A User's Guide to the evd Package (Version 2.1)

Alec Stephenson

Department of Statistics,
Division of Economic and Financial Studies,
Macquarie University, NSW 2109,
Australia.

E-mail: alec_stephenson@hotmail.com

1st May 2004

1 Introduction

1.1 What is the evd package?

The evd (extreme value distributions) package is an add-on package for the R (Ihaka and Gentleman, 1996) statistical computing system. The package contains the following (user-level) functions.

Univariate Distributions. Density, distribution, simulation and quantile (inverse distribution) functions for univariate distributions associated with extreme value theory.

```
dgev  dgpd  dgumbel  drweibull  dfrechet  dextreme  dorder
pgev  pgpd  pgumbel  prweibull  pfrechet  pextreme  porder
rgev  rgpd  rgumbel  rrweibull  rfrechet  rextreme  rorder
qgev  qgpd  qgumbel  qrweibull  qfrechet  qextreme
```

Bivariate and Multivariate Extreme Value Distributions. Density, distribution and simulation functions for bivariate and multivariate parametric extreme value models. Eight bivariate models and two multivariate models are implemented.

```
dbvevd  dmvevd  pbvevd  pmvevd  rbvevd  rmvevd
```

Dependence Functions for Bivariate and Trivariate Extreme Value Distributions. Calculate and plot dependence functions for bivariate and trivariate extreme value distributions; either nonparametric estimates, or parametric models, at specified parameter values.

```
abvnonpar  abvpar  atvnonpar  atvpar
```

Stochastic Processes. Generate stochastic processes associated with extreme value theory, identify extreme clusters and estimate the extremal index.

```
evmc  marma  mar  mma  clusters  exi
```

Fitting Models. Obtain maximum likelihood estimates for models used in extreme value theory, including generalized extreme value models, peaks over threshold models and eight parametric bivariate extreme value models.

```
fbvevd  fgev  fpot  forder  fextreme
```

Threshold Identification. Identify suitable thresholds for peaks over threshold modelling.

```
mrlplot  tcplot
```

Model Diagnostics. Model diagnostics for fitted models; diagnostic plots and analysis of deviance.

```
plot.uvevd  plot.bvevd  anova.evd
```

Profile deviances. Obtain profile traces, plot profile deviances and obtain profile confidence intervals from fitted models.

```
profile.evd  plot.profile.evd  profile2d.evd  plot.profile2d.evd
```

The following datasets are also included in the package.

```
  failure  fox  lisbon  ocmulgee  oldage  oxford
  portpirie  sask  sealevel  uccle  venice
```

1.2 Obtaining the package/guide

The evd 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 `evd/doc/` underneath wherever the package is installed.

1.3 Contents

This guide contains examples* on the use of the evd package. The examples do not include any theoretical justification. See Coles (2001) for an introduction to the statistics of extreme values. See Kotz and Nadarajah (2000) for a theoretical treatment of univariate and multivariate extreme value distributions.

Section 2 covers the standard (non-fitting) functions for univariate distributions. Sections 3 and 4 do the same for bivariate and multivariate extreme value models. Dependence functions of extreme value distributions are discussed in Section 5. Stochastic processes are discussed in Section 6. Maximum likelihood fitting of univariate models, peaks over threshold models and bivariate extreme value models is discussed in Sections 7, 8 and 9 respectively. Three practical examples using the data sets `oxford`, `rain` and `sealevel` are given in Sections 10, 11 and 12 respectively.

This guide should not be viewed as an alternative to the documentation files included within the package. These remain the definitive source of information. A reference manual containing all the documentation files can be downloaded from CRAN.

1.4 Citing the package/guide

Volume 2/2 of R-News (the newsletter of the R-project) contains an article that describes (an earlier version of) the evd package. To cite the package in publications please cite the R-News article. The article and the corresponding citation can be downloaded from `http://www.cran.r-project.org/doc/Rnews/`.

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

```
@manual{key,
  title = {A User's Guide to the evd Package (Version 2.1)},
  author = {Stephenson, A. G.},
  year = {2004},
  month = {May},
  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 `alec_stephenson@hotmail.com`. Please include an appropriate subject line.

*All of the examples presented in this guide are called with `options(digits = 4)`, and with the option `show.signif.stars` set to `FALSE`.

1.6 Legalese

2 Univariate Distributions

The Gumbel, Fréchet and (reversed) Weibull distribution functions are respectively given by

$$G(z) = \exp\left\{ -\exp\left[-\left(\frac{z-a}{b} \right) \right] \right\}, \qquad -\infty < z < \infty \tag{1}$$

$$G(z) = \begin{cases} 0, & z \leq a, \\ \exp\left\{ -\left(\frac{z-a}{b} \right)^{-\alpha} \right\}, & z > a, \end{cases} \tag{2}$$

$$G(z) = \begin{cases} \exp\left\{ -\left[-\left(\frac{z-a}{b} \right) \right]^{\alpha} \right\}, & z < a, \\ 1, & z \geq a, \end{cases} \tag{3}$$

where a is a location parameter, $b > 0$ is a scale parameter and $\alpha > 0$ is a shape parameter. The distribution (3) is often referred to as the Weibull distribution. To avoid confusion I will call this the reversed Weibull, since it is related by a change of sign to the three parameter Weibull distribution used in survival analysis.

The GEV (Generalized Extreme Value) distribution function is given by

$$G(z) = \exp\left\{ -\left[1 + \xi \left(z - \mu \right) / \sigma \right]_{+}^{-1/\xi} \right\}, \tag{4}$$

where (μ, σ, ξ) are the location, scale and shape parameters respectively, $\sigma > 0$ and $h_{+} = \max(h, 0)$. When $\xi > 0$ the GEV distribution has a finite lower end point, given by $\mu - \sigma/\xi$. When $\xi < 0$ the GEV distribution has a finite upper end point, also given by $\mu - \sigma/\xi$. The parametric form of the GEV encompasses that of the Gumbel, Frèchet and reversed Weibull distributions. The Gumbel distribution is obtained in the limit as $\xi \to 0$. The Fréchet and Weibull distributions are obtained when $\xi > 0$ and $\xi < 0$ respectively. To recover the parameterization of the Fréchet distribution (2) set $\xi = 1/\alpha > 0$, $\sigma = b/\alpha > 0$ and $\mu = a + b$. To recover the parameterization of the reversed Weibull distribution (3) set $\xi = -1/\alpha < 0$, $\sigma = b/\alpha > 0$ and $\mu = a - b$.

The generalized Pareto distribution (GPD) function is given by

$$G(z) = 1 - \left[1 + \xi \left(z - \mu \right) / \sigma \right]_{+}^{-1/\xi}, \tag{5}$$

for $z > \mu$, where (μ, σ, ξ) are the location, scale and shape parameters respectively, $\sigma > 0$ and $h_{+} = \max(h, 0)$. The GPD has a finite lower end point, given by μ. When $\xi < 0$ the GPD also has a finite upper end point, given by $\mu - \sigma/\xi$. A shifted exponential distribution is obtained in the limit as $\xi \to 0$.

It is standard practice within R to concatenate the letters r, p, q and d with an abbreviated distribution name to yield the names of the corresponding simulation, distribution, quantile (inverse distribution) and density functions respectively. The evd package follows this convention. Each of the five distributions defined above has an associated set of functions, as given in Section 1.1. Some examples are given below. They should be familiar to those who have had previous experience with R.

```
> rgev(6, loc = c(20,1), scale = .5, shape = 1)
[1] 23.7290  1.2492 19.6680  0.8662 19.7939  2.6512
> rgpd(3, loc = 2)
[1] 2.483681 3.666805 2.837809

> qrweibull(seq(0.1, 0.4, 0.1), 2, 0.5, 1, lower.tail = FALSE)
> qrweibull(seq(0.9, 0.6, -0.1), loc = 2, scale = 0.5, shape = 1)
# Both give
[1] 1.947 1.888 1.822 1.745

> pfrechet(2:6, 2, 0.5, 1)
[1] 0.0000 0.6065 0.7788 0.8465 0.8825
> pfrechet(2:6, 2, 0.5, 1, low = FALSE)
[1] 1.0000 0.3935 0.2212 0.1535 0.1175

> drweibull(-1:3, 2, 0.5, log = TRUE)
[1] -5.307 -3.307 -1.307   -Inf   -Inf
> dgumbel(-1:3, 0, 1)
[1] 0.17937 0.36788 0.25465 0.11820 0.04737
```

Let F be an arbitrary distribution function, and let $X_1, \ldots, X_m$ be a random sample from F. Define $U_m = \max\{X_1, \ldots, X_m\}$ and $L_m = \min\{X_1, \ldots, X_m\}$. The distributions of U_m and L_m are given by

$$\Pr(U_m \leq x) = [F(x)]^m \tag{6}$$

$$\Pr(L_m \leq x) = 1 - [1 - F(x)]^m. \tag{7}$$

Simulation, distribution, quantile and density functions for the distributions of U_m and L_m, given an integer m and an arbitrary distribution function F, are provided by `rextreme`, `pextreme`, `qextreme` and `dextreme` respectively. The integer m should be given to the argument `mlen`. The distribution F is most easily specified by passing an abbreviated distribution name to the argument `distn`. If the distribution of U_m is required the argument `largest` should be set to `TRUE` (the default). If the distribution of L_m is required `largest` should be set to `FALSE`. Some examples are given below.

```
> rextreme(1, distn = "norm", sd = 2, mlen = 20, largest = FALSE)
> min(rnorm(20, mean = 0, sd = 2))
# Both simulate from the same distribution
[1] -2.612

> rextreme(4, distn = "exp", rate = 1, mlen = 5)
> rextreme(4, distn = "exp", mlen = 5)
# Both simulate from the same distribution
[1] 2.2001 0.8584 4.5595 3.9397
```

```
> pextreme(c(.4, .5), distn = "norm", mean = 0.5, sd = c(1, 2), mlen = 4)
[1] 0.04484 0.06250
> dextreme(c(1, 4), distn = "gamma", shape = 1, scale = 0.3, mlen = 100)
[1] 0.3261328 0.0005398
```

Let $X_{(1)} \geq X_{(2)} \geq \cdots \geq X_{(m)}$ be the order statistics of the random sample $X_1, \ldots, X_m$. The distribution of the jth largest order statistic, for $j = 1, \ldots, m$, is

$$\Pr(X_{(j)} \leq x) = \sum_{k=0}^{j-1} \binom{m}{k} [F(x)]^{m-k} [1 - F(x)]^k. \tag{8}$$

The distribution of the jth smallest order statistic is obtained by setting $j = m+1-j$. Simulation, distribution and density functions for the distribution of $X_{(j)}$, for given integers m and $j \in \{1, \ldots, m\}$, and for an arbitrary distribution function F, are provided by `rorder`, `porder` and `dorder` respectively. The integer m should again be given to the argument `mlen`. If the argument `largest` is `TRUE` (the default) the distribution of the `j`th largest order statistic $X_{(j)}$ is used. If `largest` is `FALSE` the distribution of the `j`th smallest order statistic $X_{(m+j-1)}$ is used. Some examples are given below.

```
> rorder(1, distn = "norm", mlen = 20, j = 2)
[1] 2.284
> porder(c(1, 2), distn = "gamma", shape = c(.5, .7), mlen = 10, j = 2)
[1] 0.5177 0.8259
> dorder(c(1, 2), distn = "gamma", shape = c(.5, .7), mlen = 10, j = 2)
[1] 0.7473 0.3081
```

3 Bivariate Extreme Value Distributions

The evd package contains functions associated with eight parametric bivariate extreme value distributions. The univariate marginal distributions in each case are GEV, with marginal parameters (μ_1, σ_1, ξ_1) and (μ_2, σ_2, ξ_2).

There are three symmetric models, with distribution functions

$$G(z_1, z_2) = \exp\left\{-(y_1^{1/\alpha} + y_2^{1/\alpha})^\alpha\right\}, \quad 0 < \alpha \leq 1, \tag{9}$$

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + (y_1^{-r} + y_2^{-r})^{-1/r}\right\}, \quad r > 0, \tag{10}$$

$$G(z_1, z_2) = \exp\left(-y_1\Phi\{\lambda^{-1} + \tfrac{1}{2}\lambda[\log(y_1/y_2)]\} - y_2\Phi\{\lambda^{-1} + \tfrac{1}{2}\lambda[\log(y_2/y_1)]\}\right), \quad \lambda > 0,$$

known as the logistic (Gumbel, 1960), negative logistic (Galambos, 1975) and Hüsler-Reiss (Hüsler and Reiss, 1989) models respectively, where

$$y_j = y_j(z_j) = \{1 + \xi_j(z_j - \mu_j)/\sigma_j\}_+^{-1/\xi_j} \tag{11}$$

for $j = 1, 2$. Independence[*] is obtained when $\alpha = 1$, $r \downarrow 0$ or $\lambda \downarrow 0$. Complete dependence[†] is obtained when $\alpha \downarrow 0$, $r \to \infty$ or $\lambda \to \infty$.

[*] Independence occurs when $G(z_1, z_2) = \exp\{-(y_1 + y_2)\}$.

[†] Complete dependence occurs when $G(z_1, z_2) = \exp\{-\max(y_1, y_2)\}$.

The distributions functions (9) and (10) have asymmetric extensions, given by

$$G(z_1, z_2) = \exp\left\{-(1-\theta_1)y_1 - (1-\theta_2)y_2 - [(\theta_1 y_1)^{1/\alpha} + (\theta_2 y_2)^{1/\alpha}]^\alpha\right\}, \quad 0 < \alpha \le 1, \qquad (12)$$

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + [(\theta_1 y_1)^{-r} + (\theta_2 y_2)^{-r}]^{-1/r}\right\}, \quad r > 0,$$

known as the asymmetric logistic (Tawn, 1988) and asymmetric negative logistic (Joe, 1990) models respectively, where the asymmetry parameters $0 \le \theta_1, \theta_2 \le 1$. For the asymmetric logistic model independence is obtained when either $\alpha = 1$, $\theta_1 = 0$ or $\theta_2 = 0$. Different limits occur when θ_1 and θ_2 are fixed and $\alpha \downarrow 0$. For the asymmetric negative logistic model independence is obtained when either $r \downarrow 0$, $\theta_1 \downarrow 0$ or $\theta_2 \downarrow 0$. Different limits occur when θ_1 and θ_2 are fixed and $r \to \infty$.

It can be shown, using a representation of de Haan (1984), that

$$G(z_1, z_2) = \exp\left\{-\int_0^1 \max\{y_1(1-\alpha)x^{-\alpha}, y_2(1-\beta)(1-x)^{-\beta}\}\,\mathrm{d}x\right\}, \quad \alpha, \beta < 1.$$

is a bivariate extreme value distribution function. If we further constrain the parameters to be non-negative we obtain the bivariate bilogistic model proposed by Smith (1990), which can also be expressed as

$$G(z_1, z_2) = \exp\left\{-y_1\gamma^{1-\alpha} - y_2(1-\gamma)^{1-\beta}\right\}, \quad 0 < \alpha, \beta < 1, \qquad (13)$$

where $\gamma = \gamma(y_1, y_2; \alpha, \beta)$ solves $(1-\alpha)y_1(1-\gamma)^\beta = (1-\beta)y_2\gamma^\alpha$. The logistic model is obtained when $\alpha = \beta$. Independence is obtained as $\alpha = \beta \to 1$, and when one of α, β is fixed and the other approaches one. Different limits occur when one of α, β is fixed and the other approaches zero.

Alternatively, if we constrain both parameters to be non-positive and set $\alpha_0 = -\alpha > 0$ and $\beta_0 = -\beta > 0$ we obtain the negative bilogistic model (Coles and Tawn, 1994), which has the representation

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + y_1\gamma^{1+\alpha_0} + y_2(1-\gamma)^{1+\beta_0}\right\}, \quad \alpha_0, \beta_0 > 0,$$

where $\gamma = \gamma(y_1, y_2; -\alpha_0, -\beta_0)$. The negative logistic model is obtained when $\alpha_0 = \beta_0$ (with $r = 1/\alpha_0 = 1/\beta_0$). Independence is obtained as $\alpha_0 = \beta_0 \to \infty$, and when one of α_0, β_0 is fixed and the other tends to ∞. Different limits occur when one of α_0, β_0 is fixed and the other approaches zero.

The Coles-Tawn model[‡] (Coles and Tawn, 1991) is the final model that is considered in the evd package. The distribution function is given by

$$G(z_1, z_2) = \exp\left\{-y_1[1 - \mathrm{Be}(u; \alpha+1, \beta)] - y_2\,\mathrm{Be}(u; \alpha, \beta+1)\right\}, \quad \alpha, \beta > 0,$$

where $u = \alpha y_2/(\alpha y_2 + \beta y_1)$ and Be is the incomplete beta function, given by

$$\mathrm{Be}(u; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^u x^{\alpha-1}(1-x)^{\beta-1}\,\mathrm{d}x.$$

Complete dependence is obtained in the limit as $\alpha = \beta \to \infty$. Independence is obtained as $\alpha = \beta \to 0$ and when one of α, β is fixed and the other approaches zero. Different limits occur when one of α, β is fixed and the other tends to ∞.

[‡] Coles and Tawn (1991) call this the Dirichelet model.

Density, distribution and simulation functions for each of the eight models are provided by `dbvevd`, `pbvevd` and `rbvevd` respectively. The argument `model` denotes the specified model, which must be either `"log"` (the default), `"alog"`, `"hr"`, `"neglog"`, `"aneglog"`, `"bilog"`, `"negbilog"` or `"ct"` (or any unique partial match). The first argument in `pbvevd` and `dbvevd` should be a vector of length two or a matrix with two columns, so that each row specifies a value for (z_1, z_2).

The parameters of the specified model can be passed using one or more of the arguments `dep`, `asy`, `alpha` and `beta`. The marginal parameters (μ_1, σ_1, ξ_1) and (μ_2, σ_2, ξ_2) can be passed using the arguments `mar1` and `mar2` respectively. Gumbel marginal distributions are used by default. The arguments `mar1` and `mar2` can also be matrices with three columns, in which case each column represents a vector of values to be passed to the corresponding marginal parameter. Some examples are given below.

```
> rbvevd(3, dep = .8, asy = c(.4, 1), model = "alog")
         [,1]    [,2]
[1,]  0.07876 -0.7971
[2,]  0.01091 -0.8113
[3,] -0.10491 -0.8831

> rbvevd(3, alpha = .5, beta = 1.2, model = "negb", mar1 = rep(1, 3))
       [,1]  [,2]
[1,] 0.7417 1.085
[2,] 0.8391 1.825
[3,] 2.0142 2.280

> pbvevd(c(1, 1.2), dep = .4, asy = c(.4, .6), model = "an", mar1 = rep(1, 3))
[1] 0.173
> tmp.quant <- matrix(c(1,1.2,1,2), ncol = 2, byrow = TRUE)
> tmp.mar <- matrix(c(1,1,1,1.2,1.2,1.2), ncol = 3, byrow = TRUE)
> pbvevd(tmp.quant, dep = .4, asy = c(.4, .6), model = "an", mar1 = tmp.mar)
[1] 0.173 0.175

> dbvevd(c(1, 1.2), alpha = .2, beta = .6, model = "ct", mar1 = rep(1, 3))
[1] 0.1213
> dbvevd(tmp.quant, alpha = 0.2, beta = 0.6, model = "ct", mar1 = tmp.mar)
[1] 0.1213 0.0586
```

4 Multivariate Extreme Value Distributions

Let $z = (z_1, \dots, z_d)$. The d-dimensional logistic model (Gumbel, 1960) has distribution function

$$G(z) = \exp\left\{-\left(\sum\nolimits_{j=1}^{d} y_j^{1/\alpha}\right)^{\alpha}\right\} \tag{14}$$

where $\alpha \in (0, 1]$ and $(y_1, \dots, y_d)$ is defined by the transformations (11).

This distribution can be extended to an asymmetric form. Let B be the set of all non-empty subsets of $\{1, \dots, d\}$, let $B_1 = \{b \in B : |b| = 1\}$, where $|b|$ denotes the number of elements in the set b, and let $B_{(i)} = \{b \in B : i \in b\}$. The multivariate asymmetric logistic model (Tawn, 1990) is given by

$$G(z) = \exp\left\{-\sum\nolimits_{b \in B}\left[\sum\nolimits_{i \in b}(\theta_{i,b} y_i)^{1/\alpha_b}\right]^{\alpha_b}\right\}$$

where the dependence parameters $\alpha_b \in (0,1]$ for all $b \in B \setminus B_1$, and the asymmetry parameters $\theta_{i,b} \in [0,1]$ for all $b \in B$ and $i \in b$. The constraints $\sum_{b \in B_{(i)}} \theta_{i,b} = 1$ for $i = 1, \ldots, d$ ensure that the marginal distributions are GEV. There exists further constraints which arise from the possible redundancy of asymmetry parameters in the expansion of the distributional form. Specifically, if $\alpha_b = 1$ for some $b \in B \setminus B_1$ then $\theta_{i,b} = 0$ for all $i \in b$. Let $b_{-i_0} = \{i \in b : i \neq i_0\}$. If, for some $b \in B \setminus B_1$, $\theta_{i,b} = 0$ for all $i \in b_{-i_0}$, then $\theta_{i_0,b} = 0$. The model contains $2^d - d - 1$ dependence parameters and $d2^{d-1}$ asymmetry parameters (excluding the constraints). The logistic model (14) can be obtained by setting $\theta_{i,12\ldots d} = 1$ for all $i = 1, \ldots, d$ (which implies that $\theta_{i,b} = 0$ whenever $|b| < d$) and $\alpha_{12\ldots d} = \alpha$. The density functions for the symmetric and asymmetric logistic models are given in Stephenson (2003).

Density, distribution and simulation functions for these models are provided by `dmvevd`, `pmvevd` and `rmvevd` respectively. The argument `model` denotes the specified model, which must be either `"log"` (the default) or `"alog"` (or any unique partial match). The argument `d` denotes the dimension of the model. By default, `d = 2`. The first argument in `pbvevd` and `dbvevd` should be a vector of length `d` or a matrix with `d` columns, so that each row specifies a value for $(z_1, \ldots, z_d)$. The marginal parameters (μ_i, σ_i, ξ_i), for $i = 1, \ldots, d$, can be passed using the argument `mar`. Gumbel marginal distributions are used by default. For the symmetric logistic model, the argument `dep` represents the parameter α. Some examples are given below.

```
> rmvevd(3, dep = .6, model = "log", d = 5)
        [,1]    [,2]    [,3]     [,4]  [,5]
[1,]  0.1335  0.2878 1.07886  1.55515 1.310
[2,]  1.7100  0.9453 1.02070 -0.02553 1.527
[3,] -0.3376 -0.5814 0.07426  0.10906 2.827

> tmp.mar <- matrix(c(1,1,1,1,1,1.5,1,1,2), ncol = 3, byrow = TRUE)
> rmvevd(3, dep = .6, d = 5, mar = tmp.mar)
      [,1]   [,2]   [,3]   [,4]  [,5]
[1,] 2.803 4.6415 1.8531 3.5569 8.854
[2,] 0.751 0.9704 2.3328 2.6537 1.233
[3,] 4.641 1.4321 0.5825 0.6041 2.021

> tmp.quant <- matrix(rep(c(1,1.5,2), 5), ncol = 5)
> pmvevd(tmp.quant, dep = .6, d = 5, mar = tmp.mar)
[1] 0.07233 0.16387 0.21949
> dmvevd(tmp.quant, dep = .6, d = 5, mar = tmp.mar, log = TRUE)
[1] -3.564 -6.610 -9.460
```

For the asymmetric logistic model `dep` should be a vector of length $2^{\mathtt{d}} - \mathtt{d} - 1$ containing the dependence parameters. For example, when $\mathtt{d} = 4$

$$\mathtt{dep} = \mathtt{c}(\alpha_{12}, \alpha_{13}, \alpha_{14}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{123}, \alpha_{124}, \alpha_{134}, \alpha_{234}, \alpha_{1234}).$$

The asymmetry parameters should be passed to `asy` in a list with $2^{\mathtt{d}} - 1$ elements, where each element is a vector[*] corresponding to a set $b \in B$, containing $\{\theta_{i,b} : i \in b\}$. For example, when $\mathtt{d} = 4$

$$\begin{aligned}\mathtt{asy} = \mathtt{list}(&\theta_{1,1}, \theta_{2,2}, \theta_{3,3}, \theta_{4,4}, \mathtt{c}(\theta_{1,12}, \theta_{2,12}), \mathtt{c}(\theta_{1,13}, \theta_{3,13}), \mathtt{c}(\theta_{1,14}, \theta_{4,14}), \mathtt{c}(\theta_{2,23}, \theta_{3,23}),\\ &\mathtt{c}(\theta_{2,24}, \theta_{4,24}), \mathtt{c}(\theta_{3,34}, \theta_{4,34}), \mathtt{c}(\theta_{1,123}, \theta_{2,123}, \theta_{3,123}), \mathtt{c}(\theta_{1,124}, \theta_{2,124}, \theta_{4,124}),\\ &\mathtt{c}(\theta_{1,134}, \theta_{3,134}, \theta_{4,134}), \mathtt{c}(\theta_{2,234}, \theta_{3,234}, \theta_{4,234}), \mathtt{c}(\theta_{1,1234}, \theta_{2,1234}, \theta_{3,1234}, \theta_{4,1234})).\end{aligned}$$

[*] Including vectors of length one.

All the constraints, including $\sum_{b \in B_{(i)}} \theta_{i,b} = 1$ for $i = 1, \dots, d$, must be satisfied or an error will occur. Some examples are given below.

The dependence parameters used in the following trivariate asymmetric logistic model are $(\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{123}) = (.6, .5, .8, .3)$. The asymmetry parameters are $\theta_{1,1} = .4$, $\theta_{2,2} = 0$, $\theta_{3,3} = .6$, $(\theta_{1,12}, \theta_{2,12}) = (.3, .2)$, $(\theta_{1,13}, \theta_{3,13}) = (.1, .1)$, $(\theta_{2,23}, \theta_{3,23}) = (.4, .1)$ and finally $(\theta_{1,123}, \theta_{2,123}, \theta_{3,123}) = (.2, .4, .2)$.

```
> asy <- list(.4, 0, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.4,.2))
> rmvevd(3, dep = c(.6,.5,.8,.3), asy = asy, model = "alog", d = 3)
         [,1]    [,2]    [,3]
[1,]  0.52375 -0.8844  1.4898
[2,]  1.16174 -0.4368 -0.7404
[3,] -0.03737  1.5139 -0.5996

> dmvevd(c(2, 2, 2), dep = c(.6,.5,.8,.3), asy = asy, model = "a", d = 3)
[1] 0.006636

> tmp.quant <- matrix(rep(c(1,1.5,2), 3), ncol = 3)
> pmvevd(tmp.quant, dep = c(.6,.5,.8,.3), asy = asy, model = "a", d = 3)
[1] 0.4131 0.5849 0.7223
```

The dependence parameters used in the following four dimensional asymmetric logistic model are $\alpha_b = 1$ for $|b| = 2$[†] and $(\alpha_{123}, \alpha_{124}, \alpha_{134}, \alpha_{234}, \alpha_{1234}) = (.7, .3, .8, .7, .5)$. The asymmetry parameters are $\theta_{i,b} = 0$ for all $i \in b$ when $|b| \leq 2$, $(\theta_{1,123}, \theta_{2,123}, \theta_{3,123}) = (.2, .1, .2)$, $(\theta_{1,124}, \theta_{2,124}, \theta_{4,124}) = (.1, .1, .2)$, $(\theta_{1,134}, \theta_{3,134}, \theta_{4,134}) = (.3, .4, .1)$, $(\theta_{2,234}, \theta_{3,234}, \theta_{4,234}) = (.2, .2, .2)$ and finally $(\theta_{1,1234}, \theta_{2,1234}, \theta_{3,1234}, \theta_{4,1234}) = (.4, .6, .2, .5)$.

```
> asy <- list(0, 0, 0, 0, c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0),
  c(.2,.1,.2), c(.1,.1,.2), c(.3,.4,.1), c(.2,.2,.2), c(.4,.6,.2,.5))
> rmvevd(3, dep = c(rep(1,6),.7,.3,.8,.7,.5), asy = asy, model = "alog", d = 4)
        [,1]    [,2]    [,3]    [,4]
[1,] -0.5930 -0.1916  1.0211  0.6113
[2,]  4.3522 -1.0050  2.3618 -0.1875
[3,]  0.5805  0.4443 -0.5958  0.9717
```

5 Dependence Functions

Let $z = (z_1, \dots, z_d)$ and $\omega = (\omega_1, \dots, \omega_d)$. Any d-dimensional extreme value distribution function can be represented in the form

$$G(z) = \exp\left\{ -\left\{ \sum\nolimits_{j=1}^{d} y_j \right\} A\left(\frac{y_1}{\sum_{j=1}^{d} y_j}, \dots, \frac{y_d}{\sum_{j=1}^{d} y_j} \right) \right\}, \tag{15}$$

where $(y_1, \dots, y_d)$ is defined by the transformations (11). It follows that $A(\omega) = -\log\{G(y_1^{-1}(\omega_1), \dots, y_d^{-1}(\omega_d))\}$, defined on the simplex $S_d = \{\omega \in \mathbb{R}_+^d : \sum_{j=1}^{d} \omega_j = 1\}$. $A(\cdot)$ is known as the dependence function. The dependence function characterizes the dependence structure of G. It can be shown that $A(\omega) = 1$ when ω is one of the d vertices of S_d (i.e. when one component of ω is equal to one, and all remaining components are equal to zero), and that

[†] The values taken by α_b when $|b| = 2$ are irrelevant here because $\theta_{i,b} = 0$ for all $i \in b$ when $|b| = 2$.

A is a convex function with $\max(\omega_1, \dots, \omega_d) \leq A(\omega) \leq 1$ for all $\omega \in S_d$. The lower and upper bounds are obtained at complete dependence and mutual independence respectively. In particular, $A(1/d, \dots, 1/d)$ is equal to $1/d$ at complete dependence, and 1 at mutual independence.

The dependence function of a *bivariate* extreme value distribution is a special case (because the sets S_2 and [0,1] are equivalent), and is typically defined as follows. Any bivariate extreme value distribution function can be represented in the form

$$G(z_1, z_2) = \exp\left\{-(y_1 + y_2)A\left(\frac{y_1}{y_1 + y_2}\right)\right\}, \tag{16}$$

so that $A(\omega) = -\log\{G(y_1^{-1}(\omega), y_2^{-1}(1-\omega))\}$, defined on $0 \leq \omega \leq 1$.* It follows that $A(0) = A(1) = 1$, and that $A(\cdot)$ is a convex function with $\max(\omega, 1-\omega) \leq A(\omega) \leq 1$ for all $0 \leq \omega \leq 1$. At independence $A(1/2) = 1$. At complete dependence $A(1/2) = 0.5$.

Dependence functions for parametric bivariate and trivariate extreme value models can be calculated and plotted, at given parameter values, using the functions `abvpar` and `atvpar`. Some examples are given at the end of this section. Non-parametric estimators of dependence functions can also be calculated and plotted, using the functions `abvnonpar` and `atvnonpar`. Non-parametric estimators of dependence functions of bivariate extreme value models are constructed as follows.

Suppose (z_{i1}, z_{i2}) for $i = 1, \dots, n$ are n bivariate observations that are passed to `abvnonpar` using the argument `data`. The marginal parameters are estimated (under the assumption of independence) and the data is transformed using

$$\begin{aligned} y_{i1} &= \{1 + \hat{\xi}_1(z_{i1} - \hat{\mu}_1)/\hat{\sigma}_1\}_+^{-1/\hat{\xi}_1} \\ y_{i2} &= \{1 + \hat{\xi}_2(z_{i2} - \hat{\mu}_2)/\hat{\sigma}_2\}_+^{-1/\hat{\xi}_2} \end{aligned} \tag{17}$$

for $i = 1, \dots, n$, where $(\hat{\mu}_1, \hat{\sigma}_1, \hat{\xi}_1)$ and $(\hat{\mu}_2, \hat{\sigma}_2, \hat{\xi}_2)$ are the maximum likelihood estimates for the location, scale and shape parameters on the first and second margins. If non-stationary fitting is implemented using the `nsloc1` or `nsloc2` arguments (see Sections 7 and 9) the marginal location parameters may depend on i.

The estimator is specified using the argument `method`. A number of different estimators are implemented. A short simulation study given in Appendix A compares the properties of these estimators. The default estimator is the estimator of Capéraà *et al.* (1997), which is defined (on $0 \leq \omega \leq 1$) by

$$\exp\left\{\{1 - p(\omega)\}\int_0^\omega \frac{H_n(x) - x}{x(1-x)}\,\mathrm{d}x - p(\omega)\int_\omega^1 \frac{H_n(x) - x}{x(1-x)}\,\mathrm{d}x\right\}$$

where $H_n(x)$ is the empirical distribution function of $x_1, \dots, x_n$, with $x_i = y_{i1}/(y_{i1} + y_{i2})$ for $i = 1, \dots, n$, and $p(\cdot)$ is any bounded function on $[0, 1]$, which can be specified using the argument `wf`. By default $p(\cdot)$ is the identity function.

Let $A_n(\cdot)$ be any estimator of $A(\cdot)$. $A_n(\cdot)$ will not necessarily satisfy $\max(\omega, 1-\omega) \leq A_n(\omega) \leq 1$ for all $0 \leq \omega \leq 1$. An obvious modification is

$$A_n^{'}(\omega) = \min(1, \max\{A_n(\omega), \omega, 1-\omega\}).$$

The function `abvnonpar` always implements this modification. Another estimator $A_n^{''}(\omega)$ can be derived by taking the convex minorant of $A_n^{'}(\omega)$. This can be achieved by setting the argument `convex` to `TRUE`.

Some examples of the functions described in this section are given below. The last eight lines of code produce Figure 1.

*Some authors (e.g. Pickands, 1981) use $A(\omega) = -\log\{G(y_1^{-1}(1-\omega), y_2^{-1}(\omega))\}$.

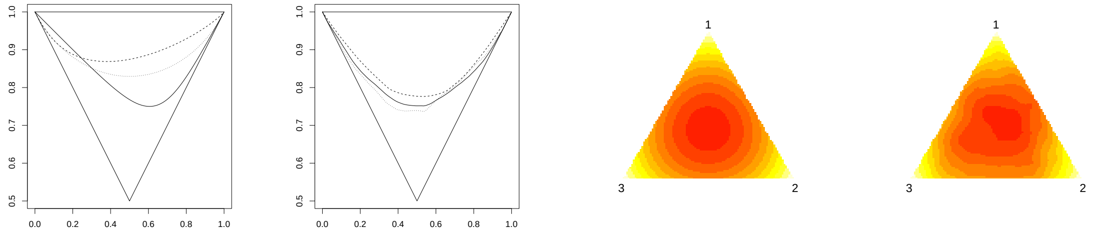

Figure 1: Extreme left: dependence functions for various parametric bivariate extreme value models. The triangular border represents the constraint $\max(\omega, 1-\omega) \leq A(\omega) \leq 1$ for all $\omega \in [0,1]$. Left: non-parametric estimates of the dependence function using data simulated from a bivariate logistic model with $\alpha = 0.6$. Right: the dependence function of a trivariate logistic model with $\alpha = 0.6$, and (extreme right) a non-parametric estimate using data simulated from that model. The colours represent twelve equally spaced intervals between `lower` $= 0.6$ and 1, with darker colours representing stronger dependence (and hence lower values). At the vertex labelled with the value i, the ith component of $\omega \in S_3$ is one (and hence the remaining components are zero).

```
> bvlsm <- rmvevd(100, dep = 0.6, model = "log", d = 2)
> tvlsm <- rmvevd(100, dep = 0.6, model = "log", d = 3)

> abvpar(seq(0,1,0.25), dep = 0.3, asy = c(.7,.9), model = "alog")
[1] 1.0000 0.8272 0.7013 0.7842 1.0000
> abvnonpar(seq(0,1,0.25), data = bvlsm)
[1] 1.0000 0.8634 0.8158 0.8392 1.0000

> abvpar(dep = .3, asy = c(.5, .9), model = "al", plot = TRUE, blty = 1)
> abvpar(alpha = .5, beta = .9, model = "bil", add = TRUE, lty = 2)
> abvpar(dep = 1.05, model = "hr", add = TRUE, lty = 3)

> abvnonpar(data = bvlsm, plot = TRUE, method = "cfg", blty = 1)
> abvnonpar(data = bvlsm, method = "tdo", add = TRUE, lty = 2)
> abvnonpar(data = bvlsm, method = "pick", add = TRUE, lty = 3)

> atvpar(dep = 0.6, model = "log", plot = TRUE, lower = 0.6)
> atvnonpar(data = tvlsm, plot = TRUE, lower = 0.6)
```

6 Stochastic Processes

The evd package contains four functions that simulate from stochastic processes associated with extreme value theory. The functions `marma`, `mar` and `mma` generate max autoregressive moving average processes, and the function `evmc` generates Markov chains with extreme value dependence structures. The function `clusters` identifies extreme clusters of a stochastic process, and `exi` estimates a quantity known as the Extremal Index.

A max autoregressive moving average process $\{X_k\}$, denoted by MARMA(p, q), satisfies

$$X_k = \max\{\phi_1 X_{k-1}, \ldots, \phi_p X_{k-p}, \epsilon_k, \theta_1 \epsilon_{k-1}, \ldots, \theta_q \epsilon_{k-q}\}$$

where $(\phi_1, \dots, \phi_p)$ and $(\theta_1, \dots, \theta_p)$ are vectors of non-negative parameters, and $\{\epsilon_k\}$ is a series of *iid* random variables with a common distribution defined by the argument `rand.gen`. The standard Fréchet distribution is used by default. A max autoregressive process $\{X_k\}$, denoted by MAR(p), is equivalent to a MARMA(p, 0) process, so that

$$X_k = \max\{\phi_1 X_{k-1}, \dots, \phi_p X_{k-p}, \epsilon_k\}.$$

A max moving average process $\{X_k\}$, denoted by MMA(q), is equivalent to a MARMA(0, q) process, so that

$$X_k = \max\{\epsilon_k, \theta_1 \epsilon_{k-1}, \dots, \theta_q \epsilon_{k-q}\}.$$

The functions `mar`, `mma` and `marma` generate MAR(p), MMA(q) and MARMA(p, q) processes respectively. Examples of calls to these functions are given below. The `n.start` argument denotes the burn-in period, which can be specified so that the output series is not unduly influenced by the p starting values, which are all zero by default.

```
> marma(100, p = 1, q = 1, psi = 0.75, theta = 0.65)
> mar(100, psi = 0.85, n.start = 20)
> mma(100, q = 2, theta = c(0.75, 0.8))
```

The function `evmc` generates first order Markov chains. Informally, a first order Markov chain $X_1, \dots, X_n$ is a stochastic process such that at any given time t the probability distribution of X_{t+1} is independent the past $X_1, \dots, X_{t-1}$, given the current state X_t. The `evmc` function generates a first order Markov chain such that each pair of consecutive values has the dependence structure of one of the parametric bivariate extreme value models given in Section 3. The main arguments of `evmc` are the same as those of `rbvevd`. The function `evmc` also has the argument `margin`, which denotes the marginal distribution of each value. This must be either `"uniform"` (the default), `"exponential"`, `"frechet"` or `"gumbel"` (or any unique partial match), for the uniform, standard exponential, standard Gumbel and standard Fréchet distributions respectively. Examples of calls to `evmc` are given below.

```
> evmc(100, alpha = 0.1, beta = 0.1, model = "bilog")
> evmc(100, dep = 10, model = "hr", margins = "exp")
```

The function `clusters` identifies extreme clusters within (stationary) stochastic processes. A simple way of determining clusters is to specify a threshold u and define consecutive exceedances of u to belong to the same cluster. It is more common though to consider a cluster to be active until r consecutive values fall below (or are equal to) u, for some given clustering interval length r. If $r > 1$ the clusters may contain any arbitrarily low value. To avoid this problem a lower threshold $u_l < u$ can be specified so that a cluster is terminated whenever any values fall below (or are equal to) u_l. The following code uses `clusters` to generate the plots depicted in Figure 2. These plots identify clusters graphically. If the argument `plot` is `FALSE` (the default), then `clusters` returns a list of extreme clusters.

```
> set.seed(150)
> x <- evmc(50, dep = 0.55, model ="log")
> clusters(x, 0.8, plot = TRUE)
> clusters(x, 0.8, 2, plot = TRUE)
> clusters(x, 0.8, 4, plot = TRUE)
> clusters(x, 0.8, 4, 0.4, plot = TRUE)
```

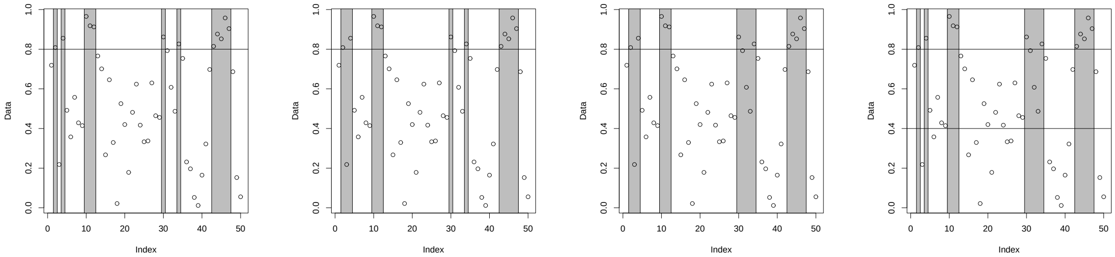

Figure 2: The identification of extreme clusters in a stochastic process. The clustering interval lengths are (from left to right) $r = 1, 2, 4, 4$. The threshold in each case is $u = 0.8$. The rightmost plot has a lower threshold $u_l = 0.4$.

The function `exi` returns estimates of the Extremal Index of a (stationary) stochastic process. The Extremal Index is defined in Chapter 3 of Leadbetter *et al.* (1983). A more informal treatment is given in Chapter 5 of Coles (2001). The extremal index can be estimated using the inverse of the average size of extreme clusters, where the cluster size is defined as the number of exceedances that it contains.

7 Fitting Univariate Distributions

This section presents functions that produce maximum likelihood estimates for some of the distributions introduced in Section 2. Peaks over threshold models are discussed in Section 8. Maximum likelihood estimates for bivariate extreme value distributions are discussed in Section 9. For illustrative purposes Sections 7, 8 and 9 use only simulated data. Three practical examples using the data sets `oxford`, `rain` and `sealevel` are given in Sections 10, 11 and 12 respectively.

The function `fgev` produces maximum likelihood estimates for the GEV distribution (4). The first argument should be a numeric vector containing data to be fitted. Missing values are allowed. If the argument `start` is given it should be a named list containing starting values, the names of which should be the parameters over which the likelihood is to be maximized. If `start` is omitted the routine attempts to find good starting values for the optimization using moment estimators.

If any of the parameters are to be set to fixed values, they can be given as separate arguments. For example, the Gumbel distribution (1) can be fitted using `shape = 0`. Arguments of the optimization function `optim` can also be specified. This includes the optimization method, which can be passed using the argument `method`. Two examples of the `fgev` function are given below.

```
> data1 <- rgev(1000, loc = 0.13, scale = 1.1, shape = 0.2)

> m1 <- fgev(data1)
> m1

Call: fgev(x = data1)
Deviance: 3650

Estimates
  loc  scale  shape
0.127  1.125  0.224
```

```
Standard Errors
   loc   scale   shape
0.0400  0.0321  0.0248

Optimization Information
  Convergence: successful
  Function Evaluations: 51
  Gradient Evaluations: 12

> m2 <- fgev(data1, loc = 0, scale = 1)
> m2

Call: fgev(x = data1, loc = 0, scale = 1)
Deviance: 3669

Estimates
shape
0.236

Standard Errors
 shape
0.0202

Optimization Information
  Convergence: successful
  Function Evaluations: 24
  Gradient Evaluations: 7
```

In the first example the likelihood is maximized over (`loc`, `scale`, `shape`). In the second example the likelihood is maximized over `shape`, with the location and scale parameters fixed at zero and one respectively. The maximum likelihood estimates from model `m1` are

```
> fitted(m1)
   loc  scale  shape
0.1271 1.1251 0.2244
```

The maximum likelihood estimators do not necessarily have the usual asymptotic properties, since the end points of the GEV distribution depend on the model parameters. Smith (1985) shows that the usual asymptotic properties hold when $\xi > -0.5$. When $-1 < \xi \leq -0.5$ the maximum likelihood estimators do not have the standard asymptotic properties, but generally exist. When $\xi \leq -1$ maximum likelihood estimators do not often exist. This occurs because of the large mass near the upper end point. The likelihood increases without bound as the upper end point is estimated to be closer and closer to the largest observed value. In terms of the reversed Weibull shape parameter α, the usual asymptotic properties hold when $\alpha > 2$, the asymptotic properties are not standard for $1 < \alpha \leq 2$, and maximum likelihood estimators do not often exist for $\alpha < 1$.

When the usual asymptotic properties hold (as here) the standard errors of the maximum likelihood estimates, approximated using the inverse of the observed information matrix, can be extracted from the fitted object using

```
> std.errors(m1)
```

```
    loc   scale   shape
0.03999 0.03214 0.02479
```

When the usual asymptotic properties do not hold the `std.errors` component will still be based on the inverse of the observed information matrix, but these values must be *interpreted with caution* (Smith, 1985).

Likelihood ratio tests can be performed using the function `anova`. We can compare the two models `m1` and `m2` to test the null hypothesis that the location parameter is zero and the scale parameter is one.

```
> anova(m1, m2)
Analysis of Deviance Table

   M.Df Deviance Df Chisq Pr(>chisq)
m1    3     3650
m2    1     3669  2  18.8    8.2e-05
```

The deviance difference, `deviance(m2)` minus `deviance(m1)`, is about 18.8, which yields a p-value of 8.2×10^{-5} when compared with a chi-squared distribution on two degrees of freedom. Diagnostic plots and profile deviances for fitted models can be constructed using the functions `plot`, `profile` and `profile2d` (see Section 10).

By default the maximum likelihood estimates are calculated under the assumption that the data to be fitted are the observed values of independent random variables $Z_1, \ldots, Z_n$, where $Z_i \sim \text{GEV}(\mu, \sigma, \xi)$ for each $i = 1, \ldots, n$. The `nsloc` argument allows non-stationary models of the form $Z_i \sim \text{GEV}(\mu_i, \sigma, \xi)$, where

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}.$$

The parameters $(\beta_0, \ldots, \beta_k)$ are to be estimated. In matrix notation $\boldsymbol{\mu} = \boldsymbol{\beta_0} + X\boldsymbol{\beta}$, where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^T$, $\boldsymbol{\beta_0} = (\beta_0, \ldots, \beta_0)^T$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$ and X is the $n \times k$ covariate matrix (excluding the intercept) with ijth element x_{ij}.

The `nsloc` argument must be a data frame containing the matrix X, or a numeric vector which is converted into a single column data frame with column name "trend". The column names of the data frame are used to derive names for the estimated parameters. This allows any of the $k + 3$ parameters $(\beta_0, \ldots, \beta_k, \sigma, \xi)$ to be set to fixed values within the optimization. The covariates must be (at least approximately) *centred and scaled*, not only for numerical reasons, but also because the starting value (if `start` is not given) for each corresponding coefficient is taken to be zero. When a linear trend is present in the data, the location parameter is often modelled as

$$\mu_i = \beta_0 + \beta_1 t_i,$$

where t_i is some centred and scaled version of the time of the ith observation. Non-stationary models are rarely fitted, but this is probably the most commonly used form of non-stationarity. More complex changes in μ may also be appropriate. For example, a change-point model

$$\mu_i = \beta_0 + \beta_1 x_i \qquad \text{where} \qquad x_i = \begin{cases} 0 & i \leq i_0 \\ 1 & i > i_0 \end{cases},$$

or a quadratic trend

$$\mu_i = \beta_0 + \beta_1 t_i + \beta_2 t_i^2.$$

See Sections 10 and 12 for examples of non-stationary modelling.

The function `fgev` also has an argument called `prob`. If `prob` $= p$ is passed a value in the interval [0,1], `fgev` again produces maximum likelihood estimates for the GEV distribution, but the model is re-parameterized from (μ, σ, ξ) to (z_p, σ, ξ), where z_p is the quantile corresponding to the upper tail probability p. This argument can be used to calculate and plot profile deviances of extreme quantiles (see Section 10). If `prob` is zero/one, then z_p is defined as the upper/lower end point $\mu - \sigma/\xi$, and ξ is restricted to the negative/positive axis. Under non-stationarity the model is re-parameterized from $(\beta_0, \beta_1, \ldots, \beta_k, \sigma, \xi)$ to $(z_p, \beta_1, \ldots, \beta_k, \sigma, \xi)$, so that z_p is the quantile corresponding to the upper tail probability p for the distribution obtained when all covariates are zero.

The `fextreme` function produces maximum likelihood estimates for the distributions (6) and (7) given an integer m and an arbitrary distribution function F. The first argument should be a numeric vector containing the data to be fitted, which should represent maxima (if the argument `largest` is `TRUE`, the default) or minima (if `largest` is `FALSE`). The argument `start` (which cannot be missing) should be a named list containing starting values, the names of which should be the parameters over which the likelihood is to be maximized. If any of the parameters are to be set to fixed values, they can be given as separate arguments. Arguments of the optimization function `optim` can also be specified. The example given below produces maximum likelihood estimates for the distribution (6), where $m = 365$ and F is the normal distribution.

```
> d2 <- rextreme(100, distn = "norm", mean = 0.56, mlen = 365)
# Simulate yearly maxima using normal distribution

> sv <- list(mean = 0, sd = 1)
> nm <- fextreme(d2, start = sv, distn = "norm", mlen = 365)
> fitted(nm)
 mean    sd
0.685 0.959
```

The `forder` function yields maximum likelihood estimates for the distribution (8) given integers m and $j \in \{1, \ldots, m\}$, and an arbitrary distribution function F. An example is given below, where $m = 365$, $j = 2$ and F is the normal distribution.

```
> d3 <- rorder(100, distn = "norm", mean = 0.56, mlen = 365, j = 2)
> sv <- list(mean = 0, sd = 1)
> nm2 <- forder(d3, sv, distn = "norm", mlen = 365, j = 2)
> fitted(nm2)
 mean    sd
0.483 1.042
```

8 Fitting Peaks Over Threshold Models

Suppose $X_1, \ldots, X_n$ is a sequence of independent and identically distributed random variables, with $M_n = \{X_1, \ldots, X_n\}$. Suppose that n is large, so that (assuming certain regularity conditions) the distribution of M_n is approximately GEV. Then for large enough u, the exceedances of the threshold u are approximately distributed as generalized Pareto, with location parameter u. The function `fpot` fits this distribution to the exceedances, and hence produces maximum likelihood estimates for the shape and scale parameters. The value of the threshold u must be specified by the user. It is typically chosen to be as small as possible, subject to the limit model providing a reasonable approximation.

The functions `mrlplot` and `tcplot`* produce diagnostic plots that facilitate the specification of

*Both of these functions are heavily based on code by Stuart Coles.

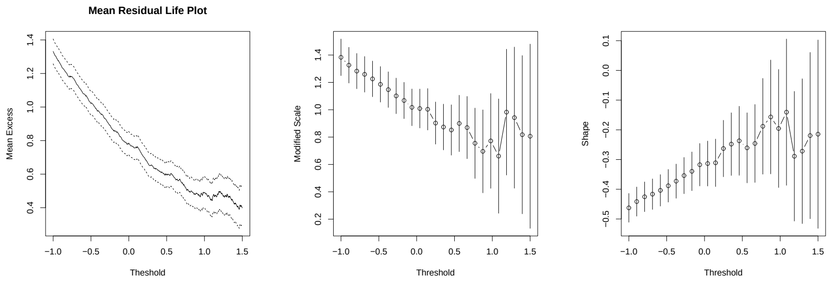

Figure 3: The identification of a threshold for the (generalized Pareto) peaks over threshold model. From left to right; the empirical mean residual life plot, modified scale parameter estimates and shape parameter estimates.

u. The function `mrlplot` produces the empirical mean residual life plot, which is a plot of the empirical mean of the excesses of u (i.e. the exceedances of u minus u), plotted against u. If the exceedances of a threshold u_0 are generalized Pareto, the empirical mean residual life plot should be approximately linear for all $u > u_0$. The function `tcplot` calculates maximum likelihood estimates for the shape and (modified) scale parameters using a number of different thresholds, and plots these estimates against u. If the exceedances of a threshold u_0 are generalized Pareto, the shape and (modified) scale parameters should be approximately constant with respect to all thresholds $u > u_0$. Threshold identification plots produced from the example given below are depicted in Figure 3. In this case, the threshold $u = 1$ was chosen.

The following code generates $n = 500$ independent standard normal random variables and fits the (generalized Pareto) peaks over threshold model to the exceedances of the threshold $u = 1$. The function `fpot` performs the fit. Many of the attributes of `fpot` are similar to those of `fgev`. In particular, the argument `start` can be given so that the user can specify his/her own starting values, and either of the `scale` or `shape` parameters can be set to fixed values by giving those parameters as arguments. For example, an exponential distribution for the excesses (or equivalently, a shifted exponential distribution for the exceedances) can be fitted using `shape = 0`. Arguments of the optimization function `optim` can also be specified. This includes the optimization method, which can be passed using the argument `method`.

```
> set.seed(50)
> tmp <- rnorm(500)
> mrlplot(tmp,  tlim = c(-1,1.5))
> tcplot(tmp, tlim = c(-1,1.5))

> pot1 <- fpot(tmp, 1)
> pot1

Call: fpot(x = tmp, threshold = 1)
Deviance: 40.5

Threshold: 1
Number Above: 76
Proportion Above: 0.152

Estimates
 scale   shape
 0.593  -0.211
```

```
[...]
```

The fitted model `pot1` gives the estimates for the scale and shape parameters of the generalized Pareto distribution fitted to the exceedances. Also given is the proportion of values above the threshold, or equivalently, the maximum likelihood estimate for the probability of an exceedance. Diagnostic plots and profile deviances for fitted models can be constructed using the functions `plot` and `profile` (see Section 11).

The peaks over thresholds model is typically extended to stationary[†] series via declustering, which corresponds to a filtering of dependent observations to obtain a set of threshold exceedances which are approximately independent. An empirical rule is used to identify clusters of exceedances, and the generalized Pareto model is then fitted to the cluster maxima, assuming those maxima to be independent. The empirical rule, as given in Section 6, is defined by the function `clusters`. A model of this form can be implemented by setting the logical argument `cmax` to `TRUE`. The clusters are identified using the threshold of the peaks over threshold model. Further arguments that affect the empirical rule for cluster identification can also be passed to `fpot`, including the clustering interval length `r` and the lower threshold `ulow`. An illustration of this technique is given below (though in practice the declustering of independent samples is unnecessary).

```
> pot2 <- fpot(tmp, 1, cmax = TRUE, r = 3, ulow = -0.1)
> pot2

Call: fpot(x = tmp, threshold = 1, cmax = TRUE, r = 3, ulow = -0.1)
Deviance: 40.46

Threshold: 1
Number Above: 76
Proportion Above: 0.152

Clustering Interval: 3
Lower Threshold: -0.1
Lower Clustering Interval: 1
Number of Clusters: 52
Extremal Index: 0.684

Estimates
 scale   shape
 0.732  -0.299

[...]
```

The Extremal Index is a quantity briefly discussed in Section 6. The estimate of the Extremal Index, 0.684, is simply the number of clusters (52) divided by the number of exceedances (76).

The function `fpot` also has an argument called `mper`. If `mper` $= m$ is passed a positive value, `fpot` again produces maximum likelihood estimates for the generalized Pareto model, but the model is re-parameterized from (σ, ξ) to (z_m, ξ), where z_m is the m-period return level, defined as follows. Let G be the fitted generalized Pareto distribution function, with location parameter equal to the specified threshold u, so that $1 - G(z)$ is the fitted probability of an exceedance over $z > u$ given an exceedance over u. The fitted probability of an exceedance over $z > u$ is therefore $p(1 - G(z))$, where p is the estimated probability of exceeding u, which is given by the empirical

[†] Further extensions to non-stationary models are not implemented.

proportion of exceedances. The m-period return level z_m satisfies $p(1 - G(z_m)) = 1/(mN\hat{\theta})$, where N is the number of observations per period, and $\hat{\theta}$ is the estimate of the extremal index if cluster maxima are fitted, with $\hat{\theta} = 1$ otherwise. The value N can be specified using the argument `npp`. For example, if observations are recorded weekly and `npp` $= 52$, then z_m is the m-year return level. If `mper` is `Inf`, then z_m is defined as the upper end point $u - \sigma/\xi$, and ξ is then restricted to be negative. The argument `mper` can be used to calculate and plot profile deviances of return levels (see Section 11).

The peaks over threshold model permits an alternative characterization in terms of point processes. Suppose again that $X_1, \dots, X_n$ is a sequence of independent and identically distributed random variables, with $M_n = \{X_1, \dots, X_n\}$, and that n is large, so that (assuming certain regularity conditions) the distribution of M_n is approximately GEV(μ, σ, ξ), with (possibly infinite) end points[‡] z_- and z_+. Then for large enough $u > z_-$, the sequence $\{X_1, \dots, X_n\}$ viewed on the interval (u, z_+) can be approximated by a non-homogeneous Poisson process (Coles, 2001). The approximation leads to a likelihood for (μ, σ, ξ), and hence maximum likelihood estimates can be obtained. The likelihood can be easily adjusted so that the maxima of a given (large) number $N \leq n$ of random variables is approximately distributed as GEV(μ, σ, ξ), so that e.g. if observations are recorded weekly and $N = 52$, then (μ, σ, ξ) corresponds to the distribution of annual maxima. The point process characterization can be fitted using the `fpot` function with `model = "pp"`. The value N can by specified using the argument `npp`. If `npp` is unspecified the default value $N = n$ is used. The following code uses the point process characterization to fit a peaks over threshold model to the simulated data `tmp`. The models `pot3` and `pot4` are equivalent; the estimates in `pot3` correspond to the GEV distribution for the maxima of the data set, whereas those in `pot4` correspond to the GEV distribution for annual maxima, assuming the observations are recorded daily.

```
> pot3 <- fpot(tmp, 1, model = "pp", npp = 500)
> pot4 <- fpot(tmp, 1, model = "pp", npp = 365.25)
> fitted(pot3)
    loc   scale   shape
 2.6839  0.2380 -0.2108
> fitted(pot4)
    loc   scale   shape
 2.6065  0.2542 -0.2108

> fitted(pot1)
 scale  shape
 0.593 -0.211
```

Also given above is the parameter estimates for the model `pot1`, fitted using the generalised Pareto characterization. Let $(\tilde{\sigma}, \tilde{\xi})$ denote the scale and shape parameters of the GPD. The relationship between the two characterizations is then given by $\tilde{\xi} = \xi$ and $\tilde{\sigma} = \sigma + \xi(u - \mu)$, where u is the threshold. This relationship can be seen in the above estimates. Under the generalized Pareto characterization, the parameter $\tilde{\sigma} - \tilde{\xi}u$ is referred to as the modified scale parameter, as plotted in the centre panel of Figure 3. Unlike $\tilde{\sigma} = \tilde{\sigma}(u)$, the modified scale parameter does not depend on the threshold u.

[‡] If $\xi > 0$, $z_- = \mu - \sigma/\xi$ and $z_+ = \infty$. If $\xi < 0$, $z_- = -\infty$ and $z_+ = \mu - \sigma/\xi$. If $\xi = 0$, the expressions given are all defined by continuity, with $z_- = -\infty$ and $z_+ = \infty$.

Bivariate Model	Constraints
Logistic	$0.1 \le \alpha \le 1$
Asymmetric Logistic	$0.1 \le \alpha \le 1$, $0.001 \le \theta_1, \theta_2 \le 1$
Hüsler-Reiss	$0.2 \le \lambda \le 10$
Negative Logistic	$0.05 \le r \le 5$
Asymmetric Negative Logistic	$0.05 \le r \le 5$, $0.001 \le \theta_1, \theta_2 \le 1$
Bilogistic	$0.1 \le \alpha, \beta \le 0.999$
Negative Bilogistic	$0.1 \le \alpha, \beta \le 20$
Coles-Tawn	$0.001 \le \alpha, \beta \le 30$

Table 1: For numerical reasons the parameters of each model are subject to the artificial constraints depicted here.

9 Fitting Bivariate Extreme Value Distributions

The function fbvevd produces maximum likelihood estimates for each of the eight bivariate models introduced in Section 3. The first argument should be a numeric matrix (or a data frame) with two columns containing the data to be fitted. Missing values are allowed. If the argument start is given it should be a named list containing starting values, the names of which should be the parameters over which the likelihood is to be maximized. If start is omitted the routine attempts to find good starting values for the optimization using maximum likelihood estimators under the assumption of independence. If any of the parameters are to be set to fixed values, they can be given as separate arguments. Arguments of the optimization function optim can also be specified.

The nsloc1 and nsloc2 arguments allow non-stationary modelling of the location parameters on the first and second margins respectively. They should be used in the same manner as the nsloc argument of fgev. Examples of bivariate models with non-stationary margins are given in Section 12.

For numerical reasons the parameters of each model are subject to the artificial constraints depicted in Table 1. The scale parameters on each GEV margin are artificially constrained to be greater than or equal to 0.01. These constraints only apply to the functions discussed in this section.

The first example given below produces maximum likelihood estimates for the (symmetric) logistic model. The second example constrains the model at independence (where dep = 1). The estimates produced in the second example are the same as those that would be produced if fgev was separately applied to each margin.

```
> bvdata <- rbvlog(100, dep = 0.6, mar1 = c(1.2,1.4,0), mar2 = c(1,1.6,0.1))

> m1 <- fbvevd(bvdata, model = "log")
> m1

Call: fbvevd(x = bvdata, model = "log")
Deviance: 728.5
AIC: 742.5

Estimates
   loc1   scale1   shape1     loc2   scale2   shape2      dep
 1.2121   1.3831  -0.1813   0.8404   1.4005   0.0834   0.7202
```

```
Standard Errors
  loc1  scale1  shape1    loc2  scale2  shape2     dep
0.1540  0.1091  0.0673  0.1537  0.1144  0.0614  0.0624

Dependence Structure
  Dependence One: 0.3526
  Dependence Two: 0.4824
  Asymmetry: 0

Optimization Information
  Convergence: successful
  Function Evaluations: 47
  Gradient Evaluations: 10

> m2 <- fbvevd(bvdata, model = "log", dep = 1)

> fitted(m2)
    loc1   scale1   shape1     loc2   scale2   shape2
 1.22311  1.37763 -0.19140  0.83671  1.40829  0.08683

> std.errors(m2)
   loc1   scale1  shape1    loc2  scale2  shape2
0.15429 0.10886 0.07245 0.15646 0.11625 0.06704

> c(logLik(m2), deviance(m2), AIC(m2))
[1] -376  752  764
```

The discussion in Section 7 regarding the properties of maximum likelihood estimators for the GEV distribution also applies to all bivariate models. The usual asymptotic properties hold only when the shape parameters on both margins are greater than -0.5. When the usual asymptotic properties do not hold the `std.errors` component will still be based on the inverse of the observed information matrix, but these values must be *interpreted with caution* (Smith, 1985).

Fitted models contain three values that summarise the fitted dependence structure. Let $A(\cdot)$ denote the dependence function (16). The three values are given by

$$\text{Dependence One} = \chi = 2\{1 - A(1/2)\}$$

$$\text{Dependence Two} = \psi_d = 4\int_0^1 1 - A(x)\,\mathrm{d}x$$

$$\text{Asymmetry} = \psi_a = \frac{4}{3 - 2\sqrt{2}} \int_0^{1/2} A(x) - A(1-x)\,\mathrm{d}x$$

The two measures of dependence χ (Coles *et al.*, 1999) and ψ_d are contained in the closed interval [0,1]. At independence $\chi = \psi_d = 0$, and at complete dependence $\chi = \psi_d = 1$. The measure of asymmetry ψ_a is contained in the closed interval [-1,1]. If $A(\cdot)$ is symmetric $\psi_a = 0$. The logistic, Hüsler-Reiss and negative logistic models are always symmetric, and hence always satisfy $\psi_a = 0$. Any value $\psi_a \in (-0.2, 0.2)$ corresponds to a dependence structure that is close to symmetric.

Diagnostic plots and profile deviances for fitted models can be constructed using the functions `plot`, `profile` and `profile2d` (see Section 12). The function `anova` performs likelihood ratio tests. The null hypothesis of the test performed below specifies that the margins are Gumbel distributions (`shape1` = `shape2` = 0). The deviance of the constrained model is compared with the deviance of the unconstrained model, and the p-value is calculated to be 0.78. The hypothesis would not be rejected at any reasonable significance level.

```
> m3 <- fbvevd(bvdata, model = "log", shape1 = 0, shape2 = 0)
> anova(m1, m3)
Analysis of Deviance Table

   M.Df Deviance Df Chisq Pr(>chisq)
m1    7      708
m3    5      708  2   0.5       0.78
```

In the following example I attempt to fit the asymmetric logistic model to the simulated data set used above, which is known to be distributed as symmetric logistic.

```
> m4 <- fbvevd(bvdata, model = "alog")
> fitted(m4)
   loc1  scale1   shape1    loc2  scale2  shape2    asy1    asy2     dep
1.20969 1.39280 -0.18529 0.84210 1.38311 0.07725 0.83313 0.99957 0.69248
```

A boundary of the parameter space has been reached; the maximum likelihood estimate for the second asymmetry parameter is (effectively) one. This may cause difficulties for the optimizer. There are two solutions to this problem: the second asymmetry parameter can be fixed at one, or the `L-BFGS-B` method can be used. The `L-BFGS-B` method allows box-constraints using the arguments `lower` and `upper`. The following snippet illustrates both approaches.

```
> mb <- fbvevd(bvdata, model = "alog", asy2 = 1)
> round(fitted(mb), 3)
  loc1 scale1 shape1   loc2 scale2 shape2   asy1    dep
 1.212  1.385 -0.176  0.834  1.396  0.086  0.867  0.693

> up <- c(rep(Inf, 6), 1, 1, 1)
> mb <- fbvevd(bvdata, model = "alog", method = "L-BFGS-B", upper = up)
> round(fitted(mb), 3)
  loc1 scale1 shape1   loc2 scale2 shape2   asy1   asy2    dep
 1.212  1.385 -0.176  0.834  1.396  0.086  0.867  1.000  0.693
```

10 Example: Oxford Temperature Data

The numeric vector `oxford` contains annual maximum temperatures (in degrees Fahrenheit) at Oxford, England, from 1901 to 1980. It is included in the evd package, and can be made available using `data(oxford)`. The data has previously been analysed by Tabony (1983).

I begin by plotting the data. The assumptions of stationarity and independence seem sensible, given the plot generated using the code below, depicted in Figure 4.

```
> data(oxford) ; ox <- oxford
> plot(1901:1980, ox, xlab = "year", ylab = "temperature")
```

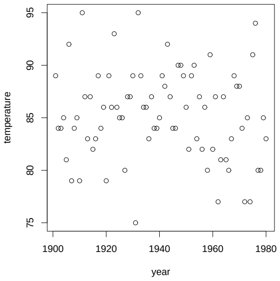

Figure 4: The `oxford` data.

The following code fits two models based on the GEV distribution. The first model assumes stationarity. The second model allows for a trend term in the location parameter (even though the plot appears to show that this is unnecessary). The `nsloc` argument is centred and scaled so that the intercept `loc` represents the location parameter in 1950 and the trend `loctrend` represents the increase in the location parameter (or decrease, if negative) over a period of 100 years.

```
> ox.fit <- fgev(ox)

> tt <- (1901:1980 - 1950)/100
> ox.fit.trend <- fgev(ox, nsloc = tt)

> fitted(ox.fit.trend)
     loc loctrend    scale    shape
 83.6617  -1.8812   4.2233  -0.2841

> std.errors(ox.fit.trend)
     loc loctrend    scale    shape
 0.55566  1.96754  0.36504  0.07068
```

The trend term not statistically significant (at any reasonable level). The stationary model `ox.fit` is retained for further analysis.

```
> ox.fit

Call: fgev(x = oxford)
Deviance: 457.8

Estimates
   loc   scale   shape
83.839   4.260  -0.287

Standard Errors
   loc   scale   shape
0.5231  0.3658  0.0683

Optimization Information
  Convergence: successful
```

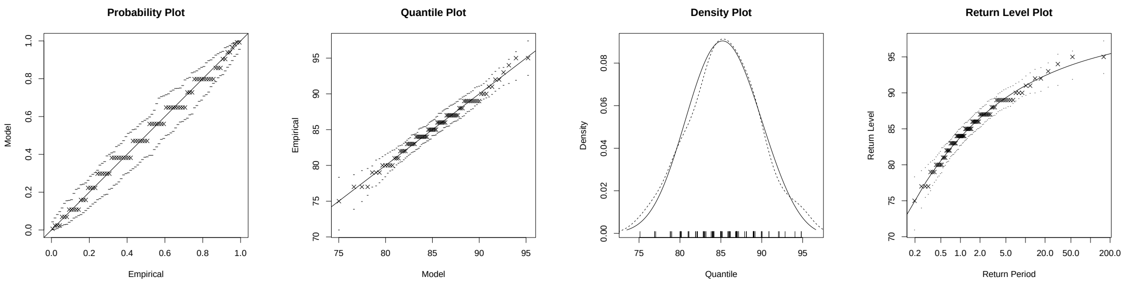

Figure 5: Diagnostic plots for the model `ox.fit`.

```
  Function Evaluations: 29
  Gradient Evaluations: 11
```

The fitted shape is negative, so the fitted distribution is Weibull. It is often of interest to test the hypothesis that the shape is zero (the Gumbel distribution). A 95% confidence interval for the shape parameter can be constructed using $-0.287 \pm 1.96 \times 0.0683$. The corresponding Wald test can be performed by dividing the maximum likelihood estimate by its standard error. The Wald test would be rejected at significance level 0.05 since the 95% confidence interval does not contain zero. A likelihood ratio test is performed in the following snippet. The hypothesis is rejected at any significance level above 0.00053.

```
> ox.fit.gum <- fgev(ox, shape = 0)
> anova(ox.fit, ox.fit.gum)
Analysis of Deviance Table

           M.Df Deviance Df Chisq Pr(>chisq)
ox.fit        3      458
ox.fit.gum    2      470  1    12    0.00053
```

Diagnostic plots can be produced using `plot`. The plots produced by the following line of code, depicted in Figure 5, compare parametric distributions, densities and quantiles to their empirical counterparts (see the documentation for `plot.uvevd` for details of the construction of each plot, and for an explanation of the argument `jitter`).

```
> plot(ox.fit, jitter = TRUE)
```

The horizontal bars on the P-P, Q-Q and return level plots represent simulated (pointwise) 95% confidence intervals. The model `ox.prof` is seen to be a good fit. The fitted density is close to the non-parametric estimator, and most points lie within the confidence intervals. Profile deviances (minus twice the profile likelihood) of the parameters can be plotted using

```
> ox.prof <- profile(ox.fit)
> pcis <- plot(ox.prof)
> pcis[["scale"]]
  lower upper
  3.643 5.117
```

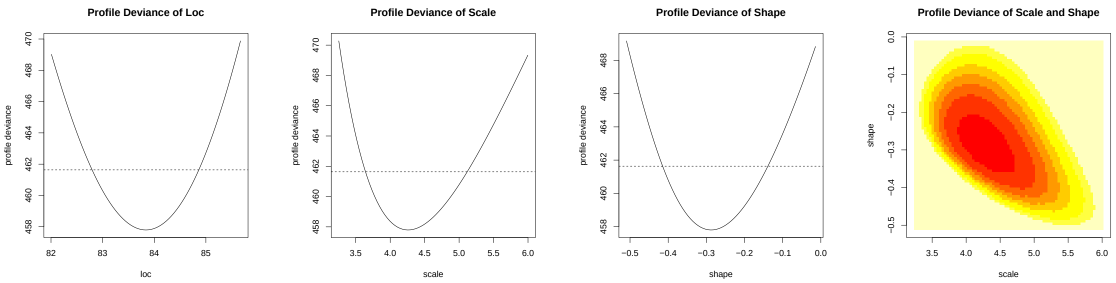

Figure 6: Profile deviance surfaces for the model `ox.fit`.

This produces the first three plots within Figure 6. A horizontal line is (optionally) drawn on each plot so that the intersection of the line with the profile deviance yields a profile confidence interval, with (default) confidence coefficient 0.95. The end points of the intervals can be derived by assigning the expression `plot(ox.prof)` to an object, as above. The profile confidence intervals for the location and shape parameters are approximately the same as the intervals constructed using their standard errors, since the profile deviances are approximately symmetric. The profile deviance for the scale parameter is asymmetric; both end points of the profile confidence interval $(3.64, 5.12)$ are larger than the corresponding end points of the interval $(3.54, 4.98)$, constructed using the standard error.

The joint profile deviance of the scale and shape parameters (which are typically negatively correlated) can be plotted using

```
> ox.prof2d <- profile2d(ox.fit, ox.prof, which = c("scale", "shape"))
> plot(ox.prof2d)
```

This produces the image plot in the left panel of Figure 6.* The colours of the image plot represent confidence sets with different confidence coefficients. By default, the lightest colour (ignoring the background colour) represents a confidence set with coefficient 0.995; the darkest colour represents a confidence set with coefficient 0.5.

Let G be the GEV distribution function, and let $G(z_p) = 1 - p$, so that

$$z_p = \begin{cases} \mu - \frac{\sigma}{\xi}[1 - \{-\log(1-p)\}^{-\xi}] & \xi \neq 0 \\ \mu - \sigma \log\{-\log(1-p)\} & \xi = 0, \end{cases} \tag{18}$$

is the quantile corresponding to the upper tail probability p. The profile deviance for $z_{0.1}$ can be plotted using the following. The argument $\texttt{prob} = p$ reparameterizes the GEV distribution so that `fgev` produces maximum likelihood estimates for (z_p, σ, ξ).

```
> ox.qfit <- fgev(ox, prob = 0.1)
> ox.qprof <- profile(ox.qfit, which = "quantile")
> plot(ox.qprof)
```

Figure 7 shows profile deviances for $z_{0.1}$, $z_{0.01}$ and $z_{0.001}$. The extent of the asymmetry in the profile deviance surface increases for decreasing (small) p. This is to be expected, since the data provide increasingly weaker information in the (upper) tail of the fitted distribution. If $\texttt{prob} = p$ is zero, then z_p is the upper end point of the GEV distribution, given by $\mu - \sigma/\xi$ when $\xi < 0$. The profile deviance for z_0 can be plotted using the following code.

*Assuming the package **akima** is available. If not, the image plot will look 'blocky', because bivariate interpolation will not be performed.

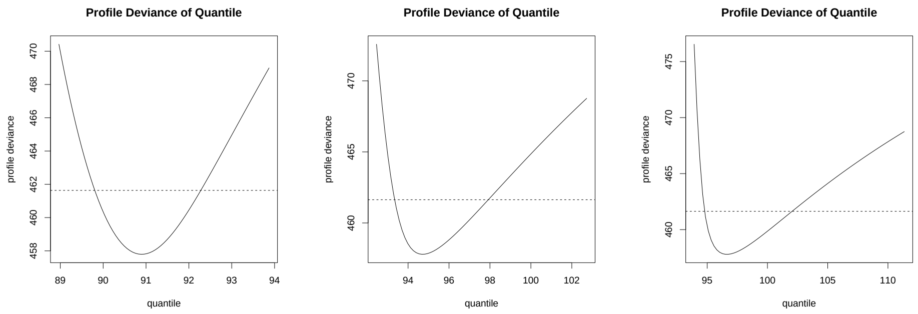

Figure 7: Profile deviance surfaces for $z_{0.1}$, $z_{0.01}$ and $z_{0.001}$.

```
> ox.qfit <- fgev(ox, prob = 0)
> ox.qprof <- profile(ox.qfit, which = "quantile", conf = 0.99)
> pcis <- plot(ox.qprof, ci = c(0.95, 0.99))
> pcis[["quantile"]]
     lower upper
0.95 95.78 113.0
0.99 95.49 131.8
```

The argument `conf` of the function `profile` controls the range of the profile trace. The profile trace is constructed so that profile confidence intervals with confidence coefficients `conf` or less can be derived from it. By default, `conf` = 0.999, though a smaller value is often appropriate when the profile deviance exhibits strong asymmetry. The 95% and 99% profile confidence intervals for the upper end point z_0 are derived as (95.8,113.0) and (95.5,131.8) respectively.

11 Example: Rainfall Data

The numeric vector `rain` contains 17531 daily rainfall accumulations at a location in south-west England, apparently* recorded over the period 1914 to 1962. The data is not included in the evd package, but it is available in the ismev package, which can be downloaded from CRAN. As usual, the package can be loaded using `library(ismev)`, and the data can be made available using `data(rain)`. The plot of the data given in Figure 1.7 of Coles (2001) shows that an assumption of stationarity is sensible. The example given here follows Coles (2001), pages 84–86 and 134.

```
> mrlplot(rain, tlim = c(0,85), nt = 100)
> par(mfrow = c(2,1))
> tcplot(rain, tlim = c(0,50), nt = 20)
> potgp <- fpot(rain, 30, npp = 365.25)
> potpp <- fpot(rain, 30, model = "pp", npp = 365.25)
> potpp2 <- fpot(rain, 30, model = "pp", npp = 365.25, cmax = TRUE, r = 7)
> clusters(rain, 30, r = 7, cmax = TRUE)
```

The first three lines of code produce the threshold diagnostic plots given in pages 80 and 85 of Coles (2001), who subsequently decides to work with the threshold $u = 30$. The models `potgp` and `potpp` use the generalized Pareto and point process characterization respectively. Both

*The length of the dataset would suggest otherwise.

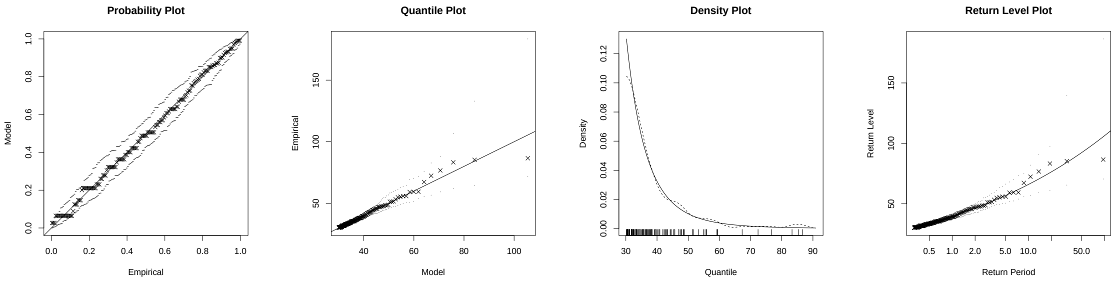

Figure 8: Diagnostic plots for the peaks over threshold model for daily rain data.

characterizations report that 152 observations lie above the threshold, giving an exceedance probability estimate of 0.00867. The estimates and standard errors of the parameters of the models potgp and potpp agree with those given in pages 85 and 134 of Coles (2001) respectively. The model potpp2 again uses the point process characterization, but now applied to cluster maxima, where clusters are defined using a clustering interval length of seven. As there is little sign of clustering in the data, this leads to relatively small changes in the parameter estimates, and relatively small increases in the standard errors. The final line of code calls the function clusters (see Section 6) in order to produce the cluster maxima that were used for the fitting of model potpp2.

Diagnostic plots can be produced using plot. The characterizations produce identical plots, so that both plot(potgp) and plot(potpp) produce the plots given in Figure 8. The plots compare parametric distributions, densities and quantiles to their empirical counterparts (see the documentation for plot.uvevd for details of the construction of each plot). The x-axis of the return level plot gives return periods in units of years, since in each of the models we specified the number of observations per period as npp = 365.25. The horizontal bars on the P-P, Q-Q and return level plots represent simulated (pointwise) 95% confidence intervals. The model potgp (or potpp) is seen to be a good fit. The fitted density tail is close to the non-parametric estimator, and most points lie within the confidence intervals.

Profile deviances of the shape parameter and the 100-year return level are given in Figure 9. They can be plotted using the following code. The argument mper $= m$ reparameterizes the GEV distribution so that fpot produces maximum likelihood estimates for (z_m, ξ), where z_m is the m period return level, as defined in Section 8. Horizontal lines denoting 95% profile confidence intervals are depicted on each plot. The end points of profile confidence intervals can be derived by assigning the plotting expression to an object (see Section 10).

```
potgp2 <- fpot(rain, 30, npp = 365.25, mper = 100)
prgp2 <- profile(potgp2)
plot(prgp2)
```

12 Example: Sea Level Data

The sealevel data frame (Coles and Tawn, 1990) has two columns containing annual sea level maxima from 1912 to 1992 at Dover and Harwich, two sites on the coast of Britain. It contains 39 missing maxima in total; nine at Dover and thirty at Harwich. There are three years for which the annual maximum is not available at either site.

I begin by plotting the data, using the code below. The resulting plots are given in Figure 10. The plot of the Harwich maxima against the Dover maxima depicts a reasonable degree of

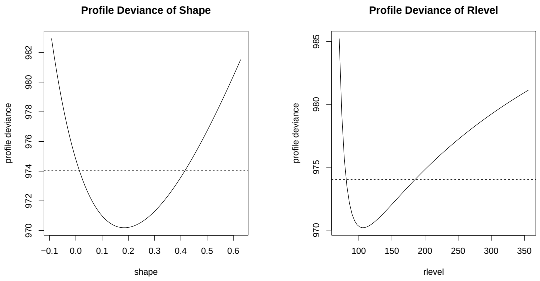

Figure 9: Profile deviances for the shape parameter and 100-year return level in the peaks over threshold model for daily rain data.

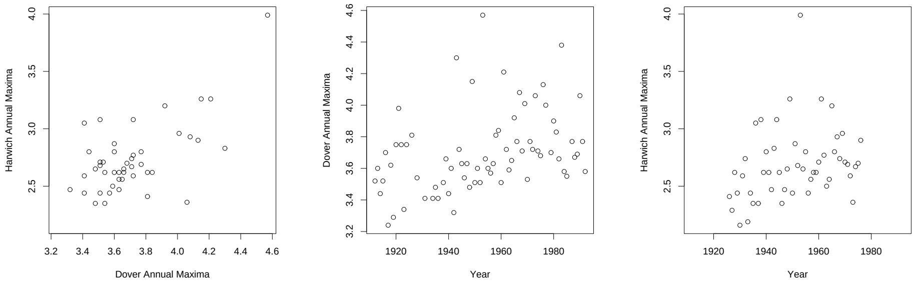

Figure 10: From left to right; Harwich Maxima vs Dover Maxima, Dover Maxima vs Year and Harwich Maxima vs Year.

dependence. The outlier corresponds to the 1953 flood resulting from a storm passing over the South-East coast of Britain on 1st February. The Harwich and Dover maxima both appear to increase with time.

```
> data(sealevel) ; sl <- sealevel
> plot(sl, xlab = "Dover Annual Maxima", ylab = "Harwich Annual Maxima")
> plot(1912:1992, sl[,1], xlab = "Year", ylab = "Dover Annual Maxima")
> plot(1912:1992, sl[,2], xlab = "Year", ylab = "Harwich Annual Maxima")
```

The following three expressions fit (symmetric) logistic models. The first model incorporates linear trend terms on both marginal location parameters. The second model incorporates a linear trend on the Dover margin only. The third model assumes stationarity. The `nsloc1` and `nsloc2` arguments are centred and scaled so that the intercepts `loc1` and `loc2` represent the marginal location parameters in 1950 and the linear trend parameters `loc1trend` and `loc2trend` represent the increase in the marginal location parameters (or decrease, if negative) over a period of 100 years.

```
> tt <- (1912:1992 - 1950)/100
> m1 <- fbvevd(sl, model = "log", nsloc1 = tt, nsloc2 = tt)
> m2 <- fbvevd(sl, model = "log", nsloc1 = tt)
> m3 <- fbvevd(sl, model = "log")
```

I’ll leave you to analyse the models in detail. In particular, notice how the trend terms affect the parameter estimates. Marginal Weibull distributions (negative shapes) are estimated when the trends are not included, but marginal Fréchet distributions (positive shapes) are estimated upon their inclusion.

The maximum likelihood estimates of the parameters can be compared with their standard errors to perform Wald tests or construct confidence intervals. Likelihood ratio tests are performed in the following snippet. The p-values confirm the statistical significance of the linear trend terms.

```
> anova(m1, m2, m3)
Analysis of Deviance Table

   M.Df Deviance Df Chisq Pr(>chisq)
m1    9    -36.5
m2    8    -29.2  1  7.26      0.007
m3    7     -9.7  1 19.56    9.7e-06
```

Quadratic trends for the location parameter on either or both margins can be incorporated using the following code. Further testing, using the models generated below, suggests that a quadratic trend may be implemented for the location parameter on the Harwich margin. Despite this, I retain the model m1 for further analysis.

```
> tdframe <- data.frame(trend = tt, quad = tt^2)
> m4 <- fbvevd(sl, model = "log", nsloc1 = tdframe, nsloc2 = tt)
> m5 <- fbvevd(sl, model = "log", nsloc1 = tt, nsloc2 = tdframe)
> m6 <- fbvevd(sl, model = "log", nsloc1 = tdframe, nsloc2 = tdframe)
```

The code given below compares two logistic models that are nested within m1. Model m7 assumes independence. The maximum likelihood estimates are the same as those that would be produced if fgev was separately applied to each margin. The deviance increase with respect to model m1 is calculated to be 13.6. *The asymptotic distribution of the deviance increase is not chi-squared.* The distribution is non-regular because the dependence parameter in the restricted (independence) model is fixed at the edge of the parameter space. Testing for the (symmetric) logistic model within the asymmetric logistic model also leads to non-regular behaviour. Tawn (1988) discusses non-regular testing procedures for bivariate extreme value models. In this case the increase in deviance is clearly too large to consider independence as a viable model.

Model m8 assumes that both marginal shape parameters are zero (or equivalently, that both marginal distributions are Gumbel). A likelihood ratio test of this hypothesis provides a p-value of 0.72. The hypothesis would not be rejected at any reasonable significance level.

```
> m7 <- fbvevd(sl, model = "log", nsloc1 = tt, nsloc2 = tt, dep = 1)
> anova(m1, m7)
# The asymptotic distribution of the deviance difference is non-regular
Analysis of Deviance Table

   M.Df Deviance Df Chisq Pr(>chisq)
m1    9    -36.5
m7    8    -22.9  1  13.6    0.00023

> m8 <- fbvevd(sl, "log", nsloc1 = tt, nsloc2 = tt, shape1 = 0, shape2 = 0)
> anova(m1, m8)
Analysis of Deviance Table
```

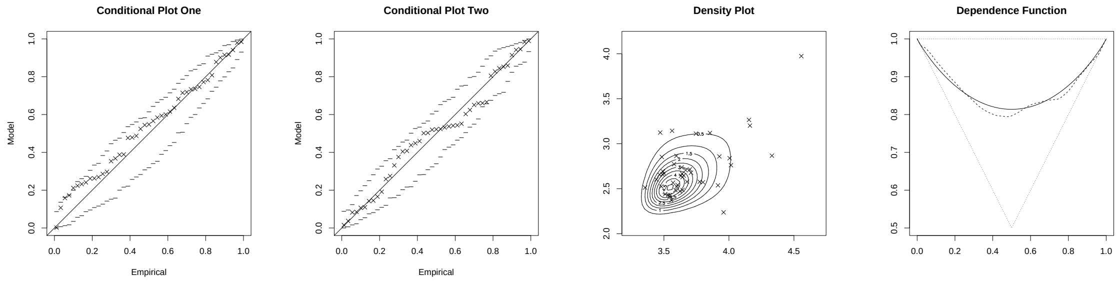

Figure 11: Diagnostic plots for the dependence structure of model `m1`.

```
   M.Df Deviance Df Chisq Pr(>chisq)
m1    9    -36.5
m8    7    -35.8  2  0.67       0.72
```

Diagnostic plots for the fitted (generalized extreme value) marginal distributions can be produced using `plot` with `mar = 1` or `mar = 2`. The plots produced are of the same structure as those given in Section 10. Diagnostic plots for the fitted dependence structure can be produced using `plot`, as shown in Figure 11 for the model `m1`.

```
> plot(m1, mar = 1)
> plot(m1, mar = 2)
> plot(m1)
```

The plots in Figure 11 compare parametric conditional distributions, densities and dependence functions to empirical counterparts (see the documentation for `plot.bvevd` for details of the construction of each plot). The horizontal bars on the conditional P-P plots represent simulated (pointwise) 95% confidence intervals. The model `m1` fits the data reasonably well. There are some minor deviations within the conditional P-P plots, but they do not represent a serious departure of the empirical estimates from the fitted model. The profile deviance (minus twice the profile likelihood) of the dependence parameter can be plotted using the following. The argument `xmax` denotes the upper bound of the dependence parameter `dep`.

```
> m1.prof <- profile(m1, which = "dep", xmax = 1)
> pcis <- plot(m1.prof)
> pcis[["dep"]]
   lower  upper
  0.5282 0.8865
```

This produces the plot in Figure 12. A horizontal line is (optionally) drawn so that the intersection of the line with the profile deviance yields a profile confidence interval, with (default) confidence coefficient 0.95. The end points of the interval can be derived by assigning the expression `plot(m1.prof)` to an object, as above.

Further analysis with models other than the (symmetric) logistic yields the following conclusions. The two models in Section 3 that include three parameters with which to describe the dependence structure (the asymmetric logistic and asymmetric negative logistic) are inappropriate. In both cases, the maximum likelihood estimate for the parameter `dep` is at an artificial boundary, because the fitted model is close to a distribution (obtained in the limit) which contains a singular

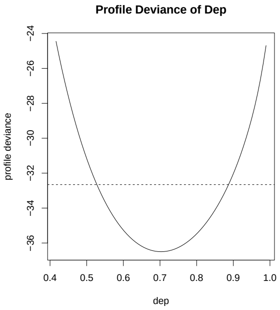

Figure 12: The profile deviance of the dependence parameter from model `m1`.

component. This is clearly illustrated in the density plots of the fitted models, which both depict a ridge of mass extending towards the 1953 outlier. The logistic and the bilogistic models have the lowest deviance of all one and two parameter models respectively. The fitted bilogistic model has $\psi_a = -0.02$, so the fitted dependence structure is almost symmetric, and hence the logistic model would appear to be preferable. In fact, the fitted bilogistic model (13) is almost logistic, because the α and β parameter estimates are almost equal, and hence the difference between the model deviances is less than 0.01.

Appendix A: Simulation Study

In this Appendix we use the tools in the package to perform a simulation study to examine the small sample properties of non-parametric estimators for the dependence function $A(\cdot)$ of the bivariate extreme value distribution. The estimators referred to in this Appendix are defined in the documentation file for the function `abvnonpar`.

Simulation studies of this form (e.g. Hall and Tajvidi, 2000) typically use the known marginal parameters $(\mu_1, \sigma_1, \xi_1, \mu_2, \sigma_2, \xi_2)$ within the transformations (17). In practice, these parameters need to be estimated. In this study we seek to replicate the behaviour of the estimators when applied to real data, and we have therefore estimated the marginal parameters by maximum likelihood.

Figure 13 depicts the behaviour of the estimators of Capéraà *et al.* (1997), Pickands (1981) and Tiago de Oliveira (1997), which we subsequently denote by A_c, A_p and A_t respectively. The estimators of Deheuvels (1991) and Hall and Tajvidi (2000) are not considered, as they produce plots that are indistinguishable from those of A_p. The first, second and third columns of the figure employ simulations from (symmetric) logistic distributions, with α equal to 0.5, 0.75 and 1 respectively. Standard Gumbel marginal distributions were used in each case. The figure shows that the estimator A_t is abysmal when estimating dependence functions with very strong ($\alpha = 0.5$) or very weak ($\alpha = 1$) levels of dependence. The estimators A_c and A_p give more consistent performances across different levels of dependence. The estimator A_c appears to outperform A_p, as the estimates of the former appear to cluster more tightly around the true dependence function for each $\alpha = 0.5, 0.75, 1$. The plots can easily be generated, using e.g.

```
> dep <- 0.5 ; method <- "cfg"
> abvpar(dep = dep, plot = TRUE, lty = 0)
> set.seed(44)
> for(i in 1:50) {
    sdt <- rbvevd(100, dep = dep)
```

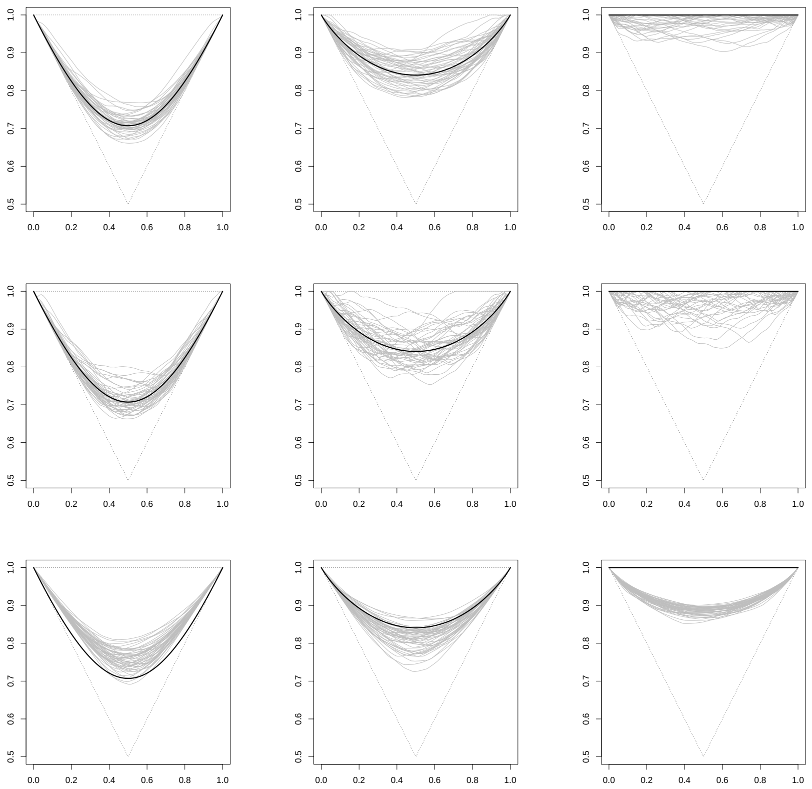

Figure 13: Simulated non-parametric dependence function estimates. The grey lines represent estimates derived using the estimators A_c (top row), A_p (middle row) and A_t (bottom row). The thick black lines represent the true dependence functions, which are (symmetric) logistic models with dependence parameters 0.5 (first column), 0.75 (second column) and 1 (third column).

```
    abvnonpar(data = sdt, add = TRUE, method = method, col = "grey")
  }
> abvpar(dep = dep, add = TRUE, lwd = 3)
```

which generates the plot in the top left corner. Only the first line of code needs to be changed in order to produce the remaining plots. The second line of code establishes the plotting region. The simulation is performed in the `for` loop, and the last line adds the true dependence function to the plot. The `set.seed` function sets the seed of the random generator, which ensures that the simulated data sets used for each plot are comparable.

Let $A_n(\cdot)$ be any estimator of $A(\cdot)$. Table 2 gives median integrated absolute errors for various non-parametric dependence function estimators. The table was constructed as follows. For $\alpha = 0.5, 0.75, 1$ we simulated 1000 datasets containing $n = 25, 100$ bivariate observations, using standard Gumbel margins. Then for each of the 1000 datasets we estimated the integrated absolute error $\int_0^1 |A_n(x) - A(x)|\,\mathrm{d}x$. The table contains the median of the 1000 values, for each value of α and n. We have extended the number of estimators to include the convex minorants of A_c and A_p, which we denote by A_c^* and A_p^*. The convex minorant of A_t is identical to A_t, because A_t is always convex.

The table again shows the poor performance of A_t when $\alpha = 0.5$, and particularly when $\alpha = 1$. A_t is the best estimator when $\alpha = 0.75$, which is not surprising given that the estimator only yields adequate estimates at mid-range levels of dependence. The estimator A_c outperforms A_p, confirming the impression given by Figure 13. Taking the convex minorant of A_c or A_p leads

	$n = 25$			$n = 100$		
	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
A_c	210	415	110	104	198	62
A_c^*	205	363	340	103	194	168
A_p	243	469	211	134	242	113
A_p^*	218	357	554	126	215	285
A_t	393	189	983	334	155	830

Table 2: Median integrated absolute errors × 10^4 for non-parametric estimates of the dependence function of the bivariate extreme value distribution, using datasets containing $n = 25, 100$ bivariate observations, simulated from the (symmetric) logistic model with dependence parameter $\alpha = 0.5, 0.75, 1$. The estimators A_c^* and A_p^* are the convex minorants of A_c and A_p respectively.

to an improvement for $\alpha = 0.5$ and $\alpha = 0.75$, but a considerable worsening for $\alpha = 1$. This worsening is expected, since taking the convex minorant always leads to estimates of stronger dependence. The values in the table can be generated using e.g.

```
> dep <- 0.5 ; n <- 25 ; method <- "cfg" ; cv <- FALSE
> nn <- 100 ; x <- (1:nn)/(nn + 1)
> a <- abvpar(x, dep = dep)
> iae <- numeric(1000)
> set.seed(44)
> for(i in 1:1000) {
    sdt <- rbvevd(n, dep = dep)
    anp <- abvnonpar(x, data = sdt, method = method, convex = cv)
    iae[i] <- sum(abs(a - anp))/nn
  }
> round(10^4 * median(iae))
```

which generates the value in the top left corner. Only the first line of code needs to be changed in order to produce the remaining values. The integrated absolute error is estimated by evaluating the absolute difference between true dependence function and the non-parametric estimate at nn $= 100$ equally spaced points in the interval $[0, 1]$. The function `numeric` merely initializes the object `iae` to be a vector of 1000 zeros.

References

Capéraà, P., Fougères, A.-L. and Genest, C. (1997) A non-parametric estimation procedure for bivariate extreme value copulas. *Biometrika*, **84**, 567–577.

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*. London: Springer–Verlag.

Coles, S. G., Heffernan, J. and Tawn, J. A. (1999) Dependence measures for extreme value analyses. *Extremes*, **2**, 339–365.

Coles, S. G. and Tawn, J. A. (1990) Statistics of coastal flood prevention. *Phil. Trans. R. Soc. Lond., A*, **332**, 457–476.

Coles, S. G. and Tawn, J. A. (1991) Modelling extreme multivariate events. *J. Roy. Statist. Soc., B*, **53**, 377–392.

Coles, S. G. and Tawn, J. A. (1994) Statistical methods for multivariate extremes: an application to structural design (with discussion). *Appl. Statist.*, **43**, 1–48.

de Haan, L. (1984) A spectral representation for max-stable processes. *Ann. Probab.*, **12**, 1194–1204.

Deheuvels, P. (1991) On the limiting behaviour of the pickands estimator for bivariate extreme-value distributions. *Statist. Probab. Letters*, **12**, 429–439.

Galambos, J. (1975) Order statistics of samples from multivariate distributions. *J. Amer. Statist. Assoc.*, **70**, 674–680.

Gumbel, E. J. (1960) Distributions des valeurs extrêmes en plusieurs dimensions. *Publ. Inst. Statist. Univ. Paris*, **9**, 171–173.

Hall, P. and Tajvidi, N. (2000) Distribution and dependence-function estimation for bivariate extreme-value distributions. *Bernoulli*, **6**, 835–844.

Hüsler, J. and Reiss, R.-D. (1989) Maxima of normal random vectors: between independence and complete dependence. *Statist. Probab. Letters*, **7**, 283–286.

Ihaka, R. and Gentleman, R. (1996) R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Joe, H. (1990) Families of min-stable multivariate exponential and multivariate extreme value distributions. *Statist. Probab. Letters*, **9**, 75–81.

Kotz, S. and Nadarajah, S. (2000) *Extreme Value Distributions*. London: Imperial College Press.

Leadbetter, M. R., Lindgren, G. and Rootzén, H. (1983) *Extremes and Related Properties of Random Sequences and Series*. New York: Springer–Verlag.

Pickands, J. (1981) Multivariate extreme value distributions. *Proc. 43rd Sess. Int. Statist. Inst.*, **49**, 859–878.

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

Smith, R. L. (1990) Extreme value theory. In *Handbook of Applicable Mathematics* (ed. W. Ledermann), vol. 7. Chichester: John Wiley, pp. 437–471.

Stephenson, A. G. (2003) *Extreme Value Distributions and their Application*. Ph.D. thesis, Lancaster University, Lancaster, UK.

Tabony, R. C. (1983) Extreme value analysis in meteorology. *The Meteorological Magazine*, **112**, 77–98.

Tawn, J. A. (1988) Bivariate extreme value theory: models and estimation. *Biometrika*, **75**, 397–415.

Tawn, J. A. (1990) Modelling multivariate extreme value distributions. *Biometrika*, **77**, 245–253.

Tiago de Oliveira, J. (1997) *Statistical Analysis of Extremes*. Pendor.