

# Computational Statistics. Chapter 1: Continuous optimization.

## Solution of exercise 1

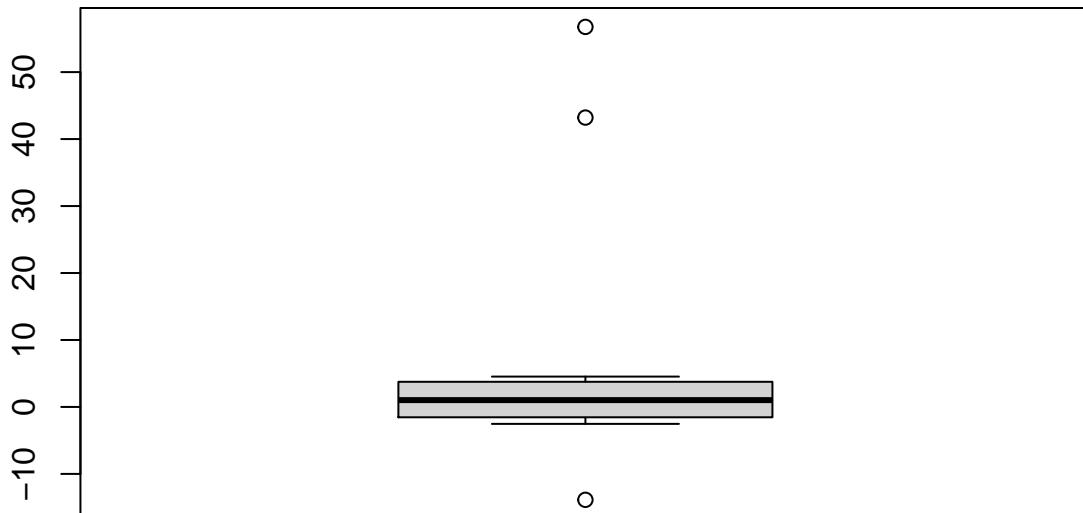
Thierry Denoeux

8/18/2021

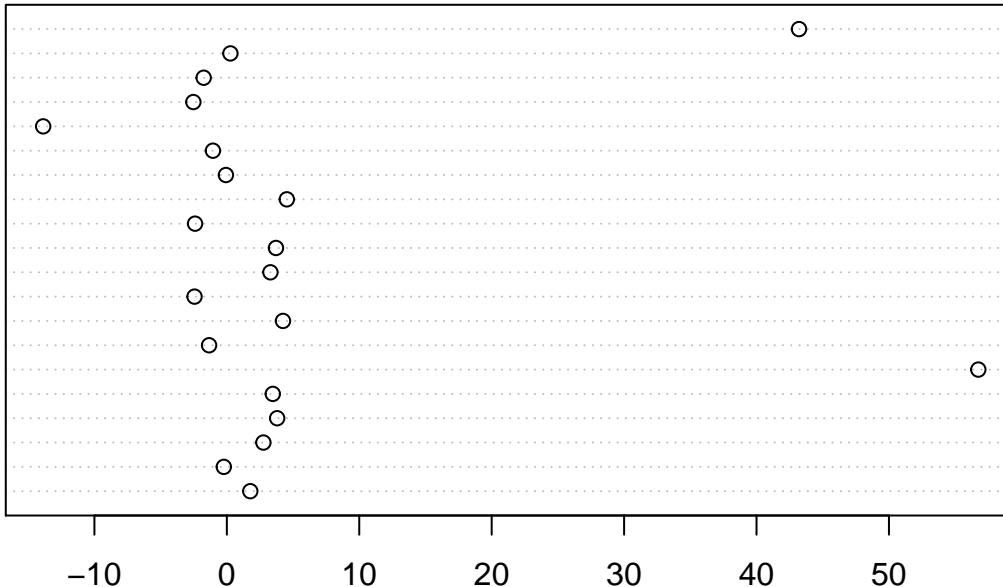
### Exercise 1

#### Question a

```
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)  
n<- length(x)  
  
boxplot(x)
```



```
dotchart(x)
```



## Question b

We first write a function to compute the log-likelihood:

```
loglik <- function(theta,x) return(sum(log(dcauchy(x,location=theta))))
```

We compute the log-likelihood for different values of  $\theta$ :

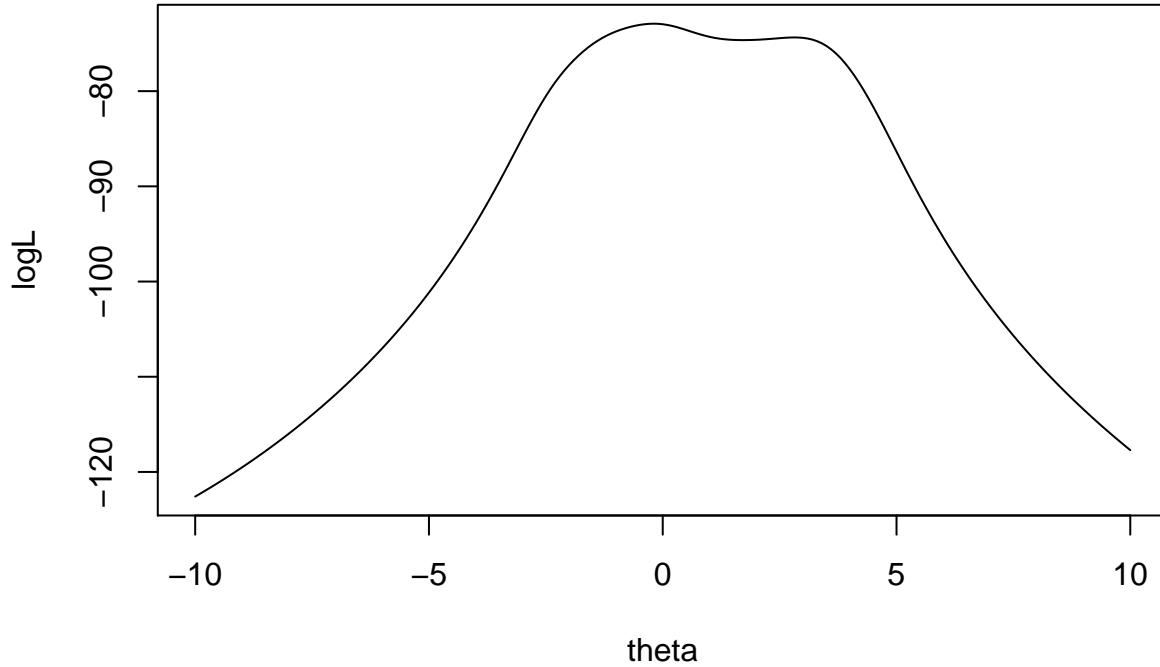
```
theta<- seq(-10,10,0.1)
N<-length(theta)
logL<-rep(0,N)
for(i in 1:N) logL[i]<- loglik(theta[i],x)
```

We can get the same result much faster without a loop, thanks to function `sapply`:

```
logL<-sapply(theta,loglik,x)
```

Finally, we plot the result:

```
plot(theta,logL,type="l")
```



We observe that the likelihood has 2 modes.

### Question c

We first need to compute the score function (first derivative of the log-likelihood). We have

$$L(\theta) = \frac{1}{\pi^n} \prod_{i=1}^n \frac{1}{(x_i - \theta)^2 + 1}$$

$$\ell(\theta) = - \sum_{i=1}^n \log[(x_i - \theta)^2 + 1] - n \log \pi$$

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

We can then write the R function:

```
dloglik <- function(theta,x) return(2*sum((x-theta)/((x-theta)^2+1)))
```

This is a function that encodes the bisection method:

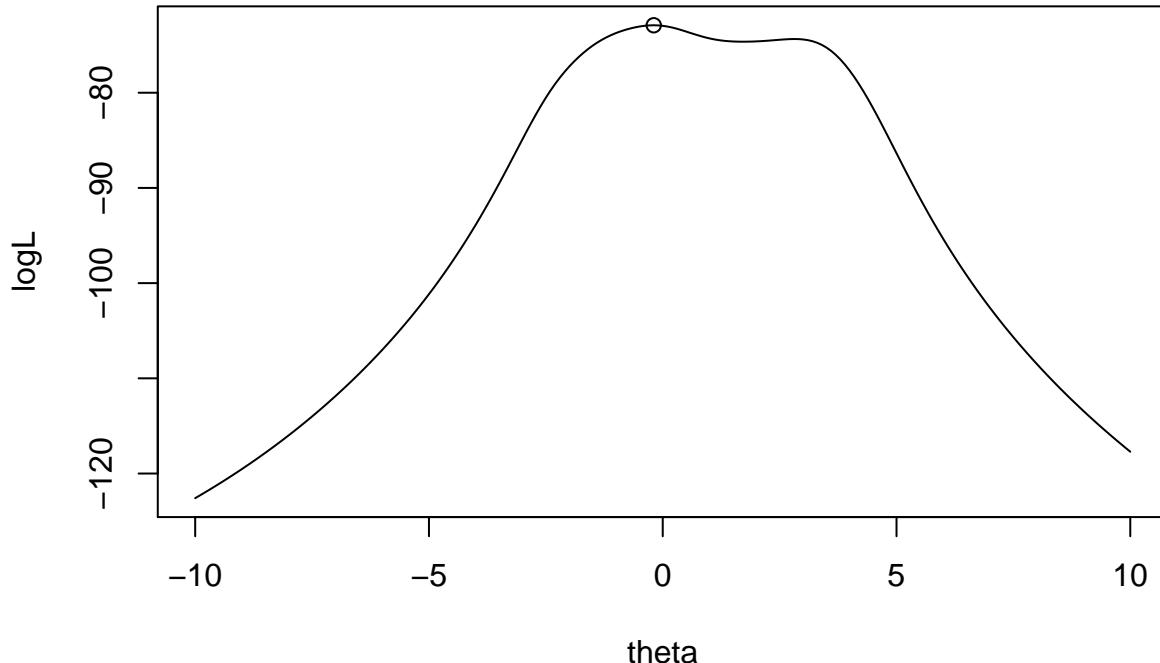
```
bisection <-function(fun,dfun,a,b,epsi,...){
  theta<-(a+b)/2
  delta<-1
  while(delta>epsi){
    theta0<-theta
    if(dfun(a,x)*dfun(theta0,x)<=0) b<-theta0 else a<-theta0
    theta<-(a+b)/2
    delta<-abs(theta-theta0)/abs(theta0)
    print(c(a,b,delta))
  }
  return(list(objective=fun(theta,x),optimum=theta))
}
```

We run it on the data and plot the result:

```
opt<-bisection(loglik,dloglik,-1,3,1e-6,x)

## [1] -1  1  1
## [1] -1    0 Inf
## [1] -0.5  0.0  0.5
## [1] -0.25  0.00  0.50
## [1] -0.250 -0.125  0.500
## [1] -0.2500000 -0.1875000  0.1666667
## [1] -0.21875000 -0.18750000  0.07142857
## [1] -0.20312500 -0.18750000  0.03846154
## [1] -0.1953125 -0.1875000  0.0200000
## [1] -0.19531250 -0.19140625  0.01020408
## [1] -0.193359375 -0.191406250  0.005050505
## [1] -0.192382812 -0.191406250  0.002538071
## [1] -0.192382812 -0.191894531  0.001272265
## [1] -0.192382812 -0.192138672  0.000635324
## [1] -0.1923828125 -0.1922607422  0.0003174603
## [1] -0.1923217773 -0.1922607422  0.0001586798
## [1] -1.922913e-01 -1.922607e-01  7.935248e-05
## [1] -1.922913e-01 -1.922760e-01  3.967939e-05
## [1] -1.922913e-01 -1.922836e-01  1.983891e-05
## [1] -1.922874e-01 -1.922836e-01  9.919257e-06
## [1] -1.922874e-01 -1.922855e-01  4.959678e-06
## [1] -1.922874e-01 -1.922865e-01  2.479827e-06
## [1] -1.922870e-01 -1.922865e-01  1.239910e-06
## [1] -1.922867e-01 -1.922865e-01  6.199559e-07

plot(theta,logL,type="l")
points(opt$optimum,opt$objective)
```



## Question d

Let us program Newton's method. For that, we need the second derivative of the log-likelihood:

$$\ell''(\theta) = 2 \sum_{i=1}^n \frac{(x_i - \theta)^2 - 1}{[(x_i - \theta)^2 + 1]^2}$$

We write the corresponding R function:

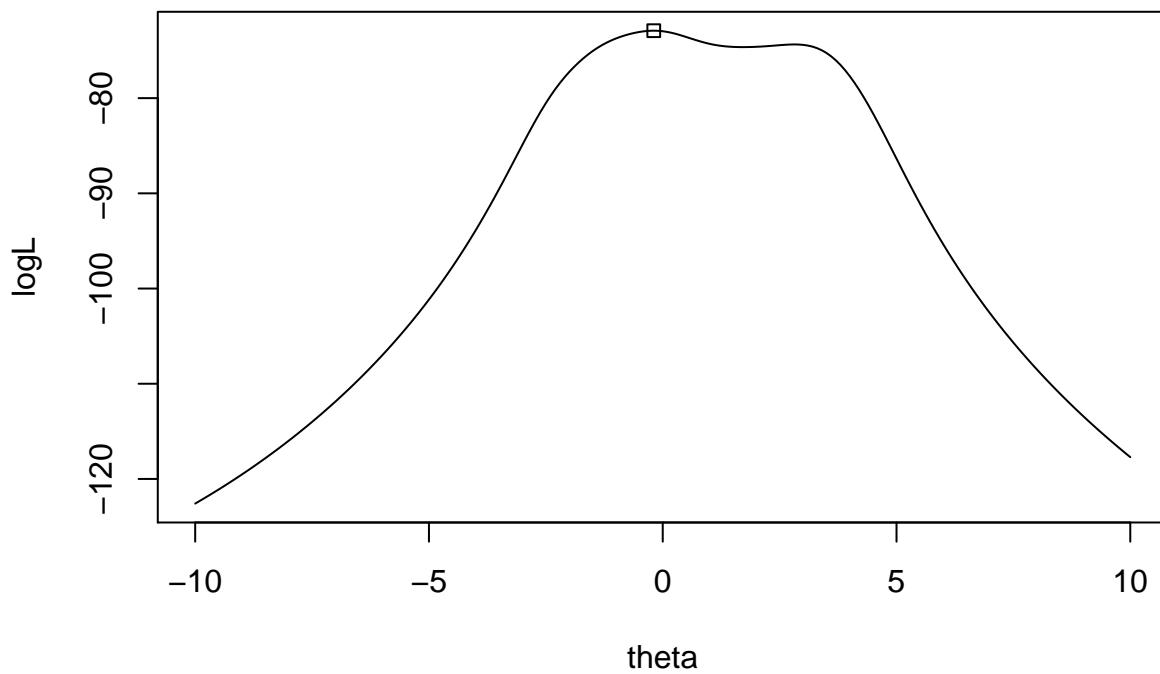
```
d2loglik <- function(theta,x) return(2*sum(((x-theta)^2-1)/((x-theta)^2+1)^2))
```

This is an implementation of Newton's method:

```
newton <-function(fun,dfun,d2fun,theta0,epsi,tmax,...){  
  delta=1  
  t<-0  
  while((delta>epsi)&(t<=tmax)){  
    t<-t+1  
    theta<-theta0-dfun(theta0,x)/d2fun(theta0,x)  
    delta<-abs(theta-theta0)/abs(theta0)  
    obj<-fun(theta,x)  
    print(c(t,theta,obj,delta))  
    theta0<-theta  
  }  
  return(list(objective=obj,optimum=theta,  
             derivative=dfun(theta,x),  
             derivative2=d2fun(theta,x)))  
}
```

We run it on our data and plot the results:

```
theta0<- 0  
opt=newton(loglik,dloglik,d2loglik,theta0,1e-6,1000,x)  
  
## [1] 1.0000000 -0.1963366 -72.9158449 Inf  
## [1] 2.000000000 -0.19228252 -72.91581962 0.02064858  
## [1] 3.000000e+00 -1.922866e-01 -7.291582e+01 2.126316e-05  
## [1] 4.000000e+00 -1.922866e-01 -7.291582e+01 2.109094e-11  
plot(theta,logL,type="l")  
points(opt$optimum,opt$objective,pch=22)
```



### Question e

Finally, we can get the same result with the R built-in function `optimize`:

```
opt <- optimize(f=loglik, lower=-2, upper=2, maximum=TRUE, x=x)
plot(theta, logL, type="l")
points(opt$maximum, opt$objective, pch=23)
```

