Absolute weights

Martin Roth

November 2, 2016

```
library(futile.logger)
library(itertools)
```

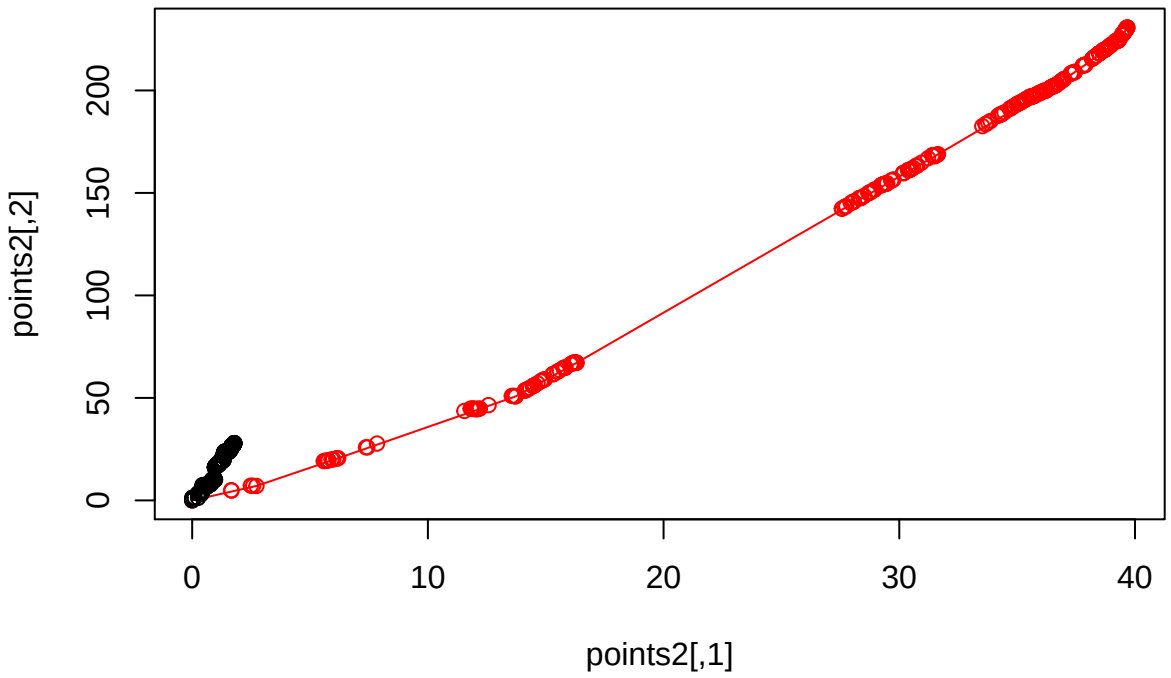
```
## Loading required package: iterators
```

```
library(data.table)
library(gpdIcm)
library(ggplot2)
```

```
shape <- -0.2
n <- 300
scale <- c(seq(5, 8, length.out = 100), rep(8, 100), seq(8, 15, length.out = 100))

set.seed(123)
y <- evd::rgpd(n, 0, scale, shape)
start1 <- start2 <- start <- isoreg(y)$yf
deviance1 <- deviance2 <- gpdIcm::compute_nll_gpd(y, start, shape)

scale1 <- start1
scale2 <- start2

directionSaved1 <- NULL
directionSaved2 <- NULL

hessianSaved1 <- NULL
hessianSaved2 <- NULL
```

```
gradient1 <- gpdIcm:::ComputeGradient(y, start1, shape)
hesDiag1  <- gpdIcm:::ComputeHessianDiagonal(y, start1, shape)

gradient2 <- gpdIcm:::ComputeGradient(y, start2, shape)
hesDiag2  <- gpdIcm:::ComputeHessianDiagonal(y, start2, shape)

hessianSaved1 <- rbind(hessianSaved1, hesDiag1)
hessianSaved2 <- rbind(hessianSaved2, hesDiag2)

nNegWeights1 <- length(hesDiag1[hesDiag1 < 0])
nNegWeights2 <- length(hesDiag2[hesDiag2 < 0])

hesDiag1[hesDiag1 < 0] <- 1e-5
hesDiag2[hesDiag2 < 0] <- pmax(-hesDiag2[hesDiag2 < 0], 1e-5)


points1  <- cbind(c(0, cumsum(hesDiag1)),
                  c(0, cumsum(start1 * hesDiag1 - gradient1)))
points2  <- cbind(c(0, cumsum(hesDiag2)),
```

```
                  c(0, cumsum(start2 * hesDiag2 - gradient2)))

projection1 <- GreatestConvexMinorant(points1[,1], points1[, 2])
projection2 <- GreatestConvexMinorant(points2[,1], points2[, 2])

plot(points2, col = 2)
lines(projection2$x.knots, projection2$y.knots, col = 2)
points(points1, col = 1)
lines(projection1$x.knots, projection1$y.knots, col = 1)
```

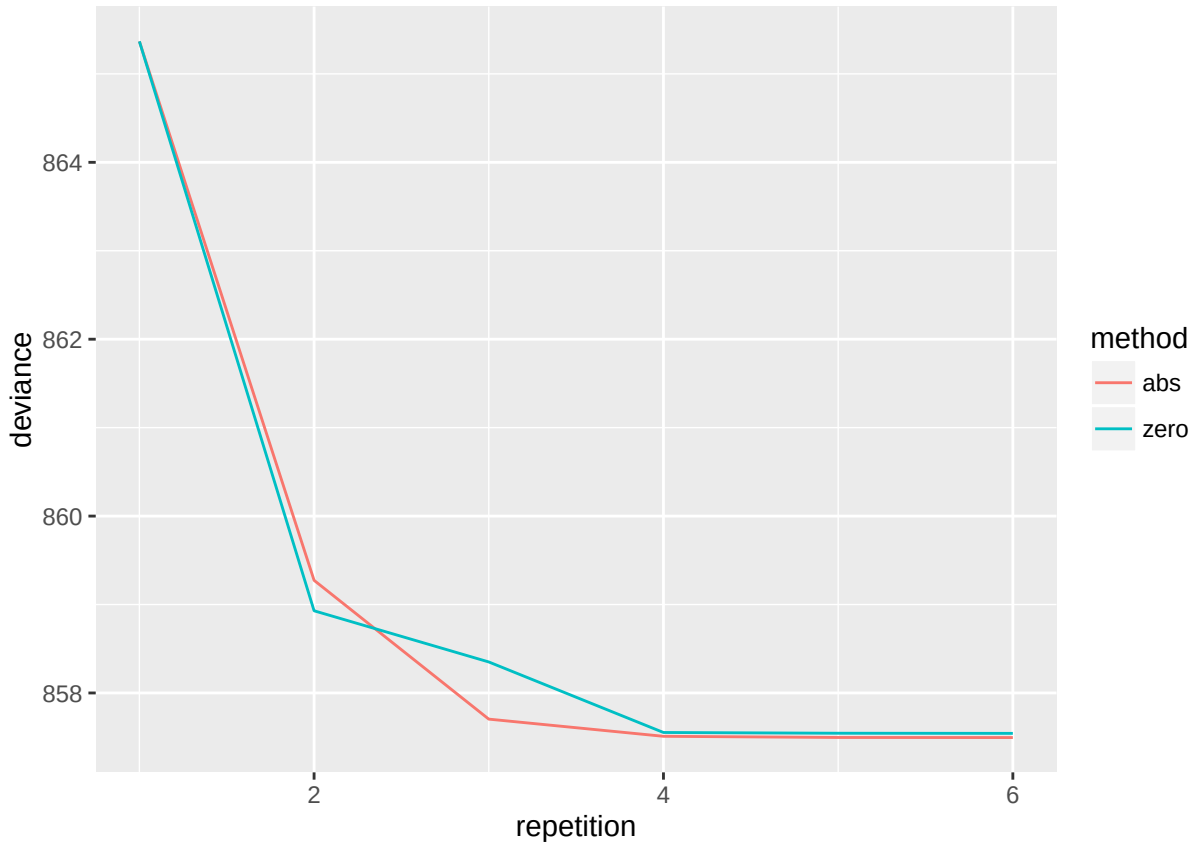



```
direction1 <- projection1$left.derivative - start1
direction2 <- projection2$left.derivative - start2

directionSaved1 <- rbind(directionSaved1, direction1)
directionSaved2 <- rbind(directionSaved2, direction2)

nextIterate1 <- gpdIcm:::LineSearchICM(y, start1, direction1, gradient1, shape)
nextIterate2 <- gpdIcm:::LineSearchICM(y, start2, direction2, gradient2, shape)

# plot(direction1, col = 4)
# points(direction2, col = 2)

# plot(start)
# points(nextIterate1$scale, col = 4)
# points(nextIterate2$scale, col = 2)

deviance1 <- c(deviance1, gpdIcm::compute_nll_gpd(y, nextIterate1$scale, shape))
deviance2 <- c(deviance2, gpdIcm::compute_nll_gpd(y, nextIterate2$scale, shape))

start1 <- nextIterate1$scale
start2 <- nextIterate2$scale
```

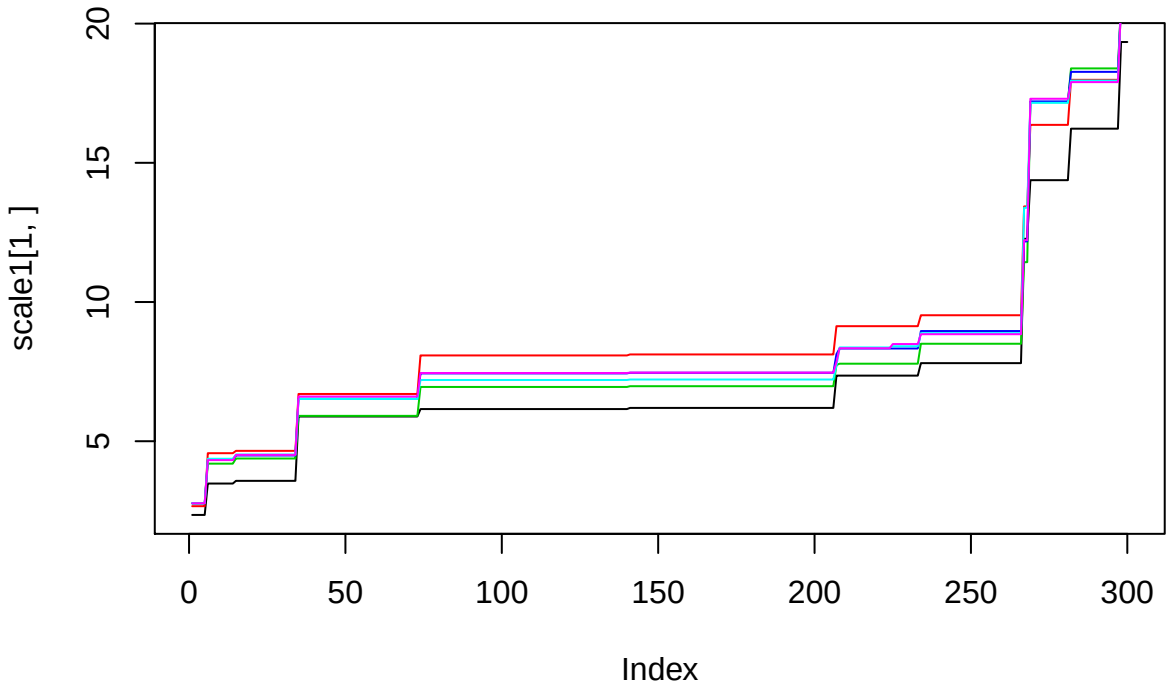
```
scale1 <- rbind(scale1, start1)
scale2 <- rbind(scale2, start2)
```

```
gradient1 <- gpdIcm:::ComputeGradient(y, start1, shape)
hesDiag1  <- gpdIcm:::ComputeHessianDiagonal(y, start1, shape)

gradient2 <- gpdIcm:::ComputeGradient(y, start2, shape)
hesDiag2  <- gpdIcm:::ComputeHessianDiagonal(y, start2, shape)

hessianSaved1 <- rbind(hessianSaved1, hesDiag1)
hessianSaved2 <- rbind(hessianSaved2, hesDiag2)

nNegWeights1 <- length(hesDiag1[hesDiag1 < 0])
nNegWeights2 <- length(hesDiag2[hesDiag2 < 0])

hesDiag1[hesDiag1 < 0] <- 1e-5
hesDiag2[hesDiag2 < 0] <- pmax(-hesDiag2[hesDiag2 < 0], 1e-5)


points1  <- cbind(c(0, cumsum(hesDiag1)),
                  c(0, cumsum(start1 * hesDiag1 - gradient1)))
points2  <- cbind(c(0, cumsum(hesDiag2)),
                  c(0, cumsum(start2 * hesDiag2 - gradient2)))

projection1 <- GreatestConvexMinorant(points1[,1], points1[, 2])
projection2 <- GreatestConvexMinorant(points2[,1], points2[, 2])

# lines(projection1$x.knots, projection1$y.knots, col = 2)

direction1 <- projection1$left.derivative - start1
direction2 <- projection2$left.derivative - start2

directionSaved1 <- rbind(directionSaved1, direction1)
directionSaved2 <- rbind(directionSaved2, direction2)

nextIterate1 <- gpdIcm:::LineSearchICM(y, start1, direction1, gradient1, shape)
nextIterate2 <- gpdIcm:::LineSearchICM(y, start2, direction2, gradient2, shape)

# plot(direction1, col = 4)
# points(direction2, col = 2)

# plot(start)
# points(nextIterate1$scale, col = 4)
# points(nextIterate2$scale, col = 2)

deviance1 <- c(deviance1, gpdIcm::compute_nll_gpd(y, nextIterate1$scale, shape))
deviance2 <- c(deviance2, gpdIcm::compute_nll_gpd(y, nextIterate2$scale, shape))

start1 <- nextIterate1$scale
start2 <- nextIterate2$scale

scale1 <- rbind(scale1, start1)
scale2 <- rbind(scale2, start2)
```

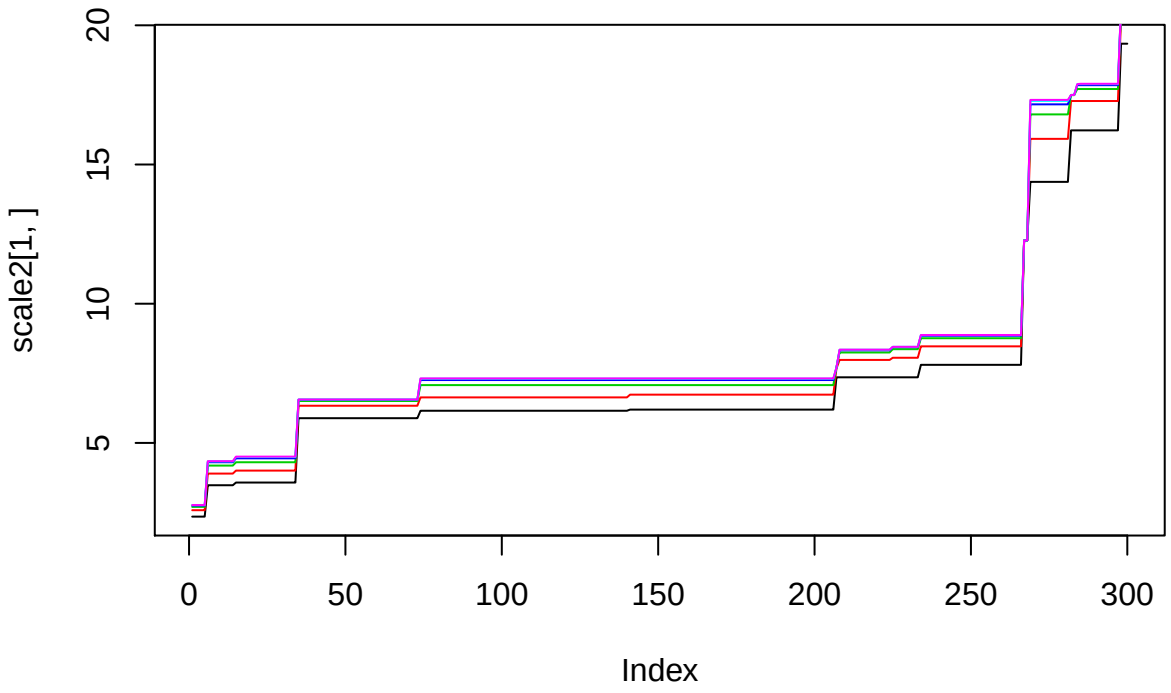
```
n <- length(deviance1)
deviance <- data.frame(method = rep(c("zero", "abs"), each = n),
                       deviance = c(deviance1, deviance2),
                       repetition = rep(1:n, 2))
ggplot(deviance, aes(x = repetition, y = deviance, col = method)) + geom_line()
```



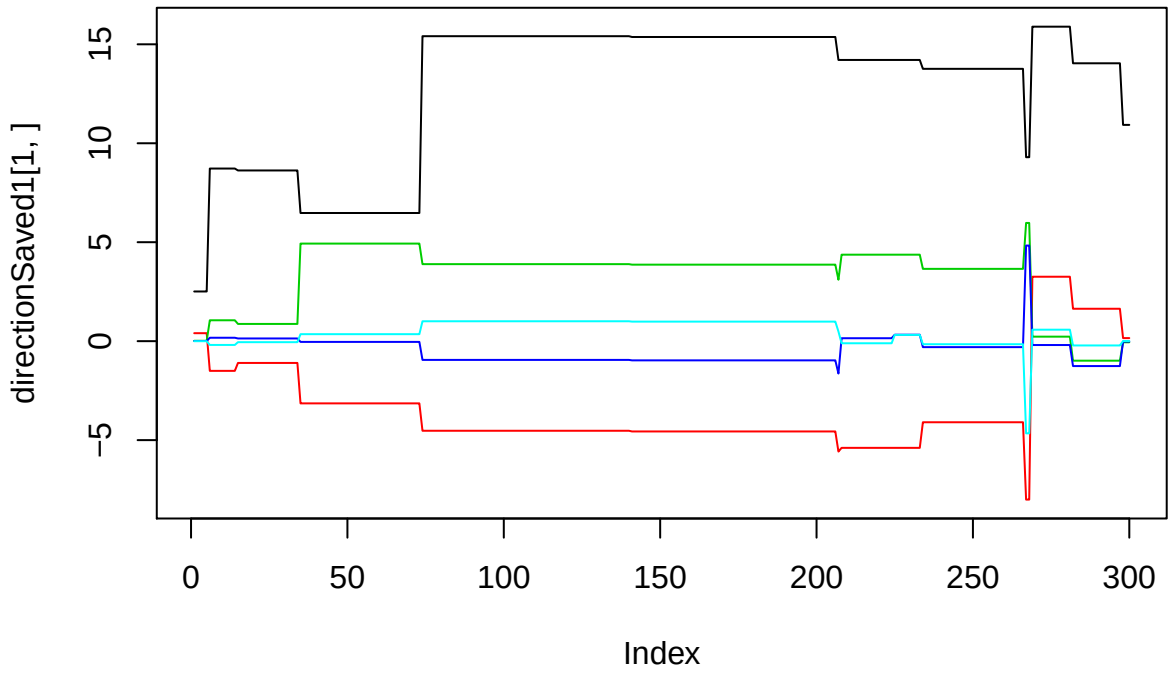

```
plot(scale1[1, ], type="l")
for(i in 2 : nrow(scale1)) lines(scale1[i, ], col = i)
```



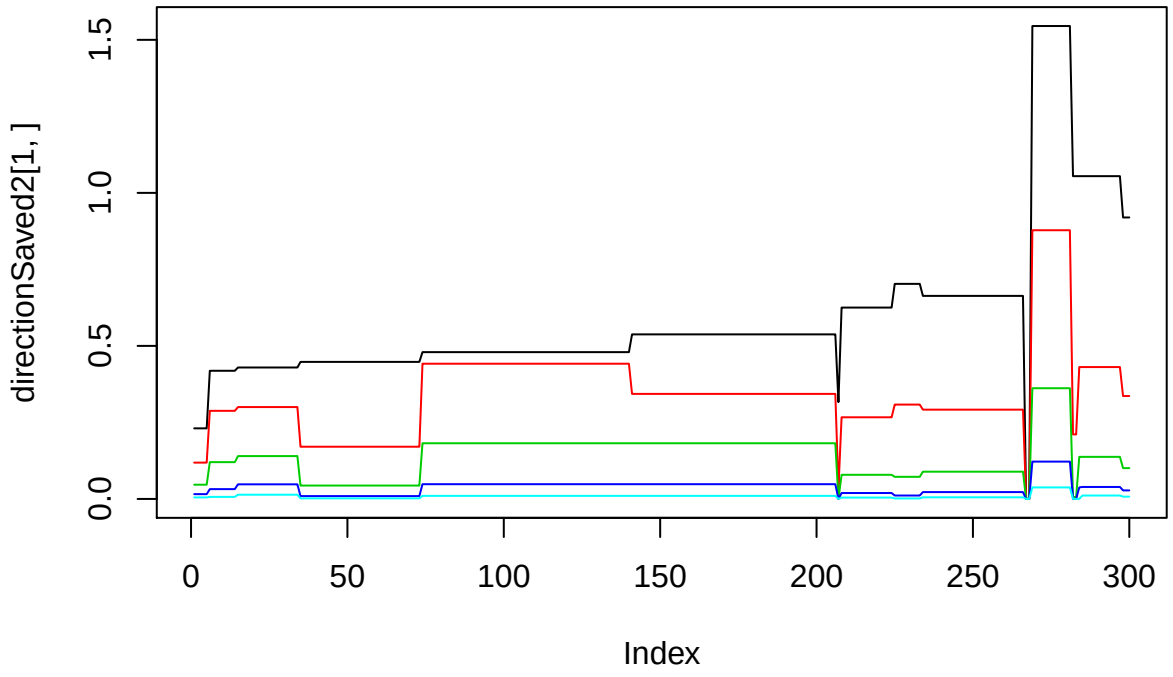

```
plot(scale2[1, ], type="l")
for(i in 2 : nrow(scale2)) lines(scale2[i, ], col = i)
```



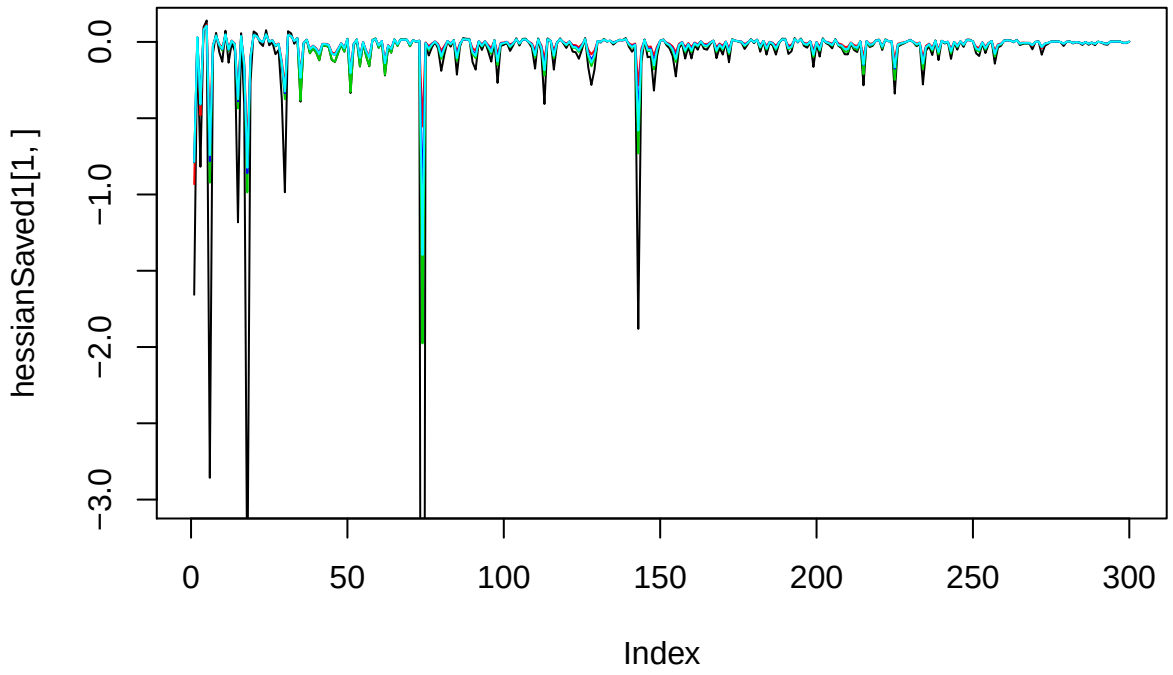

```
plot(directionSaved1[1, ], type ="l", ylim = c(min(directionSaved1), max(directionSaved1)))
for(i in 2 : nrow(directionSaved1)) lines(directionSaved1[i, ], col = i)
```



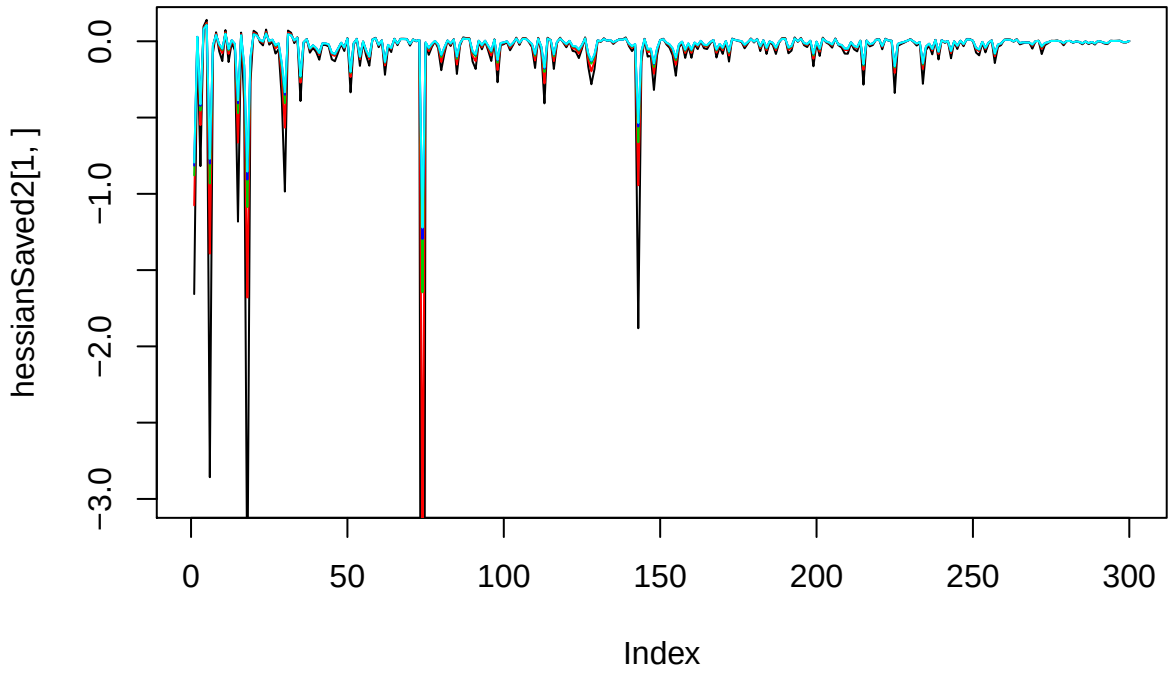

```
plot(directionSaved2[1, ], type="l", ylim = c(min(directionSaved2), max(directionSaved2)))
for(i in 2 : nrow(directionSaved2)) lines(directionSaved2[i, ], col = i)
```




```
plot(hessianSaved1[1, ], type="l", ylim = c(-3, 0.1))
for(i in 2 : nrow(hessianSaved1)) lines(hessianSaved1[i, ], col = i)
```




```
plot(hessianSaved2[1, ], type="l", ylim = c(-3, 0.1))
for(i in 2 : nrow(hessianSaved2)) lines(hessianSaved2[i, ], col = i)
```




```
apply(hessianSaved1, 1, function(x) {mean(x[x<0])})
```

```
##    hesDiag1    hesDiag1    hesDiag1    hesDiag1    hesDiag1
## -0.18664431 -0.05150981 -0.08124613 -0.06260023 -0.06588230
```

```r
apply(hessianSaved2, 1, function(x) {mean(x[x<0])})
```

```
##    hesDiag2    hesDiag2    hesDiag2    hesDiag2    hesDiag2
## -0.18664431 -0.10093147 -0.07383170 -0.06617770 -0.06406859
```