Package: FuzzyImputationTest (via r-universe)

December 20, 2024

Type Package

Title Imputation Procedures and Quality Tests for Fuzzy Data

Version 0.3.8

Description Special procedures for the imputation of missing fuzzy numbers are still underdeveloped. The goal of the package is to provide the new d-imputation method (DIMP for short, Romaniuk, M. and Grzegorzewski, P. (2023) ``Fuzzy Data Imputation with DIMP and FGAIN'' RB/23/2023) and covert some classical ones applied in R packages ('missForest','miceRanger','knn') for use with fuzzy datasets. Additionally, specially tailored benchmarking tests are provided to check and compare these imputation procedures with fuzzy datasets.

License GPL-3

NeedsCompilation yes

Imports stats, methods, FuzzySimRes, FuzzyNumbers, missForest, miceRanger, VIM, utils

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Author Maciej Romaniuk [cre, aut] (<https://orcid.org/0000-0001-9649-396X>)

Maintainer Maciej Romaniuk <mroman@ibspan.waw.pl>

Repository CRAN

Date/Publication 2024-12-20 04:50:02 UTC

Config/pak/sysreqs cmake make libicu-dev libssl-dev libx11-dev zlib1g-dev

Contents

`ApplyStatisticalTests` *Statistical epistemic tests the imputed values.*

Description

`ApplyStatisticalTests` applies the epistemic goodness-of-fit test for fuzzy data to check the quality of the imputed values.

Usage

```
ApplyStatisticalTests(
  trueData,
  imputedData,
  imputedMask,
  trapezoidal = TRUE,
  cutsNumber = 100,
  K = 50,
  ...
)
```

Arguments

trueData	Name of the input matrix (or data frame) with the true values of the variables.
imputedData	Name of the input matrix (or data frame) with the imputed values.
imputedMask	Matrix (or data frame) with logical values where TRUE indicates the cells with the imputed values.
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
cutsNumber	Number of cuts for the epistemic bootstrap tests.
K	Value of K for the res epistemic test.
...	Additional parameters passed to other functions.

Details

The procedure applies three types of the epistemic goodness-of-fit Kolmogorov-Smirnov tests (avs - averaging statistic, ms - multi-statistic, res - resampling algorithm) from the FuzzySimRes package to check the quality of the imputed values. To do this, three subsamples are used:

- true - the dataset trueData without imputed values vs the values from the same dataset that are then imputed,
- imputed - the dataset trueData without imputed values vs only the imputed values from imputedData,
- parts - only the imputed values from the dataset trueData vs their counterparts from imputedData.

To assess the respective imputation quality, p-values for true and imputed should be close to each other, and in the case of parts, they should exceed the selected significance level.

All of the input datasets can be given as matrices or data frames. The statistical tests are performed only for the input values that are proper fuzzy numbers (triangular or trapezoidal ones).

Value

The output is given as a matrix (the rows are related to various types of the test and subsamples, the columns - to the variables plus the overall mean).

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
```

```
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values (with possible repetitions!)

matrix1DImp <- FuzzyImputation(matrix1NA,method="dimp",checkFuzzy=TRUE)

# find cells with NAs

matrix1Mask <- is.na(matrix1NA)

# apply statistical epistemic bootstrap tests

ApplyStatisticalTests(matrix1,matrix1DImp,matrix1Mask,cutsNumber = 100, K=10)
```

`CalculateFuzzyMeasures` *Calculation of the fuzzy measures for the imputed values.*

Description

'CalculateFuzzyMeasures' calculates the various types of fuzzy measures between two datasets – the true and the imputed one.

Usage

```
CalculateFuzzyMeasures(
  trueData,
  imputedData,
  imputedMask,
  trapezoidal = TRUE,
  ...
)
```

Arguments

trueData	Name of the input matrix (or data frame) with the true values of the variables.
imputedData	Name of the input matrix (or data frame) with the imputed values.
imputedMask	Matrix (or data frame) with logical values where TRUE indicates the cells with the imputed values.
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
...	Additional parameters passed to other functions.

Details

The procedure calculates different types of the distance measures (Euclidean - the Euclidean measure, AHD - the AHD measure, HSD - the HSD measure) between two datasets - the first one with true values (set by trueData), and the second one (specified by imputedData) with the imputed variables. Only the truly imputed values are taken into account for these calculations. To properly distinguish the real values with their imputed counterparts, the additional matrix imputedMask should be provided. In this matrix, the logical value TRUE points out the cells with the imputed values. Otherwise, FALSE should be used.

All of the input datasets can be given as matrices or data frames.

Value

The output is given as a matrix (the rows are related to various types of the errors, the columns - to the variables and the overall mean).

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values
```

```
head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values

matrix1DImp <- ImputationDimp(matrix1NA)

# find cells with NAs

matrix1Mask <- is.na(matrix1NA)

# calculate fuzzy measures for the imputed values

CalculateFuzzyMeasures(matrix1,matrix1DImp,matrix1Mask,trapezoidal=TRUE)
```

`ErrorMatrix` *Calculation of the errors for the imputed values.*

Description

'ErrorMatrix' calculates the various types of the errors between two datasets – the true and the imputed one.

Usage

```
ErrorMatrix(trueData, imputedData, imputedMask, trapezoidal = TRUE, ...)
```

Arguments

`trueData`	Name of the input matrix (or data frame, or list) with the true values of the variables.
`imputedData`	Name of the input matrix (or data frame) with the imputed values.
`imputedMask`	Matrix (or data frame) with logical values where `TRUE` indicates the cells with the imputed values.
`trapezoidal`	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
`...`	Additional parameters passed to other functions.

Details

The procedure calculates different types of the errors (MAE - the mean absolute error, WMA - the weighted mean absolute error, MSE - the mean squared error, WMSE - the weighted mean squared error, NRMSE - the normalized root mean squared error) between two datasets - the first one with true values (set by `trueData`), and the second one (specified by `imputedData`) with the imputed variables. To properly distinguish the real values with their imputed counterparts, the additional matrix `imputedMask` should be provided. In this matrix, the logical value `TRUE` points out the cells with the imputed values. Otherwise, `FALSE` should be used.

All of the input datasets can be given as matrices or data frames.

Value

The output is given as a matrix (the rows are related to various types of the errors, the columns - to the variables).

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values

matrix1DImp <- ImputationDimp(matrix1NA)
```

```
# find cells with NAs

matrix1Mask <- is.na(matrix1NA)

# calculate errors for the imputed values

ErrorMatrix(matrix1,matrix1DImp,matrix1Mask)
```

FuzzifyMatrix *Fuzzyfing the crisp values.*

Description

'FuzzifyMatrix' converts real-valued variables into fuzzy numbers.

Usage

```
FuzzifyMatrix(
  crispMatrix,
  coreFactor = 0.2,
  supportFactor = 0.2,
  trapezoidal = TRUE,
  varNames = colnames(crispMatrix),
  ...
)
```

Arguments

crispMatrix	Name of the input matrix (or data frame) with real-valued variables to fuzzify.
coreFactor	Value used as the multiplier for the left/right end of the interval of the uniform distribution applied to randomly generated increments of the core.
supportFactor	Value used as the multiplier for the left/right end of the interval of the uniform distribution applied to randomly generated increments of the support.
trapezoidal	Logical value that indicates if trapezoidal (or triangular, otherwise) fuzzy numbers should be generated.
varNames	Names of the input variables.
...	Additional parameters passed to other functions.

Details

The procedure generates trapezoidal fuzzy numbers (when the default trapezoidal=TRUE is set) or triangular ones (for trapezoidal=FALSE) based on the real-valued data from the given matrix or the data frame. To do this, for each variable the standard deviation is calculated. Then, the left

and right increments of the core (in the case of trapezoidal fuzzy numbers) are randomly generated using the original value plus/minus two random values from the uniform distribution on the interval [0,coreFactor*(standard deviation)]. In the case of triangular fuzzy numbers, the cores are equal to the original real values. In the same manner, the left and right increments of the support are randomly generated with two random values from the uniform distribution on the interval [0,supportFactor*(standard deviation)].

Value

The output is given as a matrix with three (in the case of triangular fuzzy numbers) or four (for trapezoidal fuzzy numbers) columns for each input variable.

Examples

```
# set seed for the random generator

set.seed(12345)

# let's look at the beginning of the iris dataset (four numeric variables)

head(iris[,1:4])

# and fuzzify these variables

fuzzyOutput <- FuzzifyMatrix(iris[,1:4])

head(fuzzyOutput)
```

`FuzzyImputation` *Main method to impute fuzzy values.*

Description

'FuzzyImputation' imputes (i.e., replaces missing values) fuzzy numbers using various methods.

Usage

```
FuzzyImputation(
  dataToImpute,
  method = "dimp",
  trapezoidal = TRUE,
  checkFuzzy = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

dataToImpute	Name of the input matrix (data frame or list) of fuzzy numbers with some NAs.
method	Name of the imputation method (possible values: dimp,missForest,miceRanger,knn).
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
checkFuzzy	If TRUE is set, after each imputation, the output values are checked if they are proper fuzzy numbers. If there are some improper fuzzy numbers, they are removed, and the imputation procedure is repeated.
verbose	If TRUE is set, the current simulation number is printed.
...	Additional parameters that are passed to the imputation procedure.

Details

The procedure randomly imputes missing values (NAs) with suitable data in the case of a data frame (or a matrix, or a list) consisting of fuzzy numbers (triangular fuzzy numbers if trapezoidal=FALSE is set, or trapezoidal ones if the default trapezoidal=TRUE is used). The output is given as a matrix without NAs, where each row is related to fuzzy numbers (given by 3 values for the triangular fuzzy numbers, or 4 values in the case of trapezoidal ones) for the consecutive variables. Many fuzzy variables (not only the single one) can be used. The input has to consist of fuzzy numbers of the same type (i.e., mixing triangular and trapezoidal fuzzy numbers is not allowed).

Various possible imputation methods can be used when the parameter method is specified – both the general ones (missForest or miceRanger from the respective packages, or knn from VIM package) and a more specific ones, tailored for the fuzzy data (dimp in the case of the DIMP method). Please note that due to the imputation, some output values can be improper fuzzy variables (e.g., a core of a fuzzy number can have greater value than its right end of the support). To avoid this, checkFuzzy=TRUE should be set. In this case, the imputation procedure is repeated until all of the results are proper triangular or trapezoidal fuzzy numbers. The improper values are removed and replaced with the respective fuzzy numbers from the input dataset. However, many repetitions (even unacceptably many) are then possible.

Value

The output is given as a matrix.

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library
```

```
list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values with the DIMP method

set.seed(12345)

FuzzyImputation(matrix1NA)

# impute missing values with the miceRanger method

set.seed(12345)

FuzzyImputation(matrix1NA,method = "miceRanger")
```

`FuzzyNumbersToMatrix` *Conversion of a list of fuzzy numbers into a matrix.*

Description

'FuzzyNumbersToMatrix' converts a list of triangular or trapezoidal fuzzy numbers into a matrix.

Usage

```
FuzzyNumbersToMatrix(fuzzyList, trapezoidal = TRUE, varNames = NA, ...)
```

Arguments

`fuzzyList`	Name of the list with fuzzy numbers.
`trapezoidal`	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones).

varNames	Optional names for columns of the output matrix.
...	Additional parameters passed to other functions.

Details

The procedure converts the given list of triangular or trapezoidal fuzzy numbers into a matrix. It is necessary to select the appropriate type of these fuzzy numbers (using `trapezoidal=FALSE` for triangular fuzzy numbers or the default `trapezoidal=TRUE` for trapezoidal ones) The output matrix has 3 (for triangular fuzzy numbers) or 4 (for trapezoidal ones) columns with the values of the left supports, cores (or the left and right ends of the cores for trapezoidal fuzzy cores) and right supports. Each row is related to single fuzzy number.

Value

The output is given as a matrix with 3 (for triangular fuzzy numbers) or 4 (for trapezoidal ones) columns and the number of rows is equal to number of fuzzy values.

See Also

`MatrixToFuzzyNumbers` for conversion of a matrix to fuzzy numbers

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# check the first fuzzy number

list1$value[[1]]

# convert fuzzy numbers to a matrix and check the first value

head(FuzzyNumbersToMatrix(list1$value))
```

ImputationDimp *DIMP (d-imputation) method for fuzzy numbers.*

Description

'ImputationDimp' imputes (i.e., replaces missing values) fuzzy numbers using the DIMP (d-imputation) method.

Usage

```
ImputationDimp(dataToImpute, trapezoidal = TRUE, ...)
```

Arguments

dataToImpute	Name of the input matrix (data frame or list) of fuzzy numbers with some NAs.
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
...	Additional parameters passed to other functions

Details

The procedure randomly imputes missing values (NAs) with suitable data in the case of the dataset (or matrix, or list) consisting of fuzzy numbers (triangular fuzzy numbers if `trapezoidal=FALSE` is set, or trapezoidal if the default `trapezoidal=TRUE` is used). The output is given as a matrix without NAs, where each row is related to fuzzy numbers (given by 3 values for the triangular fuzzy numbers, or 4 values in the case of trapezoidal ones) for the consecutive variables. Many fuzzy variables (not the only one) can be used. The input has to consist of fuzzy numbers of the same types (i.e., mixing triangular and trapezoidal fuzzy numbers is not allowed).

Value

The output is given as a matrix.

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
```

```
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values

ImputationDimp(matrix1NA)
```

`ImputationTests` *Battery of test for the imputed fuzzy values.*

Description

'ImputationTests' calculates various measures and applies goodness-of-fit statistical tests to check the quality of the imputed fuzzy values.

Usage

```
ImputationTests(
  trueData,
  imputedData,
  imputedMask,
  trapezoidal = TRUE,
  cutsNumber = 100,
  K = 50,
  ...
)
```

Arguments

`trueData`	Name of the input matrix (or data frame, or list) with the true values of the variables.

imputedData	Name of the input matrix (or data frame) with the imputed values.
imputedMask	Matrix (or data frame) with logical values where TRUE indicates the cells with the imputed values.
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
cutsNumber	Number of cuts for the epistemic bootstrap tests.
K	Value of K for the res epistemic test.
...	Additional parameters passed to other functions.

Details

The procedure uses other functions embedded in this package to check the quality of the imputed fuzzy values if they are compared with the original ones. This procedure calculates number of non-FNs for each variable, error matrix (using ErrorMatrix), various statistical measures (with StatisticalMeasures), applies epistemic goodness-of-fit tests (using ApplyStatisticalTests), and evaluates the fuzzy measures (with CalculateFuzzyMeasures). Therefore, this function can be directly applied as one-click benchmark tool.

To properly distinguish the real values with their imputed counterparts, the additional matrix imputedMask should be provided. In this matrix, the logical value TRUE points out the cells with the imputed values. Otherwise, FALSE should be used.

All of the input datasets can be given as matrices or data frames.

To get overall comparison of the methods, summary(object,...) can be used for the output object from this method. The values diff are equal to the differences of p-values between the respective tests for the parts true and imputed there.

Value

The output is an S3 object of the class impTest given as a list of the matrices: trueValues - the true, input values (the same as trueData), mask - the masked (NAs) values (the same as imputedMask), nonFNNumbers - the vector with the numbers of non-FNs samples for each variable (with the overall mean), errorMatrix – the output from the function ErrorMatrix, statisticalMeasures – the output from the function StatisticalMeasures, statisticalTests – the output from the function ApplyStatisticalTests, fuzzyMeasures – the output from the function CalculateFuzzyMeasures.

See Also

MethodsComparison for the imputation benchmark for all methods, summary.impTest.

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)
```

```
# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)

# impute missing values

matrix1DImp <- ImputationDimp(matrix1NA)

# find cells with NAs

matrix1Mask <- is.na(matrix1NA)

# check the quality of the imputed values

ImputationTests(matrix1,matrix1DImp,matrix1Mask,trapezoidal=TRUE)
```

`IntroducingNA` *Introducing NAs to the specified matrix.*

Description

'IntroducingNA' introduces missing values (NAs) to the specified data matrix.

Usage

```
IntroducingNA(dataMatrix, percentage = 0.05, ...)
```

Arguments

dataMatrix	Name of the input matrix (or list, or data frame).
percentage	Desired percentage of missing values (NAs) in each row.
...	Additional parameters passed to other functions.

Details

The procedure changes randomly some values in the specified matrix to "missing values" (denoted by NA). Number of these missing values in each row is given by the parameter percentage. If the input is a list of fuzzy numbers or data frame, then it is automatically converted to a matrix.

Value

The output is given as a matrix.

Examples

```
# prepare matrix with 3 columns and 3 rows

matrix1 <- matrix(c(1,3,5,2,5,7,1,4,5),ncol=3,byrow = TRUE)

# add 1 NA in each row

set.seed(12345)

IntroducingNA(matrix1,percentage = 0.33)
```

MatrixToFuzzyNumbers *Conversion of a matrix to a list of fuzzy numbers.*

Description

'MatrixToFuzzyNumbers' converts a matrix into a list of triangular or trapezoidal fuzzy numbers.

Usage

```
MatrixToFuzzyNumbers(fuzzyMatrix, varNames = NA, ...)
```

Arguments

fuzzyMatrix	Name of the matrix with fuzzy numbers.
varNames	Optional names for values of the output list.
...	Additional parameters passed to other functions.

Details

The procedure converts the given matrix to a list of triangular or trapezoidal fuzzy numbers. If the input matrix has 3 columns, then they are treated as descriptions of consecutive triangular fuzzy numbers. In the case of 4 columns, we get trapezoidal fuzzy numbers. The values in these columns are equal to the left supports, cores (or left and right ends of the cores for trapezoidal fuzzy numbers) and right supports. Each row is related to single fuzzy number.

Value

The output is given as a list of fuzzy numbers.

See Also

FuzzyNumbersToMatrix for conversion of a list of fuzzy numbers into a matrix.

Examples

```
library(FuzzyNumbers)

# prepare matrix with 2 triangular fuzzy numbers

matrix1 <- matrix(c(1,3,5,2,5,7),ncol=3,byrow = TRUE)

# convert this matrix to list of fuzzy numbers

MatrixToFuzzyNumbers(matrix1)
```

MeasureAHD *Function to calculate the AHD distance between two fuzzy numbers.*

Description

'MeasureAHD' calculates the AHD (Area Hight Distance) measure between two trapezoidal or triangular fuzzy numbers.

Usage

```
MeasureAHD(value1, value2, trapezoidal = TRUE, ...)
```

Arguments

value1	The first input triangular or trapezoidal fuzzy number.
value2	The second input triangular or trapezoidal fuzzy number.
trapezoidal	Logical value depending on the type of input fuzzy values (triangular or trapezoidal ones).
...	Additional parameters passed to other functions.

Details

The procedure calculates the AHD (Area Hight Distance) measure of the distance between two trapezoidal or triangular fuzzy numbers. The input values can be given as triangular/trapezoidal fuzzy numbers using the objects defined in `FuzzyNumbers` package or vectors with three (in the case of triangular fuzzy numbers) or four (for trapezoidal fuzzy numbers) values related to left end of the support, the core (or its interval, respectively), and the right end of the support. The parameter `trapezoidal` is used to indicate if the input values are trapezoidal fuzzy numbers or triangular ones.

Value

The output is given as a numerical value.

References

M. Amirfakhrian, S. Yeganehmanesh, and P. Grzegorzewski, “A new distance on fuzzy semi-numbers", Soft Computing, vol. 22, no. 14, pp. 4511–4524, 2018

See Also

`MeasureHSD`, `MeasureEuclidean` for other procedures to calculate distance measures.

Examples

```
# let's define two trapezoidal fuzzy numbers

tpfn1 <- c(1,2,3,4)

tpfn2 <- c(2,6,8,10)

# calculate the distance

MeasureAHD(tpfn1,tpfn2)

# now we use objects from the FuzzyNumbers package

# load the necessary library

library(FuzzyNumbers)

tpfn1 <- TrapezoidalFuzzyNumber(1,2,3,4)

tpfn2 <- TrapezoidalFuzzyNumber(2,6,8,10)

MeasureAHD(tpfn1,tpfn2)
```

`MeasureEuclidean` *Function to calculate the Euclidean distance between two fuzzy numbers.*

Description

'MeasureEuclidean' calculates the Euclidean measure between two trapezoidal or triangular fuzzy numbers.

Usage

```
MeasureEuclidean(value1, value2, trapezoidal = TRUE, ...)
```

Arguments

`value1`	The first input triangular or trapezoidal fuzzy number.
`value2`	The second input triangular or trapezoidal fuzzy number.
`trapezoidal`	Logical value depending on the type of input fuzzy values (triangular or trapezoidal ones)
`...`	Additional parameters passed to other functions.

Details

The procedure calculates the Euclidean measure of the distance between two trapezoidal or triangular fuzzy numbers. The input values can be given as triangular/trapezoidal fuzzy numbers using the objects defined in `FuzzyNumbers` package or vectors with three (in the case of triangular fuzzy numbers) or four (for trapezoidal fuzzy numbers) values related to left end of the support, the core (or its interval, respectively), and the right end of the support. The parameter `trapezoidal` is used to indicate if the input values are trapezoidal fuzzy numbers or triangular ones.

Value

The output is given as a numerical value.

See Also

`MeasureAHD`, `MeasureHSD` for other procedures to calculate distance measures.

Examples

```
# let's define two trapezoidal fuzzy numbers

tpfn1 <- c(1,2,3,4)

tpfn2 <- c(2,6,8,10)
```

```
# calculate the distance

MeasureEuclidean(tpfn1,tpfn2)

# now we use objects from the FuzzyNumbers package

# load the necessary library

library(FuzzyNumbers)

tpfn1 <- TrapezoidalFuzzyNumber(1,2,3,4)

tpfn2 <- TrapezoidalFuzzyNumber(2,6,8,10)

MeasureEuclidean(tpfn1,tpfn2)
```

`MeasureHSD` *Function to calculate the HSD distance between two fuzzy numbers.*

Description

'MeasureHSD' calculates the HSD (Area Hight Distance) measure between two trapezoidal or triangular fuzzy numbers.

Usage

```
MeasureHSD(value1, value2, trapezoidal = TRUE)
```

Arguments

`value1`	The first input triangular or trapezoidal fuzzy number.
`value2`	The second input triangular or trapezoidal fuzzy number.
`trapezoidal`	Logical value depending on the type of input fuzzy values (triangular or trapezoidal ones).

Details

The procedure calculates the HSD (Hight Source Distance) measure of the distance between two trapezoidal or triangular fuzzy numbers. The input values can be given as triangular/trapezoidal fuzzy numbers using the objects defined in `FuzzyNumbers` package or vectors with three (in the case of triangular fuzzy numbers) or four (for trapezoidal fuzzy numbers) values related to left end of the support, the core (or its interval, respectively), and the right end of the support. The parameter `trapezoidal` is used to indicate if the input values are trapezoidal fuzzy numbers or triangular ones.

Value

The output is given as a numerical value.

References

S. Yeganehmanesh, M. Amirfakhrian, and P. Grzegorzewski, "Fuzzy semi-numbers and a distance on them with a case study in medicine," Mathematical Sciences, vol. 12, no. 1, pp. 41–52, 2018

See Also

MeasureAHD, MeasureEuclidean for other procedures to calculate distance measures.

Examples

```
# let's define two trapezoidal fuzzy numbers

tpfn1 <- c(1,2,3,4)

tpfn2 <- c(2,6,8,10)

# calculate the distance

MeasureHSD(tpfn1,tpfn2)

# now we use objects from the FuzzyNumbers package

# load the necessary library

library(FuzzyNumbers)

tpfn1 <- TrapezoidalFuzzyNumber(1,2,3,4)

tpfn2 <- TrapezoidalFuzzyNumber(2,6,8,10)

MeasureHSD(tpfn1,tpfn2)
```

---

methodNames *A vector containing names of the resampling methods.*

Description

'methodNames' is a vector containing names of the resampling methods.

Usage

```
methodNames
```

Format

An object of class character of length 4.

Value

This function returns a vector of strings.

Examples

```
# check the names

methodNames
```

`MethodsComparison` *Comparison of imputation methods for fuzzy values.*

Description

'MethodsComparison' compares the quality of built-in imputation methods using various measures and goodness-of-fit statistical tests for the given fuzzy dataset.

Usage

```
MethodsComparison(
  trueData,
  iterations = 100,
  percentage = 0.05,
  trapezoidal = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

`trueData`	Name of the input matrix (or data frame) with the true values of the variables.
`iterations`	Number of the repetitions of each analyses (introducing NAs and then imputation of the missing values).
`percentage`	Desired percentage of missing values (NAs) in each row.
`trapezoidal`	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
`verbose`	Logical value if the progress bar should be shown.
`...`	Additional parameters passed to other functions.

Details

The procedure uses the function ImputationTests to compare the quality of the imputation methods for the specified fuzzy dataset. To minimize random effects, each analysis is repeated iterations times with the new randomly generated NA values in the input dataset, and then new imputed values for all built-in methods. To generate the new NAs values, the function IntroducingNA is used. Next, the results, the same as forImputationTests (apart from trueValues and mask), are averaged.

The input dataset can be given as matrix or data frame.

To get overall comparison of the methods, summary(object,...) can be used for the output object from this method. The values diff are equal to the differences of p-values between the respective tests for the parts true and imputed there.

Value

The output is an S3 object of the class metComp given as a list of the matrices: nonFNNumbers - the vector with the numbers of non-FNs samples for each variable (with the overall mean), errorMatrix – the output from the function ErrorMatrix, statisticalMeasures – the output from the function StatisticalMeasures, statisticalTests – the output from the function ApplyStatisticalTests, fuzzyMeasures – the output from the function CalculateFuzzyMeasures.

See Also

ImputationTests for the single imputation benchmark, summary.metComp.

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)
```

```
# check the quality of the imputed values

## Not run:

MethodsComparison(matrix1,iterations=10,trapezoidal=TRUE)


## End(Not run)
```

RemoveNotFuzzy *Removing values that are not fuzzy numbers.*

Description

'RemoveNotFuzzy' removes all values that are not proper fuzzy numbers and restores the previous ones.

Usage

```
RemoveNotFuzzy(trueData, imputedData, trapezoidal = TRUE, ...)
```

Arguments

trueData	Name of the input matrix (data frame(or data frame)) that is used to restore erroneous fuzzy numbers.
imputedData	Name of the input matrix (data frame) with fuzzy numbers to check their correctness.
trapezoidal	Logical value depending on the type of fuzzy values (triangular or trapezoidal ones) in the dataset.
...	Additional parameters passed to other functions.

Details

The procedure checks all the values in the given matrix (or data frame) specified by imputedData and if some of them are not proper fuzzy numbers (e.g., their cores are outside the supports), they are removed. Instead of these erroneous values, the previous ones from the input matrix (a data frame, or a list) trueData are restored. These matrices (or data frames) should consist of fuzzy numbers (triangular fuzzy numbers if trapezoidal=FALSE is set, or trapezoidal ones if the default trapezoidal=TRUE is used). The output is given as a matrix where each row is related to fuzzy numbers (with 3 values for the triangular fuzzy numbers, or 4 values in the case of trapezoidal ones) for the consecutive variables. The input has to consist of fuzzy numbers of the same type (i.e., mixing triangular and trapezoidal fuzzy numbers is not allowed).

Value

The output is given as a matrix.

Examples

```
# matrix with proper values of triangular fuzzy numbers

matrixOK <- matrix(c(1,2,3,7,10,12,8,10,11),ncol=3,byrow = TRUE)

# matrix with the wrong third value of fuzzy triangular number (its core is greater than
# the left end of its support)

matrixFalse <- matrix(c(1,2,3,7,10,12,8,20,11),ncol=3,byrow = TRUE)

# remove the third value and restore the previous one

RemoveNotFuzzy(matrixOK,matrixFalse,trapezoidal = FALSE)
```

StatisticalMeasures *Calculation of statistical measures for errors of the imputed data.*

Description

`StatisticalMeasures` calculates various statistical measures between the real and imputed data.

Usage

```
StatisticalMeasures(trueData, imputedData, imputedMask, ...)
```

Arguments

`trueData`	Name of the input matrix (or data frame) with the true values of the variables.
`imputedData`	Name of the input matrix (or data frame) with the imputed values.
`imputedMask`	Matrix (or data frame) with logical values where `TRUE` indicates the cells with the imputed values.
`...`	Additional parameters passed to other functions.

Details

The procedure calculates different statistical measures between the real and imputed data for each column, namely:

- TrueMean - the mean only for the real but missing data,
- ImpMean - the mean only for the imputed values,
- TrueSD - the standard deviation only for the real but missing data,
- ImpSD - the standard deviation only for the imputed values,
- GenMean - the mean for the all real data (given by `trueData`),
- GenImpMean - the mean for real data with the respectively imputed values (given by `imputedData`),

- GenSD - the standard deviation for the all real data (given by `trueData`),
- GenImpSD - the standard deviation for real data with the respectively imputed values (given by `imputedData`),
- AbsDiffTrueImpMean - the absolute difference between TrueMean and ImpMean,
- AbsDiffTrueImpSD - the absolute difference between TrueSD and ImSD,
- AbsDiffGenImpMean - the absolute difference between GenMean and GenImpMean,
- AbsDiffGenImpSD - the absolute difference between GenSD and GenImpSD.

To properly distinguish the real values with their imputed counterparts, the additional matrix `imputedMask` should be provided. In this matrix, the logical value `TRUE` points out the cells with the imputed values. Otherwise, `FALSE` should be used. These input datasets should be given as matrices or data frames.

Value

The output is given as a matrix with columns related to all columns of the input dataset plus the overall mean.

Examples

```
# seed PRNG

set.seed(1234)

# load the necessary library

library(FuzzySimRes)

# generate sample of trapezoidal fuzzy numbers with FuzzySimRes library

list1<-SimulateSample(20,originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
incrCorePD="rexp", parIncrCorePD=list(rate=2),
suppLeftPD="runif",parSuppLeftPD=list(min=0,max=0.6),
suppRightPD="runif", parSuppRightPD=list(min=0,max=0.6),
type="trapezoidal")

# convert fuzzy data into a matrix

matrix1 <- FuzzyNumbersToMatrix(list1$value)

# check starting values

head(matrix1)

# add some NAs to the matrix

matrix1NA <- IntroducingNA(matrix1,percentage = 0.1)

head(matrix1NA)
```

```
# impute missing values

matrix1DImp <- ImputationDimp(matrix1NA)

# find cells with NAs

matrix1Mask <- is.na(matrix1NA)

# calculate errors for the imputed values

StatisticalMeasures(matrix1,matrix1DImp,matrix1Mask)
```

`summary.impTest` *Print summary of the benchmark for the imputation method.*

Description

Print summary of the benchmark for the imputation method.

Usage

```
## S3 method for class 'impTest'
summary(object, ...)
```

Arguments

`object`	Object of S3 class `impTest`.
`...`	Additional parameters passed to other functions.

`summary.metComp` *Print summary of the comparison of the imputation methods.*

Description

Print summary of the comparison of the imputation methods.

Usage

```
## S3 method for class 'metComp'
summary(object, ...)
```

Arguments

`object`	Object of S3 class `metComp`.
`...`	Additional parameters passed to other functions.

Index