STA410 HW3

Q1-c

```
leverage <- function(x1,x2,w,r=10,m=100) {
               qrx1 <- qr(x1)
               qrx2 <- qr(x2)
               n <- nrow(x1)
               lev1 <- NULL
               lev2 <- NULL

               for (i in 1:m) {
                 v <- ifelse(runif(n)>0.5,1,-1)
                 v[-w] <- 0
                 v1 <- qr.fitted(qrx1,v)
                 v2 <- qr.fitted(qrx2,v)
                 f1 <- v1
                 f2 <- v2

                 for (j in 2:r) {
                   v1[-w] <- 0
                   v2[-w] <- 0
                   v1 <- qr.fitted(qrx1,v1)
                   v2 <- qr.fitted(qrx2,v2)
                   f1 <- f1 + v1/j
                   f2 <- f2 + v2/j
                   }
                 lev1 <- c(lev1,sum(v*f1))
                 lev2 <- c(lev2,sum(v*f2))
               }

               std1 <- exp(-mean(lev1))*sd(lev1)/sqrt(m)
               std2 <- exp(-mean(lev2))*sd(lev2)/sqrt(m)
               lev1 <- 1 - exp(-mean(lev1))
               lev2 <- 1 - exp(-mean(lev2))
               r <- list(lev=c(lev1,lev2),std.err=c(std1,std2))
               r
}
```

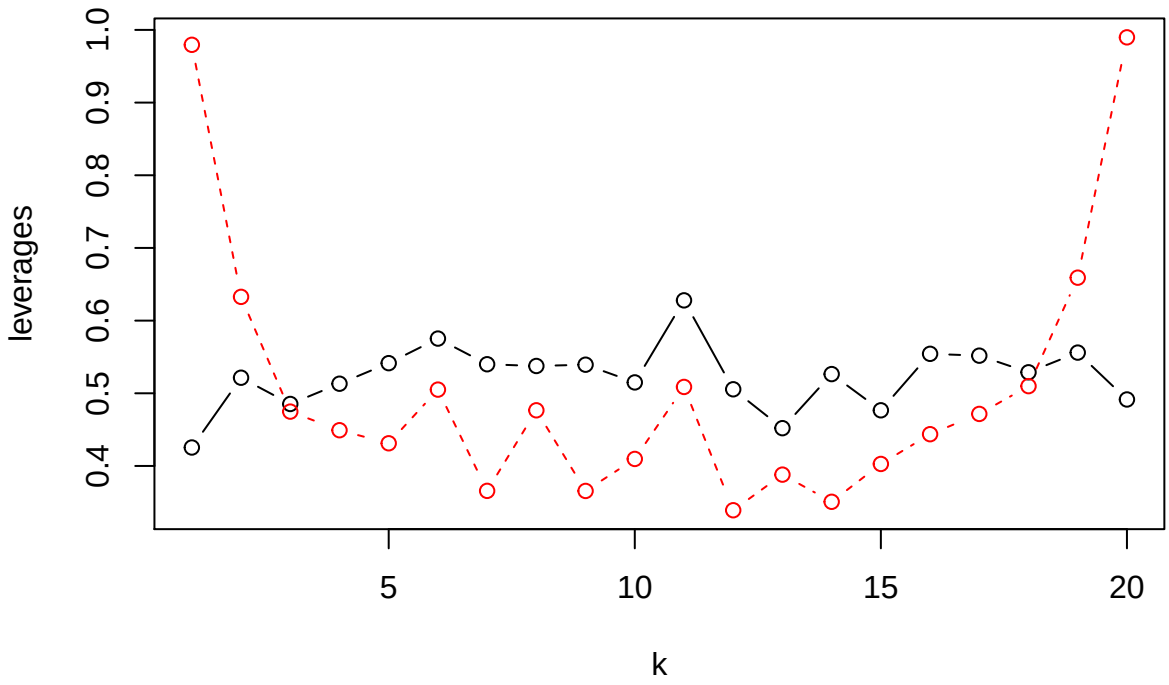
```
library(splines)
x <- c(1:1000)/1000
X1 <- 1
for (k in 1:5) X1 <- cbind(X1,cos(2*k*pi*x),sin(2*k*pi*x))
X2 <- cbind(1,bs(x,df=10))

lever<-NULL
for (k in 1:20) {
```

```
  w<-c(((k-1)*50+1):(k*50))
  result<-leverage(X1,X2,w=w,r=40,m=100)
  lever<-rbind(lever,result$lev)
  }
lever
```

```
##            [,1]      [,2]
##  [1,] 0.4254260 0.9794917
##  [2,] 0.5214235 0.6326728
##  [3,] 0.4852865 0.4747872
##  [4,] 0.5131786 0.4491966
##  [5,] 0.5414538 0.4312251
##  [6,] 0.5752806 0.5050759
##  [7,] 0.5399799 0.3657412
##  [8,] 0.5375625 0.4765854
##  [9,] 0.5393499 0.3656241
## [10,] 0.5149725 0.4096659
## [11,] 0.6278626 0.5087424
## [12,] 0.5055163 0.3390841
## [13,] 0.4520097 0.3880757
## [14,] 0.5264197 0.3507190
## [15,] 0.4765826 0.4027633
## [16,] 0.5543040 0.4437019
## [17,] 0.5517904 0.4716084
## [18,] 0.5288886 0.5098448
## [19,] 0.5559940 0.6590900
## [20,] 0.4914789 0.9897658
```

```
matplot(lever, type = c("b"),pch=1,col = 1:2,xlab="k",ylab="leverages")
```



In the sinusoidal(black) model,the 20 estimated leverages are around 0.52 from k = 1 to 20. In the B-spline (red), the 20 estimated leverages are largest at the first and last observations. In most of the points, model sinusoidal have greater leverages than B-spline.

Q2-c

```
cauchy.mle<-function(x,start1,start2,eps=1.e-8,max.iter=50){
  if (missing(start1)) start1 <- median(x)
  if (missing(start2)) start2 <- 1/2*IQR(x)

  # intial value of theta and sigma
  theta <- start1
  sigma <- start2
  parameter <-c(theta,sigma)
  n <- length(x)

  # original score S(theta) and S(sigma)
  score_theta <- 2*sum((x-theta)/((x-theta)^2+sigma^2))
  score_sigma <- n/sigma-sum(2*sigma/((x-theta)^2+sigma^2))
  score <- c(score_theta,score_sigma)

  iter <- 1
  conv <- T

  while (max(abs(score))>eps && iter<=max.iter){
    # calculate the fisher information
    info_aa <- 2*sum((sigma^2-(x-theta)^2)/((x-theta)^2+sigma^2)^2)
    info_bb <- n/sigma^2+sum((2*((x-theta)^2-sigma^2))/((x-theta)^2+sigma^2)^2)
    info_ab <- 4*(sum((sigma*(x-theta))/((x-theta)^2+sigma^2)^2))
    info <-matrix(c(info_aa,info_ab,info_ab,info_bb),nrow=2,ncol=2)


    parameter <- parameter + solve(info,score)
    theta<-parameter[1]
    sigma <-parameter[2]
    score_theta <- 2*sum((x-theta)/(sigma^2+(x-theta)^2))
    score_sigma <- n/sigma-sum(2*sigma/(sigma^2+(x-theta)^2))
    score <- c(score_theta ,score_sigma)
    iter <- iter + 1
    }
  if (max(abs(score))>eps) {
      print("No Convergence")
      conv <- F
      }

    loglik <- n*log(sigma)-n*log(pi)-sum(log(sigma^2+(x-theta)^2))

    info_aa <- 2*sum((sigma^2-(x-theta)^2)/((x-theta)^2+sigma^2)^2)
    info_bb <- n/sigma^2+sum((2*((x-theta)^2-sigma^2))/((x-theta)^2+sigma^2)^2)
    info_ab <- 4*(sum((sigma*(x-theta))/((x-theta)^2+sigma^2)^2))
    info <-matrix(c(info_aa,info_ab,info_ab,info_bb),nrow=2,ncol=2)


    r <- list(parameter=parameter,loglik=loglik,info=info,varcov = solve(info),
              convergence=conv,score=score)
    r
}
```

```r
set.seed(123)
x <- rcauchy(1000) + 8
r <- cauchy.mle(x)
r
```

```
## $parameter
## [1] 8.011717 1.013501
##
## $loglik
## [1] -2508.996
##
## $info
##            [,1]       [,2]
## [1,] 484.614025  -3.288687
## [2,]  -3.288687 488.920748
##
## $varcov
##              [,1]         [,2]
## [1,] 2.063592e-03 1.388059e-05
## [2,] 1.388059e-05 2.045415e-03
##
## $convergence
## [1] TRUE
##
## $score
## [1] 5.595073e-13 2.046363e-12
```