

Package ‘gumbel’

October 17, 2024

Type Package

Title The Gumbel-Hougaard Copula

Version 1.10-3

Description Provides probability functions (cumulative distribution and density functions), simulation function (Gumbel copula multivariate simulation) and estimation functions (Maximum Likelihood Estimation, Inference For Margins, Moment Based Estimation and Canonical Maximum Likelihood).

Depends R (>= 2.10.0)

License GPL (>= 2)

BuildVignettes true

NeedsCompilation no

Author Christophe Dutang [aut, cre] (<<https://orcid.org/0000-0001-6732-1501>>),
Anne-Lise Caillat [ctb],
Veronique Larrieu [ctb],
Triet Nguyen [ctb]

Maintainer Christophe Dutang <dutangc@gmail.com>

Repository CRAN

Date/Publication 2024-10-17 14:30:07 UTC

Contents

| | |
|-------------------|----------|
| Gumbel | 2 |
| windata | 6 |
| Index | 8 |

Description

Density function, distribution function, random generation, generator and inverse generator function for the Gumbel Copula with parameters alpha. The 4 classic estimation methods for copulas.

Usage

```
dgumbel(u, v=NULL, alpha, dim=2, warning = FALSE)
pgumbel(u, v=NULL, alpha, dim=2)
rgumbel(n, alpha, dim=2, method=1)
phigumbel(t, alpha=1)
invphigumbel(t, alpha=1)
gumbel.MBE(x, y, marg = "exp")
gumbel.EML(x, y, marg = "exp")
gumbel.IFM(x, y, marg = "exp")
gumbel.CML(x, y)
```

Arguments

| | |
|---------|---|
| u | vector of quantiles if argument v is provided or matrix of quantiles if argument v is not provided |
| v | vector of quantiles, needed if u is not a matrix |
| n | number of observations. If length(n) > 1, the length is taken to be the number required. |
| alpha | parameter of the Copula. Must be greater than 1. |
| dim | an integer specifying the dimension of the copula. |
| t | dummy variable of the generator ϕ or the inverse generator ϕ^{-1} . could be a n-dimensional array. |
| method | an integer code for the method used in simulation. 1 is the common frailty approach, 2 uses the K function (only valid with dim=2). |
| x, y | vectors of observations, realizations of random variable X and Y . |
| marg | a character string specifying the marginals of vector (X, Y) . It must be either "exp"(default value) or "gamma". |
| warning | a logical (default value FALSE) if you want warnings. |

Details

The Gumbel Hougaard Copula with parameter alpha is defined by its generator

$$\phi(t) = (-\ln(t))^{\alpha}.$$

The generator and inverse generator are implemented in `phigumbel` and `invphigumbel` respectively. As an Archimedean copula, its distribution function is

$$C(u_1, \dots, u_n) = \phi^{-1}(\phi(u_1) + \dots + \phi(u_n)) = \exp(-((- \ln(u_1))^{\alpha} + \dots + (- \ln(u_n))^{\alpha})^{1/\alpha}).$$

`pgumbel` and `dgumbel` computes the distribution function (expression above) and the density (n times differentiation of expression above with respect to u_i). As there is no explicit formulas for the density of a Gumbel copula, `dgumbel` is not yet implemented for argument `dim>3`. This two functions works with a `dim`-dimensional array with the last dimension being equalled to `dim` or with a matrix with `dim` columns (see examples).

Random generation is carried out with 2 algorithms the common frailty algorithm (`method=1`) and the 'K' algorithm (`method=2`). The common frailty algorithm (cf. Marshall & Olkin(1988)) can be sum up in three lines

- generate y_1, y_2 from exponential distribution of mean 1,
- generate θ from a stable distribution with parameter $(1/\alpha, 1, 1, 0)$,
- $u_1 < -\text{phi}(y_1/\theta)$ and $u_2 < -\text{phi}(y_2/\theta)$.

This algorithm works with any dimension. See Chambers et al(1976) for stable random generation. The 'K' algorithm use the fact the distribution function of random variable $C(U, V)$ is $K(t) = t - \phi(y)/\phi'(t)$. The algorithm is

- generate v_1, t from uniform distribution,
- generate v_2 from the K distribution i.e. $v_2 < -K^{-1}(t)$,
- $u_1 < -\phi^{-1}(\phi(v_1)v_2)$ and $u_2 < -\phi^{-1}(\phi(v_1)(1 - v_2))$.

Warning, the 'K' algorithm does NOT work with `dim>2`.

We implements the 4 usual method of estimation for copulas, namely the Exact Maximum Likelihood (`gumbel.EML`), the Inference for Margins (`gumbel.IFM`), the Moment-base Estimation (`gumbel.MBE`) and the Canonical Maximum Likelihood (`gumbel.CML`). For the moment, only two types of marginals are available : exponential distribution (`marg="exp"`) and gamma distribution (`marg="gamma"`).

Value

`dgumbel` gives the density, `pgumbel` gives the distribution function, `rgumbel` generates random deviates, `phigumbel` gives the generator, `invphigumbel` gives the inverse generator.

`gumbel.EML`, `gumbel.IFM`, `gumbel.MBE` and `gumbel.CML` returns the vector of estimates.

Invalid arguments will result in return value `NaN`.

Author(s)

A.-L. Caillat, C. Dutang, M. V. Larrieu and T. Nguyen

References

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

Marshall & Olkin(1988), *Families of multivariate distributions*, Journal of the American Statistical Association.

Chambers et al (1976), *A method for simulating stable random variables*, Journal of the American Statistical Association.

Nelsen, R. (2005), *Dependence Modeling with Archimedean Copulas*, booklet available at www.lclark.edu/~mathsci/brazil2.p

Examples

```
#dim=2
u<-seq(0,1, .1)
v<-u
#classic parametrization
#independence case (alpha=1)
dgumbel(u,v,1)
pgumbel(u,v,1)
#another parametrization
dgumbel(cbind(u,v), alpha=1)
pgumbel(cbind(u,v), alpha=1)

#dim=3 - equivalent parametrization
x <- cbind(u,u,u)
y <- array(u, c(1,11,3))
pgumbel(x, alpha=2, dim=3)
pgumbel(y, alpha=2, dim=3)
dgumbel(x, alpha=2, dim=3)
dgumbel(y, alpha=2, dim=3)

#dim=4
x <- cbind(x,u)
pgumbel(x, alpha=3, dim=4)
y <- array(u, c(2,1,11,4))
pgumbel(y, alpha=3, dim=4)

#independence case
rand <- t(rgumbel(200,1))
plot(rand[1,], rand[2,], col="green", main="Gumbel copula")

#positive dependence
rand <- t(rgumbel(200,2))
plot(rand[1,], rand[2,], col="red", main="Gumbel copula")

#comparison of random generation algorithms
nbsimu <- 10000
#Marshall Olkin algorithm
system.time(rgumbel(nbsimu, 2, dim=2, method=1))[3]
#K algortihm
system.time(rgumbel(nbsimu, 2, dim=2, method=2))[3]

#pseudo animation
## Not run:
anim <-function(n, max=50)
{
for(i in seq(1,max,length.out=n))
```

```

{
u <- t(rgumbel(10000, i, method=2))
plot(u[1,], u[2,], col="green", main="Gumbel copula",
      xlim=c(0,1), ylim=c(0,1), pch=".")

cat()
}
}
anim(20, 20)

## End(Not run)

#3D plots

#plot the density
x <- seq(.05, .95, length = 30)
y <- x
z <- outer(x, y, dgumbel, alpha=2)

persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
      ltheta = 100, shade = 0.25, ticktype = "detailed",
      xlab = "x", ylab = "y", zlab = "density")

#with wonderful colors
#code of P. Soutiras
zlim <- c(0, max(z))
ncol <- 100
nrz <- nrow(z)
ncz <- ncol(z)
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
"cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#F00000"))
couleurs <- tail(jet.colors(1.2*ncol),ncol)
fcol <- couleurs[trunc(z/zlim[2]*(ncol-1))+1]
dim(fcol) <- c(nrz,ncz)
fcol <- fcol[-nrz,-ncz]
persp(x, y, z, col=fcol, zlim=zlim, theta=30, phi=30, ticktype = "detailed",
      box = TRUE, xlab = "x", ylab = "y", zlab = "density")

#plot the distribution function
z <- outer(x, y, pgumbel, alpha=2)
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
      ltheta = 100, shade = 0.25, ticktype = "detailed",
      xlab = "u", ylab = "v", zlab = "cdf")

#parameter estimation
#true value : lambdaX=lambdaY=1, alpha=2
simu <- qexp(rgumbel(200, 2))
gumbel.MBE(simu[,1], simu[,2])
gumbel.IFM(simu[,1], simu[,2])
gumbel.EML(simu[,1], simu[,2])
gumbel.CML(simu[,1], simu[,2])

```

```
#true value : lambdaX=lambdaY=1, alphaX=alphaY=2, alpha=3
simu <- qgamma(rgumbel(200, 3), 2, 1)
gumbel.MBE(simu[,1], simu[,2], "gamma")
gumbel.IFM(simu[,1], simu[,2], "gamma")
gumbel.EML(simu[,1], simu[,2], "gamma")
gumbel.CML(simu[,1], simu[,2])
```

windata

Daily Climatological data between August 2005 and May 2007

Description

Daily Climatological data recorded in two French cities: Echirolles and St Martin-En-Haut. Weather stations are located at Echirolles (ELEV: 237m, LAT: 45 06' 00" N LONG: 5 42' 00" E) and La Rafiliere (ELEV: 575m, LAT: 45 39' 00" N LONG: 4 33' 00" E), respectively.

Usage

```
data(windStMartin)
data(windEchirolles)
```

Format

windStMartin and windEchirolles are data frames of 15 columns:

YEAR Year.

MONTH Month number.

DAY Day number.

TEMP.MEAN Average temperature (Celsius degree).

TEMP.HIGH Maximum temperature.

TIME.TH Time of the maximum temperature (hh:mm).

TEMP.LOW Minimum temperature.

TIME.TL Time of the minimum temperature.

HDD Heating Degree Days.

CDD Cooling Degree Days.

RAIN Rain (mm).

WIND.MEAN Wind speed average (km/h).

WIND.HIGH Wind speed maximum.

WIND.TH Time of the wind speed maximum.

DOM.DIR Dominant direction of the wind, a character string where "N" for North, "NE" for North-East, etc...

windata

7

Source

See <https://meteo.data.gouv.fr/datasets>

Index

- * **datasets**

- windata, 6

- * **distribution**

- Gumbel, 2

dgumbel (Gumbel), 2

Gumbel, 2

gumbel (Gumbel), 2

invphigumbel (Gumbel), 2

pgumbel (Gumbel), 2

phigumbel (Gumbel), 2

rgumbel (Gumbel), 2

windata, 6

windEchirolles (windata), 6

windStMartin (windata), 6