# Package 'Depela'

February 9, 2012

**Version** 0.0

**Date** 2009-01-19

**Title** Depela

**Author** Andrew C. Chou <andrewchengchou@gmail.com>, Jing Tao <amandajingtao@gmail.com>.

**Maintainer** Andrew C. Chou <andrewchengchou@gmail.com>

**Depends** stats, stats4, copula, R (>= 2.8.0)

**Description** Implement semiparametric estimation of copula model, and deal with structural break problems in copula modelling.

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2009-01-20 07:41:57

## R topics documented:

copulavar *Copula-VAR Estimation*

## Description

Estimate the VAR system when the innorvation terms are generated from some copula models. This function can also be used to estimate the copula model when the observations are dependent acroos the time.

## Usage

```
copulavar(z, oder = 0, dens, init)
```

## Arguments

| | |
|---|---|
| z | observations, a N*K matrix |
| oder | the order of the VAR system. If order=0, then the program will select the order automatically based on the PACF criterion. |
| dens | user-defined function returning the copula density given the value of dependence parameter. The definition of dens should follow the rule that the first argument is the value of the dependence parameter, and the second argument is the observation vector. |
| init | initial guess of the value of the dependence parameter. |

## Value

| | |
|---|---|
| m.est | A list containing the marginal parameter |
| mledp | The ML estimate of the dependence parameter |

## Examples

```
##
# generate random samples of a copula-var system
x<-gendepela(500,2,list(c(0.2,0.1),c(0.2,0.11)),"clayton",8,rep(1,2))
# define dens function
dens<-function(theta,dat){
u<-dat[1]
v<-dat[2]
return((1+theta)*u^(-1-theta)*v^(-1-theta)*(-1+u^(-theta)+v^(-theta))^(-2-1/theta))
}
out<-copulavar(x,2,dens,3)
```

gendepela *Generate Copula VAR Time Series*

## Description

gendepela generate multivariate time series allowing significant richer dependence structure among innovation terms.

## Usage

```
gendepela(N, d, phi, copfamily, theta, bstd)
```

## Arguments

| | |
|---|---|
| N | number of observations |
| d | the order of the VAR system |
| phi | a list containing the value of marginal parameter. The length of each element of this list should be equal |
| copfamily | the family name of the copula. "clayton", "frank", and "gumbel" are supported |
| theta | the value of dependence parameter |
| bstd | a vector containing the values of sd of for each innovation term |

## Details

The marginal distribution of each disturbance term is normal distribution with mean 0, and sd assgined by bstd. However, their dependence is determined by the assgined copula parameterized by theta.

## Value

A data set generated from user-defined depela model.

## Examples

```
##
# Generate a 50*3 multivariate time series whose innovation terms are contructed from a Gumbel copula.
 dat<-gendepela(50,2,list(c(0.2,0.1),c(0.2,0.11),c(0.2,0.1)),"gumbel",4,rep(1,3))
```

spestimator *Two-Stage Semiparametric Estimator of Copula Model*

## Description

spestimator estimates the dependence parameter of a copula model given observations by implementing the semi-parametric estimator.

## Usage

```
spestimator(u, dens, init)
```

## Arguments

| | |
|---|---|
| u | observations, a n*k matrix |
| dens | user-defined function returning the copula density given the value of dependence parameter. The definition of dens should follow the rule that the first argument is the value of the dependence parameter, and the second argument is the observation vector. |
| init | initial guess of the value of the dependence parameter. a k*1 vector. |

## Details

spestimator first estimate the marginal CDF for each random variable using nonparametric method, then estimate the dependence parameter by implementing MLE.

## Value

A ML estimate of the dependence parameter which is an object of class mle-class.

## Examples

```
##
# Example: Two-stage estimation of a two-dimensional Clayton copula model.
## Define the density function for Clayton copula.
library(copula)
dens<-function(theta,dat){
u<-dat[1]
v<-dat[2]
return((1+theta)*u^(-1-theta)*v^(-1-theta)*(-1+u^(-theta)+v^(-theta))^(-2-1/theta))
}
x <- mvdc(claytonCopula(8), c("norm", "exp"),list(list(mean = 0, sd =2), list(rate = 2)))
x.samp <- rmvdc(x, 500)
tsmle<-spestimator(x.samp,dens,init=2)
```

# Index