Package: ExtremalDep (via r-universe)

October 11, 2026

Version 1.0.0

Date 2025-08-15

Title Extremal Dependence Models

Author Boris Beranger [aut], Simone Padoan [cre, aut], Giulia Marcon [aut], Steven G. Johnson [ctb] (Author of included cubature fragments), Rudolf Schuerer [ctb] (Author of included cubature fragments), Brian Gough [ctb] (Author of included cubature fragments), Alec G. Stephenson [ctb], Anne Sabourin [ctb] (Author of included BMAmevt fragments), Philippe Naveau [ctb] (Author of PAMfmado function)

Maintainer Simone Padoan <simone.padoan@unibocconi.it>

Imports numDeriv, evd, sn, quadprog, copula, nloptr, gtools, mvtnorm, fda, parallel, doParallel, foreach, cluster, methods

Suggests fields, extraDistr

BugReports https://github.com/borisberanger/ExtremalDep/issues

Description A set of procedures for parametric and non-parametric modelling of the dependence structure of multivariate extreme-values is provided. The statistical inference is performed with non-parametric estimators, likelihood-based estimators and Bayesian techniques. It adapts the methodologies of Beranger and Padoan (2015) <doi:10.48550/arXiv.1508.05561>, Marcon et al. (2016) <doi:10.1214/16-EJS1162>, Marcon et al. (2017) <doi:10.1002/sta4.145>, Marcon et al. (2017) <doi:10.1016/j.jspi.2016.10.004> and Beranger et al. (2021) <doi:10.1007/s10687-019-00364-0>. This package also allows for the modelling of spatial extremes using flexible max-stable processes. It provides simulation algorithms and fitting procedures relying on the Stephenson-Tawn likelihood as per Beranger at al. (2021) <doi:10.1007/s10687-020-00376-1>.

License GPL (>= 2)

LazyData yes

NeedsCompilation yes

Repository/R-Forge/Project extremaldep

Repository/R-Forge/Revision 107

Repository/R-Forge/DateTimeStamp 2019-08-12 22:22:37

Depends R (>= 3.5.0)

URL https://faculty.unibocconi.it/simonepadoan/

Repository https://cran.r-universe.dev

Date/Publication 2025-08-21 11:40:02 UTC

RemoteUrl https://github.com/cran/ExtremalDep

RemoteRef HEAD

RemoteSha 55c675e6d3e375c7137278c02110ac440f53854f

Contents

angular . . . 3
angular.plot . . . 5
beed . . . 6
beed.boot . . . 9
beed.confband . . . 10
bic . . . 12
desn . . . 13
dest . . . 14
dExtDep . . . 15
dGEV . . . 18
diagnostics . . . 19
dim_ExtDep . . . 21
dmesn . . . 22
dmest . . . 24
ellipse . . . 25
est . . . 27
ExtQ . . . 28
fExtDep . . . 32
fExtDep.np . . . 36
fExtDepSpat . . . 44
fGEV . . . 47
heat . . . 50
index.ExtDep . . . 51
lambda.HR . . . 53
logReturns . . . 54
madogram . . . 55
method . . . 57
MilanPollution . . . 58
model . . . 58
PAMfmado . . . 59
pExtDep . . . 60

pFailure 62
pickands.plot 64
pollution 65
PrecipFrance 66
returns 67
returns.plot 69
rExtDep 71
rExtDepSpat 73
simplex 75
StdErr 76
tic 77
trans2GEV 78
trans2UFrechet 79
Wind 80
WindSpeedGust 81

Index **82**

`angular` *Estimation of the angular density, angular measure, and random generation from the angular distribution*

Description

Empirical estimation of the Pickands dependence function, the angular density, the angular measure, and random generation of samples from the estimated angular density.

Usage

```
angular(data, model, n, dep, asy, alpha, beta, df, seed, k, nsim,
        plot = TRUE, nw = 100)
```

Arguments

`data`	The dataset in vector form.
`model`	A character string specifying the model. Must be one of: `"log"`, `"alog"`, `"hr"`, `"neglog"`, `"aneglog"`, `"bilog"`, `"negbilog"`, `"ct"`, `"amix"`, `"Extremalt"`.
`n`	The number of random generations from the `model`. Required if `data = NULL`.
`dep`	The dependence parameter for the `model`.
`asy`	A vector of length two for asymmetry parameters, required for asymmetric logistic (`model = 'alog'`) and asymmetric negative logistic (`model = 'aneglog'`) models.
`alpha, beta`	Parameters for the bilogistic, negative bilogistic, Coles-Tawn, and asymmetric mixed models.
`df`	The degrees of freedom for the Extremal-t model.
`seed`	Seed for data generation. Required if `data = NULL`.

k	The polynomial order.
nsim	The number of generations from the estimated angular density.
plot	Logical; if TRUE, plots the fitted angular density, histogram of generated observations, and true angular density (if model is specified).
nw	The number of points at which the estimated functions are evaluated.

Details

See Marcon et al. (2017) for details.

Value

A list containing:

model The specified model.

n Number of random generations.

dep Dependence parameter.

data Input dataset.

Aest Estimated Pickands dependence function.

hest Estimated angular density.

Hest Estimated angular measure.

p0,p1 Point masses at the edge of the simplex.

wsim Simulated sample from the angular density.

Atrue,htrue True Pickands dependence function and angular density, if model is specified.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Marcon, G., Naveau, P. and Padoan, S. A. (2017). A semi-parametric stochastic generator for bivariate extreme events, *Stat 6*(1), 184–201.

Examples

```
################################################
# The following examples correspond to left panels
# of Figures 1, 2 & 3 from Marcon et al. (2017)
################################################

## Figure 1 - symmetric logistic

# Strong dependence
a <- angular(model = 'log', n = 50, dep = 0.3,
```

```
             seed = 4321, k = 20, nsim = 10000)
# Mild dependence
b <- angular(model = 'log', n = 50, dep = 0.6,
             seed = 212, k = 10, nsim = 10000)
# Weak dependence
c <- angular(model = 'log', n = 50, dep = 0.9,
             seed = 4334, k = 6, nsim = 10000)


## Figure 2 - asymmetric logistic

# Strong dependence
d <- angular(model = 'alog', n = 25, dep = 0.3,
             asy = c(0.3,0.8), seed = 43121465, k = 20, nsim = 10000)
# Mild dependence
e <- angular(model = 'alog', n = 25, dep = 0.6,
             asy = c(0.3,0.8), seed = 1890, k = 10, nsim = 10000)
# Weak dependence
f <- angular(model = 'alog', n = 25, dep = 0.9,
             asy = c(0.3,0.8), seed = 2043, k = 5, nsim = 10000)
```

`angular.plot` *Angular density plot.*

Description

This function displays the angular density for bivariate and trivariate extreme values models.

Usage

```
angular.plot(model, par, log, contour, labels, cex.lab, cex.cont, ...)
```

Arguments

`model`	A string with the name of the model considered. Takes value `"PB"` (Pairwise Beta), `"HR"` (Husler-Reiss), `"ET"` (Extremal-t), `"EST"` (Extremal Skew-t), `TD` (Tilted Dirichlet) or `AL` (Asymmetric Logistic).
`par`	A vector representing the parameters of the model.
`log`	A logical value specifying if the log density is computed (default is `TRUE`).
`contour`	A logical value; if `TRUE` the contour levels are displayed. Required for trivariate models only.
`labels`	A vector of character strings indicating the labels. Must be of length 1 for bivariate models and 3 for trivariate models.
`cex.lab`	A positive real indicating the size of the labels.
`cex.cont`	A positive real indicating the size of the contour labels.
`...`	Additional graphical arguments for the `hist()` function when `style = "hist"`.

Details

The angular density is computed using the function dExtDep with arguments method = "Parametric" and angular = TRUE.

When contours are displayed, levels are chosen to be the deciles.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

See Also

dExtDep

Examples

```
angular.plot(model = "HR", par = 0.6)
## Not run:
angular.plot(mode = "ET", par = c(0.6, 0.2, 0.5, 2))

## End(Not run)
```

---

beed *Bernstein Polynomials Based Estimation of Extremal Dependence*

Description

Estimates the Pickands dependence function corresponding to multivariate data on the basis of a Bernstein polynomials approximation.

Usage

```
beed(data, x, d = 3, est = c("ht", "cfg", "md", "pick"),
     margin = c("emp", "est", "exp", "frechet", "gumbel"),
     k = 13, y = NULL, beta = NULL, plot = FALSE)
```

Arguments

data	$(n \times d)$ matrix of component-wise maxima. d is the number of variables and n is the number of replications.
x	$(m \times d)$ design matrix where the dependence function is evaluated (see **Details**).
d	positive integer greater than or equal to two indicating the number of variables (d = 3 by default).
est	string, indicating the estimation method (est = "md" by default, see **Details**).

margin	string, denoting the type marginal distributions (margin = "emp" by default, see **Details**).
k	postive integer, indicating the order of Bernstein polynomials (k = 13 by default).
y	numeric vector (of size m) with an initial estimate of the Pickands function. If NULL, the initial estimate is computed using the estimation method denoted in est.
beta	vector of polynomial coefficients (see **Details**).
plot	logical; if TRUE and d <= 3, the estimated Pickands dependence function is plotted (FALSE by default).

Details

The routine returns an estimate of the Pickands dependence function using the Bernstein polynomials approximation proposed in Marcon et al. (2017). The method is based on a preliminary empirical estimate of the Pickands dependence function. If you do not provide such an estimate, this is computed by the routine. In this case, you can select one of the empirical methods available. est = 'ht' refers to the Hall-Tajvidi estimator (Hall and Tajvidi 2000). With est = 'cfg' the method proposed by Caperaa et al. (1997) is considered. Note that in the multivariate case the adjusted version of Gudendorf and Segers (2011) is used. Finally, with est = 'md' the estimate is based on the madogram defined in Marcon et al. (2017).

Each row of the $(m \times d)$ design matrix x is a point in the unit d-dimensional simplex,

$S_d := \left\{(w_1, \ldots, w_d) \in [0,1]^d : \sum_{i=1}^{d} w_i = 1\right\}.$

With this "regularization"" method, the final estimate satisfies the neccessary conditions in order to be a Pickands dependence function.

$A(\boldsymbol{w}) = \sum_{\boldsymbol{\alpha} \in \Gamma_k} \beta_{\boldsymbol{\alpha}} b_{\boldsymbol{\alpha}}(\boldsymbol{w}; k)$.

The estimates are obtained by solving an optimization quadratic problem subject to the constraints. The latter are represented by the following conditions: $A(e_i) = 1; \max(w_i) \leq A(w) \leq 1; \forall i = 1, \ldots, d$; (convexity).

The order of polynomial k controls the smoothness of the estimate. The higher k is, the smoother the final estimate is. Higher values are better with strong dependence (e. g. k = 23), whereas small values (e.g. k = 6 or k = 10) are enough with mild or weak dependence.

An empirical transformation of the marginals is performed when margin = "emp". A max-likelihood fitting of the GEV distributions is implemented when margin="est". Otherwise it refers to marginal parametric GEV theorethical distributions (margin = "exp", "frechet", "gumbel").

Value

beta	vector of polynomial coefficients
A	numeric vector of the estimated Pickands dependence function A
Anonconvex	preliminary non-convex function
extind	extremal index

Note

The number of coefficients depends on both the order of polynomial k and the dimension d. The number of parameters is explained in Marcon et al. (2017).

The size of the vector beta must be compatible with the polynomial order k chosen.

With the estimated polynomial coefficients, the extremal coefficient, i.e. $d * A(1/d, \ldots, 1/d)$ is computed.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017) Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, **183**, 1-17.

See Also

beed.confband.

Examples

```
x <- simplex(2)
data <- evd::rbvevd(50, dep = 0.4, model = "log", mar1 = c(1, 1, 1))

Amd <- beed(data, x, 2, "md", "emp", 20, plot = TRUE)
Acfg <- beed(data, x, 2, "cfg", "emp", 20)
Aht <- beed(data, x, 2, "ht", "emp", 20)

lines(x[,1], Aht$A, lty = 1, col = 3)
lines(x[,1], Acfg$A, lty = 1, col = 2)

##################################
# Trivariate case
##################################


x <- simplex(3)

data <- evd::rmvevd(50, dep = 0.8, model = "log", d = 3, mar = c(1, 1, 1))

op <- par(mfrow=c(1, 3))
Amd <- beed(data, x, 3, "md", "emp", 18, plot = TRUE)
Acfg <- beed(data, x, 3, "cfg", "emp", 18, plot = TRUE)
Aht <- beed(data, x, 3, "ht", "emp", 18, plot = TRUE)

par(op)
```

beed.boot *Bootstrap Resampling and Bernstein Estimation of Extremal Dependence*

Description

Computes nboot estimates of the Pickands dependence function for multivariate data (using the Bernstein polynomials approximation method) on the basis of the bootstrap resampling of the data.

Usage

```
beed.boot(data, x, d = 3, est = c("ht", "md", "cfg", "pick"),
   margin = c("emp", "est", "exp", "frechet", "gumbel"), k = 13,
   nboot = 500, y = NULL, print = FALSE)
```

Arguments

data	$n \times d$ matrix of component-wise maxima.
x	$m \times d$ design matrix where the dependence function is evaluated, see **Details**.
d	postive integer (greater than or equal to two) indicating the number of variables (d = 3 by default).
est	string denoting the preliminary estimation method (see **Details**).
margin	string denoting the type marginal distributions (see **Details**).
k	postive integer denoting the order of the Bernstein polynomial (k = 13 by default).
nboot	postive integer indicating the number of bootstrap replicates (nboot = 500 by default).
y	numeric vector (of size m) with an initial estimate of the Pickands function. If NULL, The initial estimation is performed by using the estimation method chosen in est.
print	logical; FALSE by default. If TRUE the number of the iteration is printed.

Details

Standard bootstrap is performed, in particular estimates of the Pickands dependence function are provided for each data resampling.

Most of the settings are the same as in the function beed.

An empirical transformation of the marginals is performed when margin = "emp". A max-likelihood fitting of the GEV distributions is implemented when margin = "est". Otherwise it refers to marginal parametric GEV theorethical distributions (margin = "exp", "frechet", "gumbel").

Value

A	numeric vector of the estimated Pickands dependence function.
bootA	matrix with nboot columns that reports the estimates of the Pickands function for each data resampling.
beta	matrix of estimated polynomial coefficients. Each column corresponds to a data resampling.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017) Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, 183, 1-17.

See Also

beed, beed.confband.

Examples

```
x <- simplex(2)
data <- evd::rbvevd(50, dep = 0.4, model = "log", mar1 = c(1, 1, 1))

boot <- beed.boot(data, x, 2, "md", "emp", 20, 500)
```

---

beed.confband *Nonparametric Bootstrap Confidence Intervals*

Description

Computes nonparametric bootstrap $(1-\alpha)\%$ confidence bands for the Pickands dependence function.

Usage

```
beed.confband(data, x, d = 3, est = c("ht", "md", "cfg", "pick"),
   margin = c("emp", "est", "exp", "frechet", "gumbel"), k = 13,
   nboot = 500, y = NULL, conf = 0.95, plot = FALSE, print = FALSE)
```

Arguments

`data`	$(n \times d)$ matrix of component-wise maxima.
`x`	$(m \times d)$ design matrix (see **Details**).
`d`	postive integer (greater than or equal to two) indicating the number of variables (`d = 3` by default).
`est`	string denoting the estimation method (see **Details**).
`margin`	string denoting the type marginal distributions (see **Details**).
`k`	postive integer denoting the order of the Bernstein polynomial (`k = 13` by default).
`nboot`	postive integer indicating the number of bootstrap replicates.
`y`	numeric vector (of size `m`) with an initial estimate of the Pickands function. If `NULL`, the initial estimation is performed by using the estimation method chosen in `est`.
`conf`	real value in $(0, 1)$ denoting the confidence level of the interval. The value `conf = 0.95` is the default.
`plot`	logical; `FALSE` by default. If `TRUE`, the confidence bands are plotted.
`print`	logical; `FALSE` by default. If `TRUE`, the number of the iteration is printed.

Details

Two methods for computing bootstrap $(1-\alpha)\%$ point-wise and simultaneous confidence bands for the Pickands dependence function are used.

The first method derives the confidence bands computing the point-wise $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap sample distribution of the Pickands dependence Bernstein based estimator.

The second method derives the confidence bands, first computing the point-wise $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap sample distribution of polynomial coefficient estimators, and then the Pickands dependence is computed using the Bernstein polynomial representation. See Marcon et al. (2017) for details.

Most of the settings are the same as in the function `beed`.

Value

`A`	numeric vector of the Pickands dependence function estimated.
`bootA`	matrix with `nboot` columns that reports the estimates of the Pickands function for each data resampling.
`A.up.beta/A.low.beta`	vectors of upper and lower bands of the Pickands dependence function obtained using the bootstrap sampling distribution of the polynomial coefficients estimator.
`A.up.pointwise/A.low.pointwise`	vectors of upper and lower bands of the Pickands dependence function obtained using the bootstrap sampling distribution of the Pickands dependence function estimator.
`up.beta/low.beta`	vectors of upper and lower bounds of the bootstrap sampling distribution of the polynomial coefficients estimator.

Note

This routine relies on the bootstrap routine (see beed.boot).

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017) Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, 183, 1-17.

See Also

beed, beed.boot.

Examples

```
x <- simplex(2)
data <- evd::rbvevd(50, dep = 0.4, model = "log", mar1 = c(1, 1, 1))

# Note you should consider 500 bootstrap replications.
# In order to obtain fastest results we used 50!
cb <- beed.confband(data, x, 2, "md", "emp", 20, 50, plot = TRUE)
```

---

bic *Extract the Bayesian Information Criterion*

Description

This function extract the BIC value from a fitted object..

Usage

```
bic(x, digits = 3)
```

Arguments

x	An object of class ExtDep_Bayes.
digits	Integer indicating the number of decimal places to be reported.

Value

A vector.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

See Also

fExtDep

Examples

```
## Not run:
data(pollution)
Hpar.hr <- list(mean.lambda = 0, sd.lambda = 3)
PNS.hr <- fExtDep(x = PNS, method = "BayesianPPP", model = "HR",
                  Nsim = 5e+4, Nbin = 3e+4, Hpar = Hpar.hr,
                  MCpar = 0.35, seed = 14342)
bic(PNS.hr)

## End(Not run)
```

---

desn *Univariate extended skew-normal distribution*

Description

Density function, distribution function for the univariate extended skew-normal (ESN) distribution.

Usage

```
desn(x, location = 0, scale = 1, shape = 0, extended = 0)
pesn(x, location = 0, scale = 1, shape = 0, extended = 0)
```

Arguments

x	quantile.
location	location parameter. 0 is the default.
scale	scale parameter; must be positive. 1 is the default.
shape	skewness parameter. 0 is the default.
extended	extension parameter. 0 is the default.

Value

density (desn), probability (pesn) from the extended skew-normal distribution with given location, scale, shape and extended parameters or from the skew-normal if extended = 0. If shape = 0 and extended = 0 then the normal distribution is recovered.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

References

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scand. J. Statist.* **12**, 171-178.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Examples

```
dens1 <- desn(x = 1, shape = 3, extended = 2)
dens2 <- desn(x = 1, shape = 3)
dens3 <- desn(x = 1)
dens4 <- dnorm(x = 1)
prob1 <- pesn(x = 1, shape = 3, extended = 2)
prob2 <- pesn(x = 1, shape = 3)
prob3 <- pesn(x = 1)
prob4 <- pnorm(q = 1)
```

---

dest *Univariate extended skew-t distribution*

Description

Density function, distribution function for the univariate extended skew-t (EST) distribution.

Usage

```
dest(x, location = 0, scale = 1, shape = 0, extended = 0, df = Inf)
pest(x, location = 0, scale = 1, shape = 0, extended = 0, df = Inf)
```

Arguments

x	quantile.
location	location parameter. 0 is the default.
scale	scale parameter; must be positive. 1 is the default.
shape	skewness parameter. 0 is the default.
extended	extension parameter. 0 is the default.
df	a single positive value representing the degrees of freedom; it can be non-integer. Default value is nu = Inf which corresponds to the skew-normal distribution.

Value

density (dest), probability (pest) from the extended skew-t distribution with given location, scale, shape, extended and df parameters or from the skew-t if extended = 0. If shape = 0 and extended = 0 then the t distribution is recovered.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

References

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew-*t* distribution. *J.Roy. Statist. Soc. B* **65**, 367–389.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-normal and Related Families*. Cambridge University Press, IMS Monographs series.

Examples

```
dens1 <- dest(x = 1, shape = 3, extended = 2, df = 1)
dens2 <- dest(x = 1, shape = 3, df = 1)
dens3 <- dest(x = 1, df = 1)
dens4 <- dt(x = 1, df = 1)
prob1 <- pest(x = 1, shape = 3, extended = 2, df = 1)
prob2 <- pest(x = 1, shape = 3, df = 1)
prob3 <- pest(x = 1, df = 1)
prob4 <- pt(q = 1, df = 1)
```

---

dExtDep *Parametric and non-parametric density of Extremal Dependence*

Description

This function calculates the density of parametric multivariate extreme distributions and corresponding angular density, or the non-parametric angular density represented through Bernstein polynomials.

Usage

```
dExtDep(x, method = "Parametric", model, par, angular = TRUE, log = FALSE,
        c = NULL, vectorial = TRUE, mixture = FALSE)
```

Arguments

x	A vector or a matrix. The value at which the density is evaluated.
method	A character string taking value "Parametric" or "NonParametric"
model	A string with the name of the model: "PB" (Pairwise Beta), "HR" (Husler-Reiss), "ET" (Extremal-t), "EST" (Extremal Skew-t), TD (Tilted Dirichlet) or AL (Asymmetric Logistic) when evaluating the angular density. Restricted to "HR", "ET" and "EST" for multivariate extreme value densities. Required when method = "Parametric".
par	A vector representing the parameters of the (parametric or non-parametric) model.
angular	A logical value specifying if the angular density is computed.
log	A logical value specifying if the log density is computed.
c	A real value in $[0, 1]$, providing the decision rule to allocate a data point to a subset of the simplex. Only required for the parametric angular density of the Extremal-t, Extremal Skew-t and Asymmetric Logistic models.
vectorial	A logical value; if TRUE a vector of nrow(x) densities is returned. If FALSE the likelihood function is returned (product of densities or sum if log = TRUE. TRUE is the default.
mixture	A logical value specifying if the Bernstein polynomial representation of distribution should be expressed as a mixture. If mixture = TRUE the density integrates to 1. Required when method = "NonParametric".

Details

Note that when method = "Parametric" and angular = FALSE, the density is only available in 2 dimensions. When method = "Parametric" and angular = TRUE, the models "AL", "ET" and "EST" are limited to 3 dimensions. This is because of the existence of mass on the subspaces on the simplex (and therefore the need to specify c).

For the parametric models, the appropriate length of the parameter vector can be obtained from the dim_ExtDep function and are summarized as follows. When model = "HR", the parameter vector is of length choose(dim, 2). When model = "PB" or model = "Extremalt", the parameter vector is of length choose(dim, 2) + 1. When model = "EST", the parameter vector is of length choose(dim, 2) + dim + 1. When model = "TD", the parameter vector is of length dim. When model = "AL", the parameter vector is of length 2^(dim - 1) * (dim + 2) - (2 * dim + 1).

Value

If x is a matrix and vectorial = TRUE, a vector of length nrow(x), otherwise a scalar.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

References

Beranger, B. and Padoan, S. A. (2015). Extreme dependence models, chapater of the book *Extreme Value Modeling and Risk Analysis: Methods and Applications*, **Chapman Hall/CRC**.

Beranger, B., Padoan, S. A. and Sisson, S. A. (2017). Models for extremal dependence derived from skew-symmetric families. *Scandinavian Journal of Statistics*, **44**(1), 21-45.

Coles, S. G., and Tawn, J. A. (1991), Modelling Extreme Multivariate Events, *Journal of the Royal Statistical Society, Series B (Methodological)*, **53**, 377–392.

Cooley, D., Davis, R. A., and Naveau, P. (2010). The pairwise beta distribution: a flexible parametric multivariate model for extremes. *Journal of Multivariate Analysis*, **101**, 2103–2117.

Engelke, S., Malinowski, A., Kabluchko, Z., and Schlather, M. (2015), Estimation of Husler-Reiss distributions and Brown-Resnick processes, *Journal of the Royal Statistical Society, Series B (Methodological)*, **77**, 239–265.

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

Marcon, G., Padoan, S. A., Naveau, P., Muliere, P. and Segers, J. (2017) Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, **183**, 1-17.

Nikoloulopoulos, A. K., Joe, H., and Li, H. (2009) Extreme value properties of t copulas. *Extremes*, **12**, 129–148.

Opitz, T. (2013) Extremal t processes: Elliptical domain of attraction and a spectral representation. *Jounal of Multivariate Analysis*, **122**, 409–413.

Tawn, J. A. (1990), Modelling Multivariate Extreme Value Distributions, *Biometrika*, **77**, 245–253.

See Also

pExtDep, rExtDep, fExtDep, fExtDep.np

Examples

```
# Example of PB on the 4-dimensional simplex
dExtDep(x = rbind(c(0.1, 0.3, 0.3, 0.3), c(0.1, 0.2, 0.3, 0.4)),
  method = "Parametric", model = "PB",
  par = c(2, 2, 2, 1, 0.5, 3, 4), log = FALSE)

# Example of EST in 2 dimensions
dExtDep(x = c(1.2, 2.3), method = "Parametric", model = "EST",
  par = c(0.6, 1, 2, 3), angular = FALSE, log = TRUE)

# Example of non-parametric angular density
beta <- c(1.0000000, 0.8714286, 0.7671560, 0.7569398,
          0.7771908, 0.8031573, 0.8857143, 1.0000000)
dExtDep(x = rbind(c(0.1, 0.9), c(0.2, 0.8)), method = "NonParametric", par = beta)
```

dGEV *The Generalized Extreme Value Distribution*

Description

Density, distribution and quantile function for the Generalized Extreme Value (GEV) distribution.

Usage

```
dGEV(x, loc, scale, shape, log = FALSE)
pGEV(q, loc, scale, shape, lower.tail = TRUE)
qGEV(p, loc, scale, shape)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
loc	Vector of locations.
scale	Vector of scales.
shape	Vector of shapes.
log	Logical; if TRUE, returns the log density.
lower.tail	Logical; if TRUE, probabilities are $P(X \leq x)$, otherwise $P(X > x)$.

Details

The GEV distribution has density

$$f(x; \mu, \sigma, \xi) = \exp\left\{-\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]_+^{-1/\xi}\right\}$$

Value

Density (dGEV), distribution function (pGEV) and quantile function (qGEV) from the Generalized Extreme Value distribution with given location, scale and shape.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com.

See Also

fGEV

Examples

```
# Densities
dGEV(x = 1, loc = 1, scale = 1, shape = 1)
dGEV(x = c(0.2, 0.5), loc = 1, scale = 1, shape = c(0, 0.3))

# Probabilities
pGEV(q = 1, loc = 1, scale = 1, shape = 1, lower.tail = FALSE)
pGEV(q = c(0.2, 0.5), loc = 1, scale = 1, shape = c(0, 0.3))

# Quantiles
qGEV(p = 0.5, loc = 1, scale = 1, shape = 1)
qGEV(p = c(0.2, 0.5), loc = 1, scale = 1, shape = c(0, 0.3))
```

---

diagnostics *Diagnostics plots for MCMC algorithm*

Description

This function displays traceplots of the scaling parameter from the proposal distribution of the adaptive MCMC scheme and the associated acceptance probability.

Usage

```
diagnostics(mcmc)
```

Arguments

mcmc	An output of the fGEV or fExtDep.np function with method = "Bayesian".

Details

When mcmc is the output of fGEV, this corresponds to a marginal estimation. In this case, diagnostics displays:

- A trace plot of τ, the scaling parameter in the multivariate normal proposal, which directly affects the acceptance rate.
- A trace plot of the acceptance probabilities of the proposal parameter values.

When mcmc is the output of fExtDep.np, this corresponds to an estimation of the dependence structure following Algorithm 1 in Beranger et al. (2021).

- If the margins are jointly estimated with the dependence (steps 1 and 2), diagnostics provides trace plots of the corresponding scaling parameters (τ_1, τ_2) and acceptance probabilities.
- For the dependence structure (step 3), a trace plot of the polynomial order κ is displayed, along with the associated acceptance probability.

Value

A graph of traceplots of the scaling parameter from the proposal distribution of the adaptive MCMC scheme and the associated acceptance probability.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Beranger, B., Padoan, S. A. and Sisson, S. A. (2021). Estimation and uncertainty quantification for extreme quantile regions. *Extremes*, **24**, 349–375.

See Also

fExtDep.np

Examples

```
################################################
### Example - Pollution levels in Milan, Italy ###
################################################

## Not run:

  # Dependence structure only
  data(MilanPollution)

  data <- Milan.winter[, c("NO2", "SO2")]
  data <- as.matrix(data[complete.cases(data), ])

  # Threshold
  u <- apply(data, 2, function(x) quantile(x, prob = 0.9, type = 3))

  # Hyperparameters
  hyperparam <- list(mu.nbinom = 6, var.nbinom = 8, a.unif = 0, b.unif = 0.2)

  # Standardise data to univariate Frechet margins
  f1 <- fGEV(data = data[, 1], method = "Bayesian", sig0 = 0.1, nsim = 5e4)
  diagnostics(f1)
  burn1 <- 1:30000
  gev.pars1 <- apply(f1$param_post[-burn1, ], 2, mean)
  sdata1 <- trans2UFrechet(data = data[, 1], pars = gev.pars1, type = "GEV")

  f2 <- fGEV(data = data[, 2], method = "Bayesian", sig0 = 0.1, nsim = 5e4)
  diagnostics(f2)
  burn2 <- 1:30000
  gev.pars2 <- apply(f2$param_post[-burn2, ], 2, mean)
  sdata2 <- trans2UFrechet(data = data[, 2], pars = gev.pars2, type = "GEV")
```

```
  sdata <- cbind(sdata1, sdata2)

  # Bayesian estimation using Bernstein polynomials
  pollut1 <- fExtDep.np(method = "Bayesian", data = sdata,
         u = TRUE, mar.fit = FALSE, k0 = 5,
         hyperparam = hyperparam, nsim = 5e4)

  diagnostics(pollut1)


## End(Not run)
```

`dim_ExtDep` *Dimensions calculations for parametric extremal dependence models*

Description

This function calculates the dimensions of an extremal dependence model for a given set of parameters, the dimension of the parameter vector for a given dimension and verifies the adequacy between model dimension and length of parameter vector when both are provided.

Usage

```
dim_ExtDep(model, par = NULL, dim = NULL)
```

Arguments

`model`	A string with the name of the model: `"PB"` (Pairwise Beta), `"HR"` (Husler-Reiss), `"ET"` (Extremal-t), `"EST"` (Extremal Skew-t), `"TD"` (Tilted Dirichlet) or `"AL"` (Asymmetric Logistic).
`par`	A vector representing the parameters of the model.
`dim`	An integer representing the dimension of the model.

Details

One of `par` or `dim` needs to be provided. If `par` is provided, the dimension of the model is calculated. If `dim` is provided, the length of the parameter vector is calculated. If both `par` and `dim` are provided, the adequacy between the dimension of the model and the length of the parameter vector is checked.

For `model = "HR"`, the parameter vector is of length `choose(dim, 2)`. For `model = "PB"` or `model = "ET"`, the parameter vector is of length `choose(dim, 2) + 1`. For `model = "EST"`, the parameter vector is of length `choose(dim, 2) + dim + 1`. For `model = "TD"`, the parameter vector is of length `dim`. For `model = "AL"`, the parameter vector is of length `2^(dim - 1) * (dim + 2) - (2 * dim + 1)`.

Value

If par is not provided and dim is provided: returns an integer indicating the length of the parameter vector. If par is provided and dim is not provided: returns an integer indicating the dimension of the model. If par and dim are provided: returns a TRUE/FALSE statement indicating whether the length of the parameter and the dimension match.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

Examples

```
dim_ExtDep(model = "EST", dim = 3)
dim_ExtDep(model = "AL", dim = 3)

dim_ExtDep(model = "PB", par = rep(0.5, choose(4, 2) + 1))
dim_ExtDep(model = "TD", par = rep(1, 5))

dim_ExtDep(model = "EST", dim = 2, par = c(0.5, 1, 1, 1))
dim_ExtDep(model = "PB", dim = 4, par = rep(0.5, choose(4, 2) + 1))
```

---

dmesn *Bivariate and Trivariate Extended Skew-Normal Distribution*

Description

Density function and distribution function for the bivariate and trivariate extended skew-normal (ESN) distribution.

Usage

```
dmesn(
  x = c(0, 0),
  location = rep(0, length(x)),
  scale = diag(length(x)),
  shape = rep(0, length(x)),
  extended = 0
)

pmesn(
  x = c(0, 0),
  location = rep(0, length(x)),
  scale = diag(length(x)),
  shape = rep(0, length(x)),
  extended = 0
)
```

Arguments

x	Quantile vector of length d = 2 or d = 3.
location	A numeric location vector of length d. Default is 0.
scale	A symmetric positive-definite scale matrix of dimension (d, d). Default is diag(d).
shape	A numeric skewness vector of length d. Default is 0.
extended	A single extension parameter. Default is 0.

Value

dmesn returns the density, and pmesn returns the probability, of the bivariate or trivariate extended skew-normal distribution with the specified location, scale, shape, and extended parameters. If extended = 0, the skew-normal distribution is obtained. If shape = 0 and extended = 0, the normal distribution is recovered.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com.

References

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J. Roy. Statist. Soc. B* **61**, 579–602.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Azzalini, A. and Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika* **83**, 715–726.

Examples

```
sigma1 <- matrix(c(2, 1.5, 1.5, 3), ncol = 2)
sigma2 <- matrix(c(
  2, 1.5, 1.8,
  1.5, 3, 2.2,
  1.8, 2.2, 3.5
), ncol = 3)

shape1 <- c(1, 2)
shape2 <- c(1, 2, 1.5)

dens1 <- dmesn(x = c(1, 1), scale = sigma1, shape = shape1, extended = 2)
dens2 <- dmesn(x = c(1, 1), scale = sigma1)
dens3 <- dmesn(x = c(1, 1, 1), scale = sigma2, shape = shape2, extended = 2)
dens4 <- dmesn(x = c(1, 1, 1), scale = sigma2)

prob1 <- pmesn(x = c(1, 1), scale = sigma1, shape = shape1, extended = 2)
prob2 <- pmesn(x = c(1, 1), scale = sigma1)
```

```
prob3 <- pmesn(x = c(1, 1, 1), scale = sigma2, shape = shape2, extended = 2)
prob4 <- pmesn(x = c(1, 1, 1), scale = sigma2)
```

---

dmest *Bivariate and trivariate extended skew-t distribution*

Description

Density function, distribution function for the bivariate and trivariate extended skew-t (EST) distribution.

Usage

```
dmest(
  x = c(0, 0),
  location = rep(0, length(x)),
  scale = diag(length(x)),
  shape = rep(0, length(x)),
  extended = 0,
  df = Inf
)

pmest(
  x = c(0, 0),
  location = rep(0, length(x)),
  scale = diag(length(x)),
  shape = rep(0, length(x)),
  extended = 0,
  df = Inf
)
```

Arguments

x	Quantile vector of length d = 2 or d = 3.
location	A numeric location vector of length d. 0 is the default.
scale	A symmetric positive-definite scale matrix of dimension (d, d). diag(d) is the default.
shape	A numeric skewness vector of length d. 0 is the default.
extended	A single value extension parameter. 0 is the default.
df	A single positive value representing the degree of freedom; it can be non-integer. Default value is nu = Inf, which corresponds to the skew-normal distribution.

Value

Density (dmest), probability (pmest) from the bivariate or trivariate extended skew-t distribution with given location, scale, shape, extended and df parameters, or from the skew-t distribution if extended = 0. If shape = 0 and extended = 0, then the t distribution is recovered.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

References

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew *t* distribution. *J. Roy. Statist. Soc. B* **65**, 367–389.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monograph series.

Examples

```
sigma1 <- matrix(c(2, 1.5, 1.5, 3), ncol = 2)
sigma2 <- matrix(c(
  2, 1.5, 1.8,
  1.5, 3, 2.2,
  1.8, 2.2, 3.5
), ncol = 3)

shape1 <- c(1, 2)
shape2 <- c(1, 2, 1.5)

dens1 <- dmest(x = c(1, 1), scale = sigma1, shape = shape1, extended = 2, df = 1)
dens2 <- dmest(x = c(1, 1), scale = sigma1, df = 1)
dens3 <- dmest(x = c(1, 1, 1), scale = sigma2, shape = shape2, extended = 2, df = 1)
dens4 <- dmest(x = c(1, 1, 1), scale = sigma2, df = 1)

prob1 <- pmest(x = c(1, 1), scale = sigma1, shape = shape1, extended = 2, df = 1)
prob2 <- pmest(x = c(1, 1), scale = sigma1, df = 1)


prob3 <- pmest(x = c(1, 1, 1), scale = sigma2, shape = shape2, extended = 2, df = 1)
prob4 <- pmest(x = c(1, 1, 1), scale = sigma2, df = 1)
```

---

ellipse *Level sets for bivariate normal, student-t and skew-normal distributions probability densities.*

Description

Level sets of the bivariate normal, student-t and skew-normal distributions probability densities for a given probability.

Usage

```
ellipse(
  center = c(0, 0),
  alpha = c(0, 0),
  sigma = diag(2),
  df = 1,
  prob = 0.01,
  npoints = 250,
  pos = FALSE
)
```

Arguments

`center`	A vector of length 2 corresponding to the location of the distribution.
`alpha`	A vector of length 2 corresponding to the skewness of the skew-normal distribution.
`sigma`	A 2 by 2 variance-covariance matrix.
`df`	An integer corresponding to the degree of freedom of the student-t distribution.
`prob`	The probability level. See `details`.
`npoints`	The maximum number of points at which it is evaluated.
`pos`	If `pos = TRUE` only the region on the positive quadrant is kept.

Details

The level sets are defined as

$R(f_\alpha) = \{x : f(x) \geq f_\alpha\}$

where f_α is the largest constant such that

$P(X \in R(f_\alpha)) \geq 1 - \alpha$.

Here we consider $f(x)$ to be the bivariate normal, student-t or skew-normal density.

Value

Returns a bivariate vector of 250 rows if `pos = FALSE`, and half otherwise.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com.

Examples

```
library(mvtnorm)

# Data simulation (Bivariate-t on positive quadrant)
rho <- 0.5
sigma <- matrix(c(1, rho, rho, 1), ncol = 2)
df <- 2

set.seed(101)
n <- 1500
data <- rmvt(5 * n, sigma = sigma, df = df)
data <- data[data[, 1] > 0 & data[, 2] > 0, ]
data <- data[1:n, ]

P <- c(1 / 750, 1 / 1500, 1 / 3000)

ell1 <- ellipse(prob = 1 - P[1], sigma = sigma, df = df, pos = TRUE)
ell2 <- ellipse(prob = 1 - P[2], sigma = sigma, df = df, pos = TRUE)
ell3 <- ellipse(prob = 1 - P[3], sigma = sigma, df = df, pos = TRUE)

plot(
  data,
  xlim = c(0, max(data[, 1], ell1[, 1], ell2[, 1], ell3[, 1])),
  ylim = c(0, max(data[, 2], ell1[, 2], ell2[, 2], ell3[, 2])),
  pch = 19
)
points(ell1, type = "l", lwd = 2, lty = 1)
points(ell2, type = "l", lwd = 2, lty = 1)
points(ell3, type = "l", lwd = 2, lty = 1)
```

---

est *Extract the estimated parameter*

Description

This function extracts the estimated parameters from a fitted object.

Usage

```
est(x, digits = 3)
```

Arguments

x	An object of class ExtDep_Bayes, ExtDep_Freq or ExtDep_Spat.
digits	Integer indicating the number of decimal places to be reported.

Value

A vector.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com.

See Also

fExtDepSpat, fExtDep

Examples

```
data(pollution)

f.hr <- fExtDep(
  x = PNS,
  method = "PPP",
  model = "HR",
  par.start = rep(0.5, 3),
  trace = 2
)

est(f.hr)
```

ExtQ *Univariate Extreme Quantile*

Description

Computes the extreme-quantiles of a univariate random variable corresponding to some exceedance probabilities.

Usage

```
ExtQ(
  P = NULL,
  method = "Frequentist",
  pU = NULL,
  cov = NULL,
  param = NULL,
  param_post = NULL
)
```

Arguments

P	A vector with values in $[0, 1]$ indicating the probabilities of the quantiles to be computed.
method	A character string indicating the estimation method. Takes value "bayesian" or "frequentist".

pU — A value in $[0, 1]$ indicating the probability of exceeding a high threshold. In the estimation procedure, observations below the threshold are censored.

cov — A $q \times c$ matrix indicating q observations of $c - 1$ covariates for the location parameter.

param — A $(c+2)$ vector indicating the estimated parameters. Required when `method="Frequentist"`.

param_post — A $n \times (c+2)$ matrix indicating the posterior sample for the parameters, where n is the number of MCMC replicates after removal of the burn-in period. Required when `method="Bayesian"`.

Details

The first column of `cov` is a vector of 1s corresponding to the intercept.

When `pU` is `NULL` (default), then it is assumed that a block maxima approach was taken and quantiles are computed using the `qGEV` function. When `pU` is provided, it is assumed that a threshold exceedances approach is taken and the quantiles are computed as

$$\mu + \sigma \left(\left(\frac{pU}{P} \right)^{\xi} - 1 \right) \frac{1}{\xi}$$

Value

When `method == "frequentist"`, the function returns a vector of length `length(P)` if `ncol(cov) = 1` (constant mean) or a `(length(P) x nrow(cov))` matrix if `ncol(cov) > 1`.

When `method == "bayesian"`, the function returns a `(length(param_post) x length(P))` matrix if `ncol(cov) = 1` or a list of `ncol(cov)` elements each taking a `(length(param_post) x length(P))` matrix if `ncol(cov) > 1`.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com

References

Beranger, B., Padoan, S. A. and Sisson, S. A. (2021). Estimation and uncertainty quantification for extreme quantile regions. *Extremes*, **24**, 349-375.

See Also

`fGEV`, `qGEV`

Examples

```
#################################################
### Example - Pollution levels in Milan, Italy ###
#################################################
```

```
## Not run:

data(MilanPollution)

# Frequentist estimation
fit <- fGEV(Milan.winter$PM10)
fit$est

q1 <- ExtQ(
  P = 1 / c(600, 1200, 2400),
  method = "Frequentist",
  param = fit$est
)
q1

# Bayesian estimation with high threshold
cov <- cbind(
  rep(1, nrow(Milan.winter)),
  Milan.winter$MaxTemp,
  Milan.winter$MaxTemp^2
)
u <- quantile(Milan.winter$PM10, prob = 0.9, type = 3, na.rm = TRUE)

fit2 <- fGEV(
  data = Milan.winter$PM10,
  par.start = c(50, 0, 0, 20, 1),
  method = "Bayesian",
  u = u,
  cov = cov,
  sig0 = 0.1,
  nsim = 5e+4
)

r <- range(Milan.winter$MaxTemp, na.rm = TRUE)
t <- seq(from = r[1], to = r[2], length = 50)
pU <- mean(Milan.winter$PM10 > u, na.rm = TRUE)

q2 <- ExtQ(
  P = 1 / c(600, 1200, 2400),
  method = "Bayesian",
  pU = pU,
  cov = cbind(rep(1, 50), t, t^2),
  param_post = fit2$param_post[-c(1:3e+4), ]
)

R <- c(min(unlist(q2)), 800)
qseq <- seq(from = R[1], to = R[2], length = 512)
Xl <- "Max Temperature"
Yl <- expression(PM[10])

for (i in seq_along(q2)) {
  K_q2 <- apply(q2[[i]], 2, function(x) density(x, from = R[1], to = R[2])$y)
  D <- cbind(expand.grid(t, qseq), as.vector(t(K_q2)))
```

```
  colnames(D) <- c("x", "y", "z")
  fields::image.plot(
    x = t, y = qseq, z = matrix(D$z, 50, 512),
    xlim = r, ylim = R, xlab = Xl, ylab = Yl
  )
}


## End(Not run)

############################################################
### Example - Simulated data from Frechet distribution ###
############################################################

if (interactive()) {

set.seed(999)
data <- extraDistr::rfrechet(n = 1500, mu = 3, sigma = 1, lambda = 1/3)

u <- quantile(data, probs = 0.9, type = 3)
fit3 <- fGEV(
  data = data,
  par.start = c(1, 2, 1),
  method = "Bayesian",
  u = u,
  sig0 = 1,
  nsim = 5e+4
)

pU <- mean(data > u)
P <- 1 / c(750, 1500, 3000)

q3 <- ExtQ(
  P = P,
  method = "Bayesian",
  pU = pU,
  param_post = fit3$param_post[-c(1:3e+4), ]
)

### Illustration

# Tail index estimation
ti_true <- 3
ti_ps <- fit3$param_post[-c(1:3e+4), 3]

K_ti <- density(ti_ps) # KDE of the tail index
H_ti <- hist(
  ti_ps, prob = TRUE, col = "lightgrey",
  ylim = range(K_ti$y), main = "", xlab = "Tail Index",
  cex.lab = 1.8, cex.axis = 1.8, lwd = 2
)
ti_ic <- quantile(ti_ps, probs = c(0.025, 0.975))
```

```
points(x = ti_ic, y = c(0, 0), pch = 4, lwd = 4)
lines(K_ti, lwd = 2, col = "dimgrey")
abline(v = ti_true, lwd = 2)
abline(v = mean(ti_ps), lwd = 2, lty = 2)

# Quantile estimation
q3_true <- extraDistr::qfrechet(
  p = P, mu = 3, sigma = 1, lambda = 1/3, lower.tail = FALSE
)

ci <- apply(log(q3), 2, function(x) quantile(x, probs = c(0.025, 0.975)))
K_q3 <- apply(log(q3), 2, density)

R <- range(log(c(q3_true, q3, data)))
Xlim <- c(log(quantile(data, 0.95)), R[2])
Ylim <- c(0, max(K_q3[[1]]$y, K_q3[[2]]$y, K_q3[[3]]$y))

plot(
  0, main = "", xlim = Xlim, ylim = Ylim,
  xlab = expression(log(x)), ylab = "Density",
  cex.lab = 1.8, cex.axis = 1.8, lwd = 2
)

cval <- c(211, 169, 105)
for (j in seq_along(P)) {
  col <- rgb(cval[j], cval[j], cval[j], 0.8 * 255, maxColorValue = 255)
  col2 <- rgb(cval[j], cval[j], cval[j], 255, maxColorValue = 255)
  polygon(K_q3[[j]], col = col, border = col2, lwd = 4)
}

points(log(data), rep(0, n), pch = 16)
# add posterior means
abline(v = apply(log(q3), 2, mean), lwd = 2, col = 2:4)
# add credible intervals
abline(v = ci[1, ], lwd = 2, lty = 3, col = 2:4)
abline(v = ci[2, ], lwd = 2, lty = 3, col = 2:4)

}
```

`fExtDep` *Extremal Dependence Estimation*

Description

Estimate the parameters of extremal dependence models using frequentist, composite likelihood, or Bayesian approaches.

Usage

```
fExtDep(x, method = "PPP", model, par.start = NULL,
        c = 0, optim.method = "BFGS", trace = 0,
        Nsim, Nbin = 0, Hpar, MCpar, seed = NULL)

## S3 method for class 'ExtDep_Freq'
plot(x, type, log = TRUE, contour = TRUE, style, labels,
                          cex.dat = 1, cex.lab = 1, cex.cont = 1, Q.fix, Q.range,
                            Q.range0, cond = FALSE, ...)

## S3 method for class 'ExtDep_Freq'
logLik(object, ...)

## S3 method for class 'ExtDep_Bayes'
plot(x, type, log = TRUE, contour = TRUE, style, labels,
                          cex.dat = 1, cex.lab = 1, cex.cont = 1, Q.fix, Q.range,
                            Q.range0, cond = FALSE, cred.ci = TRUE, subsamp, ...)

## S3 method for class 'ExtDep_Bayes'
summary(object, cred = 0.95, plot = FALSE, ...)
```

Arguments

x	fExtDep: A matrix containing the data. plot methods: an object returned by fExtDep.
object	For summary.ExtDep_Bayes: an object of class ExtDep_Bayes. For logLik: an object returned by fExtDep.
method	Estimation method: "PPP", "BayesianPPP", or "Composite".
model	Name of the model. For "PPP" or "BayesianPPP": "PB", "HR", "ET", "EST", "TD", "AL". For "Composite": "HR", "ET", or "EST".
par.start	Vector of initial parameter values for optimization.
c	Real in $[0, 1]$, required for some models under "PPP" or "BayesianPPP" ("ET", "EST", "AL"). See dExtDep.
optim.method	Optimization algorithm (see optim). Required for "PPP" or "Composite".
trace	Non-negative integer controlling optimization progress output (see optim).
Nsim	Number of MCMC simulations (for "BayesianPPP").
Nbin	Burn-in length (for "BayesianPPP").
Hpar	List of hyper-parameters (see Details). Required for "BayesianPPP".
MCpar	Variance of the proposal distribution (see Details). Required for "BayesianPPP".
seed	Integer seed for reproducibility (passed to set.seed).
type	For plot methods: plot type, one of "angular", "pickands", or "returns".
log	Logical; applies to "angular" and "pickands" plots (see angular.plot, pickands.plot).
contour	Logical; applies to "angular" and "pickands" plots.

style	For "angular" plots: "hist" or "ticks" (default).
labels	Labels for axes in plot methods.
cex.dat	Point size for 3D angular plots.
cex.lab	Label size in plots.
cex.cont	Contour line size in "angular" or "pickands" plots.
Q.fix, Q.range, Q.range0, cond	Arguments for "returns" plots (see returns.plot).
cred.ci	Logical, for "returns" plots under "BayesianPPP"; if TRUE, compute 95% credible bands.
subsamp	Posterior subsample percentage (used with cred.ci=TRUE).
cred	Credible interval coverage probability (default 0.95).
plot	Logical; if TRUE, plot kernel densities of posterior parameters (for summary.ExtDep_Bayes).
...	Additional graphical or density arguments (see Details).

Details

Estimation::

- method="PPP": Approximate likelihood estimation using dExtDep(method="Parametric", angular=TRUE).
- method="BayesianPPP": Bayesian estimation of the spectral measure (Sabourin et al., 2013; Sabourin & Naveau, 2014). Requires Hpar and MCpar. Hyper-parameters depend on the model (see references for details).
- method="Composite": Pairwise composite likelihood using dExtDep(method="Parametric", angular=FALSE).

Plotting:: See angular.plot, pickands.plot, and returns.plot. Angular plots can display data as histograms (style="hist") or ticks (style="ticks"). For trivariate cases, use cex.dat to control point size.

Value

fExtDep:

- For "PPP" or "Composite": an object of class ExtDep_Freq with elements

  model The fitted model.
  par Estimated parameters.
  LL Maximized log-likelihood.
  SE Standard errors.
  TIC Takeuchi Information Criterion.
  data Input data.

- For "BayesianPPP": an object of class ExtDep_Bayes with elements

  stored.values Posterior sample matrix of size $(Nsim - Nbin) \times d$.
  llh Log-likelihoods at posterior samples.
  lprior Log-priors at posterior samples.

arguments Algorithm details.
elapsed Elapsed run time.
Nsim, Nbin Simulation settings.
n.accept, n.accept.kept MCMC acceptance counts.
emp.mean Posterior means.
emp.sd Posterior standard deviations.
BIC Bayesian Information Criterion.

`logLik`: numerical log-likelihood value.

Author(s)

Simone Padoan (<simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/)
Boris Beranger (<borisberanger@gmail.com>, https://www.borisberanger.com)

References

Beranger, B. and Padoan, S. A. (2015). *Extreme Dependence Models*, in *Extreme Value Modeling and Risk Analysis: Methods and Applications*, Chapman & Hall/CRC.

Sabourin, A., Naveau, P., and Fougeres, A.-L. (2013). Bayesian model averaging for multivariate extremes. *Extremes*, **16**, 325-350.

Sabourin, A. and Naveau, P. (2014). Bayesian Dirichlet mixture model for multivariate extremes: A re-parametrization. *Computational Statistics & Data Analysis*, **71**, 542-567.

See Also

dExtDep, pExtDep, rExtDep, fExtDep.np

Examples

```
# Poisson Point Process approach
data(pollution)

  f.hr <- fExtDep(x = PNS, method = "PPP", model = "HR",
                  par.start = rep(0.5, 3), trace = 2)
  plot(f.hr, type = "angular",
       labels = c(expression(PM[10]), expression(NO), expression(SO[2])),
       cex.lab = 2)
  plot(f.hr, type = "pickands",
       labels = c(expression(PM[10]), expression(NO), expression(SO[2])),
       cex.lab = 2) # may be slow


# Pairwise composite likelihood

  set.seed(1)
  data <- rExtDep(n = 300, model = "ET", par = c(0.6, 3))
  f.et <- fExtDep(x = data, method = "Composite", model = "ET",
                  par.start = c(0.5, 1), trace = 2)
  plot(f.et, type = "angular", cex.lab = 2)
```

fExtDep.np *Non-parametric extremal dependence estimation*

Description

This function estimates the bivariate extremal dependence structure using a non-parametric approach based on Bernstein Polynomials.

Usage

```
fExtDep.np(
  x, method, cov1 = NULL, cov2 = NULL, u, mar.fit = TRUE,
  mar.prelim = TRUE, par10, par20, sig10, sig20, param0 = NULL,
  k0 = NULL, pm0 = NULL, prior.k = "nbinom", prior.pm = "unif",
  nk = 70, lik = TRUE,
  hyperparam = list(mu.nbinom = 3.2, var.nbinom = 4.48),
  nsim, warn = FALSE, type = "rawdata"
)

## S3 method for class 'ExtDep_npBayes'
plot(
  x, type, summary.mcmc, burn, y, probs,
  A_true, h_true, est.out, mar1, mar2, dep,
  QatCov1 = NULL, QatCov2 = QatCov1, P,
  CEX = 1.5, col.data, col.Qfull, col.Qfade, data = NULL, ...
)

## S3 method for class 'ExtDep_npFreq'
plot(
  x, type, est.out, mar1, mar2, dep, P, CEX = 1.5,
  col.data, col.Qfull, data = NULL, ...
)

## S3 method for class 'ExtDep_npEmp'
plot(
  x, type, est.out, mar1, mar2, dep, P, CEX = 1.5,
  col.data, col.Qfull, data = NULL, ...
)

## S3 method for class 'ExtDep_npBayes'
summary(
  object, w, burn, cred = 0.95, plot = FALSE, ...
)
```

Arguments

x	fExtDep.np: A matrix containing the data.

plot method functions: any object returned by fExtDep.np.

object — A list object of class ExtDep_npBayes.

method — A character string indicating the estimation method inlcuding "Bayesian", "Frequentist" and "Empirical".

cov1, cov2 — A covariate vector/matrix for linear model on the location parameter of the marginal distributions. length(cov1)/nrow(cov1) needs to match nrow(data). If NULL it is assumed to be constant. Required when method="Bayesian".

u — When method="Bayesian": a bivariate indicating the threshold for the exceedance approach. If logical TRUE the threshold are calculated by default as the 90% quantiles. When missing, a block maxima approach is considered. When method="Frequentist": the associated quantile is used to select observations with the largest radial components. If missing, the 90% quantile is used.

mar.fit — A logical value indicated whether the marginal distributions should be fitted. When method="Frequentist", data are empirically transformed to unit Frechet scale if mar.fit=TRUE. Not required when method="Empirical".

rawdata — A character string specifying if the data is "rawdata" or "maxima". Required when method="Frequentist".

mar.prelim — A logical value indicated whether a preliminary fit of marginal distributions should be done prior to estimating the margins and dependence. Required when method="Bayesian".

par10, par20 — Vectors of starting values for the marginal parameter estimation. Required when method="Bayesian" and mar.fit=TRUE.

sig10, sig20 — Positive reals representing the initial value for the scaling parameter of the multivariate normal proposal distribution for both margins. Required when method="Bayesian" and mar.fit=TRUE.

param0 — A vector of initial value for the Bernstein polynomial coefficients. It should be a list with elements $eta and $beta which can be generated by the internal function rcoef if missing. Required when method="Bayesian".

k0 — An integer indicating the initial value of the polynomial order. Required when method="Bayesian".

pm0 — A list of initial values for the probability masses at the boundaries of the simplex. It should be a list with two elements $p0 and $p1. Randomly generated if missing. Required when method="Bayesian".

prior.k — A character string indicating the prior distribution on the polynomial order. By default prior.k="nbinom" (negative binomial) but can also be "pois" (Poisson). Required when method="Bayesian".

prior.pm — A character string indicating the prior on the probability masses at the endpoints of the simplex. By default prior.pm="unif" (uniform) but can also be "beta" (beta). Required when method="Bayesian".

nk — An integer indicating the maximum polynomial order. Required when method="Bayesian".

lik — A logical value; if FALSE, an approximation of the likelihood, by means of the angular measure in Bernstein form is used (TRUE by default). Required when method="Bayesian".

`hyperparam` A list of the hyper-parameters depending on the choice of `prior.k` and `prior.pm`. See **details**. Required when `method="Bayesian"`.

`nsim` An integer indicating the number of iterations in the Metropolis-Hastings algorithm. Required when `method="Bayesian"`.

`warn` A logical value. If `TRUE` warnings are printed when some specifics (e.g., `param0`, `k0`, `pm0` and `hyperparam`) are not provided and set by default. Required when `method="Bayesian"`.

`type` `fExtDep.np`: A character string indicating whther the data are `"rawdata"` or `"maxima"`. Required when `method="Bayesian"`.

`plot` method function: A character string indicating the type of graphical summary to be plotted. Can take value `"summary"` or `"Qsets"` for any class type of `x`, `"A"` when `x` is of class `ExtDep_npBayes` or `ExtDep_npFreq`, `"returns"`, `"h"`, `"pm"` and `"k"` when `x` is of class `ExtDep_npBayes` or `"psi"` `x` is of class `ExtDep_npEmp`.

`summary.mcmc` The output of the `summary.ExtDep_npBayes` function.

`burn` The burn-in period.

`y` A 2-column matrix of unobserved thresholds at which the returns are calculated. Required when `type="returns"`.

`probs` The probability of joint exceedances, the output of the `return` function.

`A_true` A vector representing the true pickands dependence function evaluated at the grid points on the simplex given in the `summary.mcmc` object.

`h_true` A vector representing the true angular density function evaluated at the grid points on the simplex given in the `summary.mcmc` object.

`est.out` A list containing:

ghat: a 3-row matrix giving the posterior summary for the estimate of the bound;

Shat and Shat_post: a matrix of the posterior and a 3-row matrix giving the posterior summary for the estimate of the basic set S;

nuShat and nuShat_post: a matrix of the posterior and a 3-row matrix giving the posterior summary for the estimate of the measure $\nu(S)$;

Note that a posterior summary is made of its mean and 90% credibility interval.

Only required when `x` is of class `ExtDep_npBayes` and `type="Qsets"`.

`mar1, mar2` Vectors of marginal GEV parameters. Required when `type="Qsets"` except when `x` is of class `ExtDep_npBayes` and the marginal parameter were fixed.

`dep` A logical value; if `TRUE` the estimate of the dependence is plotted when computing extreme quantile regions (`type="Qsets"`).

`QatCov1, QatCov2` Matrices representing the value of the covariates at which extreme quantile regions should be computed. Required when `type="Qsets"`. See arguments `cov1` and `cov2` from `fExtDep.np`.

`P` A vector indicating the probabilities associated with the quantiles to be computed. Required when `type="Qsets"`.

`CEX` Label and axis sizes.

col.data, col.Qfull, col.Qfade
: Colors for data, estimate of extreme quantile regions and its credible interval (when applicable). Required when type="Qsets".

data
: A 2-column matrix providing the original data to be plotted when type="Qsets".

w
: A matrix or vector of values on the unit simplex. If vector values need to be between 0 and 1, if matrix each row need to sum to one.

cred
: A probability for the credible intervals.

plot
: A logical value indicating whether plot.ExtDep_npBayes should be called.

...
: Additional graphical parameters

Details

Regarding the fExtDep.np function:

When method="Bayesian", the vector of hyper-parameters is provided in the argument hyperparam. It should include:

If prior.pm="unif" requires hyperparam$a.unif and hyperparam$b.unif.

If prior.pm="beta" requires hyperparam$a.beta and hyperparam$b.beta.

If prior.k="pois" requires hyperparam$mu.pois.

If prior.k="nbinom" requires hyperparam$mu.nbinom and hyperparam$var.nbinom or hyperparam$pnb and hyperparam$rnb. The relationship is pnb = mu.nbinom/var.nbinom and rnb = mu.nbinom^2 / (var.nbinom - mu.nbinom).

When u is specified Algorithm 1 of Beranger et al. (2021) is applied whereas when it is not specified Algorithm 3.5 of Marcon et al. (2016) is considered.

When method="Frequentist", if type="rawdata" then pseudo-polar coordinates are extracted and only observations with a radial component above some high threshold (the quantile equivalent of u for the raw data) are retained. The inferential approach proposed in Marcon et al. (2017) based on the approximate likelihood is applied.

When method="Empirical", the empirical estimation procedure presented in Einmahl et al. (2013) is applied.

Regarding the plot method function:

type="returns": produces a (contour) plot of the probabilities of exceedances for some threshold. This corresponds to the output of the returns function.

type="A": produces a plot of the estimated Pickands dependence function. If A_true is specified the plot includes the true Pickands dependence function and a functional boxplot for the estimated function.

type="h": produces a plot of the estimated angular density function. If h_true is specified the plot includes the true angular density and a functional boxplot for the estimated function.

type="pm": produces a plot of the prior against the posterior for the mass at $\{0\}$.

type="k": produces a plot of the prior against the posterior for the polynomial degree k.

type="summary": when the estimation was performed in a Bayesian framework then a 2 by 2 plot with types "A", "h", "pm" and "k" is produced. Otherwise a 1 by 2 plot with types "A" and "h" is produced.

`type="Qsets"`**:** displays extreme quantile regions according to the methodology developped in Beranger et al. (2021).

Regarding the code `summary` method function:

It is obvious that the value of `burn` cannot be greater than the number of iterations in the mcmc algorithm. This can be found as `object$nsim`.

Value

Regarding the `fExtDep.np` function:

Returns lists of class `ExtDep_npBayes`, `ExtDep_npFreq` or `ExtDep_npEmp` depending on the value of the `method` argument. Each list includes:

method: The argument.

type: whether it is `"maxima"` or `"rawdata"` (in the broader sense that a threshold exceedance model was taken).

If `method="Bayesian"` the list also includes:

mar.fit: The argument.

pm: The posterior sample of probability masses.

eta: The posterior sample for the coeficients of the Bernstein polynomial.

k: The posterior sample for the Bernstein polynoial order.

accepted: A binary vector indicating if the proposal was accepted.

acc.vec: A vector containing the acceptance probabilities for the dependence parameters at each iteration.

prior: A list containing `hyperparam`, `prior.pm` and `prior.k`.

nsim: The argument.

data: The argument.

In addition if the marginal parameters are estimated (`mar.fit=TRUE`):

cov1, cov2: The arguments.

accepted.mar, accepted.mar2: Binary vectors indicating if the marginal proposals were accepted.

straight.reject1, straight.reject2: Binary vectors indicating if the marginal proposals were rejected straight away as not respecting existence conditions (proposal is multivariate normal).

acc.vec1, acc.vec2: Vectors containing the acceptance probabilities for the marginal parameters at each iteration.

sig1.vec, sig2.vec: Vectors containing the values of the scaling parameter in the marginal proposal distributions.

Finally, if the argument `u` is provided, the list also contains:

threshold: A bivariate vector indicating the threshold level for both margins.

kn: The empirical estimate of the probability of being greater than the threshold.

When `method="Frequentist"`, the list includes:

hhat: An estimate of the angular density.

Hhat: An estimate of the angular measure.

p0, p1: The estimates of the probability mass at 0 and 1.

Ahat: An estimate of the Pickands dependence function.

w: A sequence of value on the bivariate unit simplex.

q: A real in $[0, 1]$ indicating the quantile associated with the threshold `u`. Takes value `NULL` if `type="maxima"`.

data: The data on the unit Frechet scale (empirical transformation) if `type="rawdata"` and `mar.fit=TRUE`. Data on the original scale otherwise.

When `method="Empirical"`, the list includes:

fi: An estimate of the angular measure.

h_hat: An estimate of the angular density.

theta_seq: A sequence of angles from 0 to $\pi/2$

data The argument.

Regarding the code `summary` method function:

The function returns a list with the following objects:

k.median, k.up, k.low: Posterior median, upper and lower bounds of the CI for the estimated Bernstein polynomial degree κ.

h.mean, h.up, h.low: Posterior mean, upper and lower bounds of the CI for the estimated angular density h.

A.mean, A.up, A.low: Posterior mean, upper and lower bounds of the CI for the estimated Pickands dependence function A.

p0.mean, p0.up, p0.low: Posterior mean, upper and lower bounds of the CI for the estimated point mass p_0.

p1.mean, p1.up, p1.low: Posterior mean, upper and lower bounds of the CI for the estimated point mass p_1.

A_post: Posterior sample for Pickands dependence function.

h_post: Posterior sample for angular density.

eta.diff_post: Posterior sample for the Bernstein polynomial coefficients (η parametrisation).

beta_post: Posterior sample for the Bernstein polynomial coefficients (β parametrisation).

p0_post, p1_post: Posterior sample for point masses p_0 and p_1.

w: A vector of values on the bivariate simplex where the angular density and Pickands dependence function were evaluated.

burn: The argument provided.

If the margins were also fitted, the list given as `object` would contain `mar1` and `mar2` and the function would also output:

mar1.mean, mar1.up, mar1.low: Posterior mean, upper and lower bounds of the CI for the estimated marginal parameter on the first component.

mar2.mean, mar2.up, mar2.low: Posterior mean, upper and lower bounds of the CI for the estimated marginal parameter on the second component.

mar1_post: Posterior sample for the estimated marginal parameter on the first component.

mar2_post: Posterior sample for the estimated marginal parameter on the second component.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

References

Beranger, B., Padoan, S. A. and Sisson, S. A. (2021). Estimation and uncertainty quantification for extreme quantile regions. *Extremes*, **24**, 349-375.

Einmahl, J. H. J., de Haan, L. and Krajina, A. (2013). Estimating extreme bivariate quantile regions. *Extremes*, **16**, 121-145.

Marcon, G., Padoan, S. A. and Antoniano-Villalobos, I. (2016). Bayesian inference for the extremal dependence. *Electronic Journal of Statistics*, **10**, 3310-3337.

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017) Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, **183**, 1-17.

See Also

dExtDep, pExtDep, rExtDep, fExtDep

Examples

```
############################################################
### Example 1 - Wind Speed and Differential of pressure ###
############################################################

data(WindSpeedGust)

years <- format(ParcayMeslay$time, format = "%Y")
attach(ParcayMeslay[which(years %in% c(2004:2013)), ])

# Marginal quantiles
WS_th <- quantile(WS, .9)
DP_th <- quantile(DP, .9)

# Standardisation to unit Frechet (requires evd package)
pars.WS <- evd::fpot(WS, WS_th, model = "pp")$estimate
pars.DP <- evd::fpot(DP, DP_th, model = "pp")$estimate

# transform the marginal distribution to common unit Frechet:
data_uf <- trans2UFrechet(cbind(WS, DP), type = "Empirical")

# compute exceedances
rdata <- rowSums(data_uf)
```

```
r0 <- quantile(rdata, probs = .90)
extdata_WSDP <- data_uf[rdata >= r0, ]

# Fit
SP_mle <- fExtDep.np(
  x = extdata_WSDP, method = "Frequentist", k0 = 10, type = "maxima"
)

# Plot
plot(x = SP_mle, type = "summary")

#####################################################
### Example 2 - Pollution levels in Milan, Italy ###
#####################################################

## Not run:

### Here we will only model the dependence structure
data(MilanPollution)

data <- Milan.winter[, c("NO2", "SO2")]
data <- as.matrix(data[complete.cases(data), ])

# Thereshold
u <- apply(
  data, 2, function(x) quantile(x, prob = 0.9, type = 3)
)

# Hyperparameters
hyperparam <- list(mu.nbinom = 6, var.nbinom = 8, a.unif = 0, b.unif = 0.2)

### Standardise data to univariate Frechet margins

f1 <- fGEV(data = data[, 1], method = "Bayesian", sig0 = 0.0001, nsim = 5e+4)
diagnostics(f1)
burn1 <- 1:30000
gev.pars1 <- apply(f1$param_post[-burn1, ], 2, mean)
sdata1 <- trans2UFrechet(data = data[, 1], pars = gev.pars1, type = "GEV")

f2 <- fGEV(data = data[, 2], method = "Bayesian", sig0 = 0.0001, nsim = 5e+4)
diagnostics(f2)
burn2 <- 1:30000
gev.pars2 <- apply(f2$param_post[-burn2, ], 2, mean)
sdata2 <- trans2UFrechet(data = data[, 2], pars = gev.pars2, type = "GEV")

sdata <- cbind(sdata1, sdata2)

### Bayesian estimation using Bernstein polynomials

pollut1 <- fExtDep.np(
  x = sdata, method = "Bayesian", u = TRUE,
  mar.fit = FALSE, k0 = 5, hyperparam = hyperparam, nsim = 5e+4
)
```

```
diagnostics(pollut1)
pollut1_sum <- summary(object = pollut1, burn = 3e+4, plot = TRUE)
pl1 <- plot(
  x = pollut1, type = "Qsets", summary.mcmc = pollut1_sum,
  mar1 = gev.pars1, mar2 = gev.pars2, P = 1 / c(600, 1200, 2400),
  dep = TRUE, data = data, xlim = c(0, 400), ylim = c(0, 400)
)

pl1b <- plot(
  x = pollut1, type = "Qsets", summary.mcmc = pollut1_sum, est.out = pl1$est.out,
  mar1 = gev.pars1, mar2 = gev.pars2, P = 1 / c(1200),
  dep = FALSE, data = data, xlim = c(0, 400), ylim = c(0, 400)
)

### Frequentist estimation using Bernstein polynomials

pollut2 <- fExtDep.np(
  x = sdata, method = "Frequentist", mar.fit = FALSE, type = "rawdata", k0 = 8
)
plot(x = pollut2, type = "summary", CEX = 1.5)

pl2 <- plot(
  x = pollut2, type = "Qsets", mar1 = gev.pars1, mar2 = gev.pars2,
  P = 1 / c(600, 1200, 2400),
  dep = TRUE, data = data, xlim = c(0, 400), ylim = c(0, 400),
  xlab = expression(NO[2]), ylab = expression(SO[2]),
  col.Qfull = c("red", "green", "blue")
)

### Frequentist estimation using EKdH estimator

pollut3 <- fExtDep.np(x = data, method = "Empirical")
plot(x = pollut3, type = "summary", CEX = 1.5)

pl3 <- plot(
  x = pollut3, type = "Qsets", mar1 = gev.pars1, mar2 = gev.pars2,
  P = 1 / c(600, 1200, 2400),
  dep = TRUE, data = data, xlim = c(0, 400), ylim = c(0, 400),
  xlab = expression(NO[2]), ylab = expression(SO[2]),
  col.Qfull = c("red", "green", "blue")
)


## End(Not run)
```

---

fExtDepSpat *Fitting of a max-stable process*

Description

This function uses the Stephenson-Tawn likelihood to estimate parameters of max-stable models.

Usage

```
fExtDepSpat(
  x, model, sites, hit, jw, thresh, DoF, range, smooth,
  alpha, par0, acov1, acov2, parallel, ncores, args1, args2,
  seed = 123, method = "BFGS", sandwich = TRUE,
  control = list(trace = 1, maxit = 50, REPORT = 1, reltol = 0.0001)
)

## S3 method for class 'ExtDep_Spat'
logLik(object, ...)
```

Arguments

x	A $(n \times d)$ matrix containing n observations at d locations (fExtDepSpat).
object	An object of class ExtDep_Spat (logLik).
model	A character string indicating the max-stable model, currently extremal-t ("ET") and extremal skew-t ("EST") only available. Note that the Schlather model can be obtained as a special case by specifying the extremal-t model with DoF = 1.
sites	A $(d \times 2)$ matrix corresponding to the coordinates of locations where the processes are simulated. Each row corresponds to a location.
hit	A $(n \times d)$ matrix containing the hitting scenarios for each observation.
jw	An integer between 2 and d, indicating the tuples considered in the composite likelihood. If jw = d the full likelihood is considered.
thresh	A positive real indicating the threshold value for pairwise distances. See details.
DoF	A positive real indicating a fixed value of the degree of freedom of the extremal-t and extremal skew-t models.
range	A positive real indicating a fixed value of the range parameter for the power exponential correlation function (only correlation function currently available).
smooth	A positive real in $(0, 2]$ indicating a fixed value of the smoothness parameter for the power exponential correlation function (only correlation function currently available).
alpha	A vector of length 3 indicating fixed values of the skewness parameters α_0, α_1, and α_2 for the extremal skew-t model. If some components are NA, then the corresponding parameter will be estimated.
par0	A vector of initial values of the parameter vector, in order: the degree of freedom ν, the range r, the smoothness η, and the skewness parameters α_0, α_1. Its length depends on the model and the number of fixed parameters.
acov1, acov2	Vectors of length d representing covariates to model the skewness parameter of the extremal skew-t model.
parallel	A logical value; if TRUE parallel computing is done.
ncores	An integer indicating the number of cores considered in the parallel socket cluster of type 'PSOCK', based on the makeCluster routine from the parallel package. Required if parallel = TRUE.

args1, args2	Lists specifying details about the Monte Carlo simulation scheme to compute multivariate CDFs. See details.
seed	An integer for reproducibility in the CDF computations.
method	A character string indicating the optimisation method to be used. See optim for more details.
sandwich	A logical value; if TRUE the standard errors of the estimates are computed from the Sandwich (Godambe) information matrix.
control	A list of control parameters for the optimisation. See optim for more details.
...	For the logLik() function: can provide a digits argument, an integer indicating the number of decimal places to be reported.

Details

This routine follows the methodology developed by Beranger et al. (2021). It uses the Stephenson-Tawn likelihood which relies on the knowledge of time occurrences of each block maxima. Rather than considering all partitions of the set $\{1, \ldots, d\}$, the likelihood is computed using the observed partition. Let $\Pi = (\pi_1, \ldots, \pi_K)$ denote the observed partition, then the Stephenson-Tawn likelihood is given by

$$L(\theta; x) = \exp\{-V(x;\theta)\} \times \prod_{k=1}^{K} -V_{\pi_k}(x;\theta),$$

where V_π represents the partial derivative(s) of $V(x;\theta)$ with respect to π.

When jw = d the full Stephenson-Tawn likelihood is considered, whereas for values lower than d a composite likelihood approach is taken.

The argument thresh is required when the composite likelihood is used. A tuple of size jw is assigned a weight of one if the maximum pairwise distance between corresponding locations is less than thresh, and a weight of zero otherwise.

Arguments args1 and args2 relate to specifications of the Monte Carlo simulation scheme to compute multivariate CDF evaluations. These should take the form of lists including the minimum and maximum number of simulations used (Nmin and Nmax), the absolute error (eps), and whether the error should be controlled on the log-scale (logeps).

Value

fExtDepSpat: An object of class ExtDep_Spat in the form of a list comprising:

- the vector of estimated parameters (est),
- the composite likelihood order (jw),
- the maximised log-likelihood value (LL),
- if sandwich = TRUE, the standard errors from the sandwich information matrix (stderr.sand),
- the TIC for model selection (TIC),
- if the composite likelihood is considered, a matrix of all tuples with weight 1 (cmat).

logLik: The value of the maximised log-likelihood.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com

References

Beranger, B., Stephenson, A. G., and Sisson, S. A. (2021). High-dimensional inference using the extremal skew-t process. *Extremes*, **24**, 653–685.

See Also

fExtDepSpat

Examples

```
set.seed(14342)

# Simulation of 20 locations
Ns <- 20
sites <- matrix(runif(Ns * 2) * 10 - 5, nrow = Ns, ncol = 2)
for (i in 1:2) sites[, i] <- sites[, i] - mean(sites[, i])

# Simulation of 50 replicates from the Extremal-t model
Ny <- 50
x <- rExtDepSpat(
  Ny, sites, model = "ET", cov.mod = "powexp", DoF = 1,
  range = 3, nugget = 0, smooth = 1.5,
  control = list(method = "exact")
)

# Fit the extremal-t using the full Stephenson-Tawn likelihood
args1 <- list(Nmax = 50L, Nmin = 5L, eps = 0.001, logeps = FALSE)
args2 <- list(Nmax = 500L, Nmin = 50L, eps = 0.001, logeps = TRUE)

## Not run:
fit1 <- fExtDepSpat(
  x = x$vals, model = "ET", sites = sites, hit = x$hits,
  par0 = c(3, 1, 1), parallel = TRUE, ncores = 6,
  args1 = args1, args2 = args2, control = list(trace = 0)
)
stderr(fit1)

## End(Not run)
```

Description

Maximum-likelihood and Metropolis-Hastings algorithm for the estimation of the generalized extreme value distribution.

Usage

```
fGEV(data, par.start, method="Frequentist", u, cov,
     optim.method="BFGS", optim.trace=0, sig0, nsim)
```

Arguments

`data`	A vector representing the data, which may contain missing values.
`par.start`	A vector of length 3 giving the starting values for the parameters to be estimated. If missing, moment estimates are used.
`method`	A character string indicating whether the estimation is done following a `"Frequentist"` or `"Bayesian"` approach.
`u`	A real indicating a high threshold. If supplied a threshold exceedance approach is taken and computations use the censored likelihood. If missing, a block maxima approach is taken and the regular GEV likelihood is used.
`cov`	A matrix of covariates to define a linear model for the location parameter.
`optim.method`	The optimization method to be used. Required when `method="Frequentist"`. See `optim` for more details.
`optim.trace`	A non-negative integer tracing the progress of the optimization. Required when `method="Frequentist"`. See `optim` for more details.
`sig0`	Positive reals representing the initial value for the scaling parameter of the multivariate normal proposal distribution for both margins. Required when `method="Bayesian"`.
`nsim`	An integer indicating the number of iterations in the Metropolis-Hastings algorithm. Required when `method="Bayesian"`.

Details

When `cov` is a vector of ones then the location parameter μ is constant. On the contrary, when `cov` is provided, it represents the design matrix for the linear model on μ (the number of columns in the matrix indicates the number of linear predictors).

When `u=NULL` or missing, the likelihood function is given by

$$\prod_{i=1}^{n} g(x_i; \mu, \sigma, \xi)$$

where $g(\cdot; \mu, \sigma, \xi)$ represents the GEV pdf, whereas when a threshold value is set the likelihood is given by

$$k_n \log\left(G(u; \mu, \sigma, \xi)\right) \times \prod_{i=1}^{n} \frac{\partial}{\partial x} G(x; \mu, \sigma, \xi)|_{x=x_i}$$

where $G(\cdot; \mu, \sigma, \xi)$ is the GEV cdf and k_n is the empirical estimate of the probability of being greater than the threshold `u`.

Note that the case $\xi \leq 0$ is not yet considered when u is used.

The choice `method="Bayesian"` applies a random walk Metropolis-Hastings algorithm as described in Section 3.1 and Step 1 and 2 of Algorithm 1 from Beranger et al. (2021). The algorithm may restart for several reasons including if the proposed value of the parameters changes too much from the current value (see Garthwaite et al. (2016) for more details).

The choice `method="Frequentist"` uses the `optim` function to find the maximum likelihood estimator.

Value

- When `method="Frequentist"` the routine returns a list including the parameter estimates (`est`), associated variance-covariance matrix (`varcov`), and standard errors (`stderr`).
- When `method="Bayesian"` the routine returns a list including:

  param_post the parameter posterior sample;

  accepted a binary vector indicating which proposals were accepted;

  straight.reject a binary vector indicating which proposals were rejected immediately, given that the proposal is multivariate normal and there are constraints on the parameter values;

  nsim the number of simulations in the algorithm;

  sig.vec the vector of updated scaling parameters in the multivariate normal proposal distribution at each iteration;

  sig.restart the value of the scaling parameter in the multivariate normal proposal distribution when the algorithm needs to restart;

  acc.vec a vector of acceptance probabilities at each iteration.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

References

Beranger, B., Padoan, S. A., and Sisson, S. A. (2021). Estimation and uncertainty quantification for extreme quantile regions. *Extremes*, **24**, 349–375.

Garthwaite, P. H., Fan, Y., and Sisson, S. A. (2016). Adaptive optimal scaling of Metropolis-Hastings algorithms using the Robbins-Monro process. *Communications in Statistics - Theory and Methods*, **45**(17), 5098–5111.

See Also

`dGEV`

Examples

```
################################################
### Example - Pollution levels in Milan, Italy ###
################################################
```

```
data(MilanPollution)

# Frequentist estimation
fit <- fGEV(Milan.winter$PM10)
fit$est

# Bayesian estimation with high threshold
cov <- cbind(rep(1, nrow(Milan.winter)), Milan.winter$MaxTemp,
             Milan.winter$MaxTemp^2)
u <- quantile(Milan.winter$PM10, prob=0.9, type=3, na.rm=TRUE)


fit2 <- fGEV(data=Milan.winter$PM10, par.start=c(50,0,0,20,1),
             method="Bayesian", u=u, cov=cov, sig0=0.1, nsim=5e+4)
```

---

heat *Summer temperature maxima in Melbourne, Australia between 1961 and 2010*

Description

The dataset corresponds to summer temperature maxima taken over the period from August to April inclusive, recorded between 1961 and 2010 at 90 stations arranged on a 0.15 degree grid in a 9 by 10 formation.

Details

The first maximum is taken over the August 1961 to April 1962 period, and the last maximum is taken over the August 2010 to April 2011 period.

The object `heatdata` contains the core of the data:

vals A 50×90 matrix containing the 50 summer maxima at the 90 locations.

sitesLL A 90×2 matrix containing the site locations in latitude-longitude, recentered (means subtracted).

sitesEN A 90×2 matrix containing the site locations in eastings-northings, recentered (means subtracted).

hits A 50×90 integer matrix indicating the "heatwave" number associated with each summer maximum. Locations on the same row with the same integer indicate maxima arising from the same heatwave, defined over a three-day window.

sitesLLO A 90×2 matrix containing the site locations in latitude-longitude, on the original scale.

sitesENO A 90×2 matrix containing the site locations in eastings-northings, on the original scale.

ufvals A 50×90 matrix containing the 50 summer maxima at the 90 locations, standardized to the unit Frechet scale.

Standardisation to the unit Frechet scale is performed as in Beranger et al. (2021) by fitting the GEV distribution marginally, using unconstrained location and scale parameters and a shape parameter specified as a linear function of eastings and northings (in 100 km units). The resulting estimates are stored in the objects `locgrid`, `scalegrid`, and `shapegrid`, which are 10×9 matrices.

Details about the study region are given in `mellat` and `mellon`, vectors of length 10 and 11, which provide the latitude and longitude coordinates of the grid.

References

Beranger, B., Stephenson, A. G. and Sisson, S. A. (2021). High-dimensional inference using the extremal skew-t process. *Extremes*, **24**, 653-685.

Examples

```
image(x = mellon, y = mellat, z = locgrid)
points(heatdata$sitesLLO, pch = 16)
```

`index.ExtDep` *Index of extremal dependence*

Description

Computes the extremal coefficient, Pickands dependence function, and the coefficients of upper and lower tail dependence for several parametric models. Also computes the lower tail dependence function for the bivariate skew-normal distribution.

Usage

```
index.ExtDep(object, model, par, x, u)
```

Arguments

`object`	A character string indicating the index of extremal dependence to compute. Options are: • `"extremal"`: extremal coefficient, • `"pickands"`: Pickands dependence function, • `"upper.tail"`: coefficient of upper tail dependence, • `"lower.tail"`: coefficient of lower tail dependence.
`model`	A character string indicating the model/distribution. • For `object="extremal"`, `"pickands"`, or `"upper.tail"`: Husler-Reiss (`"HR"`), extremal-t (`"ET"`), extremal skew-t (`"EST"`). • For `object="lower.tail"`: extremal-t (`"ET"`), extremal skew-t (`"EST"`), or skew-normal (`"SN"`).
`par`	A vector of parameter values for the specified model/distribution.
`x`	A vector on the bivariate or trivariate unit simplex indicating where to evaluate the Pickands dependence function.
`u`	A real number in $[0, 1]$ indicating the value at which to evaluate the lower tail dependence function of the bivariate skew-normal distribution.

Details

The extremal coefficient is defined as

$$\theta = V(1,\ldots,1) = d\int_W \max_{j\in\{1,\ldots,d\}}(w_j)dH(w) = -\log G(1,\ldots,1),$$

where W is the unit simplex, V is the exponent function, and $G(\cdot)$ the distribution function of a multivariate extreme value model. Bivariate and trivariate versions are available.

The Pickands dependence function is defined as

$$A(x) = -\log G(1/x)$$

for x in the bivariate/trivariate simplex W.

The coefficient of upper tail dependence is defined as

$$\vartheta = R(1,\ldots,1) = d\int_W \min_{j\in\{1,\ldots,d\}}(w_j)dH(w).$$

In the bivariate case, using the inclusion-exclusion principle this reduces to

$$\vartheta = 2 + \log G(1,1) = 2 - V(1,1).$$

For the skew-normal distribution, the lower tail dependence function is defined as in Bortot (2010). This approximation is obtained in the limiting case as `u` tends to 1. The `par` argument should be a vector of length 3, consisting of the correlation parameter (between -1 and 1) and two real-valued skewness parameters.

Value

- `object="extremal"`: returns a value in $[1, d]$ ($d = 2, 3$).
- `object="pickands"`: returns a value in $[\max(x), 1]$.
- `object="upper.tail"`: returns a value in $[0, 1]$.
- `object="lower.tail"`: returns a value in $[-1, 1]$.

Author(s)

Simone Padoan <simone.padoan@unibocconi.it> https://faculty.unibocconi.it/simonepadoan/
Boris Beranger <borisberanger@gmail.com> https://www.borisberanger.com

References

Bortot, P. (2010). Tail dependence in bivariate skew-normal and skew-t distributions. *Unpublished*.

Examples

```
############################
### Extremal skew-t model ###
############################

## Extremal coefficient
```

```
index.ExtDep(object = "extremal", model = "EST", par = c(0.5, 1, -2, 2))

## Pickands dependence function
w <- seq(0.00001, 0.99999, length = 100)
pick <- numeric(100)
for (i in 1:100) {
  pick[i] <- index.ExtDep(
    object = "pickands", model = "EST", par = c(0.5, 1, -2, 2),
    x = c(w[i], 1 - w[i])
  )
}

plot(w, pick, type = "l", ylim = c(0.5, 1), ylab = "A(t)", xlab = "t")
polygon(c(0, 0.5, 1), c(1, 0.5, 1), lwd = 2, border = "grey")

## Upper tail dependence coefficient
index.ExtDep(object = "upper.tail", model = "EST", par = c(0.5, 1, -2, 2))

## Lower tail dependence coefficient
index.ExtDep(object = "lower.tail", model = "EST", par = c(0.5, 1, -2, 2))

################################
### Skew-normal distribution ###
################################

## Lower tail dependence function
index.ExtDep(object = "lower.tail", model = "SN", par = c(0.5, 1, -2), u = 0.5)
```

`lambda.HR` *Valid set of parameters for the 3-dimensional Husler-Reiss model*

Description

Given two parameters of the Husler-Reiss model, this function evaluates the range of values the third parameter can take to ensure a positive definite matrix in the model.

Usage

```
lambda.HR(lambda)
```

Arguments

`lambda`	A vector of length 3 containing one `NA` and two positive values. The `NA` corresponds to the parameter for which the range is calculated.

Details

As indicated in Engelke et al. (2015), the matrix with zero diagonal and squared `lambda` parameters on the off-diagonal needs to be strictly conditionally negative definite.

Value

A 2×3 matrix with, on the top, the lowest value of the parameter corresponding to the NA value in the input, and on the bottom the largest value.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com.

References

Engelke, S., Malinowski, A., Kabluchko, Z., and Schlather, M. (2015). Estimation of Husler-Reiss distributions and Brown-Resnick processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, **77**, 239–265.

Examples

```
ls <- lambda.HR(c(1, 2, NA))
dExtDep(
  x = c(0.1, 0.7, 0.2),
  method = "Parametric",
  model = "HR",
  par = ls[1, ],
  angular = TRUE
)
```

---

logReturns *Monthly maxima of log-return exchange rates of the Pound Sterling (GBP) against the US dollar (USD) and the Japanese yen (JPY), between March 1991 and December 2014*

Description

The dataset logReturns contains four columns: date_USD and USD give the date and value of the monthly maxima of the log-return exchange rate GBP/USD, while date_JPY and JPY give the date and value of the monthly maxima of the log-return exchange rate GBP/JPY.

Format

A 286×4 matrix. The first and third columns are of type "character", while the second and fourth columns are of type "numeric".

`madogram` *Madogram-based estimation of the Pickands Dependence Function*

Description

Computes a non-parametric estimate of the Pickands dependence function $A(w)$ for multivariate data, based on the madogram estimator.

Usage

```
madogram(w, data, margin = c("emp", "est", "exp", "frechet", "gumbel"))
```

Arguments

`w`	An $m \times d$ design matrix (see **Details**).
`data`	An $n \times d$ matrix of data or data frame with `d` columns. Here, `d` is the number of variables and `n` the number of replications.
`margin`	A string indicating the type of marginal distributions (`"emp"` by default, see **Details**).

Details

The estimation procedure is based on the madogram as proposed in Marcon et al. (2017). The madogram is defined by

$$\nu(\mathbf{w}) = \mathbb{E}\left(\max_{i=1,\ldots,d}\left\{F_i^{1/w_i}(X_i)\right\} - \frac{1}{d}\sum_{i=1}^{d} F_i^{1/w_i}(X_i)\right),$$

where $0 < w_i < 1$ and $w_d = 1 - (w_1 + \ldots + w_{d-1})$.

Each row of the design matrix `w` is a point in the d-dimensional unit simplex.

If X is a d-dimensional max-stable random vector with exponent measure $V(\mathbf{x})$ and Pickands dependence function $A(\mathbf{w})$, then

$$\nu(\mathbf{w}) = \frac{V(1/w_1, \ldots, 1/w_d)}{1 + V(1/w_1, \ldots, 1/w_d)} - c(\mathbf{w}),$$

where

$$c(\mathbf{w}) = \frac{1}{d}\sum_{i=1}^{d}\frac{w_i}{1+w_i}.$$

From this, it follows that

$$V(1/w_1, \ldots, 1/w_d) = \frac{\nu(\mathbf{w}) + c(\mathbf{w})}{1 - \nu(\mathbf{w}) - c(\mathbf{w})},$$

and

$$A(\mathbf{w}) = \frac{\nu(\mathbf{w}) + c(\mathbf{w})}{1 - \nu(\mathbf{w}) - c(\mathbf{w})}.$$

Marginal treatment:

- `"emp"`: empirical transformation of the marginals,
- `"est"`: maximum-likelihood fitting of the GEV distributions,
- `"exp"`, `"frechet"`, `"gumbel"`: parametric GEV theoretical distributions.

Value

A numeric vector of estimates of the Pickands dependence function.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com; Giulia Marcon, <giuliamarcongm@gmail.com>

References

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017). Multivariate Nonparametric Estimation of the Pickands Dependence Function using Bernstein Polynomials. *Journal of Statistical Planning and Inference*, **183**, 1–17.

Naveau, P., Guillou, A., Cooley, D. and Diebolt, J. (2009). Modelling pairwise dependence of maxima in space. *Biometrika*, **96**(1), 1–17.

See Also

`beed`, `beed.confband`

Examples

```
x <- simplex(2)
data <- evd::rbvevd(50, dep = 0.4, model = "log", mar1 = c(1,1,1))

Amd <- madogram(x, data, "emp")
Amd.bp <- beed(data, x, 2, "md", "emp", 20, plot = TRUE)

lines(x[,1], Amd, lty = 1, col = 2)
```

`method` *Extract the method attribute*

Description

This function extracts the method name from a fitted object.

Usage

```
method(x)
```

Arguments

`x`	An object of class `ExtDep_Bayes`, `ExtDep_Freq` or `ExtDep_Spat`.

Value

A character string.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com> https://www.borisberanger.com;

See Also

`fExtDepSpat`, `fExtDep`

Examples

```
data(pollution)
f.hr <- fExtDep(
  x = PNS,
  method = "PPP",
  model = "HR",
  par.start = rep(0.5, 3),
  trace = 2
)
method(f.hr)
```

MilanPollution *Pollution data for summer and winter months in Milan, Italy*

Description

Two datasets, `Milan.summer` and `Milan.winter`, each containing 5 air pollutants (daily maximum of NO2, NO, O3 and SO2, daily mean of PM10) and 6 meteorological covariates (maximum precipitation, maximum temperature, maximum humidity, mean precipitation, mean temperature and mean humidity).

Format

A 1968×12 data frame and a 1924×12 data frame.

Details

The summer period corresponds to 30 April-30 August between 2003 and 2017, giving 1968 observations. The winter period corresponds to 1 November-27(28) February. The records start from 31 December 2001 until 30 December 2017, giving 1924 observations.

model *Extract the model attribute*

Description

Extracts the model name from a fitted object.

Usage

```
model(x)
```

Arguments

`x`	An object of class `ExtDep_Bayes`, `ExtDep_Freq`, or `ExtDep_Spat`.

Value

A character string.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com

See Also

`fExtDepSpat`, `fExtDep`

Examples

```
data(pollution)
f.hr <- fExtDep(
  x = PNS, method = "PPP", model = "HR",
  par.start = rep(0.5, 3), trace = 2
)
model(f.hr)
```

PAMfmado *Clustering of maxima*

Description

Performs clustering of time series of maxima using the pam algorithm tailored for the F-madogram distance.

Usage

```
PAMfmado(x, K, J = 0, threshold = 0.99, max.min = 0)
```

Arguments

x	A matrix of maxima. For example, for weekly maxima of precipitation, the number of stations is ncol(x) and the time series length is nrow(x).
K	Number of clusters.
J	Number of resamplings for which the stations are randomly moved to break the dependence. By default, J = 0 means no resampling.
threshold	Quantile level used for the resampling threshold. The corresponding quantile is printed (when J is not 0).
max.min	A lower threshold to remove very small values. For example, some rain gauges cannot go below 2 mm. Default is 0.

Value

An object of class "pam" representing the clustering. See pam.object for details.

Author(s)

Philippe Naveau

References

Bernard, E., Naveau, P., Vrac, M. and Mestre, O. (2013). Clustering of maxima: Spatial dependencies among heavy rainfall in France. *Journal of Climate* **26**, 7929–7937.

Naveau, P., Guillou, A., Cooley, D. and Diebolt, J. (2009). Modeling pairwise dependence of maxima in space. *Biometrika* **96**(1).

Cooley, D., Naveau, P. and Poncet, P. (2006). Variograms for spatial max-stable random fields. In: Bertail, P., Soulier, P., Doukhan, P. (eds) *Dependence in Probability and Statistics. Lecture Notes in Statistics*, vol 187. Springer, New York, NY.

Reynolds, A., Richards, G., de la Iglesia, B. and Rayward-Smith, V. (1992). Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms* **5**, 475–504.

See Also

pam

Examples

```
data(PrecipFrance)
PAMmado <- PAMfmado(PrecipFrance$precip, 7)
```

`pExtDep` *Parametric and Non-Parametric Distribution Function of Extremal Dependence*

Description

Evaluate the distribution function of parametric multivariate extreme-value distributions and the angular probability distribution represented through Bernstein polynomials.

Usage

```
pExtDep(q, type, method = "Parametric", model, par, plot = TRUE,
        main, xlab, cex.lab, cex.axis, lwd, ...)
```

Arguments

`q`	A vector or matrix of quantiles.
`type`	A character string: `"lower"`, `"inv.lower"` or `"upper"`. Required when `method = "Parametric"`.
`method`	A character string: `"Parametric"` or `"NonParametric"`.
`model`	A character string with the model name: `"HR"` (Husler-Reiss), `"ET"` (Extremal-t), or `"EST"` (Extremal Skew-t). Required when `method = "Parametric"`.
`par`	A vector or matrix of parameters for the model. If a matrix, rows correspond to different parameter sets.

`plot` Logical; if `TRUE` (default), a plot is displayed. See **Details**.

`main`, `xlab`, `cex.lab`, `cex.axis`, `lwd`
Graphical arguments passed to `hist()`.

`...` Additional graphical arguments passed to `hist()` when `plot = TRUE`.

Details

When `method = "Parametric"`, the distribution function is available in 2 or 3 dimensions only. See `dim_ExtDep` for the correct length of the parameter vector.

- If `type = "lower"`, the cumulative distribution function is computed:

$$G(x) = P(X \leq x), \quad x \in \mathbb{R}^d,\ d = 2, 3.$$

- If `type = "inv.lower"`, the survival function is computed:

$$1 - G(x) = P(\exists i : X_i > x_i).$$

- If `type = "upper"`, the joint probability of exceedance is computed:

$$P(X \geq x).$$

When `method = "NonParametric"`, the distribution function is available in 2 dimensions only.

If `par` is a matrix and `plot = TRUE`, a histogram of the probabilities is displayed across parameter sets. A kernel density estimator, $2.5\%, 50\%, 97.5\%$ quantiles (crosses) and the mean (dot) are added.

Value

- If `par` is a vector: returns a scalar (if `q` is a vector) or a vector of length `nrow(q)` (if `q` is a matrix).
- If `par` is a matrix: returns a vector of length `nrow(par)` (if `q` is a vector) or a matrix with `nrow(par)` rows and `nrow(q)` columns (if `q` is a matrix).

Author(s)

Simone Padoan <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/
Boris Beranger <borisberanger@gmail.com>, https://www.borisberanger.com

References

Beranger, B. and Padoan, S.A. (2015). *Extreme Value Modeling and Risk Analysis: Methods and Applications*. Chapman & Hall/CRC.

Beranger, B., Padoan, S.A. and Sisson, S.A. (2017). Models for extremal dependence derived from skew-symmetric families. *Scandinavian Journal of Statistics*, **44**(1), 21–45.

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

Marcon, G., Padoan, S.A., Naveau, P., Muliere, P. and Segers, J. (2017). Multivariate nonparametric estimation of the Pickands dependence function using Bernstein polynomials. *Journal of Statistical Planning and Inference*, **183**, 1–17.

See Also

dExtDep, rExtDep, fExtDep, fExtDep.np

Examples

```
# Trivariate Extremal Skew-t
pExtDep(q = c(1, 1.2, 0.6), type = "lower", method = "Parametric",
        model = "EST", par = c(0.2, 0.4, 0.6, 2, 2, 2, 1))

# Bivariate Extremal-t
pExtDep(q = rbind(c(1.2, 0.6), c(1.1, 1.3)), type = "inv.lower",
        method = "Parametric", model = "ET", par = c(0.2, 1))

# Bivariate Extremal Skew-t
pExtDep(q = rbind(c(1.2, 0.6), c(1.1, 1.3)), type = "inv.lower",
        method = "Parametric", model = "EST", par = c(0.2, 0, 0, 1))

# Non-parametric angular density
beta <- c(1.0000000, 0.8714286, 0.7671560, 0.7569398,
          0.7771908, 0.8031573, 0.8857143, 1.0000000)
pExtDep(q = rbind(c(0.1, 0.9), c(0.2, 0.8)),
        method = "NonParametric", par = beta)
```

---

pFailure *Probability of Falling into a Failure Region*

Description

Compute the empirical probability of falling into two types of failure regions, based on exceedances simulated from a fitted extreme-value model.

Usage

```
pFailure(n, beta, u1, u2, mar1, mar2, type, plot, xlab, ylab, nlevels = 10)
```

Arguments

n	Integer. Number of points generated to compute the empirical probability.
beta	Numeric vector. Bernstein polynomial coefficients.
u1, u2	Numeric vectors of positive thresholds.

mar1, mar2	Numeric vectors. Marginal GEV parameters.
type	Character. Type of failure region: • "or" - at least one exceedance, • "and" - both exceedances, • "both" - compute both probabilities.
plot	Logical. If TRUE, display contour plots of the probabilities.
xlab, ylab	Character strings for axis labels in plots.
nlevels	Integer. Number of contour levels for plots.

Details

The two failure regions are:

$$A_u = \{(v_1, v_2) : v_1 > u_1 \text{ or } v_2 > u_2\}$$

and

$$B_u = \{(v_1, v_2) : v_1 > u_1 \text{ and } v_2 > u_2\}$$

for $(v_1, v_2) \in \mathbb{R}^2_+$, with thresholds $u_1, u_2 > 0$.

Exceedance samples are generated following Algorithm 3 of Marcon et al. (2017). The empirical estimates are

$$\hat{P}_{A_u} = \frac{1}{n} \sum_{i=1}^{n} I(y_{i1} > u_1^* \text{ or } y_{i2} > u_2^*)$$

and

$$\hat{P}_{B_u} = \frac{1}{n} \sum_{i=1}^{n} I(y_{i1} > u_1^* \text{ and } y_{i2} > u_2^*)$$

.

Value

A list with elements AND and/or OR, depending on type. Each element is a matrix of size length(u1) x length(u2).

Author(s)

Simone Padoan <simone.padoan@unibocconi.it> (https://faculty.unibocconi.it/simonepadoan/) \ Boris Beranger <borisberanger@gmail.com> (https://www.borisberanger.com)

References

Marcon, G., Naveau, P. and Padoan, S.A. (2017). A semi-parametric stochastic generator for bivariate extreme events. *Stat*, **6**, 184–201.

See Also

dExtDep, rExtDep, fExtDep, fExtDep.np

Examples

```
# Example: Wind speed and gust data
data(WindSpeedGust)
years <- format(ParcayMeslay$time, format = "%Y")
attach(ParcayMeslay[years %in% 2004:2013, ])

WS_th <- quantile(WS, .9)
DP_th <- quantile(DP, .9)

pars.WS <- evd::fpot(WS, WS_th, model = "pp")$estimate
pars.DP <- evd::fpot(DP, DP_th, model = "pp")$estimate

data_uf <- trans2UFrechet(cbind(WS, DP), type = "Empirical")
rdata <- rowSums(data_uf)
r0 <- quantile(rdata, probs = .90)
extdata <- data_uf[rdata >= r0, ]

SP_mle <- fExtDep.np(x = extdata, method = "Frequentist", k0 = 10, type = "maxima")


pF <- pFailure(
  n = 50000, beta = SP_mle$Ahat$beta,
  u1 = seq(19, 28, length = 200), mar1 = pars.WS,
  u2 = seq(40, 60, length = 200), mar2 = pars.DP,
  type = "both", plot = TRUE,
  xlab = "Daily-maximum Wind Speed (m/s)",
  ylab = "Differential of Pressure (mbar)",
  nlevels = 15
)
```

`pickands.plot` *Plot for the Pickands dependence function*

Description

This function displays the Pickands dependence function for bivariate and trivariate extreme values models.

Usage

```
pickands.plot(model, par, labels, cex.lab, contour, cex.cont)
```

Arguments

`model`	A string with the name of the model considered. Takes value `"HR"`, `"ET"` or `"EST"`.
`par`	A vector representing the parameters of the model.

labels	A vector of character strings indicating the labels. Must be of length 1 for bivariate models and 3 for trivariate models.
cex.lab	A positive real indicating the size of the labels.
contour	A logical value; if TRUE the contour levels are displayed. Required for trivariate models only.
cex.cont	A positive real indicating the size of the contour labels.

Details

The Pickands dependence function is computed using the function index.ExtDep with argument object="pickands". When contours are displayed, levels are chosen to be the deciles.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com

See Also

dExtDep

Examples

```
pickands.plot(model="HR", par=0.6)
## Not run:
pickands.plot(model="ET", par=c(0.6, 0.2, 0.5, 2))

## End(Not run)
```

---

pollution *Air quality datasets recorded in Leeds (U.K.)*

Description

Air quality datasets containing daily maxima of five air pollutants (PM10, NO, NO2, O3 and SO2) recorded in Leeds, U.K., during five winter seasons (November–February) between 1994 and 1998. Six derived datasets are included: PNS, PNN, NSN, PNNS, winterdat and Leeds.frechet.

Details

The dataset winterdat contains 590 transformed observations for each of the five pollutants. Contains NAs. Outliers have been removed according to Heffernan and Tawn (2004).

The following datasets have been obtained by applying transformations to winterdat:

- Leeds.frechet: 590 observations corresponding to the daily maxima of five air pollutants transformed to unit Frechet scale.

- NSN: 100 observations in the 3-dimensional unit simplex for the daily maxima of nitrogen dioxide (NO2), sulfur dioxide (SO2) and nitrogen oxide (NO).
- PNN: 100 observations in the 3-dimensional unit simplex for the daily maxima of particulate matter (PM10), nitrogen oxide (NO) and nitrogen dioxide (NO2).
- PNS: 100 observations in the 3-dimensional unit simplex for the daily maxima of particulate matter (PM10), nitrogen oxide (NO) and sulfur dioxide (SO2).
- PNNS: 100 observations in the 4-dimensional unit simplex for the daily maxima of particulate matter (PM10), nitrogen oxide (NO), nitrogen dioxide (NO2) and sulfur dioxide (SO2).

The transformation to unit Frechet margins of the raw data was considered by Cooley et al. (2010). Only the 100 data points with the largest radial components were kept.

Source

https://uk-air.defra.gov.uk/

References

Cooley, D., Davis, R. A., and Naveau, P. (2010). The pairwise beta distribution: a flexible parametric multivariate model for extremes. *Journal of Multivariate Analysis*, **101**, 2103–2117.

Heffernan, J. E., and Tawn, J. A. (2004). A conditional approach for multivariate extreme values. *Journal of the Royal Statistical Society: Series B (Methodology)*, **66**, 497–546.

PrecipFrance *Weekly maxima of hourly rainfall in France*

Description

A list containing the weekly maxima of hourly rainfall during the fall season from 1993 to 2011, recorded at 92 stations across France (precip). Coordinates of the monitoring stations are provided in lat and lon.

Format

A list containing:

- precip: a 228×92 matrix of weekly maxima of hourly rainfall.
- lat: a numeric vector of length 92 giving the station latitudes.
- lon: a numeric vector of length 92 giving the station longitudes.

Details

The fall season corresponds to the September–November (SON) period. The data cover a 12-week period over 19 years, yielding a sample of 228 observations (rows) and 92 stations (columns).

`returns` *Compute return values*

Description

Predicts the probability of future simultaneous exceedances.

Usage

```
returns(x, summary.mcmc, y, plot = FALSE, data = NULL, ...)
```

Arguments

`x`	An object of class `ExtDep_npBayes`, the output of the `fExtDep.np` function with `method = "Bayesian"`.
`summary.mcmc`	The output of the `summary.ExtDep_npBayes` function.
`y`	A 2-column matrix of unobserved thresholds.
`plot`	Logical. If `TRUE`, then the returns are plotted using `plot.ExtDep_npBayes`.
`data`	As in `plot.ExtDep_npBayes`. Required if `plot = TRUE`.
`...`	Additional graphical parameters when `plot = TRUE`. See `plot.ExtDep_npBayes`.

Details

Computes for a range of unobserved extremes (larger than those observed in a sample) the pointwise mean from the posterior predictive distribution of such predictive values. The probabilities are calculated through

$$P(Y_1 > y_1, Y_2 > y_2) = \frac{2}{k}\sum_{j=0}^{k-2}(\eta_{j+1}-\eta_j)\times\left(\frac{(j+1)B\left(\frac{y_1}{y_1+y_2} \mid j+2, k-j-1\right)}{y_1} - \frac{(k-j-1)B\left(\frac{y_2}{y_1+y_2} \mid k-j, j+1\right)}{y_2}\right.$$

where $B(x \mid a, b)$ denotes the cumulative distribution function of a Beta random variable with shape parameters $a, b > 0$. See Marcon et al. (2016, p. 3323) for details.

Value

A numeric vector whose length equals the number of rows of the input `y`.

Author(s)

Simone Padoan <simone.padoan@unibocconi.it> https://faculty.unibocconi.it/simonepadoan/; Boris Beranger <borisberanger@gmail.com> https://www.borisberanger.com; Giulia Marcon <giuliamarcongm@gmail.com>

References

Marcon, G., Padoan, S. A. and Antoniano-Villalobos, I. (2016). Bayesian inference for the extremal dependence. *Electronic Journal of Statistics*, **10**, 3310–3337.

Examples

```
############################################################
### Example 1 - daily log-returns between the GBP/USD ###
###             and GBP/JPY exchange rates            ###
############################################################

if(interactive()){

  data(logReturns)

  mm_gbp_usd <- ts(logReturns$USD, start = c(1991, 3), end = c(2014, 12), frequency = 12)
  mm_gbp_jpy <- ts(logReturns$JPY, start = c(1991, 3), end = c(2014, 12), frequency = 12)

  # Detect seasonality and trend in the time series of maxima:
  seas_usd <- stl(mm_gbp_usd, s.window = "period")
  seas_jpy <- stl(mm_gbp_jpy, s.window = "period")

  # Remove the seasonality and trend:
  mm_gbp_usd_filt <- mm_gbp_usd - rowSums(seas_usd$time.series[,-3])
  mm_gbp_jpy_filt <- mm_gbp_jpy - rowSums(seas_jpy$time.series[,-3])

  # Estimation of margins and dependence
  mm_gbp <- cbind(as.vector(mm_gbp_usd_filt), as.vector(mm_gbp_jpy_filt))

  hyperparam <- list(mu.nbinom = 3.2, var.nbinom = 4.48)
  gbp_mar <- fExtDep.np(x = mm_gbp, method = "Bayesian", par10 = rep(0.1, 3),
                        par20 = rep(0.1, 3), sig10 = 0.0001, sig20 = 0.0001, k0 = 5,
                        hyperparam = hyperparam, nsim = 5e4)

  gbp_mar_sum <- summary(object = gbp_mar, burn = 3e4, plot = TRUE)

  mm_gbp_range <- apply(mm_gbp, 2, quantile, c(0.9, 0.995))

  y_gbp_usd <- seq(from = mm_gbp_range[1, 1], to = mm_gbp_range[2, 1], length = 20)
  y_gbp_jpy <- seq(from = mm_gbp_range[1, 2], to = mm_gbp_range[2, 2], length = 20)
  y <- as.matrix(expand.grid(y_gbp_usd, y_gbp_jpy, KEEP.OUT.ATTRS = FALSE))

  ret_marg <- returns(x = gbp_mar, summary.mcmc = gbp_mar_sum, y = y, plot = TRUE,
               data = mm_gbp, xlab = "GBP/USD exchange rate", ylab = "GBP/JPY exchange rate")
}

############################################################
### Example 2 - reproducing results from Marcon et al. ###
############################################################

## Not run:
  set.seed(1890)
```

```
data <- evd::rbvevd(n = 100, dep = 0.6, asy = c(0.8, 0.3),
                    model = "alog", mar1 = c(1, 1, 1))

hyperparam <- list(a.unif = 0, b.unif = .5, mu.nbinom = 3.2, var.nbinom = 4.48)
pm0 <- list(p0 = 0.06573614, p1 = 0.3752118)

mcmc <- fExtDep.np(method = "Bayesian", data = data, mar.fit = FALSE, k0 = 5,
                   pm0 = pm0, prior.k = "nbinom", prior.pm = "unif",
                   hyperparam = hyperparam, nsim = 5e5)

w <- seq(0.001, 0.999, length = 100)
summary.mcmc <- summary(object = mcmc, w = w, burn = 4e5, plot = TRUE)

plot(x = mcmc, type = "A", summary.mcmc = summary.mcmc)
plot(x = mcmc, type = "h", summary.mcmc = summary.mcmc)
plot(x = mcmc, type = "pm", summary.mcmc = summary.mcmc)
plot(x = mcmc, type = "k", summary.mcmc = summary.mcmc)

y <- seq(10, 100, 2)
y <- as.matrix(expand.grid(y, y))
probs <- returns(out = mcmc, summary.mcmc = summary.mcmc, y = y, plot = TRUE)

## End(Not run)
```

`returns.plot` *Plot return levels*

Description

Displays return levels for bivariate and trivariate extreme value models.

Usage

```
returns.plot(model, par, Q.fix, Q.range, Q.range0, cond, labels, cex.lab, ...)
```

Arguments

`model`	A string giving the name of the model considered. Takes values `"HR"`, `"ET"` or `"EST"`.
`par`	A numeric vector representing the parameters of the model.
`Q.fix`	A vector of length equal to the model dimension, indicating fixed quantiles for computing joint return levels. Must contain `NA` (maximum 2) for components whose quantiles are allowed to vary.
`Q.range`	A vector or matrix indicating quantile values on the unit Frechet scale, for the components allowed to vary. Must be a vector or a one-column matrix if there is one `NA` in `Q.fix`, or a two-column matrix if there are two `NA`s.
`Q.range0`	An object of the same format as `Q.range`, corresponding to quantiles on the original scale.

cond	Logical; if TRUE, conditional return levels are computed where the conditional event is the fixed event. Default is FALSE.
labels	A character vector giving axis labels. Must be of length 1 or 2 for uni/bi-variate return levels.
cex.lab	A positive numeric value indicating label size.
...	Additional graphical arguments passed to plot and contour.

Details

Two cases are possible: univariate and bivariate return levels. Model dimensions are restricted to a maximum of three. In this case:

- A univariate return level fixes two components.
- A bivariate return level fixes one component.

The choice of fixed components is determined by the positions of the NA values in Q.fix.

If par is a vector, the corresponding return level(s) are printed. If par is a matrix, return level(s) are evaluated for each parameter vector and the mean and empirical 95% interval are displayed. This is typically used with posterior samples. If par has only two rows, the resulting plots may be uninformative.

When contours are displayed, levels correspond to deciles.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com

See Also

dExtDep, index.ExtDep

Examples

```
data(pollution)

X.range <- seq(from = 68, to = 400, length = 10)
Y.range <- seq(from = 182.6, to = 800, length = 10)

transform <- function(x, data, par) {
  data <- na.omit(data)
  if (x > par[1]) {
    emp.dist <- mean(data <= par[1])
    dist <- 1 - (1 - emp.dist) *
      max(0, 1 + par[3] * (x - par[1]) / par[2])^(-1 / par[3])
  } else {
    dist <- mean(data <= x)
  }
  return(-1 / log(dist))
}
```

```
Q.range <- cbind(
  sapply(X.range, transform, data = winterdat[, 1],
         par = c(68, 36.7, 0.29)),
  sapply(Y.range, transform, data = winterdat[, 1],
         par = c(183, 136.7, 0.13))
)
Q.range0 <- cbind(X.range, Y.range)


  returns.plot(model = "HR", par = c(0.6, 0.9, 1.0),
               Q.fix = c(NA, NA, 7),
               Q.range = Q.range, Q.range0 = Q.range0,
               labels = c("PM10", "NO"))
```

---

rExtDep *Parametric and semi-parametric random generator of extreme events*

Description

Generates random samples of iid observations from extremal dependence models and semi-parametric stochastic generators.

Usage

```
rExtDep(n, model, par, angular = FALSE, mar = c(1,1,1), num, threshold, exceed.type)
```

Arguments

n	An integer indicating the number of observations.
model	A character string with the name of the model. Parametric models include "HR" (Husler-Reiss), "ET" (Extremal-t), "EST" (Extremal Skew-t). Semi-parametric generators include "semi.bvevd" and "semi.bvexceed".
par	A vector representing the parameters of the (parametric or non-parametric) model.
angular	Logical; TRUE for angular outputs.
mar	A vector or matrix of marginal parameters.
num	An integer indicating the number of observations over which the componentwise maxima is computed. Required for model = "HR", "ET" or "EST". Default: 5e+5.
threshold	A bivariate vector indicating the level of exceedances. Required for model = "semi.bvexceed".
exceed.type	A character string taking values "and" or "or" indicating the type of exceedances. Required for model = "semi.bvexceed".

Details

There is no limit on the dimensionality when `model = "HR"`, `"ET"` or `"EST"`, while `model = "semi.bvevd"` and `"semi.bvexceed"` can only generate bivariate observations.

When `angular = TRUE` and `model = "semi.bvevd"` or `"semi.bvexceed"`, the simulation of pseudo-angles follows Algorithm 1 of Marcon et al. (2017).

When `model = "semi.bvevd"` and `angular = FALSE`, maxima samples are generated according to Algorithm 2 of Marcon et al. (2017).

When `model = "semi.bvexceed"` and `angular = FALSE`, exceedance samples are generated above the value specified by `threshold`, according to Algorithm 3 of Marcon et al. (2017). `exceed.type = "and"` generates samples that exceed both thresholds while `exceed.type = "or"` generates samples exceeding at least one threshold.

If `mar` is a vector, the marginal distributions are identical. If a matrix is provided, each row corresponds to a set of marginal parameters. No marginal transformation is applied when `mar = c(1,1,1)`.

Value

A matrix with n rows and $p \geq 2$ columns. $p = 2$ when `model = "semi.bvevd"` or `"semi.bvexceed"`.

Author(s)

Simone Padoan <simone.padoan@unibocconi.it> https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger <borisberanger@gmail.com> https://www.borisberanger.com;

References

Marcon, G., Naveau, P. and Padoan, S. A. (2017). A semi-parametric stochastic generator for bivariate extreme events. *Stat*, **6**, 184–201.

See Also

dExtDep, pExtDep, fExtDep, fExtDep.np

Examples

```
# Example using the trivariate Husler-Reiss
set.seed(1)
data <- rExtDep(n = 10, model = "HR", par = c(2,3,3))

# Example using the semi-parametric generator of maxima
set.seed(2)
beta <- c(1.0000000, 0.8714286, 0.7671560, 0.7569398,
          0.7771908, 0.8031573, 0.8857143, 1.0000000)
data <- rExtDep(n = 10, model = "semi.bvevd", par = beta,
                mar = rbind(c(0.2, 1.5, 0.6), c(-0.5, 0.4, 0.9)))

# Example using the semi-parametric generator of exceedances
set.seed(3)
data <- rExtDep(n = 10, model = "semi.bvexceed", par = beta,
```

```
threshold = c(0.2, 0.4), exceed.type = "and")
```

`rExtDepSpat` *Random generation of max-stable processes*

Description

Generates realizations from a max-stable process.

Usage

```
rExtDepSpat(n, coord, model = "SCH", cov.mod = "whitmat", grid = FALSE,
            control = list(), cholsky = TRUE, ...)
```

Arguments

`n`	An integer indicating the number of observations.
`coord`	A vector or matrix corresponding to the coordinates of locations where the process is simulated. Each row corresponds to a location.
`model`	A character string indicating the max-stable model. See `details`.
`cov.mod`	A character string indicating the correlation function. See `details`.
`grid`	Logical; `TRUE` if coordinates are on a grid.
`control`	A named list with arguments: **nlines** Number of lines of the TBM simulation. **method** Character string specifying the simulation method: `'exact'`, `'tbm'` or `'circ'`. **uBound** Uniform upper bound. See `details`.
`cholsky`	Logical; if `TRUE`, a Cholesky decomposition is performed. Used for the extremal-t and extremal skew-t models when `control = list(method = 'exact')`.
`...`	Additional parameters of the max-stable model. See `details`.

Details

This function extends the `rmaxstab` function from the `SpatialExtremes` package in two ways:

1. The **extremal skew-t** model is included.
2. The function returns the hitting scenarios, i.e., the index of which 'storm' (or process) led to the maximum value for each location and observation.

Available max-stable models:

Smith model (`model = 'SMI'`) does not require `cov.mod`. If `coord` is univariate, `var` needs to be specified. For higher dimensions, covariance parameters such as `cov11`, `cov12`, `cov22`, etc. should be provided.

Schlather model (`model = 'SCH'`) requires `cov.mod` as one of `'whitmat'`, `'cauchy'`, `'powexp'` or `'bessel'`. Parameters `nugget`, `range` and `smooth` must be specified.

Extremal-t model (`model = 'ET'`) requires `cov.mod` as above. Parameters `nugget`, `range`, `smooth` and `DoF` must be specified.

Extremal skew-t model (`model = 'EST'`) requires `cov.mod` as above. Parameters `nugget`, `range`, `smooth`, `DoF`, `alpha` (vector of length 3) and `acov1`, `acov2` (vectors of length equal to number of locations) must be specified. The skewness vector is $\alpha = \alpha_0 + \alpha_1 \text{acov1} + \alpha_2 \text{acov2}$.

Geometric Gaussian model (`model = 'GG'`) requires `cov.mod` as above. Parameters `sig2`, `nugget`, `range` and `smooth` must be specified.

Brown-Resnick model (`model = 'BR'`) does not require `cov.mod`. Parameters `range` and `smooth` must be specified.

method In `control`: NULL by default, meaning the function selects the most appropriate simulation technique. For the extremal skew-t model, only `'exact'` or `'direct'` are allowed.

nlines In `control`: default 1000 if `NULL`.

uBound In `control`: default reasonable values, e.g., 3.5 for the Schlather model.

Value

A list containing:

vals A $n \times d$ matrix with n observations at d locations.

hits A $n \times d$ matrix of hitting scenarios. Elements with the same integer indicate maxima coming from the same 'storm' or process.

Author(s)

Simone Padoan <simone.padoan@unibocconi.it> https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger <borisberanger@gmail.com> https://www.borisberanger.com;

References

Beranger, B., Stephenson, A. G. and Sisson, S. A. (2021). High-dimensional inference using the extremal skew-t process. *Extremes*, **24**, 653–685.

See Also

`fExtDepSpat`

Examples

```
# Generate some locations
set.seed(1)
lat <- lon <- seq(from = -5, to = 5, length = 20)
sites <- as.matrix(expand.grid(lat, lon))

# Example using the extremal-t
set.seed(2)
z <- rExtDepSpat(1, sites, model = "ET", cov.mod = "powexp", DoF = 1,
                 nugget = 0, range = 3, smooth = 1.5,
                 control = list(method = "exact"))
fields::image.plot(lat, lon, matrix(z$vals, ncol = 20))

# Example using the extremal skew-t
set.seed(3)
z2 <- rExtDepSpat(1, sites, model = "EST", cov.mod = "powexp", DoF = 5,
                  nugget = 0, range = 3, smooth = 1.5, alpha = c(0,5,5),
                  acov1 = sites[,1], acov2 = sites[,2],
                  control = list(method = "exact"))
fields::image.plot(lat, lon, matrix(z2$vals, ncol = 20))
```

---

simplex *Definition of a multivariate simplex*

Description

Generation of grid points over the multivariate simplex

Usage

```
simplex(d, n = 50, a = 0, b = 1)
```

Arguments

d	A positive integer indicating the dimension of the simplex.
n	A positive integer indicating the number of grid points to be generated on the univariate components of the simplex.
a, b	Two numeric values indicating the lower and upper bounds of the simplex. By default a = 0 and b = 1, indicating the unit-simplex.

Details

A d-dimensional simplex is defined by

$$S = \{(\omega_1, \ldots, \omega_d) \in \mathbb{R}_+^d : \sum_{i=1}^{d} \omega_i = 1\}.$$

Here the function defines the simplex as

$$S = \{(\omega_1, \ldots, \omega_d) \in [a, b]^d : \sum_{i=1}^{d} \omega_i = 1\}.$$

When d = 2 and $[a, b] = [0, 1]$, a grid of points of the form $\{(\omega_1, \omega_2) \in [0, 1] : \omega_1 + \omega_2 = 1\}$ is generated.

Value

Returns a matrix with d columns.

When d = 2, the number of rows is n.

When d > 2, the number of rows is equal to

$$\sum_{i_{d-1}=0}^{n-1} \sum_{i_{d-2}=0}^{n-i_{d-1}} \cdots \sum_{i_1=1}^{n-i_{d-1}-\cdots-i_2} i_1.$$

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/; Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

Examples

```
### 3-dimensional unit simplex
W <- simplex(d = 3, n = 10)
plot(W[,-3], pch = 16)
```

StdErr *Extract the standard errors of estimated parameters*

Description

This function extracts the standard errors of estimated parameters from a fitted object.

Usage

```
StdErr(x, digits = 3)
```

Arguments

x	An object of class ExtDep_Bayes, ExtDep_Freq, or ExtDep_Spat.
digits	Integer indicating the number of decimal places to report. Default is 3.

Value

A numeric vector containing the standard errors of the estimated parameters.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

See Also

fExtDepSpat, fExtDep

Examples

```
data(pollution)

f.hr <- fExtDep(
  x = PNS,
  method = "PPP",
  model = "HR",
  par.start = rep(0.5, 3),
  trace = 2
)

StdErr(f.hr)
```

---

tic *Extract the Takeuchi Information Criterion*

Description

This function extracts the TIC value from a fitted object.

Usage

```
tic(x, digits = 3)
```

Arguments

x	An object of class ExtDep_Freq or ExtDep_Spat.
digits	Integer indicating the number of decimal places to report. Default is 3.

Value

A numeric vector containing the TIC value(s) of the fitted object.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

See Also

fExtDep, fExtDepSpat

Examples

```
data(pollution)

f.hr <- fExtDep(
  x = PNS,
  method = "PPP",
  model = "HR",
  par.start = rep(0.5, 3),
  trace = 2
)

tic(f.hr)
```

---

trans2GEV *Transformation to GEV Distribution*

Description

Transforms marginal distributions from unit Frechet to the GEV scale.

Usage

```
trans2GEV(data, pars)
```

Arguments

data	A vector of length n or a $(n \times p)$ matrix representing the data on its original scale.
pars	A (1×3) vector or a $(p \times 3)$ matrix of marginal GEV parameters.

Details

The transformation is given by $(x^{\xi} - 1)\frac{\sigma}{\xi} + \mu$ if $\xi \neq 0$, and by $\sigma/x + \mu$ if $\xi = 0$.

Value

An object of the same format and dimensions as data.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

See Also

trans2UFrechet

Examples

```
data(pollution)

pars <- fGEV(Leeds.frechet[,1])$est

par_new <- c(2, 1.5, 0.5)
data_new <- trans2GEV(Leeds.frechet[,1], pars = par_new)

fGEV(data_new)
```

---

trans2UFrechet *Transformation to Unit Frechet Distribution*

Description

Empirical and parametric transformation of a dataset to unit Frechet marginal distribution.

Usage

```
trans2UFrechet(data, pars, type = "Empirical")
```

Arguments

data	A vector of length n or a $(n \times p)$ matrix representing the data on its original scale.
pars	A (1×3) vector or a $(p \times 3)$ matrix of marginal GEV parameters. Required when type = "GEV".
type	A character string indicating the type of transformation. Can be "Empirical" or "GEV".

Details

When type = "Empirical", the transformation is $t(x) = -1/\log(F_{\text{emp}}(x))$ where $F_{\text{emp}}(x)$ denotes the empirical cumulative distribution function.

When type = "GEV", the transformation is $\left(1 + \xi\frac{x-\mu}{\sigma}\right)^{1/\xi}$ if $\xi \neq 0$, and $\sigma/(x-\mu)$ if $\xi = 0$. If pars is missing, a GEV is fitted on the columns of data using the fGEV function.

Value

An object of the same format and dimensions as data.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://faculty.unibocconi.it/simonepadoan/;
Boris Beranger, <borisberanger@gmail.com>, https://www.borisberanger.com;

See Also

trans2GEV, fGEV

Examples

```
data(MilanPollution)

pars <- fGEV(Milan.winter$PM10)$est

data_uf <- trans2UFrechet(data = Milan.winter$PM10, pars = pars, type = "GEV")
fGEV(data_uf)$est

data_uf2 <- trans2UFrechet(data = Milan.winter$PM10, type = "Empirical")
fGEV(data_uf2)$est
```

Wind *Weekly Maximum Wind Speed Data Across 4 Stations in Oklahoma, USA (March-May, 1996-2012)*

Description

Four datasets of weekly maximum wind speed for each triplet of locations: CLOU.CLAY.SALL, CLOU.CLAY.PAUL, CLAY.SALL.PAUL, and CLOU.SALL.PAUL.

Details

CLOU.CLAY.SALL is a data.frame with 3 columns and 212 rows. CLOU.CLAY.PAUL is a data.frame with 3 columns and 217 rows. CLAY.SALL.PAUL is a data.frame with 3 columns and 211 rows. CLOU.SALL.PAUL is a data.frame with 3 columns and 217 rows.

Missing observations have been discarded for each triplet.

References

Beranger, B., Padoan, S. A., & Sisson, S. A. (2017). Models for extremal dependence derived from skew-symmetric families. *Scandinavian Journal of Statistics*, **44**(1), 21-45.

`WindSpeedGust` *Hourly Wind Gust, Wind Speed, and Air Pressure at Lingen (GER), Ossendorf (GER), and Parcay-Meslay (FRA)*

Description

Three `data.frame` objects, one for each location.

Details

Each object contains the following columns:

WS: Hourly wind speed in metres per second (m/s).

WG: Hourly wind gust in metres per second (m/s).

DP: Hourly air pressure at sea level in millibars.

Details for each object:

Lingen: `data.frame` with 1083 rows and 4 columns. Measurements recorded between January 1982 and June 2003.

Ossendorf: `data.frame` with 676 rows and 4 columns. Measurements recorded between March 1982 and August 1995.

ParcayMeslay: `data.frame` with 2140 rows and 4 columns. Measurements recorded between November 1984 and July 2013.

References

Marcon, G., Naveau, P., & Padoan, S.A. (2017). A semi-parametric stochastic generator for bivariate extreme events. *Stat*, **6**, 184-201.

Index

∗ **Bootstrap confidence bands**
beed.confband, 10
∗ **Nonparametric**
beed, 6
diagnostics, 19
fExtDep.np, 36
∗ **bootstrap**
beed.boot, 9
∗ **datasets**
heat, 50
logReturns, 54
MilanPollution, 58
pollution, 65
PrecipFrance, 66
Wind, 80
WindSpeedGust, 81
∗ **distribution**
desn, 13
dest, 14
dGEV, 18
dmesn, 22
dmest, 24
∗ **htest**
ExtQ, 28
∗ **models**
angular.plot, 5
bic, 12
dExtDep, 15
dim_ExtDep, 21
ellipse, 25
est, 27
fExtDep, 32
fExtDep.np, 36
fExtDepSpat, 44
fGEV, 47
index.ExtDep, 51
lambda.HR, 53
madogram, 55
method, 57
model, 58
pExtDep, 60
pFailure, 62
pickands.plot, 64
returns, 67
returns.plot, 69
rExtDep, 71
rExtDepSpat, 73
StdErr, 76
tic, 77
trans2GEV, 78
trans2UFrechet, 79
∗ **multivariate**
simplex, 75

angular, 3
angular.plot, 5, *33, 34*

beed, 6, *9–12*, *56*
beed.boot, 9, *12*
beed.confband, *8*, 10, *10*, *56*
bic, 12

CLAY.SALL.PAUL *(*Wind*)*, 80
CLOU.CLAY.PAUL *(*Wind*)*, 80
CLOU.CLAY.SALL *(*Wind*)*, 80
CLOU.SALL.PAUL *(*Wind*)*, 80
contour, *70*

desn, 13
dest, 14
dExtDep, *6*, 15, *33*, *35*, *42*, *62, 63*, *65*, *70*, *72*
dGEV, 18, *49*
diagnostics, 19
dim_ExtDep, 21, *61*
dmesn, 22
dmest, 24

ellipse, 25
est, 27
ExtQ, 28

fExtDep, *13*, *17*, *28*, 32, *42*, *57, 58*, *62, 63*, *72*, *77, 78*
fExtDep.np, *17*, *19, 20*, *35*, 36, *62, 63*, *67*, *72*
fExtDepSpat, *28*, 44, *47*, *57, 58*, *74*, *77, 78*
fGEV, *18, 19*, *29*, 47, *79, 80*

heat, 50
heatdata *(*heat*)*, 50

index.ExtDep, 51, *65*, *70*

lambda.HR, 53
lat *(*PrecipFrance*)*, 66
Leeds.frechet *(*pollution*)*, 65
Lingen *(*WindSpeedGust*)*, 81
locgrid *(*heat*)*, 50
logLik.ExtDep_Freq *(*fExtDep*)*, 32
logLik.ExtDep_Spat *(*fExtDepSpat*)*, 44
logReturns, 54
lon *(*PrecipFrance*)*, 66

madogram, 55
mellat *(*heat*)*, 50
mellon *(*heat*)*, 50
method, 57
Milan.summer *(*MilanPollution*)*, 58
Milan.winter *(*MilanPollution*)*, 58
MilanPollution, 58
model, 58

NSN *(*pollution*)*, 65

optim, *33*, *46*, *48, 49*
Ossendorf *(*WindSpeedGust*)*, 81

pam, *60*
pam.object, *59*
PAMfmado, 59
ParcayMeslay *(*WindSpeedGust*)*, 81
pesn *(*desn*)*, 13
pest *(*dest*)*, 14
pExtDep, *17*, *35*, *42*, 60, *72*
pFailure, 62
pGEV *(*dGEV*)*, 18
pickands.plot, *33, 34*, 64
plot, *70*
plot.ExtDep_Bayes *(*fExtDep*)*, 32
plot.ExtDep_Freq *(*fExtDep*)*, 32
plot.ExtDep_npBayes, *67*
plot.ExtDep_npBayes *(*fExtDep.np*)*, 36
plot.ExtDep_npEmp *(*fExtDep.np*)*, 36
plot.ExtDep_npFreq *(*fExtDep.np*)*, 36
pmesn *(*dmesn*)*, 22
pmest *(*dmest*)*, 24
PNN *(*pollution*)*, 65
PNNS *(*pollution*)*, 65
PNS *(*pollution*)*, 65
pollution, 65
precip *(*PrecipFrance*)*, 66
PrecipFrance, 66

qGEV, *29*
qGEV *(*dGEV*)*, 18

return, *38*
returns, 67
returns.plot, *34*, 69
rExtDep, *17*, *35*, *42*, *62, 63*, 71
rExtDepSpat, 73

scalegrid *(*heat*)*, 50
shapegrid *(*heat*)*, 50
simplex, 75
StdErr, 76
summary.ExtDep_Bayes *(*fExtDep*)*, 32
summary.ExtDep_npBayes, *67*
summary.ExtDep_npBayes *(*fExtDep.np*)*, 36

tic, 77
trans2GEV, 78, *80*
trans2UFrechet, 79, *79*

Wind, 80
WindSpeedGust, 81
winterdat *(*pollution*)*, 65