Package: FuzzyStatTraEOO (via r-universe)

December 4, 2024

Type Package

Title Package 'FuzzyStatTra' in Encapsulated Object Oriented Programming

Version 1.0

Date 2022-12-12

Description The aim of the package is to contain the package 'FuzzyStatTra' in Encapsulated Object Oriented Programming using R6. 'FuzzyStatTra' contains Statistical Methods for Trapezoidal Fuzzy Numbers, whose aim is to provide some basic functions for doing statistical analysis with trapezoidal fuzzy numbers. For more details, you can visit the website of the SMIRE+CoDiRE (Statistical Methods with Imprecise Random Elements and Comparison of Distributions of Random Elements) Research Group (<https://bellman.ciencias.uniovi.es/smire+codire/>). The most related paper can be found in References. Now, those functions are organized in specific classes and methods. This object-based approach is an important step in making statistical computing more accessible to users.

Depends R (>= 4.1)

Imports R6

Suggests testthat, vdiffr

License LGPL (>= 3)

NeedsCompilation no

URL https://bellman.ciencias.uniovi.es/smire+codire/FuzzyStatTraRpackage.html

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Config/testthat/edition 3

Author Andrea García Cernuda [aut, cre], Asun Lubiano [aut] (<https://orcid.org/0000-0002-9847-5164>), Sara de la Rosa de Sáa [ctb]

Maintainer Andrea García Cernuda <uo270115@uniovi.es>

Repository CRAN

Date/Publication 2022-12-16 09:40:02 UTC

Contents

`FuzzyStatTraEOO-package`

Package 'FuzzyStatTra' in Encapsulated Object Oriented Programming

---

Description

'FuzzyStatTraEOO' is an open source package for R. The aim of the package is to contain the package 'FuzzyStatTra' in Encapsulated Object Oriented Programming using R6. 'FuzzyStatTra' contains Statistical Methods for Trapezoidal Fuzzy Numbers, whose aim is to provide some basic functions for doing statistical analysis with trapezoidal fuzzy numbers. For more details, you can visit the website of the SMIRE+CoDiRE (Statistical Methods with Imprecise Random Elements and Comparison of Distributions of Random Elements) Research Group (<https://bellman.ciencias.uniovi.es/smire+codire/>). The most related paper can be found in References. Now, those functions are organized in specific classes and methods. This object-based approach is an important step in making statistical computing more accessible to users.

Details

Package: FuzzyStatTraEOO
Type: Package
Version: 1.0
Date: 2022-12-12
License: LGPL (>= 3)

For a complete list of classes and their methods call `help(package="FuzzyStatTraEOO")`, call `??FuzzyStatTraEOO` or use the Index link below, at the end of this help window.

Author(s)

Andrea Garcia Cernuda <uo270115@uniovi.es>,
Asun Lubiano <lubiano@uniovi.es> and Sara de la Rosa de Saa.

Maintainer: Andrea Garcia Cernuda <uo270115@uniovi.es>

References

- Blanco-Fernandez, A.; Casals, R.M.; Colubi, A.; Corral, N.; Garcia-Barzana, M.; Gil, M.A.; Gonzalez-Rodriguez, G.; Lopez, M.T.; Lubiano, M.A.; Montenegro, M.; Ramos-Guajardo, A.B.; de la Rosa de Saa, S.; Sinova, B.: Random fuzzy sets: A mathematical tool to develop statistical fuzzy data analysis, Iranian Journal on Fuzzy Systems 10(2), 1-28 (2013)
- De la Rosa de Saa, S.; Gil, M.A.; Gonzalez-Rodriguez, G.; Lopez, M.T.; Lubiano M.A.: Fuzzy rating scale-based questionnaires and their statistical analysis, IEEE Transactions on Fuzzy Systems 23(1), 111-126 (2015)
- De la Rosa de Sáa, S.; Lubiano, M.A.; Sinova, B.; Filzmoser, P.; Gil, M.Á.: Location-free robust scale estimates for fuzzy data, IEEE Transactions on Fuzzy Systems 29(6), 1682-1694 (2021)
- De la Rosa de Saa, S.; Lubiano M.A.; Sinova, B.; Filzmoser, P.: Robust scale estimators for fuzzy data, Advances in Data Analysis and Classification 11(4), 731-758 (2017)
- Diamond, P.; Kloeden, P.: Metric spaces of fuzzy sets, Fuzzy Sets and Systems 35, 241-249 (1990)
- Gil, M.A.; Lubiano, M.A.; De la Rosa de Saa, S.; Sinova, B.: Analyzing data from a fuzzy rating scale-based questionnaire. A case study, Psicothema 27(2), 182-191 (2015)
- Lubiano, M.A.; De la Rosa de Saa, S.; Montenegro, M.; Sinova, B.; Gil, M.A.: Descriptive analysis of responses to items in questionnaires. Why not a fuzzy rating scale?, Information Sciences 360, 131-148 (2016)
- Lubiano, M.A.; Gil, M.A.: f-Inequality indices for fuzzy random variables, in Statistical Modeling, Analysis and Management of Fuzzy Data (Bertoluzza, C., Gil, M.A., Ralescu, D.A., Eds.), Physica-Verlag, 43-63 (2002)
- Lubiano, M.A.; Montenegro, M.; Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications, European Journal of Operational Research 251, 918-929 (2016)
- Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)
- Lubiano, M.A.; Salas, A.; Gil, M.Á.: A hypothesis testing-based discussion on the sensitivity of means of Fuzzy data with respect to data shape, Fuzzy Sets and Systems 328(1), 54-69 (2017)
- Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: A generalized L1-type metric between fuzzy numbers for an approach to central tendency of fuzzy data, Information Sciences 242, 22-34 (2013)
- Sinova, B.; De la Rosa de Saa, S.; Lubiano, M.A.; Gil, M.A.: An overview on the statistical central tendency for fuzzy datasets, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 23 (Suppl. 1), 105-132 (2015)

- Sinova, B.; Gil, M.A.; Colubi, A.; Van Aelst, S.: The median of a random fuzzy number. The 1-norm distance approach, Fuzzy Sets and Systems 200, 99-115 (2012)
- Sinova, B.; Gil, M.A.; Lopez, M.T.; Van Aelst, S.: A parameterized L2 metric between fuzzy numbers and its parameter interpretation, Fuzzy Sets and Systems 245, 101-115 (2014)
- Sinova, B.; Gil, M.A.; Van Aelst, S.: M-estimates of location for the robust central tendency of fuzzy data, IEEE Transactions on Fuzzy Systems 24(4), 945-956 (2016)

See Also

https://bellman.ciencias.uniovi.es/smire+codire/FuzzyStatTraRpackage.html

---

`FuzzyNumber` *R6 Class representing a 'FuzzyNumber'.*

Description

A 'FuzzyNumber' is an array of dimension nl x 3 x 1. It must be valid.

Methods

Public methods:

- `FuzzyNumber$new()`
- `FuzzyNumber$getAlphaLevels()`
- `FuzzyNumber$getInfimums()`
- `FuzzyNumber$getSupremums()`
- `FuzzyNumber$plot()`
- `FuzzyNumber$clone()`

Method `new()`**:** This method creates a valid 'FuzzyNumber' object with all its attributes set.

Usage:

```
FuzzyNumber$new(fnLevels = NA)
```

Arguments:

`fnLevels` is an array of dimension nl x 3 x 1 (general fuzzy number). nl is the number of considered α-levels and 3 is the number of columns of the array. The first column represents the number of considered α-levels, the second one represents their infimum values and the third and last column represents their supremum values.

Details: See examples.

Returns: The FuzzyNumber object created with all its attributes set if it is valid.

Examples:

```
# Example 1:
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3)))

# Example 2:
FuzzyNumber$new(array(c(0.0,1.0,1,2,4,3),dim=c(2,3)))
```

Method `getAlphaLevels()`**:** This method gives the 'alphaLevels' array of the 'FuzzyNumber'.

Usage:

```
FuzzyNumber$getAlphaLevels()
```

Details: See examples.

Returns: The array alphaLevels of the FuzzyNumber object.

Examples:

```
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getAlphaLevels()
```

Method `getInfimums()`**:** This method gives the 'imfimums' array of the 'FuzzyNumber'.

Usage:

```
FuzzyNumber$getInfimums()
```

Details: See examples.

Returns: The array imfimums of the FuzzyNumber object.

Examples:

```
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getInfimums()
```

Method `getSupremums()`**:** This method gives the 'supremums' array of the 'FuzzyNumber'.

Usage:

```
FuzzyNumber$getSupremums()
```

Details: See examples.

Returns: The array supremums of the FuzzyNumber object.

Examples:

```
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getSupremums()
```

Method `plot()`**:** This method shows in a graph the values of the alphaLevels, infimums and supremums attributes of the corresponding 'FuzzyNumber'.

Usage:

```
FuzzyNumber$plot(color = "grey")
```

Arguments:

`color` is the color of the lines representing the number to be shown in the graph. The default value is grey, other colors can be specified, the option palette() too.

Details: See examples.

Returns: a graph with the values of the alphaLevels, infimums and supremums attributes of the corresponding 'FuzzyNumber'.

Examples:

```
# Example 1:
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$plot()

# Example 2:
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 2, 1.7),dim=c(2,3))
)$plot("blue")

# Example 3:
Simulation$new()$simulCase1(1L)$transfTra()$getDimension(1L)$plot(palette())

# Example 4:
Simulation$new()$simulCase1(1L)$transfTra()$getDimension(1L)$plot(palette()[7])
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
FuzzyNumber$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method, comments are welcome to the above mentioned mail addresses.

Author(s)

Andrea Garcia Cernuda <uo270115@uniovi.es>

Examples

```
## ------------------------------------------------
## Method `FuzzyNumber$new`
## ------------------------------------------------

# Example 1:
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3)))

# Example 2:
FuzzyNumber$new(array(c(0.0,1.0,1,2,4,3),dim=c(2,3)))

## ------------------------------------------------
## Method `FuzzyNumber$getAlphaLevels`
## ------------------------------------------------

FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getAlphaLevels()

## ------------------------------------------------
## Method `FuzzyNumber$getInfimums`
```

```
## ------------------------------------------------

FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getInfimums()

## ------------------------------------------------
## Method `FuzzyNumber$getSupremums`
## ------------------------------------------------

FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$getSupremums()

## ------------------------------------------------
## Method `FuzzyNumber$plot`
## ------------------------------------------------

# Example 1:
FuzzyNumber$new(array(c(0.0,0.5,1.0,-1.5,-1.0,-1.0,2.0,1.5,1.0),dim=c(3,3))
)$plot()

# Example 2:
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 2, 1.7),dim=c(2,3))
)$plot("blue")

# Example 3:
Simulation$new()$simulCase1(1L)$transfTra()$getDimension(1L)$plot(palette())

# Example 4:
Simulation$new()$simulCase1(1L)$transfTra()$getDimension(1L)$plot(palette()[7])
```

FuzzyNumberList *'FuzzyNumberList' is a child class of 'StatList'.*

Description

'FuzzyNumberList' must contain valid 'FuzzyNumbers'. This class implements a version of the empty 'StatList' methods.

Super class

FuzzyStatTraEOO::StatList -> FuzzyNumberList

Methods

Public methods:

- FuzzyNumberList$new()
- FuzzyNumberList$dthetaphi()
- FuzzyNumberList$dwablphi()
- FuzzyNumberList$rho1()

- FuzzyNumberList$addFuzzyNumber()
- FuzzyNumberList$removeFuzzyNumber()
- FuzzyNumberList$getDimension()
- FuzzyNumberList$plot()
- FuzzyNumberList$getLength()
- FuzzyNumberList$clone()

Method `new()`**:** This method creates a 'FuzzyNumberList' object with the columns and dimensions attributes set where the 'FuzzyNumbers' must be valid.

Usage:

```
FuzzyNumberList$new(numbers = NA)
```

Arguments:

`numbers` is a list of dimension nl x 3 x n which contains n fuzzy numbers. nl is the number of considered α-levels and 3 is the number of columns of the list. The first column represents the number of considered α-levels, the second one represents their infimum values and the third and last column represents their supremum values.

Details: See examples.

Returns: The FuzzyNumberList object created with the columns and dimensions attributes set where the 'FuzzyNumbers' must be valid.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.0,-1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0, 3.0, 2.0,
1.0), dim = c(3, 3)))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2.0, 1.5), dim = c(2, 3)))))
```

Method `dthetaphi()`**:** This method calculates the mid/spr distance between the FuzzyNumbers contained in the current object and the one passed as parameter. See Blanco-Fernandez et al. (2013) [1].

Usage:

```
FuzzyNumberList$dthetaphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` FuzzyNumberList containing FuzzyNumbers characterized by means of nl α-levels each. The α-levels of the FuzzyNumberList s should coincide with the ones of the current FuzzyNumberList (the method checks this condition).

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance.

Details: See examples.

Returns: a matrix containing the mid/spr distances between the two previous mentioned FuzzyNumberLists. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$dthetaphi(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3)))),
1,5,1)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3)))))$dthetaphi(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))
), 1, 1, 1/3)

# Example 3:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(20L)
F=F$transfTra()
S=S$transfTra()
F$dthetaphi(S,1,5,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F$dthetaphi(S,2,1,1/3)

# Example 5:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F=F$transfTra()
S=S$transfTra(50L)
F$dthetaphi(S,2,1,1)
```

Method `dwablphi()`**:** This method calculates the (ϕ,θ)-wabl/ldev/rdev distance between the 'FuzzyNumbers' contained in two 'FuzzyNumberLists'. The method checks if the α-levels of all 'FuzzyNumbers' coincide. See Sinova et al. (2013) [3] and Sinova et al. (2016) [4].

Usage:

```
FuzzyNumberList$dwablphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

s FuzzyNumberList containing FuzzyNumbers characterized by means of nl α-levels each. The α-levels should coincide with ones of the other FuzzyNumberList (the method checks this condition).

a real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

b real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

theta real number > 0, by default theta=1. It is the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: a matrix containing the (ϕ,θ)-wabl/ldev/rdev distances between the two previous mentioned FuzzyNumberLists. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$dwablphi(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3))))),
1,5,1)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3))))$dwablphi(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))
), 1, 1, 1/3)

# Example 3:
F=Simulation$new()$simulCase1(3L)
S=Simulation$new()$simulCase1(4L)
F=F$transfTra()
S=S$transfTra()
F$dwablphi(S,2,1,1)

# Example 4:
```

```
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F$dwablphi(S)

# Example 5:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F=F$transfTra()
S=S$transfTra(50L)
F$dwablphi(S,2,1,1)
```

Method `rho1()`**:** This method calculates the 1-norm distance between the 'FuzzyNumbers' contained in two 'FuzzyNumberLists'. The method checks if the α-levels of all 'FuzzyNumbers' coincide. See Diamond and Kloeden. (1990) [2].

Usage:

```
FuzzyNumberList$rho1(s = NA)
```

Arguments:

`s` FuzzyNumberList containing FuzzyNumbers characterized by means of nl α-levels each. The method checks that the α-levels should coincide with ones of the other FuzzyNumberList.

Details: See examples.

Returns: a matrix containing the 1-norm distances between the two previous mentioned FuzzyNumberLists. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$rho1(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3))))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3)))))$rho1(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))))

# Example 3:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F=F$transfTra()
```

```
S=S$transfTra()
F$rho1(S)
S$rho1(F)

# Example 4:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F=F$transfTra()
S=S$transfTra(10L)
F$rho1(S)
S$rho1(F)
```

Method `addFuzzyNumber()`**:** This method adds a 'FuzzyNumber' to the current collection of fuzzy numbers. Therefore, the dimensions' field is increased in a unit.

Usage:

```
FuzzyNumberList$addFuzzyNumber(n = NA, verbose = TRUE)
```

Arguments:

`n` is the FuzzyNumber to be added to the current collection of fuzzy numbers.

`verbose` if TRUE the messages are written to the console unless the user actively decides to set verbose=FALSE.

Details: See examples.

Returns: NULL.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3)))))$addFuzzyNumber(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3)))
))$addFuzzyNumber( FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75,
1.5), dim = c(3, 3))))
```

Method `removeFuzzyNumber()`**:** This method removes a 'FuzzyNumber' to the current collection of fuzzy numbers. Therefore, the dimensions' field is decreased in a unit.

Usage:

```
FuzzyNumberList$removeFuzzyNumber(i = NA, verbose = TRUE)
```

Arguments:

`i` is the position of the FuzzyNumber to be removed in the current collection of fuzzy numbers.

`verbose` if TRUE the messages are written to the console unless the user actively decides to set verbose=FALSE.

Details: See examples.

Returns: NULL.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1, -0.5, 1.5, 1.25), dim = c(2, 3)))
))$removeFuzzyNumber(1L)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$removeFuzzyNumber(2L)
```

Method `getDimension()`**:** This method gives the number contained in the dimension passed as parameter when the dimension is greater than 0 and not greater than the dimensions of the 'FuzzyNumberList's' numbers array.

Usage:

```
FuzzyNumberList$getDimension(i = NA)
```

Arguments:

`i` is the dimension of the FuzzyNumber wanted to be retrieved.

Details: See examples.

Returns: The FuzzyNumber contained in the dimension passed as parameter or an error if the dimension is not valid.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getDimension(1L)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getDimension(2L)
```

Method `plot()`**:** This method shows in a graph the values of the attribute numbers of the corresponding 'FuzzyNumberList'.

Usage:

```
FuzzyNumberList$plot(color = "grey")
```

Arguments:

`color` is the color of the lines representing the numbers to be shown in the graph. The default value is grey, other colors can be specified, the option palette() too.

Details: See examples.

Returns: a graph with the values of the attribute numbers of the corresponding 'FuzzyNumberList'.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$plot()

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.85, 1.7), dim = c(2, 3))))
)$plot("blue")

# Example 3:
Simulation$new()$simulCase1(8L)$transfTra()$plot(palette())

# Example 4:
Simulation$new()$simulCase1(5L)$transfTra()$plot(palette()[2:6])
```

Method `getLength()`**:** This method returns the number of dimensions that are equivalent to the number of 'FuzzyNumbers' in the corresponding 'FuzzyNumberList'.

Usage:

```
FuzzyNumberList$getLength()
```

Details: See examples.

Returns: the number of dimensions that are equivalent to the number of 'FuzzyNumbers' in the corresponding 'FuzzyNumberList'.

Examples:

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getLength()

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3))))
)$getLength()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
FuzzyNumberList$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method, comments are welcome to the above mentioned mail addresses.

Author(s)

(s) Andrea Garcia Cernuda <uo270115@uniovi.es>, Asun Lubiano <lubiano@uniovi.es>, Sara de la Rosa de Saa

References

[1] Blanco-Fernandez, A.; Casals, R.M.; Colubi, A.; Corral, N.; Garcia-Barzana, M.; Gil, M.A.; Gonzalez-Rodriguez, G.; Lopez, M.T.; Lubiano, M.A.; Montenegro, M.; Ramos-Guajardo, A.B.; de la Rosa de Saa, S.; Sinova, B.: Random fuzzy sets: A mathematical tool to develop statistical fuzzy data analysis, Iranian Journal on Fuzzy Systems 10(2), 1-28 (2013)

[2] Diamond, P.; Kloeden, P.: Metric spaces of fuzzy sets, Fuzzy Sets and Systems 35, 241-249 (1990)

[3] Sinova, B.; de la Rosa de Saa, S.; Gil, M.A.: A generalized L1-type metric between fuzzy numbers for an approach to central tendency of fuzzy data, Information Sciences 242, 22-34 (2013)

[4] Sinova, B.; Gil, M.A.; Van Aelst, S.: M-estimates of location for the robust central tendency of fuzzy data, IEEE Transactions on Fuzzy Systems 24(4), 945-956 (2016)

Examples

```
## ------------------------------------------------
## Method `FuzzyNumberList$new`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.0,-1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0, 3.0, 2.0,
1.0), dim = c(3, 3)))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2.0, 1.5), dim = c(2, 3)))))

## ------------------------------------------------
## Method `FuzzyNumberList$dthetaphi`
## ------------------------------------------------
```

```
# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$dthetaphi(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3)))))),
1,5,1)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3)))))$dthetaphi(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))
), 1, 1, 1/3)

# Example 3:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(20L)
F=F$transfTra()
S=S$transfTra()
F$dthetaphi(S,1,5,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F$dthetaphi(S,2,1,1/3)

# Example 5:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F=F$transfTra()
S=S$transfTra(50L)
F$dthetaphi(S,2,1,1)

## ------------------------------------------------
## Method `FuzzyNumberList$dwablphi`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$dwablphi(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3)))))),
1,5,1)
```

```
# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3)))))$dwablphi(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))
), 1, 1, 1/3)

# Example 3:
F=Simulation$new()$simulCase1(3L)
S=Simulation$new()$simulCase1(4L)
F=F$transfTra()
S=S$transfTra()
F$dwablphi(S,2,1,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F$dwablphi(S)

# Example 5:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(10L)
F=F$transfTra()
S=S$transfTra(50L)
F$dwablphi(S,2,1,1)

## ------------------------------------------------
## Method `FuzzyNumberList$rho1`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3)))
))$rho1(
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3))))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.0, -1.0, 2.0, 1.5, 1.0), dim =
c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1.5, -1.25, -1.0, 3.0, 2.0,
1.0), dim = c(3, 3))), FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0, 1.0, 1.0, 2.5,
2.0, 1.5), dim = c(3, 3))),FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 0.5 , 1, 1.5,
3, 2.0, 2), dim = c(3, 3)))))$rho1(FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,1,1.25,1.5, 2, 1.75, 1.5), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1,-0.5,0, 1.5, 1.25, 1), dim = c(3, 3))))))
```

```
# Example 3:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F=F$transfTra()
S=S$transfTra()
F$rho1(S)
S$rho1(F)

# Example 4:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F=F$transfTra()
S=S$transfTra(10L)
F$rho1(S)
S$rho1(F)

## ------------------------------------------------
## Method `FuzzyNumberList$addFuzzyNumber`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3)))))$addFuzzyNumber(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))))

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3)))
))$addFuzzyNumber( FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75,
1.5), dim = c(3, 3))))

## ------------------------------------------------
## Method `FuzzyNumberList$removeFuzzyNumber`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1, -0.5, 1.5, 1.25), dim = c(2, 3)))
))$removeFuzzyNumber(1L)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$removeFuzzyNumber(2L)

## ------------------------------------------------
## Method `FuzzyNumberList$getDimension`
## ------------------------------------------------

# Example 1:
```

```
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getDimension(1L)

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getDimension(2L)

## ------------------------------------------------
## Method `FuzzyNumberList$plot`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$plot()

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.85, 1.7), dim = c(2, 3))))
)$plot("blue")

# Example 3:
Simulation$new()$simulCase1(8L)$transfTra()$plot(palette())

# Example 4:
Simulation$new()$simulCase1(5L)$transfTra()$plot(palette()[2:6])

## ------------------------------------------------
## Method `FuzzyNumberList$getLength`
## ------------------------------------------------

# Example 1:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, -1, -0.5, 0, 1.5, 1.25, 1), dim = c(3, 3))),
FuzzyNumber$new(array(c(0.0, 0.5, 1.0, 1, 1.25, 1.5, 2, 1.75, 1.5), dim = c(3, 3)))
))$getLength()

# Example 2:
FuzzyNumberList$new(c(
FuzzyNumber$new(array(c(0.0, 1.0, -1.5, -1.0, 2, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -1.0, -1.0, 1.5, 1.0), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, -0.5, 0, 1.5, 1), dim = c(2, 3))),
FuzzyNumber$new(array(c(0.0, 1.0, 1, 1.5, 1.5, 1.5), dim = c(2, 3))))
)$getLength()
```

M1 *69 trapezoidal fuzzy numbers.*

Description

A dataset containing 69 trapezoidal fuzzy numbers. The data correspond to the well-knows questionaire TIMSS-PIRLS2011. This questionnaire was adapted to allow a double-type response, namely, the original Likert and a fuzzy rating scale-based (to simplify, trapezoidal). The questionnaire was conducted on 69 fourth grade students from Colegio San Ignacio (Oviedo-Asturias, Spain). Trapezoidal fuzzy rating responses to the Question M1 "I like mathematics" are collected in this dataset.

Usage

```
M1
```

Format

A matrix with 69 rows and 4 columns:

inf0 infimum of support of the trapezoidal fuzzy numbers, real number

inf1 infimum of core of the trapezoidal fuzzy numbers, real number

sup1 supremum of core of the trapezoidal fuzzy numbers, real number

sup0 supremum of support of the trapezoidal fuzzy numbers, real number

Source

http://bellman.ciencias.uniovi.es/SMIRE/FuzzyRatingScaleQuestionnaire-SanIgnacio.html

References

[1] Gil, M.A.; Lubiano, M.A.; De la Rosa de Saa, S.; Sinova, B.: Analyzing data from a fuzzy rating scale-based questionnaire. A case study, Psicothema 27(2), 182-191 (2015)

[2] Lubiano, M.A.; De la Rosa de Saa, S.; Montenegro, M.; Sinova, B.; Gil, M.A.: Descriptive analysis of responses to items in questionnaires. Why not a fuzzy rating scale?, Information Sciences 360, 131-148 (2016)

[3] Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)

Examples

```
M1
```

M2 *69 trapezoidal fuzzy numbers.*

Description

A dataset containing 69 trapezoidal fuzzy numbers. The data correspond to the well-knows questionaire TIMSS-PIRLS2011. This questionnaire was adapted to allow a double-type response, namely, the original Likert and a fuzzy rating scale-based (to simplify, trapezoidal). The questionnaire was conducted on 69 fourth grade students from Colegio San Ignacio (Oviedo-Asturias, Spain). Trapezoidal fuzzy rating responses to the Question M2 "My teacher is easy to understand" are collected in this dataset.

Usage

```
M2
```

Format

A matrix with 69 rows and 4 columns:

inf0 infimum of support of the trapezoidal fuzzy numbers, real number

inf1 infimum of core of the trapezoidal fuzzy numbers, real number

sup1 supremum of core of the trapezoidal fuzzy numbers, real number

sup0 supremum of support of the trapezoidal fuzzy numbers, real number

Source

http://bellman.ciencias.uniovi.es/SMIRE/FuzzyRatingScaleQuestionnaire-SanIgnacio.html

References

[1] Gil, M.A.; Lubiano, M.A.; De la Rosa de Saa, S.; Sinova, B.: Analyzing data from a fuzzy rating scale-based questionnaire. A case study, Psicothema 27(2), 182-191 (2015)

[2] Lubiano, M.A.; De la Rosa de Saa, S.; Montenegro, M.; Sinova, B.; Gil, M.A.: Descriptive analysis of responses to items in questionnaires. Why not a fuzzy rating scale?, Information Sciences 360, 131-148 (2016)

[3] Lubiano, M.A.; Montenegro, M.; Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications, European Journal of Operational Research 251, 918-929 (2016)

[3] Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)

[4] Lubiano, M.A.; Salas, A.; Gil, M.Á.: A hypothesis testing-based discussion on the sensitivity of means of Fuzzy data with respect to data shape, Fuzzy Sets and Systems 328(1), 54-69 (2017)

Examples

```
M2
```

M3 *69 trapezoidal fuzzy numbers.*

Description

A dataset containing 69 trapezoidal fuzzy numbers. The data correspond to the well-knows questionaire TIMSS-PIRLS2011. This questionnaire was adapted to allow a double-type response, namely, the original Likert and a fuzzy rating scale-based (to simplify, trapezoidal). The questionnaire was conducted on 69 fourth grade students from Colegio San Ignacio (Oviedo-Asturias, Spain). Trapezoidal fuzzy rating responses to the Question M3 "Mathematics is harder for me than any other subject" are collected in this dataset.

Usage

```
M3
```

Format

A matrix with 69 rows and 4 columns:

inf0 infimum of support of the trapezoidal fuzzy numbers, real number

inf1 infimum of core of the trapezoidal fuzzy numbers, real number

sup1 supremum of core of the trapezoidal fuzzy numbers, real number

sup0 supremum of support of the trapezoidal fuzzy numbers, real number

Source

http://bellman.ciencias.uniovi.es/SMIRE/FuzzyRatingScaleQuestionnaire-SanIgnacio.html

References

[1] Gil, M.A.; Lubiano, M.A.; De la Rosa de Saa, S.; Sinova, B.: Analyzing data from a fuzzy rating scale-based questionnaire. A case study, Psicothema 27(2), 182-191 (2015)

[2] Lubiano, M.A.; De la Rosa de Saa, S.; Montenegro, M.; Sinova, B.; Gil, M.A.: Descriptive analysis of responses to items in questionnaires. Why not a fuzzy rating scale?, Information Sciences 360, 131-148 (2016)

[3] Lubiano, M.A.; Montenegro, M.; Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications, European Journal of Operational Research 251, 918-929 (2016)

[4] Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)

Examples

```
M3
```

S1 *69 trapezoidal fuzzy numbers.*

Description

A dataset containing 69 trapezoidal fuzzy numbers. The data correspond to the well-knows questionaire TIMSS-PIRLS2011. This questionnaire was adapted to allow a double-type response, namely, the original Likert and a fuzzy rating scale-based (to simplify, trapezoidal). The questionnaire was conducted on 69 fourth grade students from Colegio San Ignacio (Oviedo-Asturias, Spain). Trapezoidal fuzzy rating responses to the Question S1 "My teacher taught me to discover science in daily life" are collected in this dataset.

Usage

```
S1
```

Format

A matrix with 69 rows and 4 columns:

inf0 infimum of support of the trapezoidal fuzzy numbers, real number

inf1 infimum of core of the trapezoidal fuzzy numbers, real number

sup1 supremum of core of the trapezoidal fuzzy numbers, real number

sup0 supremum of support of the trapezoidal fuzzy numbers, real number

Source

http://bellman.ciencias.uniovi.es/SMIRE/FuzzyRatingScaleQuestionnaire-SanIgnacio.html

References

[1] Gil, M.A.; Lubiano, M.A.; De la Rosa de Saa, S.; Sinova, B.: Analyzing data from a fuzzy rating scale-based questionnaire. A case study, Psicothema 27(2), 182-191 (2015)

[2] Lubiano, M.A.; De la Rosa de Saa, S.; Montenegro, M.; Sinova, B.; Gil, M.A.: Descriptive analysis of responses to items in questionnaires. Why not a fuzzy rating scale?, Information Sciences 360, 131-148 (2016)

[3] Lubiano, M.A.; Montenegro, M.; Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications, European Journal of Operational Research 251, 918-929 (2016)

[4] Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)

[5] Lubiano, M.A.; Salas, A.; Gil, M.Á.: A hypothesis testing-based discussion on the sensitivity of means of Fuzzy data with respect to data shape, Fuzzy Sets and Systems 328(1), 54-69 (2017)

Examples

```
S1
```

`Simulation` *'Simulation' contains several methods to simulate 'TrapezoidalFuzzyNumberLists'.*

Description

Simulation contains 5 different methods that gives the user a 'TrapezoidalFuzzyNumberList'.

Methods

Public methods:

- `Simulation$simulCase1()`
- `Simulation$simulCase2()`
- `Simulation$simulCase3()`
- `Simulation$simulCase4()`
- `Simulation$simulFRSTra()`
- `Simulation$clone()`

Method `simulCase1()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' from a symmetric distribution and with independent components (for a detailed explanation of the simulation see Sinova et al. (2012) [3], namely, the Case 1 for noncontaminated samples).

Usage:

```
Simulation$simulCase1(n = NA)
```

Arguments:

`n` positive integer. It is the number of trapezoidal fuzzy numbers to be generated.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers. Each one is characterized by its four values inf0, inf1, sup1, sup0.

Examples:

```
Simulation$new()$simulCase1(10L)
```

Method `simulCase2()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' from a symmetric distribution and with dependent components (for a detailed explanation of the simulation see Sinova et al. (2012) [3], namely, the Case 2 for noncontaminated samples).

Usage:

```
Simulation$simulCase2(n = NA)
```

Arguments:

`n` positive integer. It is the number of trapezoidal fuzzy numbers to be generated.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers. Each one is characterized by its four values inf0, inf1, sup1, sup0.

Examples:

```
Simulation$new()$simulCase2(10L)
```

Method `simulCase3()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' from a asymmetric distribution and with independent components (for a detailed explanation of the simulation see Sinova et al. (2012) [4], namely, the Case 3 for noncontaminated samples).

Usage:

```
Simulation$simulCase3(n = NA)
```

Arguments:

`n` positive integer. It is the number of trapezoidal fuzzy numbers to be generated.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers. Each one is characterized by its four values inf0, inf1, sup1, sup0.

Examples:

```
Simulation$new()$simulCase3(10L)
```

Method `simulCase4()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' from a asymmetric distribution and with dependent components (for a detailed explanation of the simulation see Sinova et al. (2012) [4], namely, the Case 4 for noncontaminated samples).

Usage:

```
Simulation$simulCase4(n = NA)
```

Arguments:

`n` positive integer. It is the number of trapezoidal fuzzy numbers to be generated.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers. Each one is characterized by its four values inf0, inf1, sup1, sup0.

Examples:

```
Simulation$new()$simulCase4(10L)
```

Method `simulFRSTra()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' based on the fuzzy rating scale. They are simulated mimicking the human behavior, considering for it a finite mixture of three different procedures (for a detailed explanation of the simulation see De la Rosa de Saa et al. (2012) [1]), and generated in the interval [0,1].

Usage:

```
Simulation$simulFRSTra(n = NA, w1 = NA, w2 = NA, w3 = NA, p = NA, q = NA)
```

Arguments:

`n` positive integer. It is the number of trapezoidal fuzzy numbers to be generated.

`w1` real number in [0,1]. It should be fulfilled that w1+w2+w3=1.

`w2` real number in [0,1]. It should be fulfilled that w1+w2+w3=1.

`w3` real number in [0,1]. It should be fulfilled that w1+w2+w3=1.

`p` real number > 0. It is the first parameter of the beta distribution.

`q` real number > 0. It is the second parameter of the beta distribution.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers with values in the interval [0,1]. Each trapezoidal fuzzy rating response is characterized by its four values inf0, inf1, sup1, sup0.

Examples:

```
Simulation$new()$simulFRSTra(100L,0.05,0.35,0.6,2,1)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
Simulation$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method comments are welcome to the below mentioned mail addresses.

Author(s)

(s) Andrea Garcia Cernuda <uo270115@uniovi.es>, Asun Lubiano <lubiano@uniovi.es>, Sara de la Rosa de Saa

References

[1] De la Rosa de Saa, S.; Gil, M.A.; Gonzalez-Rodriguez, G.; Lopez, M.T.; Lubiano M.A.: Fuzzy rating scale-based questionnaires and their statistical analysis, IEEE Transactions on Fuzzy Systems 23(1), 111-126 (2015)

[2] Lubiano, M.A.; Salas, A.; Carleos, C.; De la Rosa de Sáa, S.; Gil, M.Á.: Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data, International Journal of Approximate Reasoning 88, 128-147 (2017)

[3] Sinova, B.; Gil, M.A.; Colubi, A.; Van Aelst, S.: The median of a random fuzzy number. The 1-norm distance approach, Fuzzy Sets and Systems 200, 99-115 (2012)

[4] Sinova, B.; Gil, M.A.; Van Aelst, S.: M-estimates of location for the robust central tendency of fuzzy data, IEEE Transactions on Fuzzy Systems 24(4), 945-956 (2016)

Examples

```
## ------------------------------------------------
## Method `Simulation$simulCase1`
## ------------------------------------------------

Simulation$new()$simulCase1(10L)

## ------------------------------------------------
## Method `Simulation$simulCase2`
## ------------------------------------------------

Simulation$new()$simulCase2(10L)

## ------------------------------------------------
## Method `Simulation$simulCase3`
## ------------------------------------------------

Simulation$new()$simulCase3(10L)

## ------------------------------------------------
## Method `Simulation$simulCase4`
## ------------------------------------------------

Simulation$new()$simulCase4(10L)

## ------------------------------------------------
## Method `Simulation$simulFRSTra`
## ------------------------------------------------

Simulation$new()$simulFRSTra(100L,0.05,0.35,0.6,2,1)
```

`StatList` *'StatList' is an "abstract class" representing a super class, useful for 'FuzzyNumberList' and 'TrapezoidalFuzzyNumberList' implementation.*

Description

'StatList' defines the common attributes and methods of 'FuzzyNumberList' and 'TrapezoidalFuzzyNumberList'. All methods are empty except for some attribute checking, the child classes are the ones that have to give the implementation for the empty methods.

Methods

Public methods:

- `StatList$new()`
- `StatList$dthetaphi()`
- `StatList$dwablphi()`

- StatList$rho1()
- StatList$plot()
- StatList$getLength()
- StatList$clone()

Method `new()`**:** This method warns the user that this class can not be initialized as it is abstract.

Usage:

```
StatList$new()
```

Returns: shows a message telling that this class can not be initialized.

Method `dthetaphi()`**:** This method calculates the mid/spr distance between the numbers contain in two 'StatLists'.

Usage:

```
StatList$dthetaphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` can be a FuzzyNumberList or a TrapezoidalFuzzyNumberList.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance.

Returns: a matrix containing the mid/spr distances between the two previous mentioned StatLists.

Method `dwablphi()`**:** This method calculates the (ϕ,θ)-wabl/ldev/rdev distance between the numbers contained in two 'StatLists'.

Usage:

```
StatList$dwablphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` can be a FuzzyNumberList or a TrapezoidalFuzzyNumberList.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Returns: a StatList containing the (ϕ,θ)-wabl/ldev/rdev distances between the two previous mentioned StatLists.

Method `rho1()`**:** This method calculates the 1-norm distance between the numbers contained in two 'StatLists'.

Usage:

```
StatList$rho1(s = NA)
```

Arguments:

`s` can be a FuzzyNumberList or a TrapezoidalFuzzyNumberList.

Returns: a StatList containing the 1-norm distances between the two previous mentioned StatLists.

Method `plot()`**:** This method shows in a graph the inner numbers of the corresponding 'StatList'.

Usage:

```
StatList$plot(color = "grey")
```

Arguments:

`color` is the color of the lines representing the numbers to be shown in the graph. The default value is grey, other colors can be specified, the option palette() too.

Returns: a graph with the inner numbers of the corresponding 'StatList' represented.

Method `getLength()`**:** This method returns the number of dimensions that are equivalent to the number of numbers in the corresponding 'StatList'.

Usage:

```
StatList$getLength()
```

Returns: the number of dimensions that are equivalent to the number of numbers in the corresponding 'StatList'.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
StatList$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In order to have the documentation completed, we had had to write the documentation of this class. Taking into account that this class is part of the software design and it cannot be initialized, all its documentation is not needed, in particular the new and clone methods, apart from the Usage: section of each method documentation. All its methods can be used and can be found in FuzzyNumberList's and TrapezoidalFuzzyNumberList's documentation.

We are working to improve this issue. In case you find (almost surely existing) bugs or have recommendations for improving the method, comments are welcome to the above mentioned mail addresses.

Author(s)

Andrea Garcia Cernuda <uo270115@uniovi.es>

TrapezoidalFuzzyNumber

R6 Class representing a 'TrapezoidalFuzzyNumber'.

Description

A 'TrapezoidalFuzzyNumber' is characterized by their four values inf0, inf1, sup1 and sup0. Its' values are checked in order to only provide a valid 'TrapezoidalFuzzyNumber'.

Methods

Public methods:

- `TrapezoidalFuzzyNumber$new()`
- `TrapezoidalFuzzyNumber$getInf0()`
- `TrapezoidalFuzzyNumber$getInf1()`
- `TrapezoidalFuzzyNumber$getSup1()`
- `TrapezoidalFuzzyNumber$getSup0()`
- `TrapezoidalFuzzyNumber$is_positive()`
- `TrapezoidalFuzzyNumber$plot()`
- `TrapezoidalFuzzyNumber$clone()`

Method `new()`**:** This method creates a valid 'TrapezoidalFuzzyNumber' object with all its attributes set.

Usage:

```
TrapezoidalFuzzyNumber$new(inf0 = NA, inf1 = NA, sup1 = NA, sup0 = NA)
```

Arguments:

`inf0` is a real number that corresponds to the infimum of support of the trapezoidal fuzzy number.

`inf1` is a real number that corresponds to the infimum of core of the trapezoidal fuzzy number

`sup1` is a real number that corresponds to the supremum of core of the trapezoidal fuzzy numbers

`sup0` is a real number that corresponds to the supremum of support of the trapezoidal fuzzy numbers

Details: See examples.

Returns: The TrapezoidalFuzzyNumber object created with all its attributes set if it is valid.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)
```

```
# Example 3:
TrapezoidalFuzzyNumber$new(-1,-1,2,3)

# Example 4:
TrapezoidalFuzzyNumber$new(1,2,3,3)
```

Method `getInf0()`**:** This method gives the inf0 attribute of the 'TrapezoidalFuzzyNumber'.

Usage:

```
TrapezoidalFuzzyNumber$getInf0()
```

Details: See examples.

Returns: The inf0 attribute of the TrapezoidalFuzzyNumber object.

Examples:

```
TrapezoidalFuzzyNumber$new(1,2,3,4)$getInf0()
```

Method `getInf1()`**:** This method gives the inf1 attribute of the 'TrapezoidalFuzzyNumber'.

Usage:

```
TrapezoidalFuzzyNumber$getInf1()
```

Details: See examples.

Returns: The inf1 attribute of the TrapezoidalFuzzyNumber object.

Examples:

```
TrapezoidalFuzzyNumber$new(1,2,3,4)$getInf1()
```

Method `getSup1()`**:** This method gives the sup1 attribute of the 'TrapezoidalFuzzyNumber'.

Usage:

```
TrapezoidalFuzzyNumber$getSup1()
```

Details: See examples.

Returns: The sup1 attribute of the TrapezoidalFuzzyNumber object.

Examples:

```
TrapezoidalFuzzyNumber$new(1,2,3,4)$getSup1()
```

Method `getSup0()`**:** This method gives the sup0 attribute of the 'TrapezoidalFuzzyNumber'.

Usage:

```
TrapezoidalFuzzyNumber$getSup0()
```

Details: See examples.

Returns: The sup0 attribute of the TrapezoidalFuzzyNumber object.

Examples:

```
TrapezoidalFuzzyNumber$new(1,2,3,4)$getSup0()
```

Method `is_positive()`**:** This method gives information whether the 'TrapezoidalFuzzyNumber' is positive regarding its attributes.

Usage:

```
TrapezoidalFuzzyNumber$is_positive()
```

Details: See examples.

Returns: TRUE whether the TrapezoidalFuzzyNumber object has all its attributes greater than -1, otherwise FALSE.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)$is_positive()

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)$is_positive()
```

Method `plot()`**:** This method shows in a graph the values of the corresponding 'TrapezoidalFuzzyNumber'.

Usage:

```
TrapezoidalFuzzyNumber$plot(color = "grey")
```

Arguments:

`color` is the color of the lines representing the number to be shown in the graph. The default value is grey, other colors can be specified, the option palette() too.

Details: See examples.

Returns: a graph with the values of the corresponding 'TrapezoidalFuzzyNumber'.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)$plot()

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)$plot("blue")

# Example 3:
TrapezoidalFuzzyNumber$new(0,0,0.5,3)$plot(palette())

# Example 4:
TrapezoidalFuzzyNumber$new(-8,-3.55,0,10)$plot(palette()[5])
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
TrapezoidalFuzzyNumber$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method, comments are welcome to the above mentioned mail addresses.

Author(s)

Andrea Garcia Cernuda <uo270115@uniovi.es>

Examples

```
## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$new`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)

# Example 3:
TrapezoidalFuzzyNumber$new(-1,-1,2,3)

# Example 4:
TrapezoidalFuzzyNumber$new(1,2,3,3)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$getInf0`
## ------------------------------------------------

TrapezoidalFuzzyNumber$new(1,2,3,4)$getInf0()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$getInf1`
## ------------------------------------------------

TrapezoidalFuzzyNumber$new(1,2,3,4)$getInf1()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$getSup1`
## ------------------------------------------------

TrapezoidalFuzzyNumber$new(1,2,3,4)$getSup1()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$getSup0`
## ------------------------------------------------

TrapezoidalFuzzyNumber$new(1,2,3,4)$getSup0()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumber$is_positive`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)$is_positive()

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)$is_positive()

## ------------------------------------------------
```

```
## Method `TrapezoidalFuzzyNumber$plot`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumber$new(1,2,3,4)$plot()

# Example 2:
TrapezoidalFuzzyNumber$new(-8,-6,-4,-2)$plot("blue")

# Example 3:
TrapezoidalFuzzyNumber$new(0,0,0.5,3)$plot(palette())

# Example 4:
TrapezoidalFuzzyNumber$new(-8,-3.55,0,10)$plot(palette()[5])
```

`TrapezoidalFuzzyNumberList`

'TrapezoidalFuzzyNumberList' is a child class of 'StatList'.

Description

'TrapezoidalFuzzyNumberList' must contain valid 'TrapezoidalFuzzyNumbers'. This class implements a version of the empty 'StatList' methods.

Super class

`FuzzyStatTraEOO::StatList` -> `TrapezoidalFuzzyNumberList`

Methods

Public methods:

- `TrapezoidalFuzzyNumberList$new()`
- `TrapezoidalFuzzyNumberList$add()`
- `TrapezoidalFuzzyNumberList$dthetaphi()`
- `TrapezoidalFuzzyNumberList$dwablphi()`
- `TrapezoidalFuzzyNumberList$gsi()`
- `TrapezoidalFuzzyNumberList$hyperI()`
- `TrapezoidalFuzzyNumberList$mEstimator()`
- `TrapezoidalFuzzyNumberList$mdd()`
- `TrapezoidalFuzzyNumberList$mean()`
- `TrapezoidalFuzzyNumberList$median1Norm()`
- `TrapezoidalFuzzyNumberList$medianWabl()`
- `TrapezoidalFuzzyNumberList$qn()`
- `TrapezoidalFuzzyNumberList$rho1()`
- `TrapezoidalFuzzyNumberList$sn()`
- `TrapezoidalFuzzyNumberList$tn()`

- TrapezoidalFuzzyNumberList$transfTra()
- TrapezoidalFuzzyNumberList$var()
- TrapezoidalFuzzyNumberList$wablphi()
- TrapezoidalFuzzyNumberList$addTrapezoidalFuzzyNumber()
- TrapezoidalFuzzyNumberList$removeTrapezoidalFuzzyNumber()
- TrapezoidalFuzzyNumberList$getDimension()
- TrapezoidalFuzzyNumberList$plot()
- TrapezoidalFuzzyNumberList$getLength()
- TrapezoidalFuzzyNumberList$clone()

Method `new()`**:** This method creates a 'TrapezoidalFuzzyNumberList' object with all the attributes set if the 'TrapezoidalFuzzyNumbers' are valid.

Usage:

```
TrapezoidalFuzzyNumberList$new(numbers = NA)
```

Arguments:

`numbers` is a list which contains n TrapezoidalFuzzyNumbers.

Details: See examples.

Returns: The TrapezoidalFuzzyNumberList object created with all attributes set if the 'TrapezoidalFuzzyNumbers' are valid.

Examples:

```
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1, 2, 3, 4),
TrapezoidalFuzzyNumber$new(-8, -6, -4, -2),
TrapezoidalFuzzyNumber$new(-1, -1, 2, 3),
TrapezoidalFuzzyNumber$new(1, 2, 3, 3)))
```

Method `add()`**:** This method calculates the scale measure Average Distance Deviation (ADD) of a 'TrapezoidalFuzzyNumberList' with respect to a 'TrapezoidalFuzzyNumberList' or with respect to a 'FuzzyNumberList' containing a unique valid fuzzy number. The employed metric in the calculation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance. See De la Rosa de Saa et al. (2017) [2].

Usage:

```
TrapezoidalFuzzyNumberList$add(s = NA, type = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` is a TrapezoidalFuzzyNumberList containing a unique valid TrapezoidalFuzzyNumber or it is a FuzzyNumberList containing a unique valid FuzzyNumber.

`type` positive integer 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the scale measure ADD, which is a real number. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3)))),1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3)))),2L,2,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(5,6,7,8))),3L,1,1,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
S=F$mean()
F$add(S,1L)

# Example 5:
F=Simulation$new()$simulCase1(100L)
S=F$median1Norm()
F$add(S,2L,2,1,1)

# Example 6:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(1L)
F$add(U,2L)

# Example 7:
F=Simulation$new()$simulCase2(10L)
U=F$transfTra()
F$add(U,2L)

# Example 8:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(2L)
F$add(U,2L)
```

Method `dthetaphi()`**:** This method calculates the mid/spr distance between the 'TrapezoidalFuzzyNumbers' contained in the current object and the one passed as parameter. See Lubiano et al. (2016) [5].

Usage:

```
TrapezoidalFuzzyNumberList$dthetaphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` TrapezoidalFuzzyNumberList containing valid TrapezoidalFuzzyNumbers characterized by their four values inf0, inf1, sup1, sup0.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance.

Details: See examples.

Returns: a matrix containing the mid/spr distances between the two previous mentioned TrapezoidalFuzzyNumberLists.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dthetaphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dthetaphi(
TrapezoidalFuzzyNumberList$new( c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))),1,1,1)

# Example 3:
F=Simulation$new()$simulCase1(6L)
S=Simulation$new()$simulCase1(8L)
F$dthetaphi(S,1,5,1)
```

Method `dwablphi()`**:** This method calculates the (ϕ,θ)-wabl/ldev/rdev distance between the 'TrapezoidalFuzzyNumbers' contained in two 'TrapezoidalFuzzyNumberLists'. See Sinova et al. (2013) [6] and Sinova et al. (2016) [10].

Usage:

```
TrapezoidalFuzzyNumberList$dwablphi(s = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` TrapezoidalFuzzyNumberList containing valid TrapezoidalFuzzyNumbers characterized by their four values inf0, inf1, sup1, sup0.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: a matrix containing the (ϕ,θ)-wabl/ldev/rdev distances between the two previous mentioned TrapezoidalFuzzyNumberLists.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dwablphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dwablphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))),5,1,1)

# Example 3:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(20L)
F$dwablphi(S)
```

Method `gsi()`**:** This method calculates the Gini-Simpson diversity index for a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList'. See De la Rosa de Saa et al. (2015) [1].

Usage:

```
TrapezoidalFuzzyNumberList$gsi()
```

Details: See examples.

Returns: the Gini-Simpson diversity index, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$gsi()

# Example 2:
F=Simulation$new()$simulCase1(50L)
F$gsi()
```

Method `hyperI()`**:** This method calculates the hyperbolic inequality index for a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList'. The method checks if all 'TrapezoidalFuzzyNumbers' are positive. See De la Rosa de Saa et al. (2015) [1] and Lubiano and Gil (2002) [4].

Usage:

```
TrapezoidalFuzzyNumberList$hyperI(c = 0, verbose = TRUE)
```

Arguments:

`c` number in [0,0.5]. The c*100 of the hyperbolic inequality index.

`verbose` if TRUE the messages are written to the console unless the user actively decides to set verbose=FALSE.

Details: See examples.

Returns: the hyperbolic inequality index, which is a real number. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$hyperI()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$hyperI(0.5)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,4,6,8)))$hyperI()

# Example 4:
F=Simulation$new()$simulFRSTra(100L,0.05,0.35,0.6,2,1)
F$hyperI()

# Example 5:
F=Simulation$new()$simulCase2(10L)
F$hyperI(0.5)
```

Method `mEstimator()`**:** This method calculates the M-estimator of scale with loss method given in a 'TrapezoidalFuzzyNumberList' containing 'TrapezoidalFuzzyNumbers'. For computing the M-estimator, a method called "iterative reweighting" is used. The employed metric in the M-equation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

Usage:

```
TrapezoidalFuzzyNumberList$mEstimator(
  f = NA,
  estInitial = NA,
  delta = NA,
  epsilon = NA,
  type = NA,
  a = 1,
  b = 1,
  theta = 1
)
```

Arguments:

`f` is the name of the loss function. It can be "Huber", "Tukey" or "Cauchy".

`estInitial` real number > 0.

`delta` real number in (0,1). It is present in the f-equation.

`epsilon` real number > 0. It is the tolerance allowed in the algorithm.

`type` positive integer 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the value of the M-estimator of scale, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Huber",0.321,0.5,10^(-5),
1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Tukey",0.123,0.5,10^(-5),
2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Cauchy",0.123,0.5,10^(-5),
3L,0.75,0.5,1)

# Example 4:
F=Simulation$new()$simulCase1(100L)
U=F$median1Norm()
estInitial=F$mdd(U,1L)
delta=0.5
epsilon=10^(-5)
F$mEstimator("Huber",estInitial,delta,epsilon,1L)
```

Method `mdd()`**:** This method calculates the scale measure Median Distance Deviation (MDD) of a 'TrapezoidalFuzzyNumberList' with respect to a 'TrapezoidalFuzzyNumberList' or with respect to a 'FuzzyNumberList' with a unique valid fuzzy number. The employed metric in the calculation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance. See De la Rosa de Saa et al. (2015) [2] and De la Rosa de Saa et al. (2021) [3].

Usage:

```
TrapezoidalFuzzyNumberList$mdd(s = NA, type = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`s` is a TrapezoidalFuzzyNumberList containing a unique TrapezoidalFuzzyNumber or it is a FuzzyNumberList containing a unique FuzzyNumber.

`type` positive integer 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or in the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the scale measure MDD, which is a real number.If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3))))),1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3))))),2L,2,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(5,6,7,8))),3L,1,1,1)

# Example 4:
F=Simulation$new()$simulCase3(10L)
U=F$mean()
F$mdd(U,3L,1,2,1)

# Example 5:
F=Simulation$new()$simulCase2(10L)
U=F$median1Norm()
F$mdd(U,2L)
```

```
# Example 6:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(1L)
F$mdd(U,2L)

# Example 7:
F=Simulation$new()$simulCase2(10L)
U=F$transfTra()
F$mdd(U,2L)

# Example 8:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(2L)
F$mdd(U,2L)
```

Method `mean()`**:** Given a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList', the method calculates the Aumann-type mean of these numbers (which is a 'TrapezoidalFuzzyNumber' too). See Sinova et al. (2015) [7].

Usage:

```
TrapezoidalFuzzyNumberList$mean()
```

Details: See examples.

Returns: the Aumann-type mean, given as a TrapezoidalFuzzyNumber contained in a TrapezoidalFuzzyNumberList.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mean()

# Example 2:
TrapezoidalFuzzyNumberList$new(
c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$mean()

# Example 3:
F=Simulation$new()$simulCase1(100L)
F$mean()
```

Method `median1Norm()`**:** Given a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList', the method calculates the 1-norm median of these numbers, characterized by means of nl equidistant α-levels (by default nl=101), including always the 0 and 1 levels, with their infimum and supremum values. See Sinova et al. (2012) [8].

Usage:

```
TrapezoidalFuzzyNumberList$median1Norm(nl = 101L)
```

Arguments:

`nl` integer greater or equal to 2, by default nl=101. It indicates the number of desired α-levels for characterizing the 1-norm median.

Details: See examples.

Returns: the 1-norm median, given in form of a FuzzyNumber contained in a FuzzyNumberList.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$median1Norm()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$median1Norm(200L)

# Example 3:
F=Simulation$new()$simulCase1(10L)
F$median1Norm(200L)
```

Method `medianWabl()`**:** Given a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList', the method calculates the ϕ-wabl/ldev/rdev median of these numbers, characterized by means of nl equidistant α-levels (by default nl=101), including always the 0 and 1 levels, with their infimum and supremum values. See Sinova et al. (2013) [6] and Sinova et al. (2016) [10].

Usage:

```
TrapezoidalFuzzyNumberList$medianWabl(nl = 101L, a = 1, b = 1)
```

Arguments:

`nl` integer greater or equal to 2, by default nl=101. It indicates the number of desired α-levels for characterizing the ϕ-wabl/ldev/rdev median.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

Details: See examples.

Returns: the ϕ-wabl/ldev/rdev median in form of a FuzzyNUmber given in a FuzzyNumberList.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl(3L)
```

```
# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl(3L,2.2,2.8)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$medianWabl(3L)
```

Method `qn()`**:** This method calculates scale measure Qn for a matrix of 'TrapezoidalFuzzyNumbers' contained in the current 'TrapezoidalFuzzyNumber'. The employed metric in the calculation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance. See De la Rosa de Saa et al. (2021) [3].

Usage:

```
TrapezoidalFuzzyNumberList$qn(type = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`type` integer number that can be 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the scale measure Qn, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(2L,5,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(3L,1,1,1)

# Example 4:
```

```
F=Simulation$new()$simulCase1(10L)
F$qn(3L,1,1,1)
```

Method `rho1()`**:** This method calculates the 1-norm distance between the 'TrapezoidalFuzzyNumbers' contained in two 'TrapezoidalFuzzyNumberLists'.

Usage:

```
TrapezoidalFuzzyNumberList$rho1(s = NA)
```

Arguments:

`s` TrapezoidalFuzzyNumberList containing valid TrapezoidalFuzzyNumbers characterized by their four values inf0, inf1, sup1, sup0.

Details: See examples.

Returns: a matrix containing the 1-norm distances between the two previous mentioned TrapezoidalFuzzyNumberLists.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$rho1(TrapezoidalFuzzyNumberList$new(
c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F$rho1(S)
S$rho1(F)
```

Method `sn()`**:** This method calculates scale measure Sn for a matrix of 'TrapezoidalFuzzyNumbers' contained in the current 'TrapezoidalFuzzyNumber'. The employed metric in the calculation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance. See De la Rosa de Saa et al. (2021) [3].

Usage:

```
TrapezoidalFuzzyNumberList$sn(type = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`type` integer number that can be 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the scale measure Sn, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(3L,5,1,0.5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$sn(2L,5,1,0.5)
```

Method `tn()`**:** This method calculates scale measure Tn for a matrix of 'TrapezoidalFuzzyNumbers' contained in the current 'TrapezoidalFuzzyNumber'. The employed metric in the calculation can be the 1-norm distance, the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance. See De la Rosa de Saa et al. (2021) [3].

Usage:

```
TrapezoidalFuzzyNumberList$tn(type = NA, a = 1, b = 1, theta = 1)
```

Arguments:

`type` integer number that can be 1, 2 or 3: if type==1, the 1-norm distance will be considered in the calculation of the measure ADD. If type==2, the mid/spr distance will be considered. By contrast, if type==3, the (ϕ,θ)-wabl/ldev/rdev distance will be used.

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1] in the mid/spr distance or the (ϕ,θ)-wabl/ldev/rdev distance.

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance and the weight of the ldev and rdev in the (ϕ,θ)-wabl/ldev/rdev distance.

Details: See examples.

Returns: the scale measure Tn, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
```

```
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(3L,5,1,0.5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$tn(1L)
```

Method `transfTra()`**:** This method transforms a 'TrapezoidalFuzzyNumberList' containing valid 'TrapezoidalFuzzyNumbers' characterized by their four values inf0, inf1, sup1, sup0 into a 'FuzzyNumberList' containing these same amount of fuzzy numbers, characterized by means of nl equidistant α-levels each (by default nl=101).

Usage:

```
TrapezoidalFuzzyNumberList$transfTra(nl = 101L)
```

Arguments:

`nl` integer greater or equal to 2, by default nl=101. It indicates the number of desired α-levels for characterizing the trapezoidal fuzzy numbers.

Details: See examples.

Returns: a FuzzyNumberList containing the transformed TrapezoidalFuzzyNumbers into FuzzyNumbers.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra(3L)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra(10L)

# Example 4:
F=Simulation$new()$simulCase3(10L)
F$transfTra(200L)
```

Method `var()`**:** Given a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList', the method calculates the variance of these numbers with respect to the mid/spr distance. See De la Rosa de Saa et al. (2017) [2].

Usage:

```
TrapezoidalFuzzyNumberList$var(a = 1, b = 1, theta = 1)
```

Arguments:

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`theta` real number > 0, by default theta=1. It is the weight of the spread in the mid/spr distance.

Details: See examples.

Returns: the variance of the sample with respect to the mid/spr distance, which is a real number.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var(1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var(1/3,1/3,1/5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$var(1,1,1)
```

Method `wablphi()`**:** Given a sample of 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList', the method calculates the ϕ-wabl value for each of these numbers. See Sinova et al. (2014) [9].

Usage:

```
TrapezoidalFuzzyNumberList$wablphi(a = 1, b = 1)
```

Arguments:

`a` real number > 0, by default a=1. It is the first parameter of a beta distribution which corresponds to a weighting measure on [0,1].

`b` real number > 0, by default b=1. It is the second parameter of a beta distribution which corresponds to a weighting measure on [0,1].

Details: See examples.

Returns: a vector giving the ϕ-wabl values of each TrapezoidalFuzzyNumber.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi(2,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi(2.2,1.1)

# Example 4:
F=Simulation$new()$simulCase4(60L)
F$wablphi(2,1)
```

Method `addTrapezoidalFuzzyNumber()`**:** This method adds a 'TrapezoidalFuzzyNumber' to the current collection inside the current 'TrapezoidalFuzzyNumberList'. Therefore, the dimensions' field is increased in a unit.

Usage:

```
TrapezoidalFuzzyNumberList$addTrapezoidalFuzzyNumber(n = NA, verbose = TRUE)
```

Arguments:

`n` is the TrapezoidalFuzzyNumber to be added to the current collection inside the current TrapezoidalFuzzyNumberList.

`verbose` if TRUE the messages are written to the console unless the user actively decides to set verbose=FALSE.

Details: See examples.

Returns: nothing.

Examples:

```
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4))
)$addTrapezoidalFuzzyNumber(TrapezoidalFuzzyNumber$new(3,4,5,6))
```

Method `removeTrapezoidalFuzzyNumber()`**:** This method removes a 'TrapezoidalFuzzyNumber' to the current collection inside the current 'TrapezoidalFuzzyNumberList'. Therefore, the dimensions' field is decreased in a unit.

Usage:

```
TrapezoidalFuzzyNumberList$removeTrapezoidalFuzzyNumber(i = NA, verbose = TRUE)
```

Arguments:

`i` is the position of the TrapezoidalFuzzyNumber to be removed in the current collection inside the current TrapezoidalFuzzyNumberList.

`verbose` if TRUE the messages are written to the console unless the user actively decides to set verbose=FALSE.

Details: See examples.

Returns: nothing.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$removeTrapezoidalFuzzyNumber(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$removeTrapezoidalFuzzyNumber(2L)
```

Method `getDimension()`**:** This method gives the number contained in the dimension passed as parameter when the dimension is greater than 0 and not greater than the dimensions of the TrapezoidalFuzzyNumberList's numbers array.

Usage:

```
TrapezoidalFuzzyNumberList$getDimension(i = NA)
```

Arguments:

`i` is the dimension of the TrapezoidalFuzzyNumber wanted to be retrieved.

Details: See examples.

Returns: The TrapezoidalFuzzyNumber contained in the dimension passed as parameter or an error if the dimension is not valid.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$getDimension(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$getDimension(2L)
```

Method `plot()`**:** This method shows in a graph the values of the attribute numbers of the corresponding 'TrapezoidalFuzzyNumberList'.

Usage:

```
TrapezoidalFuzzyNumberList$plot(color = "grey")
```

Arguments:

`color` is the color of the lines representing the numbers to be shown in the graph. The default value is grey, other colors can be specified, the option palette() too.

Details: See examples.

Returns: a graph with the values of the attribute numbers of the corresponding 'TrapezoidalFuzzyNumberList'.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,5)))$plot()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4)))$plot()

# Example 3:
Simulation$new()$simulCase1(8L)$plot(palette())

# Example 4:
Simulation$new()$simulCase1(5L)$plot(palette()[2:6])
```

Method `getLength()`**:** This method returns the number of dimensions that are equivalent to the number of 'TrapezoidalFuzzyNumbers' in the corresponding 'TrapezoidalFuzzyNumberList'.

Usage:

```
TrapezoidalFuzzyNumberList$getLength()
```

Details: See examples.

Returns: the number of dimensions that are equivalent to the number of 'TrapezoidalFuzzyNumbers' in the corresponding 'TrapezoidalFuzzyNumberList'.

Examples:

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4))
)$getLength()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4)))$getLength()

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4),TrapezoidalFuzzyNumber$new(1,2,3,4))
)$getLength()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
TrapezoidalFuzzyNumberList$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method, comments are welcome to the above mentioned mail addresses.

Author(s)

(s) Andrea Garcia Cernuda <uo270115@uniovi.es>, Asun Lubiano <lubiano@uniovi.es>, Sara de la Rosa de Saa

References

[1] De la Rosa de Saa, S.; Gil, M.A.; Gonzalez-Rodriguez, G.; Lopez, M.T.; Lubiano M.A.: Fuzzy rating scale-based questionnaires and their statistical analysis, IEEE Transactions on Fuzzy Systems 23(1), 111-126 (2015)

[2] De la Rosa de Saa, S.; Lubiano M.A.; Sinova, B.; Filzmoser, P.: Robust scale estimators for fuzzy data, Advances in Data Analysis and Classification 11(4), 731-758 (2017)

[3] De la Rosa de Sáa, S.; Lubiano, M.A.; Sinova, B.; Filzmoser, P.; Gil, M.Á.: Location-free robust scale estimates for fuzzy data, IEEE Transactions on Fuzzy Systems 29(6), 1682-1694 (2021)

[4] Lubiano, M.A.; Gil, M.A.: f-Inequality indices for fuzzy random variables, in Statistical Modeling, Analysis and Management of Fuzzy Data (Bertoluzza, C., Gil, M.A., Ralescu, D.A., Eds.), Physica-Verlag, 43-63 (2002)

[5] Lubiano, M.A.; Montenegro, M.; Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications, European Journal of Operational Research 251, 918-929 (2016)

[6] Sinova, B.; De la Rosa de Saa, S.; Gil, M.A.: A generalized L1-type metric between fuzzy numbers for an approach to central tendency of fuzzy data, Information Sciences 242, 22-34 (2013)

[7] Sinova, B.; De la Rosa de Saa, S.; Lubiano, M.A.; Gil, M.A.: An overview on the statistical central tendency for fuzzy datasets, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 23 (Suppl. 1), 105-132 (2015)

[8] Sinova, B.; Gil, M.A.; Colubi, A.; Van Aelst, S.: The median of a random fuzzy number. The 1-norm distance approach, Fuzzy Sets and Systems 200, 99-115 (2012)

[9] Sinova, B.; Gil, M.A.; Lopez, M.T.; Van Aelst, S.: A parameterized L2 metric between fuzzy numbers and its parameter interpretation, Fuzzy Sets and Systems 245, 101-115 (2014)

[10] Sinova, B.; Gil, M.A.; Van Aelst, S.: M-estimates of location for the robust central tendency of fuzzy data, IEEE Transactions on Fuzzy Systems 24(4), 945-956 (2016)

Examples

```
## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$new`
## ------------------------------------------------

TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1, 2, 3, 4),
TrapezoidalFuzzyNumber$new(-8, -6, -4, -2),
TrapezoidalFuzzyNumber$new(-1, -1, 2, 3),
TrapezoidalFuzzyNumber$new(1, 2, 3, 3)))

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$add`
## ------------------------------------------------

# Example 1:
```

```
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3))))),1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3))))),2L,2,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$add(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(5,6,7,8))),3L,1,1,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
S=F$mean()
F$add(S,1L)

# Example 5:
F=Simulation$new()$simulCase1(100L)
S=F$median1Norm()
F$add(S,2L,2,1,1)

# Example 6:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(1L)
F$add(U,2L)

# Example 7:
F=Simulation$new()$simulCase2(10L)
U=F$transfTra()
F$add(U,2L)

# Example 8:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(2L)
F$add(U,2L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$dthetaphi`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dthetaphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
```

```
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dthetaphi(
TrapezoidalFuzzyNumberList$new( c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))),1,1,1)

# Example 3:
F=Simulation$new()$simulCase1(6L)
S=Simulation$new()$simulCase1(8L)
F$dthetaphi(S,1,5,1)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$dwablphi`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dwablphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$dwablphi(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))),5,1,1)

# Example 3:
F=Simulation$new()$simulCase1(10L)
S=Simulation$new()$simulCase1(20L)
F$dwablphi(S)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$gsi`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$gsi()

# Example 2:
F=Simulation$new()$simulCase1(50L)
F$gsi()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$hyperI`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$hyperI()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$hyperI(0.5)
```

```
# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,4,6,8)))$hyperI()

# Example 4:
F=Simulation$new()$simulFRSTra(100L,0.05,0.35,0.6,2,1)
F$hyperI()

# Example 5:
F=Simulation$new()$simulCase2(10L)
F$hyperI(0.5)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$mEstimator`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Huber",0.321,0.5,10^(-5),
1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Tukey",0.123,0.5,10^(-5),
2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mEstimator("Cauchy",0.123,0.5,10^(-5),
3L,0.75,0.5,1)

# Example 4:
F=Simulation$new()$simulCase1(100L)
U=F$median1Norm()
estInitial=F$mdd(U,1L)
delta=0.5
epsilon=10^(-5)
F$mEstimator("Huber",estInitial,delta,epsilon,1L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$mdd`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3))))),1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
```

```
FuzzyNumberList$new(c(FuzzyNumber$new(array(c(0.0, 0.5, 1.0,-1.5,-1.25,-1.0,
3.0, 2.0, 1.0), dim = c(3, 3)))),2L,2,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(0,0.1,0.2,0.3),
TrapezoidalFuzzyNumber$new(1,2,3,4),TrapezoidalFuzzyNumber$new(2,3,4,5)))$mdd(
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(5,6,7,8))),3L,1,1,1)

# Example 4:
F=Simulation$new()$simulCase3(10L)
U=F$mean()
F$mdd(U,3L,1,2,1)

# Example 5:
F=Simulation$new()$simulCase2(10L)
U=F$median1Norm()
F$mdd(U,2L)

# Example 6:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(1L)
F$mdd(U,2L)

# Example 7:
F=Simulation$new()$simulCase2(10L)
U=F$transfTra()
F$mdd(U,2L)

# Example 8:
F=Simulation$new()$simulCase2(10L)
U=Simulation$new()$simulCase2(2L)
F$mdd(U,2L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$mean`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$mean()

# Example 2:
TrapezoidalFuzzyNumberList$new(
c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$mean()

# Example 3:
F=Simulation$new()$simulCase1(100L)
F$mean()

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$median1Norm`
## ------------------------------------------------
```

```
# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$median1Norm()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$median1Norm(200L)

# Example 3:
F=Simulation$new()$simulCase1(10L)
F$median1Norm(200L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$medianWabl`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl(3L)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$medianWabl(3L,2.2,2.8)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$medianWabl(3L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$qn`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(2L,5,1,1)

# Example 3:
```

```
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$qn(3L,1,1,1)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$qn(3L,1,1,1)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$rho1`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1)))$rho1(TrapezoidalFuzzyNumberList$new(
c(TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58))))

# Example 2:
F=Simulation$new()$simulCase1(4L)
S=Simulation$new()$simulCase1(5L)
F$rho1(S)
S$rho1(F)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$sn`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$sn(3L,5,1,0.5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$sn(2L,5,1,0.5)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$tn`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
```

```
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(2L,1,1,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$tn(3L,5,1,0.5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$tn(1L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$transfTra`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra(3L)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$transfTra(10L)

# Example 4:
F=Simulation$new()$simulCase3(10L)
F$transfTra(200L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$var`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var(1,1,1)
```

```
# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$var(1/3,1/3,1/5)

# Example 4:
F=Simulation$new()$simulCase1(10L)
F$var(1,1,1)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$wablphi`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi(2,1)

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),TrapezoidalFuzzyNumber$new(1.5,2,3.75,4),
TrapezoidalFuzzyNumber$new(-4.2,-3.6,-2,-1.58)))$wablphi(2.2,1.1)

# Example 4:
F=Simulation$new()$simulCase4(60L)
F$wablphi(2,1)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$addTrapezoidalFuzzyNumber`
## ------------------------------------------------

TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4))
)$addTrapezoidalFuzzyNumber(TrapezoidalFuzzyNumber$new(3,4,5,6))

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$removeTrapezoidalFuzzyNumber`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$removeTrapezoidalFuzzyNumber(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$removeTrapezoidalFuzzyNumber(2L)

## ------------------------------------------------
```

```
## Method `TrapezoidalFuzzyNumberList$getDimension`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$getDimension(1L)

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,4)))$getDimension(2L)

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$plot`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4),
TrapezoidalFuzzyNumber$new(2,3,4,5)))$plot()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4)))$plot()

# Example 3:
Simulation$new()$simulCase1(8L)$plot(palette())

# Example 4:
Simulation$new()$simulCase1(5L)$plot(palette()[2:6])

## ------------------------------------------------
## Method `TrapezoidalFuzzyNumberList$getLength`
## ------------------------------------------------

# Example 1:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(1,2,3,4))
)$getLength()

# Example 2:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4)))$getLength()

# Example 3:
TrapezoidalFuzzyNumberList$new(c(TrapezoidalFuzzyNumber$new(-4,-3,-2,-1),
TrapezoidalFuzzyNumber$new(-6,0,1,4),TrapezoidalFuzzyNumber$new(1,2,3,4))
)$getLength()
```

Utils *'Utils' to convert real data into the corresponding 'StatList'.*

Description

'Utils' contain an auxiliary method that perform the conversion of an archive with rda extension (R Data File), that contains real data, to the corresponding 'TrapezoidalFuzzyNumberList'.

Methods

Public methods:

- `Utils$convertTra()`
- `Utils$clone()`

Method `convertTra()`**:** This method generates n 'TrapezoidalFuzzyNumbers' contained in a 'TrapezoidalFuzzyNumberList' obtained from the rows and columns of R Data File. If the data contains any NA value, the row will be deleted as a 'TrapezpidalFuzzyNumber' have to be created with double values.

Usage:

```
Utils$convertTra(d = NA)
```

Arguments:

`d` is the R Data File already loaded in the environment with data("example"). If the user wants to use M1, M2, M3 or S1, they are already loaded in the package environment through the archive data.R.

Details: See examples.

Returns: a TrapezoidalFuzzyNumberList with n TrapezoidalFuzzyNumbers. Each one is characterized by its four values inf0, inf1, sup1, sup0. The TrapezoidalFuzzyNumbers are obtained from the rows and columns of R Data File. If the body's method inner conditions are not met, NA will be returned.

Examples:

```
# Example 1:
Utils$new()$convertTra(M1)

# Example 2:
Utils$new()$convertTra(M2)

# Example 3:
Utils$new()$convertTra(M3)

# Example 4:
Utils$new()$convertTra(S1)

# Example 5:
m=as.data.frame(matrix(c(NA, 1, 2, NA, 3, 2,2,NA,1,3,NA,NA,6,4,NA,NA),ncol=4))
Utils$new()$convertTra(m)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
Utils$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

In case you find (almost surely existing) bugs or have recommendations for improving the method comments are welcome to the below mentioned mail addresses.

Author(s)

Andrea Garcia Cernuda <uo270115@uniovi.es>

Examples

```
## ------------------------------------------------
## Method `Utils$convertTra`
## ------------------------------------------------

# Example 1:
Utils$new()$convertTra(M1)

# Example 2:
Utils$new()$convertTra(M2)

# Example 3:
Utils$new()$convertTra(M3)

# Example 4:
Utils$new()$convertTra(S1)

# Example 5:
m=as.data.frame(matrix(c(NA, 1, 2, NA, 3, 2,2,NA,1,3,NA,NA,6,4,NA,NA),ncol=4))
Utils$new()$convertTra(m)
```

Index