

A User's Guide to the POT Package (Version 1.0)

Mathieu Ribatet

Copyright ©2006

Department of Hydrological Statistic, INRS,
University of Québec, 490 de la Couronne, G1K 9A9 Canada

Cemagref UR HH, 3bis quai Chauveau
69336 Lyon Cedex 09 France

E-mail: ribatet@hotmail.com

7th August 2006

1 Introduction

1.1 Why the POT package?

The **POT** package is an add-on package for the R statistical software (R Development Core Team, 2006). The main goal of this package is to develop tools to perform statistical analyses of Peaks Over a Threshold (**POT**).

Most of functions are related to the Extreme Value Theory (**EVT**). Coles (2001) gives a comprehensive introduction to the EVT, while Kluppelberg and Mikosch (1997) present advanced results.

1.2 Obtaining the package/guide

The package can be downloaded from CRAN (The Comprehensive R Archive Network) at <http://cran.r-project.org/>. This guide (in pdf) will be in the directory **POT/doc/** underneath wherever the package is installed.

1.3 Contents

To help users to use properly the **POT** package, this guide contains practical examples on the use of this package. Section 2 introduce quickly the Extreme Value Theory (**EVT**). Some basic examples are described in section 3, while section 4 gives a concrete statistical analysis of extreme value for river Adieères at Beaujeu (FRANCE).

1.4 Citing the package/guide

To cite this guide or the package in publications please use the following bibliographic database entry.

```
@Manual{key,  
  title = {A User's Guide to the POT Package (Version 1.0)},  
  author = {Ribatet, M. A.},  
  year = {2006},  
  month = {August},  
  url = {http://cran.r-project.org/}  
}
```

1.5 Caveat

I have checked these functions as best I can but, as ever, they may contain bugs. If you find a bug or suspected bug in the code or the documentation please report it to me at ribatet@hotmail.com. Please include an appropriate subject line.

1.6 Legalese

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

A copy of the GNU General Public License can be obtained from <http://www.gnu.org/copyleft/gpl.html>. You can also obtain it by writing to the Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA.

2 An Introduction the EVT

Even if this package is only related to peaks over a threshold, a classical introduction to the EVT must deal with “block maxima”. Let X_1, \dots, X_n be a series of independent and identically distributed random variables with common distribution function F . Let $M_n = \max(X_1, \dots, X_n)$.

Suppose there exists normalizing constants $a_n > 0$ and b_n such that:

$$\Pr \left[\frac{M_n - b_n}{a_n} \leq y \right] = F^n(a_n y + b_n) \longrightarrow G(y), \quad n \rightarrow +\infty \quad (2.1)$$

for all $y \in \mathbb{R}$, where G is a non-degenerate distribution function. According to the Extremal Types Theorem (Fisher and Tippett, 1928), G must be either Fréchet, Gumbel or negative Weibull. Jenkinson (1955) noted that these three distributions can be merged into a single parametric family: the Generalized Extreme Value (**GEV**) distribution. The GEV has a distribution function defined by:

$$G(y) = \exp \left[- \left(1 + \xi \frac{y - \mu}{\sigma} \right)_+^{-1/\xi} \right], \quad (2.2)$$

where (μ, σ, ξ) are the location, scale and shape parameters respectively, $\sigma > 0$ and $z_+ = \max(z, 0)$.

The Fréchet case is obtained when $\xi > 0$, the negative Weibull when $\xi < 0$ while the Gumbel case is defined by continuity when $\xi \rightarrow 0$.

From this result, Pickands (1975) showed that the limiting distribution of normalized excesses of a threshold μ as the threshold approaches the endpoint μ_{end} of the variable of interest is the Generalized Pareto Distribution (**GPD**). That is, if X is a random variable which holds (2.1), then:

$$\Pr [X \leq y | X > \mu] \longrightarrow H(y), \quad \mu \rightarrow \mu_{\text{end}} \quad (2.3)$$

with

$$H(y) = 1 - \left(1 + \xi \frac{y - \mu}{\sigma} \right)_+^{-1/\xi}, \quad (2.4)$$

where (μ, σ, ξ) are the location, scale and shape parameters respectively, $\sigma > 0$ and $z_+ = \max(z, 0)$. Note that the Exponential distribution is obtained by continuity as $\xi \rightarrow 0$.

In practice, these two asymptotical results motivated modelling block maxima with a GEV, while peaks over threshold with a GPD.

3 Basic Use

3.1 Random Numbers and Distribution Functions

First of all, let's start with basic stuffs. The **POT** package uses the R convention for random numbers generation and distribution function features.

```
> ##Random number generation
> rgpd(5, loc = 1, scale = 2, shape = -0.2)
[1] 4.672547 2.365295 1.899087 1.577886 2.409450
> ##Varying threshold can be performed also
> rgpd(6, c(1, -5), 2, -0.2)
[1] 2.424368 -3.389774 3.965086 -3.332016 4.707819 -4.985408
> ##The same but with a varying scale parameter
> rgpd(6, 0, c(2, 3), 0)
[1] 2.9850740 3.1486256 1.0705649 0.7401753 3.1231517 2.3994109
> ##Probability of non exceedence
> pgpd(c(9, 15, 20), 1, 2, 0.25)
[1] 0.9375000 0.9825149 0.9922927
> ##Quantile associated to probability of non exceedence
> qgpd(c(.25, .5, .75), 1, 2, 0)
[1] 1.575364 2.386294 3.772589
> ##Evaluate the density at point...
> dgpdc(c(9, 15, 20), 1, 2, 0.25)
[1] 0.015625000 0.003179117 0.001141829
```

Several options can be passed to three of these four functions. In particular:

- for “pgpd”, user can specify if non exceedence or exceedence probability should be computed with option `lower.tail = TRUE` or `lower.tail = FALSE` respectively;
- for “qgpd”, user can specify if quantile is related to non exceedence or exceedence probability with option `lower.tail = TRUE` or `lower.tail = FALSE` respectively;
- for “dgpdc”, user can specify if the density or the log-density should be computed with option `log = FALSE` or `log = TRUE` respectively.

3.2 Threshold Selection

The location for the GPD or equivalently the threshold is a particular parameter as must often it is not estimated as the other ones. All methods to define a suitable threshold use the asymptotic approximation defined by equation (2.3). In other words, we select a threshold for which the asymptotic distribution H in equation (2.4) is a good approximation.

The **POT** package has several tools to define a reasonable threshold. For this purpose, the user must use `tcplot`, `mrlplot`, `lmomplot`, `explot` and `diplot` functions.

The main goal of threshold selection is to select enough events to reduce the variance; but not too much as we could select events coming from the central part of the distribution¹ and induce bias.

3.2.1 Threshold Choice plot: *tcplot*

Let $X \sim GP(\mu_0, \sigma_0, \xi_0)$. Let μ_1 be another threshold as $\mu_1 > \mu_0$. The random variable $X|X > \mu_1$ is also GPD with updated parameters $\sigma_1 = \sigma_0 + \xi_0(\mu_1 - \mu_0)$ and $\xi_1 = \xi_0$. Let

$$\sigma_* = \sigma_1 - \xi_1 \mu_1 \quad (3.1)$$

With this new parametrization, σ_* is independent of μ_1 . Thus, estimates of σ_* and ξ_1 are constant for all $\mu_1 > \mu_0$ if μ_0 is a suitable threshold for the asymptotic approximation.

Threshold choice plots represent the points defined by:

$$\{(\mu_1, \sigma_*) : \mu_1 \leq x_{\max}\} \quad \text{and} \quad \{(\mu_1, \xi_1) : \mu_1 \leq x_{\max}\} \quad (3.2)$$

where x_{\max} is the maximum of the observations \mathbf{x} .

Moreover, confidence intervals can be computed using Fisher information.

Here is an application.

```
> x <- runif(10000)
> par(mfrow=c(1,2))
> tcplot(x, tlim = c(0.9, 0.995))
```

Results of the *tcplot* function is displayed in Figure 1. We can see clearly that a threshold around 0.98 is a reasonable choice. However, in practice decision are not so clear-cut as for this synthetic example.

3.2.2 Mean Residual Life Plot: *mrlplot*

The **mean residual life plot** is based on the theoretical mean of the GPD. Let X be a *r.v.* distributed as $GPD(\mu, \sigma, \xi)$. Then, theoretically we have:

$$\mathbb{E}[X] = \mu + \frac{\sigma}{1 - \xi}, \quad \text{for } \xi < 1 \quad (3.3)$$

When $\xi \geq 1$, the theoretical mean is infinite.

In practice, if X represents excess over a threshold μ_0 , and if the approximation by a GPD is good enough, we have:

$$\mathbb{E}[X - \mu_0 | X > \mu_0] = \frac{\sigma_{\mu_0}}{1 - \xi} \quad (3.4)$$

For all new threshold μ_1 such as $\mu_1 > \mu_0$, excesses above the new threshold are also approximate by a GPD with updated parameters - see section 3.2.1. Thus,

$$\mathbb{E}[X - \mu_1 | X > \mu_1] = \frac{\sigma_{\mu_1}}{1 - \xi} = \frac{\sigma_{\mu_0} + \xi \mu_1}{1 - \xi} \quad (3.5)$$

The quantity $\mathbb{E}[X - \mu_1 | X > \mu_1]$ is linear in μ_1 . Or, $\mathbb{E}[X - \mu_1 | X > \mu_1]$ is simply the mean of excesses above the threshold μ_1 which can easily be estimated using the empirical mean.

¹i.e. not extreme events.

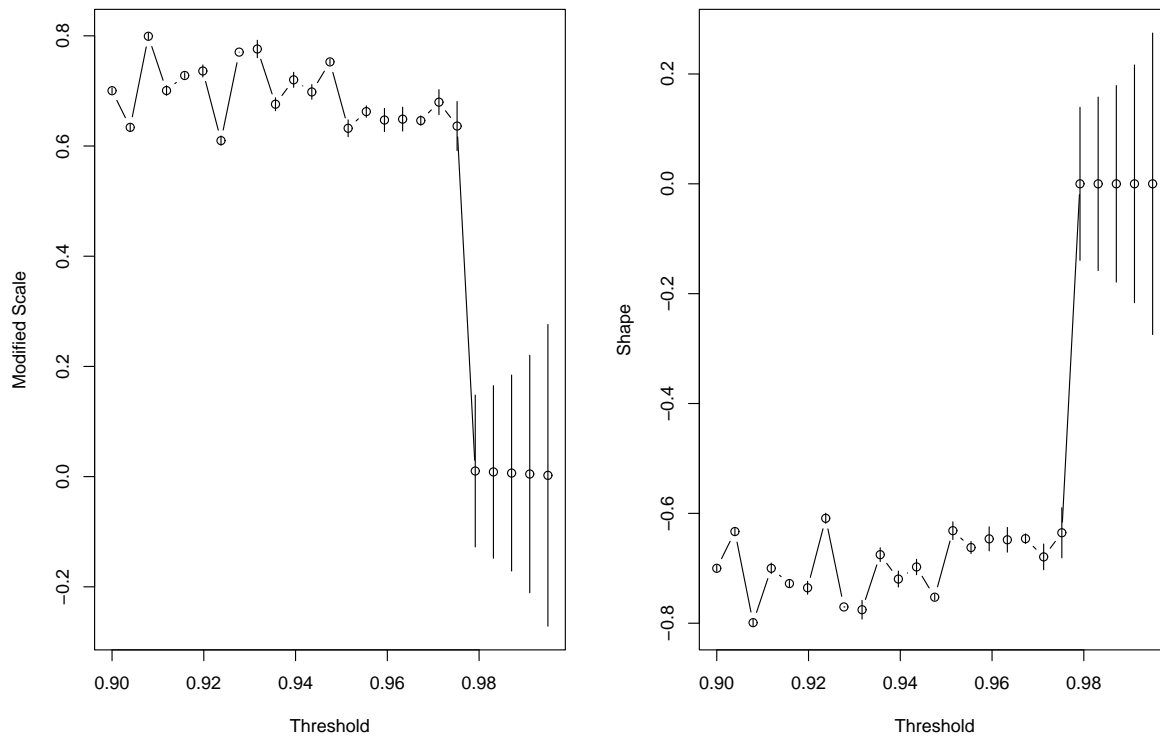


Figure 1: Threshold Choice plot on synthetic data

A mean residual life plot consists in representing points:

$$\left\{ \left(\mu, \frac{1}{n_\mu} \sum_{i=1}^{n_\mu} x_{i,n_\mu} - \mu \right) : \mu \leq x_{\max} \right\} \quad (3.6)$$

where n_μ is the number of observations \mathbf{x} above the threshold μ , x_{i,n_μ} is the i -th observation above the threshold μ and x_{\max} is the maximum of the observations \mathbf{x} .

Confidence intervals can be added to this plot as the empirical mean can be supposed to be normally distributed (Central Limit Theorem). However, normality doesn't hold anymore for high threshold as there are less and less excesses. Moreover, by construction, this plot always converge to the point $(x_{\max}, 0)$.

Here is another synthetic example.

```
> x <- rnorm(10000)
  mrlplot(x, tlim = c(1, 3.5), col = c("green", "black", "green"))
```

Figure 2 displays the mean residual life plot. A threshold around 2.5 should be reasonable.

3.2.3 L-Moments plot: *lmomplot*

L-moments are summary statistics for probability distributions and data samples. They are analogous to ordinary moments – they provide measures of location, dispersion, skewness, kurtosis, and other aspects of the shape of probability distributions or data samples – but are computed from linear combinations of the ordered data values (hence the prefix L).

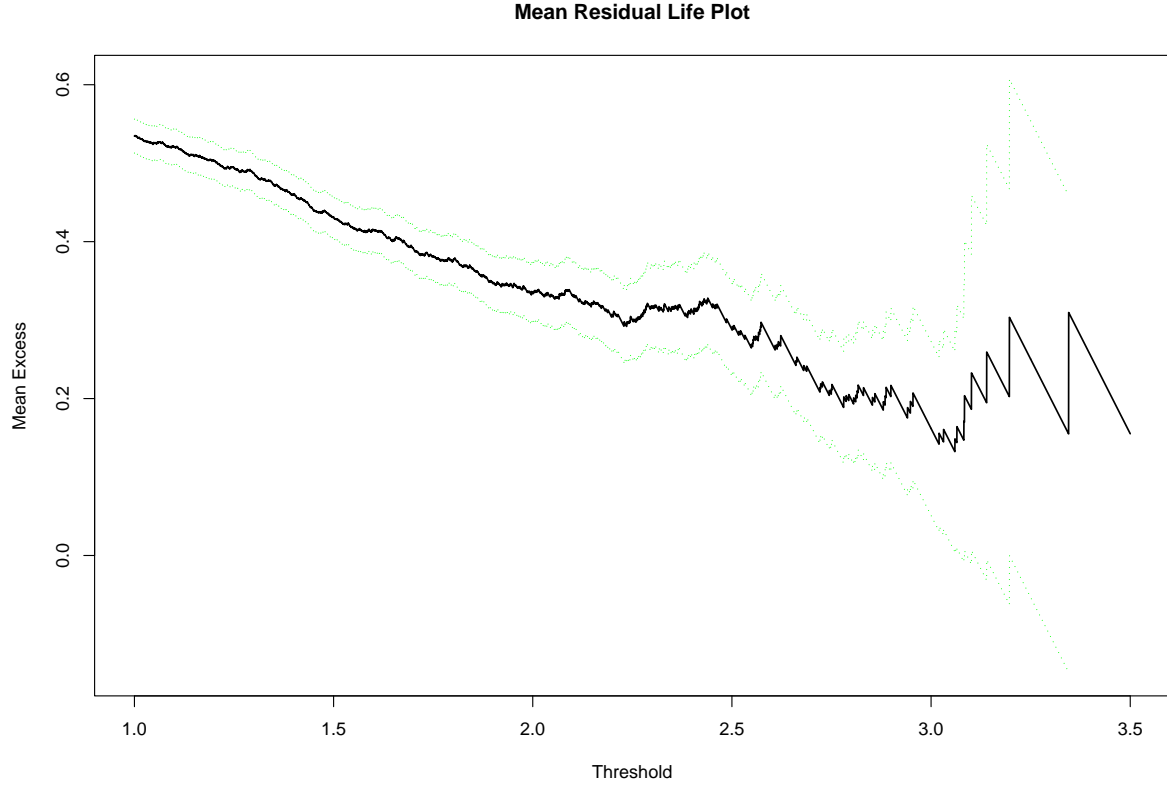


Figure 2: Mean residual life plot on synthetic data

For the GPD, the following relation holds:

$$\tau_4 = \tau_3 \frac{1 + 5\tau_3}{5 + \tau_3} \quad (3.7)$$

where τ_4 is the **L-Kurtosis** and τ_3 is the **L-Skewness**.

The **L-Moment** plot represents points defined by:

$$\{(\hat{\tau}_{3,u}, \hat{\tau}_{4,u}) : u \leq x_{\max}\} \quad (3.8)$$

where $\hat{\tau}_{3,u}$ and $\hat{\tau}_{4,u}$ are estimations of the L-Kurtosis and L-Skewness based on excesses over threshold u and x_{\max} is the maximum of the observations \mathbf{x} . The theoretical curve defined by equation (3.7) is traced as a guideline.

Here is a trivial example.

```
> x <- c(1 - abs(rnorm(200, 0, 0.2)), rgpd(100, 1, 2, 0.25))
> lmomplot(x, tlim = c(0.9, 2), identify = FALSE)
```

Figure 3 displays the L-Moment plot. By passing option `identify = TRUE` user can click on the graphic to identify the threshold related to the point selected.

We found that this graphic has often poor performance on real data.

3.2.4 Dispersion Index Plot: *diplot*

The **Dispersion Index plot** is particularly useful when dealing with time series. The EVT states that excesses over a threshold can be approximated by a GPD. However, the EVT also states that the occurrences of these excesses must be represented by a Poisson process.

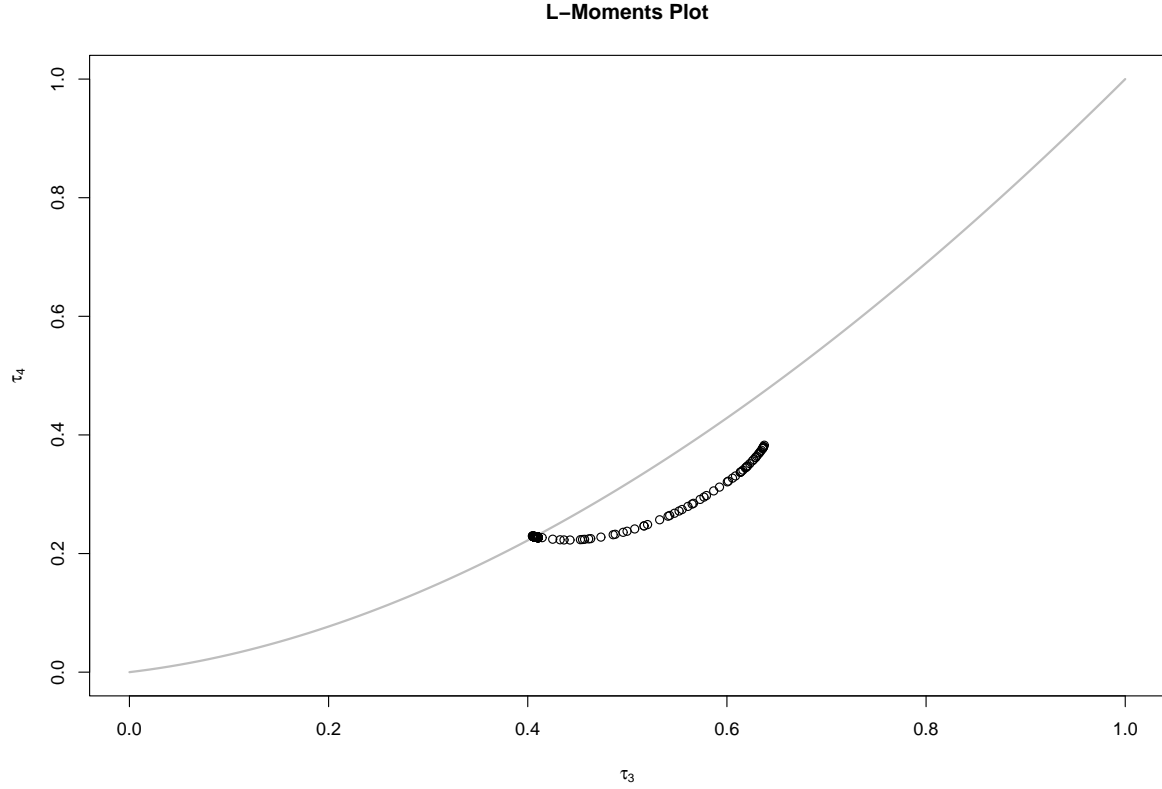


Figure 3: L-Moment plot on synthetic data

Let X be a *r.v.* distributed as a Poisson distribution with parameter λ . That is:

$$\Pr[X = k] = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k \in \mathbb{N}. \quad (3.9)$$

Thus, we have $\mathbb{E}[X] = \text{Var}[X]$. Cunnane (1979) introduced a **Dispersion Index** statistic defined by:

$$DI = \frac{s^2}{\lambda} \quad (3.10)$$

where s^2 is the intensity of the Poisson process and λ the mean number of events in a block - most often this is a year. Moreover, a confidence interval can be computed by using a χ^2 test:

$$I_\alpha = \left[\frac{\chi_{(1-\alpha)/2, M-1}^2}{M-1}, \frac{\chi_{1-(1-\alpha)/2, M-1}^2}{M-1} \right] \quad (3.11)$$

where $\Pr[DI \in I_\alpha] = \alpha$.

For the next example, we use the data set *ardieres* included in the **POT** package. Moreover, as *ardieres* is a time series, and thus strongly auto-correlated, we must “extract” extreme events while preserving independence between events. This is achieved using function **clust**².

```
> data(ardieres)
> events <- clust(ardieres, u = 2, tim.cond = 8 / 365,
+ clust.max = TRUE)
> diplot(events, tlim = c(2, 20))
```

The Dispersion Index plot is presented in Figure 4. From this figure, a threshold around 5 should be reasonable.

²The **clust** function will be presented later in section 3.6.

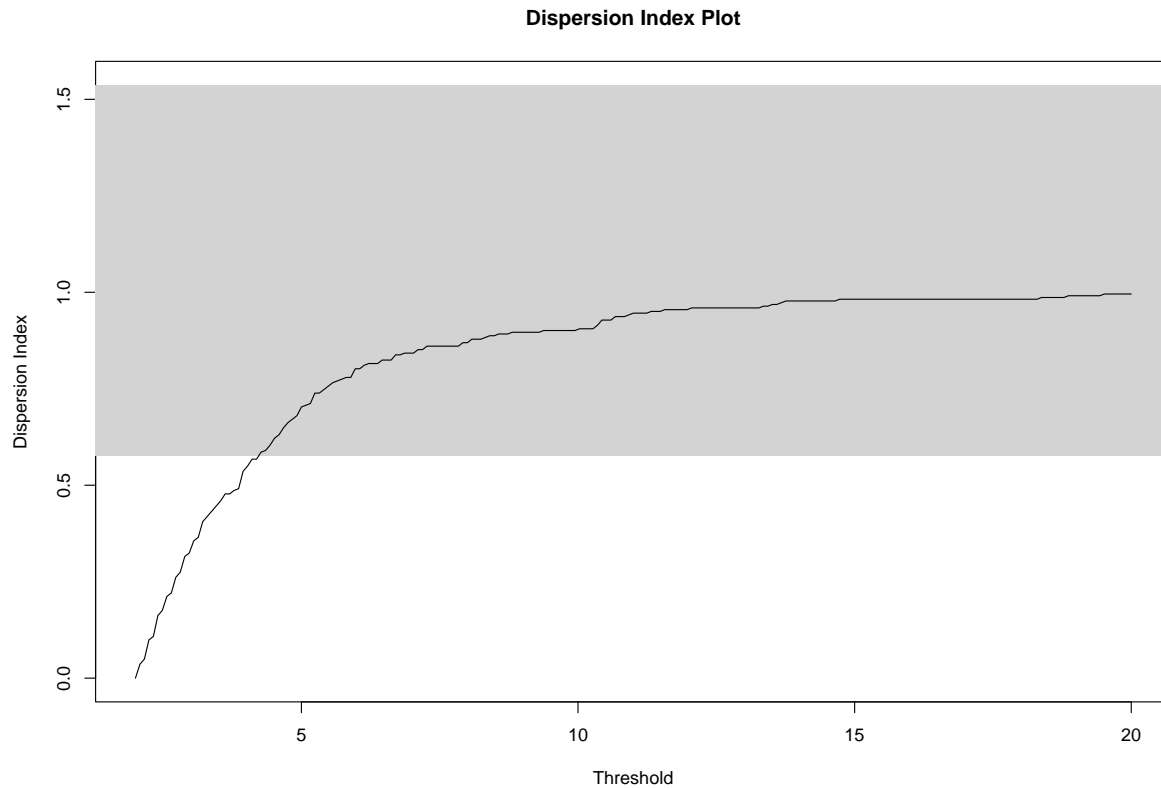


Figure 4: Dispersion index plot for the dataset *ardieres*

3.3 Fitting the GPD

The main function to fit the GPD is called `fitgpd`. This is a generic function which can fit the GPD according several estimators. There are currently 7 estimators available: method of moments `moments`, maximum likelihood `mle`, biased and unbiased probability weighted moments `pwmb`, `pwmu`, mean power density divergence `mdpd`, median `med` and pickands' `pickands` estimators. Details for these estimators can be found in (Coles, 2001), (Hosking and Wallis, 1987), (Juárez and Schucany, 2004), (Peng and Welsh, 2001) and (Pickands, 1975).

The MLE is a particular case as it is the only one which allows varying threshold. Moreover, two types of standard errors are available: “expected” or “observed” information of Fisher. The option `obs.fish` specifies if we want observed (`obs.fish = TRUE`) or expected (`obs.fish = FALSE`).

As Pickands’ estimator is not always feasible, user must check the message of feasibility return by function `fitgpd`.

We give here several didactic examples.

```
> x <- rgpd(200, 1, 2, 0.25)
> mom <- fitgpd(x, 1, "moments")$param
> mle <- fitgpd(x, 1, "mle")$param
> pwmu <- fitgpd(x, 1, "pwmu")$param
> pwmb <- fitgpd(x, 1, "pwmb")$param
> pickands <- fitgpd(x, 1, "pickands")$param
> med <- fitgpd(x, 1, "med", start = mle)$param
> mdpd <- fitgpd(x, 1, "mdpd")$param
> print(rbind(mom, mle, pwmu, pwmb, pickands, med, mdpd))
```


	scale	shape
mom	1.693222	0.22258795
mle	1.758304	0.18825359
pwmu	1.774835	0.18511688
pwmb	1.783859	0.18097373
pickands	1.867132	0.02210745
med	1.826378	0.08445534
mdpd	1.788432	0.16608265

##Convergence: iteration limit reached

The MLE method allows to fix either the scale or the shape parameter. For example, if we want to fit a Exponential distribution, just do:

```
> x <- rgpd(100, 1, 2, 0)
> fitgpd(x, thresh = 1, shape = 0, method = "mle")
> ##The same but with a fixed scale value
> fitgpd(x, thresh = 1, scale = 2, method = "mle")
```

If now, we want to fit a GPD with a varying threshold, just do:

```
> x <- rgpd(500, 1:2, 0.3, 0.01)
> fitgpd(x, 1:2, method = "mle")
```

Note that the varying threshold is repeated cyclically until it matches the length of object `x`.

3.4 Confidence Intervals

Once a statistical model is fitted, it is usual to gives confidence intervals. Currently, only `mle`, `pwmu`, `pwmb`, `moments` estimators can computed confidence intervals. Moreover, for method `mle`, “standard” and “profile” confidence intervals are available.

If we want confidence intervals for the scale parameters:

```
> x <- rgpd(100, 1, 2, 0.25)
> mle <- fitgpd(x, 1, method = "mle")
> mom <- fitgpd(x, 1, method = "moments")
> pwmb <- fitgpd(x, 1, method = "pwmb")
> pwmu <- fitgpd(x, 1, method = "pwmu")
> gpd.fiscale(mle, conf = 0.9)
> gpd.fiscale(mom, conf = 0.9)
> gpd.fiscale(pwmu, conf = 0.9)
> gpd.fiscale(pwmb, conf = 0.9)
```

For shape parameter confidence intervals, simply use function `gpd.fishape` instead of `gpd.fiscale`. Note that the *fi* stands for “Fisher Information”.

Thus, if we want profile confidence intervals, we must use functions `gpd.pfscale` and `gpd.pfshape`. The *pf* stands for “profile”. These functions are only available with a model fitted with MLE.

```
> gpd.pfscale(mle, range = c(1, 2.5), conf = 0.9)
> gpd.pfshape(mle, range = c(-0.1, 0.6), conf = 0.95)
```

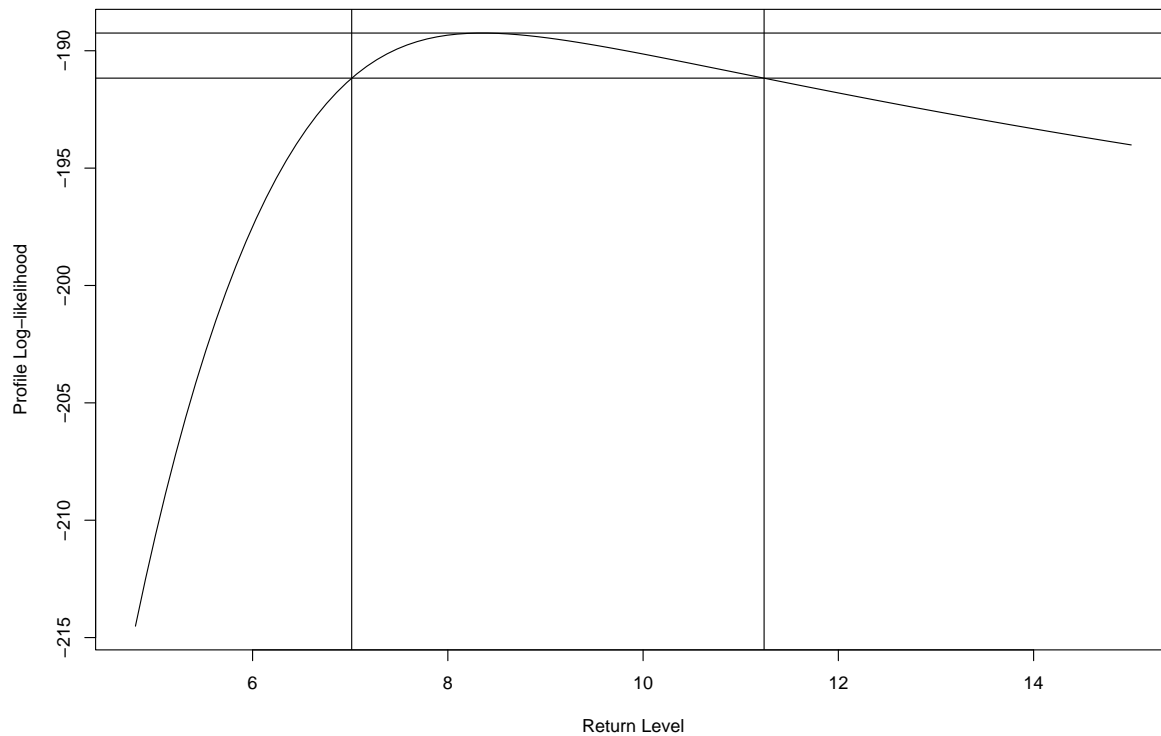


Figure 5: Profile log-likelihood function for a given return level

Confidence interval for quantiles - or return levels - are also available. This is achieved using: (a) the Delta method or (b) profile likelihood.

```
> gpd.firl(pwmu, prob = 0.95)
> gpd.pfml(mle, prob = 0.95, range = c(4.8, 15))
```

The profile confidence interval functions both returns the confidence interval and plot the profile log-likelihood function. Figure 5 depicts the graphic window returned by function `gpd.pfml` for the return level associated to non exceedence probability 0.95.

3.5 Model Checking

To check the fitted model, users must call function `plotgpd`. This is a generic function which calls functions: `pp.gpd` (probability/probability plot), `qq.gpd` (quantile/quantile plot), `dens.gpd` (density plot) and `retlev.gpd` (return level plot).

Here is a basic illustration of the function `plotgpd`.

```
> x <- rgpd(200, 10, 0.5, -0.2)
> fitted <- fitgpd(x, 10, method = "mle")
> par(mfrow=c(2,2))
> plotgpd(fitted, npy = 1)
```

Figure 6 displays the graphic windows obtained with the latter execution.

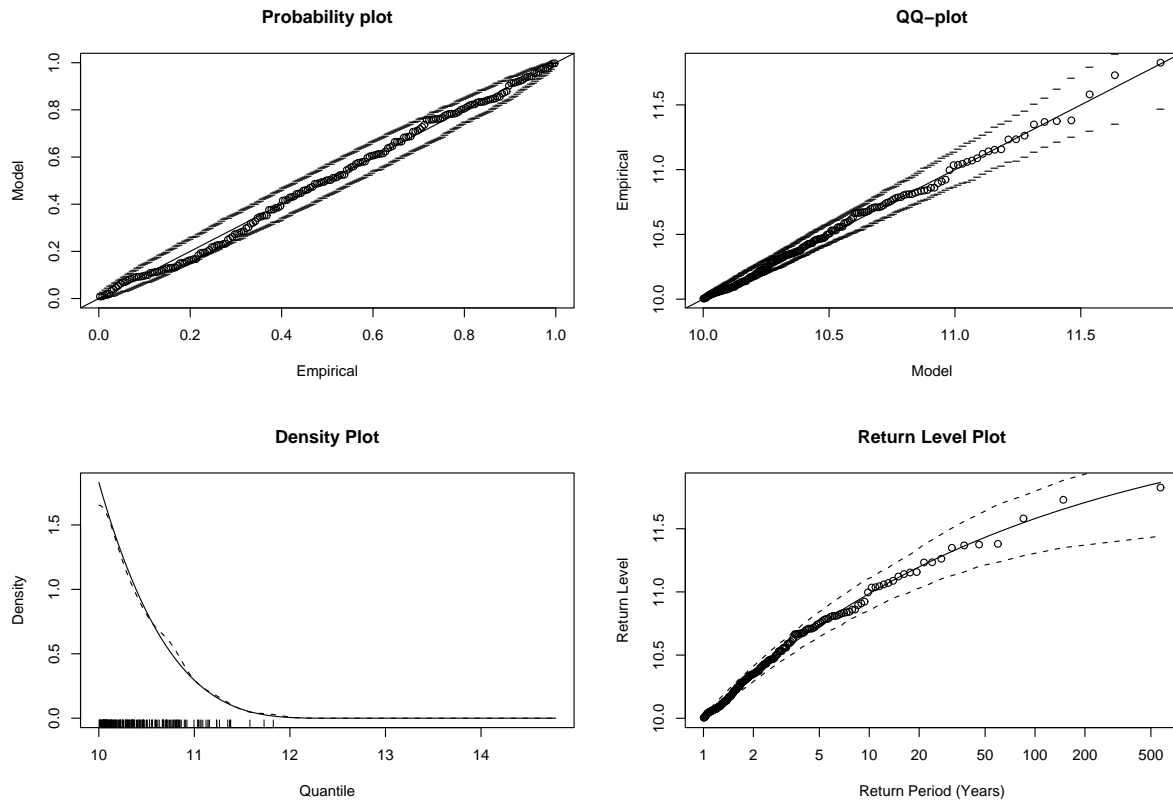


Figure 6: Checking plots from function *plotgpd*

If one is interested in only a probability/probability plot, there is two options. We can call function `pp.gpd` or equivalently `plotgpd` with the **which** option. The “which” option select which graph you want to plot. That is:

- which = 1 for a probability/probability plot;
- which = 2 for a quantile/quantile plot;
- which = 3 for a density plot;
- which = 4 for a return level plot;

Note that “which” can be a vector like `c(1,3)` or `1:3`.

Thus, the following instruction gives the same graphic.

```
> plotgpd(fitted, which = 1)
> pp.gpd(fitted)
```

If a return level plot is asked ($4 \in \text{which}$), a value for `npv` is needed. “npv” corresponds to the *mean number of events per year*. This is required to define the “return period”.

3.6 Declustering Techniques

In opposition to block maxima, a peak over threshold can be problematic when dealing with time series. Indeed, as often time series are strongly auto-correlated, select naively events above a threshold may lead to dependent events.

The function **clust** tries to identify peaks over a threshold while meeting independence criteria. For this purpose, this function needs at least two arguments: the threshold **u** and a time condition for independence **tim.cond**. Clusters are identified as follows:

1. The first exceedence initiates the first cluster;
2. The first observation under the threshold **u** “ends” the cluster unless **tim.cond** does not hold;
3. The next exceedence which holds **tim.cond** initiates a new cluster;
4. The process is iterated as needed.

Here is an application on flood discharges for river Ardière at Beaujeu. A preliminary study shows that two flood events can be considered independent if they do not lie within a 8 days window. Note that unit to define **tim.cond** must be the same than the data analyzed.

```
> data(ardieres)
> clust(ardieres, u = 2, tim.cond = 8 / 365)
```

Several options can be passed to the “clust” function. By default, it will return a list with the identified clusters. Usually, we want only cluster maxima, this is achieved by passing option **clust.max = TRUE**. Users can also ask for a graphic representation of clusters by passing option **plot = TRUE** - see Figure 7.

```
> clustMax <- clust(ardieres, u = 2, tim.cond = 8 / 365,
+ clust.max = TRUE, plot = TRUE, xlim = c(1971.1, 1972.9))
```

3.7 Miscellaneous functions

3.7.1 Return periods: *rp2prob* and *prob2rp*

The functions **rp2prob** and **prob2rp** are useful to convert return periods to non exceedence probabilities and vice versa. It needs either a return period or a non exceedence probability. Moreover, the mean number of events per year “**npv**” must be specified.

```
> rp2prob(50, 1.8)
  npv retper   prob
1 1.8    50 0.9888889
> prob2rp(0.6, 2.2)
  npv  retper prob
1 2.2 1.136364 0.6
```

3.7.2 Unbiased Sample L-Moments: *samlmu*

The function **samlmu** computes the unbiased sample L-Moments.

```
> x <- runif(50)
> samlmu(x, nmom = 5)
      l_1      l_2      t_3      t_4      t_5
0.53337554 0.16743489 -0.04026843 0.01243610 0.01386457
```

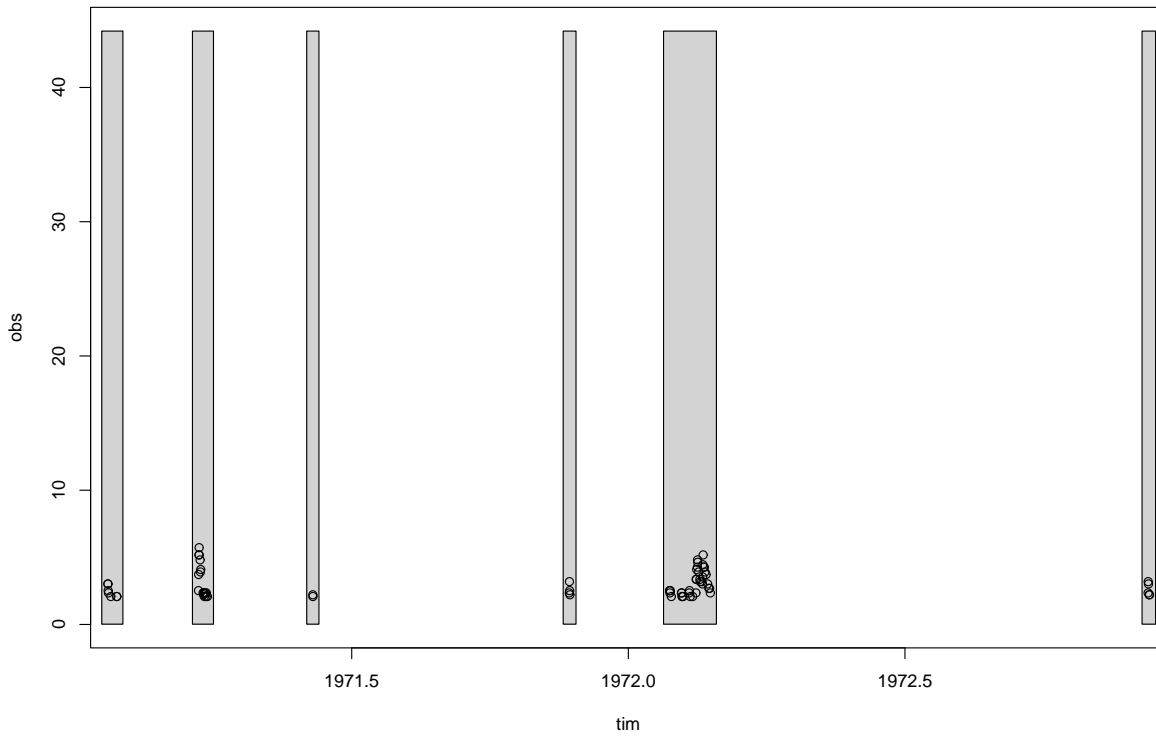


Figure 7: The identified clusters. Data Ardières, $u = 2$, $\text{tim.cond} = 8$

3.7.3 Mobile average window on time series: *ts2tsd*

The function `ts2tsd` computes an “average” time series `tsd` from the initial time series `ts`. This is achieved by using a mobile average window of length `d` on the initial time series.

```
> data(ardieres)
> tsd <- ts2tsd(ardieres, 3 / 365)
> plot(ardieres, type = "l", col = "blue")
> lines(tsd, col = "green")
```

The latter execution is depicted in Figure 8.

4 A Concrete Statistical Analysis of Peaks Over a Threshold

In this section, we provide a full and detailed analysis of peaks over a threshold for the river Ardières at Beaujeu. Figure 8 depicts instantaneous flood discharges - blue line.

As this is a time series, we must select independent events above a threshold. First, we fix a relatively low threshold to “extract” more events. Thus, some of them are not extreme but regular events. This is necessary to select a reasonable threshold for the asymptotic approximation by a GPD - see section 2.

```
> summary(ardieres)
      time      obs
```

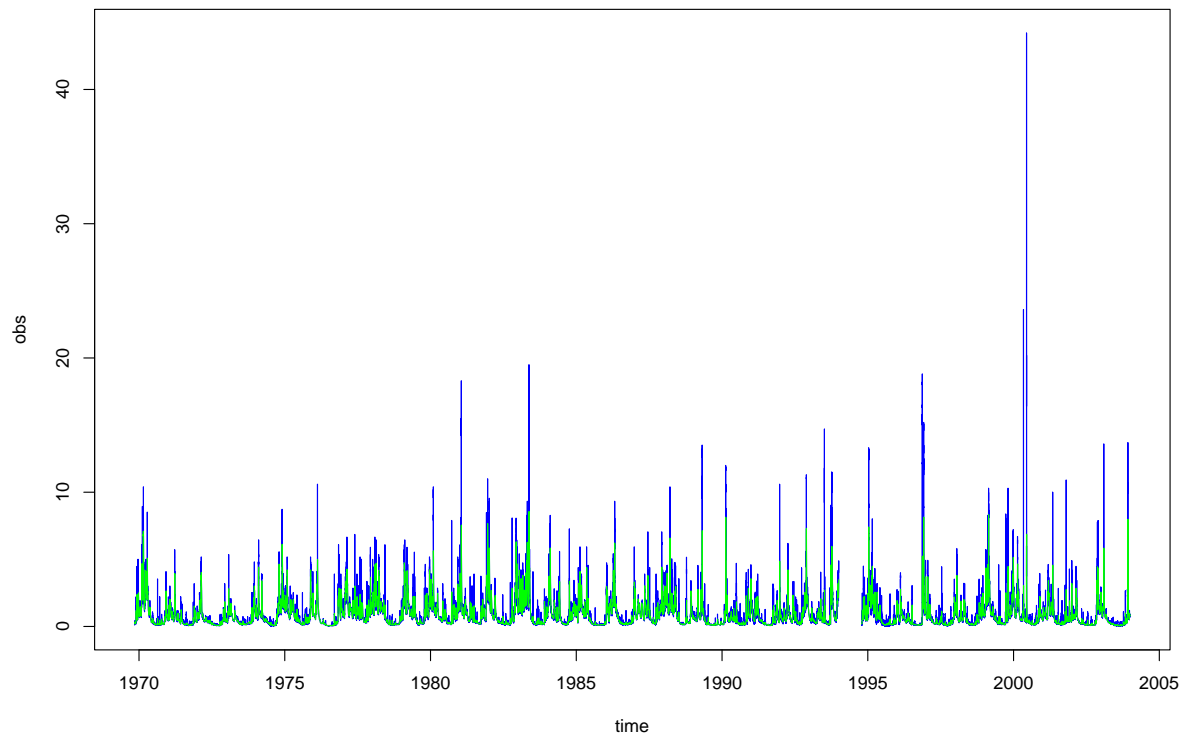


Figure 8: Instantaneous flood discharges and averaged discharged over duration 3 days. Data ardieres

```

Min.    :1970    Min.    : 0.022
1st Qu.:1981    1st Qu.: 0.236
Median :1991    Median : 0.542
Mean    :1989    Mean    : 1.024
3rd Qu.:1997    3rd Qu.: 1.230
Max.    :2004    Max.    :44.200
        NA's    : 1.000
> events0 <- clust(ardieres, u = 1.5, tim.cond = 8/365,
+   clust.max = TRUE)
> par(mfrow=c(2,2))
> mrlplot(events0[, "obs"])
> abline(v = 6, col = "green")
> diplot(events0)
> abline(v = 6, col = "green")
> tcplot(events0[, "obs"])

```

From Figure 9, a threshold value of $6m^3/s$ should be reasonable. The Mean residual life plot - top left panel- indicates that a threshold around $10m^3/s$ should be adequate. However, the selected threshold must be low enough to have enough events above it to reduce variance while not too low as it increase the bias³.

Thus, we can now “re-extract” events above the threshold $6m^3/s$, obtaining object **events1**. This is necessary as sometimes **events1** is not equal to observations of **events0** greater than $6m^3/s$.

³As the asymptotic approximation by a GPD is not accurate anymore.

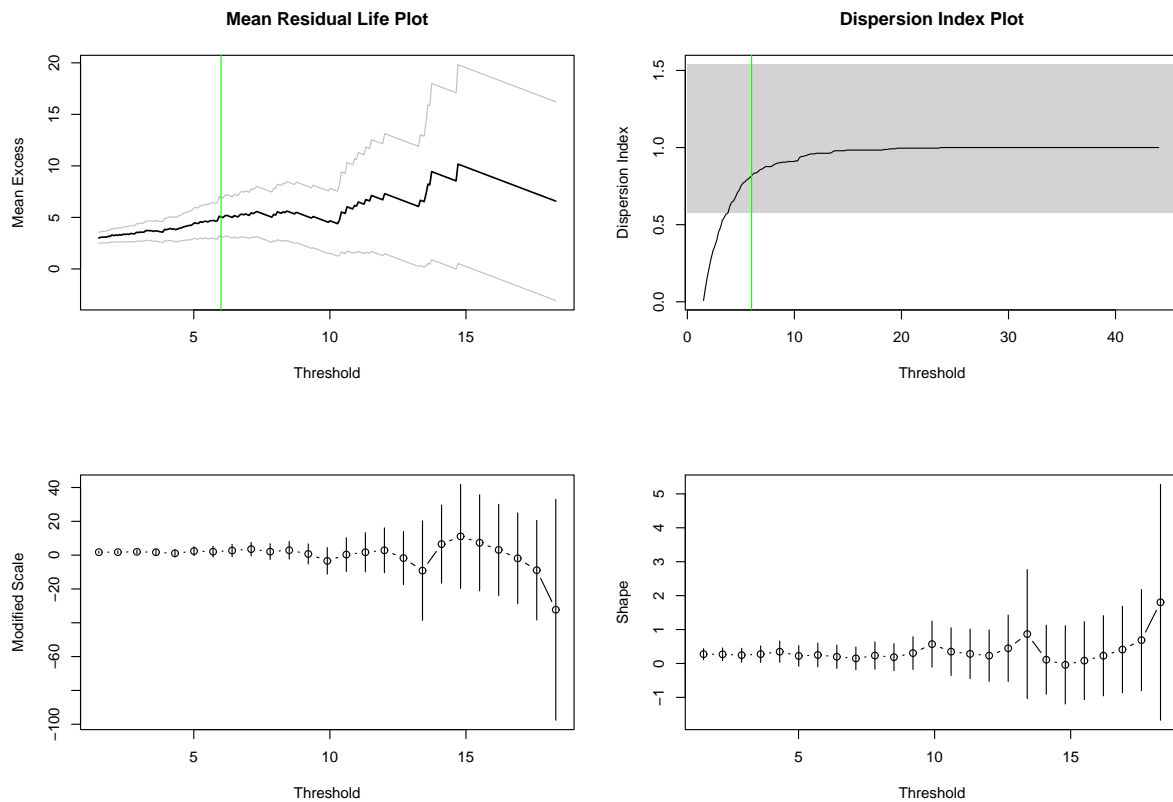


Figure 9: Threshold selection for river Ardieres at Beaujeu.

We can now define the mean number of events per year “npv”. Note that an estimation of the extremal index is available.

```
> events1 <- clust(ardieres, u = 6, tim.cond = 8/365,
+ clust.max = TRUE)
> npv <- length(events1[, "obs"]) / (diff(range(ardieres[, "time"],
na.rm = TRUE)) - diff(ardieres[c(20945, 20947), "time"]))
> ##Because there is a gap !!!
> print(npv)
[1] 1.677934
> attributes(events1)$exi
[1] 0.1225383
```

Let's fit the GPD.

```
> mle <- fitgpd(events1[, "obs"], thresh = 6, method = "mle")
Estimator: MLE
```

Varying Threshold: FALSE

```
Threshold: 6
Number Above: 56
Proportion Above: 1
```

Estimates

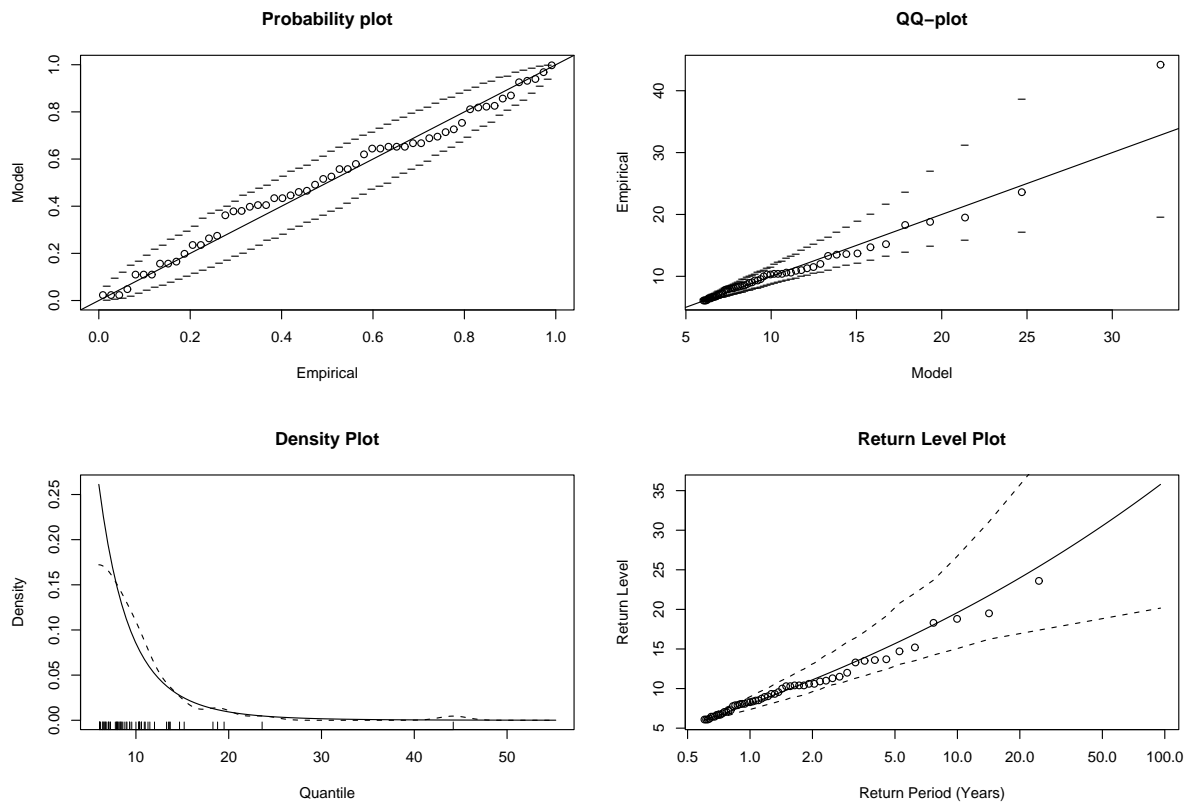


Figure 10: Graphic diagnostics for river Ardières at Beaujeu

```

scale    shape
3.8285   0.1579

```

Standard Error Type: Observed

Standard Errors

```

scale    shape
0.7224   0.1349

```

Asymptotic Variance Covariance

```

scale    shape
scale    0.52180 -0.05714
shape    -0.05714  0.01819

```

Optimization Information

```

Convergence: successful
Function Evaluations: 36
Gradient Evaluations: 9

```

The result of function `fitgpd` gives the name of the estimator, if a varying threshold was used, the threshold value, the number and the proportion of observations above the threshold, parameter estimates, standard error estimates and type, the asymptotic variance-covariance matrix and convergence diagnostic.

Figure 10 shows graphic diagnostics for the fitted model. It can be seen that the fitted model “mle” seems to be appropriate. Suppose we want to know the return level associated to the

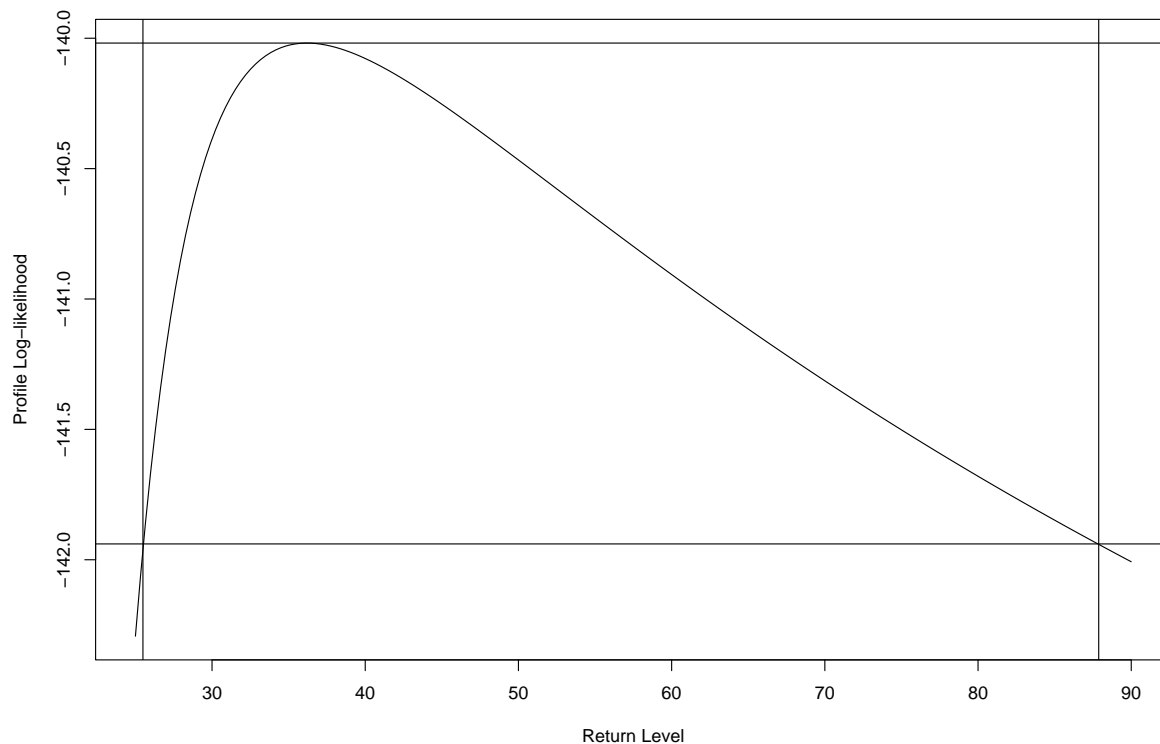


Figure 11: Profile-likelihood function for the 100-year return period quantile

100-year return period.

```
> ##First convert return period in prob
> rp2prob(retper = 100, npy = npy)
      npy retper      prob
1 1.677934    100 0.9940403
> prob <- rp2prob(retper = 100, npy = npy)[,"prob"]
> qgpd(prob, loc = 6, scale = mle$param["scale"],
+   shape = mle$param["shape"])
36.19317
```

To take into account uncertainties, Figure 11 depicts the profile confidence interval for the quantile associated to the 100-year return period.

```
> gpd.pfml(mle, prob, range = c(25, 90), nrang = 200)
If there is some troubles try to put vert.lines = FALSE or change
the range...
conf.inf conf.sup
25.48995 87.87688
```

Sometimes it is necessary to know the estimated return period of a specified events. Lets do it with the larger events in “events1”.

```
> maxEvent <- max(events1[, "obs"])
> print(maxEvent)
```

```
[1] 44.2
> prob <- pgpd(maxEvent, loc = 6, scale = mle$param["scale"],
+ shape = mle$param["shape"])
> print(prob)
0.997501
> prob2rp(prob, npy = npy)
      npy  retper  prob
1 1.677934 238.4804 0.997501
```

Thus, the largest events that occurs in June 2000 has approximately a return period of 240 years. Maybe it is a good idea to fit the GPD with the other estimators available in the **POT** package.

References

- Coles, S. (2001) *An Introduction to Statistical Modelling of Extreme Values*. Springer Series in Statistics. London: Springer Series in Statistics.
- Cunnane, C. (1979) Note on the poisson assumption in partial duration series model. *Water Resour Res*, **15**, 489–494.
- Fisher, R. and Tippett, L. (1928) Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Proceedings of the Cambridge Philosophical Society*, vol. 24.
- Hosking, J. and Wallis, J. (1987) Parameter and quantile estimation for the generalized pareto distribution. *Technometrics*, **29**, 339–349.
- Jenkinson, A. F. (1955) The frequency distribution of the annual maximum (or minimum) values of meteorological events. *Quarterly Journal of the Royal Meteorological Society*, **81**, 158–172.
- Juárez, S. and Schucany, W. (2004) Robust and efficient estimation for the generalized pareto distribution. *Extremes*, **7**, 237–251.
- Kluppelberg, C. and Mikosch, T. (1997) Large deviations of heavy-tailed random sums with applications in insurance and finance. *Journal of Applied Probability*, **34**, 293–308.
- Peng, L. and Welsh, A. (2001) Robust estimation of the generalized pareto distribution. *Extremes*, **4**, 53–65.
- Pickands, J. I. (1975) Statistical inference using extreme order statistics. *Annals of Statistics*, **3**, 119–131.
- R Development Core Team (2006) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>, ISBN 3-900051-07-0.