**Package ‘pbox’**

May 27, 2024

**Type**  Package

**Title**  Exploring Multivariate Spaces with Probability Boxes

**Version**  0.1.8

**Maintainer**  Ahmed T. Hammad <ahmed.t.hammad@gmail.com>

**Description**  
Advanced statistical library offering a method to encapsulate and query the probability space of a dataset effortlessly using Probability Boxes (p-boxes). Its distinctive feature lies in the ease with which users can navigate and analyze marginal, joint, and conditional probabilities while taking into account the underlying correlation structure inherent in the data using copula theory and models. A comprehensive explanation is available in the paper “pbox: Exploring Multivariate Spaces with Probability Boxes” to be published in the Journal of Statistical Software.

**URL**  [https://www.r-project.org](https://www.r-project.org), [https://github.com/athammad/pbox](https://github.com/athammad/pbox)

**BugReports**  [https://github.com/athammad/pbox/issues](https://github.com/athammad/pbox/issues)

**Depends**  gamlss.dist, R (>= 2.10)

**Imports**  copula, data.table, gamlss, methods, purrr, stats, stringr, utils

**Suggests**  ggplot2, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

**VignetteBuilder**  knitr

**Config/testthat/edition**  3

**Encoding**  UTF-8

**LazyData**  true

**RoxygenNote**  7.3.1

**Collate**  'SEAx.R' 'pbox.R' 'coefAll2.R' 'copula_families.R'
'deltaCI.R' 'final_pbox.R' 'fit_copula.R' 'fit_copula_pbox.R'
'fit_dist_pbox.R' 'fun_stats.R' 'gen_scenario.R' 'grid_pbox.R'
'make_pbox.R' 'match_maker.R' 'modify_pbox.R' 'param_dev.R'
'perProb.R' 'perturbate_params.R' 'probCI.R' 'q_parser.R'
'qpbox.R' 'scenario_pbox.R' 'set_pbox.R' 'show.R'
'stats_calc.R'
License GPL-3
Language en-US
NeedsCompilation no
Author Ahmed T. Hammad [aut, cre, cph]
(https://orcid.org/0000-0003-3327-2435)
Repository CRAN
Date/Publication 2024-05-27 17:40:06 UTC

R topics documented:

coeffAll2 ......................................................... 3
coeffAll2,ANY-method ....................................... 3
copula_families ................................................ 4
deltaCI ............................................................. 4
final_pbox ..................................................... 5
fit_copula ....................................................... 6
fit_copula_pbox ............................................... 6
fit_dist_pbox .................................................. 7
fun_stats ....................................................... 8
fun_stats,numeric-method .................................... 9
gen_scenario .................................................. 9
grid_pbox ....................................................... 10
make_pbox ..................................................... 11
match_maker ................................................... 12
match_maker,ANY-method ................................... 12
modify_pbox ................................................... 13
param_dev ..................................................... 14
pbox-class .................................................... 14
perProb .......................................................... 15
perturbate_params ......................................... 16
perturbate_params,ANY-method ............................ 16
probCI ............................................................ 17
probCI,numeric-method ..................................... 18
qpbox ............................................................. 18
q_parser ........................................................ 20
q_parser,ANY-method ....................................... 20
scenario_pbox .................................................. 21
SEAex ............................................................ 22
set_pbox ........................................................ 22
show ............................................................. 23
stats_calc ....................................................... 24

Index 25
Extract Coefficients

Descripton
This is an internal method to extract coefficients from the list of the fitted distributions for each variable resulting from fit_dist_pbox. This method handles potential issues with parameter extraction from the complex objects created by GAM-like models.

Usage
coefAll2(obj, deviance = FALSE)

Arguments
- obj: An object typically resulting from fit_dist_pbox.
- deviance: Logical value indicating whether to compute deviance for the fitted model.

Value
A list of coefficients, possibly including 'mu', 'sigma', 'nu', and 'tau', depending on the model specification in obj. If deviance is TRUE, it also includes the deviance of the model.

Examples
```r
data(SEAex)
pbx <- set_pbox(SEAex)
coefAll2(pbx@fit[[1]]$allDistrs$Thailand)
```

Method for extracting coefficients from GAM-like models

Description
Method for extracting coefficients from GAM-like models

Usage
```r
## S4 method for signature 'ANY'
coefAll2(obj, deviance = FALSE)
```

Arguments
- obj: A model object, typically from a GAM-like fitting procedure.
- deviance: A Boolean flag that when TRUE calculates the deviance of the model.
**Value**

A list containing model coefficients and optionally deviance.

**Description**

Internal list of defined copula families and their corresponding parameters.

**Usage**

`.copula_families`

**Format**

An object of class `list` of length 3.

**deltaCI**

*Compute Confidence Interval using Delta Method*

**Description**

Internal method to compute the probability using delta method which approximates the variance of a function of random variables (in this case, the ratio) based on the variance of the original estimates.

‘deltaCI’ general method. Internal method to compute the probability using delta method which approximates the variance of a function of random variables (in this case, the ratio) based on the variance of the original estimates.

**Usage**

`deltaCI(cond)`

```
## S4 method for signature 'ANY'
deltaCI(cond)
```

**Arguments**

`cond` list with the result of the perturbed probability for ‘mj’ and ‘co’ and corresponding CI.

**Value**

The Confidence Interval for the conditional probability.

Numeric vector representing the computed probability and confidence intervals using the perturbed copula and delta method.
Examples

```r
cond <- list(
  c(P = 0.3597117, `2.5%` = 0.3074215, `97.5%` = 0.4075315),
  c(P = 0.5682882, `2.5%` = 0.4560553, `97.5%` = 0.6823438))
deltaCI(cond)
```

---

**Build a Multivariate Distribution from Copula**

**Description**

Combines the results from ‘fit_copula_pbox’ and ‘fit_dist_pbox’ to build a multivariate distribution from copula, selecting the best copula based on AIC and utilizing the best-fitted marginal distributions. Note that Method to construct a ‘mvdc’ object by combining best-fit copula and marginal distribution results. The method uses the best copula model as determined by the lowest AIC and combines it with marginal distributions fitted to each variable.

**Usage**

```r
final_pbox(results_df, allDitrs, data, verbose = TRUE)
```

## S4 method for signature 'ANY'

```r
final_pbox(results_df, allDitrs, data, verbose = TRUE)
```

**Arguments**

- **results_df**: A data.table with AIC and parameter estimates of evaluated copulas and families from ‘fit_copula_pbox’.
- **allDitrs**: A list containing fitted distributions for each variable from ‘fit_dist_pbox’.
- **data**: A data frame or data table; this will be coerced to a ‘data.table’ internally.
- **verbose**: control verbosity of the output. Default to TRUE.

**Value**

An object of class ‘mvdc’ representing the combined multivariate distribution.

**Examples**

```r
data("SEAex")
copulaFits <- fit_copula_pbox(data = SEAex, .copula_families)
distFits <- fit_dist_pbox(data = SEAex)
final_mvd <- final_pbox(copulaFits, distFits$allDitrs, SEAex)
print(final_mvd)
```
fit_copula  
Copula Fit

Description

Internal method to automatically find the best Copula given a data.frame. Wrapper around the function fitCopula. Automatically fits a copula model using the provided pseudo-observations. This method supports various families of copulas and calculates the corresponding AIC and parameter estimates.

Usage

.fit_copula(copula, family, dim, u)

## S4 method for signature 'ANY'
.fit_copula(copula, family, dim, u)

Arguments

copula  A data.frame or data.table (the data will be coerced to a data.table internally).
family  List of copula types and their corresponding families. Currently supported families are "clayton", "frank", "amh", "gumbel", and "joe" for Archimedean Copula; "galambos", "gumbel", and "huslerReiss" for Extreme-Value copula; "normal" and "t" for Elliptical copula.
dim  number of columns of data.
u  matrix of (pseudo-)observations. Consider applying the function pobs() first in order to obtain such data.

Value

A data.table with the corresponding AIC and the parameter estimates of the evaluated copulas and families.

fit_copula_pbox  
Fit Copula Models to Data

Description

Automatically fits various copula models specified in a list to the provided data. This function is a wrapper around the underlying copula fitting function, facilitating the exploration of multiple copula families to identify the best fitting model based on criteria such as AIC.

'fit_copula_pbox' method to fit a variety of copula models to data. This method performs a grid search over specified copula families to find the best fit. It employs the pseudoinverse of the empirical distribution functions to standardize the data.
**Usage**

```r
fit_copula_pbox(data, .copula_families)
```

## S4 method for signature 'ANY'

```r
fit_copula_pbox(data, .copula_families)
```

**Arguments**

- `data`: A data frame or data table; the data will be coerced to a `data.table` internally.
- `copula_families`: A list specifying copula families to evaluate. The list should be structured with names corresponding to the type of copula (e.g., `archmCopula`, `evCopula`, `ellipCopula`) and elements being vectors of strings naming the copula families (e.g., "clayton", "frank").

**Value**

A data table summarizing the AIC and parameter estimates for each copula family evaluated.

**Examples**

```r
data("SEAex")
copula_families <- list(
  archmCopula = c("clayton", "frank", "gumbel", "joe"),
  evCopula = c("galambos", "gumbel", "huslerReiss"),
  ellipCopula = c("normal")
)
distFits <- fit_copula_pbox(data = SEAex, copula_families)
print(distFits)
```

---

**Description**

Fits the best marginal distribution for each variable in a data frame using the `gamlss::fitDist` function from the GAMLSS package. This function is designed to evaluate multiple distributions, returning a summary of fit for each, along with the Akaike Information Criterion (AIC) for comparison.

Implements the generic function `fit_dist_pbox` for data frames and data tables. This method utilizes statistical techniques to fit distributions to each column in the `data` argument, evaluating fit using criteria like AIC to determine the best fitting model.

**Usage**

```r
fit_dist_pbox(data, ...)
```

## S4 method for signature 'ANY'

```r
fit_dist_pbox(data, ...)
```
fun_stats

Arguments

data 	A data frame or data table.

Value

A list containing two elements:

allDitrs List of the fitted distributions for each variable.
distTable A data table displaying the AIC for each tested distribution.

Examples

data(SEAex)
distFits <- fit_dist_pbox(data=SEAex)
print(distFits$allDitrs)
print(distFits$distTable)

fun_stats 

Summary Statistics

Description

Computes summary statistics for a numeric vector. This function is an S4 method for the generic 'fun_stats', specifically tailored for numeric vectors. It calculates the minimum, maximum, mean, and median values.

Usage

fun_stats(x)

Arguments

x 	A numeric vector for which summary statistics are to be computed.

Value

A list containing the minimum, maximum, mean, and median of the input vector.

Examples

x <- c(1, 2, 3, 4, 5)
fun_stats(x)
fun_statsnumeric-method

Summary statistics method for numeric vectors

Description

This method is a specific implementation of the 'fun_stats' function for numeric vectors. It efficiently calculates and returns summary statistics including the minimum, maximum, mean, and median, excluding NA values.

Usage

```r
## S4 method for signature 'numeric'
fun_stats(x)
```

Arguments

- `x` Numeric vector for which summary statistics are computed.

Value

A list with components min, max, mean, and median.

gen_scenario

Generate Scenarios

Description

Internal method to Generate scenarios based on parameter list variations.

Usage

```r
gen_scenario(params = "list")
```

Arguments

- `params` List of parameters where each parameter can vary across scenarios.

Value

Nested list of scenarios.
Examples

```r
some_distr <- list(A=list(mu = 31.07, sigma = 0.28),
B=list(mu = c(34.4,31.4,25.6), sigma = 0.98, nu = 1.7),# note mu!
C=list(mu = 31.4, sigma = 0.34),
D=list(mu = 25.6, sigma = 0.24))
gen_scenario(some_distr)
```

Description

This function queries the probabilistic space of a pbox object to calculate probabilities associated with specific marginal or conditional distributions on a quantile grid. It supports conditional probability calculations as well.

This method processes the pbox object to compute probabilities based on the specified marginal and conditional parameters. It handles both simple probability calculations and complex queries involving joint and conditional distributions, with an option for bootstrap confidence interval estimation.

Usage

```r
grid_pbox(pbx, mj = character(), co = NULL, probs = seq(0, 1, 0.1), ...)
```

Arguments

- `pbx`:
  An object of class pbox from which to query the probabilistic space.

- `mj`:
  A character vector specifying the variables to query.

- `co`:
  A character vector specifying the variables to query.

- `probs`:
  A numeric vector of quantiles to calculate probabilities for (default: `seq(0, 1, 0.1)`).

- `...`:
  Additional parameters passed to `qpbox`.

Value

A data.table containing estimated probabilities for each combination of quantiles and distributions queried.

Examples

```r
data("SEAex")
pbx <- set_pbox(SEAex)
grid_pbox(pbx, mj = c("Vietnam", "Malaysia"))
```
Create a Probability Box (Pbox) Object

Description

Constructs a probability box (Pbox) object from a given dataset and a pre-defined copula model. This auxiliary method facilitates the integration of data with a copula to form a comprehensive probabilistic model known as a Pbox.

Method for creating a 'pbox' object using a specified copula and data. This method ensures that the input data and copula are compatible in terms of dimensions and structurally fit to form a Pbox.

Usage

```r
make_pbox(data, cop)
```

## S4 method for signature 'ANY'

```r
make_pbox(data, cop)
```

Arguments

- `data`: A dataframe or data table; this data will be coerced to a 'data.table' internally.
- `cop`: An object of class 'mvdc' representing the multivariate dependency structure (copula).

Value

An object of class 'pbox' with slots: - `$data`: The data coerced into a 'data.table'. - `$copula`: The provided copula object.

Examples

```r
library(copula)
data("SEAex")
cop <- normalCopula(param = 0.5, dim = 4) 
distList <- c("RG", "SN1", "RG", "RG") 
allDistrs <- list(list(mu = 31.07, sigma = 0.28), 
                list(mu = 34.4, sigma = 0.98, nu = 1.7), 
                list(mu = 31.4, sigma = 0.34), 
                list(mu = 25.6, sigma = 0.24))
copSEA <- mvdc(cop, distList, allDistrs) 
pbx <- make_pbox(data = SEAex, cop = copSEA) 
print(class(pbx))
```
**match_maker**  
*Generate Query Vector*

**Description**

This function defines a generic function for creating a query vector to explore the probabilistic space based on provided matches and data. It is used internally to handle different types of inputs efficiently.

**Usage**

```r
match_maker(varSet, matches, data)
```

**Arguments**

- **varSet**  
  A data frame or list describing the variable set.

- **matches**  
  A data frame describing the matches with potential additional control parameters.

- **data**  
  A data frame representing the data to be queried.

**Value**

A modified version of `varSet` with values updated based on `matches`.

---

**match_maker,ANY-method**  
*Method for match_maker*

**Description**

This method implements the `match_maker` function for handling specific types of `varSet`, `matches`, and `data`. It modifies the `varSet` based on `matches` which can contain variable names and values to be matched or operations to be performed. It supports operations and direct value assignment.

**Usage**

```r
## S4 method for signature 'ANY'
match_maker(varSet, matches, data)
```

**Arguments**

- **varSet**  
  A data frame or list describing the variable set.

- **matches**  
  A data frame describing the matches with variable names and corresponding values or operators.

- **data**  
  A data frame representing the data to be queried.
Value

A modified version of `varSet` that integrates conditions or values from `matches`.

See Also

`match_maker` for the generic function and additional details.

modify_pbox

Modify Parameters Box

Description

Internal method to modify specific parameters in a nested list structure by applying deviations.

Usage

```r
modify_pbox(all_parms, params_list, sigma = 0.05, range = seq(-3, 3, 1))
## S4 method for signature 'ANY'
modify_pbox(all_parms, params_list, sigma = 0.05, range = seq(-3, 3, 1))
```

Arguments

- `all_parms` nested list of parameters from the pbox object.
- `params_list` Named list where each name corresponds to a variable in the dataset and the value is a vector of parameter names to modify (e.g. list(Vietnam="mu")).
- `sigma` Standard deviation used for calculating parameter deviations.
- `range` Range values for generating deviations.

Value

Modified list of parameters.

Examples

```r
some_distr<-list(A=list(mu = 31.07, sigma = 0.28),
B=list(mu = 34.4, sigma = 0.98, nu = 1.7),
C=list(mu = 31.4, sigma = 0.34),
D=list(mu = 25.6, sigma = 0.24))
modify_pbox(some_distr, list(A = "mu"))
```
**param_dev**  
*Compute Parameter Deviations*

**Description**

Internal method to calculate ± 1, 2, 3 standard deviations for given parameters.

**Usage**

```r
param_dev(param = "numeric", sigma = 0.05, range = seq(-3, 3, 1))
```

```r
## S4 method for signature 'ANY'
param_dev(param = "numeric", sigma = 0.05, range = seq(-3, 3, 1))
```

**Arguments**

- `param` Numeric vector of parameters.
- `sigma` Numeric value representing standard deviation (default is 0.05).
- `range` Numeric vector specifying range of deviations (default is seq(-3, 3, 1)).

**Value**

Numeric vector of parameters adjusted by the specified deviations.

**Examples**

```r
param_dev(31)
```

---

**pbox-class**  
*Class "pbox": Main S4 class of the library pbox.*

**Description**

"pbox" is a class representing the probabilistic space which combines data, copula and margins.

**Slots**

- `data` The original data coerced to a `data.table`.
- `copula` The copula object of class `mvdc`.
- `fit` The results of the automated selection for both the marginal distribution and the copula.
**perProb**  

*Compute Probability Using a Perturbed Copula*

**Description**

Computes the probability by applying a perturbation to the copula parameters within a 'pbox' object, and then evaluating the probability for specified query values. This method ensures that variations in the copula parameters can be assessed for their impact on the computed probabilities.

'perProb' method for objects of class 'pbox'. This method perturbs the parameters of the copula contained in the 'pbox' and then computes the probability of the vector query using the perturbed copula. The perturbation process adjusts the copula parameters and evaluates the impact on the outcome probability.

**Usage**

```r
perProb(x, vecQuery)
```

## S4 method for signature 'pbox'

```r
perProb(x, vecQuery)
```

**Arguments**

- `x`  
  A 'pbox' object, which is expected to contain a copula.

- `vecQuery`  
  A numeric vector representing the query values.

**Value**

The probability computed using a perturbed copula.

Numeric value representing the computed probability using the perturbed copula.

**See Also**

- `set_pbox`, `pMvdc`

**Examples**

```r
data(SEAex)
pbx <- set_pbox(SEAex[, .(Malaysia, Thailand)])
vecQuery <- c(31, 34)
perProb(pbx, vecQuery)
```
perturbate_params, ANY-method

perturbate_params

**Perturb Parameters**

**Description**
This function defines a generic function to perturbate parameter values for each distribution within a copula, using random perturbations to simulate variability or uncertainty.

**Usage**

```r
perturbate_params(paramMargins)
```

**Arguments**

- `paramMargins` A list containing lists of parameter values for each distribution in the copula.

**Value**
A list of lists containing perturbed parameter values.

**Examples**

```r
paramMargins <- list(list(0.2, 0.3), list(0.4, 0.5))
perturbed <- perturbate_params(paramMargins)
print(perturbed)
```

perturbate_params, ANY-method

**Perturb Parameters Method**

**Description**
This method implements the generic `perturbate_params` function specifically for lists of copula distribution parameters. It applies a random perturbation to each parameter based on a normal distribution centered at zero with a standard deviation of 0.05.

**Usage**

```r
## S4 method for signature 'ANY'
perturbate_params(paramMargins)
```

**Arguments**

- `paramMargins` A list containing lists of parameter values for each distribution in the copula.
probCI

Value

A list of lists containing perturbed parameter values.

See Also

perturbate_params for the generic function definition.

Description

Calculates the confidence interval around a vector of probabilities using the quantiles based on the specified significance level.

Usage

probCI(probabilities, alpha=0.05)

Arguments

probabilities  A numeric vector of probabilities for which the confidence interval is desired.
alpha  The significance level used for constructing the confidence interval; default is 0.05.

Value

A list containing the lower and upper bounds of the confidence intervals for each probability.

Examples

probabilities <- c(0.1, 0.2, 0.3, 0.4, 0.5)
probCI(probabilities)
probCI(probabilities, alpha = 0.1)
probCI, numeric-method

Method to calculate confidence intervals for a vector of probabilities

Description

This method calculates the lower and upper bounds of the confidence interval for each element in the input vector of probabilities using the given alpha level.

Usage

```r
## S4 method for signature 'numeric'
probCI(probabilities, alpha = 0.05)
```

Arguments

- probabilities: A numeric vector of probabilities.
- alpha: A numeric value specifying the significance level for the confidence intervals; defaults to 0.05.

Value

A numeric vector containing the lower and upper quantile bounds for each probability in the input vector.

qpbox

Query the probabilistic space of a pbox object.

Description

This function queries the probabilistic space of a pbox object to calculate probabilities associated with specific marginal or conditional distributions. It supports conditional probability calculations and can optionally estimate confidence intervals through bootstrapping.

This method processes the pbox object to compute probabilities based on the specified marginal and conditional parameters. It handles both simple probability calculations and complex queries involving joint and conditional distributions, with an option for bootstrap confidence interval estimation.

Usage

```r
qpbox(
    pbx, 
    mj = "character", 
    co = "character", 
    lower.tail = TRUE, 
    fixed = FALSE, 
    CI = FALSE, 
```
qpbox

iter = 1000

## S4 method for signature 'pbox'
qpbox(
  pbx,
  mj = "character",
  co = "character",
  lower.tail = TRUE,
  fixed = FALSE,
  CI = FALSE,
  iter = 1000
)

### Arguments

- **pbx**
  An object of class pbox from which to query the probabilistic space.

- **mj**
  A character string specifying the marginal and or joint distribution of the variable. It must specify the variable and the value in the format ‘Var:Val’.

- **co**
  A character string specifying the marginal and conditional distribution of the variable. It must specify the variable and the value in the format ‘Var:Val’.

- **lower.tail**
  Logical; if TRUE (default), probabilities are calculated for the area to the right of the specified value.

- **fixed**
  Logical; if TRUE, calculates conditional probabilities with conditions treated as fixed.

- **CI**
  Logical; if TRUE, calculates bootstrap confidence intervals.

- **iter**
  Integer; the number of replications for the confidence interval calculation. Default is 1000.

### Value

Estimated probabilities as a numeric value or a named vector including confidence intervals if requested.

### Examples

```r
data("SEAex")
pbx <- set_pbox(SEAex)
# Get marginal distribution
qpbox(pbx, mj="Malaysia:33")
# Get conditional distribution
qpbox(pbx, mj="Malaysia:33 & Vietnam:31", co="avgRegion:26")
```
Description

This function defines a generic function to parse a string query into structured data that can be used to explore a pbox object. It extracts components of the query using regular expression matching.

Usage

q_parser(query)

Arguments

query A string representing the query.

Value

A data table with columns ‘Varnames’, ‘Value’, ‘Operator’, and ‘Varnames2’, where numeric values are converted to numeric type, and unnecessary columns are removed.

Examples

query <- "Vietnam:23"
q_parser(query)
**scenario_pbox**

**See Also**

-q_parser for the generic function definition.

---

**scenario_pbox**  
*Scenario Analysis*

**Description**

Performs scenario analysis by modifying underlying parameters of a pbox object. Query the probabilistic space under different scenarios with different combinations of parameters for a single query.

**Usage**

```r
scenario_pbox(
  pbx,  
  param_list = "list",  
  sigma = 0.05,  
  range = seq(-3, 3, 1),  
  ...  
)

## S4 method for signature 'pbox'
scenario_pbox(
  pbx,  
  param_list = "list",  
  sigma = 0.05,  
  range = seq(-3, 3, 1),  
  ...
)
```

**Arguments**

- `pbx`: object of class pbox
- `param_list`: List specifying which parameters to modify.
- `sigma`: Standard deviation for parameter deviations, defaulting to 0.05.
- `range`: Range of deviation multipliers, default is seq(-3, 3, 1).
- `...`: Additional arguments passed to qpbox.

**Value**

Named list of results from each scenario evaluation.

**Examples**

```r
data("SEAex")
pbx <- set_pbox(SEAex)
scenario_pbox(pbx,mj = "Vietnam:31 & avgRegion:26", param_list = list(Vietnam="mu"))
```
Description

Maximum yearly temperature data from 1901 to 2022 in 11 countries in Southeast Asia and the average temperature of the entire region extracted from Climatic Research Unit gridded Time Series Version 4. Data contains only temperatures for Malaysia, Thailand, Vietnam and the average regional temperature.

Usage

SEAx

Format

## 'SEAx' A data frame with 122 rows and 4 columns:

- **Malaysia, Thailand, Vietnam** Yearly max temperatures in Celsius for each country over 122 years.
- **avgRegion** Average temperature in Celsius over the whole South East Asia region

Source

[<https://crudata.uea.ac.uk/cru/data/hrg/cru_ts_4.07/crucy.2304181636.v4.07/countries/>](https://crudata.uea.ac.uk/cru/data/hrg/cru_ts_4.07/crucy.2304181636.v4.07/countries/)

Examples

```r
data(SEAex)
head(SEAex)
```

Description

Constructs a probability box (pbox) by automatically selecting the best marginal distribution and copula for a given dataset. This function facilitates the creation of a pbox object, which encapsulates the uncertainty and dependencies of the input data.

‘set_pbox’ method that utilizes data frames or data tables to configure a comprehensive pbox structure. The method involves stages of distribution fitting and copula selection, executed through external functions presumed to be available in the working environment or described in the package.
show

Usage

set_pbox(data, verbose = TRUE, ...)

## S4 method for signature 'ANY'
set_pbox(data, verbose = TRUE, ...)

Arguments

data A data frame or data table. The data will be coerced to a 'data.table' internally.
verbose control verbosity of the output. Default to TRUE.
... Other arguments to be passed to the 'fitDist' function.

Value

An object of class 'pbox' with the following slots: - '@data': The original data coerced into a 'data.table'. - '@copula': The selected copula object, typically of class 'mvdc'. - '@fit': A list containing results from the automated selection processes for both the marginal distributions and the copula.

Examples

data("SEAex")
pbx <- set_pbox(data = SEAex)
print(pbx)
print(class(pbx))
stats_calc  

*Calculate Basic Statistics*

**Description**

Computes basic statistics such as mean and median for specified variables in a data frame or data table based on a set of operations specified in the `matches` data frame. This function updates the `varSet` with the computed results for each variable.

Method implementation for calculating statistics using `data.table` and `stats`. This method allows the computation of mean and median for subsets of data defined in `matches` and updates `varSet` with these results.

**Usage**

```r
stats_calc(data, matches, varSet)
```

```r
## S4 method for signature 'ANY'
stats_calc(data, matches, varSet)
```

**Arguments**

- `data` A data frame or data table.
- `matches` A data frame describing the operations to apply.
- `varSet` A data frame to be updated with results.

**Value**

Returns a modified version of `varSet` with updated values based on the calculations.
Index

* datasets
  copula_families, 4
  SEAex, 22
  .copula_families (copula_families), 4
  .fit_copula (fit_copula), 6
  .fit_copula, ANY-method (fit_copula), 6
  coefAll2, 3
  coefAll2, ANY-method, 3
  coefAll2-method (coefAll2), 3
  copula_families, 4
  deltaCI, 4
  deltaCI, ANY-method (deltaCI), 4
  final_pbox, 5
  final_pbox, ANY-method (final_pbox), 5
  fit_copula, 6
  fit_copula_pbox, 6
  fit_copula_pbox, ANY-method
    (fit_copula_pbox), 6
  fit_dist_pbox, 3, 7
  fit_dist_pbox, ANY-method
    (fit_dist_pbox), 7
  fun_stats, 8
  fun_stats, numeric (fun_stats), 8
  fun_stats, numeric-method, 9
  gen_scenario, 9
  gen_scenario, ANY-method (gen_scenario), 9
  grid_pbox, 10
  grid_pbox, pbox-method (grid_pbox), 10
  make_pbox, 11
  make_pbox, ANY-method (make_pbox), 11
  match_maker, 12, 13
  match_maker, ANY-method, 12
  modify_pbox, 13
  modify_pbox, ANY-method (modify_pbox), 13
  param_dev, 14
  param_dev, ANY-method (param_dev), 14
  pbox-class, 14
  perProb, 15
  perProb, pbox-method (perProb), 15
  perturbate_params, 16, 17
  perturbate_params, ANY-method, 16
  pMvdc, 15
  probCI, 17
  probCI, numeric-method, 18
  probCI-method (probCI), 17
  q_parser, 20, 21
  q_parser, ANY-method, 20
  qpbox, 18
  qpbox, pbox-method (qpbox), 18
  scenario_pbox, 21
  scenario_pbox, pbox-method
    (scenario_pbox), 21
  SEAex, 22
  set_pbox, 15, 22
  set_pbox, ANY-method (set_pbox), 22
  show, 23
  show, pbox-method (show), 23
  show-pbox (show), 23
  stats_calc, 24
  stats_calc, ANY-method (stats_calc), 24