Package ‘AIDA’

September 4, 2026

Type Package

Title Analysis of Interval DAta

Version 0.2.1

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. (2026b) <doi:10.48550/arXiv.2604.26769>.
Explainable outlier detection is supported through Shapley value based decomposition of the squared robust Interval-Mahalanobis distance, allowing assessment of variable contributions to outlyingness (Loureiro et al. (2026a) <doi:10.48550/arXiv.2606.26307>).
Shapley interaction indices are also implemented, along with visualization tools to support interpretation of the results.

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

LazyData true

LazyDataCompression xz

VignetteBuilder knitr

Language en-US

Imports cellWise, cowplot, fmsb, ggbeeswarm, ggplot2, kde1d, MASS, methods

Depends R (>= 3.6)

Suggests CerioliOutlierDetection, corrplot, ggrepel, knitr, plotly, RColorBrewer, rmarkdown, robustbase, scales, testthat (>= 3.0.0)

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

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 2026-09-04 12:10:02 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.

`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 and package `imputeTS` (available at https://cran.r-project.org/package=imputeTS) was employed to deal with the remaining missing values using interpolation. 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/.

Loureiro, C.P., Oliveira, M.R., Brito, P., Oliveira, L. (2025). Air Quality Data Analysis with Symbolic Principal Components. In: Henriques-Rodrigues, L., Menezes, R., Machado, L.M., Faria, S., de Carvalho, M. (eds) New Frontiers in Statistics and Data Science. SPE 2021. Springer Proceedings in Mathematics & Statistics, vol 469. Springer, Cham. doi:10.1007/9783031689499_25

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

Either a vector of farness probabilities for each observation (if `cutoff_value` is not provided) or a list containing (if `cutoff_value` is provided):

`farness_probs`	A vector of farness probabilities for each observation. Values between 0 and 1.
`cutoff_value`	The cutoff distance value in the original distance scale corresponding to the specified `cutoff_value` probability.

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
credit_card_dist <- IMah_dist(credit_card_int)

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

---

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

  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", "Degenerated"), if not a vector must be provided with the distribution for each variable. The default is "Unif" if LatentCase="U_id_symmetric" or if Umicro is not provided, and "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.

Umicro
: Latent microdata observations. Needed if estimate.DistParam is TRUE or LatentDist is "KDE".

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:
  - $[\mathfrak{E}_{UU}]_{j\ell} = \mathcal{E}(U_j, U_\ell)$, $j \neq \ell$, with $\mathcal{E}(U_j, U_\ell) = \int_0^1 F_{U_j}^{-1}(t) F_{U_\ell}^{-1}(t)\, dt$, and $[\mathfrak{E}_{UU}]_{jj} = \mathbb{E}(U_j^2)$, $j, \ell = 1, \ldots, p$
  - $\mathbf{\Psi} = \mathrm{Diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$

Value

A list composed by:

LatentParam	A list with the parameters of the latent variables distribution.
TriangParam	The mode of the triangular distribution.
BetaParam.a	The alpha parameter of the Beta distribution.
BetaParam.b	The beta parameter of the Beta distribution.
LatentCase	The case of the latent variables.
LatentDist	The distribution 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

# Define grouping variable for microdata aggregation
credit_agrby <- paste(CreditCard_microdata$Name, CreditCard_microdata$Month, sep = "_")

# Obtain latent variables inherent to the macrodata (normalized to [-1,1])
credit_card_U <- get_latent_var(microdata = CreditCard_microdata[,3:7],
                                macrodata = CreditCard_min_max,
                                agrby = credit_agrby,
                                agrlevels = row.names(CreditCard_min_max),
                                Seq = "LbUb_VarbyVar")

# Obtain parameters of the latent variables
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_j, 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

$$V_j = C_j + U_j \frac{R_j}{2}, \quad j = 1, \ldots, p,$$

the **microdata** with U_j 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

# Define grouping variable for microdata aggregation
credit_agrby <- paste(CreditCard_microdata$Name, CreditCard_microdata$Month, sep = "_")

# Obtain latent variables inherent to the macrodata (normalized to [-1,1])
credit_card_U <- get_latent_var(microdata = CreditCard_microdata[,3:7],
                                macrodata = CreditCard_min_max,
                                agrby = 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`	(Optional) A vector of 0 and 1, indicating which observations should be considered for the calculation. If `z` is not `NULL`, `mean_c`, `mean_r`, and `cov` will be computed using only the observations with `z=1` (see `int_mean_z` and `int_cov_z`). Defaults to `NULL`.
`mean_c`	(Optional) A vector specifying the mean of centers. Defaults to `NULL`, in which case it will be computed using the `IMCD` function, if `z` is also `NULL`.
`mean_r`	(Optional) A vector specifying the mean of ranges. Defaults to `NULL`, in which case it will be computed using the `IMCD` function, if `z` is also `NULL`.
`cov`	(Optional) A covariance matrix. Defaults to `NULL`, in which case it will be computed using the `IMCD` function, if `z` is also `NULL`.

Details

The squared Interval-Mahalanobis distance between $\boldsymbol{x} = (\boldsymbol{c}^\top, \boldsymbol{r}^\top)^\top$ and the barycenter $\boldsymbol{\mu}_B = (\boldsymbol{\mu}_C^\top, \boldsymbol{\mu}_R^\top)^\top$ of a population with symbolic covariance matrix $\boldsymbol{\Sigma}_B$ (see `int_cov`) is defined according to the `LatentCase`:

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

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

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

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

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

  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:

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

  where:

- $\boldsymbol{\Psi} = \mathrm{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
- $[\mathfrak{E}_{UU}]_{j\ell} = \mathcal{E}(U_j, U_\ell)$, $j \neq \ell$, with $\mathcal{E}(U_j, U_\ell) = \int_0^1 F_{U_j}^{-1}(t) F_{U_\ell}^{-1}(t)\, dt$,
- $[\mathfrak{E}_{UU}]_{jj} = \mathbb{E}(U_j^2)$, $j, \ell = 1, \ldots, p$,
- $\bullet$ 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

# Compute squared Interval-Mahalanobis distance using IMCD estimates of mean and covariance
credit_card_dist <- IMah_dist(credit_card_int)
```

`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`	(Optional) A covariance matrix. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.

Details

The squared Interval-Mahalanobis distance between $\boldsymbol{x}_1 = (\boldsymbol{c}_1^\top, \boldsymbol{r}_1^\top)^\top$ and $\boldsymbol{x}_2 = (\boldsymbol{c}_2^\top, \boldsymbol{r}_2^\top)^\top$ of a population with symbolic covariance matrix $\boldsymbol{\Sigma}_B$ (see `int_cov`) is defined according to the `LatentCase`:

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

$$d_{\text{IMah}}(\boldsymbol{x}_1, \boldsymbol{x}_2)^2 = (\boldsymbol{c}_1 - \boldsymbol{c}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c}_1 - \boldsymbol{c}_2) + \delta(\boldsymbol{r}_1 - \boldsymbol{r}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r}_1 - \boldsymbol{r}_2),$$

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

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

$$\begin{aligned} d_{\text{IMah}}(\boldsymbol{x}_1, \boldsymbol{x}_2)^2 = (\boldsymbol{c}_1 - \boldsymbol{c}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c}_1 - \boldsymbol{c}_2) + \delta(\boldsymbol{r}_1 - \boldsymbol{r}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r}_1 - \boldsymbol{r}_2) \\ + \mathbb{E}(U)(\boldsymbol{c}_1 - \boldsymbol{c}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r}_1 - \boldsymbol{r}_2), \end{aligned}$$

  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:

$$\begin{aligned} d_{\text{IMah}}(\boldsymbol{x}_1, \boldsymbol{x}_2)^2 = (\boldsymbol{c}_1 - \boldsymbol{c}_2)^\top \boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c}_1 - \boldsymbol{c}_2) + \frac{1}{4}(\boldsymbol{r}_1 - \boldsymbol{r}_2)^\top \left(\boldsymbol{\mathfrak{E}}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1}\right)(\boldsymbol{r}_1 - \boldsymbol{r}_2) \\ + (\boldsymbol{c}_1 - \boldsymbol{c}_2)^\top \boldsymbol{\Sigma}_B^{-1} \boldsymbol{\Psi}(\boldsymbol{r}_1 - \boldsymbol{r}_2), \end{aligned}$$

  where:

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

If cov is not provided, it will be computed using the IMCD function. Additionally, if cov is set as the identity matrix, the computed distance is the Mallows distance between pairs of observations.

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% Chi-Squared quantile.
- `"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

# Obtain reweighted IMCD estimates using farness cutoff
credit_card_IMCD <- IMCD(credit_card_int,
                         m = floor(nrow(credit_card_int)*0.75),
                         cutoff = "farness",
                         cutoff_lvl = 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:

microdata 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$microdata)
head(intCars$intData)
```

---

intData *Interval Data Constructor*

Description

Constructs an interval data object.

Usage

```
intData(
  macrodata,
  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(macrodata),
  NbMicroUnits = integer(0)
)
```

Arguments

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

`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. Expects a list with a single number if `LatentCase` is "U_id_symmetric", a list of two numbers if `LatentCase` is "U_id", and a list of two matrices if `LatentCase` is "General".

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

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

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", "Degenerated"), if not a vector must be provided with the distribution for each variable. The default is "Unif" if `LatentCase="U_id_symmetric"` or if `Umicro` is not provided, and "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`.

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

`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 vector indicating the number of individual observations (microdata) aggregated by interval (macrodata).

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).

Examples

```
# Load microdat and macrodata
data(creditcard)
CreditCard_microdata <- creditcard$microdata
CreditCard_min_max <- creditcard$min_max

# Create an intData object using the min_max component of the dataset
# Assume a continuous uniform distribution for the latent variables
# This corresponds to LatentCase="U_id_symmetric"
# This is the default setting for the intData class
credit_card_int_unif <- intData(CreditCard_min_max,
                                Seq = "LbUb_VarbyVar",
                                VarNames = colnames(CreditCard_microdata)[3:7])
```

`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:

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

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", "Degenerated"), 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 vector indicating the number of individual observations (microdata) aggregated by interval (macrodata).

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. If `data` is provided, the covariance matrix is calculated based on the the sample covariance of the centers and ranges and the sample covariance between centers and ranges, and the parameters of the latent variables contained in the `intData` object. If `data` is not provided, the covariance matrix is calculated based on `sigma_cc`, `sigma_rr`, `sigma_cr`, `LatentParam`, and `LatentCase`.
`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. Expects a list with a single number if LatentCase is "U_id_symmetric", a list of two numbers if LatentCase is "U_id", and a list of two matrices if LatentCase is "General".

LatentCase
: A string specifying which of the three scenarios applies to the latent variables:
  - "U_id_symmetric": The case where the latent variables are identically distributed and symmetric.
  - "U_id": The case where the latent variables are identically distributed.
  - "General": The case where the latent variables do not have any nice properties.

  Defaults to "U_id_symmetric".

Details

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

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

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

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

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

  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:

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

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

The covariance matrix can be calculated either based on the covariance matrices of the centers and ranges or based on the data. If the data is provided, the covariance matrices are calculated using the sample covariance of the centers and ranges and the sample covariance between centers and ranges. For the robust estimation of the covariance matrix, see IMCD.

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:

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

  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:

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

  where:

  - $\boldsymbol{\Psi} = \mathrm{diag}(\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))$,
  - $[\mathfrak{E}_{UU}]_{j\ell} = \mathcal{E}(U_j, U_\ell)$, $j \neq \ell$, with $\mathcal{E}(U_j, U_\ell) = \int_0^1 F_{U_j}^{-1}(t)F_{U_\ell}^{-1}(t)\,dt$,
  - $[\mathfrak{E}_{UU}]_{jj} = \mathbb{E}(U_j^2)$, $j, \ell = 1, \ldots, p$,
  - $\bullet$ 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

# Compute sample interval-valued covariance matrix using the all the observations
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

# Compute robust distances using IMCD estimates of mean and covariance
credit_card_dist <- IMah_dist(credit_card_int)

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)
```

`int_Shapley` *Compute Shapley Values for Interval-valued Data*

Description

Outlier explanation based on Shapley values for interval-valued data. Decomposes the squared interval-valued Mahalanobis distance into additive outlyingness contributions of the variables.

Usage

```
int_Shapley(data, mean_c = NULL, mean_r = NULL, cov = NULL)
```

Arguments

`data`	An `intData` object containing the interval-valued dataset (macrodata).
`mean_c`	(Optional) A vector specifying the mean of centers. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`mean_r`	(Optional) A vector specifying the mean of ranges. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`cov`	(Optional) A covariance matrix. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.

Details

The Shapley value decomposes the squared Interval-Mahalanobis distance (see `IMah_dist`) into additive outlyingness contributions of the variables. Let $\boldsymbol{\mu}_B = (\boldsymbol{\mu}_C^\top, \boldsymbol{\mu}_R^\top)^\top$ be the barycenter and $\boldsymbol{\Sigma}_B$ the symbolic covariance matrix (see `int_cov`). The Shapley value of an interval-valued observation $\boldsymbol{x} = (\boldsymbol{c}^\top, \boldsymbol{r}^\top)^\top$, for the Interval-Mahalanobis distance, is defined according to the `LatentCase`:

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

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right] + \delta(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right],$$

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

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

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{x}) &= (\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right] + \delta(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] \\ &+ \frac{\mathbb{E}(U)}{2}(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] + \frac{\mathbb{E}(U)}{2}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],\end{aligned}$$

  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:

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{x}) &= (\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right] + \frac{1}{4}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\left(\mathfrak{C}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1}\right)(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] \\ &+ \frac{1}{2}(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Psi}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] + \frac{1}{2}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Psi}\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],\end{aligned}$$

  where:

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

Value

A matrix of Shapley values with row and column names corresponding to the rows and columns of the input data.

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Explainable Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2606.26307. https://arxiv.org/abs/2606.26307

Examples

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

# Compute Shapley values based on IMCD estimates of mean and covariance
credit_card_shapley <- int_Shapley(credit_card_int)
```

`int_Shapley_decomp` *Compute Shapley Decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) for Interval-valued Data*

Description

Decomposes the squared interval-valued Mahalanobis distance of each observation into outlyingness contributions of (Centers, Ranges, and CrossCentersRanges) per variable for interval-valued data.

Usage

```
int_Shapley_decomp(data, mean_c = NULL, mean_r = NULL, cov = NULL)
```

Arguments

`data`	An `intData` object containing the interval-valued dataset (macrodata).
`mean_c`	(Optional) A vector specifying the mean of centers. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`mean_r`	(Optional) A vector specifying the mean of ranges. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`cov`	(Optional) A covariance matrix. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.

Details

Let $\boldsymbol{\mu}_B = (\boldsymbol{\mu}_C^\top, \boldsymbol{\mu}_R^\top)^\top$ be the barycenter and $\boldsymbol{\Sigma}_B$ the symbolic covariance matrix (see `int_cov`). Based on the Shapley value (see `int_Shapley`), we can further decompose the Interval-Mahalanobis distance of an interval-valued observation $\boldsymbol{x} = (\boldsymbol{c}^\top, \boldsymbol{r}^\top)^\top$ into contributions of the centers, ranges and cross-centers-ranges of the variables. The decomposition is defined according to the `LatentCase`:

- `"U_id_symmetric"`: The latent variables are identically distributed and symmetric:
  - Centers contribution:
    $$(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],$$
  - Ranges contribution:
    $$\delta(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right],$$

  where $\delta = \mathbb{E}(U^2)/4$ is the parameter of the latent variables.
- `"U_id"`: The latent variables are identically distributed:
  - Centers contribution:
    $$(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],$$
  - Ranges contribution:
    $$\delta(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right],$$
  - CrossCentersRanges contribution:
    $$\frac{\mathbb{E}(U)}{2}(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] + \frac{\mathbb{E}(U)}{2}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],$$

  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:
  - Centers contribution:
    $$(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],$$
  - Ranges contribution:
    $$\frac{1}{4}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\left(\boldsymbol{\mathfrak{E}}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1}\right)(\boldsymbol{r} - \boldsymbol{\mu}_R)\right],$$
  - CrossCentersRanges contribution:
    $$\frac{1}{2}(\boldsymbol{c} - \boldsymbol{\mu}_C) \bullet \left[\boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Psi}(\boldsymbol{r} - \boldsymbol{\mu}_R)\right] + \frac{1}{2}(\boldsymbol{r} - \boldsymbol{\mu}_R) \bullet \left[\boldsymbol{\Psi}\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_C)\right],$$

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

Value

A list containing the matrix of Shapley value decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) per variable for each observation.

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Explainable Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2606.26307. https://arxiv.org/abs/2606.26307

Examples

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

# Compute Shapley decomposition into contributions of (Centers, Ranges, and CrossCentersRanges)
# based on IMCD estimates of mean and covariance
credit_card_shap_decomp <- int_Shapley_decomp(credit_card_int)
```

`int_Shapley_interaction` *Compute Shapley interaction indices for Interval-valued Data*

Description

Obtains a $p \times p$ matrix containing pairwise outlyingness scores based on Shapley interaction indices for each observation. Decomposes the squared interval-valued Mahalanobis distance of each observation into outlyingness contributions of pairs of variables.

Usage

```
int_Shapley_interaction(data, mean_c = NULL, mean_r = NULL, cov = NULL)
```

Arguments

`data`	An `intData` object containing the interval-valued dataset (macrodata).
`mean_c`	(Optional) A vector specifying the mean of centers. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`mean_r`	(Optional) A vector specifying the mean of ranges. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.
`cov`	(Optional) A covariance matrix. Defaults to `NULL`, in which case it will be computed using the `IMCD` function.

Details

Let $\boldsymbol{\mu}_B = (\boldsymbol{\mu}_C^\top, \boldsymbol{\mu}_R^\top)^\top$ be the barycenter and $\boldsymbol{\Sigma}_B$ the symbolic covariance matrix (see `int_cov`). Let also $\boldsymbol{\phi}(\boldsymbol{x})$ be the Shapley value of $\boldsymbol{x}$ (see `int_Shapley`) and $\mathrm{diag}(\boldsymbol{v})$ be the diagonal matrix whose main diagonal is the vector $\boldsymbol{v}$. The Shapley interaction index of an interval-valued observation $\boldsymbol{x} = (\boldsymbol{c}^\top, \boldsymbol{r}^\top)^\top$, for the Interval-Mahalanobis distance, is defined according to the `LatentCase`:

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

$$\boldsymbol{\Phi}(\boldsymbol{x}) = 2(\boldsymbol{c}-\boldsymbol{\mu}_C)(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \bullet \boldsymbol{\Sigma}_B^{-1} + 2\delta(\boldsymbol{r}-\boldsymbol{\mu}_R)(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \bullet \boldsymbol{\Sigma}_B^{-1} - \operatorname{diag}\left(\boldsymbol{\phi}(\boldsymbol{x})\right),$$

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

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

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{x}) = {} & 2(\boldsymbol{c}-\boldsymbol{\mu}_C)(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \bullet \boldsymbol{\Sigma}_B^{-1} + 2\delta(\boldsymbol{r}-\boldsymbol{\mu}_R)(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \bullet \boldsymbol{\Sigma}_B^{-1} \\ & + \mathbb{E}(U)(\boldsymbol{c}-\boldsymbol{\mu}_C)(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \bullet \boldsymbol{\Psi} + \mathbb{E}(U)(\boldsymbol{r}-\boldsymbol{\mu}_R)(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \bullet \boldsymbol{\Sigma}_B^{-1} - \operatorname{diag}\left(\boldsymbol{\phi}(\boldsymbol{x})\right),\end{aligned}$$

  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:

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{x}) = {} & 2(\boldsymbol{c}-\boldsymbol{\mu}_C)(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \bullet \boldsymbol{\Sigma}_B^{-1} + \frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu}_R)(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \bullet \mathfrak{E}_{UU} \bullet \boldsymbol{\Sigma}_B^{-1} \\ & + (\boldsymbol{c}-\boldsymbol{\mu}_C)(\boldsymbol{r}-\boldsymbol{\mu}_R)^\top \bullet \boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Psi} + (\boldsymbol{r}-\boldsymbol{\mu}_R)(\boldsymbol{c}-\boldsymbol{\mu}_C)^\top \bullet \boldsymbol{\Psi}\boldsymbol{\Sigma}_B^{-1} - \operatorname{diag}\left(\boldsymbol{\phi}(\boldsymbol{x})\right),\end{aligned}$$

  where:

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

Value

A list containing the matrix of Shapley interaction indices for each observation.

References

Loureiro, C. P., Oliveira, M. R., Brito, P., & Oliveira, L. (2026). Explainable Outlier Detection for Interval-valued Data. arXiv preprint arXiv:2606.26307. https://arxiv.org/abs/2606.26307

Examples

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

# Compute Shapley interaction indices based on the mean and covariance matrix estimated by IMCD
credit_card_shap_inter <- int_Shapley_interaction(credit_card_int)
```

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`	(Optional) A vector specifying the mean of centers. Defaults to `NULL`, in which case it will be computed using the sample mean of centers.
`mean_r`	(Optional) A vector specifying the mean of ranges Defaults to `NULL`, in which case it will be computed using the sample mean of ranges.

Details

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

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

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

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

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

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

  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:

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

  where:

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

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(
  microdata,
  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,
  removeDegenerate = FALSE
)
```

Arguments

`microdata` 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) A list with the parameters of the latent variables. Expects a list with a single number if `LatentCase` is `"U_id_symmetric"`, a list of two numbers if `LatentCase` is `"U_id"`, and a list of two matrices if `LatentCase` is `"General"`.

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

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

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"`, `"Degenerated"`), if not a vector must be provided with the distribution for each variable. The default is `"Unif"` if `LatentCase="U_id_symmetric"`, and `"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.

removeDegenerate
: Logical parameter indicating if observations with at least one degenerate interval should be removed. 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. Degenerate intervals (i.e., intervals whose lower and upper bounds are equal) can be removed by setting removeDegenerate to TRUE.

Value

An intData object containing the aggregated interval-valued data. If removeDegenerate is TRUE, observations with degenerate intervals are removed, and NULL is returned if all observations are removed.

References

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

Examples

```
data(creditcard)
CreditCard_microdata <- creditcard$microdata

# Define grouping variable for microdata aggregation
credit_agrby <- factor(paste(CreditCard_microdata$Name, CreditCard_microdata$Month, sep = "_"))

# Create intData object by aggregating microdata using the default minmax criterion
# and using KDE for estimation of the latent distribution in the general case
credit_agr <- micro2intData(CreditCard_microdata[,3:7],
                            agrby = 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_bar_int_Shapley *Barplot of Shapley values for Interval-valued Data*

Description

Barplot of Shapley values for Interval-valued Data

Usage

```
plot_bar_int_Shapley(
  x,
  cutoff_value = NULL,
  cutoff_label = NULL,
  palette = NULL,
  abbrev.var = 20,
  abbrev.obs = 20,
  sort.obs = TRUE,
  plot_IMah = TRUE,
  IMah_label = expression(Robust ~ d[IMah]^2 * (bold(x))),
  rotate_x = TRUE
)
```

Arguments

x	A $n \times p$ matrix containing the Shapley values of n observations and p variables.
cutoff_value	Numeric. The cutoff value used for detecting outliers. If cutoff_value is not NULL (default), the cutoff value is included in the plot.
cutoff_label	Character. Label for the cutoff value line in the plot.
palette	A vector with colors for each variable. If palette is NULL (default), the colors are generated using RColorBrewer.
abbrev.var	Integer. If abbrev.var > 0, column names are abbreviated using abbreviate with minlenght = abrev.var.
abbrev.obs	Integer. If abbrev.obs > 0, row names are abbreviated using abbreviate with minlenght = abrev.obs.
sort.obs	Logical. If TRUE (default), observations are sorted according to their squared (robust) Interval-Mahalanobis distance.
plot_IMah	Logical. If TRUE (default), the squared (robust) Interval-Mahalanobis distance will be included in the plot.
IMah_label	Character. Label for the Interval-Mahalanobis distance in the plot legend. Default is "Robust $d^2_{\mathrm{IMah}}(\boldsymbol{x})$".
rotate_x	Logical. If TRUE (default), the x-axis labels are rotated.

Value

Returns a barplot that displays the Shapley values (int_Shapley) for each observation and optionally (plot_IMah = TRUE) includes the squared (robust) Interval-Mahalanobis distance (IMah_dist) (black bar) and the corresponding outlier detection cut-off value (dotted line).

References

Adapted from package ShapleyOutlier (https://CRAN.R-project.org/package=ShapleyOutlier).

Examples

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

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

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Compute Shapley values
credit_card_shapley <- int_Shapley(credit_card_int,
                                   mean_c = credit_card_IMCD$mean_IMCD_c,
                                   mean_r = credit_card_IMCD$mean_IMCD_r,
                                   cov = credit_card_IMCD$cov_IMCD)

# Plot Shapley values with cutoff line and Interval-Mahalanobis distance
plot_bar_int_Shapley(credit_card_shapley,
                     cutoff_value = credit_card_outliers$cutoff_value,
                     cutoff_label = "Farness 0.9",
                     palette = rainbow(credit_card_int@NIVar))
```

---

plot_bar_int_Shapley_decomp

Barplot of Shapley value decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) for interval-valued data.

Description

Barplot of Shapley value decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) for interval-valued data.

Usage

```
plot_bar_int_Shapley_decomp(
  shapley_decomp,
  palette = NULL,
  rotate_x = TRUE,
  abbrev.obs = 20,
  sort.obs = TRUE,
  plot_IMah = FALSE
)
```

Arguments

shapley_decomp	A list of matrices containing the Shapley value decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) for each observation.
palette	A vector with colors for each feature. If palette is NULL (default), the colors are generated using RColorBrewer.
rotate_x	Logical. If TRUE (default), the x-axis labels are rotated.
abbrev.obs	Integer. If abbrev.obs > 0, row names are abbreviated using abbreviate with minlenght = abbrev.obs.
sort.obs	Logical. If TRUE (default), observations are sorted according to their total Shapley value.
plot_IMah	Logical. If TRUE, the Interval-Mahalanobis distance (sum of all Shapley values) will be included in the plot.

Value

Returns a barplot that displays the Shapley value decomposition into contributions of (Centers, Ranges, and CrossCentersRanges) for each observation.

Examples

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

# Compute Shapley decomposition into contributions of Centers, Ranges, and CrossCentersRanges
# based on IMCD estimates of mean and covariance matrix
credit_card_shap_decomp <- int_Shapley_decomp(credit_card_int)

# Plot Shapley decomposition with contributions of Centers, Ranges, and CrossCentersRanges
plot_bar_int_Shapley_decomp(credit_card_shap_decomp, palette = rainbow(credit_card_int@NIVar))
```

plot_beeswarm_int_Shapley
Beeswarm plot of Shapley values for interval-valued data.

Description

Beeswarm plot of Shapley values for interval-valued data.

Usage

```
plot_beeswarm_int_Shapley(
  shapley,
  color_class,
  color_label = NULL,
  palette = NULL,
  rotate_x = TRUE,
  shape_class = NULL,
  shape_label = NULL,
  ggplotly = FALSE,
  label_obs = NULL,
  y_limits = NULL
)
```

Arguments

shapley	A $n \times p$ matrix containing the Shapley values of n observations and p variables.
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. Default is NULL.
rotate_x	Logical. If TRUE (default), the x-axis labels are rotated.
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.
ggplotly	Logical. If TRUE (default), the plot is converted to an interactive plotly object.
label_obs	A vector with the names of the observations to be labeled in the plot when ggplotly = FALSE. Default is NULL.
y_limits	A numeric vector of length 2 specifying the limits for the y-axis. Default is NULL, so the limits are determined automatically.

Value

Returns a beeswarm plot that displays the Shapley values (int_Shapley) for each observation and feature.

Examples

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

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

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Compute Shapley values
credit_card_shapley <- int_Shapley(credit_card_int,
                                   mean_c = credit_card_IMCD$mean_IMCD_c,
                                   mean_r = credit_card_IMCD$mean_IMCD_r,
                                   cov = credit_card_IMCD$cov_IMCD)

# Beeswarm plot of Shapley values colored by outlier status
plot_beeswarm_int_Shapley(credit_card_shapley,
                     color_class = credit_card_outliers$is_outlier,
                     palette = c("gray50", "darkred"),
                     color_label = "Outlier Status")
```

`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 = FALSE,
  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 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 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

# Compute robust distances using IMCD estimates of mean and covariance
credit_card_dist <- IMah_dist(credit_card_int)
```

```
# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Compute classical distances and outliers
class_dist <- IMah_dist(credit_card_int, z = rep(1,credit_card_int@NObs))
class_outliers <- int_outliers(class_dist,
                               cutoff = "chi-squared",
                               p = credit_card_int@NIVar)

# Create a vector indicating if the observations are outliers or inliers
# based on the robust distance outlier detection
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 Distance-Distance plot
plot_dist_dist(class_dist,
               class_cutoff = class_outliers$cutoff_value,
               class_cutoff_label = "0.975 chi-squared",
               rob_dist = credit_card_dist,
               rob_cutoff = credit_card_outliers$cutoff_value,
               rob_cutoff_label = "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 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

# Compute robust distances using IMCD estimates of mean and covariance
credit_card_dist <- IMah_dist(credit_card_int)

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Create a vector indicating if the observations are outliers or inliers
# based on the robust distance outlier detection
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_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"))
```

---

plot_int_Shapley_inter *Plot Shapley interaction indices*

Description

Plot Shapley interaction indices

Usage

```
plot_int_Shapley_inter(
  x,
  abbrev = 10,
  title = NULL,
  legend = TRUE,
  text_size = 22
)
```

Arguments

x	A $p \times p$ matrix containing the Shapley interaction indices of a single observation.
abbrev	Integer. If abbrev.var > 0, variable names are abbreviated using abbreviate with minlenght = abrev.
title	Character. Title of the plot.
legend	Logical. If TRUE (default), a legend is plotted.
text_size	Integer. Size of the text in the plot

Value

Returns a figure consisting of two panels. The right panel shows the Shapley values, and the left panel the Shapley interaction indices.

References

Adapted from package ShapleyOutlier (https://CRAN.R-project.org/package=ShapleyOutlier).

Examples

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

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

# Compute Shapley interaction indices
credit_card_shap_inter <- int_Shapley_interaction(credit_card_int,
                                                  mean_c = credit_card_IMCD$mean_IMCD_c,
                                                  mean_r = credit_card_IMCD$mean_IMCD_r,
                                                  cov = credit_card_IMCD$cov_IMCD)

# Plot Shapley interaction for 1st observation
plot_int_Shapley_inter(credit_card_shap_inter[[1]])
```

---

plot_pairs_int — *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

```
plot_pairs_int(
  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

# Compute covariance and correlation matrices
credit_card_cov <- int_cov(credit_card_int)
credit_card_cor <- cov2cor(credit_card_cov)
plot_pairs_int(credit_card_int,
                  corr = credit_card_cor,
                  labels = colnames(credit_card_int))

# Alternatively, highlight outliers in the scatter plot and use the robust correlation matrix
# Obtain reweighted IMCD estimates using farness cutoff
credit_card_IMCD <- IMCD(credit_card_int,
                         m = floor(nrow(credit_card_int)*0.75),
                         cutoff = "farness",
                         cutoff_lvl = 0.9)

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 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'

plot_pairs_int(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)
```

---

plot_radar_int_Shapley
Radar plot of Shapley values for interval-valued data.

Description

Radar plot of Shapley values for interval-valued data.

Usage

```
plot_radar_int_Shapley(shapley, palette = NULL, sort.obs = FALSE)
```

Arguments

shapley	A $n \times p$ matrix containing the Shapley values of n observations and p variables.
palette	A vector of palette for each observation. Default is black.
sort.obs	Logical. If TRUE (default), observations are sorted according to their squared (robust) Interval-Mahalanobis distance.

Value

Returns a radar plot that displays the Shapley values (int_Shapley) for each observation.

Examples

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

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

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Compute Shapley values
credit_card_shapley <- int_Shapley(credit_card_int,
                                   mean_c = credit_card_IMCD$mean_IMCD_c,
                                   mean_r = credit_card_IMCD$mean_IMCD_r,
                                   cov = credit_card_IMCD$cov_IMCD)

# colors
outliers_colors <- rep('black',credit_card_int@NObs)
names(outliers_colors) <- rownames(credit_card_int)
```

```
outliers_colors[credit_card_outliers$outliers_names] = '#009de0'

plot_radar_int_Shapley(credit_card_shapley, palette = outliers_colors)
```

---

plot_scatter_int   *Scatter Plot for Interval-valued Data*

Description

Create a scatter plot 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

```
plot_scatter_int(
  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 scatter plot is drawn in the graphic window. The scatter plot 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

plot_scatter_int(credit_card_int[, c(3, 5)])

# Alternatively, highlight outliers in the scatter plot
# Compute robust distances using IMCD estimates of mean and covariance
credit_card_dist <- IMah_dist(credit_card_int)

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_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'

plot_scatter_int(credit_card_int[, c(3, 5)],
           palette = outliers_colors,
           is_outlier = credit_card_outliers$is_outlier)
```

---

plot_tile_int_Shapley *Tileplot of Shapley values for interval-valued data.*

Description

Tileplot of Shapley values for interval-valued data.

Usage

```
plot_tile_int_Shapley(
  shapley,
  outliers = NULL,
  rotate_x = TRUE,
  abbrev.var = FALSE,
  abbrev.obs = FALSE,
  sort.var = FALSE,
  sort.obs = FALSE,
  show_values = FALSE
)
```

Arguments

`shapley`	A $n \times p$ matrix containing the Shapley values of n observations and p variables.
`outliers`	A list containing the outliers' names as returned by `int_outliers`. If `outliers` is not `NULL` (default), only the outliers are highlighted in the plot.
`rotate_x`	Logical. If `TRUE` (default), the x-axis labels are rotated.

abbrev.var — Integer. If abbrev.var > 0, column names are abbreviated using abbreviate with minlenght = abrev.var.

abbrev.obs — Integer. If abbrev.obs > 0, row names are abbreviated using abbreviate with minlenght = abrev.obs.

sort.var — Logical. If TRUE, variables are sorted according to the distance.

sort.obs — Logical. If TRUE, observations are sorted according to their squared Interval-Mahalanobis distance.

show_values — Logical. If TRUE, the Shapley values are displayed in each tile.

Value

Returns a tileplot that displays the Shapley values (int_Shapley) for each observation and variable. Optionally, only the outliers are highlighted in the plot.

References

Adapted from package ShapleyOutlier (https://CRAN.R-project.org/package=ShapleyOutlier).

Examples

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

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

# Detect outliers using farness cutoff
credit_card_outliers <- int_outliers(credit_card_IMCD$robust_dist,
                                     cutoff = "farness",
                                     cutoff_lvl = 0.9)

# Compute Shapley values
credit_card_shapley <- int_Shapley(credit_card_int,
                                   mean_c = credit_card_IMCD$mean_IMCD_c,
                                   mean_r = credit_card_IMCD$mean_IMCD_r,
                                   cov = credit_card_IMCD$cov_IMCD)

plot_tile_int_Shapley(credit_card_shapley,
                      outliers = credit_card_outliers,
                      sort.var = TRUE,
                      sort.obs = TRUE)
```

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.

`rbind` *Row Bind for* `intData`

Description

Combine multiple `intData` objects by rows.

Usage

```
rbind(..., deparse.level = 1)

## S4 method for signature 'intData'
rbind(..., deparse.level = 1)
```

Arguments

`...`	`intData` objects to combine.
`deparse.level`	An integer controlling the construction of labels in the result (default is `1`).

Value

An `intData` object with rows combined from the input `intData` objects.

`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. From the raw Kaggle dataset, the duplicates were removed and to handle tracks associated with multiple genres, robust centroids were estimated for each genre based on the single-genre tracks, and, then, multi-genre tracks were assigned to the closest genre according to the robust Mahalanobis distance.

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 (DOI:10.34740/KAGGLE/DSV/4372070; Spotify Tracks Dataset by Maharshi Pandya).

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.

---

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