

2022년 2학기

Computational Statistics

HW#2

222STG10

김희숙

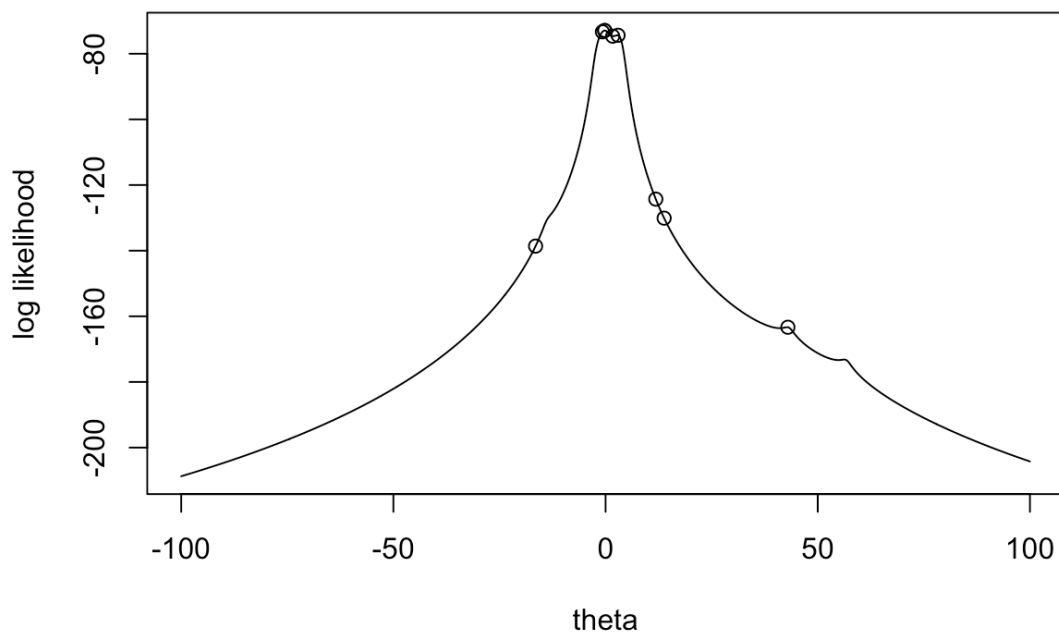
연습문제 2.1

The following data are an i.i.d. sample from a $\text{Cauchy}(\theta, 1)$ distribution: 1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44, 3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75, 0.27, 43.21.

- a) Graph the log likelihood function. Find the MLE for θ using the Newton–Raphson method. Try all of the following starting points: -11, -1, 0, 1.5, 4, 4.7, 7, 8, and 38. Discuss your results. Is the mean of the data a good starting point?

Starting points	Mle_theta	Log ikelihood
-11	-16.4933088	-138.61111
-1	-0.2414496	-72.91951
0	-0.1963366	-72.91584
1.5	1.6756107	-74.64136
4	2.9389009	-74.37868
4.7	-0.7466157	-73.32159
7	11.8353926	-124.29561
8	13.8020203	-130.09316
38	42.9572680	-163.33032

Graph

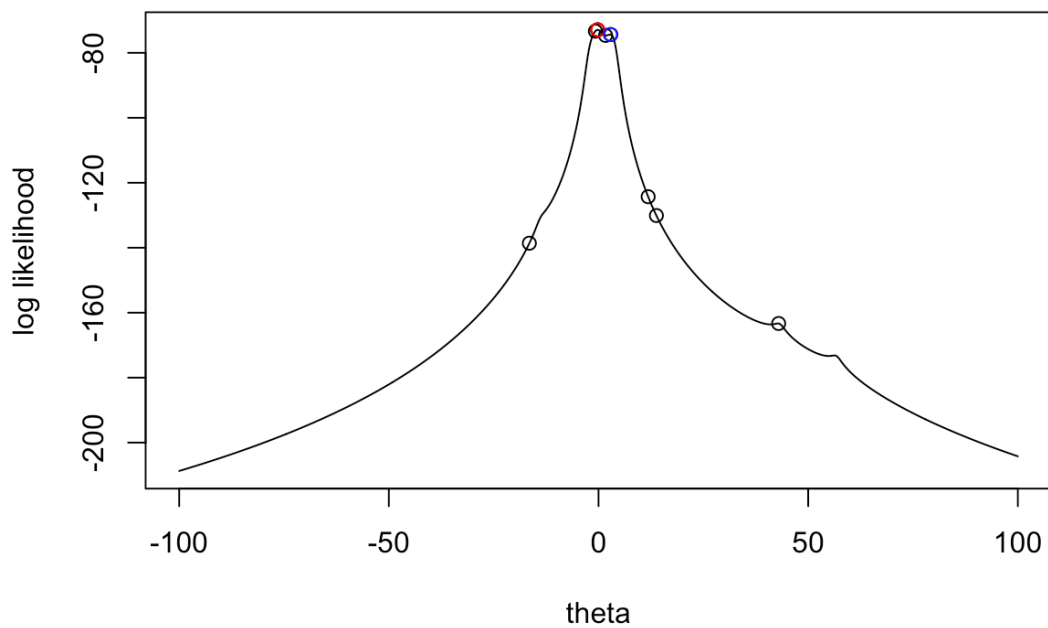


Cauchy 분포에 대해서 mean of the data 은 good starting point 가 아니라고 볼 수 있다.

- b) Apply the bisection method with starting points -1 and 1. Use additional runs to illustrate manners in which the bisection method may fail to find the global maximum.

	x0: mle_theta	g(bisect()):Log ikelihood	Graph point
bisect(-1,1)	-0.1922866	-72.91582	col = red
bisect(1,3)	2.817472	-74.36046	col = blue

Graph



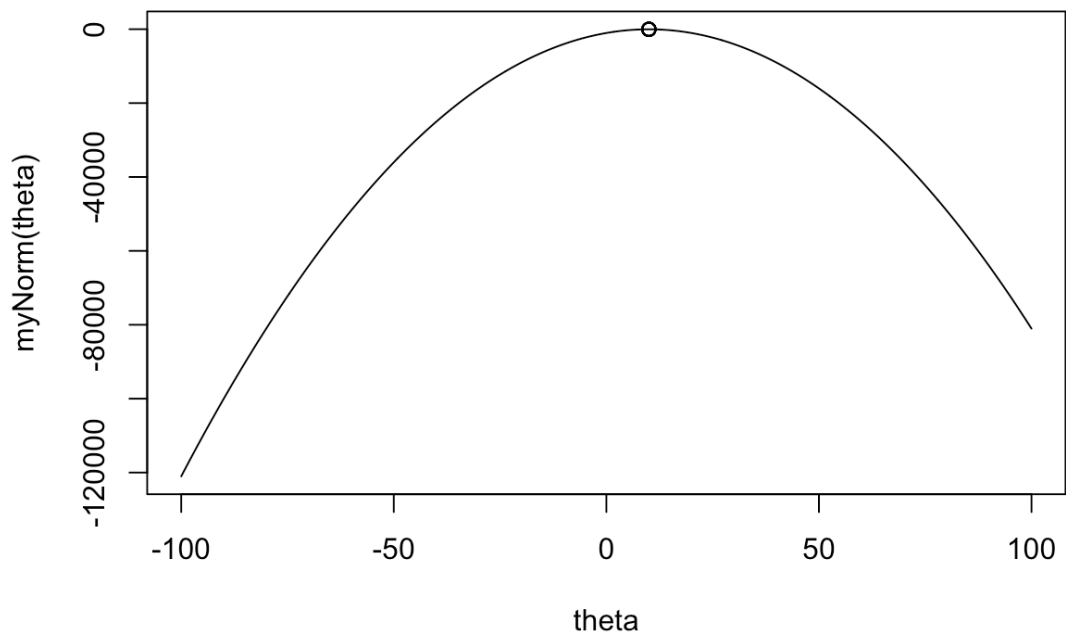
- d) From starting values of $(\theta^{(0)}, \theta^{(1)}) = (-2, -1)$, apply the secant method to estimate θ . What happens when $(\theta^{(0)}, \theta^{(1)}) = (-3, 3)$, and for other starting choices?

	x0: mle_theta	g(secant()):Log ikelihood
secant(-2,-1)	-0.1922866	-72.91582
secant(-3,3)	2.817472	-74.36046

- e) Use this example to compare the speed and stability of the Newton–Raphson method, bisection, fixed-point iteration, and the secant method. Do your conclusions change when you apply the methods to a random sample of size 20 from a $N(\theta, 1)$ distribution?

Starting points	Mle_theta	Log ikelihood
-11	9.997636	-29.73224
-1		
0		
1.5		
4		
4.7		
7		
8		
38		

Graph

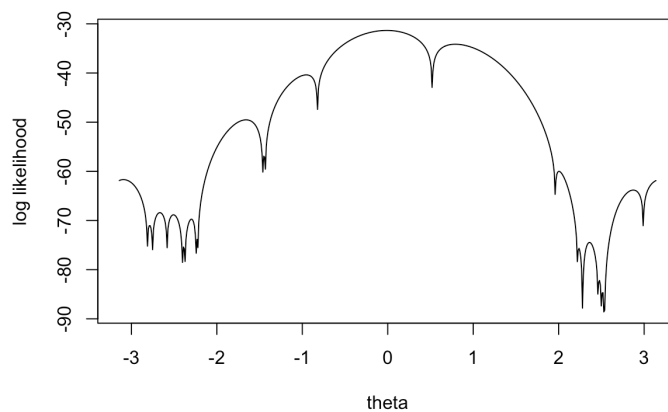


	x0:mle_theta	g(x0): Log ikelihood	niter	Speed(Time complexity)
newton(g,0)	9.997636	-29.732244	3	0.000438
bisect(g,0,11)	9.997636	-29.732244	36	0.00538
secant(g,4,11)	9.997636	-29.732244	3	0.000754

연습문제 2.2

Consider the density $f(x) = [1 - \cos\{x - \theta\}]/2\pi$ on $0 \leq x \leq 2\pi$, where θ is a parameter between $-\pi$ and π . The following i.i.d. data arise from this density: 3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96, 2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52, 2.50. We wish to estimate θ .

- a) Graph the log likelihood function between $-\pi$ and π .



- b) Find the method-of-moments estimator of θ .

$$E(X) = \int_0^{2\pi} \frac{x(1 - \cos(x - \theta))}{2\pi} dx$$

$$= \pi + \sin\theta = \bar{x}$$

$$\sin\theta = \bar{x} - \pi$$

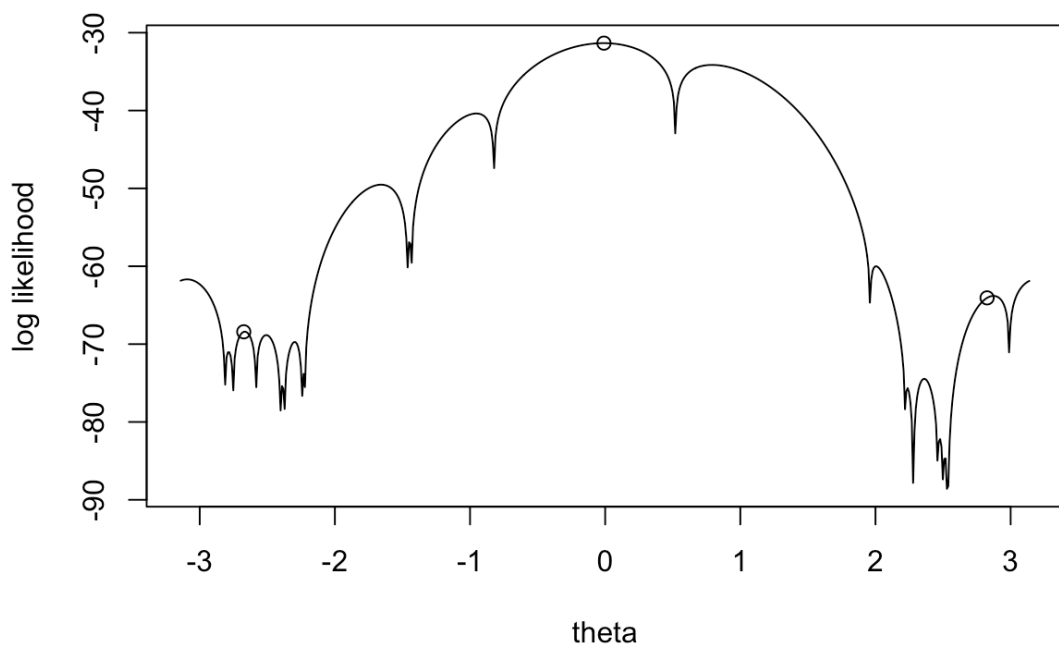
$$\therefore \theta = \sin^{-1}(\bar{x} - \pi) = 0.0584$$

따라서, mme = 0.0584

- c) Find the MLE for θ using the Newton–Raphson method, using the result from (b) as the starting value. What solutions do you find when you start at -2.7 and 2.7 ?

starting value	Mle_theta	Log ikelihood
mme(=0.0584)	-0.009102464	-31.343
-2.7	-2.674114	-68.41969
2.7	2.825724	-64.06015

Graph



- d) Repeat part(c) using 200 equally spaced starting values between $-\pi$ and π . Partition the interval between $-\pi$ and π into sets of attraction. In other words, divide the set of starting values into separate groups, with each group corresponding to a separate unique outcome of the optimization (a local mode). Discuss your results.

start_point: seq(-pi,pi, length.out = 200)
which.max(mle_theta)=100; mle_theta= -0.0119665; log ikelihood= -31.34291
start_point: gr1 <- start_point[1:100]

<code>which.max(mle_theta)=100; mle_theta= -0.0119665; log ikelihood= -31.34291</code>
<code>start_point: gr2 <- start_point[101:200]</code>
<code>which.max(mle_theta)=1; mle_theta= -0.01161673; log ikelihood= -31.34291</code>

$-\pi \sim \pi$ 까지 200 개의 초기값의 결과와 이 초기값들을 그룹으로 나누어 각 그룹에 대해 method 를 진행했을 때 값이 똑같은 것을 확인 할 수 있다.

- e) Find two starting values, as nearly equal as you can, for which the Newton–Raphson method converges to two different solutions.

starting value	Mle_theta	Log ikelihood
2.1	1.978709	-60.58919
2.2	2.169329	-67.82944

starting value는 0.1 차이로 매우 가깝게 위치하나, Log ikelihood는 7차이로 다르게 converge한다.

Code appendix

2.1 #####

a

```
myCauchy<-function(theta){  
  
  x<-c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44, 3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75, 0.27, 43.21)  
  
  loglik = function(theta,x=x){  
  
    sum(-log(pi)-log(1+(x-theta)^2))  
  
  }  
  
  clv = rep(NA,length(theta))  
  
  if (length(theta)>1) {  
  
    clv = theta  
  
    for(i in 1:length(theta)) {  
  
      clv[i] = loglik(theta[i],x)  
  
    }  
  
  }  
  
  else if (length(theta) == 1) {  
  
    clv = loglik(theta, x)  
  
  }  
  
  return(clv)  
}
```

```
theta = seq(-100,100,length=1000)
```

```
plot(theta,myCauchy(theta),type="l", ylab='log likelihood') #Graph the log likelihood function
```

####

Cauchy ikelihood function

```
g = function(theta,x=x){  
  
  sum(-log(pi)-log(1+(x-theta)^2))  
  
}
```

```

g_df1 = function(theta,x){ # 1번미분
  return(sum(2*(x-theta)/(1+(x-theta)^2)))
}

g_df2 = function(theta,x){ # 2번미분
  return(-2*sum((1-(x-theta)^2)/(1+(x-theta)^2)^2))
}

x<-c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44, 3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75, 0.27, 43.21)

x0 <- c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38) #theta: starting points

# Newton-Raphson method
maxiter<-1000
threshold<-10^(-10)

#업데이트된theta 그릇
theta = rep(NA,length(x0))
mle_theta = rep(NA,length(x0))

for(i in 1:length(x0)){
  niter<-0
  err<-1
  while(niter<=maxiter && err >= threshold){ #update
    #oldx0<-x0
    oldx0<-x0[i]
    oldx0 <- oldx0 - g_df1(oldx0,x)/g_df2(oldx0,x)
    #update error and niter
    err<-abs(g_df1(oldx0,x))#abs(oldx0-x0)
    niter<-niter+1
  }
  theta[i] <- oldx0 #theta mle
}

```



```

mle_theta[i] <- myCauchy(oldx0) # log likelihood theta mle
}

print(theta)

print(mle_theta)

points(theta,mle_theta)


#### b

# Bisection

bisect<-function(a,b){

  maxiter<-1000

  threshold<-10^(-10)

  err<-1

  niter<-0

  x0<-(a+b)/2

  while ( niter<=maxiter && err >= threshold){ #update interval

    if (g_df1(-1,x)*g_df1(x0,x) <=0) {b<-x0}

    else {a<-x0}

    #update x

    oldx0<-x0

    x0<-(a+b)/2 #새로 업데이트

    #update error and niter

    err<-abs(oldx0-x0)

    niter<-niter+1

  }

  return(c(x0))

}

bisect(-1,1) #starting points -1 and 1.

print(g(bisect(-1,1),x=x)) #-72.91582

points(bisect(-1,1),g(bisect(-1,1),x=x), col = "red")

```

```
bisect(1,3) #찾지 못하는 경우
```

```
print(g(bisect(1,3),x=x))
```

```
points(bisect(1,3),g(bisect(1,3),x=x), col = "blue")
```

```
#### d
```

```
secant<-function(x0,x1){
```

```
  maxiter<-1000
```

```
  threshold<-10^(-10)
```

```
  err<-1
```

```
  niter<-0
```

```
  while ( niter<=maxiter && err >= threshold){
```

```
    #update x
```

```
    newx0<-x1
```

```
    x1<- x1 - g_df1(x1,x)*((x1-x0)/(g_df1(x1,x)-g_df1(x0,x))) # new
```

```
    #update error and niter
```

```
    err<-abs(newx0-x0) #x1 - x0
```

```
    niter<-niter+1
```

```
    x0 <- newx0
```

```
  }
```

```
  return(c(x1))
```

```
}
```

```
secant(-2,-1) #-0.1922866
```

```
print(g(secant(-2,-1),x=x))
```

```
secant(-3,3) # 2.817472
```

```
print(g(secant(-3,3),x=x))
```

```
#### e
```

```
library("numDeriv")
```

```
n<-20
```

```
x <- rnorm(n, mean=10, sd=1) #theta=10
```

```
# Normal
```

```
g <- function(theta){
```

```
  -n*log(sqrt(2*pi))-sum((x-theta)^2/2)
```

```
}
```

```
myNorm<-function(theta){
```

```
  #x<-c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44, 3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75, 0.27, 43.21)
```

```
  loglik = function(theta,x=x){
```

```
    -n*log(sqrt(2*pi))-sum((x-theta)^2/2) #sum(-log(pi)-log(1+(x-theta)^2))
```

```
  }
```

```
  nlv = rep(NA,length(theta))
```

```
  if (length(theta)>1) {
```

```
    nlv = theta
```

```
    for(i in 1:length(theta)) {
```

```
      nlv[i] = loglik(theta[i],x)
```

```
    }
```

```
  }
```

```
  else if (length(theta) == 1) {
```

```
    nlv = loglik(theta, x)
```

```
  }
```

```
  return(nlv)
```

```
}
```

```
theta = seq(-100,100,length=1000)
```

```
plot(theta,myNorm(theta),type="l") #Graph the log likelihood function
```

```
x0 = c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38) #theta
```

```
maxiter<-1000
```

```
threshold<-10^(-10)
```

```
#업데이트된theta 그릇
```

```
theta = rep(NA,length(x0))
```

```
mle_theta = rep(NA,length(x0))
```

```
for(i in 1:length(x0)){
```

```
  niter<-0
```

```
  err<-1
```

```
  oldx0<-x0[i]
```

```
  x00<-x0[i]
```

```
  while(niter<=maxiter && err >= threshold){ #update
```

```
    oldx0<-x00
```

```
    x00 <- oldx0 - genD(g,x00)$D[1]/genD(g,x00)$D[2]
```

```
    #update error and niter
```

```
    err<-abs(oldx0-x00) #abs(genD(g,x0)$D[1])#abs(oldx0-x0)
```

```
    niter<-niter+1
```

```
  }
```

```
  theta[i] <- x00
```

```
  mle_theta[i] <- myNorm(x00)
```

```
  #print(i)
```

```
}
```

```
print(theta)
```

```
print(mle_theta)
```

```
points(theta,mle_theta)
```

```
####
```

```
# Newton's
```

```
newton<-function(g,x0){
```

```
  maxiter<-1000
```

```

threshold<-10^(-10)

err<-1

niter<-0

while ( niter<=maxiter && err >= threshold){ #update

  oldx0<-x0

  x0 <- oldx0 - genD(g,oldx0)$D[1]/genD(g,oldx0)$D[2]

  #update error and niter

  err<-abs(oldx0-x0)

  niter<-niter+1

}

return(c(x0, g(x0), niter))
}

newton(g,0)

bisect<-function(g,a,b){

  maxiter<-1000

  threshold<-10^(-10)

  err<-1

  niter<-0

  x0<-(a+b)/2

  while ( niter<=maxiter && err >= threshold){ #update interval

    if (genD(g,a)$D[1]*genD(g,x0)$D[1] <=0) {b<-x0}

    else {a<-x0}

    #update x

    oldx0<-x0

    x0<-(a+b)/2 #새로 업데이트

    #update error and niter

    err<-abs(oldx0-x0)

    niter<-niter+1

  }

```

```

    return(c(x0, g(x0), niter))
}

bisect(g,0,11)

secant<-function(g,x0,x1){
  maxiter<-1000
  threshold<-10^(-10)
  err<-1
  niter<-0

  while ( niter<=maxiter && err >= threshold){
    #update x
    newx0<-x1
    x1 <- x1 - genD(g,x1)$D[1]*((x1-x0)/(genD(g,x1)$D[1]-genD(g,x0)$D[1])) # new
    #update error and niter
    err<-abs(newx0-x0) #x1 - x0
    niter<-niter+1
    x0 <- newx0
  }
  return(c(x1, g(x1), niter))
}

secant(g,4,11)

system.time(for(i in 1:1000){newton(g,0)})
system.time(for(i in 1:1000){bisect(g,0,11)})
system.time(for(i in 1:1000){secant(g,4,11)})

```

```

2.2 #####

#### a

x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96, 2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52, 2.50)

f <- function(x){

  -n*log(2*pi) + sum(log(1-cos(x-theta)))

  #sum(log((1-cos(x-theta))/(2*pi)))

  #((1-cos(x-theta))/2*pi)

}

f_df1 <- function(theta,x=x){

  sum(-sin(x-theta)/(1-cos(x-theta)))# loglikelihood 1차 미분

}

f_df2 <- function(theta,x=x){

  sum(1/(cos(x-theta)-1))# loglikelihood 2차 미분

}


myCos <- function(theta){

  #x <- seq(0,2*pi,0.01)

  x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96, 2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52, 2.50)

  n <- length(x)

  loglik = function(theta,x=x){

    -n*log(2*pi) + sum(log(1-cos(x-theta)))

    #sum(log(((1-cos(x-theta))/2*pi)))

  }

  clv = rep(NA,length(theta))

  if (length(theta)>1) {

    clv = theta

    for(i in 1:length(theta)) {

      clv[i] = loglik(theta[i],x)

    }

  }

}

```

```

else if (length(theta) == 1) {

  clv = loglik(theta, x)

}

return(clv)

}

```

```

theta <- seq(-pi, pi, 0.01)

plot(theta, myCos(theta), type="l", ylab='log likelihood')

```

```

##### b

mme <- 0.0584

```

```

##### c

start_point <- c(mme, -2.7, 2.7)

```

```

# Newton's

maxiter<-1000

threshold<-10^(-10)

theta = rep(NA,length(start_point)) #mle_theta

mle_theta = rep(NA,length(start_point)) #mle를 넣은 loglik

for(i in 1:length(start_point)){

  err<-1

  niter<-0

  while (niter<=maxiter && err >= threshold){

    #update theta

    old_theta <- start_point[i]

    new_theta <- old_theta - f_df1(old_theta,x)/f_df2(old_theta,x)

    #update error and niter

    err <- abs(f_df1(old_theta,x))

    niter <- niter+1
  }
}

```



```

    }

    theta[i] <- new_theta

    mle_theta[i] <- myCos(new_theta)
}

theta

mle_theta

points(theta,mle_theta)

##### d

start_point <- seq(-pi,pi, length.out = 200)

# Newton's

maxiter<-1000

threshold<-10^(-10)

theta = rep(NA,length(start_point)) #mle_theta

mle_theta = rep(NA,length(start_point)) #mle를 넣은 loglik

for(i in 1:length(start_point)){

    err<-1

    niter<-0

    while (niter<=maxiter && err >= threshold){

        #update theta

        old_theta <- start_point[i]

        new_theta <- old_theta - f_df1(old_theta,x)/f_df2(old_theta,x)

        #update error and niter

        err <- abs(f_df1(old_theta,x))

        niter <- niter+1

    }

    theta[i] <- new_theta

    mle_theta[i] <- myCos(new_theta)

}

which.max(mle_theta) #100

```

```

theta[100]

mle_theta[100]

##

#2개 그룹으로 구분

gr1 <- start_point[1:100]

gr2 <- start_point[101:200]

start_point <- gr1

# Newton's

maxiter<-1000

threshold<-10^(-10)

theta = rep(NA,length(start_point)) #mle_theta

mle_theta = rep(NA,length(start_point)) #mle를 넣은 loglik

for(i in 1:length(start_point)){

  err<-1

  niter<-0

  while (niter<=maxiter && err >= threshold){

    #update theta

    old_theta <- start_point[i]

    new_theta <- old_theta - f_df1(old_theta,x)/f_df2(old_theta,x)

    #update error and niter

    err <- abs(f_df1(old_theta,x))

    niter <- niter+1

  }

  theta[i] <- new_theta

  mle_theta[i] <- myCos(new_theta)

}

```

```
which.max(mle_theta) #100
```

```
start_point <- gr2
```

```
# Newton's
```

```
maxiter<-1000
```

```
threshold<-10^(-10)
```

```
theta = rep(NA,length(start_point)) #mle_theta
```

```
mle_theta = rep(NA,length(start_point)) #mle를 넣은 loglik
```

```
for(i in 1:length(start_point)){
```

```
  err<-1
```

```
  niter<-0
```

```
  while (niter<=maxiter && err >= threshold){
```

```
    #update theta
```

```
    old_theta <- start_point[i]
```

```
    new_theta <- old_theta - f_df1(old_theta,x)/f_df2(old_theta,x)
```

```
    #update error and niter
```

```
    err <- abs(f_df1(old_theta,x))
```

```
    niter <- niter+1
```

```
  }
```

```
  theta[i] <- new_theta
```

```
  mle_theta[i] <- myCos(new_theta)
```

```
}
```

```
which.max(mle_theta) #1
```

```
theta[1]
```

```
mle_theta[1]
```

```
##### e 스타팅 포인트가 가깝지만 다르게 수렴하는 경우 찾기
```

```
## 여러 값 넣어가면서 찾기
```

```

start_point <- c(2.1,2.2)

# Newton's

maxiter<-1000

threshold<-10^(-10)

theta = rep(NA,length(start_point)) #mle_theta

mle_theta = rep(NA,length(start_point)) #mle를 넣은 loglik


for(i in 1:length(start_point)){

  err<-1

  niter<-0

  while (niter<=maxiter && err >= threshold){

    #update theta

    old_theta <- start_point[i]

    new_theta <- old_theta - f_df1(old_theta,x)/f_df2(old_theta,x)

    #update error and niter

    err <- abs(f_df1(old_theta,x))

    niter <- niter+1

  }

  theta[i] <- new_theta

  mle_theta[i] <- myCos(new_theta)

}

theta

mle_theta

```