Package 'nCopula'

November 30, 2018

Type Package

Title Hierarchical Archimedean Copulas Constructed with Multivariate Compound Distributions

Version 0.1.1

Maintainer Simon-Pierre Gadoury <spgadou@me.com>

Description Construct and manipulate hierarchical Archimedean copulas with multivariate compound distributions. The model used is the one of Cossette et al. (2017) <doi:10.1016/j.insmatheco.2017.06.001>.

Depends R (>= 3.4.0), copula

Imports Deriv, stringr, stringi, compiler, methods,

License GPL (>= 2)

LazyData FALSE

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

NeedsCompilation no

Author Simon-Pierre Gadoury [cre, aut], David Beauchemin [ctb]

Repository CRAN

Date/Publication 2018-11-30 12:40:07 UTC

R topics documented:

AMH *Construction of an Archimedean Copula Class Object.*

Description

Constructs an AMH Archimedean copula object with a given parameter and dimension.

Usage

```
AMH(param, dim = 2L, density = FALSE)
```

Arguments

`param`	parameter of the copula.
`dim`	dimension of the copula (>= 2), which is, by default, 2.
`density`	compute the expression of the density of the copulas.

Details

Constructs an AMH Archimedean copula object with a given parameter and dimension.

Value

An archm S4 class object.

Author(s)

Simon-Pierre Gadoury

Clayton *Construction of an Archimedean Copula Class Object*

Description

Constructs a Clayton Archimedean copula object with a given parameter and dimension.

Usage

```
Clayton(param, dim = 2L, density = FALSE)
```

Arguments

param	the parameter of the copula.
dim	the dimension of the copula (>= 2), which is, by default, 2.
density	logical. Should the expression of the density of the copula be computed?

Value

An archm S4 class object.

Author(s)

Simon-Pierre Gadoury

Frank *Construction of an Archimedean Copula Class Object*

Description

Constructs a Frank Archimedean copula object with a given parameter and dimension.

Usage

```
Frank(param, dim = 2L, density = FALSE)
```

Arguments

param	parameter of the copula.
dim	dimension of the copula (>= 2), which is, by default, 2.
density	compute the expression of the density of the copulas.

Value

An archm S4 class object.

Author(s)

Simon-Pierre Gadoury

GAMMA *Construction of a GAMMA Child Class Object*

Description

The function GAMMA constructs a gamma Child class object for a given parameter and arguments.

Usage

```
GAMMA(par, unif, structure = NULL)
```

Arguments

par	parameter of the distribution.
unif	uniform structure, a numeric vector of grouped numbers, i.e. c(1,2,3) is translated as being c(u1, u2, u3).
structure	nesting structure of the form X(par1, c(i,...), list(Y(par2, c(j,...), NULL), Z(par3, c(k,...), NULL))), where X, Y, and Z are compatible functions (see 'details'). It is to note that if structure is NULL, the function will automatically be of class Child. For continuous distributions (i.e. GAMMA), structure is always NULL.

Author(s)

Simon-Pierre Gadoury

See Also

Other mother or child class objects.: GEO, LOG

Examples

```
GEO(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                    GEO(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                                        GAMMA(1/30, c(3,4), NULL)))))
```

GeneticCodes *Obtain the Genetic Codes of a Structure*

Description

Function to obtain the list of all genetic codes of a structure.

Usage

```
GeneticCodes(structure)
```

Arguments

structure	an object of class Mother (the structure)

Value

A list of the structure's genetic codes.

Author(s)

Simon-Pierre Gadoury

Examples

```
## Create the structure
structure <- GEO(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                          GEO(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                                              GAMMA(1/30, c(3,4), NULL)))))
## Get the genetic codes
GeneticCodes(structure)
```

GEO *Construction of a GEO Mother or Child Class Object*

Description

Constructs either a GEO Mother or Child class object for a given parameter, arguments, and nesting structure.

Usage

```
GEO(par, unif, structure)
```

Arguments

par	parameter of the distribution.
unif	uniform structure, a numeric vector of grouped. numbers, i.e. c(1,2,3) is translated as being c(u1, u2, u3).
structure	nesting structure of the form X(par1, c(i,...), list(Y(par2, c(j,...), NULL), Z(par3, c(k,...), NULL))), where X, Y, and Z are compatible functions (see 'details'). It is to note that if structure is NULL, the function will automatically be of class Child. For continuous distributions (i.e. GAMMA), structure is always NULL.

Author(s)

Simon-Pierre Gadoury

See Also

Other mother or child class objects.: GAMMA, LOG

Examples

```
GEO(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                    GEO(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                                        GAMMA(1/30, c(3,4), NULL)))))
```

Gumbel *Construction of an Archimedean Copula Class Object*

Description

Constructs a Gumbel Archimedean copula object with a given parameter and dimension.

Usage

```
Gumbel(param, dim = 2L)
```

Arguments

param	parameter of the copula
dim	dimension of the copula (>= 2), which is, by default, 2

Value

An archm S4 class object.

Author(s)

Simon-Pierre Gadoury

`InvLap` *Inverse LST of a Node*

Description

With a specific path and a predefined structure (S4 class of a type 'Mother'), returns the inverse Laplace-Stieltjes Transform expression of the corresponding node with a specific variable.

Usage

```
InvLap(code, structure, outVar = "z", par = "value")
```

Arguments

`code`	the genetic code (numeric vector) of the node (can be a leaf i.e. end by 0).
`structure`	an object of class Mother (the structure).
`outVar`	the output variable to be used ('z' by default).
`par`	logical. Should the parameters be values ('value') or variables ('variable') ?

Details

For mother nodes, parameters are always called 'gamma' and for child nodes, parameters are always called 'alpha'. Furthermore, to recognize the parameters, the path is inserted at the end. For exemple, a child node with path (0,2,1) will have the parameter 'alpha021'.

Value

A character string giving the inverse LST of the specified node.

Author(s)

Simon-Pierre Gadoury

See Also

Lap

Examples

```
structure <- GEO(0.1, NULL, list(GAMMA(0.1, 1:2, NULL),
                           GAMMA(0.2, 3:4, NULL)))

InvLap(c(0,2), structure, outVar = 'z', par = 'value')
```

Lap *LST of a Node*

Description

With a specific path and a predefined structure (S4 class of a type 'Mother'), returns the Laplace-Stieltjes Transform expression of the corresponding node with a specific variable.

Usage

```
Lap(code, structure, outVar = "z", par = "value")
```

Arguments

code	genetic code (numeric vector) of the node (can be a leaf i.e. end by 0).
structure	object of class Mother (the structure).
outVar	output variable to be used ('z' by default).
par	Should the parameters be values ('value') or variables ('variable') ?

Details

For mother nodes, parameters are always called 'gamma' and for child nodes, parameters are always called 'alpha'. Furthermore, to recognize the parameters, the path is inserted at the end. For exemple, a child node with path (0,2,1) will have the parameter 'alpha021'.

Value

A character string giving the LST of the specified node.

Author(s)

Simon-Pierre Gadoury

See Also

InvLap

Examples

```
structure <- GEO(0.1, NULL, list(GAMMA(0.1, 1:2, NULL),
                           GAMMA(0.2, 3:4, NULL)))

Lap(c(0,2), structure, outVar = 'z', par = 'value')
```

LOG *Construction of a LOG Mother or Child Class Object*

Description

Constructs either a LOG Mother or Child class object for a given parameter, arguments, and nesting structure.

Usage

```
LOG(par, unif, structure)
```

Arguments

par	parameter of the distribution.
unif	uniform structure, a numeric vector of grouped numbers, i.e. c(1,2,3) is translated as being c(u1, u2, u3).
structure	nesting structure of the form X(par1, c(i,...), list(Y(par2, c(j,...), NULL), Z(par3, c(k,...), NULL))), where X, Y, and Z are compatible functions (see 'details'). It is good to note that if structure is NULL, the function will automatically be of class Child. For continuous distributions (i.e. GAMMA), structure is always NULL.

Author(s)

Simon-Pierre Gadoury

See Also

Other mother or child class objects.: GAMMA, GEO

Examples

```
LOG(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                    LOG(0.1, NULL, list(GAMMA(1/30, c(1.2), NULL),
                                        GAMMA(1/30, c(3,4), NULL)))))
```

Node *Obtain a node in mother class object*

Description

Use a path (numeric vector) to obtain a subgroup of a structure (mother class object).

Usage

```
Node(path, structure)
```

Arguments

path	the path of the node (numeric vector).
structure	a mother class object (S4).

Details

Every node of a mother object (structure) can be identified with a numeric vector that indicates the path used from the root to the node. The vector is the 'path' argument and is used to find specific nodes of a given structure. For a complete explanation, we refer to Cossette et al. (2017).

Value

Either a child or mother class object.

Author(s)

Simon-Pierre Gadoury

Examples

```
# We directly give the path of the desired node.
Node(c(0,2,2), LOG(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                              LOG(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                              GAMMA(1/30, c(3,4), NULL))))))

# Here we provide the path with the GeneticCodes function of this package.
structure <- LOG(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                                      LOG(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                                      GAMMA(1/30, c(3,4), NULL)))))
Node(GeneticCodes(structure)[[3]], structure)
```

pCompCop *Distribution function of Mother class objects*

Description

Distribution function of a Mother class object.

Usage

```
pCompCop(structure, vector = FALSE, express = TRUE)
```

Arguments

structure	object of class Mother.
vector	logical. If false, returns a function or a character string with (u_1, u_2, ...) as arguments, else, just (u).
express	logical. If false, returns a function, else, a character string.

Value

The distribution function in the form of either a function or a character string.

Examples

```
## Create the structure
structure <- LOG(0.5, NULL, list(GAMMA(1/30, c(5,6), NULL),
                              LOG(0.1, NULL, list(GAMMA(1/30, c(1,2), NULL),
                              GAMMA(1/30, c(3,4), NULL)))))

## Character string
pCompCop(structure, vector = TRUE, express = TRUE)
pCompCop(structure, vector = FALSE, express = TRUE)

## Function
pCompCop(structure, vector = TRUE, express = FALSE)
pCompCop(structure, vector = FALSE, express = FALSE)
```

pCop *Distribution function of archm class objects*

Description

Distribution function of an Archimedean copula (archm) class object.

Usage

```
pCop(copula, vector = FALSE, express = TRUE)
```

Arguments

copula	an Archimedean copula (archm) class object.
vector	logical. If false, returns a function or a character string with (u_1, u_2, ..., u_dim) as arguments, else, just (u).
express	logical. If false, returns a function, else, a character string.

Value

The distribution function in the form of either a function or a character string.

Author(s)

Simon-Pierre Gadoury

See Also

rCop, Clayton, AMH, Gumbel, Frank

Examples

```
cop <- Clayton(5, 2)
pCop(cop, vector = TRUE, express = TRUE)
pCop(cop, vector = FALSE, express = TRUE)
```

---

rCompCop *Random number generator for Mother class objects*

Description

Samples from a Mother class object.

Usage

```
rCompCop(n, structure)
```

Arguments

n	the number of realisations.
structure	an object of class Mother.

Value

A numeric matrix of sampled data from the structure

Author(s)

Simon-Pierre Gadoury

Examples

```
## Create the structure
structure <- GEO(0.1, 1, list(GAMMA(0.2, 2:3, NULL),
                        GEO(0.3, 4:5, NULL)))

## Sample from the structure
rCompCop(1000, structure)
```

---

rCop *Random number generator for Archimedean copula class objects*

Description

Random number generator for archm class objects.

Usage

```
rCop(n, copula)
```

Arguments

n	number of realisations.
copula	an Archimedean copula (archm) class object.

Details

For bivariate archm copula objects, the function uses the conditional approach. As for dimensions higher than 2, the Marshall-Olkin (1988) approach is chosen instead.

Value

A numeric matrix containing the samples.

Author(s)

Simon-Pierre Gadoury

See Also

pCop, Clayton, AMH, Frank, Gumbel

Examples

```
## Create the trivariate archm copula object
cop <- Clayton(5, 3)

## Generate the samples
res <- rCop(10000, cop)

## Plot the values
pairs(res, pch = 16, cex = 0.7)
```

Index