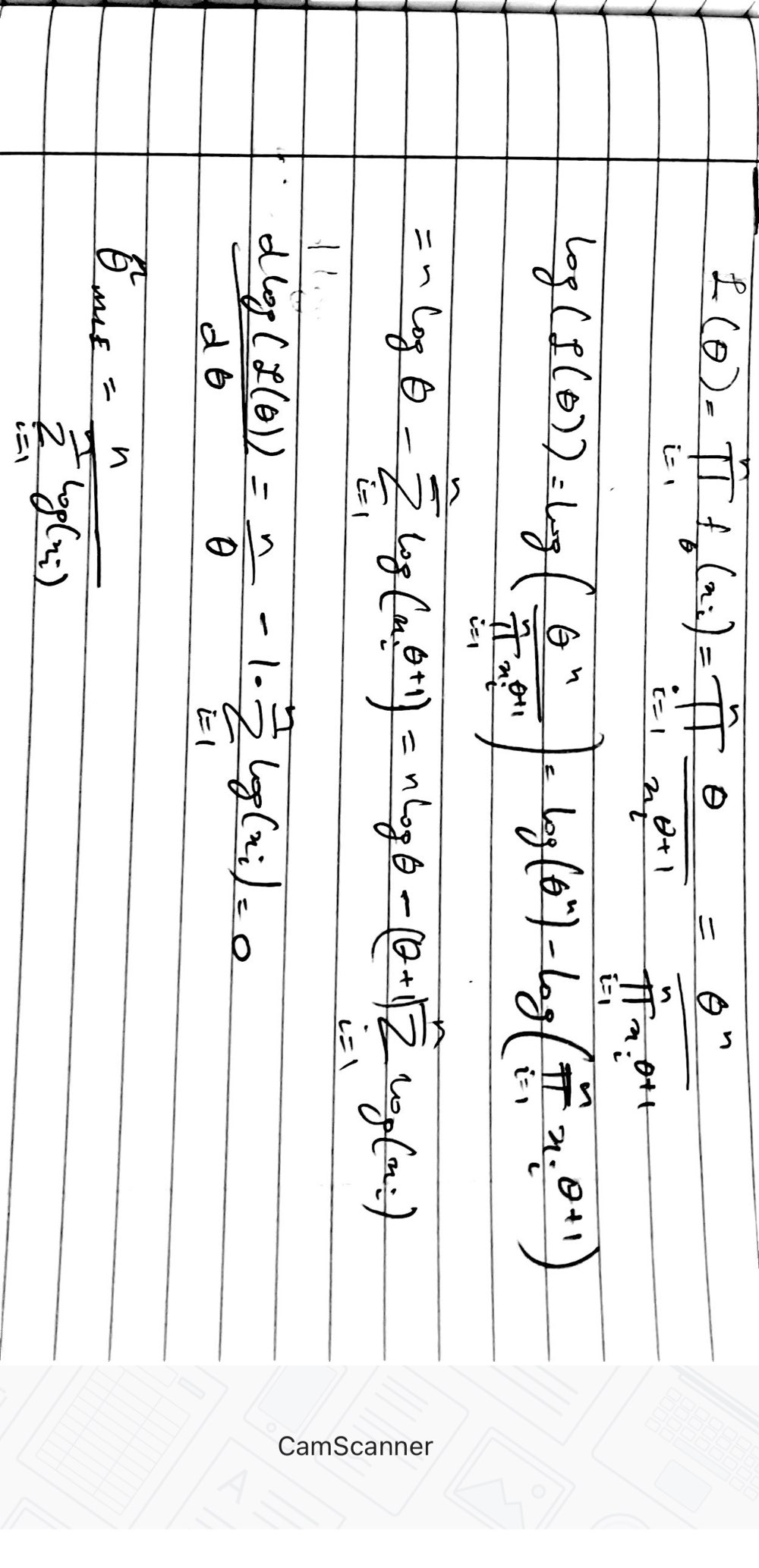
For low values of n like n=1, the percent difference between them is reliably within 5%. Here, we see estimator 1 beat estimator 2 about 50% of the time. But given larger values of n, such as n=30, the percent difference is well over 100%, and estimator2 is always better.

The value of theta doesn’t seem to make a large impact on the percent difference. It does of course affect the MSE, because the larger numbers being squared means a 2x increase in the uniform parameter means a 4x increase in the MSE.

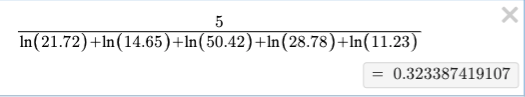
Overall for all but very small values of n, MLE is better.

**2.**

**a)** We find the likelihood function, log it, and then differentiate and solve for 0 to get Theta\_MLE.

****

**b)**

****

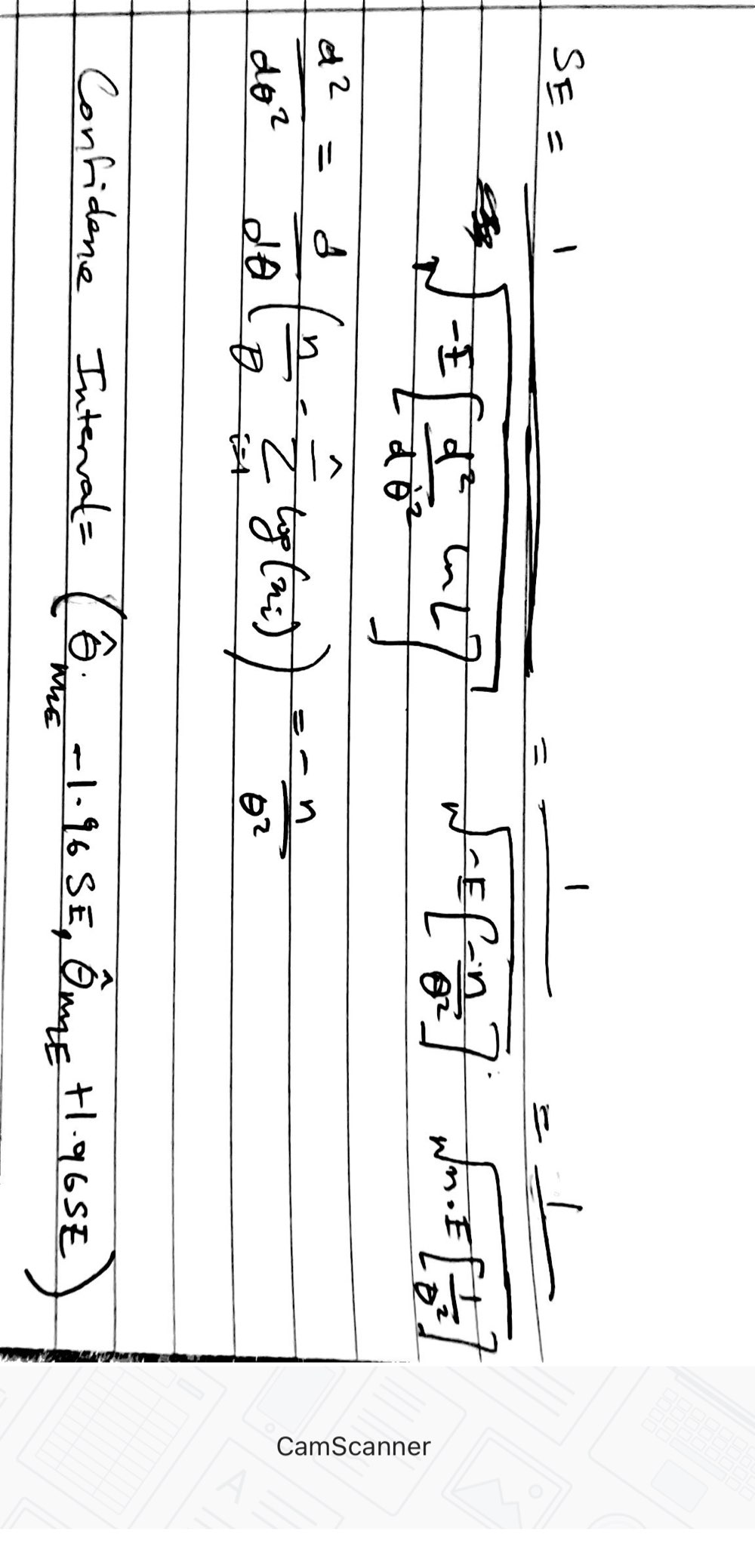
**c)**

[1] 0.3234375

Yes, these answers match up to the 4th decimal point. It is probably sufficient for our needs.

**d)**

The answer is something like this:

****

**Section 2**

**Question 1**

**n= 5**

**theta= 100**

**bigN=1000 #this won't change**

**MSE1=0**

**MSE2=0**

**estimator1=c()**

**estimator2=c()**

**values <- replicate(n = bigN, runif(n, 0, theta), simplify=FALSE) #create a giant 2D array bigN x n**

**#go row by row**

**for(i in values){**

**estimator1 <- c(estimator1, mean(i) \* 2) #c for combine**

**estimator2 <- c(estimator2, max(i))**

**}**

**for(i in range(estimator1)){**

**MSE1 = MSE1 + (theta-estimator1[i]) \*\* 2 #MSE formula**

**MSE2 = MSE2 + (theta-estimator2[i]) \*\* 2**

**}**

**print(MSE1/bigN) #average**

**print(MSE2/bigN)**

**print(MSE1/bigN) #average**

**print(MSE2/bigN)**

**Question 2**

**n=5 #number of datapoints**

**theta=2; #starting value, does not really matter**

**logfunction <- function(theta) {**

**mainsum=0 #this portion to compute sum log(xi)**

**for(xi in x){**

**mainsum= mainsum + log(xi) #defaults to natural log**

**}**

**result= (n\* log(theta) ) - ( (theta+1) \* mainsum )**

**return(result)**

**}**

**x= c(21.72, 14.65, 50.42, 28.78, 11.23)**

**answer= optim(par= theta, fn= logfunction, ,control=list(fnscale=-1)) #we call the last parameter because optim defaults to minimization**

**print(answer$par) #I only want the parameter value, not the loss function value**