**Statistical Methods for Data Science (Fall 2018)**

**Mini Project 3 (Solution)**

# Haoran Lou

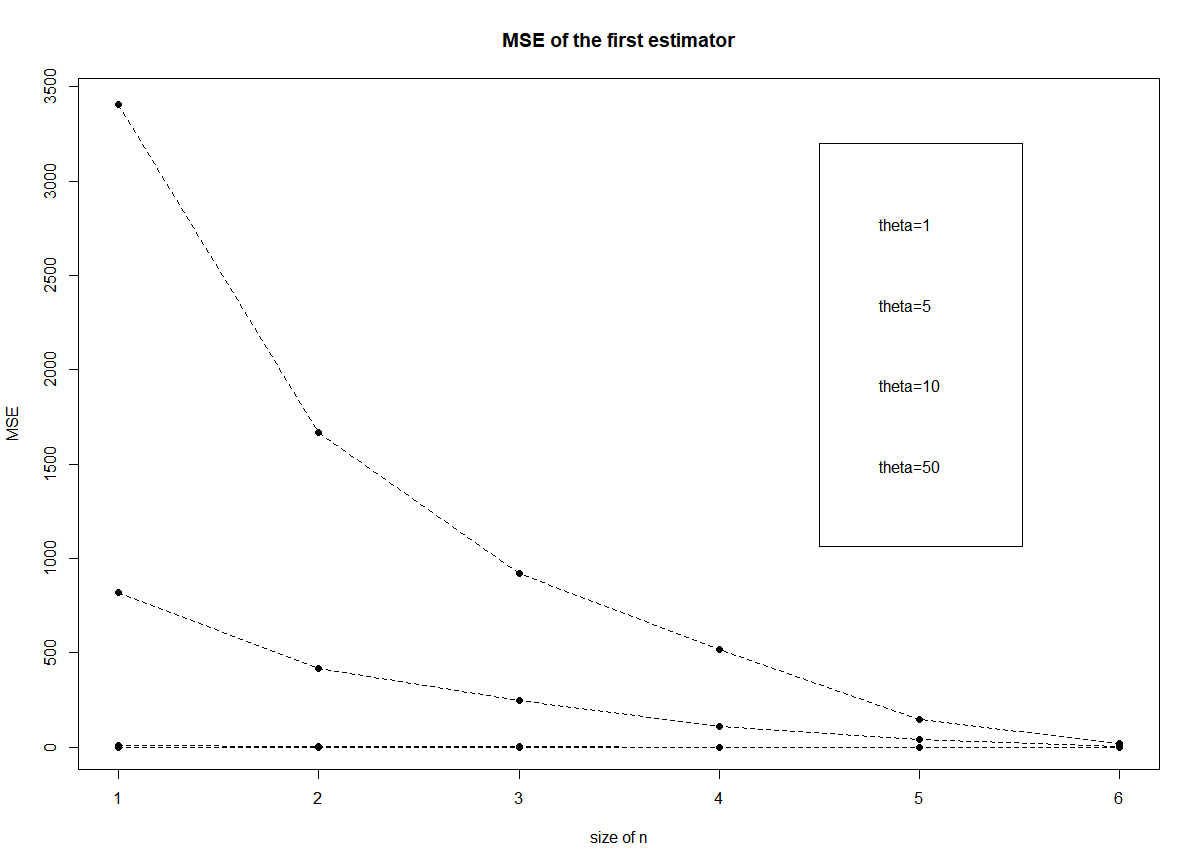
# hxl180012

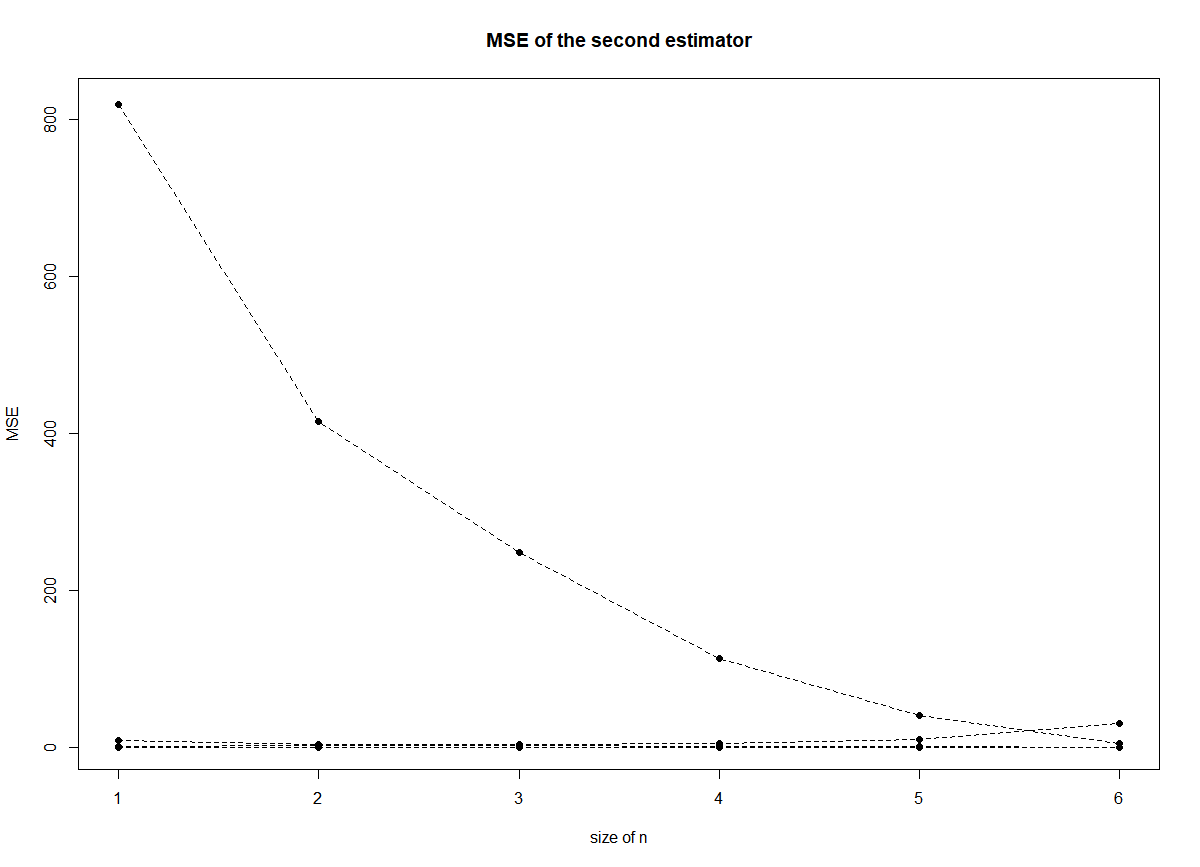
1.

(a) I use Monte Carlo to generate 1000 theta1 and theta2 heads and use the 1000 thetas to compute the mean squared error of theta1 and theta2, use ruinf() to generate theta, the mean of the 1000 compute will be the result of simulation.

(b)First I use combination of (1,1), use set seed(xxxx), to make sure the data used is same. And the mean squared error of theta1= 0.3458157, mean squared error of theta2=0.3364429 (computed data based on same set.seed).

(c)The summary plot is blow.( The xlab mark 1,2,3,4,5,6, stand for n=1,2,3,4,10,30).

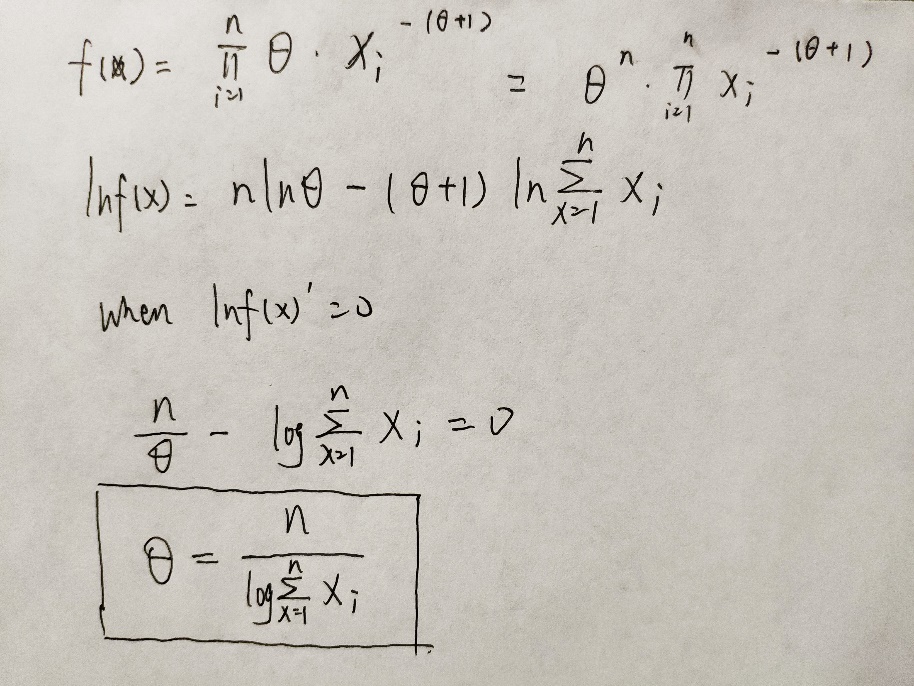




(d) Based on (c), may be the first estimator is better because when n going up the first estimator’s MSE becomes more stable than the second estimator. I think the answer depend on n. When n is large enough the answer will be more accurate to the real value.

2.

(a) The expression for maximum likelihood estimator of theta is below.



(b)

theta = 5/(log(21.72+14.65+50.42+28.78+11.23))=1.0325

(c)

The answer is not match. (I think it may because the numbers of X is less)

(d)

The stand error is sqrt(diag(solve(est11$hessian)))=0.1632

95% CI for theta=[0.0919,0.7317]

I think these approximations is going to be good.

1. #####problem 1######
2. ####################
3. #(a)
5. #############################################
7. #monte carlo simulation function
8. simulation=function(N,theta,n){
10. theta\_stat1=c(0)
11. theta\_stat2=c(0)
13. **for** (i **in** 1:N){
15. sample\_number=runif(n,min=0,max=theta)
17. theta1=max(sample\_number)
18. theta2=2\*mean(sample\_number)
20. theta\_stat1=c(theta\_stat1,theta1)
21. theta\_stat2=c(theta\_stat2,theta2)
23. }
25. mse1=mean((theta\_stat1[-1]-theta)^2)
26. mse2=mean((theta\_stat2[-1]-theta)^2)
28. **return**(c(mse1,mse2))
30. }
31. #use combination(1,1) to compute
32. simulation(1000,1,1)
33. #############################
35. #main function
36. n\_range=c(1,2,3,5,10,30)
37. theta\_range=c(1,5,50,100)
38. N=1000
40. MSE1=matrix(NA,nrow=length(n\_range),ncol=length(theta\_range))
41. MSE2=matrix(NA,nrow=length(n\_range),ncol=length(theta\_range))
43. coldata=MSE1
44. rowdata=MSE1
46. **for** (i **in** 1:length(n\_range)){
47. **for** (j **in** 1:length(theta\_range)){
49. n=n\_range[i]
50. theta=theta\_range[j]
52. mse=simulation(N,theta,n)
54. MSE1[i,j]=mse[1]
55. MSE2[i,j]=mse[2]
57. coldata[i,j]=theta\_range[j]
58. rowdata[i,j]=n\_range[i]
60. }
61. }
63. ###################plot of N and Theta#########################
65. #first estimator plot
66. mse1data=as.data.frame(MSE1)
67. mse1data=data.frame(x=n\_range,mse1data)
68. colnames(mse1data)=c('size\_of\_n','theta\_1','theta\_5','theta\_50','theta\_100')
70. plot(1:6,mse1data[,(length(theta\_range)+1)],type="l",lty=2,xlab="",ylab="")
71. points(1:6,mse1data[,(length(theta\_range)+1)],pch=16)
72. **for** (i **in** 1:(length(theta\_range)-1)){
73. lines(1:6,mse1data[,(i+1)],type="l",lty=2,xlab="",ylab="")
74. points(1:6,mse1data[,(i+1)],pch=16)
75. }
76. legend(4.5,3200,c("theta=1","theta=5","theta=10","theta=50"))
77. #xmark<-c(NA,"1","2","3","5","10","30",NA)
78. #axis(1,0:7,labels=xmark)#add x mark
79. #axis(2,0:3500)
81. title(main="MSE of the first estimator", ylab="MSE", xlab="size of n")

84. #second estimator plot
85. mse2data=as.data.frame(MSE2)
86. mse2data=data.frame(x=n\_range,mse1data)
87. colnames(mse2data)=c('size\_of\_n','theta\_1','theta\_5','theta\_50','theta\_100')
89. plot(1:6,mse2data[,(length(theta\_range)+1)],type="l",lty=2,xlab="",ylab="")
90. points(1:6,mse2data[,(length(theta\_range)+1)],pch=16)
91. **for** (i **in** 1:(length(theta\_range)-1)){
92. lines(1:6,mse2data[,(i+1)],type="l",lty=2,xlab="",ylab="")
93. points(1:6,mse2data[,(i+1)],pch=16)
94. }
95. legend(4.5,3200,c("theta=1","theta=5","theta=10","theta=50"))
96. #xmark<-c(NA,"1","2","3","5","10","30",NA)
97. #axis(1,0:7,labels=xmark)#add x mark
98. #axis(2,0:3500)
100. title(main="MSE of the second estimator", ylab="MSE", xlab="size of n")
101. #####problem 2#######
102. #####################
103. #b.##theta = 1.03250084
104. theta = 5/(log(21.72+14.65+50.42+28.78+11.23))
106. #c.##use optim funtion
107. ## negative log likelyhood function fun2
108. fun2 = function(theta,data1){
109. f1 = sum(log(theta/(data1^theta+1)))
110. **return**(-f1)
111. }
112. data1 = c(21.72, 14.65, 50.42, 28.78, 11.23)
114. ##use optim function
115. est11 = optim(par = 0.4, fn = fun2, method = "L-BFGS-B",
116. lower = 0, hessian = TRUE, dat = data1)
117. est11
119. #Standard Error
120. I = sqrt(diag(solve(est11**$hessian**)))
121. alpha = 1 - 0.95
122. z = qnorm(1 - alpha/2)
123. #lower and upper bound of confidencial interval.
124. lowerBound = est11**$par** - z \* I
125. upperBound = est11**$par** + z \* I
127. ConItv = c(lowerBound, upperBound)
128. ConItv