Package ‘MultiHazard’

January 28, 2023

Title Tools for modeling compound events

Version 1.12

Description The `MultiHazard` package provides tools for stationary multivariate statistical modeling, for example, to estimate the joint distribution of MULTIple co-occurring HAZARDs. The package contains functions for pre-processing data including imputing missing values, detrending and declustering time series as well as analyzing pairwise correlations over a range of lags. Functionality is also built in to impliment the conditional sampling - copula theory approach in Jane et al. (2020) including the automated threshold selection approach in Solari et al. (2017). Tools are provided for selecting the best fitting amongst an array of (non-extreme, truncated and non-truncated) parametric marginal distributions, and, copulas to model the dependence structure. The package contains a function that calculates joint probability contours using the method of overlaying (conditional) contours given in Bender et al. (2016), and extracting design events such as the 'most likely' event or an ensemble of possible design events. The package also provides the capability of fitting and simulating synthetic records from three higher dimensional approaches - standard (elliptic/Archimedean) copulas, Pair Copula Constructions (PCCs) and the conditional threshold exceedance approach of Heffernan and Tawn (2004). Finally, a function that calculates the time for a user-specified height of sea level rise to occur under various scenarios is supplied.

License GPL (>= 3)

URL https://rdrr.io/github/rjaneUCF/MultiHazard-R-Package/

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports copula,
dplyr,
fitdistrplus,
GeneralizedHyperbolic,
ks,
lubridate,
MASS,
RColorBrewer,

rmatio,
statmod,
texmex,
truncnorm,
tweedie,
VGAM,
VineCopula

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Depends R (>= 2.10)

LazyData true

R topics documented:

`Annual_Max` *Generate annual maximum series*

Description

Extract annual maximum in years with over a user-defined proportion of non-missing values.

Usage

```
Annual_Max(Data_Detrend, Complete_Prop = 0.8)
```

Arguments

`Complete_Prop` Minimum proportion of non-missing values in an annual record for the annual maximum to be extracted. Default is `0.8`.

`Data` Data frame containing two columns. In column:

- 1 A `"Date"` object of equally spaced discrete time steps.
- 2 Numeric vector containing corresponding time series values.

Value

List comprising the index of the annual maximum `Event` and the annual maximum values `AM`.

Examples

```
Annual_Max(Data=S20_T_MAX_Daily_Completed_Detrend$Detrend)
```

`Conditional_RP_2D` *Calculates joint and conditional return periods*

Description

Univariate return period events are obtained from the GPDs to be consistent with the isolines produced by the `Design_Event_2D` function. To find the conditional probabilities a large number of realizations are simulated from the copulas fit to the conditioned samples, in proportion with the sizes of the conditional samples. The realizations are transformed to the original scale and the relevant probabilities estimated empirically.

Usage

```
Conditional_RP_2D(
  Data,
  Data_Con1,
  Data_Con2,
  Thres1,
  Thres2,
  Copula_Family1,
  Copula_Family2,
  Marginal_Dist1,
  Marginal_Dist2,
  Con1 = "Rainfall",
  Con2 = "OsWL",
  mu = 365.25,
  Con_Var,
  RP_Con,
  RP_Non_Con,
  Var1,
  Var2,
  x_lab = "Rainfall (mm)",
  y_lab = "O-sWL (mNGVD 29)",
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = NA,
  y_lim_max = NA,
  N
)
```

Arguments

`Data`	Data frame of dimension `nx2` containing two co-occurring time series of length `n`.
`Data_Con1`	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the first column.
`Data_Con2`	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the second column. Can be obtained using the `Con_Sampling_2D` function.
`Thres1`	Numeric vector of length one specifying the threshold above which the variable in the first column was sampled in Data_Con1.
`Thres2`	Numeric vector of length one specifying the threshold above which the variable in the second column was sampled in Data_Con2.
`Copula_Family1`	Numeric vector of length one specifying the copula family used to model the `Data_Con1` dataset.
`Copula_Family2`	Numeric vector of length one specifying the copula family used to model the `Data_Con2` dataset. Best fitting of 40 copulas can be found using the `Copula_Threshold_2D` function.
`Marginal_Dist1`	Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in `Data_Con1`.

Marginal_Dist2	Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in Data_Con2.
Con1	Character vector of length one specifying the name of variable in the first column of Data.
Con2	Character vector of length one specifying the name of variable in the second column of Data.
mu	Numeric vector of length one specifying the (average) occurrence frequency of events in Data. Default is 365.25, daily data.
Con_Var	Character vector of length one specifying the (column) name of the conditioning variable.
RP_Con	Numeric vector of length one specifying the return period of the conditioning variable Con_Var.
RP_Non_Con	Numeric vector of length one specifying the return period of the non-conditioning variable.
x_lab	Character vector specifying the x-axis label.
y_lab	Character vector specifying the y-axis label.
x_lim_min	Numeric vector of length one specifying x-axis minimum. Default is NA.
x_lim_max	Numeric vector of length one specifying x-axis maximum. Default is NA.
y_lim_min	Numeric vector of length one specifying y-axis minimum. Default is NA.
y_lim_max	Numeric vector of length one specifying y-axis maximum. Default is NA.
N	Numeric vector of length one specifying the size of the sample from the fitted joint distributions used to estimate the density along an isoline. Samples are collected from the two joint distribution with proportions consistent with the total number of extreme events conditioned on each variable. Default is 10^6

Value

Console output:

- Con_Var Name of the conditioning variable
- RP_Var1 Return period of variable Con1 i.e., variable in second column of Data
- RP_Var2 Return period of variable Con2 i.e., variable in third column of Data
- Var1 Value of Con1 at the return period of interest i.e. RP_Var1
- Var2 Value of Con2 at the return period of interest i.e. RP_Var2
- RP_Full_Dependence Joint return period of the (Var1,Var2) event under full dependence
- RP_Independence Joint return period of the (Var1,Var2) event under independence
- RP_Copula Joint return period of the (Var1,Var2) event according to the two sided conditional sampling - copula theory approach
- Prob Probability associated with RP_Copula
- N_Excess Number of realizations of the Con_Var above RP_Con-year return period value
- Non_Con_Var_X Values of the non-conditioned variable of the (conditional) Cummulative Distribution Function (CDF) i.e. x-axis of bottom left plot
- Con_Prob Con_Prob CDF of the non-conditioned variable given the return period of Con_Var exceeds RP_Con
- Con_Prob_Est Probability the non-conditioned variable is less than or equal to RP_Non_Con given the return period of Con_Var exceeds RP_Con

Graphical output:

- Top left: Sample conditioned on Con1 (red crosses) and Con2 (blue circles). Black dot is the event with a marginal return period of the conditioned variable `Var_Con` and non-conditioned variable equal to `RP_Con` and `RP_Non_Con`, respectively. The joint return period of the event using the conditional sampling - copula theory approach and under the assumptions of full dependence and independence between the variables are printed.
- Top right: Sample conditioned on Con1 (red crosses) and Con2 (blue circles). Only the region where Con_Var exceeds RP_Con is visible. This is the region for which the conditional distribution (of the non-conditioned variable given Con_Var exceeds RP_Con) and in turn conditional return periods are calculated.
- Bottom left: Conditional Cumulative Distribution Function (CDF) of the non-conditioned variable given the marginal return period of the conditioned variable `Var_Con` exceeds `RP_Con` years i.e. the points visible in the top right plot.
- Bottom right: Conditional return period of the non-conditioned variable given the conditioned variable `Var_Con` has a return period longer than `RP_Con`.

See Also

`Design_Event_2D`

Examples

```
#Under a 10yr (or greater) rainfall event condition, what is the joint probability that a 10yr
#O-sWLevent occurs simultaneously?  What is the cumulative probability of events with the
#frequency equal to or less than a 10yr O-sWL event?
Conditional_RP_2D(Data=S22.Detrend.df,
                  Data_Con1=con.sample.Rainfall$Data, Data_Con2=con.sample.OsWL$Data,
                  Thres1=0.98, Thres2=0.98,
                  Copula_Family1=cop.Rainfall,Copula_Family2=cop.OsWL,
                  Marginal_Dist1="Logis", Marginal_Dist2="BS",
                  Con1 = "Rainfall", Con2 = "OsWL",
                  mu = 365.25,
                  Con_Var="Rainfall",
                  RP_Con=10, RP_Non_Con=10,
                  x_lab = "Rainfall (Inches)", y_lab = "O-sWL (ft NGVD 29)",
                  y_lim_max = 10,
                  N=10^7)
```

`Conditional_RP_2D_Equal` *Calculates joint and conditional return periods*

Description

A large number of realizations are simulated from the copulas fit to the conditioned samples, in proportion with the sizes of the conditional samples. The realization are transformed to the original scale and the relevant probabilities estimated empirically. The conditional probabilities return period of the conditioning variable equals

Usage

```
Conditional_RP_2D_Equal(
  Data,
  Data_Con1,
  Data_Con2,
  Thres1,
  Thres2,
  Copula_Family1,
  Copula_Family2,
  Marginal_Dist1,
  Marginal_Dist2,
  Con1 = "Rainfall",
  Con2 = "OsWL",
  mu = 365.25,
  Con_Var,
  RP_Con,
  RP_Non_Con,
  Width = 0.1,
  x_lab = "Rainfall (mm)",
  y_lab = "O-sWL (mNGVD 29)",
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = NA,
  y_lim_max = NA,
  N
)
```

Arguments

Data	Data frame of dimension nx2 containing two co-occurring time series of length n.
Data_Con1	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the first column.
Data_Con2	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the second column. Can be obtained using the Con_Sampling_2D function.
Thres1	Numeric vector of length one specifying the threshold above which the variable in the first column was sampled in Data_Con1.
Thres2	Numeric vector of length one specifying the threshold above which the variable in the second column was sampled in Data_Con2.
Copula_Family1	Numeric vector of length one specifying the copula family used to model the Data_Con1 dataset.
Copula_Family2	Numeric vector of length one specifying the copula family used to model the Data_Con2 dataset. Best fitting of 40 copulas can be found using the Copula_Threshold_2D function.
Marginal_Dist1	Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in Data_Con1.
Marginal_Dist2	Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in Data_Con2.

Con1	Character vector of length one specifying the name of variable in the first column of Data.
Con2	Character vector of length one specifying the name of variable in the second column of Data.
mu	Numeric vector of length one specifying the (average) occurrence frequency of events in Data. Default is 365.25, daily data.
Con_Var	Character vector of length one specifying the (column) name of the conditioning variable.
RP_Con	Numeric vector of length one specifying the return period of the conditioning variable Con_Var.
RP_Non_Con	Numeric vector of length one specifying the return period of the non-conditioning variable.
Width	Numeric vector of length one specifying the distance above and below the RP_Con event of Con_Var the simulated events are used to estimate the conditional probability.
x_lab	Character vector specifying the x-axis label.
y_lab	Character vector specifying the y-axis label.
x_lim_min	Numeric vector of length one specifying x-axis minimum. Default is NA.
x_lim_max	Numeric vector of length one specifying x-axis maximum. Default is NA.
y_lim_min	Numeric vector of length one specifying y-axis minimum. Default is NA.
y_lim_max	Numeric vector of length one specifying y-axis maximum. Default is NA.
N	Numeric vector of length one specifying the size of the sample from the fitted joint distributions used to estimate the density along an isoline. Samples are collected from the two joint distribution with proportions consistent with the total number of extreme events conditioned on each variable. Default is 10^6

Value

Console output:

- Con_Var Name of the conditioning variable
- RP_Var1 Return period of variable Con1 i.e., variable in second column of Data
- RP_Var2 Return period of variable Con2 i.e., variable in third column of Data
- Var1 Value of Con1 at the return period of interest
- Var2 Value of Con2 at the return period of interest
- RP_Full_Dependence Joint return period of the (Var1,Var2) event under full dependence
- RP_Independence Joint return period of the (Var1,Var2) event under independence
- RP_Copula Joint return period of the (Var1,Var2) event according to the two sided conditional sampling - copula theory approach
- Prob Probability associated with RP_Copula
- N_Sub_Sample Number of realizations of the Con_Var within +/- width of the value of Con_Var with retunr period .
- Non_Con_Var_X Values of the non-conditioned variable of the (conditional) Cummulative Distribution Function (CDF) i.e. x-axis of bottom left plot
- Con_Prob Con_Prob CDF of the non-conditioned variable given the return period of Con_Var equals RP_Con

- Con_Prob_Est Probability the non-conditioned variable is less than or equal to `RP_Non_Con` given the return period of `Con_Var` equals `RP_Con`

Graphical output:

- Top Left: Sample conditioned on rainfall (red crosses) and O-sWL (blue circles). Black dot is the event with a marginal return period of the conditioned variable `Var_Con` and non-conditioned variable equal to `RP_Con` and `RP_Non_Con`, respectively. The joint return period of the event using the conditional sampling - copula theory approach and under the assumptions of full dependence and independence between the variables are printed.
- Top Right: Sample used to estimate the joint return period of the event of interest. Black dots denote the `N_Excess` sized subset of the sample where the marginal return period of the conditioned variable `Var_Con` exceeds `RP_Con` (years). The subset is used to estimate the conditional probabilities in part two of the question.
- Bottom Left: Conditional Cumulative Distribution Function (CDF) of the non-conditioned variable given the marginal return period of the conditioned variable `Var_Con` exceeds `RP_Con` years i.e. the black dots in the top right plot.
- Bottom Right: Conditional return period of the non-conditioned variable given the conditioned variable `Var_Con` has a return period longer than `RP_Con`.

See Also

`Design_Event_2D` `Conditional_RP_2D`

Examples

```
#Under a 10yr rainfall event condition, what is the joint probability that a 10yr surge (O-sWL)
#event occurs simultaneously? What is the cumulative probability of events with the frequency
#equal to or less than a 10yr surge event?
Conditional_RP_2D_Equal(Data=S22.Detrend.df,
                        Data_Con1=con.sample.Rainfall$Data, Data_Con2=con.sample.OsWL$Data,
                           Thres1=0.98, Thres2=0.98,
                           Copula_Family1=cop.Rainfall,Copula_Family2=cop.OsWL,
                           Marginal_Dist1="Logis", Marginal_Dist2="Twe",
                           Con1 = "Rainfall", Con2 = "OsWL",
                           mu = 365.25,
                           Con_Var="Rainfall",
                           RP_Con=10, RP_Non_Con=10,
                           x_lab = "Rainfall (Inches)", y_lab = "O-sWL (ft NGVD 29)",
                           y_lim_max = 10,
                           N=10^8)
```

`Con_Sampling_2D` *Conditionally sampling a two-dimensional dataset*

Description

Creates a data frame where the declustered excesses of a (conditioning) variable are paired with co-occurences of another variable.

Usage

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

Arguments

Data_Detrend	Data frame containing two at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned NA. First column may be a "Date" object. Can be Dataframe_Combine output.
Data_Declust	Data frame 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 column 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.
u	Threshold, as a quantile of the observations of the conditioning variable. Default is 0.97.
Thres	Threshold expressed on the original scale of the observations. Only one of u and Thres should be supplied. Default is NA.

Value

List comprising the specified Threshold as the quantile of the conditioning variable above which declustered excesses are paired with co-occurences of the other variable, the resulting two-dimensional sample data and name of the conditioning variable. The index of the input dataset that correspond to the events in the conditonal sample x.con are also provided.

Examples

```
S20.Rainfall<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                              Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                              Con_Variable="Rainfall",u=0.97)
```

Con_Sampling_2D_Lag *Conditionally sampling a two dimensional dataset*

Description

Creates a data frame where the declustered excesses of a (conditioning) variable are paired with the maximum value of a second variable over a specified lag.

Usage

```
Con_Sampling_2D_Lag(
  Data_Detrend,
  Data_Declust,
  Con_Variable,
  u = 0.97,
  Thres,
  Lag_Backward = 0,
  Lag_Forward = 0
)
```

Arguments

- `Data_Detrend` Data frame 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` Data frame 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`.
- `u` Threshold, as a quantile of the observations of the conditioning variable. Default is `0.97`.
- `Thres` Threshold expressed on the original scale of the observations. Only one of `u` and `Thres` should be supplied. Default is `NA`.
- `Lag_Backward` Positieve lag applied to variable not assigned as the `Con_Variable`. Default is `0`
- `Lag_Forward` Negative lag to variable not assigned as the `Con_Variable`. Default is `0`

Value

List comprising the specified `Threshold` as the quantile of the conditioning variable above which declustered excesses are paired with co-occurences of the other variable, the resulting two-dimensional sample `data` and `name` of the conditioning variable. The index of the input dataset that correspond to the events of the conditioning variable `x.con` and the non-conditioning variable `x.noncon` in the conditonal sample are also provided.

Examples

```
S20.Rainfall<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                              Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                              Con_Variable="Rainfall",u=0.97)
```

Cooley19 *Derives bivariate isolines using the non-parametric approach of Cooley et al. (2019).*

Description

The Cooley et al. (2019) method exploits bivariate regular variation and kernel density estimation to generate isolines of bivariate exceedance probabilities. The function utilizes the `ks` and `texmex` packages, and works for both asymptotic dependence and independence.

Usage

```
Cooley19(
  Data,
  Migpd,
  p.base = 0.01,
  p.proj = 0.001,
  u = 0.95,
```

```
  PLOT = FALSE,
  x_lim_min_T = NA,
  x_lim_max_T = NA,
  y_lim_min_T = NA,
  y_lim_max_T = NA,
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = NA,
  y_lim_max = NA
)
```

Arguments

Data	Data frame consisting of two columns.
Migpd	An Migpd object, containing the generalized Pareto models fitted (independently) to the variables comprising the columns of Data.
p.base	Numeric vector of length one specifying the exceedance probability of the base isoline. Default is 0.01.
p.proj	Numeric vector of length one specifying the exceedance probability of the projected isoline. Default is 0.001.
u	Numeric vector of length one specifying the quantile at which to estimate the asymptotic nature of the data i.e. chi and chibar. Default is 0.95.
PLOT	Logical; indicating whether to plot the base and projected isolines on the original and transformed scale. Default is FALSE.
x_lim_min_T	Numeric vector of length one specifying the lower x-axis limit of the transformed scale plot. Default is NA.
x_lim_max_T	Numeric vector of length one specifying the upper x-axis limit of the transformed scale plot. Default is NA.
y_lim_min_T	Numeric vector of length one specifying the lower y-axis limit of the transformed scale plot. Default is NA.
y_lim_max_T	Numeric vector of length one specifying the upper y-axis limit of the transformed scale plot. Default is NA.
x_lim_min	Numeric vector of length one specifying the lower x-axis limit of the plot on the original scale. Default is NA.
x_lim_max	Numeric vector of length one specifying the upper x-axis limit of the plot on the original scale. Default is NA.
y_lim_min	Numeric vector of length one specifying the lower y-axis limit of the plot on the original scale. Default is NA.
y_lim_max	Numeric vector of length one specifying the lower y-axis limit of the plot on the original scale. Default is NA.

Value

List comprising a description of the type of (asymptoptic) dependence Asym, the values the extremal dependence measures Chi and n.bar, exceedance probabilities of the base p.base and projected p.proj isolines, as well as the points on the base I.base and projected I.proj isolines.

See Also

Dataframe_Combine Decluster GPD_Fit Migpd_Fit

Examples

```
S20.GPD<-Migpd_Fit(Data=S20.Detrend.Declustered.df[,-1], mqu =c(0.99,0.99,0.99))
Cooley19(Data=na.omit(S20.Detrend.df[,3:4]),Migpd=s.Migpd,
p.base=0.01,p.proj=0.001,PLOT=TRUE,x_lim_max_T=500,y_lim_max_T=500)
```

Copula_Threshold_2D *Copula Selection With threshold 2D - Fit*

Description

Declustered excesses of a (conditioning) variable are paired with co-occurences 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,
  u1 = seq(0.9, 0.99, 0.01),
  u2 = seq(0.9, 0.99, 0.01),
  PLOT = TRUE,
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = -1,
  y_lim_max = 1,
  Upper = NA,
  Lower = NA,
  GAP = 0.05,
  Legend = TRUE
)
```

Arguments

Data_Detrend	Data frame containing two at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned NA.
Data_Declust	Data frame containing two (independently) declustered at least partially concurrent time series. Time steps must be equally spaced, with missing values assigned NA.
u1	A single or sequence of thresholds, given as a quantile of the observations of the variable in the first column of Data_Detrend when it is used as the conditioning variable. Default, sequence from 0.9 to 0.99 at intervals of 0.01.
u2	A single or sequence of thresholds, given as a quantile of the observations of the variable in the second column of Data_Detrend when it is used as the conditioning variable. Default, sequence from 0.9 to 0.99 at intervals of 0.01.
PLOT	Logical; whether to plot the results. Default is "TRUE".
x_lim_min	Numeric vector of length one specifying x-axis minimum. Default is NA.
x_lim_max	Numeric vector of length one specifying x-axis maximum. Default is NA.

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 u1 argument for which the copula family name label to appear above the corresponding point on the Kendall's tau coefficient vs threshold plot, when conditioning on the variable in column 1. Default is 0.
Lower	Numeric vector specifying the element number of the u2 argument for which the copula family name label to appear below the corresponding point on the Kendall's tau coefficient vs threshold plot, when conditioning 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.
Legend	Logic vector of length one specifying whether a legend should be plotted. Default is TRUE.

Value

List comprising:

- Kendalls_Tau1 Kendall's tau of a sample
- p_value_Var1 p-value when testing the null hypothesis H_0: tau=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 vectors Kendalls_Tau2,p_value_Var2, N_Var2 and Copula_Family_Var2 for the specified thresholds when the dataset is conditioned on the variable in column 2.

See Also

Dataframe_Combine

Examples

```
Copula_Threshold_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                    Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                    y_lim_min=-0.075, y_lim_max =0.25,
                    Upper=c(6,8), Lower=c(6,8),GAP=0.1)
```

Copula_Threshold_2D_Lag

Copula Selection With threshold 2D - Fit

Description

Declustered excesses of a (conditioning) variable are paired with co-occurences 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_Lag(
  Data_Detrend,
  Data_Declust,
  u1 = seq(0.9, 0.99, 0.01),
  u2 = seq(0.9, 0.99, 0.01),
  PLOT = TRUE,
  Lag_Backward_Var1 = 1,
  Lag_Forward_Var1 = 1,
  Lag_Backward_Var2 = 1,
  Lag_Forward_Var2 = 1,
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = -1,
  y_lim_max = 1,
  Upper = NA,
  Lower = NA,
  GAP = 0.05,
  Legend = TRUE
)
```

Arguments

`Data_Detrend` Data frame containing two at least partially concurrent time series, detrended if necessary. Time steps must be equally spaced, with missing values assigned `NA`.

`Data_Declust` Data frame containing two (independently) declustered at least partially concurrent time series. Time steps must be equally spaced, with missing values assigned `NA`.

`u1` A single or sequence of thresholds, given as a quantile of the observations of the variable in the first column of `Data_Detrend` when it is used as the conditioning variable. Default, sequence from `0.9` to `0.99` at intervals of `0.01`.

`u2` A single or sequence of thresholds, given as a quantile of the observations of the variable in the second column of `Data_Detrend` when it is used as the conditioning variable. Default, sequence from `0.9` to `0.99` at intervals of `0.01`.

`PLOT` Logical; whether to plot the results. Default is `"TRUE"`.

`Lag_Backward_Var1` Numeric vector of length one specifying the negative lag applied to variable in the first column of `Data_Detrend`. Default `1`.

`Lag_Forward_Var1` Numeric vector of length one specifying positive lag applied to variable in the first column of `Data_Detrend`. Default `1`.

`Lag_Backward_Var2` Numeric vector of length one specifying negative lag applied to variable in the second column of `Data_Detrend`. Default `1`.

`Lag_Forward_Var2` Numeric vector of length one specifying positive lag applied to variable in the second column of `Data_Detrend`. Default `1`.

`x_lim_min` Numeric vector of length one specifying x-axis minimum. Default is `NA`.

`x_lim_max` Numeric vector of length one specifying x-axis maximum. Default is `NA`.

`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 u1 argument for which the copula family name label to appear above the corresponding point on the Kendall's tau coefficient vs uhold plot, when conditioning on the variable in column 1. Default is NA.

Lower — Numeric vector specifying the element number of the u2 argument for which the copula family name label to appear below the corresponding point on the Kendall's tau coefficient vs uhold plot, when conditioning on the variable in column 2. Default is NA.

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 uhold plot. Default is 0.05.

Legend — Logic vector of length one specifying whether a legend should be plotted. Default is TRUE.

Value

List comprising:

- Kendalls_Tau1 Kendall's tau of a sample
- p_value_Var1 p-value when testing the null hypothesis H_0=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_Tau2, p_value_Var2, N_Var2 and Copula_Family_Var2 for the specified thresholds when the dataset is conditioned on the variable in column 2.

See Also

Dataframe_Combine

Examples

```
Copula_uhold_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                      Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                      y_lim_min=-0.075, y_lim_max =0.25,
                      Upper=c(6,8), Lower=c(6,8),GAP=0.1)
```

C_Sample — *Implements bootstrap procedure in Serinaldi and Kilsby (2013)*

Description

Implements the conditional bootstrap procedure outlined in Serinaldi and Kilsby (2013) to generate non-peak rainfall totals for a simulated peak. The function also calculates hyetograph properties including net characteristics.

Usage

```
C_Sample(Data, Cluster_Max, D, Start, End, Xp)
```

Arguments

Data	Vector of the rainfall time series.
Cluster_Max	Vector of the index of Data containing the cluster maximum. If declustering is carried out using Decluster_SW() set equal to $EventsMax output.
D	Numeric vector of the duration of the cluster maximum events.
Start	Numeric vector of the index of Data where each cluster maximum event begins.
End	Numeric vector of the index of Data where each cluster maximum event ends.
Xp	Numeric vector of simulated peaks. To implement the method exactly as in Serinaldi and Kilsby (2013), set equal to a sample (taken with replacement) of the observed cluster maximum (peaks).

Value

A data frame with the following columns:

- Xp Simulated event peaks i.e. input Xp.
- D Duration sampled from the duration vector D for each simulated event.
- Samp Index of the cluster maximum event, sampled conditionally on D, that provides non-peak rainfall depths.
- V Volume of simulated events.
- Vn Net volume of simulated events.
- I Intensity of simulated events.
- In Net intensity of simulated events.
- Start Index of Data where the sampled (Samp) event begins.
- End Index of Data where the sampled (Samp) event ends.

See Also

Decluster Time_Series_Plot WL_Curve

Examples

```
Set very small rainfall measurements to zero.
#Assumed to be the result of uncertainty in measuring equipment.
S13_Rainfall$Rainfall[which(S13_Rainfall$Rainfall<0.01)] = 0
#Find NAs in rainfall series
z = which(is.na(S13_Rainfall$Rainfall)==T)
#Temporarily set NAs to zero
S13_Rainfall$Rainfall[z] = 0
#Find times where there is 6-hours of no rainfall
no.rain = rep(NA,length(S13_Rainfall$Rainfall))
for(i in 6:length(S13_Rainfall$Rainfall)){
  no.rain[i] = ifelse(sum(S13_Rainfall$Rainfall[(i-5):i])==0,i,NA)
}
#Remove NAs from results vector as these correspond to times where there is
#rainfall at certain points in the 6 hour period.
```

```
no.rain = na.omit(no.rain)
#Reset missing values in the rainfall record back to NA
S13_Rainfall$Rainfall[z] = NA
#Find the start and end times of the 500 events.
start = rep(NA,length(S13.Rainfall.Declust$EventsMax))
end = rep(NA,length(S13.Rainfall.Declust$EventsMax))
for(i in 1:length(S13.Rainfall.Declust$EventsMax)){
 start[i] = max(no.rain[which(no.rain<S13.Rainfall.Declust$EventsMax[i])])
 end[i] = min(no.rain[which(no.rain>S13.Rainfall.Declust$EventsMax[i])])
}
start = start + 1
end = end - 6
d = end - start + 1 #Duration
#Simulate some peaks by sampling observed peaks with replacement
#I.e., applying the method exactly as in Serinaldi and Kilsby (2013)
sim.peak = sample(S13.Rainfall.Declust$EventsMax,size=500,replace=TRUE)
#Derive the hyetographs
S13.oswl.sample = C_Sample(Data=S13_Rainfall$Rainfall,
                           Cluster_Max=S13.Rainfall.Declust$EventsMax,
                           D=d,Start=start,End=end,
                           Xp=sim.peak)
```

`Dataframe_Combine` *Creates a data frame containing up to five time series*

Description

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

Usage

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

Arguments

`n`	Integer `1-5` specifying the number of time series. Default is `3`.
`data.1:5`	Data frames with two columns containing in column • 1 - Continuous sequence of times spanning from the first to the final recorded observations. • 2 - Corresponding values detrended where necessary.

Value

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

See Also

Detrend

Examples

```
#Formatting data
S20.Rainfall.df<-Perrine_df
S20.Rainfall.df$Date<-as.Date(S20.Rainfall.df$Date)
S20.OsWL.df<-S20_T_MAX_Daily_Completed_Detrend_Declustered[,c(2,4)]
S20.OsWL.df$Date<-as.Date(S20.OsWL.df$Date)
#Detrending O-sWL series at Site S20
S20.OsWL.Detrend<-Detrend(Data=S20.OsWL.df,Method = "window",PLOT=FALSE,
                          x_lab="Date",y_lab="O-sWL (ft NGVD 29)")
#Creating a dataframe with the date alongside the detrended OsWL series
S20.OsWL.Detrend.df<-data.frame(as.Date(S20.OsWL.df$Date),S20.OsWL.Detrend)
colnames(S20.OsWL.Detrend.df)<-c("Date","OsWL")
#Combining the two datasets by Date argument
S20.Detrend.df<-Dataframe_Combine(data.1<-S20.Rainfall.df,
                                  data.2<-S20.OsWL.Detrend.df,
                                  data.3=0,
                                  names=c("Rainfall","OsWL"))
```

Decluster *Declusters a time series*

Description

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

Usage

```
Decluster(Data, u = 0.95, Thres = NA, 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.
Thres	Threshold expressed on the original scale of the observations. Only one of u and Thres should be supplied. Default is NA.
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) occurrence frequency of events in Data. Numeric vector of length one. Default is 365.25, daily data.

Value

List comprising the Threshold above which cluster maxima are identified, rate of cluster maxima Rate, a vector containing the original time series Detrended and the Declustered series.

See Also

Detrend

Examples

```
Decluster(data=S20_T_MAX_Daily_Completed_Detrend$Detrend)
```

Decluster_SW *Declusters a time series using a storm window approach*

Description

Find peaks with a moving window. The code is based on the `IDEVENT` function provided by Sebastian Solari.

Usage

```
Decluster_SW(Data, Window_Width)
```

Arguments

`Data`	Data frame containing two columns. In column: • 1 A `"Date"` object of equally spaced discrete time steps. • 2 Numeric vector containing corresponding time series values.
`Window_Width`	Numeric vector of length one specifying the width, in days, of the window used to ensure events are independent.

Value

List comprising vectors containing the original time series `Detrended`, independent (declustered) events `Declustered` and the elements of the original series containing the declustered events `EventID`.

Examples

```
#Declustering the O-sWL at site S22 using a 3-day window.
v<-Decluster_SW(Data=S22.Detrend.df[,c(1:2)],Window_Width=7)
plot(as.Date(S22.Detrend.df$Date),S22.Detrend.df$Rainfall,pch=16)
points(as.Date(S22.Detrend.df$Date)[v$EventID],v$Event,col=2,pch=16)
```

Decluster_S_SW *Declusters a Summed time series using a moving (Storm) Window approach*

Description

Finds the sum of a time series within a moving window then declusters the summed series using another moving window.

Usage

```
Decluster_S_SW(Data, Window_Width_Sum, Window_Width)
```

Arguments

Data — Data frame containing two columns. In column:

- 1 A "Date" object of equally spaced discrete time steps.
- 2 Numeric vector containing corresponding time series values.

Window_Width_Sum — Numeric vector of length one specifying the window width over which to sum the data.

Window_Width — Numeric vector of length one specifying the width, in days, of the window used to ensure events are independent.

Value

List comprising vectors containing the original time series Detrend, the summed series Totals, independent (declustered) events Declustered, the elements of the original series containing the start (Event_Start), center EventID, and end (Event_End) of the declustered events. Note for Window_Width_Sum_Type="End", Event_End and EventID are identical.

Examples

```
#Declustering 24 hour rainfall totals at site S13 using a 7-day window for declustering the events.
plot(S13_Precip$Date,S13_Rainfall$Rainfall)
S13_Rainfall_Totals_Declust<-Decluster_S_SW(Data=S13_Rainfall, Window_Width_Sum=24,
                                            Window_Width=7*24)
plot(S13_Rainfall[,1],
     S13_Rainfall_Totals_Declust$Totals,
     pch=16,ylim=c(0,10))
points(S13_Rainfall[S13_Rainfall_Totals_Declust$EventID,1],
       S13_Rainfall_Totals_Declust$Totals[S13_Rainfall_Totals_Declust$EventID],
       col=2,pch=16)
```

Design_Event_2D — *Derives a single or ensemble of bivariate design events*

Description

Calculates the isoline and relative probability of events on the isoline, given the observational data, for one or more user-specified return periods. Outputs the single "most-likely" design event or an ensemble of possible design events obtained by sampling along the isoline according to these relative probabilities. The design event under the assumption of full dependence is also computed.

Usage

```
Design_Event_2D(
  Data,
  Data_Con1,
  Data_Con2,
  u1,
  u2,
  Thres1 = NA,
  Thres2 = NA,
  Copula_Family1,
```

```
  Copula_Family2,
  Marginal_Dist1,
  Marginal_Dist2,
  Marginal_Dist1_Par = NA,
  Marginal_Dist2_Par = NA,
  Con1 = "Rainfall",
  Con2 = "OsWL",
  GPD1 = NA,
  GPD2 = NA,
  mu = 365.25,
  GPD_Bayes = FALSE,
  Decimal_Place = 2,
  RP,
  Interval = 10000,
  End = F,
  Resolution = "Low",
  x_lab = "Rainfall (mm)",
  y_lab = "O-sWL (mNGVD 29)",
  x_lim_min = NA,
  x_lim_max = NA,
  y_lim_min = NA,
  y_lim_max = NA,
  Isoline_Probs = "Sample",
  N = 10^6,
  N_Ensemble = 0,
  Sim_Max = 10,
  Plot_Quantile_Isoline = FALSE,
  Isoline_Type = "Combined"
)
```

Arguments

Data	Data frame of dimension nx2 containing two co-occurring time series of length n.
Data_Con1	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the first column.
Data_Con2	Data frame containing the conditional sample (declustered excesses paired with concurrent values of other variable), conditioned on the variable in the second column. Can be obtained using the Con_Sampling_2D function.
u1	Numeric vector of length one specifying the threshold, expressed as a quantile, above which the variable in the first column was sampled in Data_Con1.
u2	Numeric vector of length one specifying the threshold, expressed as a quantile, above which the variable in the second column was sampled in Data_Con2.
Thres1	Numeric vector of length one specifying the threshold above which the variable in the first column was sampled in Data_Con1. Only one of u1 and Thres1 should be supplied. Default is NA.
Thres2	Numeric vector of length one specifying the threshold above which the variable in the second column was sampled in Data_Con2. Only one of u2 and Thres2 should be supplied. Default is NA.

Copula_Family1 Numeric vector of length one specifying the copula family used to model the Data_Con1 dataset.

Copula_Family2 Numeric vector of length one specifying the copula family used to model the Data_Con2 dataset. Best fitting of 40 copulas can be found using the Copula_Threshold_2D function.

Marginal_Dist1 Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in Data_Con1.

Marginal_Dist2 Character vector of length one specifying (non-extreme) distribution used to model the marginal distribution of the non-conditioned variable in Data_Con2.

Con1 Character vector of length one specifying the name of variable in the first column of Data.

Con2 Character vector of length one specifying the name of variable in the second column of Data.

GPD1 Output of GPD_Fit applied to variable con1 i.e., GPD fit con1. Default NA. Only one of u1, Thres1 and GPD1 is required.

GPD2 Output of GPD_Fit applied to variable con2 i.e., GPD fit con2. Default NA. Only one of u2, Thres2 and GPD2 is required.

mu Numeric vector of length one specifying the (average) occurrence frequency of events in Data. Default is 365.25, daily data.

GPD_Bayes Logical; indicating whether to use a Bayesian approach to estimate GPD parameters. This involves applying a penalty to the likelihood to aid in the stability of the optimization procedure. Default is FALSE.

RP Numeric vector specifying the return periods of interest.

Interval Numeric vector specifying the number of equally spaced points comprising the combined isoline.

x_lab Character vector specifying the x-axis label.

y_lab Character vector specifying the y-axis label.

x_lim_min Numeric vector of length one specifying x-axis minimum. Default is NA.

x_lim_max Numeric vector of length one specifying x-axis maximum. Default is NA.

y_lim_min Numeric vector of length one specifying y-axis minimum. Default is NA.

y_lim_max Numeric vector of length one specifying y-axis maximum. Default is NA.

N Numeric vector of length one specifying the size of the sample from the fitted joint distributions used to estimate the density along an isoline. Samples are collected from the two joint distribution with proportions consistent with the total number of extreme events conditioned on each variable. Default is 10^6

N_Ensemble Numeric vector of length one specifying the number of possible design events sampled along the isoline of interest.

Sim_Max Numeric vector of length one specifying the maximum value, given as a multiple of the largest observation of each variable, permitted in the sample used to estimate the (relative) probabilities along the isoline.

Plot_Quantile_Isoline
Logical; indicating whether to first plot the quantile isoline. Default is FALSE.

Isoline_Type Character vector of length one specifying the type of isoline. For isolines obtained using the overlaying method in Bender et al. (2016) use "Combined" (default). For quantile isoline from the sample conditioned on variable Con1|(Con2) use "Con1"("Con2").

Decimal_Palace Numeric vector specifying the number of decimal places to which to specifiy the isoline. Defulat is 2.

Value

Plot of all the observations (grey circles) as well as the declustered excesses above Thres1 (blue circles) or Thres2 (blue circles), observations may belong to both conditional samples. Also shown is the isoline associated with `RP` contoured according to their relative probability of occurrence on the basis of the sample from the two joint distributions, the "most likely" design event (black diamond), and design event under the assumption of full dependence (black triangle) are also shown in the plot. The function also returns a list comprising the design events assuming full dependence `"FullDependence"`, as well as once the dependence between the variables is accounted for the "Most likley" `"MostLikelyEvent"` as well as an `"Ensemble"` of possible design events and relative probabilities of events on the isoline `Contour`. The quantile isolines with `Quantile_Isoline_1` and `Quantile_Isoline_2`, and GPD thresholds with `Threshold_1` and `Threshold_2`.

See Also

Copula_Threshold_2D Diag_Non_Con Diag_Non_Con_Trunc

Examples

```
S22.Rainfall<-Con_Sampling_2D(Data_Detrend=S22.Detrend.df[,-c(1,4)],
                             Data_Declust=S22.Detrend.Declustered.df[,-c(1,4)],
                             Con_Variable="Rainfall",u=0.97)
S22.OsWL<-Con_Sampling_2D(Data_Detrend=S22.Detrend.df[,-c(1,4)],
                         Data_Declust=S22.Detrend.Declustered.df[,-c(1,4)],
                         Con_Variable="OsWL",u=0.97)
S22.Copula.Rainfall<-Copula_Threshold_2D(Data_Detrend=S22.Detrend.df[,-c(1,4)],
                             Data_Declust=S22.Detrend.Declustered.df[,-c(1,4)],u1 =0.97,
                                        y_lim_min=-0.075,y_lim_max=0.25,
                                 Upper=c(2,9),Lower=c(2,10),GAP=0.15)$Copula_Family_Var1
S22.Copula.OsWL<-Copula_Threshold_2D(Data_Detrend=S22.Detrend.df[,-c(1,4)],
                             Data_Declust=S22.Detrend.Declustered.df[,-c(1,4)],u2 =0.97,
                                    y_lim_min=-0.075, y_lim_max =0.25,
                                 Upper=c(2,9),Lower=c(2,10),GAP=0.15)$Copula_Family_Var2
Design.Event<-Design_Event_2D(Data=S22.Detrend.df[,-c(1,4)],
                             Data_Con1=S22.Rainfall$Data, Data_Con2=S22.OsWL$Data,
                             u1=0.97, u2=0.97,
                     Copula_Family1=S22.Copula.Rainfall, Copula_Family2=S22.Copula.OsWL,
                             Marginal_Dist1="Logis", Marginal_Dist2="Twe",
                             RP=c(5,100),Interval=10000,N=10^6,N_Ensemble=10,
                             Plot_Quantile_Isoline=FALSE)
#Extracting the 100-year isoline from the output
Design.Event$`100`$Isoline
```

`Detrend` *Detrends a time series.*

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 = "Date",
  y_lab = "Data"
)
```

Arguments

Data	Data frame containing two columns. In column: • 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.

Examples

```
#Detrending ocean-side water level at site S22 using a 3 month moving average window and the last
#five years of observations to calculate the present day average.
Detrend(S22_T_MAX_Daily_Completed_Detrend,Method = "window",Window_Width= 89,
       End_Length = 1826, PLOT=FALSE,x_lab="Data",y_lab="Data")
```

Diag_Non_Con *Goodness of fit of non-extreme marginal distributions*

Description

Fits two (unbounded) 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

Dataframe `$AIC` giving the AIC associated with each distribution and the name of the best fitting distribution `$Best_fit`. Panel consisting of three plots. Upper plot: Plot depicting the AIC of the two 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.

See Also

Copula_Threshold_2D

Examples

```
S20.Rainfall<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                              Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                              Con_Variable="Rainfall",Thres=0.97)
Diag_Non_Con(Data=S20.Rainfall$Data$OsWL,x_lab="O-sWL (ft NGVD 29)",
             y_lim_min=0,y_lim_max=1.5)
```

Diag_Non_Con_Sel — *Demonstrate the goodness of fit of the selected non-extreme marginal distribution*

Description

Plots demonstrating the goodness of fit of a selected (not truncated) 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 Gaussian "Gaus" and logistic "Logis".

Value

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

See Also

Diag_Non_Con

Examples

```
S20.Rainfall<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                              Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                              Con_Variable="Rainfall",Thres=0.97)
Diag_Non_Con(Data=S20.Rainfall$Data$OsWL,x_lab="O-sWL (ft NGVD 29)",
             y_lim_min=0,y_lim_max=1.5)
Diag_Non_Con_Sel(Data=S20.Rainfall$Data$OsWL,x_lab="O-sWL (ft NGVD 29)",
                 y_lim_min=0,y_lim_max=1.5,Selected="Twe")
```

Diag_Non_Con_Trunc *Goodness of fit of non-extreme marginal distributions*

Description

Fits ten (truncated) non-extreme marginal distributions to a dataset and returns three plots demonstrating their relative goodness of fit. The distributions are the Birnbaum-Saunders "BS", exponential "Exp", two-parameter gamma "Gam(2)", three-parameter gamma "Gam(3)", mixed two-parameter gamma "GamMix(2)", mixed three-parameter gamma "GamMix(3)", lognormal "LNorm", truncated normal "TNorm", Tweedie "Twe" and the Weibull "Weib".

Usage

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

Arguments

Data	Numeric vector containing realizations of the variable of interest.
Omit	Character vector specifying any distributions that are not to be tested. Default "NA", all distributions are fit.
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

Dataframe $AIC giving the AIC associated with each distribution and the name of the best fitting distribution $Best_fit. Panel consisting of three plots. Upper plot: Plot depicting the AIC of the ten 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.

See Also

Copula_Threshold_2D

Examples

```
S20.OsWL<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                          Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                          Con_Variable="OsWL",Thres=0.97)
Diag_Non_Con_Trunc(Data=S20.OsWL$Data$Rainfall,x_lab="Rainfall (Inches)",
                   y_lim_min=0,y_lim_max=2)
```

Diag_Non_Con_Trunc_Sel

Godness of fit of the selected non-extreme marginal distribution

---

Description

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

Usage

```
Diag_Non_Con_Trunc_Sel(
  Data,
  Selected,
  Omit = NA,
  x_lab = "Data",
  y_lim_min = 0,
  y_lim_max = 1
)
```

Arguments

Data	Numeric vector containing realizations of the variable of interest.
Selected	Character vector of length one specifying the chosen distribution, options are the Birnbaum-Saunders "BS", exponential "Exp", two-parameter gamma "Gam(2)", three-parameter gamma "Gam(3)", mixed two-parameter gamma "GamMix(2)", mixed three-parameter gamma "GamMix(3)", lognormal "LogN", Tweedie "Twe" and Weibull "Weib".
Omit	Character vector specifying any distributions that are not to be tested. Default "NA", all distributions are fit.
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	Numericr 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 eight 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.

See Also

Diag_Non_Con_Trunc

Examples

```
S20.OsWL<-Con_Sampling_2D(Data_Detrend=S20.Detrend.df[,-c(1,4)],
                          Data_Declust=S20.Detrend.Declustered.df[,-c(1,4)],
                          Con_Variable="OsWL",Thres=0.97)
Diag_Non_Con_Trunc(Data=S20.OsWL$Data$Rainfall,x_lab="Rainfall (Inches)",
                   y_lim_min=0,y_lim_max=2)
Diag_Non_Con_Sel_Trunc(Data=S20.OsWL$Data$Rainfall,x_lab="Rainfall (Inches)",
                       y_lim_min=0,y_lim_max=2,Selected="Twe")
```

GPD_Fit *Fits a single generalized Pareto distribution - Fit*

Description

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

Usage

```
GPD_Fit(
  Data,
  Data_Full,
  u = 0.95,
  Thres = NA,
  mu = 365.25,
  GPD_Bayes = TRUE,
  Method = "Standard",
  min.RI = 1,
  PLOT = FALSE,
  xlab_hist = "Data",
  y_lab = "Data"
)
```

Arguments

Data	Numeric vector containing the declusted data.
Data_Full	Numeric vector containing the non-declustered data.
u	GPD threshold expressed as a quantile [0,1] of Data vector. Default is 0.95.
Thres	GPD threshold expressed on the original scale of the "Data". Only one of u and Thres should be supplied. Default is NA.
mu	Numeric vector of length one specifying (average) occurrence frequency of events in the Data_Full input. Default is 365.25.
GPD_Bayes	Logical; indicating whether to use a Bayesian approach to estimate GPD parameters. This involves applying a penalty to the likelihood to aid in the stability of the optimization procedure. Default is TRUE.
Method	Character vector of length one specifying the method of choosing the threshold. "Standard" (default) chooses the exact threshold specified aswither "u" or "th", whereas "Solari" selects the minimum exceedence of the "Data" above the user-specified threshold.
min.RI	Numeric vector of length one specifying the minimum return period in the return level plot. Default is 1.
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 xi and scale parameters sigma along with their standard errors sigma.SE and xi.SE.

Details

For excesses of a variable X over a suitably high threshold u the fitted GPD model is parameterized as follows:

$$P(X > x|X > u) = \left[1 + \xi \frac{(x-u)}{\sigma}\right]_{+}^{-\frac{1}{\xi}}$$

where ξ and $\sigma > 0$ are the shape and scale parameters of the GPD and $[y]_{+} = max(y, 0)$.

Examples

```
Decluster(Data=S20_T_MAX_Daily_Completed_Detrend$Detrend)
```

GPD_Parameter_Stability_Plot

GPD parameter stability plots

Description

Plots showing the stability of the GPD scale and shape parameter estimates across a specified range of thresholds.

Usage

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

Arguments

`Data`	Numeric vector containing the declusted data.
`Data_Full`	Numeric vector containing the non-declustered data.
`u`	Numeric vector of GPD thresholds; given as a quantiles `[0,1]` of `Data` vector. Default is `0.9` to `0.999` in intervals of `0.001`.
`Plot`	Logical; indicating whether to plot diagnostics. Default is `FALSE`.

Value

Plot of the shape and modified scale parameter estimates along with their errors bars over the range of specified thresholds.

See Also

`Decluster`

Examples

```
GPD_Parameter_Stability_Plot(Data = S20.Detrend.Declustered.df$Rainfall,
                             Data_Full= na.omit(S20.Detrend.df$Rainfall),
                             u=seq(0.9,0.999,0.001))
```

GPD_Threshold_Solari *Solari et al (2017) automatic GPD threshold selection*

Description

Automatic threshold selection method in Solari et al. (2017) is implemented to find the threshold above which excesses are follow a GPD. The code is based on the `ANALISIS_POT_LNORM` function provided by Sebastian Solari.

Usage

```
GPD_Threshold_Solari(
  Event,
  Data,
  RPs = c(10, 50, 100, 500, 1000),
  RPs_PLOT = c(2, 3, 4),
  Min_Quantile = 0.95,
  Alpha = 0.1,
  mu = 365.25,
  N_Sim = 10
)
```

Arguments

`Event`	Numeric vector containing the declustered events.
`Data`	Original time series. Dataframe containing two columns. In column: • 1 A `"Date"` object of equally spaced discrete time steps. • 2 Numeric vector containing corresponding time series values.
`RPs`	Numeric vector specifying the return levels calculated from the GPD fits over the thresholds. Default is `c(50,100,500,100)` plus the return period associated with the minimum candidate threshold.
`RPs_PLOT`	Numeric vector of length three specifying which elements of `RPs` are plotted in the middle row of the graphical output. Default is `c(1,2,3)`.
`Min_Quantile`	Numeric vector of length one specifying the minimum threshold, expressed as a quantile of the original time series (2nd column of `Data`) to be tested. Default `0.95`.
`Alpha`	Numeric vector of length one specifying the level of confidence associated with the confidence interval i.e., the probability that the interval contains the true value of the parameter is $1-\frac{Alpha}{2}$. The interval is referred to as the $100(1-\frac{Alpha}{2})\%$ confidence interval. Default is `0.1`.
`mu`	(average) occurrence frequency of events in the original time series `Data`. Numeric vector of length one. Default is `365.25`, daily data.
`N_Sim`	Numeric vector of length one specifying the number of bootstrap samples. Default is `10`.

Value

List comprising

- Thres_Candidate Thresholds tested which are the cluster maxima in Events exceeding the Min_Quantile quantile of the original time series (given in column 2 of Data).
- GPD_MLE GPD parameter estimates, Mean Residual Life Plot (MRLP) values and return level estimates associated with each Thres_Candidate.
- CI_Upper Upper limits of the confidence interval for the point estimates of the corresponding element of GPD_MLE.
- CI_Lower Lower limits of the confidence interval for the point estimates of the corresponding element of GPD_MLE.
- AR2 Value of the right-tail weighted Anderson Darling statistic A^2_R, the test statistic used in the Solari et al. (2017) method for each Thres_Candidate.
- AR2_pValue p-value asssociated with A^2_R.

To interpret the graphical output. Top row: The GPD exhibits certain threshold stability properties. The guiding principle for threshold choice is to find the lowest value of the threshold such that the parameter estimates stabilize to a constant value which is sustained at all higher thresholds, once the sample uncertainty has been accounted for (typically assessed by pointwise uncertainty intervals). Mean residual life plot (left). If the GPD is a valid model for excesses above a threshold then the mean of these excesses will be a linear function of the threshold. We therefore select the lowest threshold where there is a linear trend in the mean residual life plot. Parameter stability plots for the shape (center) and scale (right) parameters. If the GPD is a suitable model for a threshold then for all higher thresholds it will also be suitable, with the shape and scale parameters being constant. The lowest threshold - to reduce the associated uncertainty - at which the parameter estimates are stable for all higher thresholds should be selected. Middle row: Return levels estimated from the GPD fitted at various thresholds. Lower row: Right-tail weighted Anderson Darling statistic A^2_R associated with the GPD fitted using various thresholds. Lower A^2_R statistic values signify less (quadratic) distance between the empirical distribution and the GPD i.e., GPD is a better fit for these thresholds (left). $1 - p_{value}$ associated with the A^2_R for each threshold. The A^2_R goodness of fit tests, tests the null hypothesis that the observations are from a GPD. At smaller $1 - p_{value}$ figure there is less chance of rejecting the null hypothesis i.e., the GPD is more suitable at these thresholds (center). Events per year at each threshold (right).

Details

EDF-statistics are goodness-of-fit statistics based on a comparison of the Empirical Distribution Function (EDF) F_n and a candidate parametric probability distribution F Stephens et al. (1974). Quadratic EDF test measure the distance between F and F_n by:

$$n \int_{-\infty}^{\infty} = (F(x) - F_n(x))^2 w(x) dx$$

where n is the number of elements in the original sample and $w(x)$ is a weighting function. In the Cramer Von Misses statistic $w(x) = 1$, whereas the Anderson-Darling statistic A^2, assigns more weight to the tails of the data by setting $w(x) = \frac{1}{F(x)(1-F(x))}$. Under the null hypothesis that the sample $x_1, \ldots, x_n$ is from a GPD, the transformation $z = F_1(x)$ a sample z uniformly distribution between 0 and 1.

$$A^2 = -\frac{1}{n} \sum_{i=1}^{n} \{(2i-1)[log(z_i) + log(1 - z_{n+z-i})]\} - n$$

Sinclair et al. (1990) proposed the right-tail weighted Anderson Darling statistic A_R^2 which allocates more weight to the upper tail and less to the lower tail of the distribution than A^2 and is given by:

$$A_R^2 = \frac{n}{2} \sum_{i=1}^{n} \left[2 - \frac{(2i-1)}{n} log(1 - z_i) + 2z_i \right]$$

Solari et al. (2017) formalized EDF statistic - GOF test threshold selection procedures used to test the null hypothesis that a sample is from a GPD distribution. creating an automated approach adopting the A_R^2 as the EDF statistic. The authors also proposed combining the approach with a bootstrapping technique to assess the influence of threshold on the uncertainly of higher return period quantiles. The approach in Solari et al. (2017) comprises the following steps:

1. Decluster the time series to produce a series of n_p independent cluster maxima $\{x_i : i = 1, \ldots, n_p\}$ and sort such that $\{x_1 \leq \ldots \leq x_p\}$.
2. The sorted series defines a series of n_u thresholds after excluding repeated values i.e., $n_u \leq n_p$. For each threshold $\{u_j, j = 1, \ldots, n_u\}$ fit the GPD via L-Moments using only the excesses satisfying $x > u_j$. Then, calculate the R-AD statistic and its associated p-value for each threshold.
3. Select the threshold that minimizes one minus the p-value i.e.,

$$u_0 = argmin_{u_j}(1 - p(u_j)).$$

Examples

```
#Declustering the rainfall at site S22 using a 7-day window.
Rainfall_Declust_SW<-Decluster_SW(Data=S22.Detrend.df[,c(1:2)],Window_Width=7)
#Finding an appropriate threshold for the declustered series
GPD_Threshold_Solari(Event=Rainfall_Declust_SW$Declustered,
                     Data=22.Detrend.df[,2])
```

GPD_Threshold_Solari_Sel

Goodness-of-fit for the GPD

Description

A nonparametric bootstrapping procedure is undertaken to assess the uncertainty in the GPD parameters and associated return levels for a GPD fit to observations above a user specified threshold. The estimates are compared with those obtained at other thresholds by running the `GPD_Threshold_Solari` function beforehand, and using its output as an input of this function. The code is based on the `AUTOMATICO_MLE_BOOT` function provided by Sebastian Solari.

Usage

```
GPD_Threshold_Solari_Sel(
  Event,
  Data,
  Solari_Output,
  Thres,
  Alpha = 0.1,
  N_Sim = 10^4,
```

```
  RP_Min = 1,
  RP_Max = 1000,
  RP_Plot = 100,
  mu = 365.25,
  y_lab = "Data"
)
```

Arguments

Event — Numeric vector containing independent events declustered using a moving window approach.

Data — Original time series. Dataframe containing two columns. In column:

- 1 A "Date" object of equally spaced discrete time steps.
- 2 Numeric vector containing corresponding time series values.

Solari_Output — Output of the GPD_Threshold_Solari function.

Thres — Numeric vector of length one specifying the threshold to analyze, chosen by the user based on plots from the GPD_Threshold_Solari function.

Alpha — Numeric vector of length one specifying the level of confidence associated with the confidence interval i.e., the probability that the interval contains the true value of the parameter is $1 - \frac{Alpha}{2}$. The interval is referred to as the $100(1 - \frac{Alpha}{2})\%$ confidence interval. Default is 0.1.

N_Sim — Numeric vector of length one specifying the number of bootstrap samples. Default is 10^4.

RP_Min — Numeric vector of length one specifying the minimum return level to be calculated. Default is 1.

RP_Max — Numeric vector of length one specifying the maximum return level to be calculated. Default is 1000.

RP_Plot — Numeric vector of length one specifying the return level in the lower right plot. Default is 100.

mu — (average) occurrence frequency of events in the original time series Data. Numeric vector of length one. Default is 365.25, daily data.

y_lab — Character vector specifying the y-axis label of the return level plot.

Value

List containing three objects: Estimate, CI_Upper and CI_Lower. The Estimate dataframe comprises

- xi GPD shape parameter estimate for the threshold is Thres.
- sigma GPD scale parameter estimate for the threshold is Thres.
- Thres GPD location parameter estimate for the threshold is Thres.
- rate GPD rate parameter i.e., number of independent excesses per year for a threshold of Thres.
- The remaining columns are RL Return level estimates from the GPD using a threshold of Thres.

CI_Upper and CI_Lower give the upper and lower bounds of the $100(1 - \frac{Alpha}{2})\%$ confidence interval for the corresponding element in Estimate. Top row: Histograms of the GPD parameter estimates based on a nonparametric bootstrapping simulation. Grey bars correspond to the estimates obtained as the threshold (Thres) is varied, found by running the function a necessary input of the function. Continuous black lines correspond to results obtained by fixing the threshold at Thres. Dashed blue lines correspond to the expected values for the fixed threshold. Lower left: Return level plot. Return levels of the observations estimated from the empirical distribution. Grey bars correspond to the maximum of the upper and lower bounds of the $100(1 - \frac{Alpha}{2})\%$ confidence intervals as the threshold is varied. Continuous black lines correspond to results obtained by fixing the threshold at Thres. Dashed blue lines correspond to the expected values for the fixed threshold. Lower right: As in the top row but for the 100 years return period quantile.

Examples

```
Rainfall_Declust_SW<-Decluster_SW(Data=S22.Detrend.df[,c(1:2)],Window_Width=7)
Finding an appropriate threshold for the declustered series
S22_OsWL_Solari<-GPD_Threshold_Solari(Event=Rainfall_Declust_SW$Declustered,
                                      Data=na.omit(S22.Detrend.df[,2]))
S22_OsWL_Solari_Sel<-GPD_Threshold_Solari_Sel(Event=Rainfall_Declust_SW$Declustered,
                                          Data=S22.Detrend.df[,2],
                                          Solari_Output=S22_OsWL_Solari,
                                          Thres=S22_OsWL_Solari$Candidate_Threshold)
```

HT04 *Fits and simulates from the conditional multivariate approach of Heffernan and Tawn (2004)*

Description

Fits the conditional multivariate approach of Heffernan and Tawn (2004) to a dataset and simulates realizations from the fitted model. Function utilizes the mexDependence and predict.mex.conditioned 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

data_Detrend_Dependence_df
: A data frame with (n+1) columns, containing in column

- 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.

data_Detrend_Declustered_df
: A data frame with (n+1) columns, containing in column
  - 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.

u_Dependence
: 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.

Migpd
: An `Migpd` object, containing the generalized Pareto models fitted (independently) to each of the variables.

Margins
: 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.

V
: See documentation for mexDependence.

Maxit
: See documentation for mexDependence.

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`, residuals `z`, as well as the simulated values on the transformed `u.sim` and original x.sim scales.

See Also

`Dataframe_Combine` `Migpd_Fit`

Examples

```
#Fitting and simulating from the Heffernan and Tawn (2004) model
S20.HT04<-HT04(data_Detrend_Dependence_df=S20.Detrend.df,
               data_Detrend_Declustered_df=S20.Detrend.Declustered.df,
               u_Dependence=0.995,Migpd=S20.Migpd,mu=365.25,N=1000)
#View model conditioning on rainfall
S20.HT04$Model$Rainfall
#Assigning simulations (transformed back to the original scale) a name
S20.HT04.Sim<-S20.HT04$x.sim
#Plotting observed (black) and simulated (red) values
S20.Pairs.Plot.Data<-data.frame(rbind(na.omit(S20.Detrend.df[,-1]),S20.HT04.Sim),
                                c(rep("Observation",nrow(na.omit(S20.Detrend.df))),
                                  rep("Simulation",nrow(S20.HT04.Sim))))
colnames(S20.Pairs.Plot.Data)<-c(names(S20.Detrend.df)[-1],"Type")
pairs(S20.Pairs.Plot.Data[,1:3],
```

```
col=ifelse(S20.Pairs.Plot.Data$Type=="Observation","Black","Red"),
upper.panel=NULL,pch=16)
```

`HT04_Lag` *Implements the version of the conditional multivariate approach of Heffernan and Tawn (2004) proposed in Keef et al. (2013) which incorporates lags between the variables.*

Description

Implements the version of the conditional multivariate approach of Heffernan and Tawn (2004) proposed in Keef et al. (2013) which incorporates lags between the variables. Function utilizes the `mexDependence` and `predict.mex.conditioned` functions from the `texmex` package.

Usage

```
HT04_Lag(
  data_Detrend_Dependence_df,
  data_Detrend_Declustered_df,
  Lags,
  u_Dependence,
  Migpd,
  mu = 365.25,
  N = 100,
  Margins = "gumbel",
  V = 10,
  Maxit = 10000
)
```

Arguments

`data_Detrend_Dependence_df`
: A data frame with (n+1) columns, containing in column
  - 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.

`data_Detrend_Declustered_df`
: A data frame with (n+1) columns, containing in column
  - 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.

`u_Dependence`
: 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.

`Migpd`
: An `Migpd` object, containing the parameterized Pareto models fitted (independently) to each of the variables.

Margins — 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.

V — See documentation for mexDependence.

Maxit — See documentation for mexDependence.

Lag — Matrix specifying the lags. The no lag i.e. 0 lag cases need to be specified. Row n denotes the lags applied to the variable in the nth column of data_Detrend_Dependence_df. Column n corresponds to the nth largest lag applied to any variable. NA. Default is matrix(c(0,1,0,NA),nrow=2,byrow = T), which corresponds to a lag of 1 being applied to variable in the first column of data_Detrend_Dependence_df and no lag being applied to the variable in the second column of data_Detrend_Dependence_df.

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, residuals z, as well as the simulated values on the transformed u.sim and original x.sim scales.

See Also

Dataframe_Combine Decluster GPD_Fit Migpd_Fit

Examples

```
HT04(data_Detrend_Dependence_df = S22.Detrend.df,
data_Detrend_Declustered_df = S22.Detrend.Declustered.df,
Migpd = S22_GPD, u_Dependence=0.7,Margins = "gumbel")
```

Imputation — *Imputing missing values through linear regression*

Description

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

Usage

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

Arguments

Data — Data frame 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 a

- `Data` data frame 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 parameters including its coefficient of determination

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
##Viewing first few rows of G_3356
head(G_3356)
#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))

##Viewing first few rows of G_3355
head(G_3355)
#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
library('dplyr')
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 29)",
           y_lab="Groundwater level (ft NGVD 29)")
```

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 up to three 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`.
`GAP`	Numeric vector of length one. Length of y-axis above and below max and min Kendall's tau values.
`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-value<0.05). Lag applied to variable named second in the legend.

See Also

`Dataframe_Combine`

Examples

```
Kendall_Lag(Data=S20.Detrend.df,GAP=0.1)
```

`Mean_Excess_Plot` *Mean excess plot - GPD threshold selection*

Description

The empirical mean excess function is linear in the case of a GPD.

Usage

```
Mean_Excess_Plot(Data)
```

Arguments

`data`	A vector comprising a declustered and if necessary detrended time series to be modelled.

Value

Plot of the empirical mean excess function (black line), average of all observations exceeding a threshold decreased by the threshold, for thresholds spanning the range of the observations. Also provided are `95%` confidence intervals (blue dotted lines) and the observations (black dots).

See Also

Decluster Detrend

Examples

```
Mean_Excess_Plot(Data=S20_Detrend_Declustered_df$Rainfall)
```

Migpd_Fit *Fits Multiple independent generalized Pareto models - Fit*

Description

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

Usage

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

Arguments

Data	A data frame with n columns, each comprising a declustered and if necessary detrended time series to be modelled.
Data_Full	A data frame with n columns, each comprising the original (detrended if necessary) time series to be modelled. Only required if threshold is specified using mqu.
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))
#Same GPDs fit as above but thresholds given on the original scale
S22.Rainfall.Quantile<-quantile(na.omit(S22.Detrend.Declustered.df$Rainfall),0.99)
S22.OsWL.Quantile<-quantile(na.omit(S22.Detrend.Declustered.df$OsWL),0.99)
S22.GW.Quantile<-quantile(na.omit(S22.Detrend.Declustered.df$Groundwater),0.99)
S22.GPD<-Migpd_Fit(Data=S22.Detrend.Declustered.df[,-1],
                   mqu =c(S22.Rainfall.Quantile,S22.OsWL.Quantile,S22.GW.Quantile))
```

NOAA_SLR *NOAA sea-level rise scenarios*

Description

Time (in years) for a specified amount of sea-level rise (SLR) to occur at Miami Beach according to the five SLR scenarios in NOAA 2017 report titled "Global and Regional Sea Level Rise Scenarios for the United States".

Usage

```
NOAA_SLR(
  OsWL_req,
  SLR_scen = c("High", "Intermediate", "Low"),
  Input_unit = "m",
  Year.Inital = 2020
)
```

Arguments

OsWL_req	Numeric vector of SLR required.
SLR_scen	Character vector specifying which of the NOAA (2017) scenarios to consider. Options include High, Intermediate high Int.High, Intermediate, Intermediate low (Int.Low) and Low.
Input_unit	Character vector of length one; specifying units of SLR. Default is meters "m", other option is feet "ft".
Year	Character vector of length one; specifying

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

```
NOAA_SLR<-function(OsWL_req=seq(0,1,0.01),SLR_scen = c("High","Intermediate","Low"),
                  Input_unit="m")
```

OsWL_Intensity *Ocean-side Water Level Intensity*

Description

Calculates the "intensity" of extreme water levels, as defined in Wahl et al. (2011).

Usage

```
OsWL_Intensity(
  Data,
  Cluster_Max,
  Base_Line = mean(Data$OsWL, na.rm = T),
  Rainfall_Interval = 24
)
```

Arguments

Data	A data frame with co-occurring rainfall and O-sWL time series in two columns labeled "Rainfall" and "OsWL", respectively.
Cluster_Max	Numeric vector containing indexes of peaks in the O-sWL column of Data. If analyzing a sample conditioned on O-sWL derived using Con_Sample_2D() set equal to the $xcon output.
Base_Line	Numeric vector of length one, specifying water level about which to calculate the intensity. Default is the mean O-sWL.
Rainfall_Interval	Numeric vector of length one, specifying length of time before and after a peak over which to sum rainfall totals. Total window width is 2*Rainfall_Interval+1. Default is 24.

Value

A data frame with the following columns:

- Pre.High Index of the OsWL column of Data containing the preceding high water level.
- Fol.High Index of the OsWL column of Data containing the following high water level.
- Pre.Low Index of the OsWL column of Data containing the preceding low water level.
- Fol.Low Index of the OsWL column of Data containing the following low water level.
- Intensity Intensity of the O-sWL.
- V Total rainfall volume within Rainfall_Interval before and after the peak.

See Also

Decluster WL_Curve

Examples

```
#Decluster O-sWL series at S-13 using a runs method
S13.OsWL.Declust = Decluster(Data=S13.Detrend.df$OsWL,
                             SepCrit=24*7, u=0.99667)
#Calculate O-sWL of the identified cluster maximum
intensity = OsWL_Intensity(Data=S13.Detrend.df,Cluster_Max=S13.OsWL.Declust$EventsMax)
#Plot O-sWL series identifying cluster maximum (in red) and print "intensity" above each maximum
plot(S13.Detrend.df$Date_Time,
     S13.Detrend.df$OsWL)
points(S13.Detrend.df$Date_Time[S13.OsWL.Declust$EventsMax],
       S13.Detrend.df$OsWL[S13.OsWL.Declust$EventsMax],pch=16,col=2)
text(S13.Detrend.df$Date_Time[S13.OsWL.Declust$EventsMax],
     S13.Detrend.df$OsWL[S13.OsWL.Declust$EventsMax]+0.2,
     round(intensity$Intensity,0),cex=0.5)
```

SLR_Scenarios *Sea level rise scenarios*

Description

Time (in years) for a specified change in sea level according to various sea level projections. Contained within the function are: (1) the three scenarios for Key West in the Southeast Florida Regional Climate Change Compact, (2) those for Miami Beach in "Global and Regional Sea Level Rise Scenarios for the United States" NOAA et al. (2017) and (3) those in the Interagency Sea Level Rise Scenario Tool (NOAA et al. 2022) for Naples and Miami Beach. Users can also input scenarios of their choice.

Usage

```
SLR_Scenarios(
  SeaLevelRise,
  Scenario = "Compact",
  Unit = "m",
  Year = 2022,
  Location = "Key West",
  New_Scenario = NA
)
```

Arguments

`SeaLevelRise`	Numeric vector of length one, specifying the sea level rise required.
`Scenario`	Character vector of length one, specifying the sea level rise scenarios to be adopted. Options are `"Compact"` for those for Key West in the Southeast Florida Regional Climate Change Compact, `"NOAA2017"` for those in "Global and Regional Sea Level Rise Scenarios for the United States" at Miami Beach used in Jane et al. (2020), `"NOAA2022"` for those for Miami Beach and Naples in the Interagency Sea Level Rise Scenario Tool, or `NA` if a set of scenarios are specified by the user (see `New_Scenario`).
`Unit`	Character vector of length one, specifying units of `SeaLevelRise`. Options are meters `m` and Inches `"Inches"`. Default is `"m"`.
`Year`	Numeric vector of length one, specifying the current year. Default is `2022`.

`Location` Character vector of length one, specifying the location associated with the scenarios. Projections for "Key West" (Compact), "Miami Beach" (NOAA2017 AND NOAA2022) and "Naples" (NOAA2022) are contained within the package. If a user specified scenarios are employed, set to the name of the site. Default is "Key West".

`New_Scenario` Dataframe containing sea level rise scenarios. First column must be a year and the scenarios provided in the remaining columns. For the color scale to correlate with the severity of the scenarios they should be listed from most to least severe i.e., the highest SLR scenario should appear in column 2. All entries must be numeric.

Value

For "Compact", "NOAA2017" and "NOAA2022" a list length of time for SeaLevelRise of sea level rise is expected to arise under the High, Intermediate and Low. For user specified scenarios, the time for SeaLevelRise to occur under each is returned as SLR_Year. Upper panel: A plot of the scenarios. Scenarios are in bold until the time the SeaLevelRise is reached and are transparent thereafter. Lower panel: A plot showing the number of years before is expected to occur.

Examples

```
#Calculate the estimated time required for 0.45m of SLR in Key West according to the scenarios
in the Southeast Florida Regional Climate Change Compact
SLRScenarios(0.45)
#Calculate the estimated time required for 0.8 inches of SLR in Naples according
to the scenarios in the 2022 Interagency Sea Level Rise Scenario Tool
SLRScenarios(0.45,Scenario="NOAA2022", Unit = "Inches", Location="Naples")
#Read in the scenarios for Fort Myers downloaded
from https://sealevel.nasa.gov/task-force-scenario-tool/?psmsl_id=1106
SeaLevelRise.2022<-read.csv("sl_taskforce_scenarios_psmsl_id_1106_Fort_Myers.csv")
#Convert data to the appropriate format for the SLRScenarios function
#i.e. first column years, following columns the scenarios most to least extreme,
converted from millimeters to meters
SeaLevelRise.2022_input<-data.frame(Year=seq(2020,2150,10),
                                    "High"=as.numeric(SeaLevelRise.2022[14,-(1:5)])/1000,
                                  "Medium"=as.numeric(SeaLevelRise.2022[8,-(1:5)])/1000,
                                     "Low"=as.numeric(SeaLevelRise.2022[2,-(1:5)])/1000)
#Calculate the estimated time required for 0.8 inches of SLR at Fort Myers
SLR_Scenarios(SeaLevelRise=0.8, Scenario="Other", Unit = "m", Year=2022,
              Location="Fort Myers", New_Scenario=SeaLevelRise.2022_input)
```

Standard_Copula_Fit *Fit an Archimedean/elliptic copula model - Fit*

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`	Data frame 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` `Standard_Copula_Sel`

Examples

```
cop<-Standard_Copula_Fit(Data=S20.Detrend.df,Copula_Type="Gaussian")
cop<-Standard_Copula_Fit(Data=S20.Detrend.df,Copula_Type="tcopula")
cop<-Standard_Copula_Fit(Data=S20.Detrend.df,Copula_Type="Gumbel")
cop<-Standard_Copula_Fit(Data=S20.Detrend.df,Copula_Type="Clayton")
cop<-Standard_Copula_Fit(Data=S20.Detrend.df,Copula_Type="Frank")
```

`Standard_Copula_Sel` *Selecting best fitting standard (elliptical and Archimedean) copula*

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

`Dataframe_Combine` `Standard_Copula_Fit`

Examples

```
Standard_Copula_Sel(Data_Detrend=S20.Detrend.df)
```

Standard_Copula_Sim *Archimedean/elliptic copula model - Simulation*

Description

Simulating from a fitted Archimedean or elliptic copula model.

Usage

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

Arguments

Data	Data frame 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 n-independent generalized Pareto models.
Copula	An Archimedean or elliptic copula model. Can be specified as an Standard_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

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

See Also

Standard_Copula_Sel Standard_Copula_Fit

Examples

```
#Fitting multiple independent GPDs to the data
#(required to transform realisation back to origional scale)
S20.Migpd<-Migpd_Fit(Data=S20.Detrend.Declustered.df[,-1],mqu=c(0.975,0.975,0.9676))
#Fitting Gaussian copula
Standard_Copula_Sim(Data=S20.Detrend.df,Marginals=S20.Migpd,Copula=S20.Gaussian,
                    mu=365.25,N=10000)
```

Surge_Criterion *Surge identification criterion*

Description

Classify extreme water levels as either tidally dominated or surge driven.

Usage

```
Surge_Criterion(
  Data,
  Cluster_Max,
  Criterion_Number = "NA",
  Surge_Thres = 0.25,
  Rainfall_Thres = NA,
  Pre_Sur = 7,
  MaxMin = "Max",
  Rainfall_Interval = NA
)
```

Arguments

`Data`	A data frame with co-occurring rainfall and O-sWL time series in two columns labeled `"Rainfall"` and `"OsWL"`, respectively.
`Cluster_Max`	Numeric vector containing indexes of peaks in the O-sWL column of `Data`. If analyzing a sample conditioned on O-sWL derived using `Con_Sample_2D()` set equal to the `$xcon` output.
`Criterion_Number`	Numeric vector of length one, specifying which of the five criterion detailed in the report to adopt. If a user-defined criterion is adopted set to `NA` which is the default.
`Surge_Thres`	Numeric vector of length one, specifying the minimum elevation difference between a peak and prior maximum or minimum for the peak to be classified as surge driven. Default is `0.25`.
`Rainfall_Thres`	Numeric vector of length one, specifying minimum rainfall within a +/- `Rainfall_Interval` period of a peak for the peak to be classified as surge driven. Default is `NA`.
`Pre_Sur`	Numeric vector of length one, specifying, minimum length of time allowed between preceding maximum or minimum and the peak. Default is `7`.
`MaxMin`	Character vector of length one, specifying whether elevation difference refers to the preceding minimum (`"Min"`) or maximum (`"Max"`). Default is `"Max"`.
`Rainfall_Interval`	Numeric vector of length one, specifying length of time before and after a peak over which to sum rainfall totals. Total window width is 2*`Rainfall_Interval`+1. Default is `NA`.

Value

A vector with each cluster maximum classified as either `Tide` or `Surge` driven.

See Also

Con_Sampling_2D

Examples

```
#Decluster O-sWL series at S-13 using a runs method
S13.OsWL.Declust = Decluster(Data=S13.Detrend.df$OsWL,
                             SepCrit=24*7, u=0.99667)
#Classify peak water levels as either surge or tidally driven
surge_class = Surge_Criterion(Data = S13.Detrend.df,
                             Cluster_Max = S13.OsWL.Declust$EventsMax,
                             Criterion_Number = 5)
#Plot O-sWL time series with peaks the color of peaks representing classification
plot(S13.Detrend.df$Date_Time,S13.Detrend.df$OsWL)
points(S13.Detrend.df$Date_Time[S13.OsWL.Declust$EventsMax],
      S13.Detrend.df$OsWL[S13.OsWL.Declust$EventsMax],
      col=ifelse(surge_class=="Tide","Blue","Red"),pch=16)
legend("topleft",c("Tide","Surge"),pch=16,col=c("Blue","Red"))
#Example of a custom surge criterion. Peak is classified as tidal if
#Elevation difference between peak and preceding minimum at least 7 hrs before is less than 0.25.
#Total rainfall from 72 hours before and to 72 hrs after the peak is less than 2 Inches
surge_class = Surge_Criterion(Data = S13.Detrend.df,
                              Cluster_Max = S13.OsWL.Declust$EventsMax,
                              Surge_Thres=2.5,Rainfall_Thres=2,Pre_Sur=7,
                              MaxMin="Min",Rainfall_Interval=72)
```

Time_Series_Plot *Rainfall and O-sWL time series plots*

Description

Plots a user specified number of synthetic events where at least O-sWL or rainfall peak exceeds a high threshold. .

Usage

```
Time_Series_Plot(
  Rainfall_Series,
  Oswl_Time_Series,
  Sample,
  Con_Variable,
  Buffer = 6,
  Intensity = NA,
  Event_ID = 1:16,
  Row = 4,
  Col = 4,
  Mar = c(4.2, 4.5, 1.5, 3.5)
)
```

Arguments

`Rainfall_Series`
: Data frame with rows comprising time series of rainfall totals associated with cluster maximum of the rainfall series.

`Oswl_Time_Series`
: Data frame with rows comprising the water level curves associated with the simulated events in `Sample`.

`Sample`
: Data frame containing the simulated events. Columns (and their names) required by the function are rainfall peak (Rainfall), O-sWL peak (OsWL), their lag time (Lag), and the ID of the sampled rainfall event (samp).

`Con_Variable`
: Character vector of length one specifying the conditioning variable of the events in `Sample`.

`Buffer`
: Numeric vector of length one specifying the extension of the x-axis before and after the rainfall event when `Con_Variable == "Rainfall"`. Default is `6`.

`Intensity`
: Numeric vector specifying the "intensity" of the O-sWL events in `Sample`. Default is `NA`.

`Event_ID`
: Numeric vector specifying the events in `Sample` to be plot.

`Row`
: Numeric vector of length one specifying the number of rows of subplots in the Figure.

`Col`
: Numeric vector of length one specifying the number of columns of subplots in the Figure. Product of `Row` and `Col` must be equal to or greater than `Event_ID`.

`Mar`
: Numeric vector of length one specifying the margin at the (bottom,left,top,right) of the subplots. Default is `c(4.2,4.5,1.5,3.5)`.

Value

Figure containing a (Row * Col) matrix of subplots each displaying the hyetogaph (grey bars) and water level curve (blue lines) comprising an event.

See Also

`C_Sample` `WL_Curve`

Examples

```
Time_Series_Plot(Rainfall_Series=x_n,Oswl_Time_Series=oswl.ts.rainfall,
                 Sample=S13.rainfall.sample,Con_Variable="Rainfall",Event_ID=1:16)
```

`Vine_Copula_Fit` *C and D-vine Copula - Fitting*

Description

Fit either a C- or D-vine copula model. Function is a repackaging the `RVineStructureSelect` and `RVineCopSelect` functions from the `RVine` package into a single function.

Usage

```
Vine_Copula_Fit(Data)
```

Arguments

Data	Data frame containing n at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.

Value

List comprising the vine copula Structure, pair-copula families composing the C- or D-vine copula Family, its parameters Par and Par2.

See Also

Dataframe_Combine Vine_Copula_Sim

Examples

```
S20.Vine<-Vine_Copula_Fit(Data=S20.Detrend.df)
```

Vine_Copula_Sim *C and D-vine Copula - Simulation*

Description

Simulating from specified C- and D-vine copula models. Function is a repackaging of the RVineMatrix and RVineMatrix functions from the VineCopula package into a single function.

Simulating from specified C- and D-vine copula models. Function is a repackaging of the RVineMatrix and RVineMatrix functions from the VineCopula package into a single function.

Usage

```
Vine_Copula_Sim(Data, Vine_Model, Marginals, mu = 365.25, N = 10000)

Vine_Copula_Sim(Data, Vine_Model, Marginals, mu = 365.25, N = 10000)
```

Arguments

Data	Data frame containing n at least partially concurrent time series. First column may be a "Date" object. Can be Dataframe_Combine output.
Vine_Model	An RVineMatrix object i.e., output of Vine_Copula_Fit specifying the structure and copula families composing the vine copula.
Marginals	An migpd object containing the d-independent generalized Pareto models.
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 `Vine_Type`. In addition, data frames of the simulated observations: `u.Sim` on the transformed `$[0,1]^n$` and `x.Sim` the original scales.

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 `Vine_Type`. In addition, data frames of the simulated observations: `u.Sim` on the transformed `$[0,1]^n$` and `x.Sim` the original scales.

See Also

`Vine_Copula_Fit`

`Vine_Copula_Fit`

Examples

```
#Fitting vine copula
S20.Vine<-Vine_Copula_Fit(Data=S20.Detrend.df)
#Simulating from fitted copula
S20.Vine.Sim<-Vine_Copula_Sim(Data=S20.Detrend.df,Vine_Model=S20.Vine,
                              Marginals=S20.Migpd,N=10)
#Plotting observed (black) and simulated (red) values
S20.Pairs.Plot.Data<-data.frame(rbind(na.omit(S20.Detrend.df[,-1]),S22.Vine.Sim$x.Sim),
                                c(rep("Observation",nrow(na.omit(S20.Detrend.df))),
                                rep("Simulation",nrow(S20.Vine.Sim$x.Sim))))
colnames(S20.Pairs.Plot.Data)<-c(names(S20.Detrend.df)[-1],"Type")
pairs(S20.Pairs.Plot.Data[,1:3],
      col=ifelse(S20.Pairs.Plot.Data$Type=="Observation","Black","Red"),
      upper.panel=NULL)
```


```
#Fitting vine copula
S20.Vine<-Vine_Copula_Fit(Data=S20.Detrend.df)
#Simulating from fitted copula
S20.Vine.Sim<-Vine_Copula_Sim(Data=S20.Detrend.df,Vine_Model=S20.Vine,
                              Marginals=S20.Migpd,N=10)
#Plotting observed (black) and simulated (red) values
S20.Pairs.Plot.Data<-data.frame(rbind(na.omit(S20.Detrend.df[,-1]),S22.Vine.Sim$x.Sim),
                                c(rep("Observation",nrow(na.omit(S20.Detrend.df))),
                                rep("Simulation",nrow(S20.Vine.Sim$x.Sim))))
colnames(S20.Pairs.Plot.Data)<-c(names(S20.Detrend.df)[-1],"Type")
pairs(S20.Pairs.Plot.Data[,1:3],
      col=ifelse(S20.Pairs.Plot.Data$Type=="Observation","Black","Red"),
      upper.panel=NULL)
```

`WL_Curve` *Derive water level curves*

Description

Generates water level curves for simulated extreme water levels based on a simulated "intensity".

Usage

```
WL_Curve(
  Data,
  Cluster_Max,
  Pre_Low,
  Fol_Low,
  Thres,
  Base_Line = mean(Data$OsWL, na.rm = T),
  Limit,
  Peak,
  Intensity,
  Length = 144
)
```

Arguments

Data	A data frame of the time series with the column containing ocean-side water levels labeled "OsWL".
Cluster_Max	Numeric vector containing indexes of peaks in the O-sWL column of Data. If analyzing a sample conditioned on O-sWL derived using Con_Sample_2D() set equal to the $xcon output.
Pre_Low	Numeric vector of the indexes of the O-sWL column in Data containing the preceding low water level.
Fol_Low	Numeric vector of the indexes of the O-sWL column in Data containing the following low water level.
Thres	Numeric vector of length one, specifying threshold above which to apply the method. Below the threshold an observed curve with an intensity less than limit is randomly sampled.
Base_Line	Numeric vector of length one, specifying water level about which to calculate the intensity. Default is the mean O-sWL.
Limit	Numeric vector of length one, specifying an upper limit on the observed water level curve intensities to sample for simulated peaks less than Thres.
Peak	Numeric vector of simulated peak water levels.
Intensity	Numeric vector of the intensity associated with each simulated Peak.
Length	Numeric vector of length one, specifying the length of time over which the water level curve is simulated before (and after) the time of the simulated peak. Total duration of the water level curve is 2*Length+1. Minimum is 5. Default is 144.

Value

A data frame, where each row contains the water level curve generated for corresponding simulated peak in the Peak input. A vector of the intensity Intensity of the generated water level curve.

See Also

Surge_Criterion OsWL_Intensity

Examples

```
#Declustering O-sWL series
S13.OsWL.Declust = Decluster(Data=S13.Detrend.df$OsWL,
                             SepCrit=24*7, u=0.99667)
#Use O-sWL intensity function to obtain index of preceding and following low water levels
intensity = OsWL_Intensity(Data=S13.Detrend.df,Cluster_Max=S13.OsWL.Declust$EventsMax)

#Four synthetic events
sim.peaks = c(3.4,4,4.2,5)
sim.intensity = c(38,48,120,1400)

#Generating the water level curves
oswl_ts_oswl = WL_Curve(Data = S13.Detrend.df,
                        Cluster_Max = S13.OsWL.Declust$EventsMax,
                        Pre_Low = intensity$Pre.Low,
                        Fol_Low = intensity$Fol.Low,
                        Thres = S13.OsWL.Declust$Threshold, Limit = 45,
                        Peak = sim.peaks,
                        Intensity = sim.intensity)

#Plot the water level curves of the observed peaks
plot(1:289,
     S13.Detrend.df$OsWL[(S13.OsWL.Declust$EventsMax[1]-144):
                         (S13.OsWL.Declust$EventsMax[1]+144)],
     type='l',ylim=c(1,5))
for(i in 2:length(S13.OsWL.Declust$EventsMax-144)){
  lines(1:289,
        S13.Detrend.df$OsWL[(S13.OsWL.Declust$EventsMax[i]-144):
                            (S13.OsWL.Declust$EventsMax[i]+144)])
}
#Superimpose the curves generated ro the four synthetic events
for(i in 1:4){
  lines(1:289,oswl_ts_oswl$Series[i,],col=2)
}
```

Index