Package 'CDVine'

July 2, 2012

Type Package

Title Statistical inference of C- and D-vine copulas

Version 1.1-8

Date 2012-07-02

Author Ulf Schepsmeier, Eike Christian Brechmann

Maintainer Ulf Schepsmeier <schepsme@ma.tum.de>

Depends R (>= 2.11.0), MASS, mvtnorm, igraph

Description This package provides functions for statistical inference
of canonical vine (C-vine) and D-vine copulas. It contains
tools for bivariate exploratory data analysis and for bivariate
as well as vine copula selection. Models can be estimated
either sequentially or by joint maximum likelihood estimation.
Sampling algorithms and plotting methods are also included.
Data is assumed to lie in the unit hypercube (so-called copula data).

License GPL (>= 2)

LazyLoad yes

R topics documented:

CDVine-package   *Statistical inference of C- and D-vine copulas*

Description

This package provides functions for statistical inference of canonical vine (C-vine) and D-vine copulas. It contains tools for bivariate exploratory data analysis and for bivariate as well as vine copula selection. Models can be estimated either sequentially or by joint maximum likelihood estimation. Sampling algorithms and plotting methods are also included. Data is assumed to lie in the unit hypercube (so-called copula data).

Details

Package:	CDVine
Type:	Package
Version:	1.1-8
Date:	2012-07-02
License:	GPL (>=2)
Depends:	R ($\geq$ 2.11.0), MASS, mvtnorm, igraph
LazyLoad:	yes

Bivariate copula families

In this package several bivariate copula families are included for bivariate analysis as well as for multivariate analysis using vine copulas. It provides functionality of elliptical (Gaussian and Student t) as well as Archimedean (Clayton, Gumbel, Frank, Joe, BB1, BB6, BB7 and BB8) copulas to cover a large bandwidth of possible dependence structures. For the Archimedean copula families rotated versions are included to cover negative dependence too. The two parameter BB1, BB6, BB7 and BB8 copulas are however numerically instable for large parameters, in particular, if BB6, BB7 and BB8 copulas are close to the Joe copula which is a boundary case of these three copula families. In general, the user should be careful with extreme parameter choices.

The following table shows the parameter ranges of bivariate copula families with parameters `par` and `par2`:

Copula family	par	par2
Gaussian	$(-1, 1)$	-
Student t	$(-1, 1)$	$(2, \infty)$
(Survival) Clayton	$(0, \infty)$	-
(Survival) Gumbel	$[1, \infty)$	-
Frank	$R \backslash \{0\}$	-
(Survival) Joe	$(1, \infty)$	-
Rotated Clayton (90 and 270 degrees)	$(-\infty, 0)$	-
Rotated Gumbel (90 and 270 degrees)	$(-\infty, -1]$	-
Rotated Joe (90 and 270 degrees)	$(-\infty, -1)$	-
(Survival) Clayton-Gumbel (BB1)	$(0, \infty)$	$[1, \infty)$
(Survival) Joe-Gumbel (BB6)	$[1, \infty)$	$[1, \infty)$
(Survival) Joe-Clayton (BB7)	$[1, \infty)$	$(0, \infty)$
(Survival) Joe-Frank (BB8)	$[1, \infty)$	$(0, 1]$
Rotated Clayton-Gumbel (90 and 270 degrees)	$(-\infty, 0)$	$-\infty, -1]$
Rotated Joe-Gumbel (90 and 270 degrees)	$(-\infty, -1]$	$(-\infty, -1]$
Rotated Joe-Clayton (90 and 270 degrees)	$(-\infty, -1]$	$(-\infty, 0)$
Rotated Joe-Frank (90 and 270 degrees)	$(-\infty, -1]$	$[-1, 0)$

C- and D-vine copula models

When specifying C- and D-vine copula models, one has to select an order of the variables. For a D-vine the order of the variables in the first tree has to be chosen and for a C-vine the root nodes for each tree need to be determined. Functions for inference of C- and D-vine copula models in this package assume that the order of the variables in the data set under investigation exactly corresponds to this C- or D-vine order. E.g., in a C-vine the first column of a data set is the first root node, the second column the second root node, etc. According to this order arguments have to be provided to functions for C- and D-vine copula inference. After choosing the type of the vine model, the copula families (`family`) and parameters (`par` and `par2`) have to be specified as vectors of length $d(d-1)/2$, where d is the number of variables. In a C-vine, the entries of this vector correspond to the following pairs and associated pair-copula terms

$(1, 2), (1, 3), (1, 4), ..., (1, d), (2, 3|1), (2, 4|1), ..., (2, d|1), (3, 4|1, 2), (3, 5|1, 2), ..., (3, d|1, 2), ..., (d-1, d|1, ..., d-2)$.

Similarly, the pairs of a D-vine are denoted in the following order:

$(1, 2), (2, 3), (3, 4), ..., (d-1, d), (1, 3|2), (2, 4|3), ..., (d-2, d|d-1), (1, 4|2, 3), (2, 5|3, 4), ..., (d-3, d|d-2, d-1), ..., (1, d|2, ..., d-1)$.

Acknowledgment

grateful to Harry Joe for his contributions to the implementation of the bivariate Archimedean copulas.

Author(s)

Ulf Schepsmeier, Eike Christian Brechmann <CDVine@ma.tum.de>

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

Bedford, T. and R. M. Cooke (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. Annals of Mathematics and Artificial intelligence 32, 245-268.

Bedford, T. and R. M. Cooke (2002). Vines - a new graphical model for dependent random variables. Annals of Statistics 30, 1031-1068.

Brechmann, E. C., C. Czado, and K. Aas (2012). Truncated regular vines in high dimensions with applications to financial data. Canadian Journal of Statistics 40 (1), 68-85.

Brechmann, E. C. and C. Czado (2011). Risk Management with High-Dimensional Vine Copulas: An Analysis of the Euro Stoxx 50. Submitted for publication. http://mediatum.ub.tum.de/doc/1079276/1079276.pdf.

Czado, C., U. Schepsmeier, and A. Min (2012). Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling, 12 (3), 229-255.

Dissmann, J. F., E. C. Brechmann, C. Czado, and D. Kurowicka (2011). Selecting and estimating regular vine copulae and application to financial returns. Submitted for publication. http://mediatum.ub.tum.de/doc/1079277/1079277.pdf

Joe, H. (1996). Families of m-variate distributions with given margins and m(m-1)/2 bivariate dependence parameters. In L. Rueschendorf, B. Schweizer, and M. D. Taylor (Eds.), Distributions with fixed marginals and related topics, pp. 120-141. Hayward: Institute of Mathematical Statistics.

Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman and Hall, London.

Knight, W. R. (1966). A computer method for calculating Kendall's tau with ungrouped data. Journal of the American Statistical Association 61 (314), 436-439.

Kurowicka, D. and R. M. Cooke (2006). Uncertainty Analysis with High Dimensional Dependence Modelling. Chichester: John Wiley.

Kurowicka, D. and H. Joe (Eds.) (2011). DEPENDENCE MODELING: Vine Copula Handbook. Singapore: World Scientific Publishing Co.

BiCopCDF   *Distribution function of a bivariate copula*

Description

This function evaluates the cumulative distribution function (CDF) of a given parametric bivariate copula.

Usage

```
BiCopCDF(u1, u2, family, par, par2=0)
```

Arguments

`u1,u2`	Numeric vectors of equal length with values in [0,1].
`family`	An integer defining the bivariate copula family:

0 = independence copula
1 = Gaussian copula
2 = Student t copula (t-copula)
3 = Clayton copula
4 = Gumbel copula
5 = Frank copula
6 = Joe copula
7 = BB1 copula
8 = BB6 copula
9 = BB7 copula
10 = BB8 copula
13 = rotated Clayton copula (180 degrees; "survival Clayton")
14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
16 = rotated Joe copula (180 degrees; "survival Joe")
17 = rotated BB1 copula (180 degrees; "survival BB1")
18 = rotated BB6 copula (180 degrees; "survival BB6")
19 = rotated BB7 copula (180 degrees; "survival BB7")
20 = rotated BB8 copula (180 degrees; "survival BB8")
23 = rotated Clayton copula (90 degrees)
24 = rotated Gumbel copula (90 degrees)
26 = rotated Joe copula (90 degrees)
27 = rotated BB1 copula (90 degrees)
28 = rotated BB6 copula (90 degrees)
29 = rotated BB7 copula (90 degrees)
30 = rotated BB8 copula (90 degrees)
33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

`par`	Copula parameter.
`par2`	Second parameter for bivariate copulas with two parameters (t, BB1, BB6, BB7, BB8; default: `par2 = 0`).

Value

A numeric vector of the bivariate copula distribution function evaluated at u1 and u2.

Author(s)

Eike Brechmann

See Also

[BiCopPDF](#), [BiCopHfunc](#), [BiCopSim](#)

Examples

```
# simulate from a bivariate t-copula
simdata = BiCopSim(300,2,-0.7,par2=4)

# evaluate the distribution function of the bivariate t-copula
u1 = simdata[,1]
u2 = simdata[,2]
BiCopCDF(u1,u2,2,-0.7,par2=4)
```

BiCopChiPlot   *Chi-plot for bivariate copula data*

Description

This function creates a chi-plot of given bivariate copula data.

Usage

```
BiCopChiPlot(u1, u2, PLOT=TRUE, mode="NULL", ...)
```

Arguments

u1,u2	Data vectors of equal length with values in [0,1].
PLOT	Logical; whether the results are plotted. If PLOT = FALSE, the values lambda, chi and control.bounds are returned (see below; default: PLOT = TRUE).
mode	Character; whether a general, lower or upper chi-plot is calculated. Possible values are mode = "NULL", "upper" and "lower". "NULL" = general chi-plot (default) "upper" = upper chi-plot "lower" = lower chi-plot
...	Additional plot arguments.

Details

For observations $u_{i,j}, \; i = 1, ..., N, \; j = 1, 2$, the chi-plot is based on the following two quantities: the chi-statistics

$$\chi_i = \frac{\hat{F}_{U_1 U_2}(u_{i,1}, u_{i,2}) - \hat{F}_{U_1}(u_{i,1})\hat{F}_{U_2}(u_{i,2})}{\sqrt{\hat{F}_{U_1}(u_{i,1})(1 - \hat{F}_{U_1}(u_{i,1}))\hat{F}_{U_2}(u_{i,2})(1 - \hat{F}_{U_2}(u_{i,2}))}},$$

and the lambda-statistics

$$\lambda_i = 4 sgn\left(\tilde{F}_{U_1}(u_{i,1}), \tilde{F}_{U_2}(u_{i,2})\right) \cdot \max\left(\tilde{F}_{U_1}(u_{i,1})^2, \tilde{F}_{U_2}(u_{i,2})^2\right),$$

where $\hat{F}_{U_1}$, $\hat{F}_{U_2}$ and $\hat{F}_{U_1 U_2}$ are the empirical distribution functions of the uniform random variables U_1 and U_2 and of (U_1, U_2), respectively. Further, $\tilde{F}_{U_1} = \hat{F}_{U_1} - 0.5$ and $\tilde{F}_{U_2} = \hat{F}_{U_2} - 0.5$.

These quantities only depend on the ranks of the data and are scaled to the interval $[0, 1]$. λ_i measures a distance of a data point $(u_{i,1}, u_{i,2})$ to the center of the bivariate data set, while χ_i corresponds to a correlation coefficient between dichotomized values of U_1 and U_2. Under independence it holds that $\chi_i \sim \mathcal{N}(0, \frac{1}{N})$ and $\lambda_i \sim \mathcal{U}[-1, 1]$ asymptotically, i.e., values of χ_i close to zero indicate independence—corresponding to $F_{U_1 U_2} = F_{U_1} F_{U_2}$.

When plotting these quantities, the pairs of (λ_i, χ_i) will tend to be located above zero for positively dependent margins and vice versa for negatively dependent margins. Control bounds around zero indicate whether there is significant dependence present.

If mode = "lower" or "upper", the above quantities are calculated only for those $u_{i,1}$'s and $u_{i,2}$'s which are smaller/larger than the respective means of u1= $(u_{1,1}, ..., u_{N,1})$ and u2= $(u_{1,2}, ..., u_{N,2})$.

Value

lambda   Lambda-statistics (x-axis).

chi   Chi-statistics (y-axis).

control.bounds   A 2-dimensional vector of bounds $((1.54/\sqrt{n}, -1.54/\sqrt{n})$, where n is the length of u1 and where the chosen values correspond to an approximate significance level of 10%.

Author(s)

Natalia Belgorodski, Ulf Schepsmeier

References

Abberger, K. (2004). A simple graphical method to explore tail-dependence in stock-return pairs. Discussion Paper, University of Konstanz, Germany.

Genest, C. and A. C. Favre (2007). Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering, 12 (4), 347-368.

See Also

BiCopMetaContour, BiCopKPlot, BiCopLambda

Examples

```
# chi-plots for bivariate Gaussian copula data
n = 500
tau = 0.5

# simulate copula data
fam = 1
theta = BiCopTau2Par(fam,tau)
dat = BiCopSim(n,fam,theta)

# create chi-plots
dev.new(width=16,height=5)
par(mfrow=c(1,3))
BiCopChiPlot(dat[,1],dat[,2],xlim=c(-1,1),ylim=c(-1,1),
             main="General chi-plot")
BiCopChiPlot(dat[,1],dat[,2],mode="lower",xlim=c(-1,1),
             ylim=c(-1,1),main="Lower chi-plot")
BiCopChiPlot(dat[,1],dat[,2],mode="upper",xlim=c(-1,1),
             ylim=c(-1,1),main="Upper chi-plot")
```

BiCopEst   *Parameter estimation for bivariate copula data using inversion of Kendall's tau or maximum likelihood estimation*

Description

This function estimates the parameter(s) for a bivariate copula using either inversion of empirical Kendall's tau for single parameter copula families or maximum likelihood estimation for one and two parameter copula families supported in this package.

Usage

```
BiCopEst(u1, u2, family, method="mle", se=FALSE, max.df=30,
         max.BB=list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1)))
```

Arguments

u1,u2   Data vectors of equal length with values in [0,1].

family   An integer defining the bivariate copula family:
                0 = independence copula
                1 = Gaussian copula
                2 = Student t copula (t-copula)
                3 = Clayton copula
                4 = Gumbel copula
                5 = Frank copula
                6 = Joe copula
                7 = BB1 copula
                8 = BB6 copula
                9 = BB7 copula
                10 = BB8 copula
                13 = rotated Clayton copula (180 degrees; "survival Clayton")
                14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
                16 = rotated Joe copula (180 degrees; "survival Joe")
                17 = rotated BB1 copula (180 degrees; "survival BB1")
                18 = rotated BB6 copula (180 degrees; "survival BB6")
                19 = rotated BB7 copula (180 degrees; "survival BB7")
                20 = rotated BB8 copula (180 degrees; "survival BB8")
                23 = rotated Clayton copula (90 degrees)
                24 = rotated Gumbel copula (90 degrees)
                26 = rotated Joe copula (90 degrees)
                27 = rotated BB1 copula (90 degrees)
                28 = rotated BB6 copula (90 degrees)
                29 = rotated BB7 copula (90 degrees)
                30 = rotated BB8 copula (90 degrees)
                33 = rotated Clayton copula (270 degrees)
                34 = rotated Gumbel copula (270 degrees)
                36 = rotated Joe copula (270 degrees)
                37 = rotated BB1 copula (270 degrees)
                38 = rotated BB6 copula (270 degrees)
                39 = rotated BB7 copula (270 degrees)
                40 = rotated BB8 copula (270 degrees)

method	Character indicating the estimation method: either maximum likelihood estimation (method = "mle"; default) or inversion of Kendall's tau (method = "itau"). For method = "itau" only one parameter bivariate copula families can be used (family = 1,3,4,5,6,13,14,16,23,24,26,33,34 or 36).
se	Logical; whether standard error(s) of parameter estimates is/are estimated (default: se = FALSE).
max.df	Numeric; upper bound for the estimation of the degrees of freedom parameter of the t-copula (default: max.df = 30).
max.BB	List; upper bounds for the estimation of the two parameters (in absolute values) of the BB1, BB6, BB7 and BB8 copulas (default: max.BB = list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1))).

Details

If method = "itau", the function computes the empirical Kendall's tau of the given copula data and exploits the one-to-one relationship of copula parameter and Kendall's tau which is available for many one parameter bivariate copula families (see [BiCopPar2Tau](#) and [BiCopTau2Par](#)). The inversion of Kendall's tau is however not available for all bivariate copula families (see above). If a two parameter copula family is chosen and method = "itau", a warning message is returned and the MLE is calculated.

For method = "mle" copula parameters are estimated by maximum likelihood using starting values obtained by method = "itau". If no starting values are available by inversion of Kendall's tau, starting values have to be provided given expert knowledge and the boundaries max.df and max.BB respectively.

A warning message is returned if the estimate of the degrees of freedom parameter of the t-copula is larger than max.df. For high degrees of freedom the t-copula is almost indistinguishable from the Gaussian and it is advised to use the Gaussian copula in this case. As a rule of thumb max.df = 30 typically is a good choice. Moreover, standard errors of the degrees of freedom parameter estimate cannot be estimated in this case.

Value

par, par2	Estimated copula parameter(s).
se,se2	Standard error(s) of the parameter estimate(s) (if se = TRUE).

Author(s)

Ulf Schepsmeier, Eike Brechmann, Jakob Stoeber, Carlos Almeida

References

Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman and Hall, London.

See Also

[BiCopPar2Tau](#), [BiCopTau2Par](#), [CDVineSeqEst](#), [BiCopSelect](#)

Examples

```
## Example 1: bivariate Gaussian copula
dat = BiCopSim(500,1,0.7)
u1 = dat[,1]
v1 = dat[,2]

# empirical Kendall's tau
tau1 = cor(u1,v1,method="kendall")

# inversion of empirical Kendall's tau
BiCopTau2Par(1,tau1)
BiCopEst(u1,v1,family=1,method="itau")$par

# maximum likelihood estimate for comparison
BiCopEst(u1,v1,family=1,method="mle")$par


## Example 2: bivariate Clayton and survival Gumbel copulas
# simulate from a Clayton copula
dat = BiCopSim(500,3,2.5)
u2 = dat[,1]
v2 = dat[,2]

# empirical Kendall's tau
tau2 = cor(u2,v2,method="kendall")

# inversion of empirical Kendall's tau for the Clayton copula
BiCopTau2Par(3,tau2)
BiCopEst(u2,v2,family=3,method="itau",se=TRUE)

# inversion of empirical Kendall's tau for the survival Gumbel copula
BiCopTau2Par(14,tau2)
BiCopEst(u2,v2,family=14,method="itau",se=TRUE)

# maximum likelihood estimates for comparison
BiCopEst(u2,v2,family=3,method="mle",se=TRUE)
BiCopEst(u2,v2,family=14,method="mle",se=TRUE)

## Example 3: fit of a t-copula to standardized residuals of
## S&P 500 and DAX returns
data(worldindices)
BiCopEst(worldindices[,1],worldindices[,4],family=2,method="mle",se=TRUE)
```

BiCopGofKendall	*Goodness-of-fit test based on Kendall's process for bivariate copula data*

Description

This function performs the goodness-of-fit test based on Kendall's process for bivariate copula data. It computes the Cramer-von Mises and Kolmogorov-Smirnov test statistics, respectively, as well as the according p-values using bootstrapping.

Usage

```
BiCopGofKendall(u1, u2, family, B=100, level=0.05)
```

Arguments

u1,u2   Data vectors of equal length with values in [0,1].

family   An integer defining the bivariate copula family for which the test is performed:
               1 = Gaussian copula
               2 = Student t copula (t-copula)
               3 = Clayton copula
               4 = Gumbel copula
               5 = Frank copula
               6 = Joe copula
               7 = BB1 copula
               8 = BB6 copula
               9 = BB7 copula
               10 = BB8 copula
               13 = rotated Clayton copula (180 degrees; "survival Clayton")
               14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
               16 = rotated Joe copula (180 degrees; "survival Joe")
               17 = rotated BB1 copula (180 degrees; "survival BB1")
               18 = rotated BB6 copula (180 degrees; "survival BB6")
               19 = rotated BB7 copula (180 degrees; "survival BB7")
               20 = rotated BB8 copula (180 degrees; "survival BB8")
               23 = rotated Clayton copula (90 degrees)
               24 = rotated Gumbel copula (90 degrees)
               26 = rotated Joe copula (90 degrees)
               27 = rotated BB1 copula (90 degrees)
               28 = rotated BB6 copula (90 degrees)
               29 = rotated BB7 copula (90 degrees)
               30 = rotated BB8 copula (90 degrees)
               33 = rotated Clayton copula (270 degrees)
               34 = rotated Gumbel copula (270 degrees)
               36 = rotated Joe copula (270 degrees)
               37 = rotated BB1 copula (270 degrees)
               38 = rotated BB6 copula (270 degrees)
               39 = rotated BB7 copula (270 degrees)
               40 = rotated BB8 copula (270 degrees)

B   Integer; number of bootstrap samples (default: B = 100). For B = 0 only the
               the test statistics are returned. WARNING: If B is chosen too large, computations
               will take very long.

level   Numeric; significance level of the goodness-of-fit test (default: level = 0.05).

Details

This copula goodness-of-fit test is based on Kendall's process as investigated by Genest and Rivest (1993) and Wang and Wells (2000). For rotated copulas the input arguments are transformed and the goodness-of-fit procedure for the corresponding non-rotated copula is used.

Value

`p.value.CvM`	P-value of the goodness-of-fit test using the Cramer-von Mises statistic (if B > 0).
`p.value.KS`	P-value of the goodness-of-fit test using the Kolmogorov-Smirnov statistic (if B > 0).
`statistic.CvM`	The observed Cramer-von Mises test statistic.
`statistic.KS`	The observed Kolmogorov-Smirnov test statistic.

Author(s)

Jiying Luo, Eike Brechmann

References

Genest, C. and L.-P. Rivest (1993). Statistical inference procedures for bivariate Archimedean copulas. Journal of the American Statistical Association, 88 (423), 1034-1043.

Luo J. (2011). Stepwise estimation of D-vines with arbitrary specified copula pairs and EDA Tools. Diploma thesis, Technische Universitaet Muenchen.
http://mediatum.ub.tum.de/doc/1079291/1079291.pdf.

Wang, W. and M. T. Wells (2000). Model selection and semiparametric inference for bivariate failure-time data. Journal of the American Statistical Association, 95 (449), 62-72.

See Also

BiCopIndTest, BiCopSelect, BiCopVuongClarke, BiCopKPlot, BiCopLambda

Examples

```
# sample from a Gaussian copula
par1 = 3
fam1 = 3
dat1 = BiCopSim(500,fam1,par1)

# perform the goodness-of-fit test for the true copula
gof = BiCopGofKendall(dat1[,1],dat1[,2],fam1)
gof$p.value.CvM
gof$p.value.KS

# perform the goodness-of-fit test for the Frank copula
gof = BiCopGofKendall(dat1[,1],dat1[,2],5)
gof$p.value.CvM
gof$p.value.KS
```

BiCopHfunc   *Conditional distribution function (h-function) of a bivariate copula*

Description

This function evaluates the conditional distribution function (h-function) of a given parametric bivariate copula.

Usage

```
BiCopHfunc(u1, u2, family, par, par2=0)
```

Arguments

u1,u2	Numeric vectors of equal length with values in [0,1].
family	An integer defining the bivariate copula family:
	0 = independence copula
	1 = Gaussian copula
	2 = Student t copula (t-copula)
	3 = Clayton copula
	4 = Gumbel copula
	5 = Frank copula
	6 = Joe copula
	7 = BB1 copula
	8 = BB6 copula
	9 = BB7 copula
	10 = BB8 copula
	13 = rotated Clayton copula (180 degrees; "survival Clayton")
	14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
	16 = rotated Joe copula (180 degrees; "survival Joe")
	17 = rotated BB1 copula (180 degrees; "survival BB1")
	18 = rotated BB6 copula (180 degrees; "survival BB6")
	19 = rotated BB7 copula (180 degrees; "survival BB7")
	20 = rotated BB8 copula (180 degrees; "survival BB8")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)
	27 = rotated BB1 copula (90 degrees)
	28 = rotated BB6 copula (90 degrees)
	29 = rotated BB7 copula (90 degrees)
	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)
	39 = rotated BB7 copula (270 degrees)
	40 = rotated BB8 copula (270 degrees)
par	Copula parameter.
par2	Second parameter for bivariate copulas with two parameters (t, BB1, BB6, BB7, BB8; default: par2 = 0).

Details

The h-function is defined as the conditional distribution function of a bivariate copula, i.e.,

$$h(u|v, \boldsymbol{\theta}) := F(u|v) = \frac{\partial C(u, v)}{\partial v},$$

where C is a bivariate copula distribution function with parameter(s) $\boldsymbol{\theta}$. For more details see Aas et al. (2009).

Value

hfunc1	Numeric vector of the conditional distribution function (h-function) evaluated at u2 given u1, i.e., $h(u2|u1, \boldsymbol{\theta})$.
hfunc2	Numeric vector of the conditional distribution function (h-function) evaluated at u1 given u2, i.e., $h(u1|u2, \boldsymbol{\theta})$.

Author(s)

Ulf Schepsmeier

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

See Also

BiCopPDF, BiCopCDF, CDVineLogLik, CDVineSeqEst

Examples

```
## Example 1: 4-dimensional C-vine model with mixed pair-copulas
data(worldindices)
Data = as.matrix(worldindices)[,1:4]
d = dim(Data)[2]
fam = c(5,1,3,14,3,2)

# sequential estimation
seqpar1 = CDVineSeqEst(Data,fam,type=1,method="itau")

# calculate the inputs of the second tree using h-functions
h1 = BiCopHfunc(Data[,1],Data[,2],fam[1],seqpar1$par[1])
h2 = BiCopHfunc(Data[,1],Data[,3],fam[2],seqpar1$par[2])
h3 = BiCopHfunc(Data[,1],Data[,4],fam[3],seqpar1$par[3])

# compare estimated parameters
BiCopEst(h1$hfunc1,h2$hfunc1,fam[4],method="itau")
seqpar1$par[4]

BiCopEst(h1$hfunc1,h3$hfunc1,fam[5],method="itau")
seqpar1$par[5]


## Example 2: 4-dimensional D-vine model with mixed pair-copulas

# sequential estimation
seqpar2 = CDVineSeqEst(Data,fam,type=2,method="itau")

# calculate the inputs of the second tree using h-functions
h1 = BiCopHfunc(Data[,1],Data[,2],fam[1],seqpar2$par[1])
h2 = BiCopHfunc(Data[,2],Data[,3],fam[2],seqpar2$par[2])
h3 = BiCopHfunc(Data[,3],Data[,4],fam[3],seqpar2$par[3])

# compare estimated parameters
BiCopEst(h1$hfunc2,h2$hfunc1,fam[4],method="itau")
```

```
seqpar2$par[4]

BiCopEst(h2$hfunc2,h3$hfunc1,fam[5],method="itau")
seqpar2$par[5]
```

BiCopIndTest	*Independence test for bivariate copula data*

Description

This function returns the p-value of a bivariate asymptotic independence test based on Kendall's tau.

Usage

```
BiCopIndTest(u1, u2)
```

Arguments

u1,u2   Data vectors of equal length with values in [0,1].

Details

The test exploits the asymptotic normality of the test statistic

$$\text{statistic} := T = \sqrt{\frac{9N(N-1)}{2(2N+5)}} \times |\hat{\tau}|,$$

where N is the number of observations (length of u1) and $\hat{\tau}$ the empirical Kendall's tau of the data vectors u1 and u2. The p-value of the null hypothesis of bivariate independence hence is asymptotically

$$\text{p.value} = 2 \times (1 - \Phi(T)),$$

where Φ is the standard normal distribution function.

Value

statistic   Test statistic of the independence test.

p.value   P-value of the independence test.

Author(s)

Jeffrey Dissmann

References

Genest, C. and A. C. Favre (2007). Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering, 12 (4), 347-368.

See Also

[BiCopPar2Tau](#), [BiCopTau2Par](#), [BiCopSelect](#), [CDVineCopSelect](#)

Examples

```
## Example 1: Gaussian copula with large dependence parameter
par1 = 0.7
fam1 = 1
dat1 = BiCopSim(500,fam1,par1)

# perform the asymptotic independence test
BiCopIndTest(dat1[,1],dat1[,2])


## Example 2: Gaussian copula with small dependence parameter
par2 = 0.01
fam2 = 1
dat2 = BiCopSim(500,fam2,par2)

# perform the asymptotic independence test
BiCopIndTest(dat2[,1],dat2[,2])
```

BiCopKPlot   *Kendall's plot (K-plot) for bivariate copula data*

Description

This function creates a Kendall's plot (K-plot) of given bivariate copula data.

Usage

```
BiCopKPlot(u1, u2, PLOT=TRUE, ...)
```

Arguments

u1,u2	Data vectors of equal length with values in [0,1].
PLOT	Logical; whether the results are plotted. If PLOT = FALSE, the values W.in and Hi.sort are returned (see below; default: PLOT = TRUE).
...	Additional plot arguments.

Details

For observations $u_{i,j}$, $i = 1, ..., N$, $j = 1, 2$, the K-plot considers two quantities: First, the ordered values of the empirical bivariate distribution function $H_i := \hat{F}_{U_1 U_2}(u_{i,1}, u_{i,2})$ and, second, $W_{i:N}$, which are the expected values of the order statistics from a random sample of size N of the random variable $W = C(U_1, U_2)$ under the null hypothesis of independence between U_1 and U_2. $W_{i:N}$ can be calculated as follows

$$W_{i:n} = N \binom{N-1}{i-1} \int_0^1 \omega k_0(\omega)(K_0(\omega))^{i-1}(1 - K_0(\omega))^{N-i} d\omega,$$

where

$$K_0(\omega) = \omega - \omega \log(\omega),$$

and $k_0(\cdot)$ is the corresponding density.

K-plots can be seen as the bivariate copula equivalent to QQ-plots. If the points of a K-plot lie approximately on the diagonal $y = x$, then U_1 and U_2 are approximately independent. Any deviation from the diagonal line points towards dependence. In case of positive dependence, the points of the K-plot should be located above the diagonal line, and vice versa for negative dependence. The larger the deviation from the diagonal, the stronger is the degree of dependency. There is a perfect positive dependence if points $(W_{i:N}, H_i)$ lie on the curve $K_0(\omega)$ located above the main diagonal. If points $(W_{i:N}, H_i)$ however lie on the x-axis, this indicates a perfect negative dependence between U_1 and U_2.

Value

`W.in`	W-statistics (x-axis).
`Hi.sort`	H-statistics (y-axis).

Author(s)

Natalia Belgorodski, Ulf Schepsmeier

References

Genest, C. and A. C. Favre (2007). Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering, 12 (4), 347-368.

See Also

`BiCopMetaContour`, `BiCopChiPlot`, `BiCopLambda`, `BiCopGofKendall`

Examples

```
# Gaussian and Clayton copulas
n = 500
tau = 0.5

# simulate from Gaussian copula
fam1 = 1
theta1 = BiCopTau2Par(fam1,tau)
dat1 = BiCopSim(n,fam1,theta1)

# simulate from Clayton copula
fam2 = 3
theta2 = BiCopTau2Par(fam2,tau)
dat2 = BiCopSim(n,fam2,theta2)

# create K-plots
dev.new(width=10,height=5)
par(mfrow=c(1,2))
BiCopKPlot(dat1[,1],dat1[,2],main="Gaussian copula")
BiCopKPlot(dat2[,1],dat2[,2],main="Clayton copula")
```

BiCopLambda   *Lambda-function (plot) for bivariate copula data*

Description

This function plots the lambda-function of given bivariate copula data.

Usage

```
BiCopLambda(u1=NULL, u2=NULL, family="emp", par=0, par2=0,
            PLOT=TRUE, ...)
```

Arguments

u1,u2	Data vectors of equal length with values in [0,1] (default: u1 and u2 = NULL).
family	An integer defining the bivariate copula family or indicating the empirical lambda-function: "emp" = empirical lambda-function (default) 1 = Gaussian copula; the theoretical lambda-function is simulated (no closed formula available) 2 = Student t copula (t-copula); the theoretical lambda-function is simulated (no closed formula available) 3 = Clayton copula 4 = Gumbel copula 5 = Frank copula 6 = Joe copula 7 = BB1 copula 8 = BB6 copula 9 = BB7 copula 10 = BB8 copula
par	Copula parameter; if the empirical lambda-function is chosen, par = NULL or 0 (default).
par2	Second copula parameter for t-, BB1, BB6, BB7 and BB8 copulas (default: par2 = 0).
PLOT	Logical; whether the results are plotted. If PLOT = FALSE, the values empLambda and/or theoLambda are returned (see below; default: PLOT = TRUE).
...	Additional plot arguments.

Value

empLambda	If the empirical lambda-function is chosen and PLOT=FALSE, a vector of the empirical lambda's is returned.
theoLambda	If the theoretical lambda-function is chosen and PLOT=FALSE, a vector of the theoretical lambda's is returned.

Note

The λ-function is characteristic for each bivariate copula family and defined by Kendall's distribution function K:

$$\lambda(v, \theta) := v - K(v, \theta)$$

with

$$K(v, \theta) := P(C_\theta(U_1, U_2) \le v), \;\; v \in [0, 1].$$

For Archimedean copulas one has the following closed form expression in terms of the generator function φ of the copula C_θ:

$$\lambda(v, \theta) = \frac{\varphi(v)}{\varphi'(v)},$$

where φ' is the derivative of φ. For more details see Genest and Rivest (1993) or Schepsmeier (2010).

For the bivariate Gaussian and t-copula no closed form expression for the theoretical λ-function exists. Therefore it is simulated based on samples of size 1000. For all other implemented copula families there are closed form expressions available.

The plot of the theoretical λ-function also shows the limits of the λ-function corresponding to Kendall's tau $= 0$ and Kendall's tau $= 1$ ($\lambda = 0$).

For rotated bivariate copulas one has to transform the input arguments u1 and/or u2. In particular, for copulas rotated by 90 degrees u1 has to be set to 1-u1, for 270 degrees u2 to 1-u2 and for survival copulas u1 and u2 to 1-u1 and 1-u2, respectively. Then λ-functions for the corresponding non-rotated copula families can be considered.

Author(s)

Ulf Schepsmeier

References

Genest, C. and L.-P. Rivest (1993). Statistical inference procedures for bivariate Archimedean copulas. Journal of the American Statistical Association, 88 (423), 1034-1043.

Schepsmeier, U. (2010). Maximum likelihood estimation of C-vine pair-copula constructions based on bivariate copulas from different families. Diploma thesis, Technische Universitaet Muenchen. http://mediatum.ub.tum.de/doc/1079296/1079296.pdf.

See Also

BiCopMetaContour, BiCopKPlot, BiCopChiPlot

Examples

```
# Clayton and rotated Clayton copulas
n = 1000
tau = 0.5

# simulate from Clayton copula
fam = 3
theta = BiCopTau2Par(fam,tau)
dat = BiCopSim(n,fam,theta)

# create lambda-function plots
dev.new(width=16,height=5)
```

```
par(mfrow=c(1,3))
BiCopLambda(dat[,1],dat[,2]) # empirical lambda-function
BiCopLambda(family=fam,par=theta) # theoretical lambda-function
BiCopLambda(dat[,1],dat[,2],family=fam,par=theta) # both

# simulate from rotated Clayton copula (90 degrees)
fam = 23
theta = BiCopTau2Par(fam,-tau)
dat = BiCopSim(n,fam,theta)

# rotate the data to standard Clayton copula data
rot_dat = 1-dat[,1]

dev.new(width=16,height=5)
par(mfrow=c(1,3))
BiCopLambda(rot_dat,dat[,2])  # empirical lambda-function
BiCopLambda(family=3,par=-theta) # theoretical lambda-function
BiCopLambda(rot_dat,dat[,2],family=3,par=-theta) # both
```

BiCopMetaContour	*Contour plot of bivariate meta distribution with different margins and copula (theoretical and empirical)*

Description

This function plots a bivariate contour plot corresponding to a bivariate meta distribution with different margins and specified bivariate copula and parameter values or creates corresponding empirical contour plots based on bivariate copula data.

Usage

```
BiCopMetaContour(u1=NULL, u2=NULL, bw=1, size=100,
                 levels=c(0.01,0.05,0.1,0.15,0.2),
                 family="emp", par=0, par2=0, PLOT=TRUE,
                 margins="norm", margins.par=0, xlim=NA, ...)
```

Arguments

u1,u2	Data vectors of equal length with values in [0,1] (default: u1 and u2 = NULL).
bw	Bandwidth (smoothing factor; default: bw = 1).
size	Number of grid points; default: size = 100.
levels	Vector of contour levels. For Gaussian, Student t or exponential margins the default value (levels = c(0.01,0.05,0.1,0.15,0.2)) typically is a good choice. For uniform margins we recommend levels = c(0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.5) and for Gamma margins levels = c(0.005,0.01,0.03,0.05,0.07,0.09).
family	An integer defining the bivariate copula family or indicating an empirical contour plot: "emp" = empirical contour plot (default; margins can be specified by margins)

0 = independence copula
1 = Gaussian copula
2 = Student t copula (t-copula)
3 = Clayton copula
4 = Gumbel copula
5 = Frank copula
6 = Joe copula
7 = BB1 copula
8 = BB6 copula
9 = BB7 copula
10 = BB8 copula
13 = rotated Clayton copula (180 degrees; "survival Clayton")
14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
16 = rotated Joe copula (180 degrees; "survival Joe")
17 = rotated BB1 copula (180 degrees; "survival BB1")
18 = rotated BB6 copula (180 degrees; "survival BB6")
19 = rotated BB7 copula (180 degrees; "survival BB7")
20 = rotated BB8 copula (180 degrees; "survival BB8")
23 = rotated Clayton copula (90 degrees)
24 = rotated Gumbel copula (90 degrees)
26 = rotated Joe copula (90 degrees)
27 = rotated BB1 copula (90 degrees)
28 = rotated BB6 copula (90 degrees)
29 = rotated BB7 copula (90 degrees)
30 = rotated BB8 copula (90 degrees)
33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

par	Copula parameter; if empirical contour plot, par = NULL or 0 (default).
par2	Second copula parameter for t-, BB1, BB6, BB7 and BB8 copulas (default: par2 = 0).
PLOT	Logical; whether the results are plotted. If PLOT = FALSE, the values x, y and z are returned (see below; default: PLOT = TRUE).
margins	Character; margins for the bivariate copula contour plot. Possible margins are: "norm" = standard normal margins (default) "t" = Student t margins with degrees of freedom as specified by margins.par "gamma" = Gamma margins with shape and scale as specified by margins.par "exp" = Exponential margins with rate as specified by margins.par "unif" = uniform margins
margins.par	Parameter(s) of the distribution of the margins if necessary (default: margins.par = 0), i.e.,

- a positive real number for the degrees of freedom of Student t margins (see dt),
- a 2-dimensional vector of positive real numbers for the shape and scale parameters of Gamma margins (see dgamma),
- a positive real number for the rate parameter of exponential margins (see dexp).

xylim   A 2-dimensional vector of the x- and y-limits. By default (xylim = NA) stan-
                     dard limits for the selected margins are used.

...   Additional plot arguments.

Value

x   A vector of length `size` with the x-values of the kernel density estimator with
                     Gaussian kernel if the empirical contour plot is chosen and a sequence of values
                     in `xylim` if the theoretical contour plot is chosen.

y   A vector of length `size` with the y-values of the kernel density estimator with
                     Gaussian kernel if the empirical contour plot is chosen and a sequence of values
                     in `xylim` if the theoretical contour plot is chosen.

z   A matrix of dimension `size` with the values of the density of the meta distribu-
                     tion with chosen margins (see `margins` and `margins.par`) evaluated at the grid
                     points given by x and y.

Note

Warning: The combination `family = 0` (independence copula) and `margins = "unif"` (uniform
margins) is not possible because all z-values are equal.

Author(s)

Ulf Schepsmeier, Alexander Bauer

See Also

[BiCopChiPlot](), [BiCopKPlot](), [BiCopLambda]()

Examples

```
## Example 1: contour plot of meta Gaussian copula distribution
## with Gaussian margins
tau = 0.5
fam = 1
theta = BiCopTau2Par(fam,tau)
BiCopMetaContour(u1=NULL,u2=NULL,bw=1,size=100,
                 levels=c(0.01,0.05,0.1,0.15,0.2),
                 family=fam,par=theta,main="tau=0.5")


## Example 2: empirical contour plot with standard normal margins
dat = BiCopSim(N=1000,fam,theta)
BiCopMetaContour(dat[,1],dat[,2],bw=2,size=100,
                 levels=c(0.01,0.05,0.1,0.15,0.2),
                 par=0,family="emp",main="N=1000")

# empirical contour plot with exponential margins
BiCopMetaContour(dat[,1],dat[,2],bw=2,size=100,
                 levels=c(0.01,0.05,0.1,0.15,0.2),
                 par=0,family="emp",main="n=500",
                 margins="exp",margins.par=1)
```

BiCopName	*Bivariate copula family names*

Description

This function transforms the bivariate copula family number into its character expression and vice versa.

Usage

```
BiCopName(family, short=TRUE)
```

Arguments

family   Bivariate copula family, either its number or its character expression (see table below).

No.	Short name	Long name
0	"I"	"Independence"
1	"N"	"Gaussian"
2	"t"	"t"
3	"C"	"Clayton"
4	"G"	"Gumbel"
5	"F"	"Frank"
6	"J"	"Joe"
7	"BB1"	"Clayton-Gumbel"
8	"BB6"	"Joe-Gumbel"
9	"BB7"	"Joe-Clayton"
10	"BB8"	"Frank-Joe"
13	"SC"	"Survival Clayton"
14	"SG"	"Survival Gumbel"
16	"SJ"	"Survival Joe"
17	"SBB1"	"Survival Clayton-Gumbel"
18	"SBB6"	"Survival Joe-Gumbel"
19	"SBB7"	"Survival Joe-Clayton"
20	"SBB8"	"Survival Joe-Frank"
23	"C90"	"Rotated Clayton 90 degrees"
24	"G90"	"Rotated Gumbel 90 degrees"
26	"J90"	"Rotated Joe 90 degrees"
27	"BB1_90"	"Rotated Clayton-Gumbel 90 degrees"
28	"BB6_90"	"Rotated Joe-Gumbel 90 degrees"
29	"BB7_90"	"Rotated Joe-Clayton 90 degrees"
30	"BB8_90"	"Rotated Frank-Joe 90 degrees"
33	"C270"	"Rotated Clayton 270 degrees"
34	"G270"	"Rotated Gumbel 270 degrees"
36	"J270"	"Rotated Joe 270 degrees"
37	"BB1_270"	"Rotated Clayton-Gumbel 270 degrees"
38	"BB6_270"	"Rotated Joe-Gumbel 270 degrees"
39	"BB7_270"	"Rotated Joe-Clayton 270 degrees"
40	"BB8_270"	"Rotated Frank-Joe 270 degrees"

short   Logical; if the number of a bivariate copula family is used and `short = TRUE`
                 (default), a short version of the corresponding character expression is returned,
                 otherwise the long version.

Value

The transformed bivariate copula family (see table above).

Author(s)

Ulf Schepsmeier

See Also

[CDVineTreePlot](#)

Examples

```
# family as number
family = 1
BiCopName(family,short=TRUE)  # short version
BiCopName(family,short=FALSE) # long version

# family as character expression (short version)
family = "C"
BiCopName(family) # as number

# long version
family = "Clayton"
BiCopName(family) # as number
```

BiCopPar2TailDep   *Tail dependence coefficients of a bivariate copula*

Description

This function computes the theoretical tail dependence coefficients of a bivariate copula for given
parameter values.

Usage

```
BiCopPar2TailDep(family, par, par2=0)
```

Arguments

family   An integer defining the bivariate copula family:
                  0 = independence copula
                  1 = Gaussian copula
                  2 = Student t copula (t-copula)
                  3 = Clayton copula
                  4 = Gumbel copula
                  5 = Frank copula
                  6 = Joe copula

        7 = BB1 copula
        8 = BB6 copula
        9 = BB7 copula
        10 = BB8 copula
        13 = rotated Clayton copula (180 degrees; "survival Clayton")
        14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
        16 = rotated Joe copula (180 degrees; "survival Joe")
        17 = rotated BB1 copula (180 degrees; "survival BB1")
        18 = rotated BB6 copula (180 degrees; "survival BB6")
        19 = rotated BB7 copula (180 degrees; "survival BB7")
        20 = rotated BB8 copula (180 degrees; "survival BB8")
        23 = rotated Clayton copula (90 degrees)
        24 = rotated Gumbel copula (90 degrees)
        26 = rotated Joe copula (90 degrees)
        27 = rotated BB1 copula (90 degrees)
        28 = rotated BB6 copula (90 degrees)
        29 = rotated BB7 copula (90 degrees)
        30 = rotated BB8 copula (90 degrees)
        33 = rotated Clayton copula (270 degrees)
        34 = rotated Gumbel copula (270 degrees)
        36 = rotated Joe copula (270 degrees)
        37 = rotated BB1 copula (270 degrees)
        38 = rotated BB6 copula (270 degrees)
        39 = rotated BB7 copula (270 degrees)
        40 = rotated BB8 copula (270 degrees)

par   Copula parameter.

par2   Second parameter for the two parameter t-, BB1, BB6, BB7 and BB8 copulas (default: `par2 = 0`).

Value

lower   Lower tail dependence coefficient of the given bivariate copula family C:

$$\lambda_L = \lim_{u \searrow 0} \frac{C(u,u)}{u}$$

upper   Upper tail dependence coefficient of the given bivariate copula family C:

$$\lambda_U = \lim_{u \nearrow 1} \frac{1 - 2u + C(u,u)}{1 - u}$$

Lower and upper tail dependence coefficients for bivariate copula families and parameters (θ for one parameter families and the first parameter of the t-copula with ν degrees of freedom, θ and δ for the two parameter BB1, BB6, BB7 and BB8 copulas) are given in the following table.

No.	Lower tail dependence	Upper tail dependence
1	-	-
2	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\theta}{1+\theta}}\right)$	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\theta}{1+\theta}}\right)$
3	$2^{-1/\theta}$	-
4	-	$2 - 2^{1/\theta}$
5	-	-
6	-	$2 - 2^{1/\theta}$

7	$2^{-1/(\theta\delta)}$	$2 - 2^{1/\delta}$
8	-	$2 - 2^{1/(\theta\delta)}$
9	$2^{-1/\delta}$	$2 - 2^{1/\theta}$
10	-	$2 - 2^{1/\theta}$ if $\delta = 1$ otherwise 0
13	-	$2^{-1/\theta}$
14	$2 - 2^{1/\theta}$	-
16	$2 - 2^{1/\theta}$	-
17	$2 - 2^{1/\delta}$	$2^{-1/(\theta\delta)}$
18	$2 - 2^{1/(\theta\delta)}$	-
19	$2 - 2^{1/\theta}$	$2^{-1/\delta}$
20	$2 - 2^{1/\theta}$ if $\delta = 1$ otherwise 0	-
23, 33	-	-
24, 34	-	-
26, 36	-	-
27, 37	-	-
28, 38	-	-
29, 39	-	-
30, 40	-	-

Author(s)

Eike Brechmann

References

Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman and Hall, London.

See Also

BiCopPar2Tau

Examples

```
## Example 1: Gaussian copula
BiCopPar2TailDep(1,0.7)

## Example 2: t copula
BiCopPar2TailDep(2,0.7,4)
```

BiCopPar2Tau   *Kendall's tau value of a bivariate copula*

Description

This function computes the theoretical Kendall's tau value of a bivariate copula for given parameter values.

Usage

```
BiCopPar2Tau(family, par, par2=0)
```

Arguments

family	An integer defining the bivariate copula family:

                            0 = independence copula
                            1 = Gaussian copula
                            2 = Student t copula (t-copula)

2 = Student t copula (t-copula)
3 = Clayton copula
4 = Gumbel copula
5 = Frank copula
6 = Joe copula
7 = BB1 copula
8 = BB6 copula
9 = BB7 copula
10 = BB8 copula
13 = rotated Clayton copula (180 degrees; "survival Clayton")
14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
16 = rotated Joe copula (180 degrees; "survival Joe")
17 = rotated BB1 copula (180 degrees; "survival BB1")
18 = rotated BB6 copula (180 degrees; "survival BB6")
19 = rotated BB7 copula (180 degrees; "survival BB7")
20 = rotated BB8 copula (180 degrees; "survival BB8")
23 = rotated Clayton copula (90 degrees)
24 = rotated Gumbel copula (90 degrees)
26 = rotated Joe copula (90 degrees)
27 = rotated BB1 copula (90 degrees)
28 = rotated BB6 copula (90 degrees)
29 = rotated BB7 copula (90 degrees)
30 = rotated BB8 copula (90 degrees)
33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

par	Copula parameter.
par2	Second parameter for the two parameter BB1, BB6, BB7 and BB8 copulas (default: par2 = 0). Note that the degrees of freedom parameter of the t-copula does not need to be set, because the theoretical Kendall's tau value of the t-copula is independent of this choice.

Value

Theoretical value of Kendall's tau corresponding to the bivariate copula family and parameter(s) (θ for one parameter families and the first parameter of the t-copula, θ and δ for the two parameter BB1, BB6, BB7 and BB8 copulas).

No.	Kendall's tau
1, 2	$\frac{2}{\pi}\arcsin(\theta)$
3, 13	$\frac{\theta}{\theta+2}$
4, 14	$1 - \frac{1}{\theta}$
5	$1 - \frac{4}{\theta} + 4\frac{D_1(\theta)}{\theta}$

with $D_1(\theta) = \int_0^\theta \frac{x/\theta}{\exp(x)-1}dx$ (Debye function)

6, 16	$1 + \frac{4}{\theta^2}\int_0^1 x\log(x)(1-x)^{2(1-\theta)/\theta}dx$
7, 17	$1 - \frac{2}{\delta(\theta+2)}$
8, 18	$1 + 4\int_0^1 -\log(-(1-t)^\theta+1)(1-t-(1-t)^{-\theta}+(1-t)^{-\theta}t)/(\delta\theta)dt$
9, 19	$1 + 4\int_0^1((1-(1-t)^\theta)^{-\delta}-)/(-\theta\delta(1-t)^{\theta-1}(1-(1-t)^\theta)^{-\delta-1})dt$
10, 20	$1 + 4\int_0^1 -\log\left(((1-t\delta)^\theta-1)/((1-\delta)^\theta-1)\right)$
	$*(1-t\delta-(1-t\delta)^{-\theta}+(1-t\delta)^{-\theta}t\delta)/(\theta\delta)dt$
23, 33	$\frac{\theta}{2-\theta}$
24, 34	$-1 - \frac{1}{\theta}$
26, 36	$-1 - \frac{4}{\theta^2}\int_0^1 x\log(x)(1-x)^{-2(1+\theta)/\theta}dx$
27, 37	$-1 - \frac{2}{\delta(2-\theta)}$
28, 38	$-1 - 4\int_0^1 -\log(-(1-t)^{-\theta}+1)(1-t-(1-t)^\theta+(1-t)^\theta t)/(\delta\theta)dt$
29, 39	$-1 - 4\int_0^1((1-(1-t)^{-\theta})^\delta-)/(-\theta\delta(1-t)^{-\theta-1}(1-(1-t)^{-\theta})^{\delta-1})dt$
30, 40	$-1 - 4\int_0^1 -\log\left(((1+t\delta)^{-\theta}-1)/((1+\delta)^{-\theta}-1)\right)$
	$*(1+t\delta-(1+t\delta)^\theta-(1+t\delta)^\theta t\delta)/(\theta\delta)dt$

Author(s)

Ulf Schepsmeier

References

Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman and Hall, London.

Czado, C., U. Schepsmeier, and A. Min (2012). Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling, 12(3), 229-255.

See Also

[CDVinePar2Tau](), [BiCopTau2Par]()

Examples

```
## Example 1: Gaussian copula
tt1 = BiCopPar2Tau(1,0.7)

# transform back
BiCopTau2Par(1,tt1)


## Example 2: Clayton copula
BiCopPar2Tau(3,1.3)
```

BiCopPDF   *Density of a bivariate copula*

Description

This function evaluates the probability density function (PDF) of a given parametric bivariate copula.

Usage

```
BiCopPDF(u1, u2, family, par, par2=0)
```

Arguments

u1,u2   Numeric vectors of equal length with values in [0,1].

family   An integer defining the bivariate copula family:
               0 = independence copula
               1 = Gaussian copula
               2 = Student t copula (t-copula)
               3 = Clayton copula
               4 = Gumbel copula
               5 = Frank copula
               6 = Joe copula
               7 = BB1 copula
               8 = BB6 copula
               9 = BB7 copula
               10 = BB8 copula
               13 = rotated Clayton copula (180 degrees; "survival Clayton")
               14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
               16 = rotated Joe copula (180 degrees; "survival Joe")
               17 = rotated BB1 copula (180 degrees; "survival BB1")
               18 = rotated BB6 copula (180 degrees; "survival BB6")
               19 = rotated BB7 copula (180 degrees; "survival BB7")
               20 = rotated BB8 copula (180 degrees; "survival BB8")
               23 = rotated Clayton copula (90 degrees)
               24 = rotated Gumbel copula (90 degrees)
               26 = rotated Joe copula (90 degrees)
               27 = rotated BB1 copula (90 degrees)
               28 = rotated BB6 copula (90 degrees)
               29 = rotated BB7 copula (90 degrees)
               30 = rotated BB8 copula (90 degrees)
               33 = rotated Clayton copula (270 degrees)
               34 = rotated Gumbel copula (270 degrees)
               36 = rotated Joe copula (270 degrees)
               37 = rotated BB1 copula (270 degrees)
               38 = rotated BB6 copula (270 degrees)
               39 = rotated BB7 copula (270 degrees)
               40 = rotated BB8 copula (270 degrees)

par   Copula parameter.

par2   Second parameter for bivariate copulas with two parameters (t, BB1, BB6, BB7, BB8; default: par2 = 0).

Value

A numeric vector of the bivariate copula density evaluated at u1 and u2.

Author(s)

Eike Brechmann

See Also

BiCopCDF, BiCopHfunc, BiCopSim

Examples

```
# simulate from a bivariate t-copula
simdata = BiCopSim(300,2,-0.7,par2=4)

# evaluate the density of the bivariate t-copula
u1 = simdata[,1]
u2 = simdata[,2]
BiCopPDF(u1,u2,2,-0.7,par2=4)
```

BiCopSelect   *Selection and maximum likelihood estimation of bivariate copula fam-*
                          ilies

Description

This function selects an appropriate bivariate copula family for given bivariate copula data using one
of a range of methods. The corresponding parameter estimates are obtained by maximum likelihood
estimation.

Usage

```
BiCopSelect(u1, u2, familyset=NA, selectioncrit="AIC",
            indeptest=FALSE, level=0.05)
```

Arguments

u1,u2   Data vectors of equal length with values in [0,1].

familyset   Vector of bivariate copula families to select from (the independence copula
               MUST NOT be specified in this vector, otherwise it will be selected). The vector
               has to include at least one bivariate copula family that allows for positive and
               one that allows for negative dependence. If familyset = NA (default), selection
               among all possible families is performed. Coding of bivariate copula families:
               1 = Gaussian copula
               2 = Student t copula (t-copula)
               3 = Clayton copula
               4 = Gumbel copula
               5 = Frank copula
               6 = Joe copula
               7 = BB1 copula
               8 = BB6 copula
               9 = BB7 copula
               10 = BB8 copula
               13 = rotated Clayton copula (180 degrees; "survival Clayton")
               14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
               16 = rotated Joe copula (180 degrees; "survival Joe")
               17 = rotated BB1 copula (180 degrees; "survival BB1")
               18 = rotated BB6 copula (180 degrees; "survival BB6")

19 = rotated BB7 copula (180 degrees; "survival BB7")
20 = rotated BB8 copula (180 degrees; "survival BB8")
23 = rotated Clayton copula (90 degrees)
24 = rotated Gumbel copula (90 degrees)
26 = rotated Joe copula (90 degrees)
27 = rotated BB1 copula (90 degrees)
28 = rotated BB6 copula (90 degrees)
29 = rotated BB7 copula (90 degrees)
30 = rotated BB8 copula (90 degrees)
33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

selectioncrit   Character indicating the criterion for bivariate copula selection. Possible choices:
                 `selectioncrit = "AIC"` (default) or `"BIC"`.

indeptest   Logical; whether a hypothesis test for the independence of u1 and u2 is per-
                 formed before bivariate copula selection (default: `indeptest = FALSE`; cp. BiCopIndTest).
                 The independence copula is chosen if the null hypothesis of independence can-
                 not be rejected.

level   Numeric; significance level of the independence test (default: `level = 0.05`).

Details

Copulas can be selected according to the Akaike and Bayesian Information Criteria (AIC and BIC,
respectively). First all available copulas are fitted using maximum likelihood estimation. Then the
criteria are computed for all available copula families (e.g., if u1 and u2 are negatively dependent,
Clayton, Gumbel, Joe, BB1, BB6, BB7 and BB8 and their survival copulas are not considered) and
the family with the minimum value is chosen. For observations $u_{i,j}$, $i = 1, ..., N$, $j = 1, 2$, the
AIC of a bivariate copula family c with parameter(s) $\boldsymbol{\theta}$ is defined as

$$AIC := -2 \sum_{i=1}^{N} \ln[c(u_{i,1}, u_{i,2}|\boldsymbol{\theta})] + 2k,$$

where $k = 1$ for one parameter copulas and $k = 2$ for the two parameter t-, BB1, BB6, BB7 and
BB8 copulas. Similarly, the BIC is given by

$$BIC := -2 \sum_{i=1}^{N} \ln[c(u_{i,1}, u_{i,2}|\boldsymbol{\theta})] + \ln(N)k.$$

Evidently, if the BIC is chosen, the penalty for two parameter families is stronger than when using
the AIC.

Additionally a test for independence can be performed beforehand.

Value

family   The selected bivariate copula family.

par, par2   The estimated bivariate copula parameter(s).

p.value.indeptest
                 P-value of the independence test if performed.

Note

When the bivariate t-copula is considered and the degrees of freedom are estimated to be larger than 30, then the bivariate Gaussian copula is taken into account instead. Similarly, when BB1 (Clayton-Gumbel), BB6 (Joe-Gumbel), BB7 (Joe-Clayton) or BB8 (Joe-Frank) copulas are considered and the parameters are estimated to be very close to one of their boundary cases, the respective one parameter copula is taken into account instead.

Author(s)

Eike Brechmann, Jeffrey Dissmann

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki (Eds.), Proceedings of the Second International Symposium on Information Theory Budapest, Akademiai Kiado, pp. 267-281.

Brechmann, E. C. (2010). Truncated and simplified regular vines and their applications. Diploma thesis, Technische Universitaet Muenchen.
http://mediatum.ub.tum.de/doc/1079285/1079285.pdf.

Manner, H. (2007). Estimation and model selection of copulas with an application to exchange rates. METEOR research memorandum 07/056, Maastricht University.

Schwarz, G. E. (1978). Estimating the dimension of a model. Annals of Statistics 6 (2), 461-464.

See Also

CDVineCopSelect, BiCopIndTest

Examples

```
## Example 1: Gaussian copula with large dependence parameter
par1 = 0.7
fam1 = 1
dat1 = BiCopSim(500,fam1,par1)

# select the bivariate copula family and estimate the parameter(s)
cop1 = BiCopSelect(dat1[,1],dat1[,2],familyset=c(1:10),indeptest=FALSE,level=0.05)
cop1$family
cop1$par
cop1$par2


## Example 2: Gaussian copula with small dependence parameter
par2 = 0.01
fam2 = 1
dat2 = BiCopSim(500,fam2,par2)

# select the bivariate copula family and estimate the parameter(s)
cop2 = BiCopSelect(dat2[,1],dat2[,2],familyset=c(1:10),indeptest=TRUE,level=0.05)
cop2$family
cop2$par
cop2$par2


## Example 3: empirical data
```

```
data(worldindices)
cop3 = BiCopSelect(worldindices[,1],worldindices[,4],familyset=c(1:10,13,14,16,23,24,26))
cop3$family
cop3$par
cop3$par2
```

BiCopSim   *Simulation from a bivariate copula*

Description

This function simulates from a given parametric bivariate copula.

Usage

```
BiCopSim(N, family, par, par2=0)
```

Arguments

N	Number of bivariate observations simulated.
family	An integer defining the bivariate copula family:
	0 = independence copula
	1 = Gaussian copula
	2 = Student t copula (t-copula)
	3 = Clayton copula
	4 = Gumbel copula
	5 = Frank copula
	6 = Joe copula
	7 = BB1 copula
	8 = BB6 copula
	9 = BB7 copula
	10 = BB8 copula
	13 = rotated Clayton copula (180 degrees; "survival Clayton")
	14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
	16 = rotated Joe copula (180 degrees; "survival Joe")
	17 = rotated BB1 copula (180 degrees; "survival BB1")
	18 = rotated BB6 copula (180 degrees; "survival BB6")
	19 = rotated BB7 copula (180 degrees; "survival BB7")
	20 = rotated BB8 copula (180 degrees; "survival BB8")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)
	27 = rotated BB1 copula (90 degrees)
	28 = rotated BB6 copula (90 degrees)
	29 = rotated BB7 copula (90 degrees)
	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)

       39 = rotated BB7 copula (270 degrees)

       40 = rotated BB8 copula (270 degrees)

par	Copula parameter.
par2	Second parameter for bivariate copulas with two parameters (t, BB1,BB6, BB7, BB8; default: `par2 = 0`).

Value

An `N` x 2 matrix of data simulated from the bivariate copula.

Author(s)

Ulf Schepsmeier

See Also

[BiCopCDF](), [BiCopPDF](), [CDVineSim]()

Examples

```
# simulate from a bivariate t-copula
simdata = BiCopSim(300,2,-0.7,par2=4)
```

BiCopTau2Par	*Parameter of a bivariate copula for a given Kendall's tau value*

Description

This function computes the parameter of a one parameter bivariate copula for a given value of Kendall's tau.

Usage

```
BiCopTau2Par(family, tau)
```

Arguments

tau	Kendall's tau value (numeric in [-1,1]).
family	An integer defining the bivariate copula family:
	0 = independence copula
	1 = Gaussian copula
	3 = Clayton copula
	4 = Gumbel copula
	5 = Frank copula
	6 = Joe copula
	13 = rotated Clayton copula (180 degrees; "survival Clayton")
	14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
	16 = rotated Joe copula (180 degrees; "survival Joe")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)

33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
Note that two parameter bivariate copula families cannot be used.

Value

Parameter corresponding to the bivariate copula family and the value of Kendall's tau (τ).

No.	Parameter
1, 2	$\sin(\tau\frac{\pi}{2})$
3, 13	$\max(0, 2\frac{\tau}{1-\tau})$
4, 14	$\max(1, \frac{1}{1-\tau})$
5	no closed form expression (numerical inversion)
6, 16	no closed form expression (numerical inversion)
23, 33	$\max(0, 2\frac{\tau}{1+\tau})$
24, 34	$\min(-1, -\frac{1}{1+\tau})$
26, 36	no closed form expression (numerical inversion)

Author(s)

Jakob Stoeber, Eike Brechmann

References

Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman and Hall, London.

Czado, C., U. Schepsmeier, and A. Min (2012). Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling, 12(3), 229-255.

See Also

BiCopTau2Par

Examples

```
## Example 1: Gaussian copula
tt1 = BiCopTau2Par(1,0.5)

# transform back
BiCopPar2Tau(1,tt1)


## Example 2: Clayton copula
BiCopTau2Par(3,0.4)
```

BiCopVuongClarke	*Scoring goodness-of-fit test based on Vuong and Clarke tests for bivariate copula data*

Description

Based on the Vuong and Clarke tests this function computes a goodness-of-fit score for each bivariate copula family under consideration. For each possible pair of copula families the Vuong and the Clarke tests decides which of the two families fits the given data best and assigns a score—pro or contra a copula family—according to this decision.

Usage

```
BiCopVuongClarke(u1, u2, familyset=NA,
                 correction=FALSE, level=0.05)
```

Arguments

u1,u2   Data vectors of equal length with values in [0,1].

familyset   An integer vector of bivariate copula families under consideration, i.e., which are
               compared in the goodness-of-fit test. If familyset = NA (default), all possible
               families are compared. Possible families are:
               0 = independence copula
               1 = Gaussian copula
               2 = Student t copula (t-copula)
               3 = Clayton copula
               4 = Gumbel copula
               5 = Frank copula
               6 = Joe copula
               7 = BB1 copula
               8 = BB6 copula
               9 = BB7 copula
               10 = BB8 copula
               13 = rotated Clayton copula (180 degrees; "survival Clayton")
               14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
               16 = rotated Joe copula (180 degrees; "survival Joe")
               17 = rotated BB1 copula (180 degrees; "survival BB1")
               18 = rotated BB6 copula (180 degrees; "survival BB6")
               19 = rotated BB7 copula (180 degrees; "survival BB7")
               20 = rotated BB8 copula (180 degrees; "survival BB8")
               23 = rotated Clayton copula (90 degrees)
               24 = rotated Gumbel copula (90 degrees)
               26 = rotated Joe copula (90 degrees)
               27 = rotated BB1 copula (90 degrees)
               28 = rotated BB6 copula (90 degrees)
               29 = rotated BB7 copula (90 degrees)
               30 = rotated BB8 copula (90 degrees)
               33 = rotated Clayton copula (270 degrees)
               34 = rotated Gumbel copula (270 degrees)
               36 = rotated Joe copula (270 degrees)
               37 = rotated BB1 copula (270 degrees)
               38 = rotated BB6 copula (270 degrees)
               39 = rotated BB7 copula (270 degrees)
               40 = rotated BB8 copula (270 degrees)

correction   Correction for the number of parameters. Possible choices: correction = FALSE
               (no correction; default), "Akaike" and "Schwarz".

level   Numerical; significance level of the tests (default: level = 0.05).

Details

The Vuong as well as the Clarke test compare two models against each other and based on their null hypothesis, allow for a statistically significant decision among the two models (see the documentations of CDVineVuongTest and CDVineClarkeTest for descriptions of the two tests). In the goodness-of-fit test proposed by Belgorodski (2010) this is used for bivariate copula selection. It compares a model 0 to all other possible models under consideration. If model 0 is favored over another model, a score of "+1" is assigned and similarly a score of "-1" if the other model is determined to be superior. No score is assigned, if the respective test cannot discriminate between two models. Both tests can be corrected for the numbers of parameters used in the copulas. Either no correction (correction = FALSE), the Akaike correction (correction = "Akaike") or the parsimonious Schwarz correction (correction = "Schwarz") can be used.

The models compared here are bivariate parametric copulas and we would like to determine which family fits the data better than the other families. E.g., if we would like to test the hypothesis that the bivariate Gaussian copula fits the data best, then we compare the Gaussian copula against all other copulas under consideration. In doing so, we investigate the null hypothesis "The Gaussian copula fits the data better than all other copulas under consideration", which corresponds to $k-1$ times the hypothesis "The Gaussian copula C_j fits the data better than copula C_i" for all $i = 1, ..., k, i \neq j$, where k is the number of bivariate copula families under consideration (length of familyset). This procedure is done not only for one family but for all families under consideration, i.e., two scores, one based on the Vuong and one based on the Clarke test, are returned for each bivariate copula family. If used as a goodness-of-fit procedure, the family with the highest score should be selected.

For more and detailed information about the goodness-of-fit test see Belgorodski (2010).

Value

A matrix with Vuong test scores in the first and Clarke test scores in the second row. Column names correspond to bivariate copula families (see above).

Author(s)

Ulf Schepsmeier, Eike Brechmann, Natalia Belgorodski

References

Belgorodski, N. (2010) Selecting pair-copula families for regular vines with application to the multivariate analysis of European stock market indices Diploma thesis, Technische Universitaet Muenchen. http://mediatum.ub.tum.de/doc/1079284/1079284.pdf.

Clarke, K. A. (2007). A Simple Distribution-Free Test for Nonnested Model Selection. Political Analysis, 15, 347-363.

Vuong, Q. H. (1989). Ratio tests for model selection and non-nested hypotheses. Econometrica 57 (2), 307-333.

See Also

BiCopGofKendall, CDVineVuongTest, CDVineClarkeTest, BiCopSelect

Examples

```
# simulate from a t-copula
dat = BiCopSim(500,2,0.7,5)
```

```
# apply the test for families 1-10
vcgof = BiCopVuongClarke(dat[,1],dat[,2],familyset=c(1:10))

# display the Vuong test scores
vcgof[1,]
```

CDVineAIC–BIC   *AIC and BIC of C- and D-vine copula models*

Description

These functions calculate the Akaike and Bayesian Information criteria of d-dimensional C- and D-vine copula models for a given copula data set.

Usage

```
CDVineAIC(data, family, par,
          par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2), type)
CDVineBIC(data, family, par,
          par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2), type)
```

Arguments

data   An N x d data matrix (with uniform margins).

family   A d*(d-1)/2 integer vector of C-/D-vine pair-copula families with values
              0 = independence copula
              1 = Gaussian copula
              2 = Student t copula (t-copula)
              3 = Clayton copula
              4 = Gumbel copula
              5 = Frank copula
              6 = Joe copula
              7 = BB1 copula
              8 = BB6 copula
              9 = BB7 copula
              10 = BB8 copula
              13 = rotated Clayton copula (180 degrees; "survival Clayton")
              14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
              16 = rotated Joe copula (180 degrees; "survival Joe")
              17 = rotated BB1 copula (180 degrees; "survival BB1")
              18 = rotated BB6 copula (180 degrees; "survival BB6")
              19 = rotated BB7 copula (180 degrees; "survival BB7")
              20 = rotated BB8 copula (180 degrees; "survival BB8")
              23 = rotated Clayton copula (90 degrees)
              24 = rotated Gumbel copula (90 degrees)
              26 = rotated Joe copula (90 degrees)
              27 = rotated BB1 copula (90 degrees)
              28 = rotated BB6 copula (90 degrees)
              29 = rotated BB7 copula (90 degrees)
              30 = rotated BB8 copula (90 degrees)
              33 = rotated Clayton copula (270 degrees)

34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

par   A d*(d-1)/2 vector of pair-copula parameters.

par2   A d*(d-1)/2 vector of second parameters for two parameter pair-copula families
                (default: par2 = rep(0,dim(data)[2]*(dim(data)[2]-1)/2)).

type   Type of the vine model:
                1 or "CVine" = C-vine
                2 or "DVine" = D-vine

Details

If k denotes the number of parameters of a C-vine copula model with log-likelihood l_{CVine} and parameter set $\boldsymbol{\theta}$, then the Akaike Information Criterion (AIC) by Akaike (1973) is defined as

$$AIC := -2l_{CVine}\left(\boldsymbol{\theta}|\boldsymbol{u}\right) + 2k,$$

for observations $\boldsymbol{u} = (\boldsymbol{u}'_1, ..., \boldsymbol{u}'_N)'$.

Similarly, the Bayesian Information Criterion (BIC) by Schwarz (1978) is given by

$$BIC := -2l_{CVine}\left(\boldsymbol{\theta}|\boldsymbol{u}\right) + \log(N)k.$$

The AIC and BIC expressions for D-vine copula models are defined accordingly.

Value

AIC, BIC   The computed AIC or BIC value, respectively.

pair.AIC, pair.BIC

                An array of individual contributions to the AIC or BIC value for each pair-
                copula, respectively. Note: AIC = sum(pair.AIC) and similarly BIC = sum(pair.BIC).

Author(s)

Eike Brechmann

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki (Eds.), Proceedings of the Second International Symposium on Information Theory Budapest, Akademiai Kiado, pp. 267-281.

Schwarz, G. E. (1978). Estimating the dimension of a model. Annals of Statistics 6 (2), 461-464.

See Also

CDVineLogLik, CDVineVuongTest, CDVineClarkeTest

Examples

```
## Example 1: 3-dimensional D-vine model with Gaussian pair-copulas
data(worldindices)
Data = as.matrix(worldindices)[,1:3]
fam1 = c(1,1,1)
par1 = c(0.2,0.3,0.4)

# calculate AIC and BIC
CDVineAIC(Data,fam1,par1,type=2)
CDVineBIC(Data,fam1,par1,type=2)


## Example 2: 6-dimensional C-vine model with Student t pair-copulas
## with 5 degrees of freedom
data(worldindices)
Data = as.matrix(worldindices)
dd = dim(Data)[2]*(dim(Data)[2]-1)/2
fam2 = rep(2,dd)
par2 = rep(0.5,dd)
nu2 = rep(5,dd)

# calculate AIC and BIC
CDVineAIC(Data,fam2,par2,nu2,type=1)
CDVineBIC(Data,fam2,par2,nu2,type=1)


## Example 3: 4-dimensional C-vine model with mixed pair-copulas
fam3 = c(5,1,3,14,3,2)
par3 = c(0.9,0.3,0.2,1.1,0.2,0.7)
nu3 = c(0,0,0,0,0,7)

# calculate AIC and BIC
CDVineAIC(Data[,1:4],fam3,par3,nu3,type=1)
CDVineBIC(Data[,1:4],fam3,par3,nu3,type=1)
```

CDVineClarkeTest   *Clarke test comparing two vine copula models*

Description

This function performs a Clarke test between two d-dimensional C- or D-vine copula models, respectively.

Usage

```
CDVineClarkeTest(data, Model1.order=1:dim(data)[2],
                 Model2.order=1:dim(data)[2], Model1.family,
                 Model2.family, Model1.par, Model2.par,
                 Model1.par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2),
                 Model2.par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2),
                 Model1.type, Model2.type)
```

Arguments

data
: An N x d data matrix (with uniform margins).

Model1.order, Model2.order
: Two numeric vectors giving the order of the variables in the first D-vine trees or of the C-vine root nodes in models 1 and 2 (default: Model1.order and Model2.order = 1:dim(data)[2], i.e., standard order).

Model1.family, Model2.family
: Two d*(d-1)/2 numeric vectors of the pair-copula families of models 1 and 2, respectively, with values
  0 = independence copula
  1 = Gaussian copula
  2 = Student t copula (t-copula)
  3 = Clayton copula
  4 = Gumbel copula
  5 = Frank copula
  6 = Joe copula
  7 = BB1 copula
  8 = BB6 copula
  9 = BB7 copula
  10 = BB8 copula
  13 = rotated Clayton copula (180 degrees; "survival Clayton")
  14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
  16 = rotated Joe copula (180 degrees; "survival Joe")
  17 = rotated BB1 copula (180 degrees; "survival BB1")
  18 = rotated BB6 copula (180 degrees; "survival BB6")
  19 = rotated BB7 copula (180 degrees; "survival BB7")
  20 = rotated BB8 copula (180 degrees; "survival BB8")
  23 = rotated Clayton copula (90 degrees)
  24 = rotated Gumbel copula (90 degrees)
  26 = rotated Joe copula (90 degrees)
  27 = rotated BB1 copula (90 degrees)
  28 = rotated BB6 copula (90 degrees)
  29 = rotated BB7 copula (90 degrees)
  30 = rotated BB8 copula (90 degrees)
  33 = rotated Clayton copula (270 degrees)
  34 = rotated Gumbel copula (270 degrees)
  36 = rotated Joe copula (270 degrees)
  37 = rotated BB1 copula (270 degrees)
  38 = rotated BB6 copula (270 degrees)
  39 = rotated BB7 copula (270 degrees)
  40 = rotated BB8 copula (270 degrees)

Model1.par, Model2.par
: Two d*(d-1)/2 numeric vectors of the (first) copula parameters of models 1 and 2, respectively.

Model1.par2, Model2.par2
: Two d*(d-1)/2 numeric vectors of the second copula parameters of models 1 and 2, respectively; necessary for t, BB1, BB6, BB7 and BB8 copulas. If no such families are included in Model1.family/Model2.family, these arguments do not need to be specified (default: Model1.par2 and Model2.par2 = rep(0,dim(data)[2]*(dim(da

Model1.type, Model2.type
: Type of the respective vine model:

       1 or "CVine" = C-vine
       2 or "DVine" = D-vine

Details

The test proposed by Clarke (2007) allows to compare non-nested models. For this let c_1 and c_2 be two competing vine copulas in terms of their densities and with estimated parameter sets $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$. The null hypothesis of statistical indistinguishability of the two models is

$$H_0 : P(m_i > 0) = 0.5 \; \forall i = 1, .., N,$$

where $m_i := \log \left[\frac{c_1(\boldsymbol{u}_i | \hat{\boldsymbol{\theta}}_1)}{c_2(\boldsymbol{u}_i | \hat{\boldsymbol{\theta}}_2)} \right]$ for observations $\boldsymbol{u}_i, \; i = 1, ..., N$.

Since under statistical equivalence of the two models the log likelihood ratios of the single observations are uniformly distributed around zero and in expectation 50% of the log likelihood ratios greater than zero, the tets statistic

$$\mathtt{statistic} := B = \sum_{i=1}^{N} \mathbf{1}_{(0,\infty)}(m_i),$$

where $\mathbf{1}$ is the indicator function, is distributed Binomial with parameters N and $p = 0.5$, and critical values can easily be obtained. Model 1 is interpreted as statistically equivalent to model 2 if B is not significantly different from the expected value $Np = \frac{N}{2}$.

Like AIC and BIC, the Clarke test statistic may be corrected for the number of parameters used in the models. There are two possible corrections; the Akaike and the Schwarz corrections, which correspond to the penalty terms in the AIC and the BIC, respectively.

Value

`statistic, statistic.Akaike, statistic.Schwarz`
               Test statistics without correction, with Akaike correction and with Schwarz correction.

`p.value, p.value.Akaike, p.value.Schwarz`
               P-values of tests without correction, with Akaike correction and with Schwarz correction.

Author(s)

Jeffrey Dissmann, Ulf Schepsmeier, Eike Brechmann

References

Clarke, K. A. (2007). A Simple Distribution-Free Test for Nonnested Model Selection. Political Analysis, 15, 347-363.

See Also

CDVineVuongTest, CDVineAIC, CDVineBIC

Examples

```
# compare 6-dimensional C-vine copula models
# with Gaussian and with Student t pair-copulas
d = 6
dd = d*(d-1)/2
fam1 = rep(1,dd)
par1 = c(0.2,0.69,0.73,0.22,-0.09,0.51,0.32,0.01,0.82,0.01,
         -0.2,-0.32,-0.19,-0.17,-0.06)

fam2 = rep(2,dd)
par2 = par1
nu2 = rep(4,dd)

# simulate a sample of size 300 from the first C-vine copula model
simdata = CDVineSim(300,fam1,par1,type=1)

# compare the two models based on this sample
clarke = CDVineClarkeTest(simdata,1:d,1:d,fam1,fam2,par1,par2,
                         Model2.par2=nu2,Model1.type=1,Model2.type=1)
clarke$statistic
clarke$statistic.Schwarz
clarke$p.value
clarke$p.value.Schwarz
```

CDVineCopSelect | *Sequential copula selection and estimation of C- and D-vine copula models*

Description

This function fits either a C- or a D-vine copula model to a d-dimensional copula data set. Appropriate pair-copula families are selected using `BiCopSelect` and estimated sequentially.

Usage

```
CDVineCopSelect(data, familyset=NA, type, selectioncrit="AIC",
               indeptest=FALSE, level=0.05)
```

Arguments

data
: An N x d data matrix (with uniform margins).

familyset
: An integer vector of pair-copula families to select from (the independence copula MUST NOT be specified in this vector unless one wants to fit an independence vine!). The vector has to include at least one pair-copula family that allows for positive and one that allows for negative dependence. If `familyset = NA` (default), selection among all possible families is performed. The coding of pair-copula families is shown below.

type
: Type of the vine model:
: 1 or "CVine" = C-vine
: 2 or "DVine" = D-vine

selectioncrit
: Character indicating the criterion for pair-copula selection. Possible choices: `selectioncrit = "AIC"` (default) or `"BIC"` (see `BiCopSelect`).

indeptest	Logical; whether a hypothesis test for the independence of u1 and u2 is performed before bivariate copula selection (default: indeptest = FALSE; cp. BiCopIndTest). The independence copula is chosen for a (conditional) pair if the null hypothesis of independence cannot be rejected.
level	Numeric; significance level of the independence test (default: level = 0.05).

Value

family	A d*(d-1)/2 vector of pair-copula families with values
	0 = independence copula
	1 = Gaussian copula
	2 = Student t copula (t-copula)
	3 = Clayton copula
	4 = Gumbel copula
	5 = Frank copula
	6 = Joe copula
	7 = BB1 copula
	8 = BB6 copula
	9 = BB7 copula
	10 = BB8 copula
	13 = rotated Clayton copula (180 degrees; "survival Clayton")
	14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
	16 = rotated Joe copula (180 degrees; "survival Joe")
	17 = rotated BB1 copula (180 degrees; "survival BB1")
	18 = rotated BB6 copula (180 degrees; "survival BB6")
	19 = rotated BB7 copula (180 degrees; "survival BB7")
	20 = rotated BB8 copula (180 degrees; "survival BB8")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)
	27 = rotated BB1 copula (90 degrees)
	28 = rotated BB6 copula (90 degrees)
	29 = rotated BB7 copula (90 degrees)
	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)
	39 = rotated BB7 copula (270 degrees)
	40 = rotated BB8 copula (270 degrees)
par	A d*(d-1)/2 vector of pair-copula parameters.
par2	A d*(d-1)/2 vector of second pair-copula parameters for the t-, BB1, BB6, BB7 and BB8 copulas.

Author(s)

Eike Brechmann

See Also

BiCopSelect, CDVineSeqEst

Examples

```
# simulate from a 4-dimensional D-vine copula with mixed pair-copulas
d = 4
dd = d*(d-1)/2
family = c(1,2,3,4,7,3)
par = c(0.5,0.4,2,1.5,1.2,1.5)
par2 = c(0,5,0,0,2,0)
type = 2
simdata = CDVineSim(1000,family,par,par2,type)

# determine appropriate pair-copula families and parameters
# of a D-vine structure
CDVineCopSelect(simdata,type=2,familyset=c(1:10,13,14,23,24))
```

CDVineLogLik   *Log-likelihood of C- and D-vine copula models*

Description

This function calculates the log-likelihood of d-dimensional C- and D-vine copula models for a given copula data set.

Usage

```
CDVineLogLik(data, family, par,
             par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2), type)
```

Arguments

data   An N x d data matrix (with uniform margins).

family   A d*(d-1)/2 integer vector of C-/D-vine pair-copula families with values
            0 = independence copula
            1 = Gaussian copula
            2 = Student t copula (t-copula)
            3 = Clayton copula
            4 = Gumbel copula
            5 = Frank copula
            6 = Joe copula
            7 = BB1 copula
            8 = BB6 copula
            9 = BB7 copula
            10 = BB8 copula
            13 = rotated Clayton copula (180 degrees; "survival Clayton")
            14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
            16 = rotated Joe copula (180 degrees; "survival Joe")
            17 = rotated BB1 copula (180 degrees; "survival BB1")
            18 = rotated BB6 copula (180 degrees; "survival BB6")
            19 = rotated BB7 copula (180 degrees; "survival BB7")
            20 = rotated BB8 copula (180 degrees; "survival BB8")
            23 = rotated Clayton copula (90 degrees)
            24 = rotated Gumbel copula (90 degrees)

26 = rotated Joe copula (90 degrees)
27 = rotated BB1 copula (90 degrees)
28 = rotated BB6 copula (90 degrees)
29 = rotated BB7 copula (90 degrees)
30 = rotated BB8 copula (90 degrees)
33 = rotated Clayton copula (270 degrees)
34 = rotated Gumbel copula (270 degrees)
36 = rotated Joe copula (270 degrees)
37 = rotated BB1 copula (270 degrees)
38 = rotated BB6 copula (270 degrees)
39 = rotated BB7 copula (270 degrees)
40 = rotated BB8 copula (270 degrees)

par	A d*(d-1)/2 vector of pair-copula parameters.
par2	A d*(d-1)/2 vector of second parameters for two parameter pair-copula families (default: par2 = rep(0,dim(data)[2]*(dim(data)[2]-1)/2)).
type	Type of the vine model: 1 or "CVine" = C-vine 2 or "DVine" = D-vine

Details

Let $\boldsymbol{u} = (\boldsymbol{u}'_1, ..., \boldsymbol{u}'_N)'$ be d-dimensional observations with $\boldsymbol{u}_i = (u_{i,1}, ..., u_{i,d})' \in [0,1]^d$, $i = 1, ..., N$. Then the log-likelihood of a C-vine copula is given by

$$\texttt{loglik} := l_{CVine}(\boldsymbol{\theta}|\boldsymbol{u}) = \sum_{i=1}^{N} \sum_{j=1}^{d-1} \sum_{k=1}^{d-j} \ln \left[c_{j,j+k|1,...,j-1} \right],$$

where

$$c_{j,j+k|1,...,j-1} := c_{j,j+k|1:(j-1)}(F(u_{i,j}|u_{i,1}, ..., u_{i,j-1}), F(u_{i,j+k}|u_{i,1}, ..., u_{i,j-1})|\boldsymbol{\theta}_{j,j+k|1,...,j-1})$$

denote pair-copulas with parameter(s) $\boldsymbol{\theta}_{j,j+k|1,...,j-1}$.

Similarly, the log-likelihood of a d-dimensional D-vine copula is

$$\texttt{loglik} := l_{DVine}(\boldsymbol{\theta}|\boldsymbol{u}) = \sum_{i=1}^{N} \sum_{j=1}^{d-1} \sum_{k=1}^{d-j} \ln \left[c_{k,k+j|k+1,...,k+j-1} \right],$$

again with pair-copula densities denoted by

$$c_{k,k+j|k+1,...,k+j-1} :=$$

$$c_{k,k+j|k+1,...,k+j-1}(F(u_{i,k}|u_{i,k+1}, ..., u_{i,k+j-1}), F(u_{i,k+j}|u_{i,k+1}, ..., u_{i,k+j-1})|\boldsymbol{\theta}_{k,k+j|k+1,...,k+j-1}).$$

Conditional distribution functions in both expressions are obtained recursively using the relationship

$$h(u|\boldsymbol{v}, \boldsymbol{\theta}) := F(u|\boldsymbol{v}) = \frac{\partial C_{uv_j|\boldsymbol{v}_{-j}}(F(u|\boldsymbol{v}_{-j}), F(v_j|\boldsymbol{v}_{-j}))}{\partial F(v_j|\boldsymbol{v}_{-j})},$$

where $C_{uv_j|\boldsymbol{v}_{-j}}$ is a bivariate copula distribution function with parameter(s) $\boldsymbol{\theta}$ and $\boldsymbol{v}_{-j}$ denotes a vector with the j-th component v_j removed. The notation of h-functions is introduced for convenience. For more details see Aas et al. (2009).

Note that both log-likelihoods can also be written as $\texttt{loglik} = \sum_{k=1}^{d(d-1)/2} ll_k$, where ll_k are the individual contributions to the log-likelihood of each pair-copula.

Value

loglik	The calculated log-likelihood value of the C- or D-vine copula model.
ll	An array of individual contributions to the log-likelihood for each pair-copula. Note: `loglik = sum(ll)`.
vv	The stored transformations (h-functions) which may be used for posterior updates.

Author(s)

Carlos Almeida, Ulf Schepsmeier

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

See Also

BiCopHfunc, CDVineMLE, CDVineAIC, CDVineBIC

Examples

```
## Example 1: 3-dimensional D-vine model with Gaussian pair-copulas
data(worldindices)
Data = as.matrix(worldindices)[,1:3]
fam1 = c(1,1,1)
par1 = c(0.2,0.3,0.4)

# calculate the log-likelihood
logLik1 = CDVineLogLik(Data,fam1,par1,type=2)

# check the above formula
sum(logLik1$ll)
logLik1$loglik


## Example 2: 6-dimensional C-vine model with Student t pair-copulas
## with 5 degrees of freedom
data(worldindices)
Data = as.matrix(worldindices)
dd = dim(Data)[2]*(dim(Data)[2]-1)/2
fam2 = rep(2,dd)
par2 = rep(0.5,dd)
nu2 = rep(5,dd)

# calculate the log-likelihood
logLik2 = CDVineLogLik(Data,fam2,par2,nu2,type=1)
logLik2$loglik


## Example 3: 4-dimensional C-vine model with mixed pair-copulas
fam3 = c(5,1,3,14,3,2)
par3 = c(0.9,0.3,0.2,1.1,0.2,0.7)
nu3 = c(0,0,0,0,0,7)
```

```
# calculate the log-likelihood
logLik3 = CDVineLogLik(Data[,1:4],fam3,par3,nu3,type=2)
logLik3$loglik
```

CDVineMLE   *Maximum likelihood estimation of C- and D-vine copula models*

Description

This function calculates the MLE of C- or D-vine copula model parameters using sequential estimates as initial values (if not provided).

Usage

```
CDVineMLE(data, family, start=NULL, start2=NULL, type, maxit=200,
         max.df=30, max.BB=list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1)), ...)
```

Arguments

data	An N x d data matrix (with uniform margins).
family	A d*(d-1)/2 integer vector of C-/D-vine pair-copula families with values
	0 = independence copula
	1 = Gaussian copula
	2 = Student t copula (t-copula)
	3 = Clayton copula
	4 = Gumbel copula
	5 = Frank copula
	6 = Joe copula
	7 = BB1 copula
	8 = BB6 copula
	9 = BB7 copula
	10 = BB8 copula
	13 = rotated Clayton copula (180 degrees; "survival Clayton")
	14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
	16 = rotated Joe copula (180 degrees; "survival Joe")
	17 = rotated BB1 copula (180 degrees; "survival BB1")
	18 = rotated BB6 copula (180 degrees; "survival BB6")
	19 = rotated BB7 copula (180 degrees; "survival BB7")
	20 = rotated BB8 copula (180 degrees; "survival BB8")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)
	27 = rotated BB1 copula (90 degrees)
	28 = rotated BB6 copula (90 degrees)
	29 = rotated BB7 copula (90 degrees)
	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)

	39 = rotated BB7 copula (270 degrees)
	40 = rotated BB8 copula (270 degrees)
start	A d*(d-1)/2 numeric vector of starting values for C-/D-vine pair-copula parameters (optional; otherwise they are calculated via [CDVineSeqEst](#); default: start = NULL).
start2	A d*(d-1)/2 numeric vector of starting values for second C-/D-vine pair-copula parameters (optional; otherwise they are calculated via [CDVineSeqEst](#); default: start2 = NULL).
type	Type of the vine model: 1 or ″CVine″ = C-vine 2 or ″DVine″ = D-vine
maxit	The maximum number of iteration steps (optional; default: maxit = 200).
max.df	Numeric; upper bound for the estimation of the degrees of freedom parameter of the t-copula (default: max.df = 30; for more details see [BiCopEst](#)).
max.BB	List; upper bounds for the estimation of the two parameters (in absolute values) of the BB1, BB6, BB7 and BB8 copulas (default: max.BB = list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1))).
...	Additional control parameters for [optim](#).

Value

par	Estimated (first) C-/D-vine pair-copula parameters.
par2	Estimated second C-/D-vine pair-copula parameters for families with two parameters (t, BB1,BB6, BB7, BB8). All other entries are zero.
loglik	Optimized log-likelihood value corresponding to the estimated pair-copula parameters.
convergence	An integer code indicating either successful convergence (convergence = 0) or an error (cp. [optim](#); the CDVineMLE-function uses the "L-BFGS-B" method): 1 = the iteration limit maxit has been reached 51 = a warning from the "L-BFGS-B" method; see component message for further details 52 = an error from the "L-BFGS-B" method; see component message for further details
message	A character string giving any additional information returned by [optim](#), or NULL.

Author(s)

Carlos Almeida, Ulf Schepsmeier

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

See Also

[CDVineLogLik](#), [CDVineSeqEst](#)

Examples

```
## Example 1: 4-dimensional D-vine model with Gaussian pair-copulas
data(worldindices)
Data = as.matrix(worldindices)[,1:4]
fam = rep(1,6)

# maximum likelihood estimation
CDVineMLE(Data,family=fam,type=2,maxit=100)


## Example 2: 4-dimensional D-vine model with mixed pair-copulas
fam2 = c(5,1,3,14,3,2)

# sequential estimation
m = CDVineSeqEst(Data,family=fam2,type=2)
m

# calculate the log-likelihood
LogLik0 = CDVineLogLik(Data,fam2,m$par,m$par2,type=2)
LogLik0$loglik

# maximum likelihood estimation
CDVineMLE(Data,family=fam2,type=2,maxit=5)  # 5 iterations
CDVineMLE(Data,family=fam2,type=2)  # default: 200 iterations
```

CDVinePar2Tau   *Kendall's tau values of a vine copula model*

Description

This function computes the values of Kendall's tau corresponding to the parameters of a C- or D-vine copula model.

Usage

```
CDVinePar2Tau(family, par, par2=rep(0,length(family)))
```

Arguments

family   A d*(d-1)/2 integer vector of C-/D-vine pair-copula families with values
                0 = independence copula
                1 = Gaussian copula
                2 = Student t copula (t-copula)
                3 = Clayton copula
                4 = Gumbel copula
                5 = Frank copula
                6 = Joe copula
                7 = BB1 copula
                8 = BB6 copula
                9 = BB7 copula
                10 = BB8 copula
                13 = rotated Clayton copula (180 degrees; "survival Clayton")
                14 = rotated Gumbel copula (180 degrees; "survival Gumbel")

        16 = rotated Joe copula (180 degrees; "survival Joe")
        17 = rotated BB1 copula (180 degrees; "survival BB1")
        18 = rotated BB6 copula (180 degrees; "survival BB6")
        19 = rotated BB7 copula (180 degrees; "survival BB7")
        20 = rotated BB8 copula (180 degrees; "survival BB8")
        23 = rotated Clayton copula (90 degrees)
        24 = rotated Gumbel copula (90 degrees)
        26 = rotated Joe copula (90 degrees)
        27 = rotated BB1 copula (90 degrees)
        28 = rotated BB6 copula (90 degrees)
        29 = rotated BB7 copula (90 degrees)
        30 = rotated BB8 copula (90 degrees)
        33 = rotated Clayton copula (270 degrees)
        34 = rotated Gumbel copula (270 degrees)
        36 = rotated Joe copula (270 degrees)
        37 = rotated BB1 copula (270 degrees)
        38 = rotated BB6 copula (270 degrees)
        39 = rotated BB7 copula (270 degrees)
        40 = rotated BB8 copula (270 degrees)

par   A $d*(d-1)/2$ vector of pair-copula parameters.

par2   A $d*(d-1)/2$ vector of second parameters for pair-copula families with two parameters (t, BB1, BB6, BB7, BB8; default: `par2 = rep(0,length(family))`).

Value

A $d*(d-1)/2$ vector of theoretical Kendall's tau values corresponding to the given pair-copula families and parameters.

Author(s)

Eike Brechmann

See Also

[BiCopPar2Tau](#)

Examples

```
# specify the vine
fam3 = c(5,1,3,14,3,2)
par3 = c(0.9,0.3,0.2,1.1,0.2,0.7)
nu3 = c(0,0,0,0,0,7)

# compute the corresponding Kendall's tau values
tau = CDVinePar2Tau(fam3,par3,nu3)
```

CDVineSeqEst   *Sequential estimation of C- and D-vine copula models*

Description

This function sequentially estimates the pair-copula parameters of d-dimensional C- or D-vine copula models.

Usage

```
CDVineSeqEst(data, family, type, method="mle", se=FALSE, max.df=30,
             max.BB=list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1)),
             progress=FALSE)
```

Arguments

data   An N x d data matrix (with uniform margins).

family   A $d*(d-1)/2$ integer vector of C-/D-vine pair-copula families with values
                0 = independence copula
                1 = Gaussian copula
                2 = Student t copula (t-copula)
                3 = Clayton copula
                4 = Gumbel copula
                5 = Frank copula
                6 = Joe copula
                7 = BB1 copula
                8 = BB6 copula
                9 = BB7 copula
                10 = BB8 copula
                13 = rotated Clayton copula (180 degrees; "survival Clayton")
                14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
                16 = rotated Joe copula (180 degrees; "survival Joe")
                17 = rotated BB1 copula (180 degrees; "survival BB1")
                18 = rotated BB6 copula (180 degrees; "survival BB6")
                19 = rotated BB7 copula (180 degrees; "survival BB7")
                20 = rotated BB8 copula (180 degrees; "survival BB8")
                23 = rotated Clayton copula (90 degrees)
                24 = rotated Gumbel copula (90 degrees)
                26 = rotated Joe copula (90 degrees)
                27 = rotated BB1 copula (90 degrees)
                28 = rotated BB6 copula (90 degrees)
                29 = rotated BB7 copula (90 degrees)
                30 = rotated BB8 copula (90 degrees)
                33 = rotated Clayton copula (270 degrees)
                34 = rotated Gumbel copula (270 degrees)
                36 = rotated Joe copula (270 degrees)
                37 = rotated BB1 copula (270 degrees)
                38 = rotated BB6 copula (270 degrees)
                39 = rotated BB7 copula (270 degrees)
                40 = rotated BB8 copula (270 degrees)

type	Type of the vine model: 1 or "CVine" = C-vine 2 or "DVine" = D-vine
method	Character indicating the estimation method: either pairwise maximum likelihood estimation (method = "mle"; default) or inversion of Kendall's tau (method = "itau"; see BiCopEst). For method = "itau" only one parameter pair-copula families can be used (family = 1, 3, 4, 5, 6, 13, 14, 16, 23, 24, 26, 33, 34 or 36).
se	Logical; whether standard errors are estimated (default: se=FALSE).
max.df	Numeric; upper bound for the estimation of the degrees of freedom parameter of the t-copula (default: max.df = 30; for more details see BiCopEst).
max.BB	List; upper bounds for the estimation of the two parameters (in absolute values) of the BB1, BB6, BB7 and BB8 copulas (default: max.BB = list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1))).
progress	Logical; whether the pairwise estimation progress is printed (default: progress = FALSE).

Details

The pair-copula parameter estimation is performed tree-wise, i.e., for each C-/D-vine tree the results from the previous tree(s) are used to calculate the new copula parameters using BiCopEst.

Value

par	Estimated (first) C-/D-vine pair-copula parameters.
par2	Estimated second C-/D-vine pair-copula parameters for families with two parameters (t, BB1, BB6, BB7, BB8). All other entries are zero.
se	Estimated standard errors of the (first) pair-copula parameter estimates (if se = TRUE).
se2	Estimated standard errors of the second pair-copula parameter estimates (if se = TRUE).

Author(s)

Carlos Almeida, Ulf Schepsmeier

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

Czado, C., U. Schepsmeier, and A. Min (2012). Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling, 12(3), 229-255.

See Also

BiCopEst, BiCopHfunc, CDVineLogLik, CDVineMLE

Examples

```
## Example 1: 4-dimensional D-vine model with Gaussian pair-copulas
data(worldindices)
Data = as.matrix(worldindices)[,1:4]
d = dim(Data)[2]
```

```
fam = rep(1,d*(d-1)/2)

# sequential estimation
CDVineSeqEst(Data,fam,type=2,method="itau")$par
CDVineSeqEst(Data,fam,type=2,method="mle")$par


## Example 2: 4-dimensional D-vine model with mixed pair-copulas
fam2 = c(5,1,3,14,3,2)

# sequential estimation
CDVineSeqEst(Data,fam2,type=2,method="mle",se=TRUE,progress=TRUE)
```

CDVineSim   *Simulation from C- and D-vine copula models*

Description

This function simulates from given C- and D-vine copula models.

Usage

```
CDVineSim(N, family, par, par2=rep(0,length(family)), type)
```

Arguments

N   Number of d-dimensional observations simulated.

family   A d*(d-1)/2 integer vector of C-/D-vine pair-copula families with values
               0 = independence copula
               1 = Gaussian copula
               2 = Student t copula (t-copula)
               3 = Clayton copula
               4 = Gumbel copula
               5 = Frank copula
               6 = Joe copula
               7 = BB1 copula
               8 = BB6 copula
               9 = BB7 copula
               10 = BB8 copula
               13 = rotated Clayton copula (180 degrees; "survival Clayton")
               14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
               16 = rotated Joe copula (180 degrees; "survival Joe")
               17 = rotated BB1 copula (180 degrees; "survival BB1")
               18 = rotated BB6 copula (180 degrees; "survival BB6")
               19 = rotated BB7 copula (180 degrees; "survival BB7")
               20 = rotated BB8 copula (180 degrees; "survival BB8")
               23 = rotated Clayton copula (90 degrees)
               24 = rotated Gumbel copula (90 degrees)
               26 = rotated Joe copula (90 degrees)
               27 = rotated BB1 copula (90 degrees)
               28 = rotated BB6 copula (90 degrees)
               29 = rotated BB7 copula (90 degrees)

	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)
	39 = rotated BB7 copula (270 degrees)
	40 = rotated BB8 copula (270 degrees)
par	A d*(d-1)/2 vector of pair-copula parameters.
par2	A d*(d-1)/2 vector of second parameters for pair-copula families with two parameters (t, BB1, BB6, BB7, BB8; default: par2 = rep(0,length(family))).
type	Type of the vine model: 1 or "CVine" = C-vine 2 or "DVine" = D-vine

Value

An N x d matrix of data simulated from the given C- or D-vine copula model.

Author(s)

Carlos Almeida, Ulf Schepsmeier, Eike Brechmann, Jakob Stoeber

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

See Also

CDVineLogLik, BiCopSim

Examples

```
## Example 1: simulate from a 6-dimensional C-vine model
## with Gaussian pair-copulas
d = 6
dd = d*(d-1)/2
fam1 = rep(1,dd)
par1 = c(0.2,0.69,0.73,0.22,-0.09,0.51,0.32,0.01,0.82,0.01,
         -0.2,-0.32,-0.19,-0.17,-0.06)
N = 100
U1 = CDVineSim(N,fam1,par1,type=1)
head(U1)

# calculate the log-likelihood
logLik = CDVineLogLik(U1,fam1,par1,type=1)
logLik$loglik


## Example 2: simulate from a 6-dimensional C-vine model
## with Student t pair-copulas each with three degrees of freedom
fam2 = rep(2,dd)
par2 = rep(0.5,dd)
```

```
nu2 = rep(3,dd)
U2 = CDVineSim(N,fam2,par2,nu2,type=1)

# calculate the log-likelihood
logLik2 = CDVineLogLik(U2,fam2,par2,nu2,type=1)
logLik2$loglik


## Example 3: simulate from a 6-dimensional D-vine model
## with Student t pair-copulas each with five degrees of freedom
fam3 = rep(2,dd)
par3 = rep(0.4,dd)
nu3 = rep(5,dd)
U3 = CDVineSim(N,fam3,par3,nu3,type=2)

# calculate the log-likelihood
logLik3 = CDVineLogLik(U3,fam3,par3,nu3,type=2)
logLik3$loglik
```

CDVineTreePlot   *Plot function for C- or D-vine trees*

Description

This function plots one or all trees of a given C- and D-vine copula model.

Usage

```
CDVineTreePlot(data=NULL, family,
               par=rep(0,length(family)), par2=rep(0,length(family)),
               names=NULL, type, method="mle", max.df=30,
               max.BB=list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1)),
               tree="ALL", edge.labels=c("family"), P=NULL, ...)
```

Arguments

data   An N x d data matrix (with uniform margins); default: data = NULL.

family   A d*(d-1)/2 vector of pair-copula families with values
                0 = independence copula
                1 = Gaussian copula
                2 = Student t copula (t-copula)
                3 = Clayton copula
                4 = Gumbel copula
                5 = Frank copula
                6 = Joe copula
                7 = BB1 copula
                8 = BB6 copula
                9 = BB7 copula
                10 = BB8 copula
                13 = rotated Clayton copula (180 degrees; "survival Clayton")
                14 = rotated Gumbel copula (180 degrees; "survival Gumbel")
                16 = rotated Joe copula (180 degrees; "survival Joe")

	17 = rotated BB1 copula (180 degrees; "survival BB1")
	18 = rotated BB6 copula (180 degrees; "survival BB6")
	19 = rotated BB7 copula (180 degrees; "survival BB7")
	20 = rotated BB8 copula (180 degrees; "survival BB8")
	23 = rotated Clayton copula (90 degrees)
	24 = rotated Gumbel copula (90 degrees)
	26 = rotated Joe copula (90 degrees)
	27 = rotated BB1 copula (90 degrees)
	28 = rotated BB6 copula (90 degrees)
	29 = rotated BB7 copula (90 degrees)
	30 = rotated BB8 copula (90 degrees)
	33 = rotated Clayton copula (270 degrees)
	34 = rotated Gumbel copula (270 degrees)
	36 = rotated Joe copula (270 degrees)
	37 = rotated BB1 copula (270 degrees)
	38 = rotated BB6 copula (270 degrees)
	39 = rotated BB7 copula (270 degrees)
	40 = rotated BB8 copula (270 degrees)

par
: A d*(d-1)/2 vector of pair-copula parameters (optional; default: `par = rep(0,length(family))`).

par2
: A d*(d-1)/2 vector of second parameters for pair-copula families with two parameters (optional; default: `par2 = rep(0,length(family))`).

names
: A vector of names for the d variables. If `names = NULL` (default), column names of `data` (if available) or simply numbers are used.

type
: Type of the vine model:
  1 or `"CVine"` = C-vine
  2 or `"DVine"` = D-vine

method
: Character indicating the estimation method: either maximum likelihood estimation (`method = "mle"`; default) or inversion of Kendall's tau (`method = "itau"`).

max.df
: Numeric; upper bound for the estimation of the degrees of freedom parameter of the t-copula (default: `max.df = 30`; for more details see `BiCopEst`).

max.BB
: List; upper bounds for the estimation of the two parameters (in absolute values) of the BB1, BB6, BB7 and BB8 copulas (default: `max.BB = list(BB1=c(5,6),BB6=c(6,6),BB7=c(5,6),BB8=c(6,1))`).

tree
: Number of the tree to be plotted or `tree = "ALL"` (default) to plot all trees.

edge.labels
: Vector of edge labels. Possible choices:
  FALSE: no edge labels
  `"family"`: pair-copula families (default)
  `"par"`: pair-copula parameters
  `"par2"`: second pair-copula parameters
  `"theotau"`: theoretical Kendall's tau values corresponding to pair-copula families and parameters (see `BiCopPar2Tau`)
  `"emptau"`: empirical Kendall's tau values (only if data is provided!)

P
: A list of matrices with two columns for the x-y-coordinates of the nodes in the plot(s) (optional; default: `P = NULL`).

...
: further graphical parameters

Note

The function computes the positions of the nodes automatically with the Fruchterman-Reingold algorithm (see `plot.igraph` for a detailed description). If one would like to set the positions

manually, one has to specify a list of matrices P in the argument list. A good starting point may be to run the function `CDVineTreePlot` and manipulate the returning matrix P.

The user can set the copula parameters par and par2. If not set and data is provided, the parameters of the C- or D-vine copula model are estimated sequentially using `CDVineSeqEst`/`BiCopEst`. Then the edge width is chosen according to the empirical Kendall's tau values. Otherwise theoretical values are used.

Author(s)

Ulf Schepsmeier

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.

See Also

`CDVineMLE`

Examples

```
# simulate from a 6-dimensional C-vine model with Gaussian pair-copulas
d = 6
dd = d*(d-1)/2
par1 = c(0.2,0.69,0.73,0.22,-0.09,0.51,0.32,0.01,0.82,0.01,-0.2,
         -0.32,-0.19,-0.17,-0.06);
fam1 = rep(1,dd)
N = 100
U = CDVineSim(N,fam1,par1,type=1)

# plot the first tree with pair-copula families and
# empirical Kendall's tau values as edge labels
CDVineTreePlot(U,fam1,type=1,tree=1,edge.labels=c("family","emptau"))

# plot all trees without edge labels and without sequential estimation
CDVineTreePlot(data=NULL,family=fam1,par=par1,type=1, edge.labels=FALSE)
```

CDVineVuongTest   *Vuong test comparing two vine copula models*

Description

This function performs a Vuong test between two d-dimensional C- or D-vine copula models, respectively.

Usage

```
CDVineVuongTest(data, Model1.order=1:dim(data)[2],
                Model2.order=1:dim(data)[2], Model1.family,
                Model2.family, Model1.par, Model2.par,
                Model1.par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2),
                Model2.par2=rep(0,dim(data)[2]*(dim(data)[2]-1)/2),
                Model1.type, Model2.type)
```

Arguments

data   An N x d data matrix (with uniform margins).

Model1.order, Model2.order

Two numeric vectors giving the order of the variables in the first D-vine trees or of the C-vine root nodes in models 1 and 2 (default: Model1.order and Model2.order = 1:dim(data)[2], i.e., standard order).

Model1.family, Model2.family

Two d*(d-1)/2 numeric vectors of the pair-copula families of models 1 and 2, respectively, with values

0 = independence copula

1 = Gaussian copula

2 = Student t copula (t-copula)

3 = Clayton copula

4 = Gumbel copula

5 = Frank copula

6 = Joe copula

7 = BB1 copula

8 = BB6 copula

9 = BB7 copula

10 = BB8 copula

13 = rotated Clayton copula (180 degrees; "survival Clayton")

14 = rotated Gumbel copula (180 degrees; "survival Gumbel")

16 = rotated Joe copula (180 degrees; "survival Joe")

17 = rotated BB1 copula (180 degrees; "survival BB1")

18 = rotated BB6 copula (180 degrees; "survival BB6")

19 = rotated BB7 copula (180 degrees; "survival BB7")

20 = rotated BB8 copula (180 degrees; "survival BB8")

23 = rotated Clayton copula (90 degrees)

24 = rotated Gumbel copula (90 degrees)

26 = rotated Joe copula (90 degrees)

27 = rotated BB1 copula (90 degrees)

28 = rotated BB6 copula (90 degrees)

29 = rotated BB7 copula (90 degrees)

30 = rotated BB8 copula (90 degrees)

33 = rotated Clayton copula (270 degrees)

34 = rotated Gumbel copula (270 degrees)

36 = rotated Joe copula (270 degrees)

37 = rotated BB1 copula (270 degrees)

38 = rotated BB6 copula (270 degrees)

39 = rotated BB7 copula (270 degrees)

40 = rotated BB8 copula (270 degrees)

Model1.par, Model2.par

Two d*(d-1)/2 numeric vectors of the (first) copula parameters of models 1 and 2, respectively.

Model1.par2, Model2.par2

Two d*(d-1)/2 numeric vectors of the second copula parameters of models 1 and 2, respectively; necessary for t, BB1, BB6, BB7 and BB8 copulas. If no such families are included in Model1.family/Model2.family, these arguments do not need to be specified (default: Model1.par2 and Model2.par2 = rep(0,dim(data)[2]*(dim(da

Model1.type, Model2.type

Type of the respective vine model:

1 or "CVine" = C-vine

2 or "DVine" = D-vine

Details

The likelihood-ratio based test proposed by Vuong (1989) can be used for comparing non-nested models. For this let c_1 and c_2 be two competing vine copulas in terms of their densities and with estimated parameter sets $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$. We then compute the standardized sum, ν, of the log differences of their pointwise likelihoods $m_i := \log\left[\frac{c_1(\boldsymbol{u}_i|\hat{\boldsymbol{\theta}}_1)}{c_2(\boldsymbol{u}_i|\hat{\boldsymbol{\theta}}_2)}\right]$ for observations $\boldsymbol{u}_i \in [0,1]$, $i = 1, ..., N$, i.e.,

$$\texttt{statistic} := \nu = \frac{\frac{1}{n}\sum_{i=1}^{N} m_i}{\sqrt{\sum_{i=1}^{N}\left(m_i - \bar{m}\right)^2}}.$$

Vuong (1989) shows that ν is asymptotically standard normal. According to the null-hypothesis

$$H_0 : E[m_i] = 0 \,\forall i = 1, ..., N,$$

we hence prefer vine model 1 to vine model 2 at level α if

$$\nu > \Phi^{-1}\left(1 - \frac{\alpha}{2}\right),$$

where Φ^{-1} denotes the inverse of the standard normal distribution function. If $\nu < -\Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$ we choose model 2. If, however, $|\nu| \le \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$, no decision among the models is possible.

Like AIC and BIC, the Vuong test statistic may be corrected for the number of parameters used in the models. There are two possible corrections; the Akaike and the Schwarz corrections, which correspond to the penalty terms in the AIC and the BIC, respectively.

Value

statistic, statistic.Akaike, statistic.Schwarz

Test statistics without correction, with Akaike correction and with Schwarz correction.

p.value, p.value.Akaike, p.value.Schwarz

P-values of tests without correction, with Akaike correction and with Schwarz correction.

Author(s)

Jeffrey Dissmann, Ulf Schepsmeier

References

Vuong, Q. H. (1989). Ratio tests for model selection and non-nested hypotheses. Econometrica 57 (2), 307-333.

See Also

CDVineClarkeTest, CDVineAIC, CDVineBIC

Examples

```
# compare 6-dimensional C-vine copula models
# with Gaussian and with Student t pair-copulas
d = 6
dd = d*(d-1)/2
fam1 = rep(1,dd)
par1 = c(0.2,0.69,0.73,0.22,-0.09,0.51,0.32,0.01,0.82,0.01,
         -0.2,-0.32,-0.19,-0.17,-0.06)

fam2 = rep(2,dd)
par2 = par1
nu2 = rep(4,dd)

# simulate a sample of size 300 from the first C-vine copula model
simdata = CDVineSim(300,fam1,par1,type=1)

# compare the two models based on this sample
vuong = CDVineVuongTest(simdata,1:d,1:d,fam1,fam2,par1,par2,
                        Model2.par2=nu2,Model1.type=1,Model2.type=1)
vuong$statistic
vuong$statistic.Schwarz
vuong$p.value
vuong$p.value.Schwarz
```

---

worldindices   *Major World Indices*

Description

This data set contains transformed standardized residuals of daily log returns of major world stock indices in 2009 and 2010. The considered indices are the leading stock exchanges of the six largest economies in the world: the US American S&P 500, the Japanese Nikkei 225, the Chinese SSE Composite Index, the German DAX, the French CAC 40 and the British FTSE 100 Index. Each time series is filtered using an ARMA(1,1)-GARCH(1,1) model with Student t innovations.

Format

A data frame with 396 observations on 6 variables. Column names correspond to ticker symbols of the indices.

Source

Yahoo! Finance

Examples

```
# load the data set
data(worldindices)

# compute the empirical Kendall's tau matrix
cor(worldindices,method="kendall")
```

Index