

Package ‘Multi’

December 19, 2019

Title Tools for modeling compound events

Version 0.0.0.9000

Description What the package does (one paragraph).

License What license it uses

Encoding UTF-8

LazyData true

Imports texmex,
fitdistrplus,
tweedie,
MASS,
dplyr,
VGAM,
copula,
GeneralizedHyperbolic,
statmod,
RColorBrewer,
VineCopula,
ks,
truncnorm,
actuar

RoxygenNote 6.1.1

R topics documented:

Con_Sampling_2D	2
Copula_Threshold_2D	3
Dataframe_Combine	4
Decluster	5
Detrend	5
Diag_Non_Con	6
Diag_Non_Con_Sel	7
Diag_Non_Con_Sel_Trunc	8
Diag_Non_Con_Trunc	8
GPD_Fit	9
HT04	10
Imputation	11
Kendall_Lag	12

Migpd_Fit	13
SLR_Scenarios	14
Standard_Copula_Fit	14
Standard_Copula_Sel	15
Standard_Copula_Sim	16
Vine_Copula_Fit	16
Vine_Copula_Sim	17
Index	19

Con_Sampling_2D	<i>Conditionally sampling a two dimensional dataset</i>
-----------------	---

Description

Creates a dataframe where the declustered excesses of a (conditioning) variable are paired with co-occurrences of another variable.

Usage

```
Con_Sampling_2D(Data_Detrend, Data_Declust, Con_Variable, Thres = 0.97)
```

Arguments

Data_Detrend	Dataframe containing two at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned NA. First object may be a "Date" object. Can be Dataframe_Combine output.
Data_Declust	Dataframe containing two (independently) declustered at least partially concurrent time series. Time steps must be equally spaced, with missing values assigned NA. Columns must be in the same order as in Data_Detrend. First object may be a "Date" object. Can be Dataframe_Combine output.
Con_Variable	Column number (1 or 2) or the column name of the conditioning variable. Default is 1.
Thres	Threshold, as a quantile of the observations of the conditioning variable. Default is 0.97.

Value

List comprising the specified Threshold as the quantile of the conditioning variable above which declustered excesses are paired with co-occurrences of the other variable, the resulting two dimensional sample data and name of the conditioning variable.

Examples

```
Conditional_Sampling_2D(Data_Detrend=Vic[, -1], Data_Declust=Vic.Data.Declustered[, -1], Con_Variable=1, Thres=
```

Description

Declustered excesses of a (conditioning) variable are paired with co-occurrences of the other variable before the best fitting bivariate copula is selected, using BiCopSelect function in the VineCopula package, for a single or range of thresholds. The procedure is automatically repeated with the variables switched.

Usage

```
Copula_Threshold_2D(Data_Detrend, Data_Declust, Thres = seq(0.9, 0.99,
  0.01), x_lim_min = min(Thres), x_lim_max = max(Thres),
  y_lim_min = -1, y_lim_max = 1, Upper = 0, Lower = 0,
  GAP = 0.05)
```

Arguments

Data_Detrend	Dataframe containing two at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned NA.
Data_Declust	Dataframe containing two (independently) declustered at least partially concurrent time series. Time steps must be equally spaced, with missing values assigned NA.
Thres	A single or sequence of thresholds, given as a quantile of the observations of the conditioning variable. Default, sequence from 0.9 to 0.99 at intervals of 0.01.
x_lim_min	Numeric vector of length one specifying x-axis minimum. Default is the maximum argument in \cpdeThres.
x_lim_max	Numeric vector of length one specifying x-axis maximum. Default is the minimum argument in Thres.
y_lim_min	Numeric vector of length one specifying y-axis minimum. Default -1.0.
y_lim_max	Numeric vector of length one specifying y-axis maximum. Default 1.0.
Upper	Numeric vector specifying the element number of the Thres argument for which the copula family name label to appear above the corresponding point on the Kendall's tau coefficient vs threshold plot, when condition on the variable in column 1. Default is 0.
Lower	Numeric vector specifying the element number of the Thres argument for which the copula family name label to appear below the corresponding point on the Kendall's tau coefficient vs threshold plot, when condition on the variable in column 2. Default is 0.
GAP	Numeric vector of length one specifying the distance above or below the copula family name label appears the corresponding point on the Kendall's tau coefficient vs threshold plot. Default is 0.05.

Value

- List comprising:
- `Kendalls_Tau_Var1` \newline Kendalls tau of a sample
 - `p_value_Var1` p-value when testing the null hypothesis H_0 i.e. that there is no correlation between the variables
 - `N_Var1` size of the dataset
 - `Copula_Family_Var1` best fitting copula for the specified thresholds

when the dataset is conditioned on the variable in column 1. Analogous vector `Kendalls_Tau_Var2`, `p_value_Var2`, `N_Var2` and `Copula_Family_Var2` for the specified thresholds when the dataset is conditioned on the variable in column 2.

Examples

`Copula_Threshold_2D(Data_Detrend=S28_Detrend,Data_Declust=S28_Detrend_Declustered,y_lim=c(-0.075,0.25),Upper=0.05)`

Dataframe_Combine	<i>Creates a dataframe containing up to five time series</i>
-------------------	--

Description

Combines up to five time series, detrended where necessary, into a single dataframe.

Usage

```
Dataframe_Combine(data.1, data.2, data.3, data.4 = 0, data.5 = 0,
  n = 3, names)
```

Arguments

- | | |
|-----------------------|---|
| <code>n</code> | Integer 1-5 specifying the number of time series. Default is 3. |
| <code>data.1:5</code> | Dataframes with two columns containing in column <ul style="list-style-type: none">• 1 - Continuous sequence of times spanning from the first to the final recorded observations.• 2 - Corresponding values detrended where necessary. |

Value

A dataframe containing all times from the first to the most up to date reading of any of the variables.

See Also

[Detrend](#)

Examples

```
#Reading in data for site S22
Miami_Airport_df<-read.csv("C:\\Users\\ro327497\\Documents\\SFWMD\\SFWMD Data\\S22\\Miami_Airport_df.csv")
S22_T_MAX_Daily_Completed_Detrend_Declustered<-read.csv("C:\\Users\\ro327497\\Documents\\SFWMD\\SFWMD Data\\S22\\S22_T_MAX_Daily_Completed_Detrend_Declustered.csv")
G580A_GWValueFilled_Detrend_Declustered<-read.csv("C:\\Users\\ro327497\\Documents\\SFWMD\\SFWMD Data\\S22\\G580A_GWValueFilled_Detrend_Declustered.csv")
S22_Detrend_Declustered.df<-Dataframe_Combine(data.1<-Miami_Airport_df,data.2<-S22_T_MAX_Daily_Completed_Detrend_Declustered.df,data.3<-G580A_GWValueFilled_Detrend_Declustered.df)
S22_Detrend.df<-Dataframe_Combine(data.1<-Miami_Airport_df,data.2<-S22_T_MAX_Daily_Completed_Detrend_Declustered.df,data.3<-G580A_GWValueFilled_Detrend_Declustered.df)
```

Decluster	<i>Declusters a time series</i>
-----------	---------------------------------

Description

Identify cluster maxima above a threshold, using the runs method of Smith and Weissman (1994).

Usage

```
Decluster(Data, u = 0.95, SepCrit = 3, mu = 365.25)
```

Arguments

Data	Numeric vector of the time series.
u	Numeric vector of length one specifying the declustering threshold; as a quantile $[0, 1]$ of Data vector. Default is 0.95.
SepCrit	Integer; specifying the separation criterion under which events are declustered. Default is 3 corresponding to a storm window of three days in the case of daily data.
mu	(average) Number of events per year. Numeric vector of length one. Default is 365.25, daily data.

Value

List comprising the Threshold above which cluster maxima are identified, average number of declustered excesses per year EventsPerYear, a vector containing the original time series Detrended and the Declustered series.

Examples

```
Decluster(data=S28_T_MAX_Daily_Completed_Detrend$Detrend)
```

Detrend	<i>Detrends a time series.</i>
---------	--------------------------------

Description

Detrends a time series using either a linear fit covering the entire dataset or moving average trend correction with a user-specified window width.

Usage

```
Detrend(Data, Method = "window", Window_Width = 89,  
End_Length = 1826, PLOT = FALSE, x_lab = "Data", y_lab = "Data")
```

Arguments

Data	Dataframe containing two columns. In column: <ul style="list-style-type: none"> • 1 A "Date" object of equally spaced discrete time steps. • 2 Numeric vector containing corresponding time series values. No NAs allowed.
Method	Character vector of length one specifying approach used to detrend the data. Options are moving average "window" (default) and "linear".
Window_Width	Numeric vector of length one specifying length of the moving average window. Default is 89, window comprises the observation plus 44 days either side, which for daily data corresponds to an approximate 3 month window.
End_Length	Numeric vector of length one specifying number of observations at the end of the time series used to calculate the present day average. Default is 1826, which for daily data corresponds to the final five years of observations.
PLOT	Logical; whether to plot original and detrended series. Default is "FALSE".
x_lab	Character vector of length one specifying x-axis label. Default is "Date".
y_lab	Character vector of length one specifying y-axis label. Default is "Data".

Value

Numeric vector of the detrended time series.

Diag_Non_Con	<i>Goodness of fit of non-extreme marginal distributions</i>
--------------	--

Description

Fits nine non-extreme marginal distributions to a dataset and returns three plots demonstrating their relative goodness of fit.

Usage

```
Diag_Non_Con(Data, x_lab, y_lim_min = 0, y_lim_max = 1)
```

Arguments

Data	Numeric vector containing realizations of the variable of interest.
x_lab	Character vector of length one specifying the label on the x-axis of histogram and cumulative distribution plot.
y_lim_min	Numeric vector of length one specifying the lower y-axis limit of the histogram. Default is 0.
y_lim_max	Numeric vector of length one specifying the upper y-axis limit of the histogram. Default is 1.

Value

Panel consisting of three plots. Upper plot: Plot depicting the AIC of the nine fitted distributions. Middle plot: Probability Density Functions (PDFs) of the fitted distributions superimposed on a histogram of the data. Lower plot: Cumulative Distribution Functions (CDFs) of the fitted distributions overlaid on a plot of the empirical CDF.

Examples

```
Diag_Non_Con(Vic.Data.Surge$Discharge,x_lab="Discharge",y_lim_min=0,y_lim_max=0.002)
```

Diag_Non_Con_Sel	<i>Demonstrate the goodness of fit of the slected non-extreme marginal distribution</i>
------------------	---

Description

Plots demonstrating the goodness of fit of a selected non-extreme marginal distribution to a dataset.

Usage

```
Diag_Non_Con_Sel(Data, x_lab = "Data", y_lim_min = 0, y_lim_max = 1,
  Selected)
```

Arguments

Data	Numeric vector containing realizations of the variable of interest.
x_lab	Numeric vector of length one specifyingLabel on the x-axis of histogram and cummulative distribution plot.
y_lim_min	Numeric vector of length one specifying the lower y-axis limit of the histogram.
y_lim_max	Numeric vector of length one specifying the upper y-axis limit of the histogram.
Selected	Charactor vector of length one specifying the chosen distribution, options are the Birnbaum-Saunders "BS", exponential "Exp", gamma "Gam", Gaussian "Gaus", inverse Gaussian "InvG", logistic "Logis", lognormal "LogN", Tweedie "Twe" and Weibull "Weib".

Value

Panel consisting of three plots. Upper plot: Plots depicting the AIC of the nine fitted distributions. Middle plot: Probabilty Density Functions (PDFs) of the selected distriptions superimposed on a histogram of the data. Lower plot: Cummulative distribution function (CDFs) of the selected distribution overlaid on a plot of the empirical CDF.

Examples

```
Diag_Non_Con_Sel(Data,x_lab,y_lim_min,y_lim_max,Selected)
```

Diag_Non_Con_Sel_Trunc

Goodness of fit of non-extreme marginal distributions

Description

Fits nine non-extreme marginal distributions to a dataset and returns three plots demonstrating their relative goodness of fit.

Usage

```
Diag_Non_Con_Sel_Trunc(Data, x_lab, y_lim_min = 0, y_lim_max = 1,
  Selected)
```

Arguments

Data	Numeric vector containing realizations of the variable of interest.
x_lab	Character vector of length one specifying the label on the x-axis of histogram and cummulative distribution plot.
y_lim_min	Numeric vector of length one specifying the lower y-axis limit of the histogram. Default is 0.
y_lim_max	Numeric vector of length one specifying the upper y-axis limit of the histogram. Default is 1.

Value

Panel consisting of three plots. Upper plot: Plot depicting the AIC of the nine fitted distributions. Middle plot: Probabilty Density Functions (PDFs) of the fitted distriptions superimposed on a histogram of the data. Lower plot: Cummulaibre Distribution Functions (CDFs) of the fitted distributions overlaid on a plot of the empirical CDF.

Examples

```
Diag_Non_Con(Vic.Data.Surge$Discharge,x_lab="Discharge",y_lim_min=0,y_lim_max=0.002)
```

Diag_Non_Con_Trunc

Goodness of fit of non-extreme marginal distributions

Description

Fits nine non-extreme marginal distributions to a dataset and returns three plots demonstrating their relative goodness of fit.

Usage

```
Diag_Non_Con_Trunc(Data, x_lab, y_lim_min = 0, y_lim_max = 1)
```


Arguments

Data	Numeric vector containing realizations of the variable of interest.
x_lab	Character vector of length one specifying the label on the x-axis of histogram and cumulative distribution plot.
y_lim_min	Numeric vector of length one specifying the lower y-axis limit of the histogram. Default is 0.
y_lim_max	Numeric vector of length one specifying the upper y-axis limit of the histogram. Default is 1.

Value

Panel consisting of three plots. Upper plot: Plot depicting the AIC of the nine fitted distributions. Middle plot: Probability Density Functions (PDFs) of the fitted distributions superimposed on a histogram of the data. Lower plot: Cumulative Distribution Functions (CDFs) of the fitted distributions overlaid on a plot of the empirical CDF.

Examples

```
Diag_Non_Con(Vic.Data.Surge$Discharge,x_lab="Discharge",y_lim_min=0,y_lim_max=0.002)
```

GPD_Fit

*Fits a single generalised Pareto distribution - Fit***Description**

Fit a Generalized Pareto Distribution (GPD) to a declustered dataset.

Usage

```
GPD_Fit(Data, Data_Full, u = 0.95, PLOT = FALSE, xlab_hist = "Data",
        y_lab = "Data")
```

Arguments

Data	Numeric vector containing the declustered data.
Data_Full	Numeric vector containing the non-declustered data.
u	GPD threshold; as a quantile $[0, 1]$ of Data vector. Default is 0.95.
xlab_hist	Character vector of length one. Histogram x-axis label. Default is "Data".
y_lab	Character vector of length one. Histogram x-axis label. Default is "Data".
Plot	Logical; indicating whether to plot diagnostics. Default is FALSE.

Value

List comprising the GPD Threshold, shape parameter ξ and scale parameters σ along with their standard errors $\sigma.SE$ and $\xi.SE$.

Details

The fitted GPD model, is following parameterised as follows: $P(X > x|X > u)$

Examples

```
Decluster(Data=S28_T_MAX_Daily_Completed_Detrend$Detrend)
```

HT04	<i>Fits and simulates from the conditional multivariate approach of Heffernan and Tawn (2004)</i>
------	---

Description

Fitting and simulating the conditional multivariate approach of Heffernan and Tawn (2004) to a dataset comprising 3 variables. Function utilizes the `mexDependence` and functions from the `texmex` package.

Usage

```
HT04(data_Detrend_Dependence_df, data_Detrend_Declustered_df, u_Dependence,
      Migpd, mu = 365.25, N = 100, Margins = "gumbel", V = 10,
      Maxit = 10000)
```

Arguments

<code>data_Detrend_Dependence_df</code>	<p>A dataframe with (n+1) columns, containing in column</p> <ul style="list-style-type: none"> • 1 - Continuous sequence of dates spanning the first to the final time of any of the variables are recorded. • 2:(n+1) - Values, detrended where necessary, of the variables to be modelled.
<code>data_Detrend_Declustered_df</code>	<p>A dataframe with (n+1) columns, containing in column</p> <ul style="list-style-type: none"> • 1 - Continuous sequence of dates spanning the first to the final time of any of the variables are recorded. • 2:(n+1) - Declustered and if necessary detrended values of the variables to be modelled.
<code>u_Dependence</code>	Dependence quantile. Specifies the (sub-sample of) data to which the dependence model is fitted, that for which the conditioning variable exceeds the threshold associated with the prescribed quantile. Default is 0.7, thus the dependence parameters are estimated using the data with the highest 30% of values of the conditioning variables.
<code>Migpd</code>	An <code>Migpd</code> object, containing the generalised Pareto models fitted (independently) to each of the variables.
<code>Margins</code>	Character vector specifying the form of margins to which the data are transformed for carrying out dependence estimation. Default is "gumbel", alternative is "laplace". Under Gumbel margins, the estimated parameters a and b describe only positive dependence, while c and d describe negative dependence in this case. For Laplace margins, only parameters a and b are estimated as these capture both positive and negative dependence.
<code>V</code>	See documentation for <code>mexDependence</code> .
<code>Maxit</code>	See documentation for <code>mexDependence</code> .

Value

List comprising the fitted HT04 models Models, proportion of the time each variable is most extreme, given at least one variable is extreme Prop, as well as the simulated values on the transformed u.sim and original x.sim scales.

See Also

[Detrend_Combine](#) [Detrend_Declustered_Combine](#)

Examples

```
HT04(data_Detrend_Dependence_df = S22.Detrend.df, data_Detrend_Declustered_df = S22.Detrend.Declustered.df , M
```

Imputation	<i>Imputing missing values through linear regression</i>
------------	--

Description

Fits a simple linear regression model, impute missing values of the dependent variable.

Usage

```
Imputation(Data, Variable, x_lab, y_lab)
```

Arguments

Data	Dataframe containing two at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.
Variable	Character vector of length one specifying the (column) name of the variable to be imputed i.e. dependent variable in the fitted regression.
x_lab	Character vector of length one specifying the name of the independent variable to appear as the x-axis label on a plot showing the data, imputed values and the linear regression model.
y_lab	Character vector of length one specifying the name of the dependent variable to appear as the y-axis label on plot showing the data, imputed values and the linear regression model.

Value

List comprising

- Data dataframe containing the original data plus an additional column named Value where the NA values of the Variable of interest have been imputed where possible.
- Model linear regression model paramters including its coefficient of determenation

and a scatter plot of the data (black points), linear regression model (red line) and fitted (imputed) values (blue points).

Examples

```
####Objective: Fill in missing values at groundwater well G_3356 using record at G_3355
##Reading in data at G_3356
G_3356<-read.csv("C:\\Users\\ro327497\\Documents\\SFWMD\\SFWMD Data\\S20\\G-3356.csv",header=TRUE)[-12273,]
#Converting date column to a "Date" object
G_3356$Date<-seq(as.Date("1985-10-23"), as.Date("2019-05-29"), by="day")
#Converting readings to numeric object
G_3356$Value<-as.numeric(as.character(G_3356$Value))

##Reading in data at G_3355
G_3355<-read.csv("C:\\Users\\ro327497\\Documents\\SFWMD\\SFWMD Data\\S20\\G-3355.csv")[-12341,c(3,4)]
#Converting date column to a "Date" object
G_3355$Date<-seq(as.Date("1985-08-20"), as.Date("2019-06-02"), by="day")
#Converting readings to numeric object
G_3355$Value<-as.numeric(as.character(G_3355$Value))

##Merge the two dataframes by date
GW_S20<-merge(G_3356,G_3355,by="Date")
colnames(GW_S20)<-c("Date","G3356","G3355")
#Carrying out imputation
Imputation(Data=GW_S20,Variable="G3356",x_lab="Groundwater level (ft NGVD)",y_lab="Groundwater level (ft NGVD)")
```

Kendall_Lag

Kendall's tau correlation coefficient between pairs of variables over a range of lags

Description

Kendall's tau correlation coefficient between pairs of variables over a range of lags

Usage

```
Kendall_Lag(Data, Lags = seq(-6, 6, 1), PLOT = TRUE, GAP = 0.1)
```

Arguments

Data	A data frame with 3 columns, containing concurrent observations of three time series.
Lags	Integer vector giving the lags over which to calculate coefficient. Default is a vector from -6 to 6.
Plot	Logical; whether to show plot of Kendall's coefficient vs lag. Default is TRUE.

Value

List comprising Kendall's tau coefficients between the variables pairs composing columns of Data with the specified lags applied to the second named variable Values and the p-values Test when testing the null hypothesis H_0 : $\tau=0$ i.e. there is no correlation between a pair of variables. Plot of the coefficient with a filled point of hypothesis test ($p\text{-value}<0.05$). Lag applied to variable named second in the legend.

Examples

```
Kendall_Lag(Data=S22.Detrend.df,GAP=0.1)
```

Migpd_Fit

*Fits Multiple independent generalized Pareto models - Fit***Description**

Fit multiple independent generalized Pareto models to each column of a dataframe. Edited version of the migpd function in texmex, to allow for NAs in a time series.

Usage

```
Migpd_Fit(Data, mth, mqu, penalty = "gaussian", maxit = 10000,
  trace = 0, verbose = FALSE, priorParameters = NULL)
```

Arguments

Data	A dataframe with n columns, each comprising a declustered and if necessary detrended time series to be modelled.
mth	Marginal thresholds, above which generalized Pareto models are fitted. Numeric vector of length n.
mqu	Marginal quantiles, above which generalized Pareto models are fitted. Only one of mth and mqu should be supplied. Numeric vector of length n.
penalty	See ggplot.migpd .
maxit	See ggplot.migpd .
trace	See ggplot.migpd .
verbose	See ggplot.migpd .
priorParameters	See ggplot.migpd .

Value

An object of class "migpd". There are coef, print, plot, ggplot and summary functions available.

See Also

[Decluster Detrend Dataframe_Combine](#)

Examples

```
#With date as first column
S22.GPD<-Migpd_Fit(Data=S22.Detrend.Declustered.df, mqu =c(0.99,0.99,0.99))
#Without date as first column
S22.GPD<-Migpd_Fit(Data=S22.Detrend.Declustered.df[, -1], mqu =c(0.99,0.99,0.99))
```

SLR_Scenarios	<i>Sea level rise scenarios in the Southeast Florida Regional Climate Change Compact:</i>
---------------	---

Description

Calculates and plots time required for sea level rise to reach a specified level according to the three scenarios in the Compact.

Usage

```
SLR_Scenarios(SeaLevelRise, Unit = "m")
```

Arguments

SeaLevelRise	Numeric vector of length one, sea level rise required.
data	A dataframe with n columns, each comprising a declustered and if necessary detrended time series to be modelled.

Value

An object of class "migpd". There are coef, print, plot, ggplot and summary functions available.

Examples

```
SLRScenarios(0.45)
```

Standard_Copula_Fit	<i>Fit an Archimedean/elliptic copula model - Fit</i>
---------------------	---

Description

Fit a n-dimensional Archimedean or elliptic copula model. Function is simply a repackaging of the fitCopula function in the copula package.

Usage

```
Standard_Copula_Fit(Data, Copula_Type = "Gaussian")
```

Arguments

Data	Dataframe containing n at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.
Copula_Type	Type of elliptical copula to be fitted, options are "Gaussian" (Default), "tcopula", "Gumbel", "Clayton" and "Frank".

Value

List comprising the Copula_Type and the fitted copula Model object.

See Also

[Dataframe_Combine](#) [CDVineCopSelect](#) [BiCopSelect](#)

Examples

```
cop<-Standard_Copula_Fit(Data=S22.Detrend.df,Copula_Type="Gaussian")
cop<-Standard_Copula_Fit(Data=S22.Detrend.df,Copula_Type="tcopula")
cop<-Standard_Copula_Fit(Data=S22.Detrend.df,Copula_Type="Gumbel")
cop<-Standard_Copula_Fit(Data=S22.Detrend.df,Copula_Type="Clayton")
cop<-Standard_Copula_Fit(Data=S22.Detrend.df,Copula_Type="Frank")
```

Standard_Copula_Sel	<i>Selecting best fitting standard (elliptical and Archimeadean) copula</i>
---------------------	---

Description

Fits five n-dimensional standard copula to a dataset and returns their corresponding AIC values.

Usage

```
Standard_Copula_Sel(Data)
```

Arguments

Data	Data frame containing n at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned NA. First object may be a "Date" object. Can be Dataframe_Combine output.
------	--

Value

Data frame containing copula name in column 1 and associated AIC in column 2. Parameters are estimated using the `fitCopula()` function in `copula` package using maximum pseudo-likelihood estimator "mpl". See [fitCopula](#) for a more thorough explanation.

See Also

[Standard_Copula_Fit](#)

Examples

```
Standard_Copula_Sel(Data_Detrend=S22.Detrend.df)
```

Standard_Copula_Sim	<i>Archimedean/elliptic copula model - Simulation</i>
---------------------	---

Description

Simulating from a fitted Archimedean or elliptic copula Model. Builds on the `in texmex` package.

Usage

```
Standard_Copula_Sim(Data, Marginals, Copula, mu = 365.25, N = 10000)
```

Arguments

Marginals	An <code>migpd</code> object containing the n -independent generalized Pareto models.
Copula	An Archimedean or elliptic copula model. Can be specified as an <code>Standard_Copula_Fit</code> object.
mu	(average) Number of events per year. Numeric vector of length one. Default is 365.25, daily data.
N	Number of years worth of extremes to be simulated. Numeric vector of length one. Default 10,000 (years).

Value

Each n -dimensional realisation is given on the transformed $[0, 1]^n$ scale (first n columns) in the first dataframe `u.Sim` and on the original scale in the second dataframe `x.Sim`.

See Also

[HT04_Fit](#)

Examples

```
Standard_Copula_Sim(Data=S22.Detrend.df, Marginals=S22.GPD, Copula=S22.Gaussian, mu=365.25, N=10000)
```

Vine_Copula_Fit	<i>C and D-vine Copula - Fitting</i>
-----------------	--------------------------------------

Description

Fit either a C- or D-vine copula model. Function is a repackaging of the `CDVineCopSelect` function in the `CDVine` package.

Usage

```
Vine_Copula_Fit(Data, FamilySet = NA, Type = "DVine",
  SelCrit = "AIC", Indeptest = FALSE, Level = 0.05)
```


Arguments

Data	Dataframe containing n at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.
FamilySet	Integer vector which must 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. See help file of the CDVineSim function to find out copula represented by integers 0-40.
Type	Type of the vine model: <ul style="list-style-type: none">• 1 or "CVine" = C-vine• 2 or "DVine" = D-vine
SelCrit	Character vector specifying the criterion for choosing among the competing pair-copula. Possible choices: "AIC" (default) or "BIC".
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

List comprising the pair-copula families composing the C- or D-vine copula Family, its parameters Par and Par2 as well as whether it is a C or D-vine Type.

See Also

[Detrend_Declustered_Combine](#) [CDVineCopSelect](#) [BiCopSelect](#)

Examples

```
S22.Vine<-Vine_Copula_Fit(Data=S22.Detrend.df, FamilySet=NA, Type="DVine", SelCrit="AIC",Indeptest=FALSE, Le
```

Vine_Copula_Sim	<i>C and D-vine Copula - Simulation</i>
-----------------	---

Description

Simulating from specified C- and D-vine copula models. Builds on the CDVineSim in CDVine.

Usage

```
Vine_Copula_Sim(Data, Marginals, Vine_family, Vine_par, Vine_par2,  
  Vine_Type = "DVine", mu = 365.25, N = 10000)
```

Arguments

Data	Dataframe containing n at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.
Marginals	An migpd object containing the d-independent generalized Pareto models.
Vine_family	A $n*(n-1)/2$ integer vector specifying the pair-copula families defining the fitting C- or a D-vine copula models. Can be specified as the Family argument of a Vine_Copula_Fit object. See help file of the CDVineSim function to find out copula represented by integers 0-40.
Vine_par	A $n*(n-1)/2$ vector of pair-copula parameters.
Vine_par2	A $n*(n-1)/2$ vector of second parameters for pair-copula families with two parameters.
Vine_Type	Type of the vine model: <ul style="list-style-type: none"> • 1 or "CVine" = C-vine • 2 or "DVine" = D-vine Can be specified as the Type argument of a Vine_Copula_Fit object.
mu	(average) Number of events per year. Numeric vector of length one. Default is 365.25, daily data.
N	Number of years worth of extremes to be simulated. Numeric vector of length one. Default 10,000 (years).

Value

List comprising an integer vector specifying the pair-copula families composing the C- or D-vine copula Vine_family, its parameters Vine_par and Vine_par2 and type of regular vine \odeVine_Type. In addition, dataframes of the simulated observations: u.Sim on the transformed $[0,1]^n$ and x.Sim the original scales.

See Also

Detrend_Declustered_Combine CD_Vine_Select migpd.edit

Examples

```
S22.Vine.Sim<-Vine_Copula_Sim(Data=S22.Detrend.df,Marginals=S22.GPD,Vine_family=S22.Vine$Family, Vine_par=S22.Vine$Par,Vine_par2=S22.Vine$Par2,Vine_Type=S22.Vine$Type, mu=mu,N=N)

S22.Pairs.Plot.Data<-data.frame(rbind(na.omit(S22.Detrend.df[, -1]),S22.Vine.Sim$x.Sim),c(rep("Observation",nrow(S22.Detrend.df)-1),rep("Sim",1)))
colnames(S22.Pairs.Plot.Data)<-c(names(S22.Detrend.df)[-1],"Type")
pairs(S22.Pairs.Plot.Data[,1:3],col=ifelse(S22.Pairs.Plot.Data$Type=="Observation","Black","Red"),upper.panel=FALSE)
```

Index

BiCopSelect, [15](#), [17](#)

CDVineCopSelect, [15](#), [17](#)

Con_Sampling_2D, [2](#)

Copula_Threshold_2D, [3](#)

Dataframe_Combine, [4](#), [13](#), [15](#)

Decluster, [5](#), [13](#)

Detrend, [4](#), [5](#), [13](#)

Detrend_Combine, [11](#)

Detrend_Declustered_Combine, [11](#), [17](#)

Diag_Non_Con, [6](#)

Diag_Non_Con_Sel, [7](#)

Diag_Non_Con_Sel_Trunc, [8](#)

Diag_Non_Con_Trunc, [8](#)

fitCopula, [15](#)

ggplot.migpd, [13](#)

GPD_Fit, [9](#)

HT04, [10](#)

HT04_Fit, [16](#)

Imputation, [11](#)

Kendall_Lag, [12](#)

Migpd_Fit, [13](#)

SLR_Scenarios, [14](#)

Standard_Copula_Fit, [14](#), [15](#)

Standard_Copula_Sel, [15](#)

Standard_Copula_Sim, [16](#)

Vine_Copula_Fit, [16](#)

Vine_Copula_Sim, [17](#)