2018 Fall

Computational Statistics
HW#1

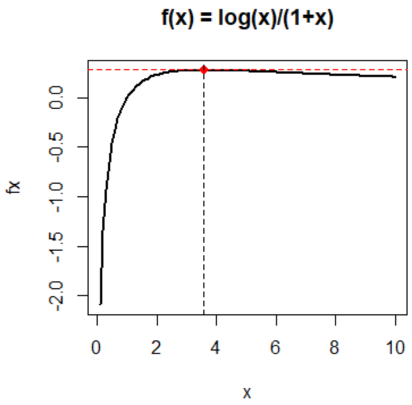
182STG18 이하경

[Problem 1] Implement of 3 Optimization Methods

Goal : Bisection, Newton, Secant 방법을 이용해 최적 값을 찾는 알고리즘을 각각 생성하고 손으로 직접 계산이 어려운 함수들에 직접 적용하여 해를 찾고 방법 별 비교를 해보려고 한다.

function 1. $\mathbf{f(x) = log(x)/(1+x)}$

method	start	x*	f(x*)	niter	comp.time
Bisection	(1, 5)	3.59112	0.27846	28	0.00413
Newton	1	3.59112	0.27846	9	0.00118
Secant	(1, 5)	3.59112	0.27846	14	0.00212
Optimize	(1, 5)	3.59112	0.27846	-	0.00002

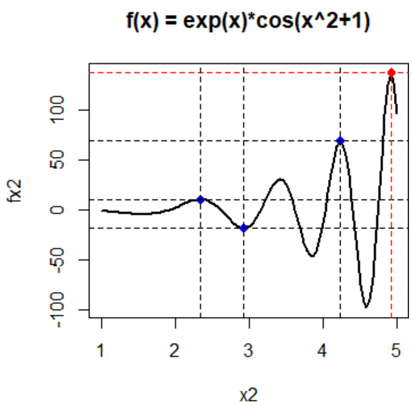


함수의 그래프를 (0,10) 범위에서 그려 보았을 때 2와 4 사이에서 하나의 최댓값을 가지는 것을 확인하고 각 방법의 적절한 초기값으로 1과 5를 설정하였다. 반복 수는 Newton 방법에서 제일 작았으며 소요된 시간 또한 제일 짧았다. 그러나 세 방법 모두 빠른 시간 내에 해를 찾아냈고 R의 optimize 함수를 이용해 찾은 값과 비교했을 때 정확도에도 차이가 없다.

function 2. $\mathbf{f(x)} = e^{x} \cdot \mathbf{cos(x^2+1)}$

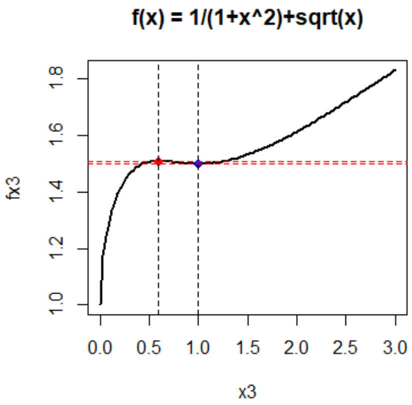
method	start	x*	f(x*)	niter	comp.time
Bisection	(3, 5)	4.2387	68.8436	27	0.00425
Newton	3	2.9315	-18.4888	4	0.00081
Newton	5	**4.9228**	**136.6836**	5	0.00095
Secant	(3, 5)	2.3438	10.1914	9	0.00133
Optimize	(3, 5)	4.2387	68.8436	-	0.00001

두번째 함수의 경우 그래프를 보면 x가 점점 커질수록 local maximum과 minimum 값을 반복해서 가지며 점점 발산하는 형태를 확인할 수 있다. 초기값의 범위를 3, 5로 임의 지정하고 최적 값을 찾은 결과가 방법 별로 서로 다름을 확인하였다. (3, 5)의 범위에서 제일 큰 값을 가지는 x의 값은 4.9228로 Newton 방법의 초기값을 5로 지정했을 때 해당 값을 가장 정확하고 빠르게 찾아냈다. 하지만 초기값을 3으로 하였을 경우에는 전혀 다른 지점으로 수렴하였다. 3가지 방법 모두 값이 수렴하지 않는 경우는 발생하지 않았고 모두 안정적인 계산을 하였다.



function 3. $\mathbf{f(x) = 1/(1 + x^2) + \sqrt{x}}$

method	start	x*	f(x*)	niter	comp.time
all 3 methods	(0, 1)	-	-	-	(try-error) 0
Bisection	(0.1, 1)	0.59326	1.5099	26	0.00638
Newton	0.1	0.59326	1.5099	8	0.00286
Secant	(0.1, 1)	1	1.5	1	0.00080
Secant	(0.1, 0.7)	0.59326	1.5099	9	0.00283
Optimize	(0, 1)	0.59236	1.5099	-	0.00004



이 함수의 경우 $x > 0$인 전 범위에서 local maximum 값과 local minimum 값을 하나씩 가지며 $x \to 0$일 때 $-\infty$로, $x \to \infty$일 때는 ∞로 발산한다. 함수의 식에 $\sqrt{x}$을 포함하고 있어 초기 값에 0을 입력하였을 경우 1차 미분 값이 계산되지 않아 3가지 방법에서 모두 값이 수렴하지 못하고 Error가 발생하였다. 따라서 다음 시행에서 초기값을 (0.1, 1) 또는 0.1으로 조정하여 입력하였을 때 Bisection과 Newton 방법에서는 local maximum 0.593으로, Secant 방법에서는 local minimum인 1으로 수렴하였다. 따라서 Secant 방법에서 초기 값을 (0.1, 0.7)로 좁혀 다시 계산한 결과 이전의 x* 값과 같은 결과를 얻을 수 있었다.

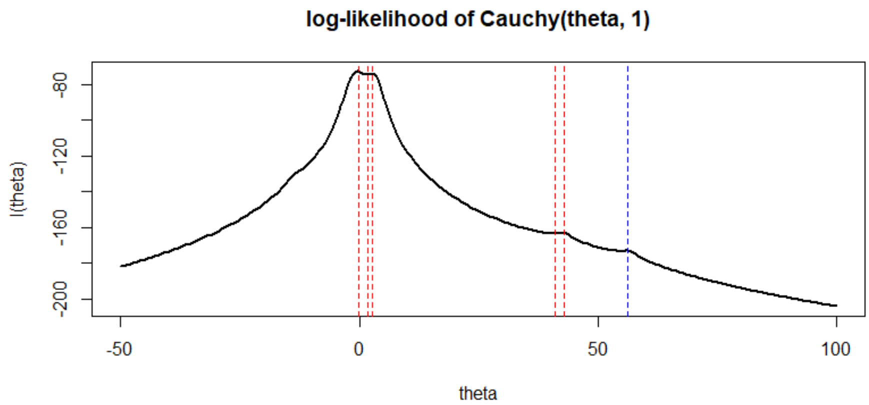
Discussion

직접 생성한 세 가지 다양한 함수의 maximum 또는 minimum 값을 찾은 결과를 통해, 먼저 Newton 방법에서의 계산 속도가 상당히 빠름을 알 수 있었다. Bisection의 경우 반복수가 다른 두 방법에 비해 눈에 띄게 크고 시스템 소요 시간 또한 길었지만 안정적으로 원하는 값으로 잘 수렴하였다. 2차 미분을 계산해 분모에 사용하는 Newton 방법의 경우 수렴 여부가 함수의 형태에 따라 영향을 크게 받으며 비교적 불안정한 것을 알 수 있었다. Newton 방법의 근사적 방법으로서 2차 미분을 계산하지 않아도 되는 Secant 방법은 결과 값이 Newton 방법과 거의 동일할 것으로 예상하였으나 초기 값이 거의 다르지 않음에도 불구하고 서로 다른 값으로 수렴하는 경우가 있었다. 따라서 최적 값을 찾아내기 위해서는 초기 값의 적절한 설정이 중요하며 임의로 선택하기보다는 더 객관적인 기준을 바탕으로 지정할 필요가 있음을 알게 되었다.

[Problem 2.1]

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 of the Log-Likelihood $l(\theta)$ & Finding MLE $\hat{\theta}$ using Newton-Raphson method



start	θ*	$l(\theta^*)$	niter	comp.time
-11	-	-	-	(stopped) 0.28
-1	-0.19229	-72.91582	5	0.00169
0	-0.19229	-72.91582	4	0.00170
1.5	1.71359	-74.64202	5	0.00161
4	2.81747	-74.36046	6	0.00198
4.7	-0.19229	-72.91582	6	0.00197
7	41.04085	-163.60772	10	0.00293
8	-	-	-	(stopped) 0.11
38	42.79538	-163.31289	8	0.00380
$\bar{x}$ = 5.106	56.25336	-173.1686	12	0.00331

log-likelihood의 그래프로 (-50, 100)의 범위에서 약 4개의 local maximum을 가지는 것을 확인하였다. 그 중 θ의 값이 0의 근처일 때 global maximum을 가진다. 9개의 초기값 중 -11과 8에서는 함수가 수렴하지 않아 최적 값을 찾지 못했으며, -1, 0, 4.7에서 $\hat{\theta} = -0.19229$로 global maximum 값 -72.91582를 가지는 MLE를 찾아냈다. 나머지 초기값에서 구해진 값은 local maximum이나 local minimum에 해당한다. 주어진 sample의 평균 5.106을 초기값으로 하였을 때 반복 수는 다른 초기값에 비해 크며 수렴한 값은 56.25336으로 local maximum에 해당하므로 global maximum을 찾아내는 데에는 좋은 초기값이 아니라고 할 수 있다.

b. MLE $\hat{\theta}$ using Bisection method

start	θ*	$l(\theta^*)$	niter	comp.time
(-1, 1)	-0.19229	-72.91582	27	0.00514

초기값 (-1,1)으로 Bisection 방법을 적용하여 찾아낸 θ*는 -0.19229로 θ의 전 범위에서 log-likelihood 값이 제일 크다.

d. MLE $\hat{\theta}$ using Secant method

start	θ^*	$l(\theta^*)$	niter	comp.time
(-2, 1)	1.71359	-74.64202	9	0.00199
(-3, 3)	2.81747	-74.36046	7	0.00150
(-1, 1)	-0.19229	-72.91582	9	0.00183

(-2, 1)과 (-3, 3)에서 수렴한 θ^*의 값은 서로 다르며 두 값 모두 global maximum에 해당하지는 않는다. 따라서 Bisection 방법과 마찬가지로 (-1, 1)을 초기값으로 추가적인 시행을 해보았을 때 MLE 값을 잘 찾아냈다.

e. Comparing the speed and stability of 3 methods & Applying to a sample of $N(\theta, 1)$

1) Speed and Stability of 3 methods

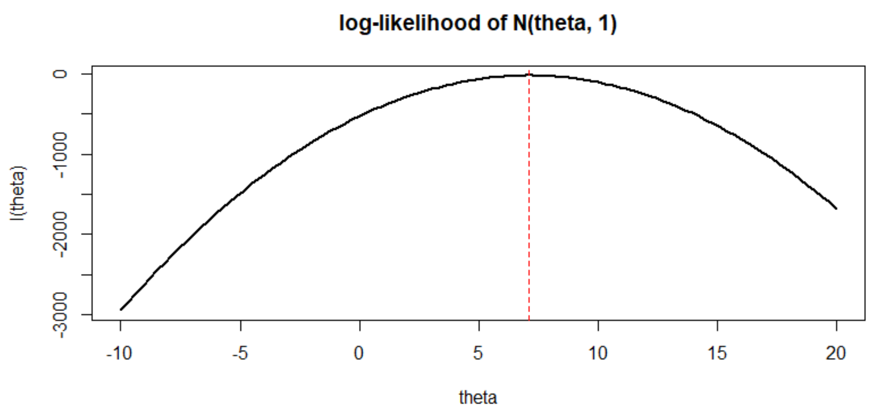
위의 세 가지 방법에서의 각각 반복횟수와 소요된 계산 시간은 비슷한 초기값을 기준으로 비교하였을 때 Newton, Secant, Bisection 방법 순으로 짧았다. Newton 방법의 경우 특히 7, 8과 같이 인접한 두 값을 초기값으로 각각 추정하였을 때 값의 수렴 여부가 상이한 점이 눈에 띄었다. 따라서 계산 속도 면에서는 Newton 방법이 가장 우수하지만 계산 시 2차 미분 값까지 포함하기 때문에 안정성이 다소 떨어진다는 것을 확인할 수 있다. Bisection 방법의 경우 속도가 느리지만 초기값의 설정 범위가 최적 값을 포함하고 있을 경우 잘 수렴하므로 안정성 면에서는 우수하다. 세 가지 방법에서 모두 초기값에 따라 수렴하는 θ^*이 달라져, 함수의 maximum 값이 여러 개 존재하는 경우 global maximum 값을 잘 찾기 위해서는 적절한 초기값의 설정이 중요함을 알 수 있다.

2) Result of applying to a random sample of $N(\theta, 1)$

(6.50, 7.13, 6.92, 7.89, 7.12, 7.32, 6.42, 7.71, 6.17, 6.64, 7.09, 7.10, 6.80, 7.74, 7.12, 6.97, 6.61, 7.51, 6.09, 9.31)

i. 임의의 $\theta = 7$에 대해 $N(\theta, 1)$ 분포에서 n=20의 random sample을 추출하였다.

ii. log-likelihood의 그래프 확인 결과 정규 분포는 코시 분포와 달리 하나의 global maximum을 가진다. 초기값을 (5, 10)으로 설정하고 각 방법으로 MLE를 추정한 후 R의 optimize 함수를 이용하여 구한 최적 값과 비교하였다.



method	start	θ^*	$l(\theta^*)$	niter	comp.time
Bisection	(5, 10)	7.10787	-23.28395	28	0.01061

Newton	5	7.10787	-23.28395	2	0.00119
Newton	$\bar{x}$ = 7.108	7.10787	-23.28395	1	0.00119
Secant	(5, 10)	7.10787	-23.28395	2	0.00135
Optimize	(5, 10)	7.10787	-23.28295	-	0.00008

각 방법에서 설정한 초기 값에서 모두 MLE $\hat{\theta} = 7.10787$을 안정적으로 잘 찾아냈으며 이 값은 표본의 평균 $\bar{x}$와 같다. 속도 면에서는 Cauchy 분포에서와 동일하게 Newton 방법이 반복횟수와 시스템 소요 시간이 가장 빨랐다.

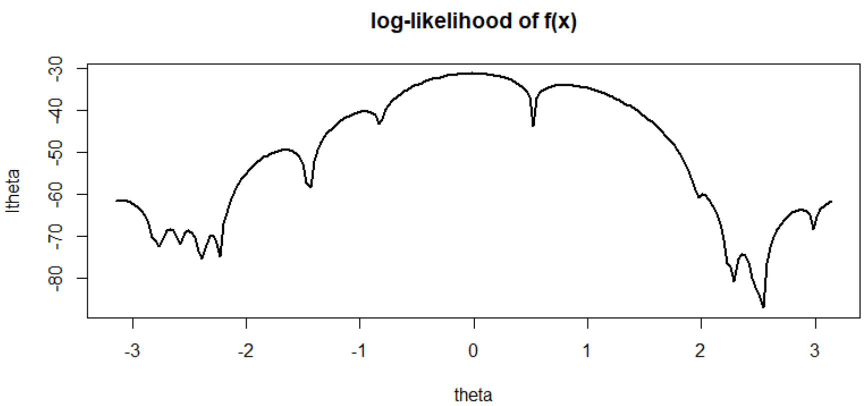
[Problem 2.2]

$$\mathbf{X \sim f(x) = [1 - \cos(x - \theta)\,]/2\pi,\ \ 0 \le x \le 2\pi,\ \ -\pi \le \theta \le \pi}$$

i.i.d. Data from the 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)

a. Graph of the Log-Likelihood $\boldsymbol{l(\theta)}$ $(-\pi \le \theta \le \pi)$

log-likelihood의 그래프가 주어진 범위에서 여러 개의 optimum 값을 가지는 것을 확인할 수 있다.



b. Method-of-Moments Estimator of θ

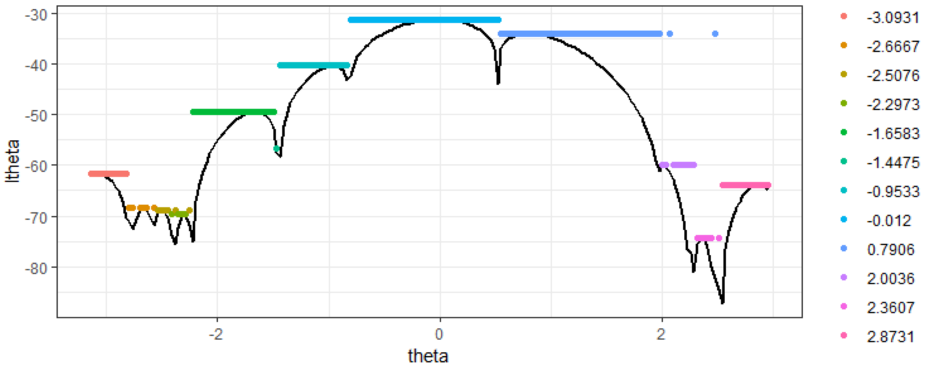
$$E(X) = \int_0^{2\pi} xf(x)\,dx = \int_0^{2\pi} \frac{x(1-\cos\theta)}{2\pi}dx = \pi + \sin\theta\,, \quad \rightarrow\ \hat{\theta}^{MME} = \sin^{-1}(\bar{x} - \pi) = 0.5844 \quad \cdots\ \bar{x} = 3.2$$

c. MLE $\hat{\theta}$ using Newton-Raphson Method

start	θ^*	$l(\theta^*)$	niter	comp.time
MME = 0.0584	-0.0120	-31.3429	4	0.00225
-2.7	-2.6667	-68.3992	10	0.00515
2.7	2.8731	-63.8057	7	0.00411

MME 0.0584를 초기값으로 하였을 때 global maximum 값으로 잘 수렴하였으며 -2.7과 2.7에서는 각각 해당 초기값과 인접한 곳의 local 값으로 다르게 수렴하였다.

d. MLE using Newton method from 200 equally spaced starting values of $(-\pi, \pi)$
& Partitioning the interval into several groups



interval	θ^*	$l(\theta^*)$	
(-3.1416, -2.8259)	-3.0931	-61.6817	
(-2.7943, -2.2575)	-2.6667	-68.3992	
	-2.5076	-68.8378	
	-2.2973	-69.7194	
(-2.2260, -1.4998)	-1.4475	-56.7603	
(-1.4366, -0.8367)	-0.9533	-40.3785	
(-0.7736, 0.5210)	**-0.0120**	**-31.3429**	**MLE**
(0.5525, 1.9734)	0.7906	-34.1414	
(2.0049, 2.2891)	2.0036	-59.9967	
(2.3207, 2.5101)	2.3607	-74.4577	
(2.5417, 2.9521)	2.8731	-63.8057	

200개의 초기 값에서 총 11개의 local 또는 global maximum 값이 추정되었다. 추정되는 값에 따라 9개의 구간으로 나눌 수 있으며, 그래프에서 변동이 심한 (-2.5417, -2.2575) 구간에서는 θ^*의 값이 -2.6667, -2.5076 또는 -2.2973 세가지 값 중 하나로 다양하게 수렴하였다. 그래프에서 확인할 수 있듯이 전체에서 가장 큰 log-likelihood 값을 가지는 MLE $\theta^* = -0.0120$이며 총 200개의 초기 값 중 43개에서 해당 값으로 잘 수렴하였다.

e. 2 Starting values nearly equal which the Newton-Raphson method converges to 2 different solutions

start	θ^*	$l(\theta^*)$	niter	comp.time
2.4785390	2.36072	-74.45772	12	0.00800
2.4785391	2.87310	-63.80565	13	0.01057
2.4785392	0.79060	-34.14135	13	0.01014
2.4785393	2.00364	-59.99666	12	0.00962

log-likelihood의 그래프에서 변동이 심한 구간을 확인할 때 바로 인접한 초기값에서 서로 다른 해를 찾는 경우가 일부 발생하였다. 그 중 특히 초기값이 2.4785390에서 2.4785393까지 10^-7 씩 차이나는 구간에서 Newton 방법으로 찾아진 최적 값은 각각 다르고, 이 때의 log-likelihood 값은 -74.457부터 -34.1414로 매우 상이하다. 4개의 초기 값 중 2.4785392에서만 global maximum likelihood 값으로 수렴하였다.

[Appendix] R Code

```
library(numDeriv)
library(dplyr)
library(ggplot2)

  mybisec <- function(f, a, b, E=10^-8) {

    maxiter = 1000
    error = 1
    niter = 0
    xt = (a+b)/2

    while (niter <= maxiter & error >= E) {

      if (grad(f,a)*grad(f,xt) < 0) { b <- xt }
      else if (grad(f,a)*grad(f,xt) > 0) { a <- xt }

      xt_1 <- xt
      xt <- (a+b)/2

      error <- abs(xt-xt_1)
      niter <- niter + 1
    }
    return(data.frame(x = xt, fx = f(xt), niter))
  }

  mynewton <- function(f, x0, E=10^-8) {

    maxiter = 1000
    error = 1
    niter = 0
    xt = x0

    while (niter <= maxiter & error >= E)  {

      xt_1 <- xt
      xt <- xt - grad(f, xt)/as.double(hessian(f, xt))
      error <- abs(xt-xt_1)
      niter <- niter + 1
    }
    return(data.frame(x = xt, fx = f(xt), niter))
  }

  mysecant <- function(f, x0, x1, E=10^-8) {

    maxiter = 1000
    error = 1
    niter = 0
    xt = x1
    xt_1 = x0

    while (niter <= maxiter & error >= E) {

      xt_new <- xt - grad(f, xt)*(xt-xt_1)/(grad(f, xt)-grad(f, xt_1))
      xt_1 <- xt
      xt <- xt_new

      error <- abs(xt-xt_1)
      niter <- niter + 1
    }
    return(data.frame(x = xt, fx = f(xt), niter))
  }

  bisectime <- function(f, a, b) { system.time( for (i in 1:1000) mybisec(f, a, b) ) / 1000 }
  newtontime <- function(f, x0) { system.time( for (i in 1:1000) mynewton(f, x0) ) / 1000 }
  secanttime <- function(f, x0, x1) {
    system.time( for (i in 1:1000) mysecant(f, x0, x1) ) / 1000 }
```

Problem 1

```
myf1 <- function(x) {
  f = log(x)/(1+x)
  return(f)
}

mybisec(myf1, 1, 5) ; bisectime(myf1, 1, 5)
mynewton(myf1, 1) ; newtontime(myf1, 1)
mysecant(myf1, 1, 5) ; secantime(myf1, 1, 5)
optimize(myf1, c(1, 5), maximum = T)

x_star = mynewton(myf1, 1)$x

x = seq(0.1, 10, by = 0.1) ; fx = myf1(x)
plot(x, fx, main = paste("f(x) = log(x)/(1+x)"), type = "l", lwd = 2)
points(x_star, myf1(x_star), pch = 16, col = "red")
abline(v = x_star, h = myf1(x_star), lty = c(2,2), col = c("red","black"))

  myf2 <- function(x) {
  f = exp(x)*cos(x^2+1)
  return(f)
}

mybisec(myf2, 3, 5) ; bisectime(myf2, 3, 5)
mynewton(myf2, 3) ; newtontime(myf2, 3)
mysecant(myf2, 3, 5) ; secanttime(myf2, 3, 5)
optimize(myf2, c(3, 5), maximum = T)

x2_star = mynewton(myf2, 5)$x
x2_newton1 = mynewton(myf2, 3)$x
x2_bisec = mybisec(myf2, 3, 5) ; x2_secant = mysecant(myf2, 3, 5)

x2 = seq(1, 5, length = 100) ; fx2 = myf2(x2)
plot(x2, fx2, main = paste("f(x) = exp(x)*cos(x^2+1)"), type = "l", lwd = 2)
points(x2_star, myf2(x2_star), pch = 16, col = "red")
points(x2_newton1, myf2(x2_newton1), pch = 16, col = "blue")
points(x2_bisec, myf2(x2_bisec), pch = 16, col = "blue")
points(x2_secant, myf2(x2_secant), pch = 16, col = "blue")
abline(v = c(x2_star, x2_newton1, x2_bisec, x2_secant),
       h = c(myf2(x2_star), myf2(x2_newton1), myf2(x2_bisec), myf2(x2_secant)),
       lty = c(2,2,2,2), col = c("red", "black", "black", "black"))

myf3 <- function(x) {
  f = 1/(1+x^2)+sqrt(x)
  return(f)
}

x3 = seq(0, 3, length = 100) ; fx3 = myf3(x3)
plot(x3, fx3, main = paste("f(x) = 1/(1+x^2)+sqrt(x)"), type = "l", lwd = 2)

mybisec(myf3, 0.1, 1) ; bisectime(myf3, 0.1, 1)
mynewton(myf3, 0.1) ; newtontime(myf3, 0.1)
mysecant(myf3, 0.1, 1) ; secanttime(myf3, 0.1, 1)
mysecant(myf3, 0.1, 0.7) ; secanttime(myf3, 0.1, 0.7)
optimize(myf3, c(0, 1), maximum = T)

x3_star = mybisec(myf3, 0.1, 1)$x
x3_local = mysecant(myf3, 0.1, 1)$x
plot(x3, fx3, main = paste("f(x) = 1/(1+x^2)+sqrt(x)"), type = "l", lwd = 2)
points(x3_star, myf3(x3_star), pch = 16, col = "red")
points(x3_local, myf3(x3_local), pch = 16, col = "blue")
abline(v = c(x3_star, x3_local), h = c(myf3(x3_star), myf3(x3_local)),
      lty = c(2,2,2,2), col = c("red", "red", "black", "black"))
```

Problem 2.1

```
sample = 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)


  llcauchy <- function(theta) {
```

```
    loglik = c()
    for (i in 1:length(theta)) {
      density = dcauchy(sample, location = theta[i], scale = 1)
      loglik[i] = sum(log(density))
    }
    return(loglik)
  }

  tb <- data.frame(theta = seq(-50, 100, length = 300)) %>% mutate(ltheta = llcauchy(theta))
  plot(tb$theta, tb$ltheta, type = "l", lwd = 2,
       xlab = "theta", ylab = "l(theta)", main = "log-likelihood of Cauchy(theta, 1)")

  start = c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)
  newton21 = data.frame(theta = numeric(length(start)), loglik = numeric(length(start)),
                        niter = numeric(length(start)))
  for (j in 1:3) {
    for (i in 1:length(start)) {
      newton21[i,j] <- ifelse(is(try(mynewton(llcauchy, x0 = start[i]), silent = T),
"try-error"), t(rep(NA,3)), mynewton(llcauchy, x0 = start[i])[j])
    }
  }

  for (i in 1:length(start)) { try(newtontime(llcauchy, x0 = start[i]), silent = T) }

  mean(sample)
  mynewton(llcauchy, x0 = mean(sample))
  newtontime(llcauchy, x0 = mean(sample))

  plot(tb$theta, tb$ltheta, type = "l", lwd = 2,
       xlab = "theta", ylab = "l(theta)", main = "log-likelihood of Cauchy(theta, 1)")
  abline(v = newton21$theta, col = "red", lty = 2)
  abline(v = mynewton(llcauchy, x0 = mean(sample))$x, col = "blue", lty = 2)


# b
  mybisec(llcauchy, -1, 1) ; bisectime(llcauchy, -1, 1)

# d
  mysecant(llcauchy, -2, 1) ; secanttime(llcauchy, -2, 1)
  mysecant(llcauchy, -3, 3) ; secanttime(llcauchy, -3, 3)
  mysecant(llcauchy, -1, 1) ; secanttime(llcauchy, -1, 1)

# e
  set.seed(100)
  sample2 = rnorm(20, 7, 1)

  llnorm <- function(theta) {

    loglik = c()
    for (i in 1:length(theta)) {
      density = dnorm(sample2, mean = theta[i], sd = 1)
      loglik[i] = sum(log(density))
    }
    return(loglik)
  }

  tb2 <- data.frame(theta = seq(-10, 20, length = 100)) %>% mutate(ltheta = llnorm(theta))
  plot(tb2$theta, tb2$ltheta, type = "l", lwd = 2,
      xlab = "theta", ylab = "l(theta)", main = "log-likelihood of N(theta, 1)")

  optimize(llnorm, c(5,10), maximum = T)

  mybisec(llnorm, 5, 10) ; bisectime(llnorm, 5, 10)
  mynewton(llnorm, 5) ; newtontime(llnorm, 5)
  mysecant(llnorm, 5, 10) ; secanttime(llnorm, 5, 10)

  mean(sample2)
  mynewton(llnorm, mean(sample2)) ; newtontime(llnorm, mean(sample2))

  plot(tb2$theta, tb2$ltheta, type = "l", lwd = 2,
      xlab = "theta", ylab = "l(theta)", main = "log-likelihood of N(theta, 1)")
  abline(v = mean(sample2), col = "red", lty = 2)
```

Problem 2.2

```
data = 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)

  ex2.2 <- function(x, theta) {
    fx = ifelse((x>=0 & x<=2*pi), (1-cos(x-theta))/(2*pi), 0)
  }

  llex2.2 <- function(theta) {
    loglik = c()
    for (i in 1:length(theta)) {
      density = ex2.2(data, theta[i])
      loglik[i] = sum(log(density))
    }
    return(loglik)
  }

# a
  theta = seq(-pi, pi, length = 200)
  ltheta = llex2.2(theta)

  plot(theta, ltheta, type = "l", lwd = 2, main = "log-likelihood of f(x)")

# b
  mme = asin(mean(data)-pi)
# c
  mynewton(llex2.2, mme) ; newtontime(llex2.2, mme)
  mynewton(llex2.2, -2.7) ; newtontime(llex2.2, -2.7)
  mynewton(llex2.2, 2.7) ; newtontime(llex2.2, 2.7)

# d
  newton22_x = c() ; newton22_fx = c()
  for (i in 1:length(theta)) {
    newton22_x[i] <- ifelse(is(try(mynewton(llex2.2, x0 = theta[i]), silent = T),"try-error"),
                            NA, mynewton(llex2.2, x0 = theta[i])$x)
    newton22_fx[i] <- ifelse(is(try(mynewton(llex2.2, x0 = theta[i]), silent = T),
"try-error"), NA, mynewton(llex2.2, x0 = theta[i])$fx)
  }
  newton22 = data.frame(theta, ltheta, theta_star = newton22_x, ltheta_star = newton22_fx)
  newton22 <- filter(newton22, theta_star >= -pi & theta_star <= pi)

  plot(theta, ltheta, type = "l", lwd = 2,
       xlab = "theta", ylab = "l(theta)", main = "log-likelihood of f(x)")
  points(newton22$theta_star, newton22$ltheta_star, col = "red", pch = 16)

  ggplot(newton22) + geom_line(aes(theta, ltheta), size = 0.8) +
    geom_point(aes(theta, ltheta_star, color = factor(round(theta_star,4))), size = 1.5) +
theme_bw() + labs(color = "theta_star")

# e
  mynewton(llex2.2, x0 = 2.4785390)
  mynewton(llex2.2, x0 = 2.4785391)
  mynewton(llex2.2, x0 = 2.4785392)
  mynewton(llex2.2, x0 = 2.4785393)
  mynewton(llex2.2, x0 = 2.4785394)
```