Package: GRIDCOPULA (via r-universe)

August 22, 2024

Title Bivariate Copula Functions Based on Regular Grid

Description Estimates grid type bivariate copula functions, calculates some association measures and provides several copula graphics.

Version 1.0.0

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.1

Imports Rsolnp, fields, limSolve, ggplot2, pracma, reshape2

NeedsCompilation no

Author Rogelio Salinas Gutiérrez Developer [aut, cre, cph], Angélica Hernández Quintero Dev [aut], Pedro Abraham Montoya Calzada Dev [cph]

Maintainer Rogelio Salinas Gutiérrez Developer

<rsalinas@correo.uaa.mx>

Repository CRAN

Date/Publication 2022-09-13 11:40:02 UTC

Contents

`aic.grid` *Calculates the AIC of a grid type copula*

Description

This function receives a grid type copula as a parameter and returns the value of the AIC.

Usage

```
aic.grid(gc)
```

Arguments

`gc`	a grid type copula object.

Value

Returns a number with the AIC of a grid type copula.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
```

```
v <- Fy(y)
df <- cbind(u,v)
k <- 5
m <- 4
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
aic.grid(copula.grid)
```

`bic.grid` *Calculates the BIC of a grid type copula*

Description

This function receives a grid type copula as a parameter and returns the value of the BIC.

Usage

```
bic.grid(gc)
```

Arguments

gc	a grid type copula object.

Value

Returns a number with the BIC of a grid type copula.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 5
m <- 4
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
bic.grid(copula.grid)
```

`calculate.ls` *Creates a grid type copula according to a given data set and the Pfeifer proposal*

Description

Returns a list with a matrix with the density over the grid, a matrix with the quantity of data over the grid, the number of subintervals for the U2 variable, the number of subintervals for the U1 variable.

Usage

```
calculate.ls(U, k, m)
```

Arguments

`U`	a matrix of size nx2 with the observed values.
`k`	positive integer indicating the number of subintervals for the U2 variable.
`m`	positive integer indicating the number of subintervals for the U1 variable.

`calculate.ml` *Estimates the parameters of a grid type copula according to their log-likelihood and a given data set*

Description

Returns a list with a matrix with the density over the grid, a matrix with the quantity of data over the grid, the number of subintervals for the U2 variable, the number of subintervals for the U1 variable.

Usage

```
calculate.ml(U, k, m, D.ini = NULL)
```

Arguments

`U`	a matrix of size nx2 with the observed values.
`k`	positive integer indicating the number of subintervals for the U2 variable.
`m`	positive integer indicating the number of subintervals for the U1 variable.
`D.ini`	a matrix with the initial values for the density copula.package: the name of the package for numerical optimization.

`cdf.grid` *Calculates the distribution function of a grid copula for a single data point*

Description

Returns the corresponding distribution function value.

Usage

```
cdf.grid(u, mg, Vm)
```

Arguments

`u`	vector of size 1x2 with the values of the variables.
`mg`	a grid type copula object.
`Vm`	Volume matrix.

`contour_color_grid` *Draws the density / distribution function of a grid copula with contours and colors*

Description

Draws the density / distribution function of a grid copula with contours and colors

Usage

```
contour_color_grid(
  gc,
  FUN = "d.grid",
  u1 = seq(0, 1, length.out = 21),
  u2 = seq(0, 1, length.out = 21),
  color.name = "heat.colors",
  color.size = 7
)
```

Arguments

`gc`	a grid type copula object.
`FUN`	the name of the function to be applied (d.grid, p.grid), default is 'd.grid'.
`u1`	indicates the place for lines on axis u1.
`u2`	indicates the place for lines on axis u2.
`color.name`	indicates the palette of colors.
`color.size`	indicates the number of colors.

Value

Returns a graph of the density / distribution.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
contour_color_grid(gc = copula.grid, FUN = 'd.grid', color.name = "rainbow")
contour_color_grid(gc = copula.grid, FUN = 'p.grid', color.name = "rainbow")
```

---

contour_grid *Draws the density / distribution function of a grid copula with contours*

Description

Draws the density / distribution function of a grid copula with contours

Usage

```
contour_grid(
  gc,
  FUN = "d.grid",
  u1 = seq(0, 1, length.out = 21),
  u2 = seq(0, 1, length.out = 21)
)
```

Arguments

gc	a grid type copula object.
FUN	the name of the function to be applied (d.grid, p.grid), default is 'd.grid'.
u1	indicates the place for lines on axis u1.
u2	indicates the place for lines on axis u2.

Value

Returns a graph of the density / distribution.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
contour_grid(gc = copula.grid, FUN = 'd.grid')
contour_grid(gc = copula.grid, FUN = 'p.grid')
```

`contour_image_grid` *Draws the density / distribution function of a grid copula with contours and colors*

Description

Draws the density / distribution function of a grid copula with contours and colors

Usage

```
contour_image_grid(
  gc,
  FUN = "p.grid",
  u1 = seq(0, 1, length.out = 100),
  u2 = seq(0, 1, length.out = 100),
  color.name = "heat.colors",
  color.size = 40
)
```

Arguments

`gc`	a grid type copula object.
`FUN`	the name of the function to be applied (d.grid, p.grid), default is 'p.grid'.
`u1`	indicates the place for lines on axis u1.
`u2`	indicates the place for lines on axis u2.
`color.name`	indicates the palette of colors.
`color.size`	indicates the number of colors.

Value

Returns a graph of the density / distribution.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ls")
contour_image_grid(gc = copula.grid, FUN = 'd.grid', color.name= "rainbow", color.size = 10)
contour_image_grid(gc = copula.grid, FUN = 'p.grid', color.name = "rainbow", color.size = 10)
```

`count.grid` *Counts the data according to a specified grid*

Description

Returns a matrix of size kxm indicating the quantity of data.

Usage

```
count.grid(U, k, m)
```

Arguments

`U`	a matrix of size nx2 with the observed values.
`k`	a positive integer indicating the number of subintervals for the U2 variable.
`m`	a positive integer indicating the number of subintervals for the U1 variable.

`d.grid` *Evaluates the density of a grid type copula*

Description

Returns the corresponding density values of a grid type copula.

Usage

```
d.grid(U, V = NULL, gc)
```

Arguments

U	a matrix of size nx2 with the observed values. It can also be a vector of size kx1 with the values of the U1 variable.
V	optional, a vector of size kx1 with the values of the U2 variable.
gc	a grid type copula object.

Value

Returns a vector with the corresponding density.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 5
m <- 4
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
d.grid(df,gc=copula.grid)
```

---

data.grid *Draws the scatter plot of bivariate data in the unit square*

Description

Draws the scatter plot of bivariate data in the unit square

Usage

```
data.grid(U, draw.lines = TRUE, k = 4, m = 4)
```

Arguments

U	matrix of size kx2 with the values of both variables.
draw.lines	draws lines inside the unit square or not.
k	positive integer indicating the number of subintervals for the U2 variable.
m	positive integer indicating the number of subintervals for the U1 variable.

Value

Returns a scatter plot of bivariate data in the unit square.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
data.grid(U=df, draw.lines = FALSE, k = k, m = m)
data.grid(U=df, draw.lines = TRUE, k = k, m = m)
```

---

equalities.grid *Evaluates the equalities for the parameters of a grid type copula*

Description

Returns the corresponding value of the equalities.

Usage

```
equalities.grid(x, A)
```

Arguments

x	a vector with the density values of a grid type copula.
A	a matrix considered as a reference with the size of the grid.

---

estimate.gridCopula *Estimates the parameters of a grid type copula*

Description

This function estimates grid type copulas by one of the following methods: maximum likelihood or least squares (See reference).

Usage

```
estimate.gridCopula(
  U,
  k = NULL,
  m = NULL,
  method = "ml",
  D.ini = NULL,
  criterion = "AIC"
)
```

Arguments

U	a matrix of size nx2 with the observed values.
k	positive integer indicating the number of subintervals for the U2 variable.
m	positive integer indicating the number of subintervals for the U1 variable.
method	Method that uses, least squares "ls" or maximum likelihood "ml", by default "ml".
D.ini	a matrix with the initial values for the density copula.package: the name of the package for numerical optimization.
criterion	If the values of k i m are not specified, they will be obtained by the "AIC" or "BIC" criteria, by default "AIC".

Value

Returns a list with a matrix with the density over the grid, a matrix with the quantity of data over the grid, the number of subintervals for the U2 variable, the number of subintervals for the U1 variable.

References

@mischttps://doi.org/10.48550/arxiv.2010.15709, doi = 10.48550/ARXIV.2010.15709, url = https://arxiv.org/abs/2010.1570
author = Pfeifer, Dietmar and Strassburger, Doreen and Philipps, Joerg, keywords = Methodology (stat.ME), Risk Management (q-fin.RM), FOS: Computer and information sciences, FOS: Computer and information sciences, FOS: Economics and business, FOS: Economics and business, 62H05, 62H12, 62H17, 11K45, title = Modelling and simulation of dependence structures in nonlife insurance with Bernstein copulas, publisher = arXiv, year = 2020, copyright = arXiv.org perpetual, non-exclusive license

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 5
m <- 4
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
print(copula.grid$Density)
```

image_color_grid *Draws the density of a grid copula with colors*

Description

Draws the density of a grid copula with colors

Usage

```
image_color_grid(gc, color.name = "heat.colors", color.size = 7)
```

Arguments

gc	a grid type copula object.
color.name	indicates the palette of colors.
color.size	indicates the number of colors.

Value

Returns a graph of the density.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
image_color_grid(gc = copula.grid, color.name = "rainbow", color.size = 10)
```

ll.grid *Calculates the log-likelihood of a grid type copula*

Description

Returns the log-likelihood of a grid type copula.

Usage

```
ll.grid(x, y)
```

Arguments

x	a vector with the density values of a grid type copula.
y	a vector with the counting values in the grid.

---

mi.grid *Calculates the mutual information of a grid type copula*

Description

Calculates the mutual information of a grid type copula

Usage

```
mi.grid(gc)
```

Arguments

gc	a grid type copula object.

Value

Returns a number with the mutual information.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.ml <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
copula.ls <- estimate.gridCopula(U=df, k=k, m=m , method = "ls")
mi.grid(gc = copula.ml)
mi.grid(gc = copula.ls)
```

mosaic.grid *Draws the density of a grid copula with mosaics*

Description

Draws the density of a grid copula with mosaics

Usage

```
mosaic.grid(gc, number.size = 5)
```

Arguments

gc	a grid type copula object.
number.size	indicates the size of numbers.

Value

Returns a graph.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
mosaic.grid(gc = copula.grid, number.size = 5)
```

normal.color.contour.grid *Draws the density of a grid copula with contours, normal marginals and colors*

Description

Draws the density of a grid copula with contours, normal marginals and colors

Usage

```
normal.color.contour.grid(
  gc,
  x1 = seq(-3, 3, length.out = 21),
  x2 = seq(-3, 3, length.out = 21),
  color.name = "gray",
  color.size = 7
)
```

Arguments

gc	a grid type copula object.
x1	indicates the place for lines on axis x1.
x2	indicates the place for lines on axis x2.
color.name	indicates the palette of colors.
color.size	indicates the number of colors.

Value

Returns a graph.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
normal.color.contour.grid(gc = copula.grid, color.name = "heat.colors")
```

---

normal.contour.grid *Draws the density of a grid copula with contours and normal marginals*

Description

Draws the density of a grid copula with contours and normal marginals

Usage

```
normal.contour.grid(
  gc,
  x1 = seq(-3, 3, length.out = 21),
  x2 = seq(-3, 3, length.out = 21)
)
```

Arguments

gc	a grid type copula object.
x1	indicates the place for lines on axis x1.
x2	indicates the place for lines on axis x2.

Value

Returns a graph.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
normal.contour.grid(gc = copula.grid)
```

---

normal.multiplication *Calculates the multiplication of normal densities*

Description

Returns a vector with the values of the product of densities.

Usage

```
normal.multiplication(x, y)
```

Arguments

x	a vector with the values of the variable x.
y	a vector with the values of the variable y.

objective.grid — *Evaluates the negative log-likelihood function associated to a grid type copula*

Description

Returns the corresponding value of the least squares function.

Usage

```
objective.grid(x, A)
```

Arguments

x	a vector with the density values of a grid type copula.
A	a matrix with the counting values in the grid.

p.grid — *Evaluates the distribution function of a grid type copula*

Description

Returns the corresponding distribution function values.

Usage

```
p.grid(U, V = NULL, gc)
```

Arguments

U	a matrix of size nx2 with the observed values. It can also be a vector of size kx1 with the values of the U1 variable.
V	optional, a vector of size kx1 with the values of the U2 variable.
gc	a grid type copula object.

Value

Returns a vector with the corresponding distribution.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 5
m <- 4
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
p.grid(df,gc=copula.grid)
```

`pdf.grid` *Evaluates the density of a grid copula for a single data point*

Description

Returns the corresponding distribution function values.

Usage

```
pdf.grid(u, mg)
```

Arguments

`u`	vector of size 1x2 with the values of the variables.
`mg`	a grid type copula object.

`perspective.grid` *Draws the density / distribution function of a grid copula with perspective*

Description

Draws the density / distribution function of a grid copula with perspective

Usage

```
perspective.grid(
  gc,
  FUN = "d.grid",
  u1 = seq(0, 1, length.out = 21),
  u2 = seq(0, 1, length.out = 21),
  ang.theta = -30,
  ang.phi = 25,
  distancia = 10
)
```

Arguments

gc	a grid type copula object.
FUN	the name of the function to be applied (d.grid, p.grid), default is 'd.grid'.
u1	indicates the place for lines on axis u1.
u2	indicates the place for lines on axis u2.
ang.theta	angle for the azimuthal direction.
ang.phi	angle for the colatitude.
distancia	the distance of the eyepoint from the centre of the box.

Value

Returns a graph of the density / distribution.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
perspective.grid(gc = copula.grid, ang.theta = 90 , ang.phi = 80, distancia = 3)
perspective.grid(gc = copula.grid, FUN = "p.grid")
```

r.cond.grid *Generates a conditional random sample from a grid type copula*

Description

Returns a vector of size nx1 with the random sample of the U2 variable.

Usage

```
r.cond.grid(U, mg)
```

Arguments

U	a vector of size nx1 with values of the U1 variable.
mg	a grid type copula object.

r.grid *Generates a random sample from a grid type copula*

Description

Generates a random sample from a grid type copula

Usage

```
r.grid(n, gc)
```

Arguments

n	an integer number indicating the size of the sample.
gc	a grid type copula object.

Value

Returns a matrix of size nx2 with the random sample.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
```

```
df <- cbind(u,v)
k <- 15
m <- 15
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
df2 <- r.grid(n = n, gc = copula.grid)
data.grid(df, k=k, m=m)
data.grid(df2, k=k, m=m)
```

`rho.grid` *Calculates the Spearman's rho concordance measure for a grid type copula*

Description

Calculates the Spearman's rho concordance measure for a grid type copula

Usage

```
rho.grid(gc)
```

Arguments

gc	a grid type copula object.

Value

Returns a number with the corresponding value.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
rho.grid(gc = copula.grid)
```

`rho.integrand.grid` *Evaluates the integrand function for the Spearman's rho concordance measure of a grid type copula.*

Description

Returns the corresponding value of the integrand function.

Usage

```
rho.integrand.grid(x, y, gc)
```

Arguments

`x`	a vector with values of the U1 variable.
`y`	a vector with values of the U2 variable.
`gc`	a grid type copula object.

`tau.grid` *Calculates the Kendall's tau concordance measure for a grid type copula*

Description

Calculates the Kendall's tau concordance measure for a grid type copula

Usage

```
tau.grid(gc)
```

Arguments

`gc`	a grid type copula object.

Value

Returns a number with the corresponding value.

Examples

```
n <- 500
x <- rgamma(n,4,1/2)
e <- rnorm(n,0,.3)
y <- sin(x+e)
Fx <- ecdf(x)
Fy <- ecdf(y)
u <- Fx(x)
v <- Fy(y)
df <- cbind(u,v)
k <- 10
m <- 10
copula.grid <- estimate.gridCopula(U = df, k = k, m = m , method = "ml")
tau.grid(gc = copula.grid)
```

---

tau.integrand.grid *Evaluates the integrand function for the Kendall's tau concordance measure of a grid type copula.*

Description

Returns the corresponding value of the integrand function.

Usage

```
tau.integrand.grid(x, y, gc)
```

Arguments

x	a vector with values of the U1 variable.
y	a vector with values of the U2 variable.
gc	a grid type copula object.

---

validate.density *Check if the matrix is a density of a copula*

Description

Return a logical value

Usage

```
validate.density(D.ini, k, m)
```

Arguments

D.ini	a matrix with the initial values for the density copula.package: the name of the package for numerical optimization.
k	positive integer indicating the number of subintervals for the U2 variable.
m	positive integer indicating the number of subintervals for the U1 variable.

Index