Lab1 - Group 6

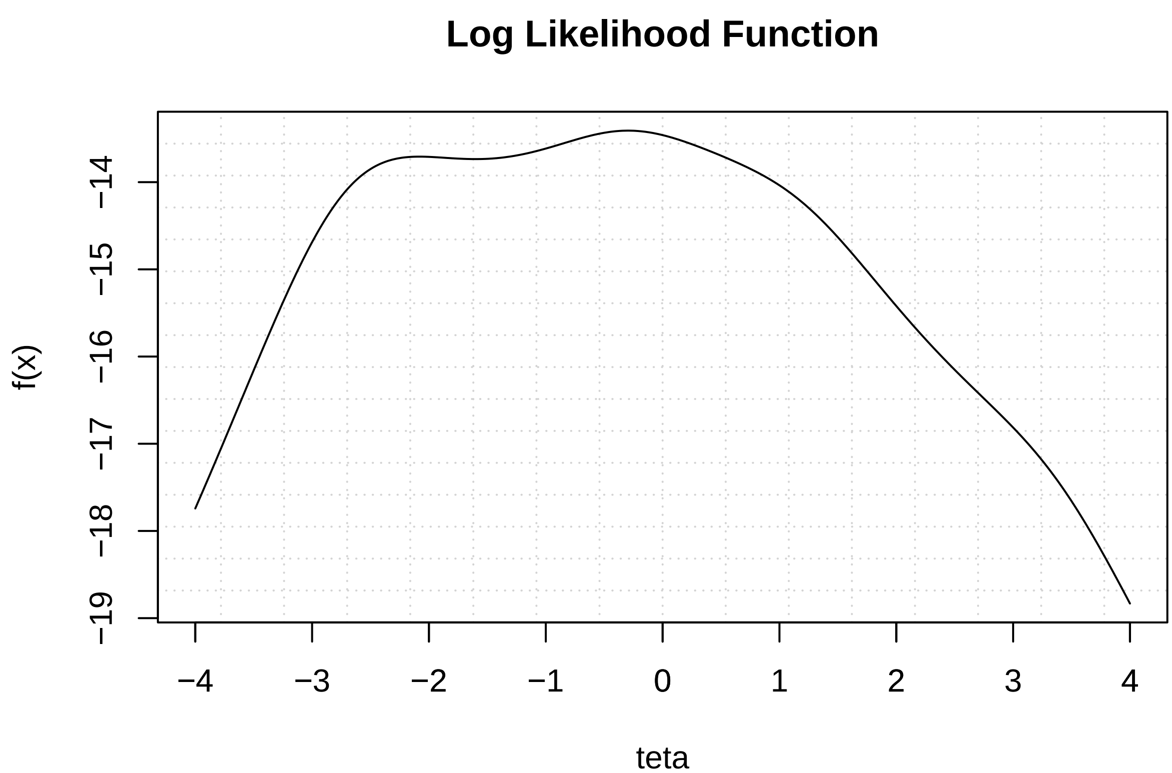
Ayesha Gamage - ayega981, N Muditha Cherangani-mudch175

2023-11-01

Question 1:

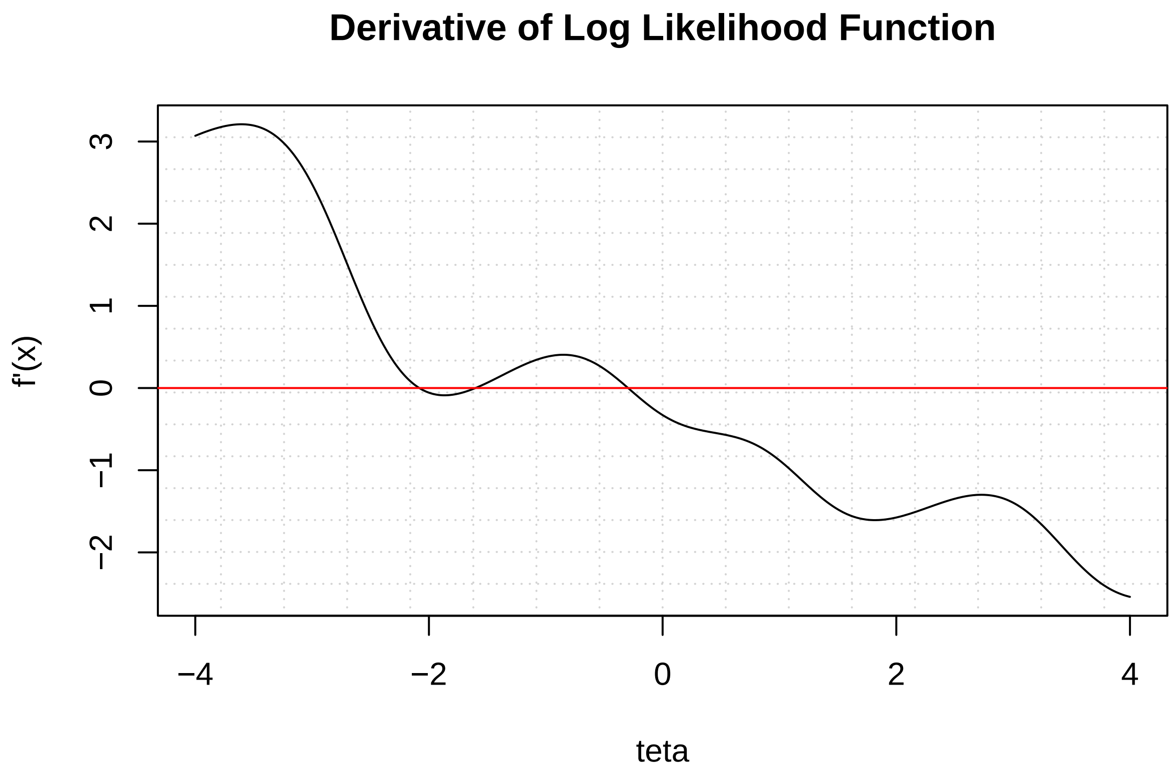
The log likelihood function,

$$f(x) = -nlog(\pi) - \sum_{i=1}^{n} log(1 + (x_i - \theta)^2)$$



$f(x)$ derivative with respect to θ is,

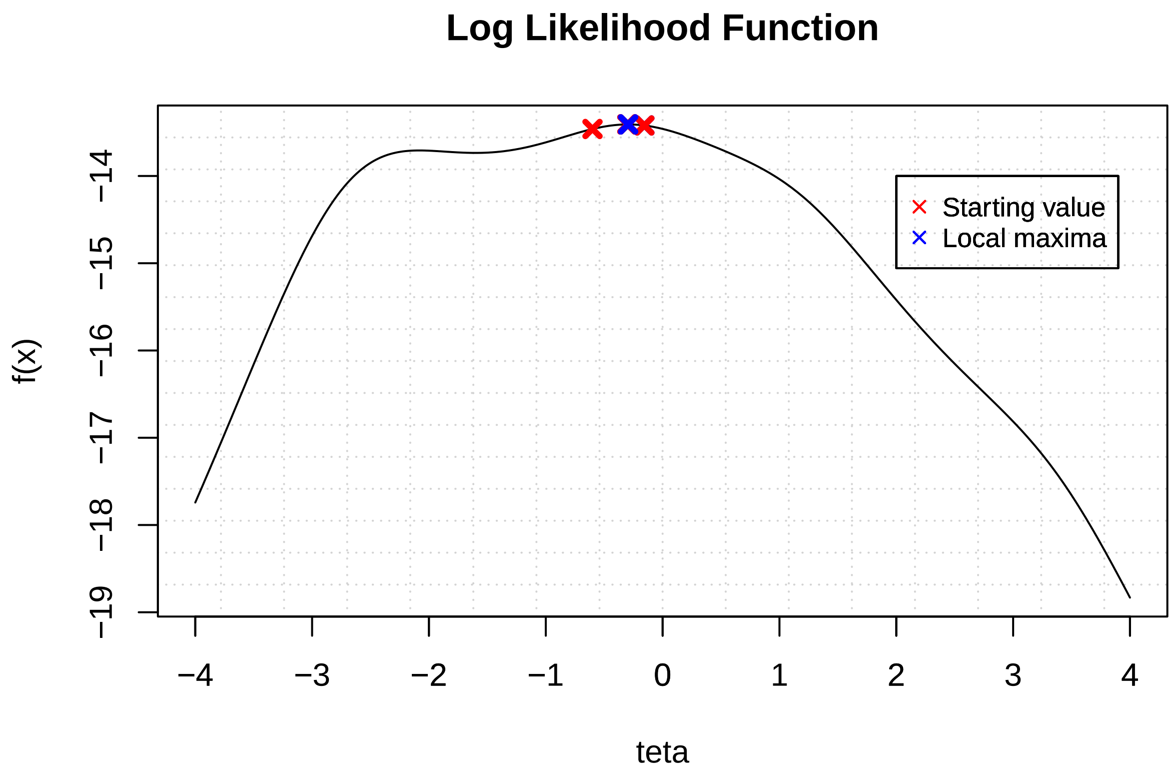
$$f'(x) = \sum_{i=1}^{n} \frac{2(x_i - \theta)}{1 + (x_i - \theta)^2}$$



A derivative of zero given when y=0.In this graph shown 3 points which derivative is 0.

b).Newton-Raphson Method,

In the question has first derivative of log likelihood function, using the “Deriv” r library got second derivative by derivative first derivative function. In the following r code, “d2fx” function return the second derivative.

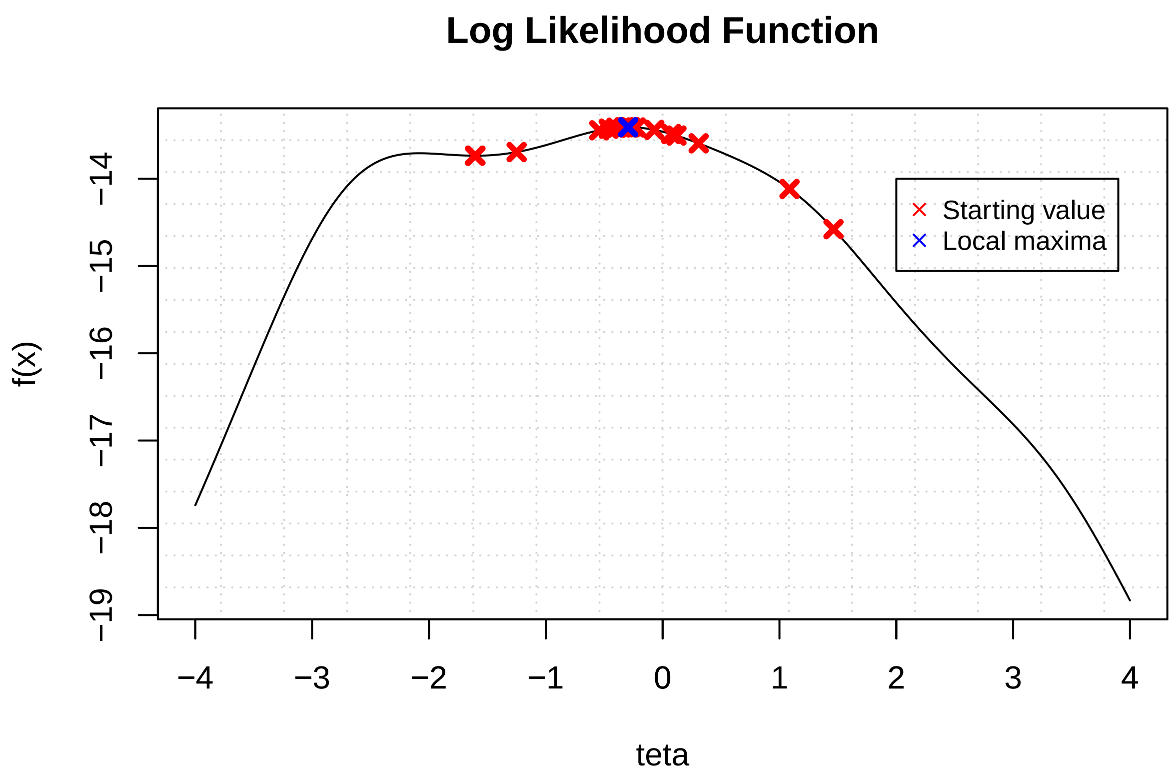



```
## [1] -0.2952455
```

c).

Two local maxima identified by using log likelihood and derivative log likelihood plots.Using the following example for newton functions found suitable starting values to lead local maxima based on the plots. The starting values between -0.6 to 0.1 lead one local maxima and values between -3 and -1.9 lead other local maxima. In here identify staring values which not lead local maxima such as ,-3.5,-3.7,-3.8,-3.9,-4 and starting values between 4 and 2. I got two local maxima values from the program.Those are -2.082124 and -0.2952455. When consider the log likelihood function plot i get the -0.2952455 as global maxima.

d).




```
## [1] -0.2952455
```

Question 2

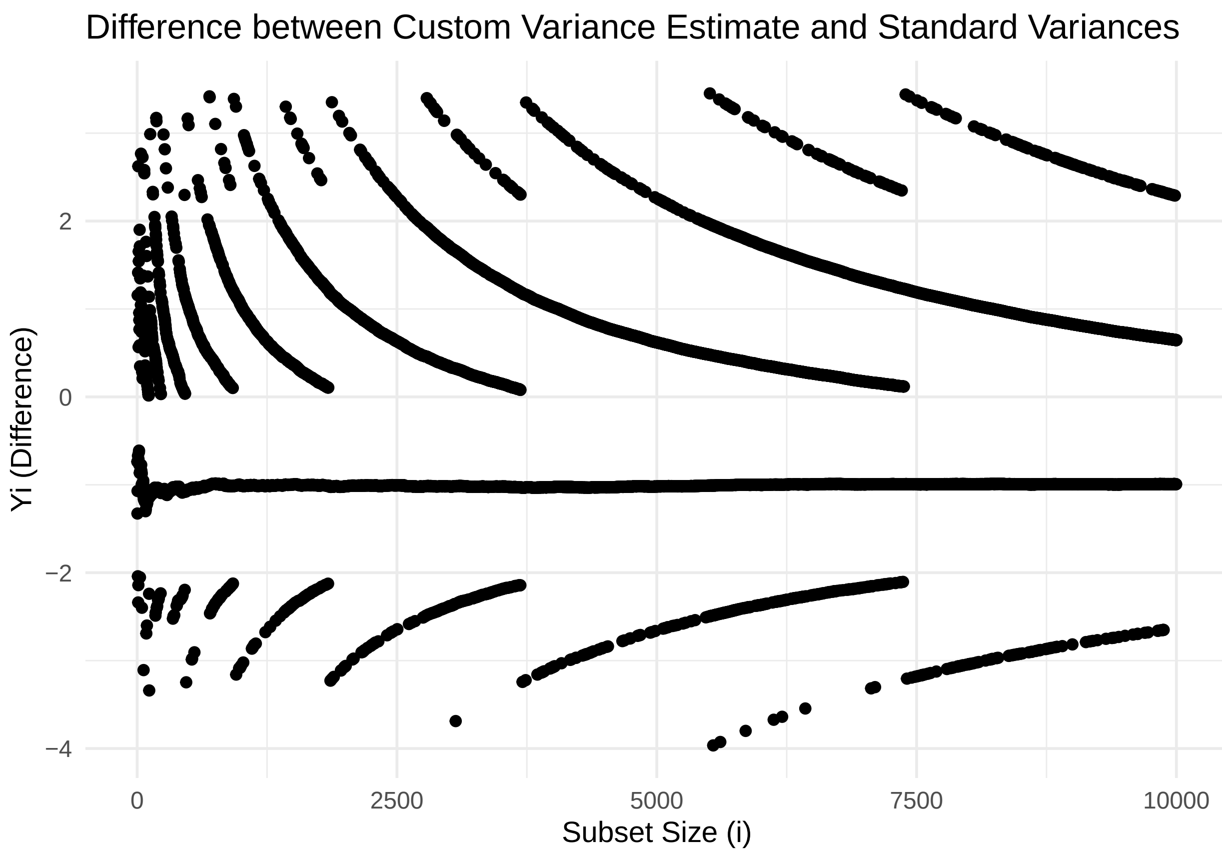
(a).R function, myvar

```r
#############Question 2, (a)####################
library(ggplot2)

myvar <- function(X) {
  n <- length(X)
  if (n <= 1) {
    stop("Sample size must be greater than 1.")
  }
  sum_xi_squared <- sum((X^2))
  sum_xi <- sum(X)

  myvar_x <- (sum_xi_squared - ((sum_xi^2)/n))/(n - 1)
  return(myvar_x)
}
```

(b). Generate a vector x = (x1, . . . , x10000) with 10000 random numbers

```r
#############Question 2, (b)#####################
x <- rnorm(10000, mean = 10^8, sd = 1)
```

(c). Plot the dependence Y_{i} on i.



With an increasing sample size, variance different should be close to 0.But, here variance difference is fluctuate between -4 and 4. The plot shows that. When consider function given variance values, there is a Precision error when divided by n and n-1.

(d).

To implement a better variance estimator, can use the sample variance formula, which is more numerically stable and equivalent to standard variances. In here used following formula for better variance,

$$bettervar = \sum_{i=1}^{n}(X - meanx)^2/(n-1)$$

```r
#############Question 2, (c)#####################
better_var <- function(X) {
  n <- length(X)
  if (n <= 1) {
```

```r
    stop("Sample size must be greater than 1.")
  }

  mean_x <- mean(X)
  better_var_x <- sum((X - mean_x)^2) / (n - 1)

  return(better_var_x)
}
# i <- 2:10000
# Y<-sapply(i, function(i){better_var(x[1:i])-var(x[1:i])})
# data <- data.frame(i, Y)

# ggplot(data, aes(x = i, y = Y)) +
#   geom_point() +
#   labs(x = "Subset Size (i)", y = "Yi (Difference)") +
#   ggtitle("Difference between Better Variance Estimate and  Standard Variances") +
#   theme_minimal()
```

Appendix: All r code for this report

```r
knitr::opts_chunk$set(echo = TRUE)


#############Question 1, (a)#####################
library("Deriv")

n <- 5
x <-c(-2.8, 3.4, 1.2, -0.3, -2.6)
teta <- seq(-4,4,by=0.01)

#log function
fx <- function(teta,x){
  return( -n*log(pi) - sum(log(1+((x-teta)^2))))
}
#print(fx(0.2,x))

#plot the log function
log_plot <- function(){
  points <- (sapply(teta, fx,x))
  plot(teta ,points,type = "l", xlab="teta",ylab = "f(x)",xlim =c(-4,4),panel.first = grid(nx=16),main =
  axis(side=1, at=c(-4:4))

}
log_plot()
#############Question 1, (a)#####################
#derivative function
dfx <- function(teta,x){
  return(sum((2*(x-teta))/(1+(x-teta)^2)))
}

#plot the derivative function
```

```
derivative_plot <- function(){
  points <- (sapply(teta, dfx,x))
  plot(teta ,points,type = "l", xlab="teta",ylab = "f'(x)",xlim =c(-4,4),panel.first = grid(nx=16),main
  abline(h=0,col="red")
}
derivative_plot()
############Question 1, (b)####################
#second derivative
d2fx <- function(teta,x){
  f <- expression(sum((2*(x-teta))/(1+(x-teta)^2)))
  d2val <-  Deriv(f,'teta')
  return(eval({teta; d2val}))

}
# Newton function
newton <- function(x0, eps=0.0001){
  xt  <- x0
  xt1 <- x0 + 2  # xt1 is x(t-1); starting value
  log_plot()
   while(abs(xt-xt1)>eps)
  {
    xt1 <- xt
    dfx_v <- dfx(xt1,x)
    d2fx_v <- d2fx(xt1,x)
    xt  <- xt1 - (dfx_v/d2fx_v)
    points(xt1, fx(xt1,x), col="red", pch=4, lwd=3)# starting value in plot
    points(xt, fx(xt,x), col="blue", pch=4, lwd=3)  # result in plot
    legend(2,-14,legend=c("Starting value","Local maxima"), col =c("red","blue"),ncol = 1,pch=4, cex=0.8
   }
  xt
}
newton(-0.6)

############Question 1, (d)####################
get_stating <- function(min_range,max_range,N,eps=0.0001){

  log_plot()
  rand_points <- runif(N,min_range,max_range)
  global_max<- rand_points[1]
  for(i in rand_points){
    x0<- i
    xt  <- x0
    xt1 <- x0 + 2
    found<-TRUE
    while(abs(xt-xt1)>eps)
    {
      prev_dif <- xt1-xt
      xt1 <- xt
      dfx_v <- dfx(xt1,x)
      d2fx_v <- d2fx(xt1,x)
      xt  <- xt1 - (dfx_v/d2fx_v)
      current_dif <-xt1-xt
```

```
      #check weather value in given range
      if((abs(xt)>4) && abs(prev_dif)<abs(current_dif)){
        found<-FALSE
        break
      }
       points(xt1, fx(xt1,x), col="red", pch=4, lwd=3)# starting value in plot
    }
    if(found && fx(xt,x)>fx(global_max,x)){
        local_max <- xt
    }
  }

    points(local_max, fx(local_max,x), col="blue", pch=4, lwd=3)  # result in plot
    legend(2,-14,legend=c("Starting value","Local maxima"), col =c("red","blue"),ncol = 1,pch=4, cex=0.8

  local_max

}
get_stating(-4,4,5)
############Question 2, (a)####################
library(ggplot2)

myvar <- function(X) {
  n <- length(X)
  if (n <= 1) {
    stop("Sample size must be greater than 1.")
  }
  sum_xi_squared <- sum((X^2))
  sum_xi <- sum(X)

  myvar_x <- (sum_xi_squared - ((sum_xi^2)/n))/(n - 1)
  return(myvar_x)
}
############Question 2, (b)####################
x <- rnorm(10000, mean = 10^8, sd = 1)
options(digits = 22)
i <- 2:10000
Y<-sapply(i, function(i){myvar(x[1:i])-var(x[1:i])})
data <- data.frame(i, Y)

ggplot(data, aes(x = i, y = Y)) +
  geom_point() +
  labs(x = "Subset Size (i)", y = "Yi (Difference)") +
  ggtitle("Difference between Custom Variance Estimate and Standard Variances") +
  theme_minimal()
############Question 2, (c)####################
better_var <- function(X) {
  n <- length(X)
  if (n <= 1) {
    stop("Sample size must be greater than 1.")
  }

  mean_x <- mean(X)
```

```r
  better_var_x <- sum((X - mean_x)^2) / (n - 1)

  return(better_var_x)
}
# i <- 2:10000
# Y<-sapply(i, function(i){better_var(x[1:i])-var(x[1:i])})
# data <- data.frame(i, Y)

# ggplot(data, aes(x = i, y = Y)) +
#   geom_point() +
#   labs(x = "Subset Size (i)", y = "Yi (Difference)") +
#   ggtitle("Difference between Better Variance Estimate and  Standard Variances") +
#   theme_minimal()
```