Package `tscopula`

July 7, 2021

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Time Series Copula Models</td>
</tr>
<tr>
<td>Version</td>
<td>0.2.1</td>
</tr>
<tr>
<td>Date</td>
<td>2021-07-03</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Alexander McNeil <a href="mailto:alexanderjmcneil@gmail.com">alexanderjmcneil@gmail.com</a></td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.5.0)</td>
</tr>
<tr>
<td>License</td>
<td>GPL-3</td>
</tr>
<tr>
<td>LazyData</td>
<td>true</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>7.1.1</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>Imports</td>
<td>methods, stats, graphics, utils, zoo, xts, FKF, ltsa, rvinecopulib, FitAR, arfima, Matrix</td>
</tr>
<tr>
<td>Collate</td>
<td>'basic_objects.R' 'armacopula.R' 'dvinecopula.R' 'dvinecopula2.R' 'fitting_basic.R' 'margins.R' 'full_models.R' 'vtransforms.R' 'fitting_vtscopula.R' 'helper_vtarma.R' 'data.R'</td>
</tr>
<tr>
<td>Suggests</td>
<td>knitr, rmarkdown</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>no</td>
</tr>
<tr>
<td>Author</td>
<td>Alexander McNeil [aut, cre], Martin Bladt [aut]</td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2021-07-07 08:20:02 UTC</td>
</tr>
</tbody>
</table>
R topics documented:

- `acf2pacf` .......................................................... 3
- `armacopula` ......................................................... 4
- `armacopula-class` ............................................... 4
- `bitcoin` .............................................................. 5
- `coerce,tscopula,tscm-method` ................................ 6
- `coerce,tscopulafit,tscmfit-method` ......................... 6
- `cpi` ................................................................. 7
- `dmarg` ............................................................... 7
- `doubleweibull` ..................................................... 8
- `dvinecopula` ......................................................... 9
- `dvinecopula-class` ............................................... 9
- `dvinecopula2` ....................................................... 10
- `dvinecopula2-class` ............................................. 11
- `fit` ................................................................. 12
- `fit,margin-method` ............................................... 13
- `fit,tscm-method` .................................................. 13
- `fit,tscopulafit-method` ....................................... 14
- `fit,tscopulaU-method` ......................................... 15
- `fit,vtscopula-method` .......................................... 15
- `glag` ............................................................... 16
- `kendall` ............................................................ 17
- `kfilter` ............................................................. 17
- `kpacf_arfima` ...................................................... 18
- `kpacf_arma` ........................................................ 18
- `kpacf_fbn` .......................................................... 19
- `laplace` ............................................................. 19
- `margin` .............................................................. 20
- `margin-class` ....................................................... 20
- `marginfit-class` .................................................. 21
- `non_invert` .......................................................... 22
- `non_stat` ............................................................ 22
- `pacf2acf` ............................................................ 23
- `pcoincide` ........................................................... 23
- `plot,marginfit,missing-method` ............................... 24
- `plot,tscmfit,missing-method` .................................. 24
- `plot,tscopulafit,missing-method` ......................... 24
- `plot,Vtransform,missing-method` ......................... 24
- `pmarg` .............................................................. 25
- `profilefulcrum` ................................................... 25
- `qmarg` .............................................................. 26
- `quantile,tscmfit-method` ...................................... 26
- `safe_ses` ............................................................ 26
- `sdoubleweibull` .................................................... 29
- `sigmastarma` ....................................................... 30
- `sim` ................................................................. 30
- `slaplace` ............................................................ 31
Compute partial autocorrelations from autocorrelations

acf2pacf

acf2pacf(rho)

rho vector of autocorrelation values (excluding 1).

Value

A vector of partial autocorrelation values with same length as rho.
Examples

\begin{verbatim}
rho <- ARMAacf(ar = -0.9, ma = 0.8, lag.max = 50)[-1]
alpha <- acf2pacf(rho)
\end{verbatim}

\textbf{armacopula}  \hspace{1cm}  \textit{Constructor function for ARMA copula process}

Description

Constructor function for ARMA copula process

Usage

\texttt{armacopula(pars = list\{ar = 0, ma = 0\})}

Arguments

\texttt{pars}

\begin{itemize}
  \item \texttt{pars} list consisting of vector of AR parameters named \texttt{ar} and vector of MA parameters named \texttt{ma}.
\end{itemize}

Value

An object of class \texttt{armacopula}.

Examples

\begin{verbatim}
armacopula(list(ar = 0.5, ma = 0.4))
\end{verbatim}

\textbf{armacopula-class}  \hspace{1cm}  \textit{ARMA copula processes}

Description

Class of objects for ARMA copula processes.

Usage

\begin{verbatim}
## S4 method for signature 'armacopula'
coef(object)

## S4 method for signature 'armacopula'
show(object)

## S4 method for signature 'armacopula'
sim(object, n = 1000)

## S4 method for signature 'armacopula'
kendall(object, lagmax = 20)
\end{verbatim}
Arguments

- **object**: an object of the class.
- **n**: length of realization.
- **lagmax**: maximum value of lag.

Methods (by generic)

- **coef**: Coef method for ARMA copula class
- **show**: Show method for ARMA copula process
- **sim**: Simulation method for armacopula class
- **kendall**: Calculate Kendall's tau values for armacopula model

Slots

- **name**: name of ARMA copula process.
- **modelspec**: vector containing number of AR and MA parameters.
- **pars**: list consisting of vector of AR parameters named 'ar' and vector of MA parameters named 'ma'.

Examples

```r
sim(armacopula(list(ar = c(0.5, 0.4), ma = -0.8)), n = 1000)
mod <- armacopula(list(ar = 0.95, ma = -0.85))
kendall(mod)
```

---

**bitcoin**  
*Bitcoin price data 2016-19*

Description

Time series of Bitcoin closing prices from 31 December 2015 to 31 December 2019 (1044 values). This permits the calculation of 4 calendar years of returns.

Usage

```r
data(bitcoin)
```

Format

An object of class "xts".

Examples

```r
data(bitcoin)
plot(bitcoin)
X <- (diff(log(bitcoin))[-1]) * 100
plot(X)
```
coerce,tscopula,tscm-method

Convert tscopula object to tscm object

Description

Convert tscopula object to tscm object

Usage

## S4 method for signature 'tscopula,tscm'
coerce(from, to = "tsc", strict = TRUE)

Arguments

from a tscopula object.
to a tscm object.
strict logical variable stating whether strict coercion should be enforced.

Value

A tscm object.

coerce,tscopulafit,tscmfit-method

Convert tscopulafit object to be tscmfit object

Description

Convert tscopulafit object to be tscmfit object

Usage

## S4 method for signature 'tscopulafit,tscmfit'
coerce(from, to = "tscmfit", strict = TRUE)

Arguments

from a tscopulafit object.
to a tscmfit object.
strict logical variable stating whether strict coercion should be enforced.

Value

A tscmfit object.
Description

Time series of US quarterly CPI (consumer price index) data Q4 1959 to Q4 2020 (245 values) for studying inflation. These data were sourced from the OECD webpage and represent the total ‘perspective’ on inflation, including food and energy. They have been based to have a value of 100 in 2015.

Usage

data(cpi)

Format

An object of class "xts".

Examples

data(cpi)
plot(cpi)
X <- (diff(log(cpi))[-1]) * 100
plot(X)

---

dmarg

Description

Compute the density function of the marginal model.

Usage

dmarg(x, y, log = FALSE)

Arguments

x an object of class margin.
y vector of values for which density should be computed.
log logical variable specifying whether log density should be returned.

Value

A vector of values for the density.
doubleweibull

Double Weibull distribution

Examples

```r
margmod <- margin("norm", pars = c(mean = 0, sd = 1))
dmarg(margmod, c(-2, 0, 2), log = TRUE)
```

Description

Double Weibull distribution

Usage

```r
ddoubleweibull(x, mu = 0.05, shape = 1, scale = 1, log = FALSE)
pdoubleweibull(q, mu = 0.05, shape = 1, scale = 1)
qdoubleweibull(p, mu = 0.05, shape = 1, scale = 1)
rdoubleweibull(n, mu = 0.05, shape = 1, scale = 1)
```

Arguments

- `x`: vector of values.
- `mu`: location parameter.
- `shape`: shape parameter.
- `scale`: scale parameter.
- `log`: flag for log density.
- `q`: vector of quantiles.
- `p`: vector of probabilities.
- `n`: number of observations.

Value

A vector of density, distribution function, quantile or random values.
dvinecopula

Constructor function for dvinecopula process

Description

Constructor function for dvinecopula process

Usage

dvinecopula(family = "indep", pars = list(NULL), rotation = 0)

Arguments

family a vector of family names
pars a list containing the parameters of each lag
rotation a vector of rotations

Value

An object of class dvinecopula.

Examples

dvinecopula(family = c("joe", "gauss", "t"), pars = list(3, .5, c(1, 2)), rotation = c(180, 0, 0))

dvinecopula-class

D-vine copula processes

Description

Class of objects for d-vine copula processes.

Usage

## S4 method for signature 'dvinecopula'
coef(object)

## S4 method for signature 'dvinecopula'
show(object)

## S4 method for signature 'dvinecopula'
sim(object, n = 1000, innov = NA, start = NA)

## S4 method for signature 'dvinecopula'
kendall(object, lagmax = 20)
**dvinecopula2**

**Arguments**
- object: an object of the class.
- n: length of realization.
- innov: vector of innovations of length n.
- start: vector of start values with length equal to order of process.
- lagmax: maximum value of lag.

**Methods (by generic)**
- coef: Coef method for dvinecopula class
- show: Show method for dvinecopula class
- sim: Simulation method for dvinecopula class
- kendall: Calculate Kendall’s tau values for pair copulas in d-vine copula

**Slots**
- name: name of the d-vine copula process.
- modelspec: list containing the family, number of parameters and rotations
- pars: list comprising of the parameters.

**Examples**
```r
sim(dvinecopula("gauss", 0.5))
mixmod <- dvinecopula(family = c("gumbel", "gauss"), pars = list(1.5, -0.6))
kendall(mixmod)
```

---

**Description**

Constructor function for dvinecopula2 process

**Usage**
```r
dvinecopula2(
  family = "gauss",
  rotation = 0,
  kpacf = "kpacf_arma",
  pars = list(ar = 0.1, ma = 0.1),
  maxlag = Inf,
  negtau = "none"
)
```
Arguments

- family: family name
- rotation: rotation
- kpacf: character string giving name of Kendal pacf
- pars: a list containing the parameters of each lag
- maxlag: scalar specifying maximum lag
- negtau: character specifying treatment of negative tau values

Value

An object of class `dvinecopula2`.

Examples

```r
dvinecopula2(family = "joe", kpacf = "kpacf_arma", pars = list(ar = 0.95, ma = -0.85), maxlag = 30)
```

---

**dvinecopula2-class**  
*D-vine copula processes of type 2*

Description

Class of objects for d-vine copula processes.

Usage

```r
## S4 method for signature 'dvinecopula2'
coef(object)

## S4 method for signature 'dvinecopula2'
show(object)

## S4 method for signature 'dvinecopula2'
sim(object, n = 1000)

## S4 method for signature 'dvinecopula2'
kendall(object, lagmax = 20)
```

Arguments

- object: an object of the class.
- n: length of realization.
- lagmax: maximum value of lag.
Methods (by generic)

- coef: Coef Method for dvinecopula2 class
- show: Show method for dvinecopula2 class
- sim: Simulation method for dvinecopula2 class
- kendall: Calculate Kendall’s tau values for pair copulas in type 2 d-vine copula

Slots

- name: name of the d-vine copula process.
- modelspec: list containing the family, rotation, and name of KPACF
- pars: list comprising of the parameters.

Examples

copmod <- dvinecopula2(family = "joe", kpacf = "kpacf_arma",
                      pars = list(ar = 0.95, ma = -0.85), maxlag = 30)
kendall(copmod)

Description

Methods are available for objects of class tscopulaU, vtscopula, tscopulafit, margin and tscm.

Usage

fit(x, y, ...)

Arguments

x an object of the model class.

y a vector or time series of data.

... further arguments to be passed on.

Value

An object of the fitted model class.
Description
Fit method for margin class

Usage
## S4 method for signature 'margin'
fit(x, y, tsoptions = list(), control = list(maxit = 1000))

Arguments
- x: an object of class margin.
- y: a vector or time series of data.
- tsoptions: list of optional arguments: hessian is logical variable specifying whether Hessian matrix should be returned; start is vector of named starting values
- control: list of control parameters to be passed to the optim function.

Value
An object of class marginfit.

Examples
margmod <- margin("norm", pars = c(mean = 0, sd = 1))
data <- sim(margmod, n = 500)
fit(margmod, data)

Description
Fit method for tscm class

Usage
## S4 method for signature 'tscm'
fit(x, y, tsoptions = list(), control = list(warn.1d.NelderMead = FALSE, trace = FALSE, maxit = 5000), method = "IFM")
Arguments

- **x**: an object of class tscm.
- **y**: a vector or time series of data.
- **tsoptions**: a list of parameters passed to fitting.
- **control**: list of control parameters to be passed to the optim function.
- **method**: character string specifying method.

Value

An object of class tscmfit.

Examples

```r
mod <- tscm(dvinecraft(family = "gauss", pars = 0.5), margin="doubleweibull")
y <- sim(mod)
fit(mod, y)
```

Additional Examples

```r
ar1 <- armacopula(list(ar = 0.7))
data <- sim(ar1, 1000)
ar1fit <- fit(fit(ar1, data), sim(ar1, 1000))
```
fit,tscopulaU-method  Fit method for tscopulaU class

Description

Fit method for tscopulaU class

Usage

```r
## S4 method for signature 'tscopulaU'
fit(x, y, tsoptions = list(), control = list(warn.1d.NelderMead = FALSE))
```

Arguments

- `x`: an object of class `tscopulaU`
- `y`: vector or time series of data to which the copula process is to be fitted.
- `tsoptions`: list of options
- `control`: list of control parameters to be passed to the `optim` function.

Value

An object of class `tscopulafit`.

Examples

```r
data <- sim(armacopula(list(ar = 0.5, ma = 0.4)), n = 1000)
fit(armacopula(list(ar = 0.5, ma = 0.4)), data)
```

fit,vtscopula-method  Fit method for vtscopula class

Description

Fit object of class `vtscopula` to data using maximum likelihood.

Usage

```r
## S4 method for signature 'vtscopula'
fit(
  x,  
y,  
tsoptions = list(),
  control = list(maxit = 2000, warn.1d.NelderMead = FALSE)
)
```
Arguments

- **x**: an object of class `vtscopula`.
- **y**: a vector or time series of data.
- **tsoptions**: list of optional arguments: `hessian` is logical variable specifying whether Hessian matrix should be returned; `method` is choice of optimization method.
- **control**: list of control parameters to be passed to the `optim` function.

Value

An object of class `tscopulafit`.

Examples

```r
copobject <- armacopula(pars = list(ar = 0.6, ma = 0.2))
vtcop <- vtscopula(copobject, Vtransform = V2p())
y <- sim(vtcop)
fit(vtcop, y)
```

---

**glag**

*Generalized lagging function*

Description

Generalized lagging function

Usage

```r
glag(x, lagmax = 20, glagplot = FALSE)
```

Arguments

- **x**: an object of class `tscopulafit`.
- **lagmax**: maximum value for lag.
- **glagplot**: logical value indicating generalized lag plot.

Value

If `glagplot` is `TRUE` a list of generalized lagged datasets of maximum length 9 is returned to facilitate a generalized lag plot. If `glagplot` is `FALSE` a vector of length `lagmax` containing the Kendall rank correlations for the generalized lagged datasets is returned.
kendall

Generic for Kendall correlations

Description

Methods are available for objects of class armacopula, dvinecopula, dvinecopula2 and vscopula.

Usage

kendall(object, ...)

Arguments

object an object of the model class.
...

Value

A vector of Kendall correlations.

kfilter

Kalman filter for ARMA copula model

Description

Kalman filter for ARMA copula model

Usage

kfilter(x, y)

Arguments

x an object of class armacopula.
y a vector of data.

Value

A matrix or multivariate time series with columns consisting of conditional mean, standard deviation and residuals.

Examples

data <- sim(armacopula(list(ar = c(0.5, 0.4), ma = -0.8)), n = 1000)
kfilter(armacopula(list(ar = c(0.5, 0.4), ma = -0.8)), data)
kpacf_arfima  
**KPACF of ARFIMA process**

**Description**
KPACF of ARFIMA process

**Usage**
kpacf_arfima(k, theta)

**Arguments**
k
number of lags.
theta
list with components ar, ma and d specifying the ARFIMA parameters

**Value**
A vector of Kendall partial autocorrelations of length k.

kpacf_arma  
**KPACF of ARMA process**

**Description**
KPACF of ARMA process

**Usage**
kpacf_arma(k, theta)

**Arguments**
k
number of lags.
theta
list with components ar and ma specifying the ARMA parameters.

**Value**
A vector of Kendall partial autocorrelations of length k.
kpacf_fbn

kpacf_fbn

\textit{kpacf\_fbn} \quad \textit{KPACF of fractional Brownian noise}

\textbf{Description}

KPACF of fractional Brownian noise

\textbf{Usage}

\texttt{kpacf\_fbn(k, theta)}

\textbf{Arguments}

- \texttt{k} \quad \text{number of lags}
- \texttt{theta} \quad \text{parameter of process}

\textbf{Value}

A vector of Kendall partial autocorrelations of length \texttt{k}.

\begin{itemize}
  \item [laplace]
\end{itemize}

\textit{laplace} \quad \textit{Laplace distribution}

\textbf{Description}

Laplace distribution

\textbf{Usage}

- \texttt{dlaplace(x, mu = 0.05, scale = 1, log = FALSE)}
- \texttt{plaplace(q, mu = 0.05, scale = 1)}
- \texttt{qlaplace(p, mu = 0.05, scale = 1)}
- \texttt{rlaplace(n, mu = 0.05, scale = 1)}

\textbf{Arguments}

- \texttt{x} \quad \text{vector of values.}
- \texttt{mu} \quad \text{location parameter.}
- \texttt{scale} \quad \text{scale parameter.}
- \texttt{log} \quad \text{flag for log density.}
- \texttt{q} \quad \text{vector of quantiles.}
- \texttt{p} \quad \text{vector of probabilities.}
- \texttt{n} \quad \text{number of observations.}
Value

A vector of density, distribution function, quantile or random values.

```
margin
```

Constructor function for margin

Description

Constructor function for margin

Usage

```
margin(name, pars = NULL)
```

Arguments

- `name`: character string giving name of distribution
- `pars`: parameters of the distribution

Value

An object of class `margin`.

Examples

```
margin("sst")
```

margin-class

Marginal model for time series

Description

Class of objects for marginal models for stationary time series. The object is given a name and there must exist functions `pname`, `qname`, `dname` and `rname`. For example, the object could be named `norm` and make use of `pnorm`, `qnorm`, `dnorm` and `rnorm`. As well as the parameters of the distribution, `dname` must have the logical argument `log` specifying whether log density should be computed.

Usage

```
## S4 method for signature 'margin'
coef(object)

## S4 method for signature 'margin'
sim(object, n = 1000)

## S4 method for signature 'margin'
show(object)
```
Arguments

object an object of the class.
n length of realization.

Methods (by generic)

• coef: Coef method for margin class
• sim: Simulation method for margin class
• show: Show method for margin class

Slots

name name of the marginal model class.
pars a numeric vector containing the named parameters of the distribution which are passed as arguments to pname, qname, dname and rname.

Examples

new("margin", name = "norm", pars = c(mu = 0, sigma = 1))
margmod <- margin("norm", pars = c(mean = 0, sd = 1))
sim(margmod, n = 500)

Description

Fitted marginal model for time series

Usage

## S4 method for signature 'marginfit'
logLik(object)

Arguments

object an object of the class.

Methods (by generic)

• logLik: logLik method for marginfit class

Slots

margin an object of class margin.
data numeric vector or time series of data.
fit a list containing details of the maximum likelihood fit.
non_invert  
*Check for invertibility of ARMA process*

**Description**
Check for invertibility of ARMA process

**Usage**

non_invert(ma)

**Arguments**

*ma*  
vector of moving average parameters.

**Value**
A logical variable stating whether ARMA process is invertible.

---

non_stat  
*Check for causality of ARMA process*

**Description**
Check for causality of ARMA process

**Usage**

non_stat(ar)

**Arguments**

*ar*  
vector of autoregressive parameters

**Value**
A logical variable stating whether ARMA process is causal.
pacf2acf

Compute autocorrelations from partial autocorrelations

**Description**

Compute autocorrelations from partial autocorrelations

**Usage**

```r
pacf2acf(alpha)
```

**Arguments**

- `alpha` vector of partial autocorrelation values.

**Value**

A vector of autocorrelation values with same length as `alpha`.

**Examples**

```r
alpha <- ARMAacf(ar = -0.9, ma = 0.8, lag.max = 50, pacf = TRUE)
rho <- pacf2acf(alpha)
```

pcoincide

Compute coincidence probability for v-transform

**Description**

Computes the probability that if we v-transform a uniform random variable and then stochastically invert the v-transform, we get back to the original value.

**Usage**

```r
pcoincide(x)
```

**Arguments**

- `x` an object of class `Vtransform`.

**Value**

The probability of coincidence.

**Examples**

```r
pcoincide(Vlinear(delta = 0.4))
pcoincide(V3p(delta = 0.45, kappa = 0.5, xi = 1.3))
```
Description

Plot method for marginfit class

Usage

## S4 method for signature 'marginfit,missing'
plot(x, bw = FALSE)

Arguments

x an object of class marginfit.
bw logical variable specifying whether black-white options should be chosen.

Value

No return value, generates plot.

Description

Plot method for tscmfit class

Usage

## S4 method for signature 'tscmfit,missing'
plot(x, plottype = "residual", bw = FALSE, lagmax = 30)

Arguments

x an object of class tscmfit.
plottype type of plot required.
bw logical variable specifying whether black-white options should be chosen.
lagmax maximum lag value for dvinecopula2 plots

Value

No return value, generates plot.
### plot.tscopulafit,missing-method

Plot method for tscopulafit class

**Description**

Plot method for tscopulafit class

**Usage**

```r
## S4 method for signature 'tscopulafit,missing'
plot(x, plottype = "residual", bw = FALSE, lagmax = 30)
```

**Arguments**

- `x` an object of class `tscopulafit`.
- `plottype` type of plot required.
- `bw` logical variable specifying whether black-white options should be chosen.
- `lagmax` maximum lag value for Kendall plots

**Value**

No return value, generates plot.

**Examples**

```r
data <- sim(armacopula(list(ar = 0.5, ma = 0.4)), n = 1000)
fit <- fit(armacopula(list(ar = 0.5, ma = 0.4)), data)
plot(fit)
```

---

### plot,Vtransform,missing-method

Plot method for Vtransform class

**Description**

Plots the v-transform as well as its gradient or inverse. Can also plot the conditional probability that a series PIT falls below the fulcrum for a given volatility PIT value $v$.
## pmarg

### Usage

```r
## S4 method for signature 'Vtransform,missing'
plot(
  x,
  type = "transform",
  shading = TRUE,
  npoints = 200,
  lower = 0,
  upper = 1
)
```

### Arguments

- **x**: an object of class `Vtransform`
- **type**: type of plot: 'transform' for plot of transform, 'inverse' for plot of inverse, 'gradient' for plot of gradient or 'pdown' for plot of conditional probability.
- **shading**: logical variable specifying whether inadmissible zone for v-transform should be shaded
- **npoints**: number of plotting points along x-axis.
- **lower**: the lower x-axis value for plotting.
- **upper**: the upper x-axis value for plotting

### Value

No return value, generates plot.

### Examples

```r
plot(Vsymmetric())
plot(V2p(delta = 0.45, kappa = 0.8), type = "inverse")
plot(V2p(delta = 0.45, kappa = 0.8), type = "gradient")
```

### Description

Compute CDF of marginal model

### Usage

```r
pmarg(x, q)
```

### Arguments

- **x**: an object of class `margin`
- **q**: vector of values at which CDF should be computed.
Profile likelihood for fulcrum parameter

Description
Profile likelihood for fulcrum parameter

Usage
profilefulcrum(
  data,
  tscopula = dvinecopula(family = 1, pars = list(0.1)),
  locations = seq(0, 1, by = 0.1),
  plot = TRUE
)

Arguments
  data: a vector or time series of data on (0,1).
  tscopula: an object of class tscopulaU or vtscopula.
  locations: vector containing locations of different values for fulcrum.
  plot: logical values specifying whether plot should be created.

Value
A matrix containing fulcrum values and log likelihood values.

Examples
  copobject <- armacopula(pars = list(ar = 0.6, ma = 0.2))
  vtcop <- vtscopula(copobject, Vtransform = V2p())
  y <- sim(vtcop)
  profilefulcrum(y, vtcop)
qmarg

Compute quantiles of marginal model

Description

Compute the quantile function of the marginal model.

Usage

qmarg(x, p)

Arguments

x
an object of class margin.
p
vector of probabilities for which quantiles should be computed.

Value

A vector of values for the quantile function.

Examples

margmod <- margin("norm", pars = c(mean = 0, sd = 1))
qmarg(margmod, c(0.05, 0.5, 0.95))
safe_ses

*Calculate standard errors safely*

**Description**

Calculate standard errors safely

**Usage**

```
safe_ses(hess)
```

**Arguments**

- **hess**
  - A Hessian matrix from a model fit.

**Value**

A vector of standard errors.

---

sdoubleweibull

*Skew double Weibull distribution*

**Description**

Skew double Weibull distribution

**Usage**

```
dsdoubleweibull(x, mu = 0.05, shape = 1, scale = 1, gamma = 1, log = FALSE)
pssdoubleweibull(q, mu = 0.05, shape = 1, scale = 1, gamma = 1)
qsddoubleweibull(p, mu = 0.05, shape = 1, scale = 1, gamma = 1)
rssdoubleweibull(n, mu = 0.05, shape = 1, scale = 1, gamma = 1)
```

**Arguments**

- **x**
  - A vector of values.
- **mu**
  - Location parameter.
- **shape**
  - Shape parameter.
- **scale**
  - Scale parameter.
- **gamma**
  - Skewness parameter.
- **log**
  - Flag for log density.
- **q**
  - A vector of quantiles.
- **p**
  - A vector of probabilities.
- **n**
  - Number of observations.
**Value**

A vector of density, distribution function, quantile or random values.

---

**sigmastarma**

*Standard deviation of innovations for armacopula*

---

**Description**

Uses the function `tacvfARMA` in the `ltsa` library.

**Usage**

```r
sigmastarma(x)
```

**Arguments**

- `x` an object of class `armacopula`.

**Value**

The standard deviation of the standardized ARMA innovation distribution.

**Examples**

```r
sicpatarma(armacopula(list(ar = c(0.5, 0.4), ma = -0.8)))
```

---

**sim**

*Generic for simulating time series copula models*

---

**Description**

Methods are available for objects of class `swncopula`, `armacopula`, `dvinecopula`, `dvinecopula2`, `margin` and `tscm`.

**Usage**

```r
sim(object, ...)
```

**Arguments**

- `object` an object of the model class.
- `...` further arguments to be passed to the simulation.

**Value**

A simulated realization from the time series model.
slaplace  

Description

Skew Laplace distribution

Usage

dslaplace(x, mu = 0.05, scale = 1, gamma = 1, log = FALSE)

pslaplace(q, mu = 0.05, scale = 1, gamma = 1)

qslaplace(p, mu = 0.05, scale = 1, gamma = 1)

rslaplace(n, mu = 0.05, scale = 1, gamma = 1)

Arguments

x vector of values.
mu location parameter.
scale scale parameter.
gamma skewness parameter.
log flag for log density.
q vector of quantiles.
p vector of probabilities.
n number of observations.

Value

A vector of density, distribution function, quantile or random values.

sst  

Description

Skew Student t distribution
Usage

\texttt{psst(q, df = 10, gamma = 1, mu = 0, sigma = 1)}
\texttt{qst(p, df, gamma, mu, sigma)}
\texttt{dsst(x, df, gamma, mu, sigma, log = FALSE)}
\texttt{rsst(n, df, gamma, mu, sigma)}

Arguments

- \texttt{q} vector of quantiles.
- \texttt{df} degrees of freedom.
- \texttt{gamma} skewness parameter.
- \texttt{mu} location parameter.
- \texttt{sigma} scale parameter.
- \texttt{p} vector of probabilities.
- \texttt{x} vector of values.
- \texttt{log} flag for log density.
- \texttt{n} number of observations.

Value

A vector of density, distribution function, quantile or random values.

\begin{tabular}{l}
\texttt{st} \hspace{1cm} \textit{Student t distribution} \\
\end{tabular}

Description

Student t distribution

Usage

\texttt{pst(q, df = 10, mu = 0, sigma = 1)}
\texttt{qst(p, df, mu, sigma)}
\texttt{dst(x, df, mu, sigma, log = FALSE)}
\texttt{rst(n, df, mu, sigma)}
**stochinverse**

**Arguments**

- `q` vector of quantiles.
- `df` degrees of freedom.
- `mu` location parameter.
- `sigma` scale parameter.
- `p` vector of probabilities.
- `x` vector of values.
- `log` flag for log density.
- `n` number of observations.

**Value**

A vector of density, distribution function, quantile or random values.

---

**stochinverse**  
*Stochastic inverse of a v-transform*

**Description**

Stochastic inverse of a v-transform

**Usage**

```r
stochinverse(x, v, tscopula = NULL, tol = .Machine$double.eps^0.75)
```

**Arguments**

- `x` an object of class `Vtransform`.
- `v` a vector, matrix or time series with values in [0, 1].
- `tscopula` a time series copula object.
- `tol` the desired accuracy (convergence tolerance) that is passed to `uniroot` if numerical inversion is used.

**Value**

A vector, matrix or time series with values in [0, 1].

**Examples**

```r
stochinverse(Vsymmetric(), c(0, 0.25, 0.5, 0.75, 1))
```
strank  \hspace{1cm} \textit{Calculate standardized ranks of data}

**Description**

Calculate standardized ranks of data

**Usage**

\texttt{strank(x)}

**Arguments**

\texttt{x} \hspace{1cm} a vector or time series of data.

**Value**

A vector or time series of standardized ranks in the interval \((0,1)\)

**Examples**

\texttt{strank(rnorm(100))}

\texttt{swncopula}  \hspace{1cm} \textit{Constructor function for strict white noise copula process}

**Description**

Constructor function for strict white noise copula process

**Usage**

\texttt{swncopula()}

**Value**

Object of class \texttt{swncopula}.

**Examples**

\texttt{swncopula()}


swncopula-class

Strict white noise copula process

Description

Strict white noise copula process

Usage

## S4 method for signature 'swncopula'

sim(object, n = 1000)

## S4 method for signature 'swncopula'

coeff(object)

## S4 method for signature 'swncopula'

show(object)

Arguments

object an object of class swncopula.
n numeric value for length of simulated realisation.

Methods (by generic)

- sim: Simulation method for strict white noise copula
- coeff: Coef method for strict white noise copula
- show: Show method for strict white noise copula

Examples

sim(swncopula())

tscm

Constructor function for time series

Description

Constructor function for time series

Usage

tscm(tscopula, margin = new("margin", name = "unif"))
Arguments

- `tscopula` an object of class `tscopula`.
- `margin` an object of class `margin`.

Value

An object of class `tscm`.

Examples

```r
tscm(dvinecopula(family = "gauss", pars = 0.5), margin("doubleweibull"))
```

---

tscm-class

Full models

Description

Class of objects for composite time series models consisting of stationary copula processes and marginal distributions.

Usage

```r
## S4 method for signature 'tscm'
show(object)

## S4 method for signature 'tscm'
coef(object)

## S4 method for signature 'tscm'
sim(object, n = 1000)
```

Arguments

- `object` an object of the class.
- `n` length of realization.

Methods (by generic)

- `show`: Show method for `tscm` class
- `coef`: Coefficient method for `tscm` class
- `sim`: Simulation method for `tscm` class

Slots

- `tscopula` an object of class `tscopula`.
- `margin` an object of class `margin`.
Examples

```r
mod <- tscm(dvinecopula(family = "gauss", pars = 0.5), margin("doubleweibull"))
sim(mod)
```

---

**tscmfit-class**

**Fitted tscm model**

**Description**

Class of objects for fitted tscm models.

**Usage**

```r
## S4 method for signature 'tscmfit'
logLik(object)

## S4 method for signature 'tscmfit'
resid(object, trace = FALSE)
```

**Arguments**

- `object`: an object of the class.
- `trace`: extract trace instead of residuals.

**Methods (by generic)**

- `logLik`: method for tscmfit class
- `resid`: Residual method for tscmfit class

**Slots**

- `tscopula`: an object of class `tscopula`.
- `margin`: an object of class `margin`.
- `data`: a vector or time series of data to which process has been fitted.
- `fit`: a list containing details of the fit.

---

**tscopula-class**

**Time series copula processes**

**Description**

Class of objects for time series copula processes.
**tscopulafit-class**  
*Fitted time series copula processes*

**Description**

Class of objects for fitted time series copula processes.

**Usage**

```r
## S4 method for signature 'tscopulafit'
sim(object, n = 1000)

## S4 method for signature 'tscopulafit'
coef(object)

## S4 method for signature 'tscopulafit'
show(object)

## S4 method for signature 'tscopulafit'
logLik(object)

## S4 method for signature 'tscopulafit'
resid(object, trace = FALSE)
```

**Arguments**

- `object`: an object of class `tscopulafit`.
- `n`: length of realization.
- `trace`: extract trace instead of residuals.

**Methods (by generic)**

- `sim`: Simulation method for tscopulafit class
- `coef`: Coef method for tscopulafit class
- `show`: Show method for tscopulafit objects
- `logLik`: logLik method for tscopulafit class
- `resid`: Residual method for tscopulafit class

**Slots**

- `tscopula`: an object of class `tscopula`.
- `data`: a vector or time series of data.
- `fit`: a list containing details of the fit.
Examples

```r
ar1 <- armacopula(list(ar = 0.7))
data <- sim(ar1, 1000)
ar1fit <- fit(ar1, data)
sim(ar1fit)
```

tscopulaU-class  
Time series copulas of class tscopulaU

Description

S4 Class union for basic time series copula types. These are armacopula, dvinecopula and dvinecopula2.

V2b  
Constructor function for 2-parameter beta v-transform

Description

Constructor function for 2-parameter beta v-transform

Usage

```r
V2b(delta = 0.5, kappa = 1)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delta</td>
<td>a value in (0, 1) specifying the fulcrum of the v-transform.</td>
</tr>
<tr>
<td>kappa</td>
<td>additional positive parameter of v-transform.</td>
</tr>
</tbody>
</table>

Value

An object of class Vtransform.

Examples

```r
V2b(delta = 0.45, kappa = 1.2)
```
**V2p**

*Constructor function for 2-parameter v-transform*

**Description**

Constructor function for 2-parameter v-transform

**Usage**

\[ V2p(\delta = 0.5, \kappa = 1) \]

**Arguments**

- \( \delta \): a value in \((0, 1)\) specifying the fulcrum of the v-transform.
- \( \kappa \): additional positive parameter of v-transform.

**Value**

An object of class `Vtransform`.

**Examples**

\[ V2p(\delta = 0.45, \kappa = 1.2) \]

---

**V3b**

*Constructor function for 3-parameter beta v-transform*

**Description**

Constructor function for 3-parameter beta v-transform

**Usage**

\[ V3b(\delta = 0.5, \kappa = 1, \xi = 1) \]

**Arguments**

- \( \delta \): a value in \((0, 1)\) specifying the fulcrum of the v-transform.
- \( \kappa \): additional positive parameter of v-transform.
- \( \xi \): additional positive parameter of v-transform.

**Value**

An object of class `Vtransform`.

**Examples**

\[ V3b(\delta = 0.45, \kappa = 1.2, \xi = 1.2) \]
**V3p**

*Constructor function for 3-parameter v-transform*

---

**Description**

Constructor function for 3-parameter v-transform

**Usage**

\[ \text{V3p}(\text{delta} = 0.5, \text{kappa} = 1, \text{xi} = 1) \]

**Arguments**

- **delta**: a value in \((0, 1)\) specifying the fulcrum of the v-transform.
- **kappa**: additional positive parameter of v-transform.
- **xi**: additional positive parameter of v-transform.

**Value**

An object of class `Vtransform`.

**Examples**

\[ \text{V3p}(\text{delta} = 0.45, \text{kappa} = 0.8, \text{xi} = 1.1) \]

---

**Vdegenerate**

*Constructor function for degenerate v-transform*

---

**Description**

Constructor function for degenerate v-transform

**Usage**

\[ \text{Vdegenerate}() \]

**Value**

An object of class `VtransformI`.

**Examples**

\[ \text{Vdegenerate}() \]
vdownprob  
**Calculate conditional down probability of v-transform**

**Description**
Calculate conditional down probability of v-transform

**Usage**

```r
vdownprob(x, v)
```

**Arguments**
- `x` an object of class `Vtransform`.
- `v` a vector or time series with values in \([0, 1]\).

**Value**
A vector or time series of values of gradient.

**Examples**

```r
vdownprob(V2p(delta = 0.55, kapp = 1.2), c(0, 0.25, 0.5, 0.75, 1))
```

vgradient  
**Calculate gradient of v-transform**

**Description**
Calculate gradient of v-transform

**Usage**

```r
vgradient(x, u)
```

**Arguments**
- `x` an object of class `Vtransform`.
- `u` a vector or time series with values in \([0, 1]\).

**Value**
A vector or time series of values of gradient.

**Examples**

```r
vgradient(Vsymmetric(), c(0, 0.25, 0.5, 0.75, 1))
```
vinverse

Calculate inverse of v-transform

Description

If the Vtransform object is also a VtransformI object (an invertible v-transform) then the analytical inverse is used. Otherwise an inverse is found by numerical root finding with `uniroot`.

Usage

\[
\text{vinverse}(x, v, \text{tol} = \text{.Machine}$\text{double.eps}^0.75)
\]

Arguments

- **x**: an object of class Vtransform.
- **v**: a vector or time series with values in \([0, 1]\).
- **tol**: the desired accuracy (convergence tolerance) that is passed to `uniroot` if numerical inversion is used.

Value

A vector or time series with values in \([0, 1]\).

Examples

\[
\text{vinverse(Vsymmetric()}, \text{c(0, 0.25, 0.5, 0.75, 1)})
\]

Vlinear

Constructor function for linear v-transform

Description

Constructor function for linear v-transform

Usage

\[
\text{Vlinear}(\text{delta} = 0.5)
\]

Arguments

- **delta**: a value in \((0, 1)\) specifying the fulcrum of the v-transform.

Value

An object of class VtransformI.
Examples
Vlinear(delta = 0.45)

Vsymmetric
Constructor function for symmetric v-transform

Description
Constructor function for symmetric v-transform

Usage
Vsymmetric()

Value
An object of class VtransformI.

Examples
Vsymmetric()

vtrans
Evaluate a v-transform

Description
Evaluate a v-transform

Usage
vtrans(x, u)

Arguments
x
an object of class Vtransform.
u
a vector or time series with values in [0, 1].

Value
A vector or time series with values in [0, 1].

Examples
vtrans(Vsymmetric(), c(0, 0.25, 0.5, 0.75, 1))
Vtransform-class

Class of v-transforms

Description

This is the class of v-transforms. It contains the VtransformI subclass consisting of v-transforms with an analytical expression for the inverse.

Usage

```r
## S4 method for signature 'Vtransform'
show(object)
```

```r
## S4 method for signature 'Vtransform'
coef(object)
```

Arguments

object an object of the class.

Methods (by generic)

- show: Show method for Vtransform class
- coef: Coef method for Vtransform class

Slots

name a name for the v-transform of class character.

Vtrans function to evaluate the v-transform.

pars vector containing the named parameters of the v-transform.

gradient function to evaluate the gradient of the v-transform.

Examples

```r
V2p(delta = 0.5, kappa = 1.2)
```
VtransformI-class  

Class of invertible v-transforms

Description

This class inherits from the Vtransform class and contains v-transforms with an analytical expression for the inverse.

Slots

- name: a name for the v-transform of class character.
- Vtrans: function to evaluate the v-transform.
- pars: vector containing the named parameters of the v-transform.
- gradient: function to evaluate the gradient of the v-transform.
- inverse: function to evaluate the inverse of the v-transform.

Examples

Vlinear(delta = 0.55)

vtscopula

Constructor function for vtscopula object

Description

Constructor function for vtscopula object

Usage

vtscopula(tscopulaU, Vtransform = Vlinear(), Wcopula = swncopula())

Arguments

- tscopulaU: an object of class armacopula, dvinecopula or dvinecopula2.
- Vtransform: an object of class Vtransform.
- Wcopula: an object of class tscopula.

Value

An object of class vtscopula.

Examples

copobject <- armacopula(pars = list(ar = 0.6, ma = 0.2))
vtscopula(copobject, Vtransform = V2p())
Description

Class of objects for v-transformed time series copula processes.

Usage

## S4 method for signature 'vtscopula'
show(object)

## S4 method for signature 'vtscopula'
coef(object)

## S4 method for signature 'vtscopula'
sim(object, n = 1000)

## S4 method for signature 'vtscopula'
kendall(object, lagmax = 20)

Arguments

object     an object of the class.
n          length of realization.
lagmax     maximum value of lag.

Methods (by generic)

- show: Show method for vtscopula objects
- coef: Coef method for vtscopula class
- sim: Simulation method for vtscopula class
- kendall: Calculate Kendall’s tau values for vtscopula model

Slots

Vcopula  object of class tscopulaU.
Vtransform  object of class Vtransform.
Wcopula  object of class tscopula.

Examples

copobject <- armacopula(pars = list(ar = 0.6, ma = 0.2))
sim(vtscopula(copobject, Vtransform = V2p()))
mod <- vtscopula(aramacopula(list(ar = 0.95, ma = -0.85)))
kendall(mod)
Index

* datasets
  bitcoin, 5
  cpi, 7

acf2pacf, 3
armacopula, 4, 4, 17, 28, 30, 39, 46
armacopula-class, 4

bitcoin, 5
coef, armacopula-method
  (armacopula-class), 4
coef, dvinecopula-method
  (dvinecopula-class), 9
coef, dvinecopula2-method
  (dvinecopula2-class), 11
coef, margin-method (margin-class), 20
coef, swncopula-method
  (swncopula-class), 35
coef, tscm-method (tscm-class), 36
coef, tscopulafit-method
  (tscopulafit-class), 38
coef, Vtransform-method
  (Vtransform-class), 45
coef, vtscopula-method
  (vtscopula-class), 47
c coerce, tscopula, tscm-method, 6
c coerce, tscopulafit, tscmfit-method, 6
cpi, 7
ddoubleweibull (doubleweibull), 8
dlaplace (laplace), 19
dnorm, 20
doubleweibull, 8
dsdoubleweibull (sddoubleweibull), 29
dslaplace (slaplace), 31
dsst (sst), 31
dst (st), 32
dvinecopula, 9, 9, 17, 30, 39, 46
dvinecopula-class, 9
dvinecopula2, 10, 11, 17, 30, 39, 46
dvinecopula2-class, 11

fit, 12
fit, margin-method, 13
fit, tscm-method, 13
fit, tscopulafit-method, 14
fit, tscopulaU-method, 15
fit, vtscopula-method, 15
glag, 16

kendall, 17
kendall, armacopula-method
  (armacopula-class), 4
kendall, dvinecopula-method
  (dvinecopula-class), 9
kendall, dvinecopula2-method
  (dvinecopula2-class), 11
kendall, vtscopula-method
  (vtscopula-class), 47

kfilter, 17
kpacf_arfima, 18
kpacf_arma, 18
kpacf_fbn, 19

laplace, 19
logLik, marginfit-method
  (marginfit-class), 21
logLik, tscmfit-method (tscmfit-class), 37
logLik, tscopulafit-method
  (tscopulafit-class), 38

margin, 7, 12, 13, 20, 20, 21, 26, 28, 30, 36, 37
margin-class, 20
marginfit, 13, 24
marginfit-class, 21

non_invert, 22
INDEX

non_stat, 22
optim, 13–16
pacf2acf, 23
pcoincide, 23
pdoubleweibull (doubleweibull), 8
plaplace (laplace), 19
plot, marginfit, missing-method, 24
plot, tscmfit, missing-method, 24
plot, tscopulafit, missing-method, 25
plot, Vtransform, missing-method, 25
pmarg, 26
pnorm, 20
profilefulcrum, 27
psdoubleweibull (sdoubleweibull), 29
pslaplace (slaplace), 31
psst (sst), 31
pstat (st), 32
qdoubleweibull (doubleweibull), 8
qlaplace (laplace), 19
qmarg, 28
qnorm, 20
qsdoubleweibull (sdoubleweibull), 29
qslaplace (slaplace), 31
qsst (sst), 31
qstat (st), 32
quantile, tscmfit-method, 28
rdoubleweibull (doubleweibull), 8
resid, tscmfit-method (tscmfit-class), 37
resid, tscopulafit-method (tscopulafit-class), 38
rlaplace (laplace), 19
rnorm, 20
rsdoubleweibull (sdoubleweibull), 29
rslaplace (slaplace), 31
rsst (sst), 31
rstat (st), 32
safe_ses, 29
sdoubleweibull, 29
show, armacopula-method (armacopula-class), 4
show, dvinecopula-method (dvinecopula-class), 9
show, dvinecopula2-method (dvinecopula2-class), 11
show, margin-method (margin-class), 20
show, swncopula-method (swncopula-class), 20
show, tscm-method (tscm-class), 36
show, tscopulafit-method (tscopulafit-class), 38
show, Vtransform-method (Vtransform-class), 45
show, vtscopula-method (vtscopula-class), 47
sigmastarma, 30
sim, 30
sim, armacopula-method (armacopula-class), 4
sim, dvinecopula-method (dvinecopula-class), 9
sim, dvinecopula2-method (dvinecopula2-class), 11
sim, margin-method (margin-class), 20
sim, swncopula-method (swncopula-class), 35
sim, tscm-method (tscm-class), 36
sim, tscopulafit-method (tscopulafit-class), 38
sim, vtscopula-method (vtscopula-class), 47
slaplace, 31
sst, 31
st, 32
stochinverse, 33
strank, 34
swncopula, 30, 34, 35, 35
swncopula-class, 35
tacvfARMA, 30
tscm, 6, 12, 14, 30, 35, 36, 37
tscm-class, 36
tscmfit, 6, 12, 14, 24, 28
tscmfit-class, 37
tscopula, 6, 36–38, 46, 47
tscopula-class, 37
tscopulafit, 6, 12, 14–16, 25, 38
tscopulafit-class, 38
tscopulaU, 12, 15, 27, 47
tscopulaU-class, 39
uniroot, 43
V2b, 39
V2p. 40
V3b. 40
V3p. 41
Vdegenerate. 41
vdowp, 42
vgradient. 42
vinverse. 43
Vlinear. 43
Vsymmetric. 44
vtrans. 44
Vtransform. 23, 26, 33, 39–44, 46, 47
Vtransform-class. 45
VtransformI. 41, 43–45
VtransformI-class. 46
vtscopula. 12, 15–17, 27, 46, 46
vtscopula-class. 47