Package 'HistDAWass'

October 29, 2026

Type Package

Title Histogram-Valued Data Analysis

Version 1.0.10

Date 2026-10-22

Maintainer Antonio Irpino <antonio.irpino@unicampania.it>

Description In the framework of Symbolic Data Analysis, a relatively new approach to the statistical analysis of multi-valued data, we consider histogram-valued data, i.e., data described by univariate histograms. The methods and the basic statistics for histogram-valued data are mainly based on the L2 Wasserstein metric between distributions, i.e., the Euclidean metric between quantile functions. The package contains unsupervised classification techniques, least square regression and tools for histogram-valued data and for histogram time series. An introducing paper is Irpino A. Verde R. (2015) <doi:10.1007/s11634-014-0176-4>.

License GPL (>= 2)

Imports graphics, class, FactoMineR, ggplot2 (>= 3.4.0), ggridges, grid, histogram, grDevices, stats, utils, Rcpp

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/Airpino/HistDAWass

BugReports https://github.com/Airpino/HistDAWass/issues

Depends R(>= 3.1), methods

LazyData true

Collate 'For_Rccp_int.R' 'All_classes.R' 'RcppExports.R' 'Utility.R' 'Met_HTS.R' 'Met_MatH.R' 'Met_distributionH.R' 'Fuzzy_cmeans.R' 'H_time_series.R' 'HistDAWass-package.R' 'Kohonen_maps.R' 'principal_components.R' 'regression.R' 'unsuperv_classification.R' 'Plotting_with_ggplot.R'

Encoding UTF-8

NeedsCompilation yes

LinkingTo Rcpp,RcppArmadillo

Config/roxygen2/version 8.1.0

Author Antonio Irpino [aut, cre] (ORCID:
<https://orcid.org/0000-0001-9293-7180>)

Repository CRAN

Date/Publication 2026-10-29 10:40:09 UTC

Contents

HistDAWass-package 4
Age_Pyramids_2014 5
Agronomique 6
Arith-distributionH 6
BLOOD 7
BloodBRITO 8
Center.cell.MatH 8
checkEmptyBins 9
China_Month 9
China_Seas 10
compP 11
compQ 11
crwtransform 12
data2hist 13
distributionH-class 14
dotpW 15
DouglasPeucker 16
extract-MatH 17
get.cell.MatH 17
get.distr 18
get.histo 19
get.m 19
get.MatH.main.info 20
get.MatH.stats 21
get.s 21
HTS-class 22
HTS.exponential.smoothing 23
HTS.moving.averages 23
HTS.predict.knn 24
kurtH 25
MatH-class 25
meanH 27
OzoneFull 27
OzoneH 28
plot-distributionH 28
plot-HTS 29

plot-MatH . . . 30
plot-TdistributionH . . . 30
plotPredVsObs . . . 31
plot_errors . . . 32
register . . . 33
registerMH . . . 33
RetHTS . . . 34
rQQ . . . 35
ShortestDistance . . . 35
show-distributionH . . . 36
show-MatH . . . 37
skewH . . . 37
stations_coordinates . . . 38
stdH . . . 38
subsetHTS . . . 39
summaryHTS . . . 39
TdistributionH-class . . . 40
TMatH-class . . . 41
WassSqDistH . . . 41
WH.1d.PCA . . . 42
WH.bind . . . 43
WH.mat.prod . . . 44
WH.mat.sum . . . 45
WH.MultiplePCA . . . 45
WH.plot_multiple_indivs . . . 46
WH.plot_multiple_Spanish.funs . . . 48
WH.regression.GOF . . . 49
WH.regression.two.components . . . 49
WH.regression.two.components.predict . . . 50
WH.var.covar . . . 51
WH.vec.mean . . . 52
WH_2d_Adaptive_Kohonen_maps . . . 53
WH_2d_Kohonen_maps . . . 55
WH_adaptive.kmeans . . . 57
WH_adaptive_fcmeans . . . 59
WH_fcmeans . . . 60
WH_hclust . . . 61
WH_kmeans . . . 63
WH_MAT_DIST . . . 64

Index **66**

HistDAWass-package *HistDAWass: Histogram-Valued Data Analysis*

Description

Methods for the analysis of histogram-valued data, i.e. data described by univariate histograms, in the framework of Symbolic Data Analysis. The basic statistics and the methods are based on the L2 Wasserstein metric between distributions, i.e. the Euclidean metric between quantile functions.

Data structures

- distributionH: a histogram; data2hist builds one from raw data.
- MatH: a matrix of histograms (individuals by histogram-valued variables).
- HTS-class: a histogram time series.

Basic statistics

Moments of a histogram (meanH, stdH, skewH, kurtH), quantiles and cdf (compQ, compP), the L2 Wasserstein distance (WassSqDistH, WH_MAT_DIST), the mean histogram (WH.vec.mean), variance-covariance and correlation matrices (WH.var.covar).

Methods

- Clustering: WH_kmeans, WH_adaptive.kmeans, WH_fcmeans, WH_adaptive_fcmeans, WH_hclust, Kohonen maps (WH_2d_Kohonen_maps, WH_2d_Adaptive_Kohonen_maps).
- Principal component analysis: WH.1d.PCA, WH.MultiplePCA.
- Regression: WH.regression.two.components.
- Histogram time series: HTS.moving.averages, HTS.exponential.smoothing, HTS.predict.knn.

Author(s)

Maintainer: Antonio Irpino <antonio.irpino@unicampania.it> (ORCID)

Authors:

- Antonio Irpino <antonio.irpino@unicampania.it> (ORCID)

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

See Also

Useful links:

- https://github.com/Airpino/HistDAWass
- Report bugs at https://github.com/Airpino/HistDAWass/issues

Examples

```
# a matrix of four histograms (2 individuals x 2 variables)
a <- list(
  distributionH(x = c(80, 100, 120, 135, 150, 165, 180, 200, 240),
                p = c(0, 0.025, 0.1, 0.275, 0.525, 0.725, 0.887, 0.975, 1)),
  distributionH(x = c(80, 100, 120, 135, 150, 165, 180, 195, 210, 240),
                p = c(0, 0.013, 0.101, 0.255, 0.508, 0.718, 0.895, 0.961, 0.987, 1)),
  distributionH(x = c(95, 110, 125, 140, 155, 170, 185, 200, 215, 230, 245),
              p = c(0, 0.012, 0.041, 0.154, 0.36, 0.595, 0.781, 0.929, 0.972, 0.992, 1)),
  distributionH(x = c(105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 260),
              p = c(0, 0.009, 0.035, 0.081, 0.186, 0.385, 0.633, 0.832, 0.932, 0.977, 1))
)
M <- MatH(x = a, nrows = 2, ncols = 2,
          rownames = c("u1", "u2"), varnames = c("Var_1", "Var_2"))
M
WH.vec.mean(M)
```

`Age_Pyramids_2014` *Age pyramids of the countries of the World in 2014*

Description

Distributions of the population by 5-year age classes (relative frequencies) for 228 countries and the World (first row): whole population, male and female population.

Format

A `MatH` object with 229 rows (the World and 228 countries) and 3 columns: `Both.Sexes.Population`, `Male.Population`, `Female.Population`.

Source

United States Census Bureau, https://www.census.gov/data.html

Examples

```
get.MatH.main.info(Age_Pyramids_2014)
```

Agronomique *Agronomique data*

Description

Distributions of the marginal costs of farms in 22 French regions.

Format

A `MatH` object with 22 rows (regions) and 4 columns: `Y_TSC` (total costs of a farm), `X_Wheat` (costs for wheat), `X_Pig` (costs for pigs), `X_Cmilk` (costs for cow milk).

Source

Verde, R., Irpino, A., Desbois, D. (2014). *How to cope with modelling and privacy concerns? A regression model and a visualization tool for aggregated data*, Conference of European Statistics Stakeholders, Rome.

Examples

```
Agronomique
```

Arith-distributionH *Arithmetic operators for histograms*

Description

Sum, difference and product of `distributionH` objects and numbers, consistent with the L2 Wasserstein metric: the operations are applied to the quantile functions of the distributions.

- `e1 + e2`: the sum of the quantile functions. A number is treated as an impulse distribution, so `x + a` shifts `x` by `a`.
- `e1 - e2`: the difference of the quantile functions. The difference of two distributions is a distribution only if `e1` is more variable than `e2` (the result must be a non-decreasing quantile function); otherwise it may not be valid. `x - a` shifts `x` by `-a`.
- `a * x` or `x * a`: the product of a distribution and a number, i.e. the distribution of `a` times the random variable (for `a < 0` the distribution is reflected). Similarly, `a - x` is the distribution of `a` minus the random variable.

Usage

```
## S4 method for signature 'distributionH,distributionH'
e1 + e2

## S4 method for signature 'distributionH,distributionH'
e1 - e2

## S4 method for signature 'distributionH,distributionH'
e1 * e2
```

Arguments

e1, e2	Two distributionH objects, or a distributionH and a number.

Value

A distributionH object.

Examples

```
d1 <- distributionH(x = c(1, 2, 3), p = c(0, 0.4, 1))
d2 <- distributionH(x = c(7, 8, 10, 15), p = c(0, 0.2, 0.7, 1))
d1 + d2
d2 + 10
2 * d1
d2 - d1
```

BLOOD *Blood data*

Description

Histograms of cholesterol, hemoglobin and hematocrit for 14 groups of patients (by sex and age).

Format

A MatH object with 14 rows (groups of patients) and 3 columns (variables).

Source

Billard L. and Diday E. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining*, Wiley.

Examples

```
BLOOD
plot(BLOOD)
```

BloodBRITO *Blood data from Brito*

Description

Histograms of hematocrit and hemoglobin for 10 patients.

Format

A MatH object with 10 rows (patients) and 2 columns (variables).

Source

Dias, S. and Brito P. (2013). *Distribution and Symmetric Distribution Regression Model for Histogram-Valued Variables*, arXiv:1303.6199.

Examples

```
BloodBRITO
```

Center.cell.MatH *Centre the histograms of a matrix*

Description

Shifts each histogram of a MatH object so that its mean is zero.

Usage

```
Center.cell.MatH(object)
```

Arguments

object	A MatH object.

Value

A MatH object whose histograms have zero mean.

Examples

```
CEN_BLOOD <- Center.cell.MatH(BLOOD)
get.MatH.stats(CEN_BLOOD, stat = "mean")
```

checkEmptyBins *Fix the empty bins of histograms*

Description

Checks for empty bins, i.e. bins with zero weight (two consecutive equal values of the cdf). Each empty bin gets a weight of `1e-8` and the cdf is renormalised. Since version 1.0.9 empty bins are fixed when a `distributionH` object is created (see `distributionH`), so this method is useful for objects created with previous versions of the package or modified by assigning their slots directly. The `MatH` method applies the check to all the cells of the matrix.

Usage

```
checkEmptyBins(object)
```

Arguments

`object`	A `distributionH` or a `MatH` object.

Value

An object of the same class of `object`, without empty bins.

Examples

```
## an empty bin in an object modified by slot assignment
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.5, 0.6, 1))
mydist@p <- c(0, 0.5, 0.5, 1)
diff(checkEmptyBins(mydist)@p)
## all the distributions of a MatH
newMat <- checkEmptyBins(China_Seas[1:5, 1:3])
```

China_Month *Monthly climatic data of China*

Description

Distributions of 14 climatic variables observed in 60 stations of China in each month (168 histogram-valued variables).

Format

A `MatH` object with 60 rows (stations) and 168 columns (14 variables x 12 months).

Details

The 14 variables are: mean station pressure (mb), mean temperature, mean maximum temperature, mean minimum temperature, total precipitation (mm), sunshine duration (h), mean cloud amount (percentage of sky cover), mean relative humidity (%), snow days (days with snow cover), dominant wind direction (degrees), mean wind speed (m/s), dominant wind frequency (%), extreme maximum temperature (C), extreme minimum temperature (C). The coordinates of the stations are in `stations_coordinates`.

Source

Climate Data Bases of the People's Republic of China 1841-1988 (TR055), doi:10.3334/CDIAC/cli.tr055

Examples

```
get.MatH.main.info(China_Month)$ncols
```

China_Seas *Seasonal climatic data of China*

Description

Distributions of 14 climatic variables observed in 60 stations of China in each season (56 histogram-valued variables).

Format

A `MatH` object with 60 rows (stations) and 56 columns (14 variables x 4 seasons).

Details

The 14 variables are: mean station pressure (mb), mean temperature, mean maximum temperature, mean minimum temperature, total precipitation (mm), sunshine duration (h), mean cloud amount (percentage of sky cover), mean relative humidity (%), snow days (days with snow cover), dominant wind direction (degrees), mean wind speed (m/s), dominant wind frequency (%), extreme maximum temperature (C), extreme minimum temperature (C). The coordinates of the stations are in `stations_coordinates`.

Source

Climate Data Bases of the People's Republic of China 1841-1988 (TR055), doi:10.3334/CDIAC/cli.tr055

Examples

```
get.MatH.main.info(China_Seas)$ncols
```

compP *Cumulative distribution function of a histogram at a value*

Description

Computes the value $F(q)$ of the cumulative distribution function of a `distributionH` object.

Usage

```
compP(object, q)
```

Arguments

`object`	A `distributionH` object.
`q`	A number.

Value

A number between 0 and 1.

See Also

`compQ`

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
compP(mydist, 5)
```

compQ *Quantile of a histogram*

Description

Computes the quantile of a `distributionH` object at a given level, i.e. the value $Q(p) = F^{-1}(p)$ of its quantile function.

Usage

```
compQ(object, p)
```

Arguments

`object`	A `distributionH` object.
`p`	A number between 0 and 1.

Value

A number, the quantile of order p.

See Also

compP

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
compQ(mydist, 0.5) # the median
compQ(mydist, 0) # the minimum
compQ(mydist, 1) # the maximum
compQ(mydist, 0.25) # the first quartile
```

---

crwtransform *Centres, radii and weights of the bins of a histogram*

Description

Computes the centre (midpoint), the radius (half-width) and the weight (probability) of each bin of a distributionH object. This representation allows a fast computation of the statistics based on the L2 Wasserstein distance.

Usage

```
crwtransform(object)
```

Arguments

object	A distributionH object.

Value

A list with three numeric vectors: Centers, the midpoints of the bins; Radii, the half-widths of the bins; Weights, the probabilities of the bins (summing to one).

References

Irpino, A., Verde, R., Lechevallier, Y. (2006) *Dynamic clustering of histograms using Wasserstein metric*, In: Proceedings of COMPSTAT 2006, Physica-Verlag, 869-876.

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
crwtransform(mydist)
```

`data2hist` *Build a histogram from raw data*

Description

Builds a `distributionH` object from a vector of numeric values, using one of several algorithms for choosing the bins.

Usage

```
data2hist(
  data,
  algo = "histogram",
  type = "combined",
  qua = 10,
  breaks = numeric(0),
  epsilon = 0.01
)
```

Arguments

`data`	A numeric vector.
`algo`	The algorithm: `"histogram"` (default), the penalised likelihood method of `histogram` (accurate but slow for large datasets); `"base"`, the bins of `hist`; `"FixedQuantiles"`, `qua` bins with equal probability; `"ManualBreaks"`, the bins given in `breaks`; `"PolyLine"`, a piecewise linear approximation of the empirical cdf obtained with the Ramer-Douglas-Peucker algorithm (see `DouglasPeucker`).
`type`	For `algo = "histogram"`: `"combined"` (default), `"regular"` (equal-width bins) or `"irregular"`, see `histogram`.
`qua`	For `algo = "FixedQuantiles"`, the number of bins (default 10).
`breaks`	For `algo = "ManualBreaks"`, a vector of breaks.
`epsilon`	For `algo = "PolyLine"`, the tolerance of the approximation (default 0.01; the data are scaled to unit standard deviation).

Value

A `distributionH` object. Empty bins (without observations) get a weight of `1e-8`.

See Also

`distributionH`, `histogram`

Examples

```
data <- rnorm(n = 1000, mean = 2, sd = 3)
mydist <- data2hist(data)
plot(mydist)
data2hist(data, algo = "FixedQuantiles", qua = 20)
```

`distributionH-class` *Histogram objects: the distributionH class*

Description

The `distributionH` class describes a histogram (a distribution) by means of its cumulative distribution function (cdf): the bounds of the bins and the values of the cdf at the bounds. All the methods of the package are based on the L2 Wasserstein distance between distributions.

Usage

```
distributionH(x = numeric(0), p = numeric(0))
```

Arguments

`x`	A numeric vector, the bounds of the bins, in non-decreasing order. A single value defines an impulse distribution.
`p`	A numeric vector of the same length of `x`, the values of the cdf at `x`, non-decreasing from 0 to 1.

Value

`distributionH()` returns a `distributionH` object.

Slots

`x` A non-decreasing numeric vector: the bounds of the bins (the domain of the distribution).

`p` A non-decreasing numeric vector of the same length of `x`, from 0 to 1: the cdf at the values in `x`.

`m` The mean of the distribution.

`s` The standard deviation of the distribution.

Creating objects

Objects are created with `distributionH(x, p)` or, equivalently, with `new("distributionH", x, p, m, s)`, where the optional `m` and `s` (mean and standard deviation) are computed when not given.

Empty bins, i.e. bins with zero weight (two consecutive equal values in `p`), may produce numerical problems. Since version 1.0.9 each empty bin gets a weight of `1e-8` and the cdf is renormalised when the object is created (see also `checkEmptyBins`).

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

See Also

`data2hist` to build a histogram from raw data; `meanH`, `stdH`, `plot,distributionH-method`.

Examples

```
# the histogram
# | Bins  | Prob | cdf |
# | [1,2) | 0.4  | 0.4 |
# | [2,3] | 0.6  | 1.0 |
mydist <- distributionH(x = c(1, 2, 3), p = c(0, 0.4, 1))
mydist
str(mydist)
```

`dotpW` *Dot product of two histograms*

Description

Computes the dot product, inducing the L2 Wasserstein metric, between two `distributionH` objects, or between a `distributionH` and a number (considered as an impulse distribution).

Usage

```
dotpW(e1, e2)
```

Arguments

`e1, e2`	Two `distributionH` objects, or a `distributionH` and a number.

Value

A number.

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

Examples

```
mydist1 <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
mydist2 <- distributionH(x = c(5, 7, 15), p = c(0, 0.7, 1))
dotpW(mydist1, mydist2) # 39.51429
dotpW(mydist1, 3) # 13.2
dotpW(3, mydist1) # 13.2
```

`DouglasPeucker` *Ramer-Douglas-Peucker algorithm*

Description

Approximates a curve, given by a sequence of 2d points, with a polyline (Ramer-Douglas-Peucker algorithm). It is used by `data2hist` with `algo = "PolyLine"`.

Usage

```
DouglasPeucker(points, epsilon)
```

Arguments

`points`	A two-column matrix with the coordinates of the points.
`epsilon`	A positive number, the tolerance of the approximation (recommended 0.01).

Value

A two-column matrix with the vertices of the polyline.

See Also

`data2hist`, `ShortestDistance`

Examples

```
x <- seq(0, 1, length.out = 50)
DouglasPeucker(cbind(x, x^2), epsilon = 0.01)
```

extract-MatH *Extract a sub-matrix of histograms*

Description

Extracts rows and/or columns of a MatH object, as for an ordinary matrix.

Usage

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

Arguments

x	A MatH object.
i, j	Row and column indices (integers, names or logical vectors).
...	Not used.
drop	Not used (the result is always a MatH object).

Value

A MatH object.

Examples

```
SUB_D <- BLOOD[c(1, 2, 5), c(1, 2)]
SUB_D
```

get.cell.MatH *Get or set a cell of a matrix of histograms*

Description

get.cell.MatH returns the histogram in a cell of a MatH object; set.cell.MatH assigns a histogram to a cell.

Usage

```
get.cell.MatH(object, r, c)

set.cell.MatH(object, mat, r, c)
```

Arguments

object	For get.cell.MatH, a MatH object; for set.cell.MatH, the distributionH object to assign.
r, c	Integers, the row and the column of the cell.
mat	A MatH object.

Value

get.cell.MatH returns a distributionH object, set.cell.MatH the modified MatH object.

Examples

```
get.cell.MatH(BLOOD, r = 1, c = 1)
mydist <- distributionH(x = c(0, 1, 2, 3, 4), p = c(0, 0.1, 0.6, 0.9, 1))
MAT <- set.cell.MatH(mydist, BLOOD, r = 1, c = 1)
```

---

get.distr *Cumulative distribution function of a histogram*

Description

Returns the points of the cumulative distribution function (cdf) of a distributionH object.

Usage

```
get.distr(object)
```

Arguments

object	A distributionH object.

Value

A data.frame with two columns: x, the bounds of the bins, and p, the values of the cdf at x.

See Also

get.histo

Examples

```
D <- distributionH(x = c(1, 2, 3, 4), p = c(0, 0.2, 0.6, 1))
get.distr(D)
```

get.histo *Bins of a histogram*

Description

Returns the bins of a distributionH object and their probabilities.

Usage

```
get.histo(object)
```

Arguments

object	A distributionH object.

Value

A data.frame with one row for each bin and three columns: min.x and max.x, the bounds of the bin, and p, its probability (relative frequency).

See Also

get.distr

Examples

```
D <- distributionH(x = c(1, 2, 3, 4), p = c(0, 0.2, 0.6, 1))
get.histo(D)
```

get.m *Mean of a histogram (stored value)*

Description

Returns the mean stored in the m slot of a distributionH object. To compute it from the distribution use meanH.

Usage

```
get.m(object)
```

Arguments

object	A distributionH object.

Value

A number, the mean of the distribution.

See Also

meanH, get.s

Examples

```
D <- distributionH(x = c(1, 2, 3, 4), p = c(0, 0.2, 0.6, 1))
get.m(D)
```

---

get.MatH.main.info *Dimensions and names of a matrix of histograms*

Description

Functions returning the number of rows and of columns, the names of the rows (individuals) and of the columns (variables) of a MatH object.

Usage

```
get.MatH.nrows(object)

get.MatH.ncols(object)

get.MatH.rownames(object)

get.MatH.varnames(object)

get.MatH.main.info(object)
```

Arguments

object	A MatH object.

Value

get.MatH.nrows and get.MatH.ncols return an integer; get.MatH.rownames and get.MatH.varnames return a character vector; get.MatH.main.info returns a list with all of them (nrows, ncols, rownames, varnames).

Examples

```
get.MatH.nrows(BLOOD)
get.MatH.ncols(BLOOD)
get.MatH.rownames(BLOOD)
get.MatH.varnames(BLOOD)
get.MatH.main.info(BLOOD)
```

get.MatH.stats *Statistics of the histograms of a matrix*

Description

Computes a statistic (mean, median, standard deviation, ...) for each histogram of a MatH object.

Usage

```
## S4 method for signature 'MatH'
get.MatH.stats(object, stat = "mean", prob = 0.5)
```

Arguments

object	A MatH object.
stat	The statistic: "mean" (default), "median", "min", "max", "std", "skewness", "kurtosis" or "quantile".
prob	A number between 0 and 1, the level of the quantile when stat = "quantile".

Value

A list with the name of the statistic (stat), the level prob (only when stat = "quantile") and the matrix of values (mat).

Examples

```
get.MatH.stats(BLOOD) # means
get.MatH.stats(BLOOD, stat = "median")
get.MatH.stats(BLOOD, stat = "quantile", prob = 0.05) # 5th percentiles
get.MatH.stats(BLOOD, stat = "std")
```

get.s *Standard deviation of a histogram (stored value)*

Description

Returns the standard deviation stored in the s slot of a distributionH object. To compute it from the distribution use stdH.

Usage

```
get.s(object)
```

Arguments

object	A distributionH object.

Value

A non-negative number, the standard deviation of the distribution.

See Also

stdH, get.m

Examples

```
D <- distributionH(x = c(1, 2, 3, 4), p = c(0, 0.2, 0.6, 1))
get.s(D)
```

HTS-class *Histogram time series: the HTS class*

Description

The HTS class describes a histogram time series, i.e. a sequence of histograms (TdistributionH objects) observed along time.

Slots

data A list of TdistributionH objects.

Creating objects

Objects are created with new("HTS", epocs, ListOfTimedElements), where epocs is the number of time points and ListOfTimedElements a list of TdistributionH objects.

See Also

RetHTS for an example dataset; subsetHTS, plot,HTS-method, HTS.moving.averages, HTS.exponential.smoothing, HTS.predict.knn.

Examples

```
RetHTS
subsetHTS(RetHTS, from = 1, to = 5)
```

HTS.exponential.smoothing *Exponential smoothing of a histogram time series*

Description

(Beta version) Exponential smoothing of a histogram time series, based on the L2 Wasserstein distance.

Usage

```
HTS.exponential.smoothing(HTS, alpha = 0.9)
```

Arguments

HTS	A HTS object.
alpha	A number between 0 and 1, the smoothing parameter (default 0.9).

Value

A list with smoothing.alpha, the smoothing parameter, and SmoothedHTS, the smoothed HTS object.

See Also

HTS.moving.averages

Examples

```
mov.expo.smooth <- HTS.exponential.smoothing(HTS = RetHTS, alpha = 0.8)
mov.expo.smooth$SmoothedHTS
```

HTS.moving.averages *Moving averages of a histogram time series*

Description

(Beta version) Smoothing of a histogram time series by (weighted) moving averages, based on the L2 Wasserstein distance.

Usage

```
HTS.moving.averages(HTS, k = 3, weights = rep(1, k))
```

Arguments

HTS	A HTS object.
k	An odd integer, the number of elements of each average (default 3).
weights	A vector of k non-negative weights (default: equal weights).

Value

A list with k, the (normalised) weights and AveragedHTS, the smoothed HTS object.

See Also

HTS.exponential.smoothing

Examples

```
mov.av.smoothed <- HTS.moving.averages(HTS = RetHTS, k = 5)
mov.av.smoothed$AveragedHTS
```

HTS.predict.knn *K-NN forecast of a histogram time series*

Description

(Beta version) Forecasts a histogram of a time series with the k-nearest neighbours method, based on the L2 Wasserstein distance: the forecast is the mean of the histograms following the k histograms most similar to the one at position.

Usage

```
HTS.predict.knn(HTS, position = length(HTS@data), k = 3)
```

Arguments

HTS	A HTS object.
position	An integer, the position of the last observed histogram (default: the last one).
k	The number of neighbours (default 3).

Value

A distributionH object, the forecast.

References

Arroyo, J., Mate, C. (2009). Forecasting histogram time series with k-nearest neighbours methods. International Journal of Forecasting, 25(1), 192-207. doi:10.1016/j.ijforecast.2008.07.003

Examples

```
prediction <- HTS.predict.knn(HTS = RetHTS, position = 108, k = 3)
```

kurtH *Kurtosis of a histogram*

Description

Computes the kurtosis (fourth standardised moment) of a `distributionH` object.

Usage

```
kurtH(object)
```

Arguments

`object`	A `distributionH` object.

Value

A number, the kurtosis index (3 for a Gaussian distribution).

See Also

`meanH`, `stdH`, `skewH`

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
kurtH(mydist) # 2.133533
```

MatH-class *Matrices of histograms: the MatH class*

Description

The `MatH` class describes a matrix of histograms: each cell contains a `distributionH` object, rows are the individuals and columns the (histogram-valued) variables.

Usage

```
MatH(
  x = NULL,
  nrows = 1,
  ncols = 1,
  rownames = NULL,
  varnames = NULL,
  by.row = FALSE
)
```

Arguments

nrows, ncols	Integers, the number of rows (individuals) and of columns (variables). Default 1.
by.row	A logical value (default FALSE): if TRUE the matrix is filled by rows, otherwise by columns.
x	A list of distributionH objects (default NULL, an empty matrix).
rownames, varnames	Character vectors (or lists of strings), the names of the rows and of the columns (variables). Default NULL.

Value

MatH() returns a MatH object.

Slots

M A matrix of mode list, each cell containing a distributionH object.

Creating objects

Objects are created with MatH() or with new("MatH", nrows, ncols, ListOfDist, names.rows, names.cols, by.row), where ListOfDist is a list of distributionH objects and names.rows, names.cols are the names of rows and columns.

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

See Also

get.MatH.main.info, plot,MatH-method, WH.vec.mean, WH.var.covar.

Examples

```
## a list of six distributionH objects
ListOfDist <- vector("list", 6)
ListOfDist[[1]] <- distributionH(c(1, 2, 3), c(0, 0.4, 1))
ListOfDist[[2]] <- distributionH(c(7, 8, 10, 15), c(0, 0.2, 0.7, 1))
ListOfDist[[3]] <- distributionH(c(9, 11, 20), c(0, 0.5, 1))
ListOfDist[[4]] <- distributionH(c(2, 5, 8), c(0, 0.3, 1))
ListOfDist[[5]] <- distributionH(c(8, 10, 15), c(0, 0.75, 1))
ListOfDist[[6]] <- distributionH(c(20, 22, 24), c(0, 0.12, 1))
## a 3 x 2 MatH filled by columns
MyMAT <- MatH(
  x = ListOfDist, nrows = 3, ncols = 2,
  rownames = c("I1", "I2", "I3"), varnames = c("Var1", "Var2")
)
MyMAT
```

```
## an empty 10 x 4 matrix of histograms
MAT <- MatH(nrows = 10, ncols = 4)
```

---

meanH *Mean of a histogram*

Description

Computes the mean (first moment) of a distributionH object.

Usage

```
meanH(object)
```

Arguments

object	A distributionH object.

Value

A number, the mean of the distribution.

See Also

stdH, skewH, kurtH

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
meanH(mydist) # 4.4
```

OzoneFull *Ozone data (stations without missing data)*

Description

Histograms of four variables recorded by 78 stations in the USA, without missing data: ozone concentration (ppb), temperature (C), solar radiation (W/m2) and wind speed (m/s).

Format

A MatH object with 78 rows (stations) and 4 columns (variables).

Source

US EPA, CASTNET (Clean Air Status and Trends Network), meteorological data.

Examples

```
get.MatH.main.info(OzoneFull)
```

OzoneH *Ozone data (all the stations)*

Description

Histograms of four variables recorded by 84 stations in the USA: ozone concentration (ppb), temperature (C), solar radiation (W/m2) and wind speed (m/s). Some stations have missing data.

Format

A `MatH` object with 84 rows (stations) and 4 columns (variables).

Source

US EPA, CASTNET (Clean Air Status and Trends Network), meteorological data.

Examples

```
get.MatH.main.info(OzoneH)
```

plot-distributionH *Plot a histogram*

Description

Plots a `distributionH` object as a histogram, a cumulative distribution function, a quantile function, a density or a boxplot.

Usage

```
## S4 method for signature 'distributionH'
plot(x, type = "HISTO", col = "green", border = "black")
```

Arguments

`x`	A `distributionH` object.
`type`	The type of plot: `"HISTO"` (default, histogram), `"CDF"` (cumulative distribution function), `"QF"` (quantile function), `"DENS"` (kernel density approximation), `"HBOXPLOT"` (horizontal boxplot) or `"VBOXPLOT"` (vertical boxplot).
`col`	The fill colour (default `"green"`).
`border`	The colour of the borders (default `"black"`).

Value

The plot is drawn on the current device and the `ggplot` object is returned invisibly, so that it can be further modified, e.g. `p <- plot(mydist); p + ggplot2::theme_bw()`.

Examples

```
mydist <- distributionH(x = c(7, 8, 10, 15), p = c(0, 0.2, 0.7, 1))
plot(mydist)
plot(mydist, type = "HISTO", col = "red", border = "blue")
plot(mydist, type = "DENS", col = "red", border = "blue")
plot(mydist, type = "HBOXPLOT")
plot(mydist, type = "VBOXPLOT")
plot(mydist, type = "CDF")
plot(mydist, type = "QF")
```

---

plot-HTS *Plot a histogram time series*

Description

Plots a HTS object as a sequence of violin plots or boxplots, one for each timestamp.

Usage

```
## S4 method for signature 'HTS'
plot(x, y = "missing", type = "VIOLIN", border = "black", maxno.perplot = 30)
```

Arguments

x	A HTS object.
y	Not used.
type	The type of plot: "VIOLIN" (default) or "BOXPLOT".
border	The colour of the borders (default "black").
maxno.perplot	The maximum number of timestamps in each row of the plot (default 30): longer series are split into several rows.

Value

The plot is drawn on the current device and the ggplot object is returned invisibly, so that it can be further modified.

Examples

```
plot(subsetHTS(RetHTS, from = 1, to = 10))

plot(RetHTS, type = "BOXPLOT", border = "blue", maxno.perplot = 20)
plot(RetHTS, type = "VIOLIN", maxno.perplot = 20)
```

plot-MatH *Plot a matrix of histograms*

Description

Plots all the histograms of a MatH object: one panel for each variable, one row for each individual.

Usage

```
## S4 method for signature 'MatH'
plot(x, y = "missing", type = "HISTO", border = "black", angL = 330)
```

Arguments

x	A MatH object.
y	Not used.
type	The type of plot: "HISTO" (default, histograms), "DENS" (kernel density approximations) or "BOXPLOT" (boxplots).
border	The colour of the borders (default "black").
angL	Not used (kept for compatibility).

Value

The plot is drawn on the current device and the ggplot object is returned invisibly, so that it can be further modified.

Examples

```
plot(BLOOD)

plot(BLOOD, type = "DENS", border = "blue")
plot(BLOOD, type = "BOXPLOT")
```

plot-TdistributionH *Plot a histogram with a time reference*

Description

Plots a TdistributionH object: see plot,distributionH-method for the types of plot.

Usage

```
## S4 method for signature 'TdistributionH'
plot(x, type = "HISTO", col = "green", border = "black")
```

Arguments

x	A TdistributionH object.
type	The type of plot: "HISTO" (default, histogram), "CDF" (cumulative distribution function), "QF" (quantile function), "DENS" (kernel density approximation), "HBOXPLOT" (horizontal boxplot) or "VBOXPLOT" (vertical boxplot).
col	The fill colour (default "green").
border	The colour of the borders (default "black").

Value

The plot is drawn on the current device and the ggplot object is returned invisibly.

Examples

```
td <- new("TdistributionH", tstamp = 1, x = c(1, 2, 3), p = c(0, 0.4, 1))
plot(td)
```

---

plotPredVsObs *Plot observed versus predicted histograms*

Description

Compares observed and predicted histograms (e.g. from WH.regression.two.components.predict), one panel for each individual.

Usage

```
plotPredVsObs(PRED, OBS, type = "HISTO", ncolu = 2)
```

Arguments

PRED	A MatH object with one column, the predicted histograms.
OBS	A MatH object with one column, the observed histograms.
type	The type of plot: "HISTO" (default, histograms), "CDF" (cumulative distribution functions) or "DENS" (kernel density approximations).
ncolu	The number of columns of panels (default 2).

Value

The plot is drawn on the current device and the ggplot object is returned invisibly.

See Also

plot_errors

Examples

```
pars <- WH.regression.two.components(BLOOD, Yvar = 1, Xvars = c(2:3))
PRED <- WH.regression.two.components.predict(data = BLOOD[, 2:3], parameters = pars)
OBS <- BLOOD[, 1]
plotPredVsObs(PRED, OBS, "HISTO")

plotPredVsObs(PRED, OBS, "CDF")
plotPredVsObs(PRED, OBS, "DENS")
```

`plot_errors` *Plot the errors of predicted histograms*

Description

Plots the differences between observed and predicted histograms (e.g. from `WH.regression.two.components.predict`) and prints the root mean squared error based on the L2 Wasserstein distance.

Usage

```
plot_errors(PRED, OBS, type = "HISTO_QUA", np = 200)
```

Arguments

PRED	A MatH object with one column, the predicted histograms.
OBS	A MatH object with one column, the observed histograms.
type	The type of plot: "HISTO_QUA" (default, differences of the quantile functions), "HISTO_DEN" (differences of the densities) or "DENS_KDE" (differences of kernel density approximations).
np	The number of points used to compute the differences (default 200).

Value

The root mean squared error, invisibly. The plot is drawn on the current device.

See Also

`plotPredVsObs`, `WH.regression.GOF`

Examples

```
pars <- WH.regression.two.components(BLOOD, Yvar = 1, Xvars = c(2:3))
PRED <- WH.regression.two.components.predict(data = BLOOD[, 2:3], parameters = pars)
OBS <- BLOOD[, 1]
plot_errors(PRED, OBS, "HISTO_QUA")
plot_errors(PRED, OBS, "HISTO_DEN")
plot_errors(PRED, OBS, "DENS_KDE")
```

register *Register two histograms on a common cdf*

Description

Given two distributionH objects, returns two equivalent distributions sharing the same cdf values (the union of the two p vectors). This is the basis for computing the statistics based on the L2 Wasserstein distance.

Usage

```
register(object1, object2)
```

Arguments

object1, object2
: Two distributionH objects.

Value

A list: the two distributionH objects, with the same p slot, and a matrix with their quantiles and the common cdf.

References

Irpino, A., Lechevallier, Y. and Verde, R. (2006): *Dynamic clustering of histograms using Wasserstein metric* In: Rizzi, A., Vichi, M. (eds.) COMPSTAT 2006. Physica-Verlag, Berlin, 869-876.

Examples

```
mydist1 <- distributionH(c(1, 2, 3), c(0, 0.4, 1))
mydist2 <- distributionH(c(7, 8, 10, 15), c(0, 0.2, 0.7, 1))
regDist <- register(mydist1, mydist2)
regDist[[1]]@p
regDist[[1]]@x
regDist[[2]]@x
```

registerMH *Register the histograms of a matrix on a common cdf*

Description

registerMH recomputes all the histograms of a MatH object so that they share the same cdf values (the same p slot), without changing the distributions. This allows a fast computation of all the methods based on the L2 Wasserstein metric. is.registeredMH checks whether the histograms already share the same cdf.

Usage

```
is.registeredMH(object)

registerMH(object)
```

Arguments

object	A MatH object.

Value

registerMH returns a MatH object whose histograms share the same cdf; is.registeredMH returns TRUE or FALSE.

References

Irpino, A., Lechevallier, Y. and Verde, R. (2006): *Dynamic clustering of histograms using Wasserstein metric* In: Rizzi, A., Vichi, M. (eds.) COMPSTAT 2006. Physica-Verlag, Berlin, 869-876.

Examples

```
mydist1 <- distributionH(c(1, 2, 3), c(0, 0.4, 1))
mydist2 <- distributionH(c(7, 8, 10, 15), c(0, 0.2, 0.7, 1))
mydist3 <- distributionH(c(9, 11, 20), c(0, 0.8, 1))
MyMAT <- MatH(x = list(mydist1, mydist2, mydist3), nrows = 1, ncols = 3)
is.registeredMH(MyMAT) # FALSE
MATregistered <- registerMH(MyMAT)
is.registeredMH(MATregistered) # TRUE
MATregistered@M[[1, 1]]@p
```

RetHTS *Histogram time series of returns*

Description

Daily histograms of the returns of the US dollar / Japanese yen exchange rate.

Format

A HTS-class object with 108 histograms.

Examples

```
summaryHTS(subsetHTS(RetHTS, from = 1, to = 5))
plot(subsetHTS(RetHTS, from = 1, to = 20))
```

rQQ *Correlation between the quantile functions of two histograms*

Description

Computes the correlation between the quantile functions of two `distributionH` objects.

Usage

```
rQQ(e1, e2)
```

Arguments

`e1, e2`	Two `distributionH` objects.

Value

A number, the correlation between the quantile functions.

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

Examples

```
mydist1 <- distributionH(x = c(1, 2, 3), p = c(0, 0.4, 1))
mydist2 <- distributionH(x = c(7, 8, 10, 15), p = c(0, 0.2, 0.7, 1))
rQQ(mydist1, mydist2) # 0.916894
```

ShortestDistance *Distance between a point and a line*

Description

Computes the Euclidean distance between a point and the line through two points (used by `DouglasPeucker`).

Usage

```
ShortestDistance(p, line)
```

Arguments

`p`	A vector with the coordinates of the point.
`line`	A 2 x 2 matrix with the coordinates of two points of the line.

Value

A number, the distance of p from line.

See Also

DouglasPeucker

Examples

```
ShortestDistance(c(0, 1), rbind(c(0, 0), c(1, 0)))
```

---

show-distributionH *Print a histogram*

Description

Prints a distributionH object as a table of bins and probabilities. If the histogram has many bins, only the first and the last five are shown.

Usage

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

Arguments

object	A distributionH object.

Value

Invisibly NULL; called for its side effect.

Examples

```
mydist <- distributionH(x = c(7, 8, 10, 15), p = c(0, 0.2, 0.7, 1))
mydist
```

show-MatH *Print a matrix of histograms*

Description

Prints a MatH object showing, for each histogram, its mean and its standard deviation.

Usage

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

Arguments

object	A MatH object.

Value

Invisibly NULL; called for its side effect.

Examples

```
BLOOD
```

skewH *Skewness of a histogram*

Description

Computes the skewness (third standardised moment) of a distributionH object.

Usage

```
skewH(object)
```

Arguments

object	A distributionH object.

Value

A number, the skewness index.

See Also

meanH, stdH, kurtH

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
skewH(mydist) # 0.7148855
```

`stations_coordinates` *Coordinates of the stations of the China datasets*

Description

Geographical coordinates of the 60 stations of `China_Month` and `China_Seas`.

Format

A data frame with 60 rows and 9 columns: `ID`, `Name_ST` (name of the station), `Code_ST` (code of the station), `LAT.deg`, `LAT.min`, `LON.deg`, `LON.min` (latitude and longitude, degrees and minutes), `Elevation`, `Beginning` (first year of observation).

Source

Climate Data Bases of the People's Republic of China 1841-1988 (TR055), doi:10.3334/CDIAC/cli.tr055

Examples

```
head(stations_coordinates)
```

`stdH` *Standard deviation of a histogram*

Description

Computes the standard deviation (square root of the second centred moment) of a `distributionH` object.

Usage

```
stdH(object)
```

Arguments

`object`	A `distributionH` object.

Value

A non-negative number, the standard deviation of the distribution.

See Also

meanH, skewH, kurtH

Examples

```
mydist <- distributionH(x = c(1, 2, 3, 10), p = c(0, 0.1, 0.5, 1))
stdH(mydist) # 2.563851
```

---

subsetHTS *Subset of a histogram time series*

Description

Extracts the elements of a histogram time series between two positions.

Usage

```
subsetHTS(object, from, to)
```

Arguments

object	A HTS object.
from, to	Integers, the positions of the first and of the last element.

Value

A HTS object.

Examples

```
SUB_RetHTS <- subsetHTS(RetHTS, from = 1, to = 20) # the first 20 elements
```

---

summaryHTS *Summary statistics of a histogram time series*

Description

Computes basic statistics for each histogram of a HTS object.

Usage

```
summaryHTS(x)
```

Arguments

x	A HTS object.

Value

A data frame with one row for each histogram: timestamp and period, mean, standard deviation, skewness, kurtosis, minimum, quartiles and maximum.

Examples

```
summaryHTS(subsetHTS(RetHTS, from = 1, to = 10))
```

TdistributionH-class *Histograms with a time reference: the TdistributionH class*

Description

The `TdistributionH` class extends `distributionH` with a time reference: a timestamp or a time period. It is the element of a histogram time series (`HTS-class`).

Slots

`tstamp` A number, the timestamp.

`period` A list with two elements, `start` and `end`, the bounds of the time period (alternative to `tstamp`).

`x` The bounds of the bins (see `distributionH-class`).

`p` The cdf at `x`.

`m` The mean.

`s` The standard deviation.

Creating objects

Objects are created with `new("TdistributionH", tstamp, period, x, p, m, s)`, where `x`, `p`, `m` and `s` are as in `distributionH`.

See Also

`HTS-class`, `plot,TdistributionH-method`

Examples

```
td <- new("TdistributionH", tstamp = 1, x = c(1, 2, 3), p = c(0, 0.4, 1))
td@tstamp
```

TMatH-class *Matrices of histograms with a time reference: the TMatH class*

Description

The TMatH class extends MatH with a time reference (timestamps or time periods).

Slots

tstamp A numeric vector of timestamps.

period A list of time periods (alternative to tstamp).

M The matrix of histograms, as in MatH-class.

Creating objects

Objects are created with new("TMatH", mat, tstamp, period), where mat is a MatH object.

See Also

MatH-class, HTS-class

WassSqDistH *Squared L2 Wasserstein distance between two histograms*

Description

Computes the squared L2 Wasserstein distance between two distributionH objects, i.e. the integral of the squared difference of their quantile functions. Optionally it returns its decomposition into a position, a size and a shape component.

Usage

```
## S4 method for signature 'distributionH,distributionH'
WassSqDistH(object1 = object1, object2 = object2, details = FALSE)
```

Arguments

object1, object2	Two distributionH objects.
details	A logical value (default FALSE). If TRUE, the decomposition of the distance is returned.

Value

If `details = FALSE`, a number, the squared distance. If `details = TRUE`, a named vector with the squared distance (`SQ_W_dist`), its position (`POSITION`), size (`SIZE`) and shape (`SHAPE`) components, and the correlation between the quantile functions (`rQQ`).

References

Irpino, A. and Romano, E. (2007): *Optimal histogram representation of large data sets: Fisher vs piecewise linear approximations*. RNTI E-9, 99-110.

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

See Also

`WH_MAT_DIST` for the distance matrix of a `MatH`.

Examples

```
mydist1 <- distributionH(x = c(1, 2, 3), p = c(0, 0.4, 1))
mydist2 <- distributionH(x = c(7, 8, 10, 15), p = c(0, 0.2, 0.7, 1))
WassSqDistH(mydist1, mydist2)
WassSqDistH(mydist1, mydist2, details = TRUE)
```

WH.1d.PCA *Principal component analysis of a histogram variable*

Description

Principal component analysis (PCA) of one histogram variable based on the L2 Wasserstein distance: a centred (not standardised) PCA of the quantiles of the histograms. It provides a dimensional reduction and a visualization of the distributions.

Usage

```
WH.1d.PCA(
  data,
  var,
  quantiles = 10,
  plots = TRUE,
  listaxes = c(1:4),
  axisequal = FALSE,
  qcut = 1,
  outl = 0
)
```

Arguments

data	A MatH object.
var	An integer, the column of the variable to analyse.
quantiles	An integer, the number of quantiles used in the analysis (default 10).
plots	A logical value (default TRUE). If TRUE, for each factorial plane four plots (the variables and the individuals, plain and coloured by mean and by standard deviation) are drawn on the current graphic device, in a 2 x 2 layout.
listaxes	A vector of integers, the axes of the factorial planes to plot (default 1:4, i.e. the planes 1-2 and 3-4).
axisequal	A logical value (default FALSE). If TRUE, the x and y axes of the plots have the same scale.
qcut	A number between 0.5 and 1 (default 1): densities higher than their qcut quantile are cut in the plots, to avoid very peaked densities.
outl	A number between 0 (default) and 0.5, the probability mass removed from each tail of the distributions (e.g. 0.1 removes 10% from each tail).

Value

A list with PCAout, the results of PCA, WASSVARIANCE (not computed, always 0) and INERTIA, the total inertia.

References

Verde, R., Irpino, A., Balzanella, A. (2016). Dimension reduction techniques for distributional symbolic data. IEEE Transactions on Cybernetics, 46(2), 344-355. doi:10.1109/TCYB.2015.2389653

See Also

WH.MultiplePCA, WH.plot_multiple_Spanish.funs

Examples

```
results <- WH.1d.PCA(data = BLOOD, var = 1, listaxes = c(1:2))
```

WH.bind *Bind matrices of histograms*

Description

WH.bind.row binds by rows two MatH objects with the same number of columns, WH.bind.col binds by columns two MatH objects with the same number of rows. WH.bind does one of the two according to byrow.

Usage

```
WH.bind.row(object1, object2)

WH.bind.col(object1, object2)

WH.bind(object1, object2, byrow)
```

Arguments

object1, object2	Two MatH objects.
byrow	A logical value: if TRUE the objects are bound by rows, otherwise by columns.

Value

A MatH object.

Examples

```
WH.bind.row(BLOOD[1:3, ], BLOOD[5:8, ])
WH.bind.col(BLOOD[1:10, 1], BLOOD[1:10, 3])
WH.bind(BLOOD[1:10, 1], BLOOD[1:10, 3], byrow = FALSE)
```

WH.mat.prod *Product of two matrices of histograms*

Description

Matrix product of two MatH objects, where the product of two cells is the dot product of the histograms inducing the L2 Wasserstein metric (see dotpW).

Usage

```
## S4 method for signature 'MatH,MatH'
WH.mat.prod(object1, object2, traspose1 = FALSE, traspose2 = FALSE)
```

Arguments

object1, object2	Two MatH objects with conformable dimensions.
traspose1, traspose2	Logical values (default FALSE): if TRUE the corresponding matrix is transposed.

Value

A numeric matrix.

Examples

```
MAT <- WH.mat.prod(BLOOD[1:5, ], BLOOD[6:10, ], traspose1 = TRUE)
```

`WH.mat.sum` *Sum of two matrices of histograms*

Description

Sums two `MatH` objects of the same dimensions, cell by cell, by summing the quantile functions of the histograms (a sum consistent with the L2 Wasserstein metric).

Usage

```
WH.mat.sum(object1, object2)
```

Arguments

`object1`, `object2`
: Two `MatH` objects with the same dimensions.

Value

A `MatH` object.

Examples

```
MAT <- WH.mat.sum(BLOOD[1:5, ], BLOOD[6:10, ])
```

`WH.MultiplePCA` *Principal component analysis of a set of histogram variables*

Description

(Beta version) Principal component analysis of several histogram variables based on the L2 Wasserstein distance: a multiple factor analysis (`MFA`) of the quantiles of the histograms, where each variable is a group of quantiles. It extends `WH.1d.PCA`.

Usage

```
WH.MultiplePCA(data, list.of.vars, quantiles = 10, outl = 0, graph = TRUE)
```

Arguments

data	A MatH object.
list.of.vars	A vector of integers, the columns of the variables to analyse.
quantiles	An integer, the number of quantiles used in the analysis (default 10).
outl	A number between 0 (default) and 0.5, the probability mass removed from each tail of the distributions (e.g. 0.1 removes 10% from each tail).
graph	A logical value (default TRUE): if TRUE the plots of MFA are drawn.

Value

The results of MFA.

See Also

WH.plot_multiple_Spanish.funs, WH.plot_multiple_indivs

Examples

```
res <- WH.MultiplePCA(BLOOD, list.of.vars = c(1:3), graph = FALSE)
```

WH.plot_multiple_indivs

Plot the histograms of the individuals on a factorial plane

Description

(Beta version) Plots the histograms (or densities) of one variable for each individual, placed at the coordinates of the individuals on a factorial plane of WH.MultiplePCA.

Usage

```
WH.plot_multiple_indivs(
  data,
  res,
  axes = c(1, 2),
  indiv = 0,
  var = 1,
  strx = 0.1,
  stry = 1/8,
  HISTO = TRUE,
  coor = 0,
  stat = "mean"
)
```

Arguments

data	A MatH object, the data of the analysis.
res	The results of WH.MultiplePCA.
axes	A vector of two integers, the factorial axes (default c(1, 2)).
indiv	A vector of integers, the individuals (rows) to plot. Default 0: all the individuals.
var	An integer, the variable to plot.
strx	A number, the average width of the drawn distributions as a fraction of the range of the coordinates of the individuals on the x axis (default 0.1: on average, the domain of a distribution takes one tenth of the x axis). All the distributions share the same scale, so their spreads remain comparable.
stry	A number, the height of the highest density as a fraction of the range of the coordinates of the individuals on the y axis (default 1/8). All the densities share the same scale; spikes (bins of almost zero width) are avoided by drawing histograms with very narrow bins on 30 equal-width bins.
HISTO	A logical value: if TRUE (default) histograms are drawn, otherwise smoothed densities.
coor	A matrix of coordinates of the individuals. Default 0: the coordinates in res.
stat	The statistic used to label and colour the histograms in the second plot: "mean" (default), "std", "skewness" or "kurtosis".

Details

The distributions are measured on the scale of the original variable, while the factorial plane has the scale of the principal components: strx and stry set the size of the drawn distributions automatically, as fractions of the ranges of the coordinates of the individuals. Each distribution is drawn with its mean at the coordinates of the individual.

Value

The two plots are drawn on the current device and a list with the two ggplot objects is returned invisibly.

Examples

```
results <- WH.MultiplePCA(BLOOD, list.of.vars = c(1:3), graph = FALSE)
WH.plot_multiple_indivs(BLOOD, results, axes = c(1, 2), var = 1, HISTO = TRUE)
WH.plot_multiple_indivs(BLOOD, results, axes = c(1, 2), var = 1, HISTO = FALSE)
```

`WH.plot_multiple_Spanish.funs` *Plot the variables of a PCA of histogram variables (Spanish-fan plot)*

Description

Plots the quantiles of the histogram variables on a factorial plane after `WH.MultiplePCA` (or `WH.1d.PCA`): the quantiles of each variable form a fan.

Usage

```
WH.plot_multiple_Spanish.funs(
  res,
  axes = c(1, 2),
  var = 1,
  LABS = TRUE,
  multi = TRUE,
  corplot = TRUE
)
```

Arguments

`res`	The results of `WH.MultiplePCA` or of `WH.1d.PCA`.
`axes`	A vector of two integers, the factorial axes (default `c(1, 2)`).
`var`	A vector of integers, the variables to plot.
`LABS`	A logical value (default `TRUE`): if `TRUE` the quantiles are labelled.
`multi`	A logical value: `TRUE` (default) if `res` comes from `WH.MultiplePCA`, `FALSE` if it comes from `WH.1d.PCA`.
`corplot`	A logical value: if `TRUE` (default) the correlations are plotted (circle of correlations), otherwise the coordinates.

Value

The plot is drawn on the current device and the `ggplot` object is returned invisibly.

Examples

```
res <- WH.MultiplePCA(BLOOD, list.of.vars = c(1:3), graph = FALSE)
WH.plot_multiple_Spanish.funs(res, axes = c(1, 2), var = c(1:3))
```

WH.regression.GOF *Goodness of fit of a two-component regression of histogram variables*

Description

Computes goodness of fit indices of the predictions of a WH.regression.two.components model.

Usage

```
WH.regression.GOF(observed, predicted)
```

Arguments

observed	A MatH object with one column, the observed histograms.
predicted	A MatH object with one column, the predicted histograms.

Value

A list with the root mean squared error based on the L2 Wasserstein distance (RMSE_W), the OMEGA index and the pseudo-R2 index (PSEUDOR2, with the details of the decomposition of the sum of squares).

References

Irpino, A., Verde, R. (2015). Linear regression for numeric symbolic variables: a least squares approach based on Wasserstein Distance. Advances in Data Analysis and Classification, 9, 81-106. doi:10.1007/s1163401501977

Examples

```
model.parameters <- WH.regression.two.components(data = BLOOD, Yvar = 1, Xvars = c(2:3))
Predicted.BLOOD <- WH.regression.two.components.predict(
  data = BLOOD[, 2:3],
  parameters = model.parameters
)
WH.regression.GOF(observed = BLOOD[, 1], predicted = Predicted.BLOOD)
```

WH.regression.two.components *Two-component regression of histogram variables*

Description

Linear regression of a histogram variable on a set of histogram variables, based on the decomposition of the L2 Wasserstein distance into a component for the means and a component for the centred distributions. The parameters are estimated by least squares (non-negative least squares for the centred component).

Usage

```
WH.regression.two.components(data, Yvar, Xvars, simplify = FALSE, qua = 20)
```

Arguments

data	A MatH object.
Yvar	An integer, the column of the dependent variable.
Xvars	A vector of integers, the columns of the explanatory variables.
simplify	A logical value (default FALSE). If TRUE, the histograms are approximated by qua equally spaced quantiles, for faster computations.
qua	An integer, the number of quantiles when simplify = TRUE (default 20).

Value

A named vector with the estimated parameters: the intercept and the coefficients of the means (prefix AV_) and the coefficients of the centred distributions (prefix CEN_).

References

Irpino, A., Verde, R. (2015). Linear regression for numeric symbolic variables: a least squares approach based on Wasserstein Distance. Advances in Data Analysis and Classification, 9, 81-106. doi:10.1007/s11634015019777

See Also

WH.regression.two.components.predict, WH.regression.GOF

Examples

```
model.parameters <- WH.regression.two.components(data = BLOOD, Yvar = 1, Xvars = c(2:3))
model.parameters
```

WH.regression.two.components.predict

Predictions of a two-component regression of histogram variables

---

Description

Predicts the histograms of the dependent variable using the parameters estimated by WH.regression.two.components.

Usage

```
WH.regression.two.components.predict(data, parameters)
```

Arguments

data	A MatH object with the explanatory variables (in the same order used for the estimation).
parameters	The named vector of parameters returned by WH.regression.two.components.

Value

A MatH object with one column, the predicted histograms.

See Also

WH.regression.GOF

Examples

```
model.parameters <- WH.regression.two.components(data = BLOOD, Yvar = 1, Xvars = c(2:3))
Predicted.BLOOD <- WH.regression.two.components.predict(
  data = BLOOD[, 2:3],
  parameters = model.parameters
)
```

WH.var.covar *Sum of squares, covariance and correlation of histogram variables*

Description

Compute the matrix of the sums of squared deviations from the mean (WH.SSQ), the variance-covariance matrix (WH.var.covar) and the correlation matrix (WH.correlation) of the variables of a MatH object, according to the L2 Wasserstein metric. The functions with suffix 2 compute the same (rectangular) matrices between the variables of two MatH objects with the same number of rows.

Usage

```
## S4 method for signature 'MatH'
WH.SSQ(object, w = numeric(0))

## S4 method for signature 'MatH'
WH.var.covar(object, w = numeric(0))

## S4 method for signature 'MatH'
WH.correlation(object, w = numeric(0))

## S4 method for signature 'MatH,MatH'
WH.SSQ2(object1, object2, w = numeric(0))

## S4 method for signature 'MatH,MatH'
```

```
WH.var.covar2(object1, object2, w = numeric(0))

## S4 method for signature 'MatH,MatH'
WH.correlation2(object1, object2, w = numeric(0))
```

Arguments

object	A MatH object.
object1, object2	Two MatH objects with the same number of rows.
w	A vector of non-negative weights, one for each row (individual). Default: equal weights.

Value

A numeric matrix.

References

Irpino, A., Verde, R. (2015) *Basic statistics for distributional symbolic variables: a new metric-based approach*, Advances in Data Analysis and Classification, 9(2), 143-175. doi:10.1007/s11634-01401764

Examples

```
WH.SSQ(BLOOD)
WH.var.covar(BLOOD)
WH.correlation(BLOOD)
# weighted, with random weights
RN <- runif(get.MatH.nrows(BLOOD))
WH.var.covar(BLOOD, w = RN)
# between two sets of variables
WH.correlation2(BLOOD[, 1], BLOOD[, 2:3])
```

WH.vec.mean *Weighted sum and mean of the histograms of a matrix*

Description

WH.vec.sum and WH.vec.mean compute the (weighted) sum and the (weighted) mean of all the histograms of a MatH object, consistently with the L2 Wasserstein metric: the result is the histogram whose quantile function is the (weighted) sum or mean of the quantile functions. The mean is the Wasserstein barycenter of the histograms.

Usage

```
## S4 method for signature 'MatH'
WH.vec.sum(object, w = numeric(0))

## S4 method for signature 'MatH'
WH.vec.mean(object, w = numeric(0))
```

Arguments

object	A MatH object.
w	A vector of non-negative weights, one for each cell of the matrix. Default: equal weights.

Value

A distributionH object.

Examples

```
WH.vec.mean(BLOOD)
WH.vec.sum(BLOOD)
# weighted mean with random weights
RN <- runif(get.MatH.nrows(BLOOD) * get.MatH.ncols(BLOOD))
WH.vec.mean(BLOOD, w = RN)
```

WH_2d_Adaptive_Kohonen_maps

Batch Kohonen self-organising maps of histogram data with adaptive distances

---

Description

Batch Kohonen self-organising 2d map of the individuals of a MatH object, based on adaptive L2 Wasserstein distances.

Usage

```
WH_2d_Adaptive_Kohonen_maps(
  x,
  net = list(xdim = 4, ydim = 3, topo = c("rectangular")),
  kern.param = 2,
  TMAX = -9999,
  Tmin = -9999,
  niter = 30,
  repetitions,
  simplify = FALSE,
  qua = 10,
```

```
  standardize = FALSE,
  schema = 6,
  init.weights = "EQUAL",
  weight.sys = "PROD",
  theta = 2,
  Wfix = FALSE,
  verbose = FALSE,
  atleast = 2
)
```

Arguments

x	A MatH object.
net	A list describing the topology of the map: list(xdim = <rows>, ydim = <columns>, topo = "rectangular" or "hexagonal"), see somgrid. Default: a 4 x 3 rectangular map.
kern.param	The parameter of the RBF kernel (default 2).
TMAX, Tmin	The initial and final values of the neighbourhood radius. When negative (default), they are computed from the size of the map.
niter	The maximum number of iterations (default 30).
repetitions	The number of repetitions of the algorithm (default 5); the best solution is returned.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
schema	An integer, the system of weights: 1 = a weight for each variable; 2 = a weight for the position and one for the variability component of each variable; 3 and 4 = as 1 and 2, with a different set of weights for each neuron (product of the weights of each variable equal to one); 5 and 6 = as 1 and 2, with a different set of weights for each neuron (product of all the weights equal to one, default 6).
init.weights	The initial weights: "EQUAL" (default), "PROP" or "RANDOM".
weight.sys	The constraint on the weights: "PROD" (default) or "SUM".
theta	A number, the exponent of the weights when weight.sys = "SUM" (default 2).
Wfix	A logical value (default FALSE). If TRUE the weights are not updated (non adaptive distances).
verbose	A logical value (default FALSE). If TRUE, details of the computations are printed.
atleast	An integer (default 2): solutions where fewer than atleast neurons attract individuals are discarded.

Details

Batch Self-Organising Maps (BSOM) for histogram data, with a cost function based on the L2 Wasserstein distance, i.e. the Euclidean distance between quantile functions. The adaptive version introduces an automatic system of weights in the cost function, to take into account the different effect of the variables on the map.

Value

A list with:

`solution` The best solution among the repetitions: a list with `MAP` (the grid of the map), `IDX` (the neuron of each individual), `cardinality` (the number of individuals for each neuron), `proto` (a `MatH` object with the prototypes of the neurons), `Crit` (the criterion), `weights.comp` (the weights of the components of the variables) and `Weight.sys` (the system of weights).

`quality` The proportion of the total sum of squares explained by the map (the higher the better).

and further details on the parameters of the algorithm. The final map is plotted on the current graphic device.

References

Irpino A., Verde R., De Carvalho F.A.T. (2012). Batch self organizing maps for interval and histogram data. In: Proceedings of COMPSTAT 2012, ISI/IASC, 143-154.

See Also

`WH_2d_Kohonen_maps`

Examples

```
results <- WH_2d_Adaptive_Kohonen_maps(
  x = BLOOD,
  net = list(xdim = 2, ydim = 3, topo = c("rectangular")),
  repetitions = 2, simplify = TRUE,
  qua = 10, standardize = TRUE
)
```

`WH_2d_Kohonen_maps` *Batch Kohonen self-organising maps of histogram data*

Description

Batch Kohonen self-organising 2d map of the individuals of a `MatH` object, based on the L2 Wasserstein distance.

Usage

```
WH_2d_Kohonen_maps(
  x,
  net = list(xdim = 4, ydim = 3, topo = c("rectangular")),
  kern.param = 2,
  TMAX = 2,
  Tmin = 0.2,
  niter = 30,
  repetitions = 5,
  simplify = FALSE,
  qua = 10,
  standardize = FALSE,
  verbose = FALSE
)
```

Arguments

x	A MatH object.
net	A list describing the topology of the map: list(xdim = <rows>, ydim = <columns>, topo = "rectangular" or "hexagonal"), see somgrid. Default: a 4 x 3 rectangular map.
kern.param	The parameter of the RBF kernel (default 2).
TMAX, Tmin	The initial and final values of the neighbourhood radius (defaults 2 and 0.2).
niter	The maximum number of iterations (default 30).
repetitions	The number of repetitions of the algorithm (default 5); the best solution is returned.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
verbose	A logical value (default FALSE). If TRUE, details of the computations are printed.

Details

Batch Self-Organising Maps (BSOM) for histogram data, with a cost function based on the L2 Wasserstein distance, i.e. the Euclidean distance between quantile functions. The adaptive version introduces an automatic system of weights in the cost function, to take into account the different effect of the variables on the map.

Value

A list with:

`solution` The best solution among the repetitions: a list with `MAP` (the grid of the map), `IDX` (the neuron of each individual), `cardinality` (the number of individuals for each neuron), `proto` (a `MatH` object with the prototypes of the neurons), `Crit` (the criterion), `weights.comp` (the weights of the components of the variables) and `Weight.sys` (the system of weights).

`quality` The proportion of the total sum of squares explained by the map (the higher the better).

and further details on the parameters of the algorithm. The final map is plotted on the current graphic device.

References

Irpino A., Verde R., De Carvalho F.A.T. (2012). Batch self organizing maps for interval and histogram data. In: Proceedings of COMPSTAT 2012, ISI/IASC, 143-154.

See Also

`WH_2d_Adaptive_Kohonen_maps`

Examples

```
results <- WH_2d_Kohonen_maps(
  x = BLOOD,
  net = list(xdim = 2, ydim = 3, topo = c("rectangular")),
  repetitions = 2, simplify = TRUE,
  qua = 10, standardize = TRUE
)
```

WH_adaptive.kmeans *K-means clustering of histogram data with adaptive distances*

Description

K-means of the individuals of a `MatH` object based on adaptive L2 Wasserstein distances: the algorithm also estimates a weight for each variable (or for each component of the variables, and/or for each cluster).

Usage

```
WH_adaptive.kmeans(
  x,
  k,
  schema = 1,
  init,
  rep,
  simplify = FALSE,
```

```
  qua = 10,
  standardize = FALSE,
  weight.sys = "PROD",
  theta = 2,
  init.weights = "EQUAL",
  verbose = FALSE
)
```

Arguments

x	A MatH object.
k	An integer, the number of clusters.
schema	An integer, the system of weights: 1 = a weight for each variable (default); 2 = a weight for the position and one for the variability component of each variable; 3 = as 1, with a different set of weights for each cluster; 4 = as 2, with a different set of weights for each cluster.
init	Not used.
rep	An integer, the number of repetitions of the algorithm (with random starting partitions); the best solution is returned. Default 5.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
weight.sys	The constraint on the weights: "PROD" (default), the product of the weights is one; "SUM", the weights sum to one.
theta	A number, the exponent of the weights when weight.sys = "SUM" (default 2).
init.weights	The initial weights: "EQUAL" (default) or "RANDOM".
verbose	A logical value (default FALSE). If TRUE, details of the computations are printed.

Value

A list with IDX (the cluster of each individual), cardinality (the size of each cluster), proto (a MatH object with the prototypes), weights (the weights of the variables or components), Crit (the criterion), TOTSSQ, BSQ, WSQ (total, between and within sums of squares) and quality (the proportion of the total sum of squares explained by the partition).

References

Irpino A., Verde R., De Carvalho F.A.T. (2014). Dynamic clustering of histogram data based on adaptive squared Wasserstein distances. Expert Systems with Applications, 41, 3351-3366. doi:10.1016/j.eswa.2013.12.001

See Also

WH_kmeans, WH_adaptive_fcmeans

Examples

```
results <- WH_adaptive.kmeans(x = BLOOD, k = 2, rep = 10,
            simplify = TRUE, qua = 10, standardize = TRUE)
results$weights
```

`WH_adaptive_fcmeans` *Fuzzy c-means clustering of histogram data with adaptive distances*

Description

Fuzzy c-means of the individuals of a `MatH` object based on adaptive L2 Wasserstein distances: the algorithm also estimates a weight for each variable (or for each component of the variables, and/or for each cluster).

Usage

```
WH_adaptive_fcmeans(
  x,
  k = 5,
  schema,
  m = 1.6,
  rep,
  simplify = FALSE,
  qua = 10,
  standardize = FALSE,
  init.weights = "EQUAL",
  weight.sys = "PROD",
  theta = 2,
  verbose = FALSE
)
```

Arguments

`x`	A `MatH` object.
`k`	An integer, the number of clusters.
`schema`	An integer, the system of weights: 1 = a weight for each variable; 2 = a weight for the position and one for the variability component of each variable; 3 = as 1, with a different set of weights for each cluster; 4 = as 2, with a different set of weights for each cluster.
`m`	A number greater than 1, the fuzziness coefficient (default 1.6).
`rep`	An integer, the number of repetitions of the algorithm (with random starting partitions); the best solution is returned. Default 5.
`simplify`	A logical value (default `FALSE`). If `TRUE`, each histogram is approximated by `qua` equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.

qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
init.weights	The initial weights: "EQUAL" (default) or "RANDOM".
weight.sys	The constraint on the weights: "PROD" (default), the product of the weights is one; "SUM", the weights sum to one.
theta	A number, the exponent of the weights when weight.sys = "SUM" (default 2).
verbose	A logical value (default FALSE). If TRUE, details of the computations are printed.

Value

A list with:

solution	The best solution among the repetitions: a list with var.weights (the weights), membership (the matrix of membership degrees), IDX (the crisp assignment of each individual), cardinality (the size of each crisp cluster), proto (a MatH object with the prototypes) and Crit (the criterion).
quality	The proportion of the total sum of squares explained by the partition (the higher the better).

and further details on the decomposition of the sum of squares.

See Also

WH_fcmeans, WH_adaptive.kmeans

Examples

```
results <- WH_adaptive_fcmeans(
  x = BLOOD, k = 2, schema = 4, m = 1.5, rep = 3, simplify = TRUE,
  qua = 10, standardize = TRUE, init.weights = "EQUAL", weight.sys = "PROD"
)
results$solution$var.weights
```

WH_fcmeans *Fuzzy c-means clustering of histogram data*

Description

Fuzzy c-means of the individuals of a MatH object, based on the L2 Wasserstein distance between histograms.

Usage

```
WH_fcmeans(x, k, m = 1.6, rep, simplify = FALSE, qua = 10, standardize = FALSE)
```

Arguments

x	A MatH object.
k	An integer, the number of clusters.
m	A number greater than 1, the fuzziness coefficient (default 1.6).
rep	An integer, the number of repetitions of the algorithm (with random starting partitions); the best solution is returned. Default 5.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.

Value

A list with:

solution	The best solution among the repetitions: a list with membership (the matrix of membership degrees), IDX (the crisp assignment of each individual), cardinality (the size of each crisp cluster), proto (a MatH object with the prototypes) and Crit (the criterion).
quality	The proportion of the total sum of squares explained by the partition (the higher the better).

and further details on the decomposition of the sum of squares.

See Also

WH_adaptive_fcmeans, WH_kmeans

Examples

```
results <- WH_fcmeans(x = BLOOD, k = 2, m = 1.5, rep = 10,
                      simplify = TRUE, qua = 10, standardize = TRUE)
results$solution$membership
```

WH_hclust *Hierarchical clustering of histogram data*

Description

Hierarchical clustering of the individuals of a MatH object, based on the L2 Wasserstein distance. It is a wrapper of hclust.

Usage

```
WH_hclust(
  x,
  simplify = FALSE,
  qua = 10,
  standardize = FALSE,
  distance = "WDIST",
  method = "complete"
)
```

Arguments

x	A MatH object.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
distance	The distance: only "WDIST", the L2 Wasserstein distance, is available.
method	The agglomeration method, as in hclust: one of "ward.D", "ward.D2", "single", "complete" (default), "average", "mcquitty", "median" or "centroid".

Value

An object of class hclust.

References

Irpino A., Verde R. (2006). A new Wasserstein based distance for the hierarchical clustering of histogram symbolic data. In: Batanjeli et al. (eds.) Data Science and Classification, IFCS 2006, Springer, 185-192.

See Also

WH_MAT_DIST

Examples

```
results <- WH_hclust(x = BLOOD, simplify = TRUE, method = "complete")
plot(results)
cutree(results, k = 5)
```

WH_kmeans *K-means clustering of histogram data*

Description

K-means (dynamic clustering) of the individuals of a MatH object, based on the L2 Wasserstein distance between histograms.

Usage

```
WH_kmeans(
  x,
  k,
  rep = 5,
  simplify = FALSE,
  qua = 10,
  standardize = FALSE,
  verbose = FALSE
)
```

Arguments

x	A MatH object.
k	An integer, the number of clusters.
rep	An integer, the number of repetitions of the algorithm (with random starting partitions); the best solution is returned. Default 5.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.
verbose	A logical value (default FALSE). If TRUE, details of the computations are printed.

Value

A list with:

solution	The best solution among the repetitions (minimum criterion): a list with IDX (the cluster of each individual), cardinality (the size of each cluster), centers (a MatH object with the prototypes) and Crit (the within-cluster sum of squares).
quality	The proportion of the total sum of squares explained by the partition (the higher the better).

References

Irpino A., Verde R., Lechevallier Y. (2006). Dynamic clustering of histograms using Wasserstein metric. In: Rizzi A., Vichi M. (eds.) COMPSTAT 2006 - Advances in computational statistics, Physica-Verlag, 869-876.

See Also

WH_adaptive.kmeans, WH_fcmeans, WH_hclust

Examples

```
results <- WH_kmeans(
  x = BLOOD, k = 2, rep = 10, simplify = TRUE,
  qua = 10, standardize = TRUE
)
results$solution$IDX
results$quality
```

WH_MAT_DIST *L2 Wasserstein distance matrix*

Description

Computes the matrix of the L2 Wasserstein distances between the individuals (rows) of a MatH object, summing the squared distances over the variables.

Usage

```
WH_MAT_DIST(x, simplify = FALSE, qua = 10, standardize = FALSE)
```

Arguments

x	A MatH object.
simplify	A logical value (default FALSE). If TRUE, each histogram is approximated by qua equally spaced quantiles: the computations are much faster for large datasets, with a small approximation error.
qua	An integer, the number of quantiles used when simplify = TRUE (default 10).
standardize	A logical value (default FALSE). If TRUE, the variables are standardised by their Wasserstein-based standard deviation.

Value

A symmetric matrix of L2 Wasserstein distances.

References

Irpino A., Verde R. (2006). A new Wasserstein based distance for the hierarchical clustering of histogram symbolic data. In: Batanjeli et al. (eds.) Data Science and Classification, IFCS 2006, Springer, 185-192.

See Also

WassSqDistH, WH_hclust

Examples

```
DMAT <- WH_MAT_DIST(x = BLOOD, simplify = TRUE)
```

Index

∗ **datasets**
Age_Pyramids_2014, 5
Agronomique, 6
BLOOD, 7
BloodBRITO, 8
China_Month, 9
China_Seas, 10
OzoneFull, 27
OzoneH, 28
RetHTS, 34
stations_coordinates, 38
*,distributionH,distributionH-method *(Arith-distributionH)*, 6
*,distributionH,numeric-method *(Arith-distributionH)*, 6
*,numeric,distributionH-method *(Arith-distributionH)*, 6
+,distributionH,distributionH-method *(Arith-distributionH)*, 6
+,distributionH,numeric-method *(Arith-distributionH)*, 6
+,numeric,distributionH-method *(Arith-distributionH)*, 6
-,distributionH,distributionH-method *(Arith-distributionH)*, 6
-,distributionH,numeric-method *(Arith-distributionH)*, 6
-,numeric,distributionH-method *(Arith-distributionH)*, 6
[,MatH,ANY,ANY,ANY-method *(extract-MatH)*, 17
[,MatH-method *(extract-MatH)*, 17

Age_Pyramids_2014, 5
Agronomique, 6
Arith-distributionH, 6

BLOOD, 7
BloodBRITO, 8

Center.cell.MatH, 8
Center.cell.MatH,MatH-method *(Center.cell.MatH)*, 8
checkEmptyBins, 9, *14*
checkEmptyBins,distributionH-method *(checkEmptyBins)*, 9
checkEmptyBins,MatH-method *(checkEmptyBins)*, 9
China_Month, 9, *38*
China_Seas, 10, *38*
compP, *4*, 11, *12*
compP,distributionH,numeric-method *(compP)*, 11
compQ, *4*, *11*, 11
compQ,distributionH,numeric-method *(compQ)*, 11
crwtransform, 12
crwtransform,distributionH-method *(crwtransform)*, 12

data2hist, *4*, 13, *15, 16*
distributionH, *4*, *9*, *13*, *40*
distributionH *(distributionH-class)*, 14
distributionH-class, 14
dotpW, 15, *44*
dotpW,distributionH,distributionH-method *(dotpW)*, 15
dotpW,distributionH,numeric-method *(dotpW)*, 15
dotpW,numeric,distributionH-method *(dotpW)*, 15
DouglasPeucker, *13*, 16, *35, 36*

extract-MatH, 17

get.cell.MatH, 17
get.cell.MatH,MatH,numeric,numeric-method *(get.cell.MatH)*, 17
get.distr, 18, *19*
get.distr,distributionH-method *(get.distr)*, 18

get.histo, *18*, 19
get.histo,distributionH-method *(get.histo)*, 19
get.m, 19, *22*
get.m,distributionH-method *(get.m)*, 19
get.MatH.main.info, 20, *26*
get.MatH.main.info,MatH-method *(get.MatH.main.info)*, 20
get.MatH.ncols *(get.MatH.main.info)*, 20
get.MatH.ncols,MatH-method *(get.MatH.main.info)*, 20
get.MatH.nrows *(get.MatH.main.info)*, 20
get.MatH.nrows,MatH-method *(get.MatH.main.info)*, 20
get.MatH.rownames *(get.MatH.main.info)*, 20
get.MatH.rownames,MatH-method *(get.MatH.main.info)*, 20
get.MatH.stats, 21
get.MatH.stats,MatH-method *(get.MatH.stats)*, 21
get.MatH.varnames *(get.MatH.main.info)*, 20
get.MatH.varnames,MatH-method *(get.MatH.main.info)*, 20
get.s, *20*, 21
get.s,distributionH-method *(get.s)*, 21

hclust, *61*, *62*
hist, *13*
HistDAWass *(HistDAWass-package)*, 4
HistDAWass-package, 4
histogram, *13*
HTS-class, 22
HTS.exponential.smoothing, *4*, *22*, 23, *24*
HTS.moving.averages, *4*, *22*, *23*, 23
HTS.predict.knn, *4*, *22*, 24

initialize,distributionH-method *(distributionH-class)*, 14
initialize,HTS-method *(HTS-class)*, 22
initialize,MatH-method *(MatH-class)*, 25
initialize,TdistributionH-method *(TdistributionH-class)*, 40
initialize,TMatH-method *(TMatH-class)*, 41
is.registeredMH *(registerMH)*, 33
is.registeredMH,MatH-method *(registerMH)*, 33

kurtH, *4*, 25, *27*, *37*, *39*
kurtH,distributionH-method *(kurtH)*, 25

MatH, *4*
MatH *(MatH-class)*, 25
MatH-class, 25
meanH, *4*, *15*, *19*, *20*, *25*, 27, *37*, *39*
meanH,distributionH-method *(meanH)*, 27
MFA, *45*, *46*

OzoneFull, 27
OzoneH, 28

PCA, *43*
plot,distributionH-method *(plot-distributionH)*, 28
plot,HTS-method *(plot-HTS)*, 29
plot,MatH-method *(plot-MatH)*, 30
plot,TdistributionH-method *(plot-TdistributionH)*, 30
plot-distributionH, 28
plot-HTS, 29
plot-MatH, 30
plot-TdistributionH, 30
plot_errors, *31*, 32
plotPredVsObs, 31, *32*

register, 33
register,distributionH,distributionH-method *(register)*, 33
registerMH, 33
registerMH,MatH-method *(registerMH)*, 33
RetHTS, *22*, 34
rQQ, 35
rQQ,distributionH,distributionH-method *(rQQ)*, 35

set.cell.MatH *(get.cell.MatH)*, 17
set.cell.MatH,distributionH,MatH,numeric,numeric-method *(get.cell.MatH)*, 17
ShortestDistance, *16*, 35
show,distributionH-method *(show-distributionH)*, 36
show,MatH-method *(show-MatH)*, 37
show-distributionH, 36
show-MatH, 37
skewH, *4*, *25*, *27*, 37, *39*
skewH,distributionH-method *(skewH)*, 37
somgrid, *54*, *56*

stations_coordinates, *10*, 38
stdH, *4*, *15*, *21*, *22*, *25*, *27*, *37*, 38
stdH,distributionH-method *(*stdH*)*, 38
subsetHTS, *22*, 39
subsetHTS,HTS,numeric,numeric-method *(*subsetHTS*)*, 39
summaryHTS, 39

TdistributionH-class, 40
TMatH-class, 41

WassSqDistH, *4*, 41, *65*
WassSqDistH,distributionH,distributionH-method *(*WassSqDistH*)*, 41
WH.1d.PCA, *4*, 42, *45*, *48*
WH.bind, 43
WH.bind,MatH,MatH-method *(*WH.bind*)*, 43
WH.bind,MatH-method *(*WH.bind*)*, 43
WH.bind.col *(*WH.bind*)*, 43
WH.bind.col,MatH,MatH-method *(*WH.bind*)*, 43
WH.bind.col,MatH-method *(*WH.bind*)*, 43
WH.bind.row *(*WH.bind*)*, 43
WH.bind.row,MatH,MatH-method *(*WH.bind*)*, 43
WH.bind.row,MatH-method *(*WH.bind*)*, 43
WH.correlation *(*WH.var.covar*)*, 51
WH.correlation,MatH-method *(*WH.var.covar*)*, 51
WH.correlation2 *(*WH.var.covar*)*, 51
WH.correlation2,MatH,MatH-method *(*WH.var.covar*)*, 51
WH.correlation2,MatH-method *(*WH.var.covar*)*, 51
WH.mat.prod, 44
WH.mat.prod,MatH,MatH-method *(*WH.mat.prod*)*, 44
WH.mat.prod,MatH-method *(*WH.mat.prod*)*, 44
WH.mat.sum, 45
WH.mat.sum,MatH,MatH-method *(*WH.mat.sum*)*, 45
WH.mat.sum,MatH-method *(*WH.mat.sum*)*, 45
WH.MultiplePCA, *4*, *43*, 45, *46–48*
WH.plot_multiple_indivs, *46*, 46
WH.plot_multiple_Spanish.funs, *43*, *46*, 48
WH.regression.GOF, *32*, 49, *50*, *51*
WH.regression.two.components, *4*, *49*, 49, *50*, *51*
WH.regression.two.components.predict, *31*, *32*, *50*, 50
WH.SSQ *(*WH.var.covar*)*, 51
WH.SSQ,MatH-method *(*WH.var.covar*)*, 51
WH.SSQ2 *(*WH.var.covar*)*, 51
WH.SSQ2,MatH,MatH-method *(*WH.var.covar*)*, 51
WH.SSQ2,MatH-method *(*WH.var.covar*)*, 51
WH.var.covar, *4*, *26*, 51
WH.var.covar,MatH-method *(*WH.var.covar*)*, 51
WH.var.covar2 *(*WH.var.covar*)*, 51
WH.var.covar2,MatH,MatH-method *(*WH.var.covar*)*, 51
WH.var.covar2,MatH-method *(*WH.var.covar*)*, 51
WH.vec.mean, *4*, *26*, 52
WH.vec.mean,MatH-method *(*WH.vec.mean*)*, 52
WH.vec.sum *(*WH.vec.mean*)*, 52
WH.vec.sum,MatH-method *(*WH.vec.mean*)*, 52
WH_2d_Adaptive_Kohonen_maps, *4*, 53, *57*
WH_2d_Kohonen_maps, *4*, *55*, 55
WH_adaptive.kmeans, *4*, 57, *60*, *64*
WH_adaptive_fcmeans, *4*, *58*, 59, *61*
WH_fcmeans, *4*, *60*, 60, *64*
WH_hclust, *4*, 61, *64*, *65*
WH_kmeans, *4*, *58*, *61*, 63
WH_MAT_DIST, *4*, *42*, *62*, 64