A Guide to the agop 0.2-0 Package for R

Aggregation Operators and Preordered Sets in R

Marek Gagolewski[1,2], Anna Cena[1,2]
[1] Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Rexamine, Email: `{gagolews,cena}@rexamine.com`
http://agop.rexamine.com

February 19, 2014

The package, as well as this tutorial, is still in its early days – any suggestions are welcome!

Contents

1 Getting Started

> *"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 (...)." [28, p. xiii]*

R [39] is a free, open source software environment for statistical computing and graphics, which includes an implementation of a very powerful and quite popular high-level language called S. It runs on all major operating systems, i.e. Windows, Linux, and MacOS X. To install R and/or find some information on the S language please visit R Project's Homepage at www.R-project.org. Perhaps you may also wish to install RStudio, a convenient development environment for R. It is available at www.rsudio.org.

agop is an open source (licensed under GNU LGPL 3) package for R $\geq$ 2.12 to which anyone can contribute. It started as a fork of the CITAN (*Citation Analysis Toolpack*, [17]) package.

To install latest "official" release of the package available on *CRAN* we type[1]:

```r
install.packages('agop')
```

Alternatively, we may fetch its current development snapshot from *GitHub*:

```r
install.packages('devtools')
library('devtools')
install_github('agop', 'Rexamine')
```

Note that in this case you will need a working C/C++ compiler[2].

Each session with agop should be preceded by a call to:

```r
library('agop') # Load the package
```

To view the main page of the manual we type:

```r
library(help='agop')
```

For more information please visit the package's homepage [22]. In case of any problems, comments, or suggestions feel free to contact the authors. Good luck!

2 Theoretical Background

Let us establish some basic notation convention used throughout this tutorial. From now on let $\mathbb{I} = [a, b]$, possibly with $a = -\infty$ or $b = \infty$. Note that in many practical situations we commonly

[1] You are viewing the **development** version of the tutorial. Some of the features presented in this document may be missing in the CRAN release. Please, upgrade to the **latest** development version from *GitHub* if you need the new functionality. Note that you will need a working C/C++ compiler.

[2] Windows users should have Rtools installed, see cran.r-project.org/bin/windows/Rtools/.

choose $\mathbb{I} = [-1, 1]$, $\mathbb{I} = [0, 1]$ or $\mathbb{I} = [0, \infty]$. A set of all vectors of arbitrary length with elements in $\mathbb{I}$ is denoted by $\mathbb{I}^{1,2,\dots} = \bigcup_{n=1}^{\infty} \mathbb{I}^n$.

For two equal-length vectors $\mathbf{x}, \mathbf{y} \in \mathbb{I}^n$ we write $\mathbf{x} \le \mathbf{y}$ if and only if for all $i = 1, \dots, n$ it holds $x_i \le y_i$. Moreover, all binary arithmetic operations on vectors $\mathbf{x}, \mathbf{y} \in \mathbb{I}^n$ will be performed element-wise, e.g. $\mathbf{x} + \mathbf{y} = (x_1 + y_1, \dots, x_n + y_n) \in \mathbb{I}^n$. Similar behavior is assumed for $-$, $\cdot$, $/$, $\wedge$ (min), $\vee$ (max), etc. Additionally, each function of one variable $\mathsf{f} : \mathbb{I} \to \mathbb{I}$ can be extended to the vector space: we write $\mathsf{f}(\mathbf{x})$ to denote $(\mathsf{f}(x_1), \dots, \mathsf{f}(x_n))$.

Let $x_{(i)}$ denote the ith order statistic, i.e. the ith smallest value in $\mathbf{x}$. Moreover, for convenience, let $x_{\{i\}} = x_{|\mathbf{x}|-i+1}$ denote the ith greatest value in $\mathbf{x}$.

For any $n \in \mathbb{N}$ and $c \in \mathbb{I}$, we set $(n * c) = (c, \dots, c) \in \mathbb{I}^n$. Also, $[n] := \{1, 2, \dots, n\}$ with $[0] = \emptyset$.

2.1 A Note on Representing Numeric Data and Applying Operations in R

Recall how we create numeric vectors in R:

```
(x1 <- c(5, 2, 3, 1, 0, 0))

## [1] 5 2 3 1 0 0

class(x1)

## [1] "numeric"

(x2 <- 10:1) # the same as seq(10, 1)

##  [1] 10  9  8  7  6  5  4  3  2  1

(x3 <- seq(1, 5, length.out=6))

## [1] 1.0 1.8 2.6 3.4 4.2 5.0

(x4 <- seq(1, 5, by=1.25))

## [1] 1.00 2.25 3.50 4.75
```

To obtain $(n * c)$, e.g. for $n = 10$ and $c = 3$, we call:

```
rep(10, 3)

## [1] 10 10 10
```

Note that in R all the arithmetic operations on vectors are performed element-wise, i.e. in a manner indicated above. This is called ***vectorization***. The same holds for mathematical functions: they are extended to the vector space.

```
x <- c(1, 3,  3, 2)
y <- c(2, 3, -1, 0)
x+y

## [1] 3 6 2 2

x*y

## [1]  2  9 -3  0

pmin(x,y)

## [1]  1  3 -1  0

pmax(x,y)

## [1] 2 3 3 2
```

```
abs(y)

## [1] 2 3 1 0
```

Thus, we calculated $\mathbf{x}+\mathbf{y}$, $\mathbf{x}\cdot\mathbf{y}$, $\mathbf{x}\wedge\mathbf{y}$, $\mathbf{x}\vee\mathbf{y}$, and $|\mathbf{x}|$ (try to determine yourself what happens if we deal with vectors of unequal length is R).

Moreover, given two equal-length vectors, for the $\le$ relation we write:

```
all(x <= y)

## [1] FALSE
```

To get $x_{\{i\}}$ we have to sort the given vector nonincreasingly.

```
(xs <- sort(x, decreasing=TRUE)) # `decresing' may be misleading

## [1] 3 3 2 1

xs[3] # the third greatest value in x

## [1] 2
```

2.2 A Note on Storing Multiple Numeric Vectors in R

Vectors of the same length can be conveniently stored in a matrices. Please note that the `dimnames` attribute of a matrix may define its row and column labels. Its value may be set to `NULL` (no names given) or to a list with two character vectors (rows and columns names, respectively). Another simple way to set the labels is by using the `rownames()` and `colnames()` functions.

The `apply()` function may be called to evaluate a given method on each matrix row or column (parameter `MARGIN` set to 1 and 2, respectively).

```
expertopinions <- matrix(c(
      6,7,2,3,1, # this will be the first COLUMN
      8,3,2,1,9, # 2nd
      4,2,4,1,6  # 3rd
   ),
   ncol=3,
   dimnames=list(NULL, c("A", "B", "C")) # only column names set
)
class(expertopinions)

## [1] "matrix"

print(expertopinions)   # or print(authors)

##      A B C
## [1,] 6 8 4
## [2,] 7 3 2
## [3,] 2 2 4
## [4,] 3 1 1
## [5,] 1 9 6

apply(expertopinions, 2, mean) #  apply the mean() function on each COLUMN

##   A   B   C
## 3.8 4.6 3.4
```

Vectors that are not of the same length may be store in a list (with possibly named elements). In that case, the functionality of `apply()` is provided by `lapply()` or `sapply()` functions.

```
authors <- list(
   "John S." = c(7,6,2,1,0),
   "Kate F." = c(9,8,7,6,4,1,1,0)
)
class(authors)

## [1] "list"

str(authors)    # or print(authors)

## List of 2
##  $ John S.: num [1:5] 7 6 2 1 0
##  $ Kate F.: num [1:8] 9 8 7 6 4 1 1 0

index_h(authors[[1]]) # the h-index /see below/ for 1st author

## [1] 2

sapply(authors, index_h) # calculate the h-index for all vectors in a list

## John S. Kate F.
##       2       4

index_h(authors) # index_h() expects an numeric vector on input

## Error:  argument ‘x‘ should be a numeric vector (or an object coercible to)
```

2.3 Aggregation Operators and Their Basic Properties

Dealing with huge amounts of data faces us with the problem of constructing their synthetic descriptions. The aggregation theory, a relatively new research domain at the border of mathematics and computer science, is interested in the analysis of functions that may be used in this task. Thus, we should start with the formal definition of objects of our interest. Here is the most general setting:

Definition 1. A function $\mathsf{F} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ is called an ***(extended*[3]*) aggregation operator*** if it is at least ***nondecreasing*** in each variable, i.e. for all n and $\mathbf{x}, \mathbf{y} \in \mathbb{I}^n$ if $\mathbf{x} \le \mathbf{y}$, then $\mathsf{F}(\mathbf{x}) \le \mathsf{F}(\mathbf{y})$.

Note that each aggregation operator is a mapping into $\mathbb{I}$, thus for all n we have $\inf_{\mathbf{x}\in\mathbb{I}^n} \mathsf{F}(\mathbf{x}) \ge a$ and $\sup_{\mathbf{x}\in\mathbb{I}^n} \mathsf{F}(\mathbf{x}) \le b$. By nondecreasingness, however, these conditions reduce to $\mathsf{F}(n * a) \ge a$ and $\mathsf{F}(n * b) \le b$.

Also keep in mind that some authors assume (cf. [28]) that aggregation operators must fulfill the two following ***strong boundary conditions***: for all n we have $\inf_{\mathbf{x}\in\mathbb{I}^n} \mathsf{F}(\mathbf{x}) = a$ and $\sup_{\mathbf{x}\in\mathbb{I}^n} \mathsf{F}(\mathbf{x}) = b$. In our case, this does not necessarily hold – we want to be more general.

Here are some interesting properties of aggregation operators. Later on we will characterize the classes of aggregation operators that fulfill them.

Definition 2. We call $\mathsf{F} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ ***symmetric*** if:

$$(\forall n \in \mathbb{N})\ (\forall \mathbf{x}, \mathbf{y} \in \mathbb{I}^n)\ \mathbf{x} \cong \mathbf{y} \Longrightarrow \mathsf{F}(\mathbf{x}) = \mathsf{F}(\mathbf{y}),$$

where $\mathbf{x} \cong \mathbf{y}$ if and only if there exists a permutation σ of $[n]$ such that $\mathbf{x} = (y_{\sigma(1)}, \dots, y_{\sigma(n)})$.

[3]Extended to the space of vectors of arbitrary length, cf. e.g. [5, 28]; Classical approach considers only fixed-length vectors. In agop we are as much general as possible.

It may be shown, see [28, Thm. 2.34], that $\mathsf{F}:\mathbb{I}^n \to \mathbb{I}$ is symmetric if and only if there exists a function $\mathsf{G}':\mathbb{I}^{1,2,\dots} \to \mathbb{I}$ such that $\mathsf{F}(x_1,\dots,x_n) = \mathsf{G}'(x_{(1)},\dots,x_{(n)})$, or, equivalently, a function $\mathsf{G}'':\mathbb{I}^{1,2,\dots} \to \mathbb{I}$, for which we have $\mathsf{F}(x_1,\dots,x_n) = \mathsf{G}''(x_{\{1\}},\dots,x_{\{n\}})$. In other words, F may be defined solely using order statistics: its value is independent of the aggregated vector's elements presentation.

Idempotence is well-known from algebra, where we say that element x is idempotent with respect to binary operator $*$ if we have $x * x = x$. The following definition extends this property to n-ary aggregation functions, cf. [28].

Definition 3. We call $\mathsf{F}:\mathbb{I}^{1,2,\dots} \to \mathbb{I}$ ***idempotent*** if:

$$(\forall n \in \mathbb{N})\ (\forall x \in \mathbb{I})\ \mathsf{F}(n * x) = x.$$

Idempotent aggregation operators fulfilling the strong boundary conditions (see p. 5) are sometimes called ***averaging functions***, cf. [28].

An example of such object is the arithmetic mean or median.

Definition 4. We call $\mathsf{F}:\mathbb{I}^{1,2,\dots} \to \mathbb{I}$ ***additive*** if:

$$\mathsf{F}(\mathbf{x}+\mathbf{y}) = \mathsf{F}(\mathbf{x}) + \mathsf{F}(\mathbf{y}),$$

for all $n \in \mathbb{N}, \mathbf{x},\mathbf{y} \in \mathbb{I}^n$ such that $\mathbf{x}+\mathbf{y} \in \mathbb{I}^n$.

Please note that for $a \le 0$, if F is additive, then necessarily it holds $\mathsf{F}(\mathbf{0}) = 0$.

Definition 5. We call F ***minitive*** if:

$$(\forall n \in \mathbb{N})\ (\forall \mathbf{x},\mathbf{y} \in \mathbb{I}^n)\ \mathsf{F}(\mathbf{x}\wedge\mathbf{y}) = \mathsf{F}(\mathbf{x}) \wedge \mathsf{F}(\mathbf{y}).$$

Definition 6. We call F ***maxitive*** if:

$$(\forall n \in \mathbb{N})\ (\forall \mathbf{x},\mathbf{y} \in \mathbb{I}^n)\ \mathsf{F}(\mathbf{x}\vee\mathbf{y}) = \mathsf{F}(\mathbf{x}) \vee \mathsf{F}(\mathbf{y}).$$

Definition 7. We call F ***modular*** (cf. [4, 28, 32]) if:

$$(\forall n \in \mathbb{N})\ (\forall \mathbf{x},\mathbf{y} \in \mathbb{I}^n)\ \mathsf{F}(\mathbf{x}\vee\mathbf{y}) + \mathsf{F}(\mathbf{x}\wedge\mathbf{y}) = \mathsf{F}(\mathbf{x}) + \mathsf{F}(\mathbf{y})$$

It may easily be seen that each additive operator is also modular (i.e. modularity is more general than additivity), because for any additive aggregation operator F, since $(\mathbf{x}\vee\mathbf{y})+(\mathbf{x}\wedge\mathbf{y}) = \mathbf{x}+\mathbf{y}$, we have $\mathsf{F}(\mathbf{x}) + \mathsf{F}(\mathbf{y}) = \mathsf{F}(\mathbf{x}+\mathbf{y}) = \mathsf{F}((\mathbf{x}\vee\mathbf{y}) + (\mathbf{x}\wedge\mathbf{y})) = \mathsf{F}(\mathbf{x}\vee\mathbf{y}) + \mathsf{F}(\mathbf{x}\wedge\mathbf{y})$.

Apart from the "ordinary" minitivity, maxitivity, and modularity we may introduce their symmetrized versions, using $\mathbf{x} \overset{S}{+} \mathbf{y} = (x_{(1)}+y_{(1)},\dots,x_{(n)}+y_{(n)})$, $\mathbf{x} \overset{S}{\vee} \mathbf{y} = (x_{(1)}\vee y_{(1)},\dots,x_{(n)}\vee y_{(n)})$ and $\mathbf{x} \overset{S}{\wedge} \mathbf{y} = (x_{(1)}\wedge y_{(1)},\dots,x_{(n)}\wedge y_{(n)})$.

2.4 Impact Functions and The Producers Assessment Problem

We already noticed the important class of aggregation operators: the averaging functions. They may be used to represent the most "typical" value of a numeric vector. Here is another interesting class that represents solutions to some very interesting practical issue.

Tab. 1. The Producer Assessment Problem – typical instances

	Producer	Products	Rating method	Discipline
A	Scientist	Scientific articles	Number of citations	Scientometrics
B	Scientific institute	Scientists	The h-index	Scientometrics
C	Web server	Web pages	Number of in-links	Webometrics
D	Artist	Paintings	Auction price	Auctions
E	Billboard company	Advertisements	Sale results	Marketing
F	R package author	Packages	PageRank values on the citation graph	Software Engineering

The ***Producers Assessment Problem*** (PAP, [26]) concerns evaluation of a set of ***producers*** (e.g. scientists, artists, writers, craftsman) according to some quality or popularity ***ratings*** of ***products*** (e.g. scientific articles, works, books, artifacts) that were outputted by an entity.

PAP instances may be found in many real-life situations, like those encountered for example in scientometrics, webometrics, marketing, manufacturing, or quality engineering, see Table 1 and e.g. [14]. Our main interest here is focused on constructing and analyzing aggregation operators which may be used in the producers' rating task. Such functions should take into account the two following aspects of a producer's quality:

- his/her ability to output highly-rated products,
- his/her overall productivity.

For the sake of illustration, we will consider PAP in the scientometric context, where scientists "produce" papers that are cited by peers.

Let $\mathbb{I} = [0, \infty]$ represent the set of values that some a priori chosen paper quality measure may take. These may of course be non-integers, for example when we consider citations normalized with respect to the number of papers' authors.

It is widely accepted, see e.g. [44, 43, 45, 36, 34, 35, 26, 16, 15], that each aggregation operator $\mathsf{F} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ to be applied in PAP should at least be:

(a) nondecreasing in each variable (additional citations received by a paper or an improvement of its quality measure does not result in a decrease of the authors' overall evaluation),

(b) arity-monotonic (by publishing a new paper we never decrease the overall valuation of the entity),

(c) symmetric (independent of the order of elements' presentation, i.e. we may always assume that we aggregate vectors that are already sorted).

More formally, axiom (b) is fulfilled iff for any $\mathbf{x} \in \mathbb{I}^{1,2,\dots}$ and $y \in \mathbb{I}$ it holds $\mathsf{F}(\mathbf{x}) \leq \mathsf{F}(x_1, \dots, x_n, y)$. It may be seen that this property is ***arity-dependent***, i.e. it takes into account the number of elements to be aggregated.

Moreover, (a) and (c) were defined in the previous section.

Here is a bunch of arity-dependent properties that can be useful while aggregating vectors of varying lengths.

Definition 8. We call $\mathsf{F} \in \mathcal{E}(\mathbb{I})$ a ***zero-insensitive*** aggregation operator if for each $\mathbf{x} \in \mathbb{I}^{1,2,\dots}$ it holds $\mathsf{F}(\mathbf{x}, 0) = \mathsf{F}(\mathbf{x})$.

It may be seen that, under nondecreasingness, zero-insensitivity implies arity-monotonicity, see [24]. What is interesting, each zero-insensitive impact function F may be defined by means of $\mathsf{G} : \mathbb{I}^\infty \to \mathbb{I}$ such that $\mathsf{F}(\mathbf{x}) = \mathsf{G}(\mathbf{x}, 0, 0, \dots)$, i.e. of function which domain is the space of vectors of infinite length.

Zero-sensitivity may be strengthened as follows, cf. [24] and [44, Axiom A1].

Definition 9. $\mathsf{F} \in \mathcal{E}(\mathbb{I})$ is ***F*-insensitive** if

$$(\forall \mathbf{x} \in \mathbb{I}^{1,2,\dots})\ (\forall y \in \mathbb{I})\ y \le \mathsf{F}(\mathbf{x}) \Longrightarrow \mathsf{F}(\mathbf{x}, y) = \mathsf{F}(\mathbf{x}).$$

Note that the above property was called R-stability in [3].

Definition 10. $\mathsf{F} \in \mathcal{E}(\mathbb{I})$ is ***F*+sensitive** if

$$(\forall \mathbf{x} \in \mathbb{I}^{1,2,\dots})\ (\forall y \in \mathbb{I})\ y > \mathsf{F}(\mathbf{x}) \Longrightarrow \mathsf{F}(\mathbf{x}, y) > \mathsf{F}(\mathbf{x}).$$

2.5 Dispersion Functions

Classically, in the theory of aggregation we consider the broadly-conceived averaging functions and logical connectives. However, one often needs a very different kind of a proper synthesis of multidimensional numeric data into a single number. In [21] an axiomatization of ***dispersion functions***, which may be used to measure data variability, spread, or scatter, was proposed.

Given $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$, we write $\mathbf{x} \preccurlyeq \mathbf{x}'$ and say that $\mathbf{x}$ ***has not greater spread than*** $\mathbf{x}'$, if and only if for all $i, j = 1, \dots, n$ it holds

$$x_i \ge x_j \Longrightarrow x_i - x_j \le x'_i - x'_j. \tag{1}$$

It is easily seen that $\preccurlyeq$ is a preorder on $\mathbb{R}^n$, i.e. a relation that is reflexive and transitive, and that it is not necessarily total, i.e. not all vectors are comparable with each other, see also Sec. 4.3.

Definition 11. A ***dispersion function*** is an aggregation operator $\mathsf{V} : \mathbb{R}^n \to [0, \infty]$ such that:

(d1) For any $c \in \mathbb{R}$ it holds $\mathsf{V}(n * c) = \mathsf{V}(c, c, \dots, c) = 0$,

(d2) for each $\mathbf{x} \preccurlyeq \mathbf{x}'$ it holds $\mathsf{V}(\mathbf{x}) \le \mathsf{V}(\mathbf{x}')$.

This class includes e.g. the sample variance (see `var()`), standard veriation (see `sd()`), range, interquartile range (IQR, see `IQR()`), median absolute deviation (MAD) etc., that is functions widely used in exploratory data analysis as descriptive statistics.

3 Visualization

3.1 Depicting Producers

The `plot_producer()` function may be used to draw a graphical representation of a given numeric vector, i.e. what is sometimes called a ***citation function*** in scientometrics.

As in the PAP we are interested in symmetric agops, a given vector $\mathbf{x} = (x_1, \dots, x_n)$ may be represented by a step function defined for $0 \le y < n$ and given by:

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

This function may be obtained by setting `type='right.continuous'` argument in `plot_producer()`. Recall that $x_{\{i\}}$ denotes i-th greatest value in $\mathbf{x}$.

On the other hand, for `type='left.continuous'` (the default), we get

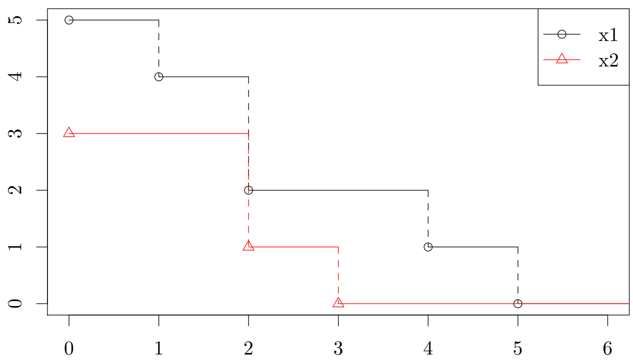
$$\pi(y) = x_{\{\lfloor y \rfloor\}}$$

for $0 < y \le n$.

Note that this function depicts the curve that joins the sequence of points $(0, x_{\{1\}}), (1, x_{\{1\}}), (1, x_{\{2\}}), (2, x_{\{2\}}), \ldots, (n, x_{\{n\}})$.

The `plot_producer()` function behaves much like the well-known R's `plot.default()` and allows for passing all its graphical parameters. For example, let us depict the state of two given producers, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$.

```
x1 <- c(5, 4, 2, 2, 1)
x2 <- c(3, 3, 1, 0, 0, 0, 0)
plot_producer(x1, extend=TRUE)
plot_producer(x2, add=TRUE, col=2, pch=2, extend=TRUE)
legend('topright', c('x1', 'x2'), col=c(1, 2), lty=1, pch=c(1, 2))
```



4 Binary Relations

The agop package includes a few functions aiming to deal with binary relations[4] defined on finite sets consisting of distinctive elements $V = \{v_1, \ldots, v_n\}$. Each binary relation $R \subseteq V \times V$ may be represented by a square 0-1[5] matrix $A = (a_{i,j})_{i,j=1,\ldots,n}$ such that $a_{i,j} = 1$ if and only if $v_i R v_j$.

Note that we write $R \subseteq R'$ if $v_i R v_j \implies v_i R' v_j$, and that if $R = V \times V$ then for all i, j it holds $v_i R v_j$.

Table 2 contains an overview of the most often considered properties of binary relations. Moreover, Table 3 lists popular classes of relations.

Reductions and closures To determine the ***reflexive closure***, i.e. the minimal reflexive $R' \supseteq R$ call `rel_closure_reflexive()`. On the other hand, with a call to `rel_reduction_reflexive()` we get the ***reflexive reduction*** of R, i.e. the minimal $R' \subseteq R$ such that the reflexive closures of R' and R are equal. In other words, R' is the largest irreflexive relation contained in R.

[4]On CRAN there is also a package relations, see [33], that provides data structures and algorithms for k-ary relations with arbitrary domains, featuring relational algebra, predicate functions, and fitters for consensus relations. For some time, agop's functionality will be a subset of relations's (yet faster). In future versions, however, we'd like to add fuzzy relations handling.

[5]Or, equivalently, logical; we have `as.logical(0) == FALSE` and `as.logical(x) == TRUE` if `x` $\neq 0$, and, on the other hand, `as.integer(FALSE) == 0 as.integer(TRUE) == 1`.

Tab. 2. Properties of binary relations.

Property	**Definition**	agop implementation
Reflexivity	$(\forall i)\ v_i R v_i$	`rel_is_reflexive()`
Irreflexivity	$(\forall i)\ \neg v_i R v_i$	`rel_is_irreflexive()`
Symmetry	$(\forall i,j)\ v_i R v_j \Rightarrow v_j R v_i$	`rel_is_symmetric()`
Antisymmetry	$(\forall i,j)\ v_i R v_j$ and $v_j R v_i \Rightarrow i = j$	`rel_is_antisymmetric()`
Asymmetry	$(\forall i,j)\ v_i R v_j \Rightarrow \neg v_j R v_i$	`rel_is_asymmetric()`
Totality	$(\forall i,j)\ v_i R v_j$ or $v_j R v_i$	`rel_is_total()`
Transitivity	$(\forall i,j,k)\ v_i R v_i$ and $v_j R v_k \Rightarrow v_i R v_k$	`rel_is_transitive()`
Cyclicity	transitive closure of R is not antisymmetric	`rel_is_cyclic()`

Tab. 3. Types of binary relations.

Class	**Properties**
preorder (quasiorder)	reflexive, transitive
total preorder (weak order, preference)	total ($\Rightarrow$ reflexive), transitive
partial order	reflexive, transitive, antisymmetric
linear order	total ($\Rightarrow$ reflexive), transitive, antisymmetric
equivalence relation	symmetric, reflexive, transitive

To find the ***transitive closure***, cf. [42], of a given binary relation R, i.e. the minimal transitive $R' \supseteq R$, we call agop's `rel_closure_transitive()` function. On the other hand, the ***transitive reduction*** of acyclic R, see [1] and the `rel_reduction_transitive()` function, is the minimal $R' \subseteq R$ such that the transitive closures of R and R' are equal.

A mixture of the reflexive reduction and some kind of transitive reduction, particularly useful when drawing Hasse diagrams of preoredered sets (which may not necessarily be represented by an acyclic relation R) may be determined with `rel_reduction_hasse()`.

In general, a total closure and a total reduction are not well-defined. However, when dealing with preorders, the following notion may be useful, see [19]. To determine the so-called ***fair total closure*** i.e. minimal total $R' \supseteq R$ such that if $\neg xRy$ and $\neg xRy$ then $xR'y$ and $yR'x$, we call `rel_closure_total_fair()`.

The ***symmetric closure***, the smallest symmetric binary relation that contains a given one, is available via a call to `rel_closure_symmetric()`.

4.1 Weak Dominance Relation (for PAP)

Let us consider the following relation on $\mathbb{I}^{1,2,\dots}$. For any $\mathbf{x} \in \mathbb{I}^n$ and $\mathbf{y} \in \mathbb{I}^m$ we write $\mathbf{x} \trianglelefteq \mathbf{y}$ if and only if $n \le m$ and $x_{\{i\}} \le y_{\{i\}}$ for all $i = 1, \dots, n \wedge m$.

Of course, $\trianglelefteq$ is symmetric and transitive, i.e. it is a preorder. Moreover, it would have been a partial order (in general it's not), if we had defined it on the set of *sorted* vectors.

Intuitively, we say that an author (scientometric context again) X is (weakly) dominated by an author Y, if X has no more papers than Y and each the ith most cited paper of X has no more citations than the ith most cited paper of Y. Not that the $(m - n)$ least cited Y's papers are not taken into account here.

Most importantly, however, there exist pairs of vectors that are *incomparable* with respect to $\trianglelefteq$ (see the illustration below). In other words, this dominance relation is not total.

Whether this relationship between a pair of vectors holds may be determined using agop's `pord_weakdom()` function.

```
c(pord_weakdom(5:1, 10:1), pord_weakdom(10:1, 5:1)) # 5:1 <= 10:1
## [1]  TRUE FALSE
c(pord_weakdom(3:1, 5:4),  pord_weakdom(5:4, 3:1))  # 3:1 ?? 5:4
## [1] FALSE FALSE
```

The following result was shown in [26]. Let $\mathsf{F} \in \mathcal{E}(\mathbb{I})$. Then F is symmetric, nondecreasing in each variable and arity-monotonic if and only if for any $\mathbf{x}, \mathbf{y}$ if $\mathbf{x} \trianglelefteq \mathbf{y}$, then $\mathsf{F}(\mathbf{x}) \le \mathsf{F}(\mathbf{y})$. Therefore, the class of impact functions may be equivalently defined as all the aggregation operators that are nondecreasing with respect to this preorder.

Additionally, we will write $\mathbf{x} \triangleleft \mathbf{y}$ if $\mathbf{x} \trianglelefteq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$ (strict dominance).

4.2 Weak Dominance Relation (for fixed arity)

Consider the following relation on $\mathbb{I}^{1,2,\dots}$. For any $\mathbf{x} \in \mathbb{I}^n$ and $\mathbf{y} \in \mathbb{I}^m$ we write $\mathbf{x} \le \mathbf{y}$ if and only if $n = m$ and $x_i \le y_i$ for all $i = 1, \dots, n$.

The `all(x <= y)` expression may not work well for vectors of unequal lengths.

...TO DO...

Moreover, a version that is independent of the vectors' elements ordering (i.e. defined using $x_{(i)} \le y_{(i)}$) is available:

...TO DO...

4.3 Vector Spread

The function `pord_spread()` can be used to compare spread, scater, or variability of two numeric vectors, see [21] with the $\preccurlyeq$ relation introduced in Sec. 2.5 and [21].

Moreover, for `pord_spreadsym()` we get the relation which acts on sorted input vectors (which leads to symmetric dispersion functions).

```
pord_spreadsym(c(1,5,2), c(1,7,3))
## [1] TRUE
```

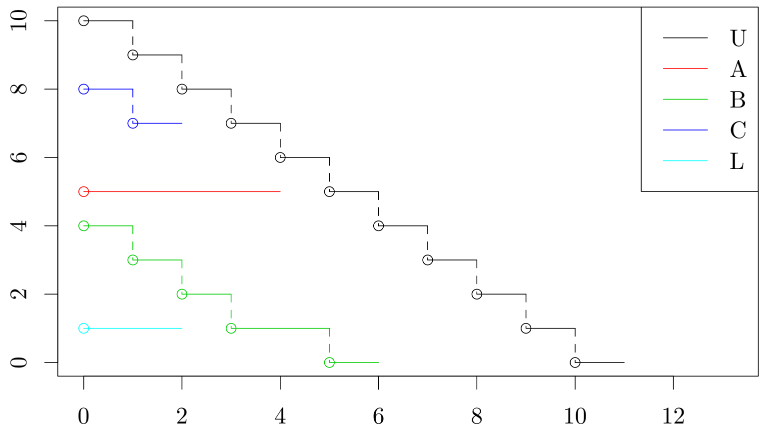
4.4 Operations on Preorders and Other Binary Relations

Example. Let us consider the 5 following vectors.

```
ex1 <- list(
   U = 10:0,             # some upper bound
   A = c(5,5,5,5),       # moderate productivity & quality
   B = c(4,3,2,1,1,0), # high productivity
   C = c(8,7),           # high quality
   L = c(1,1)            # some lower bound
)
```

Plot of "citation" curves:

```
for (i in seq_along(ex1))
plot_producer(ex1[[i]], add=(i>1), col=i)
legend("topright", legend=names(ex1), col=1:length(ex1), lty=1)
```



Here is the adjacency matrix for the preordered set $(\{A, B, C, L, U\}, \trianglelefteq)$ created with the **rel_graph()** function, that takes each pair of elements from the list passed as its first argument and compares them with a given function passed as its second argument.

```
ord <- rel_graph(ex1, pord_weakdom) # compare each (ex1[[i]], ex1[[j]]) with pord_weakdom
print(ord)

##      U     A     B     C     L
## U TRUE FALSE FALSE FALSE FALSE
## A TRUE  TRUE FALSE FALSE FALSE
## B TRUE FALSE  TRUE FALSE FALSE
## C TRUE FALSE FALSE  TRUE FALSE
## L TRUE  TRUE  TRUE  TRUE  TRUE

rel_is_reflexive(ord)  # is reflexive

## [1] TRUE

rel_is_transitive(ord) # is transitive

## [1] TRUE

rel_is_total(ord)      # not a total preorder...

## [1] FALSE
```

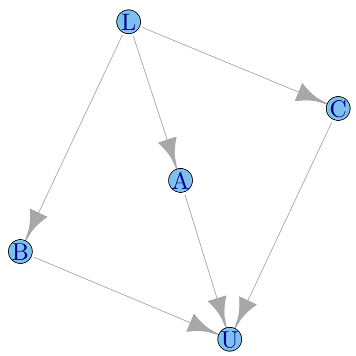
We see that we have $A??B$, $A??C$, $B??C$ (no pair from $\{A, B, C\}$ is comparable w.r.t. $\trianglelefteq$):

```
incomp <- get_incomparable_pairs(ord)
incomp <- incomp[incomp[,1]<incomp[,2],] # remove permutations: ((1,2), (2,1))->(1,2)
incomp[,] <- rownames(ord)[incomp]
print(incomp) # all incomparable pairs

##   row col
## A "A" "B"
## A "A" "C"
## B "B" "C"

# the other way: generate maximal independent sets
lapply(get_independent_sets(ord), function(set) rownames(ord)[set])

## [[1]]
```

```
## [1] "A" "B" "C"
```

To draw the Hasse diagram, we base on the reflexive and a kind of transitive reduction of the graph, which is determined with `rel_reduction_hasse()`.

```
require(igraph)
hasse <- graph.adjacency(rel_reduction_hasse(ord))
set.seed(1234567) # igraph's drawing facilities are far from perfect
plot(hasse, layout=layout.fruchterman.reingold(hasse, dim=2))
```

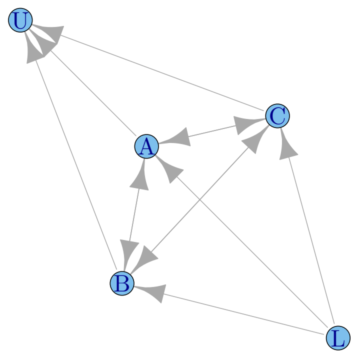


$(\{A, B, C, L, U\}, \trianglelefteq)$ is not totally ordered, let us determine the ***fair total closure*** of $\trianglelefteq$ (set $x \trianglelefteq'' y$ and $y \trianglelefteq'' x$ whenever $\neg(x \trianglelefteq y$ or $y \trianglelefteq x)$, see [19] for discussion), and then calculate its transitive closure, as the resulting matrix may not necessarily be transitive.

```
ord_total <- rel_closure_transitive(rel_closure_total_fair(ord)) # a total preorder
print(ord_total)
```

```
##      U     A     B     C     L
## U TRUE FALSE FALSE FALSE FALSE
## A TRUE  TRUE  TRUE  TRUE FALSE
## B TRUE  TRUE  TRUE  TRUE FALSE
## C TRUE  TRUE  TRUE  TRUE FALSE
## L TRUE  TRUE  TRUE  TRUE  TRUE
```

```
hasse <- graph.adjacency(rel_reduction_hasse(ord_total))
set.seed(123)
plot(hasse, layout=layout.fruchterman.reingold(hasse, dim=2))
```



Note that each total preorder $\trianglelefteq''$ induces an equivalence relation ($x \simeq y$ iff $x \trianglelefteq'' y$ and $y \trianglelefteq'' x$; the equivalence classes may be ordered with $\trianglelefteq''$).

5 Predefined Classes of Aggregation Operators in agop

5.1 A Review of Notable Classes of Aggregation Operators

Here are some well-known classes of aggregation operators. Originally, they were defined for fixed-length vector and for $\mathbb{I} = [0,1]$.

Definition 12. Let $\mathbf{w} = (w_1, \dots, w_n) \in [0,1]^n$ be a ***weighting vector*** such that $\sum_{i=1}^n w_i = 1$. Then, for any $\mathbf{x} \in \mathbb{I}^n$:

1. The ***weighted arithmetic mean*** associated with $\mathbf{w}$, $\mathsf{WAM}_\mathbf{w} : \mathbb{I}^n \to \mathbb{I}$, is defined as
$$\mathsf{WAM}_\mathbf{w}(\mathbf{x}) = \sum_{i=1}^n w_i x_i.$$

2. The ***ordered weighted averaging operator*** (cf. [46]) associated with $\mathbf{w}$, $\mathsf{OWA}_\mathbf{w} : \mathbb{I}^n \to \mathbb{I}$, is defined as
$$\mathsf{OWA}_\mathbf{w}(\mathbf{x}) = \sum_{i=1}^n w_i x_{(i)}.$$

We see that both functions are idempotent, additive, and that OWA is the symmetrized version of WAM. Moreover, for $\mathbf{w} = (n * \frac{1}{n})$, $\mathsf{WAM}_\mathbf{w}$ defines the arithmetic mean (`mean()` in R). ***Truncated mean*** is an interesting example of an OWA operator (see `mean(x, trim=...)`).

In agop the WAM and OWA operators are available as `wam()` and `owa()`.

```
wam(c(1,2,2,2), c(0.1,0.4,0.4,0.1))

## [1] 1.9

owa(c(1,3,5,2), rep(1,4)) # should be normalized

## Warning:  elements of 'w' does not sum up to 1.  correcting.
## [1] 2.75
```

Note that there is a strong, well-known connection between the OWA operators and the Choquet integral [8] w.r.t. some monotone measure, see e.g. [28].

Definition 13. Let $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{I}^n$ be a vector such that $\bigvee_{i=1}^n w_i = b = \sup \mathbb{I}$. Then, for any $\mathbf{x} \in \mathbb{I}^n$:

1. The ***weighted maximum*** associated with $\mathbf{w}$, $\mathsf{WMax}_{\mathbf{w}} : \mathbb{I}^n \to \mathbb{I}$, is defined as
$$\mathsf{WMax}_{\mathbf{w}}(\mathbf{x}) = \bigvee_{i=1}^{n} (w_i \wedge x_i).$$
2. The ***ordered weighted maximum*** (cf. [12, 11]) associated with $\mathbf{w}$, $\mathsf{OWMax}_{\mathbf{w}} : \mathbb{I}^n \to \mathbb{I}$, is defined as
$$\mathsf{OWMax}_{\mathbf{w}}(\mathbf{x}) = \bigvee_{i=1}^{n} (w_i \wedge x_{(i)}).$$

agop implementation: `wmax()` and `owmax()`.

```
wmax(c(1,3,5,2), Inf) # no vectorization here!
## Error:  'x' and 'w' should be of equal lengths
wmax(c(1,3,5,2), rep(Inf, 4)) # greatest value /default behavior/
## [1] 5
owmax(1:10, 1:10)
## [1] 10
```

Definition 14. Let $\mathbf{w} = c(w_1, \ldots, w_n) \in [0, 1]^n$ be such that $\bigwedge_{i=1}^n w_i = a = \inf \mathbb{I}$. Then, for any $\mathbf{x} \in \mathbb{I}^n$:

1. The ***weighted minimum*** $\mathsf{WMin}_{\mathbf{w}} : \mathbb{I}^n \to \mathbb{I}$ associated with the weight vector $\mathbf{w}$ is defined as
$$\mathsf{WMin}_{\mathbf{w}}(\mathbf{x}) = \bigwedge_{i=1}^{n} (w_i \vee x_i).$$
2. The ***ordered weighted minimum*** $\mathsf{OWMin}_{\mathbf{w}} : \mathbb{I}^n \to \mathbb{I}$ associated with the weight vector $\mathbf{w}$ is defined as
$$\mathsf{OWMin}_{\mathbf{w}}(\mathbf{x}) = \bigwedge_{i=1}^{n} (w_i \vee x_{(i)}).$$

agop implementation: `wmin()` and `owmin()`.

It is clear to see that OWMax operators fulfill the maxitivity property and OWMin operators fulfill the minitivity property. Interestingly, it may be shown, cf. [28], that for each OWMax operator there exist an equivalent OWMin operator and inveresly.

As stated above, "classical" aggregation operators were defined for vectors of fixed lengths. Let us present some notable generalizations of these operators.

Let $\mathbb{I}^{\mathbb{I}}$ denote the set of functions from $\mathbb{I}$ to $\mathbb{I}$. The following object will be needed for further considerations.

Definition 15. A ***triangle of functions*** is a sequence $\triangle = (\mathsf{f}_{i,n} \in \mathbb{I}^{\mathbb{I}} : i \in [n], n \in \mathbb{N})$.

Here is a graphical interpretation of $\triangle$:

$$\begin{array}{llll}
\mathsf{f}_{1,1} & & & \\
\mathsf{f}_{1,2} & \mathsf{f}_{2,2} & & \\
\mathsf{f}_{1,3} & \mathsf{f}_{2,3} & \mathsf{f}_{3,3} & \\
\vdots & \vdots & \vdots & \ddots
\end{array}$$

Definition 16. Let $\triangle = (\mathsf{f}_{i,n})_{i\in[n],n\in\mathbb{N}}$ be a triangle of functions such that $(\forall n)\ \sum_{i=1}^{n} \inf \mathsf{f}_{i,n} \geq a$ and $(\forall n)\ \sum_{i=1}^{n} \sup \mathsf{f}_{i,n} \leq b$. Then the ***quasi-L-statistic*** generated by $\triangle$ is a function $\mathsf{qL}_{\triangle} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ such that

$$\mathsf{qL}_{\triangle}(\mathbf{x}) = \sum_{i=1}^{n} \mathsf{f}_{i,n}(x_{\{i\}}).$$

It is easily seen that quasi-L-statistics generalize OWA operators if we set $\mathsf{f}_{i,n}(x) = c_{n-i+1,n}x$, $c_{i,n} \in [0,1]$, and $(\forall n)\ \sum_{i=1}^{n} c_{i,n} = 1$.

Assume that $\mathbb{I} = [0,b]$. Interestingly, it has been shown ([32], cf. also [18]) that an aggregation operator $\mathsf{F} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ fulfills the symmetric modularity property if and only if F is a nondecreasing quasi-L-statistic. What is more, in [18] we may find that $\mathsf{qL}_{\triangle}$ is nondecreasing if and only if there exists $\triangledown = (\mathsf{g}_{i,n})_{i\in[n],n\in\mathbb{N}}$ such that $(\forall n)\ (\forall i \in [n])\ \mathsf{g}_{i,n}$ is nondecreasing, $(\forall n)\ \sum_{i=1}^{n} \mathsf{g}_{i,n} \leq b$, $(\forall n)\ (\forall i > 1)\ \mathsf{g}_{i,n}(0) = 0$ and $\mathsf{qL}_{\triangle} = \mathsf{qL}_{\triangledown}$.

Definition 17. The ***quasi-S-statistic*** for a given triangle of functions $\triangle = (f_{i,n})_{i\in[n],n\in\mathbb{N}}$ is a function $\mathsf{qS}_{\triangle} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ such that

$$\mathsf{qS}_{\triangle}(\mathbf{x}) = \bigvee_{i=1}^{n} \mathsf{f}_{i,n}(x_{\{i\}}),$$

for any $\mathbf{x} \in \mathbb{I}^{1,2,\dots}$.

Quasi-S-statistic generalize the OWMax operators, if $\mathsf{f}_{i,n}(x) = x \wedge c_{n-i+1,n}$, $c_{i,n} \in \mathbb{I}$ and $(\forall n)\ \bigvee_{i=1}^{n} c_{i,n} = b$.

There is an equivalence between symmetric maxitive aggregation operators and nondecreasing quasi-S-statistics. Moreover, without loss of generality we may assume that a nondecreasing quasi-S-statistic is always generated by triangle of functions in which $(\forall n)\ (\forall i \in [n])\ \mathsf{f}_{i,n}$ is nondecreasing, $(\forall n)\ (\forall i \in [n])\ \mathsf{f}_{i,n}(a) = \mathsf{f}_{n,n}(a)$ and $(\forall n)\ \mathsf{f}_{1,n} \preceq \cdots \preceq \mathsf{f}_{n,n})$, see [18].

Definition 18. The ***quasi-I-statistic*** generated by $\triangle = (\mathsf{f}_{i,n})_{i\in[n],n\in\mathbb{N}}$ is a function $\mathsf{qI}_{\triangle} : \mathbb{I}^{1,2,\dots} \to \mathbb{I}$ such that

$$\mathsf{qI}_{\triangle}(\mathbf{x}) = \bigwedge_{i=1}^{n} \mathsf{f}_{i,n}(x_{\{i\}}),$$

for any $\mathbf{x} \in \mathbb{I}^{1,2,\dots}$.

Quasi-I-statistics are generalizations of the OWMin operators, if $\mathsf{f}_{i,n}(x) = x \vee c_{n-i+1,n}$, $c_{i,n} \in \mathbb{I}$ and $(\forall n)\ \bigwedge_{i=1}^{n} c_{i,n} = a$.

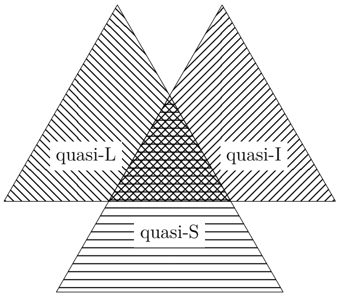
Like above, it has been shown that every symmetric minitive aggregation operator is a nondecreasing quasi-I-statistic, and conversely. Additionally, with no loss in generality we may assume that nondecreasing quasi-S-statistic is generated by triangle of functions in which

$(\forall n)$ $(\forall i \in [n])$ $\mathsf{f}_{i,n}$ is nondecreasing, $(\forall n)$ $(\forall i \in [n])$ $\mathsf{f}_{i,n}(b) = \mathsf{f}_{n,n}(b)$ and $(\forall n)$ $\mathsf{f}_{1,n} \preceq \cdots \preceq \mathsf{f}_{n,n}$, see [18].

Note: sometimes we also consider L-, S-, and I-statistics, i.e. special cases of the above-defined quasi-·-statistics, generated by triangles of coefficients (i.e. sequences $\triangle = (c_{i,n} \in \mathbb{I} : i \in [n], n \in \mathbb{N})$, cf. [5]). An ***L-statistic*** is a quasi-L-statistic for which we have $\mathsf{f}_{i,n}(x) = c_{i,n}x$. Similarly, by setting $\mathsf{f}_{i,n}(x) = x \wedge c_{i,n}$ we obtain an ***S-statistic*** from the quasi-S-statistics class, and by setting $\mathsf{f}_{i,n}(x) = x \vee c_{i,n}$ we get an ***I-statistic*** from quasi-I-statistics.

Also note that L-statistics are known from the probability theory. However, sometimes under this name some authors understand sums of a function of order statistics.

Most interestingly, in [18] it has been shown that the intersection of any two of the three "quasi" classes is the same:



Basing on this result, the ***OM3*** class (symmetric maxitive, minitive, and also modular aggregation operators) was proposed in [6, 7].

Definition 19. A sequence of nondecreasing functions $\mathbf{w} = (\mathsf{w}_1, \mathsf{w}_2, \dots)$, $\mathsf{w}_i : \mathbb{I} \to \mathbb{I}$, and a triangle of coefficients $\triangle = (c_{i,n})_{i\in[n],n\in\mathbb{N}}$, $c_{i,n} \in \mathbb{I}$ such that $(\forall n)$ $c_{1,n} \le c_{2,n} \le \cdots \le c_{n,n}$, $0 \le \mathsf{w}_n(0) \le c_{1,n}$, and $\mathsf{w}_n(b) = c_{n,n}$, generates a nondecreasing ***OM3 operator*** $\mathsf{M}_{\triangle,\mathbf{w}} : \mathbb{I}^n \to \mathbb{I}$ such that for $\mathbf{x} \in \mathbb{I}^n$ we have:

$$\begin{aligned}\mathsf{M}_{\triangle,\mathbf{w}}(\mathbf{x}) &= \bigvee_{i=1}^{n} \mathsf{w}_n(x_{(n-i+1)}) \wedge c_{i,n} = \bigwedge_{i=1}^{n} (\mathsf{w}_n(x_{(n-i+1)}) \vee c_{i-1,n}) \wedge c_{n,n} \\ &= \sum_{i=1}^{n} \left(\left(\mathsf{w}_n(x_{(n-i+1)}) \vee c_{i-1,n} \right) \wedge c_{i,n} - c_{i-1,n} \right).\end{aligned}$$

We see that the OM3 class contains i.a. all order statistics (whenever $\mathsf{w}_n(x) = x$, and $c_{i,n} = 0$, $c_{j,n} = b$ for $i < k$, $j \ge k$, and some k), OWMax operators (for $\mathsf{w}_n(x) = x$), and the famous Hirsch h-index (see below).

5.2 Interesting Impact Functions

Let us go back to the Producers Assessment Problem. Below we assume that $\mathbb{I} = [0, \infty]$.

The h-index. Given a sequence $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{I}^{1,2,\dots}$, the ***Hirsch index*** [29] of $\mathbf{x}$ is defined as $\mathsf{H}(\mathbf{x}) = \max\{i = 1, \dots, n : x_{\{i\}} \ge i\}$ if $n \ge 1$ and $x_{\{1\}} \ge 1$, or $\mathsf{H}(\mathbf{x}) = 0$ otherwise. It may be shown that the h-index is a zero-insensitive OM3 aggeration operator, see [18], with:

$$\mathsf{H}(\mathbf{x}) = \bigvee_{i=1,\dots,n}^{n} i \wedge \lfloor x_{\{i\}} \rfloor.$$

Interpretation: "an author has h-index of H if H of his/her n most cited papers have at least H citations each, and the other $n-H$ papers are cited no more that H times each". The h-index may also be expressed as a Sugeno integral [38] w.r.t. to a counting measure, cf. [40] and [27].

agop implementation: `index_h()`.

```
index_h(c(6,5,4,2,1,0,0,0,0,0,0))
## [1] 3
index_h(c(-1,3,4,2)) # only for x>=0
## Error:  all elements in 'x' should be not less than 0
```

Moreover, we have $\mathsf{H}(\mathbf{x}) \le \min\{n, x_1\}$.

Note that the h-index was defined in original context (aggregation of citation counts) for integer vectors. More generally, it is better to use the OM3 operator with $\mathsf{w}_i(x) = x = \mathrm{Id}(x)$ and $c_{i,n} = i$ (two identity "objects" = one of the simplest setting). Interestingly, such aggregation operator is then asymptotically idempotent, i.e. for all $x \in \mathbb{I}$ we have $\lim_{n\to\infty} \mathsf{M}_{\triangle,\mathbf{w}}(n * x) = x$.

The g-index. Egghe's g-index [13] is defined as $\mathsf{G}(\mathbf{x}) = \max\{g = 1, \ldots, n : \sum_{i=1}^{g} x_{\{g\}} \ge g^2\}$, and is available in agop by calling `index_g()`. We have $\mathsf{G}(\mathbf{x}) \ge \mathsf{H}(\mathbf{x})$ with $\mathsf{G}(n{*}n) = \mathsf{H}(n{*}n) = n$

Note that this aggregation operator is not zero-insensitive, for example $\mathsf{G}(9,0) = 2$ and $\mathsf{G}(9,0,0) = 3$. Thus, we also provide the `index_g_zi()` function, which treats $\mathbf{x}$ as it would be padded with 0s.

```
index_g(9)
## [1] 1
index_g(c(9,0,0))
## [1] 3
index_g_zi(9)
## [1] 3
```

The index is interesting from the computational point of view – it may be calculated on the nondecreasing vector of cumulative sums, `cumsum(sort(x, decreasing=TRUE))`, however, it cannot directly be expressed as a symmetric maxitive aggregation operator.

However, it might be shown (see [27] for the proof) that if $\mathbf{x}$ is sorted nondecreasingly, then:

$$\mathsf{G}(\mathbf{x}) = \mathsf{H}(\mathbf{x})(0 \vee \mathtt{cummin}(\mathtt{cumsum}(x) - (1:n)^2 + (1:n))),$$

where $1:n = (1, 2, 3, \ldots, n)$.

The w-index. The w-index [44] is defined as

$$\mathsf{W}(\mathbf{x}) = \max\left\{w = 0, 1, 2, \ldots : x_{\{i\}} \ge w - i + 1, i = 1, \ldots, w\right\}$$

and is available in agop by calling `index_w()`.

Interestingly, we have shown in [27] that if $\mathbf{x}$ is sorted nondecreasingly, then:

$$\mathsf{W}(\mathbf{x}) = \mathsf{H}(\mathbf{x})(\mathtt{cummin}(\mathbf{x} + (1:n) - 1)).$$

Thus, it is easily seen that this is a zero-insensitive impact function. What is more we have $\mathsf{H}(\mathbf{x}) \le \mathsf{W}(\mathbf{x}) \le 2\mathsf{H}(\mathbf{x})$ and $\mathsf{W}(\mathbf{x}) \le \min\{n, x_1\}$.

The r_p-indices. The r_p-index, for $p \geqslant 1$ is expressed as

$$\mathsf{r}_p(\mathbf{x}) = \sup\{r > 0 : \mathsf{s}^{p,r} \trianglelefteq \mathbf{x}\},$$

where $\mathbf{s}^{p,r} = \left(\sqrt[p]{r^p - 0^p}, \sqrt[p]{r^p - 1^p}, \dots, \sqrt[p]{r^p - \lfloor r \rfloor^p}\right)$. For more details see [16, 23].

Please note that for integer vectors we have $r_1 = \mathsf{W}$ and $r_\infty = \mathsf{H}$ (cf. [23]). Hence it easily seen that, this is a zero-insensitive impact function.

agop implementation: `index_rp()`.

The l_p-indices. The l_p-index (cf. [16, 23]) for $p \in [1, \infty)$, $u > 0$ and $v > 0$ is a function $\mathsf{l}_p : \mathbb{I}^{1,2,\dots} \to \mathbb{I}^2$ given by the equation

$$\mathsf{l}_p(\mathbf{x}) = \arg\sup_{(u,v)} \{uv : \mathsf{e}^{p,u,v} \trianglelefteq \mathbf{x}\},$$

where $\mathsf{e}^{p,u,v} = \left(\sqrt[p]{v^p - (\frac{v}{u}0)^p}, \sqrt[p]{v^p - (\frac{v}{u}1)^p}, \dots, \sqrt[p]{v^p - (\frac{v}{u}\lfloor u \rfloor)^p}\right)$.

agop implementation: `index_lp()`.

The MAXPROD**-index.** The MAXPROD-index [31] is given by the equation

$$\mathsf{MP}(\mathbf{x}) = \max\left\{i \cdot x_{\{i\}} : i = 1, 2, \dots\right\}$$

is another example of zero-insensitive impact function. Interestingly, this index is a particular case of a projected l_∞-index, see [23], and can be also expressed in terms of Shilkret integral [37], see [27] for discussion.

In agop the MAXPROD-index is implemented in the `index_maxprod()` function.

Simple transformations of the h-index. Bibliometricians in many papers considered very simple, direct modifications of the h-index. For example, the $h(2)$-index [30] is defined as:

$$\mathsf{H2}(\mathbf{x}) = \max\left\{h = 0, 1, 2, \dots : x_h \geq h^2\right\}.$$

Some authors introduced other settings than "h^2" on the right side of (5.2), e.g. "$2h$", "αh" for some $\alpha > 0$, or "h^β", $\beta \geq 1$, cf. [2].

It may easily be shown that these reduce to the h-index for properly transformed input vectors, e.g. $\mathsf{H2}(\mathbf{x}) = \mathsf{H}(\sqrt{\mathbf{x}})$.

5.3 Interesting Dispersion Functions

...TO DO...

D2OWA (`d2owa()`):

$$\mathsf{D2OWA}_{\mathbf{w}}(\mathbf{x}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mathsf{OWA}_{\mathbf{w}}(\mathbf{x}))^2}$$

6 Aggregation Operators from the Probabilistic Perspective

By default, theory of aggregation looks at the aggregation operators mainly from the algebraic perspective. Of course, we may also be interested in their probabilistic properties, e.g. in i.i.d. RVs models (the simplest and the most "natural" ones in statistics), cf. [16] for discussion.

Intuitively, a random variable is a method for "producing" input data. An aggregation operator applied on a random variable (possibly multidimensional) is classically called a ***statistic***.

6.1 Some Notable Probability Distributions

Let $(X_1, \dots, X_n)$ i.i.d. F, where $\operatorname{supp} F = \mathbb{I}$. In social phenomena modeling, if F is continuous, we often assume that the underlying density f is decreasing and convex on $\mathbb{I}$, possibly with heavy-tails. E.g. in the bibliometric impact assessment problem, this assumption reflects the fact that higher paper valuations are more difficult to obtain than the lower ones, most of the papers have very small valuation (near 0), and the probability of attaining a high note decreases in at least linear pace.

Let us make a review of some useful statistical distributions, that are not available through "base" R (for other ones, e.g. exponential, normal, uniform, Weibull, etc. refer to the widely-available literature).

6.1.1 Pareto-Type II Distribution

Many generalizations of the Pareto distribution have been proposed (GPD, *Generalized Pareto Distributions*, cf. e.g. [41, 47]). Here we will introduce the so-called Pareto-Type II (Lomax) distribution, which has support $\mathbb{I} = [0, \infty]$ and is defined with two parameters.

Formally, X follows the Pareto-II distribution with shape parameter $k > 0$ and scale parameter $s > 0$, denoted $X \sim \mathrm{P2}(k, s)$, if its density is of the form

$$f(x) = \frac{k s^k}{(s + x)^{k+1}} \quad (x \geq 0). \tag{2}$$

The cumulative distribution function of X is then:

$$F(x) = 1 - \frac{s^k}{(s + x)^k} \quad (x \geq 0). \tag{3}$$

The Pareto-Type II distribution is implemented in agop: `dpareto2()` gives the p.d.f. (2), `ppareto2()` gives the c.d.f. (3), `qpareto2()` calculates the quantile function, F^{-1}, and `rpareto2()` generates random deviates.

Properties. The expected value of $X \sim \mathrm{P2}(k, s)$ exists for $k > 1$ and is equal to $\mathbb{E}X = \frac{s}{k-1}$. Variance exists for $k > 2$ and is equal to $\operatorname{Var} X = \frac{ks^2}{(k-2)(k-1)^2}$. More generally, the i-th raw moment for $k > i$ is given by: $\mathbb{E}X^i = \frac{\Gamma(i+1)\Gamma(k-i)}{\Gamma(k+1)} k s^i$.

For a fixed s, if $X \sim \mathrm{P2}(k_x, s)$ and $Y \sim \mathrm{P2}(k_y, s)$, $k_x < k_y$, then X stochastically dominates Y, denoted $X \succ Y$. On the other hand, for a fixed k, if $X \sim \mathrm{P2}(k, s_x)$ and $Y \sim \mathrm{P2}(k, s_y)$, then $s_x > s_y$ implies $X \succ Y$.

Most importantly, if $X \sim \mathrm{P2}(k, s)$, then the conditional distribution of $X - t$ given $X > t$, is $\mathrm{P2}(k, s + t)$ $t \geq 0$.

Additionally, it might be shown that if $X \sim \mathrm{P2}(k, s)$, then $\ln(s + X)$ has c.d.f. $F(x) = 1 - s^k e^{-kx}$ and density $f(x) = k s^k e^{-kx}$ for $x \geq \ln s$, i.e. has the same distribution as $Z + \ln s$, where $Z \sim \mathrm{Exp}(k) \equiv \Gamma(1, 1/k)$ (exponential distribution).

Parameter estimation. Let $\mathbf{x} = (x_1, \dots, x_n)$ be a realization of the Pareto-Type II i.i.d. sample with known $s > 0$. The unbiased (corrected) maximum likelihood estimator for k:

$$\widehat{k}(\mathbf{x}) = \frac{n - 1}{\sum_{i=1}^n \ln\left(1 + \frac{1}{s} x_i\right)}.$$

It may be shown that for $n > 2$ it holds $\operatorname{Var} \widehat{k}(\mathbf{x}) = k^2 \frac{1}{n-2}$.

agop implementation: `pareto2_estimate_mle()` with explicitly set argument `s`.

```
rowMeans(replicate(1000, {
   pareto2_estimate_mle(rpareto2(50, 2, 1.5), s=1.5)
}))
```

```
##        k        s
## 2.004464 1.500000
```

For both unknown k and s we have:

$$\begin{cases} \widehat{k} = \frac{n}{\sum_{i=1}^{n} \ln\left(1+x_i/\widehat{s}\right)}, \\ 1 + \frac{1}{n}\sum_{i=1}^{n} \ln\left(1+x_i/\widehat{s}\right) - \frac{n}{\sum_{i=1}^{n}\left(1+x_i/\widehat{s}\right)^{-1}} \quad = \quad 0. \end{cases}$$

Unfortunately, the second equation must be solved numerically. It is worth noting that the above system of equations may sometimes have no solution (as the local minimum of the likelihood function may not exist, see [10] for discussion). This estimator may be heavily biased and have a large mean squared error (of course, it is only asymptotically unbiased).

agop implementation: `pareto2_estimate_mle()` with explicitly set argument `s`.

```
rowMeans(replicate(1000, {
   pareto2_estimate_mle(rpareto2(50, 2, 1.5))
}), na.rm=TRUE)
```

```
##        k        s
## 2.747034 2.311929
```

We see that the estimator’s performance is weak.

A better (in general) estimation procedure was proposed in [48]. The Zhang-Stevens MMS (*minimum mean square error*) (Bayesian) estimator has relatively small bias (often positive) and mean squared error. In agop it is available as: `pareto2_estimate_mmse`.

```
rowMeans(replicate(1000, {
   pareto2_estimate_mmse(rpareto2(50, 2, 1.5))
}))
```

```
##        k        s
## 1.579770 1.224389
```

Goodness-of-fit tests. `pareto2_test_ad()` ..... – Anderson-Darling goodness-of-fit test (approximate p-value)..... (TO DO: describe) for known s by means of the `exp_test_ad()` function and the above-mentioned relationship between Pareto-Type II distributions and Exponential ones.

```
x <- rpareto2(100, k=1, s=2)
pareto2_test_ad(x, s=2)
```

```
##
##  Anderson-Darling goodness-of-fit test for Pareto Type-II
##  distribution
##
## data:  x
## W = 0.4831, p-value = 0.5168
```

Two-sample F-test. The following simple test was introduced in [16]. Let $(X_1, X_2, \ldots, X_{n_1})$ *i.i.d.* $\mathrm{P2}(k_1, s)$ and $(Y_1, Y_2, \ldots, Y_{n_2})$ *i.i.d.* $\mathrm{P2}(k_2, s)$, where s is an a-priori known scale parameter.

We are going to verify the null hypothesis $H_0 : k_1 = k_2$ against the two-sided alternative hypothesis $K : k_1 \neq k_2$.

It might be shown that $\sum_{i=1}^{n} \ln(s + X_i) - n \ln s \sim \Gamma(n, 1/k)$. This implies that under H_0, the following test statistic follows the Snedecor F distribution:

$$R(\mathbf{X}, \mathbf{Y}) = \frac{n_1}{n_2} \frac{\sum_{i=1}^{n_2} \ln\left(1 + \frac{Y_i}{s}\right)}{\sum_{i=1}^{n_1} \ln\left(1 + \frac{X_i}{s}\right)} \overset{H_0}{\sim} \mathrm{F}^{[2n_2, 2n_1]}. \tag{4}$$

The null hypothesis is accepted iff

$$R(\mathbf{x}, \mathbf{y}) \in \left[\mathtt{qf}(\tfrac{\alpha}{2}, 2n_2, 2n_1),\ \mathtt{qf}(1 - \tfrac{\alpha}{2}, 2n_2, 2n_1)\right],$$

where $\mathtt{qf}(q, d_1, d_2)$ denotes the q-quantile of $\mathrm{F}^{[d_1,d_2]}$

The p-value may be determined as follows:

$$p = 2\left(\tfrac{1}{2} - \left|\mathtt{pf}(R(\mathbf{x}, \mathbf{y}), 2n_2, 2n_1) - \tfrac{1}{2}\right|\right), \tag{5}$$

where $\mathtt{pf}(x, d_1, d_2)$ is the c.d.f. of $\mathrm{F}^{[d_1,d_2]}$.

agop implementation: `pareto2_test_f()`.

```
x <- rpareto2(35, 1.2, 1)
y <- rpareto2(25, 2.1, 1)
pareto2_test_f(x, y, s=1)

##
##  Two-sample F-test for equality of shape parameters for Type
##  II-Pareto distributions with known common scale parameter
##
## data:  x and y
## F = 0.3858, p-value = 0.000547
## alternative hypothesis: two-sided
```

6.1.2 Discretized Pareto-Type II Distribution

We would say that $X \sim \mathrm{DP2}(k, s)$, i.e. it follows the ***discretized Pareto-Type II distribution*** with shape parameter $k > 0$ and scale parameter $s > 0$, if $X = \lfloor Y \rfloor$, where $Y \sum \mathrm{P2}(k, s)$.

......TO BE DONE......

The Discretized Pareto-Type II distribution is implemented in agop: `ddpareto2()` gives the p.m.f., `pdpareto2()` gives the c.d.f., `qdpareto2()` calculates the quantile function, and `rdpareto2()` generates random deviates.

6.2 Stochastic Properties of Aggregation Operators

Given $(X_1, X_2, \dots)$ i.i.d. following a continuous c.d.f. F it is well-known, see [9], that L-statistics with weights $c_{i,n} = \mathsf{w}(i/n)$, for $\mathsf{w} : [0, 1] \to \mathbb{I}$, are asymptotically normally distributed. A similar result for the same weight setting has been shown for S-statistics, see [25].

For i.i.d samples of finite length we have e.g. the following result [20]:

Theorem 20. *Let $\mathbf{X} = (X_1, \dots, X_n)$ be a sequence of i.i.d. random variables with continuous c.d.f. F defined on $\mathbb{R}_{0+}$. Then the c.d.f. of $\mathsf{H}(\mathbf{X})$ for $x \in [0, n)$ is given by*

$$\mathsf{D}_n(x) = \mathcal{I}(\mathsf{F}(\lfloor x + 1 \rfloor^{-0}); n - \lfloor x \rfloor, \lfloor x \rfloor + 1),$$

where $\mathcal{I}(p; a, b)$ is the regularized incomplete beta function (`pbeta()` in R).

More generally, the c.d.f. of some quasi-S-statistics may be expressed as an incomplete beta function, see [25]. Note that, unlike in the case of the distribution of "ordinary" order statistics (see [9]), the parameters a, b of $\mathcal{I}$ are functions of x here.

7 NEWS/CHANGELOG

```
                          ** agop package NEWS **
***************************************************************************

0.2-0 /under development/

* The definition of owa(), owmax(), and owmin() is now consistent with
   that of (Grabisch et al., 2009), i.e. uses nondecreasing
   vectors, and not nonincreasing ones.

* NEW functions: pareto2_test_ad() and exp_test_ad(): (approximate)
   Anderson-Darling goodness-of-fit test for the Pareto-II (with known scale
   parameter) and exponential distribution, respectively.

* *PENDING* NEW disribution implemented: discretized Pareto-Type II
   (floor of Pareto Type-II):
   useful for modeling some phenomena in social sciences;
   see dpareto2_estimate_mle() for MLE estimators;
   and ddpareto2(), pdpareto2(), rdpareto2(), qdpareto2()
   for standard d.f. implementations.

* NEW functions: pord_spread() and pord_spreadsym() --
   to compare spread of two equal-length numeric vectors
   (see Gagolewski M., Dispersion Operators, submitted, 2013).

* NEW function: d2owa(): a symmetric dispersion operator defined as
   the L2 distance between a numeric vector and an OWA operator.

* BUGFIX: rel_closure_transitive() - resulting matrix
   was not necessarily transitive; the function has been rewritten in C++
   and now uses the Warshall algorithm (1962).

* *** Possibly TO DO: remove dependency Matrix and igraph ***

* All functions dealing with binary relations now have "rel_" prefix.
  Moreover, de_transitive() has been renamed to rel_reduction_hasse().

* Rewritten in C++: ppareto2(),
   rel_is_reflexive(), rel_is_total(), rel_is_transitive(),
   rel_closure_total_fair(), rel_closure_transitive(),
   rel_reduction_transitive(), rel_reduction_hasse().

* New functions dealing with binary relations:
   rel_closure_reflexive(), rel_reduction_reflexive(), rel_is_symmetric(),
   rel_closure_symmetric(), rel_is_irreflexive(), rel_is_asymmetric(),
   rel_is_antisymmetric(), rel_is_cyclic(), ....
```

```
   Note that in future releases of agop we plan to add
   procedures that deal with fuzzy relations.

* BUGFIX: prepare_arg_numeric_sorted (internal, C++) did not sort some vectors.

* all built-in agops now throw an error on empty vectors.

* moved testthat tests and vignettes to the agop github repository
      devel/ dir; `testthat` Suggest-type dependency removed.

* ...TO DO...

****************************************************************************

0.1-3 /2013-06-27/

* Compiler errors on Solaris Studio removed.

****************************************************************************

0.1-1 /2013-06-26/

* initial release
  [the package started as a lightweight fork of the CITAN package].

****************************************************************************
```

Acknowledgments. This document has been generated with LaTeX, knitr and the tikzDevice package for R. Their authors' wonderful work is fully appreciated.

The contribution of Marek Gagolewski 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 (March-June 2013), and by the FNP START Scholarship from the Foundation for Polish Science (2013).

Bibliography

[1] Aho, A., Garey, M., and Ullman, J. The transitive reduction of a directed graph. *SIAM Journal on Computing 1*, 2 (1972), 131–137.

[2] Alonso, S., Cabrerizo, F. J., Herrera-Viedma, E., and Herrera, F. *h*-index: A review focused on its variants, computation and standardization for different scientific fields. *Journal of Informetrics 3* (2009), 273–289.

[3] Beliakov, G., and James, S. Stability of weighted penalty-based aggregation functions. *Fuzzy Sets and Systems 226*, 1 (2013), 1–18.

[4] Beliakov, G., Pradera, A., and Calvo, T. *Aggregation functions: A guide for practitioners.* Springer-Verlag, 2007.

[5] Calvo, T., and Mayor, G. Remarks on two types of extended aggregation functions. *Tatra Mountains Mathematical Publications 16* (1999), 235–253.

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

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

[8] Choquet, G. Theory of capacities. *Annales de l'institut Fourier 5* (1954), 131–295.

[9] David, H. A., and Nagaraja, H. N. *Order statistics*. Wiley, 2003.

[10] del Castillo, J., and Daoudi, J. Estimation of the Generalized Pareto Distribution. *Statistics and Probability Letters 79* (2009), 684–688.

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

[12] Dubois, D., Prade, H., and Testemale, C. Weighted fuzzy pattern matching. *Fuzzy Sets and Systems 28* (1988), 313–331.

[13] Egghe, L. An improvement of the h-index: the g-index. *ISSI Newsletter 2*, 1 (2006), 8–9.

[14] Franceschini, F., and Maisano, D. A. The Hirsch index in manufacturing and quality engineering. *Quality and Reliability Engineering International 25* (2009), 987–995.

[15] Franceschini, F., and Maisano, D. A. Structured evaluation of the scientific output of academic research groups by recent h-based indicators. *Journal of Informetrics 5* (2011), 64–74.

[16] Gagolewski, M. *Aggregation operators and their application in a formal model for quality evaluation system of scientific research (Wybrane operatory agregacji i ich zastosowanie w modelu formalnym systemu jakości w nauce)*. PhD thesis, Systems Research Institute, Polish Academy of Sciences, 2011. (In Polish).

[17] Gagolewski, M. Bibliometric impact assessment with R and the CITAN package. *Journal of Informetrics 5*, 4 (2011), 678–692.

[18] Gagolewski, M. On the relationship between symmetric maxitive, minitive, and modular aggregation operators. *Information Sciences 221* (2013), 170–180.

[19] Gagolewski, M. Scientific impact assessment cannot be fair. *Journal of Informetrics 7*, 4 (2013), 792–802.

[20] Gagolewski, M. Statistical hypothesis test for the difference between Hirsch indices of two Pareto-distributed random samples. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, R. Kruse et al., Eds., vol. 190. Springer-Verlag, 2013, pp. 359–367.

[21] Gagolewski, M. An axiomatization of tools measuring variability, spread, or scatter of numeric sequences, their relation to aggregation functions, and applications in decision making, 2014. Submitted paper.

[22] Gagolewski, M., and Cena, A. *agop: Aggregation operators and preordered sets in R*, 2014. http://agop.rexamine.com.

[23] Gagolewski, M., and Grzegorzewski, P. A geometric approach to the construction of scientific impact indices. *Scientometrics 81*, 3 (2009), 617–634.

[24] Gagolewski, M., and Grzegorzewski, P. Arity-monotonic extended aggregation operators. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, E. Hüllermeier et al., Eds., vol. 80. Springer-Verlag, 2010, pp. 693–702.

[25] Gagolewski, M., and Grzegorzewski, P. S-statistics and their basic properties. In *Combining Soft Computing and Statistical Methods in Data Analysis*, C. Borgelt et al., Eds. Springer-Verlag, 2010, pp. 281–288.

[26] Gagolewski, M., and 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), 1312–1324.

[27] Gagolewski, M., and Mesiar, R. Monotone measures and universal integrals in a uniform framework for the scientific impact assessment problem. *Information Sciences 263* (2014), 166–174.

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

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

[30] Kosmulski, M. A new Hirsch-type index saves time and works equally well as the original *h*-index. *ISSI Newsletter 2*, 3 (2006), 4–6.

[31] 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), .

[32] Mesiar, R., and Mesiarová-Zemánková, A. The ordered modular averages. *IEEE Transactions on Fuzzy Systems 19*, 1 (2011), 42–50.

[33] Meyer, D., and Hornik, K. *relations: Data Structures and Algorithms for Relations*, 2013. R package version 0.6-2.

[34] Quesada, A. Monotonicity and the Hirsch index. *Journal of Informetrics 3*, 2 (2009), 158–160.

[35] Quesada, A. More axiomatics for the Hirsch index. *Scientometrics 82* (2010), 413–418.

[36] Rousseau, R. Woeginger's axiomatisation of the *h*-index and its relation to the *g*-index, the $h(2)$-index and the r^2-index. *Journal of Informetrics 2*, 4 (2008), 335–340.

[37] Shilkret, N. Maxitive measure and integration. *Indagationes Mathematicæ33* (1971), 109–116.

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

[39] R Development Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. http://www.R-project.org.

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

[41] Villasenor-Alva, J., and Gonzalez-Estrada, E. A bootstrap goodness of fit test for the Generalized Pareto Distribution. *Computational Statistics and Data Analysis 53*, 11 (2009), 3835–3841.

[42] Warshall, S. A theorem on boolean matrices. *Journal of the ACM 9*, 1 (1962), 11–12.

[43] Woeginger, G. J. An axiomatic analysis of Egghe's *g*-index. *Journal of Informetrics 2*, 4 (2008), 364–368.

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

[45] Woeginger, G. J. A symmetry axiom for scientific impact indices. *Journal of Informetrics 2* (2008), 298–303.

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

[47] Zhang, J. Improving on estimation for the Generalized Pareto Distribution. *Technometrics 52*, 3 (2010), 335–339.

[48] Zhang, J., and Stephens, M. A. A new and efficient estimation method for the Generalized Pareto Distribution. *Technometrics 51*, 3 (2009), 316–325.

Index