An Introduction to Statistical Modelling of Extreme Values

Package

```
library(ismev)
```

```
## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-23. For overview type 'help("mgcv-package")'.
```

```
library(extRemes)
```

```
## Loading required package: Lmoments

## Loading required package: distillery

## Loading required package: car

##
## Attaching package: 'extRemes'

## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```
source("declustering.r")
source("extreme_functions.r")
```

Chapter 5: Extremes of Dependent Sequences

Wooster

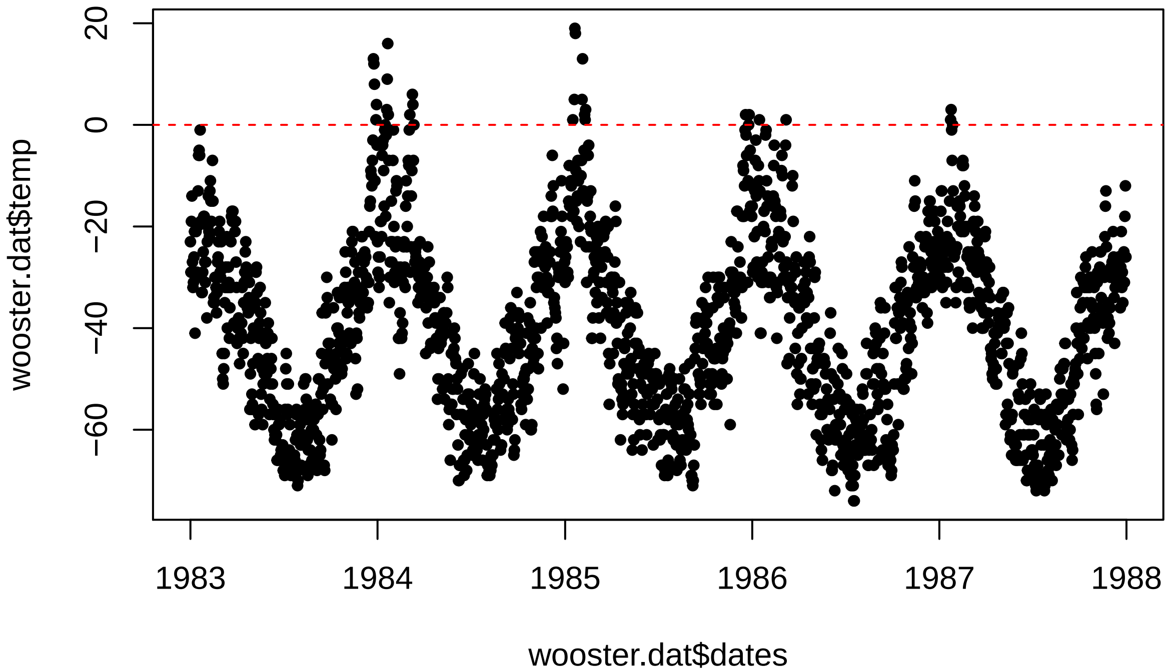
A numeric vector containing daily minimum temperatures, in degrees Fahrenheit, at Wooster, Ohio, over the period 1983 to 1988.

```
data("wooster")
wooster.dat <- data.frame(temp=-wooster)

start.date <- as.Date("01-01-1983", format="%d-%m-%Y")
end.date <- as.Date("01-01-1988", format="%d-%m-%Y")
dates <- seq(start.date, end.date, by=1)
dates <- dates[-length(dates)]

wooster.dat <- cbind(wooster.dat, dates)
wooster.dat$year <- as.numeric(format(as.Date(wooster.dat$dates,
                                              format="%d/%m/%Y"),"%Y"))
wooster.dat$month <- as.numeric(format(as.Date(wooster.dat$dates,
                                               format="%d/%m/%Y"),"%m"))
wooster.dat$spring[wooster.dat$month==3 |
                     wooster.dat$month==4 |
```

```r
                     wooster.dat$month==5 ] <- 1
wooster.dat$spring[is.na(wooster.dat$spring)] <- 0

wooster.dat$summer[wooster.dat$month==6 |
                     wooster.dat$month==7 |
                     wooster.dat$month==8 ] <- 1
wooster.dat$summer[is.na(wooster.dat$summer)] <- 0

wooster.dat$fall[wooster.dat$month==9 |
                   wooster.dat$month==10 |
                   wooster.dat$month==11 ] <- 1
wooster.dat$fall[is.na(wooster.dat$fall)] <- 0

wooster.dat$winter[wooster.dat$month==12 |
                     wooster.dat$month==1 |
                   wooster.dat$month==2 ] <- 1
wooster.dat$winter[is.na(wooster.dat$winter)] <- 0

head(wooster.dat)
```

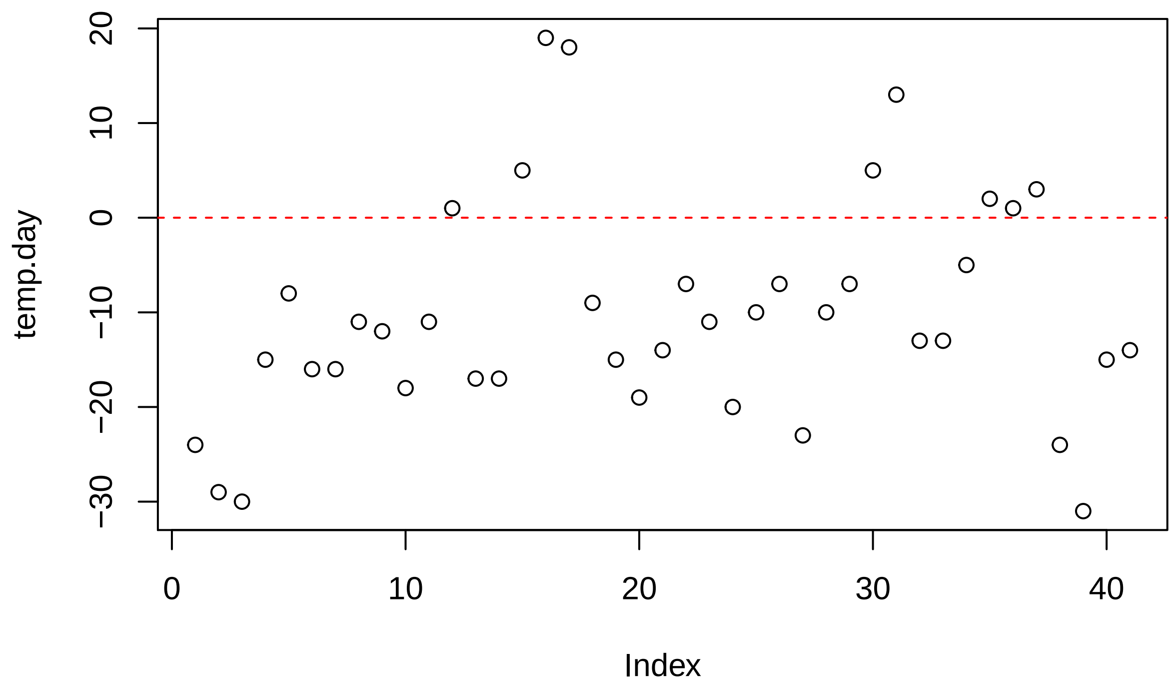
```
##   temp      dates year month spring summer fall winter
## 1  -23 1983-01-01 1983     1      0      0    0      1
## 2  -29 1983-01-02 1983     1      0      0    0      1
## 3  -19 1983-01-03 1983     1      0      0    0      1
## 4  -14 1983-01-04 1983     1      0      0    0      1
## 5  -27 1983-01-05 1983     1      0      0    0      1
## 6  -32 1983-01-06 1983     1      0      0    0      1
```

```r
plot(wooster.dat$dates, wooster.dat$temp, pch=20)
abline(h=0, col="red", lty=2)
```



Daily example

```
temp.day <- wooster.dat$temp[wooster.dat$dates>"1985-01-04" &
                        wooster.dat$dates<"1985-02-15"]
plot(temp.day)
abline(h=0, col="red", lty=2)
```



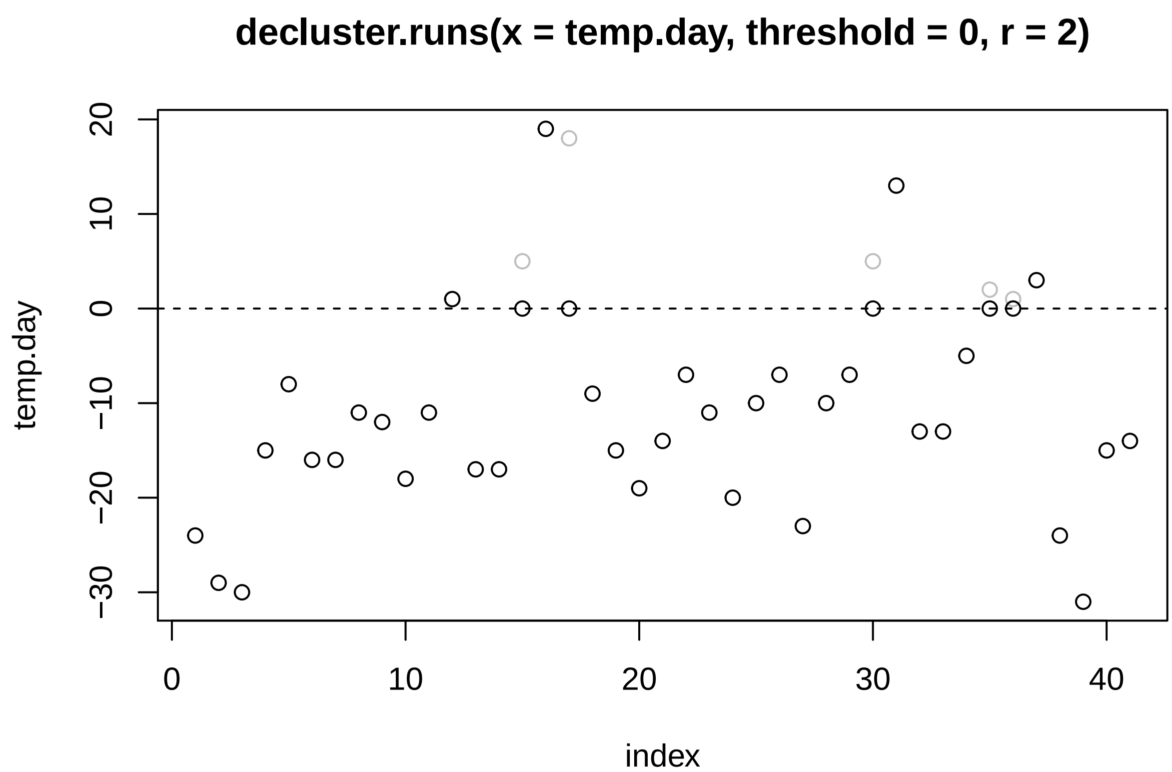

```r
cluster2(temp.day, threshold = 0)
```

```
## [1]  1 19 13  3
```

```r
y <- decluster(temp.day, threshold = 0, r = 2)
y
```

```
##
##  temp.day  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  4
##
##  Run length =  2
```

```r
plot(y)
```




```r
cluster4(temp.day, threshold = 0)
```

```
## [1] 19 13
```

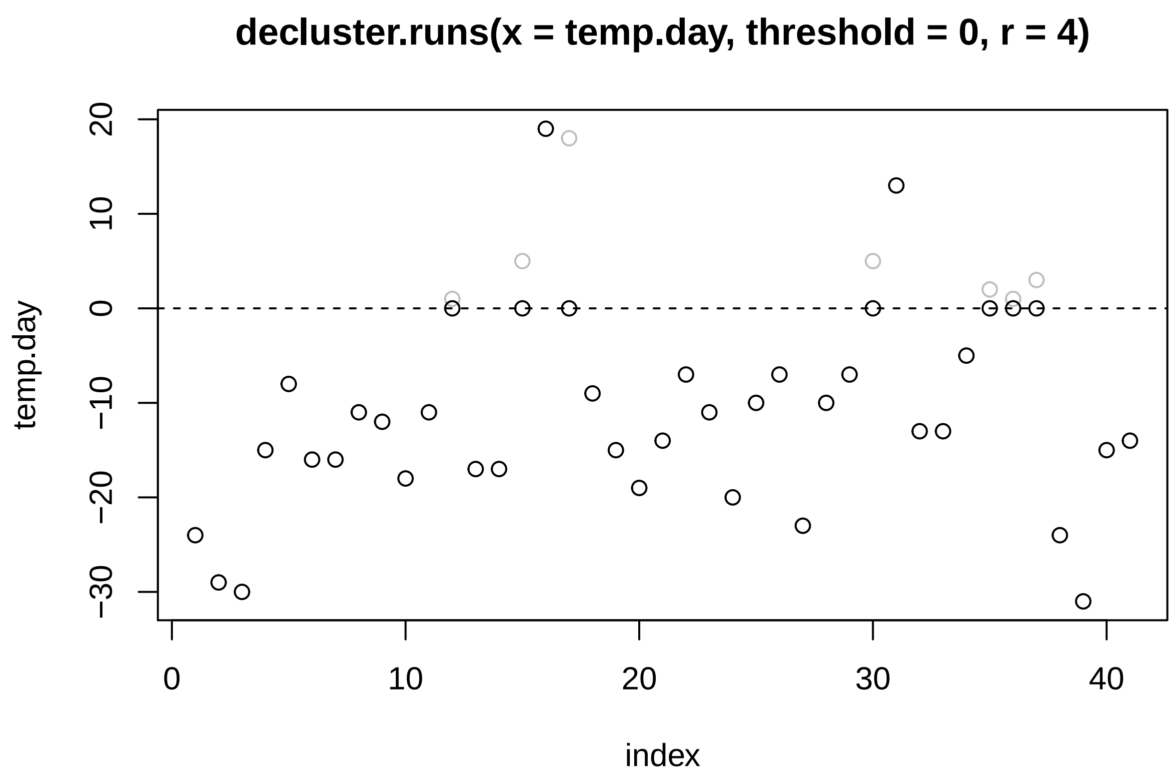
```r
y <- decluster(temp.day, threshold = 0, r = 4)
y
```

```
##
##  temp.day  declustered via runs  declustering.
```

```
## Warning in max(ind2.1[ind2.1 < K]): no non-missing arguments to max;
## returning -Inf
```

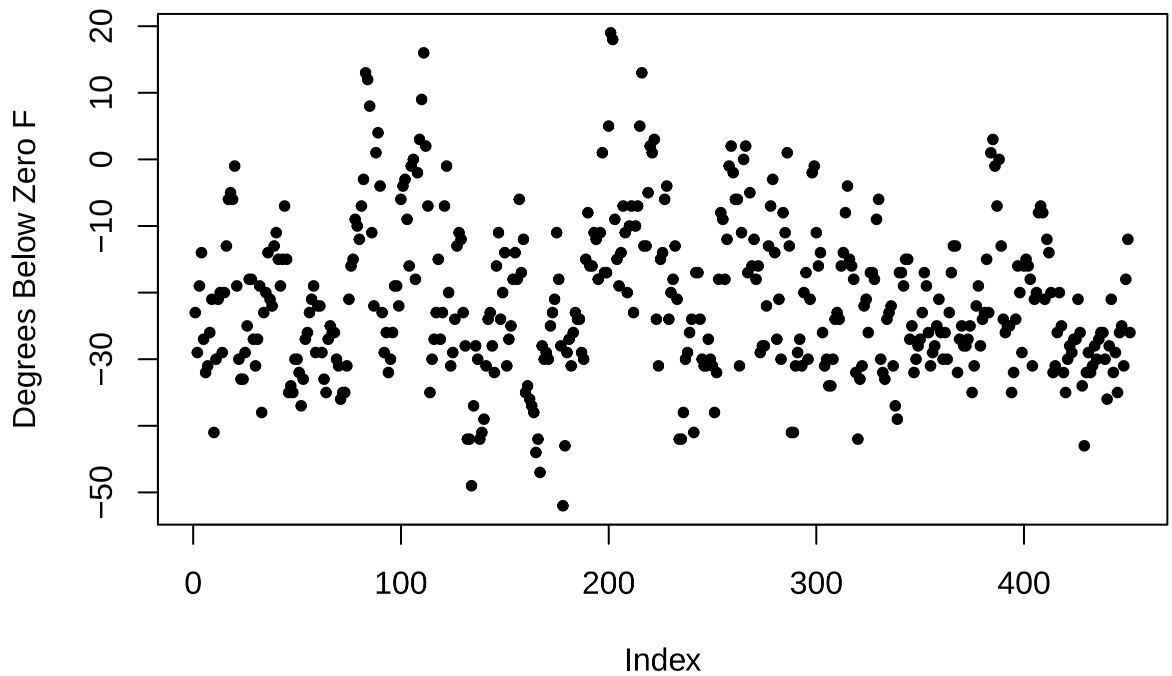
```
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  2
##
##  Run length =  4
```

```r
plot(y)
```



Winter example

```
winter <- wooster.dat$temp[wooster.dat$winter==1]
plot(winter, pch=20, ylab="Degrees Below Zero F")
```

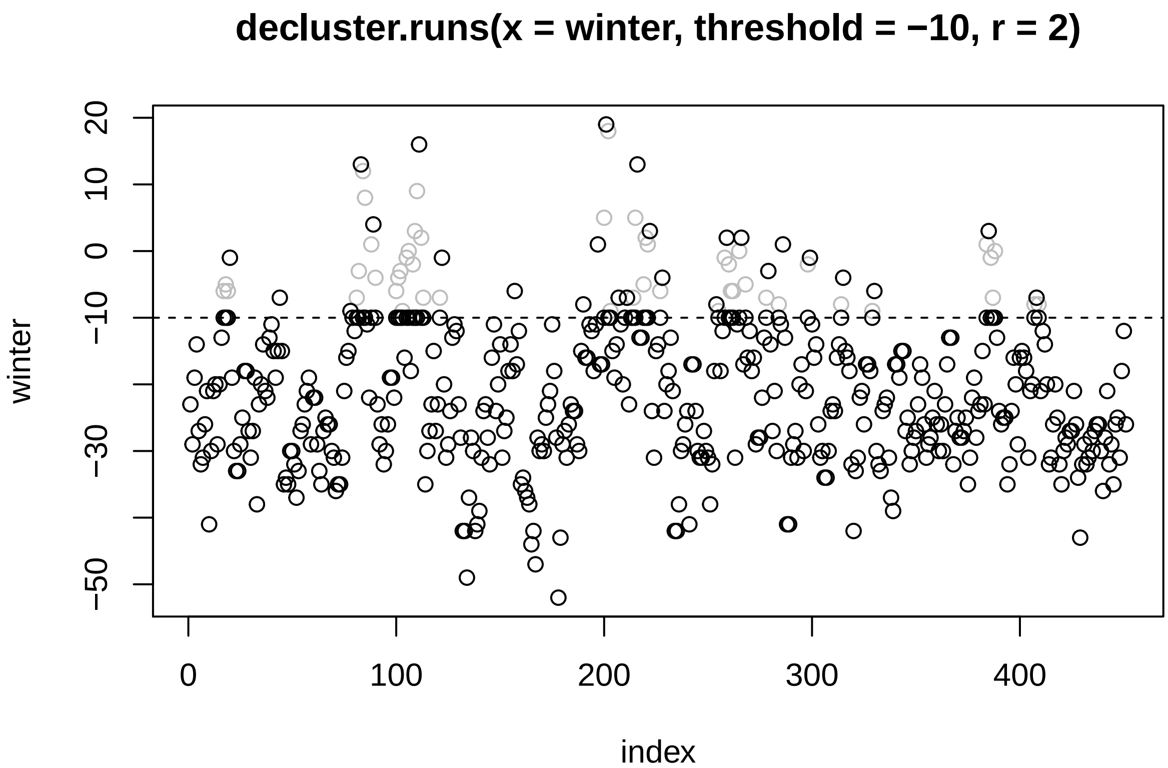


u=-10 and r=2

```
y102 <- decluster(winter, threshold = -10, r=2)
y102
```

```
##
##  winter  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  26
##
##  Run length =  2
```

```
plot(y102)
```




```r
fitgpd102 <- gpd.fit(c(y102), threshold = -10)
```

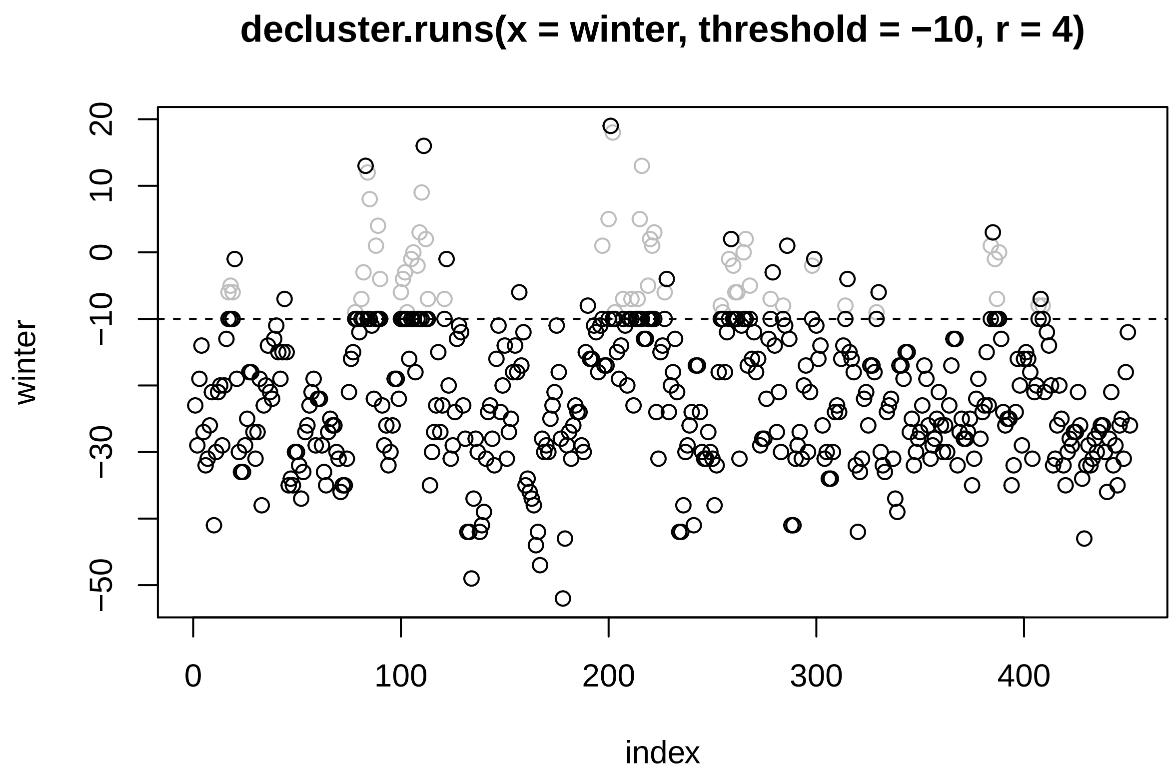
```
## $threshold
## [1] -10
##
## $nexc
## [1] 26
##
## $conv
## [1] 0
##
## $nllh
## [1] 84.41435
##
## $mle
## [1] 14.0110333 -0.3933294
##
## $rate
## [1] 0.05764967
##
## $se
## [1] 3.9404105 0.2158193
```

u=-10 and r=4

```r
y104 <- decluster(winter, threshold = -10, r=4)
y104
```

```
##
##  winter  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  17
##
##  Run length =  4
```

```r
plot(y104)
```




```r
fitgpd104 <- gpd.fit(c(y104), threshold = -10)
```

```
## $threshold
## [1] -10
##
## $nexc
## [1] 17
##
## $conv
## [1] 0
##
## $nllh
## [1] 55.92855
```

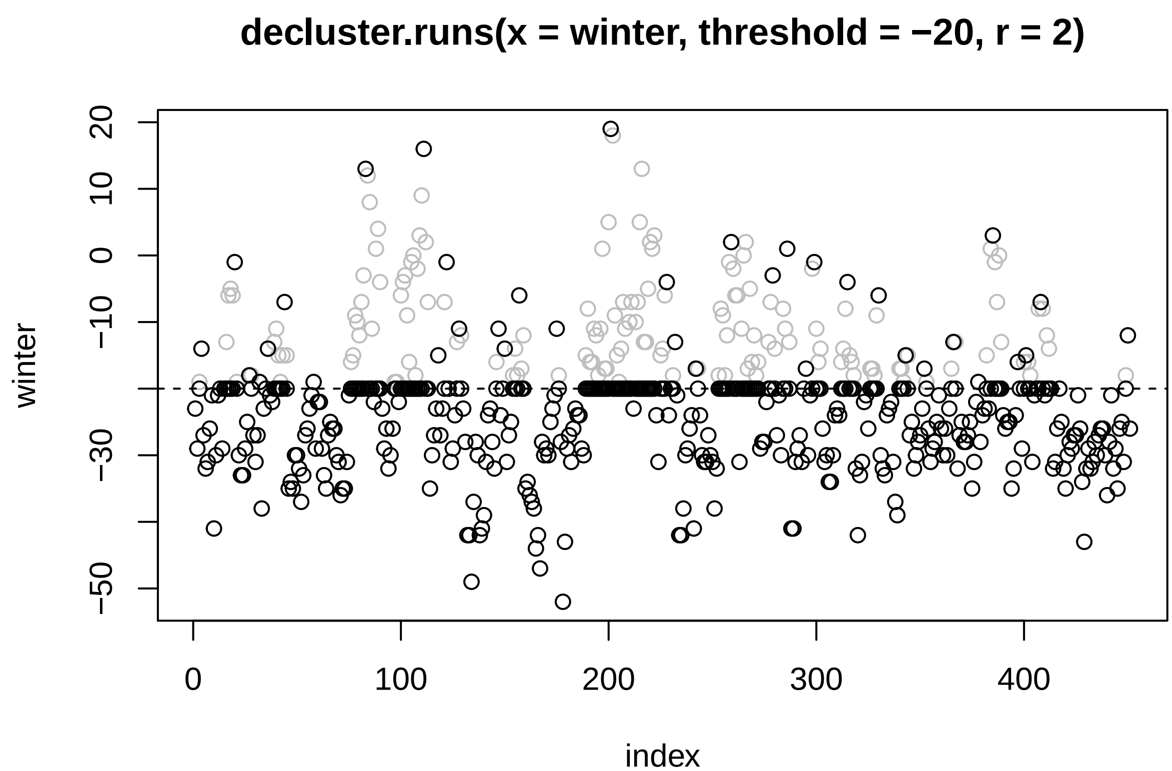
```
##
## $mle
## [1] 14.8270931 -0.4065942
##
## $rate
## [1] 0.03769401
##
## $se
## [1] 5.5446169 0.3011815
```

u=-20 and r=2

```r
y202 <- decluster(winter, threshold = -20, r=2)
y202
```

```
##
##  winter  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  36
##
##  Run length =  2
```

```r
plot(y202)
```



```r
fitgpd202 <- gpd.fit(c(y202), threshold = -20)
```

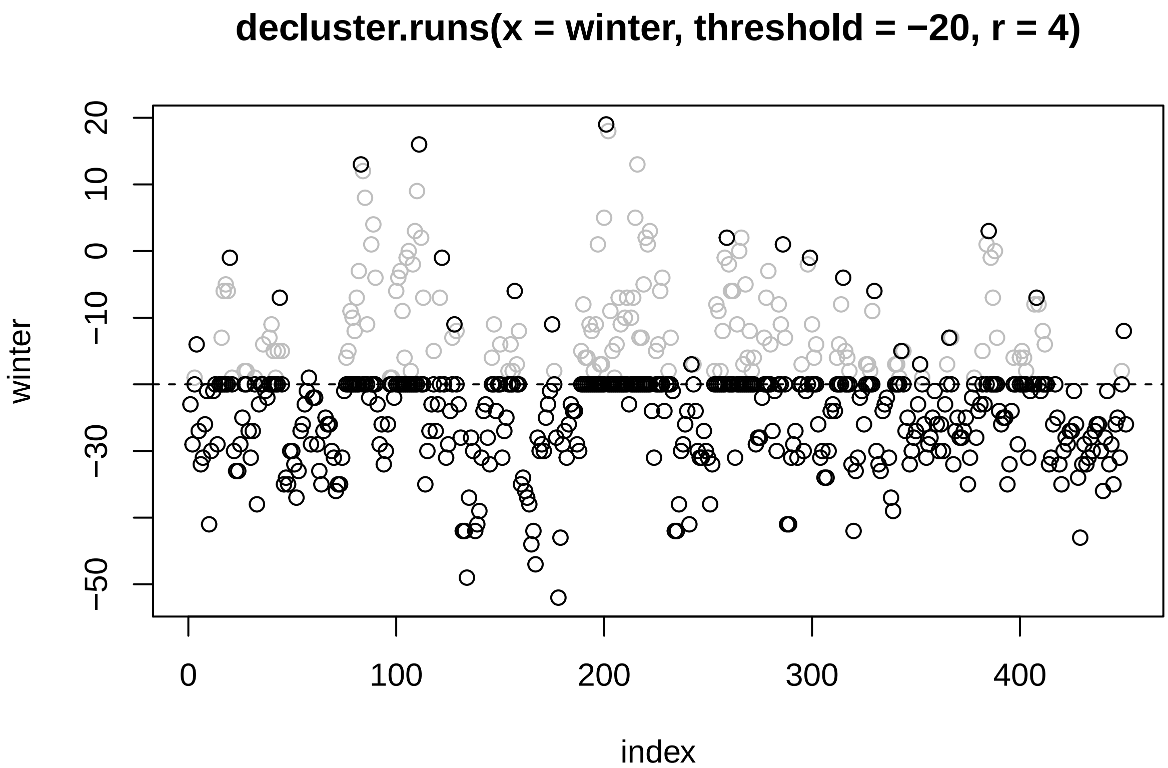
```
## $threshold
## [1] -20
##
## $nexc
## [1] 36
##
## $conv
## [1] 0
##
## $nllh
## [1] 124.5006
##
## $mle
## [1] 15.6393726 -0.2916171
##
## $rate
## [1] 0.07982262
##
## $se
## [1] 3.5890459 0.1650612
```

u=-20 and r=4

```r
y204 <- decluster(winter, threshold = -20, r=4)
y204
```

```
##
##  winter  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  1
##
##  Number of clusters =  23
##
##  Run length =  4
```

```r
plot(y204)
```

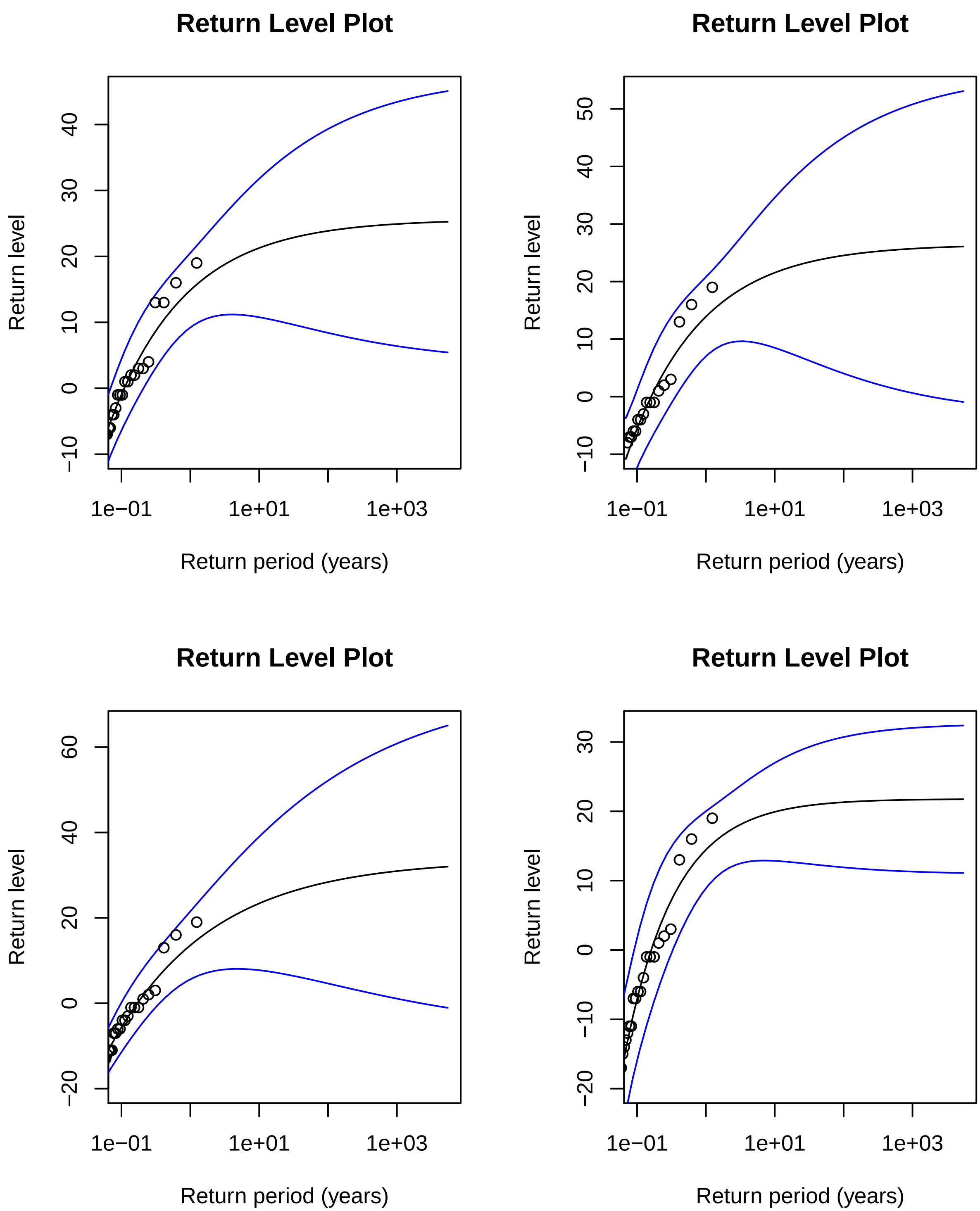



```
fitgpd204 <- gpd.fit(c(y204), threshold = -20)
```

```
## $threshold
## [1] -20
##
## $nexc
## [1] 23
##
## $conv
## [1] 0
##
## $nllh
## [1] 83.22574
##
## $mle
## [1] 24.8745884 -0.5953015
##
## $rate
## [1] 0.05099778
##
## $se
## [1] 7.0967735 0.2308971
```

```
par(mfrow=c(2,2))
z <- fitgpd102
gpd.rl(z$mle, z$threshold, z$rate, z$n, z$npy, z$cov, z$data, z$xdata)
z <- fitgpd104
```

```r
gpd.rl(z$mle, z$threshold, z$rate, z$n, z$npy, z$cov, z$data, z$xdata)
z <- fitgpd202
gpd.rl(z$mle, z$threshold, z$rate, z$n, z$npy, z$cov, z$data, z$xdata)
z <- fitgpd204
gpd.rl(z$mle, z$threshold, z$rate, z$n, z$npy, z$cov, z$data, z$xdata)
```


Return Level Plot
Return level
Return period (years)
1e−01
1e+01
1e+03
−10
0
10
20
30
40
Return Level Plot
Return level
Return period (years)
1e−01
1e+01
1e+03
−10
0
10
20
30
40
50
Return Level Plot
Return level
Return period (years)
1e−01
1e+01
1e+03
−20
0
20
40
60
Return Level Plot
Return level
Return period (years)
1e−01
1e+01
1e+03
−20
−10
0
10
20
30

Dow Jones Index Series

Data

```
library(ismev)
source("extreme_functions.r")
data("dowjones")
str(dowjones)
```

```
## 'data.frame':    1304 obs. of  2 variables:
##  $ Date : POSIXt, format: "1995-09-11 09:00:00" "1995-09-12 09:00:00" ...
##  $ Index: num  4705 4747 4766 4802 4798 ...
```

Date

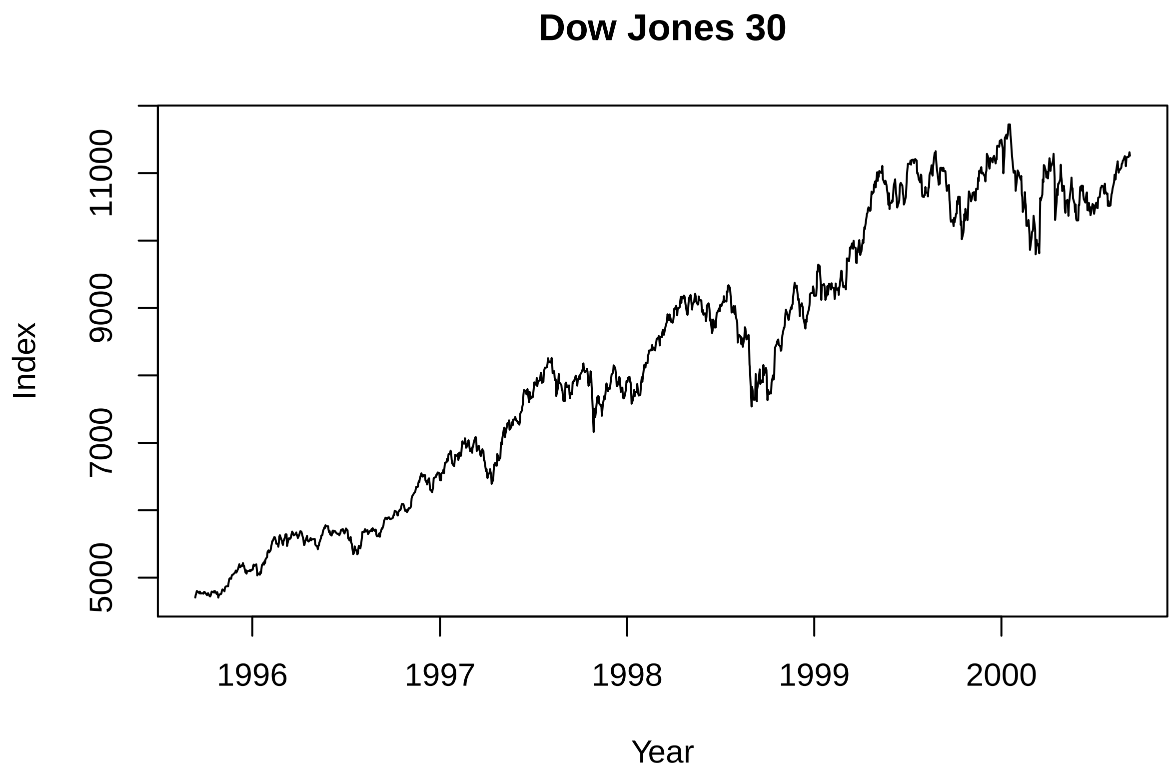
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
dates <- parse_date_time(x = dowjones$Date, orders ="Y-m-d H:M:S")
```

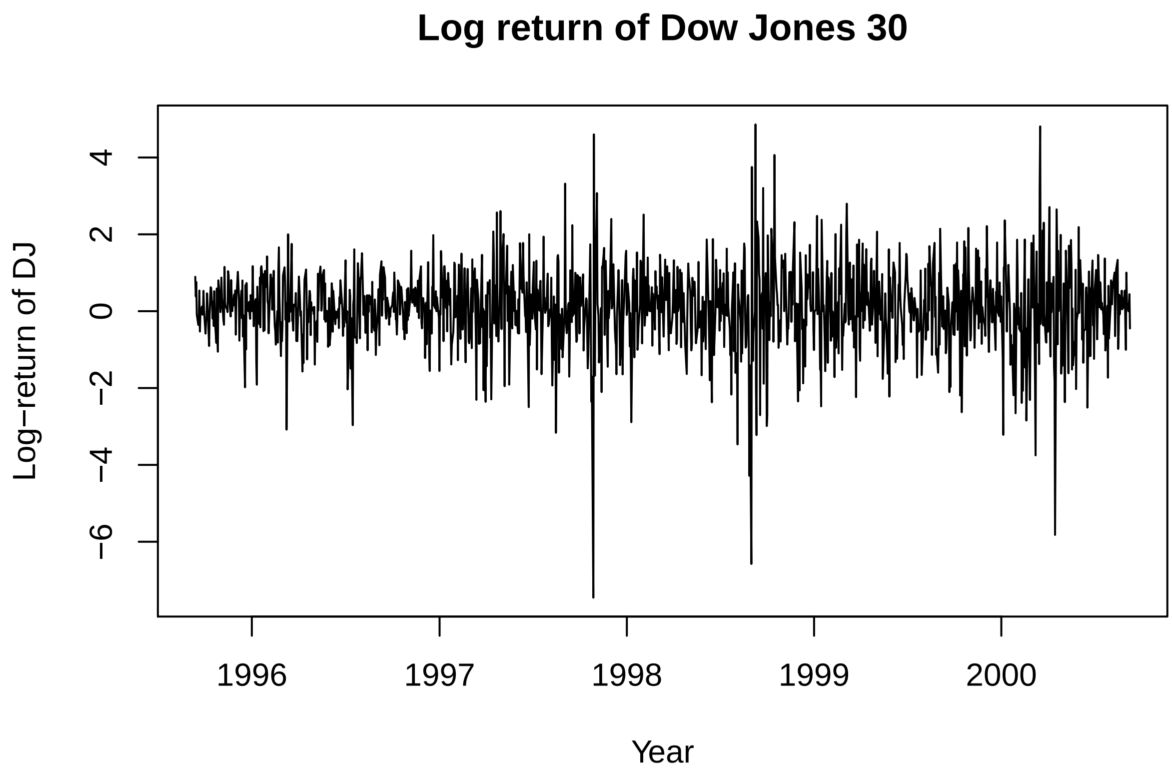
Price

```
price <- dowjones$Index
plot(x= dates, y=price, t="l",
     ylab="Index", xlab="Year", main="Dow Jones 30")
```



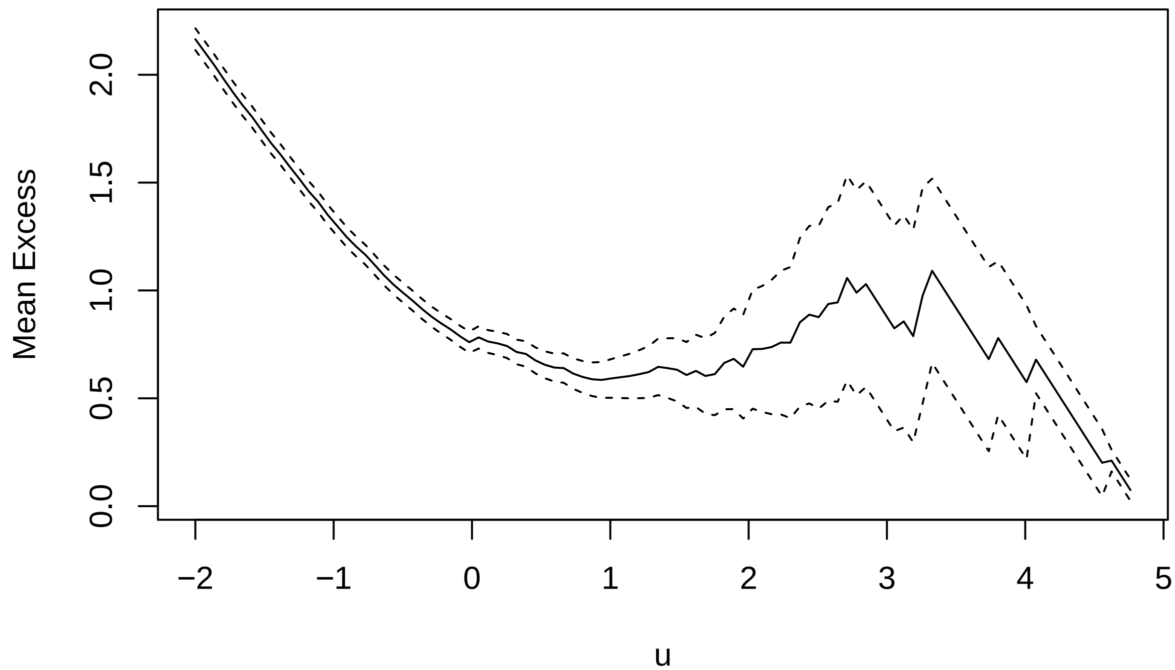
Log-return

```
ret <- diff(log(price))*100
plot(x= dates[-1], y=ret, t="l",
     ylab="Log-return of DJ", xlab="Year",
     main="Log return of Dow Jones 30")
```



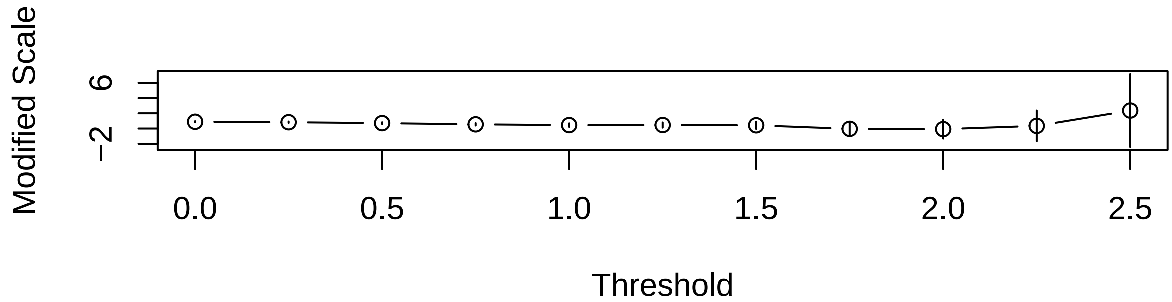
Mean residual life plot

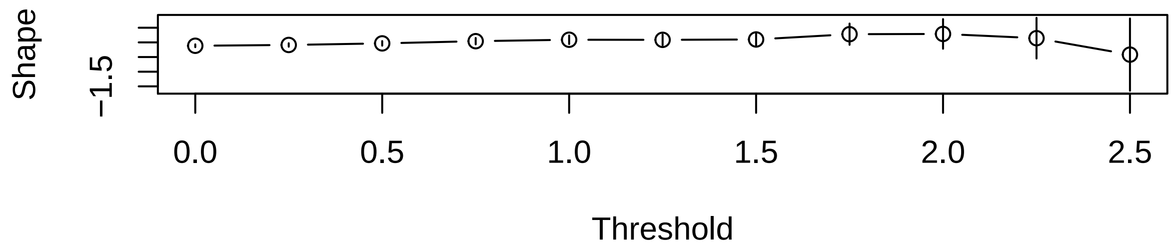
```
mrl.plot(ret, umin = -2)
```



Fitting the GPD Model Over a Range of Thresholds

```
gpd.fitrange(ret, umin = 0, umax = 2.5, nint = 11)
```

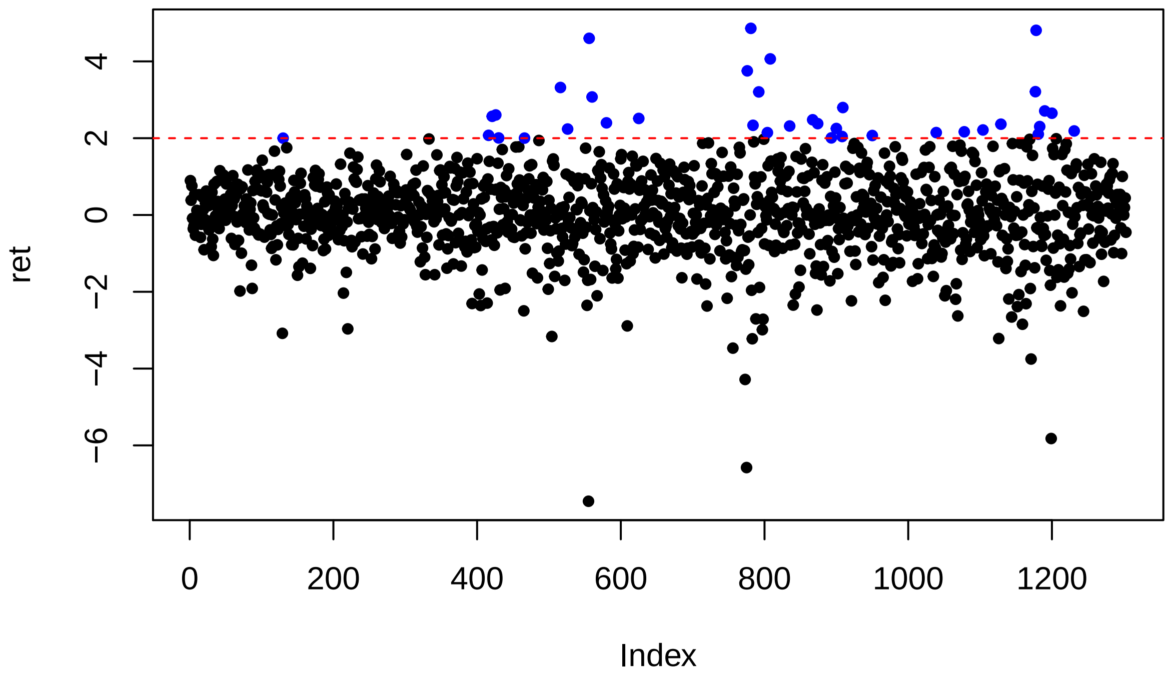




Threshold

```
u <- 2
```

```
col.exceed <- ret
col.exceed[col.exceed > u] <- "blue"
col.exceed[col.exceed <= u] <- "black"

plot(ret, pch=20, col=col.exceed)
abline(h=u, col="red", lty=2)
```



Fit the GPD model

```
fitgpd1 <- gpd.fit(ret, threshold = u, npy = 365)
```

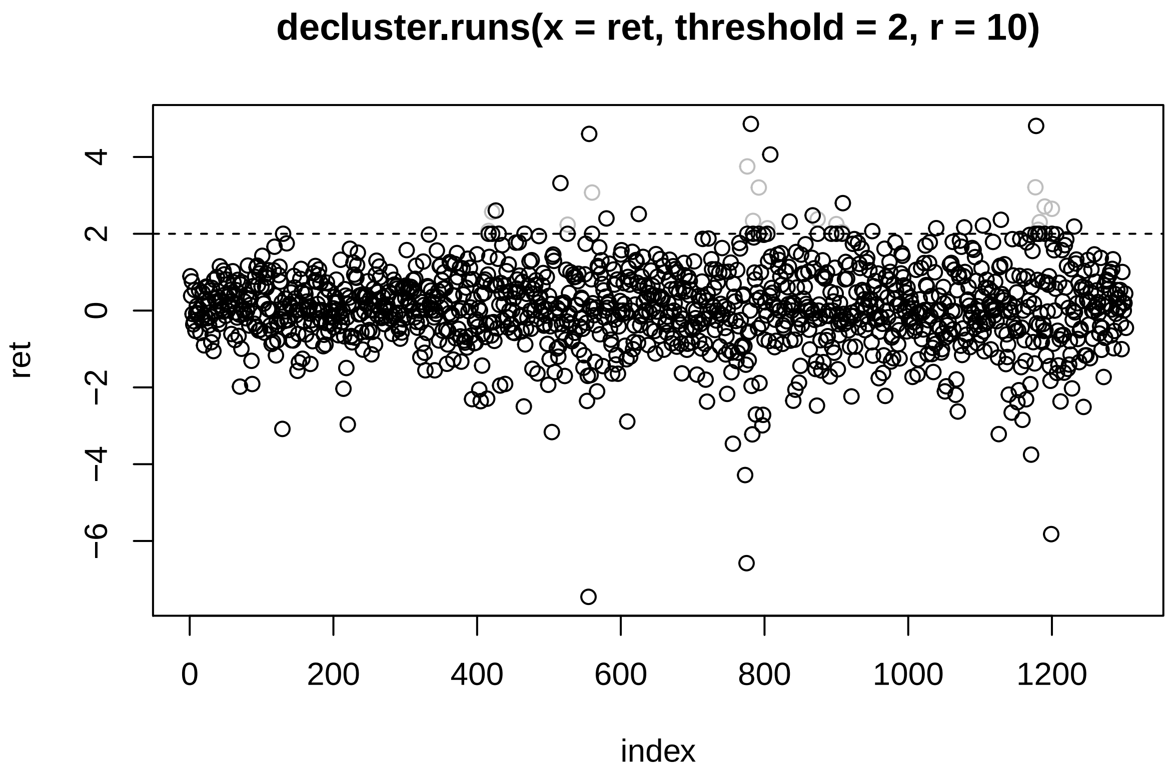
```
## $threshold
## [1] 2
##
## $nexc
## [1] 37
##
## $conv
## [1] 0
##
## $nllh
## [1] 21.64016
##
## $mle
## [1] 0.4951310 0.2879248
##
## $rate
## [1] 0.02839601
##
## $se
## [1] 0.1495846 0.2578784
```

Decluster model

```
y <- decluster(ret, threshold = 2, r = 10)
y
```

```
##
##  ret  declustered via runs  declustering.
##
##  Estimated extremal index (intervals estimate) =  0.9384659
##
##  Number of clusters =  19
##
##  Run length =  10
```

```
plot(y)
```




```
fitgpd2 <- gpd.fit(c(y), threshold = 2)
```

```
## $threshold
## [1] 2
##
## $nexc
## [1] 19
##
## $conv
## [1] 0
##
```

```
## $nllh
## [1] 15.12354
##
## $mle
## [1] 0.5635274 0.3694016
##
## $rate
## [1] 0.01458173
##
## $se
## [1] 0.2557343 0.4005749
```