Package 'UncerIn2'

November 24, 2015

Version 2.0

Date 2015-11-10

Type Package

Title Implements Models of Uncertainty into the Interpolation Functions

Author Tomas Burian <Buri777@seznam.cz>

Maintainer Tomas Burian <Buri777@seznam.cz>

Description Provides a basic (random) data, grids, 6 models of uncertainty, 3 automatic interpolations (idw, spline, kriging), variogram and basic data visualization.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 3.0.0), sp, RandomFields, automap, fields, gstat

Imports geoR, methods, stats, Rcpp, utils

Collate 'class-01-Points.R' 'class-02-UncertainPoints.R' 'class-03-UncertainInterpolation.R' 'class-04-FuzzyInterpolation.R' 'class-05-Variogram.R' 'methods-as.data.frame.R' 'methods-as.UncertainPoints.R' 'methods-uncertaintyConstant.R' 'methods-uncertaintyError.R' 'methods-uncertaintyPercent.R' 'methods-uncertaintyRandomDeviate.R' 'methods-uncertaintyRandomNumber.R' 'methods-uncertaintyRandomPercent.R' 'methods-Grid.R' 'methods-idwUncertain.R' 'methods-splineUncertain.R' 'methods-krigingUncertain.R' 'methods-variogram.R' 'methods-plot.R'

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-24 18:37:19

R topics documented:

as.dataframe . . . 2
as.UncertainPoints . . . 3
FuzzyInterpolation-class . . . 4
Grid.box . . . 4
Grid.def . . . 5
Grid.interpolation . . . 6
idwUncertain . . . 6
krigingUncertain . . . 7
Plot . . . 9
Points . . . 9
Points-class . . . 10
splineUncertain . . . 10
TUTORIAL . . . 11
UncertainInterpolation-class . . . 12
UncertainPoints-class . . . 13
uncertaintyConstant . . . 14
uncertaintyError . . . 14
uncertaintyInterpolation2-information . . . 15
uncertaintyPercent . . . 17
uncertaintyRandomDeviate . . . 18
uncertaintyRandomNumber . . . 18
uncertaintyRandomPercent . . . 19
variogram . . . 20

Index **22**

`as.dataframe` *Transformation of S4 class Points or UncertainPoints into data.frame*

Description

This function provides the transformation of S4 object class `UncertainPoints` or `Points` into the `data.frame` data format.

Usage

```
## S4 method for signature 'UncertainPoints'
as.dataframe(data)

## S4 method for signature 'Points'
as.dataframe(data)
```

Arguments

`data`	Input data type of S4 object of class UncertainPoints or Points.

Value

Returns an object of class `data.frame`.

See Also

`Points-class`, `Points`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

`as.UncertainPoints` *Transformation of S4 class UncertainInterpolation into UncertainPoints*

Description

This function provides the transformation of S4 object class `UncertainInterpolation` into the S4 class `UncertainPoints`..

Usage

```
## S4 method for signature 'UncertainInterpolation'
as.UncertainPoints(object,grid)
```

Arguments

`object`	Input data type of S4 object class UncertainInterpolation.
`grid`	Input grid (default set FALSE)

Value

Returns an object of class `UncertainPoints`.

See Also

`UncertainInterpolation-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

FuzzyInterpolation-class

S4 class Representing a FuzzyInterpolation

Description

Definition of S4 class for output data from fuzzy (kriging) interpolation function of package UncerIn2. Where x, y should represent coordinates and minimal,modalValue, maximal the values of uncertainty.

Slots

x: Input numeric data (number specyfing the x grid coordinate).

y: Input numeric data (number specyfing the y grid coordinate).

minimal: Defined model values of uncertaintyLower - the bottom calculated part.

modalValue: Defined model values of modalValue - the mean values.

maximal: Defined model values of uncertaintyUpper - the upper calculated part.

See Also

uncertaintyInterpolation2-package

Other res: Points-class, UncertainInterpolation-class, UncertainPoints-class

Examples

```
showClass("FuzzyInterpolation")
```

Grid.box

Generation of grid defined by surrounding box

Description

This method creates a grid defined in two possible ways - the user should decide if spatial grid is required (gridded = TRUE/FALSE). The size of the grid is recognized by the input data coordinates. User should define size of the single cell.

Usage

```
## S4 method for signature 'Points'
Grid.box(object,gridded,cellsize)
```

Arguments

object	Input data type of S4 object class Points.
gridded	Logical value TRUE/FALSE. If true, then Spatial grid is made.
cellsize	Numeric value for the size of cell.

Value

Returns an object of class SpatialPixels or data.frame, depends on whether the spatial grid is required.

See Also

Points-class, Grid.def, Grid.interpolation, uncertaintyInterpolation2-package

---

Grid.def *Generation of grid with defined number of cels*

Description

This method creates a grid defined in two possible ways - the user should decide if spatial grid is required (gridded = TRUE/FALSE). The size of the grid should be also defined by the user.

Usage

```
## S4 method for signature 'Points'
Grid.def(object,gridded,numberOfCellsX,numberOfCellsY)
```

Arguments

object	Input data type of S4 object class Points.
gridded	Logical value TRUE/FALSE. If true, then Spatial grid is made.
numberOfCellsX	The number of cells on axle X.
numberOfCellsY	The number of cells on axle Y.

Value

Returns an object of class SpatialPixels or data.frame, depends on whether the spatial grid is required.

See Also

Points-class, Grid.box, Grid.interpolation, uncertaintyInterpolation2-package

Grid.interpolation *Generation of grid from the results of interpolation functions*

Description

This method creates a grid from S4 object class `UncertainInterpolation`, defined in two possible ways - the user should decide if spatial grid is required (`gridded = TRUE/FALSE`). The size of the grid is recognized by the input data coordinates. User should define size of the single cell.

Usage

```
## S4 method for signature 'UncertainInterpolation'
Grid.interpolation(object,gridded,cellsize)
```

Arguments

`object`	Input data type of S4 object class `UncertainInterpolation`.
`gridded`	Logical value `TRUE/FALSE`. If true, then Spatial grid is made.
`cellsize`	Numeric value for the size of cell.

Value

Returns an object of class `SpatialPixels` or `data.frame`, depends on whether the spatial grid is required.

See Also

`UncertainInterpolation-class`, `Grid.box`, `Grid.def`, `uncertaintyInterpolation2-package`

idwUncertain *IDW interpolation*

Description

This function provides IDW interpolation over the input data enriched by the uncertainty. The input data must be an S4 object class of `UncertainPoints` and grid type of `Spatial`. Output object is type of S4 class `UncertainInterpolation`.

Usage

```
## S4 method for signature 'UncertainPoints,Spatial'
idwUncertain(object,grid,nmax,idp)
```

Arguments

object	Input data. An object of UncertainPoints class.
grid	Input Spatial grid.
nmax	The number of nearest observations that should be used.
idp	Inverse distance power.

Value

Returns an object of class UncertainInterpolation.

See Also

UncertainPoints-class, UncertainInterpolation-class, Grid.def, Grid.box, Grid.interpolation, gstat, Plot, uncertaintyInterpolation2-package

---

krigingUncertain *Krigging and Fuzzy kriging interpolation*

Description

This function provides kriging interpolation over the input data enriched by the uncertainty with multiple dispatch. The input data must be an S4 object class of UncertainPoints or Points in case of Fuzzy kriging. Grid type of Spatial. Output object is type of S4 class UncertainInterpolation or in case of Fuzzy kriging S4 class FuzzyInterpolation. For more informations about fuzzy kriging interpolation read Details.

Usage

```
## S4 method for signature 'UncertainPoints,Spatial'
krigingUncertain(object, grid, data_variogram = data, block = 0,
   model = c("Sph", "Exp", "Gau", "Ste"), kappa = c(0.05, seq(0.2, 2, 0.1), 5, 10),
   fix.values = c(NA,NA,NA), remove_duplicates = TRUE, verbose = FALSE, GLS.model = NA,
    start_vals = c(NA,NA,NA), miscFitOptions = list())

## S4 method for signature 'Points,Spatial'
krigingUncertain(object, grid, krigingModel, psills, ranges, nuggets, models,
   vgm_start, logResults=FALSE)
```

Arguments

object	Input data. An object of UncertainPoints class.
grid	Input Spatial grid.
...	Additional arguments to be passed to f.
data_variogram	An optional way to provide a different dataset for the building of the variogram then for the spatial interpolation.

block	Use this parameter to pass on a specification for the block size. e.g. c(1000,1000)
model	List of models that will be tested during automatic variogram fitting
kappa	List of values for the smoothing parameter of the Matern model that will be tested during automatic variogram fitting.
fix.values	Can be used to fix a variogram parameter to a certain value. It consists of a list with a length of three. The items describe the fixed value for the nugget, range and sill respectively. Setting the value to NA means that the value is not fixed. Is passed on to autofitVariogram.
remove_duplicates	logical, remove duplicates from input
verbose	logical, if TRUE autoKrige will give extra information on the fitting process
GLS.model	If a variogram model is passed on through this parameter a Generalized Least Squares sample variogram is calculated.
start_vals	Can be used to give the starting values for the variogram fitting. The items describe the fixed value for the nugget, range and sill respectively.
miscFitOptions	Additional options to set the behavior of autofitVariogram.
krigingModel	Model of the kriging used in the calculations.
psills	Defined psills.
ranges	Defined ranges.
nuggets	Defined nuggets.
models	Variogram model of dependent variable (or its residuals).
vgm_start	Modal variogram selected for the data.
logResults	Was the dataset logaritmized prior to the calculation?

Details

The function for Fuzzy kriging and its processes were taken from source code Jan Caha. For more informations and details visit https://github.com/JanCaha/Hais2015-paper.

Value

Returns an object of class UncertainInterpolation or FuzzyInterpolation.

See Also

UncertainPoints-class, Points-class, UncertainInterpolation-class, FuzzyInterpolation-class, Grid.def,Grid.box, Grid.interpolation, autoKrige, Plot, uncertaintyInterpolation2-package, https://github.com/JanCaha/Hais2015-paper

Plot *Plotting S4 class UncertainInterpolation*

Description

This function provides the plotting of S4 object class `UncertainInterpolation`.

Usage

```
## S4 method for signature 'UncertainInterpolation'
Plot(object, attr1, attr2, attr3, cuts, pretty)

## S4 method for signature 'UncertainInterpolation'
Plot(object, attr1 = "uncertaintyLower", attr2 = "modalValue",
attr3 = "uncertaintyUpper", cuts = 10, pretty=TRUE)
```

Arguments

`object`	Input data type of S4 object class UncertainInterpolation.
`attr1`	First plotting atrribute.
`attr2`	Second plotting atrribute.
`attr3`	Third plotting atrribute.
`cuts`	Number of cuts.
`pretty`	Logical value `TRUE/FALSE`.(choose colour breaks at pretty numbers?)

See Also

`UncertainInterpolation-class`, `uncertaintyInterpolation2-package`

Points *Creates S4 object class Points*

Description

This function creates an object of S4 class `Points`.

Usage

```
Points(x, y, z)
```

Arguments

`x`	Input numeric data (number specyfing the x coordinate).
`y`	Input numeric data (number specyfing the y coordinate).
`z`	Input numeric data (number specyfing the values of variable).

Value

Returns an object of class `Points`.

`Points-class` *S4 class Representing a class Points*

Description

Definition of class for input data of numeric format `x/y/z`. Also defines input data format into the functions of package `UncerIn2`, which are building uncertainty models over input data. Where `x, y` should represent coordinates and `z` variable.

Slots

`x`: Input numeric data (number specyfing the x coordinate).

`y`: Input numeric data (number specyfing the y coordinate).

`z`: Input numeric data (number specyfing the values of variable).

See Also

`uncertaintyInterpolation2-package`

Other res: `FuzzyInterpolation-class`, `UncertainInterpolation-class`, `UncertainPoints-class`

Examples

```
showClass("Points")
```

`splineUncertain` *Spline interpolation*

Description

This function provides Spline interpolation over the input data enriched by the uncertainty. The input data must be an S4 object class of `UncertainPoints` and grid type of `data.frame`. Output object is type of S4 class `UncertainInterpolation`.

Usage

```
## S4 method for signature 'UncertainPoints,data.frame'
splineUncertain(object, grid, m = NULL, p = NULL,
  scale.type = "range", lon.lat = FALSE, miles = TRUE, method = "GCV", GCV = TRUE)
```

Arguments

object	Input data. An object of UncertainPoints class.
grid	Input grid type of dataframe.
m	A polynomial function of degree (m-1) will be included in the model as the drift (or spatial trend) component. Default is the value such that 2m-d is greater than zero where d is the dimension of x.
p	Polynomial power for Wendland radial basis functions. Default is 2m-d where d is the dimension of x.
scale.type	The independent variables and knots are scaled to the specified scale.type. By default the scale type is "range", whereby the locations are transformed to the interval (0,1) by forming (x-min(x))/range(x) for each x. Scale type of "user" allows specification of an x.center and x.scale by the user. The default for "user" is mean 0 and standard deviation 1. Scale type of "unscaled" does not scale the data.
lon.lat	If TRUE locations are interpreted as lognitude and latitude and great circle distance is used to find distances among locations.
miles	If TRUE great circle distances are in miles if FALSE distances are in kilometers.
method	Determines what "smoothing" parameter should be used. The default is to estimate standard GCV Other choices are: GCV.model, GCV.one, RMSE, pure error and REML. The differences are explained in the Krig help file.
GCV	If TRUE the decompositions are done to efficiently evaluate the estimate, GCV function and likelihood at multiple values of lambda.

Value

Returns an object of class UncertainInterpolation.

See Also

UncertainPoints-class, UncertainInterpolation-class, Grid.def, Grid.box, Grid.interpolation, Tps, Plot, uncertaintyInterpolation2-package

---

TUTORIAL *Package UncerIn2 TUTORIAL*

Description

Here is described how to work with this package. An example usage with free dataset meuse.

Author(s)

Tomas Burian, <Buri777@seznam.cz>

See Also

meuse uncertaintyInterpolation2-package

Examples

```
## package UncerIn2 TUTORIAL

## Example data import and definition
# dataset meuse import
data(meuse)
# S4 class object Points input data definition
points = Points(x = meuse$x, y = meuse$y, z = meuse$elev)

## Uncertainty creation
# building uncertainty model (based on uncertaintyError function)
uncertaintyModel = uncertaintyError(points)

## Grid creation
# generate grid for interpolation
gridDef = Grid.def(points) # data.frame
gridDef.spat = Grid.def(points, TRUE) # SpatialPixels
# OR (bounded by box)
# gridBox = Grid.box(points) # data.frame
# gridBox.spat = Grid.box(points, TRUE) # SpatialPixels

## Iinterpolation
IDW = idwUncertain(uncertaintyModel, gridDef.spat)
# spline = splineUncertain(uncertaintyModel, gridDef)
# kriging = krigingUncertain(uncertaintyModel, gridDef.spat)

## variogram estimation
# var = variogram(uncertaintyModel)
# show(var)

# visualization of results
Plot(IDW)
```

UncertainInterpolation-class

S4 class Representing a UncertainInterpolation

Description

Definition of S4 class for output data from interpolation functions of package UncerIn2. Where x, y should represent coordinates and uncertaintyLower, modalValue,uncertaintyUpper the values of uncertainty.

Slots

x: Input numeric data (number specyfing the x grid coordinate).

y: Input numeric data (number specyfing the y grid coordinate).

uncertaintyLower: Defined model values of uncertaintyLower - the bottom calculated part.

modalValue: Defined model values of modalValue - the mean values.

uncertaintyUpper: Defined model values of uncertaintyUpper - the upper calculated part.

See Also

uncertaintyInterpolation2-package

Other res: FuzzyInterpolation-class, Points-class, UncertainPoints-class

Examples

```
showClass("UncertainInterpolation")
```

UncertainPoints-class *S4 class Representing a UncertainPoints*

Description

Definition of S4 class for output data from functions of package UncerIn2, which are building uncertainty models over input data. Also defines input data format into the interpolation function of package UncerIn2. Where x, y should represent coordinates and uncertaintyLower,modalValue, uncertaintyUpper the variables of uncertainty model.

Slots

x: Input numeric data (number specyfing the x coordinate).

y: Input numeric data (number specyfing the y coordinate).

uncertaintyLower: Defined model values of uncertaintyLower - the bottom calculated part.

modalValue: Defined model values of modalValue - the mean values.

uncertaintyUpper: Defined model values of uncertaintyUpper - the upper calculated part.

See Also

uncertaintyInterpolation2-package

Other res: FuzzyInterpolation-class, Points-class, UncertainInterpolation-class

Examples

```
showClass("UncertainPoints")
```

uncertaintyConstant *Creates S4 object class UncertainPoints*

Description

Builds an uncertainty model based on the constant error over the input data. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyConstant(data, value = 2)
```

Arguments

`data`	Input data. S4 class of `Points`.
`value`	Constant value of imprecision that defines the uncertainty.

Value

Returns an object of class `UncertainPoints`.

See Also

`Points-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

uncertaintyError *Creates S4 object class UncertainPoints*

Description

Builds an uncertainty model over the input data based on the spatially correlated errors. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyError(input, T = NULL, grid = FALSE, distances = NULL,
  dim = NULL, data = NULL, given = NULL)
```

Arguments

input	Input data. S4 class of Points
T	optional vector of time coordinates, T must always be an equidistant vector. Instead of T=seq(from=From, by=By, len=Len) one may also write T=c(From, By, Len).
grid	Logical; RandomFields can find itself the correct value in nearly all cases, so that usually grid need not be given.
distances	Another alternative to pass the (relative) coordinates.
dim	Only used if distances are given.
data	For conditional simulation and random imputing only. If data is missing, unconditional simulation is performed.
given	Optional, matrix or list. If given matrix then the coordinates can be given separately, namely by given where, in each row, a single location is given.

Details

For the calculations of spatially correlated errors was used package RandomFields.

Value

Returns an object of class UncertainPoints.

See Also

Points-class, UncertainPoints-class, RFsimulate, uncertaintyInterpolation2-package

---

uncertaintyInterpolation2-information

Implements models of uncertainty into the interpolation functions

---

Description

This package is the main result of diploma thesis. It can provide a basic (random) data, grids, 6 models of uncertainty, 3 automatic interpolations (idw, spline, kriging), variogram and basic data visualization. All together it could make a basic tools to solve the problem of uncertainty in this world.

Details

Package:	uncertaintyInterpolation2
Type:	Package
Version:	2.0
Date:	2015-11-10
License:	R (>= 3.0.0)
LazyLoad:	yes

Author(s)

Tomas Burian <Buri777@seznam.cz>

Maintainer: Tomas Burian <Buri777@seznam.cz>

References

BURIAN, T. EXTENSION OF INTERPOLATION TOOLS IN R PROJECT BY MODELS OF UNCERTAINTY. Univerzita Palackeho v Olomouci, 2013.

CAHA, J. Fuzzy Surface Analyses: First Experiments. Olomouc : Terra Notitia, Palacky University for Department of Geoinformatics, 1st ed. edition, 2015. ISBN 978-80-244-4055-2.

CAHA, J., MAREK, L., DVORSKY, J. (in print) Predicting PM10 concentrations using fuzzy kriging. In Hybrid Artificial Intelligent Systems. Springer, 2015.

FISHER, P. F., TATE, N. J. Causes and consequences of error in digital elevation models. Progress in Physical Geography, 30, 4, p. 467 to 489, August 2006. ISSN 03091333.

CHAMBERS, J. M. Programming with Data A Guide to the S Language. Springer- Verlag, 1998.

HEUVELINK, G. B. M. Analysing Uncertainty Propagation in GIS: Why is it not that Simple? In FOODY, G. M., ATKINSON, P. M. (Ed.) Uncertainty in remote sensing and GIS, s. 307. Chichester : Wiley, 2002. ISBN 0470844086.

HIEMSTRA, P., PEBESMA, E., TWENHOFEL, C., HEUVELINK, G. Realtime automatic interpolation of ambient gamma dose rates from the dutch radioactivity monitoring network. Computers & Geosciences, 2008. DOI: http://dx.doi.org/10.1016/j.cageo.2008.10.011.

LODWICK, W., ANILE, M., SPINELLA, S. Introduction. In LODWICK, W. (Ed.) Fuzzy surfaces in GIS and geographical analysis : theory, analytical methods, algorithms, and applications, p. 1 to 46. Boca Raton : CRC Press, 2008. ISBN 9780849363955.

MATLOFF, N. THE ART OF R PROGRAMMING A Tour of Statistical Software Design. William Pollock, 2011.

OKSANEN, J. Digital Elevation Model Error in Terrain Analysis. PhD thesis, University of Helsinki, 2006.

R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2013. Dostupne z: http: //www.R-project.org.

VIERTL, R. Statistical methods for fuzzy data. Chichester, West Sussex : Wiley, 2011. ISBN 9780470699454.

WAELDER, O. An application of the fuzzy theory in surface interpolation and surface deformation analysis. Fuzzy Sets and Systems, 158, 14, p. 1535 to 1545, July 2007.

See Also

TUTORIAL http://www.r-project.org

Examples

```
# dataset meuse import
data(meuse)

# S4 class object Points input data definition
points = Points(x = meuse$x, y = meuse$y, z = meuse$elev)

# building uncertainty model (based on uncertaintyError function)
uncertaintyModel = uncertaintyError(points)

# generate grid for IDW interpolation
gridDef.spat = Grid.def(points, TRUE) # SpatialPixels

# IDW interpolation process
IDW = idwUncertain(uncertaintyModel, gridDef.spat)

# visualization of results
Plot(IDW)
```

`uncertaintyPercent` *Creates S4 object class UncertainPoints*

Description

Builds an uncertainty model based on the percentual error over the input data. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyPercent(data, value = 2)
```

Arguments

`data`	Input data. S4 class of `Points`.
`value`	Percent value of imprecision that defines the uncertainty.

Value

Returns an object of class `UncertainPoints`.

See Also

`Points-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

`uncertaintyRandomDeviate`

Creates S4 object class UncertainPoints

Description

Builds an uncertainty model based on the imprecision made by random deviates over the input data. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyRandomDeviate(data, mean = 0, sd = 1)
```

Arguments

`data`	Input data. S4 class of `Points`.
`mean`	The mean value of input data values.
`sd`	The standart deviation of input data values.

Value

Returns an object of class `UncertainPoints`.

See Also

`Points-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

`uncertaintyRandomNumber`

Creates S4 object class UncertainPoints

Description

Builds an uncertainty model based on the imprecision made by random number (from defined interval) over the input data. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyRandomNumber(data, min = 0, max = 1)
```

Arguments

`data`	Input data. S4 class of `Points`.
`min`	The minimum value of the interval.
`max`	The maximum value of the interval.

Value

Returns an object of class `UncertainPoints`.

See Also

`Points-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

`uncertaintyRandomPercent`

Creates S4 object class UncertainPoints

Description

Builds an uncertainty model based on the imprecision made by random percent (from defined interval) over the input data. Input data must be type of S4 class `Points`. Output object is type of S4 class `UncertainPoints`.

Usage

```
uncertaintyRandomPercent(data, min = 0, max = 5)
```

Arguments

`data`	Input data. S4 class of `Points`.
`min`	The minimum value of the interval.
`max`	The maximum value of the interval.

Value

Returns an object of class `UncertainPoints`.

See Also

`Points-class`, `UncertainPoints-class`, `uncertaintyInterpolation2-package`

variogram *Estimating of variograms*

Description

This function can estimate variogram over the input data of S4 object class `UncertainPoints`. It is a bonus function of this whole package - just for user comfort. Output object is type of S4 class `Variogram` which can be printed by function `show` from UncerIn2 package. It also visualizes automaticaly all estimated variograms during the process.

Usage

```
## S4 method for signature 'UncertainPoints'
variogram(object, model = c("Sph", "Exp", "Gau", "Ste"),
  kappa = c(0.05, seq(0.2, 2, 0.1), 5, 10), fix.values = c(NA,NA,NA), verbose = FALSE,
   GLS.model = NA, start_vals = c(NA,NA,NA), miscFitOptions = list())
```

Arguments

`object`	Input data type of S4 object class UncertainPoints.
`model`	The list of variogrammodels that will be tested.
`kappa`	Smoothing parameter of the Matern model. Provide a list if you want to check more than one value.
`fix.values`	Can be used to fix a variogram parameter to a certain value. It consists of a list with a length of three. The items describe the fixed value for the nugget, range and sill respectively.
`verbose`	Logical, if TRUE the function will give extra feedback on the fitting process.
`GLS.model`	If a variogram model is passed on through this parameter a Generalized Least Squares sample variogram is calculated.
`start_vals`	Can be used to give the starting values for the variogram fitting. The items describe the fixed value for the nugget, range and sill respectively. They need to be given in that order. Setting the value to NA means that the value will be automatically chosen.
`miscFitOptions`	A list with named arguments that provide additional control over the fitting process. For example: list(merge.small.bins = TRUE). If the list is empty, autofitVariogram uses default values.

Details

For the estimation of variogram were used functions from package automap.

Value

Returns an object of class `Variogram`.

See Also

UncertainPoints-class, autofitVariogram, uncertaintyInterpolation2-package

Index

∗Topic **package**
    uncertaintyInterpolation2-information, 15

as.dataframe, 2
as.dataframe,Points-method *(*as.dataframe*)*, 2
as.dataframe,UncertainPoints-method *(*as.dataframe*)*, 2
as.UncertainPoints, 3
as.UncertainPoints,UncertainInterpolation-method *(*as.UncertainPoints*)*, 3
autofitVariogram, *21*
autoKrige, *8*

FuzzyInterpolation-class, 4

Grid.box, 4, *5–8*, *11*
Grid.box,Points-method *(*Grid.box*)*, 4
Grid.def, *5*, 5, *6–8*, *11*
Grid.def,Points-method *(*Grid.def*)*, 5
Grid.interpolation, *5*, 6, *7, 8*, *11*
Grid.interpolation,UncertainInterpolation-method *(*Grid.interpolation*)*, 6
gstat, *7*

idwUncertain, 6
idwUncertain,UncertainPoints,Spatial-method *(*idwUncertain*)*, 6

krigingUncertain, 7
krigingUncertain,Points,Spatial-method *(*krigingUncertain*)*, 7
krigingUncertain,UncertainPoints,Spatial-method *(*krigingUncertain*)*, 7

meuse, *12*

Plot, *7, 8*, 9, *11*
Plot,UncertainInterpolation-method *(*Plot*)*, 9
Points, *3*, 9
Points-class, 10

RFsimulate, *15*

splineUncertain, 10
splineUncertain,UncertainPoints,data.frame-method *(*splineUncertain*)*, 10

Tps, *11*
TUTORIAL, 11, *16*

UncertainInterpolation-class, 12
UncertainPoints-class, 13
Uncertainty Interpolation 2-package *(*uncertaintyInterpolation2-information*)*, 15
uncertaintyConstant, 14
uncertaintyError, 14
uncertaintyInterpolation2-information, 15
uncertaintyPercent, 17
uncertaintyRandomDeviate, 18
uncertaintyRandomNumber, 18
uncertaintyRandomPercent, 19

variogram, 20
variogram,UncertainPoints-method *(*variogram*)*, 20