Package ‘FuzzyNumbers’

February 5, 2019

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 0.4-6

Date 2019-02-05

URL http://www.gagolewski.com/software/

BugReports http://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 6.1.1

Encoding UTF-8

NeedsCompilation no

Author Marek Gagolewski [aut, cre],
Jan Caha [ctb]

Maintainer Marek Gagolewski <gagolews@rexamine.com>

Repository CRAN

Date/Publication 2019-02-05 18:13:20 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 ................................. 20
distance ................................................................. 21
evaluate ................................................................. 22
expectedInterval ....................................................... 23
expectedValue .......................................................... 24
Extract ................................................................. 25
fapply ................................................................. 26
FuzzyNumber ........................................................... 27
FuzzyNumber-class ...................................................... 28
integrateAlpha .......................................................... 29
integrate_discont_val ............................................... 30
maximum ................................................................. 31
minimum ................................................................. 32
necessityExceedance .................................................... 33
necessityStrictExceedance .......................................... 34
necessityStrictUndervaluation ..................................... 35
necessityUndervaluation ............................................. 36
piecewiseLinearApproximation .................................... 37
PiecewiseLinearFuzzyNumber ....................................... 39
PiecewiseLinearFuzzyNumber-class ............................... 40
plot ................................................................. 41
possibilityExceedance ............................................... 43
possibilityStrictExceedance ........................................ 44
possibilityStrictUndervaluation .................................. 45
possibilityUndervaluation .......................................... 46
PowerFuzzyNumber ...................................................... 47
PowerFuzzyNumber-class ............................................. 48
show ................................................................. 49
supp ................................................................. 49
trapezoidalApproximation .......................................... 50
FuzzyNumbers-package

TrapezoidalFuzzyNumber ................................................................. 52
TrapezoidalFuzzyNumber-class ....................................................... 53
TriangularFuzzyNumber ................................................................. 53
value .................................................................................................. 54
weightedExpectedValue ................................................................. 55
width .................................................................................................. 56
^,PiecewiseLinearFuzzyNumber,numeric-method ............................... 57

Index 58

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


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


**alphacut**


---

**alphacut**  
*Compute Alpha-Cuts*

**Description**

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

**Usage**

```r
## 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 alpha cut bounds). If some elements in `alpha` are not in $[0,1]$, then `NA` is set.

**See Also**

Other `FuzzyNumber`-method: `Arithmetic, 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`

**Examples**

```r
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 \( \alpha - \text{Int}(A) := \left[ \int_0^1 \alpha A_L(\alpha) \, d\alpha, \int_0^1 \alpha A_U(\alpha) \, d\alpha \right] \).

**Usage**

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

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

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

```r
## 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, 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.TrapezoidalFuzzyNumber, expectedInterval, plot
ambiguity

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, 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: 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**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>a fuzzy number</td>
</tr>
<tr>
<td>...</td>
<td>additional arguments passed to alphaInterval</td>
</tr>
</tbody>
</table>

**Details**

The ambiguity is defined as

\[
\text{amb}(A) := \int_0^1 \alpha (A_U(\alpha) - A_L(\alpha)) \, d\alpha.
\]

**Value**

Returns a single numeric value.

**References**


**See Also**

Other FuzzyNumber-method: Arithmetic, 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)
Arithmetic

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, PiecewiseLinearFuzzyNumber-class, 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 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'
\( e_1 + e_2 \)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
\( e_1 + e_2 \) # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'numeric,FuzzyNumber'
\( e_1 - e_2 \) # \( e_2 \times (-1) + e_1 \)

## S4 method for signature 'TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber'
\( e_1 - e_2 \)

## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
\( e_1 - e_2 \)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
\( e_1 - e_2 \)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
\( e_1 - e_2 \) # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'FuzzyNumber,ANY'
\( e_1 - e_2 \) # \(-e_1\)

## S4 method for signature 'numeric,FuzzyNumber'
\( e_1 \times e_2 \) # \( e_2 \times e_1 \)

## S4 method for signature 'TrapezoidalFuzzyNumber,numeric'
\( e_1 \times e_2 \)

## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
\( e_1 \times e_2 \)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'
\( e_1 \times e_2 \) # calls as.PiecewiseLinearFuzzyNumber()

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
\( e_1 \times e_2 \)

## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'
\( e_1 / e_2 \)

## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
\( e_1 / e_2 \)
# Arithmetic

## S4 method for signature 'PiecewiseLinearFuzzyNumber,FuzzyNumber'

```r
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: `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: `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `*,PiecewiseLinearFuzzyNumber`, `numeric-method`, `alphaInterval`, `arctan`, `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
```r
## 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`, `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`, `PiecewiseLinearFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber`, `^`, `PiecewiseLinearFuzzyNumber`, `numeric-method`, `alphaInterval`, `as.character`
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

```r
## 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 object of class FuzzyNumber.

See Also

Other FuzzyNumber-method: Arithmetic, FuzzyNumber-class, FuzzyNumber, alphaInterval, alphacut, ambiguity, as.PiecewiseLinearFuzzyNumber, as.PowerFuzzyNumber, as.TrapezoidalFuzzyNumber, 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

Converting 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

```r
# 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, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber, numeric-method, alphaInterval, arctan2, as.PowerFuzzyNumber, as.TrapezoidalFuzzyNumber, as.character, expectedInterval, fapply, maximum, minimum, necessityExceedance, necessityStrictExceedance, necessityStrictUndervaluation,
as.PowerFuzzyNumber

necessityUndervaluation, plot, possibilityExceedance, possibilityStrictExceedance, possibilityStrictUndervaluation, possibilityUndervaluation

Other FuzzyNumber-method: Arithmetic, 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

```r
## 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`.
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

```r
## 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)
```
**convertAlpha**

**Argument**

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

**Value**

Returns an object of class TrapezoidalFuzzyNumber.

**See Also**

Other FuzzyNumber-method: Arithmetic, 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: PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval, as.PowerFuzzyNumber, as.character, expectedInterval

Other PiecewiseLinearFuzzyNumber-method: Arithmetic, PiecewiseLinearFuzzyNumber-class, 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

<table>
<thead>
<tr>
<th>f</th>
<th>a function defined on [x1,x2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>numeric vector of length 1; if longer, only the first element is used</td>
</tr>
<tr>
<td>x2</td>
<td>numeric vector of length 1; if longer, only the first element is used</td>
</tr>
</tbody>
</table>

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
Calculate the Core of a Fuzzy Number

Description

We have \( \text{core}(A) := [a_2, a_3] \). This gives the values that a fuzzy number necessarily represents.

Usage

```r
## 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`, `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

`Create 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

```r
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))
```
**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.

**Arguments**

- `a1` a number specifying left bound of the support
- `a2` a number specifying left bound of the core
- `a3` a number specifying right bound of the core
- `a4` a number specifying 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, distance, integrateAlpha, plot`
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, 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 $d^2_E(A,B) := \int_0^1 (A_L(\alpha)-B_L(\alpha))^2 \, d\alpha, \int_0^1 +(A_U(\alpha)-B_U(\alpha))^2 \, d\alpha$, see (Grzegorzewski, 1988).

Usage

```r
## 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>e1</code></td>
<td>a fuzzy number</td>
</tr>
<tr>
<td><code>e2</code></td>
<td>a fuzzy number</td>
</tr>
<tr>
<td>...</td>
<td>additional arguments passed to <code>integrate</code></td>
</tr>
<tr>
<td><code>type</code></td>
<td>one of &quot;Euclidean&quot;, &quot;EuclideanSquared&quot;</td>
</tr>
</tbody>
</table>

Details

The calculation are done using numerical integration.

Value

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

References


See Also

Other FuzzyNumber-method: Arithmetic, 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, integrateAlpha, plot

Description

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

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>object</code></td>
<td>a fuzzy numbers</td>
</tr>
<tr>
<td><code>x</code></td>
<td>numeric vector</td>
</tr>
</tbody>
</table>
expectedInterval

Value

Returns a numeric vector.

See Also

Other FuzzyNumber-method: Arithmetic, 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 $EI(A) := \left[ \int_0^1 A_L(\alpha) \, d\alpha, \int_0^1 A_U(\alpha) \, d\alpha \right]$, 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


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, 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, 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: PowerFuzzyNumber-class, PowerFuzzyNumber, alphaInterval, as.PowerFuzzyNumber, as.TrapezoidalFuzzyNumber, as.character

description

Calculate the Expected Value of a Fuzzy Number

Description

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

Usage

```r
## 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, 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 defuzzification: 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

```r
## 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.

Examples

```r
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

```r
## 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,PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber,^,PiecewiseLinearFuzzyNumber,numerics-method,alphaInterval, arctan2,as.PiecewiseLinearFuzzyNumber,as.PowerFuzzyNumber,as.TrapezoidalFuzzyNumber, as.character,expectedInterval,maximum,minimum,necessityExceedance,necessityStrictExceedance, necessityStrictUndervaluation,necessityUndervaluation,plot,possibilityExceedance, possibilityStrictExceedance,possibilityStrictUndervaluation,possibilityUndervaluation

Other extension_principle: `Arithmetic,^,PiecewiseLinearFuzzyNumber,numerics-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 specifying left bound of the support
- `a2`: a number specifying left bound of the core
- `a3`: a number specifying right bound of the core
- `a4`: a number specifying 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, 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
Description

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

$$
\mu(x) = \begin{cases} 
0 & \text{if } x < a_1, \\
left((x - a_1)/(a_2 - a_1)) & \text{if } a_1 \leq x < a_2, \\
1 & \text{if } a_2 \leq x \leq a_3, \\
right((x - a_3)/(a_4 - a_3)) & \text{if } a_3 < x \leq a_4, \\
10 & \text{if } a_4 < x,
\end{cases}
$$

where $a_1, a_2, a_3, a_4 \in R$, $a_1 \leq a_2 \leq a_3 \leq a_4$, $\left. \right.: [0,1] \rightarrow [0,1]$ is a nondecreasing function called the left side generator of $A$, and $\right. : [0,1] \rightarrow [0,1]$ 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 $\alpha$-cuts, $A(\alpha)$. We have $A(0) = [a_1,a_4]$ and

$$
A(\alpha) = [a_1 + (a_2 - a_1) \times \left. (\alpha), \\
a_3 + (a_4 - a_3) \times \right. (\alpha)]
$$

for $0 < \alpha \leq 1$, where $\left. : [0,1] \rightarrow [0,1]$ and $\right. : [0,1] \rightarrow [0,1]$ are, respectively, strictly increasing and decreasing functions satisfying $\left. (\alpha) = \inf \{x : \mu(x) \geq \alpha\}$ and $\right. (\alpha) = \sup \{x : \mu(x) \geq \alpha\}$.

Details

Please note that many algorithms that deal with fuzzy numbers often use $\alpha$-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 $[0,1] \rightarrow [0,1]$ that gives the lower alpha-cut bound.
- upper: A nonincreasing function $[0,1] \rightarrow [0,1]$ that gives the upper alpha-cut bound.
- left: A nondecreasing function $[0,1] \rightarrow [0,1]$ that gives the left side function.
- right: A nonincreasing function $[0,1] \rightarrow [0,1]$ that gives the right side function.
**integrateAlpha**

Child/sub classes

- TrapezoidalFuzzyNumber
- PiecewiseLinearFuzzyNumber
- PowerFuzzyNumber
- DiscontinuousFuzzyNumber

References


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, 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

```r
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**

```r
## 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, 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, 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.
maximum

Value

Returns the estimate of the integral.

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


See Also

Other min_max-operators: `minimum`

Other PiecewiseLinearFuzzyNumber-method: `Arithmetic,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 numbers

Description

Determines minimum fuzzy number based on two inputs.

Usage

```r
## 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


See Also

Other min_max-operators: maximum
Other PiecewiseLinearFuzzyNumber-method: Arithmetic, PiecewiseLinearFuzzyNumber-class, PiecewiseLinearFuzzyNumber-, numeric-method, alphaInterval, arctan2, as.PiecewiseLinearFuzzyNumber, as.PowerFuzzyNumber, as.TrapezoidalFuzzyNumber, as.character, expectedInterval, fapply, maximum, possibilityExceedance, possibilityStrictExceedance, possibilityStrictUndervaluation, possibilityUndervaluation, 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 = max(alphacut(x, 0)[1], alphacut(y, 0)[1])
max = max(alphacut(x, 0)[2], alphacut(y, 0)[2])
plot(x, col="red", ylim=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 = max(alphacut(x, 0)[1], alphacut(y, 0)[1])
max = max(alphacut(x, 0)[2], alphacut(y, 0)[2])
plot(x, col="red", ylim=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.
necessityStrictExceedance

Usage

```r
## 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


See Also

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

Other `PiecewiseLinearFuzzyNumber`-method: `Arithmetic`, `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

```r
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)
```

Description

Determines value of necessity of $e_1 > e_2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.
Usage

```r
# 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


See Also

Other comparison-operators: `necessityExceedance`, `necessityStrictUndervaluation`, `necessityUndervaluation`, `possibilityExceedance`, `possibilityStrictExceedance`, `possibilityStrictUndervaluation`, `possibilityUndervaluation`

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`, `necessityStrictUndervaluation`, `necessityUndervaluation`, `plot`, `possibilityExceedance`, `possibilityStrictExceedance`, `possibilityStrictUndervaluation`, `possibilityUndervaluation`

Examples

```r
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)
```

---

### Necessity of strict undervaluation

**Description**

Determines value of necessity of $e_1 < e_2$, the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.
Usage

```r
## 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


See Also

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

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`, `necessityUndervaluation`, `plot`, `possibilityExceedance`, `possibilityStrictExceedance`, `possibilityStrictUndervaluation`, `possibilityUndervaluation`

Examples

```r
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 \(e_1 <= e_2\), the result is from range \([0,1]\). Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.
Usage

```r
## 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


See Also

Other comparison-operators: `necessityExceedance`, `necessityStrictExceedance`, `necessityStrictUndervaluation`, `possibilityExceedance`, `possibilityStrictExceedance`, `possibilityStrictUndervaluation`

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

Examples

```r
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)
```

---

**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

```r
## 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

See Also

Other approximation: trapezoidalApproximation

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, plot, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

Examples

```r
(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**

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

**Arguments**

- `a1`: a number specifying left bound of the support
- `a2`: a number specifying left bound of the core
- `a3`: a number specifying right bound of the core
- `a4`: a number specifying 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]
PiecewiseLinearFuzzyNumber-class

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

See Also

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

Value

An object of class PiecewiseLinearFuzzyNumber.
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, PiecewiseLinearFuzzyNumber, ^, 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

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**

```r
## 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,
lab=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, 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, 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, distance, integrateAlpha

Examples

```r
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 $e_1 \geq e_2$, the result is from range $[0,1]$. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

**Usage**

```r
## 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 $e_2$ by $e_1$.
References


See Also

Other comparison-operators: necessityExceedance, necessityStrictExceedance, necessityStrictUndervaluation, necessityUndervaluation, possibilityStrictExceedance, possibilityStrictUndervaluation, possibilityUndervaluation

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, possibilityStrictExceedance, possibilityStrictUndervaluation, possibilityStrictUndervaluation

Examples

```r
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 \( e_1 > e_2 \), the result is from range \([0,1]\). Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

**Usage**

```r
## 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 \( e_2 \) by \( e_1 \).
**possibilityStrictUndervaluation**

References


See Also

Other comparison-operators: necessityExceedance, necessityStrictExceedance, necessityStrictUndervaluation, necessityUndervaluation, possibilityExceedance, possibilityStrictUndervaluation, possibilityUndervaluation

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, possibilityStrictUndervaluation, possibilityStrictUndervaluation, possibilityUndervaluation

Examples

```r
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 \( e_1 < e_2 \), the result is from range \([0,1]\). Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```r
## 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


See Also

Other comparison-operators: necessityExceedance, necessityStrictExceedance, necessityStrictUndervaluation, necessityUndervaluation, possibilityExceedance, possibilityStrictExceedance, possibilityUndervaluation

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, 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)

posibilityUndervaluation

Possibility of undervaluation

Description

Determines value of possibility of \( e_1 \leq e_2 \), the result is from range [0,1]. Value 0 indicates no fulfilment of the operator and 1 indicates complete fulfilment.

Usage

```r
## S4 method for signature
## 'PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber'
posibilityUndervaluation(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.
PowerFuzzyNumber

References

See Also
Other comparison-operators: necessityExceedance, necessityStrictExceedance, necessityStrictUndervaluation, necessityUndervaluation, possibilityExceedance, possibilityStrictExceedance, possibilityStrictUndervaluation
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

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 (Create 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 specifying left bound of the support
a2 a number specifying left bound of the core
a3 a number specifying right bound of the core
a4 a number specifying right bound of the support
p.left a positive number specifying the exponent for the left side
p.right a positive number specifying the exponent for the right side

Value
Object of class PowerFuzzyNumber
See Also

Other PowerFuzzyNumber-method: 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 $a_1 \leq a_2 \leq a_3 \leq a_4$, and parameters $p_{left}, p_{right}>0$, which determine exponents for the side functions.

Details

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

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

Slots

$a_1, a_2, a_3, a_4, lower, upper, left, right$: 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: PowerFuzzyNumber, alphaInterval, as.PowerFuzzyNumber, as.TrapezoidalFuzzyNumber, as.character, expectedInterval

Examples

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

Calculate the Support of a Fuzzy Number

Description

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

Usage

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

**Arguments**

- object: a fuzzy number

**Value**

Returns a numeric vector of length 2.

**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`, `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**

```r
## 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) ⊆ core(T(A)) and supp(T(A)) ⊆ 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


See Also

Other approximation: piecewiseLinearApproximation

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

```r
(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 specifying left bound of the support.
- `a2`: a number specifying left bound of the core.
- `a3`: a number specifying right bound of the core.
- `a4`: a number specifying 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)
value

Arguments

a1      a number specifying left bound of the support
amid   a number specifying the core
a4      a number specifying 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 defuzzify a fuzzy number.

Usage

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

Arguments

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

Details

The value of \( A \) (Delgado et al, 1998) is defined as \( \text{val}(A) := \int_0^1 \alpha (A_L(\alpha) + A_U(\alpha)) \, d\alpha \).

Value

Returns a single numeric value.

References

weightedExpectedValue

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, weightedExpectedValue, width

Other defuzzification: 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 defuzzify a fuzzy number.

For \( w = 0.5 \) we get the ordinary `expectedValue`.

**Usage**

```r
## 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, 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 defuzzification: 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

```r
## 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 \( \text{width}(A) := EI_U(A) - EI_L(A) \), where \( EI \) is the `expectedInterval`.

Value

Returns a single numeric value.

References


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`

Other characteristics: `ambiguity`, `expectedValue`, `value`, `weightedExpectedValue`
Description

For fuzzy numbers the equality of \( x^x = x^2 \) does not hold.

Usage

\[
\text{## S4 method for signature 'PiecewiseLinearFuzzyNumber,numeric'} \\
\text{e1} \ ^ \text{e2}
\]

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


See Also

Other extension_principle: Arithmetic, fapply
Other PiecewiseLinearFuzzyNumber-method: Arithmetic, 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

\[
x = \text{as.PiecewiseLinearFuzzyNumber}(\text{TriangularFuzzyNumber}(-2, 1, 9), \text{knot.n} = 2) \\
x^2 \\
x^3
\]
Index

*, PiecewiseLinearFuzzyNumber, FuzzyNumber-method (Arithmetic), 9
*, PiecewiseLinearFuzzyNumber, PiecewiseLinearFuzzyNumber-method (Arithmetic), 9
*, PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 9
*, TrapezoidalFuzzyNumber, numeric-method (Arithmetic), 9
+, PiecewiseLinearFuzzyNumber, PiecewiseLinearFuzzyNumber-method (Arithmetic), 9
+, PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 9
+, TrapezoidalFuzzyNumber, numeric-method (Arithmetic), 9
+, numeric, FuzzyNumber-method (Arithmetic), 9
+, numeric, FuzzyNumber-method (Arithmetic), 9
-, FuzzyNumber, ANY-method (Arithmetic), 9
-, PiecewiseLinearFuzzyNumber, FuzzyNumber-method (Arithmetic), 9
-, PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 9
-, TrapezoidalFuzzyNumber, numeric-method (Arithmetic), 9
-, numeric, FuzzyNumber-method (Arithmetic), 9
/, PiecewiseLinearFuzzyNumber, FuzzyNumber-method (Arithmetic), 9
/, PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 9
/, PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 9
[, DiscontinuousFuzzyNumber, character-method (Extract), 25

(alphaInterval, 5, 6, 7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 56
alphacut, FuzzyNumber, numeric-method (alphacut). 5
alphacut, FuzzyNumber, numeric-method (alphacut), 5
alphainterval, 5, 6, 7, 9, 11–17, 19, 22–27, 29–31, 33–37, 39–41, 43–57
alphainterval, FuzzyNumber-method (alphainterval), 6
alphainterval, PiecewiseLinearFuzzyNumber-method (alphainterval), 6
alphainterval, PowerFuzzyNumber-method (alphainterval), 6
ambiguity, 5, 6, 7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 56
ambiguity, FuzzyNumber-method (ambiguity), 7
approxfun, 8
approxinvert, 8, 18, 29
arctan2, 7, 8, 11, 13, 14, 16, 17, 24, 26, 31, 33–37, 40, 41, 43–47, 57
arctan2, PiecewiseLinearFuzzyNumber, PiecewiseLinearFuzzyNumber-method (arctan2), 8
arctan2, PiecewiseLinearFuzzyNumber, PiecewiseLinearFuzzyNumber-method (arctan2), 8
as.character, FuzzyNumber-method (as.character), 12
INDEX

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, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 56
as.FuzzyNumber, FuzzyNumber-method (as.FuzzyNumber), 13
as.FuzzyNumber, numeric-method (as.FuzzyNumber), 13
as.PiecewiseLinearFuzzyNumber, FuzzyNumber-method (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber, numeric-method (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber, PiecewiseLinearFuzzyNumber-method (as.PiecewiseLinearFuzzyNumber), 14
as.PiecewiseLinearFuzzyNumber, TrapezoidalFuzzyNumber-method (as.PiecewiseLinearFuzzyNumber), 14
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, FuzzyNumber-method (as.TrapezoidalFuzzyNumber), 16
as.TrapezoidalFuzzyNumber, numeric-method (as.TrapezoidalFuzzyNumber), 16

DiscontinuousFuzzyNumber, 3, 19, 19, 20–22, 29, 30, 43
DiscontinuousFuzzyNumber-class, 20
distance, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 56
distance, DiscontinuousFuzzyNumber, DiscontinuousFuzzyNumber-method (distance), 21
distance, DiscontinuousFuzzyNumber, FuzzyNumber-method (distance), 21
distance, FuzzyNumber, FuzzyNumber-method (distance), 21
evaluate, 5–7, 11–13, 15–17, 19, 22, 22, 24, 25, 27, 29, 30, 39, 43, 49–51, 55, 56
evaluate, FuzzyNumber, numeric-method (evaluate), 22
expectedInterval, FuzzyNumber-method (expectedInterval), 23
expectedInterval, PiecewiseLinearFuzzyNumber-method (expectedInterval), 23
expectedInterval, PowerFuzzyNumber-method (expectedInterval), 23
expectedInterval, TrapezoidalFuzzyNumber-method (expectedInterval), 23
expectedValue, 3, 5–7, 11–13, 15–17, 19, 22–24, 24, 27, 29, 30, 39, 43, 49–51, 55, 56
expectedValue,FuzzyNumber-method
(expectedValue), 24
Extract, 25

fapply, 3, 7, 9, 11, 13, 14, 16, 17, 24, 26, 31,
33–37, 40, 41, 43–47, 57

fapply,PiecewiseLinearFuzzyNumber,function-method
(fapply), 26
FuzzyNumber, 3, 5–8, 11–13, 15–19, 21–25,
27, 29, 30, 39, 40, 43, 48–51, 53,
55, 56

FuzzyNumbers-class, 28
FuzzyNumbers-package, 3

integrate, 22, 30
integrate_discont_val, 30, 30
integrateAlpha, 5–7, 11–13, 15–17, 19–25,
27, 29, 29, 38, 39, 43, 49–51, 55, 56

integrateAlpha,DiscontinuousFuzzyNumber,numeric-method
(integrateAlpha), 29
integrateAlpha,FuzzyNumber,numeric-method
(integrateAlpha), 29

maximum, 7, 9, 11, 13, 14, 16, 17, 24, 26, 31,
33–37, 40, 41, 43–47, 57

maximum,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(maximum), 31

minimum, 7, 9, 11, 13, 14, 16, 17, 24, 26, 31,
32, 34–37, 40, 41, 43–47, 57

minimum,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(minimum), 32

necessityExceedance, 7, 9, 11, 13, 14, 16,
17, 24, 26, 31, 33, 35–37, 40, 41,
43–47, 57

necessityExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(necessityExceedance), 33

necessityStrictExceedance, 7, 9, 11, 13,
14, 16, 17, 24, 26, 31, 33, 34, 36,
37, 40, 41, 43–47, 57

necessityStrictExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(necessityStrictExceedance), 34

necessityStrictUndervaluation, 7, 9, 11,
13, 14, 16, 17, 24, 26, 31, 33–35, 35,
37, 40, 41, 43–47, 57

necessityStrictUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(necessityStrictUndervaluation), 35

necessityUndervaluation, 7, 9, 11, 13,
15–17, 24, 26, 31, 33–36, 36, 40, 41,
43–47, 57

necessityUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(necessityUndervaluation), 36

piecewiseLinearApproximation, 3, 5–7,
11–13, 15–17, 19–25, 29, 30, 37, 41,
43, 49–51, 55, 56

piecewiseLinearApproximation,FuzzyNumber-method
(piecewiseLinearApproximation), 37

PiecewiseLinearFuzzyNumber, 3, 7, 9, 11,
12, 14, 16, 17, 24, 26, 29, 31, 33–39,
39, 40, 41, 43–48, 57

PiecewiseLinearFuzzyNumber-class, 40
plot, 3, 5–7, 9, 11–17, 19–27, 29–31, 33–37,
39–41, 41, 44–47, 49–57

plot,DiscontinuousFuzzyNumber,missing-method
(plot), 41
plot,FuzzyNumber,missing-method (plot), 41

plot,PiecewiseLinearFuzzyNumber,missing-method
(plot), 41
plot,TrapezoidalFuzzyNumber,missing-method
(plot), 41

plot,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(possibilityExceedance), 43
possibilityExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(possibilityExceedance), 43
possibilityStrictExceedance, 7, 9, 11, 13,
15–17, 24, 26, 31, 33–37, 40, 41, 43,
44, 44, 46, 47, 57

possibilityStrictExceedance,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(possibilityStrictExceedance), 44
possibilityStrictUndervaluation, 7, 9,
11, 13, 15–17, 24, 26, 31, 33–37, 40,
41, 43–45, 45, 47, 57

possibilityStrictUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(possibilityStrictUndervaluation), 45
possibilityUndervaluation, 7, 9, 11, 13,
15–17, 24, 26, 31, 33–37, 40, 41,
43–47, 57

possibilityUndervaluation,PiecewiseLinearFuzzyNumber,PiecewiseLinearFuzzyNumber-method
(possibilityUndervaluation), 46
INDEX

PowerFuzzyNumber. 3, 7, 13, 15–17, 24, 29, 47, 47, 48
PowerFuzzyNumber-class, 48
show, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49, 50, 51, 55, 56
show, FuzzyNumber-method (show), 49
splinefun, 8
supp, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49, 49, 51, 55, 56
supp, FuzzyNumber-method (supp), 49
trapezoidalApproximation, FuzzyNumber-method (trapezoidalApproximation), 50
TrapezoidalFuzzyNumber, 3, 6, 11, 14, 16, 17, 24, 29, 40, 43, 51, 52, 52, 53, 54
TrapezoidalFuzzyNumber-class, 53
TriangularFuzzyNumber, 6, 11, 14, 16, 17, 24, 43, 52, 53, 53
value, 3, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 54, 55, 56
value, FuzzyNumber-method (value), 54
weightedExpectedValue, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 55, 56
weightedExpectedValue, FuzzyNumber-method (weightedExpectedValue), 55
width, 5–7, 11–13, 15–17, 19, 22–25, 27, 29, 30, 39, 43, 49–51, 55, 56
width, FuzzyNumber-method (width), 56