

The copula Package

May 20, 2007

Version 0.4-1

Date 2007/05/17

Title Multivariate Dependence with Copula

Author Jun Yan <jyan@stat.uiowa.edu>

Maintainer Jun Yan <jyan@stat.uiowa.edu>

Depends methods, mvtnorm, scatterplot3d, sn, adapt

Description Classes (S4) of commonly used copulas including elliptical, Archimidean, and extreme value copulas.

License GPL version 2 or later.

R topics documented:

Copula	2
Mvdc	3
archmCopula-class	4
archmCopula	6
AssocMeasures	7
contour-methods	7
copula-class	8
ellipCopula-class	9
ellipCopula	10
evCopula-class	11
evCopula	12
fitCopula-class	13
fitCopula	13
generator	15
mvdc-class	15
persp-methods	16
plackettCopula	16
show-methods	17
summary-methods	17

Copula *The Copula Distribution*

Description

Density, distribution function, and random generation for a "copula" object.

Usage

```
dcopula(copula, u)
pcopula(copula, u)
rcopula(copula, n)
```

Arguments

copula	a "copula" object.
u	a vector of the copula dimension or a matrix with number of rows being the copula dimension, giving the coordinates of the points where the density of distribution function need to be evaluated.
n	number of observations to be generated.

Details

The density function of an Archimedean copula is obtained by differentiating the distribution function symbolically using D.

The distribution function of a t copula uses pmvt from package mvtnorm. The density function of a t copula uses the dmst from package sn.

The random number generator for an Archimedean copula uses the conditional approach for bivariate case and the Marshal-Olkin (1988) approach for dimension greater than 2.

Value

'dcopula' gives the density, 'pcopula' gives the distribution function, and 'rcopula' generates random variates.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

References

Joe (1997) Nelsen (1999)

See Also

copulaObject, ellipCopula, archmCopula

Examples

```

norm.cop <- normalCopula(0.5)
norm.cop
x <- rcopula(norm.cop, 100)
plot(x)
dcopula(norm.cop, x)
pcopula(norm.cop, x)
persp(norm.cop, dcopula)
contour(norm.cop, pcopula)
## a 3-dimensional normal copula
u <- rcopula(normalCopula(0.5, dim = 3), 1000)
## scatterplot3d(u)
## a 3-dimensional clayton copula
v <- rcopula(claytonCopula(2, dim = 3), 1000)
## scatterplot3d(v)

```

Mvdc

Multivariate Distribution via Copula

Description

Density, distribution function, and random generator for a multivariate distribution via copula.

Usage

```

mvdc(copula, margins, paramMargins)
dmvdc(mvdc, x)
pmvdc(mvdc, x)
rmvdc(mvdc, n)

```

Arguments

copula	an object of copula.
margins	a character vector specifying all the marginal distributions. See details below.
paramMargins	a list of list with names components, giving the parameter values of the marginal distributions. See details below.
mvdc	a mvdc object.
x	a vector of the copula dimension or a matrix with number of rows being the copula dimension, giving the coordinates of the points where the density of distribution function need to be evaluated.
n	number of observations to be generated.

Details

The characters in argument `margins` are used to construct function names of density, distribution, and quantile functions. For example, "norm" can be used to specify marginal distribution, because "dnorm", "pnorm", and "qnorm" are all available.

Each component list in argument `paramMargins` is a list with named component which are used to specify the parameters of the marginal distributions. For example, `paramMargins = list(list(mean = 0, sd = 2), list(rate = 2))` can be used to specify that the first margin is normal with mean 0 and sd 2, and the second margin is exponential with rate 2.

Value

'mvdc' constructs an object of class "mvdc". 'dmvdc' gives the density, 'pmvdc' gives the distribution function, and 'rmvdc' generates random variates.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[ellipCopula](#), [archmCopula](#), [mvdc-class](#), [copula-class](#)

Examples

```
## construct a bivariate distribution whose marginals
## are normal and exponential respectively, coupled
## together via a normal copula
x <- mvdc(normalCopula(0.75), c("norm", "exp"),
          list(list(mean = 0, sd = 2), list(rate = 2)))
x.samp <- rmvdc(x, 100)
dmvdc(x, x.samp)
pmvdc(x, x.samp)
persp(x, dmvdc, xlim = c(-4, 4), ylim=c(0, 1))
```

archmCopula-class *Class "archmCopula"*

Description

Archimedean copula

Objects from the Class

Objects can be created by calls of the form `new("archmCopula", ...)` or by function 'archmCopula'.

Slots

exprdist: Object of class "expression", expressions for the cdf and pdf of the copula.
These expressions are used in function 'pcopula' and 'dcopula'.

dimension: Object of class "numeric"

parameters: Object of class "numeric"

param.names: Object of class "character"

param.lowbnd: Object of class "numeric"

param.upbnd: Object of class "numeric"

message: Object of class "character"

Methods

dcopula signature (copula = "claytonCopula"): ...

pcopula signature (copula = "claytonCopula"): ...

rcopula signature (copula = "claytonCopula"): ...

dcopula signature (copula = "frankCopula"): ...

pcopula signature (copula = "frankCopula"): ...

rcopula signature (copula = "frankCopula"): ...

dcopula signature (copula = "gumbelCopula"): ...

pcopula signature (copula = "gumbelCopula"): ...

rcopula signature (copula = "gumbelCopula"): ...

dcopula signature (copula = "amhCopula"): ...

pcopula signature (copula = "amhCopula"): ...

rcopula signature (copula = "amhCopula"): ...

Extends

Class "archmCopula" extends class "copula" directly. Class "claytonCopula", "frankCopula", "gumbelCopula" and "amhCopula" extends class "archmCopula" directly.

Note

The expressions of pdf are obtained by differentiating the cdf expression using function 'D'.
"gumbelCopula" is also of class "evCopula".

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[archmCopula](#), [copula-class](#).

Description

Constructs an Archimedean copula class object with its corresponding parameter and dimension.

Usage

```
archmCopula(family, param, dim = 2, ...)  
claytonCopula(param, dim = 2)  
frankCopula(param, dim = 2)  
gumbelCopula(param, dim = 2)  
amhCopula(param, dim = 2)
```

Arguments

family	a character string specifying the family of an Archimedean copula. Implemented families are "clayton", "frank", and "gumbel".
param	a numeric vector specifying the parameter values.
dim	the dimension of the copula.
...	currently nothing.

Value

An Archimedean copula object of class "claytonCopula", "frankCopula", "gumbelCopula", or "amhCopula".

Note

"archmCopula" is a wrapper for "claytonCopula", "frankCopula", "gumbelCopula" and "amhCopula".

For $\text{dim} > 6$, the expression of pdf is not available due to intensive computing involved in symbolically differentiating the cdf. Therefore, for $\text{dim} > 6$, likelihood estimation cannot be done yet.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[ellipCopula](#)

Examples

```

clayton.cop <- claytonCopula(2, dim = 3)
##scatterplot3d(rcopula(clayton.cop, 1000))
frank.cop <- frankCopula(3)
persp(frank.cop, dcopula)
gumbel.cop <- archmCopula("gumbel", 5)
contour(gumbel.cop, dcopula)

```

AssocMeasures

Association Measures in Package 'copula'

Description

Methods to compute association measures such as Kendall's Tau, Spearman's Rho, and tail dependence index for bivariate copulas. Calibration is supplied for Kendall's Tau and Spearman's Rho.

Methods

copula = "copula" Association measure for a "copula" object.

contour-methods

Methods for Function contour in Package 'copula'

Description

Methods for function `contour` in package **copula**

Details

When `x` is of class "copula", these arguments can be supplied: `fun`: the function to be plotted, "dcopula" or "pcopula". `n = 51`: the number of points to do the plotting. `theta = -30`, `phi = 30`, `expand = 0.618`: arguments for "contour"

when `x` is of class "mvdc", these arguments are expected to replace the effect of `n = 51`: `xlim`: the range of the `x` variable. `ylim`: the range of the `y` variable. `nx`: the number of points for `x` to expand. `ny`: the number of points for `y` to expand.

Methods

x = "copula" Contour plot for a "copula" object.

x = "mvdc" Contour plot for a "mvdc" object.

Examples

```

contour(frankCopula(-0.8), dcopula)
contour(claytonCopula(2), pcpopula)
x <- mvdc(gumbelCopula(3), c("norm", "norm"),
          list(list(mean = 0, sd = 1), list(mean = 1)))
contour(x, dmvc, xlim=c(-2, 2), ylim=c(-1, 3))
contour(x, pmvc, xlim=c(-2, 2), ylim=c(-1, 3))

```

copula-class *Class "copula"*

Description

A class of multivariate distribution with uniform margins.

Objects from the Class

Objects can be created by calls of the form `new("copula", ...)`.

Slots

dimension: Object of class "numeric", dimension of the copula.
parameters: Object of class "numeric", parameter values.
param.names: Object of class "character", parameter names.
param.lowbnd: Object of class "numeric", parameter lower bounds.
param.upbnd: Object of class "numeric", parameter upper bounds.
message: Object of class "character", families of the copula.

Warning

This implementation is still at the experimental stage and is subject to change during the development.

Note

The 'copula' class is extended by the 'evCopula', 'archmCopula' and 'ellipCopula' classes. Objects of implemented copulas can be created from functions 'evCopula', 'archmCopula' and 'ellipcopula'.

"plackettCopula" is a special type of copula which does not belong to either one of the three classes above.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[archmCopula-class](#), [ellipCopula-class](#) [evCopula-class](#)

ellipCopula-class *Class "ellipCopula"*

Description

Copulas generated from elliptical multivariate distributions.

Objects from the Class

Objects can be created by calls of the form `new("ellipCopula", ...)`, or by function `'ellipCopula'`.

Slots

dispstr: Object of class "character", indicating the type of the dispersion matrix such as 'ex', 'ar1', 'toep', or 'un'.

dimension: Object of class "numeric"

parameters: Object of class "numeric"

param.names: Object of class "character"

param.lowbnd: Object of class "numeric"

param.upbnd: Object of class "numeric"

message: Object of class "character"

Extends

Class "ellipCopula" extends class "copula" directly. Class "normalCopula" and "tCopula" extends class "ellipCopula" directly.

Methods

dcopula signature(copula = "normalCopula"): ...

pcopula signature(copula = "normalCopula"): ...

rcopula signature(copula = "normalCopula"): ...

dcopula signature(copula = "tCopula"): ...

pcopula signature(copula = "tCopula"): ...

rcopula signature(copula = "tCopula"): ...

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[ellipCopula](#), [copula-class](#).

Description

Constructs an elliptical copula class object with its corresponding parameter and dimension.

Usage

```
ellipCopula(family, param, dim = 2, dispstr = "ex", df = 5, ...)  
normalCopula(param, dim = 2, dispstr = "ex")  
tCopula(param, dim = 2, dispstr = "ex", df = 5)
```

Arguments

family	a character string specifying the family of an elliptical copula. Implemented families are "normal" and "t".
param	a numeric vector specifying the parameter values.
dim	the dimension of the copula.
dispstr	a character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable, "ar1" for AR(1), "toep" for toeplitz, , and "un" for unstructured. For normal copula, this defines the structure of the correlation matrix.
df	a numerical value specifying the degree of freedom for the multivariate t distribution used to construct the t copulas.
...	currently nothing.

Value

An elliptical copula object of class "normalCopula" or "tCopula".

Note

"ellipCopula" is a wrapper for "normalCopula" and "tCopula".

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[archmCopula](#)

Examples

```

norm.cop <- normalCopula(c(0.5, 0.6, 0.7), dim = 3, dispstr = "un")
t.cop <- tCopula(c(0.5, 0.3), dim = 3, dispstr = "toep", df = 2)
## from the wrapper
norm.cop <- ellipCopula("normal", param = c(0.5, 0.6, 0.7),
                        dim = 3, dispstr = "un")
## 3d scatter plot of 1000 random observations
##scatterplot3d(rcopula(norm.cop, 1000))
##scatterplot3d(rcopula(t.cop, 1000))

```

evCopula-class *Class "evCopula"*

Description

Extreme value copula

Objects from the Class

Objects can be created by calls of the form `new("evCopula", ...)` or by function `'evCopula'`.

Slots

dimension: Object of class "numeric"
parameters: Object of class "numeric"
param.names: Object of class "character"
param.lowbnd: Object of class "numeric"
param.upbnd: Object of class "numeric"
message: Object of class "character"

Methods

dcopula signature(copula = "galambosCopula"): ...
pcopula signature(copula = "galambosCopula"): ...
rcopula signature(copula = "galambosCopula"): ...
dcopula signature(copula = "huslerReissCopula"): ...
pcopula signature(copula = "huslerReissCopula"): ...
rcopula signature(copula = "huslerReissCopula"): ...

Extends

Class "evCopula" extends class "copula" directly. Class "galambosCopula" and "huslerReissCopula" extends class "evCopula" directly.

Note

The expressions of pdf are obtained by differentiating the cdf expression using function 'deriv'.
"gumbelCopula" is also of class "archmCopula".

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[evCopula](#), [copula-class](#).

evCopula

Construction of Extreme Value Copula Class Object

Description

Constructs an extreme value copula class object with its corresponding parameter.

Usage

```
evCopula(family, param, dim = 2, ...)  
galambosCopula(param)  
huslerReissCopula(param)
```

Arguments

family	a character string specifying the family of an extreme value copula. Implemented families are "galambos" and "gumbel".
param	a numeric vector specifying the parameter values.
dim	the dimension of the copula.
...	currently nothing.

Value

An extreme value copula object of class "galambosCopula" "gumbelCopula", or "huslerReissCopula".

Note

Gumbel copula is both Archimedean copula and extreme value copula.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[ellipCopula](#), [archmCopula](#)

Examples

```
gumbel.cop <- evCopula("gumbel", param=2)
contour(gumbel.cop, dcopula)
galambos.cop <- galambosCopula(2)
contour(galambos.cop, dcopula)
```

fitCopula-class *Class "fitCopula"*

Description

Classes and summaries for fitting copula models.

Objects from the Class

Objects can be created by calls of the form `fitCopula`, `fitMvdc` or their summary methods.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

fitCopula *Maximum Likelihood Estimation of Copula Models*

Description

Fit a copula model to multivariate data.

Usage

```
loglikCopula(param, x, copula)
loglikMvdc(param, x, mvdc)
fitCopula(data, copula, start, lower=NULL, upper=NULL,
           optim.control = list(NULL), method = "BFGS")
fitMvdc(data, mvdc, start, optim.control = list(NULL), method = "BFGS")
```

Arguments

<code>param</code>	a vector of parameter values
<code>x</code>	a data matrix
<code>copula</code>	a 'copula' object
<code>mvdc</code>	a 'mvdc' object
<code>data</code>	a data matrix
<code>start</code>	a vector of starting value for param
<code>lower, upper</code>	bounds on the variables for the "L-BFGS-B" method.
<code>optim.control</code>	a list of control to be passed to optim
<code>method</code>	the method for optim

Value

The return values of 'loglikCopula' and 'loglikMvdc' are the loglikelihood evaluated at the given value of 'param'.

The return values of 'fitCopula' and 'fitMvdc' are an object of class 'fitCopula' and 'fitMvdc', respectively, containing slots:

<code>est</code>	the estimate of the parameters
<code>var.est</code>	variance matrix of the estimate
<code>loglik</code>	loglikelihood at est
<code>fit</code>	the result of optim

Note

When covariates are available for marginal distributions or copula, one can construct loglikelihood function and feed it to optim to estimate all the parameters.

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

References

Yan (2006) Multivariate Modeling with Copulas and Engineering Applications. In Handbook of Engineering Statistics, Ed. Pham, Springer.

See Also

[Copula](#), [mvdc](#)

Examples

```

gmb <- gumbelCopula(3, dim=2)
myMvd <- mvdc(gmb, c("exp","exp"), list(list(rate=2),list(rate=4)))
x <- rmvdc(myMvd, 1000)
fit <- fitMvdc(x, myMvd, c(1,1,2))
fit

```

generator

*Generators for Archimedean and EV Copulas***Description**

Methods function to evaluate generator function, inverse generator function, and derivatives of generator function.

Methods

copula = "copula" Association measure for a "copula" object.

mvdc-class

*Class "mvdc"***Description**

A class of multivariate distribution via copula.

Objects from the Class

Objects can be created by calls of the form `new("mvdc", ...)` or by function `mvdc`.

Slots

copula: Object of class "copula", specifying the copula.

margins: Object of class "character", specifying the marginal distributions.

paramMargins: Object of class "list", a list of list, with eaching list giving the names parameter values of the margins.

Methods

contour signature(x = "mvdc"):...

persp signature(x = "mvdc"):...

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also[mvdc](#)

`persp-methods`*Methods for Function persp in Package 'copula'*

DescriptionMethods for function `persp` in package **copula****Details**

When `x` is of class `"copula"`, these arguments can be supplied: `fun`: the function to be plotted, `"dcopula"` or `"pcopula"`. `n = 51`: the number of points to do the plotting. `theta = -30`, `phi = 30`, `expand = 0.618`: arguments for `"persp"`

when `x` is of class `"mvdc"`, two more arguments are expected to replace the effect of `n = 51`: `xlim`: the range of the `x` variable. `ylim`: the range of the `y` variable.

Methods`x = "copula"` Perspective plot for a `"copula"` object.`x = "mvdc"` Perspective plot for a `"mvdc"` object.**Examples**

```
persp(frankCopula(-0.8), dcopula)
persp(claytonCopula(2), pcopula)
x <- mvdc(gumbelCopula(3), c("norm", "norm"),
          list(list(mean = 0, sd = 1), list(mean = 1)))
persp(x, dmvc, xlim=c(-2, 2), ylim=c(-1, 3))
persp(x, pmvc, xlim=c(-2, 2), ylim=c(-1, 3))
```

`plackettCopula`*Construction of Plackett Copula Class Object*

Description

Constructs an Plackett copula class object with its corresponding parameter.

Usage`plackettCopula(param)`**Arguments**`param` a numeric vector specifying the parameter values.

Value

A Plackett copula object of class "plackettCopula".

Author(s)

Jun Yan <jyan@stat.uiowa.edu>

See Also

[ellipCopula](#), [archmCopula](#)

Examples

```
plackett.cop <- plackettCopula(param=2)
tailIndex(plackett.cop)
```

show-methods

Methods for Function show in Package 'copula'

Description

Methods for function `show` in package **copula**.

Methods

object = "copula" describe this method here

object = "fitCopula" describe this method here

object = "fitMvdc" describe this method here

Examples

summary-methods

Methods for Function summary in Package 'copula'

Description

Methods for function `summary` in package **copula**.

Methods

object = "fitCopula" describe this method here

object = "fitMvdc" describe this method here

Examples

Index

*Topic **classes**

archmCopula-class, 4
copula-class, 7
ellipCopula-class, 8
evCopula-class, 11
fitCopula-class, 13
mvdc-class, 15

*Topic **distribution**

archmCopula, 5
Copula, 1
ellipCopula, 9
evCopula, 12
Mvdc, 3
plackettCopula, 16

*Topic **hplot**

contour-methods, 7
persp-methods, 15

*Topic **methods**

AssocMeasures, 6
contour-methods, 7
generator, 14
persp-methods, 15
show-methods, 17
summary-methods, 17

*Topic **models**

fitCopula, 13

*Topic **multivariate**

archmCopula, 5
AssocMeasures, 6
Copula, 1
ellipCopula, 9
evCopula, 12
fitCopula, 13
generator, 14
Mvdc, 3
plackettCopula, 16

*Topic **print**

show-methods, 17

Afun (*generator*), 14

Afun, galambosCopula-method
(*generator*), 14

Afun, gumbelCopula-method
(*generator*), 14

Afun, huslerReissCopula-method
(*generator*), 14

Afun-methods (*generator*), 14

amhCopula (*archmCopula*), 5

amhCopula-class
(*archmCopula-class*), 4

archmCopula, 4, 5, 5, 10, 12, 16

archmCopula-class, 8

archmCopula-class, 4

AssocMeasures, 6

calibKendallsTau (*AssocMeasures*),
6

calibKendallsTau, ANY-method
(*AssocMeasures*), 6

calibKendallsTau, claytonCopula-method
(*AssocMeasures*), 6

calibKendallsTau, copula-method
(*AssocMeasures*), 6

calibKendallsTau, ellipCopula-method
(*AssocMeasures*), 6

calibKendallsTau, normalCopula-method
(*AssocMeasures*), 6

calibKendallsTau, tCopula-method
(*AssocMeasures*), 6

calibKendallsTau-methods
(*AssocMeasures*), 6

calibSpearmanRho
(*AssocMeasures*), 6

calibSpearmanRho, ANY-method
(*AssocMeasures*), 6

calibSpearmanRho, copula-method
(*AssocMeasures*), 6

calibSpearmanRho, ellipCopula-method
(*AssocMeasures*), 6

- calibSpearmanRho, normalCopula-method *fitMvdc*-class (*fitCopula*-class),
(*AssocMeasures*), 6
- calibSpearmanRho, tCopula-method
(*AssocMeasures*), 6
- calibSpearmanRho-methods
(*AssocMeasures*), 6
- claytonCopula (*archmCopula*), 5
- claytonCopula-class
(*archmCopula*-class), 4
- contour, copula-method
(*contour-methods*), 7
- contour, mvdc-method
(*contour-methods*), 7
- contour-methods, 7
- Copula, 1, 14
- copula-class, 4, 5, 9, 11
- copula-class, 7

- dcopula (*Copula*), 1
- dcopula, amhCopula-method
(*Copula*), 1
- dcopula, claytonCopula-method
(*Copula*), 1
- dcopula, frankCopula-method
(*Copula*), 1
- dcopula, galambosCopula-method
(*Copula*), 1
- dcopula, gumbelCopula-method
(*Copula*), 1
- dcopula, huslerReissCopula-method
(*Copula*), 1
- dcopula, normalCopula-method
(*Copula*), 1
- dcopula, plackettCopula-method
(*Copula*), 1
- dcopula, tCopula-method (*Copula*), 1
- dmvdc (*Mvdc*), 3

- ellipCopula, 4, 6, 9, 9, 12, 16
- ellipCopula-class, 8
- ellipCopula-class, 8
- evCopula, 11, 12
- evCopula-class, 8
- evCopula-class, 11

- fitCopula, 13
- fitCopula-class, 13
- fitMvdc (*fitCopula*), 13

- fitMvdc-class (*fitCopula*-class),
13
- frankCopula (*archmCopula*), 5
- frankCopula-class
(*archmCopula*-class), 4

- galambosCopula (*evCopula*), 12
- galambosCopula-class
(*evCopula*-class), 11
- generator, 14
- genFun (*generator*), 14
- genFun, amhCopula-method
(*generator*), 14
- genFun, claytonCopula-method
(*generator*), 14
- genFun, frankCopula-method
(*generator*), 14
- genFun, gumbelCopula-method
(*generator*), 14
- genFun-methods (*generator*), 14
- genFunDer1 (*generator*), 14
- genFunDer1, amhCopula-method
(*generator*), 14
- genFunDer1, claytonCopula-method
(*generator*), 14
- genFunDer1, frankCopula-method
(*generator*), 14
- genFunDer1, gumbelCopula-method
(*generator*), 14
- genFunDer1-methods (*generator*), 14
- genFunDer2 (*generator*), 14
- genFunDer2, amhCopula-method
(*generator*), 14
- genFunDer2, claytonCopula-method
(*generator*), 14
- genFunDer2, frankCopula-method
(*generator*), 14
- genFunDer2, gumbelCopula-method
(*generator*), 14
- genFunDer2-methods (*generator*), 14
- genInv (*generator*), 14
- genInv, amhCopula-method
(*generator*), 14
- genInv, claytonCopula-method
(*generator*), 14
- genInv, frankCopula-method
(*generator*), 14
- genInv, gumbelCopula-method
(*generator*), 14

- genInv-methods (*generator*), 14
- gumbelCopula (*archmCopula*), 5
- gumbelCopula-class
 (*archmCopula-class*), 4

- huslerReissCopula (*evCopula*), 12
- huslerReissCopula-class
 (*evCopula-class*), 11

- kendallsTau (*AssocMeasures*), 6
- kendallsTau, amhCopula-method
 (*AssocMeasures*), 6
- kendallsTau, ANY-method
 (*AssocMeasures*), 6
- kendallsTau, archmCopula-method
 (*AssocMeasures*), 6
- kendallsTau, claytonCopula-method
 (*AssocMeasures*), 6
- kendallsTau, copula-method
 (*AssocMeasures*), 6
- kendallsTau, frankCopula-method
 (*AssocMeasures*), 6
- kendallsTau, gumbelCopula-method
 (*AssocMeasures*), 6
- kendallsTau, normalCopula-method
 (*AssocMeasures*), 6
- kendallsTau, tCopula-method
 (*AssocMeasures*), 6
- kendallsTau-methods
 (*AssocMeasures*), 6

- loglikCopula (*fitCopula*), 13
- loglikMvdc (*fitCopula*), 13

- Mvdc, 3
- mvdc, 14, 15
- mvdc (*Mvdc*), 3
- mvdc-class, 4
- mvdc-class, 15

- normalCopula (*ellipCopula*), 9
- normalCopula-class
 (*ellipCopula-class*), 8

- pcopula (*Copula*), 1
- pcopula, amhCopula-method
 (*Copula*), 1
- pcopula, claytonCopula-method
 (*Copula*), 1
- pcopula, frankCopula-method
 (*Copula*), 1
- pcopula, galambosCopula-method
 (*Copula*), 1
- pcopula, gumbelCopula-method
 (*Copula*), 1
- pcopula, huslerReissCopula-method
 (*Copula*), 1
- pcopula, normalCopula-method
 (*Copula*), 1
- pcopula, plackettCopula-method
 (*Copula*), 1
- pcopula, tCopula-method (*Copula*), 1
- persp, copula-method
 (*persp-methods*), 15
- persp, mvdc-method
 (*persp-methods*), 15
- persp-methods, 15
- plackettCopula, 16
- plackettCopula-class
 (*copula-class*), 7
- pmvdc (*Mvdc*), 3

- rcopula (*Copula*), 1
- rcopula, amhCopula-method
 (*Copula*), 1
- rcopula, claytonCopula-method
 (*Copula*), 1
- rcopula, frankCopula-method
 (*Copula*), 1
- rcopula, galambosCopula-method
 (*Copula*), 1
- rcopula, gumbelCopula-method
 (*Copula*), 1
- rcopula, huslerReissCopula-method
 (*Copula*), 1
- rcopula, normalCopula-method
 (*Copula*), 1
- rcopula, plackettCopula-method
 (*Copula*), 1
- rcopula, tCopula-method (*Copula*), 1
- rmvdc (*Mvdc*), 3

- show, copula-method
 (*show-methods*), 17
- show, fitCopula-method
 (*show-methods*), 17
- show, fitMvdc-method
 (*show-methods*), 17

show, normalCopula-method
 (*show-methods*), 17
 show, tCopula-method
 (*show-methods*), 17
 show-methods, 17
 spearmanRho (*AssocMeasures*), 6
 spearmanRho, ANY-method
 (*AssocMeasures*), 6
 spearmanRho, copula-method
 (*AssocMeasures*), 6
 spearmanRho, frankCopula-method
 (*AssocMeasures*), 6
 spearmanRho, normalCopula-method
 (*AssocMeasures*), 6
 spearmanRho, tCopula-method
 (*AssocMeasures*), 6
 spearmanRho-methods
 (*AssocMeasures*), 6
 summary, fitCopula-method
 (*summary-methods*), 17
 summary, fitMvdc-method
 (*summary-methods*), 17
 summary-methods, 17
 summaryFitCopula-class
 (*fitCopula-class*), 13
 summaryFitMvdc-class
 (*fitCopula-class*), 13

 tailIndex (*AssocMeasures*), 6
 tailIndex, ANY-method
 (*AssocMeasures*), 6
 tailIndex, claytonCopula-method
 (*AssocMeasures*), 6
 tailIndex, copula-method
 (*AssocMeasures*), 6
 tailIndex, evCopula-method
 (*AssocMeasures*), 6
 tailIndex, frankCopula-method
 (*AssocMeasures*), 6
 tailIndex, gumbelCopula-method
 (*AssocMeasures*), 6
 tailIndex, normalCopula-method
 (*AssocMeasures*), 6
 tailIndex, tCopula-method
 (*AssocMeasures*), 6
 tailIndex-methods
 (*AssocMeasures*), 6
 tCopula (*ellipCopula*), 9

 tCopula-class
 (*ellipCopula-class*), 8