Package ‘agop’

November 30, 2023

Version 0.2.4

Date 2023-11-30

Title Aggregation Operators and Preordered Sets

Description Tools supporting multi-criteria and group decision making, including variable number of criteria, by means of aggregation operators, spread measures, fuzzy logic connectives, fusion functions, and preordered sets. Possible applications include, but are not limited to, quality management, scientometrics, software engineering, etc.

URL https://github.com/gagolews/agop/

BugReports https://github.com/gagolews/agop/issues

ByteCompile TRUE

Suggests igraph

Type Package

Depends R (>= 3.0), stats, grDevices, graphics

License LGPL (>= 3)

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation yes

Author Marek Gagolewski [aut, cre] (<https://orcid.org/0000-0003-0637-6028>), Anna Cena [ctb] (<https://orcid.org/0000-0001-8697-5383>)

Maintainer Marek Gagolewski <marek@gagolewski.com>

Repository CRAN

Date/Publication 2023-11-30 00:10:02 UTC

R topics documented:

agop-package *Aggregation Operators and Preordered Sets Package for R*

Description

Keywords: aggregation, bibliometrics, scientometrics, scientific impact, webometrics, preorders, binary relations, means, OWA, OWMax, OWMin, Hirsch's h-index, Egghe's g-index, variance, spread, decision making, fuzzy logic.

Acknowledgments: The development of the package in March-June 2013 was partially supported by the European Union from the resources of the European Social Fund, Project PO KL "Information technologies: Research and their interdisciplinary applications", agreement UDA-POKL.04.01.01-00-051/10-00.

Author(s)

Marek Gagolewski [aut,cre],
Anna Cena [ctb]

check_comonotonicity *Check If Two Vectors Are Comonotonic*

Description

This functions determines if two vectors have a common ordering permutation.

Usage

```
check_comonotonicity(x, y, incompatible_lengths = NA)
```

Arguments

`x`	numeric vector
`y`	numeric vector
`incompatible_lengths`	single logical value, value to return iff lengths of `x` and `y` differ

Details

Two vectors `x`, `y` of equal length n are *comonotonic*, if and only if there exists a permutation σ such that $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$ and $y_{\sigma(1)} \leq \cdots \leq y_{\sigma(n)}$. Thus, σ orders `x` and `y` simultaneously. Equivalently, `x` and `y` are comonotonic, iff $(x_i - x_j)(y_i - y_j) \geq 0$ for every `i,j`.

If there are missing values in `x` or `y`, the function returns `NA`.

Currently, the implemented algorithm has $O(n^2)$ time complexity.

Value

Returns a single logical value.

References

Grabisch M., Marichal J.-L., Mesiar R., Pap E., *Aggregation functions*, Cambridge University Press, 2009.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other binary_relations: `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_reflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

`d2owa_checkwts` *D2OWA Operators*

Description

Computes the D2OWA operator, i.e., the normalized L2 distance between a numeric vector and an OWA operator.

Usage

```
d2owa_checkwts(w)

d2owa(x, w = rep(1/length(x), length(x)))
```

Arguments

`w`	numeric vector of the same length as `x`, with elements in $[0, 1]$, and such that $\sum_i w_i = 1$; weights
`x`	numeric vector to be aggregated

Details

D2OWA is a symmetric spread measure. It is defined as `d2owa(x) == sqrt(mean((x-owa(x,w))^2))`. Not all weights, however, generate a proper function of this kind; `d2owa_checkwts` may be used to check that. For `d2owa`, if `w` is not appropriate, an error is thrown.

`w` is automatically normalized so that its elements sum up to 1.

Value

For `d2owa`, a single numeric value is returned. On the other hand, `d2owa_checkwts` returns a single logical value.

References

Gagolewski M., Spread measures and their relation to aggregation functions, European Journal of Operational Research 241(2), 2015, pp. 469-477. doi:10.1016/j.ejor.2014.08.034

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

Yager R.R., On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics* 18(1), 1988, pp. 183-190.

---

dpareto2_estimate_mle *Parameter Estimation in the Discretized Pareto-Type II Distribution Family (MLE)*

Description

Finds the maximum likelihood estimator of the Discretized Pareto Type-II distribution's shape parameter k and scale parameter s.

Usage

```
dpareto2_estimate_mle(
  x,
  k0 = 1,
  s0 = 1,
  kmin = 1e-04,
  smin = 1e-04,
  kmax = 100,
  smax = 100
)
```

Arguments

`x`	a non-negative numeric vector
`k0, s0`	initial points for the L-BFGS-B method
`kmin, kmax`	lower and upper bound for the shape parameter
`smin, smax`	lower and upper bound for the scale parameter

Details

Note that the maximum of the likelihood function might not exist for some input vectors. This estimator may have a large mean squared error.

Value

Returns a numeric vector with the following named components:

- `k` - estimated parameter of shape
- `s` - estimated parameter of scale

or `c(NA, NA)` if the maximum of the likelihood function could not be found.

See Also

Other DiscretizedPareto2: `rdpareto2()`

---

exp_test_ad *Anderson-Darling Test for Exponentiality*

Description

Performs an approximate Anderson-Darling goodness-of-fit test, which verifies the null hypothesis: Data follow an exponential distribution.

Usage

```
exp_test_ad(x)
```

Arguments

`x`	a non-negative numeric vector of data values

Details

Sample size should be not less than 3. Missing values are removed from `x` before applying the procedure.

The p-value is approximate: its distribution has been estimated by taking 2500000 MC samples. For performance and space reasons, the estimated distribution is recreated by a spline interpolation on a fixed number of points. As a result, the resulting p-value distribution might not necessarily be uniform for p»0.5.

Value

A list of the class `htest` is returned, just like in many other testing methods, see, e.g., `ks.test`.

References

Anderson T.W., Darling D.A., A Test of Goodness-of-Fit, *Journal of the American Statistical Association* 49, 1954, pp. 765-769.

See Also

pexp

Other Tests: pareto2_test_ad(), pareto2_test_f()

`fimplication_minimal` *Fuzzy Implications*

Description

Various fuzzy implications Each of these is a fuzzy logic generalization of the classical implication operation.

Usage

```
fimplication_minimal(x, y)

fimplication_maximal(x, y)

fimplication_kleene(x, y)

fimplication_lukasiewicz(x, y)

fimplication_reichenbach(x, y)

fimplication_fodor(x, y)

fimplication_goguen(x, y)

fimplication_goedel(x, y)

fimplication_rescher(x, y)

fimplication_weber(x, y)

fimplication_yager(x, y)
```

Arguments

x	numeric vector with elements in $[0, 1]$
y	numeric vector of the same length as x, with elements in $[0, 1]$

Details

A function $I : [0, 1] \times [0, 1] \to [0, 1]$ is a *fuzzy implication* if for all $x, y, x', y' \in [0, 1]$ it holds: (a) if $x \leq x'$, then $I(x, y) \geq I(x', y)$; (b) if $y \leq y'$, then $I(x, y) \leq I(x, y')$; (c) $I(1, 1) = 1$; (d) $I(0, 0) = 1$; (e) $I(1, 0) = 0$.

The minimal fuzzy implication is given by $I_0(x, y) = 1$ iff $x = 0$ or $y = 1$, and 0 otherwise.

The maximal fuzzy implication is given by $I_1(x, y) = 0$ iff $x = 1$ and $y = 0$, and 1 otherwise.

The Kleene-Dienes fuzzy implication is given by $I_{KD}(x, y) = max(1 - x, y)$.

The Lukasiewicz fuzzy implication is given by $I_L(x, y) = min(1 - x + y, 1)$.

The Reichenbach fuzzy implication is given by $I_{RB}(x, y) = 1 - x + xy$.

The Fodor fuzzy implication is given by $I_F(x, y) = 1$ iff $x \leq y$, and $max(1 - x, y)$ otherwise.

The Goguen fuzzy implication is given by $I_{GG}(x, y) = 1$ iff $x \leq y$, and y/x otherwise.

The Goedel fuzzy implication is given by $I_{GD}(x, y) = 1$ iff $x \leq y$, and y otherwise.

The Rescher fuzzy implication is given by $I_{RS}(x, y) = 1$ iff $x \leq y$, and 0 otherwise.

The Weber fuzzy implication is given by $I_W(x, y) = 1$ iff $x < 1$, and y otherwise.

The Yager fuzzy implication is given by $I_Y(x, y) = 1$ iff $x = 0$ and $y = 0$, and y^x otherwise.

Value

Numeric vector of the same length as `x` and `y`. The `i`th element of the resulting vector gives the result of calculating `I(x[i], y[i])`.

References

Klir G.J, Yuan B., *Fuzzy sets and fuzzy logic. Theory and applications*, Prentice Hall PTR, New Jersey, 1995.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other fuzzy_logic: `fnegation_yager()`, `tconorm_minimum()`, `tnorm_minimum()`

`fnegation_yager` *Fuzzy Negations*

Description

Various fuzzy negations. Each of these is a fuzzy logic generalization of the classical negation operation.

Usage

```
fnegation_yager(x)

fnegation_classic(x)

fnegation_minimal(x)

fnegation_maximal(x)
```

Arguments

x	numeric vector with elements in $[0, 1]$

Details

A function $N : [0, 1] \to [0, 1]$ is a *fuzzy implication* if for all $x, y \in [0, 1]$ it holds: (a) if $x \le y$, then $N(x) \ge N(y)$; (b) $N(1) = 0$; (c) $N(0) = 1$.

The classic fuzzy negation is given by $N_C(x) = 1 - x$.

The Yager fuzzy negation is given by $N_Y(x) = sqrt(1 - x^2)$.

The minimal fuzzy negation is given by $N_0(x, y) = 1$ iff $x = 0$, and 0 otherwise.

The maximal fuzzy negation is given by $N_1(x, y) = 1$ iff $x < 1$, and 0 otherwise.

Value

Numeric vector of the same length as `x`. The `i`th element of the resulting vector gives the result of calculating `N(x[i])`.

References

Klir G.J, Yuan B., *Fuzzy sets and fuzzy logic. Theory and applications*, Prentice Hall PTR, New Jersey, 1995.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other fuzzy_logic: `fimplication_minimal()`, `tconorm_minimum()`, `tnorm_minimum()`

`index_g` *Egghe's g-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \dots, x_n)$, where $x_i \ge x_j \ge 0$ for $i \le j$, the g*-index* (Egghe, 2006) for x is defined as

$$G(x) = \max\{i = 1, \dots, n : \sum_{j=1}^{i} x_i \ge i^2\}$$

if $n \ge 1$ and $x_1 \ge 1$, or $G(x) = 0$ otherwise.

Usage

```
index_g(x)

index.g(x) # same as index_g(x), deprecated alias

index_g_zi(x)
```

Arguments

x a non-negative numeric vector

Details

`index.g` is a (deprecated) alias for `index_g`.

Note that `index_g` is not a zero-insensitive impact function, see Examples section. `index_g_zi` is its zero-sensitive variant: it assumes that the aggregated vector is padded with zeros.

If a non-increasingly sorted vector is given, the function has O(n) run-time.

For historical reasons, this function is also available via an alias, `index.g` [but its usage is deprecated].

Value

a single numeric value

References

Egghe L., Theory and practise of the g-index, *Scientometrics* 69(1), 2006, pp. 131-152.

Mesiar R., Gagolewski M., H-index and other Sugeno integrals: Some defects and their compensation, IEEE Transactions on Fuzzy Systems 24(6), 2016, pp. 1668-1672. doi:10.1109/TFUZZ.2016.2516579

Gagolewski M., Mesiar R., Monotone measures and universal integrals in a uniform framework for the scientific impact assessment problem, Information Sciences 263, 2014, pp. 166-174. doi:10.1016/j.ins.2013.12.004

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other impact_functions: `index_h()`, `index_lp()`, `index_maxprod()`, `index_rp()`, `index_w()`, `pord_weakdom()`

Examples

```
sapply(list(c(9), c(9,0), c(9,0,0), c(9,0,0,0)), index_g) # not a zero-sensitive agop
```

---

index_h *Hirsch's h-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \dots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the h-*index* (Hirsch, 2005) for x is defined as

$$H(x) = \max\{i = 1, \dots, n : x_i \geq i\}$$

if $n \geq 1$ and $x_1 \geq 1$, or $H(x) = 0$ otherwise.

Usage

```
index_h(x)

index.h(x) # same as index_h(x), deprecated alias
```

Arguments

x	a non-negative numeric vector

Details

If a non-increasingly sorted vector is given, the function has O(n) run-time.

For historical reasons, this function is also available via an alias, `index.h` [but its usage is deprecated].

See `index_rp` and `owmax` for natural generalizations.

The h-index is the same as the discrete Sugeno integral of `x` w.r.t. the counting measure (see Torra, Narukawa, 2008).

Value

a single numeric value

References

Hirsch J.E., An index to quantify individual's scientific research output, *Proceedings of the National Academy of Sciences* 102(46), 2005, pp. 16569-16572.

Mesiar R., Gagolewski M., H-index and other Sugeno integrals: Some defects and their compensation, IEEE Transactions on Fuzzy Systems 24(6), 2016, pp. 1668-1672. doi:10.1109/TFUZZ.2016.2516579

Gagolewski M., Mesiar R., Monotone measures and universal integrals in a uniform framework for the scientific impact assessment problem, Information Sciences 263, 2014, pp. 166-174. doi:10.1016/j.ins.2013.12.004

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

Sugeno M., *Theory of fuzzy integrals and its applications*, PhD thesis, Tokyo Institute of Technology, 1974.

Torra V., Narukawa Y., The h-index and the number of citations: Two fuzzy integrals, *IEEE Transactions on Fuzzy Systems* 16(3), 2008, pp. 795-797.

See Also

Other impact_functions: `index_g()`, `index_lp()`, `index_maxprod()`, `index_rp()`, `index_w()`, `pord_weakdom()`

Examples

```
authors <- list(  # a list of numeric sequences
                  # (e.g. citation counts of the articles
                  # written by some authors)
    "A" =c(23,21,4,2,1,0,0),
    "B" =c(11,5,4,4,3,2,2,2,2,2,1,1,1,0,0,0,0),
    "C" =c(53,43,32,23,14,13,12,8,4,3,2,1,0)
 )
index_h(authors$A)
sapply(authors, index_h)
```

---

index_lp — *The l_p-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \dots, x_n)$, where $x_i \geq x_j$ for $i \leq j$, the l_p-*index* for $p = \infty$ equals to

$$l_p(x) = \arg\max_{(i, x_i), i=1,\dots,n} \{ix_i\}$$

if $n \geq 1$, or $l_\infty(x) = 0$ otherwise. Note that if $(i, x_i) = l_\infty(x)$, then

$$MAXPROD(x) = \mathtt{prod}(l_\infty(x)) = ix_i,$$

where $MAXPROD$ is the index proposed in (Kosmulski, 2007), see `index_maxprod`. Moreover, this index corresponds to the Shilkret integral of x w.r.t. some monotone measure, cf. (Gagolewski, Debski, Nowakiewicz, 2013).

For the definition of the l_p-index for $p < \infty$ we refer to (Gagolewski, Grzegorzewski, 2009a).

Usage

```
index_lp(x, p = Inf, projection = prod)

index.lp(x, p = Inf, projection = prod)  # deprecated alias
```

Arguments

`x`	a non-negative numeric vector
`p`	index order, $p \in [1, \infty]$; defaults ∞ (`Inf`).
`projection`	function

Details

The l_p-index, by definition, is not an impact function, as it produces 2 numeric values. Thus, it should be projected to one dimension. However, you may set the `projection` argument to `identity` so as to obtain the 2-dimensional index

If a non-increasingly sorted vector is given, the function has O(n) run-time for any p, see (Gagolewski, Debski, Nowakiewicz, 2013).

For historical reasons, this function is also available via an alias, `index.lp` [but its usage is deprecated].

Value

result of `projection(c(`i, x_i`))`

References

Gagolewski M., Grzegorzewski P., A geometric approach to the construction of scientific impact indices, *Scientometrics* 81(3), 2009a, pp. 617-634.

Gagolewski M., Debski M., Nowakiewicz M., *Efficient Algorithm for Computing Certain Graph-Based Monotone Integrals: the lp-Indices*, In: Mesiar R., Bacigal T. (Eds.), *Proc. Uncertainty Modelling*, STU Bratislava, ISBN:978-80-227-4067-8, 2013, pp. 17-23.

Kosmulski M., MAXPROD - A new index for assessment of the scientific output of an individual, and a comparison with the h-index, *Cybermetrics* 11(1), 2007.

Shilkret, N., Maxitive measure and integration, *Indag. Math.* 33, 1971, pp. 109-116.

See Also

Other impact_functions: `index_g()`, `index_h()`, `index_maxprod()`, `index_rp()`, `index_w()`, `pord_weakdom()`

Examples

```
x <- runif(100, 0, 100)
index.lp(x, Inf, identity)  # two-dimensional value, can not be used
                            # directly in the analysis
index.lp(x, Inf, prod)      # the MAXPROD-index (one-dimensional) [default]
```

`index_maxprod` *Kosmulski's MAXPROD-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \ldots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the *MAXPROD-index* (Kosmulski, 2007) for x is defined as

$$MAXPROD(x) = \max\{ix_i : i = 1, \ldots, n\}$$

Usage

```
index_maxprod(x)
```

Arguments

`x`	a non-negative numeric vector

Details

If a non-increasingly sorted vector is given, the function has O(n) run-time.

The MAXPROD index is the same as the discrete Shilkret integral of `x` w.r.t. the counting measure.

See `index_lp` for a natural generalization.

Value

a single numeric value

References

Kosmulski M., MAXPROD - A new index for assessment of the scientific output of an individual, and a comparison with the h-index, *Cybermetrics* 11(1), 2007.

Mesiar R., Gagolewski M., H-index and other Sugeno integrals: Some defects and their compensation, IEEE Transactions on Fuzzy Systems 24(6), 2016, pp. 1668-1672. doi:10.1109/TFUZZ.2016.2516579

Gagolewski M., Mesiar R., Monotone measures and universal integrals in a uniform framework for the scientific impact assessment problem, Information Sciences 263, 2014, pp. 166-174. doi:10.1016/j.ins.2013.12.004

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other impact_functions: `index_g()`, `index_h()`, `index_lp()`, `index_rp()`, `index_w()`, `pord_weakdom()`

`index_rp` *The r_p-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \dots, x_n)$, where $x_i \geq x_j$ for $i \leq j$, the r_p*-index* for $p = \infty$ equals to

$$r_p(x) = \max_{i=1,\dots,n} \{\min\{i, x_i\}\}$$

if $n \geq 1$, or $r_\infty(x) = 0$ otherwise. That is, it is equivalent to a particular OWMax operator, see `owmax`.

For the definition of the r_p-index for $p < \infty$ we refer to (Gagolewski, Grzegorzewski, 2009).

Usage

```
index_rp(x, p = Inf)

index.rp(x, p = Inf) # same as index_rp(x, p), deprecated alias
```

Arguments

x	a non-negative numeric vector
p	index order, $p \in [1, \infty]$; defaults ∞ (Inf).

Details

Note that if $x_1, \ldots, x_n$ are integers, then

$$r_\infty(x) = H(x),$$

where H is the h-index (Hirsch, 2005) and

$$r_1(x) = W(x),$$

where W is the w-index (Woeginger, 2008), see index_h and index_w.

If a non-increasingly sorted vector is given, the function has O(n) run-time.

For historical reasons, this function is also available via an alias, index.rp [but its usage is deprecated].

Value

a single numeric value

References

Gagolewski M., Grzegorzewski P., A geometric approach to the construction of scientific impact indices, *Scientometrics* 81(3), 2009, pp. 617-634.

Hirsch J.E., An index to quantify individual's scientific research output, *Proceedings of the National Academy of Sciences* 102(46), 2005, pp. 16569-16572.

Woeginger G.J., An axiomatic characterization of the Hirsch-index, *Mathematical Social Sciences* 56(2), 2008, pp. 224-232.

See Also

Other impact_functions: index_g(), index_h(), index_lp(), index_maxprod(), index_w(), pord_weakdom()

Examples

```
x <- runif(100, 0, 100);
index.rp(x);            # the r_oo-index
floor(index.rp(x));     # the h-index
index.rp(floor(x), 1);  # the w-index
```

`index_w` *Woeginger's w-index*

Description

Given a sequence of n non-negative numbers $x = (x_1, \dots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the w-*index* (Woeginger, 2008) for x is defined as

$$W(x) = \max\{i = 1, \dots, n : x_j \geq i - j + 1, \forall j = 1, \dots, i\}$$

Usage

```
index_w(x)
```

Arguments

x	a non-negative numeric vector

Details

If a non-increasingly sorted vector is given, the function has O(n) run-time.

See `index_rp` for a natural generalization.

Value

a single numeric value

References

Woeginger G. J., An axiomatic characterization of the Hirsch-index. *Mathematical Social Sciences* 56(2), 2008, pp. 224-232.

See Also

Other impact_functions: `index_g()`, `index_h()`, `index_lp()`, `index_maxprod()`, `index_rp()`, `pord_weakdom()`

owa *WAM and OWA Operators*

Description

Computes the Weighted Arithmetic Mean or the Ordered Weighted Averaging aggregation operator.

Usage

```
owa(x, w = rep(1/length(x), length(x)))

wam(x, w = rep(1/length(x), length(x)))
```

Arguments

x	numeric vector to be aggregated
w	numeric vector of the same length as x, with elements in $[0, 1]$, and such that $\sum_i w_i = 1$; weights

Details

The OWA operator is given by

$$\mathsf{OWA}_{\mathtt{w}}(\mathtt{x}) = \sum_{i=1}^{n} w_i\, x_{(i)}$$

where $x_{(i)}$ denotes the i-th smallest value in x.

The WAM operator is given by

$$\mathsf{WAM}_{\mathtt{w}}(\mathtt{x}) = \sum_{i=1}^{n} w_i\, x_i$$

If the elements in w do not sum up to 1, then they are normalized and a warning is generated.

Both functions by default return the ordinary arithmetic mean. Special cases of OWA include the trimmed mean (see mean) and Winsorized mean.

There is a strong, well-known connection between the OWA operators and the Choquet integrals.

Value

These functions return a single numeric value.

References

Choquet G., Theory of capacities, *Annales de l'institut Fourier* 5, 1954, pp. 131-295.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

Yager R.R., On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics* 18(1), 1988, pp. 183-190.

See Also

Other aggregation_operators: `owmax()`

`owmax` *WMax, WMin, OWMax, and OWMin Operators*

Description

Computes the (Ordered) Weighted Maximum/Minimum.

Usage

```
owmax(x, w = rep(Inf, length(x)))

owmin(x, w = rep(-Inf, length(x)))

wmax(x, w = rep(Inf, length(x)))

wmin(x, w = rep(-Inf, length(x)))
```

Arguments

`x`	numeric vector to be aggregated
`w`	numeric vector of the same length as `x`; weights

Details

The OWMax operator is given by

$$\mathsf{OWMax}_{\mathtt{w}}(\mathtt{x}) = \bigvee_{i=1}^{n} w_i \wedge x_{(i)}$$

where $x_{(i)}$ denotes the i-th smallest value in `x`.
The OWMin operator is given by

$$\mathsf{OWMin}_{\mathtt{w}}(\mathtt{x}) = \bigwedge_{i=1}^{n} w_i \vee x_{(i)}$$

The WMax operator is given by

$$\mathsf{WMax}_{\mathtt{w}}(\mathtt{x}) = \bigvee_{i=1}^{n} w_i \wedge x_i$$

The WMin operator is given by

$$\mathsf{WMin}_{\mathtt{w}}(\mathtt{x}) = \bigwedge_{i=1}^{n} w_i \vee x_i$$

OWMax and WMax by default return the greatest value in x and OWMin and WMin - the smallest value in x.

Classically, it is assumed that if we aggregate vectors with elements in $[a, b]$, then the largest weight for OWMax should be equal to b and the smallest for OWMin should be equal to a.

There is a strong connection between the OWMax/OWMin operators and the Sugeno integrals w.r.t. some monotone measures. Additionally, it may be shown that the OWMax and OWMin classes are equivalent.

Moreover, index_h for integer data is a particular OWMax operator.

Value

These functions return a single numeric value.

References

Dubois D., Prade H., Testemale C., Weighted fuzzy pattern matching, *Fuzzy Sets and Systems* 28, 1988, pp. 313-331.

Dubois D., Prade H., Semantics of quotient operators in fuzzy relational databases, *Fuzzy Sets and Systems* 78(1), 1996, pp. 89-93.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

Sugeno M., *Theory of fuzzy integrals and its applications*, PhD thesis, Tokyo Institute of Technology, 1974.

See Also

Other aggregation_operators: owa()

---

pareto2_estimate_mle *Parameter Estimation in the Pareto Type-II Distribution Family (MLE)*

Description

Finds the maximum likelihood estimator of the Pareto Type-II distribution's shape parameter k and, if not given explicitly, scale parameter s.

Usage

```
pareto2_estimate_mle(
  x,
  s = NA_real_,
  smin = 1e-04,
  smax = 20,
  tol = .Machine$double.eps^0.25
)
```

Arguments

x	a non-negative numeric vector
s	a-priori known scale parameter, $s > 0$ or NA if unknown (default)
smin	lower bound for the scale parameter
smax	upper bound for the scale parameter
tol	the desired accuracy (convergence tolerance)

Details

Note that if s is not given, then the maximum of the likelihood function might not exist for some input vectors. This estimator may have a large mean squared error. Consider using pareto2_estimate_mmse.

For known s, the estimator is unbiased.

Value

Returns a numeric vector with the following named components:

- k - estimated parameter of shape
- s - estimated (or known, see the s argument) parameter of scale

or c(NA, NA) if the maximum of the likelihood function could not be found.

See Also

Other Pareto2: pareto2_estimate_mmse(), pareto2_test_ad(), pareto2_test_f(), rpareto2()

---

pareto2_estimate_mmse *Parameter Estimation in the Pareto Type-II Distribution Family (MMSE)*

Description

Finds the MMS estimator of the Pareto Type-II distribution parameters using the Bayesian method (and the R code) developed by Zhang and Stevens (2009).

Usage

```
pareto2_estimate_mmse(x)
```

Arguments

x	a non-negative numeric vector

Value

Returns a numeric vector with the following named components:

- k - estimated parameter of shape,
- s - estimated parameter of scale.

References

Zhang J., Stevens M.A., A New and Efficient Estimation Method for the Generalized Pareto Distribution, *Technometrics* 51(3), 2009, pp. 316-325.

See Also

Other Pareto2: pareto2_estimate_mle(), pareto2_test_ad(), pareto2_test_f(), rpareto2()

---

pareto2_test_ad — *Anderson-Darling Test for the Pareto Type-II Distribution*

Description

Performs an approximate Anderson-Darling goodness-of-fit test, which verifies the null hypothesis: Data follow a Pareto-Type II distribution.

Usage

```
pareto2_test_ad(x, s = 1)
```

Arguments

x	a non-negative numeric vector of data values
s	the known scale parameter, $s > 0$

Details

We know that if X follows a Pareto-Type II distribution with shape parameter k, then $log(1 + X/s)$ follows an exponential distribution with parameter k. Thus, this function transforms the input vector, and performs the same steps as exp_test_ad.

Value

A list of the class htest is returned, see exp_test_ad.

See Also

Other Pareto2: pareto2_estimate_mle(), pareto2_estimate_mmse(), pareto2_test_f(), rpareto2()

Other Tests: exp_test_ad(), pareto2_test_f()

`pareto2_test_f` *Two-Sample F-test For Equality of Shape Parameters for Type II-Pareto Distributions*

Description

Performs the F-test for the equality of shape parameters of two samples from Pareto type-II distributions with known and equal scale parameters, $s > 0$.

Usage

```
pareto2_test_f(
  x,
  y,
  s,
  alternative = c("two.sided", "less", "greater"),
  significance = NULL
)
```

Arguments

`x`	a non-negative numeric vector
`y`	a non-negative numeric vector
`s`	the known scale parameter, $s > 0$
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"` (default), `"less"`, or `"greater"`
`significance`	significance level, $0 <$`significance`< 1 or `NULL`. See the Value section for details

Details

Given two samples $(X_1, ..., X_n)$ i.i.d. $P2(k_x, s)$ and $(Y_1, ..., Y_m)$ i.i.d. $P2(k_y, s)$ this test verifies the null hypothesis $H_0 : k_x = k_y$ against two-sided or one-sided alternatives, depending on the value of `alternative`. It is based on the test statistic $T(X, Y) = \frac{n \sum_{i=1}^{m} \log(1+Y_i/m)}{m \sum_{i=1}^{n} \log(1+X_i/n)}$ which, under H_0, follows the Snedecor's F distribution with $(2m, 2n)$ degrees of freedom.

Note that for $k_x < k_y$, then X dominates Y stochastically.

Value

If `significance` is not `NULL`, then the list of class `power.htest` with the following components is yield in result:

- `statistic` - the value of the test statistic.
- `result` - either FALSE (accept null hypothesis) or TRUE (reject).
- `alternative` - a character string describing the alternative hypothesis.

- `method` - a character string indicating what type of test was performed.
- `data.name` - a character string giving the name(s) of the data.

Otherwise, the list of class `htest` with the following components is yield in result:

- `statistic` the value of the test statistic.
- `p.value` the p-value of the test.
- `alternative` a character string describing the alternative hypothesis.
- `method` a character string indicating what type of test was performed.
- `data.name` a character string giving the name(s) of the data.

See Also

Other Pareto2: `pareto2_estimate_mle()`, `pareto2_estimate_mmse()`, `pareto2_test_ad()`, `rpareto2()`

Other Tests: `exp_test_ad()`, `pareto2_test_ad()`

`plot_producer` *Draws a Graphical Representation of a Numeric Vector*

Description

Draws a step function that represents a numeric vector with elements in $[a, \infty]$.

Usage

```
plot_producer(
  x,
  type = c("left.continuous", "right.continuous", "curve"),
  extend = FALSE,
  add = FALSE,
  pch = 1,
  col = 1,
  lty = 1,
  lwd = 1,
  cex = 1,
  col.steps = col,
  lty.steps = 2,
  lwd.steps = 1,
  xlab = "",
  ylab = "",
  main = "",
  xmarg = 10,
  xlim = c(0, length(x) * 1.2),
  ylim = c(a, max(x)),
  a = 0,
  ...
)
```

Arguments

- `x` — non-negative numeric vector
- `type` — character; `'left.continuous'` (the default) or `'right.continuous'` for step functions and `'curve'` for a continuous step curve
- `extend` — logical; should the plot be extended infinitely to the right? Defaults to `FALSE`
- `add` — logical; indicates whether to start a new plot, `FALSE` by default
- `pch, col, lty, lwd, cex, xmarg` — graphical parameters
- `col.steps, lty.steps, lwd.steps` — graphical parameters, used only for `type` of `'left.continuous'` and `'right.continuous'` only
- `ylim, xlim, xlab, ylab, main, ...` — additional graphical parameters, see `plot.default`
- `a` — single numeric value

Details

In **agop**, a vector $x = (x_1, \ldots, x_n)$ can be represented by a step function defined for $0 \leq y < n$ and given by:

$$\pi(y) = x_{(n-\lfloor y+1 \rfloor+1)}$$

(for `type == 'right.continuous'`) or for $0 < y \leq n$

$$\pi(y) = x_{(n-\lfloor y \rfloor+1)}$$

(for `type == 'left.continuous'`, the default) or by a curve interpolating the points $(0, x_{(n)})$, $(1, x_{(n)})$, $(1, x_{(n-1)})$, $(2, x_{(n-1)})$, ..., $(n, x_{(1)})$. Here, $x_{(i)}$ denotes the i-th smallest value in x.

In bibliometrics, a step function of one of the two above-presented types is called a citation function.

For historical reasons, this function is also available via its alias, `plot.citfun` [but its usage is deprecated].

Value

nothing interesting

Examples

```
john_s <- c(11,5,4,4,3,2,2,2,2,2,1,1,1,0,0,0,0)
plot_producer(john_s, main="Smith, John", col="red")
```

pord_nd — *Weak Dominance Relation (Preorder)*

Description

Checks whether a numeric vector of arbitrary length is (weakly) dominated (elementwise) by another vector of the same length.

Usage

```
pord_nd(x, y, incompatible_lengths = NA)
```

Arguments

`x`	numeric vector with nonnegative elements
`y`	numeric vector with nonnegative elements
`incompatible_lengths`	single logical value, value to return iff lengths of `x` and `y` differ

Details

We say that a numeric vector **x** of length n_x is *weakly dominated* by **y** of length n_y iff

1. $n_x = n_y$ and
2. for all $i = 1, \ldots, n_x$ it holds $x_i \leq y_i$.

This relation is a preorder: it is reflexive (see `rel_is_reflexive`) and transitive (see `rel_is_transitive`), but not necessarily total (see `rel_is_total`). See `rel_graph` for a convenient function to calculate the relationship between all pairs of elements of a given set.

Such a preorder is tightly related to classical aggregation functions: each aggregation function is a morphism between weak-dominance-preordered set of vectors and the set of reals equipped with standard linear ordering.

Value

Returns a single logical value indicating whether `x` is weakly dominated by `y`.

References

Grabisch M., Marichal J.-L., Mesiar R., Pap E., *Aggregation functions*, Cambridge University Press, 2009.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other binary_relations: check_comonotonicity(), pord_spread(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive(), rel_reduction_hasse()

---

pord_spread — *Compare Spread of Vectors (Preorder)*

Description

This function determines whether one numeric vector has not greater spread than the other

Usage

```
pord_spread(x, y, incompatible_lengths = NA)
```

Arguments

x	numeric vector
y	numeric vector of the same length as x
incompatible_lengths	single logical value, value to return iff lengths of x and y differ

Details

We say that **x** of size n is of *no greater spread* than **y** iff for all $i, j = 1, \ldots, n$ such that $x_i > x_j$ it holds $x_i - x_j \leq y_i - y_j ll$. Such a preorder is used in the definition of a spread measure (see Gagolewski, 2015).

Note that the class of dispersion functions includes e.g. the sample variance (see var), standard veriation (see sd), range (see range and then diff), interquartile range (see IQR), median absolute deviation (see mad).

Value

The function returns a single logical value, which states whether x has no greater spread than y

References

Gagolewski M., Spread measures and their relation to aggregation functions, *European Journal of Operational Research* 241(2), 2015, pp. 469–477.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive(), rel_reduction_hasse()

`pord_weakdom` *Weak Dominance Relation (Preorder) in the Producer Assessment Problem*

Description

Checks whether a given numeric vector of arbitrary length is (weakly) dominated by another vector, possibly of different length, in terms of (sorted) elements' values and their number.

Usage

```
pord_weakdom(x, y)
```

Arguments

`x`	numeric vector with nonnegative elements
`y`	numeric vector with nonnegative elements

Details

We say that a numeric vector **x** of length n_x is *weakly dominated* by **y** of length n_y iff

1. $n_x \leq n_y$ and
2. for all $i = 1, \ldots, n$ it holds $x_{(n_x-i+1)} \leq y_{(n_y-i+1)}$.

This relation is a preorder: it is reflexive (see `rel_is_reflexive`) and transitive (see `rel_is_transitive`), but not necessarily total (see `rel_is_total`). See `rel_graph` for a convenient function to calculate the relationship between all pairs of elements of a given set.

Note that this dominance relation gives the same value for all permutations of input vectors' element. Such a preorder is tightly related to symmetric impact functions: each impact function is a morphism between weak-dominance-preordered set of vectors and the set of reals equipped with standard linear ordering (see Gagolewski, Grzegorzewski, 2011 and Gagolewski, 2013).

Value

Returns a single logical value indicating whether `x` is weakly dominated by `y`.

References

Gagolewski M., Grzegorzewski P., Possibilistic Analysis of Arity-Monotonic Aggregation Operators and Its Relation to Bibliometric Impact Assessment of Individuals, *International Journal of Approximate Reasoning* 52(9), 2011, pp. 1312-1324.

Gagolewski M., Scientific Impact Assessment Cannot be Fair, *Journal of Informetrics* 7(4), 2013, pp. 792-802.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_spread(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive(), rel_reduction_hasse()

Other impact_functions: index_g(), index_h(), index_lp(), index_maxprod(), index_rp(), index_w()

---

rdpareto2 *Discretized Pareto Type-II (Lomax) Distribution [TO DO]*

Description

Probability mass function, cumulative distribution function, quantile function, and random generation for the Discretized Pareto Type-II distribution with shape parameter $k > 0$ and scale parameter $s > 0$.

[TO DO: rewrite in C, add NA handling, add working qdpareto2()]

Usage

```
rdpareto2(n, k = 1, s = 1)

pdpareto2(q, k = 1, s = 1, lower.tail = TRUE)

qdpareto2(p, k = 1, s = 1, lower.tail = TRUE)

ddpareto2(x, k = 1, s = 1)
```

Arguments

n	integer; number of observations
k	vector of shape parameters, $k > 0$
s	vector of scale parameters, $s > 0$
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$, and $P(X > x)$ otherwise
p	vector of probabilities
x, q	vector of quantiles

Details

If $X \sim \mathrm{DP2}(k, s)$, then $\lfloor Y \rfloor = X$, where Y has ordinary Pareto Type-II distribution, see ppareto2.

Value

numeric vector; ddpareto2 gives the probability mass function, pdpareto2 gives the cumulative distribution function, qdpareto2 calculates the quantile function, and rdpareto2 generates random deviates.

See Also

Other distributions: `rpareto2()`

Other DiscretizedPareto2: `dpareto2_estimate_mle()`

`rel_graph` *Create an Adjacency Matrix Representing a Binary Relation*

Description

Returns a binary relation that represents results of comparisons with `pord` of all pairs of elements in `x`. We have `ret[i,j] == pord(x[[i]], x[[j]], ...)`.

Usage

```
rel_graph(x, pord, ...)
```

Arguments

`x`	list with elements to compare, preferably named
`pord`	a function with two arguments, returning a single Boolean value, e.g., `pord_spread`, `pord_nd`, or `pord_weakdom`
`...`	additional arguments passed to `pord`

Value

Returns a square logical matrix. `dimnames` of the matrix correspond to `names` of `x`.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_reflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

`rel_is_antisymmetric` *Antisymmetric Binary Relations*

Description

A binary relation R is *antisymmetric*, iff for all x, y we have xRy and $yRx \Rightarrow x = y$.

Usage

```
rel_is_antisymmetric(R)
```

Arguments

`R` an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

`rel_is_antisymmetric` finds out if a given binary relation is antisymmetric. Missing values in `R` may result in `NA`.

Also, check out `rel_closure_symmetric` for the symmetric closure of `R`.

Value

`rel_is_antisymmetric` returns a single logical value.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_reflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

`rel_is_asymmetric` *Asymetric Binary Relations*

Description

A binary relation R is *asymmetric*, iff for all x, y we have $xRy \Rightarrow \neg yRx$.

Usage

```
rel_is_asymmetric(R)
```

Arguments

`R` an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

Note that an asymmetric relation is necessarily irreflexive, compare `rel_is_irreflexive`.

`rel_is_asymmetric` finds out if a given binary relation is asymmetric. Missing values in `R` may result in `NA`.

Also, check out `rel_closure_symmetric` for the symmetric closure of `R`.

Value

`rel_is_asymmetric` returns a single logical value.

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_spread(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive(), rel_reduction_hasse()

---

rel_is_cyclic *Cyclic Binary Relations*

Description

A binary relation R is *cyclic*, iff its transitive closure is not antisymmetric. Note that R may be reflexive and still acyclic, i.e., loops in R are not taken into account.

Usage

```
rel_is_cyclic(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

rel_is_cyclic has $O(n^3)$ time complexity, where n is the number of rows in R (the implemented algorithm currently verifies whether a depth-first search-based topological sorting is possible). Missing values in R always result in NA.

Value

rel_is_cyclic returns a single logical value.

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_spread(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive(), rel_reduction_hasse()

rel_is_irreflexive *Irreflexive Binary Relations*

Description

A binary relation R is *irreflexive* (or antireflexive), iff for all x we have $\neg xRx$.

Usage

```
rel_is_irreflexive(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

`rel_is_irreflexive` finds out if a given binary relation is irreflexive. The function just checks whether all elements on the diagonal of `R` are zeros, i.e., it has $O(n)$ time complexity, where n is the number of rows in `R`. Missing values on the diagonal may result in `NA`.

When dealing with a graph's loops, i.e., elements related to themselves, you may be interested in finding a reflexive closure, see `rel_closure_reflexive`, or a reflexive reduction, see `rel_reduction_reflexive`.

Value

`rel_is_irreflexive` returns a single logical value.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_reflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

rel_is_reflexive *Reflexive Binary Relations*

Description

A binary relation R is reflexive, iff for all x we have xRx.

Usage

```
rel_is_reflexive(R)

rel_closure_reflexive(R)

rel_reduction_reflexive(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

`rel_is_reflexive` finds out if a given binary relation is reflexive. The function just checks whether all elements on the diagonal of `R` are non-zeros, i.e., it has $O(n)$ time complexity, where n is the number of rows in `R`. Missing values on the diagonal may result in `NA`.

A reflexive closure of a binary relation R, determined by `rel_closure_reflexive`, is the minimal reflexive superset R' of R.

A reflexive reduction of a binary relation R, determined by `rel_reduction_reflexive`, is the minimal subset R' of R, such that the reflexive closures of R and R' are equal i.e., the largest irreflexive relation contained in R.

Value

The `rel_closure_reflexive` and `rel_reduction_reflexive` functions return a logical square matrix. `dimnames` of `R` are preserved.

On the other hand, `rel_is_reflexive` returns a single logical value.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

`rel_is_symmetric` *Symmetric Binary Relations*

Description

A binary relation R is *symmetric*, iff for all x, y we have $xRy \Rightarrow yRx$.

Usage

```
rel_is_symmetric(R)

rel_closure_symmetric(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

`rel_is_symmetric` finds out if a given binary relation is symmetric. Any missing value behind the diagonal results in `NA`.

The *symmetric closure* of a binary relation R, determined by `rel_closure_symmetric`, is the smallest symmetric binary relation that contains R. Here, any missing values in `R` result in an error.

Value

The `rel_closure_symmetric` function returns a logical square matrix. `dimnames` of `R` are preserved.

On the other hand, `rel_is_symmetric` returns a single logical value.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_reflexive()`, `rel_is_total()`, `rel_is_transitive()`, `rel_reduction_hasse()`

`rel_is_total` *Total Binary Relations*

Description

A binary relation R is *total* (or *strong complete*), iff for all x, y we have xRy or yRx.

Usage

```
rel_is_total(R)

rel_closure_total_fair(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

Note that each total relation is also reflexive, see rel_is_reflexive.

rel_is_total determines if a given binary relation R is total. The algorithm has $O(n^2)$ time complexity, where n is the number of rows in R. If R[i,j] and R[j,i] is NA for some (i, j), then the functions outputs NA.

The problem of finding a total closure or reduction is not well-defined in general.

When dealing with preorders, however, the following closure may be useful, see (Gagolewski, 2013). *Fair totalization* of R, performed by rel_closure_total_fair, is the minimal superset R' of R such that if not xRy and not yRx then $xR'y$ and $yR'x$.

Even if R is transitive, the resulting relation might not necessarily fulfil this property. If you want a total preorder, call rel_closure_transitive afterwards. Missing values in R are not allowed and result in an error.

Value

rel_is_total returns a single logical value.

rel_closure_reflexive returns a logical square matrix. dimnames of R are preserved.

References

Gagolewski M., Scientific Impact Assessment Cannot be Fair, *Journal of Informetrics* 7(4), 2013, pp. 792-802.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_spread(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_transitive(), rel_reduction_hasse()

---

rel_is_transitive *Transitive Binary Relations*

Description

A binary relation R is *transitive*, iff for all x, y, z we have xRy and $yRz \Longrightarrow xRz$.

Usage

```
rel_is_transitive(R)

rel_closure_transitive(R)

rel_reduction_transitive(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

`rel_is_transitive` finds out if a given binary relation is transitive. The algorithm has $O(n^3)$ time complexity, pessimistically, where n is the number of rows in `R`. If `R` contains missing values behind the diagonal, the result will be `NA`.

The *transitive closure* of a binary relation R, determined by `rel_closure_transitive`, is the minimal superset of R such that it is transitive. Here we use the well-known Warshall algorithm (1962), which runs in $O(n^3)$ time.

The *transitive reduction*, see (Aho et al. 1972), of an acyclic binary relation R, determined by `rel_reduction_transitive`, is a minimal unique subset R' of R, such that the transitive closures of R and R' are equal. The implemented algorithm runs in $O(n^3)$ time. Note that a transitive reduction of a reflexive relation is also reflexive. Moreover, some kind of transitive reduction (not necessarily minimal) is also determined in `rel_reduction_hasse` – it is useful for drawing Hasse diagrams.

Value

The `rel_closure_transitive` and `rel_reduction_transitive` functions return a logical square matrix. `dimnames` of `R` are preserved.

On the other hand, `rel_is_transitive` returns a single logical value.

References

Aho A.V., Garey M.R., Ullman J.D., The Transitive Reduction of a Directed Graph, *SIAM Journal on Computing* 1(2), 1972, pp. 131-137.

Warshall S., A theorem on Boolean matrices, *Journal of the ACM* 9(1), 1962, pp. 11-12.

See Also

Other binary_relations: `check_comonotonicity()`, `pord_nd()`, `pord_spread()`, `pord_weakdom()`, `rel_graph()`, `rel_is_antisymmetric()`, `rel_is_asymmetric()`, `rel_is_cyclic()`, `rel_is_irreflexive()`, `rel_is_reflexive()`, `rel_is_symmetric()`, `rel_is_total()`, `rel_reduction_hasse()`

`rel_reduction_hasse` *Hasse Diagrams*

Description

This function computes the reflexive reduction and a kind of transitive reduction which is useful for drawing Hasse diagrams.

Usage

```
rel_reduction_hasse(R)
```

Arguments

R	an object coercible to a 0-1 (logical) square matrix, representing a binary relation on a finite set.

Details

The input matrix R might not necessarily be acyclic/asymmetric, i.e., it may represent any totally preordered set (which induces an equivalence relation on the underlying preordered set). The implemented algorithm runs in $O(n^3)$ time and first determines the transitive closure of R. If an irreflexive R is given, then the transitive closures of R and of the resulting matrix are identical. Moreover, if R is additionally acyclic, then this function is equivalent to rel_reduction_transitive.

Value

The rel_reduction_hasse function returns a logical square matrix. dimnames of R are preserved.

See Also

Other binary_relations: check_comonotonicity(), pord_nd(), pord_spread(), pord_weakdom(), rel_graph(), rel_is_antisymmetric(), rel_is_asymmetric(), rel_is_cyclic(), rel_is_irreflexive(), rel_is_reflexive(), rel_is_symmetric(), rel_is_total(), rel_is_transitive()

Examples

```
## Not run:
# Let ord be a total preorder (a total and transitive binary relation)
# === Plot the Hasse diagram of ord ===
# ===  requires the igraph package  ===
library("igraph")
hasse <- graph.adjacency(rel_reduction_transitive(ord))
plot(hasse, layout=layout.fruchterman.reingold(hasse, dim=2))

## End(Not run)
```

---

rpareto2 *Pareto Type-II (Lomax) Distribution*

Description

Density, cumulative distribution function, quantile function, and random generation for the Pareto Type-II (Lomax) distribution with shape parameter $k > 0$ and scale parameter $s > 0$.

[TO DO: rewrite in C, add NA handling]

Usage

```
rpareto2(n, k = 1, s = 1)

ppareto2(q, k = 1, s = 1, lower.tail = TRUE)

qpareto2(p, k = 1, s = 1, lower.tail = TRUE)

dpareto2(x, k = 1, s = 1)
```

Arguments

n	integer; number of observations
k	vector of shape parameters, $k > 0$
s	vector of scale parameters, $s > 0$
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$, and $P(X > x)$ otherwise
p	vector of probabilities
x, q	vector of quantiles

Details

If $X \sim \mathrm{P2}(k, s)$, then $\mathrm{supp}\, X = [0, \infty)$. The c.d.f. for $x \geq 0$ is given by

$$F(x) = 1 - s^k/(s + x)^k$$

and the density by

$$f(x) = ks^k/(s + x)^{k+1}.$$

Value

numeric vector; dpareto2 gives the density, ppareto2 gives the cumulative distribution function, qpareto2 calculates the quantile function, and rpareto2 generates random deviates.

See Also

Other distributions: rdpareto2()

Other Pareto2: pareto2_estimate_mle(), pareto2_estimate_mmse(), pareto2_test_ad(), pareto2_test_f()

---

tconorm_minimum *t-conorms*

Description

Various t-conorms. Each of these is a fuzzy logic generalization of the classical alternative operation.

Usage

```
tconorm_minimum(x, y)

tconorm_product(x, y)

tconorm_lukasiewicz(x, y)

tconorm_drastic(x, y)

tconorm_fodor(x, y)
```

Arguments

x	numeric vector with elements in $[0, 1]$
y	numeric vector of the same length as x, with elements in $[0, 1]$

Details

A function $S : [0, 1] \times [0, 1] \to [0, 1]$ is a *t-conorm* if for all $x, y, z \in [0, 1]$ it holds: (a) $S(x, y) = S(y, x)$; (b) if $y \leq z$, then $S(x, y) \leq S(x, z)$; (c) $S(x, S(y, z)) = S(S(x, y), z)$; (d) $S(x, 0) = x$.

The minimum t-conorm is given by $S_M(x, y) = max(x, y)$.

The product t-conorm is given by $S_P(x, y) = x + y - xy$.

The Lukasiewicz t-conorm is given by $S_L(x, y) = min(x + y, 1)$.

The drastic t-conorm is given by $S_D(x, y) = 1$ iff $x, y \in (0, 1]$, and $max(x, y)$ otherwise.

The Fodor t-conorm is given by $S_F(x, y) = 1$ iff $x + y \geq 1$, and $max(x, y)$ otherwise.

Value

Numeric vector of the same length as x and y. The ith element of the resulting vector gives the result of calculating S(x[i], y[i]).

References

Klir G.J, Yuan B., *Fuzzy sets and fuzzy logic. Theory and applications*, Prentice Hall PTR, New Jersey, 1995.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other fuzzy_logic: fimplication_minimal(), fnegation_yager(), tnorm_minimum()

tnorm_minimum *t-norms*

Description

Various t-norms. Each of these is a fuzzy logic generalization of the classical conjunction operation.

Usage

```
tnorm_minimum(x, y)

tnorm_product(x, y)

tnorm_lukasiewicz(x, y)

tnorm_drastic(x, y)

tnorm_fodor(x, y)
```

Arguments

x	numeric vector with elements in $[0, 1]$
y	numeric vector of the same length as x, with elements in $[0, 1]$

Details

A function $T : [0, 1] \times [0, 1] \to [0, 1]$ is a *t-norm* if for all $x, y, z \in [0, 1]$ it holds: (a) $T(x, y) = T(y, x)$; (b) if $y \leq z$, then $T(x, y) \leq T(x, z)$; (c) $T(x, T(y, z)) = T(T(x, y), z)$; (d) $T(x, 1) = x$.

The minimum t-norm is given by $T_M(x, y) = min(x, y)$.

The product t-norm is given by $T_P(x, y) = xy$.

The Lukasiewicz t-norm is given by $T_L(x, y) = max(x + y - 1, 0)$.

The drastic t-norm is given by $T_D(x, y) = 0$ iff $x, y \in [0, 1)$, and $min(x, y)$ otherwise.

The Fodor t-norm is given by $T_F(x, y) = 0$ iff $x + y \leq 1$, and $min(x, y)$ otherwise.

Value

Numeric vector of the same length as x and y. The ith element of the resulting vector gives the result of calculating T(x[i], y[i]).

References

Klir G.J, Yuan B., *Fuzzy sets and fuzzy logic. Theory and applications*, Prentice Hall PTR, New Jersey, 1995.

Gagolewski M., Data Fusion: Theory, Methods, and Applications, Institute of Computer Science, Polish Academy of Sciences, 2015, 290 pp. isbn:978-83-63159-20-7

See Also

Other fuzzy_logic: fimplication_minimal(), fnegation_yager(), tconorm_minimum()

Index