Package 'FuzzyNumbers'

November 15, 2021

Title Tools to Deal with Fuzzy Numbers

Type Package

Description S4 classes and methods
to deal with fuzzy numbers. They allow for computing any arithmetic operations (e.g., by using the Zadeh extension principle), performing approximation of arbitrary fuzzy numbers by trapezoidal and piecewise linear ones, preparing plots for publications, computing possibility and necessity values for comparisons, etc.

Version 0.4-7

Date 2021-11-15

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

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

License LGPL (>= 3)

ByteCompile TRUE

VignetteBuilder knitr

Suggests knitr, digest

Depends R (>= 3.0.0), methods, grDevices, graphics, stats

RoxygenNote 7.1.2

Encoding UTF-8

NeedsCompilation no

Author Marek Gagolewski [aut, cre] (<https://orcid.org/0000-0003-0637-6028>),
Jan Caha [ctb]

Maintainer Marek Gagolewski <marek@gagolewski.com>

Repository CRAN

Date/Publication 2021-11-15 12:10:32 UTC

R topics documented:

FuzzyNumbers-package . . . 3
alphacut . . . 5
alphaInterval . . . 6
ambiguity . . . 7
approxInvert . . . 8
arctan2 . . . 8
Arithmetic . . . 9
as.character . . . 12
as.FuzzyNumber . . . 13
as.PiecewiseLinearFuzzyNumber . . . 14
as.PowerFuzzyNumber . . . 15
as.TrapezoidalFuzzyNumber . . . 16
convertAlpha . . . 17
convertSide . . . 18
core . . . 19
DiscontinuousFuzzyNumber . . . 19
DiscontinuousFuzzyNumber-class . . . 21
distance . . . 22
evaluate . . . 23
expectedInterval . . . 24
expectedValue . . . 25
Extract . . . 26
fapply . . . 27
FuzzyNumber . . . 28
FuzzyNumber-class . . . 29
integrateAlpha . . . 30
integrate_discont_val . . . 31
maximum . . . 32
minimum . . . 33
necessityExceedance . . . 35
necessityStrictExceedance . . . 36
necessityStrictUndervaluation . . . 37
necessityUndervaluation . . . 38
piecewiseLinearApproximation . . . 39
PiecewiseLinearFuzzyNumber . . . 40
PiecewiseLinearFuzzyNumber-class . . . 41
plot . . . 42
possibilityExceedance . . . 45
possibilityStrictExceedance . . . 46
possibilityStrictUndervaluation . . . 47
possibilityUndervaluation . . . 48
PowerFuzzyNumber . . . 49
PowerFuzzyNumber-class . . . 49
show . . . 50
supp . . . 51
trapezoidalApproximation . . . 52

TrapezoidalFuzzyNumber 53
TrapezoidalFuzzyNumber-class 54
TriangularFuzzyNumber 55
value 56
weightedExpectedValue 57
width 58
^,PiecewiseLinearFuzzyNumber,numeric-method 59

Index **60**

`FuzzyNumbers-package` *Tools to Deal with Fuzzy Numbers*

Description

FuzzyNumbers is an open source (LGPL 3) package for R. It provides S4 classes and methods to deal with fuzzy numbers. The package may be used by researchers in fuzzy numbers theory (e.g., for testing new algorithms, generating numerical examples, preparing figures).

Details

Fuzzy set theory gives one of many ways (in particular, see Bayesian probabilities) to represent imprecise information. Fuzzy numbers form a particular subclass of fuzzy sets of the real line. The main idea behind this concept is motivated by the observation that people tend to describe their knowledge about objects through vague numbers, e.g., "I'm about 180 cm tall" or "The event happened between 2 and 3 p.m.".

For the formal definition of a fuzzy number please refer to the `FuzzyNumber` man page. Note that this package also deals with particular types of fuzzy numbers like trapezoidal, piecewise linear, or "parametric" FNs (see `TrapezoidalFuzzyNumber` `PiecewiseLinearFuzzyNumber`, `PowerFuzzyNumber`, and *EXPERIMENTAL* `DiscontinuousFuzzyNumber`)

The package aims to provide the following functionality:

1. Representation of arbitrary fuzzy numbers (including FNs with discontinuous side functions and/or alpha-cuts), as well as their particular types, e.g. trapezoidal and piecewise linear fuzzy numbers,
2. Defuzzification and approximation by triangular and piecewise linear FNs (see e.g. `expectedValue`, `value`, `trapezoidalApproximation`, `piecewiseLinearApproximation`),
3. Visualization of FNs (see `plot`, `as.character`),
4. Basic operations on FNs (see e.g. `fapply` and Arithmetic),
5. etc.

For a complete list of classes and methods call `help(package="FuzzyNumbers")`. Moreover, you will surely be interested in a step-by-step guide to the package usage and features which is available at the project's webpage.

Keywords: Fuzzy Numbers, Fuzzy Sets, Shadowed Sets, Trapezoidal Approximation, Piecewise Linear Approximation, Approximate Reasoning, Imprecision, Vagueness, Randomness.

Acknowledgments: Many thanks to Jan Caha, Przemyslaw Grzegorzewski, Lucian Coroianu, and Pablo Villacorta Iglesias for stimulating discussion.

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.

Author(s)

Marek Gagolewski, with contributions from Jan Caha

References

Ban A.I. (2008), Approximation of fuzzy numbers by trapezoidal fuzzy numbers preserving the expected interval, Fuzzy Sets and Systems 159, pp. 1327-1344.

Ban A.I. (2009), On the nearest parametric approximation of a fuzzy number - Revisited, Fuzzy Sets and Systems 160, pp. 3027-3047.

Bodjanova S. (2005), Median value and median interval of a fuzzy number, Information Sciences 172, pp. 73-89.

Chanas S. (2001), On the interval approximation of a fuzzy number, Fuzzy Sets and Systems 122, pp. 353-356.

Coroianu L., Gagolewski M., Grzegorzewski P. (2013), Nearest Piecewise Linear Approximation of Fuzzy Numbers, Fuzzy Sets and Systems 233, pp. 26-51.

Coroianu L., Gagolewski M., Grzegorzewski P., Adabitabar Firozja M., Houlari T. (2014), Piecewise linear approximation of fuzzy numbers preserving the support and core, In: Laurent A. et al. (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II (CCIS 443), Springer, pp. 244-254.

Delgado M., Vila M.A., Voxman W. (1998), On a canonical representation of a fuzzy number, Fuzzy Sets and Systems 93, pp. 125-135.

Dubois D., Prade H. (1978), Operations on fuzzy numbers, Int. J. Syst. Sci. 9, pp. 613-626.

Dubois D., Prade H. (1987a), The mean value of a fuzzy number, Fuzzy Sets and Systems 24, pp. 279-300.

Dubois D., Prade H. (1987b), Fuzzy numbers: An overview, In: Analysis of Fuzzy Information. Mathematical Logic, vol. I, CRC Press, pp. 3-39.

Grzegorzewski P. (2010), Algorithms for trapezoidal approximations of fuzzy numbers preserving the expected interval, In: Bouchon-Meunier B. et al. (Eds.), Foundations of Reasoning Under Uncertainty, Springer, pp. 85-98.

Grzegorzewski P. (1998), Metrics and orders in space of fuzzy numbers, Fuzzy Sets and Systems 97, pp. 83-94.

Grzegorzewski P,. Pasternak-Winiarska K. (2011), Trapezoidal approximations of fuzzy numbers with restrictions on the support and core, Proc. EUSFLAT/LFA 2011, Atlantis Press, pp. 749-756.

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

Stefanini L., Sorini L. (2009), Fuzzy arithmetic with parametric LR fuzzy numbers, In: Proc. IFSA/EUSFLAT 2009, pp. 600-605.

Yeh C.-T. (2008), Trapezoidal and triangular approximations preserving the expected interval, Fuzzy Sets and Systems 159, pp. 1345-1353.

`alphacut` *Compute Alpha-Cuts*

Description

If A is a fuzzy number, then its α-cuts are always in form of intervals. Moreover, the α-cuts form a nonincreasing chain w.r.t. $alpha$.

Usage

```
## S4 method for signature 'FuzzyNumber,numeric'
alphacut(object, alpha)
```

Arguments

`object`	a fuzzy number
`alpha`	numeric vector with elements in [0,1]

Value

Returns a matrix with two columns (left and right alha cut bounds). if some elements in `alpha` are not in [0,1], then `NA` is set.

See Also

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `weightedExpectedValue()`, `width()`

Other alpha_cuts: `core()`, `supp()`

Examples

```
A <- TrapezoidalFuzzyNumber(1, 2, 3, 4)
alphacut(A, c(-1, 0.4, 0.2))
```

alphaInterval *Compute the Alpha-Interval of a Fuzzy Number*

Description

We have $\alpha - Int(A) := [\int_0^1 \alpha A_L(\alpha)\, d\alpha, \int_0^1 \alpha A_U(\alpha)\, d\alpha]$.

Usage

```
## S4 method for signature 'FuzzyNumber'
alphaInterval(object, ...)

## S4 method for signature 'TrapezoidalFuzzyNumber'
alphaInterval(object)

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
alphaInterval(object)

## S4 method for signature 'PowerFuzzyNumber'
alphaInterval(object)
```

Arguments

object	a fuzzy number
...	for FuzzyNumber and DiscontinuousFuzzyNumber - additional arguments passed to integrateAlpha

Details

Note that if an instance of the FuzzyNumber or DiscontinuousFuzzyNumber class is given, the calculation is performed via numerical integration. Otherwise, the computation is exact.

Value

Returns numeric vector of length 2.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNu
expectedInterval(), plot()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, PowerFuzzyNumber, as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval()

`ambiguity` *Calculate the Ambiguity of a Fuzzy Number*

Description

The ambiguity (Delgado et al, 1998) is a measure of nonspecificity of a fuzzy number.

Usage

```
## S4 method for signature 'FuzzyNumber'
ambiguity(object, ...)
```

Arguments

`object`	a fuzzy number
`...`	additional arguments passed to `alphaInterval`

Details

The ambiguity is defined as $amb(A) := \int_0^1 \alpha \left(A_U(\alpha) - A_L(\alpha)\right)\, d\alpha$.

Value

Returns a single numeric value.

References

Delgado M., Vila M.A., Voxman W. (1998), On a canonical representation of a fuzzy number, Fuzzy Sets and Systems 93, pp. 125-135.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other characteristics: expectedValue(), value(), weightedExpectedValue(), width()

approxInvert *Approximate the Inverse of a Given Function*

Description

The function may be used to create side generating functions from alpha-cut generators and inversely.

Usage

```
approxInvert(f, method = c("monoH.FC", "linear", "hyman"), n = 500)
```

Arguments

f	a monotonic, continuous function f: [0,1]->[0,1]
method	interpolation method: "monoH.FC', "hyman" or "linear"
n	number of interpolation points

Details

The function is a wrapper to splinefun and approxfun. Thus, interpolation is used.

Value

Returns a new function, the approximate inverse of the input.

See Also

FuzzyNumber

Other auxiliary: convertAlpha(), convertSide()

arctan2 *Arc-tangent*

Description

The arc-tangent of two arguments arctan2(y, x) returns the angle between the x-axis and the vector from the origin to (x, y) for PiecewiseLinearFuzzyNumbers.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
arctan2(y, x)
```

Arguments

y	a PiecewiseLinearFuzzyNumber
x	a PiecewiseLinearFuzzyNumber

Details

Note that resulting values are no longer from interval [-pi,pi] but [-1.5pi,pi], in order to provide valid fuzzy numbers as result.

Value

Returns a fuzzy number of the class PiecewiseLinearFuzzyNumber indicating the angle specified by the input fuzzy numbers. The range of results is [-1.5pi,pi].

See Also

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
y = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-2, 3, 5), knot.n = 9)
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-4.8, -4, 1.5), knot.n = 9)
arctan2(y,x)
```

Arithmetic *Arithmetic Operations on Fuzzy Numbers*

Description

Applies arithmetic operations using the extension principle and interval-based calculations.

Usage

```
## S4 method for signature 'numeric,FuzzyNumber'
e1 + e2 # e2 + e1

## S4 method for signature 'TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber'
e1 + e2

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

```
e1 + e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
e1 + e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
e1 + e2 # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'numeric,FuzzyNumber'
e1 - e2 # e2*(-1) + e1

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

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

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
e1 - e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
e1 - e2 # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'FuzzyNumber,ANY'
e1 - e2 # -e1

## S4 method for signature 'numeric,FuzzyNumber'
e1 * e2 # e2 * e1

## S4 method for signature 'TrapezoidalFuzzyNumber,numeric'
e1 * e2

## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
e1 * e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
e1 * e2 # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
e1 * e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
e1 / e2

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

```
e1 / e2

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
e1 / e2 # calls as.PiecewiseLinearFuzzyNumber()
```

Arguments

e1	a fuzzy number or single numeric value
e2	a fuzzy number or single numeric value

Details

Implemented operators: +, -, *, / for piecewise linear fuzzy numbers. Also some versions may be applied on numeric values and trapezoidal fuzzy numbers.

Note that according to the theory the class of PLFNs is not closed under the operations * and /. However, if you operate on a large number of knots, the results should be satisfactory.

Thanks to Jan Caha for suggestions on PLFN operations.

Value

Returns a fuzzy number of the class PiecewiseLinearFuzzyNumber or TrapezoidalFuzzyNumber.

See Also

Other FuzzyNumber-method: Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other PiecewiseLinearFuzzyNumber-method: Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other TrapezoidalFuzzyNumber-method: TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval(), plot()

Other extension_principle: ^,PiecewiseLinearFuzzyNumber,numeric-method, fapply()

as.character *Get Basic Information on a Fuzzy Number in a String*

Description

This method is especially useful if you would like to generate LaTeX equations defining a fuzzy numbers.

Usage

```
## S4 method for signature 'FuzzyNumber'
as.character(x, toLaTeX=FALSE, varnameLaTeX="A")

## S4 method for signature 'TrapezoidalFuzzyNumber'
as.character(x, toLaTeX=FALSE, varnameLaTeX="A")

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
as.character(x, toLaTeX=FALSE, varnameLaTeX="A")

## S4 method for signature 'PowerFuzzyNumber'
as.character(x, toLaTeX=FALSE, varnameLaTeX="A")
```

Arguments

x	a fuzzy number
toLaTeX	logical; should LaTeX code be output?
varnameLaTeX	character; variable name to be included in equations

Details

Consider calling the cat function on the resulting string.

Thanks to Jan Caha for suggesting the toLaTeX arg.

Value

Returns a character vector.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(),

arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval()

Other conversion: as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber()

`as.FuzzyNumber` *Converts an Object to a Fuzzy Number*

Description

Please note that applying this function on a FuzzyNumber child class causes information loss, as it drops all additional slots defined in the child classes. FuzzyNumber is the base class for all FNs. Note that some functions for TFNs or PLFNs work much faster and are more precise. This function shouldn't be used in normal computations.

Usage

```
## S4 method for signature 'numeric'
as.FuzzyNumber(object)

## S4 method for signature 'FuzzyNumber'
as.FuzzyNumber(object)
```

Arguments

`object`	a fuzzy number or a single numeric value (crisp number) or vector of length two (interval)

Value

Returns an bject of class FuzzyNumber.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumb as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other conversion: as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character()

`as.PiecewiseLinearFuzzyNumber` *Converts an Object to a Piecewise Linear Fuzzy Number*

Description

This method is only for exact conversion. For other cases (e.g. general FNs), use `piecewiseLinearApproximation`.

Usage

```
## S4 method for signature 'TrapezoidalFuzzyNumber'
as.PiecewiseLinearFuzzyNumber(object, knot.n=0,
   knot.alpha=seq(0, 1, length.out=knot.n+2)[-c(1,knot.n+2)])

## S4 method for signature 'numeric'
as.PiecewiseLinearFuzzyNumber(object, knot.n=0,
   knot.alpha=seq(0, 1, length.out=knot.n+2)[-c(1,knot.n+2)])

## S4 method for signature 'FuzzyNumber'
as.PiecewiseLinearFuzzyNumber(object, knot.n=0,
   knot.alpha=seq(0, 1, length.out=knot.n+2)[-c(1,knot.n+2)])

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
as.PiecewiseLinearFuzzyNumber(object, knot.n=0,
   knot.alpha=seq(0, 1, length.out=knot.n+2)[-c(1,knot.n+2)])
```

Arguments

`object`	a fuzzy number or a single numeric value (crisp number) or vector of length two (interval)
`knot.n`	the number of knots
`knot.alpha`	`knot.n` alpha-cut values at knots, defaults to uniformly distributed knots

Value

Returns an object of class `PiecewiseLinearFuzzyNumber`.

See Also

Other TrapezoidalFuzzyNumber-method: `Arithmetic`, `TrapezoidalFuzzyNumber-class`, `TrapezoidalFuzzyNumber`, `TriangularFuzzyNumber()`, `alphaInterval()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `expectedInterval()`, `plot()`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `^,PiecewiseLinearFuzzyNumber,numeric-method`, `alphaInterval()`, `arctan2()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `minimum()`, `necessityExceedance()`, `necessityStrictExceedance()`,

necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other conversion: as.FuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character()

`as.PowerFuzzyNumber` *Converts an Object to a Power Fuzzy Number*

Description

This method is only for exact conversion.

Usage

```
## S4 method for signature 'numeric'
as.PowerFuzzyNumber(object)

## S4 method for signature 'FuzzyNumber'
as.PowerFuzzyNumber(object)

## S4 method for signature 'PowerFuzzyNumber'
as.PowerFuzzyNumber(object)

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
as.PowerFuzzyNumber(object)

## S4 method for signature 'TrapezoidalFuzzyNumber'
as.PowerFuzzyNumber(object)
```

Arguments

`object`	a fuzzy number or a single numeric value (crisp number) or vector of length two (interval)

Value

Returns an object of class PowerFuzzyNumber.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval(), plot()

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other conversion: as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character()

---

as.TrapezoidalFuzzyNumber

Converts an Object to a Trapezoidal Fuzzy Number

Description

This method is only for exact conversion. For other cases (e.g. general FNs), use trapezoidalApproximation.

Usage

```
## S4 method for signature 'numeric'
as.TrapezoidalFuzzyNumber(object)

## S4 method for signature 'FuzzyNumber'
as.TrapezoidalFuzzyNumber(object)

## S4 method for signature 'PowerFuzzyNumber'
as.TrapezoidalFuzzyNumber(object)

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
as.TrapezoidalFuzzyNumber(object)

## S4 method for signature 'TrapezoidalFuzzyNumber'
as.TrapezoidalFuzzyNumber(object)
```

Arguments

object	a fuzzy number or a single numeric value (crisp number) or vector of length two (interval)

Value

Returns an bject of class TrapezoidalFuzzyNumber.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), expectedInterval(), plot()

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval(), as.PowerFuzzyNumber(), as.character(), expectedInterval()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other conversion: as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.character()

---

convertAlpha *Convert a Given Upper/Lower Alpha-Cut Function to an Alpha-Cut Generating Function*

Description

The resulting function calls the original function and then linearly scales its output.

Usage

```
convertAlpha(f, y1, y2)
```

Arguments

f	a function into [y1,y2]
y1	numeric vector of length 1
y2	numeric vector of length 1

Value

Returns a new function defined on [0,1] (scaled input).

See Also

FuzzyNumber

Other auxiliary: approxInvert(), convertSide()

---

convertSide *Convert a Given Side Function to Side Generating Function*

Description

The resulting function linearly scales the input and passes it to the original function.

Usage

```
convertSide(f, x1, x2)
```

Arguments

f	a function defined on [x1,x2]
x1	numeric vector of length 1; if longer, only the first element is used
x2	numeric vector of length 1; if longer, only the first element is used

Details

The function works for x1<x2 and x1>x2.

Value

Returns a new function defined on [0,1] (scaled input).

See Also

FuzzyNumber

Other auxiliary: approxInvert(), convertAlpha()

core *Calculate the Core of a Fuzzy Number*

Description

We have $\text{core}(A) := [a2, a3]$. This gives the values that a fuzzy number necessarily represents.

Usage

```
## S4 method for signature 'FuzzyNumber'
core(object)
```

Arguments

object	a fuzzy number

Value

Returns a numeric vector of length 2.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other alpha_cuts: alphacut(), supp()

DiscontinuousFuzzyNumber *Creates a Fuzzy Number with Possibly Discontinuous Side Functions or Alpha-Cut Bounds*

Description

For convenience, objects of class DiscontinuousFuzzyNumber may be created with this function.

Usage

```
DiscontinuousFuzzyNumber(
  a1,
  a2,
  a3,
  a4,
  lower = function(a) rep(NA_real_, length(a)),
  upper = function(a) rep(NA_real_, length(a)),
  left = function(x) rep(NA_real_, length(x)),
  right = function(x) rep(NA_real_, length(x)),
  discontinuities.left = numeric(0),
  discontinuities.right = numeric(0),
  discontinuities.lower = numeric(0),
  discontinuities.upper = numeric(0)
)
```

Arguments

a1	a number specyfing left bound of the support
a2	a number specyfing left bound of the core
a3	a number specyfing right bound of the core
a4	a number specyfing right bound of the support
lower	lower alpha-cut bound generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
upper	upper alpha-cut bound generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_
left	lower side function generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
right	upper side function generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_
discontinuities.left	nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
discontinuities.right	nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
discontinuities.lower	nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
discontinuities.upper	nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0

Value

Object of class DiscontinuousFuzzyNumber

See Also

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber-class, Extract, distance(), integrateAlpha(), plot()

DiscontinuousFuzzyNumber-class ***EXPERIMENTAL** S4 Class Representing a Fuzzy Number with Discontinuous Side Functions or Alpha-Cut Bounds*

Description

Discontinuity information increase the precision of some numerical integration-based algorithms, e.g. of piecewiseLinearApproximation. It also allows for making more valid fuzzy number plots.

Slots

a1, a2, a3, a4, lower, upper, left, right: Inherited from the FuzzyNumber class.

discontinuities.left: nondecreasingly sorted numeric vector with elements in (0,1); discontinuity points for the left side generating function

discontinuities.right: nondecreasingly sorted numeric vector with elements in (0,1); discontinuity points for the right side generating function

discontinuities.lower: nondecreasingly sorted numeric vector with elements in (0,1); discontinuity points for the lower alpha-cut bound generator

discontinuities.upper: nondecreasingly sorted numeric vector with elements in (0,1); discontinuity points for the upper alpha-cut bound generator

Extends

Class FuzzyNumber, directly.

See Also

DiscontinuousFuzzyNumber for a convenient constructor

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber, Extract, distance(), integrateAlpha(), plot()

Examples

```
showClass("DiscontinuousFuzzyNumber")
showMethods(classes="DiscontinuousFuzzyNumber")
```

distance *Calculate the Distance Between Two Fuzzy Numbers*

Description

Currently, only Euclidean distance may be calculated. We have $d_E^2(A,B) := \int_0^1 (A_L(\alpha) - B_L(\alpha))^2 \, d\alpha, \int_0^1 + (A_U(\alpha) - B_U(\alpha))^2 \, d\alpha$, see (Grzegorzewski, 1988).

Usage

```
## S4 method for signature 'FuzzyNumber,FuzzyNumber'
distance(e1, e2, type=c("Euclidean", "EuclideanSquared"), ...)

## S4 method for signature 'FuzzyNumber,DiscontinuousFuzzyNumber'
distance(e1, e2, type=c("Euclidean", "EuclideanSquared"), ...)

## S4 method for signature 'DiscontinuousFuzzyNumber,FuzzyNumber'
distance(e1, e2, type=c("Euclidean", "EuclideanSquared"), ...)

## S4 method for signature 'DiscontinuousFuzzyNumber,DiscontinuousFuzzyNumber'
distance(e1, e2, type=c("Euclidean", "EuclideanSquared"), ...)
```

Arguments

e1	a fuzzy number
e2	a fuzzy number
...	additional arguments passed to integrate
type	one of "Euclidean", "EuclideanSquared"

Details

The calculation are done using numerical integration,

Value

Returns the calculated distance, i.e. a single numeric value.

References

Grzegorzewski P., Metrics and orders in space of fuzzy numbers, Fuzzy Sets and Systems 97, 1998, pp. 83-94.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber-class, DiscontinuousFuzzyNumber, Extract, integrateAlpha(), plot()

---

evaluate *Evaluate the Membership Function*

Description

This function returns the value(s) of the membership function of a fuzzy number at given point(s).

Usage

```
## S4 method for signature 'FuzzyNumber,numeric'
evaluate(object, x)
```

Arguments

object	a fuzzy numbers
x	numeric vector

Value

Returns a numeric vector.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Examples

```
T <- TrapezoidalFuzzyNumber(1,2,3,4)
evaluate(T, seq(0, 5, by=0.5))
```

`expectedInterval` *Calculate the Expected Interval of a Fuzzy Number*

Description

We have $EI(A) := [\int_0^1 A_L(\alpha)\, d\alpha, \int_0^1 A_U(\alpha)\, d\alpha]$, see (Duboid, Prade, 1987).

Usage

```
## S4 method for signature 'FuzzyNumber'
expectedInterval(object, ...)

## S4 method for signature 'TrapezoidalFuzzyNumber'
expectedInterval(object)

## S4 method for signature 'PiecewiseLinearFuzzyNumber'
expectedInterval(object)

## S4 method for signature 'PowerFuzzyNumber'
expectedInterval(object)
```

Arguments

`object`	a fuzzy number
`...`	for `FuzzyNumber` and `DiscontinuousFuzzyNumber` - additional arguments passed to `integrateAlpha`

Details

Note that if an instance of the `FuzzyNumber` or `DiscontinuousFuzzyNumber` class is given, the calculation is performed via numerical integration. Otherwise, the computation is exact.

Value

Returns a numeric vector of length 2.

References

Dubois D., Prade H. (1987), The mean value of a fuzzy number, Fuzzy Sets and Systems 24, pp. 279-300.

See Also

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `weightedExpectedValue()`, `width()`

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), plot()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character()

`expectedValue` *Calculate the Expected Value of a Fuzzy Number*

Description

The calculation of the so-called expected value is one of possible methods to deffuzify a fuzzy number.

Usage

```
## S4 method for signature 'FuzzyNumber'
expectedValue(object, ...)
```

Arguments

`object`	a fuzzy number
`...`	additional arguments passed to `expectedInterval`

Details

The expected value of A is defined as $EV(A) := (EI_U(A) + EI_L(A))/2$, where EI is the `expectedInterval`.

Value

Returns a single numeric value.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other deffuzification: value(), weightedExpectedValue()

Other characteristics: ambiguity(), value(), weightedExpectedValue(), width()

Extract *FuzzyNumber Slot Accessors*

Description

For possible slot names see man pages for the FuzzyNumber class and its derivatives

Usage

```
## S4 method for signature 'FuzzyNumber,character'
x[i]

## S4 method for signature 'PiecewiseLinearFuzzyNumber,character'
x[i]

## S4 method for signature 'PowerFuzzyNumber,character'
x[i]

## S4 method for signature 'DiscontinuousFuzzyNumber,character'
x[i]
```

Arguments

x	a fuzzy number
i	character; slot name

Details

All slot accessors are read-only.

Value

Returns the slot value.

See Also

Other FuzzyNumber-method: Arithmetic, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(),

plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PowerFuzzyNumber-method: PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval()

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber-class, DiscontinuousFuzzyNumber, distance(), integrateAlpha(), plot()

Examples

```
A <- FuzzyNumber(1,2,3,4)
A["a1"]
A["right"]
```

---

fapply *Apply a Function on a Fuzzy Number*

Description

Applies a given monotonic function using the extension principle (i.e. the function is applied on alpha-cuts).

Usage

```
## S4 method for signature 'PiecewiseLinearFuzzyNumber,function'
fapply(object, fun, ...)
```

Arguments

object	a fuzzy number
fun	a monotonic, vectorized R function
...	additional arguments passed to fun

Details

Currently only a method for the PiecewiseLinearFuzzyNumber class has been defined. The computations are exact (up to a numeric error) at knots. So, make sure you have a sufficient number of knots if you want good approximation.

For other types of fuzzy numbers, consider using piecewiseLinearApproximation.

Value

Returns a PiecewiseLinearFuzzyNumber.

See Also

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedanc necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other extension_principle: Arithmetic, ^,PiecewiseLinearFuzzyNumber,numeric-method

---

`FuzzyNumber` *Creates a Fuzzy Number*

Description

For convenience, objects of class FuzzyNumber may be created with this function.

Usage

```
FuzzyNumber(
  a1,
  a2,
  a3,
  a4,
  lower = function(a) rep(NA_real_, length(a)),
  upper = function(a) rep(NA_real_, length(a)),
  left = function(x) rep(NA_real_, length(x)),
  right = function(x) rep(NA_real_, length(x))
)
```

Arguments

a1	a number specyfing left bound of the support
a2	a number specyfing left bound of the core
a3	a number specyfing right bound of the core
a4	a number specyfing right bound of the support
lower	lower alpha-cut bound generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
upper	upper alpha-cut bound generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_
left	lower side function generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
right	upper side function generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_

Value

Object of class FuzzyNumber

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

FuzzyNumber-class *S4 class Representing a Fuzzy Number*

Description

Formally, a fuzzy number A (Dubois, Prade, 1987) is a fuzzy subset of the real line R with membership function μ given by:

$$\mu(x) = \begin{cases} 0 & \text{if } x < a1, \\ left((x-a1)/(a2-a1)) & \text{if } a1 \le x < a2, \\ 1 & \text{if } a2 \le x \le a3, \\ right((x-a3)/(a4-a3)) & \text{if } a3 < x \le a4, \\ 0 & \text{if } a4 < x, \end{cases}$$

where $a1, a2, a3, a4 \in R$, $a1 \le a2 \le a3 \le a4$, $left : [0,1] \to [0,1]$ is a nondecreasing function called the *left side generator of* A, and $right : [0,1] \to [0,1]$ is a nonincreasing function called the *right side generator of* A. Note that this is a so-called L-R representation of a fuzzy number.

Alternatively, it may be shown that each fuzzy number A may be uniquely determined by specifying its α-cuts, $A(\alpha)$. We have $A(0) = [a1, a4]$ and

$$A(\alpha) = [a1 + (a2 - a1) * lower(\alpha), a3 + (a4 - a3) * upper(\alpha)]$$

for $0 < \alpha \le 1$, where $lower : [0,1] \to [0,1]$ and $upper : [0,1] \to [0,1]$ are, respectively, strictly increasing and decreasing functions satisfying $lower(\alpha) = \inf\{x : \mu(x) \ge \alpha\}$ and $upper(\alpha) = \sup\{x : \mu(x) \ge \alpha\}$.

Details

Please note that many algorithms that deal with fuzzy numbers often use α-cuts rather than side functions.

Note that the **FuzzyNumbers** package also deals with particular types of fuzzy numbers like trapezoidal, piecewise linear, or "parametric" FNs.

Slots

`a1`: Single numeric value specifying the left bound for the support.

`a2`: Single numeric value specifying the left bound for the core.

`a3`: Single numeric value specifying the right bound for the core.

`a4`: Single numeric value specifying the right bound for the support.

`lower`: A nondecreasing function [0,1]->[0,1] that gives the lower alpha-cut bound.

`upper`: A nonincreasing function [0,1]->[1,0] that gives the upper alpha-cut bound.

`left`: A nondecreasing function [0,1]->[0,1] that gives the left side function.

`right`: A nonincreasing function [0,1]->[1,0] that gives the right side function.

Child/sub classes

`TrapezoidalFuzzyNumber`, `PiecewiseLinearFuzzyNumber`, `PowerFuzzyNumber`, and `DiscontinuousFuzzyNumber`

References

Dubois D., Prade H. (1987), Fuzzy numbers: An overview, In: Analysis of Fuzzy Information. Mathematical Logic, vol. I, CRC Press, pp. 3-39.

See Also

`FuzzyNumber` for a convenient constructor, and `as.FuzzyNumber` for conversion of objects to this class. Also, see `convertSide` for creating side functions generators, `convertAlpha` for creating alpha-cut bounds generators, `approxInvert` for inverting side functions/alpha-cuts numerically.

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `weightedExpectedValue()`, `width()`

Examples

```
showClass("FuzzyNumber")
showMethods(classes="FuzzyNumber")
```

---

integrateAlpha *Numerically Integrate Alpha-Cut Bounds*

Description

Integrates numerically a transformed or weighted lower or upper alpha-cut bound of a fuzzy number.

Usage

```
## S4 method for signature 'FuzzyNumber,character,numeric,numeric'
integrateAlpha(object, which=c("lower", "upper"),
   from=0, to=1, weight=NULL, transform=NULL, ...)

## S4 method for signature 'DiscontinuousFuzzyNumber,character,numeric,numeric'
integrateAlpha(object, which=c("lower", "upper"),
   from=0, to=1, weight=NULL, transform=NULL, ...)
```

Arguments

object	a fuzzy number
which	one of "lower", "upper"
from	numeric
to	numeric
...	additional arguments passed to integrate or integrate_discont_val
weight	a function or NULL
transform	a function or NULL

Value

Returns a single numeric value.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), piecewiseLinearApproximation(), plot(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber-class, DiscontinuousFuzzyNumber, Extract, distance(), plot()

---

integrate_discont_val *Integrate a Function with at Most Finite Number of Discontinuities *EXPERIMENTAL**

Description

The function uses multiple calls to integrate.

Usage

```
integrate_discont_val(f, from, to, discontinuities = numeric(0), ...)
```

Arguments

f	an R function taking a numeric vector of length 1 as its first argument and returning a numeric vector of length 1
from	the lower limit of integration
to	the upper limit of integration
discontinuities	nondecreasingly sorted numeric vector which indicates the points at which f is discontinuous
...	further arguments to be passed to the integrate function.

Value

Returns the estimate of the integral.

---

maximum *Maximum of fuzzy nubmers*

Description

Determines maximum fuzzy number based on two inputs.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
maximum(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Details

The resulting PiecewiseLinearFuzzyNumber is only an approximation of the result it might not be very precise for small number of knots (see examples).

Value

Returns a PiecewiseLinearFuzzyNumber representing maximal value of the inputs.

References

KAUFMANN, A., GUPTA, M. M. (1985) Introduction to Fuzzy Arithmetic. New York : Van Nostrand Reinhold Company. ISBN 044230079.

See Also

Other min_max-operators: minimum()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), minimum(), necessityExceedance(), necessityStrictExceedance necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
# example with low number of knots, showing the approximate nature
# of the result
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-4.8, -3 , -1.5))
y = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-5.5, -2.5, -1.1))
maxFN = maximum(x,y)
min = min(alphacut(x,0)[1],alphacut(y,0)[1])
max = max(alphacut(x,0)[2],alphacut(y,0)[2])
plot(x, col="red", xlim=c(min,max))
plot(y, col="blue", add=TRUE)
plot(maxFN, col="green", add=TRUE)

# example with high number of knots, that does not suffer
# from the approximate nature of the result
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-4.8, -3 , -1.5), knot.n = 9)
y = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-5.5, -2.5, -1.1), knot.n = 9)
maxFN = maximum(x,y)
min = min(alphacut(x,0)[1],alphacut(y,0)[1])
max = max(alphacut(x,0)[2],alphacut(y,0)[2])
plot(x, col="red", xlim=c(min,max))
plot(y, col="blue", add=TRUE)
plot(maxFN, col="green", add=TRUE)
```

---

minimum *Minimum of fuzzy nubmers*

Description

Determines minimum fuzzy number based on two inputs.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
minimum(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Details

The resulting PiecewiseLinearFuzzyNumber is only an approximation of the result it might not be very precise for small number of knots (see examples).

Value

Returns a PiecewiseLinearFuzzyNumber representing maximal value of the inputs.

References

KAUFMANN, A., GUPTA, M. M. (1985) Introduction to Fuzzy Arithmetic. New York : Van Nostrand Reinhold Company. ISBN 044230079.

See Also

Other min_max-operators: `maximum()`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `^,PiecewiseLinearFuzzyNumber,numeric-method`, `alphaInterval()`, `arctan2()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `necessityExceedance()`, `necessityStrictExceedance` `necessityStrictUndervaluation()`, `necessityUndervaluation()`, `plot()`, `possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Examples

```
# example with low number of knots, showing the approximate nature
# of the result
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-4.8, -3 , -1.5))
y = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-5.5, -2.5, -1.1))
minFN = minimum(x,y)
min = min(alphacut(x,0)[1],alphacut(y,0)[1])
max = max(alphacut(x,0)[2],alphacut(y,0)[2])
plot(x, col="red", xlim=c(min,max))
plot(y, col="blue", add=TRUE)
plot(minFN, col="green", add=TRUE)

# example with high number of knots, that does not suffer
# from the approximate nature of the result
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-4.8, -3 , -1.5), knot.n = 9)
y = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-5.5, -2.5, -1.1), knot.n = 9)
minFN = minimum(x,y)
min = min(alphacut(x,0)[1],alphacut(y,0)[1])
max = max(alphacut(x,0)[2],alphacut(y,0)[2])
plot(x, col="red", xlim=c(min,max))
plot(y, col="blue", add=TRUE)
```

```
plot(minFN, col="green", add=TRUE)
```

`necessityExceedance` *Necessity of exceedance*

Description

Determines value of necessity of $e1 >= e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
necessityExceedance(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the necessity of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: `necessityStrictExceedance()`, `necessityStrictUndervaluation()`, `necessityUndervaluation()`, `possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `^,PiecewiseLinearFuzzyNumber,numeric-method`, `alphaInterval()`, `arctan2()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `minimum()`, `necessityStrictExceedance()`, `necessityStrictUndervaluation()`, `necessityUndervaluation()`, `plot()`, `possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Examples

```
a <- TriangularFuzzyNumber(2, 3, 5)
b <- TriangularFuzzyNumber(1.5, 4, 4.8)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
necessityExceedance(a,b)
```

`necessityStrictExceedance` *Necessity of strict exceedance*

Description

Determines value of necessity of $e1 > e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
necessityStrictExceedance(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the strict necessity of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: `necessityExceedance()`, `necessityStrictUndervaluation()`, `necessityUndervaluatio`
`possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`,
`possibilityUndervaluation()`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber-class`,
`PiecewiseLinearFuzzyNumber`, `^,PiecewiseLinearFuzzyNumber,numeric-method`, `alphaInterval()`,
`arctan2()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`,
`as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `minimum()`, `necessityExceedance()`,
`necessityStrictUndervaluation()`, `necessityUndervaluation()`, `plot()`, `possibilityExceedance()`,
`possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Examples

```
a <- TriangularFuzzyNumber(2, 3, 5)
b <- TriangularFuzzyNumber(1.5, 4, 4.8)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
necessityStrictExceedance(a,b)
```

necessityStrictUndervaluation
Necessity of strict undervaluation

Description

Determines value of necessity of $e1 < e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
necessityStrictUndervaluation(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the necessity of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityUndervaluation(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(0.2, 1.0, 2.8)
b <- TriangularFuzzyNumber(0, 1.8, 2.2)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
necessityStrictUndervaluation(a,b)
```

necessityUndervaluation *Necessity of undervaluation*

Description

Determines value of necessity of $e1 <= e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
necessityUndervaluation(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the necessity of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluat possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(0.2, 1.0, 2.8)
b <- TriangularFuzzyNumber(0, 1.8, 2.2)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
necessityUndervaluation(a,b)
```

piecewiseLinearApproximation
Piecewise Linear Approximation of a Fuzzy Number

Description

This method finds a piecewise linear approximation $P(A)$ of a given fuzzy number A by using the algorithm specified by the `method` parameter.

Usage

```
## S4 method for signature 'FuzzyNumber'
piecewiseLinearApproximation(object,
   method=c("NearestEuclidean", "SupportCorePreserving",
   "Naive"),
   knot.n=1, knot.alpha=seq(0, 1, length.out=knot.n+2)[-c(1,knot.n+2)],
   ..., verbose=FALSE)
```

Arguments

`object`	a fuzzy number
`...`	further arguments passed to `integrateAlpha` [only `"NearestEuclidean"` and `"SupportCorePreserving"`]
`method`	character; one of: `"NearestEuclidean"` (default), `"SupportCorePreserving"`, or `"Naive"`
`knot.n`	desired number of knots (if missing, then calculated from given `knot.alpha`)
`knot.alpha`	alpha-cuts at which knots will be positioned (defaults to equally distributed knots)
`verbose`	logical; should some technical details on the computations being performed be printed? [only `"NearestEuclidean"`]

Details

'`method`' may be one of:

1. `NearestEuclidean`: see (Coroianu, Gagolewski, Grzegorzewski, 2013 and 2014a); uses numerical integration, see `integrateAlpha`. Slow for large `knot.n`.
2. `SupportCorePreserving`: This method was proposed in (Coroianu et al., 2014b) and is currently only available for `knot.n==1`. It is the L2-nearest piecewise linear approximation with constraints core(A)==core(P(A)) and supp(A)==supp(P(A)); uses numerical integration.
3. `Naive`: We have core(A)==core(P(A)) and supp(A)==supp(P(A)) and the knots are taken directly from the specified alpha cuts (linear interpolation).

Value

Returns a `PiecewiseLinearFuzzyNumber` object.

References

Coroianu L., Gagolewski M., Grzegorzewski P. (2013), Nearest Piecewise Linear Approximation of Fuzzy Numbers, Fuzzy Sets and Systems 233, pp. 26-51.

Coroianu L., Gagolewski M., Grzegorzewski P., Adabitabar Firozja M., Houlari T. (2014a), Piecewise linear approximation of fuzzy numbers preserving the support and core, In: Laurent A. et al. (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II (CCIS 443), Springer, pp. 244-254.

Coroianu L., Gagolewski M., Grzegorzewski P. (2014b), Nearest Piecewise Linear Approximation of Fuzzy Numbers - General Case, submitted for publication.

See Also

Other approximation: `trapezoidalApproximation()`

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `weightedExpectedValue()`, `width()`

Examples

```
(A <- FuzzyNumber(-1, 0, 1, 3,
   lower=function(x) sqrt(x),upper=function(x) 1-sqrt(x)))
(PA <- piecewiseLinearApproximation(A, "NearestEuclidean",
   knot.n=1, knot.alpha=0.2))
```

`PiecewiseLinearFuzzyNumber`

Creates a Piecewise Linear Fuzzy Number

Description

For convenience, objects of class `PiecewiseLinearFuzzyNumber` may be created with this function.

Usage

```
PiecewiseLinearFuzzyNumber(
  a1,
  a2,
  a3,
  a4,
  knot.n = 0,
  knot.alpha = numeric(0),
  knot.left = numeric(0),
  knot.right = numeric(0)
)
```

Arguments

a1	a number specyfing left bound of the support
a2	a number specyfing left bound of the core
a3	a number specyfing right bound of the core
a4	a number specyfing right bound of the support
knot.n	the number of knots
knot.alpha	knot.n alpha-cut values at knots
knot.left	knot.n knots on the left side; a nondecreasingly sorted vector with elements in [a1,a2]
knot.right	knot.n knots on the right side; a nondecreasingly sorted vector with elements in [a3,a4]

Details

If a1, a2, a3, and a4 are missing, then knot.left and knot.right may be of length knot.n+2.

If knot.n is not given, then it guessed from length(knot.left). If knot.alpha is missing, then the knots will be equally distributed on the interval [0,1].

Value

An object of class PiecewiseLinearFuzzyNumber.

See Also

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

PiecewiseLinearFuzzyNumber-class

S4 Class Representing a Piecewise Linear Fuzzy Number

Description

A piecewise linear fuzzy number (PLFN) has side functions and alpha-cut bounds that linearly interpolate a given set of points (at fixed alpha-cuts).

Details

If knot.n is equal to 0 or all left and right knots lie on common lines, then a Piecewise Linear Fuzzy Number reduces to a TrapezoidalFuzzyNumber. Note that, however, the TrapezoidalFuzzyNumber does not inherit from PiecewiseLinearFuzzyNumber for efficiency reasons. To convert the former to the latter, call as.PiecewiseLinearFuzzyNumber.

Slots

`a1`, `a2`, `a3`, `a4`, `lower`, `upper`, `left`, `right`: Inherited from the `FuzzyNumber` class.

`knot.n`: number of knots, a single integer value, 0 for a trapezoidal fuzzy number

`knot.alpha`: alpha-cuts, increasingly sorted vector of length `knot.n` with elements in [0,1]

`knot.left`: nondecreasingly sorted vector of length `knot.n`; defines left alpha-cut bounds at knots

`knot.right`: nondecreasingly sorted vector of length `knot.n`; defines right alpha-cut bounds at knots

Extends

Class `FuzzyNumber`, directly.

See Also

`PiecewiseLinearFuzzyNumber` for a convenient constructor, `as.PiecewiseLinearFuzzyNumber` for conversion of objects to this class, and `piecewiseLinearApproximation` for approximation routines.

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber`, `^,PiecewiseLinearFuzzyNumber,numeric-method`, `alphaInterval()`, `arctan2()`, `as.PiecewiseLinearFuzzyNumber(`
`as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `minimum()`, `necessityExceedance()`, `necessityStrictExceedance()`, `necessityStrictUndervaluation()`, `necessityUndervaluation()`, `plot()`, `possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Examples

```
showClass("PiecewiseLinearFuzzyNumber")
showMethods(classes="PiecewiseLinearFuzzyNumber")
```

---

plot *Plot a Fuzzy Number*

Description

The function aims to provide a similar look-and-feel to the built-in `plot.default` and `curve` function.

Usage

```
## S4 method for signature 'FuzzyNumber,missing'
plot(x, y, from=NULL, to=NULL, n=101, at.alpha=NULL,
draw.membership.function=TRUE, draw.alphacuts=!draw.membership.function,
xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
type="l", col=1, lty=1, pch=1, lwd=1,
shadowdensity=15, shadowangle=45, shadowcol=col, shadowborder=NULL,
add=FALSE, ...)
```

```
## S4 method for signature 'TrapezoidalFuzzyNumber,missing'
plot(x, y, from=NULL, to=NULL,
draw.membership.function=TRUE, draw.alphacuts=!draw.membership.function,
xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
type="l", col=1, lty=1, pch=1, lwd=1, add=FALSE, ...)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,missing'
plot(x, y, from=NULL, to=NULL,
draw.membership.function=TRUE, draw.alphacuts=!draw.membership.function,
xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
type="l", col=1, lty=1, pch=1, lwd=1, add=FALSE, ...)

## S4 method for signature 'DiscontinuousFuzzyNumber,missing'
plot(x, y, from=NULL, to=NULL,
n=101, draw.membership.function=TRUE, draw.alphacuts=!draw.membership.function,
xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
type="l", col=1, lty=1, pch=1, lwd=1,
add=FALSE, ...)
```

Arguments

x	a fuzzy number
y	not used
from	numeric;
to	numeric;
n	numeric; number of points to probe
at.alpha	numeric vector; give exact alpha-cuts at which linear interpolation should be done
draw.membership.function	logical; you want membership function (TRUE) or alpha-cuts plot (FALSE)?
draw.alphacuts	logical; defaults !draw.membership.function
xlab	character; x-axis label
ylab	character; y-axis label
xlim	numeric;
ylim	numeric;
type	character; defaults "l"; plot type, e.g.~"l" for lines, "p" for points, or "b" for both
col	see plot.default
lty	see plot.default
pch	see plot.default
lwd	see plot.default
shadowdensity	numeric; for shadowed sets;
shadowangle	numeric; for shadowed sets;

shadowcol	color specification, see plot.default; for shadowed sets;
shadowborder	numeric; for shadowed sets;
add	logical; add another FuzzyNumber to existing plot?
...	further arguments passed to plot.default

Details

Note that if from > a1 then it is set to a1.

Value

Returns nothing really interesting.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), show(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval()

Other DiscontinuousFuzzyNumber-method: DiscontinuousFuzzyNumber-class, DiscontinuousFuzzyNumber, Extract, distance(), integrateAlpha()

Examples

```
plot(FuzzyNumber(0,1,2,3), col="gray")
plot(FuzzyNumber(0,1,2,3, left=function(x) x^2, right=function(x) 1-x^3), add=TRUE)
plot(FuzzyNumber(0,1,2,3, lower=function(x) x, upper=function(x) 1-x), add=TRUE, col=2)
```

possibilityExceedance *Possibility of exceedance*

Description

Determines value of possibility of $e1 >= e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
possibilityExceedance(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the possibility of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluat necessityUndervaluation(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityStrictExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(2, 3, 5)
b <- TriangularFuzzyNumber(1.5, 4, 4.8)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
possibilityExceedance(a,b)
```

possibilityStrictExceedance *Possibility of strict exceedance*

Description

Determines value of possibility of $e1 > e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
possibilityStrictExceedance(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the strict possibility of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluat necessityUndervaluation(), possibilityExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictUndervaluation(), possibilityUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(2, 3, 5)
b <- TriangularFuzzyNumber(1.5, 4, 4.8)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
possibilityStrictExceedance(a,b)
```

possibilityStrictUndervaluation
Possibility of strict undervaluation

Description

Determines value of possibility of $e1 < e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
possibilityStrictUndervaluation(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the necessity of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluat necessityUndervaluation(), possibilityExceedance(), possibilityStrictExceedance(), possibilityUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(0.2, 1.0, 2.8)
b <- TriangularFuzzyNumber(0, 1.8, 2.2)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
possibilityStrictUndervaluation(a,b)
```

possibilityUndervaluation
Possibility of undervaluation

Description

Determines value of possibility of $e1 <= e2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
possibilityUndervaluation(e1, e2)
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	a PiecewiseLinearFuzzyNumber

Value

Returns a value from range [0,1] indicating the possibility of exceedance of e2 by e1.

References

DUBOIS, Didier and PRADE, Henri, 1983, Ranking Fuzzy Numbers in the Setting of Possibility Theory. Information Sciences. 1983. Vol. 30, no. 3, p. 183–224.

See Also

Other comparison-operators: necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluat necessityUndervaluation(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation()

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, Extract, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, ^,PiecewiseLinearFuzzyNumber,numeric-method, alphaInterval(), arctan2(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval(), fapply(), maximum(), minimum(), necessityExceedance(), necessityStrictExceedance(), necessityStrictUndervaluation(), necessityUndervaluation(), plot(), possibilityExceedance(), possibilityStrictExceedance(), possibilityStrictUndervaluation()

Examples

```
a <- TriangularFuzzyNumber(0.2, 1.0, 2.8)
b <- TriangularFuzzyNumber(0, 1.8, 2.2)
a <- as.PiecewiseLinearFuzzyNumber(a, knot.n = 9)
b <- as.PiecewiseLinearFuzzyNumber(b, knot.n = 9)
possibilityUndervaluation(a,b)
```

PowerFuzzyNumber *Creates a Fuzzy Number with Sides Given by Power Functions*

Description

For convenience, objects of class PowerFuzzyNumber may be created with this function.

Usage

```
PowerFuzzyNumber(a1, a2, a3, a4, p.left = 1, p.right = 1)
```

Arguments

a1	a number specyfing left bound of the support
a2	a number specyfing left bound of the core
a3	a number specyfing right bound of the core
a4	a number specyfing right bound of the support
p.left	a positive number specyfing the exponent for the left side
p.right	a positive number specyfing the exponent for the right side

Value

Object of class PowerFuzzyNumber

See Also

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber-class, alphaInterval(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval()

PowerFuzzyNumber-class *S4 class Representing a Fuzzy Number with Sides Given by Power Functions*

Description

Bodjanova-type fuzzy numbers which sides are given by power functions are defined using four coefficients a1 <= a2 <= a3 <= a4, and parameters p.left, p.right>0, which determine exponents for the side functions.

Details

We have $\texttt{left}(x) = x^{\texttt{p.left}}$, and $\texttt{right}(x) = (1-x)^{\texttt{p.right}}$.

This class is a natural generalization of trapezoidal FNs. For other see PiecewiseLinearFuzzyNumber.

Slots

a1, a2, a3, a4, lower, upper, left, right: Inherited from the FuzzyNumber class.

p.left: single numeric value; 1.0 for a trapezoidal FN.

p.right: single numeric value; 1.0 for a trapezoidal FN.

Extends

Class FuzzyNumber, directly.

References

Bodjanova S. (2005), Median value and median interval of a fuzzy number, Information Sciences 172, pp. 73-89.

See Also

PowerFuzzyNumber for a convenient constructor, as.PowerFuzzyNumber for conversion of objects to this class.

PowerFuzzyNumber for a convenient constructor

Other PowerFuzzyNumber-method: Extract, PowerFuzzyNumber, alphaInterval(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), expectedInterval()

Examples

```
showClass("PowerFuzzyNumber")
showMethods(classes="PowerFuzzyNumber")
```

---

show *Print Basic Information on a Fuzzy Number*

Description

See as.character for more details.

Usage

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

Arguments

object	a fuzzy number

Details

The method as.character is called on given fuzzy number object with default arguments. The results are printed on stdout.

Value

Does not return anything interesting.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), supp(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

---

supp *Calculate the Support of a Fuzzy Number*

Description

We have $\mathrm{supp}(A) := [a1, a4]$. This gives the values that a fuzzy number possibly may represent.

Usage

```
## S4 method for signature 'FuzzyNumber'
supp(object)
```

Arguments

object	a fuzzy number

Value

Returns a numeric vector of length 2.

See Also

Other FuzzyNumber-method: Arithmetic, Extract, FuzzyNumber-class, FuzzyNumber, alphaInterval(), alphacut(), ambiguity(), as.FuzzyNumber(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), as.character(), core(), distance(), evaluate(), expectedInterval(), expectedValue(), integrateAlpha(), piecewiseLinearApproximation(), plot(), show(), trapezoidalApproximation(), value(), weightedExpectedValue(), width()

Other alpha_cuts: alphacut(), core()

`trapezoidalApproximation` *Trapezoidal Approximation of a Fuzzy Number*

Description

This method finds a trapezoidal approximation $T(A)$ of a given fuzzy number A by using the algorithm specified by the `method` parameter.

Usage

```
## S4 method for signature 'FuzzyNumber'
trapezoidalApproximation(object,
   method=c("NearestEuclidean", "ExpectedIntervalPreserving",
            "SupportCoreRestricted", "Naive"),
   ..., verbose=FALSE)
```

Arguments

`object`	a fuzzy number
`...`	further arguments passed to `integrateAlpha`
`method`	character; one of: `"NearestEuclidean"` (default), `"ExpectedIntervalPreserving"`, `"SupportCoreRestricted"`, `"Naive"`
`verbose`	logical; should some technical details on the computations being performed be printed?

Details

`method` may be one of:

1. `NearestEuclidean`: see (Ban, 2009); uses numerical integration, see `integrateAlpha`
2. `Naive`: We have core(A)==core(T(A)) and supp(A)==supp(T(A))
3. `ExpectedIntervalPreserving`: L2-nearest trapezoidal approximation preserving the expected interval given in (Grzegorzewski, 2010; Ban, 2008; Yeh, 2008) Unfortunately, for highly skewed membership functions this approximation operator may have quite unfavourable behavior. For example, if Val(A) < EV_1/3(A) or Val(A) > EV_2/3(A), then it may happen that the core of the output and the core of the original fuzzy number A are disjoint (cf. Grzegorzewski, Pasternak-Winiarska, 2011)
4. `SupportCoreRestricted`: This method was proposed in (Grzegorzewski, Pasternak-Winiarska, 2011). L2-nearest trapezoidal approximation with constraints core(A) $\subseteq$ core(T(A)) and supp(T(A)) $\subseteq$ supp(A), i.e. for which each point that surely belongs to A also belongs to T(A), and each point that surely does not belong to A also does not belong to T(A).

Value

Returns a `TrapezoidalFuzzyNumber` object.

References

Ban A.I. (2008), Approximation of fuzzy numbers by trapezoidal fuzzy numbers preserving the expected interval, Fuzzy Sets and Systems 159, pp. 1327-1344.

Ban A.I. (2009), On the nearest parametric approximation of a fuzzy number - Revisited, Fuzzy Sets and Systems 160, pp. 3027-3047.

Grzegorzewski P. (2010), Algorithms for trapezoidal approximations of fuzzy numbers preserving the expected interval, In: Bouchon-Meunier B. et al (Eds.), Foundations of Reasoning Under Uncertainty, Springer, pp. 85-98.

Grzegorzewski P, Pasternak-Winiarska K. (2011), Trapezoidal approximations of fuzzy numbers with restrictions on the support and core, Proc. EUSFLAT/LFA 2011, Atlantis Press, pp. 749-756.

Yeh C.-T. (2008), Trapezoidal and triangular approximations preserving the expected interval, Fuzzy Sets and Systems 159, pp. 1345-1353.

See Also

Other approximation: `piecewiseLinearApproximation()`

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `value()`, `weightedExpectedValue()`, `width()`

Examples

```
(A <- FuzzyNumber(-1, 0, 1, 40,
   lower=function(x) sqrt(x), upper=function(x) 1-sqrt(x)))
(TA <- trapezoidalApproximation(A,
   "ExpectedIntervalPreserving")) # Note that the cores are disjoint!
expectedInterval(A)
expectedInterval(TA)
```

`TrapezoidalFuzzyNumber`

Creates a Trapezoidal Fuzzy Number

Description

For convenience, objects of class `TrapezoidalFuzzyNumber` may be created with this function.

Usage

```
TrapezoidalFuzzyNumber(a1, a2, a3, a4)
```

Arguments

a1	a number specyfing left bound of the support
a2	a number specyfing left bound of the core
a3	a number specyfing right bound of the core
a4	a number specyfing right bound of the support

Value

Object of class TrapezoidalFuzzyNumber

See Also

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber-class, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval(), plot()

TrapezoidalFuzzyNumber-class

S4 class Representing a Trapezoidal Fuzzy Number

Description

Trapezoidal Fuzzy Numbers have linear side functions and alpha-cut bounds.

Details

Trapezoidal fuzzy numbers are among the simplest FNs. Despite their simplicity, however, they include triangular FNs, "crisp" real intervals, and "crisp" reals. Please note that currently no separate classes for these particular TFNs types are implemented in the package.

Slots

a1, a2, a3, a4, lower, upper, left, right: Inherited from the FuzzyNumber class.

Extends

Class FuzzyNumber, directly.

See Also

TrapezoidalFuzzyNumber for a convenient constructor, as.TrapezoidalFuzzyNumber for conversion of objects to this class, and trapezoidalApproximation for approximation routines.

Other TrapezoidalFuzzyNumber-method: Arithmetic, TrapezoidalFuzzyNumber, TriangularFuzzyNumber(), alphaInterval(), as.PiecewiseLinearFuzzyNumber(), as.PowerFuzzyNumber(), as.TrapezoidalFuzzyNumber(), expectedInterval(), plot()

Examples

```
showClass("TrapezoidalFuzzyNumber")
showMethods(classes="TrapezoidalFuzzyNumber")
```

`TriangularFuzzyNumber` *Creates a Triangular Fuzzy Number*

Description

For convenience, objects of class `TrapezoidalFuzzyNumber` may be created with this function.

Usage

```
TriangularFuzzyNumber(a1, amid, a4)
```

Arguments

`a1`	a number specyfing left bound of the support
`amid`	a number specyfing the core
`a4`	a number specyfing right bound of the support

Details

Currently there is no separate class of a Triangular Fuzzy Number.

Value

Object of class `TrapezoidalFuzzyNumber`

See Also

Other TrapezoidalFuzzyNumber-method: `Arithmetic`, `TrapezoidalFuzzyNumber-class`, `TrapezoidalFuzzyNumber`, `alphaInterval()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `expectedInterval()`, `plot()`

`value` *Calculate the Value of a Fuzzy Number*

Description

The calculation of the so-called value is one of possible methods to deffuzify a fuzzy number.

Usage

```
## S4 method for signature 'FuzzyNumber'
value(object, ...)
```

Arguments

`object`	a fuzzy number
`...`	additional arguments passed to `alphaInterval`

Details

The value of A (Delgrado et al, 1998) is defined as $val(A) := \int_0^1 \alpha \left(A_L(\alpha) + A_U(\alpha)\right)\, d\alpha$.

Value

Returns a single numeric value.

References

Delgado M., Vila M.A., Voxman W. (1998), On a canonical representation of a fuzzy number, Fuzzy Sets and Systems 93, pp. 125-135.

See Also

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `weightedExpectedValue()`, `width()`

Other deffuzification: `expectedValue()`, `weightedExpectedValue()`

Other characteristics: `ambiguity()`, `expectedValue()`, `weightedExpectedValue()`, `width()`

`weightedExpectedValue` *Calculate the Weighted Expected Value of a Fuzzy Number*

Description

The calculation of the so-called weighted expected value is one of possible methods to deffuzify a fuzzy number.

For $w = 0.5$ we get the ordinary `expectedValue`.

Usage

```
## S4 method for signature 'FuzzyNumber'
weightedExpectedValue(object, w=0.5, ...)
```

Arguments

`object`	a fuzzy number
`...`	additional arguments passed to `expectedInterval`
`w`	a single numeric value in [0,1]

Details

The weighted expected value of A is defined as $EV_w(A) := (1 - w)EI_L(A) + wEI_U(A)$, where EI is the `expectedInterval`.

Value

Returns a single numeric value.

See Also

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `width()`

Other deffuzification: `expectedValue()`, `value()`

Other characteristics: `ambiguity()`, `expectedValue()`, `value()`, `width()`

`width` *Calculate the Width of a Fuzzy Number*

Description

The width (Chanas, 2001) is a measure of nonspecificity of a fuzzy number.

Usage

```
## S4 method for signature 'FuzzyNumber'
width(object, ...)
```

Arguments

`object`	a fuzzy number
`...`	additional arguments passed to `expectedInterval`

Details

The width of A is defined as $width(A) := EI_U(A) - EI_L(A)$, where EI is the `expectedInterval`.

Value

Returns a single numeric value.

References

Chanas S. (2001), On the interval approximation of a fuzzy number, Fuzzy Sets and Systems 122, pp. 353-356.

See Also

Other FuzzyNumber-method: `Arithmetic`, `Extract`, `FuzzyNumber-class`, `FuzzyNumber`, `alphaInterval()`, `alphacut()`, `ambiguity()`, `as.FuzzyNumber()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `core()`, `distance()`, `evaluate()`, `expectedInterval()`, `expectedValue()`, `integrateAlpha()`, `piecewiseLinearApproximation()`, `plot()`, `show()`, `supp()`, `trapezoidalApproximation()`, `value()`, `weightedExpectedValue()`

Other characteristics: `ambiguity()`, `expectedValue()`, `value()`, `weightedExpectedValue()`

^,PiecewiseLinearFuzzyNumber,numeric-method

Integer power of fuzzy number

Description

For fuzzy numbers the equality of `X*X == X^2` does not hold.

Usage

```
## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
e1 ^ e2
```

Arguments

e1	a PiecewiseLinearFuzzyNumber
e2	numeric (if it is not integer it will be converted by function as.integer())

Details

This function calculates integer power of a PiecewiseLinearFuzzyNumber according to the reference below.

Value

Returns a fuzzy number of the class PiecewiseLinearFuzzyNumber indicating `e1^e2`.

References

KAUFMANN, A., GUPTA, M. M. (1985) Introduction to Fuzzy Arithmetic. New York : Van Nostrand Reinhold Company. ISBN 044230079.

See Also

Other extension_principle: `Arithmetic`, `fapply()`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic`, `Extract`, `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `alphaInterval()`, `arctan2()`, `as.PiecewiseLinearFuzzyNumber()`, `as.PowerFuzzyNumber()`, `as.TrapezoidalFuzzyNumber()`, `as.character()`, `expectedInterval()`, `fapply()`, `maximum()`, `minimum()`, `necessityExceedance()`, `necessityStrictExceedance()`, `necessityStrictUndervaluation()`, `necessityUndervaluation()`, `plot()`, `possibilityExceedance()`, `possibilityStrictExceedance()`, `possibilityStrictUndervaluation()`, `possibilityUndervaluation()`

Examples

```
x = as.PiecewiseLinearFuzzyNumber(TriangularFuzzyNumber(-2, 1, 9), knot.n = 2)
x^2
x^3
```

Index

∗ **DiscontinuousFuzzyNumber-method**
DiscontinuousFuzzyNumber, 19
DiscontinuousFuzzyNumber-class, 21
distance, 22
Extract, 26
integrateAlpha, 30
plot, 42
∗ **FuzzyNumber-method**
alphacut, 5
alphaInterval, 6
ambiguity, 7
Arithmetic, 9
as.character, 12
as.FuzzyNumber, 13
as.PiecewiseLinearFuzzyNumber, 14
as.PowerFuzzyNumber, 15
as.TrapezoidalFuzzyNumber, 16
core, 19
distance, 22
evaluate, 23
expectedInterval, 24
expectedValue, 25
Extract, 26
FuzzyNumber, 28
FuzzyNumber-class, 29
integrateAlpha, 30
piecewiseLinearApproximation, 39
plot, 42
show, 50
supp, 51
trapezoidalApproximation, 52
value, 56
weightedExpectedValue, 57
width, 58
∗ **PiecewiseLinearFuzzyNumber-method**
^,PiecewiseLinearFuzzyNumber,numeric-method, 59
alphaInterval, 6
arctan2, 8
Arithmetic, 9
as.character, 12
as.PiecewiseLinearFuzzyNumber, 14
as.PowerFuzzyNumber, 15
as.TrapezoidalFuzzyNumber, 16
expectedInterval, 24
Extract, 26
fapply, 27
maximum, 32
minimum, 33
necessityExceedance, 35
necessityStrictExceedance, 36
necessityStrictUndervaluation, 37
necessityUndervaluation, 38
PiecewiseLinearFuzzyNumber, 40
PiecewiseLinearFuzzyNumber-class, 41
plot, 42
possibilityExceedance, 45
possibilityStrictExceedance, 46
possibilityStrictUndervaluation, 47
possibilityUndervaluation, 48
∗ **PowerFuzzyNumber-method**
alphaInterval, 6
as.character, 12
as.PowerFuzzyNumber, 15
as.TrapezoidalFuzzyNumber, 16
expectedInterval, 24
Extract, 26
PowerFuzzyNumber, 49
PowerFuzzyNumber-class, 49
∗ **TrapezoidalFuzzyNumber-method**
alphaInterval, 6
Arithmetic, 9
as.PiecewiseLinearFuzzyNumber, 14
as.PowerFuzzyNumber, 15
as.TrapezoidalFuzzyNumber, 16
expectedInterval, 24

plot, 42
TrapezoidalFuzzyNumber, 53
TrapezoidalFuzzyNumber-class, 54
TriangularFuzzyNumber, 55
* **TriangularFuzzyNumber-method**
as.character, 12
* **alpha_cuts**
alphacut, 5
core, 19
supp, 51
* **approximation**
piecewiseLinearApproximation, 39
trapezoidalApproximation, 52
* **auxiliary**
approxInvert, 8
convertAlpha, 17
convertSide, 18
* **characteristics**
ambiguity, 7
expectedValue, 25
value, 56
weightedExpectedValue, 57
width, 58
* **comparison-operators**
necessityExceedance, 35
necessityStrictExceedance, 36
necessityStrictUndervaluation, 37
necessityUndervaluation, 38
possibilityExceedance, 45
possibilityStrictExceedance, 46
possibilityStrictUndervaluation, 47
possibilityUndervaluation, 48
* **conversion**
as.character, 12
as.FuzzyNumber, 13
as.PiecewiseLinearFuzzyNumber, 14
as.PowerFuzzyNumber, 15
as.TrapezoidalFuzzyNumber, 16
* **deffuzification**
expectedValue, 25
value, 56
weightedExpectedValue, 57
* **extension_principle**
^,PiecewiseLinearFuzzyNumber,numeric-method (Arithmetic), 59
Arithmetic, 9
fapply, 27
* **membership_function**
evaluate, 23
* **min_max-operators**
maximum, 32
minimum, 33
* **special-method**
arctan2, 8
*,PiecewiseLinearFuzzyNumber,FuzzyNumber-method (Arithmetic), 9
*,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-me (Arithmetic), 9
*,PiecewiseLinearFuzzyNumber,numeric-method (Arithmetic), 9
*,TrapezoidalFuzzyNumber,numeric-method (Arithmetic), 9
*,numeric,FuzzyNumber-method (Arithmetic), 9
+,PiecewiseLinearFuzzyNumber,FuzzyNumber-method (Arithmetic), 9
+,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-me (Arithmetic), 9
+,PiecewiseLinearFuzzyNumber,numeric-method (Arithmetic), 9
+,TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber-method (Arithmetic), 9
+,numeric,FuzzyNumber-method (Arithmetic), 9
-,FuzzyNumber,ANY-method (Arithmetic), 9
-,PiecewiseLinearFuzzyNumber,FuzzyNumber-method (Arithmetic), 9
-,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-me (Arithmetic), 9
-,PiecewiseLinearFuzzyNumber,numeric-method (Arithmetic), 9
-,TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber-method (Arithmetic), 9
-,numeric,FuzzyNumber-method (Arithmetic), 9
/,PiecewiseLinearFuzzyNumber,FuzzyNumber-method (Arithmetic), 9
/,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-me (Arithmetic), 9
/,PiecewiseLinearFuzzyNumber,numeric-method (Arithmetic), 9
[,DiscontinuousFuzzyNumber,character-method (Extract), 26
[,FuzzyNumber,character-method (Extract), 26

[,PiecewiseLinearFuzzyNumber,character-method (Extract), 26
[,PowerFuzzyNumber,character-method (Extract), 26
^,PiecewiseLinearFuzzyNumber,numeric-method, 59

alphacut, 5, *6, 7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
alphacut,FuzzyNumber,numeric-method (alphacut), 5
alphaInterval, *5*, 6, *7*, *9*, *11–17*, *19*, *23–31*, *33–38*, *40–42*, *44–51*, *53–59*
alphaInterval,FuzzyNumber-method (alphaInterval), 6
alphaInterval,PiecewiseLinearFuzzyNumber-method (alphaInterval), 6
alphaInterval,PowerFuzzyNumber-method (alphaInterval), 6
alphaInterval,TrapezoidalFuzzyNumber-method (alphaInterval), 6
ambiguity, *5, 6*, 7, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
ambiguity,FuzzyNumber-method (ambiguity), 7
approxfun, *8*
approxInvert, 8, *18*, *30*
arctan2, *7*, 8, *11*, *13, 14*, *16, 17*, *25, 26*, *28*, *33–38*, *41, 42*, *44–48*, *59*
arctan2,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method (arctan2), 8
Arithmetic, *3*, *5–7*, *9*, 9, *12–17*, *19*, *23–26*, *28–31*, *33–38*, *40–42*, *44–48*, *51*, *53–59*
as.character, *3*, *5–7*, *9*, *11*, 12, *13–17*, *19*, *23–31*, *33–38*, *40–42*, *44–51*, *53*, *56–59*
as.character,FuzzyNumber-method (as.character), 12
as.character,PiecewiseLinearFuzzyNumber-method (as.character), 12
as.character,PowerFuzzyNumber-method (as.character), 12
as.character,TrapezoidalFuzzyNumber-method (as.character), 12
as.FuzzyNumber, *5–7*, *11–13*, 13, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
as.FuzzyNumber,FuzzyNumber-method (as.FuzzyNumber), 13
as.FuzzyNumber,numeric-method (as.FuzzyNumber), 13
as.PiecewiseLinearFuzzyNumber, *5–7*, *9*, *11–13*, 14, *16, 17*, *19*, *23–26*, *28–31*, *33–38*, *40–42*, *44–48*, *51*, *53–59*
as.PiecewiseLinearFuzzyNumber,FuzzyNumber-method (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber,numeric-method (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-m (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber,TrapezoidalFuzzyNumber-metho (as.PiecewiseLinearFuzzyNumber), 14
as.PowerFuzzyNumber, *5–7*, *9*, *11–15*, 15, *17*, *19*, *23–31*, *33–38*, *40–42*, *44–51*, *53–59*
as.PowerFuzzyNumber,FuzzyNumber-method (as.PowerFuzzyNumber), 15
as.PowerFuzzyNumber,numeric-method (as.PowerFuzzyNumber), 15
as.PowerFuzzyNumber,PiecewiseLinearFuzzyNumber-method (as.PowerFuzzyNumber), 15
as.PowerFuzzyNumber,PowerFuzzyNumber-method (as.PowerFuzzyNumber), 15
as.PowerFuzzyNumber,TrapezoidalFuzzyNumber-method (as.PowerFuzzyNumber), 15
as.TrapezoidalFuzzyNumber, *5–7*, *9*, *11–16*, 16, *19*, *23–31*, *33–38*, *40–42*, *44–51*, *53–59*
as.TrapezoidalFuzzyNumber,FuzzyNumber-method (as.TrapezoidalFuzzyNumber), 16
as.TrapezoidalFuzzyNumber,numeric-method (as.TrapezoidalFuzzyNumber), 16
as.TrapezoidalFuzzyNumber,PiecewiseLinearFuzzyNumber-metho (as.TrapezoidalFuzzyNumber), 16
as.TrapezoidalFuzzyNumber,PowerFuzzyNumber-method (as.TrapezoidalFuzzyNumber), 16
as.TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber-method (as.TrapezoidalFuzzyNumber), 16

cat, *12*
convertAlpha, *8*, 17, *18*, *30*
convertSide, *8*, *18*, 18, *30*
core, *5–7*, *11–13*, *15–17*, 19, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*

core,FuzzyNumber-method *(core)*, 19
curve, *42*

DiscontinuousFuzzyNumber, *3*, *19*, 19, *20*, *21*, *23*, *27*, *30*, *31*, *44*
DiscontinuousFuzzyNumber-class, 21
distance, *5–7*, *11–13*, *15–17*, *19*, *21*, 22, *23–27*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
distance,DiscontinuousFuzzyNumber,DiscontinuousFuzzyNumber-method *(distance)*, 22
distance,DiscontinuousFuzzyNumber,FuzzyNumber-method *(distance)*, 22
distance,FuzzyNumber,DiscontinuousFuzzyNumber-method *(distance)*, 22
distance,FuzzyNumber,FuzzyNumber-method *(distance)*, 22

evaluate, *5–7*, *11–13*, *15–17*, *19*, *23*, 23, *24–26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
evaluate,FuzzyNumber,numeric-method *(evaluate)*, 23
expectedInterval, *5–7*, *9*, *11–17*, *19*, *23*, 24, *25–31*, *33–38*, *40–42*, *44–51*, *53–59*
expectedInterval,FuzzyNumber-method *(expectedInterval)*, 24
expectedInterval,PiecewiseLinearFuzzyNumber-method *(expectedInterval)*, 24
expectedInterval,PowerFuzzyNumber-method *(expectedInterval)*, 24
expectedInterval,TrapezoidalFuzzyNumber-method *(expectedInterval)*, 24
expectedValue, *3*, *5–7*, *11–13*, *15–17*, *19*, *23*, *24*, 25, *26*, *29–31*, *40*, *44*, *51*, *53*, *56–58*
expectedValue,FuzzyNumber-method *(expectedValue)*, 25
Extract, *5–7*, *9*, *11–17*, *19*, *21*, *23–25*, 26, *28–31*, *33–38*, *40–42*, *44–51*, *53*, *56–59*

fapply, *3*, *7*, *9*, *11*, *13*, *14*, *16*, *17*, *25*, *26*, 27, *33–38*, *41*, *42*, *44–48*, *59*
fapply,PiecewiseLinearFuzzyNumber,function-method *(fapply)*, 27
FuzzyNumber, *3*, *5–8*, *11–13*, *15–19*, *21*, *23–26*, *28*, 28, *29–31*, *40*, *42*, *44*, *50*, *51*, *53*, *54*, *56–58*
FuzzyNumber-class, 29
FuzzyNumbers-package, 3

integrate, *22*, *31*, *32*
integrate_discont_val, *31*, 31
integrateAlpha, *5–7*, *11–13*, *15–17*, *19*, *21*, *23–27*, *29*, *30*, 30, *39*, *40*, *44*, *51–53*, *56–58*
integrateAlpha,DiscontinuousFuzzyNumber,character,numeric,numeric-method *(integrateAlpha)*, 30
integrateAlpha,FuzzyNumber,character,numeric,numeric-method *(integrateAlpha)*, 30

maximum, *7*, *9*, *11*, *13*, *14*, *16*, *17*, *25*, *26*, *28*, 32, *34–38*, *41*, *42*, *44–48*, *59*
maximum,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(maximum)*, 32
minimum, *7*, *9*, *11*, *13*, *14*, *16*, *17*, *25*, *26*, *28*, *33*, 33, *35–38*, *41*, *42*, *44–48*, *59*
minimum,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(minimum)*, 33

necessityExceedance, *7*, *9*, *11*, *13*, *14*, *16*, *17*, *25*, *26*, *28*, *33*, *34*, 35, *36–38*, *41*, *42*, *44–48*, *59*
necessityExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(necessityExceedance)*, 35
necessityStrictExceedance, *7*, *9*, *11*, *13*, *14*, *16*, *17*, *25*, *26*, *28*, *33–35*, 36, *37*, *38*, *41*, *42*, *44–48*, *59*
necessityStrictExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(necessityStrictExceedance)*, 36
necessityStrictUndervaluation, *7*, *9*, *11*, *13*, *15–17*, *25*, *26*, *28*, *33–36*, 37, *38*, *41*, *42*, *44–48*, *59*
necessityStrictUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(necessityStrictUndervaluation)*, 37
necessityUndervaluation, *7*, *9*, *11*, *13*, *15–17*, *25*, *26*, *28*, *33–37*, 38, *41*, *42*, *44–48*, *59*
necessityUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(necessityUndervaluation)*, 38

piecewiseLinearApproximation, *3*, *5–7*, *11–17*, *19*, *21*, *23–27*, *29–31*, 39, *42*, *44*, *51*, *53*, *56–58*
piecewiseLinearApproximation,FuzzyNumber-method *(piecewiseLinearApproximation)*, 39
PiecewiseLinearFuzzyNumber, *3*, *7*, *9*, *11*, *12*, *14*, *16*, *17*, *25–28*, *30*, *33–40*, 40, *41*, *42*, *44–49*, *59*

PiecewiseLinearFuzzyNumber-class, 41
plot, *3*, *5–7*, *9*, *11–17*, *19*, *21*, *23–31*, *33–38*, *40–42*, 42, *45–48*, *51*, *53–59*
plot,DiscontinuousFuzzyNumber,missing-method *(*plot*)*, 42
plot,FuzzyNumber,missing-method *(*plot*)*, 42
plot,PiecewiseLinearFuzzyNumber,missing-method *(*plot*)*, 42
plot,TrapezoidalFuzzyNumber,missing-method *(*plot*)*, 42
plot.default, *42–44*
possibilityExceedance, *7*, *9*, *11*, *13*, *15–17*, *25*, *27, 28*, *33–38*, *41, 42*, *44*, 45, *46–48*, *59*
possibilityExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(*possibilityExceedance*)*, 45
possibilityStrictExceedance, *7*, *9*, *11*, *13*, *15–17*, *25*, *27, 28*, *33–38*, *41, 42*, *44, 45*, 46, *47, 48*, *59*
possibilityStrictExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(*possibilityStrictExceedance*)*, 46
possibilityStrictUndervaluation, *7*, *9*, *11*, *13*, *15–17*, *25*, *27, 28*, *33–38*, *41, 42*, *44–46*, 47, *48*, *59*
possibilityStrictUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(*possibilityStrictUndervaluation*)*, 47
possibilityUndervaluation, *7*, *9*, *11*, *13*, *15–17*, *25*, *27, 28*, *33–38*, *41, 42*, *44–47*, 48, *59*
possibilityUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method *(*possibilityUndervaluation*)*, 48
PowerFuzzyNumber, *3*, *7*, *13*, *15–17*, *25*, *27*, *30*, *49*, 49, *50*
PowerFuzzyNumber-class, 49

show, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, 50, *51*, *53*, *56–58*
show,FuzzyNumber-method *(*show*)*, 50
splinefun, *8*
supp, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, 51, *53*, *56–58*
supp,FuzzyNumber-method *(*supp*)*, 51

trapezoidalApproximation, *3*, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, 52, *54*, *56–58*
trapezoidalApproximation,FuzzyNumber-method *(*trapezoidalApproximation*)*, 52
TrapezoidalFuzzyNumber, *3*, *6*, *11*, *14*, *16, 17*, *25*, *30*, *41*, *44*, *52, 53*, 53, *54, 55*
TrapezoidalFuzzyNumber-class, 54
TriangularFuzzyNumber, *6*, *11*, *14*, *16, 17*, *25*, *44*, *54*, 55

value, *3*, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, 56, *57, 58*
value,FuzzyNumber-method *(*value*)*, 56

weightedExpectedValue, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56*, 57, *58*
weightedExpectedValue,FuzzyNumber-method *(*weightedExpectedValue*)*, 57
width, *5–7*, *11–13*, *15–17*, *19*, *23–26*, *29–31*, *40*, *44*, *51*, *53*, *56, 57*, 58
width,FuzzyNumber-method *(*width*)*, 58