

# Package ‘RTDE’

May 7, 2026

**Type** Package

**Title** Robust Tail Dependence Estimation

**Version** 0.2-2

**Description** Robust tail dependence estimation for bivariate models. This package is based on two papers by the authors: 'Robust and bias-corrected estimation of the coefficient of tail dependence' and 'Robust and bias-corrected estimation of probabilities of extreme failure sets'. This work was supported by a research grant (VKR023480) from VIL-LUM FONDEN and an international project for scientific cooperation (PICS-6416).

**Depends** R (>= 3.0.0), parallel, methods

**Suggests** tseries

**License** GPL (>= 2)

**NeedsCompilation** no

**Author** Christophe Dutang [aut, cre] (ORCID: <<https://orcid.org/0000-0001-6732-1501>>),  
Armelle Guillou [ctb] (ORCID: <<https://orcid.org/0000-0001-7136-8687>>),  
Yuri Goegebeur [ctb] (ORCID: <<https://orcid.org/0000-0002-9976-5040>>)

**Maintainer** Christophe Dutang <dutangc@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-10-16 14:30:05 UTC

## Contents

dataRTDE . . . . .	2
EPD . . . . .	4
FGM . . . . .	6
fitRTDE . . . . .	8
Frank . . . . .	10
Frechet . . . . .	11
MDPD . . . . .	13
qqparetoplot . . . . .	14
RTDE . . . . .	15
upareto . . . . .	18
zvalueRTDE . . . . .	19

---

dataRTDE	<i>Data object used for a Tail Dependence model</i>
----------	---

---

### Description

Data object used for a Tail Dependence model.

### Usage

```
dataRTDE(obs, simu.nb, simu.marg=c("ufrechet", "upareto"),
  simu.cop=c("indep", "FGM", "Frank"), simu.cop.par=NULL,
  contamin.eps=NULL, contamin.method=c("NA", "max+", "+"),
  contamin.marg=c("ufrechet", "upareto"),
  contamin.cop=c("indep", "FGM", "Frank"),
  contamin.cop.par=NULL, control=list())
```

```
## S3 method for class 'dataRTDE'
print(x, ...)
## S3 method for class 'dataRTDE'
summary(object, ...)
## S3 method for class 'dataRTDE'
plot(x, which=1:2, ...)
```

### Arguments

obs	bivariate numeric dataset.
simu.nb	a numeric for the sample size of simulated data.
simu.marg	a character string for the marginal distribution: either "ufrechet" (default) or "upareto".
simu.cop	a character string ofr the copula: either "indep" (default), "FGM" or "Frank".
simu.cop.par	a numeric for the copula parameter, default to NULL.
contamin.eps	a numeric for the percentage (of simu.nb) of contaminated data.
contamin.method	a character string for the contamination method: either "NA" (default), "max+" or "+".
contamin.marg	a character string for the marginal distribution: either "ufrechet" (default) or "upareto".
contamin.cop	a character string ofr the copula: either "indep" (default), "FGM" or "Frank".
contamin.cop.par	a numeric for the copula parameter, default to NULL.
control	A list of control paremeters. Unused.

x, object	an R object inheriting from "dataRTDE".
...	arguments to be passed to subsequent methods.
which	an integer (1 or 2) to specify whether to plot in original scale or unit-Pareto scale, respectively.

## Details

The function `dataRTDE` handles empirical or simulated data and may add a contamination.

**Empirical data** When `obs` is provided, `dataRTDE` just wraps the two-column matrix  $(X_i, Y_i)_i$ .

**Simulated data** When `simu.XXX` are provided, `dataRTDE` simulates random vectors  $(X_i, Y_i)_i$  from the copula `simu.cop` with parameter `simu.cop.par` and marginal `simu.marg`.

Note that end-user must choose between empirical data (`obs` is provided) and simulated data (`simu.XXX` are provided). Not both can be provided. In addition to data handling  $(X_i, Y_i)_i$ , a contamination can be processed by adding new simulated points  $(\tilde{X}_i, \tilde{Y}_i)_i$  when `contamin.method != "NA"`. Those points  $(\tilde{X}_i, \tilde{Y}_i)_i$  are simulated from the copula `contamin.cop` with parameter `contamin.cop.par` and marginal `contamin.cop.par`. If `contamin.method != "+"`, the points  $(\tilde{X}_i, \tilde{Y}_i)_i$  are the contaminations, while if `contamin.method != "max+"` the contaminations are obtained by adding the component-wise maximum of the data:  $(\tilde{X}_i + X_{n,n}, \tilde{Y}_i + Y_{n,n})_i$ , where  $X_{n,n} = \max(X_1, \dots, X_n)$ , idem for  $Y_{n,n}$ .

## Value

`dataRTDE` returns an object of class "dataRTDE" having the following components:

`n` rownumber of data.

`n0` rownumber of `contamin`.

`data` original or simulated data.

`contamin` contaminated data.

## Author(s)

Christophe Dutang

## References

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, Volume 57, Insurance: Mathematics and Economics

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

## See Also

See [fitRTDE](#) for the fitting process and [zvalueRTDE](#) for the z-value computation.

**Examples**

```
#####
# (1) simulation

n <- 100
x <- dataRTDE(simu.nb=n, simu.marg="ufrechet", simu.cop="indep")
print(x)
summary(x)
plot(x, xlab="x", ylab="y")

#####
# (2) part of the workers' compensation dataset

x1 <- c(
  21.798086, 22.640528, 22.572010, 24.789710, 25.876764, 28.033613,
  22.525887, 12.004031, 12.713178, 13.596610, 14.811727, 12.774073,
  20.245789, 24.242468, 50.216515, 56.099793, 58.109747, 67.807105,
  73.852437, 84.208474, 83.604216, 19.507341, 20.810822, 23.838122,
  24.212193, 25.367578, 35.401344, 37.580989, 12.428727, 13.492474,
  23.471988, 24.101833, 24.766193, 26.078216)

x2 <- c(
  0.538707, 0.439184, 1.059775, 0.560013, 1.004997, 1.097314, 0.609833, 0.270222,
  0.229566, 0.596850, 0.196539, 0.134248, 0.489312, 0.418218, 0.769208, 0.649707,
  0.503919, 0.675466, 0.545745, 1.562266, 0.931762, 0.291125, 0.499927, 0.151084,
  0.141910, 0.300373, 0.119761, 0.141300, 0.377662, 0.169574, 0.243585, 0.061215,
  0.055272, 0.312816, 0.160196, 0.623029, 0.280707, 0.174422, 0.176666, 0.153907,
  0.605122, 0.664457, 0.348918, 0.370878)

obs <- dataRTDE(cbind(x1, x2))
obs
summary(obs)

plot(obs)
```

---

 EPD

*The Extended Pareto Distribution*


---

**Description**

Density function, distribution function, quantile function, random generation.

**Usage**

```
dEPD(x, eta, delta, rho, tau, log = FALSE)
pEPD(q, eta, delta, rho, tau, lower.tail=TRUE, log.p = FALSE)
qEPD(p, eta, delta, rho, tau, lower.tail=TRUE, log.p = FALSE,
      control=list())
rEPD(n, eta, delta, rho, tau)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>eta</code>	first shape parameter.
<code>delta</code>	nuisance parameter.
<code>rho, tau</code>	second shape parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .
<code>control</code>	A list of control parameters. See section Details.

**Details**

The extended Pareto distribution is defined by the following density

$$f(x) = \frac{1}{\eta} x^{-1/\eta-1} [1 + \delta(1 - x^{-\tau})]^{-1/\eta-1} [1 + \delta(1 - (1 - \tau)x^{-\tau})]$$

for all  $x > 1$  when parametrized by  $\tau$ . However, a typical parametrization is obtained by setting  $\tau = -\rho/\eta$ , i.e.

$$f(x) = \frac{1}{\eta} x^{-1/\eta-1} [1 + \delta(1 - x^{\rho/\eta})]^{-1/\eta-1} [1 + \delta(1 - (1 + \rho/\eta)x^{\rho/\eta})]$$

for all  $x > 1$  when parametrized by  $\rho$ .

The control argument is a list that can supply any of the following components:

- `upperbound` The upperbound used in the `optimize` function when computing numerical quantiles, default to `1e6`.
- `tol` the desired accuracy used in the `optimize` function when computing numerical quantiles, default to `1e-9`.

**Value**

`dEPD` gives the density, `pEPD` gives the distribution function, `qEPD` gives the quantile function, and `rEPD` generates random deviates.

The length of the result is determined by `n` for `rEPD`, and is the maximum of the lengths of the numerical parameters for the other functions.

The numerical parameters other than `n` are recycled to the length of the result. Only the first elements of the logical parameters are used.

**Author(s)**

Christophe Dutang

## References

J. Beirlant, E. Joossens, J. Segers (2009), *Second-order refined peaks-over-threshold modelling for heavy-tailed distributions*, Journal of Statistical Planning and Inference, Volume 139, Issue 8, Pages 2800-2815.

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, Insurance: Mathematics and Economics

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

## Examples

```
#####
# (1) density function
x <- seq(0, 5, length=24)

cbind(x, dEPD(x, 1/2, 1/4, -1))

#####
# (2) distribution function

cbind(x, pEPD(x, 1/2, 1/4, -1, lower=FALSE))
```

---

 FGM

*The Eyraud Farlie Gumbel Morgenstern Distribution*


---

## Description

Density function, distribution function, quantile function, random generation.

## Usage

```
dFGM(u, v, alpha, log = FALSE)
pFGM(u, v, alpha, lower.tail=TRUE, log.p = FALSE)
qFGM(p, alpha, lower.tail=TRUE, log.p = FALSE)
rFGM(n, alpha)
```

## Arguments

u, v	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

alpha	shape parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

### Details

The FGM is defined by the following distribution function

$$C(u, v) = u * v * (1 + \alpha * (1 - u) * (1 - v))$$

for all  $u, v$  in  $[0,1]$  and  $\alpha$  in  $[0,1]$ . When `lower.tail=FALSE`, pFGM returns the survival copula  $P(U > u, V > v)$ .

### Value

dFGM gives the density, pFGM gives the distribution function, qFGM gives the quantile function, and rFGM generates random deviates.

The length of the result is determined by `n` for rFGM, and is the maximum of the lengths of the numerical parameters for the other functions.

The numerical parameters other than `n` are recycled to the length of the result. Only the first elements of the logical parameters are used.

### Author(s)

Christophe Dutang

### References

Nelsen, R. (2006), *An Introduction to Copula, Second Edition*, Springer.

### Examples

```
#####
# (1) density function
u <- v <- seq(0, 1, length=25)

cbind(u, v, dFGM(u, v, 1/2))
cbind(u, v, outer(u, v, dFGM, alpha=1/2))

#####
# (2) distribution function

cbind(u, v, pFGM(u, v, 1/2))
cbind(u, v, outer(u, v, pFGM, alpha=1/2))
```

fitRTDE

*Fitting a Tail Dependence model with a Robust Estimator***Description**

Fit a Tail Dependence model with a Robust Estimator.

**Usage**

```
fitRTDE(obs, nbpoint, alpha, omega, method="MDPDE", fix.arg=list(rho=-1),
        boundary.method="log", control=list())
```

```
## S3 method for class 'fitRTDE'
print(x, ...)
## S3 method for class 'fitRTDE'
summary(object, ...)
## S3 method for class 'fitRTDE'
plot(x, which=1:2, main, ...)
```

**Arguments**

obs	bivariate numeric dataset.
nbpoint	a numeric for the number of largest points to be selected.
alpha	a numeric for the power divergence parameter.
omega	a numeric for omega, see section Details.
method	a character string equals to "MDPDE".
fix.arg	a named list of fixed arguments: either <i>rho</i> only e.g. <code>list(rho=-1)</code> or <i>rho</i> , <i>delta</i> e.g. <code>list(rho=-1, delta=0)</code> .
boundary.method	a character string: either "log" or "simple", see section Details.
control	A list of control parameters. See section Details.
x, object	an R object inheriting from "fitRTDE".
...	arguments to be passed to subsequent methods.
which	an integer (1 or 2) to specify whether to plot eta or delta, respectively.
main	a main title for the plot.

**Details**

The function `fitRTDE` fits an extended Pareto distribution ( $\eta, \tau$  are fitted while  $\rho$  is fixed) on the relative excess of  $Z_\omega$  (see `zvalueRTDE`) using a robust estimator based on the minimum distance power divergence criterion (see `MDPD`). The boundary enforcement on  $\eta, \tau$  is either done by the bounded BFGS algorithm (see `optim` with `method="L-BFGS-B"`) or by the bounded Nelder-Mead algorithm (see `constrOptim` with `method="Nelder-Mead"`).

**Value**

fitRTDE returns an object of class "fitRTDE" having the following components:

n rownumber of data.  
 n0 rownumber of contamin.  
 alpha a vector of alpha parameters.  
 omega a vector of omega parameters.  
 m a vector of nbpoint.  
 rho a numeric for rho.  
 eta estimate of *eta*.  
 delta estimate of *delta*.  
 Ztilde see [zvalueRTDE](#).

**Author(s)**

Christophe Dutang

**References**

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, Volume 57, Insurance: Mathematics and Economics

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

**Examples**

```
#####
# (1) simulation

omega <- 1/2
m <- 48
n <- 100
obs <- cbind(rupareto(n), rupareto(n)) + rupareto(n)

#function of m
system.time(
x <- fitRTDE(obs, nbpoint=m:(n-m), 0, 1/2)
)
x
summary(x)
plot(x, which=1)
plot(x, which=2)
```

**Description**

Density function, distribution function, quantile function, random generation.

**Usage**

```
dfrank(u, v, alpha, log = FALSE)
pfrank(u, v, alpha, lower.tail=TRUE, log.p = FALSE)
qfrank(p, alpha, lower.tail=TRUE, log.p = FALSE)
rfrank(n, alpha)
```

**Arguments**

<code>u, v</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>alpha</code>	shape parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

The Frank is defined by the following distribution function

$$C(u, v) = -\frac{1}{\alpha} \log \left[ 1 - \frac{(1 - e^{-\alpha u})(1 - e^{-\alpha v})}{1 - e^{-\alpha}} \right],$$

for all  $u, v$  in  $[0, 1]$ . When `lower.tail=FALSE`, `pfrank` returns the survival copula  $P(U > u, V > v)$ .

**Value**

`dfrank` gives the density, `pfrank` gives the distribution function, `qfrank` gives the quantile function, and `rfrank` generates random deviates.

The length of the result is determined by `n` for `rfrank`, and is the maximum of the lengths of the numerical parameters for the other functions.

The numerical parameters other than `n` are recycled to the length of the result. Only the first elements of the logical parameters are used.

**Author(s)**

Christophe Dutang

## References

Nelsen, R. (2006), *An Introduction to Copula, Second Edition*, Springer.

## Examples

```
#####  
# (1) density function  
u <- v <- seq(0, 1, length=25)  
  
cbind(u, v, dfrank(u, v, 1/2))  
cbind(u, v, outer(u, v, dfrank, alpha=1/2))  
  
#####  
# (2) distribution function  
  
cbind(u, v, pfrank(u, v, 1/2))  
cbind(u, v, outer(u, v, pfrank, alpha=1/2))
```

---

Frechet

*The Frechet Distribution*

---

## Description

Density function, distribution function, quantile function, random generation.

## Usage

```
dfrechet(x, shape, xmin, log = FALSE)  
pfrechet(q, shape, xmin, lower.tail=TRUE, log.p = FALSE)  
qfrechet(p, shape, xmin, lower.tail=TRUE, log.p = FALSE)  
rfrechet(n, shape, xmin)  
  
dufrechet(x, log = FALSE)  
pufrechet(q, lower.tail=TRUE, log.p = FALSE)  
qufrechet(p, lower.tail=TRUE, log.p = FALSE)  
rufrechet(n)
```

## Arguments

x, q	vector of quantiles.
p	vector of probabilities.

n	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
shape	shape parameter.
xmin	lower bound parameter.
log, log.p	logical; if TRUE, probabilities p are given as <code>log(p)</code> .
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

### Details

The Frechet distribution is defined by the following density

$$f(x) = \text{shape} * (x - \text{xmin})^{(-\text{shape}-1)} * \exp(-(x - \text{xmin})^{(-\text{shape})})$$

for all  $x > \text{xmin}$ . The unit Frechet distribution corresponds to `xmin=0` and `shape=1`.

### Value

`dfrechet`, `dufrechet` give the density, `pfrechet`, `pufrechet` give the distribution function, `qfrechet`, `qufrechet` give the quantile function, and `rfrechet`, `rufrechet` generate random deviates.

The length of the result is determined by `n` for `rfrechet`, `rufrechet`, and is the maximum of the lengths of the numerical parameters for the other functions.

The numerical parameters other than `n` are recycled to the length of the result. Only the first elements of the logical parameters are used.

### Author(s)

Christophe Dutang

### References

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

Beirlant, J., Goegebeur, Y., Teugels, J., Segers (2004), *Statistics of Extremes: Theory and Applications*, John Wiley and Sons.

### Examples

```
#####
# (1) density function
x <- seq(0, 5, length=24)

cbind(x, dfrechet(x, 1/2, 1/4))

#####
# (2) distribution function

cbind(x, pfrechet(x, 1/2, 1/4))
```

**Description**

Computes the power divergence statistics then used a minimization problem.

**Usage**

```
MDPD(theta, densfun, obs, alpha, ..., control=list())
```

**Arguments**

theta	the parameter of the distribution given as a vector.
densfun	a function computing the theoretical density function.
obs	a numeric vector of observations
alpha	a numeric for the power divergence parameter.
...	further arguments to be passed to the density function.
control	A list of control paremeters. See section Details.

**Details**

The Power Divergence for a density function  $f$  and observations  $X_1, \dots, X_n$  is defined as

$$\Delta(f, \alpha) = \int_R f^{1+\alpha}(x) dx - \left(1 + \frac{1}{\alpha}\right) \frac{1}{n} \sum_{i=1}^n f^\alpha(X_i)$$

for  $\alpha > 0$

$$\Delta(f, 0) = -\frac{1}{n} \sum_{i=1}^n \log f(X_i)$$

for  $\alpha = 0$ .

The control argument is a list that can supply any of the following components:

eps a small positive floating-point number used when integrate stalled, default to 1e-3.  
 tol the desired accuracy used in the integrate function when computing the power divergence, default to 1e-3.  
 lower the lower bound of the domain of the density function, default to 1.  
 upper the lower bound of the domain of the density function, default to infinity.

**Value**

MDPD returns the power divergence against the density function densfun as a numeric.

**Author(s)**

Christophe Dutang

**References**

Basu, A., Harris, I.R., Hjort, N.L., Jones, M.C., (1998). *Robust and efficient estimation by minimizing a density power divergence*, *Biometrika*, 85, 549-559.

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, *Insurance: Mathematics and Economics*

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

**Examples**

```
#####
# (1) small example

omega <- 1/2
m <- 10
n <- 100
obs <- cbind(rupareto(n), rupareto(n)) + rupareto(n)

#unit Pareto transform
z <- zvalueRTDE(obs, omega, nbpoint=m, output="relexcess")

MDPD(c(1/2, 1/4), dEPD, z$Z, alpha=0, rho=-1)
```

---

qqparetoplot

*The QQ Pareto plot*


---

**Description**

Plot the quantile-quantile Pareto plot

**Usage**

```
qqparetoplot(x, ..., highlight=c("red","cross"))
```

**Arguments**

`x` data vector.  
`highlight` character string used in points to plot outliers.  
`...` further arguments for `plot.default`.

**Details**

`qqparetoplot` plots the quantile-quantile Pareto plot and may highlight some points having name "new".

**Value**

Invisible list with component `x` for the x-coordinates and `y` for the y-coordinates.

**Author(s)**

Christophe Dutang

**Examples**

```
#####
# (1) small examples

set.seed(1234)
x <- rupareto(100)
qqparetoplot(x)

x <- rexp(100)
qqparetoplot(x)
```

---

 RTDE

---

*Data object used for a Tail Dependence model*


---

**Description**

Data object used for a Tail Dependence model.

**Usage**

```
RTDE(obs=NULL, simu=list(), contamin=list(),
      nbpoint, alpha, omega, method="MDPDE", fix.arg=list(rho=-1),
      boundary.method="log", core=1, keepdata, control=list())
```

```
## S3 method for class 'RTDE'
print(x, ...)
```

```

## S3 method for class 'RTDE'
summary(object, ...)
## S3 method for class 'RTDE'
plot(x, which=1:3, FUN=mean, main, ...)

prob(object, q, ...)
## Default S3 method:
prob(object, q, ...)
## S3 method for class 'RTDE'
prob(object, q, ...)

```

## Arguments

<code>obs</code>	bivariate numeric dataset.
<code>simu</code>	a names list with components: "nb", "marg", "cop", "replicate". When needed, "cop.par" must be provided, see <a href="#">dataRTDE</a> .
<code>contamin</code>	a names list with components: "eps", "method", "marg", "cop". When needed, "cop.par" must be provided, see <a href="#">dataRTDE</a> .
<code>nbpoint</code>	a numeric for the number of largest points to be selected.
<code>alpha</code>	a numeric for the power divergence parameter.
<code>omega</code>	a numeric for omega, see section Details.
<code>method</code>	a character string equals to "MDPDE".
<code>fix.arg</code>	a named list of fixed arguments: either <i>rho</i> only e.g. <code>list(rho=-1)</code> or <i>rho, delta</i> e.g. <code>list(rho=-1, delta=0)</code> .
<code>boundary.method</code>	a character string: either "log" or "simple", see section Details.
<code>core</code>	a numeric for the number of core to be used, only relevant for simulated data.
<code>keepdata</code>	a logical whether to return or not the dataset.
<code>control</code>	A list of control paremeters for <a href="#">fitRTDE</a> .
<code>x, object</code>	an R object inheriting from "RTDE".
<code>...</code>	arguments to be passed to subsequent methods.
<code>which</code>	an integer to specify what to plot: 1 eta, 2 delta, 3 probability estimates.
<code>FUN</code>	the function to be applied, default to <a href="#">mean</a> .
<code>main</code>	a main title for the plot.
<code>q</code>	vector of quantiles.

## Details

The function RTDE handles (empirical or simulated) data (cf. [dataRTDE](#)) and then fits a bivariate tail model using a method criterion (cf. [fitRTDE](#) and MDPD) based on an extended Pareto distribution approximation (EPD). Typical distributions for simulated data and/or contaminations are

**Marginal** Unit Pareto [upareto](#), Frechet [Frechet](#).

**Copula** Frank [Frank](#), FGM [FGM](#).

For a good introduction, please refer to references.

### Value

RTDE returns an object of class "RTDE" having the following components:

obs.type see [dataRTDE](#).

data see [dataRTDE](#).

fit see [fitRTDE](#).

simu see [dataRTDE](#).

contamin see [dataRTDE](#).

setting a list summarizing the computation.

### Author(s)

Christophe Dutang

### References

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, Volume 57, Insurance: Mathematics and Economics

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

### See Also

See [fitRTDE](#) for the fitting process and [dataRTDE](#) for the data-handling process.

### Examples

```
#####  
# (1) simulation  
  
n <- 100  
x <- RTDE(simu=list(nb=n, marg="ufrechet", cop="indep", replicate=1),  
nbpoint=10:11, alpha=0, omega=1/2)  
x  
summary(x)
```

---

upareto

*The unit Pareto Distribution*


---

**Description**

Density function, distribution function, quantile function, random generation.

**Usage**

```
dupareto(x, log = FALSE)
pupareto(q, lower.tail=TRUE, log.p = FALSE)
qpareto(p, lower.tail=TRUE, log.p = FALSE)
rupareto(n)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
log, log.p	logical; if TRUE, probabilities p are given as $\log(p)$ .
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

The extended Pareto distribution is defined by the following density and distribution function

$$f(x) = \frac{1}{x^2}, F(x) = 1 - \frac{1}{x},$$

for all  $x > 0$ .

**Value**

`dupareto` gives the density, `pupareto` gives the distribution function, `qpareto` gives the quantile function, and `rupareto` generates random deviates.

The length of the result is determined by `n` for `rupareto`, and is the maximum of the lengths of the numerical parameters for the other functions.

The numerical parameters other than `n` are recycled to the length of the result. Only the first elements of the logical parameters are used.

**Author(s)**

Christophe Dutang

## References

Johnson, N.L., Kotz, S. and Balakrishnan, N. (2000), *Continuous Univariate Distributions, Volume 1, Second Edition*, John Wiley and Sons.

## Examples

```
#####  
# (1) density function  
x <- seq(0, 5, length=24)  
  
cbind(x, dupareto(x))  
  
#####  
# (2) distribution function  
  
cbind(x, pupareto(x))
```

---

zvalueRTDE

*The Z-value random variable*

---

## Description

Compute the Z-value variable from a bivariate dataset.

## Usage

```
zvalueRTDE(obs, omega, nbpoint, output=c("orig", "relexcess"),  
           marg=c("upareto", "ufrechet", "uunif"))  
  
## S3 method for class 'zvalueRTDE'  
print(x, ...)  
## S3 method for class 'zvalueRTDE'  
summary(object, ...)  
  
relexcess(x, nbpoint, ...)  
## Default S3 method:  
relexcess(x, nbpoint, ...)  
## S3 method for class 'zvalueRTDE'  
relexcess(x, nbpoint, ...)
```

**Arguments**

obs	bivariate numeric dataset.
omega	a numeric for omega, see Details.
nbpoint	a numeric for the number of largest points to be selected.
output	a character string for the output: either "orig" for original value or "relexcess" for relative excess.
marg	a character string for the empirical margin transformation: either "upareto" for unit Pareto, "ufrechet" for unit Frechet or "uunif" for unit uniform margin.
x, object	an R object inheriting from "zvalueRTDE".
...	arguments to be passed to subsequent methods.

**Details**

Given a bivariate dataset  $(X_i, Y_i)_i$  of  $n$  points, two variables are defined: (1) for output="orig", the  $\tilde{Z}_{\omega, i}$  variable

$$\tilde{Z}_{\omega, i} = \min \left( f \left( \frac{R_i^X}{n+1} \right), \frac{\omega}{1-\omega} f \left( \frac{R_i^Y}{n+1} \right) \right)$$

where  $f(x)$  is the margin transformation and  $i = 1, \dots, n$ ; (2) for output="relexcess", the  $Z_j$  variable

$$\frac{\tilde{Z}_{\omega, n-m+j, n}}{\tilde{Z}_{\omega, n-m, n}}$$

where  $m$  equals nbpoint,  $j = 1, \dots, m$ , and  $\tilde{Z}_{\omega, 1, n}, \dots, \tilde{Z}_{\omega, n, n}$  are the order statistics of  $\tilde{Z}_{\omega, 1}, \dots, \tilde{Z}_{\omega, n}$ . The margin transformation is

$$f(x) = \frac{1}{1-x}, f(x) = \frac{1}{-\log(x)}, f(x) = x,$$

respectively for unit Pareto (marg="upareto"), unit Frechet (marg="ufrechet") and unit uniform margin (marg="uunif").

**Value**

zvalueRTDE computes the Z-variable and returns an object of class "zvalueRTDE" having the following components type (either "orig" or "relexcess"), omega, Ztilde or Z, n, possibly m.

relexcess computes the relative excesses from a Z-variable and returns an object of class "zvalueRTDE" of type "relexcess".

**Author(s)**

Christophe Dutang

**References**

C. Dutang, Y. Goegebeur, A. Guillou (2014), *Robust and bias-corrected estimation of the coefficient of tail dependence*, Volume 57, Insurance: Mathematics and Economics

This work was supported by a research grant (VKR023480) from VILLUM FONDEN and an international project for scientific cooperation (PICS-6416).

**See Also**

See [fitRTDE](#) for the fitting process and [dataRTDE](#) for the data-handling process.

**Examples**

```
#####  
# (1) example  
  
omega <- 1/2  
m <- 10  
n <- 100  
obs <- cbind(rupareto(n), rupareto(n)) + rupareto(n)  
  
#unit Pareto transform  
zvalueRTDE(obs, omega, output="orig")  
  
relexcess(zvalueRTDE(obs, omega, output="orig"), m)  
zvalueRTDE(obs, omega, nbpoint=m, output="relexcess")
```

# Index

## \* distribution

- dataRTDE, [2](#)
- EPD, [4](#)
- FGM, [6](#)
- fitRTDE, [8](#)
- Frank, [10](#)
- Frechet, [11](#)
- MDPD, [13](#)
- qqparetoplot, [14](#)
- RTDE, [15](#)
- upareto, [18](#)
- zvalueRTDE, [19](#)

constrOptim, [8](#)

dataRTDE, [2](#), [16](#), [17](#), [21](#)

dEPD (EPD), [4](#)

dFGM (FGM), [6](#)

dfrank (Frank), [10](#)

dfrechet (Frechet), [11](#)

dufrechet (Frechet), [11](#)

dupareto (upareto), [18](#)

EPD, [4](#), [16](#)

FGM, [6](#), [17](#)

fitRTDE, [3](#), [8](#), [16](#), [17](#), [21](#)

Frank, [10](#), [17](#)

Frechet, [11](#), [17](#)

MDPD, [8](#), [13](#), [16](#)

mean, [16](#)

optim, [8](#)

pEPD (EPD), [4](#)

pFGM (FGM), [6](#)

pfrank (Frank), [10](#)

pfrechet (Frechet), [11](#)

plot.dataRTDE (dataRTDE), [2](#)

plot.fitRTDE (fitRTDE), [8](#)

plot.RTDE (RTDE), [15](#)

print.dataRTDE (dataRTDE), [2](#)

print.fitRTDE (fitRTDE), [8](#)

print.RTDE (RTDE), [15](#)

print.zvalueRTDE (zvalueRTDE), [19](#)

prob (RTDE), [15](#)

pufrechet (Frechet), [11](#)

pupareto (upareto), [18](#)

qEPD (EPD), [4](#)

qFGM (FGM), [6](#)

qfrank (Frank), [10](#)

qfrechet (Frechet), [11](#)

qqparetoplot, [14](#)

qufrechet (Frechet), [11](#)

qupareto (upareto), [18](#)

relexcess (zvalueRTDE), [19](#)

rEPD (EPD), [4](#)

rFGM (FGM), [6](#)

rfrank (Frank), [10](#)

rfrechet (Frechet), [11](#)

RTDE, [15](#)

rufrechet (Frechet), [11](#)

rupareto (upareto), [18](#)

summary.dataRTDE (dataRTDE), [2](#)

summary.fitRTDE (fitRTDE), [8](#)

summary.RTDE (RTDE), [15](#)

summary.zvalueRTDE (zvalueRTDE), [19](#)

upareto, [17](#), [18](#)

zvalueRTDE, [3](#), [8](#), [9](#), [19](#)