Package ‘AIDA’

May 12, 2026

Type Package

Title Analysis of Interval DAta

Version 0.1.5

Description Tools for the analysis of interval-
valued data, including construction, visualization, and statistical modeling.
The package provides the 'intData' class for representing interval-valued data, along with functions to aggregate microdata and to estimate parameters of latent distributions.
Barycenter and covariance matrix estimation is implemented based on the Mallows distance (Oliveira et al. (2025) <doi:10.48550/arXiv.2407.05105>).
Robust estimation of the symbolic covariance matrix is implemented via the Interval Minimum Covariance Determinant (IMCD) estimator, enabling outlier detection based on the robust squared Interval-Mahalanobis distance, as proposed by Loureiro et al. (2026) <doi:10.48550/arXiv.2604.26769>.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/catarinaploureiro/AIDA,
https://catarinaploureiro.github.io/AIDA/

BugReports https://github.com/catarinaploureiro/AIDA/issues

RoxygenNote 7.3.3

LazyData true

LazyDataCompression xz

VignetteBuilder knitr

Language en-US

Imports ggplot2, ggrepel, CerioliOutlierDetection, cellWise, geigen, kde1d, plotly, robustbase, MASS, assertthat, methods

Depends R (>= 3.6)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), corrplot

NeedsCompilation no

Author Catarina P. Loureiro [aut, cre] (ORCID:
<https://orcid.org/0009-0001-9464-5824>)

Maintainer Catarina P. Loureiro <catarinapadrela@tecnico.ulisboa.pt>

Repository CRAN

Date/Publication 2026-05-12 19:30:02 UTC

Contents

`==,intData,intData-method` *Equality Comparison for intData Objects*

Description

Compare two intData objects for equality.

Compare two intData objects for inequality.

Usage

```
## S4 method for signature 'intData,intData'
e1 == e2

## S4 method for signature 'intData,intData'
e1 != e2
```

Arguments

`e1`	An intData object.
`e2`	An intData object.

Value

A logical matrix indicating which elements are equal between the two intData objects.

A logical matrix indicating element-wise inequality of the two intData objects.

`angle_error` *Angle Error*

Description

Computes the angle error between eigenvalues of the estimated covariance matrix and of the ground truth covariance matrix.

Usage

```
angle_error(est_cov, ground_truth_cov)
```

Arguments

`est_cov` Estimated covariance matrix.

`ground_truth_cov` Ground truth covariance matrix.

Details

The angle error is given by:

$$1 - \frac{\hat{\boldsymbol{a}}^\top \boldsymbol{a}}{\sqrt{\hat{\boldsymbol{a}}^\top \hat{\boldsymbol{a}}}\sqrt{\boldsymbol{a}^\top \boldsymbol{a}}},$$

where $\hat{\boldsymbol{a}}$ and $\boldsymbol{a}$ are the eigenvalues of the estimated and ground truth covariance matrices, respectively.

Value

Angle error between eigenvalues.

`Centers` *Centers Method for intData*

Description

Centers Method for intData

Usage

```
Centers(Sdt)

## S4 method for signature 'intData'
Centers(Sdt)
```

Arguments

`Sdt`	An object of class intData.

Value

A `data.frame` containing the centers of the intervals.

`colnames,intData-method` *Column Names Method for intData*

Description

Column Names Method for intData

Usage

```
## S4 method for signature 'intData'
colnames(x)
```

Arguments

`x`	An object of class intData.

Value

A character vector of column names.

`creditcard` *Credit Card Dataset*

Description

This dataset contains interval data of credit card expenses, including min-max values, centers and ranges, microdata, and an intData object. It is composed of 5 variables: Food, Social, Travel, Gas, and Clothes. It was aggregated by person-month.

Usage

```
data(creditcard)
```

Format

A list with the following components:

`microdata` A data frame with `1000` rows and `7` columns. It contains the microdata, with individual measurements of each variable for all observations.

`min_max` A data frame with `36` rows and `10` columns. Each row corresponds to a different observation, and each column gives the minimum and maximum values for each variable.

`centers_ranges` A data frame with `36` rows and `10` columns. Each row corresponds to the centers and ranges of the interval data.

`intData` An intData object with `36` interval-valued observations and `5` variables, constructed assuming the microdata follow symmetric triangular distributions.

References

This data was retrieved from Billard, L. and Diday, E. (2006). Symbolic Data Analysis: Conceptual Statistics and Data Mining. John Wiley & Sons. doi:10.1002/9780470090183.

Examples

```
data(creditcard)
head(creditcard$min_max)
head(creditcard$microdata)
head(creditcard$intData)
```

`dim,intData-method` *Dimensions Method for intData*

Description

Dimensions Method for intData

Usage

```
## S4 method for signature 'intData'
dim(x)
```

Arguments

x	An object of class intData.

Value

A vector of the number of rows and columns.

entrecampos_air_quality

Entrecampos Air Quality Dataset

Description

This dataset contains interval data of air pollutants' concentrations, including min-max values and microdata. This air quality dataset was obtained from a monitoring station in Entrecampos, Lisbon. It is composed of 9 pollutants' concentration measures in µg/m3 during the years 2019, 2020, and 2021: sulphur dioxide (SO2), particles < 10µm, ozone (O3), nitrogen dioxide (NO2), carbon monoxide (CO), benzene (C6H6), particles < 2.5µm, nitrogen oxides (NOx), and nitrogen monoxide (NO). For the `microdata_transformed`, `min_max`, and `intData`, the pollutant "benzene" was removed due to a high number of missing values. The aggregation of the microdata was done by day.

Usage

```
data(entrecampos_air_quality)
```

Format

A list with the following components:

`microdata_raw` A data frame with `26304` rows and `11` columns. It contains the raw microdata, with individual measurements of each variable for all observations.

`microdata_transformed` A data frame with `26304` rows and `10` columns. It contains the microdata, with individual measurements of each variable for all observations. Logarithmic transformations were applied to all variables and interpolation to deal with missing values.

`min_max` A data frame with `1096` rows and `17` columns. Each row corresponds to a different observation, and each column gives the minimum and maximum values for each variable. The first column corresponds to the day, the next 8 to the minimum and the last 8 to the maximum.

`intData` An intData object, constructed using KDE for estimating the parameters of the latent distributions.

References

This data was retrieved from the Portuguese Environment Agency database available at `https://qualar.apambiente.pt/`.

Examples

```
data(entrecampos_air_quality)
head(entrecampos_air_quality$microdata_raw)
head(entrecampos_air_quality$microdata_transformed)
head(entrecampos_air_quality$min_max)
head(entrecampos_air_quality$intData)
```

farness *Farness Estimation*

Description

Estimate farness from a distance vector in order to identify outlier observations.

Usage

```
farness(dist, cutoff_value = NULL)
```

Arguments

dist	Vector of distances of each observation.
cutoff_value	Optional cutoff value between 0 and 1 to flag outliers. If provided, the function returns both the farness probabilities and the cutoff distance value in the original distance scale.

Value

Farness of each observation. Values between 0 and 1. If `cutoff_value` is provided, a list with the farness probabilities and the cutoff distance value in the original distance scale is returned.

References

J. Raymaekers and P.J. Rousseeuw (2021). Transforming variables to central normality. Machine Learning. doi:10.1007/s10994021059605

Based on the `cellWise` package: Raymaekers J, Rousseeuw P (2023). *cellWise: Analyzing Data with Cellwise Outliers*. R package version 2.5.3, `https://CRAN.R-project.org/package=cellWise`.

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

# Compute squared Interval-Mahalanobis distance
z <- rep(1, nrow(credit_card_int))
credit_card_dist<-IMah_dist(credit_card_int,z)

credit_card_farness <- farness(credit_card_dist, 0.9)
```

frobenius_error *Relative Frobenius Error*

Description

Computes the relative Frobenius error between an estimated covariance matrix and the ground truth.

Usage

```
frobenius_error(est_cov, ground_truth_cov)
```

Arguments

`est_cov`	Estimated covariance matrix.
`ground_truth_cov`	Ground truth covariance matrix.

Details

The relative Frobenius error is given by:

$$\frac{\|\boldsymbol{A} - \boldsymbol{B}\|_F}{\|\boldsymbol{B}\|_F} = \frac{\sqrt{\sum_{i=1}^{p} \sum_{j=1}^{p} |[\boldsymbol{A}]_{ij} - [\boldsymbol{B}]_{ij}|^2}}{\sqrt{\sum_{i=1}^{p} \sum_{j=1}^{p} |[\boldsymbol{B}]_{ij}|^2}},$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the estimated and ground truth covariance matrices, respectively.

Value

Frobenius error between the two matrices.

get_latent_param *Compute Latent Variables Parameters*

Description

Obtain the parameters of the latent variables inherent to the macrodata.

Usage

```
get_latent_param(
  LatentCase = c("U_id_symmetric", "U_id", "General"),
  LatentDist = c("Unif", "Triang", "TNorm", "InvTri", "Beta", "KDE", "Degenerated"),
  TriangParam = 0,
  BetaParam.a = 1,
  BetaParam.b = 1,
  Umicro = NULL,
  p = NULL,
  estimate.DistParam = FALSE
)
```

Arguments

`LatentCase` A string specifying which of the three scenarios applies to the latent variables:

- `"General"`: The case where the latent variables do not have any nice properties.
- `"U_id"`: The case where the latent variables are identically distributed.
- `"U_id_symmetric"`: The case where the latent variables are identically distributed and symmetric.

Defaults to `"U_id_symmetric"`.

`LatentDist` A string or vector of strings specifying the distribution(s) of the latent variables. If the variables are identically distributed it can be one of (`"Unif"`,`"Triang"`,`"TNorm"`,`"InvTri"`,`"Beta"`,`"` if not a vector must be provided with the distribution for each variable.

`TriangParam` Mode of the triangular distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `0`.

`BetaParam.a` Parameter alpha of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `1`.

`BetaParam.b` Parameter beta of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `1`.

`Umicro` Latent microdata observations. Needed if `LatentDist="KDE"` or `estimate.DistParam=TRUE`.

`p` Number of variables.

`estimate.DistParam`
Logical parameter indicating if estimation of the parameters of the latent distributions should be performed. Can only be set to TRUE if `LatentCase="General"`. The default is `FALSE`.

Details

The parameters of the latent variables inherent to the macrodata are defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric, so its parameters are:

  - $\delta = \mathbb{E}(U^2)/4$
- "U_id": The latent variables are identically distributed, so its parameters are:
  - $\delta = \mathbb{E}(U^2)/4$
  - $\mathbb{E}(U)$
- "General": The latent variables do not have any nice properties, so its parameters are:
  - $[\mathfrak{E}_{UU}]_{ij} = \mathcal{E}(U_i, U_j)$, $i \neq j$, with $\mathcal{E}(U_i, U_j) = \int_0^1 F_{U_i}^{-1}(t) F_{U_j}^{-1}(t)\, dt$, and $[\mathfrak{E}_{UU}]_{ii} = \mathbb{E}(U_i^2)$, $i, j = 1, \ldots, p$
  - $\mathbf{\Psi} = \text{Diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$

Value

A list with the parameters of the latent variables.

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Examples

```
data(creditcard)
CreditCard_min_max <- creditcard$min_max
CreditCard_microdata <- creditcard$microdata
credit_agrby<-paste(CreditCard_microdata$Name,CreditCard_microdata$Month,sep = "_")
credit_card_U<-get_latent_var(CreditCard_microdata[,3:7], CreditCard_min_max, credit_agrby,
                             agrlevels = row.names(CreditCard_min_max), Seq="LbUb_VarbyVar")
credit_card_param<-get_latent_param(LatentCase="General",LatentDist="KDE",Umicro=credit_card_U)
```

`get_latent_var` *Compute Latent Variables*

Description

Obtain the latent variables inherent to the macrodata.

Usage

```
get_latent_var(
  microdata,
  macrodata,
  agrby,
  agrlevels,
  Seq = c("AllLb_AllUb", "AllCen_AllRng", "LbUb_VarbyVar", "CenRng_VarbyVar")
)
```

Arguments

`microdata` A matrix containing the microdata.

`macrodata` A data frame, matrix or intData object containing the macrodata/interval data.

`agrby` A factor used to specify the grouping of the microdata.

`agrlevels` The categories/levels on which the microdata was aggregated.

`Seq` Format of macrodata if it is a data frame or matrix. Available options are:

- `"AllLb_AllUb"`: All lower bounds followed by all upper bounds, in the same variable order.
- `"AllCen_AllRng"`: All Centers followed by all Ranges, in the same variable order.
- `"LbUb_VarbyVar"`: Lower bounds followed by upper bounds, variable by variable.
- `"CenRng_VarbyVar"`: Centers followed by Ranges, variable by variable.

Details

The latent variables, U_{ij}, are defined according to the following model:

Let $X_j = (C_j, R_j)^\top = \left[C_j - \frac{R_j}{2}, C_j + \frac{R_j}{2}\right]$ represent the **macrodata** and

$$V_{ij} = C_j + U_{ij}\frac{R_j}{2}, \quad j = 1, \ldots, p,\ i = 1, \ldots, m_j$$

the **microdata** with U_{ij} being random variables with support on $[-1, 1]$, uncorrelated with (C_j, R_j).

Value

A matrix with the same size as the microdata.

References

Oliveira, M.R., Azeitona, M., Pacheco, A., Valadas, R.. Association measures for interval variables. Advances in Data Analysis and Classification 16, 491–520 (2022). doi:10.1007/s11634021004458

Examples

```
data(creditcard)
CreditCard_min_max <- creditcard$min_max
CreditCard_microdata <- creditcard$microdata
credit_agrby<-paste(CreditCard_microdata$Name,CreditCard_microdata$Month,sep = "_")
credit_card_U<-get_latent_var(CreditCard_microdata[,3:7], CreditCard_min_max, credit_agrby,
                              agrlevels = row.names(CreditCard_min_max), Seq="LbUb_VarbyVar")
```

head,intData-method *Head Method for intData*

Description

Returns the first n rows of an intData object.

Usage

```
## S4 method for signature 'intData'
head(x, n = min(nrow(x), 6L))
```

Arguments

x	An intData object.
n	The number of rows to return.

Value

A subset of the intData object.

IMah_dist *Interval-Mahalanobis Distance*

Description

Calculate the squared Interval-Mahalanobis distance of all rows in the data and the barycenter.

Usage

```
IMah_dist(data, z = NULL, mean_c = NULL, mean_r = NULL, cov = NULL)
```

Arguments

data	An intData object containing the macrodata/interval data
z	A vector of 0 and 1, indicating which observations should be considered for the calculation. You must provide either z or (mean_c, mean_r and cov)
mean_c	The mean vector of the centers
mean_r	The mean vector of the ranges
cov	The symbolic covariance matrix

Details

The squared Interval-Mahalanobis distance is defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:

$$d_{IMah}(\boldsymbol{x})^2 = (\boldsymbol{c} - \boldsymbol{\mu}_C)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C) + \delta(\boldsymbol{r} - \boldsymbol{\mu}_R)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R),$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.

- `"U_id"`: The latent variables are identically distributed:

$$d_{IMah}(\boldsymbol{x})^2 = (\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_C)+\delta(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r}-\boldsymbol{\mu}_R)+\frac{\mathbb{E}(U)}{2}(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r}-\boldsymbol{\mu}_R)+\frac{\mathbb{E}(U)}{2}(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \boldsymbol{\Sigma}$$

  where $\delta = \mathbb{E}(U^2)/4$ and $\mathbb{E}(U)$ are the parameter of the latent variables.

- `"General"`: The latent variables do not have any nice properties:

$$d_{IMah}(\boldsymbol{x})^2 = (\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_C)+\frac{1}{4}(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \left(\mathfrak{E}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1}\right)(\boldsymbol{r}-\boldsymbol{\mu}_R)+\frac{1}{2}(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Psi}(\boldsymbol{r}-\boldsymbol{\mu}_R)+\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu}_R$$

  where:

  - $\boldsymbol{\Psi} = \text{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $[\mathfrak{E}_{UU}]_{ij} = \mathcal{E}(U_i, U_j)$, $i \neq j$, with $\mathcal{E}(U_i, U_j) = \int_0^1 F_{U_i}^{-1}(t)F_{U_j}^{-1}(t)\,dt$,
  - $[\mathfrak{E}_{UU}]_{ii} = \mathbb{E}(U_i^2)$, $i, j = 1, \ldots, p$,
  - $\bullet$ denotes the Schur (or entrywise) product of matrices.

Value

A vector with the squared Interval-Mahalanobis distance of each observation.

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Minimum Covariance Determinant Estimator and Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2604.26769. https://arxiv.org/abs/2604.26769

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

z <- rep(1, nrow(credit_card_int))
credit_card_dist<-IMah_dist(credit_card_int,z)
```

IMah_dist_pairs *Interval-Mahalanobis distance for all pairs*

Description

Calculate the squared Interval-Mahalanobis distance of all pairs of observations in the data.

Usage

```
IMah_dist_pairs(data, cov = NULL)
```

Arguments

`data`	An intData object containing the macrodata/interval data
`cov`	The symbolic covariance matrix

Details

The squared Interval-Mahalanobis distance is defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:

$$d_{IMah}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = (\boldsymbol{c}_i - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{c}_i - \boldsymbol{c}_j) + \delta (\boldsymbol{r}_i - \boldsymbol{r}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{r}_i - \boldsymbol{r}_j),$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.

- `"U_id"`: The latent variables are identically distributed:

$$d_{IMah}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = (\boldsymbol{c}_i - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{c}_i - \boldsymbol{c}_j) + \delta (\boldsymbol{r}_i - \boldsymbol{r}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{r}_i - \boldsymbol{r}_j) + \frac{\mathbb{E}(U)}{2} (\boldsymbol{c}_i - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{r}_i - \boldsymbol{r}_j) + \frac{\mathbb{E}(U)}{2} (\boldsymbol{r}_i - \boldsymbol{r}_j)^\top$$

  where $\delta = \mathbb{E}(U^2)/4$ and $\mathbb{E}(U)$ are the parameter of the latent variables.

- `"General"`: The latent variables do not have any nice properties:

$$d_{IMah}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = (\boldsymbol{c}_i - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_B^{-1} (\boldsymbol{c}_i - \boldsymbol{c}_j) + \frac{1}{4} (\boldsymbol{r}_i - \boldsymbol{r}_j)^\top \left(\boldsymbol{\mathfrak{E}}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1} \right) (\boldsymbol{r}_i - \boldsymbol{r}_j) + \frac{1}{2} (\boldsymbol{c}_i - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_B^{-1} \boldsymbol{\Psi} (\boldsymbol{r}_i - \boldsymbol{r}_j) + \frac{1}{2} (\boldsymbol{r}_i -$$

  where:

  - $\boldsymbol{\Psi} = \text{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $[\boldsymbol{\mathfrak{E}}_{UU}]_{ij} = \mathcal{E}(U_i, U_j)$, $i \neq j$, with $\mathcal{E}(U_i, U_j) = \int_0^1 F_{U_i}^{-1}(t) F_{U_j}^{-1}(t)\, dt$,
  - $[\boldsymbol{\mathfrak{E}}_{UU}]_{ii} = \mathbb{E}(U_i^2)$, $i, j = 1, \ldots, p$,
  - $\bullet$ denotes the Schur (or entrywise) product of matrices.

Value

A matrix with the squared Interval-Mahalanobis distance of each pair of observations.

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Minimum Covariance Determinant Estimator and Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2604.26769. https://arxiv.org/abs/2604.26769

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_dist<-IMah_dist_pairs(credit_card_int)
```

IMCD — *Interval-valued data Minimum Covariance Determinant (IMCD) estimation*

Description

Applies an adaptation of the FAST-MCD algorithm to estimate location and scatter for interval-valued data.

Usage

```
IMCD(
  data,
  m = 0,
  cutoff = c("farness", "adjbox", "chi-squared", "F-dist", "raw"),
  cutoff_lvl = NULL
)
```

Arguments

`data` An intData object containing the interval-valued dataset (macrodata).

`m` An integer specifying the subset size to use for the estimation. Defaults to `floor(0.75*n)`.

`cutoff` Indicates which cutoff should be considered for reweighting the estimates:

- `"chi-squared"`: The traditional 97.5\
- `"raw"`: No reweighting.
- `"adjbox"`: Adjusted Boxplots (package `robustbase`).
- `"F-dist"`: The quantile of the scaled F distribution (adapted from package `CerioliOutlierDetection`).
- `"farness"`: "Farness" is estimated from the robust distance (adapted from package `cellWise`).

Defaults to `"farness"`.

`cutoff_lvl` A numeric value specifying the level of the cutoff to be used.

- If cutoff="chi-squared", cutoff_lvl is the quantile of the Chi-squared distribution (default is 0.975).
- If cutoff="adjbox", cutoff_lvl is the coefficient for the adjusted boxplot (default is 1.5).
- If cutoff="F-dist", cutoff_lvl is the quantile of the F-distribution (default is 0.975).
- If cutoff="farness", cutoff_lvl represents the threshold for farness, with a default of 0.99.
- If cutoff="raw", cutoff_lvl is ignored.

If no value is provided, the function uses the default values associated with each cutoff method.

Value

A list containing the robustly estimated parameters:

mean_IMCD_c	Estimated mean of the centers of the interval data.
mean_IMCD_r	Estimated mean of the ranges of the interval data.
cov_IMCD	Estimated covariance (scatter) matrix (int_cov) for the data.
final_z	Binary vector indicating the inclusion of each observation in the reweighted subset.
cutoff	The cutoff method used for reweighting.
cutoff_value	Cutoff value used for reweighting.
robust_dist	Robust distances (IMah_dist) for each observation.
farness_probs	Farness probabilities (if cutoff is set to "farness").

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Minimum Covariance Determinant Estimator and Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2604.26769. https://arxiv.org/abs/2604.26769

Adapted from https://github.com/frankp-0/fastMCD.

The case cutoff=="F-dist" is adapted from package CerioliOutlierDetection (https://cran.r-project.org/package=CerioliOutlierDetection).

Examples

```
# Example using creditcard dataset
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_IMCD <- IMCD(credit_card_int, floor(0.75*credit_card_int@NObs), "farness", 0.9)
```

intCars — *Cars Dataset*

Description

This dataset contains interval data of car specifications, including min-max values. It is composed of 5 variables: Engine Capacity, Top Speed, Acceleration, Price and Class. The aggregation of the microdata was done by car model.

Usage

```
data(intCars)
```

Format

A list with the following components:

- `min_max` A data frame with 27 rows and 9 columns. It contains the lower and upper bounds for each variable.
- `intData` An intData object with 27 interval-valued observations and 4 variables. The variable "Price" was log-transformed into "lnPrice". The microdata are not available, thus the default parameters of the latent distributions were used assuming a uniform distribution.

References

This data was retrieved from the `MAINT.Data` package, available at https://cran.r-project.org/package=MAINT.Data.

Examples

```
data(intCars)
head(intCars$min_max)
head(intCars$intData)
```

intData — *Interval Data Constructor*

Description

Constructs an interval data object.

Usage

```
intData(
  Data,
  Seq = c("AllLb_AllUb", "AllCen_AllRng", "LbUb_VarbyVar", "CenRng_VarbyVar"),
  LatentParam = NULL,
  LatentCase = c("U_id_symmetric", "U_id", "General"),
  LatentDist = c("Unif", "Triang", "TNorm", "InvTri", "Beta", "KDE", "Degenerated"),
  TriangParam = 0,
  BetaParam.a = 1,
  BetaParam.b = 1,
  Umicro = NULL,
  estimate.DistParam = FALSE,
  VarNames = NULL,
  ObsNames = row.names(Data),
  NbMicroUnits = integer(0)
)
```

Arguments

`Data` A data frame or matrix containing the data.

`Seq` Format of macrodata if it is a data frame or matrix. Available options are:

- "AllLb_AllUb": All lower bounds followed by all upper bounds, in the same variable order.
- "AllCen_AllRng": All Centers followed by all Ranges, in the same variable order.
- "LbUb_VarbyVar": Lower bounds followed by upper bounds, variable by variable.
- "CenRng_VarbyVar": Centers followed by Ranges, variable by variable.

`LatentParam` A list with the parameters of the latent variables.

`LatentCase` A string specifying which of the three scenarios applies to the latent variables:

- "General": The case where the latent variables do not have any nice properties.
- "U_id": The case where the latent variables are identically distributed.
- "U_id_symmetric": The case where the latent variables are identically distributed and symmetric.

Defaults to "U_id_symmetric".

`LatentDist` A string or vector of strings specifying the distribution(s) of the latent variables. If the variables are identically distributed it can be one of ("Unif","Triang","TNorm","InvTri","Beta"," if not a vector must be provided with the distribution for each variable.

`TriangParam` Mode of the triangular distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is 0.

`BetaParam.a` Parameter alpha of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is 1.

BetaParam.b — Parameter beta of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is 1.

Umicro — Latent microdata observations. Needed if LatentDist="KDE" or estimate.DistParam=TRUE.

estimate.DistParam — Logical parameter indicating if estimation of the parameters of the latent distributions should be performed. Can only be set to TRUE if LatentCase="General". The default is FALSE.

VarNames — A character vector of variable names.

ObsNames — A character vector of observation names.

NbMicroUnits — An integer specifying the number of micro units.

Value

An object of class intData.

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Adapted from package MAINT.Data (https://cran.r-project.org/package=MAINT.Data).

---

intData-class — *Interval Data Class*

Description

A class to represent interval data.

Slots

Centers — A data frame of centers of the intervals.

Ranges — A data frame of ranges of the intervals.

LatentParam — A list with the parameters of the latent variables.

LatentCase — A string specifying which of the three scenarios applies to the latent variables:

- "General": The case where the latent variables do not have any nice properties.
- "U_id": The case where the latent variables are identically distributed.
- "U_id_symmetric": The case where the latent variables are identically distributed and symmetric.

Defaults to "U_id_symmetric".

LatentDist — A string or vector of strings specifying the distribution(s) of the latent variables. If the variables are identically distributed it can be one of ("Unif","Triang","TNorm","InvTri","Beta","KDE","Degenerate if not, it is a vector with the distribution for each variable.

`ObsNames` A character vector of observation names.

`VarNames` A character vector of variable names.

`NObs` A numeric value indicating the number of observations.

`NIVar` A numeric value indicating the number of interval variables.

`NbMicroUnits` An integer indicating the number of micro units.

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Adapted from package MAINT.Data (https://cran.r-project.org/package=MAINT.Data).

`int_cov` *Interval-valued Covariance*

Description

Calculate the interval-valued covariance matrix based on the covariance matrices of the centers and ranges or data.

Usage

```
int_cov(
  data = NULL,
  sigma_cc = NULL,
  sigma_rr = NULL,
  sigma_cr = NULL,
  LatentParam = NULL,
  LatentCase = c("U_id_symmetric", "U_id", "General")
)
```

Arguments

`data`	An intData object containing the macrodata/interval data.
`sigma_cc`	Covariance matrix of the centers.
`sigma_rr`	Covariance matrix of the ranges.
`sigma_cr`	Covariance matrix between the centers and ranges.
`LatentParam`	A list with the parameters of the latent variables.
`LatentCase`	A string specifying which of the three scenarios applies to the latent variables: • `"General"`: The case where the latent variables do not have any nice properties. • `"U_id"`: The case where the latent variables are identically distributed. • `"U_id_symmetric"`: The case where the latent variables are identically distributed and symmetric. Defaults to `"U_id_symmetric"`.

Details

This function calculates the interval-valued covariance matrix, $\boldsymbol{\Sigma}_B$, based on the covariance matrices of the centers, $\boldsymbol{\Sigma}_{CC}$, ranges, $\boldsymbol{\Sigma}_{RR}$, and the covariance matrix between the centers and ranges, $\boldsymbol{\Sigma}_{CR} = \boldsymbol{\Sigma}_{RC}^\top$. The covariance matrix is defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:

$$\boldsymbol{\Sigma}_B = \boldsymbol{\Sigma}_{CC} + \delta \boldsymbol{\Sigma}_{RR},$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.

- `"U_id"`: The latent variables are identically distributed:

$$\boldsymbol{\Sigma}_B = \boldsymbol{\Sigma}_{CC} + \delta \boldsymbol{\Sigma}_{RR} + \frac{\mathbb{E}(U)}{2} \left(\boldsymbol{\Sigma}_{CR} + \boldsymbol{\Sigma}_{RC} \right),$$

  where $\delta = \mathbb{E}(U^2)/4$ and $\mathbb{E}(U)$ are the parameters of the latent variables.

- `"General"`: The latent variables do not have any nice properties:

$$\boldsymbol{\Sigma}_B = \boldsymbol{\Sigma}_{CC} + \frac{1}{4} \mathfrak{E}_{UU} \bullet \boldsymbol{\Sigma}_{RR} + \frac{1}{2} \boldsymbol{\Sigma}_{CR} \boldsymbol{\Psi} + \frac{1}{2} \boldsymbol{\Psi} \boldsymbol{\Sigma}_{RC}$$

  where:

  - $\boldsymbol{\Psi} = \text{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $[\mathfrak{E}_{UU}]_{ij} = \mathcal{E}(U_i, U_j)$, $i \neq j$, with $\mathcal{E}(U_i, U_j) = \int_0^1 F_{U_i}^{-1}(t) F_{U_j}^{-1}(t)\, dt$,
  - $[\mathfrak{E}_{UU}]_{ii} = \mathbb{E}(U_i^2)$, $i, j = 1, \ldots, p$,
  - $\bullet$ denotes the Schur (or entrywise) product of matrices.

Value

The symbolic covariance matrix.

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_cov<-int_cov(credit_card_int)
```

`int_cov_z` *Sample Interval-valued Covariance*

Description

Calculate the interval-valued covariance matrix in function of z

Usage

```
int_cov_z(z, data)
```

Arguments

`z`	A vector of 0 and 1, indicating which observations should be considered for the calculation
`data`	An intData object containing the macrodata/interval data

Details

Let $\boldsymbol{z} \in \{0,1\}^n$ be a vector indicating which m observations are "active". This function calculates the sample interval-valued covariance matrix in function of $\boldsymbol{z}$: $\boldsymbol{S}_B(\boldsymbol{z})$. Let $\boldsymbol{C}$, $\boldsymbol{R}$ be the matrices of centers and ranges, respectively. Additionally, set:

$$\overline{\boldsymbol{c}}_B(\boldsymbol{z}) = \frac{1}{m}\boldsymbol{C}^\top \boldsymbol{z}, \qquad \overline{\boldsymbol{r}}_B(\boldsymbol{z}) = \frac{1}{m}\boldsymbol{R}^\top \boldsymbol{z}.$$

The sample interval-valued covariance matrix is obtained according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:

$$\boldsymbol{S}_B(\boldsymbol{z}) = \left(\frac{1}{m}\sum_{h=1}^{n} z_h \boldsymbol{c}_h \boldsymbol{c}_h^\top\right) - \overline{\boldsymbol{c}}_B(\boldsymbol{z})\overline{\boldsymbol{c}}_B(\boldsymbol{z})^\top + \left(\frac{\delta}{m}\sum_{h=1}^{n} z_h \boldsymbol{r}_h \boldsymbol{r}_h^\top\right) - \delta \overline{\boldsymbol{r}}_B(\boldsymbol{z})\overline{\boldsymbol{r}}_B(\boldsymbol{z})^\top,$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.

- `"U_id"`: The latent variables are identically distributed:

$$\boldsymbol{S}_B(\boldsymbol{z}) = \left(\frac{1}{m}\sum_{h=1}^{n} z_h \boldsymbol{c}_h \boldsymbol{c}_h^\top\right) - \overline{\boldsymbol{c}}_B(\boldsymbol{z})\overline{\boldsymbol{c}}_B(\boldsymbol{z})^\top + \left(\frac{\delta}{m}\sum_{h=1}^{n} z_h \boldsymbol{r}_h \boldsymbol{r}_h^\top\right) - \delta \overline{\boldsymbol{r}}_B(\boldsymbol{z})\overline{\boldsymbol{r}}_B(\boldsymbol{z})^\top + \left(\frac{\mathbb{E}(U)}{2m}\sum_{h=1}^{n} z_h \boldsymbol{c}_h \boldsymbol{r}_h^\top\right) - \frac{\mathbb{E}(U)}{2}$$

  where $\delta = \mathbb{E}(U^2)/4$ and $\mathbb{E}(U)$ are the parameters of the latent variables.

- `"General"`: The latent variables do not have any nice properties:

$$\boldsymbol{S}_B(\boldsymbol{z}) = \left(\frac{1}{m}\sum_{h=1}^{n} z_h \boldsymbol{c}_h \boldsymbol{c}_h^\top\right) - \overline{\boldsymbol{c}}_B(\boldsymbol{z})\overline{\boldsymbol{c}}_B(\boldsymbol{z})^\top + \left(\frac{1}{4m}\mathfrak{E}_{UU} \bullet \sum_{h=1}^{n} z_h \boldsymbol{r}_h \boldsymbol{r}_h^\top\right) - \frac{1}{4}\mathfrak{E}_{UU} \bullet \left[\overline{\boldsymbol{r}}_B(\boldsymbol{z})\overline{\boldsymbol{r}}_B(\boldsymbol{z})^\top\right] + \left(\frac{1}{2m}\sum_{h=1}^{n} z_h$$

  where:

  - $\boldsymbol{\Psi} = \mathrm{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $[\mathfrak{E}_{UU}]_{ij} = \mathcal{E}(U_i, U_j)$, $i \neq j$, with $\mathcal{E}(U_i, U_j) = \int_0^1 F_{U_i}^{-1}(t) F_{U_j}^{-1}(t)\, dt$,
  - $[\mathfrak{E}_{UU}]_{ii} = \mathbb{E}(U_i^2)$, $i, j = 1, \ldots, p$,
  - $\bullet$ denotes the Schur (or entrywise) product of matrices.

Value

The symbolic covariance matrix

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Minimum Covariance Determinant Estimator and Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2604.26769. https://arxiv.org/abs/2604.26769

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

z <- rep(1, nrow(credit_card_int))
credit_card_cov<-int_cov_z(z,credit_card_int)
```

---

int_mean_z *Sample Mean*

Description

Calculate the mean of X in function of z

Usage

```
int_mean_z(z, X)
```

Arguments

z	A vector of 0 and 1, indicating which observations should be considered for the calculation
X	A matrix where the rows correspond to observations and the columns to variables

Details

This function calculates the mean of $\boldsymbol{X}$ in function of $\boldsymbol{z}$. If $\boldsymbol{z}$ is a vector of 0 and 1, the mean is calculated for the m observations that are equal to 1:

$$\bar{\boldsymbol{x}}(\boldsymbol{z}) = \frac{1}{m}\boldsymbol{X}^\top \boldsymbol{z}.$$

Value

A vector where each element is the mean for each variable

Examples

```
n <- 100
p <- 4
X <- matrix(rnorm(n * p), ncol = p)
#if we consider all the observations the result obtained is the same as colMeans()
z <- c(rep(1, n))
int_mean_z(z, X)
colMeans(X)
```

---

int_outliers *Outlier Detection for Interval-Valued Data Based on Robust Distances*

Description

Identifies potential outliers in interval-valued data using robust distance-based methods with customizable cutoff criteria.

Usage

```
int_outliers(
  robust_dist,
  cutoff = c("farness", "adjbox", "chi-squared", "F-dist"),
  cutoff_lvl = NULL,
  p = NULL,
  z = NULL
)
```

Arguments

`robust_dist` A numeric vector containing the robust distances for each observation.

`cutoff` A character string specifying the method for setting the outlier cutoff threshold. Options include:

- `"chi-squared"`: Outliers are identified based on a specified Chi-Squared quantile.
- `"adjbox"`: Uses adjusted boxplot statistics (from `robustbase`) to classify outliers.
- `"F-dist"`: Applies a cutoff derived from the F and Beta distributions for robust outlier detection.
- `"farness"`: Identifies outliers based on a "farness" threshold, determined by the robust distance distribution.

Default is `"farness"`.

`cutoff_lvl` A numeric value specifying the level of the cutoff to be used.

- If `cutoff="chi-squared"`, `cutoff_lvl` is the quantile of the Chi-squared distribution (default is 0.975).

- If `cutoff="adjbox"`, `cutoff_lvl` is the coefficient for the adjusted boxplot (default is 1.5).
- If `cutoff="F-dist"`, `cutoff_lvl` is the significance level for identifying outliers (default is 0.95).
- If `cutoff="farness"`, `cutoff_lvl` represents the threshold for farness, with a default of 0.99.

If no value is provided, the function uses the default values associated with each cutoff method.

`p` The number of variables in the data. Required for `"chi-squared"` and `"F-dist"` cutoff methods.

`z` A binary vector indicating the subset of observations used for initial robust estimation. Required for the `"F-dist"` cutoff method.

Details

This function classifies observations as outliers based on robust distances and user-defined cutoff methods. It supports various approaches, including Chi-Squared quantiles, adjusted boxplots, F distribution quantiles, and farness probabilities.

Value

A list with the following components:

`outliers_names` Character vector of names for observations classified as outliers.

`is_outlier` Logical vector indicating whether each observation is an outlier (TRUE) or not (FALSE).

`cutoff` The cutoff method used for detecting outliers.

`cutoff_value` Cutoff value used for detecting outliers.

`farness_probs` Numeric vector of farness probabilities for each observation (only if `cutoff` is set to `"farness"`).

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Minimum Covariance Determinant Estimator and Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2604.26769. https://arxiv.org/abs/2604.26769

Case `cutoff=="F-dist"` is adapted from package `CerioliOutlierDetection` (https://cran.r-project.org/package=CerioliOutlierDetection).

Examples

```
# Example of detecting outliers using robust distances
set.seed(42)
robust_dist <- abs(rnorm(100))
result <- int_outliers(robust_dist, cutoff="chi-squared", p=5)

# Example using creditcard dataset
```

```
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_IMCD <- IMCD(credit_card_int, floor(0.75*credit_card_int@NObs), "farness", 0.9)
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist, "farness", 0.9)
```

`KL_divergence` *Kullback-Leibler (KL) Divergence*

Description

Computes the Kullback-Leibler (KL) divergence between an estimated covariance matrix and the ground truth. Assumes normal multivariate distributions.

Usage

```
KL_divergence(est_cov, ground_truth_cov)
```

Arguments

`est_cov` Estimated covariance matrix.

`ground_truth_cov` Ground truth covariance matrix.

Details

The KL divergence between two p-dimensional Gaussians $\mathcal{N}(\boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}})$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by:

$$\frac{1}{2}\left(\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\Sigma}}) + \log\left(\frac{\det(\boldsymbol{\Sigma})}{\det(\hat{\boldsymbol{\Sigma}})}\right) - p\right),$$

where $\hat{\boldsymbol{\Sigma}}$ and $\boldsymbol{\Sigma}$ are the estimated and ground truth covariance matrices, respectively.

Value

KL divergence between the two matrices.

References

Yufeng Zhang, Wanwei Liu, Zhenbang Chen, Ji Wang, and Kenli Li. On the properties of Kullback-Leibler divergence between multivariate gaussian distributions, 2023. https://arxiv.org/abs/2102.05485

`LatentCase` *Latent Case Method for intData*

Description

Latent Case Method for intData

Usage

```
LatentCase(Sdt)

## S4 method for signature 'intData'
LatentCase(Sdt)
```

Arguments

Sdt	An object of class intData.

Value

A character with the latent case.

`LatentDist` *Latent Distribution Method for intData*

Description

Latent Distribution Method for intData

Usage

```
LatentDist(Sdt)

## S4 method for signature 'intData'
LatentDist(Sdt)
```

Arguments

Sdt	An object of class intData.

Value

A character with the latent distribution(s).

LatentParam — *Latent Parameters Method for intData*

Description

Latent Parameters Method for intData

Usage

```
LatentParam(Sdt)

## S4 method for signature 'intData'
LatentParam(Sdt)
```

Arguments

`Sdt`	An object of class intData.

Value

A list with the latent parameters.

LogRanges — *LogRanges Method for intData*

Description

LogRanges Method for intData

Usage

```
LogRanges(Sdt)

## S4 method for signature 'intData'
LogRanges(Sdt)
```

Arguments

`Sdt`	An object of class intData.

Value

A `data.frame` containing the logarithms of the ranges.

`LowerBounds` *Lower Bounds Method for intData*

Description

Lower Bounds Method for intData

Usage

```
LowerBounds(Sdt)

## S4 method for signature 'intData'
LowerBounds(Sdt)
```

Arguments

`Sdt`	An object of class intData.

Value

A `data.frame` containing the lower bounds of the intervals.

`Mallows_dist` *Mallows Distance*

Description

Calculate the squared Mallows distance between all rows in data and the barycenter.

Usage

```
Mallows_dist(data, mean_c = NULL, mean_r = NULL)
```

Arguments

`data`	An intData object containing the macrodata/interval data
`mean_c`	The mean vector of the centers
`mean_r`	The mean vector of the ranges

Details

The squared Mallows distance is defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:

$$d_M(\boldsymbol{x})^2 = (\boldsymbol{c} - \boldsymbol{\mu}_C)^\top (\boldsymbol{c} - \boldsymbol{\mu}_C) + \delta(\boldsymbol{r} - \boldsymbol{\mu}_R)^\top (\boldsymbol{r} - \boldsymbol{\mu}_R),$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.
- `"U_id"`: The latent variables are identically distributed:

$$d_M(\boldsymbol{x})^2 = (\boldsymbol{c} - \boldsymbol{\mu}_C)^\top (\boldsymbol{c} - \boldsymbol{\mu}_C) + \delta(\boldsymbol{r} - \boldsymbol{\mu}_R)^\top (\boldsymbol{r} - \boldsymbol{\mu}_R) + \mathbb{E}(U)(\boldsymbol{c} - \boldsymbol{\mu}_C)^\top (\boldsymbol{r} - \boldsymbol{\mu}_R),$$

  where $\delta = \mathbb{E}(U^2)/4$ and $\mathbb{E}(U)$ are the parameter of the latent variables.
- `"General"`: The latent variables do not have any nice properties:

$$d_M(\boldsymbol{x})^2 = (\boldsymbol{c} - \boldsymbol{\mu}_C)^\top (\boldsymbol{c} - \boldsymbol{\mu}_C) + (\boldsymbol{r} - \boldsymbol{\mu}_R)^\top \boldsymbol{\Delta}(\boldsymbol{r} - \boldsymbol{\mu}_R) + (\boldsymbol{c} - \boldsymbol{\mu}_C)^\top \boldsymbol{\Psi}(\boldsymbol{r} - \boldsymbol{\mu}_R),$$

  where:
  - $\boldsymbol{\Psi} = \text{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $\boldsymbol{\Delta} = \text{diag}(\mathbb{E}(U_1^2), \ldots, \mathbb{E}(U_p^2))/4$.

Value

A vector with the squared Mallows distance of each observation.

References

Oliveira, M. R., Pinheiro, D., & Oliveira, L. (2025). Location and association measures for interval-valued data based on Mallows' distance. arXiv preprint arXiv:2407.05105. https://arxiv.org/abs/2407.05105

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_dist<-Mallows_dist(credit_card_int)
```

---

micro2intData *Aggregate Microdata into Interval-Valued Data*

Description

Aggregates microdata from a data frame into interval-valued data using various criteria and latent distribution settings.

Usage

```
micro2intData(
  MicDtDF,
  agrby,
  agrcrt = "minmax",
  LatentParam = NULL,
  LatentCase = c("U_id_symmetric", "U_id", "General"),
 LatentDist = c("Unif", "Triang", "TNorm", "InvTri", "Beta", "KDE", "Degenerated"),
  TriangParam = 0,
  BetaParam.a = 1,
  BetaParam.b = 1,
  estimate.DistParam = FALSE
)
```

Arguments

`MicDtDF` A data frame containing the microdata. All columns should be numeric.

`agrby` A factor used to specify the grouping of the microdata for aggregation.

`agrcrt` A string or numeric vector of length 2 specifying the aggregation criterion. The default is `"minmax"`, which takes the minimum and maximum values for each variable. If a numeric vector is provided, it should specify the lower and upper percentiles for aggregation (e.g., `c(0.05, 0.95)`).

`LatentParam` Optional latent parameter used for certain types of latent distributions.

`LatentCase` A string specifying which of the three scenarios applies to the latent variables:

- `"General"`: The case where the latent variables do not have any nice properties.
- `"U_id"`: The case where the latent variables are identically distributed.
- `"U_id_symmetric"`: The case where the latent variables are identically distributed and symmetric.

Defaults to `"U_id_symmetric"`.

`LatentDist` A string or vector of strings specifying the distribution(s) of the latent variables. If the variables are identically distributed it can be one of `("Unif","Triang","TNorm","InvTri","Beta","` if not a vector must be provided with the distribution for each variable. The default is `"KDE"` if `LatentCase="General"`.

`TriangParam` Mode of the triangular distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `0`.

`BetaParam.a` Parameter alpha of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `1`.

`BetaParam.b` Parameter beta of the Beta distribution. If the latent variables are identically distributed, it is only necessary to provide a number, if not a vector is needed. The default is `1`.

`estimate.DistParam`
: Logical parameter indicating if estimation of the parameters of the latent distributions should be performed. Can only be set to TRUE if `LatentCase="General"`. The default is `FALSE`.

Details

This function processes a data frame of microdata and aggregates it into interval-valued data according to the specified grouping factor and aggregation criteria. It can handle different latent distribution cases and parameter settings.

If some rows contain invalid (non-finite or missing) values, those rows are removed before aggregation. If all rows in the resulting interval-valued data are degenerate (i.e., the lower bound equals the upper bound), the function will return `NULL`.

Value

An intData object containing the aggregated interval-valued data, or `NULL` if all units lead to degenerate intervals.

References

Adapted from package `MAINT.Data` (https://cran.r-project.org/package=MAINT.Data).

Examples

```
data(creditcard)
CreditCard_microdata <- creditcard$microdata
credit_agrby<-factor(paste(CreditCard_microdata$Name,CreditCard_microdata$Month,sep = "_"))
credit_agr<-micro2intData(CreditCard_microdata[,3:7],credit_agrby,LatentCase = "General")
```

`names,intData-method` *Variable Names Method for intData*

Description

Variable Names Method for intData

Usage

```
## S4 method for signature 'intData'
names(x)
```

Arguments

`x`
: An object of class intData.

Value

A character vector of variable names.

NbMicroUnits *Number of Micro Units Method for intData*

Description

Number of Micro Units Method for intData

Usage

```
NbMicroUnits(x)

## S4 method for signature 'intData'
NbMicroUnits(x)
```

Arguments

x	An object of class intData.

Value

An integer specifying the number of micro units.

ncol,intData-method *Number of Columns Method for intData*

Description

Number of Columns Method for intData

Usage

```
## S4 method for signature 'intData'
ncol(x)
```

Arguments

x	An object of class intData.

Value

The number of columns.

nrow,intData-method — *Number of Rows Method for intData*

Description

Number of Rows Method for intData

Usage

```
## S4 method for signature 'intData'
nrow(x)
```

Arguments

x	An object of class intData.

Value

The number of rows.

plot,intData,intData-method — *Plot Method for Two intData Objects*

Description

Plots one intData object against another, with options to visualize the intervals as crosses or rectangles.

Plots a single intData object, either in a vertical or horizontal layout.

Usage

```
## S4 method for signature 'intData,intData'
plot(
  x,
  y,
  type = c("crosses", "rectangles", "crosses2"),
  append = FALSE,
  palette = rainbow(x@NObs),
  ...
)

## S4 method for signature 'intData,missing'
plot(
  x,
```

```
  casen = NULL,
  layout = c("vertical", "horizontal"),
  append = FALSE,
  ...
)
```

Arguments

`x`	An intData object.
`y`	An intData object to plot on the y-axis.
`type`	The type of plot to generate: "crosses" or "rectangles" or "crosses2". Default is "crosses".
`append`	Logical, if `TRUE`, the plot is added to the current plot.
`palette`	A vector with colors for each observation.
`...`	Additional graphical parameters.
`casen`	A vector specifying the case numbers to plot. Default is `NULL`.
`layout`	The layout of the plot: "vertical" or "horizontal".

Value

A plot showing the relationship between the two intData objects.

A plot showing the intervals of the intData object.

`plot_dist_dist` *Distance-Distance plot for interval-valued data.*

Description

Distance-Distance plot for interval-valued data.

Usage

```
plot_dist_dist(
  class_dist,
  class_cutoff = NULL,
  class_cutoff_label = NULL,
  rob_dist,
  rob_cutoff = NULL,
  rob_cutoff_label = NULL,
  obs_names = NULL,
  ggplotly = TRUE,
  color_class = NULL,
  color_label = NULL,
  palette = NULL,
  shape_class = NULL,
```

```
  shape_label = NULL,
  label_obs = NULL
)
```

Arguments

- `class_dist` — A numeric vector containing the classical distances for each observation.
- `class_cutoff` — Numeric. The cutoff value for the classical distances.
- `class_cutoff_label` — Character. Label for the classical cutoff. If NULL (default), no legend for the classical cutoff is shown.
- `rob_dist` — A numeric vector containing the robust distances for each observation.
- `rob_cutoff` — Numeric. The cutoff value for the robust distances.
- `rob_cutoff_label` — Character. Label for the robust cutoff. If NULL (default), no legend for the robust cutoff is shown.
- `obs_names` — A character vector containing the names of the observations. If NULL (default), the names are taken from the names of class_dist.
- `ggplotly` — Logical. If `TRUE` (default), the plot is converted to an interactive plotly::plotly object.
- `color_class` — A vector indicating the color class of each observation. If NULL (default), all points have the same color.
- `color_label` — Character. Label for the color class. If NULL (default), no legend for the color class is shown.
- `palette` — A vector with colors for each color class. If NULL (default), default ggplot2::ggplot2 colors are used.
- `shape_class` — A vector indicating the shape class of each observation. If NULL (default), all points have the same shape.
- `shape_label` — Character. Label for the shape class. If NULL (default), no legend for the shape class is shown.
- `label_obs` — A vector with the names of the observations to be labeled in the plot when `ggplotly` = `FALSE`. Default is NULL.

Value

Returns a Distance-Distance plot that displays the classical distances against the robust distances for each observation, highlighting outliers.

Examples

```
#Create intData object
data(creditcard)
credit_card_int <- creditcard$intData

#Estimate the mean and covariance matrix
credit_card_IMCD<-IMCD(credit_card_int, floor(nrow(credit_card_int)*0.75), "farness", 0.9)
```

```
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                      p=credit_card_int@NIVar, cutoff_lvl = 0.9)

#Plot Distance-Distance plot
class_dist <- IMah_dist(credit_card_int, z=rep(1,credit_card_int@NObs))
class_outliers <- int_outliers(class_dist,cutoff = "adjbox",p=p,cutoff_lvl = 1.5)
credit_card_is_outliers <- as.character(credit_card_outliers$is_outlier)
credit_card_is_outliers[credit_card_outliers$is_outlier] <- "Outlier"
credit_card_is_outliers[!credit_card_outliers$is_outlier] <- "Inlier"
plot_dist_dist(class_dist, class_outliers$cutoff_value[2], "1.5 adjusted boxplot",
         credit_card_IMCD$robust_dist, credit_card_outliers$cutoff_value, "0.9 farness",
             color_class = credit_card_is_outliers, palette = c("grey50", "red"))
```

`plot_interval_dist` *Interval-Mahalanobis distance plot for interval-valued data.*

Description

Interval-Mahalanobis distance plot for interval-valued data.

Usage

```
plot_interval_dist(
  dist,
  cutoff = NULL,
  cutoff_label = NULL,
  obs_names = NULL,
  sort.obs = TRUE,
  color_class = NULL,
  color_label = NULL,
  palette = NULL,
  shape_class = NULL,
  shape_label = NULL,
  label_obs = NULL
)
```

Arguments

`dist`	A numeric vector containing the Interval-Mahalanobis distances for each observation.
`cutoff`	A numeric vector containing cutoff values to be displayed as horizontal lines.
`cutoff_label`	A character vector containing labels for each cutoff. If NULL (default), default labels are generated.
`obs_names`	A character vector containing the names of the observations. If NULL (default), the names are taken from the names of dist.
`sort.obs`	Logical. If `TRUE` (default), observations are sorted according to their distances.

color_class	A vector indicating the color class of each observation. If NULL (default), all points have the same color.
color_label	Character. Label for the color class. If NULL (default), no legend for the color class is shown.
palette	A vector with colors for each color class. If NULL (default), default ggplot2::ggplot2 colors are used.
shape_class	A vector indicating the shape class of each observation. If NULL (default), all points have the same shape.
shape_label	Character. Label for the shape class. If NULL (default), no legend for the shape class is shown.
label_obs	A vector with the names of the observations to be labeled in the plot. If NULL (default), no labels are shown and x-axis labels are displayed.

Value

Returns a plot that displays the Interval-Mahalanobis distances for each observation, highlighting outliers based on specified cutoffs.

Examples

```
#Create intData object
data(creditcard)
credit_card_int <- creditcard$intData

#Estimate the mean and covariance matrix
credit_card_IMCD<-IMCD(credit_card_int, floor(nrow(credit_card_int)*0.75), "farness", 0.9)
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                            p=credit_card_int@NIVar, cutoff_lvl = 0.9)
credit_card_is_outliers <- as.character(credit_card_outliers$is_outlier)
credit_card_is_outliers[credit_card_outliers$is_outlier] <- "Outlier"
credit_card_is_outliers[!credit_card_outliers$is_outlier] <- "Inlier"

#Plot Interval-Mahalanobis distance plot
plot_interval_dist(credit_card_IMCD$robust_dist,
                   cutoff = credit_card_outliers$cutoff_value,
                   cutoff_label = c("0.9 farness"),
                   obs_names = rownames(credit_card_int),
                   sort.obs = FALSE,
                   color_class = credit_card_is_outliers,
                   palette = c("grey50", "red"))
```

---

print,summaryintData-method

Print Method for Summary intData

Description

Print Method for Summary intData

Usage

```
## S4 method for signature 'summaryintData'
print(x, ...)
```

Arguments

x	An object of class summaryintData.
...	Additional arguments passed to print.

Value

The object itself, returned invisibly. Called for its side effects (printing).

Ranges — *Ranges Method for intData*

Description

Ranges Method for intData

Usage

```
Ranges(Sdt)

## S4 method for signature 'intData'
Ranges(Sdt)
```

Arguments

Sdt	An object of class intData.

Value

A data.frame containing the ranges of the intervals.

`row.names,intData-method`
Row.Names Method for intData

Description

Row.Names Method for intData

Usage

```
## S4 method for signature 'intData'
row.names(x)
```

Arguments

x	An object of class intData.

Value

A character vector of row names.

`rownames,intData-method`
Row Names Method for intData

Description

Row Names Method for intData

Usage

```
## S4 method for signature 'intData'
rownames(x)
```

Arguments

x	An object of class intData.

Value

A character vector of row names.

show,intData-method — *Show Method for intData*

Description

Show Method for intData

Show Method for Summary intData

Usage

```
## S4 method for signature 'intData'
show(object)

## S4 method for signature 'summaryintData'
show(object)
```

Arguments

`object`	An object of class `summaryintData`.

Value

The object itself, returned invisibly. Called for its side effects (printing).

spotify_tracks — *Spotify Tracks Dataset*

Description

This dataset contains interval data of Spotify tracks' audio features, including min-max values and trimmed intervals, as well as the microdata. It is composed of 11 audio features: duration, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and popularity. The aggregation of the microdata was done by track genre.

Usage

```
data(spotify_tracks)
```

Format

A list with the following components:

`microdata` A data frame with `81033` rows and `20` columns. It contains the microdata, with individual measurements of each variable for all observations.

`microdata_transformed` A data frame with `81033` rows and `20` columns. It contains the transformed microdata, with individual measurements of each variable for all observations. Logarithmic transformations were applied to "loudness" and "tempo". "duration_ms" in milliseconds was converted to "duration" in minutes. "popularity" was scaled to the range `[0,1]`.

`intData_minmax` An intData object with `111` interval-valued observations and `11` variables, constructed using min-max aggregation based on the transformed microdata.

`intData_trimmed` An intData object with `111` interval-valued observations and `11` variables, constructed using trimmed aggregation (`1\%` trimming) based on the transformed microdata.

References

This data was retrieved from Kaggle, available at `https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset`.

Examples

```
data(spotify_tracks)
head(spotify_tracks$intData_minmax)
head(spotify_tracks$intData_trimmed)
head(spotify_tracks$microdata)
head(spotify_tracks$microdata_transformed)
```

`summary,intData-method`

Summary Method for intData

Description

Summary Method for intData

Usage

```
## S4 method for signature 'intData'
summary(object)
```

Arguments

`object`	An object of class intData.

Value

An object of class `summaryintData`.

summaryintData-class *Summary Interval Data Class*

Description

A class to represent the summary of interval data.

Slots

`Centersumar` A table summarizing the centers.

`Rngsumar` A table summarizing the ranges.

SYMB.biplot *Symbolic Biplot for Interval-valued Data*

Description

Create a biplot for interval-valued symbolic data, visualizing the symbolic data as rectangles or crosses, with the first two variables on the x and y axes. The function allows customization of colors, fill colors, and outlier representation.

Usage

```
SYMB.biplot(
  data,
  type = c("rectangles", "crosses", "crosses2"),
  palette = rainbow(nrow(data)),
  fill_col = "gray50",
  is_outlier = NULL,
  ...
)
```

Arguments

`data`	An intData object containing the macrodata/interval data. The first two variables are used for the x and y axes.
`type`	The type of plot to generate: "rectangles", "crosses" or "crosses2". Default is "rectangles".
`palette`	A vector with colors for each observation. Default is `rainbow(nrow(data))`.
`fill_col`	If `type="rectangles"`, a vector with colors for the fill of each observation, or a single color for all observations. Default is "gray50".
`is_outlier`	A vector with logical values indicating if the observation is an outlier or not. It makes the line width of the outlying observations thicker. Default is NULL.
`...`	Additional graphical parameters.

Value

A biplot is drawn in the graphic window. The biplot shows the symbolic data as rectangles or crosses, with the first two variables on the x and y axes.

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

SYMB.biplot(credit_card_int[,c(3,5)])

# Highlight outliers in the biplot
credit_card_IMCD <- IMCD(credit_card_int, floor(0.75*credit_card_int@NObs), "farness", 0.9)
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist, "farness", 0.9)
outliers_colors<-rep('gray50',credit_card_int@NObs)
names(outliers_colors)<-rownames(credit_card_int)
outliers_colors[credit_card_outliers$outliers_names] = 'red'
SYMB.biplot(credit_card_int[,c(3,5)], palette = outliers_colors,
            is_outlier = credit_card_outliers$is_outlier)
```

SYMB.pairs.panels *Pairs-plot for Interval-valued Symbolic data.*

Description

Adapted from pairs.panels (R package "psych") shows a scatter plot of matrices, with bivariate symbolic scatter plots below the diagonal, variables' names on the diagonal, and all the symbolic correlations above the diagonal. Useful for descriptive statistics of symbolic objects described by interval variables.

Usage

```
SYMB.pairs.panels(
  data,
  type = c("rectangles", "crosses", "crosses2"),
  cex.cor = 2,
  corr = NULL,
  palette = rainbow(nrow(data)),
  fill_col = "gray50",
  is_outlier = NULL,
  ...
)
```

Arguments

data	An intData object containing the macrodata/interval data
type	The type of plot to generate: "rectangles" or "crosses" or "crosses2". Default is "rectangles".

cex.cor	Character expansion factor
corr	A matrix with the symbolic correlations; if not provided the upper panel is omitted
palette	A vector with colors for each observation.
fill_col	If type="rectangles", a vector with colors for the fill of each observation, or a single color for all observations. Default is "gray50".
is_outlier	A vector with logical values indicating if the observation is an outlier or not. It makes the line width of the outlying observations thicker. Default is NULL.
...	Additional graphical parameters.

Value

A scatter plot matrix is drawn in the graphic window. The lower off diagonal draws scatter plots, the diagonal variables' names, the upper off diagonal reports all the symbolic correlations.

Examples

```
data(creditcard)
credit_card_int <- creditcard$intData

credit_card_cov<-int_cov(credit_card_int)
credit_card_cor<-cov2cor(credit_card_cov)
SYMB.pairs.panels(credit_card_int,corr=credit_card_cor,labels=colnames(credit_card_int))

# Highlight outliers in the biplot
credit_card_IMCD <- IMCD(credit_card_int, floor(0.75*credit_card_int@NObs), "farness", 0.9)
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist, "farness", 0.9)
outliers_colors<-rep('gray50',credit_card_int@NObs)
names(outliers_colors)<-rownames(credit_card_int)
outliers_colors[credit_card_outliers$outliers_names] = 'red'
SYMB.pairs.panels(credit_card_int,corr=cov2cor(credit_card_IMCD$cov_IMCD),
                  palette = outliers_colors,labels=colnames(credit_card_int),
                  type = "rectangles",is_outlier = credit_card_outliers$is_outlier)
```

---

tail,intData-method *Tail Method for intData*

Description

Returns the last n rows of an intData object.

Usage

```
## S4 method for signature 'intData'
tail(x, n = min(nrow(x), 6L))
```

Arguments

`x`	An intData object.
`n`	The number of rows to return.

Value

A subset of the intData object.

`UpperBounds` *Upper Bounds Method for intData*

Description

Upper Bounds Method for intData

Usage

```
UpperBounds(Sdt)

## S4 method for signature 'intData'
UpperBounds(Sdt)
```

Arguments

`Sdt`	An object of class intData.

Value

A `data.frame` containing the upper bounds of the intervals.

`[,intData-method` *Subset an intData Object*

Description

Extract a subset of rows and columns from an intData object.

Usage

```
## S4 method for signature 'intData'
x[i, j, ..., drop = TRUE]
```

Arguments

`x`	An intData object.
`i`	Row indices or names to subset. Defaults to all rows.
`j`	Column indices or names to subset. Defaults to all columns.
`...`	Additional arguments (not used).
`drop`	Logical, passed to the underlying `[`. Defaults to `TRUE`.

Value

An intData object containing the specified subset of rows and columns.

Index