Package ‘agop’

August 29, 2013

Version 0.1-3

Date 2013-06-27

Title Aggregation Operators and Preordered Sets

Description Tools supporting multi-criteria decision making, including variable number of criteria, by means of aggregation operators and preordered sets. Possible applications include, but are not limited to, scientometrics and bibliometrics.

URL http://www.rexamine.com/resources/agop/

BugReports http://github.com/Rexamine/agop/issues

ByteCompile TRUE

Type Package

Depends R (>= 2.12.0), base, stats, grDevices, graphics, Matrix,igraph

License LGPL (>= 3)

Suggests testthat

Collate ’agop-package.R’ ’visualization.R’ ’preorders.R’’agops-impact.R’ ’agops-classical.R’ ’distrib-pareto2.R’’distrib-pareto2-estimators.R’ ’distrib-pareto2-ftest.R’

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

Maintainer Marek Gagolewski <gagolews@rexamine.com>

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-06-27 16:54:17

R topics documented:

`agop-package` *Aggregation Operators Package for R*

Description

"The process of combining several numerical values into a single representative one is called aggregation, and the numerical function performing this process is called aggregation function. This simple definition demonstrates the size of the field of application of aggregation: applied mathematics (e.g. probability, statistics, decision theory), computer science (e.g. artificial intelligence, operation research), as well as many applied fields (economics and finance, pattern recognition and image processing, data fusion, multicriteria decision making, automated reasoning etc.). Although history of aggregation is probably as old as mathematics (think of the arithmetic mean), its existence has reminded underground till only recent (...)." (Grabisch et al, 2009, p. xiii)

Details

agop is an open source (LGPL 3) package for R, to which anyone can contribute. It started as a fork of the **CITAN** package (Gagolewski, 2011).

For more infrmation refer to the Package Vignette. Its most recent version is available at http://github.com/Rexamine/agop/raw/master/inst/doc/agop-Tutorial.pdf.

Author(s)

Marek Gagolewski <gagolews@rexamine.com> [aut,cre],
Anna Cena <cena@rexamine.com> [ctb]

Keywords: aggregation, bibliometrics, scientometrics, scientific impact, webometrics, preorders, means, OWA, OWMax, OWMin, Hirsch's h-index, Egghe's g-index.

Acknowledgments: The development of the package in March-June 2013 was partially supported by the European Union from 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.

References

Beliakov G., Pradera A., Calvo T., Aggregation Functions: A Guide for Practitioners, Springer-Verlag, 2007.

Cena A., Gagolewski M., OM3: ordered maxitive, minitive, and modular aggregation operators - Part I: Axiomatic analysis under arity-dependence, In: Bustince H. et al (Eds.), Aggregation Functions in Theory and in Practise (AISC 228), Springer-Verlag, Heidelberg, 2013, pp. 93-103.

Cena A., Gagolewski M., OM3: ordered maxitive, minitive, and modular aggregation operators - Part II: A simulation study, In: Bustince H. et al (Eds.), Aggregation Functions in Theory and in Practise (AISC 228), Springer-Verlag, Heidelberg, 2013, pp. 105-115.

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

Gagolewski M., On the Relationship Between Symmetric Maxitive, Minitive, and Modular Aggregation Operators, Information Sciences 221, 2013, pp. 170-180.

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., Mesiar R., Aggregating Different Paper Quality Measures with a Generalized h-index, Journal of Informetrics 6(4), 2012, pp. 566-579.

Gagolewski M., Bibliometric Impact Assessment with R and the CITAN Package, Journal of Informetrics 5(4), 2011, pp. 678-692.

Gagolewski M., Grzegorzewski P., A Geometric Approach to the Construction of Scientific Impact Indices, Scientometrics 81(3), 2009, pp. 617-634.

Gagolewski M., Statistical Hypothesis Test for the Difference between Hirsch Indices of Two Pareto-Distributed Random Samples, In: Kruse R. et al (Eds.), Synergies of Soft Computing and Statistics for Intelligent Data Analysis (AISC 190), Springer-Verlag, Heidelberg, 2013, pp. 359-367.

Gagolewski M., On the Relation Between Effort-Dominating and Symmetric Minitive Aggregation Operators, In: Greco S. et al (Eds.), Advances in Computational Intelligence, Part III (CCIS 299), Springer-Verlag, Heidelberg, 2012, pp. 276-285.

Gagolewski M., Grzegorzewski P., Axiomatic Characterizations of (quasi-) L-statistics and S-statistics and the Producer Assessment Problem, In: Galichet S., Montero J., Mauris G. (Eds.), Proc. EUSFLAT/LFA 2011, Atlantic Press, 2011, pp. 53-58.
Gagolewski M., Grzegorzewski P., S-Statistics and Their Basic Properties, In: Borgelt C. et al (Eds.), Combining Soft Computing and Statistical Methods in Data Analysis (AISC 77), Springer-Verlag, Heidelberg, 2010, pp. 281-288.
Gagolewski M., Grzegorzewski P., Arity-Monotonic Extended Aggregation Operators, In: Hullermeier E., Kruse R., Hoffmann F. (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems (CCIS 80), Springer-Verlag, Heidelberg, 2010, pp. 693-702.
Grabisch M., Marichal J.-L., Mesiar R., Pap E., Aggregation functions, Cambridge University Press, 2009.
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.
Shilkret, N., Maxitive measure and integration, Indag. Math. 33, 1971, pp. 109-116.
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.

`closure_total_fair` *Total Closure of Adjacency Matrix [Fair Totalization]*

Description

Fair totalization: for each pair (x,y) s.t. not xRy and not xRy let from now on xRy and yRx

Usage

```
closure_total_fair(B)
```

Arguments

`B`	object of class `igraph` or a square 0-1 matrix of class `Matrix` or `matrix`

Details

If you want a total preorder, call `closure_transitive`.

Value

object of class `Matrix`

See Also

Other binary_relations: `closure_transitive`, `de_transitive`, `get_equivalence_classes`, `get_incomparable_pairs`, `get_independent_sets`, `is_reflexive`, `is_total`, `is_transitive`, `pord_weakdom`, `rel_graph` closure_transitive *Transitive Closure of Adjacency Matrix*

Description

This may be slow for large graphs.

Usage

```
closure_transitive(B)
```

Arguments

B	object of class igraph or a square 0-1 matrix of class Matrix or matrix

Value

object of class Matrix

See Also

Other binary_relations: closure_total_fair, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_total, is_transitive, pord_weakdom, rel_graph

de_transitive *De-transitivitize Graph*

Description

Useful for draving Hasse diagrams.

Usage

```
de_transitive(B)
```

Arguments

B	object of class igraph or a square 0-1 matrix of class Matrix or matrix

Value

object of class Matrix

See Also

Other binary_relations: closure_total_fair, closure_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_total, is_transitive, pord_weakdom, rel_graph

get_equivalence_classes

Get All Equivalence Classes of a Total Binary Relation

Description

Note that we assume that B is total, reflexive and transitive.

Usage

```
get_equivalence_classes(B)
```

Arguments

B	object of class `igraph` or a square 0-1 matrix of class `Matrix` or `matrix`

Value

list of integer vectors; each list element defines an equivalence class by listing vertices' numbers; each vector is ordered by the outdegrees of their nodes (they are the same in each class)

See Also

Other binary_relations: `closure_total_fair`, `closure_transitive`, `de_transitive`, `get_incomparable_pairs`, `get_independent_sets`, `is_reflexive`, `is_total`, `is_transitive`, `pord_weakdom`, `rel_graph`

get_incomparable_pairs

Get Incomparable Pairs in an Adjacency Matrix

Description

A pair (x,y) is incomparable iff not xRy and not xRy

Usage

```
get_incomparable_pairs(B)
```

Arguments

B	object of class `igraph` or a square 0-1 matrix of class `Matrix` or `matrix`

Details

See also `get_independent_sets` of how to generate all maximal independent sets.

Value

integer matrix with two columns (indices of incomparable elements, not that these are pairs, and not sets: you'll get (i,j) and (j,i))

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_independent_sets, is_reflexive, is_total, is_transitive, pord_weakdom, rel_graph

---

get_independent_sets *Get All Maximal Independent Sets*

Description

The function generates vectors of indices $S_j = \{i_1, ..., i_{k_j}\}$ such that all pairs from S_j are incomparable (A pair (i,i') is incomparable iff not iRi' and not $i'Ri$, see also get_incomparable_pairs.

Usage

```
get_independent_sets(B)
```

Arguments

B	object of class `igraph` or a square 0-1 matrix of class `Matrix` or `matrix`

Details

Note that we assume that B is transitive. Loops are not taken into account at all.

Value

list of integer vectors; each list element defines an independent set of vertices numbers

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, is_reflexive, is_total, is_transitive, pord_weakdom, rel_graph index_g *Egghe's g-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 g-index (Egghe, 2006) for x is defined as

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

if $n \geq 1$ and $x_1 \geq 1$, 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.

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

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

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

Value

a single numeric value

References

Egghe L., Theory and practise of the g-index, Scientometrics 69(1), 131-152, 2006.
Torra V., Narukawa Y., The h-index and the number of citations: Two fuzzy integrals. IEEE Transactions on Fuzzy Systems 16(3), 2008, 795-797.

See Also

Other impact_functions: index_h, index_lp, index_maxprod, index_rp, index_w, index.h, index.lp, index.rp

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 $x = (x_1, \ldots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the h-*index* (Hirsch, 2005) for x is defined as

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

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 non-increasingly sorted vector is given, the function is O(n).

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

See index_rp and owmax for natural generalizations.

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), 16569-16572, 2005.

See Also

Other impact_functions: `index_g`, `index_g_zi`, `index_lp`, `index_maxprod`, `index_rp`, `index_w`, `index.g`, `index.lp`, `index.rp`

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 $x = (x_1, \ldots, x_n)$, where $x_i \geq x_j$ for $i \leq j$, the l_p*-index* for $p = \infty$ equals to

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

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

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

where $MAXPROD$ is the index proposed in (Kosmulski, 2007), see `index_maxprod`.

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 `projection` to `identity` to obtain the 2-dimensional index

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

For historical reasons, this function is also available via its 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 algorithms for computing "geometric" scientific impact indices, Research Report of Systems Research Institute, Polish Academy of Sciences RB/1/2009, 2009b.
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.

See Also

Other impact_functions: `index_g`, `index_g_zi`, `index_h`, `index_maxprod`, `index_rp`, `index_w`, `index.g`, `index.h`, `index.rp`

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 $x = (x_1, \dots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the *MAXPROD-index* (Kosmulski, 2007) for x is defined as

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

Usage

```
index_maxprod(x)
```

Arguments

x	a non-negative numeric vector

Details

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

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.

See Also

Other impact_functions: `index_g`, `index_g_zi`, `index_h`, `index_lp`, `index_rp`, `index_w`, `index.g`, `index.h`, `index.lp`, `index.rp`

`index_rp` *The r_p-index*

Description

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

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

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 $x_1, \dots, x_n$ 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 non-increasingly sorted vector is given, the function is O(n).

For historical reasons, this function is also available via its 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), 16569-16572, 2005.
Woeginger G.J., An axiomatic characterization of the Hirsch-index, Mathematical Social Sciences, 56(2), 224-232, 2008.

See Also

Other impact_functions: `index_g`, `index_g_zi`, `index_h`, `index_lp`, `index_maxprod`, `index_w`, `index.g`, `index.h`, `index.lp`

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 $x = (x_1, \ldots, x_n)$, where $x_i \geq x_j \geq 0$ for $i \leq j$, the w-*index* (Woeginger, 2008) for x is defined as

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

Usage

```
index_w(x)
```

Arguments

`x`	a non-negative numeric vector

Details

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

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, 224-232.

See Also

Other impact_functions: `index_g`, `index_g_zi`, `index_h`, `index_lp`, `index_maxprod`, `index_rp`, `index.g`, `index.h`, `index.lp`, `index.rp`

is_reflexive *Check if Given Adjacency Matrix is Reflexive*

Description

A binary relation R is reflexive, iff for all x we have xRx. The function just checks whether all elements on the diagonal of B are non-zeros.

Usage

```
is_reflexive(B)
```

Arguments

B	object of class igraph or a square 0-1 matrix of class Matrix or matrix

Value

single logical value

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_total, is_transitive, pord_weakdom, rel_graph

is_total *Check if Given Adjacency Matrix is Total*

Description

A binary relation R is total, iff for all x, y we have xRy or yRx.

Usage

```
is_total(B)
```

Arguments

B	object of class igraph or a square 0-1 matrix of class Matrix or matrix

Value

single logical value

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_transitive, pord_weakdom, rel_graph

---

`is_transitive` *Check if Given Adjacency Matrix is Transitive*

Description

A binary relation R is transitive, iff for all x, y, z we have xRy and yRz => xRz

Usage

```
is_transitive(B)
```

Arguments

B	object of class `igraph` or a square 0-1 matrix of class `Matrix` or `matrix`

Value

single logical value

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_total, pord_weakdom, rel_graph

---

`owa` *WAM and OWA Operators*

Description

Computes the Weghted 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 greatest 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 of `w` does not sum up to 1, then they are normalized and a warning is generated.

Both functions return the ordinary arithmetic mean by default. Special cases of OWA include the trimmed mean (cf. `mean`) and winsorized mean.

There is a strong connection between the OWA operators and the Choquet integrals.

Value

single numeric value

References

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`, `owmin`, `wmax`, `wmin`

`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 greatest 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` return the greatest value in `x` by default, and `OWMin` and `WMin` - the smallest value in `x`.

Note that e.g. in the case of OWMax operator the aggregation w.r.t. `w` gives the same result as that of w.r.t. `sort(w)`. Moreover, classically, it is assumed that if we agregate vectors with elements in $[a, b]$, then the largest weight should be equal to b.

There is a strong connection between the OWMax/OWMin operators and the Sugeno integrals. 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

single numeric value

References

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

See Also

Other aggregation_operators: `owa`, `wam`

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

Description

Finds the maximum likelihood estimator of the type II Pareto 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 to look for
`smax`	upper bound for the scale parameter to look for
`tol`	the desired accuracy (convergence tolerance)

Details

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

For known s, the estimator is unbiased.

Value

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: dpareto2, pareto2_estimate_mmse, pareto2_test_f, ppareto2, qpareto2, rpareto2

---

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

Description

Finds the MMS estimator of the type II Pareto 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

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, 316-325.

See Also

Other Pareto2: dpareto2, pareto2_estimate_mle, pareto2_test_f, ppareto2, qpareto2, rpareto2 pareto2_test_f — *Two-Sample F-test For Equality of Shape Parameters for Type II-Pareto Distributions*

Description

Performs F-test for equality of shape parameters of two samples from the 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 bases on 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, has 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 passed as a 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 passed as a 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: `dpareto2`, `pareto2_estimate_mle`, `pareto2_estimate_mmse`, `ppareto2`, `qpareto2`, `rpareto2`

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

Description

Draws a step function that represents given numeric vector with elements in $[0, \infty]$.

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(0, max(x)), ...)
```

Arguments

`x`	non-negative numeric vector
`type`	character; type of the graphical '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`

Details

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

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

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

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

(for `type == 'left.continuous'`, the default) or by a curve joining 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_weakdom` *Weak Dominance Relation (Preorder)*

Description

$x <= y$ iff $nx <= ny$ AND for all $i = 1, ..., n$ $x_{(n-i+1)} <= y_{(m-i+1)}$, where $nx = length(x)$ and $ny = length(y)$.

Usage

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

Arguments

x	numeric vector
y	numeric vector

Details

This function accepts only vectors with nonnegative elements.

Value

single logical value; whether x <= y

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_total, is_transitive, rel_graph

---

rel_graph — *Create Adjacency Matrix of Given Binary Relation*

Description

Note that adjacency matrix can also be conceived as a directed graph (DAG). ret[i,j] iff i <= j

Usage

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

Arguments

x	list with elements to compare, preferrably named
pord	function with 2 arguments, returning boolean value
...	additional arguments passed to pord

Value

square 0-1 Matrix (of class Matrix)

See Also

Other binary_relations: closure_total_fair, closure_transitive, de_transitive, get_equivalence_classes, get_incomparable_pairs, get_independent_sets, is_reflexive, is_total, is_transitive, pord_weakdom

---

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 $s > 0$.

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

x,q	vector of quantiles
p	vector of probabilities
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

Details

If $X \sim \mathrm{P2}(k, s)$, then $\operatorname{supp} X = [0, \infty)$. 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 Pareto2: pareto2_estimate_mle, pareto2_estimate_mmse, pareto2_test_f

Index