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
```

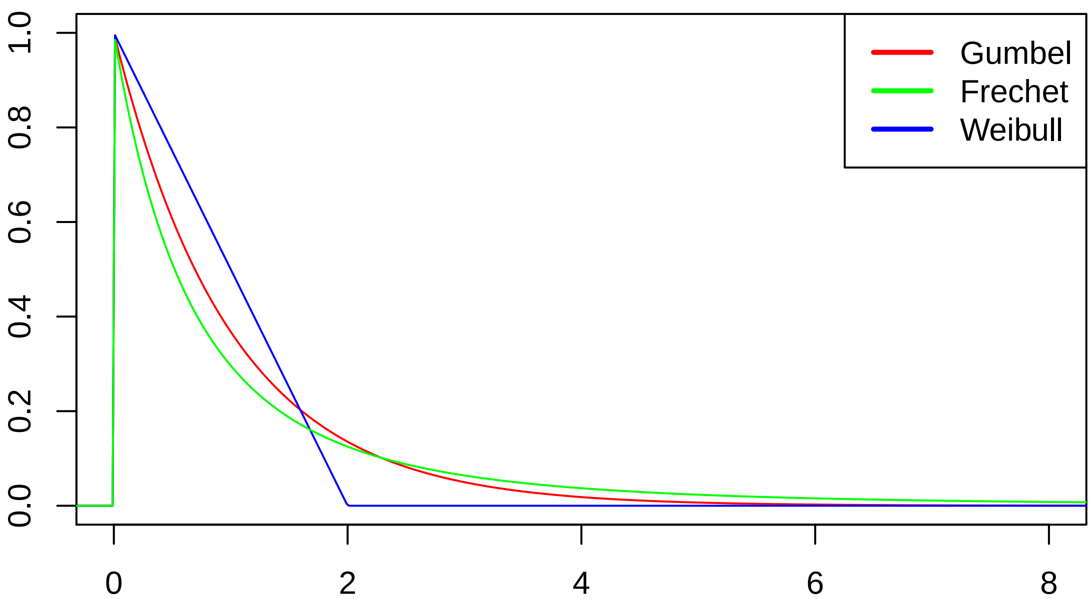
```
source("extreme_functions.r")
```

Chapter 4: Threshold Models

The Generalsed pareto distribution

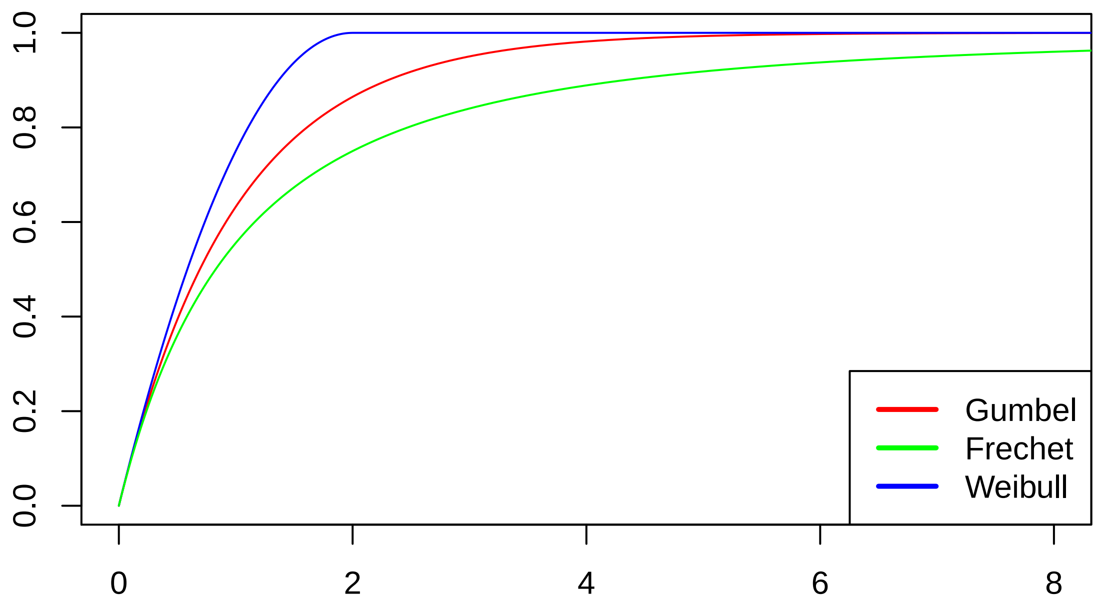
PDF

```
library(extRemes)
xx <- seq(-10, 10, length=1000)
gum <- devd(xx, loc = 0, scale = 1, shape = 0, type = c("GP"))
fre <- devd(xx, loc = 0, scale = 1, shape = -0.5, type = c("GP"))
wei <- devd(xx, loc = 0, scale = 1, shape = 0.5, type = c("GP"))

plot(x=xx, y=gum, t="l", xlim=c(0,8), ylim=c(0,1), col="red", ylab="", xlab="")
lines(x=xx, fre, col="blue")
lines(x=xx, wei, col="green")

legend("topright", legend = c("Gumbel", "Frechet", "Weibull"),
       lwd=c(2.5,2.5,2.5),col=c("red","green","blue"))
```



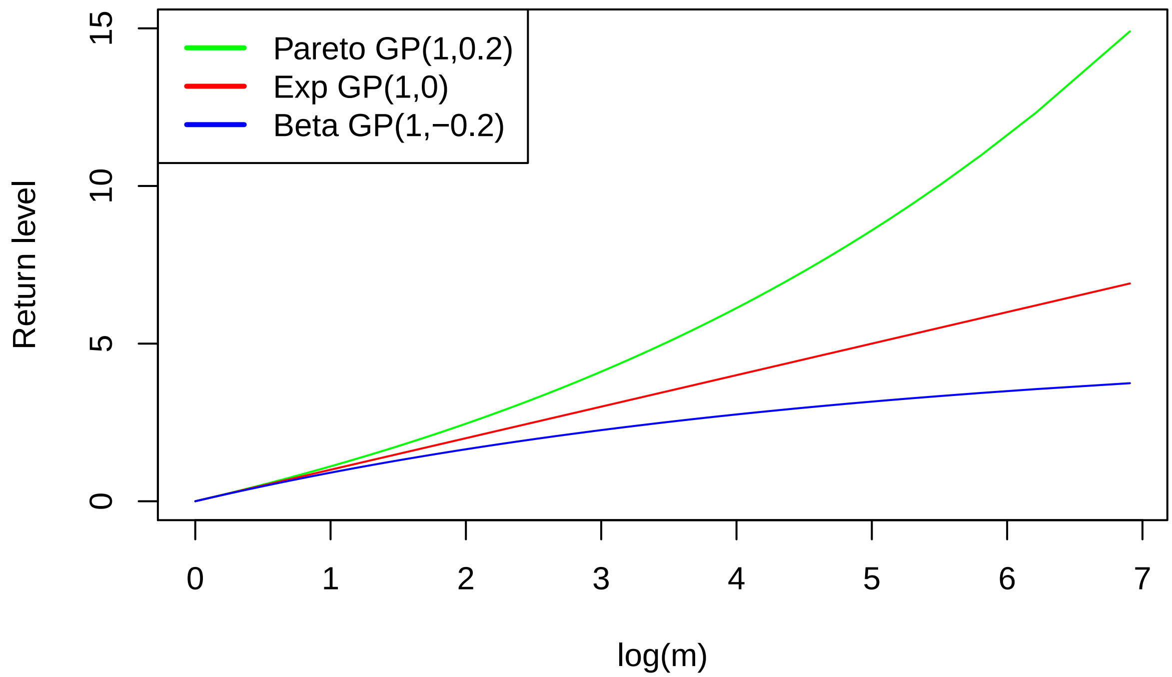
CDF

```
library(extRemes)
xx <- seq(0, 20, length=1000)
gum <- pevd(xx, loc = 0, scale = 1, shape = 0, type = c("GP"))
fre <- pevd(xx, loc = 0, scale = 1, shape = -0.5, type = c("GP"))
wei <- pevd(xx, loc = 0, scale = 1, shape = 0.5, type = c("GP"))

plot(x=xx, y=gum, t="l", xlim=c(0,8), ylim=c(0,1), col="red", ylab="", xlab="")
lines(x=xx, fre, col="blue")
lines(x=xx, wei, col="green")

legend("bottomright", legend = c("Gumbel", "Frechet", "Weibull"),
       lwd=c(2.5,2.5,2.5),col=c("red","green","blue"))
```



m-observation return level

```
p <- seq(0, 1, by = 0.001)
m <- 1/p

rl.pare <- (m^(0.2)-1)/0.2
rl.exp <- log(m)
rl.beta <- (m^(-0.2)-1)/-0.2

plot(x=log(m), y=rl.pare, t="l", col="green",
     ylim=c(0,15), ylab="Return level", xlab="log(m)")
lines(x=log(m), y=rl.exp, col="red")
lines(x=log(m), y=rl.beta, col="blue")
legend("topleft", legend =
         c("Pareto GP(1,0.2)", "Exp GP(1,0)", "Beta GP(1,-0.2)"),
       lwd=c(2.5,2.5,2.5),col=c("green","red","blue"))
```



Daily Rainfall data (using ismev)

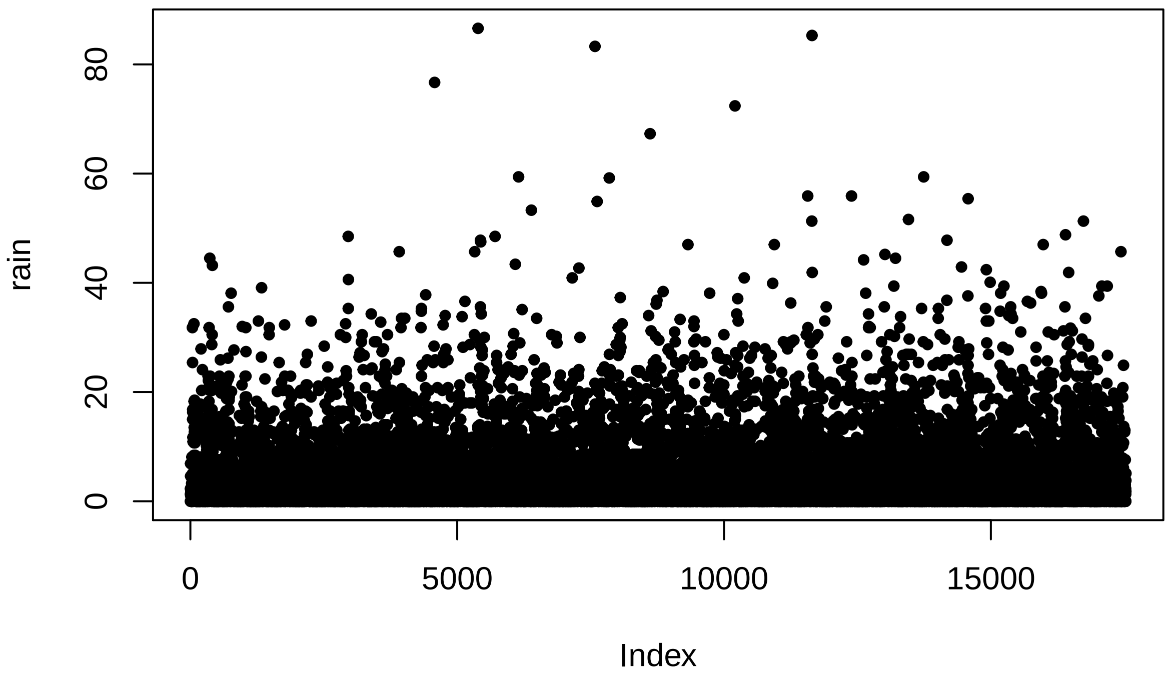
Data

A numeric vector containing daily rainfall accumulations at a location in south-west England over the period 1914 to 1962

```
library(ismev)
data(rain)
str(rain)
```

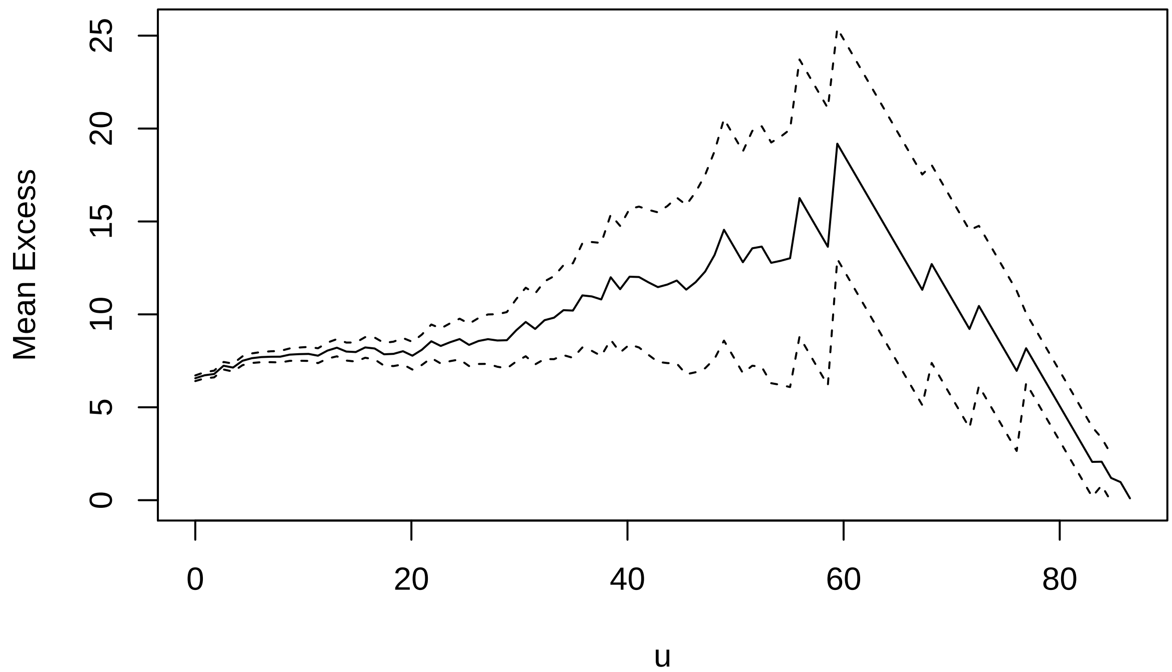
```
##  num [1:17531] 0 2.3 1.3 6.9 4.6 0 1 1.5 1.8 1.8 ...
```

```
plot(rain, pch=20)
```



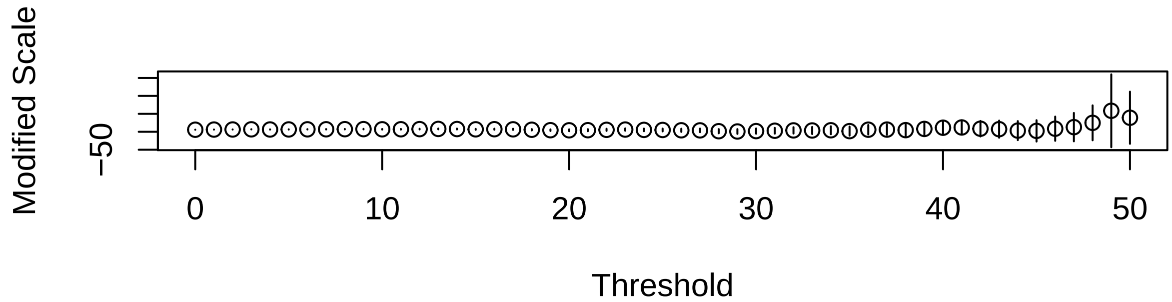
Mean residual life plot

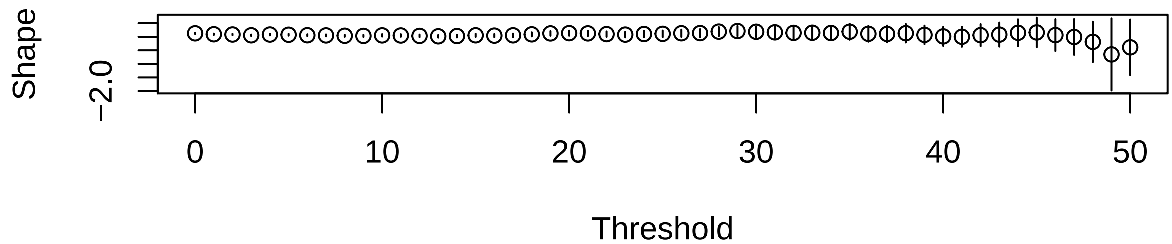
```
mrl.plot(rain)
```



Fitting the GPD Model Over a Range of Thresholds

```
gpd.fitrange(rain, umin = 0, umax = 50, nint = 51)
```

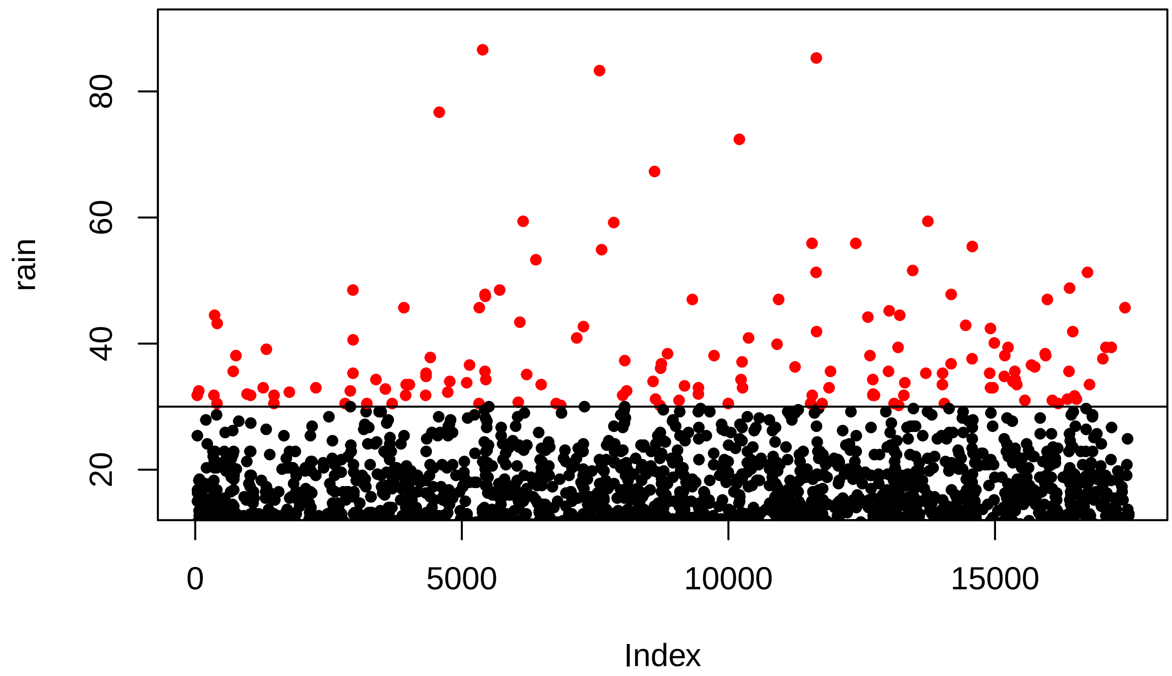




Threshold

```
u <- 30
```

```
col.exceed <- rain
col.exceed[col.exceed > u] <- "red"
col.exceed[col.exceed <= u] <- "black"

plot(rain, pch=20, ylim=c(15,90), col=col.exceed)
abline(h=u)
```



Fit the GPD model

```
fitgpd <- gpd.fit(rain, threshold = u, npy = 365)
```

```
## $threshold
## [1] 30
##
## $nexc
## [1] 152
##
## $conv
## [1] 0
##
## $nllh
## [1] 485.0937
##
## $mle
## [1] 7.4422639 0.1843027
##
## $rate
## [1] 0.008670355
##
## $se
## [1] 0.9587773 0.1011714
```

```r
options(digits = 3)
# parameters
fitgpd$mle
```

```
## [1] 7.442 0.184
```

```r
# log-likelihood
-fitgpd$nllh
```

```
## [1] -485
```

```r
# covariance
cov.two <- fitgpd$cov; cov.two
```

```
##         [,1]    [,2]
## [1,]  0.9193 -0.0655
## [2,] -0.0655  0.0102
```

```r
# standard deviation
fitgpd$se
```

```
## [1] 0.959 0.101
```

```r
# confidential interval
ci <- cbind(fitgpd$mle-1.96*fitgpd$se, fitgpd$mle+1.96*fitgpd$se)
colnames(ci) <- c("lower", "upper")
rownames(ci) <- c("sigma", "xi")
ci
```

```
##        lower upper
## sigma  5.563 9.321
## xi    -0.014 0.383
```

```r
# number of complete sample
n <- fitgpd$n; n
```

```
## [1] 17531
```

```r
# number of exceedance
fitgpd$nexc
```

```
## [1] 152
```

```r
# zeta(exceedance rate)
zeta <- fitgpd$rate; zeta
```

```
## [1] 0.00867
```

```r
# var of zeta
var.zeta <- zeta*(1-zeta)/n; var.zeta
```

```
## [1] 4.9e-07
```

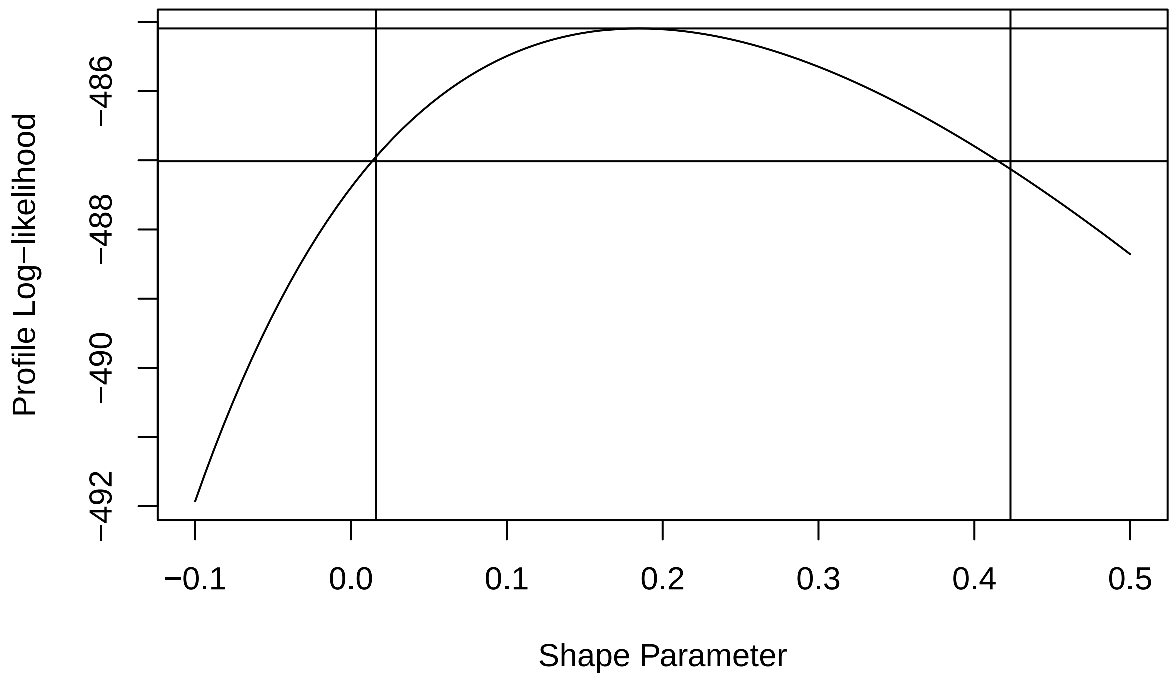
```r
# complete covariance
cov <- matrix(c(var.zeta, 0, 0, 0, cov.two[1,], 0, cov.two[2,]),
       nrow = 3, byrow = T, dimnames = list(c("zeta","sigma","xi"),
                                            c("zeta","sigma","xi")))
cov
```

```
##          zeta   sigma      xi
## zeta  4.9e-07  0.0000  0.0000
```

```
## sigma 0.0e+00  0.9193 -0.0655
## xi    0.0e+00 -0.0655  0.0102
```

```r
# Profile likelihood for shape
cgpd.profxi(fitgpd, xlow = -0.1, xup = 0.5, nint = 100)
```

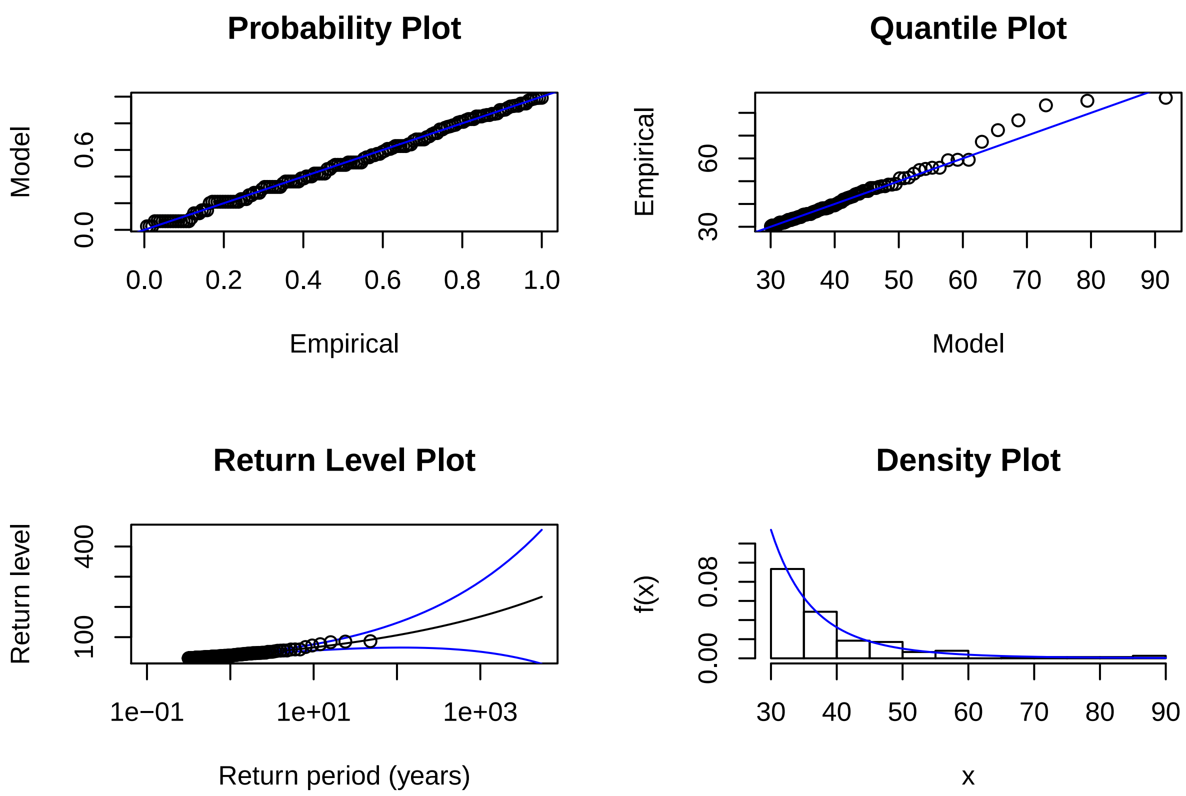
```
## If routine fails, try changing plotting interval
```




```
## $optimal.value
## [1] -485
##
## $optimal.x
## [1] 0.185
##
## $confint.left
## [1] 0.0162
##
## $confint.right
## [1] 0.423
```

Diagnosis

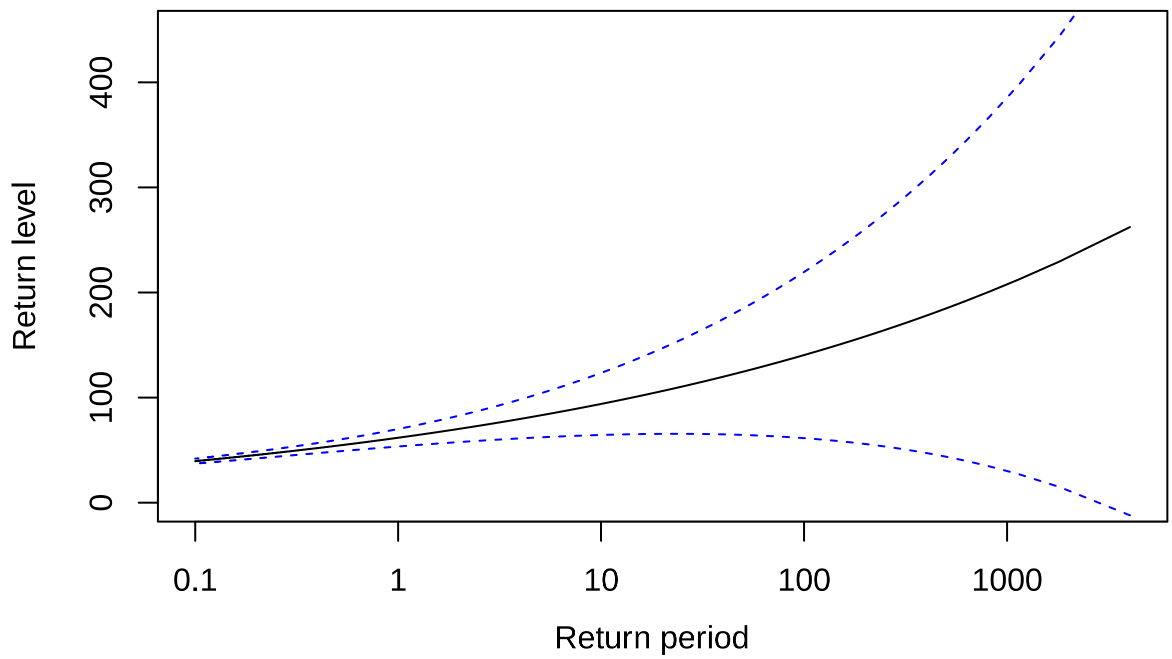
```r
gpd.diag(fitgpd)
```



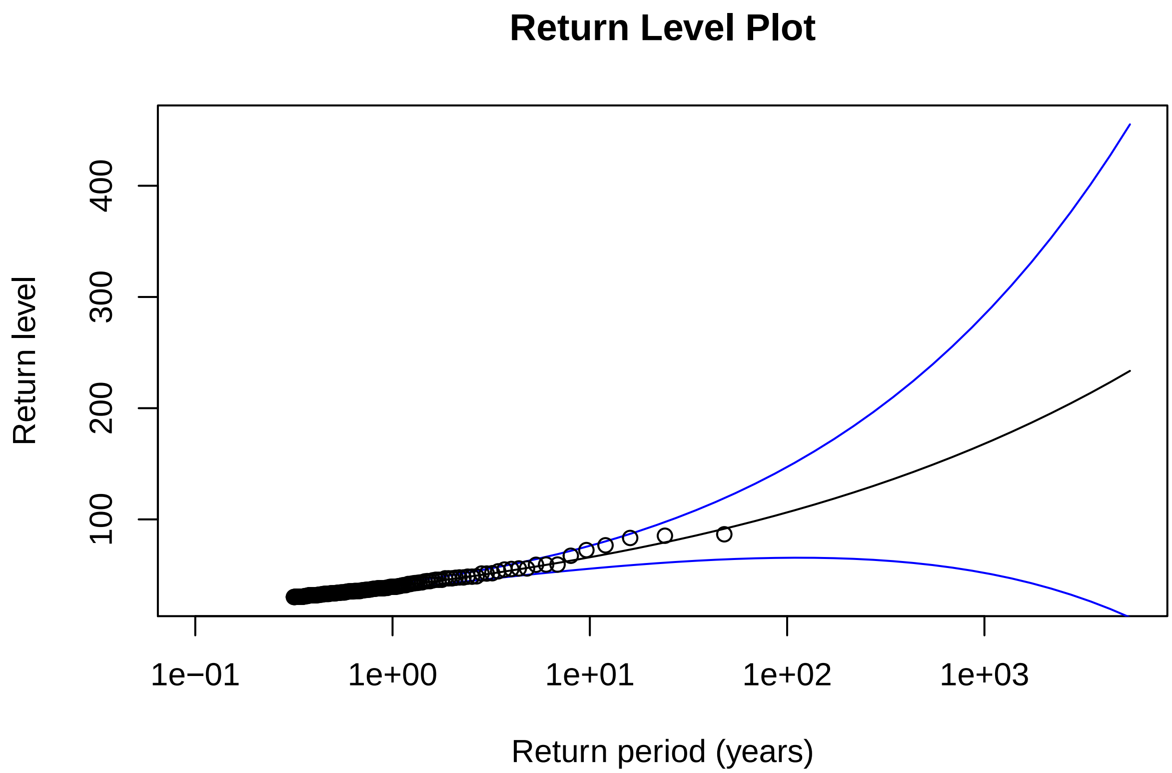
Return level

```
sigma <- fitgpd$mle[1]; xi <- fitgpd$mle[2]

# return level data
p <- seq(0, 1, by = 0.0001)
N <- 1/p
m <- (1/p)*365
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

```r
plot(x=log(N), y=rl.data$return.level, t="l",
     ylim = c(0,450), xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period",side = 1, line=2.5, at=4.5)

lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)
```




```r
gpd.rl(fitgpd$mle, fitgpd$threshold, fitgpd$rate,
       fitgpd$n, fitgpd$npy, fitgpd$cov, fitgpd$data, fitgpd$xdata)
```



100-year return

```
#return level
rl <- rl.data$return.level[rl.data$prob==0.01]

#standard deviation
sd <- sqrt(var.data$var[var.data$prob==0.01])

#CI at 95%
rl-1.96*sd
```

```
## [1] 65.5
```

```
rl+1.96*sd
```

```
## [1] 147
```

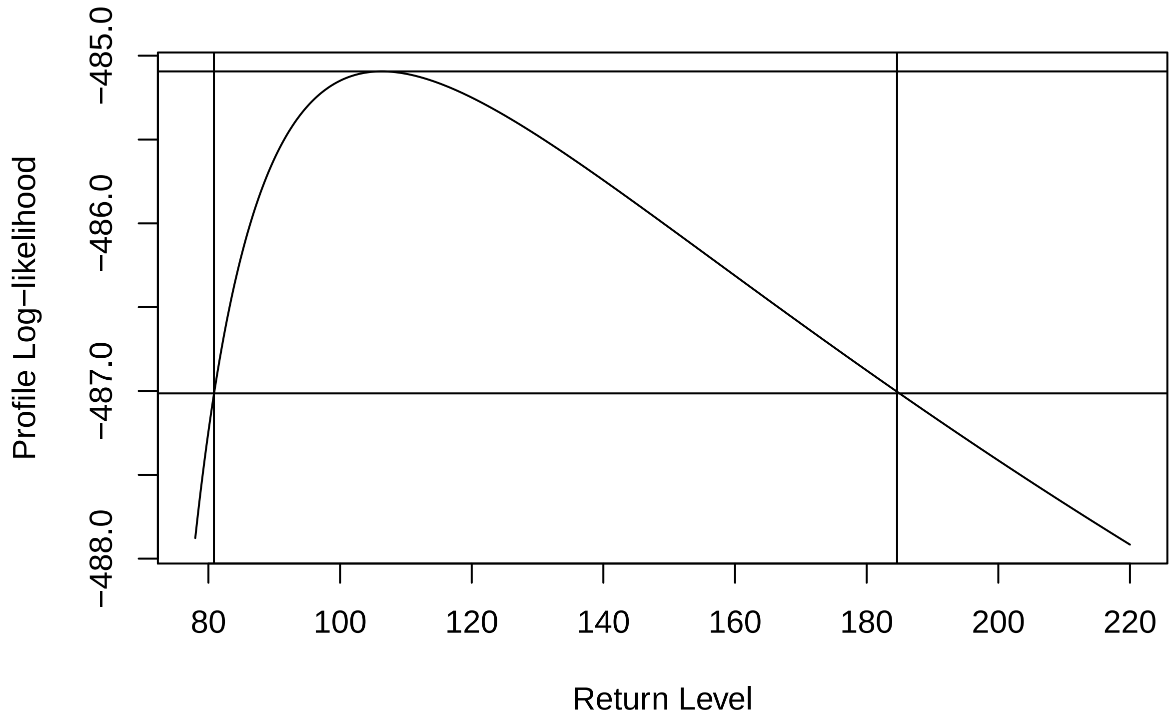
```
length(rain[rain>rl-1.96*sd])
```

```
## [1] 6
```

Priofile likelihood

```
cgpd.prof(fitgpd, m = 100, xlow = 78, xup = 220)
```

```
## If routine fails, try changing plotting interval
```




```
## $optimal.value
## [1] -485
##
## $optimal.x
## [1] 106
##
## $confint.lower
## [1] 80.8
##
## $confint.upper
## [1] 185
```

Daily Rainfall data (using extRemes)

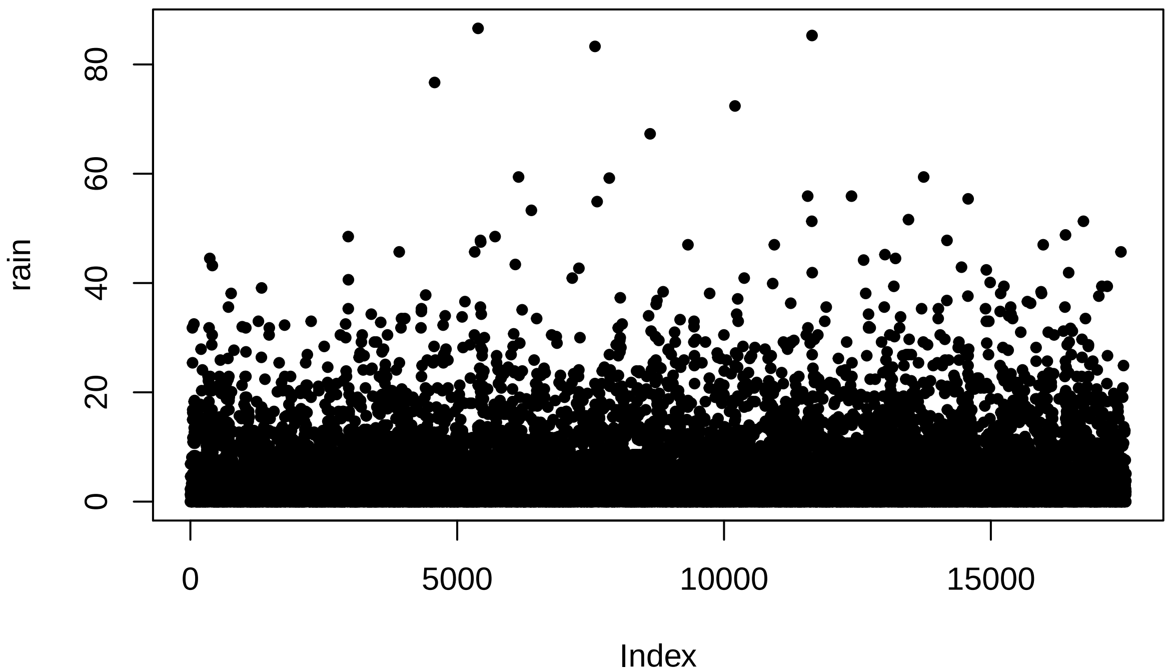
Data

A numeric vector containing daily rainfall accumulations at a location in south-west England over the period 1914 to 1962

```r
library(ismev)
data(rain)
str(rain)
```

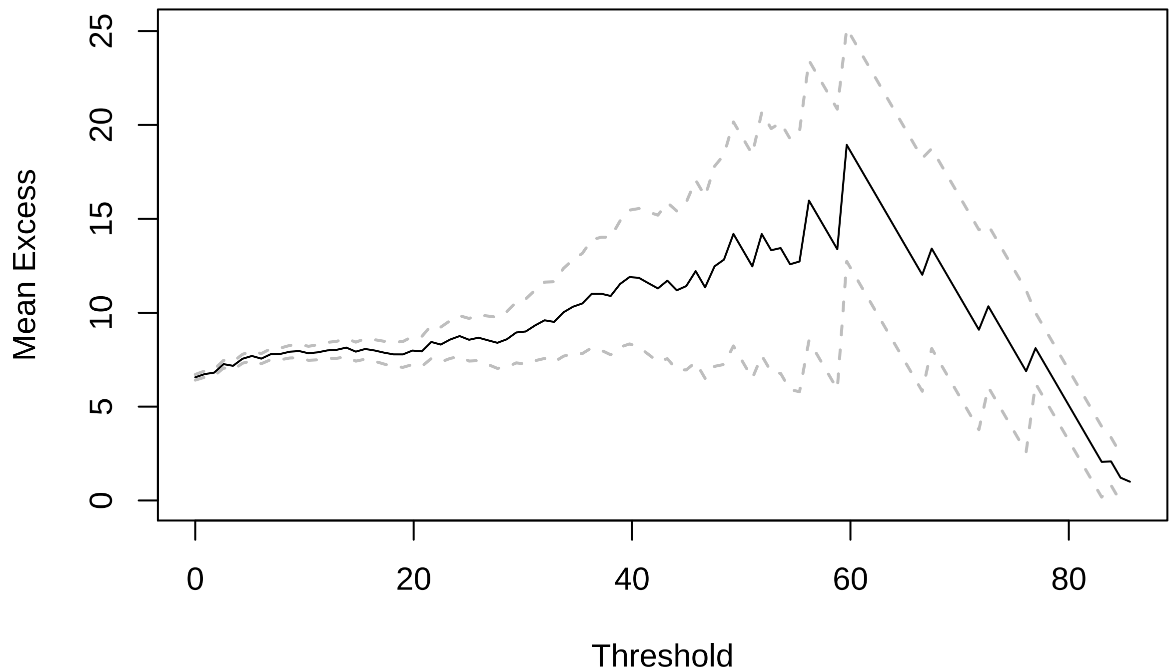
```
##  num [1:17531] 0 2.3 1.3 6.9 4.6 0 1 1.5 1.8 1.8 ...
```

```r
plot(rain, pch=20)
```



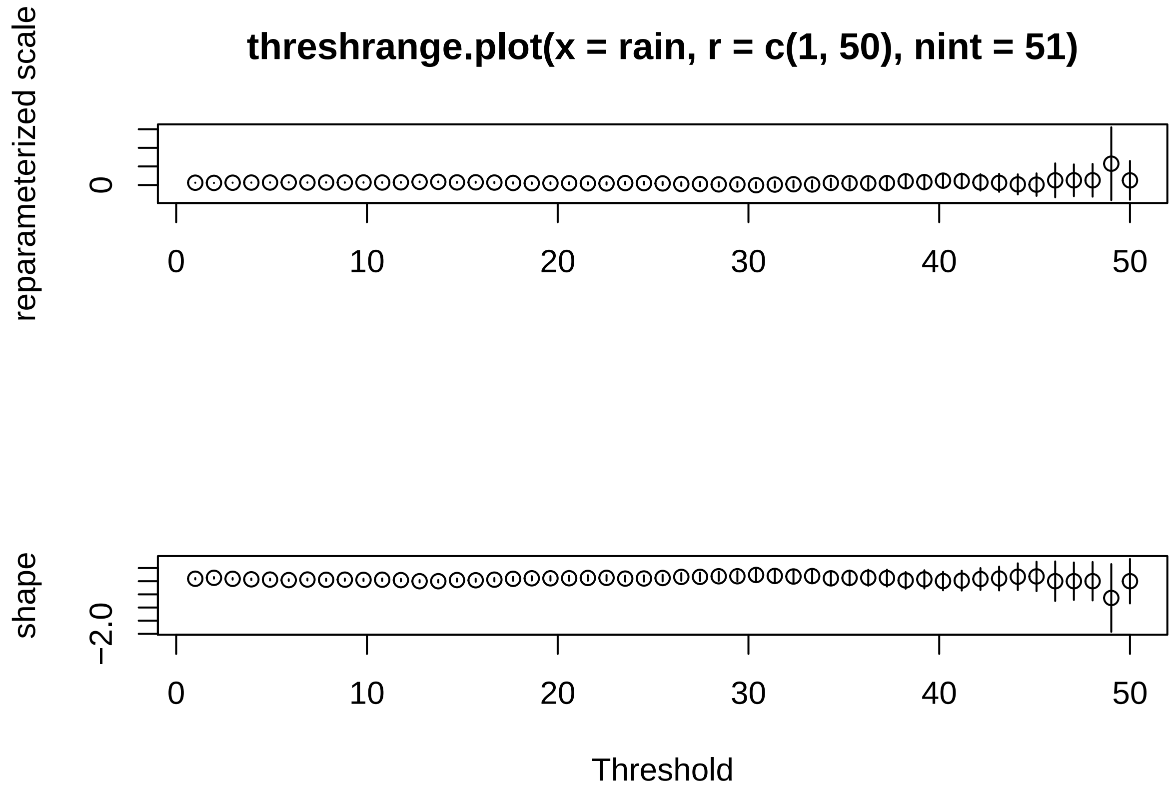
Mean residual life plot

```
library(extRemes)
mrlplot(rain)
```



Fitting the GPD Model Over a Range of Thresholds

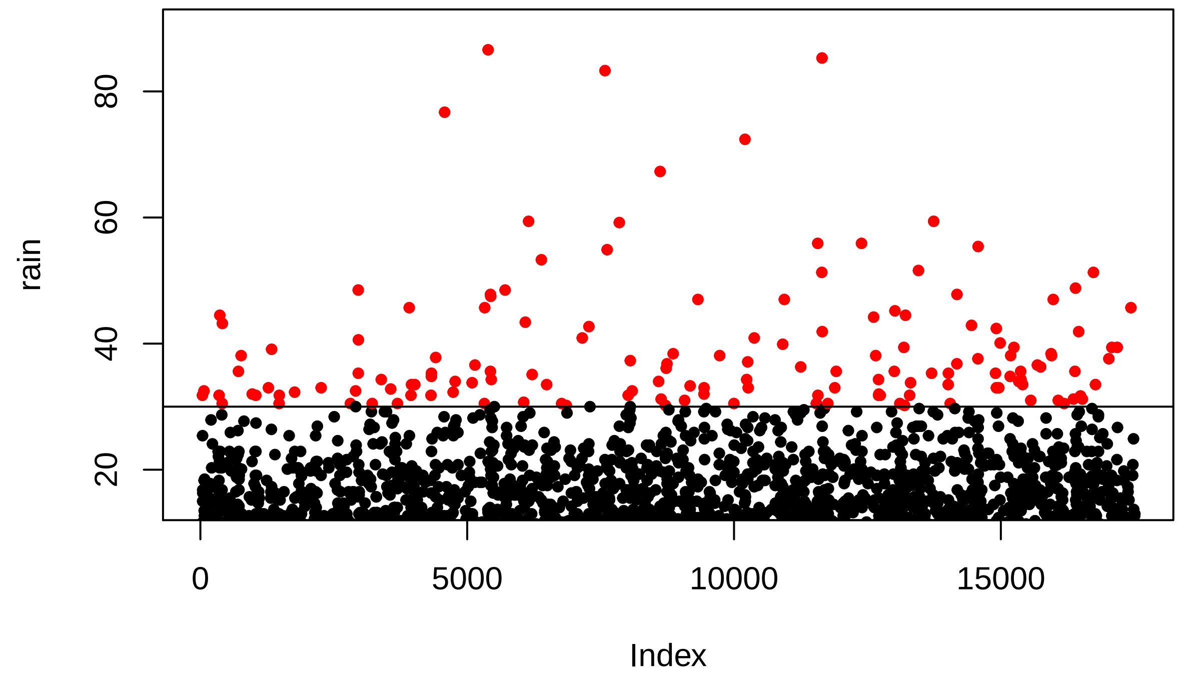
```
threshrange.plot(rain, r = c(1, 50), nint = 51)
```



Threshold

```
u <- 30
```

```
col.exceed <- rain
col.exceed[col.exceed > u] <- "red"
col.exceed[col.exceed <= u] <- "black"

plot(rain, pch=20, ylim=c(15,90), col=col.exceed)
abline(h=u)
```



Fit the GPD model

```
options(digits = 3)
fitgpd <- fevd(rain, threshold = u, span = 365, type = "GP")
fitgpd
```

```
##
## fevd(x = rain, threshold = u, type = "GP", span = 365)
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  485
##
##
##  Estimated parameters:
## scale shape
## 7.440 0.184
##
##  Standard Error Estimates:
## scale shape
## 0.959 0.101
##
##  Estimated parameter covariance matrix.
##         scale    shape
```

```
## scale  0.9188 -0.0655
## shape -0.0655  0.0102
##
##  AIC = 974
##
##  BIC = 980
```

```r
# parameters
par.gdp <- distill.fevd(fitgpd)[c(1,2)]
par.gdp
```

```
## scale shape
## 7.440 0.184
```

```r
# log-likelihood
-distill.fevd(fitgpd)["nllh"]
```

```
## nllh
## -485
```

```r
# covariance
cov.two <- matrix(distill.fevd(fitgpd)[4:7], nrow = 2, byrow = T,
                  dimnames = list(c("scale", "shape"), c("scale", "shape")))
cov.two
```

```
##         scale   shape
## scale  0.9188 -0.0655
## shape -0.0655  0.0102
```

```r
# standard deviation
se <- sqrt(diag(cov.two))
se
```

```
## scale shape
## 0.959 0.101
```

```r
# confidential interval
ci.fevd(fitgpd, type = "parameter")
```

```
## fevd(x = rain, threshold = u, type = "GP", span = 365)
##
## [1] "Normal Approx."
##
##       95% lower CI Estimate 95% upper CI
## scale       5.5616    7.440        9.319
## shape      -0.0139    0.184        0.383
```

```r
# number of complete sample
n <- fitgpd$n; n
```

```
## [1] 17531
```

```r
# number of exceedance
length(rain[rain > u])
```

```
## [1] 152
```

```r
# zeta(exceedance rate)
zeta <- fitgpd$rate; zeta
```

```
## [1] 0.00867
```

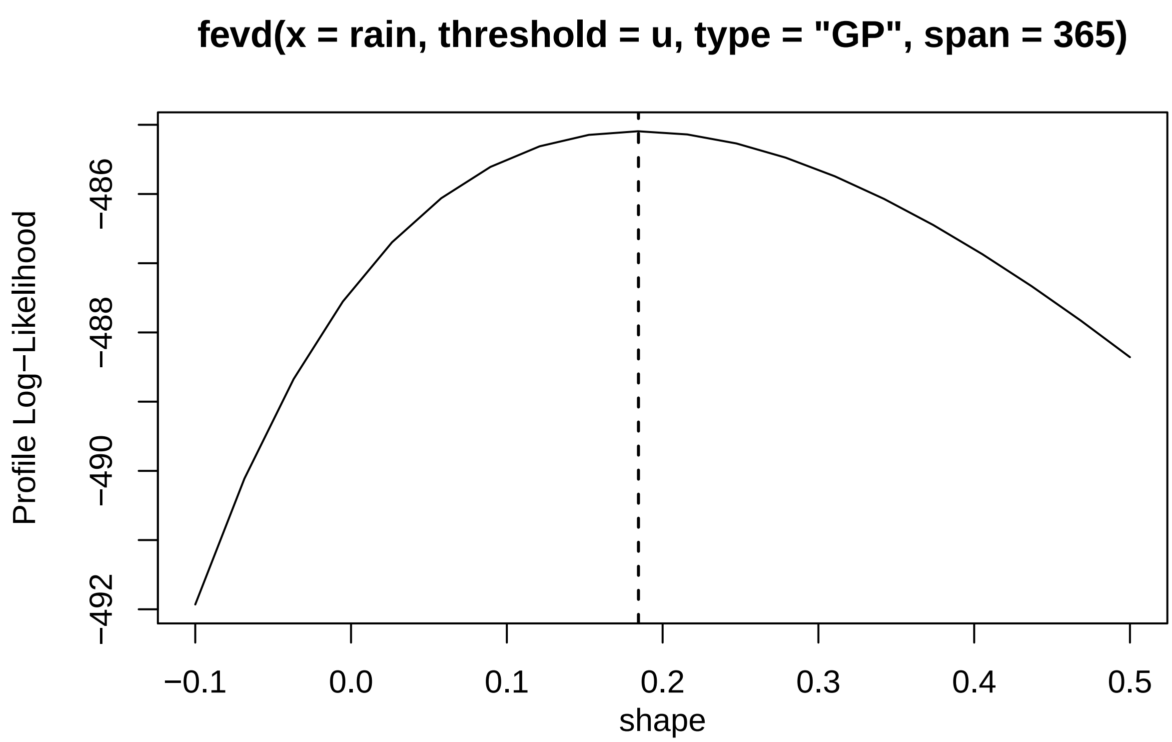
```r
# var of zeta
var.zeta <- zeta*(1-zeta)/n; var.zeta
```

```
## [1] 4.9e-07
```

```r
# complete covariance
cov <- matrix(c(var.zeta, 0, 0, 0, cov.two[1,], 0, cov.two[2,]),
       nrow = 3, byrow = T, dimnames = list(c("zeta","sigma","xi"),
                                            c("zeta","sigma","xi")))
cov
```

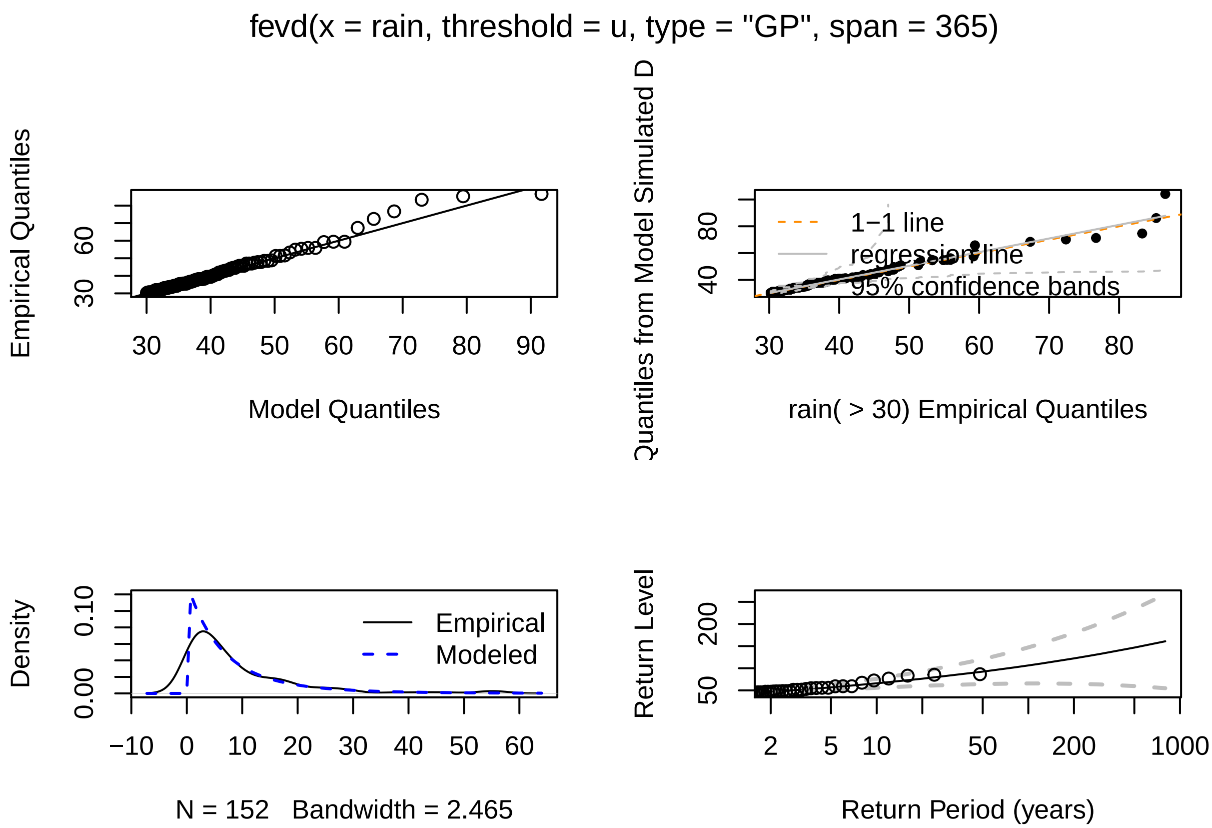
```
##          zeta   sigma      xi
## zeta  4.9e-07  0.0000  0.0000
## sigma 0.0e+00  0.9188 -0.0655
## xi    0.0e+00 -0.0655  0.0102
```

```r
# Profile likelihood for shape
profliker(fitgpd, type = "parameter", which.par = 2, xrange = c(-0.1, 0.5))
```



Diagnosis

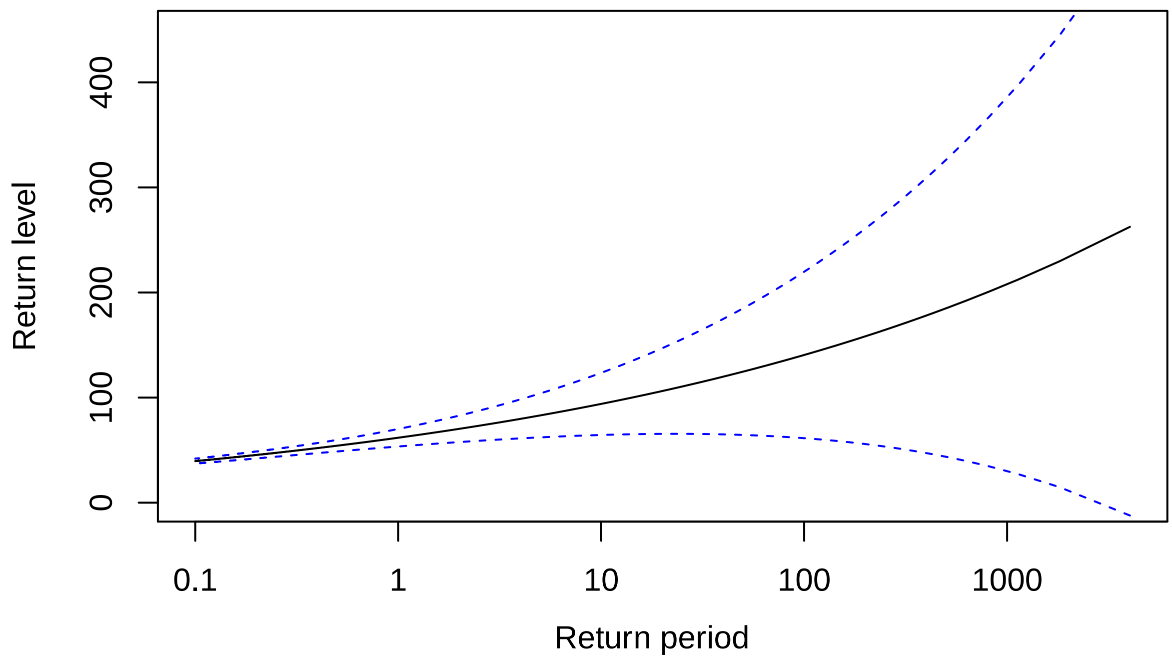
```r
plot(fitgpd)
```



Return level

```
sigma <- par.gdp[1]; xi <- par.gdp[2]

# return level data
p <- seq(0, 1, by = 0.0001)
N <- 1/p
m <- (1/p)*365
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

```r
plot(x=log(N), y=rl.data$return.level, t="l",
     ylim = c(0,450), xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period",side = 1, line=2.5, at=4.5)

lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)
```



100-year return

```r
#return level
rl <- rl.data$return.level[rl.data$prob==0.01]

#standard deviation
sd <- sqrt(var.data$var[var.data$prob==0.01])

#CI at 95%
rl-1.96*sd
```

```
## [1] 65.5
```

```r
rl+1.96*sd
```

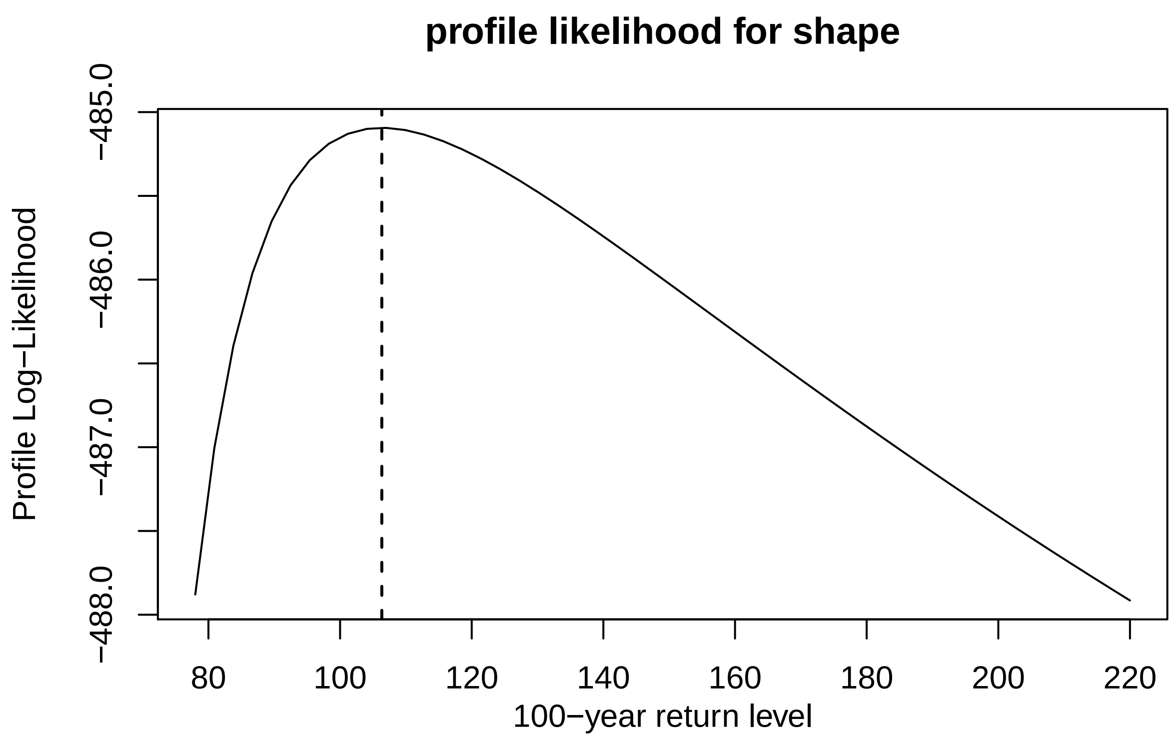
```
## [1] 147
```

```
length(rain[rain>rl-1.96*sd])
```

```
## [1] 6
```

Priofile likelihood

```
profliker(fitgpd, return.period = 100, xrange = c(78,220), nint = 50,
          main="profile likelihood for shape")
```



Dow Jones Index Series (using ismev)

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

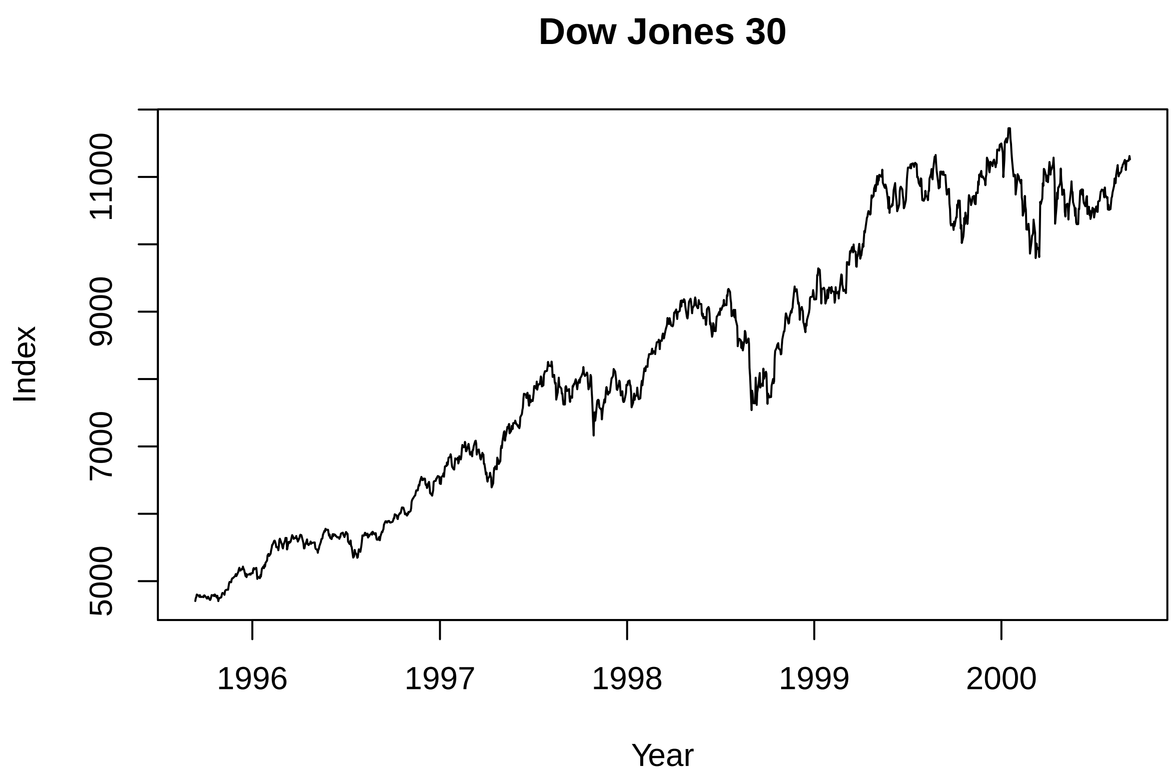
```r
library(lubridate)
```

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

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

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

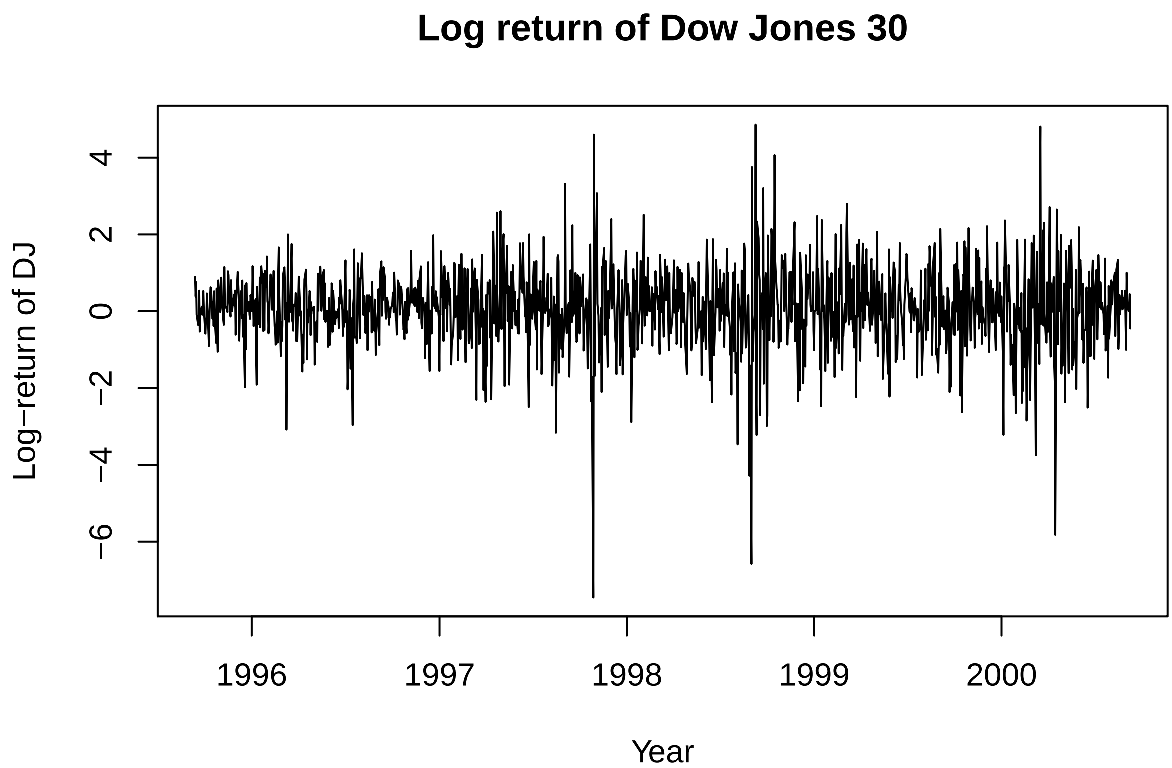
Price

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



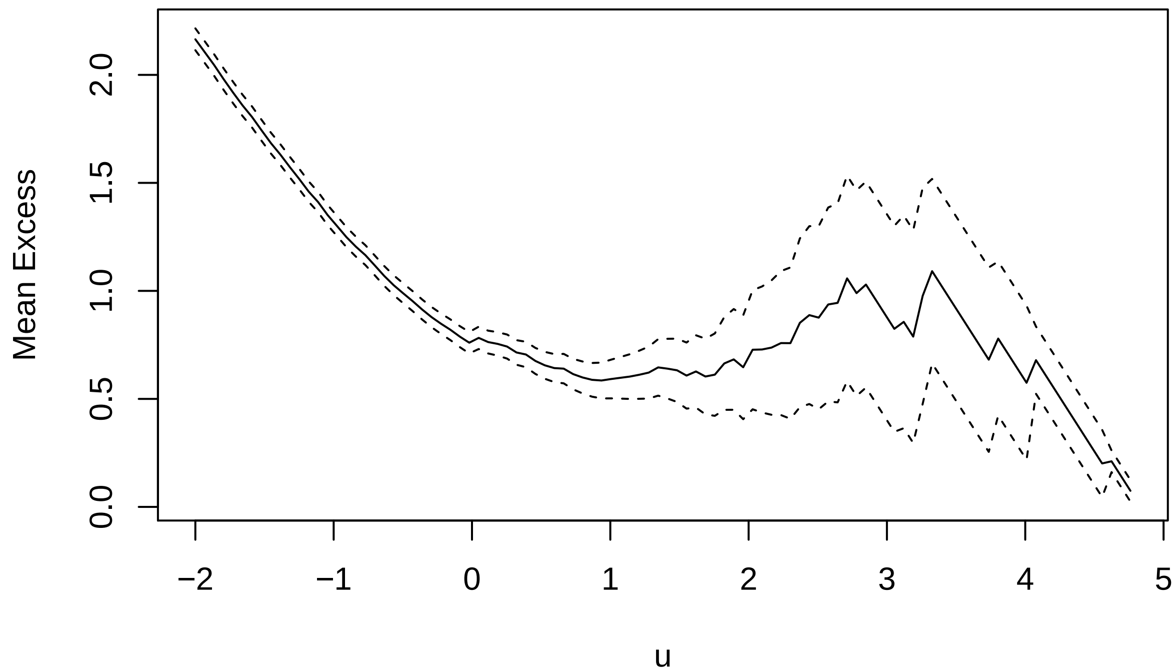
Log-return

```r
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")
```



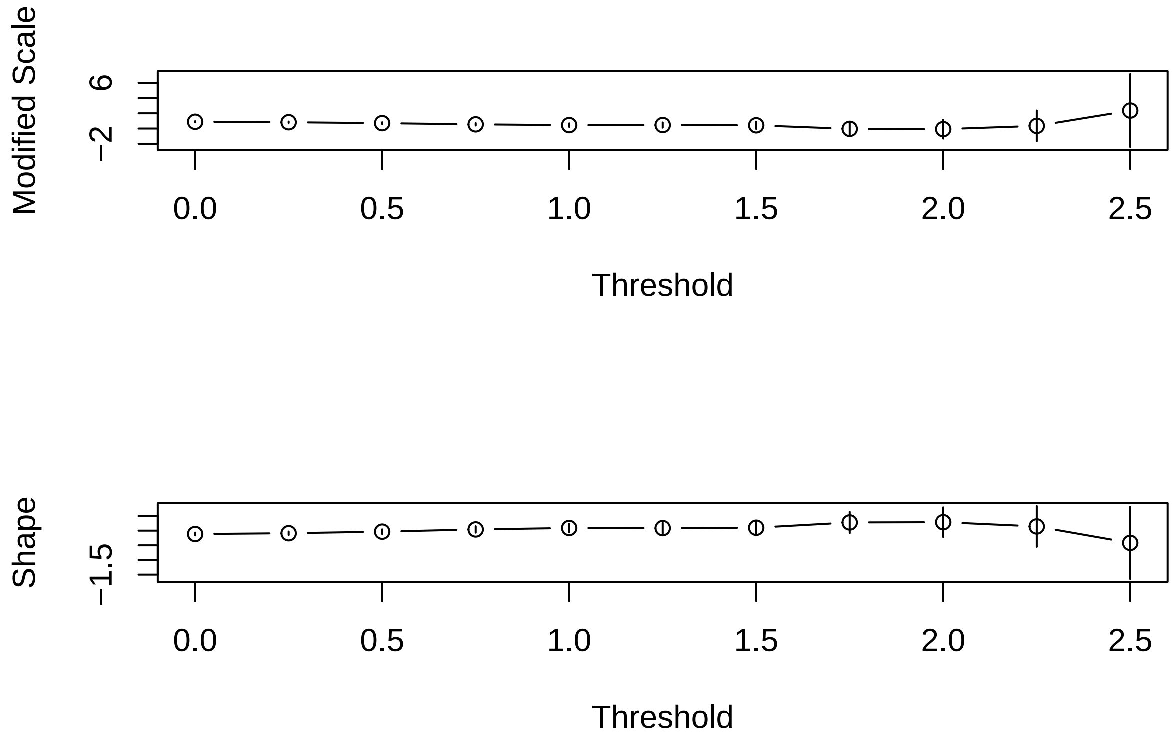
Mean residual life plot

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



Fitting the GPD Model Over a Range of Thresholds

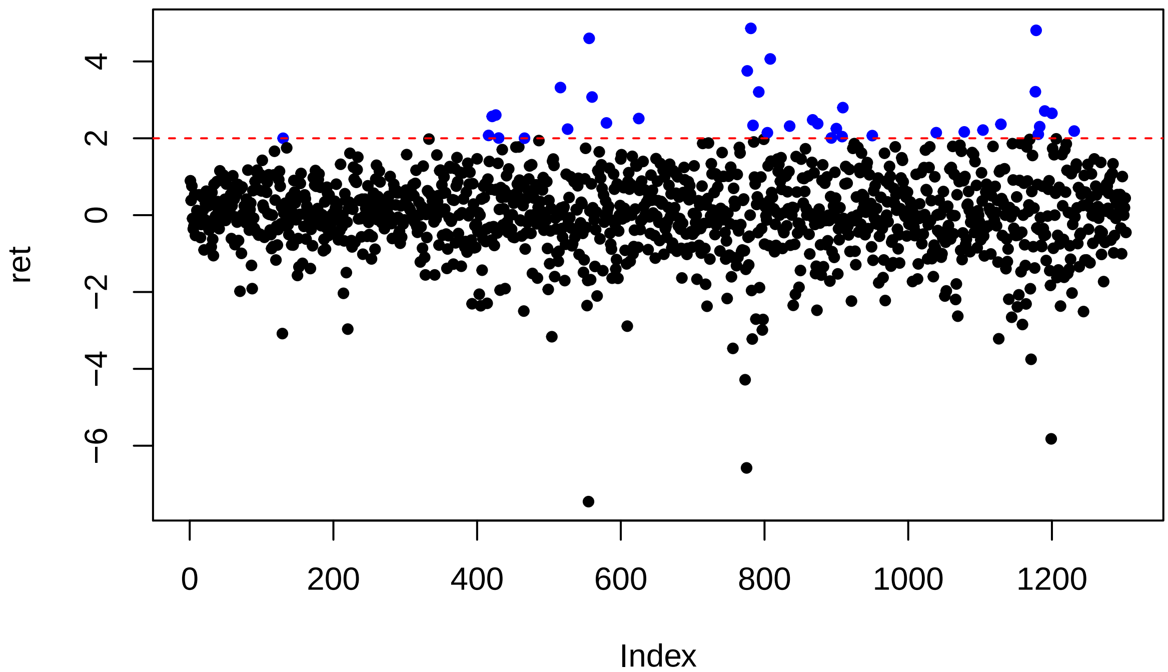
```
gpd.fitrange(ret, umin = 0, umax = 2.5, nint = 11)
```



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

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

```
## $threshold
## [1] 2
##
## $nexc
## [1] 37
##
## $conv
## [1] 0
##
## $nllh
## [1] 21.6
##
## $mle
## [1] 0.495 0.288
##
## $rate
## [1] 0.0284
##
## $se
## [1] 0.150 0.258
```

```r
options(digits = 3)
# parameters
fitgpd$mle
```

```
## [1] 0.495 0.288
```

```r
# log-likelihood
-fitgpd$nllh
```

```
## [1] -21.6
```

```r
# covariance
cov.two <- fitgpd$cov; cov.two
```

```
##          [,1]    [,2]
## [1,]  0.0224 -0.0280
## [2,] -0.0280  0.0665
```

```r
# standard deviation
fitgpd$se
```

```
## [1] 0.150 0.258
```

```r
# confidential interval
ci <- cbind(fitgpd$mle-1.96*fitgpd$se, fitgpd$mle+1.96*fitgpd$se)
colnames(ci) <- c("lower", "upper")
rownames(ci) <- c("sigma", "xi")
ci
```

```
##        lower upper
## sigma  0.202 0.788
## xi    -0.218 0.793
```

```r
# number of complete sample
n <- fitgpd$n; n
```

```
## [1] 1303
```

```r
# number of exceedance
fitgpd$nexc
```

```
## [1] 37
```

```r
# zeta(exceedance rate)
zeta <- fitgpd$rate; zeta
```

```
## [1] 0.0284
```

```r
# var of zeta
var.zeta <- zeta*(1-zeta)/n; var.zeta
```

```
## [1] 2.12e-05
```

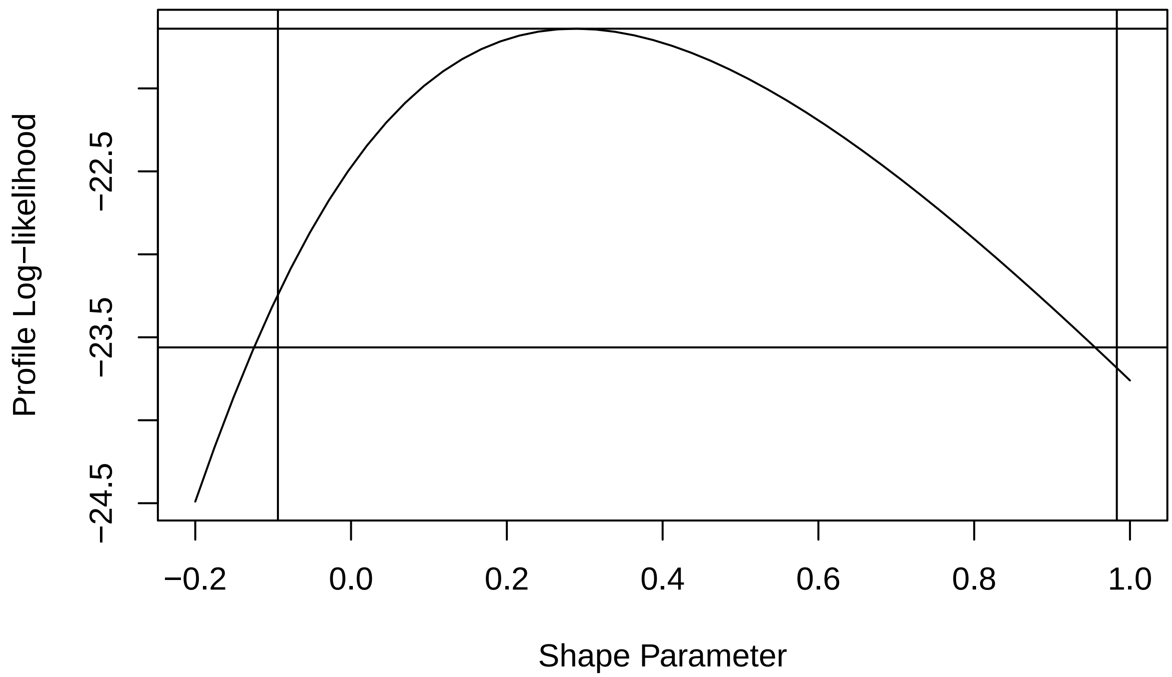
```r
# complete covariance
cov <- matrix(c(var.zeta, 0, 0, 0, cov.two[1,], 0, cov.two[2,]),
       nrow = 3, byrow = T, dimnames = list(c("zeta","sigma","xi"),
                                            c("zeta","sigma","xi")))
cov
```

```
##           zeta   sigma      xi
## zeta  2.12e-05  0.0000  0.0000
```

```
## sigma 0.00e+00  0.0224 -0.0280
## xi    0.00e+00 -0.0280  0.0665
```

```r
# Profile likelihood for shape
cgpd.profxi(fitgpd, xlow = -0.2, xup = 1, nint = 50)
```

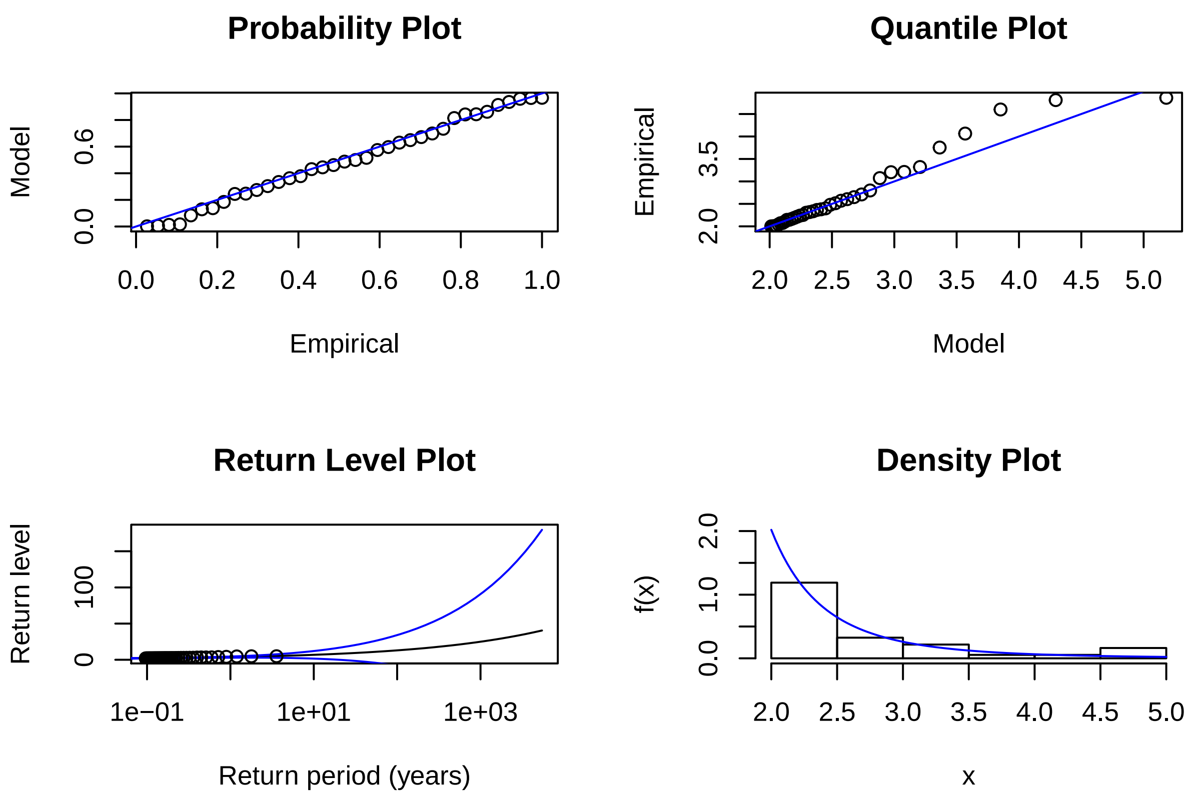
```
## If routine fails, try changing plotting interval
```




```
## $optimal.value
## [1] -21.6
##
## $optimal.x
## [1] 0.29
##
## $confint.left
## [1] -0.0939
##
## $confint.right
## [1] 0.983
```

Diagnosis

```r
gpd.diag(fitgpd)
```



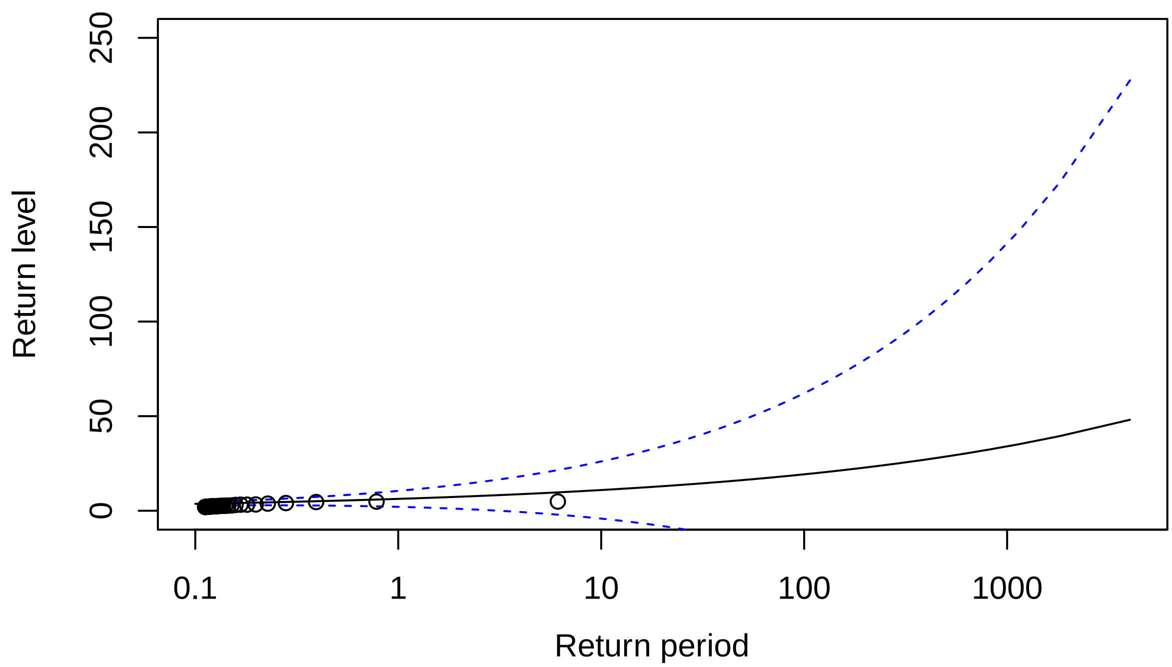

```
gpd.rl
```

```
## function (a, u, la, n, npy, mat, dat, xdat)
## {
##     a <- c(la, a)
##     eps <- 1e-06
##     a1 <- a
##     a2 <- a
##     a3 <- a
##     a1[1] <- a[1] + eps
##     a2[2] <- a[2] + eps
##     a3[3] <- a[3] + eps
##     jj <- seq(-1, 3.75 + log10(npy), by = 0.1)
##     m <- c(1/la, 10^jj)
##     q <- gpdq2(a[2:3], u, la, m)
##     d <- t(gpd.rl.gradient(a = a, m = m))
##     mat <- matrix(c((la * (1 - la))/n, 0, 0, 0, mat[1, 1], mat[1,
##         2], 0, mat[2, 1], mat[2, 2]), ncol = 3)
##     v <- apply(d, 1, q.form, m = mat)
##     plot(m/npy, q, log = "x", type = "n", xlim = c(0.1, max(m)/npy),
##         ylim = c(u, max(xdat, q[q > u - 1] + 1.96 * sqrt(v)[q >
##             u - 1])), xlab = "Return period (years)", ylab = "Return level",
##         main = "Return Level Plot")
##     lines(m[q > u - 1]/npy, q[q > u - 1])
##     lines(m[q > u - 1]/npy, q[q > u - 1] + 1.96 * sqrt(v)[q >
##         u - 1], col = 4)
```

```
##     lines(m[q > u - 1]/npy, q[q > u - 1] - 1.96 * sqrt(v)[q >
##         u - 1], col = 4)
##     nl <- n - length(dat) + 1
##     sdat <- sort(xdat)
##     points((1/(1 - (1:n)/(n + 1))/npy)[sdat > u], sdat[sdat >
##         u])
## }
## <bytecode: 0x000000002eac2878>
## <environment: namespace:ismev>
```

```
sigma <- fitgpd$mle[1]; xi <- fitgpd$mle[2]

# return level data
p <- seq(0, 1, by = 0.0001)
N <- 1/p
m <- (1/p)*365
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

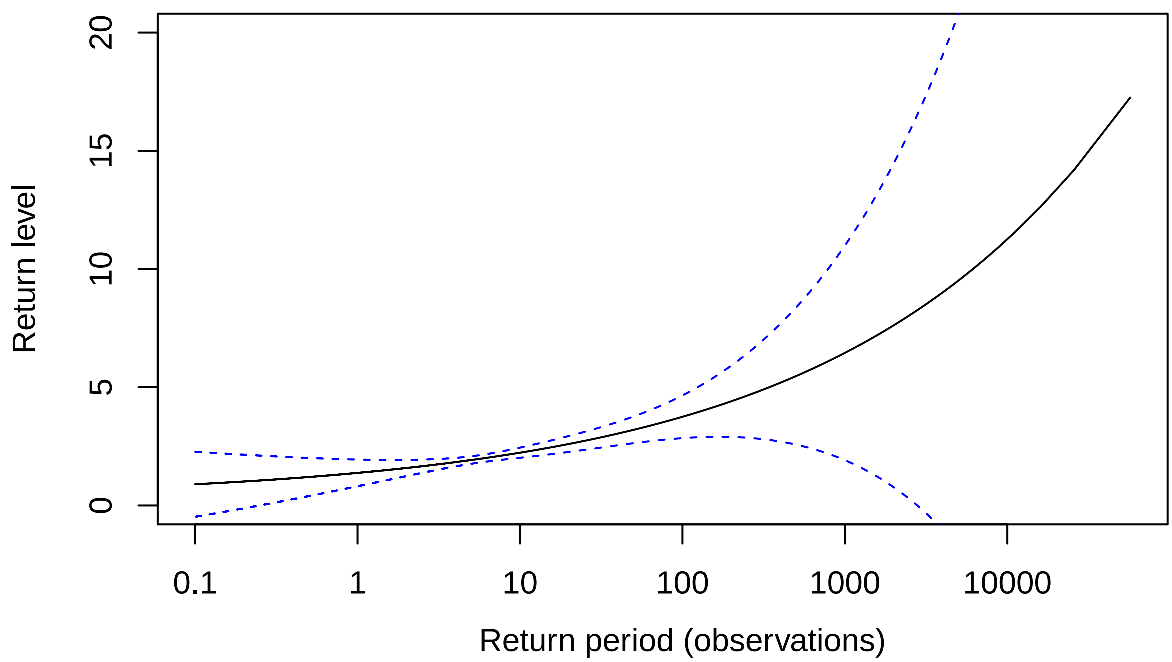
```
plot(x=log(N), y=rl.data$return.level, t="l",
     ylim = c(0,250), xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period",side = 1, line=2.5, at=4.5)

lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)

sdat <- sort(ret)
points((1/(1 - (1:n)/(n + 1))/365)[sdat > u], sdat[sdat > u])
```




```
sigma <- fitgpd$mle[1]; xi <- fitgpd$mle[2]

# return level data
p <- seq(0, 1, by = 0.00001)
N <- 1/p
m <- (1/p)
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

```
plot(x=log(N), y=rl.data$return.level, t="l", xaxt='n',
     ylim = c(0,20), ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period (observations)",side = 1, line=2.5, at=6)
```

```r
lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)
```



Dow Jones Index Series (using extRemes)

Data

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

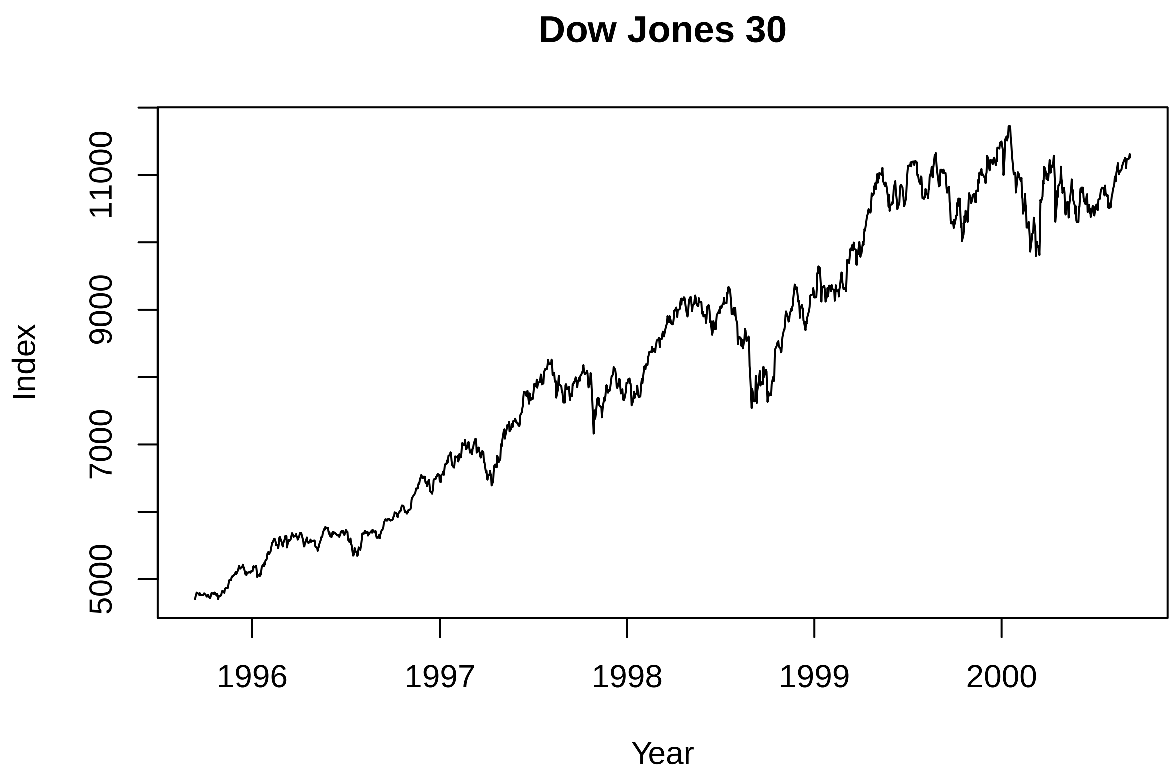
```
## 'data.frame':    1304 obs. of  2 variables:
##  $ Date : POSIXt, format: "1995-09-11 09:00:00" "1995-09-12 09:00:00" ...
##  $ Index: num  4705 4747 4766 4802 4798 ...
```

Date

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

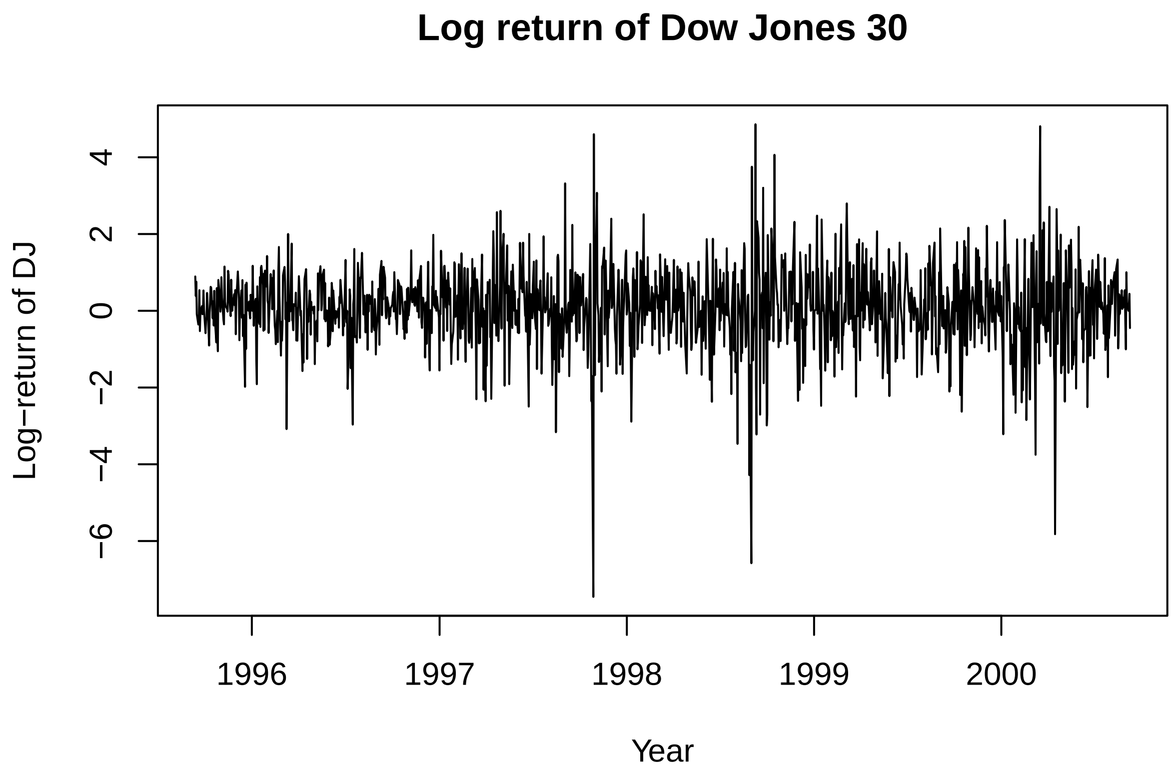
Price

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



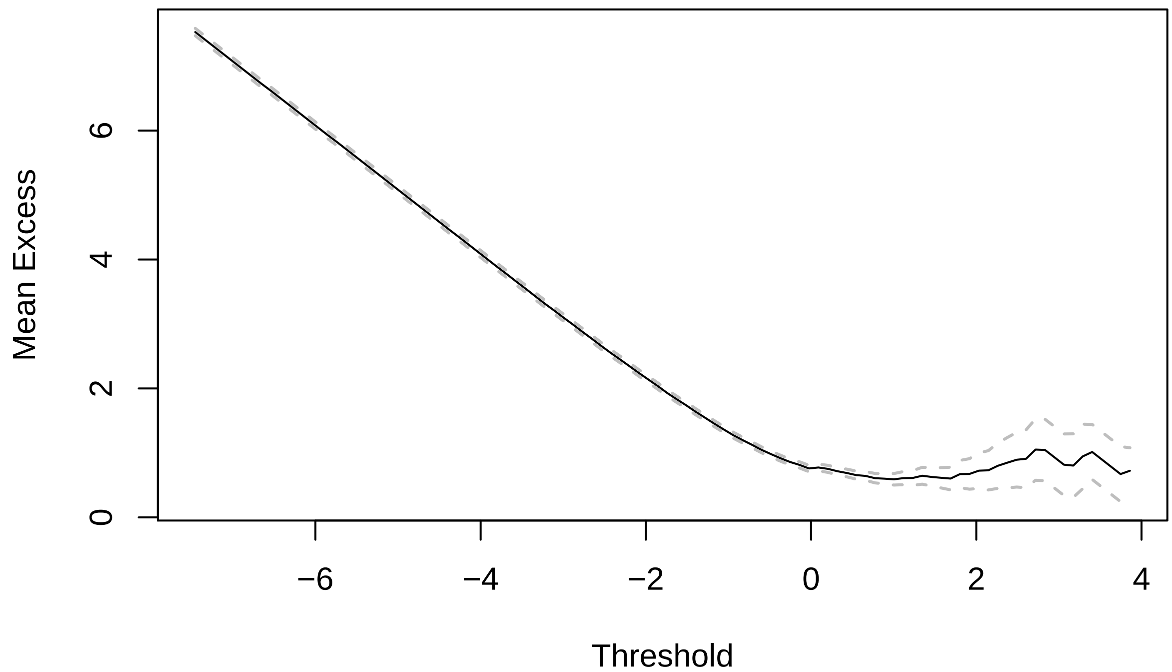
Log-return

```
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")
```



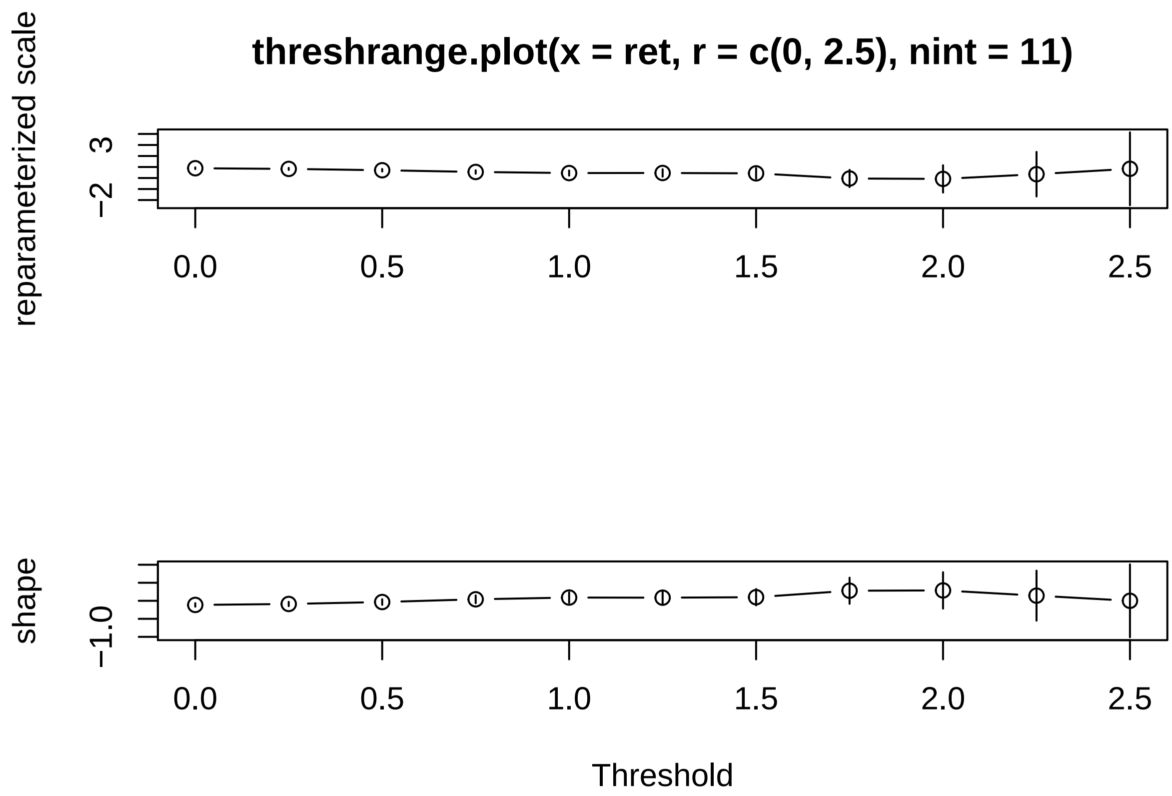
Mean residual life plot

```
mrlplot(ret)
```



Fitting the GPD Model Over a Range of Thresholds

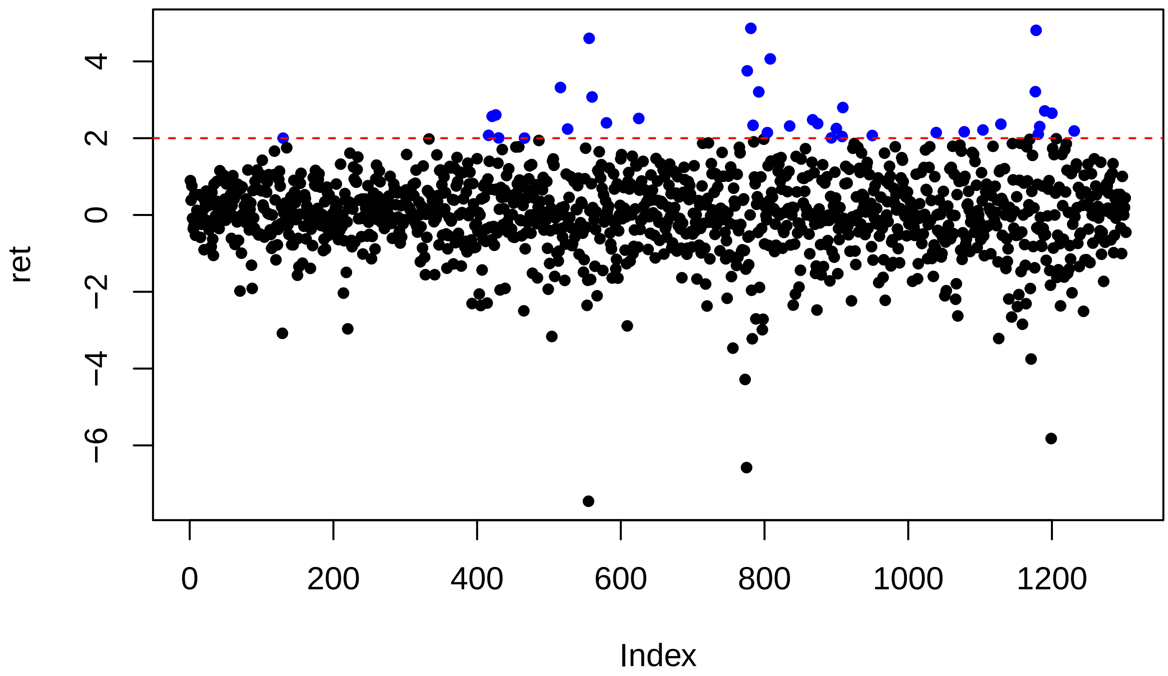
```
threshrange.plot(ret, r = c(0,2.5), nint = 11)
```



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

```
options(digits = 3)
fitgpd <- fevd(ret, threshold = u, type = "GP", span = 365)
fitgpd
```

```
##
## fevd(x = ret, threshold = u, type = "GP", span = 365)
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  21.6
##
##
##  Estimated parameters:
## scale shape
## 0.495 0.288
##
##  Standard Error Estimates:
## scale shape
## 0.150 0.258
##
##  Estimated parameter covariance matrix.
##         scale   shape
```

```
## scale  0.0224 -0.0279
## shape -0.0279  0.0665
##
##  AIC = 47.3
##
##  BIC = 50.5
```

```r
# parameters
par.gdp <- distill.fevd(fitgpd)[c(1,2)]
par.gdp
```

```
## scale shape
## 0.495 0.288
```

```r
# log-likelihood
-distill.fevd(fitgpd)["nllh"]
```

```
##  nllh
## -21.6
```

```r
# covariance
cov.two <- matrix(distill.fevd(fitgpd)[4:7], nrow = 2, byrow = T,
                  dimnames = list(c("scale", "shape"), c("scale", "shape")))
cov.two
```

```
##         scale   shape
## scale  0.0224 -0.0279
## shape -0.0279  0.0665
```

```r
# standard deviation
se <- sqrt(diag(cov.two))
se
```

```
## scale shape
## 0.150 0.258
```

```r
# confidential interval
ci.fevd(fitgpd, type = "parameter")
```

```
## fevd(x = ret, threshold = u, type = "GP", span = 365)
##
## [1] "Normal Approx."
##
##       95% lower CI Estimate 95% upper CI
## scale        0.202    0.495        0.788
## shape       -0.217    0.288        0.793
```

```r
# number of complete sample
n <- fitgpd$n; n
```

```
## [1] 1303
```

```r
# number of exceedance
length(ret[ret > u])
```

```
## [1] 37
```

```r
# zeta(exceedance rate)
zeta <- fitgpd$rate; zeta
```

```
## [1] 0.0284
```

```r
# var of zeta
var.zeta <- zeta*(1-zeta)/n; var.zeta
```

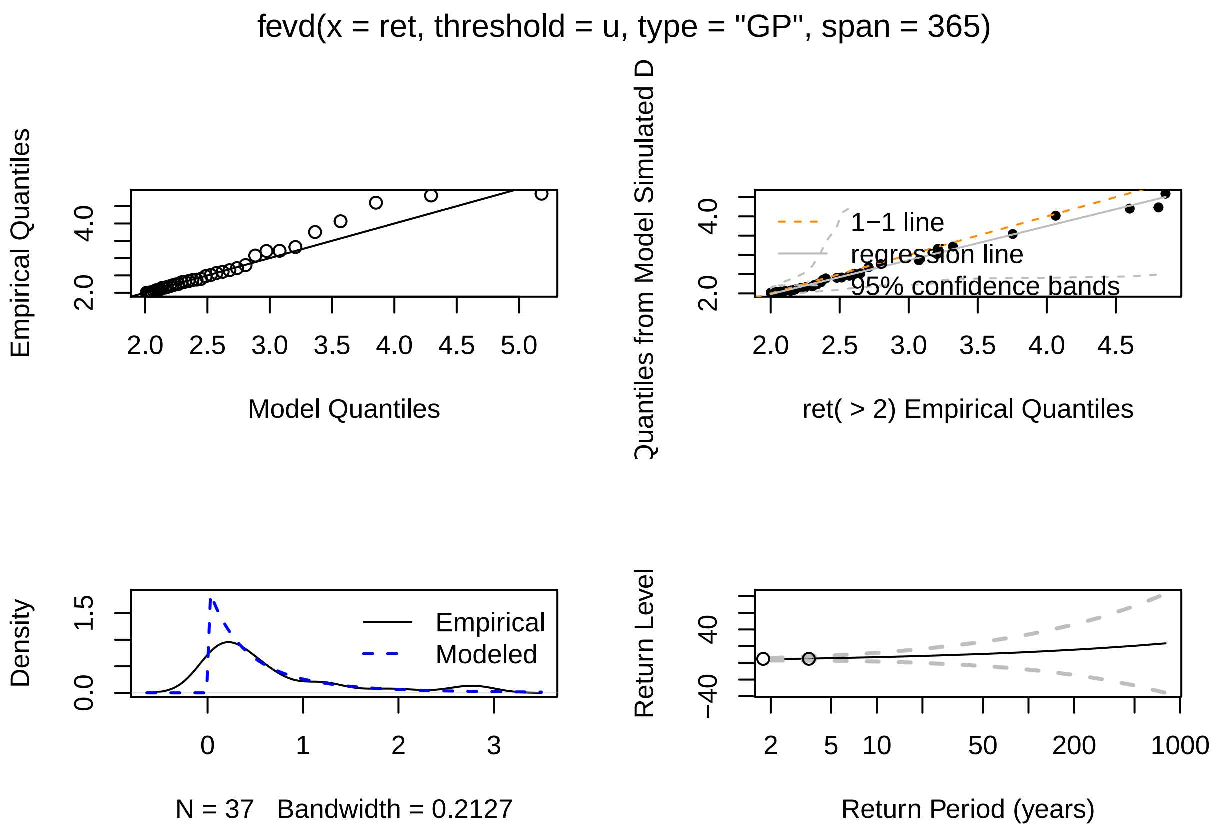
```
## [1] 2.12e-05
```

```r
# complete covariance
cov <- matrix(c(var.zeta, 0, 0, 0, cov.two[1,], 0, cov.two[2,]),
       nrow = 3, byrow = T, dimnames = list(c("zeta","sigma","xi"),
                                            c("zeta","sigma","xi")))
cov
```

```
##           zeta   sigma      xi
## zeta  2.12e-05  0.0000  0.0000
## sigma 0.00e+00  0.0224 -0.0279
## xi    0.00e+00 -0.0279  0.0665
```

Diagnosis

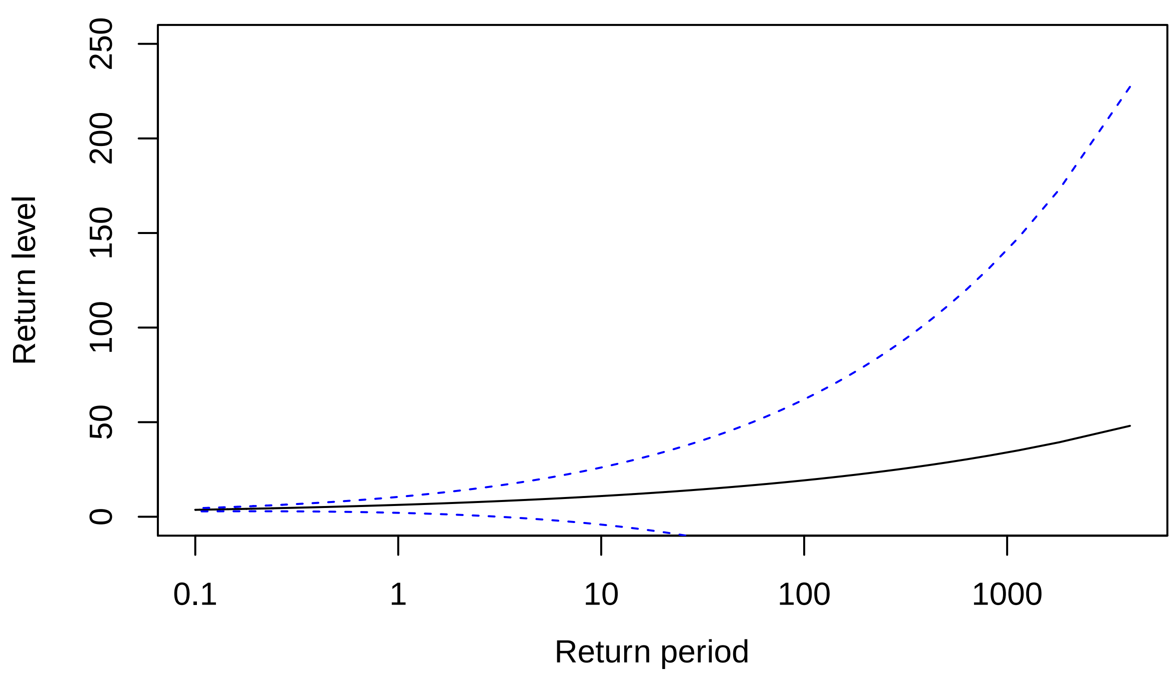
```r
plot(fitgpd)
```




```r
sigma <- par.gdp[1]; xi <- par.gdp[2]

# return level data
p <- seq(0, 1, by = 0.0001)
N <- 1/p
m <- (1/p)*365
```

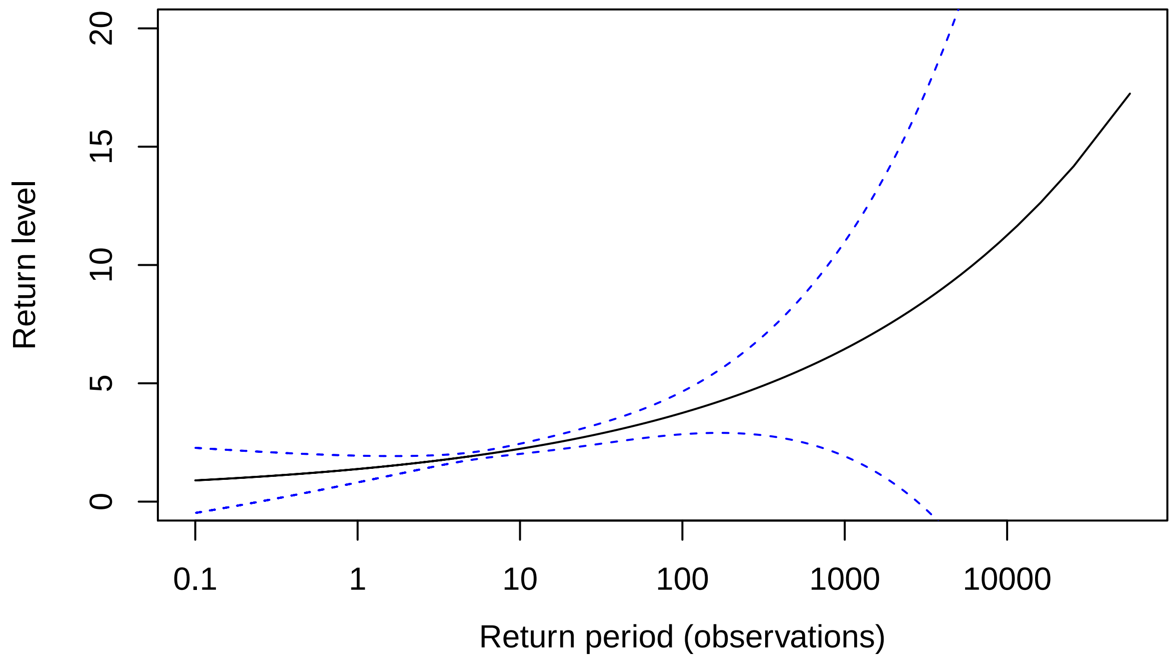
```r
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

```r
plot(x=log(N), y=rl.data$return.level, t="l",
     ylim = c(0,250), xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period",side = 1, line=2.5, at=4.5)

lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)
```



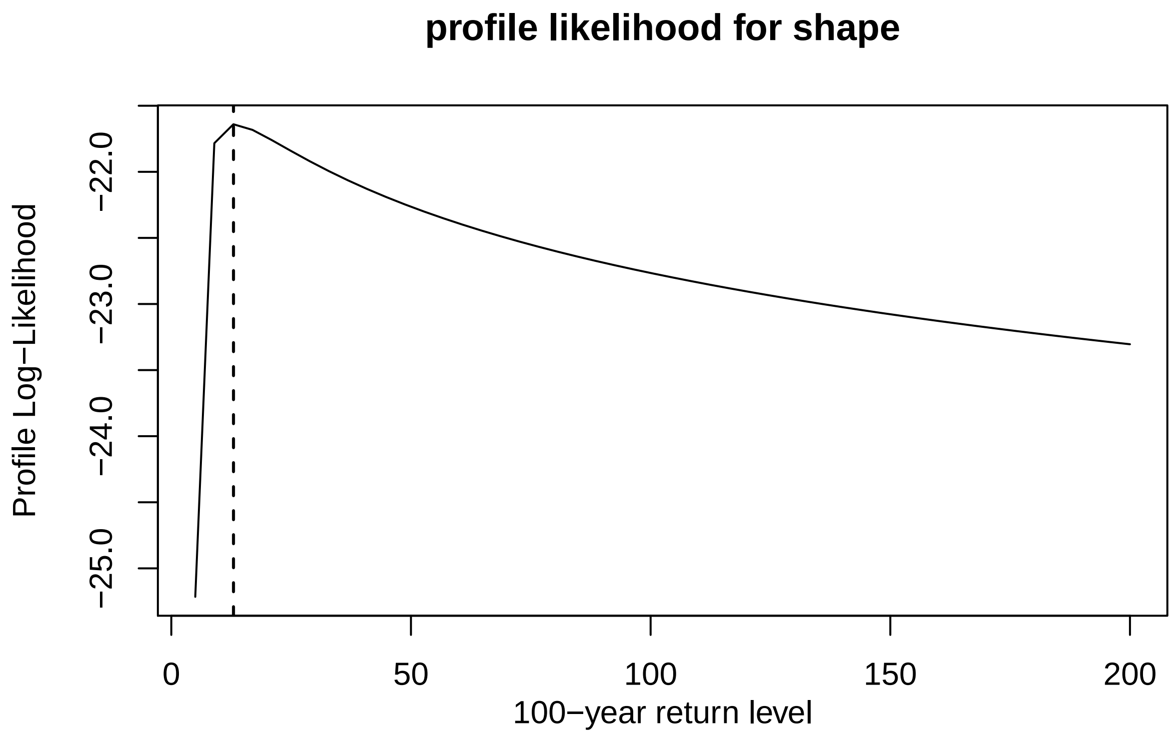
```r
sigma <- par.gdp[1]; xi <- par.gdp[2]

# return level data
p <- seq(0, 1, by = 0.00001)
N <- 1/p
m <- (1/p)
rl <- u + (sigma/xi)*((m*zeta)^(xi)-1)
rl.data <- data.frame(prob=p, m=m, return.level=rl)

# Variance data
var <- matrix(nrow = length(m), ncol = 1)
for(i in 1:length(m)){
  dz1 <- sigma*m[i]^(xi)*zeta^(xi-1)
  dz2 <- xi^(-1)*((m[i]*zeta)^(xi)-1)
  dz3 <- -sigma*xi^(-2)*((m[i]*zeta)^(xi)-1)+
    sigma*xi^(-1)*((m[i]*zeta)^(xi)*log(m[i]*zeta))
  dz <- matrix(c(dz1,dz2,dz3), nrow=3, byrow=T)
  each.var <- t(dz) %*% cov %*% dz
  var[i] <- each.var
}
var.data <- data.frame(prob=p, m=m, var=var)
```

```r
plot(x=log(N), y=rl.data$return.level, t="l", xaxt='n',
     ylim = c(0,20), ylab="Return level", xlab="")
axis(side = 1, at = c(0,2,4,6,8,10), labels= c(0.1,1,10,100,1000,10000))
mtext("Return period (observations)",side = 1, line=2.5, at=6)

lower.line <- rl.data$return.level-1.96*sqrt(var.data$var)
upper.line <- rl.data$return.level+1.96*sqrt(var.data$var)

lines(x=log(N), y=lower.line, col="blue", lty=2)
lines(x=log(N), y=upper.line, col="blue", lty=2)
```



Profile likelihood for return level

```
# profile likelihood for 100 years
profliker(fitgpd, return.period = 100,
          xrange = c(5,200), nint = 50,
          main="profile likelihood for shape")
```

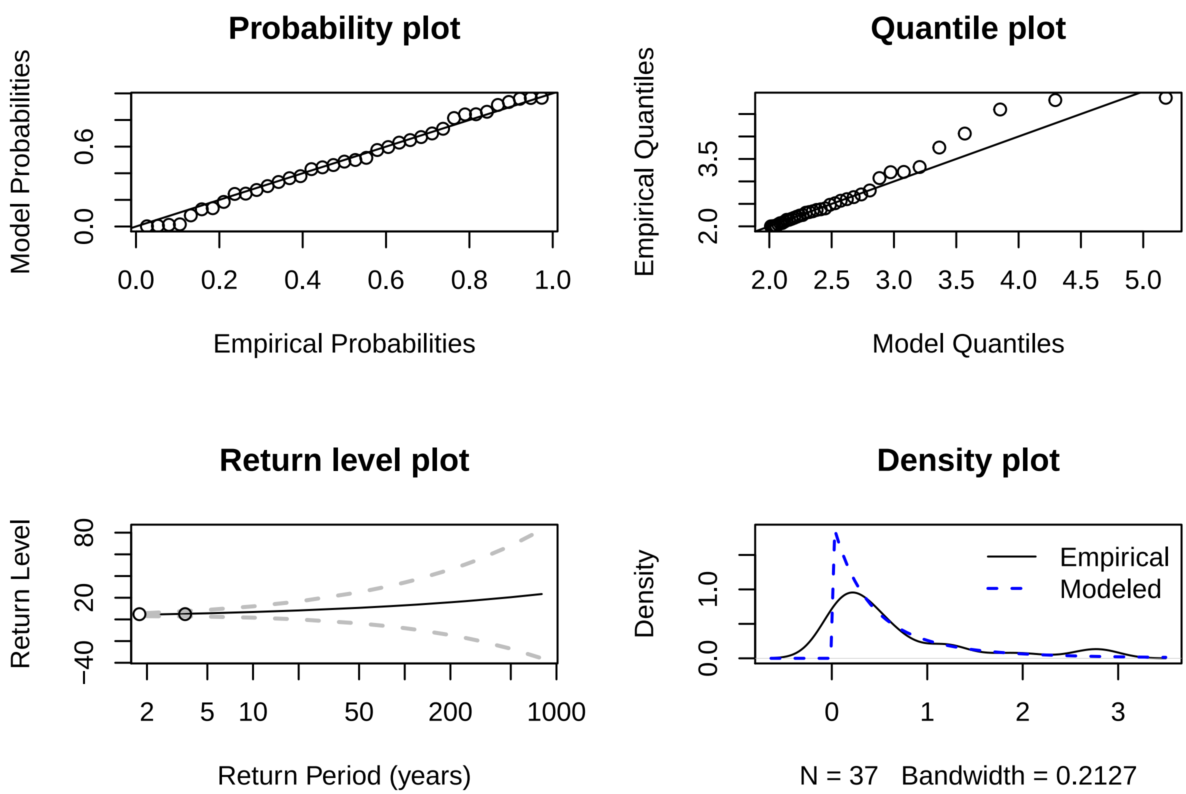



```r
ci.fevd(fitgpd, type = "return.level", method = "proflik")
```

```
## Warning in ci.fevd.mle(fitgpd, type = "return.level", method = "proflik"):
## NaNs produced

## fevd(x = ret, threshold = u, type = "GP", span = 365)
##
## [1] "Profile Likelihood"
##
## [1] "100-year return level: 12.975"
##
## [1] "95% Confidence Interval: (12.5354, 170.123)"
```

Diagnosis

```r
par(mfrow=c(2,2))
plot.fevd(fitgpd, type = "probprob", main="Probability plot")
plot.fevd(fitgpd, type = "qq", main="Quantile plot")
plot.fevd(fitgpd, type = "rl", main="Return level plot")
plot.fevd(fitgpd, type = "density", main="Density plot")
```


Probability plot
Model Probabilities
Empirical Probabilities
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Quantile plot
Empirical Quantiles
Model Quantiles
2.0
3.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Return level plot
Return Level
Return Period (years)
−40
20
80
2
5
10
50
200
1000
Density plot
Density
Empirical
Modeled
0.0
1.0
0
1
2
3
N = 37   Bandwidth = 0.2127