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
```

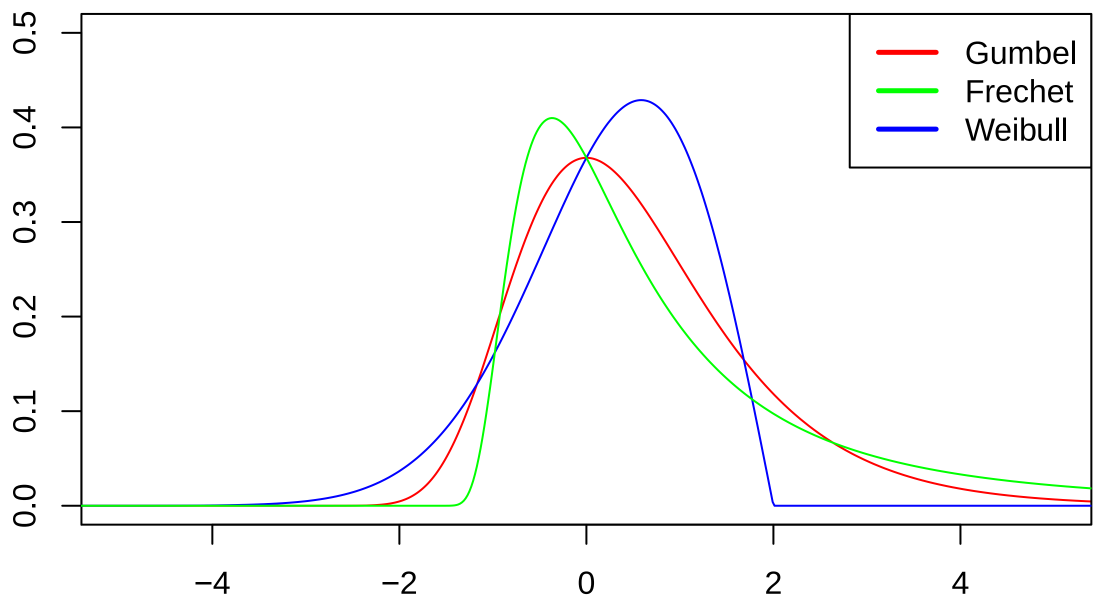
```
source("extreme_functions.r")
```

Chapter 3: Classical Extreme Value Theory and Models

The Generalsed extreme value distribution

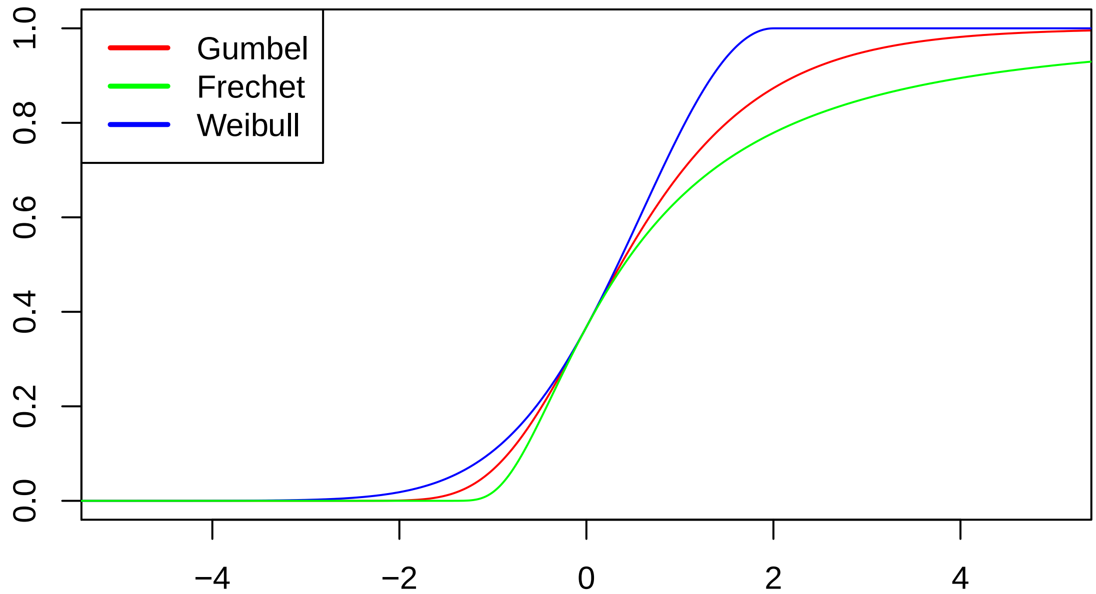
PDF

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

plot(x=xx, y=gum, t="l", xlim=c(-5,5), ylim=c(0,0.5),
     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"))
```



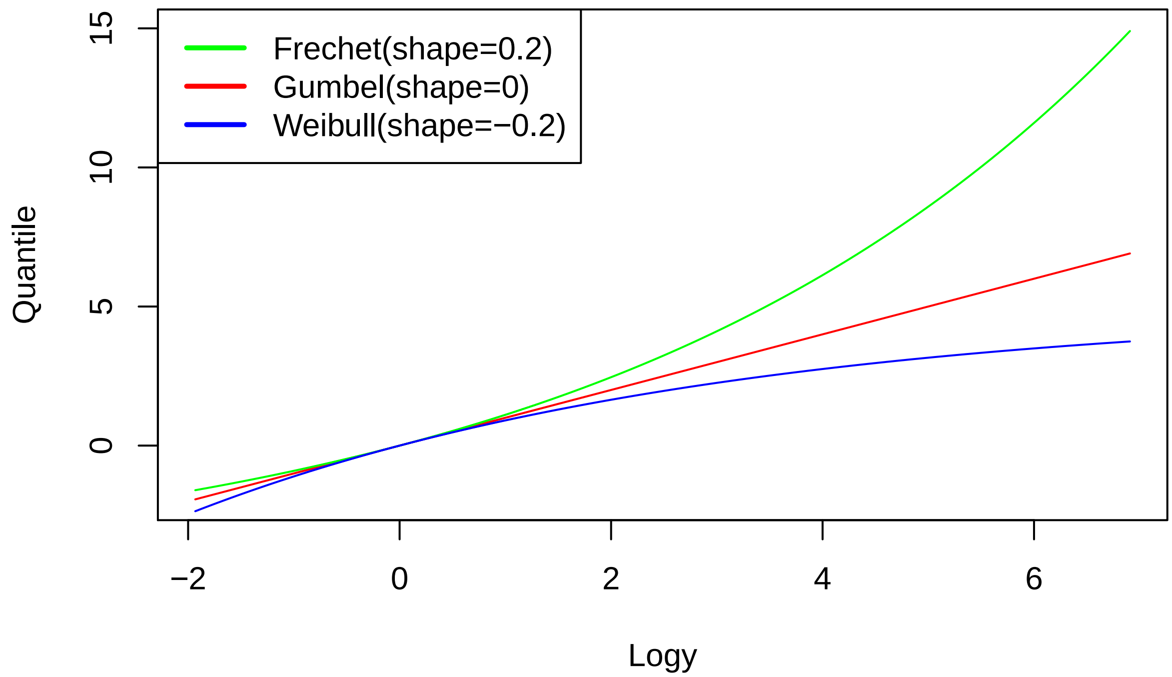
CDF

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

plot(x=xx, y=gum, t="l", xlim=c(-5,5), ylim=c(0,1),
     col="red", ylab="", xlab="")
lines(x=xx, fre, col="blue")
lines(x=xx, wei, col="green")

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

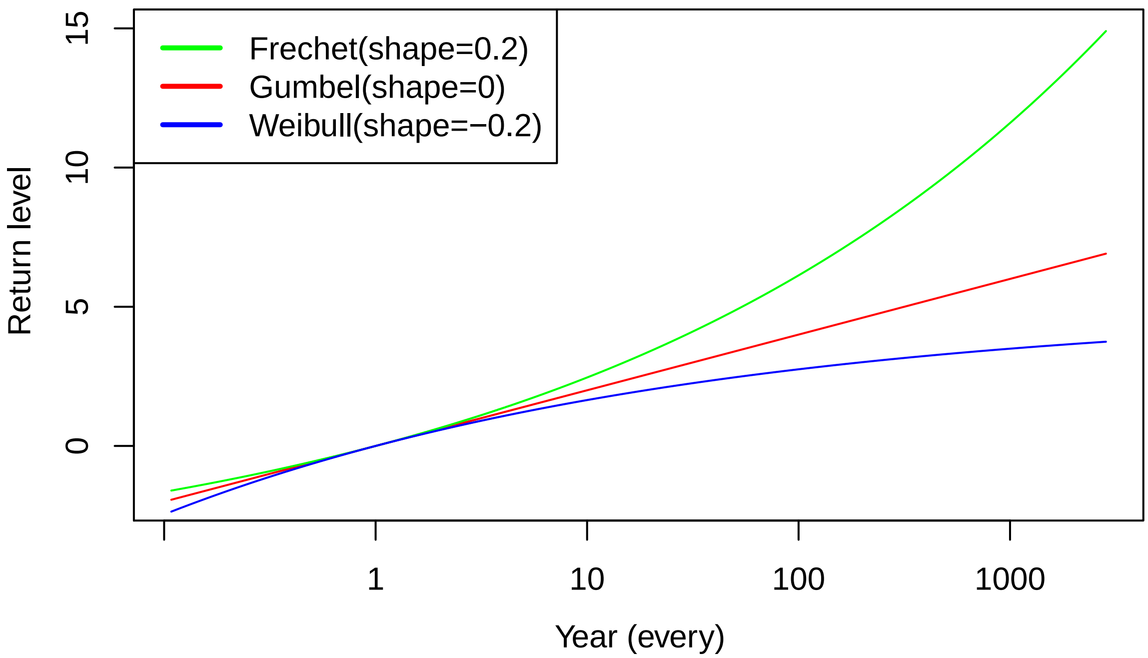


Return level

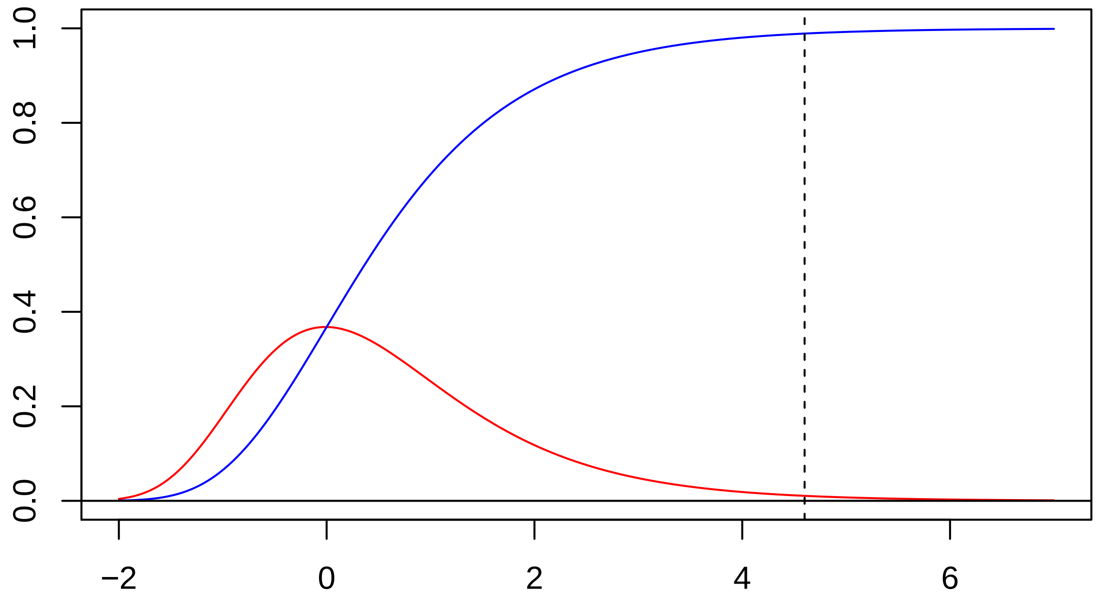
```
p <- seq(0.001, 0.999, length.out = 10000)
r.p <- -log(1-p)

rl.gum <- -log(r.p)
rl.fre <- -(1-r.p^(-0.2))/( 0.2)
rl.wei <- -(1-r.p^( 0.2))/(-0.2)

plot(x=-log(r.p), y=rl.gum, t="l", col="red",
     ylim=c(-2,15), ylab="Quantile", xlab="Logy")
lines(x=-log(r.p), y=rl.fre, col="green")
lines(x=-log(r.p), y=rl.wei, col="blue")
legend("topleft", legend =
         c("Frechet(shape=0.2)", "Gumbel(shape=0)", "Weibull(shape=-0.2)"),
       lwd=c(2.5,2.5,2.5),col=c("green","red","blue"))
```




```
p <- seq(0.001, 0.999, length.out = 10000)
r.p <- -1/log(1-p)

rl.gum <- log(r.p)
rl.fre <- (r.p^( 0.2)-1)/( 0.2)
rl.wei <- (r.p^(-0.2)-1)/(-0.2)

plot(x=log(r.p), y=rl.gum, t="l", col="red",
     ylim=c(-2,15), xaxt='n', ann=FALSE,
     ylab="Return level", xlab="Logy")
lines(x=log(r.p), y=rl.fre, col="green")
lines(x=log(r.p), y=rl.wei, col="blue")
legend("topleft", legend =
         c("Frechet(shape=0.2)", "Gumbel(shape=0)", "Weibull(shape=-0.2)"),
       lwd=c(2.5,2.5,2.5),col=c("green","red","blue"))
axis(side = 1, at = c(-2,0,2,4,6), labels= c("",1,10,100,1000))
mtext("Year (every)",side = 1, line=2.5, at=2.5)
mtext("Return level",side = 2, line=2.5, at=7)
```




```r
library(extRemes)
xx <- seq(-2, 7, length=1000)
pdf <- devd(xx, loc = 0, scale = 1, shape = 0.01, type = c("GEV"))
cdf <- pevd(xx, loc = 0, scale = 1, shape = 0.01, type = c("GEV"))
plot(x=xx, y=pdf, t="l", col="red", ylab="", xlab="", ylim = c(0,1))
lines(x=xx, cdf, col="blue")
abline(h=0)

return.level <- -log(-log(1-0.01))
abline(v=return.level, lty=2)
```

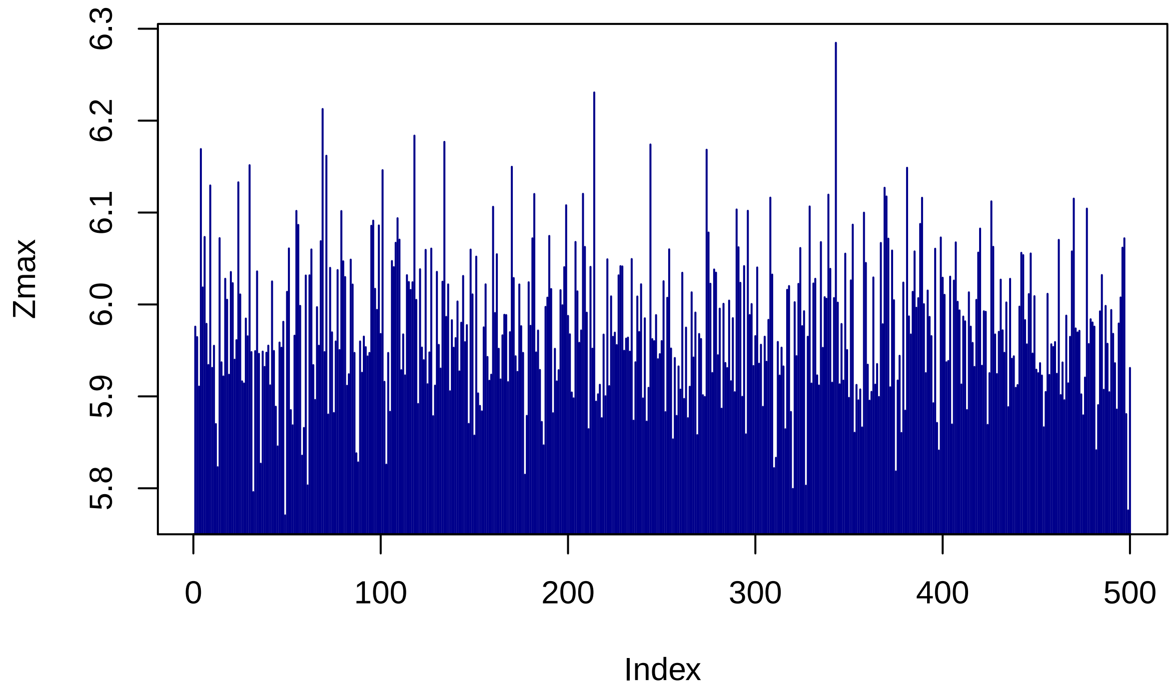


Inference for return levels

Case for xi < 0

Random variates from beta distribution

```
set.seed(100)
n <- 500
Z <- matrix(rbeta(100*n, shape1=1, shape2=2), nrow=100, ncol=n, byrow=T) + 5 +
  rnorm(n, mean = 0, sd = 0.1)

Zmax <- apply(Z, 2, max)
plot(Zmax, t="h", col = "darkblue")
```



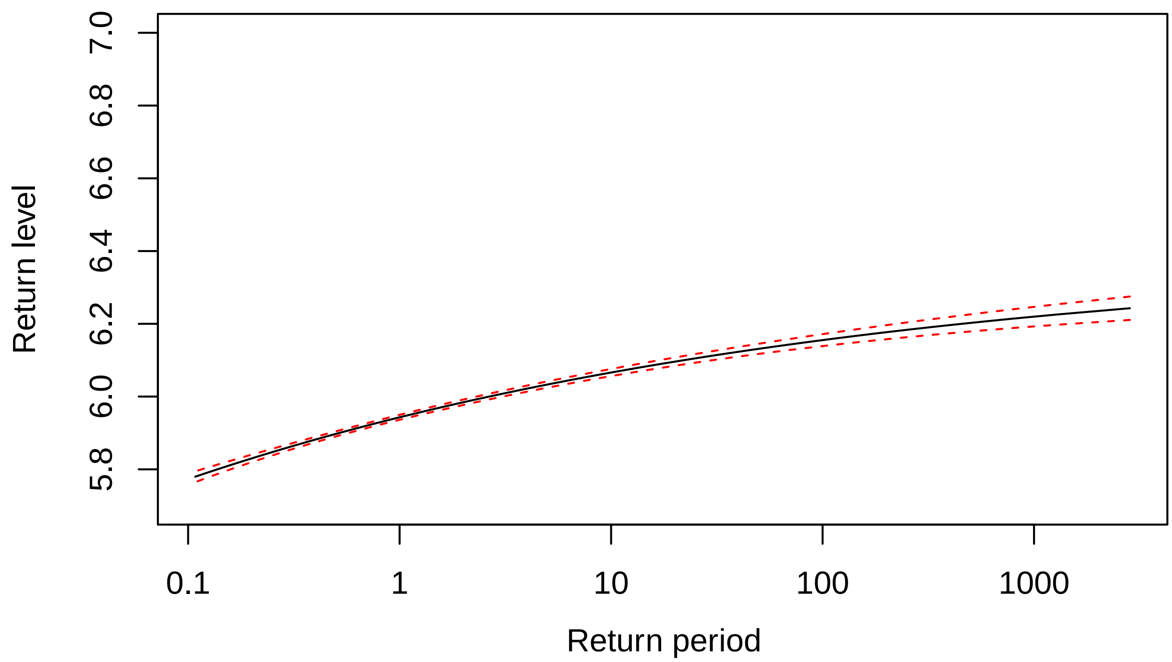

```
fit <- gev.fit(Zmax)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] -576.9642
##
## $mle
## [1]  5.94301637  0.07199454 -0.16131124
##
## $se
## [1] 0.003497106 0.002394182 0.021856572
```

```
mu <- fit$mle[1]; sigma <- fit$mle[2]; xi <- fit$mle[3]; cov <- fit$cov

# Return level data
p <- seq(0, 1, by = 0.001)
y.p <- -log(1-p)
rl <- mu-sigma*(1-y.p^(-xi))/(xi)
rl.data <- data.frame(prob=p, y.p=y.p, return.level=rl)

# Variance fdata
var <- matrix(nrow = length(y.p), ncol = 1)
for(i in 1:length(y.p)){
  dz1 <- 1
  dz2 <- -xi^(-1)*(1-y.p[i]^(-xi))
```

```r
  dz3 <- sigma*xi^(-2)*(1-y.p[i]^(-xi)) - sigma*xi^(-1)*y.p[i]^(-xi)*log(y.p[i])
  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, y.p=y.p, var=var)
```

```r
plot(x=-log(y.p), y=rl.data$return.level, t="l",
     ylim = c(5.7, 7),
     xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(-2,0,2,4,6), labels= c(0.1,1,10,100,1000))
mtext("Return period",side = 1, line=2.5, at=2.5)

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(y.p), y=lower.line, col="red", lty=2)
lines(x=-log(y.p), y=upper.line, col="red", lty=2)
```



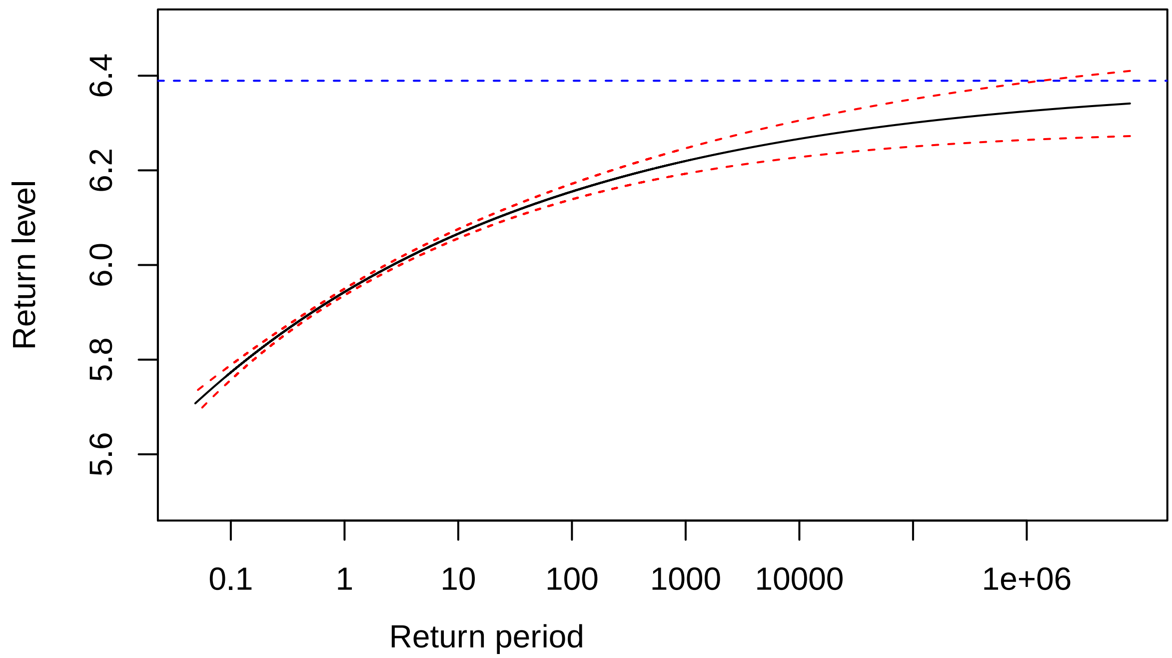

```r
# upper <- mu-sigma/xi
# abline(h=upper, col="blue", lty=2)
```

Precise level

```r
mu <- fit$mle[1]; sigma <- fit$mle[2]; xi <- fit$mle[3]; cov <- fit$cov

# Return level data
```

```r
p <- seq(0, 1, by = 0.000001)
y.p <- -log(1-p)
rl <- mu-sigma*(1-y.p^(-xi))/(xi)
rl.data <- data.frame(prob=p, y.p=y.p, return.level=rl)

# Variance fdata
var <- matrix(nrow = length(y.p), ncol = 1)
for(i in 1:length(y.p)){
  dz1 <- 1
  dz2 <- -xi^(-1)*(1-y.p[i]^(-xi))
  dz3 <- sigma*xi^(-2)*(1-y.p[i]^(-xi)) - sigma*xi^(-1)*y.p[i]^(-xi)*log(y.p[i])
  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, y.p=y.p, var=var)

upper <- mu - sigma/xi
```

```r
plot(x=-log(y.p), y=rl.data$return.level, t="l",
     ylim = c(5.5, 6.5),
     xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(-2,0,2,4,6,8,10,12),
     labels= c(0.1,1,10,100,1000,10000,100000,1000000))
mtext("Return period",side = 1, line=2.5, at=2.5)

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(y.p), y=lower.line, col="red", lty=2)
lines(x=-log(y.p), y=upper.line, col="red", lty=2)

abline(h=upper, col="blue", lty=2)
```



Annual maximum sea-levels at port price (using ismev)

```
library(ismev)
library(extRemes)
source("extreme_functions.r")
```

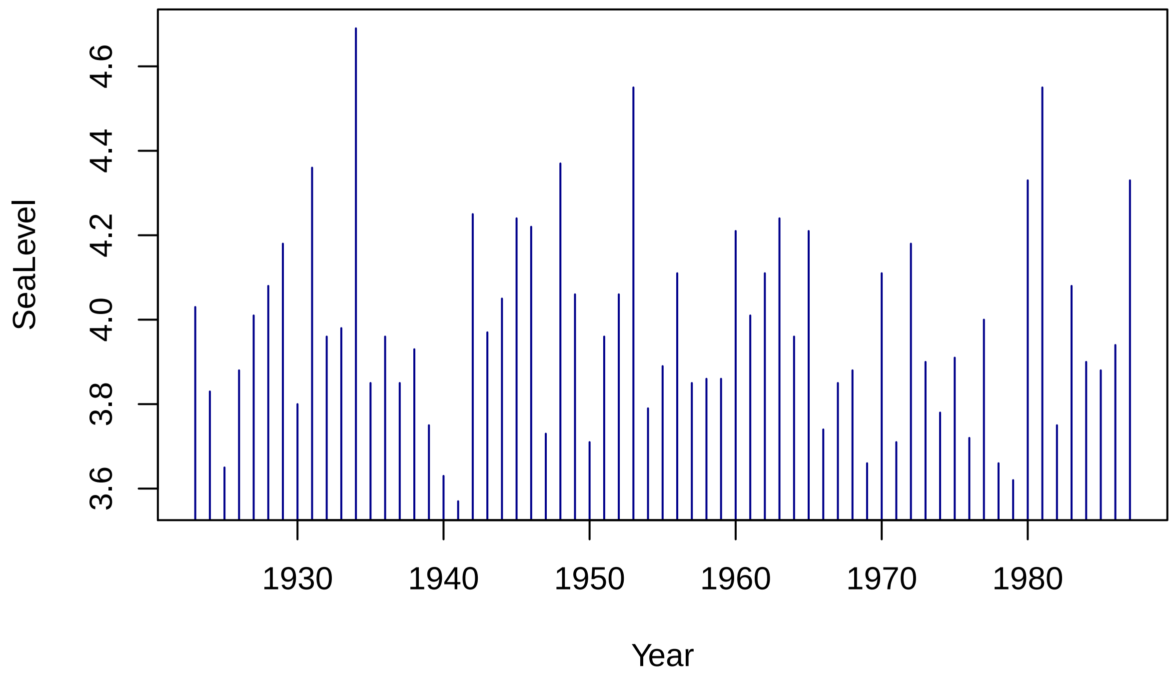
Data

The portpirie data frame has 65 rows and 2 columns. The second column gives annual maximimum sea levels recorded at Port Pirie, South Australia, from 1923 to 1987. The first column gives the corresponding years.

```
data("portpirie")
str(portpirie)
```

```
## 'data.frame':    65 obs. of  2 variables:
##  $ Year    : num  1923 1924 1925 1926 1927 ...
##  $ SeaLevel: num  4.03 3.83 3.65 3.88 4.01 4.08 4.18 3.8 4.36 3.96 ...
```

```
plot(portpirie, t="h", col="darkblue")
```



GEV fit

```r
options(digits = 3)
fitgev <- gev.fit(portpirie$SeaLevel)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] -4.34
##
## $mle
## [1]  3.8747  0.1980 -0.0501
##
## $se
## [1] 0.0279 0.0202 0.0983
```

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

```
## [1] 4.34
```

```r
# Covariance
fitgev$cov
```

```
##           [,1]      [,2]      [,3]
```

```
## [1,]  0.000780  0.000197 -0.001074
## [2,]  0.000197  0.000410 -0.000777
## [3,] -0.001074 -0.000777  0.009654
```

```r
# Standard errors
fitgev$se
```

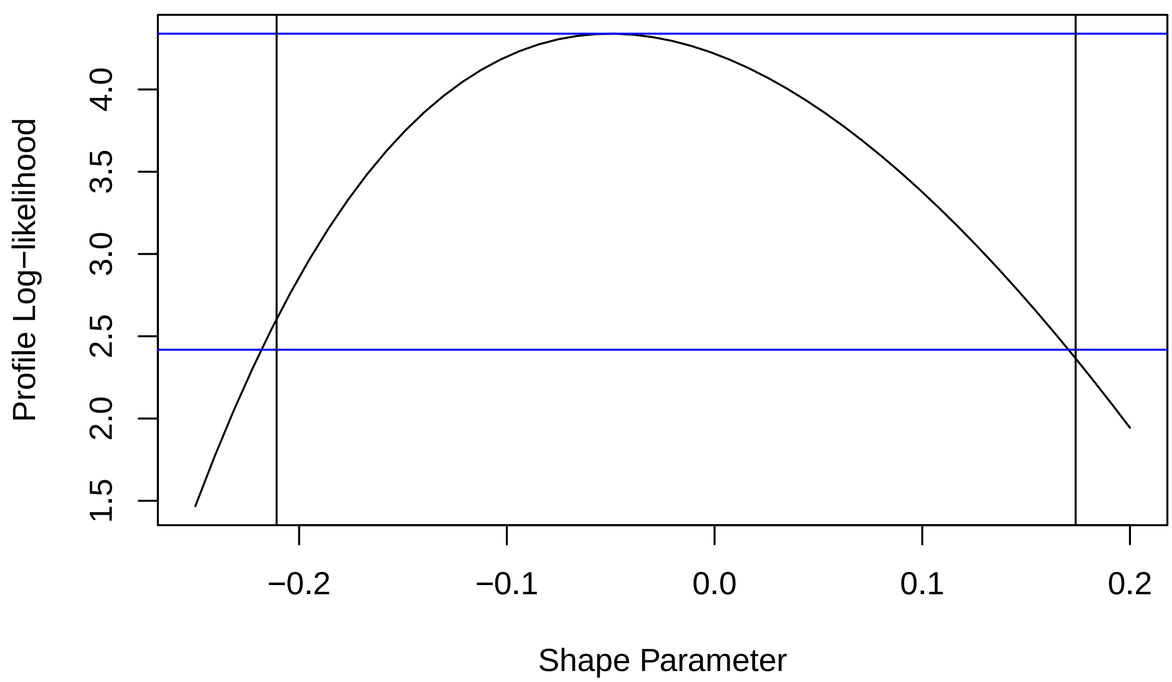
```
## [1] 0.0279 0.0202 0.0983
```

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

```
##        lower upper
## mu     3.820 3.929
## sigma  0.158 0.238
## xi    -0.243 0.142
```

```r
# Profile likelihood for shape
cgev.profxi(fitgev, xlow = -0.25, xup = 0.2, nint = 50)
```

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

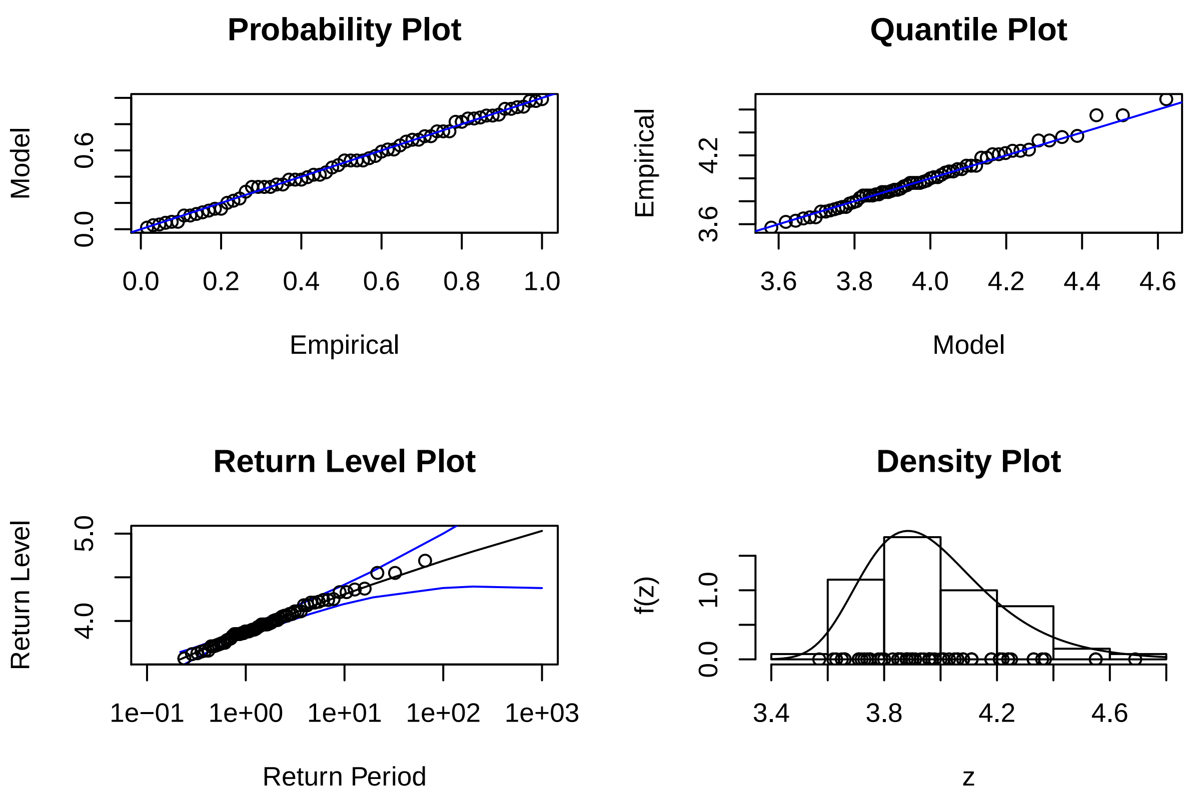



```
## $optimal.value
## [1] 4.34
##
## $optimal.x
```

```
## [1] -0.048
##
## $confint.lower
## [1] -0.211
##
## $confint.upper
## [1] 0.174
```

Diagnosis

```r
gev.diag(fitgev)
```

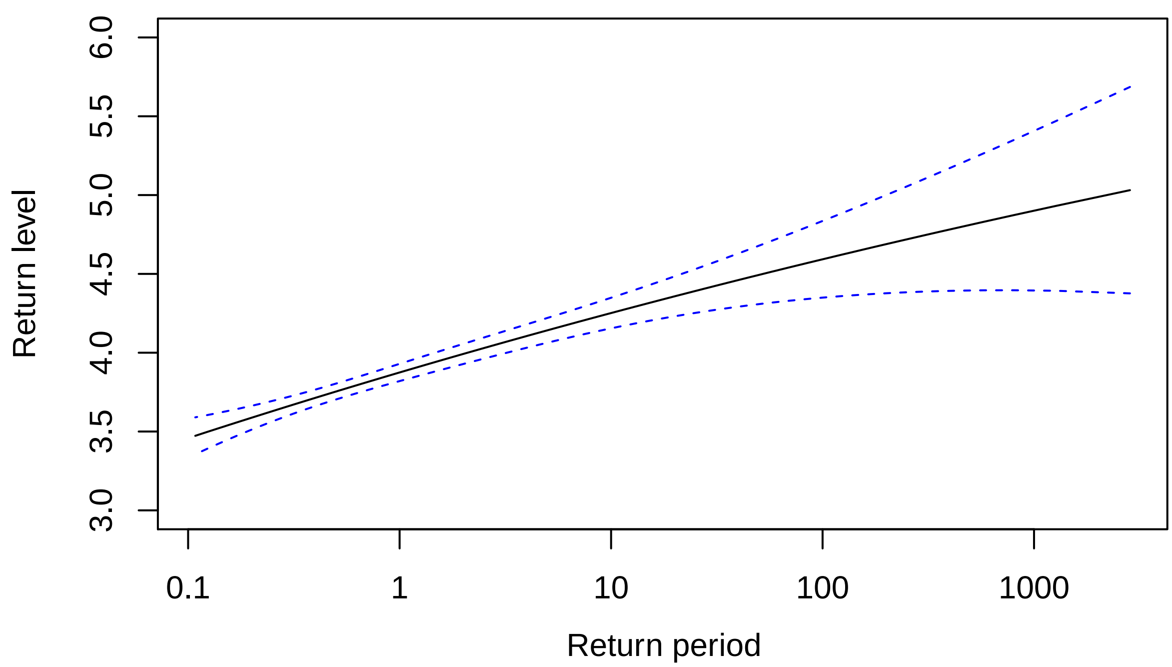


Return level

```r
mu <- fitgev$mle[1]; sigma <- fitgev$mle[2]; xi <- fitgev$mle[3]; cov <- fitgev$cov

# Return level data
p <- seq(0, 1, by = 0.001)
y.p <- -log(1-p)
rl <- mu-sigma*(1-y.p^(-xi))/(xi)
rl.data <- data.frame(prob=p, y.p=y.p, return.level=rl)

# Variance fdata
var <- matrix(nrow = length(y.p), ncol = 1)
```

```
for(i in 1:length(y.p)){
  dz1 <- 1
  dz2 <- -xi^(-1)*(1-y.p[i]^(-xi))
  dz3 <- sigma*xi^(-2)*(1-y.p[i]^(-xi)) - sigma*xi^(-1)*y.p[i]^(-xi)*log(y.p[i])
  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, y.p=y.p, var=var)
```

```
plot(x=-log(y.p), y=rl.data$return.level, t="l",
     ylim = c(3, 6),
     xaxt='n', ylab="Return level", xlab="")
axis(side = 1, at = c(-2,0,2,4,6), labels= c(0.1,1,10,100,1000))
mtext("Return period",side = 1, line=2.5, at=2.5)

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



10-year return

```r
#return level
rl <- rl.data$return.level[rl.data$prob==0.1]

#standard deviation
sd <- sqrt(var.data$var[var.data$prob==0.1])

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

```
## [1] 4.19
```

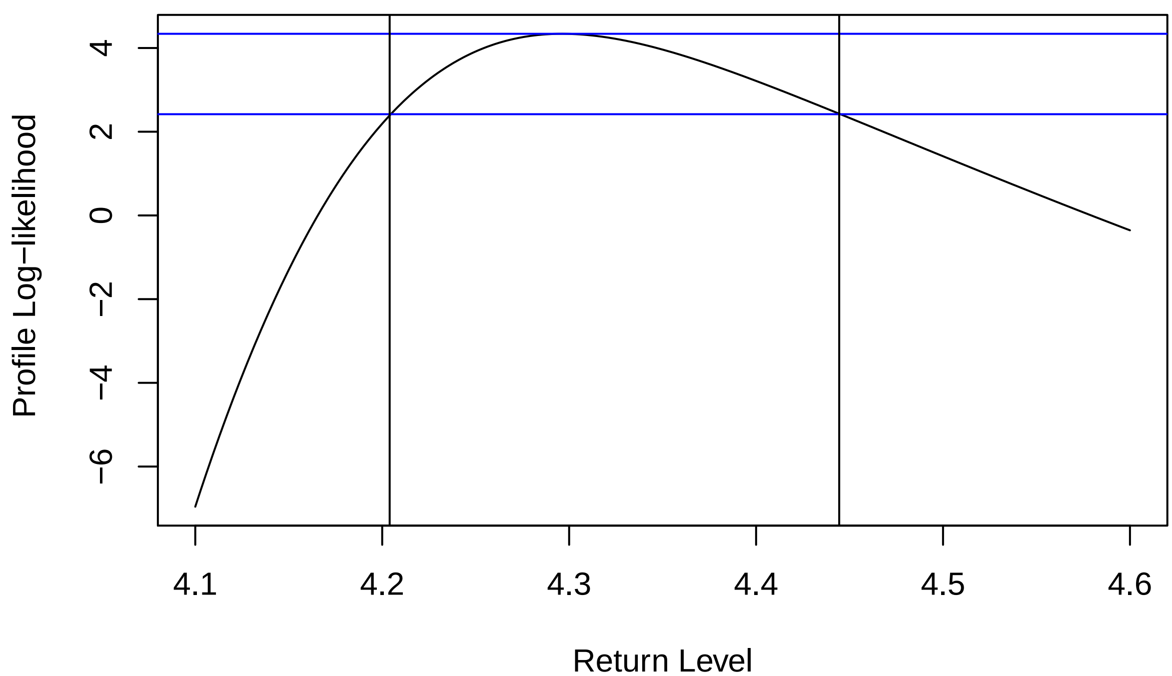
```r
rl+1.96*sd
```

```
## [1] 4.4
```

Priofile likelihood

```r
cgev.prof(fitgev, m = 10, xlow = 4.1, xup = 4.6)
```

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




```
## $optimal.value
## [1] 4.34
##
## $optimal.x
## [1] 4.3
##
## $confint.lower
## [1] 4.2
```

```
##
## $confint.upper
## [1] 4.44
```

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] 4.38
```

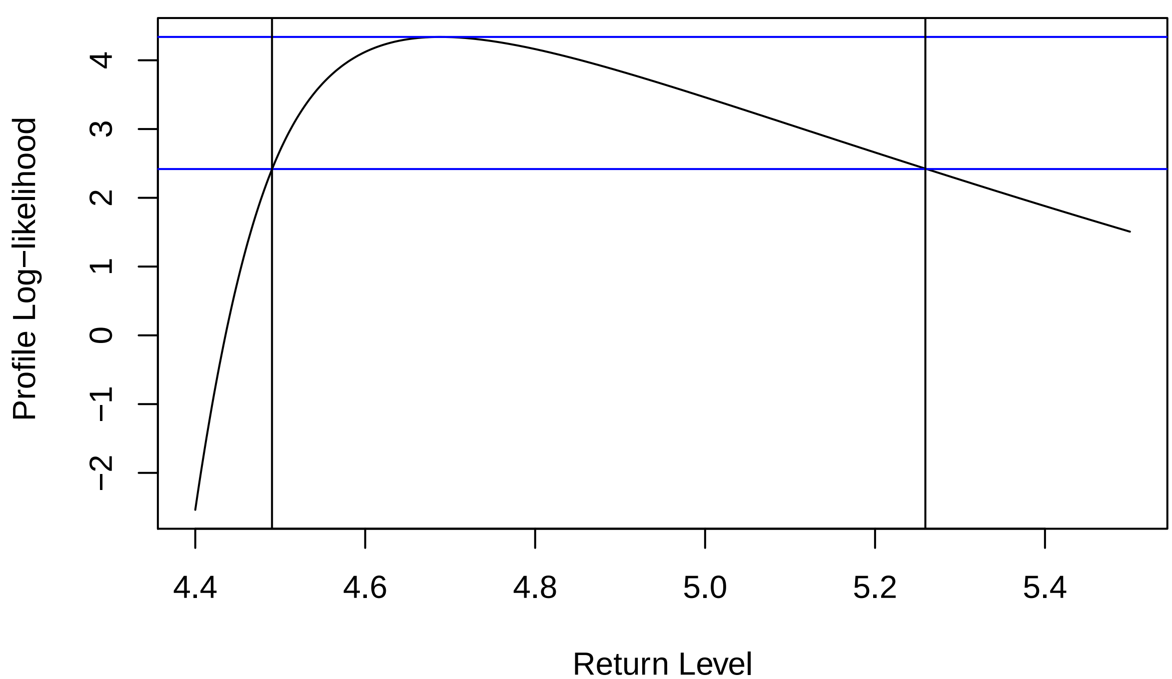
```r
rl+1.96*sd
```

```
## [1] 5
```

Priofile likelihood

```r
cgev.prof(fitgev, m = 100, xlow = 4.4, xup = 5.5)
```

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




```
## $optimal.value
## [1] 4.34
```

```
##
## $optimal.x
## [1] 4.69
##
## $confint.lower
## [1] 4.49
##
## $confint.upper
## [1] 5.26
```

Gumbel fit

```r
fitgum <- gum.fit(portpirie$SeaLevel)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] -4.22
##
## $mle
## [1] 3.869 0.195
##
## $se
## [1] 0.0255 0.0189
```

```r
# Covariance
fitgum$cov
```

```
##          [,1]     [,2]
## [1,] 0.000650 0.000153
## [2,] 0.000153 0.000355
```

```r
# Standard errors
fitgum$se
```

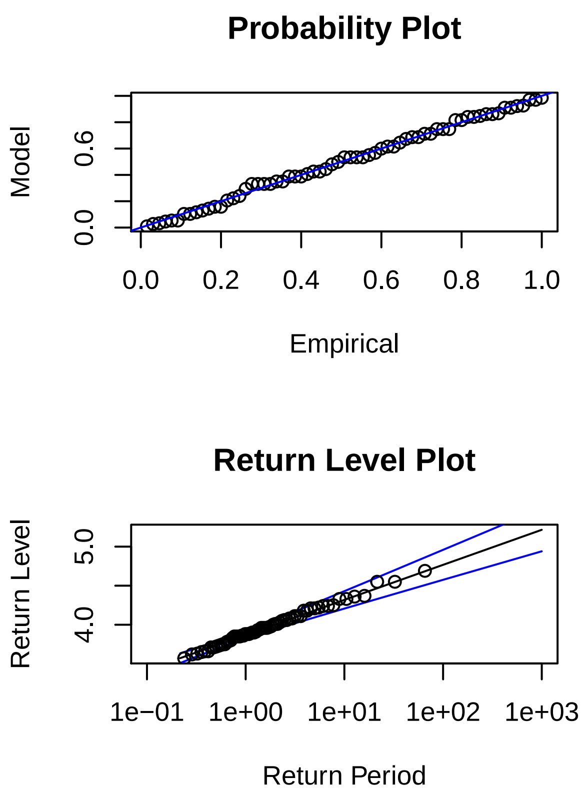
```
## [1] 0.0255 0.0189
```

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

```
##       lower upper
## mu    3.819 3.919
## sigma 0.158 0.232
```

Diagnosis

```r
gum.diag(fitgum)
```



Model comparison(Log ratio test)

```
#statistic
2*(-fitgev$nllh - -fitgum$nllh)
```

```
## [1] 0.243
```

```
#p-value
as.numeric(pchisq(-fitgev$nllh - -fitgum$nllh, df=1, lower.tail=FALSE))
```

```
## [1] 0.728
```

Annual maximum sea-levels at port price (using extRemes)

GEV fit

```
options(digits = 3)
fitgev <- fevd(portpirie$SeaLevel, type = "GEV")
```

```
# estimated paramters
par.gev <- distill(fitgev)[1:3]
par.gev
```

```
## location    scale    shape
##   3.8747   0.1980  -0.0501
```

```r
# log-likelihood
loglik.gev <- -distill(fitgev)[4]
loglik.gev
```

```
## nllh
## 4.34
```

```r
# covariance
cov.gev <- matrix(as.numeric(distill(fitgev)[5:13]),nrow = 3, byrow = T,
              dimnames =list(c("location","scale","shape"),
                             c("location","scale","shape")))
cov.gev
```

```
##           location     scale     shape
## location  0.000780  0.000197 -0.001074
## scale     0.000197  0.000410 -0.000777
## shape    -0.001074 -0.000777  0.009654
```

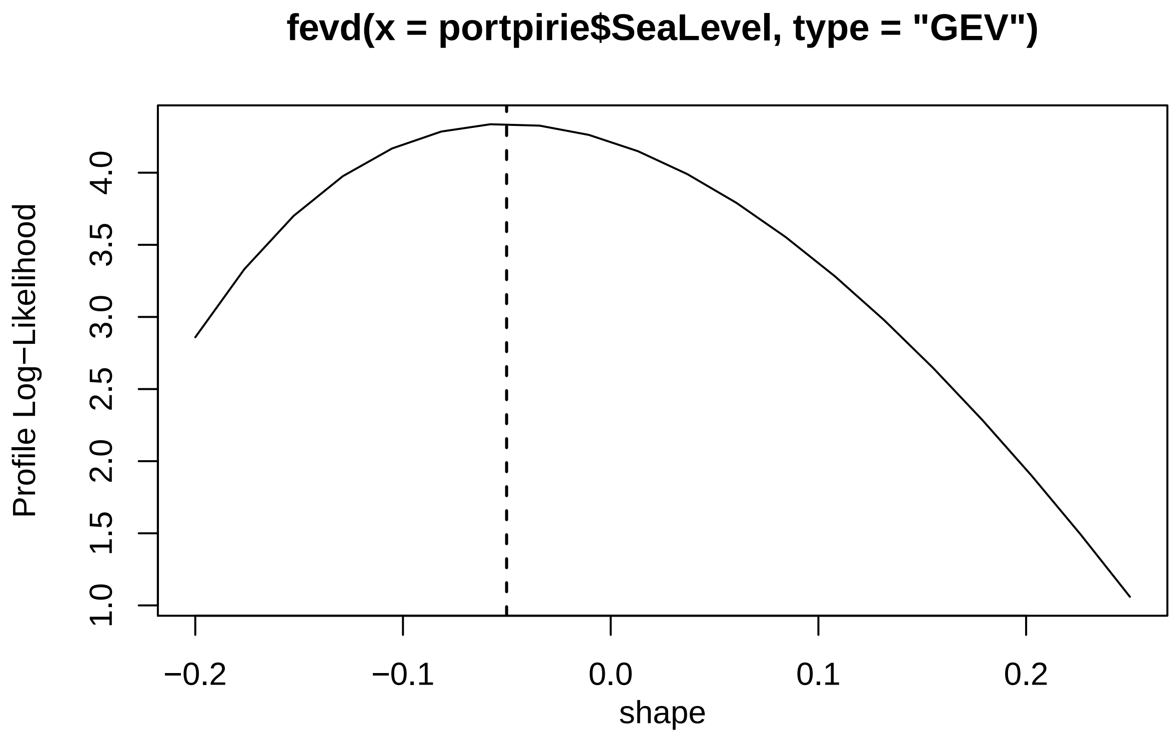
```r
# Standard errors
se.gev <- sqrt(diag(cov.gev))
se.gev
```

```
## location    scale    shape
##   0.0279   0.0202   0.0983
```

```r
# confidence interval
ci.fevd(fitgev, type = "parameter")
```

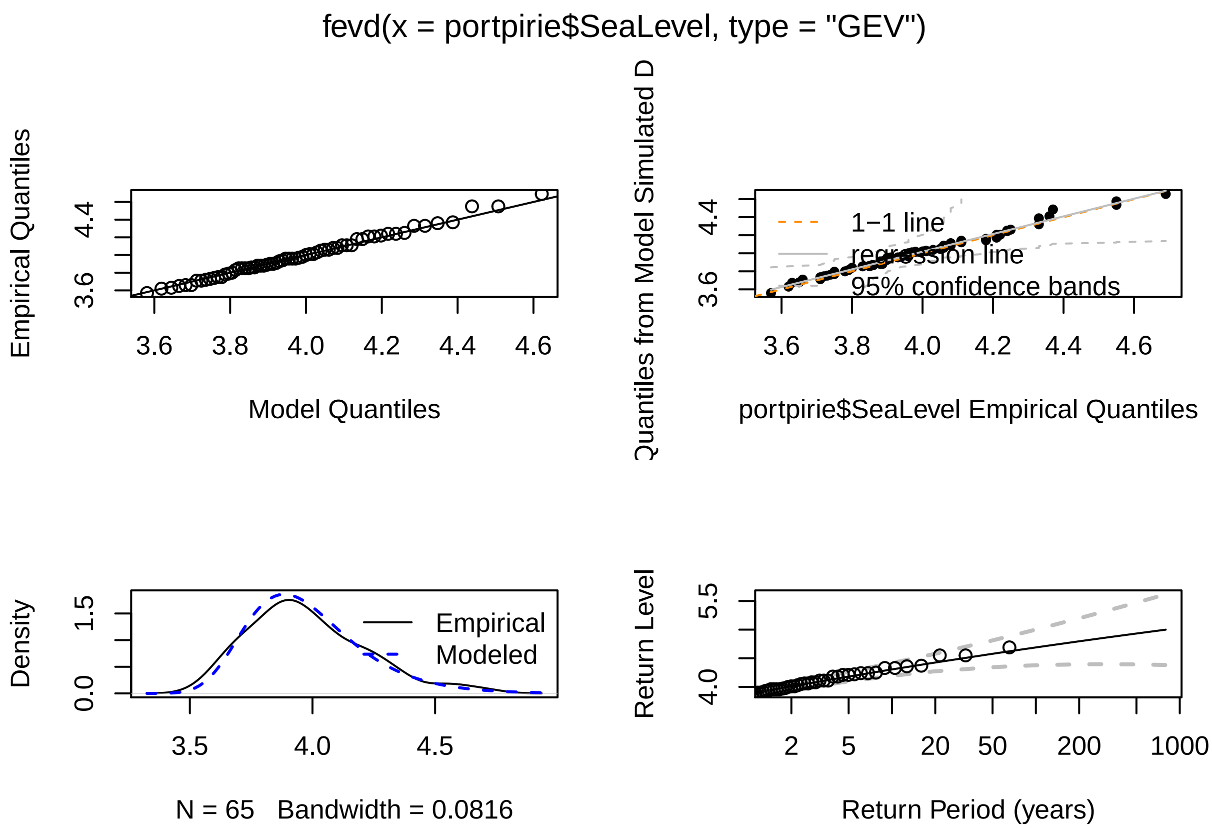
```
## fevd(x = portpirie$SeaLevel, type = "GEV")
##
## [1] "Normal Approx."
##
##          95% lower CI Estimate 95% upper CI
## location        3.820   3.8747        3.929
## scale           0.158   0.1980        0.238
## shape          -0.243  -0.0501        0.142
```

```r
# Profile likelihood for shape
profliker(fitgev, type = "parameter", which.par = 3, xrange = c(-0.2, 0.25))
```



Diagnosis

```
plot(fitgev)
```

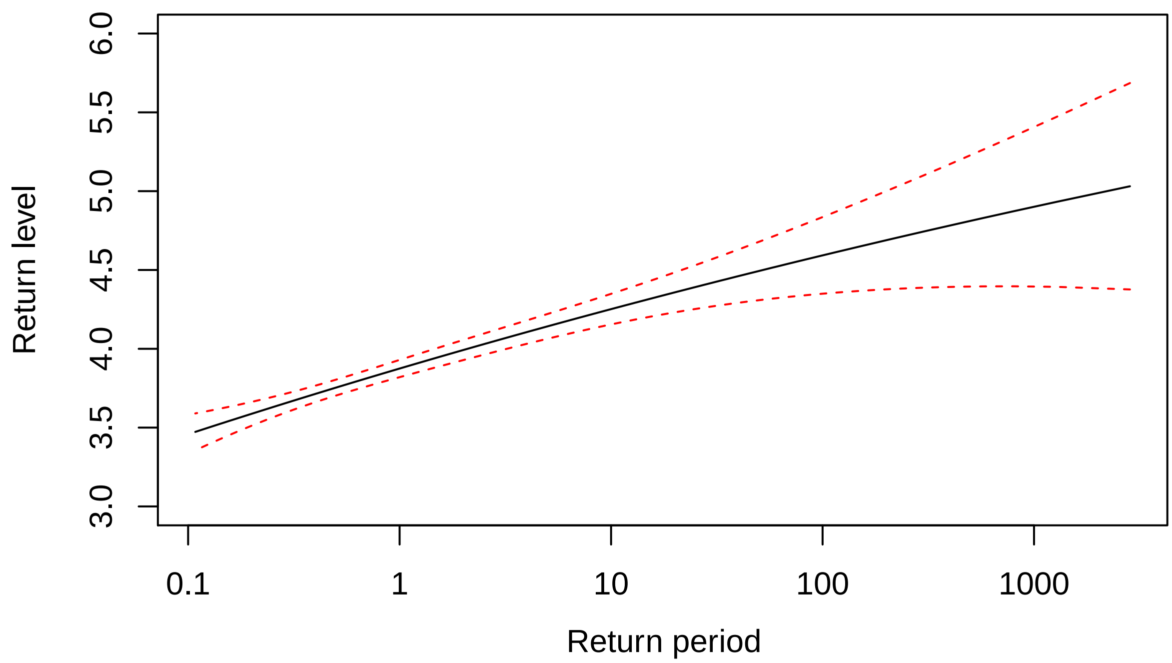


Return level

```
mu <- par.gev[1]; sigma <- par.gev[2]; xi <- par.gev[3]; cov <- cov.gev

# Return level data
p <- seq(0, 1, by = 0.001)
y.p <- -log(1-p)
rl <- mu-sigma*(1-y.p^(-xi))/(xi)
rl.data <- data.frame(prob=p, y.p=y.p, return.level=rl)

# Variance fdata
var <- matrix(nrow = length(y.p), ncol = 1)
for(i in 1:length(y.p)){
  dz1 <- 1
  dz2 <- -xi^(-1)*(1-y.p[i]^(-xi))
  dz3 <- sigma*xi^(-2)*(1-y.p[i]^(-xi)) - sigma*xi^(-1)*y.p[i]^(-xi)*log(y.p[i])
  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, y.p=y.p, var=var)
```

```
plot(x=-log(y.p), y=rl.data$return.level, t="l",
     ylim = c(3, 6),
     xaxt='n', ylab="Return level", xlab="")
```

```
axis(side = 1, at = c(-2,0,2,4,6), labels= c(0.1,1,10,100,1000))
mtext("Return period",side = 1, line=2.5, at=2.5)

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(y.p), y=lower.line, col="red", lty=2)
lines(x=-log(y.p), y=upper.line, col="red", lty=2)
```



10-year return

```
#return level
rl <- rl.data$return.level[rl.data$prob==0.1]

#standard deviation
sd <- sqrt(var.data$var[var.data$prob==0.1])

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

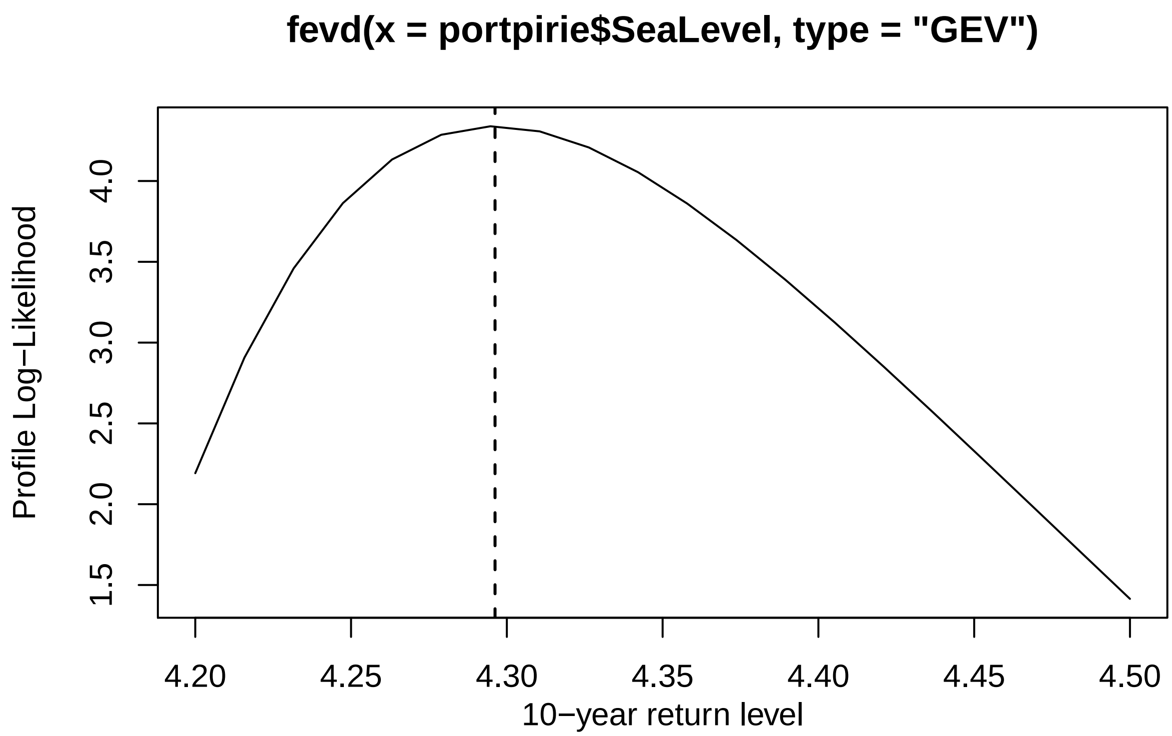
```
## [1] 4.19
```

```
rl+1.96*sd
```

```
## [1] 4.4
```

Priofile likelihood

```r
profliker(fitgev, type = "return.level", return.period = 10, xrange = c(4.2, 4.5))
```



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
```

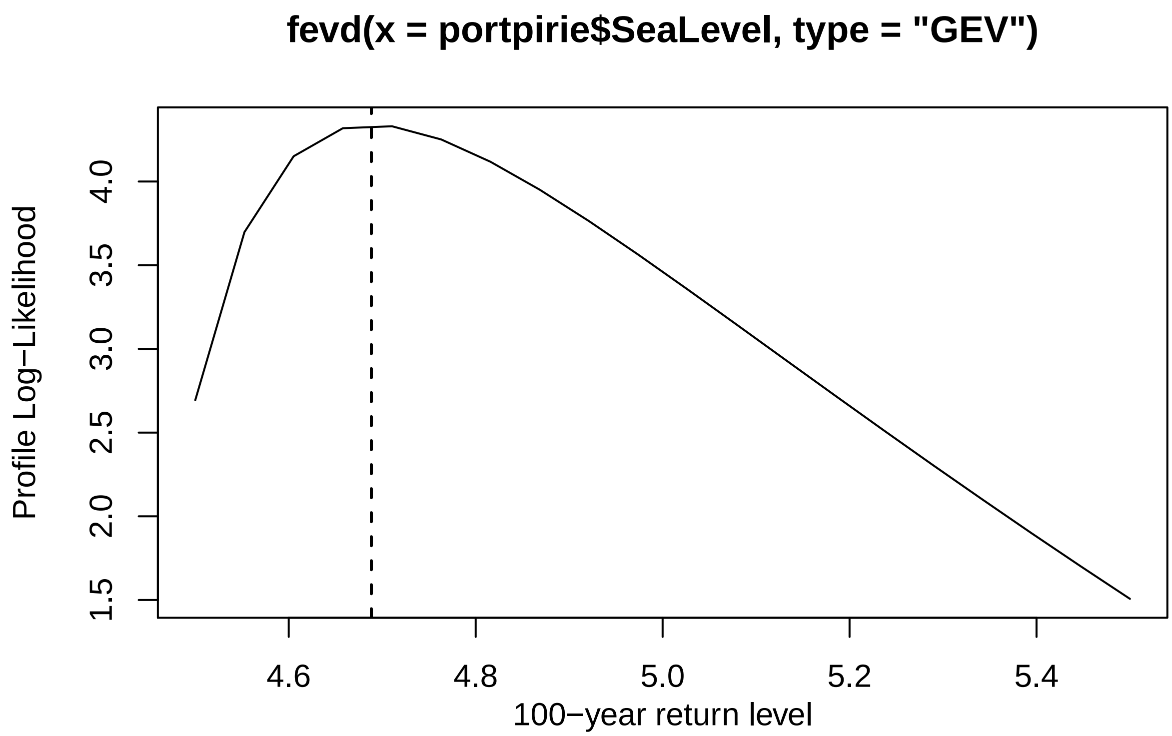
```
## [1] 4.38
```

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

```
## [1] 5
```

Priofile likelihood

```r
profliker(fitgev, type = "return.level", return.period = 100, xrange = c(4.5, 5.5))
```



Gumbel fit

```r
fitgum <- fevd(portpirie$SeaLevel, type = "Gumbel")
```

```r
# estimated paramters
par.gum <- distill(fitgum)[1:2]
par.gum
```

```
## location    scale
##    3.869    0.195
```

```r
# log-likelihood
loglik.gum <- -distill(fitgum)["nllh"]
loglik.gum
```

```
## nllh
## 4.22
```

```r
# Covariance
cov.gum <- matrix(as.numeric(distill(fitgum)[4:7]),nrow = 2, byrow = T,
              dimnames =list(c("location","scale"), c("location","scale")))
cov.gum
```

```
##            location    scale
```

```
## location 0.000650 0.000153
## scale    0.000153 0.000355
```

```r
# Standard errors
se.gum <- sqrt(diag(cov.gum))
se.gum
```

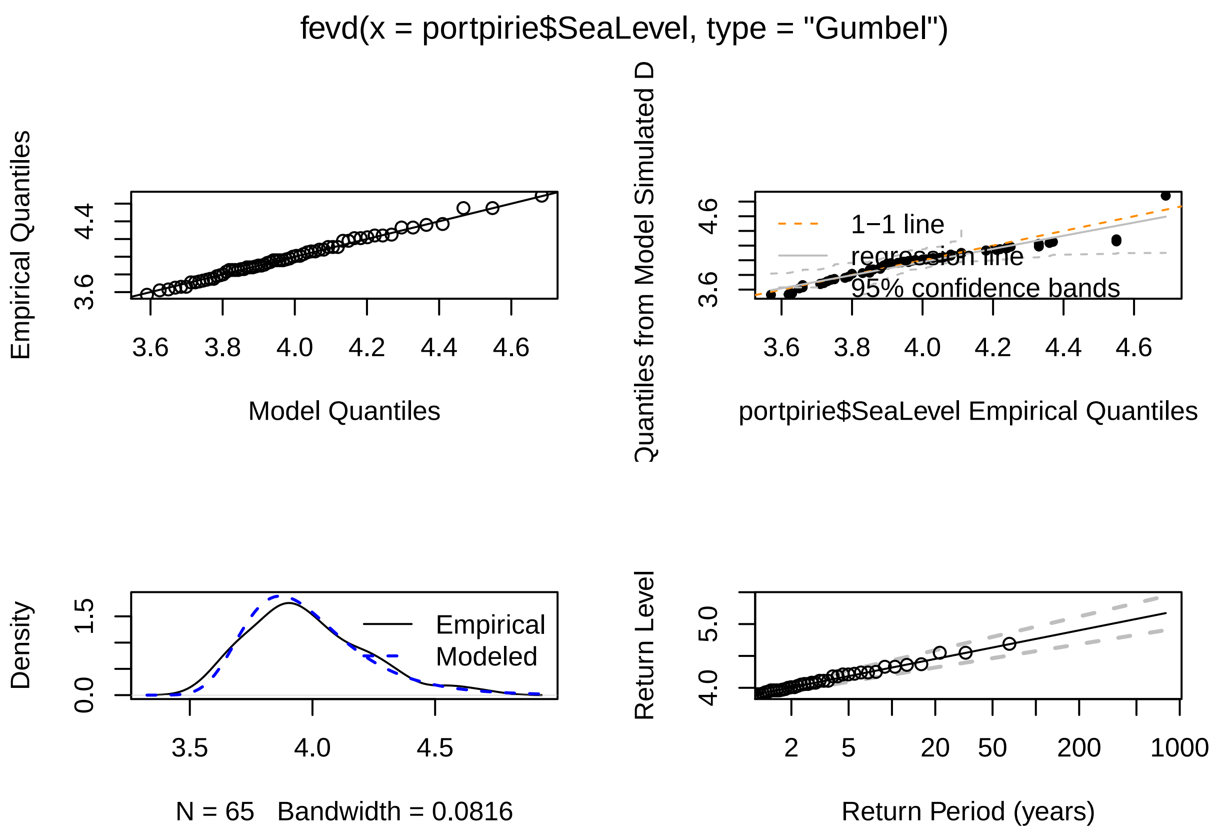
```
## location     scale
##   0.0255    0.0189
```

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

```
## fevd(x = portpirie$SeaLevel, type = "Gumbel")
##
## [1] "Normal Approx."
##
##          95% lower CI Estimate 95% upper CI
## location        3.819    3.869        3.919
## scale           0.158    0.195        0.232
```

Diagnosis

```r
plot(fitgum)
```



Model comparison(Log ratio test)

```
#statistic
2*(loglik.gev - loglik.gum)
```

```
##  nllh
## 0.243
```

```
#p-value
as.numeric(pchisq(loglik.gev - loglik.gum, df=1, lower.tail=FALSE))
```

```
## [1] 0.728
```