# An Introduction to Statistical Modelling of Extreme Values

```r
library(ismev)
```

```
## Loading required package: mgcv

## Loading required package: nlme

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

```r
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
```

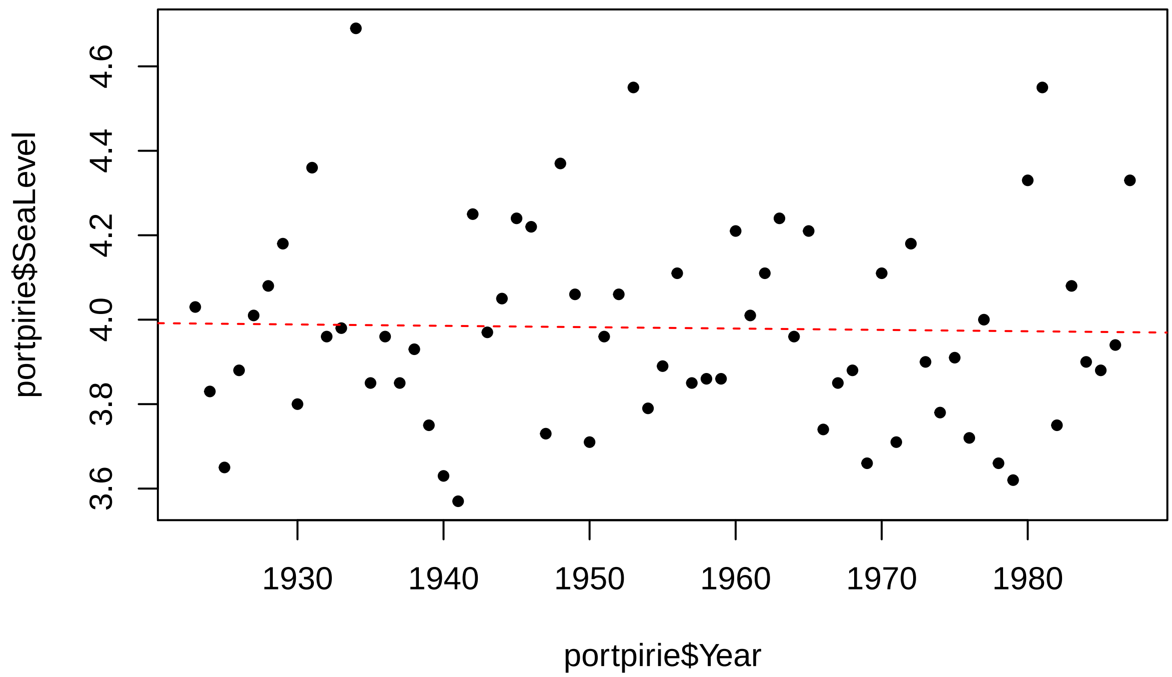
```r
source("extreme_functions.r")
```

## Chapter 6: Non-stationary squence

### Port Pirie

```r
library(ismev)
data("portpirie")
```

```r
plot(x=portpirie$Year, y=portpirie$SeaLevel, pch=20)
fit <- lm(SeaLevel ~ Year, data = portpirie)
abline(fit, col="red", lty=2)
```




```r
summary(fit)
```

```
##
## Call:
## lm(formula = SeaLevel ~ Year, data = portpirie)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4151 -0.1606 -0.0280  0.1316  0.7026
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.6086420  3.1322646   1.471    0.146
## Year        -0.0003212  0.0016021  -0.201    0.842
##
## Residual standard error: 0.2423 on 63 degrees of freedom
## Multiple R-squared:  0.0006378,  Adjusted R-squared:  -0.01523
## F-statistic: 0.0402 on 1 and 63 DF,  p-value: 0.8417
```

**Fit with normal model**

```r
fitgev <- gev.fit(portpirie$SeaLevel)
```

```
## $conv
## [1] 0
```

```
##
## $nllh
## [1] -4.339058
##
## $mle
## [1]  3.87474692  0.19804120 -0.05008773
##
## $se
## [1] 0.02793211 0.02024610 0.09825633
```

### Fit with linear time model for u

```r
ti <- matrix(ncol=1,nrow=length(portpirie$SeaLevel))
ti[,1] <- seq(1,65,1)

fitgev.ut <- gev.fit(portpirie$SeaLevel, ydat = ti, mul = 1)
```

```
## $model
## $model[[1]]
## [1] 1
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
## [1] -4.375107
##
## $mle
## [1]  3.8865633240 -0.0003548552  0.1979732663 -0.0504552135
##
## $se
## [1] 0.051248878 0.001306864 0.020207615 0.097786572
```

### Ratio test

```r
2*(-fitgev.ut$nllh - -fitgev$nllh)
```

```
## [1] 0.07209759
```

No difference

## Sea levels ans Southern Oscillation Index

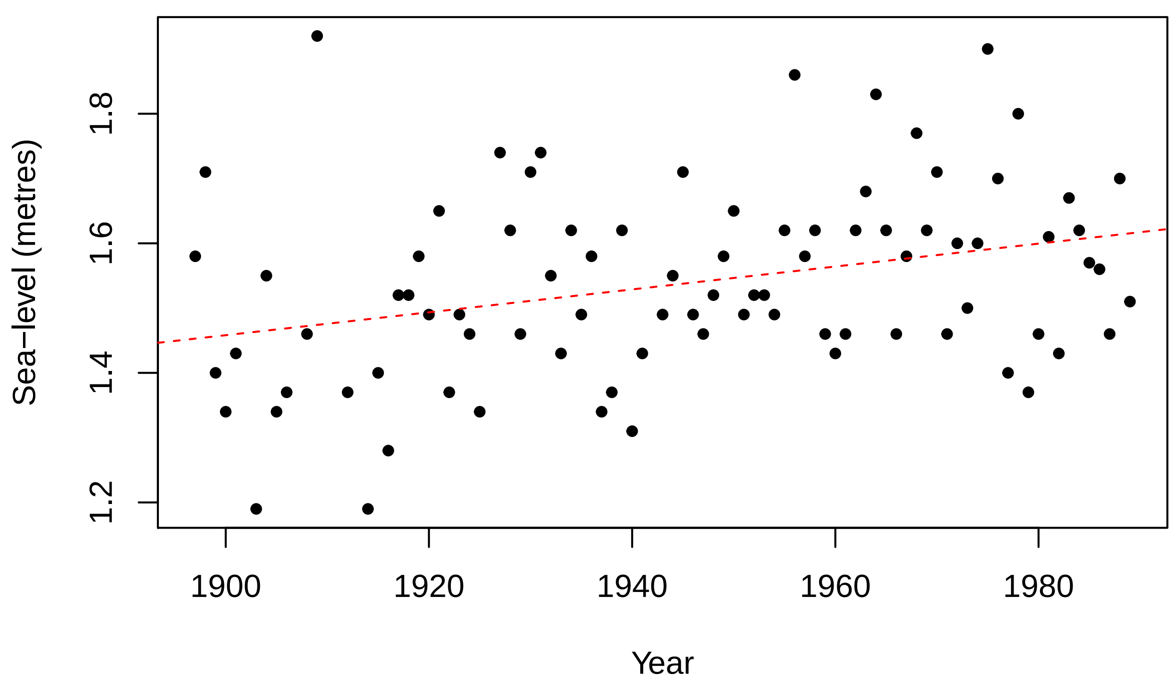
### Data

The fremantle data frame has 86 rows and 3 columns. The second column gives 86 annual maximimum sea levels recorded at Fremantle,Western Australia, within the period 1897 to 1989. The first column gives the corresponding years. The third column gives annual mean values of the Southern Oscillation Index (SOI), which is a proxy for meteorological volitility.

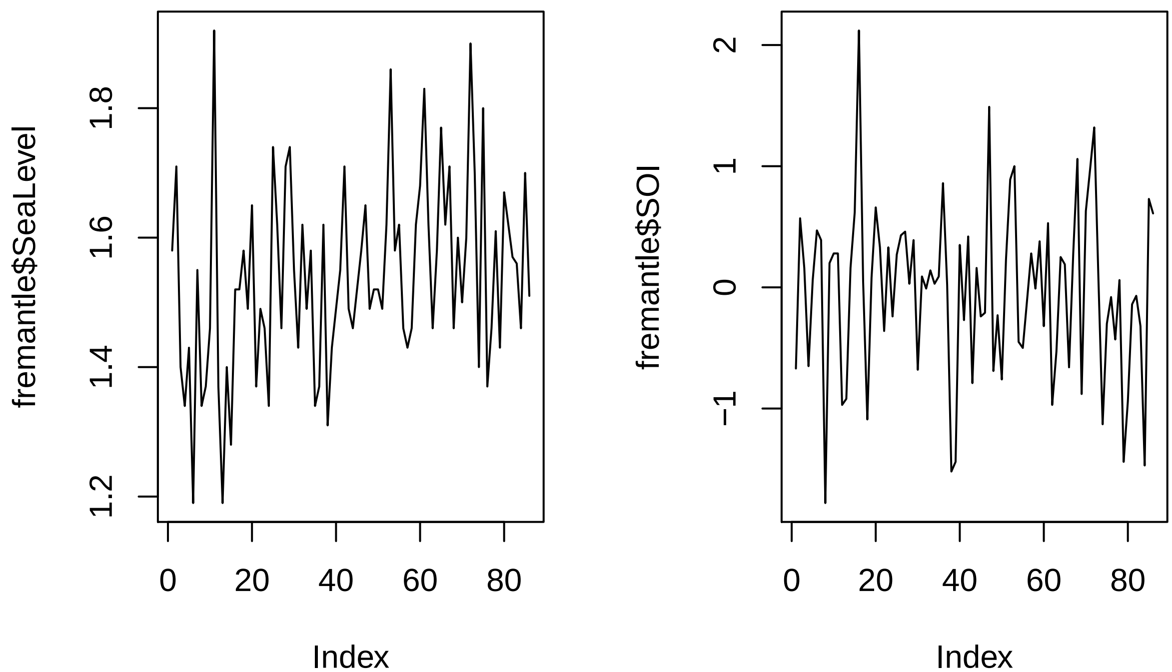
```
data("fremantle")
head(fremantle)
```

```
##   Year SeaLevel   SOI
## 1 1897     1.58 -0.67
## 2 1898     1.71  0.57
## 3 1899     1.40  0.16
## 4 1900     1.34 -0.65
## 5 1901     1.43  0.06
## 7 1903     1.19  0.47
```

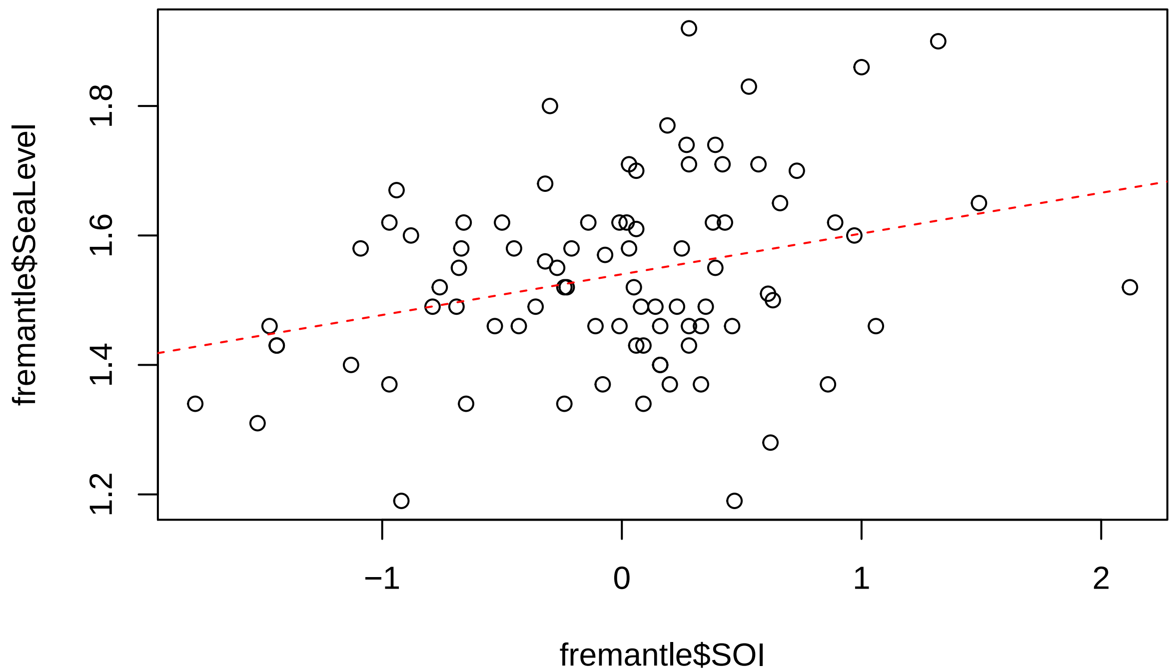
```
plot(x=fremantle$Year, fremantle$SeaLevel, pch=20,
       ylab="Sea-level (metres)", xlab="Year")
fit <- lm(SeaLevel ~ Year, data = fremantle)
abline(fit, col="red", lty=2)
```




```
par(mfrow=c(1,2))
plot(fremantle$SeaLevel, t="l")
plot(fremantle$SOI, t="l")
```




```
par(mfrow=c(1,1))
plot(fremantle$SOI, fremantle$SeaLevel)
abline(lm(SeaLevel ~ SOI, data = fremantle), col="red", lty=2)
```




### Covariates

```
n <- dim(fremantle)[1]

#Covariates
covar <- matrix(ncol = 3, nrow = n)
covar[,1] <- fremantle$SOI
covar[,2] <- seq(1, n, 1) # Linear trend
covar[,3] <- covar[,2]^2  # Quadratic trend

head(covar)
```

```
##       [,1] [,2] [,3]
## [1,] -0.67    1    1
## [2,]  0.57    2    4
## [3,]  0.16    3    9
## [4,] -0.65    4   16
## [5,]  0.06    5   25
## [6,]  0.47    6   36
```

### Model stationary

```
fitgev <- gev.fit(fremantle$SeaLevel, show = F)
fitgev$nllh
```

```
## [1] -43.56663
```

```r
fitgev$mle
```

```
## [1]  1.4823409  0.1412671 -0.2174320
```

```r
fitgev$se
```

```
## [1] 0.01672502 0.01149461 0.06377394
```

### Model with SOI

```r
fitgev.SOI <- gev.fit(fremantle$SeaLevel, ydat = covar, mul = 1, show = F)
fitgev.SOI$nllh
```

```
## [1] -47.21114
```

```r
fitgev.SOI$mle
```

```
## [1]  1.48985338  0.06188902  0.13960518 -0.26848380
```

```r
fitgev.SOI$se
```

```
## [1] 0.01655406 0.02315637 0.01150991 0.06399288
```

### Model with Linear time

```r
fitgev.ut <- gev.fit(fremantle$SeaLevel, ydat = covar, mul = 2, show = F)
fitgev.ut$nllh
```

```
## [1] -49.78972
```

```r
fitgev.ut$mle
```

```
## [1]  1.387186155  0.002140832  0.124716473 -0.128545018
```

```r
fitgev.ut$se
```

```
## [1] 0.0274796482 0.0005215259 0.0104146285 0.0679844086
```

### Model with quadratic time

```r
fitgev.ut2 <- gev.fit(fremantle$SeaLevel, ydat = covar, mul = c(2,3), show = F)
```

```
## Warning in sqrt(diag(z$cov)): NaNs produced
```

```r
fitgev.ut2$nllh
```

```
## [1] -50.95252
```

```r
fitgev.ut2$mle
```

```
## [1]  1.331932e+00  5.642570e-03 -3.921111e-05  1.208444e-01 -9.821101e-02
```

```r
fitgev.ut2$se
```

```
## [1] 0.0265635286 0.0005200551          NaN 0.0090983484 0.0060969134
```

### Model with SOI and Linear time

```r
fitgev.SOIut <- gev.fit(fremantle$SeaLevel, ydat = covar, mul = c(1, 2), show = F)
fitgev.SOIut$nllh
```

```
## [1] -53.8257
```

```r
fitgev.SOIut$mle
```

```
## [1]  1.389381297  0.055171074  0.002232467  0.121147089 -0.154480161
```

```r
fitgev.SOIut$se
```

```
## [1] 0.0272538644 0.0197789753 0.0005178779 0.0100390306 0.0636920071
```

### Model with linear time for sigma

```r
fitgev.st <- gev.fit(fremantle$SeaLevel, ydat = covar, sigl = 2, show = F)
fitgev.st$nllh
```

```
## [1] -44.67998
```

```r
fitgev.st$mle
```

```
## [1]  1.4920666018  0.1666643314 -0.0007118848 -0.1609604077
```

```r
fitgev.st$se
```

```
## [1] 1.587629e-02 1.122782e-02 1.999940e-06 7.249364e-02
```

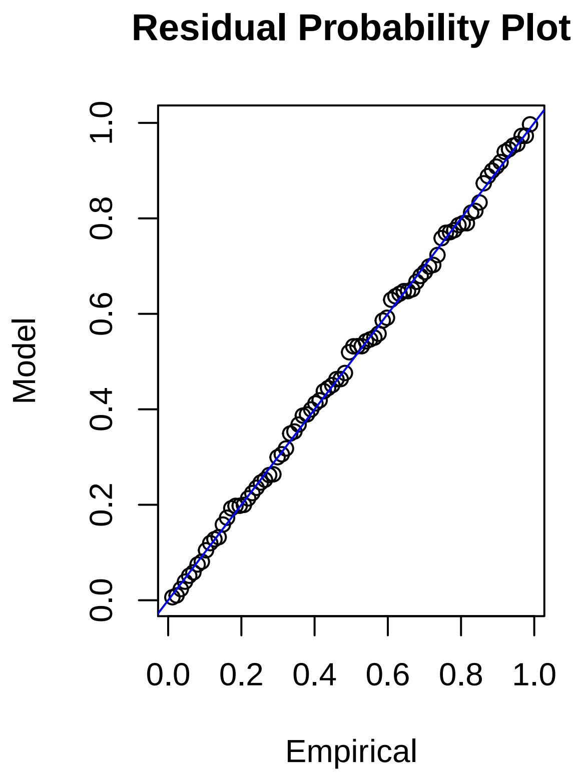
### Ratio test between stationary and linear trend

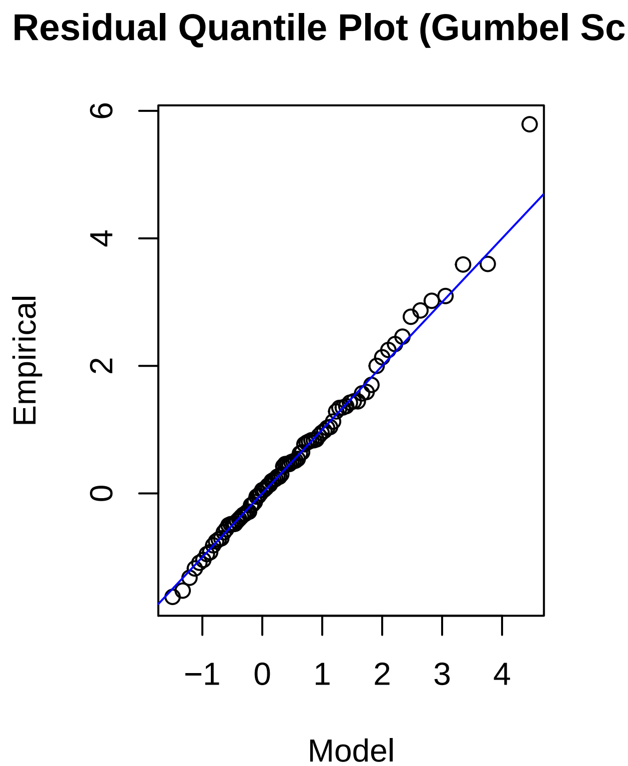
```r
2*(-fitgev.ut$nllh - -fitgev$nllh)
```

```
## [1] 12.44618
```

### Diag

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




## Sea level in Venice

### Data

The venice data frame has 51 rows and 11 columns. The final ten columns contain the 10 largest sea levels observed within the year given by the first column. The ten largest sea levels are given for every year in the period 1931 to 1981, excluding 1935 in which only the six largest measurements are available. SeaLevai is measured in cm.

```
library(ismev)
data("venice")
head(venice)
```

```
##   Year  r1  r2  r3  r4  r5 r6 r7 r8 r9 r10
## 1 1931 103  99  98  96  94 89 86 85 84  79
## 2 1932  78  78  74  73  73 72 71 70 70  69
## 3 1933 121 113 106 105 102 89 89 88 86  85
## 4 1934 116 113  91  91  91 89 88 88 86  81
## 5 1935 115 107 105 101  93 91 NA NA NA  NA
## 6 1936 147 106  93  90  87 87 87 84 82  81
```

### Covariates

```r
n <- dim(venice)[1]

#Covariates
covar <- matrix(ncol = 1, nrow = n)
covar[,1] <- seq(1, n, 1) # Linear trend

head(covar)
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
## [4,]    4
## [5,]    5
## [6,]    6
```

### Fit

### r=1 model

```r
venice.r1 <- venice$r1
fitgev.r1 <- gev.fit(venice.r1, show = F)
fitgev.r1.ut <- gev.fit(venice.r1, ydat = covar, mul = 1, show = F)
```

### r=5 model

```r
venice.r5 <- venice[,c(2:6)]
fitgev.r5 <- rlarg.fit(venice.r5, show = F)
fitgev.r5.ut <- rlarg.fit(venice.r5, ydat = covar, mul = 1, show = F)
```

### r=10 model

```r
venice.r10 <- venice[,-1]
fitgev.r10 <- rlarg.fit(venice.r10, show = F)
fitgev.r10.ut <- rlarg.fit(venice.r10, ydat = covar, mul = 1, show = F)
```

### Results for parameters

```r
gev.result <- function(gevfit){
  loglik <- -gevfit$nllh
  par.mu0 <- gevfit$mle[1]
  par.mu1 <- NaN
  par.sigma <- gevfit$mle[2]
  par.xi <- gevfit$mle[3]
  se.mu0 <- gevfit$se[1]
  se.mu1 <- NaN
  se.sigma <- gevfit$se[2]
  se.xi <- gevfit$se[3]

  result <- data.frame(loglik=loglik, mu0=par.mu0, se.mu0=se.mu0,
                       mu1=par.mu1, se.mu1=se.mu1,
```

```
                       sigma=par.sigma, se.sigma=se.sigma,
                       xi=par.xi, se.xi=se.xi)
  return(round(result, 3))
}

gev.ut.result <- function(gevfit){
  loglik <- -gevfit$nllh
  par.mu0 <- gevfit$mle[1]
  par.mu1 <- gevfit$mle[2]
  par.sigma <- gevfit$mle[3]
  par.xi <- gevfit$mle[4]
  se.mu0 <- gevfit$se[1]
  se.mu1 <- gevfit$se[2]
  se.sigma <- gevfit$se[3]
  se.xi <- gevfit$se[4]

  result <- data.frame(loglik=loglik, mu0=par.mu0, se.mu0=se.mu0,
                       mu1=par.mu1, se.mu1=se.mu1,
                       sigma=par.sigma, se.sigma=se.sigma,
                       xi=par.xi, se.xi=se.xi)
  return(round(result, 3))
}
```

**Stationary models**

```
rbind(gev.result(fitgev.r1), gev.result(fitgev.r5),gev.result(fitgev.r10))
```

```
##      loglik     mu0 se.mu0 mu1 se.mu1  sigma se.sigma     xi se.xi
## 1  -222.715 111.099  2.628 NaN    NaN 17.175    1.803 -0.077 0.074
## 2  -731.967 118.569  1.567 NaN    NaN 13.662    0.776 -0.088 0.033
## 3 -1139.090 120.548  1.362 NaN    NaN 12.784    0.549 -0.113 0.020
```

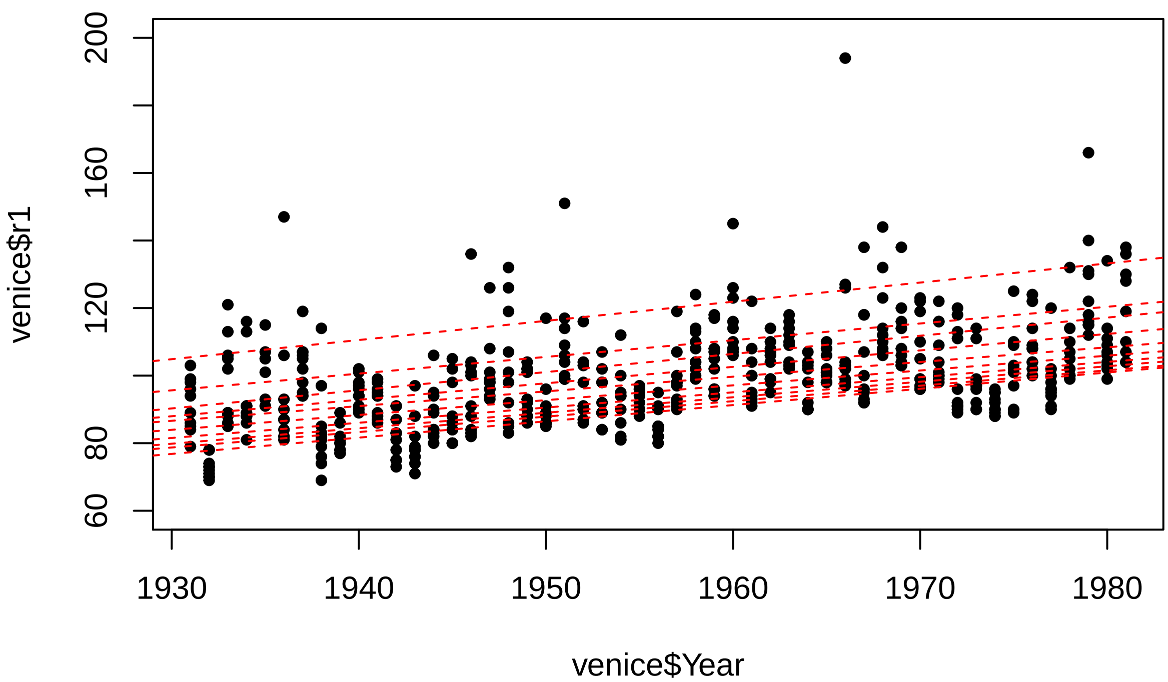
**Non-stationary models**

```
rbind(gev.ut.result(fitgev.r1.ut),
      gev.ut.result(fitgev.r5.ut),
      gev.ut.result(fitgev.r10.ut))
```

```
##      loglik     mu0 se.mu0   mu1 se.mu1  sigma se.sigma     xi se.xi
## 1  -216.063  96.986  4.249 0.564  0.139 14.584    1.578 -0.027 0.083
## 2  -704.760 104.233  2.038 0.458  0.055 12.290    0.805 -0.037 0.042
## 3 -1084.059 104.513  1.667 0.482  0.041 11.737    0.641 -0.065 0.028
```

**Plot with linear lines**

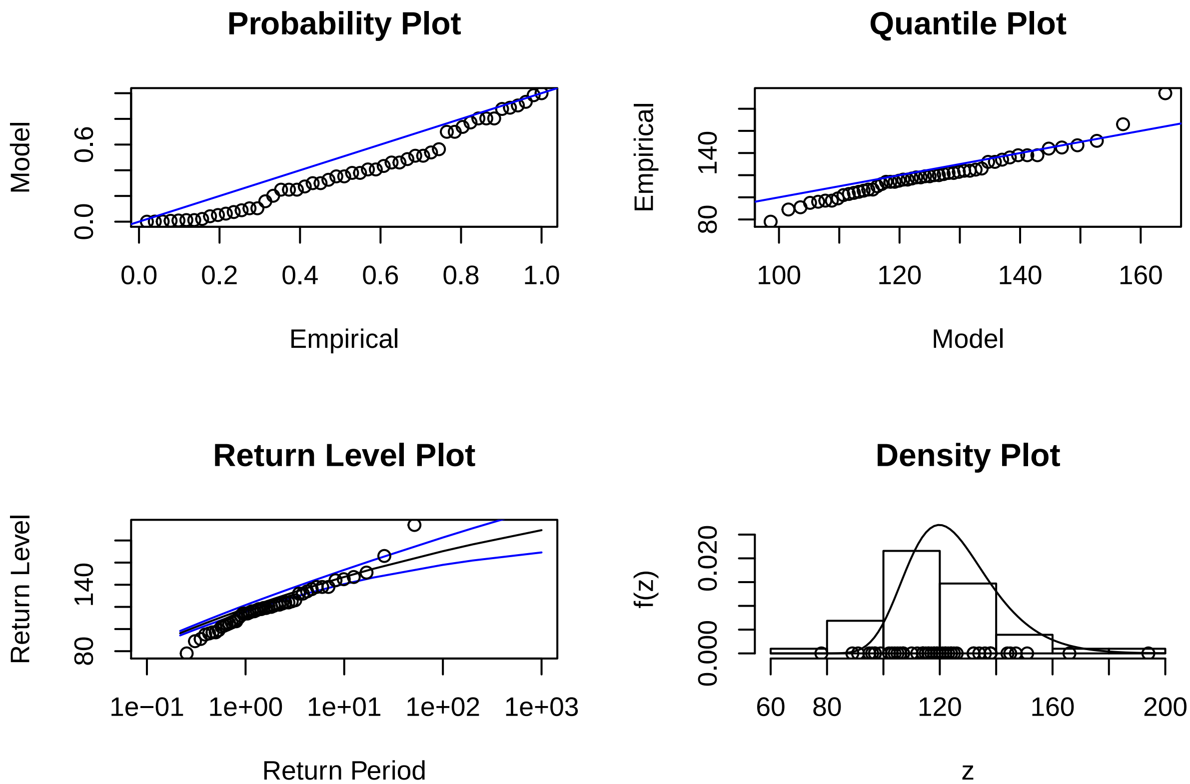
```
plot(x=venice$Year, y=venice$r1, pch=20, ylim=c(60,200))
for(i in 3:11){
  points(x=venice$Year, y=venice[,i], pch=20)
}
for(i in 2:11){
abline(lm(venice[,i] ~ venice$Year), col="red", lty=2)
}
```

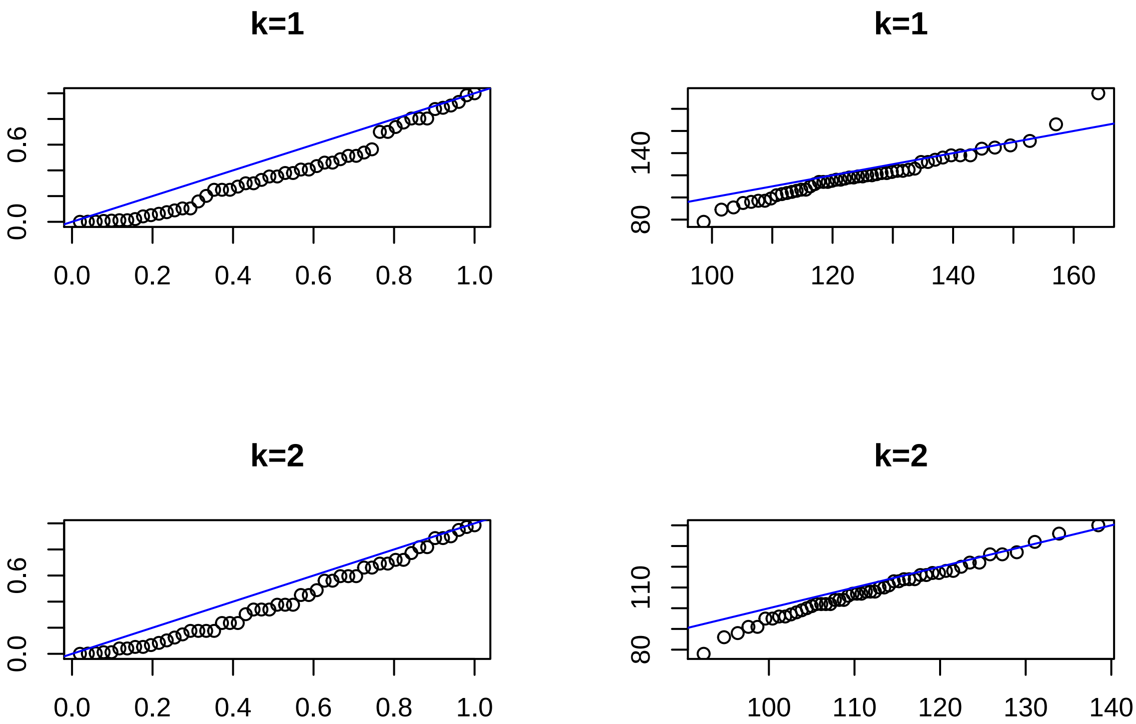


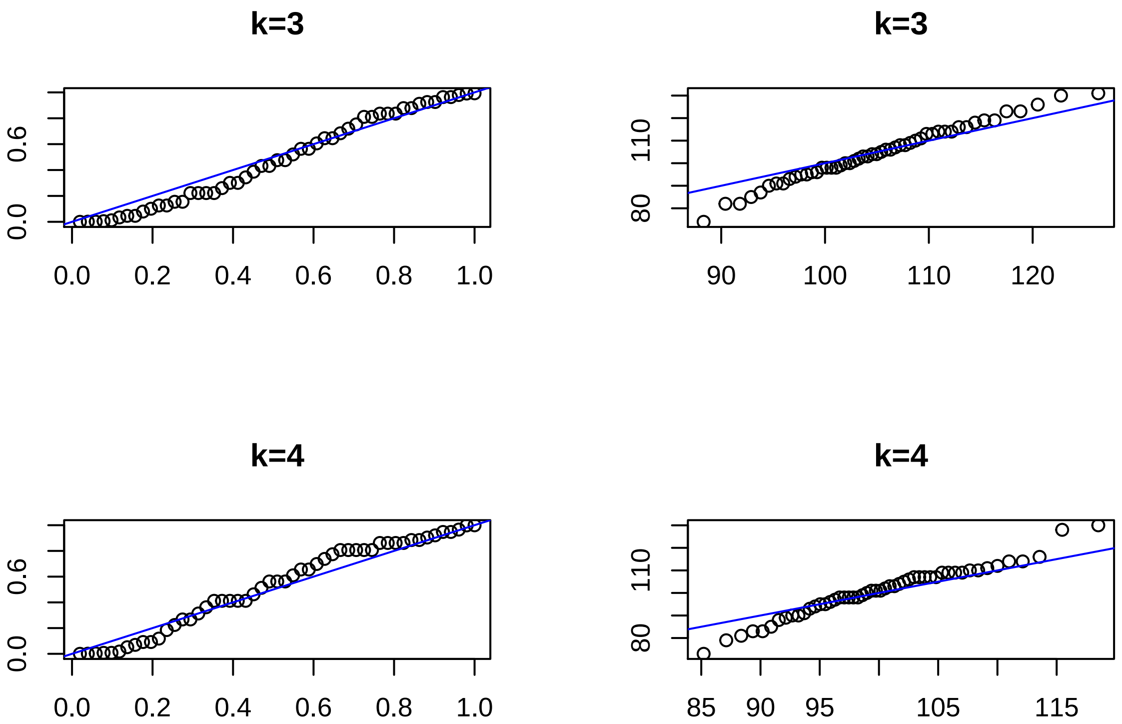

**Models check**

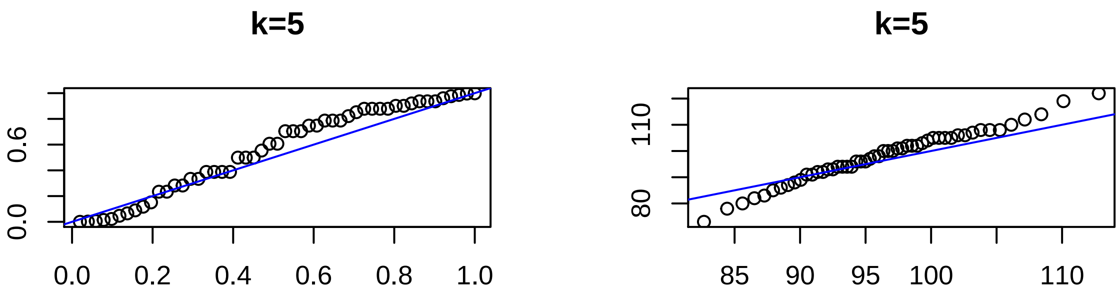
**Stationary model**

```
rlarg.diag(fitgev.r5)
```

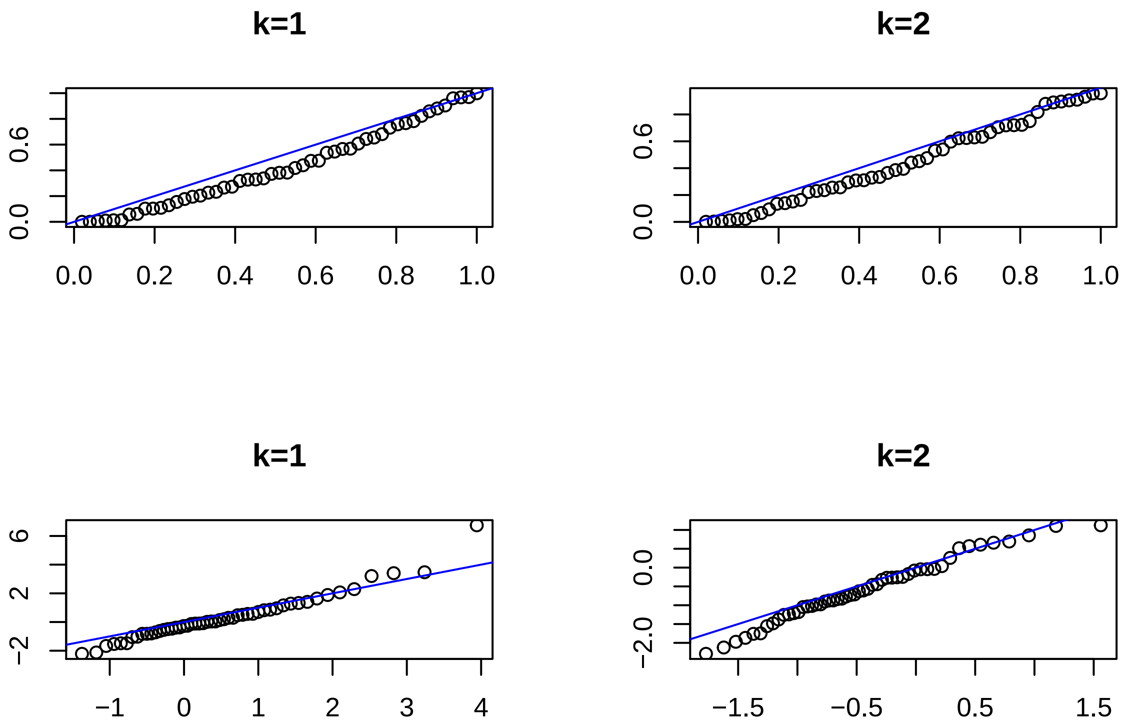

Probability Plot
Model
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Empirical
Quantile Plot
Empirical
80
140
100
120
140
160
Model
Return Level Plot
Return Level
80
140
1e−01
1e+00
1e+01
1e+02
1e+03
Return Period
Density Plot
f(z)
0.000
0.020
60
80
120
160
200
z

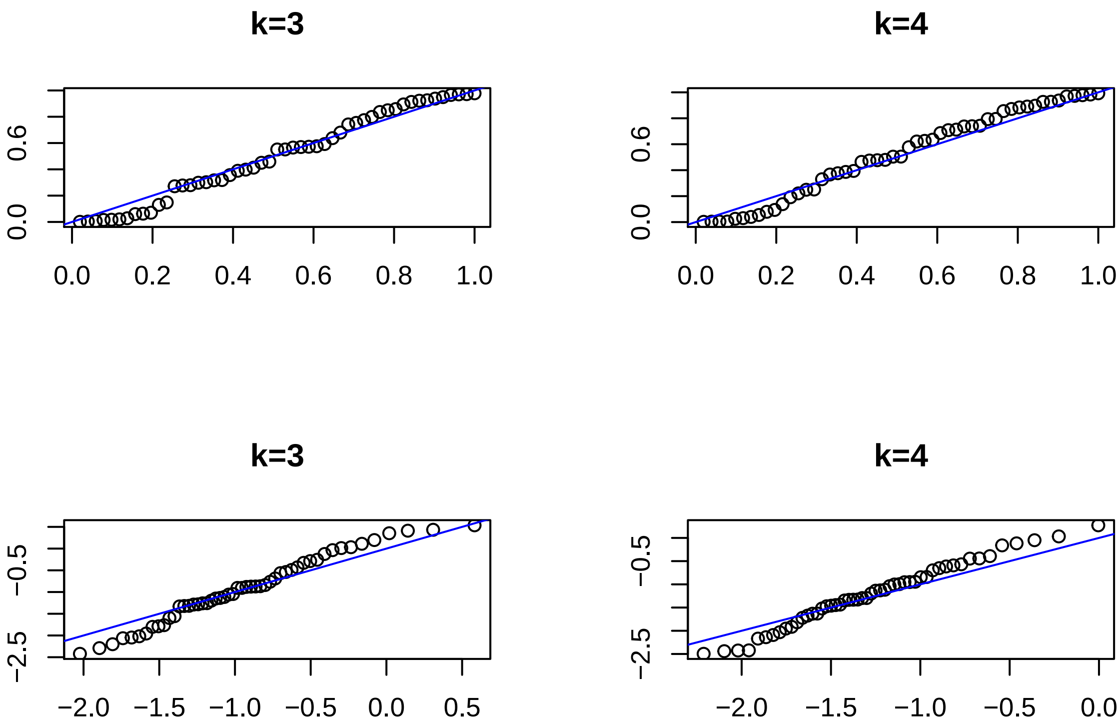

k=1
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
k=1
80
140
100
120
140
160
k=2
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
k=2
80
110
100
110
120
130
140

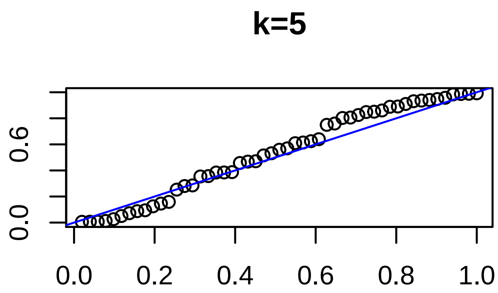

k=3
0.0
0.6
0.0 0.2 0.4 0.6 0.8 1.0
k=3
80
110
90 100 110 120
k=4
0.0
0.6
0.0 0.2 0.4 0.6 0.8 1.0
k=4
80
110
85 90 95 105 115

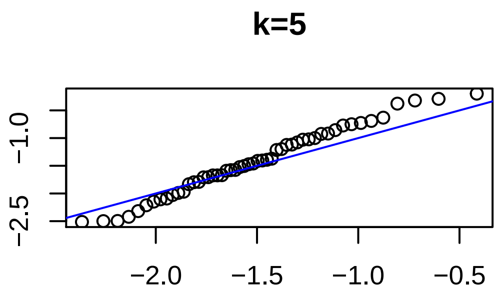



```r
rlarg.diag(fitgev.r5.ut)
```


k=1
k=2
0.0
0.6
0.0 0.2 0.4 0.6 0.8 1.0
0.0
0.6
0.0 0.2 0.4 0.6 0.8 1.0
k=1
k=2
−2
2
6
−1 0 1 2 3 4
−2.0
0.0
−1.5 −0.5 0.5 1.5


k=3
k=4
k=3
k=4
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
−2.5
−0.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5




## Daily rainfall

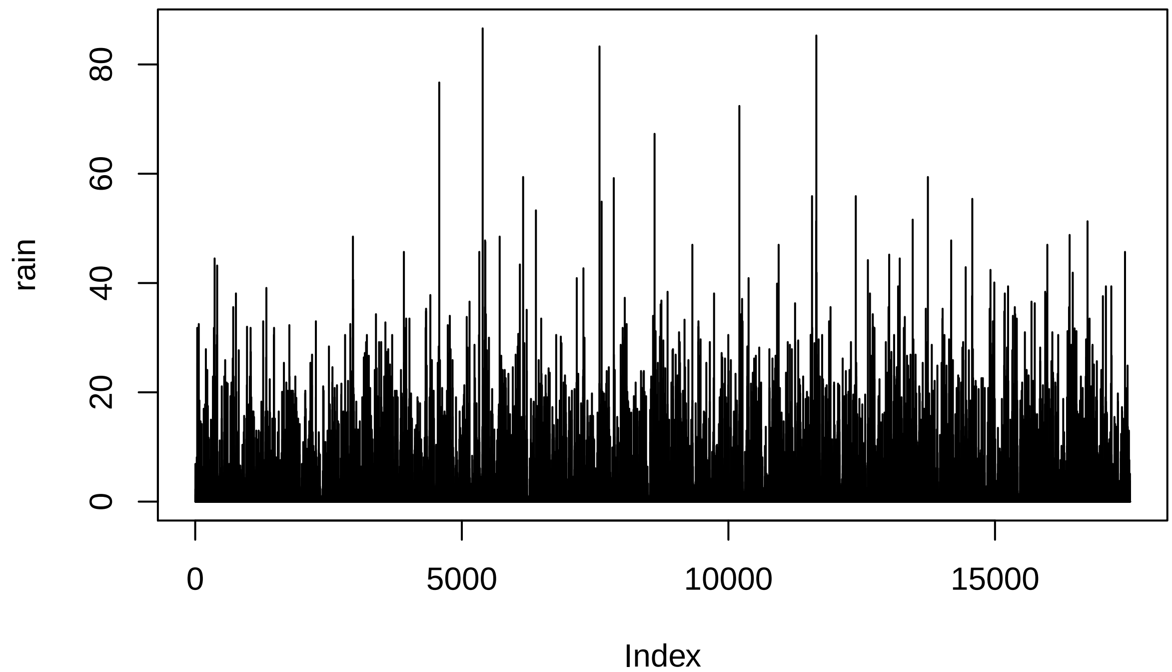
### Data

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")
head(rain)
```

```
## [1] 0.0 2.3 1.3 6.9 4.6 0.0
```

```r
plot(rain, t="h")
```




**Covariates**

```r
n <- length(rain)[1]

#Covariates
covar <- matrix(ncol = 1, nrow = n)
covar[,1] <- seq(1, n, 1) # Linear trend

head(covar,3)
```

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

**Fit with stationary model**

```r
fitgpd <- gpd.fit(rain, threshold = 30, show = F)
fitgpd$nllh
```

```
## [1] 485.0937
```

```r
fitgpd$mle
```

```
## [1] 7.4422639 0.1843027
```

**Fit with non-stationary model**

```
fitgpd.st <- gpd.fit(rain, 30, ydat = covar, sigl = 1, siglink = exp, show = F)
```

```
## Warning in sqrt(diag(z$cov)): NaNs produced
```

```
fitgpd.st$nllh
```

```
## [1] 484.6016
```

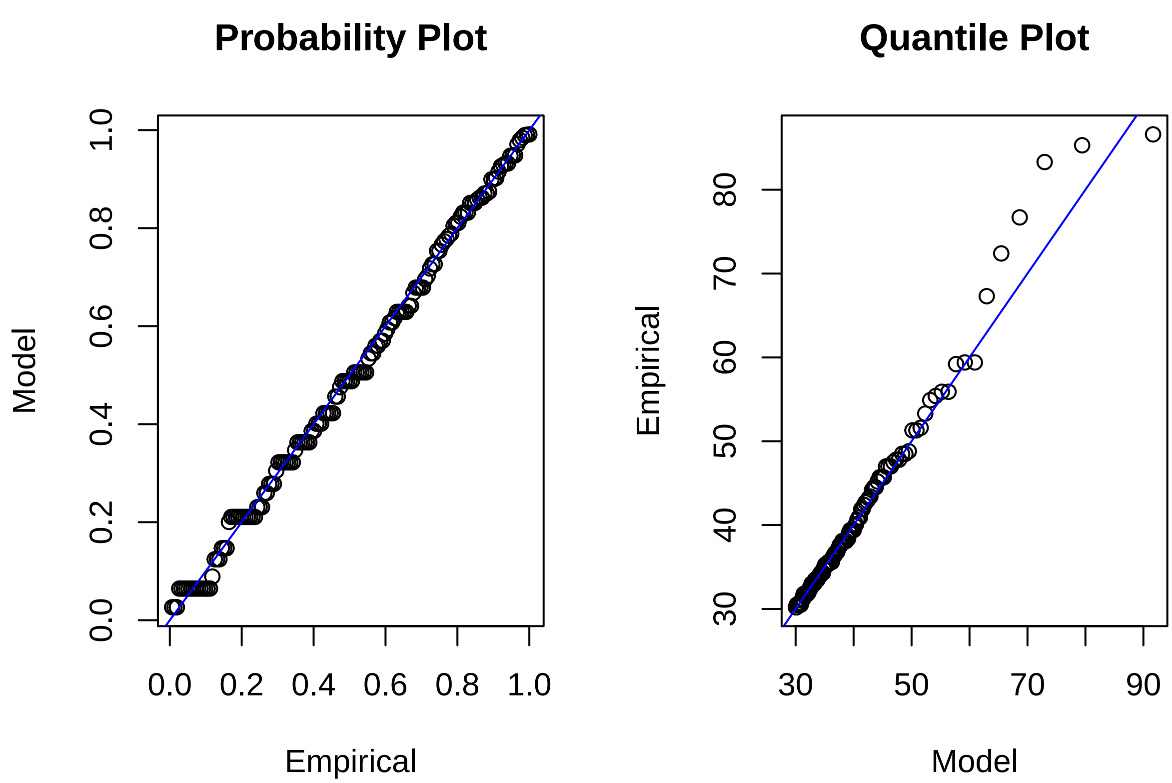
```
fitgpd.st$mle
```

```
## [1] 1.8039221883 0.0000196265 0.1977440505
```

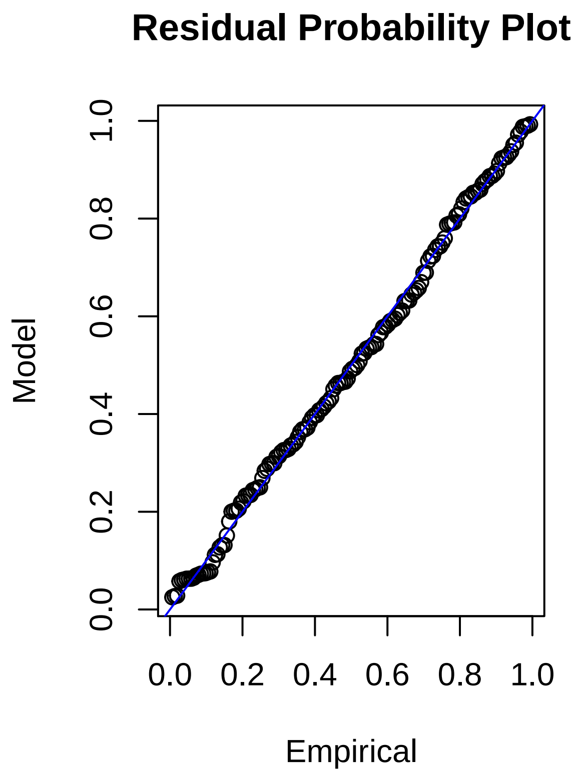
**Model check**

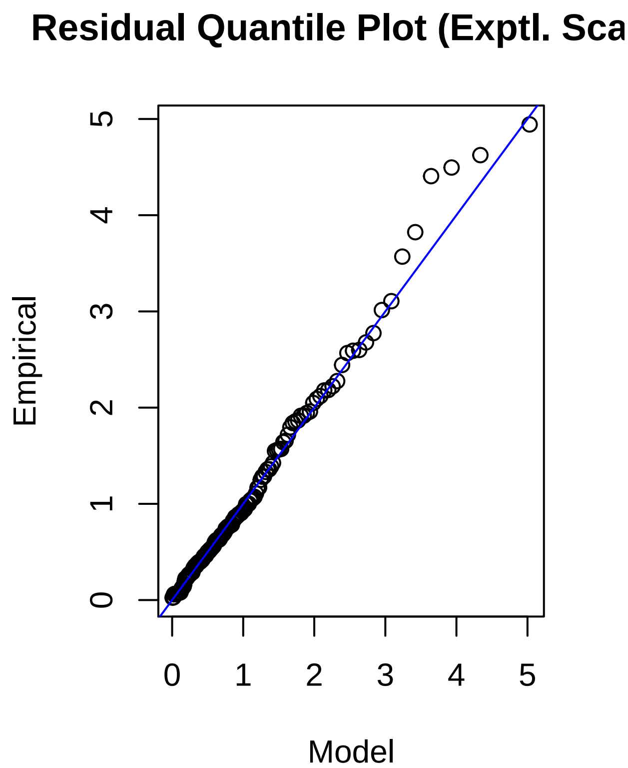
**Stationary model**

```
par(mfrow=c(1,2))
cgpd.pp(fitgpd); cgpd.qq(fitgpd)
```




```
gpd.diag(fitgpd.st)
```




### Wooster temperature

#### Data

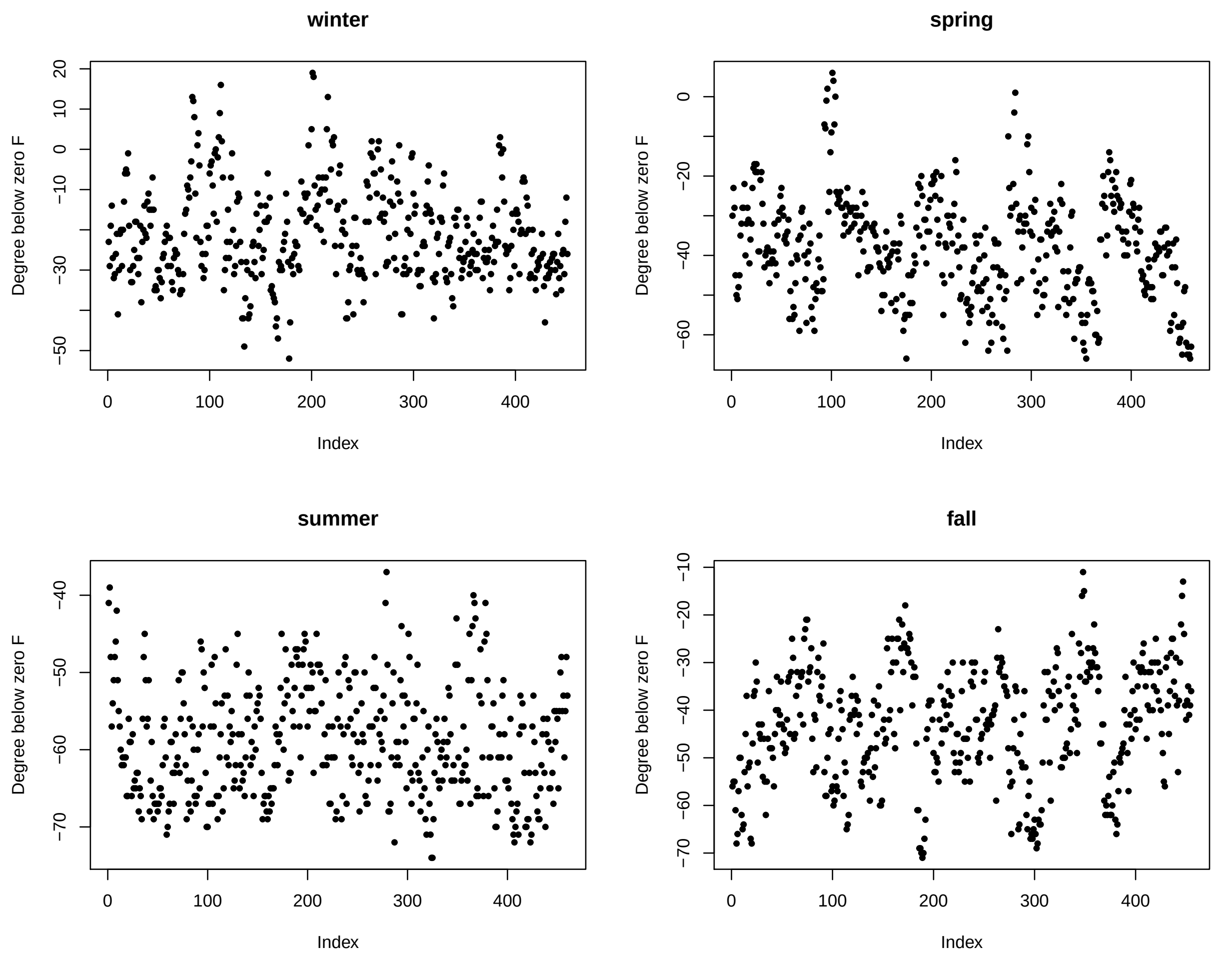
```
data("wooster")
wooster.dat <- data.frame(temp=-wooster)

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$month==4 |
                     wooster.dat$month==5 ] <- 1
wooster.dat$spring[is.na(wooster.dat$spring)] <- 0

wooster.dat$summer[wooster.dat$month==6 |
                     wooster.dat$month==7 |
```

```r
                     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
```

### Plot by season

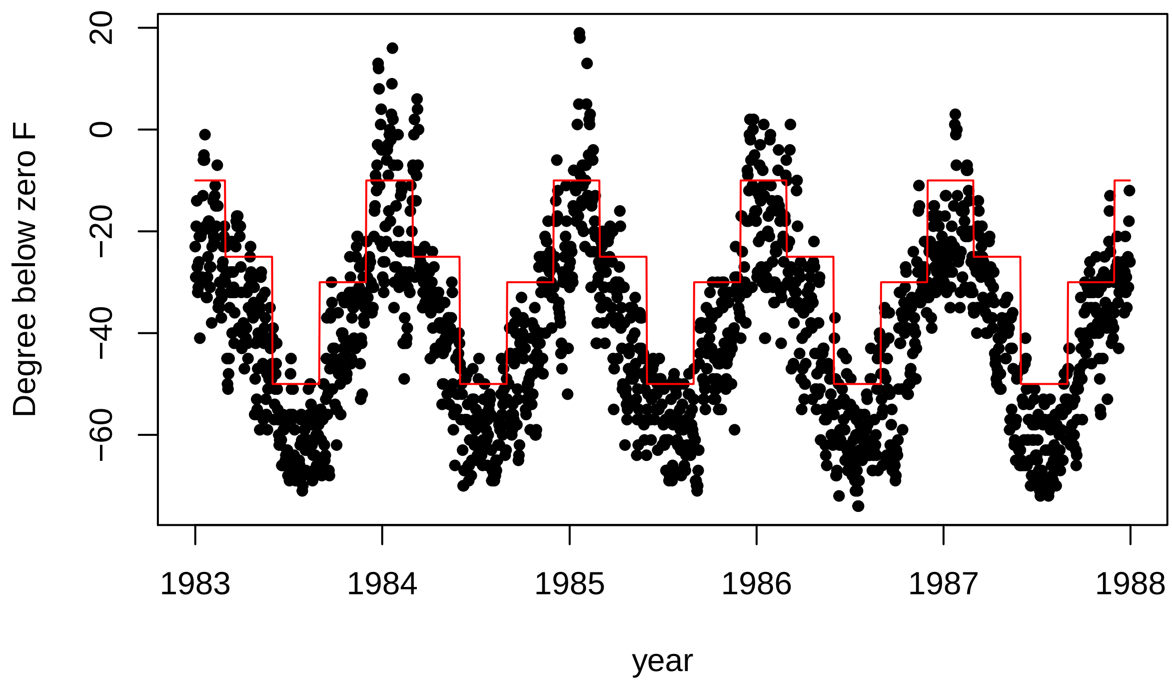
```r
par(mfrow=c(2,2))
plot(wooster.dat$temp[wooster.dat$winter==1], pch=20,
     ylab="Degree below zero F", main="winter")
plot(wooster.dat$temp[wooster.dat$spring==1], pch=20,
     ylab="Degree below zero F", main="spring")
plot(wooster.dat$temp[wooster.dat$summer==1], pch=20,
     ylab="Degree below zero F", main="summer")
plot(wooster.dat$temp[wooster.dat$fall==1], pch=20,
     ylab="Degree below zero F", main="fall")
```




### Seasonal threshold

```
wooster.dat$thres <- 0
wooster.dat$thres[wooster.dat$winter==1] <- -10
wooster.dat$thres[wooster.dat$spring==1] <- -25
wooster.dat$thres[wooster.dat$summer==1] <- -50
wooster.dat$thres[wooster.dat$fall==1] <- -30
```

```
plot(x=wooster.dat$dates, y=wooster.dat$temp, pch=20,
     ylab="Degree below zero F", xlab="year")
lines(x=wooster.dat$dates, y=wooster.dat$thres, col="red")
```




**Fit with stationary model**

```
fitgpd21 <- gpd.fit(wooster.dat$temp, threshold = -10)
```

```
## $threshold
## [1] -10
##
## $nexc
## [1] 85
##
## $conv
## [1] 0
##
## $nllh
## [1] 262.1725
##
## $mle
## [1] 10.6402671 -0.2802218
##
## $rate
## [1] 0.04654984
##
## $se
## [1] 1.48317853 0.09270039
```

### Non-stationary model

```r
fitgpd2 <- gpd.fit(wooster.dat$temp, threshold = wooster.dat$thres)
```

```
## $model
## $model[[1]]
## NULL
##
## $model[[2]]
## NULL
##
##
## $link
## [1] "c(identity, identity)"
##
## $nexc
## [1] 235
##
## $conv
## [1] 0
##
## $nllh
## [1] 684.7317
##
## $mle
## [1]  7.41154486 -0.08939142
##
## $rate
## [1] 0.1286966
##
## $se
## [1] 0.6664141 0.0621433
```

## Sea level in Venice (UNC)

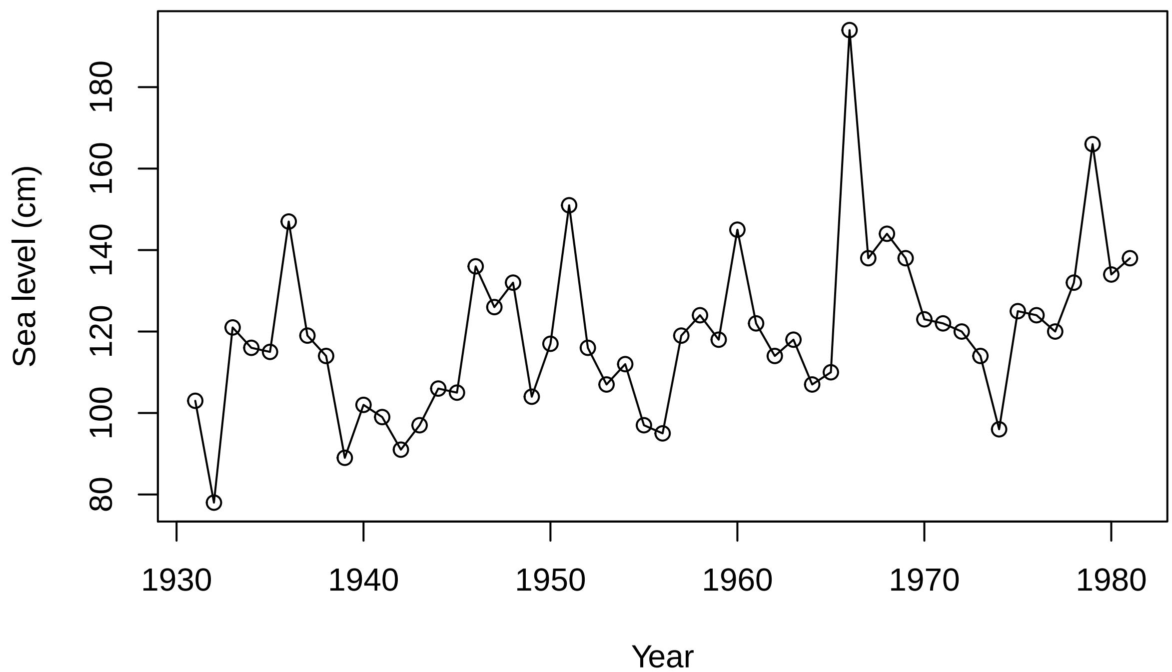
### Data

```r
library(ismev)
data("venice")
head(venice)
```

```
##   Year  r1  r2  r3  r4  r5 r6 r7 r8 r9 r10
## 1 1931 103  99  98  96  94 89 86 85 84  79
## 2 1932  78  78  74  73  73 72 71 70 70  69
## 3 1933 121 113 106 105 102 89 89 88 86  85
## 4 1934 116 113  91  91  91 89 88 88 86  81
## 5 1935 115 107 105 101  93 91 NA NA NA  NA
## 6 1936 147 106  93  90  87 87 87 84 82  81
```

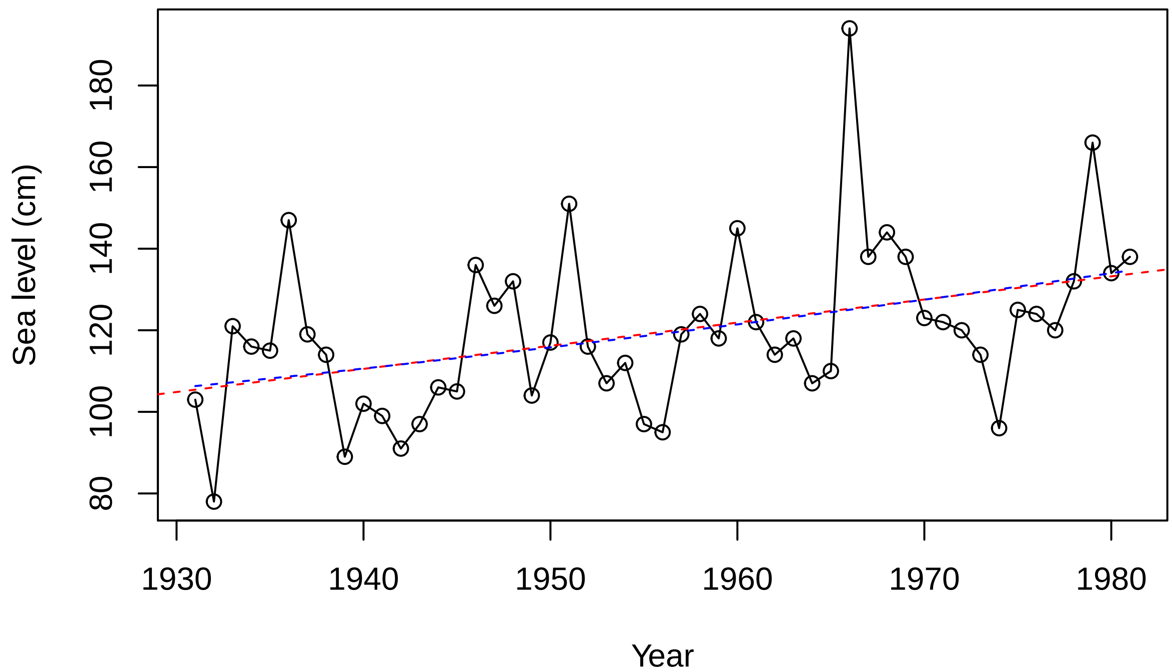
**Maxima**

```
venice.anmax <- venice[,c(1,2)]
plot(x=venice.anmax$Year, y=venice.anmax$r1, t="o",
     xlab="Year", ylab="Sea level (cm)")
```




**Trend analysis**

```
plot(x=venice.anmax$Year, y=venice.anmax$r1, t="o",
     xlab="Year", ylab="Sea level (cm)")

#Linear
fit <- lm(r1 ~ Year, data = venice.anmax)
abline(fit, col="red", lty=2)

#Quandratic
fit2 <- lm(r1 ~ poly(Year,2,raw=TRUE),data = venice.anmax)
curve(predict(fit2, newdata=data.frame(Year=x)), add=T, col="blue", lty=2)
```




```r
summary(fit)
```

```
##
## Call:
## lm(formula = r1 ~ Year, data = venice.anmax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.813 -11.211  -3.309   9.515  68.722
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -989.3822   346.4770  -2.856  0.00628 **
## Year           0.5670     0.1771   3.201  0.00241 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.62 on 49 degrees of freedom
## Multiple R-squared:  0.1729, Adjusted R-squared:  0.1561
## F-statistic: 10.25 on 1 and 49 DF,  p-value: 0.002406
```

```r
summary(fit2)
```

```
##
## Call:
## lm(formula = r1 ~ poly(Year, 2, raw = TRUE), data = venice.anmax)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.047 -10.865  -3.322   9.067  68.976
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.334e+03  5.202e+04   0.141    0.888
## poly(Year, 2, raw = TRUE)1 -7.944e+00  5.319e+01  -0.149    0.882
## poly(Year, 2, raw = TRUE)2  2.176e-03  1.360e-02   0.160    0.874
##
## Residual standard error: 18.81 on 48 degrees of freedom
## Multiple R-squared:  0.1734, Adjusted R-squared:  0.1389
## F-statistic: 5.034 on 2 and 48 DF,  p-value: 0.01036
```

### GEV

#### Stationary

```r
fitgev <- gev.fit(venice.anmax$r1)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 222.7145
##
## $mle
## [1] 111.09925486  17.17548761  -0.07673265
##
## $se
## [1] 2.6280070 1.8033672 0.0735214
```

#### Linear trend for location

```r
#Time index
ti <- matrix(ncol=1,nrow=length(venice.anmax$Year))
ti[,1] <- seq(1,length(venice.anmax$Year),1)

#Fit
fitgev.ut <- gev.fit(venice.anmax$r1, ydat = ti, mul = 1)
```

```
## $model
## $model[[1]]
## [1] 1
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
```

```
##
## $conv
## [1] 0
##
## $nllh
## [1] 216.0626
##
## $mle
## [1] 96.98579330  0.56414269 14.58435088 -0.02731421
##
## $se
## [1] 4.24930969 0.13948421 1.57840034 0.08270996
```

### Quandratic trend for location

```r
#Time index
ti2 <- matrix(ncol = 2, nrow = length(venice.anmax$Year))
ti2[,1] <- seq(1,length(venice.anmax$Year),1)
ti2[,2] <- (ti2[,1])^2

#Fit
fitgev.ut2 <- gev.fit(venice.anmax$r1, ydat = ti2, mul = c(1,2))
```

```
## $model
## $model[[1]]
## [1] 1 2
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
## [1] 216.0555
##
## $mle
## [1] 96.385006916  0.632499970 -0.001340599 14.564915436 -0.025471589
##
## $se
## [1] 6.64848069 0.59400203 0.01130770 1.59752694 0.08578017
```

### Model selection

### Linear and Stationary

```r
2*(-fitgev.ut$nllh - -fitgev$nllh)
```

```
## [1] 13.30386
```

```r
as.numeric(pchisq(-fitgev.ut$nllh - -fitgev$nllh, df=1, lower.tail=FALSE))
```

```
## [1] 0.009904848
```

fitgev.ut is better

#### Linear and Quadratic

```r
2*(-fitgev.ut2$nllh - -fitgev.ut$nllh)
```

```
## [1] 0.01417277
```

```r
as.numeric(pchisq(-fitgev.ut2$nllh - -fitgev.ut$nllh, df=1, lower.tail=FALSE))
```

```
## [1] 0.9329128
```

No difference -> choose linear model

#### Model diagnostics

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