A User's Guide to the evd Package (Version 1.2)

Alec Stephenson

Department of Mathematics and Statistics,
Lancaster University,
Lancaster, LA1 4YF.

E-mail: a.stephenson@lancaster.ac.uk
30th May 2002

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 extends simulation, distribution, quantile (inverse distribution) and density functions to univariate, bivariate and multivariate parametric extreme value distributions. It also provides fitting functions which calculate maximum likelihood estimates for univariate and bivariate models.

All comments, criticisms and queries on the package or associated documentation are gratefully received.

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. It can also be downloaded directly from `http://www.maths.lancs.ac.uk/~stephena/` (in postscript or pdf).

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 functions for univariate extreme value distributions. Sections 3 and 4 do the same for bivariate and multivariate models. Maximum likelihood fitting of univariate and bivariate models is discussed in Sections 5 and 6 respectively. Two extended examples, one univariate and one bivariate, using the data sets `oxford` and `sealevel` (both included in the package) are given in Sections 7 and 8.

This guide should not be viewed as an alternative to the help files included within the package. These remain the definitive source of help. A reference manual containing all the help files can be downloaded from `http://www.maths.lancs.ac.uk/~stephena/` or from CRAN.

All of the examples presented in this guide are called with `options(digits = 4)`.

1.4 Citing the package/guide

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

```
@MANUAL{key,
TITLE = {A User's Guide to the evd Package (Version 1.2)},
AUTHOR = {Stephenson, A. G.},
YEAR = {2002},
NOTE = {Available from \verb+http://www.maths.lancs.ac.uk/~stephena/+}
}
```

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 `a.stephenson@lancaster.ac.uk`. If you do find a bug and are the first person to report it, I guarantee to buy you the drink of your choice. If you ever manage to find me.

1.6 Legalese

1.7 Acknowledgments

Thanks to Paulo Ribeiro Jr. for much needed advice.

2 Standard Univariate Functions

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

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

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

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

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 $h_+ = \max(h, 0)$. 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 $\mu = a - b$.

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 four distributions defined above has an associated set of functions, as shown here for the GEV.

```
rgev(n, loc = 0, scale = 1, shape = 0)
pgev(q, loc = 0, scale = 1, shape = 0, lower.tail = TRUE)
qgev(p, loc = 0, scale = 1, shape = 0, lower.tail = TRUE)
dgev(x, loc = 0, scale = 1, shape = 0, log = FALSE)
```

Parameters arguments can be vectors* (the standard recycling rules are applied). 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

> 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
```

*With one exception; the **shape** parameter of the GEV functions cannot be a vector.

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

$$\Pr(U_m \leq x) = [F(x)]^m \tag{5}$$

$$\Pr(L_m \leq x) = 1 - [1 - F(x)]^m. \tag{6}$$

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

```
rext(n, quantfun, ..., distn, mlen = 1, largest = TRUE)
pext(q, distnfun, ..., distn, mlen = 1, largest = TRUE, lower.tail = TRUE)
qext(p, quantfun, ..., distn, mlen = 1, largest = TRUE, lower.tail = TRUE)
dext(x, densfun, distnfun, ..., distn, mlen = 1, largest = TRUE, log = FALSE)
```

The integer m should be given to the argument `mlen`. The distribution F can be specified by passing the corresponding quantile, distribution and density functions to `quantfun`, `distnfun` and `densfun` respectively (both the distribution and density functions are required for `dext`). Alternatively, a string can be passed to `distn` such that the name of the quantile/distribution/density function is derived when the string is prefixed by the letter q/p/d. If the distribution of U_m is required use `largest = TRUE` (the default). If the distribution of L_m is required use `largest = FALSE`. Some examples should make this clear.

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

> rext(1, distn = "norm", sd = 2, mlen = 20, largest = FALSE)
[1] -4.403
# Simulates from the same distribution as
> min(rnorm(20, mean = 0, sd = 2))
[1] -2.612

> pext(c(.4, .5), distn = "norm", mean = 0.5, sd = c(1, 2), mlen = 4)
[1] 0.04484 0.06250

> dext(c(1, 4), distn = "gamma", shape = 1, scale = 0.3, mlen = 100)
[1] 0.3261328 0.0005398
```

Parameters can be given as vectors assuming that this is implemented in the functions passed as arguments (either directly, using `[quant/dist/dens]fun`, or indirectly, using `distn`). If any of the parameters of F are omitted the defaults defined in the corresponding distribution/density/quantile function are used. If default values do not exist an error occurs. Although the examples above use functions provided in R, user defined functions can be specified. Density functions must have `log` arguments.[†]

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

$$\Pr(X_{(j)} \leq x) = \sum_{k=0}^{j-1} \binom{m}{k} [F(x)]^{m-k} [1 - F(x)]^k. \tag{7}$$

[†]A simple wrapper can always be constructed to achieve this.

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(n, quantfun, ..., distn, mlen = 1, j = 1, largest = TRUE)
porder(q, distn, ..., mlen = 1, j = 1, largest = TRUE, lower.tail = TRUE)
dorder(x, densfun, distnfun, ..., distn, mlen = 1, j = 1, largest = TRUE,
       log = FALSE)
```

The integer m should again be given to the argument `mlen`. If `largest = TRUE` (the default) the distribution of the `j`th largest order statistic $X_{(j)}$ is used. If `largest = FALSE` the distribution of the `j`th smallest order statistic $X_{(m+j-1)}$ is used.

For computational reasons it is better to specify `j` to be an integer in the interval $[1, $`ceiling(mlen/2)`$]$. This can always be achieved using the argument `largest`. Some examples are given below.

```
> rorder(1, distn = "norm", mlen = 20, j = 2)
> rorder(1, distn = "norm", mlen = 20, j = 19, largest = FALSE)
# Both simulate the second largest order statistic from 20 standard normals
# The first expression is preferred since j is in the interval [1,10]
[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 Standard Bivariate Functions

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, given by

$$G(z_1, z_2) = \exp\left\{-(y_1^{1/\alpha} + y_2^{1/\alpha})^{\alpha}\right\}, \quad 0 < \alpha \leq 1, \tag{8}$$

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

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

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

$$y_j = y_j(z_j) = \{1 + \xi_j(z_j - \mu_j)/\sigma_j\}_+^{-1/\xi_j} \tag{10}$$

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 (8) and (9) have asymmetric extensions, given by

$$G(z_1, z_2) = \exp\left\{-(1-\theta_1)y_1 - (1-\theta_2)y_2 - [(\theta_1 y_1)^{1/\alpha} + (\theta_2 y_2)^{1/\alpha}]^{\alpha}\right\}, \quad 0 < \alpha \leq 1, \qquad (11)$$

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

known as the asymmetric logistic (Tawn, 1988) and asymmetric negative logistic (Joe, 1990) models respectively, where the asymmetry parameters $0 \leq \theta_1, \theta_2 \leq 1$. 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 $r \to \infty$.

Any bivariate extreme value distribution function can be expressed as (de Haan, 1984)

$$G(z_1, z_2) = \exp\left\{-\int_0^1 \max\{y_1 f_1(x), y_2 f_2(x)\}\, \mathrm{d}x\right\}$$

where (y_1, y_2) are again defined by the transformations (10), and where f_1 and f_2 are density functions with support [0,1].

In particular, if we take the beta densities $f_1(x) = (1-\alpha)x^{-\alpha}$ and $f_2(x) = (1-\beta)(1-x)^{-\beta}$ we obtain

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

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

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

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

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

[‡]Coles and Tawn (1991) call this the Dirichelet model. I avoid this term because it could be confused with the Dirichelet distribution.

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 ∞.

Each of the eight models has a set of functions of the type given here for the asymmetric logistic.

```
rbvalog(n, dep, asy = c(1, 1), mar1 = c(0, 1, 0), mar2 = mar1)
pbvalog(q, dep, asy = c(1, 1), mar1 = c(0, 1, 0), mar2 = mar1)
dbvalog(x, dep, asy = c(1, 1), mar1 = c(0, 1, 0), mar2 = mar1, log = FALSE)
abvalog(x = 0.5, dep, asy = c(1, 1), plot = FALSE, add = FALSE, lty = 1,
    blty = 3, xlim = c(0, 1), ylim = c(0.5, 1), xlab = "",  ylab = "", ...)
```

The first three functions are for simulation and for the calculation of the distribution and density functions. The arguments `q` and `x` in `pbvalog` and `dbvalog` respectively should be vectors of length two or matrices with two columns, so that each row specifies a value for (z_1, z_2).

The argument `dep` is the dependence parameter. In this case `dep` represents the parameter α in distribution (11). The argument `asy` is a vector containing the asymmetry parameters (θ_1, θ_2). The marginal parameters (μ_1, σ_1, ξ_1) and (μ_2, σ_2, ξ_2) should be passed to `mar1` and `mar2` respectively. The arguments `mar1` and `mar2` can also be given as matrices with three columns, in which case each column represents a vector of values to be passed to the corresponding marginal parameter (the standard recycling rules are applied). Vector/matrix arguments for `dep` and `asy` are not implemented.

The `abvalog` function calculates (by default) or plots the dependence function $A(\cdot)$, which is defined as follows. Any bivariate extreme value distribution function can be represented in the form

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

so that $A(\omega) = -\log\{G(y_1^{-1}(\omega), y_2^{-1}(1-\omega))\}$, defined on $0 \le \omega \le 1$.[§] $A(\cdot)$ does not depend on the marginal parameters. It can be shown that $A(0) = A(1) = 1$, and that $A(\cdot)$ is a convex function with $\max(\omega, 1-\omega) \le A(\omega) \le 1$ for all $0 \le \omega \le 1$. $A(\cdot)$ is differentiable when the distribution is jointly continuous. The value $A(1/2) \in [0.5, 1]$ is returned by default. At independence $A(1/2) = 1$. At complete dependence $A(1/2) = 0.5$.

Non-parametric estimators of the dependence function can also be calculated or plotted, using the function `abvnonpar`. Suppose (z_{i1}, z_{i2}) for $i = 1, \ldots, n$ are n bivariate observations that are passed to `abvnonpar` using its `data` argument. The marginal parameters are estimated (under the assumption of independence) and the data is transformed using

$$y_{i1} = \{1 + \hat{\xi}_1(z_{i1} - \hat{\mu}_1)/\hat{\sigma}_1\}_+^{-1/\hat{\xi}_1}$$
$$y_{i2} = \{1 + \hat{\xi}_2(z_{i2} - \hat{\mu}_2)/\hat{\sigma}_2\}_+^{-1/\hat{\xi}_2}$$

for $i = 1, \ldots, n$, 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 5 and 6) the marginal location parameters may depend on i.

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

Three different estimators can be implemented. They are defined (on $0 \leq \omega \leq 1$) as follows.

Pickands (1981)

$$A_p(\omega) = n \left\{ \sum_{i=1}^{n} \min \left(\frac{y_{i1}}{\omega}, \frac{y_{i2}}{1-\omega} \right) \right\}^{-1}$$

Deheuvels (1991)

$$A_d(\omega) = n \left\{ \sum_{i=1}^{n} \min \left(\frac{y_{i1}}{\omega}, \frac{y_{i2}}{1-\omega} \right) - \omega \sum_{i=1}^{n} y_{i1} - (1-\omega) \sum_{i=1}^{n} y_{i2} + n \right\}^{-1}$$

Capéraà *et al.* (1997); The Default Estimator

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

In the estimator of Capéraà *et al.* (1997), $H_n(x)$ is the empirical distribution function of $x_1, \ldots, x_n$, where $x_i = y_{i2}/(y_{i1} + y_{i2})$ for $i = 1, \ldots, n$, 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 $A(\cdot)$. The estimator proposed by Deheuvels (1991) satisfies $A_n(0) = A_n(1) = 1$. $A_c(\cdot)$ satisfies this constraint when $p(0) = 0$ and $p(1) = 1$. None of the estimators satisfy $\max(\omega, 1-\omega) \leq A_n(\omega) \leq 1$ for all $0 \leq \omega \leq 1$. An obvious modification is

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

Another estimator $A_n^{''}(\omega)$ can be derived by taking the convex minorant of $A_n^{'}(\omega)$. These modifications can be implemented using the `modify` argument. Set `modify` $= 1$ to plot or calculate $A_n^{'}(\omega)$. Set `modify` $= 2$ to plot or calculate $A_n^{''}(\omega)$. Please refer to Section 8 for examples.

Some of the functions outlined in this section are illustrated below.

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

> rbvnegbilog(3, alpha = .5, beta = 1.2, mar1 = c(1, 1, 1))
       [,1]  [,2]
[1,] 0.7417 1.085
[2,] 0.8391 1.825
[3,] 2.0142 2.280

> pbvaneglog(c(1, 1.2), dep = .4, asy = c(.4, .6), mar1 = c(1, 1, 1))
[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)
> pbvaneglog(tmp.quant, dep = .4, asy = c(.4, .6), mar1 = tmp.mar)
[1] 0.173 0.175

> dbvct(c(1, 1.2), alpha = .2, beta = .6, mar1 = c(1, 1, 1))
[1] 0.1213
```

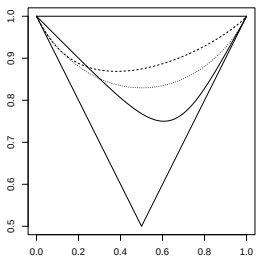

Figure 1: Dependence functions for various bivariate extreme value distributions. All dependence functions must be convex and must lie within the triangular region.

```
> dbvct(tmp.quant, alpha = 0.2, beta = 0.6, mar1 = tmp.mar)
[1] 0.1213 0.0586

> abvlog(dep = .3)
[1] 0.6156
> abvlog(seq(0, 1, 0.25), dep = .3)
[1] 1.0000 0.7557 0.6156 0.7557 1.0000

> abvalog(dep = .3, asy = c(.5, .9), plot = TRUE, blty = 1)
> abvbilog(alpha = .5, beta = .9, add = TRUE, lty = 2)
> abvhr(dep = 1.05, add = TRUE, lty = 3)
```

The last three lines of code produce Figure 1.

The simulation functions `rbvlog` and `rbvalog` use bivariate versions of Algorithms 1.1 and 1.2 in Stephenson (2002). All other simulation functions use a root finding algorithm to generate random vectors from the conditional distribution function. The simulation functions `rbvbilog` and `rbvnegbilog` for the bilogistic and negative bilogistic models are relatively slow (about 2.8 seconds per 1000 random vectors on a 450MHz PIII, 512Mb RAM) because each evaluation of either distribution function requires a root finding algorithm to evaluate γ.

4 Standard Multivariate Functions

Let $z = (z_1, \ldots, z_d)$. The d-dimensional logistic model (Gumbel, 1960) is given by

$$G(z) = \exp\left\{-\left(\sum_{j=1}^{d} y_j^{-1/\alpha}\right)^{\alpha}\right\} \tag{12}$$

where $\alpha \in (0, 1]$ and $(y_1, \ldots, y_d)$ is defined by the transformations (10).

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

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

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 (12) can be obtained by setting $\theta_{i,12\ldots d} = 1$ for all $i = 1, \ldots, d$ (which implies, using the sum constraints, that $\theta_{i,b} = 0$ whenever $|b| < d$) and $\alpha_{12\ldots d} = \alpha$.

The evd package provides the following functions for simulating from and calculating the distribution function of these models.

```
rmvlog(n, dep, d = 2, mar = c(0, 1, 0))
pmvlog(q, dep, d = 2, mar = c(0, 1, 0))
rmvalog(n, dep, asy, d = 2, mar = c(0, 1, 0))
pmvalog(q, dep, asy, d = 2, mar = c(0, 1, 0))
```

The argument `mar` represents the GEV marginal parameters for every univariate margin, and may again be a matrix. The simulation functions `rmvlog` and `rmvalog` use Algorithms 2.1 and 2.2 in Stephenson (2002).

For the symmetric model `dep` $= \alpha$. Some examples are given below.

```
> rmvlog(3, dep = .6, 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)
> rmvlog(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)
> pmvlog(tmp.quant, dep = .6, d = 5, mar = tmp.mar)
[1] 0.07233 0.16387 0.21949
```

For the asymmetric model `dep` should be a vector of length $2^{\mathtt{d}} - \mathtt{d} - 1$ containing the dependence parameters. Specifically, 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 $\{\theta_{i,b} : i \in b\}$. Specifically, 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 $i = 1, \ldots, d$, must be satisfied or an error will occur.

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 $(\theta_{1,123}, \theta_{2,123}, \theta_{3,123}) = (.2, .4, .2)$. Notice that the constraints are satisfied.

```
> asy <- list(.4, 0, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.4,.2))
> rmvalog(3, dep = c(.6,.5,.8,.3), asy = asy, 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

> pmvalog(c(2, 2, 2), dep = c(.6,.5,.8,.3), asy = asy, d = 3)
[1] 0.7223

> tmp.quant <- matrix(rep(c(1,1.5,2), 3), ncol = 3)
> pmvalog(tmp.quant, dep = c(.6,.5,.8,.3), asy = asy, 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 $i \in b$ such that $|b| \leq 2$, $(\theta_{1,123}, \theta_{2,123}, \theta_{3,123}) = (.2, .1, .2)$, $(\theta_{1,124}, \theta_{2,124}, \theta_{4,124}) = (.1, .1, .2)$, $(\theta_{1,134}, \theta_{3,134}, \theta_{4,134}) = (.3, .4, .1)$, $(\theta_{2,234}, \theta_{3,234}, \theta_{4,234}) = (.2, .2, .2)$ 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))
> rmvalog(3, dep = c(rep(1,6),.7,.3,.8,.7,.5), asy = asy, 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
```

I will end this section with some examples that may be helpful in deciphering errors.

```
> asy <- list(.4, 0, .5, c(.3,.2), c(.1,.15), c(.4,.075), c(.2,.4,.25))
> rmvalog(3, dep = c(.6,.5,.8,.3), asy = asy, d = 3)
Error in rmvalog(3, dep = c(0.6, 0.5, 0.8, 0.3), asy = asy, d = 3) :
        'asy' does not satisfy the appropriate constraints

# 0.5 + 0.15 + 0.075 + 0.25 does not equal one; the sum constraint on the third
margin is not satisfied.

> asy <- list(.4, 0, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.4,.2))
> rmvalog(3, dep = c(.6,1,.8,.3), asy = asy, d = 3)
Error in rmvalog(3, dep = c(0.6, 1, 0.8, 0.3), asy = asy, d = 3) :
```

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

```
        'asy' does not satisfy the appropriate constraints

# A dependence parameter is equal to one but the corresponding asymmetry
parameters are not zero (the first 'further constraint').
# One possible alternative which preserves dep (and still satisfies the sum
constraints) is

> asy <- list(.4, 0, .6, c(.3,.2), c(0,0), c(.4,.1), c(.3,.4,.3))
> rmvalog(3, dep = c(.6,1,.8,.3), asy = asy, d = 3)
       [,1]    [,2]    [,3]
[1,]  4.627  2.9125  0.9414
[2,]  1.200  0.1556  0.2048
[3,] -1.159 -0.8749 -1.0340

> asy <- list(.5, 0, .6, c(.3,.2), c(0,.1), c(.4,.1), c(.2,.4,.2))
> rmvalog(3, dep = c(.6,.5,.8,.3), asy = asy, d = 3)
Error in rmvalog(3, dep = c(0.6, 0.5, 0.8, 0.3), asy = asy, d = 3) :
        'asy' does not satisfy the appropriate constraints

# The fifth element in asy contains exactly one non-zero asymmetry parameter
(the second 'further constraint').

> asy <- list(.4, 0, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.4,.2))
> rmvalog(3, dep = c(.6,.5,.8,.3), asy = asy)
Error in rmvalog(3, dep = c(0.6, 0.5, 0.8, 0.3), asy = asy) :
        'asy' is not of the correct form

# asy is not of the correct form only because the dimension has been
mis-specified (the default dimension is 2).
```

5 Fitting Univariate Distributions by Maximum Likelihood

This section presents functions that produce maximum likelihood estimates for distributions introduced in Section 2. Maximum likelihood estimates for bivariate distributions are discussed in Section 6. For illustrative purposes Sections 5 and 6 use only simulated data. Two extended examples (one univariate and one bivariate) using the data sets `oxford` and `sealevel` (both included in the evd package) are given in Sections 7 and 8.

The following function produces maximum likelihood estimates for the GEV distribution (4).

```
fgev(x, start, ..., nsloc = NULL, std.err = TRUE, corr = FALSE,
    method = "BFGS", warn.inf = TRUE)
```

The argument `x` should be a numeric vector containing data to be fitted. Missing values are allowed. If `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 some of the parameters are to be set to fixed values, they can be given as separate arguments. The Gumbel distribution (1) can be fitted using `shape = 0`. Any arguments that can be passed to

the optimization function `optim` can also be specified. This includes the optimization method, which is explicitly passed using the argument `method`. If `std.err` = `TRUE` (the default), the (asymptotic) standard errors* of the maximum likelihood estimates are returned. If `corr` = `TRUE`, the (asymptotic) correlation matrix is returned. When `warn.inf` = `TRUE` (the default), a warning is given if the negative log-likelihood is infinite at the starting values. The `nsloc` argument is explained subsequently.

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

*This is not strictly true. Read on for the details.

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
```

Maximum likelihood estimators of GEV parameters do not necessarily have the usual asymptotic properties, since the (lower or upper) end point of the GEV distribution (given by $\mu - \sigma/\xi$) depends on the 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 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 asymptotic 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 "standard errors" and the "correlation matrix" must be *interpreted with caution* (Smith, 1985).

The `deviance(m1)` is minus twice the log-likelihood function, evaluated at `fitted(m1)`. Likelihood ratio tests can be performed by comparing with chi-squared distributions the difference between the deviances of nested models.[†] We can compare the two examples given above, using the function `anova`, 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 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The deviance difference 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. Model diagnostics and profile deviances can be calculated/plotted using `plot`, `profile` and `profile2d` (see Section 7).

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

[†]There are non-regular cases for which the asymptotic distribution of the test statistic is not chi-squared (see Section 8).

$Z_i \sim \text{GEV}(\mu, \sigma, \xi)$ 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) *centered 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 centered 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 7 and 8 for examples of non-stationary modelling.

The function `fgev.quant` 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. The function can be used to calculate and plot profile deviances of extreme quantiles (see Section 7). 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 following functions produce maximum likelihood estimates for the distributions (5), (6) and (7).

```
fext(x, start, densfun, distnfun, ..., distn, mlen = 1, largest = TRUE,
     std.err = TRUE, corr = FALSE, method = "Nelder-Mead")
forder(x, start, densfun, distnfun, ..., distn, mlen = 1, j = 1, largest = TRUE,
     std.err = TRUE, corr = FALSE, method = "Nelder-Mead")
```

The `fext` function yields maximum likelihood estimates for the distributions (5) and (6) given an integer m and an arbitrary distribution function F.

The arguments `densfun`, `distnfun`, `distn`, `mlen` and `largest` are the same as those used in the density function `dext` (see Section 2). The argument `x` should be a numeric vector containing the data to be fitted, which should be assumed to represent maxima (if `largest` is `TRUE`, the default) or minima (if `largest` is `FALSE`). `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 some of the parameters are to be set to fixed values, they can be given as separate arguments. If parameters are missing, they are fixed at their default values specified in the density/distribution function. Any of the arguments that can be given to `optim` can also be specified.

The optimizer will be affected by the way in which the density and distribution functions passed to `fext` behave when given values outside of the valid parameter space.[‡] Functions in R base generally produce `NaN` values which may result in warnings being printed. These warnings can usually be ignored.

If the density and distribution functions are user defined, the order of the arguments must mimic those in R base (i.e. data first, parameters second). The density function must have a `log` argument.[§]

Two examples are given below. The second example (incorrectly) takes F to be a gamma distribution with the shape parameter fixed at 0.5.

```
> data2 <- rext(100, qnorm, mean = 0.56, mlen = 365)
# Simulate yearly maxima using normal distribution

> nm <- fext(data2, list(mean = 0, sd = 1), distn = "norm", mlen = 365)
> fitted(nm)
 mean    sd
0.685 0.959

> ga <- fext(data2, list(scale = 1), shape = 0.5, distn = "gamma", mlen = 365,
        method = "L-BFGS-B", lower = 0.01)
> fitted(ga)
 scale
0.7386
```

The `forder` function yields maximum likelihood estimates for the distribution (7) given integers m and $j \in \{1, \ldots, m\}$, and an arbitrary distribution function F. The arguments `densfun`, `distnfun`, `distn`, `mlen`, `j` and `largest` are the same as those used in the density function `dorder` (see Section 2). The argument `x` should be a numeric vector containing the data to be fitted, and `start` (which cannot be missing) should again be a named list containing starting values.

6 Fitting Bivariate Distributions by Maximum Likelihood

For each of the eight bivariate models introduced in Section 3 there is a function that produces maximum likelihood estimates. Each function has the same formal arguments. The function corresponding to the logistic model is

```
fbvlog(x, start, ..., nsloc1 = NULL, nsloc2 = NULL, std.err = TRUE,
    corr = FALSE, method = "BFGS", warn.inf = TRUE)
```

The argument `x` should be given a numeric matrix (or a data frame) containing two columns of data to be fitted. Missing values are allowed on either or both margins/columns within any

[‡] The `L-BFGS-B` method can be used to specify box-constraints. See the help page of `optim` for details.

[§] A simple wrapper can always be constructed to achieve this.

Bivariate Model	Constraints
Logistic	$0.1 \leq \alpha \leq 1$
Asymmetric Logistic	$0.1 \leq \alpha \leq 1$, $0 \leq \theta_1, \theta_2 \leq 1$
Hüsler-Reiss	$0.2 \leq \lambda \leq 10$
Negative Logistic	$0.05 \leq r \leq 5$
Asymmetric Negative Logistic	$0.05 \leq r \leq 5$, $0.001 \leq \theta_1, \theta_2 \leq 1$
Bilogistic	$0.1 \leq \alpha, \beta \leq 0.999$
Negative Bilogistic	$0.1 \leq \alpha, \beta \leq 20$
Coles-Tawn	$0.001 \leq \alpha, \beta \leq 30$

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

observation/row. If `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. See the help file for details. If `start` is omitted the routine attempts to find good starting values for the optimization using maximum likelihood estimators under the assumption of independence.

A separate name is associated with each individual parameter so that any parameter subset can be fixed at specified values. The parameters on the first margin can be fixed using the arguments `loc1`, `scale1` and `shape1`. The parameters on the second margin can be fixed using `loc2`, `scale2` and `shape2`. The asymmetry parameters in the asymmetric logistic and asymmetric negative logistic models can be fixed using `asy1` and `asy2`. As usual, any arguments that can passed to `optim` can be specified.

The `nsloc1` and `nsloc2` arguments allow non-stationary modelling of the location parameter on the first and second margin respectively. They should be used in the same manner as the `nsloc` argument in univariate fitting functions. Examples of bivariate models with non-stationary margins are given in Section 8. The `std.err`, `corr`, `method` and `warn.inf` arguments are equivalent to those used in univariate fitting functions.

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 uses the `fbvlog` function to obtain 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 <- fbvlog(bvdata)
> m1

Call: fbvlog(x = bvdata)
Deviance: 707.8

Estimates
    loc1    scale1    shape1      loc2    scale2    shape2       dep
 1.19491   1.39859  -0.00922   1.02453   1.41791   0.03395   0.54230
```

```
Standard Errors
  loc1  scale1  shape1    loc2  scale2  shape2     dep
0.1519  0.1089  0.0517  0.1570  0.1169  0.0645  0.0540

Optimization Information
  Convergence: successful
  Function Evaluations: 51
  Gradient Evaluations: 11

> m2 <- fbvlog(bvdata, dep = 1)
> fitted(m2)
    loc1   scale1   shape1     loc2   scale2   shape2
 1.19070  1.39358 -0.01697  1.02258  1.41097  0.04831
```

The discussion in Section 5 regarding the properties of maximum likelihood estimators for the GEV distribution 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 "standard errors" and the "correlation matrix" must be *interpreted with caution* (Smith, 1985).

Model diagnostics and profile deviances can be calculated/plotted using `plot`, `profile` and `profile2d` (see Section 8). Components of the fitted object can be extracted using `fitted`, `std.errors` and `deviance`.

The following snippet uses `anova` to perform a likelihood ratio test. The null hypothesis 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. The p-value is calculated to be 0.78.

```
> m3 <- fbvlog(bvdata, 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
```

Boundary Problems

In the following example I attempt to fit the asymmetric logistic model to the simulated data set used above. The data set is distributed as symmetric logistic; both asymmetry parameters equal one. This illustrates the difficulties that arise when parameter estimates are on the edge of the parameter space.

```
> m4 <- fbvalog(bvdata)
> fitted(m4)
   loc1  scale1  shape1    loc2  scale2  shape2    asy1    asy2     dep
1.19537 1.33235 0.01946 1.03591 1.41390 0.13776 0.99949 0.89746 0.47705
```

A boundary of the parameter space has been reached; the maximum likelihood estimate for the first asymmetry parameter is (effectively) one. This will cause difficulties for the optimizer. There are two approaches to this problem. The 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 these approaches.

```
> fitted(fbvalog(bvdata, asy1 = 1))
     loc1    scale1    shape1      loc2    scale2    shape2      asy2       dep
 1.178964  1.391445 -0.002429  1.036365  1.430401  0.036784  0.869307  0.495566

> upper <- c(rep(Inf, 6), 1, 1, 1)
> fitted(fbvalog(bvdata, method = "L-BFGS-B", upper = upper))
     loc1    scale1    shape1      loc2    scale2    shape2      asy1      asy2
 1.178968  1.391463 -0.002437  1.036320  1.430364  0.036788  1.000000  0.869414
      dep
 0.495590
```

Fitting Every Model

The `fbvall` function produces maximum likelihood estimates for all eight models.

```
fbvall(x, ..., nsloc1 = NULL, nsloc2 = NULL, which = NULL, boxcon = TRUE,
  std.err = TRUE, orderby = c("AIC", "BIC", "SC"), control = list(maxit = 250))
```

The argument `x` should again be given a numeric matrix (or a data frame) containing two columns of data to be fitted. Marginal parameters can be set to fixed values within each optimization (if Gumbel margins are required, use `shape1 = 0` and `shape2 = 0`). The `nsloc1` and `nsloc2` arguments allow non-stationary modelling of the location parameters, for every fitted model. A subset of models can be specified using `which` (all models are fitted by default). The (column) order of the models in the components of the returned object is controlled by `orderby`. The `control` argument is passed to the optimization function `optim`. The `boxcon` argument defines the method used for the optimizations. If `boxcon` is `TRUE` (the default), the `L-BFGS-B` optimization method is used, which incorporates box-constraints. If `boxcon` is `FALSE` the `BFGS` method is used. The `BFGS` method is faster, and should be preferred if the estimates are known to lie in the interior of the parameter space. The help page provides further details. An example is provided in Section 8.

7 Extended Example: Oxford 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, as depicted in Figure 2.

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

The following snippet 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 centered 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.

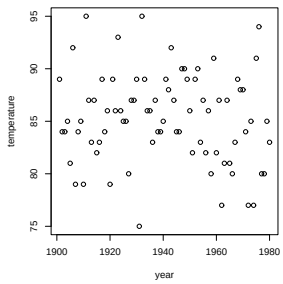

Figure 2: The `oxford` data.

```
> 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
  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% (asymptotic) 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%

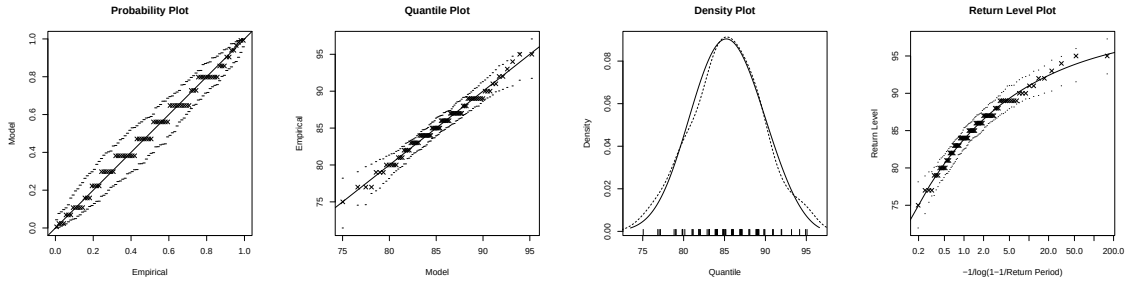

Figure 3: Diagnostic plots for the model `ox.fit`.

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 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Diagnostic plots can be produced using `plot`. The argument `jitter` is specified because the `oxford` data contains repeated values (see the help file for `plot.evd`).

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

The plots, depicted in Figure 3, compare parametric distributions, densities and quantiles to their empirical counterparts (see the help files for details on the construction of each plot). 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)
> plot(ox.prof)
```

This produces the first three plots within Figure 4. 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 calculated explicitly using `pcint(ox.prof)`. 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.

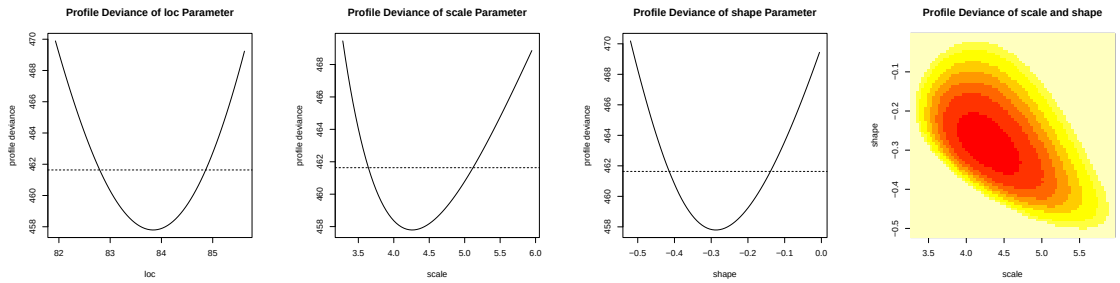

Figure 4: Profile deviance surfaces for the model `ox.fit`.

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 4.† 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

$$z_p = \begin{cases} \mu - \frac{\sigma}{\xi}[1 - \{-\log(1-p)\}^{-\xi}] & \xi \neq 0 \\ \mu - \sigma \log\{-\log(1-p)\} & \xi = 0, \end{cases} \tag{13}$$

is the quantile corresponding to the upper tail probability p. The profile deviance for $z_{0.1}$ can be plotted using the following. The function `fgev.quantile` calculates maximum likelihood estimates for the GEV distribution under the parameterization (z_p, σ, ξ), where p is specified using the argument `prob`.

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

Figure 5 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.

Imagine that Oxford is on another planet (it's easy if you try), and that maximum *daily* temperatures are both stationary and independent. The snippet below fits normal and gamma distributions to the (unrecorded) daily observations.

```
> ox.nm <- fext(ox, list(mean = 40, sd = 1), distn = "norm", mlen = 365)
> fitted(ox.nm)
 mean    sd
```

*For R Version 1.5.0 and later this will produce warnings, because the robust `"Nelder-Mead"` method is not recommended for optimizations over one variable, although it seems to work okay in this example.

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

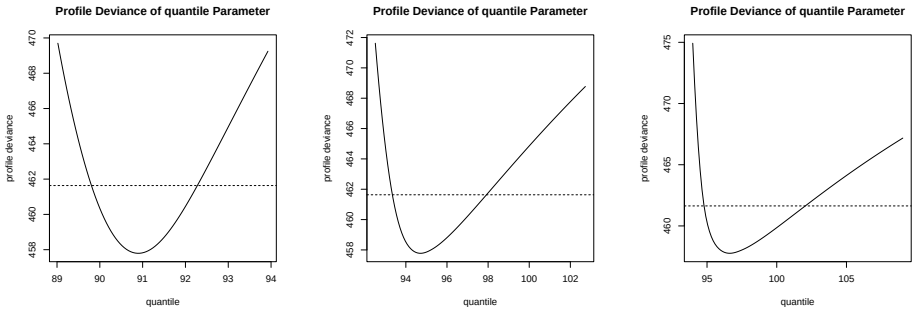

Figure 5: Profile deviance surfaces for $z_{0.1}$, $z_{0.01}$ and $z_{0.001}$.

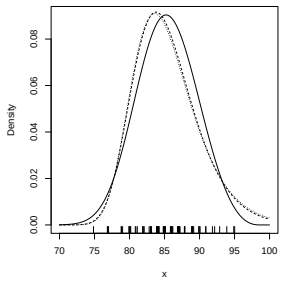

Figure 6: The fitted GEV density (solid line), and fitted densities for models assuming normal (dashed line) and gamma (dotted line) distributions for (unrecorded) daily maxima (valid under extremely tenuous assumptions). A rug plot of the (jittered) data is also included.

```
48.85 12.43

> ox.ga <- fext(ox, list(scale = 1, shape = 1), distn = "gamma", mlen = 365)
> fitted(ox.ga)
scale shape
 1.63 32.94

> x <- seq(70, 100, length = 100)
> mle <- fitted(ox.fit)
> plot(x, dgev(x, mle[1], mle[2], mle[3]), type = "l", ylab = "Density")
> nm.max <- dext(x, mean = 48.85, sd = 12.43, distn = "norm", mlen = 365)
> ga.max <- dext(x, scale = 1.63, shape = 32.94, distn = "gamma", mlen = 365)
> lines(x, nm.max, lty = 2)
> lines(x, ga.max, lty = 3)
> rug(jitter(ox))
```

The fitted densities for the *annual* maxima, derived by passing normal and gamma parameter estimates to `dext`, are compared to the GEV model in Figure 6. The normal and gamma models yield very similar distributions,[‡] and both are marginally more right skewed than the GEV fit. This example is not particularly relevant for the `oxford` data, but it may be relevant for other data sets.

[‡]Which will not be surprising to those who know about domains of attraction. The limiting distribution (as `mlen` tends to ∞) for both models is Gumbel.

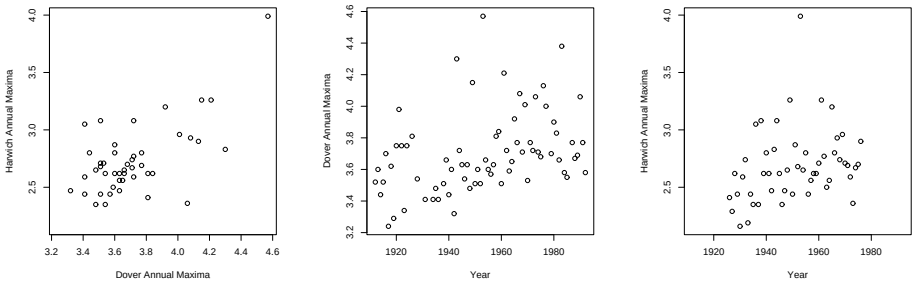

Figure 7: From left to right; Harwich Maxima vs Dover Maxima, Dover Maxima vs Year and Harwich Maxima vs Year.

8 Extended 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 depicted in Figure 7. The plot of the Harwich maxima against the Dover maxima shows a reasonable degree of 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 centered 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 <- fbvlog(sl, nsloc1 = tt, nsloc2 = tt)
> m2 <- fbvlog(sl, nsloc1 = tt)
> m3 <- fbvlog(sl)
```

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 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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 <- fbvlog(sl, nsloc1 = tdframe, nsloc2 = tt)
> m5 <- fbvlog(sl, nsloc1 = tt, nsloc2 = tdframe)
> m6 <- fbvlog(sl, 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 13.59. *The asymptotic distribution of the test statistic 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 within bivariate extreme value models. In this case the increase in deviance is clearly too large to consider independence.

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 <- fbvlog(sl, nsloc1 = tt, nsloc2 = tt, dep = 1)
> deviance(m7) - deviance(m1)
# WARNING: The asymptotic distribution of this statistic is non-regular
[1] 13.59

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

   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 the same as those given in Section 7. Diagnostic plots for the fitted dependence structure can be produced using `plot`, as shown in Figure 8 for the model `m1`.

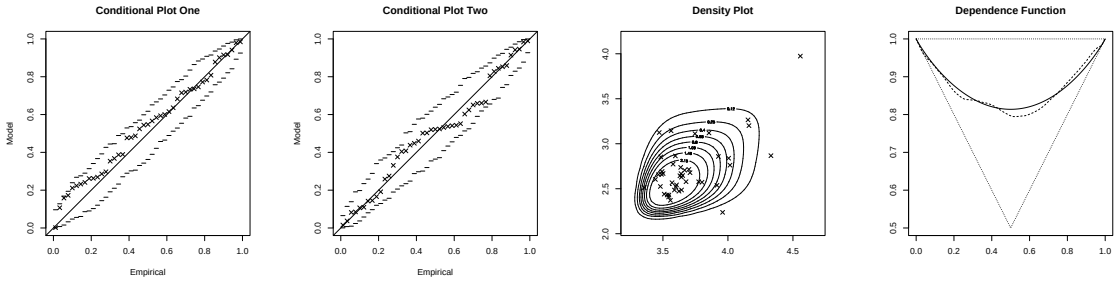

Figure 8: Diagnostic plots for the dependence structure of model `m1`.

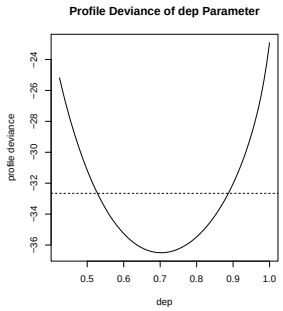

Figure 9: The profile deviance of the dependence parameter from model `m1`.

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

The plots compare parametric conditional distributions, densities and dependence functions to empirical counterparts (see the help files for details on 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.

```
> m1.prof <- profile(m1, which = "dep", xmax = 1)
> plot(m1.prof)
```

This produces the plot in Figure 9. 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 calculated explicitly using `pcint(m1.prof)`.

Models other than the logistic can be fitted in a similar fashion, using `fbvhr`, `fbvbilog`, `fbvct` and others. The function `fbvall` attempts to fit all bivariate models simultaneously (see Section 6). The code below provides an example. Linear trend terms for the marginal location parameters are implemented within each fit.

```
> m.all <- fbvall(sl, nsloc1 = tt, nsloc2 = tt)

> fitted(m.all)
```

```
                log  neglog     alog aneglog      hr   bilog      ct negbilog
loc1        3.57566 3.57510  3.58582 3.58285 3.57528 3.57548 3.57544  3.57516
loc1trend   0.43944 0.44014  0.46393 0.44098 0.44032 0.43871 0.43903  0.44104
scale1      0.15881 0.15843  0.17088 0.16577 0.15809 0.15857 0.15833  0.15863
shape1      0.06634 0.07259 -0.03219 0.00221 0.07325 0.06623 0.06841  0.07315
loc2        2.54899 2.55061  2.55091 2.55022 2.55167 2.54913 2.54967  2.55054
loc2trend   0.53283 0.53568  0.49654 0.51825 0.53605 0.53504 0.53811  0.53379
scale2      0.20701 0.20792  0.19952 0.20468 0.20852 0.20718 0.20683  0.20784
shape2      0.04781 0.03951  0.06158 0.07265 0.03038 0.04898 0.04233  0.03889
asy1/alpha       NA      NA  0.37812 0.45615      NA 0.69068 0.49647  1.37009
asy2/beta        NA      NA  0.25716 0.32518      NA 0.71339 0.48443  1.54953
dep         0.70282 0.68503  0.10000 5.00000 1.05590      NA      NA       NA
```

The dependence parameter estimates for the asymmetric logistic and asymmetric negative logistic models are at artificial boundaries. It appears that both models are fitting close to a (singular) distribution obtained in the limit. I will return to this issue later.

The `m.all` object contains standard errors and deviances for each model, which can be accessed using the functions `std.errors` and `deviance`. The fitting function `fbvall` and every accessor function has an argument `which`, that can be used to fit or access components of a specified subset of models. The object also contains the components `dep.summary` and `criteria`. The `dep.summary` component contains simple summary statistics of the dependence structure of each model, based on the dependence function $A(\cdot)$ defined in Section 3.

```
> m.all$dep.summary
          log neglog    alog aneglog     hr    bilog       ct negbilog
dep    0.3723 0.3635  0.2564  0.3144 0.3436  0.37284 0.359591  0.36404
intdep 0.5072 0.5039  0.3035  0.3694 0.4837  0.50794 0.495147  0.50458
intasy 0.0000 0.0000 -0.2708 -0.2695 0.0000 -0.02149 0.003764  0.01997
```

These summaries are given by

$$\texttt{dep} = \chi = 2\{1 - A(1/2)\}$$

$$\texttt{intdep} = \psi_d = 4\int_0^1 1 - A(x)\ \mathrm{d}x$$

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

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 summary statistic ψ_a is a measure of asymmetry, and is contained in the closed interval [-1,1].* If $A(\cdot)$ is symmetric $\psi_a = 0$. As a rough guide, any value $\psi_a \in (-0.2, 0.2)$ corresponds to a dependence structure that is close to symmetric.† The (unoptimized) asymmetric logistic and the (unoptimized) asymmetric negative logistic are the only models for which $|\psi_a| > 0.2$. The Coles-Tawn, bilogistic and negative bilogistic models can incorporate asymmetry, but for all three models $|\psi_a| < 0.05$.

Models that are not nested can be compared by adding penalty terms to the deviances. The penalty terms take into account the number of parameters fitted. (If both models have the same

*Conjecture.

†The integral measures ψ_d and ψ_a are not standard theory; I have created them for the purpose of this package. They have no underlying interpretation that I am aware of. I am open to suggestions of other simple summaries of the dependence structure.

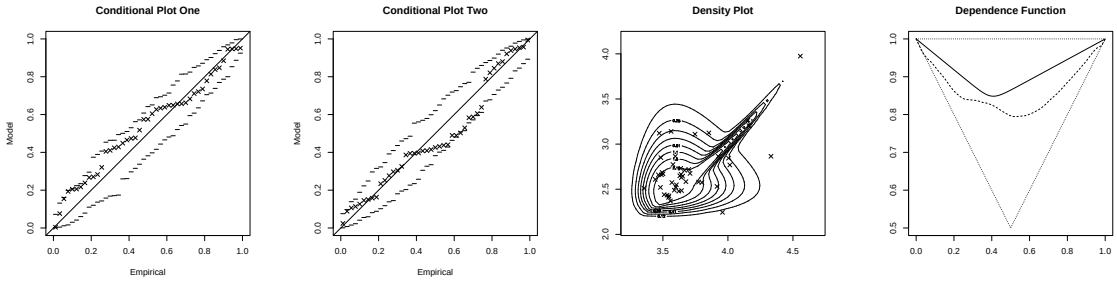

Figure 10: Diagnostic plots for the dependence structure of model `m9`.

number of parameters the deviances can be compared directly.) Three commonly used penalty terms are $2p$ (Akaike's information criterion, or AIC), $p\log(n)$ (Schwarz's criterion, or SC) and $p\{1+\log(n)\}$ (Bayesian information criterion, or BIC), where p is the number of parameters estimated and n is the number of observations.[‡] The AIC, SC and BIC values for each model are contained in the `criteria` component of the `m.all` object.

```
> m.all$criteria
        log  neglog   alog aneglog      hr   bilog      ct negbilog
AIC -18.498 -17.905 -17.85 -17.544 -17.376 -16.506 -16.218  -15.913
SC   -2.238  -1.645   2.02   2.330  -1.116   1.560   1.849    2.153
BIC   6.762   7.355  13.02  13.330   7.884  11.560  11.849   12.153
```

By default, the (column) order of the models in the components of `m.all` is determined by AIC (lowest/best first, highest/worst last). The default behaviour can be changed using the argument `orderby`. The (symmetric) logistic model is seen to give the best fit under all three criteria.

All models other than the asymmetric logistic and the asymmetric negative logistic produce similar dependence structures. The code below produces diagnostic plots for the (unoptimized) asymmetric logistic model, as depicted in Figure 10.

```
> m9 <- fbvalog(sl, nsloc1 = tt, nsloc2 = tt)
> plot(m9)
```

The near singular component can clearly be seen in the contour plot of the fitted density. Diagnostic plots for the (unoptimized) asymmetric negative logistic model depict the same behaviour. Both fits are unrealistic for sea level data.

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.

[‡]Since `fbvall` compares models for the dependence structure, n is taken as the number of observations which are complete (i.e. not missing on either margin).

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.

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.

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. (2002) Simulating multivariate extreme value distributions of logistic type. To be published - available on request.

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.