

# Package ‘elsa’

July 22, 2025

**Type** Package

**Title** Entropy-Based Local Indicator of Spatial Association

**Version** 1.1-28

**Date** 2020-03-13

**Depends** methods, sp ( $\geq 1.2-0$ ), raster, R ( $\geq 3.0.0$ )

**Suggests** knitr, rmarkdown

## Description

A framework that provides the methods for quantifying entropy-based local indicator of spatial association (ELSA) that can be used for both continuous and categorical data. In addition, this package offers other methods to measure local indicators of spatial associations (LISA). Furthermore, global spatial structure can be measured using a variogram-like diagram, called entrogram. For more information, please check that paper: Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019) <[doi:10.1016/j.spasta.2018.10.001](https://doi.org/10.1016/j.spasta.2018.10.001)>.

**License** GPL ( $\geq 3$ )

**VignetteBuilder** knitr

**URL** <http://r-gis.net>

**BugReports** <https://github.com/babaknaimi/elsa/issues/>

**NeedsCompilation** yes

**Author** Babak Naimi [cre, aut] (ORCID: <<https://orcid.org/0000-0001-5431-2729>>),  
Roger Bivand [ctb] (part of the dnn C code, from the spdep package),  
William Venables [ctb] (part of the dnn C code, taken from the spdep package),  
Brian Ripley [ctb] (part of the dnn C code, taken from the spdep package)

**Maintainer** Babak Naimi <naimi.b@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-03-19 14:30:09 UTC

Contents

categorize . . . . .	2
correlogram . . . . .	3
dif2list . . . . .	4
dneigh . . . . .	6
elsa . . . . .	8
elsa.test . . . . .	10
entrogram . . . . .	11
Entrogram-class . . . . .	13
lisa . . . . .	13
moran . . . . .	15
nclass . . . . .	16
neighbours-class . . . . .	17
Variogram . . . . .	18
<b>Index</b>	<b>20</b>

---

categorize	<i>Categorizing continious data</i>
------------	-------------------------------------

---

Description

A function to categorize (discretize or binning) numerical data in the form of a raster layer, or a vector.

Usage

```
categorize(x,nc,probs,...)
```

Arguments

x	A RasterLayer or a numerical vector
nc	Number of classes, if missing, it will be automatically detected
probs	a two-item numeric vector of the lower and upper probabilities within the range of [0,1], e.g., c(0.025, 0.975) (default: NULL) to consider a quantile range of values in categorization (to avoid the effects of outliers); if NULL or FALSE, then the whole range is used.
...	Additional parameters for <a href="#">writeRaster</a> function (filename should be specified as additional argument)

Details

If nc is not specified, the function calls the function nclass to find the best number of classes. For the details of how nclass works, see the reference.

When outliers do exist in x, it is likely that they affect the quality of categorization. To avoid their affects, a quantile with the specified probabilities can be used. default is recommended: c(0.025,0.975), but to ignore it, you can turn the probs argument to NULL.

**Value**

RasterLayer      if x is a RasterLayer  
 numeric vector   if x is a numeric vector

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
categorize(1:10,3) # categorizeing a numeric vector into 3 categories

file <- system.file('external/dem_example.grd',package='elsa')
r <- raster(file)
plot(r,main='a continuous raster map')
rc <- categorize(r,nc=4)
plot(rc, main='categorized map')
```

---

 correlogram

---

*Correlogram of Spatial Data*


---

**Description**

Compute Correlogram of spatial data. The function returns a binned correlogram by calculating Moran's I (spatial autocorrelation) in different range of distances.

**Usage**

```
correlogram(x, width, cutoff,...)
```

**Arguments**

x	a spatial object (RasterLayer or SpatialPointsDataFrame or SpatialPolygonsDataFrame)
width	the lag size (width of subsequent distance intervals) into which cell pairs are grouped for semivariance estimates. If missing, the cell size (raster resolution) is assigned.
cutoff	spatial separation distance up to which cell pairs are included in semivariance estimates; as a default, the length of the diagonal of the box spanning the data is divided by three.

... Additional arguments including `zcol` (when `x` is `Spatial*` object, specifies the name of the variable in the dataset; `longlat` (when `x` is `Spatial*` object, specifies whether the dataset has a geographic coordinate system); `s` (only when `x` is a `Raster` object, it would be useful when the dataset is big, so then by specifying `s`, the calculation would be based on a sample with size `s` drawn from the dataset, default is `NULL` means all cells should be contributed in the calculations)

## Details

Correlogram is a graph to explore spatial structure in a single variable. A correlogram summarizes the spatial relations in the data, and can be used to understand within what range (distance) the data is spatially autocorrelated.

## Value

Correlogram      an object containing Moran's I values within each distance interval

## Author(s)

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

## References

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

## Examples

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)
plot(r, main='a continuous raster map')

co <- correlogram(r, width=2000, cutoff=30000)

plot(co)
```

---

dif2list

*Convert differences in the level of categorical map to a list*

---

## Description

This function converts a data.frame including the values specifying the differences (contrast or degrees of dissimilarities) between the classes (categories) in a categorical map, to a list.

**Usage**

```
dif2list(x,pattern,fact=1)
```

**Arguments**

<code>x</code>	a data.frame containing degrees of dissimilarities between different categories
<code>pattern</code>	a numeric vector, specifies the pattern the data is organised in <code>x</code> ; (only when the number of classes or subclasses is greater than 9; see the examples)
<code>fact</code>	a numeric value (default = 1), specifies the factor that multiplied to the estimated dissimilarities

**Details**

When ELSA is calculated for a categorical map, by default it is assumed that the level of dissimilarities (or level of contrast) between different classes are the same. For example, if a categorical map has four classes of "A","B","C", and "D", dissimilarity or contrast between "A" & "B" is the same as between "A" & "C", or between "C" & "D", etc. Sometimes, it is not a valid assumption as some classes might be more similar than the others. For example, a landuse map may contain several classes of forest, and several classes of agriculture. The level of dissimilarity between a class of forest and a class of agriculture is not the same as between two different types of forest.

ELSA is flexible enough to incorporate different levels of dissimilarity. To do that, a list can be specified in which you specify the contrast or dissimilarity between each class (name of an item in the list) to the other classes (named vector with items corresponding to the other classes). For instance, `list(A=c(A=0,B=1,C=2,D=2), B=c(A=1,B=0,C=2,D=2),...)` simply specifies the dissimilarities between class A and class B (the first two items in the list) to the four classes of A, B, C, and D (the vectors assigned to the first and second item in the list).

Alternatively, a simple coding approach can be used when the classes can be organised in a hierarchical way (see Naimi et al. (2019) for more details), and if that is the case, a code can be assigned to each class within a data.frame. Then, `dif2list` function can be used to convert the data.frame to a list with the structure required by the function (like the above example). Using `dif2list` is not necessary to introduce the differences between different categories, as a user can either specify them directly in a list. However, defining them in a data.frame would be easier specially when the classes are in related in a hierarchical form.

In the data.frame required by `dif2list`, a code is assigned to each class. The code is specified according to a hierarchical structure and the number of levels in it. For instance, a land cover may have four classes including forest broadleaves, forest needle-leaves, cropland rainfed, and cropland irrigated. These four classes can be organised hierarchically at two levels, the first level would be forest and cropland, and the second level would be the name of the classes. The codes assigned to these four classes can be 11, 12, 21, and 22, respectively. The dissimilarities between, for example, classes of 11 & 12 would be 1 and between 11 & 21 would be 2. The estimated dissimilarities can be adjusted by changing the `fact` that is multiplied to the dissimilarity values.

**Value**

A list

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
# imagine we have a categorical map including 4 classes (values 1:4), and the first two classes
# (i.e., 1 and 2) belong to the major class 1 (so can have legends of 11, 12, respectively), and
# the second two classes (i.e, 3 and 4) belong to the major class 2 (so can have legends of 21,
# and 22 respectively). Then we can construct the data.frame as:

d <- data.frame(g=c(1,2,3,4),leg=c(11,12,21,22))

d

# dif2list generates a list including 4 values each corresponding to each value (class in the map
# , i.e, 1:4). Each item then has a numeric vector containing a relative dissimilarity between the
# main class (the name of the item in the list) and the other classes. If one wants to change
# the relative dissimilarity between two specific classes, then the list can easily be edited and
# used in the elsa function

dif2list(d)

# As you see in the legend, each value contains a sequence of numbers specifying the class,
# subclass, sub-subclass, .... and so on in a hierarchical manner (for example, 12 means class 1
# and subclass 2). In case if there is more than 9 classes or subclasses (for example, 112 should
# refer to class 1, and subclass 12, not class 1 , subclass 1, and sub-subclass 2), then the
# pattern should be specified as a vector like c(1,2) means that the length of the major class in
# the hierarchy is 1, while the length of the sub class is 2.

d <- data.frame(g=c(1,2,3,4),leg=c(101,102,201,202))

d

dif2list(d,pattern=c(1,2))

dif2list(d,pattern=c(1,2), fact=2) # dissimilarities are multiplied by 2 (fact=2).
```

## Description

This function identifies the neighbours features (points or polygons) given the specified distance (in kilometer for geographic coordinates, i.e., if `longlat=TRUE`; and in the map unit for projected datasets, i.e., if `longlat = FALSE`) and builds a list of neighbours.

The `neighd` function returns a list including distance of each feature to neighbourhood locations.

## Usage

```
dneigh(x,d1,d2,longlat,method,...)
```

```
neighd(x,d1,d2,longlat,...)
```

## Arguments

<code>x</code>	a <code>SpatialPoints</code> , or <code>SpatialPolygons</code> or a matrix (or <code>data.frame</code> ) of point coordinates or a <code>SpatialPoints</code> object
<code>d1</code>	lower local distance bound (if <code>longlat = TRUE</code> , in kilometer; otherwise in the spatial unit of the dataset, e.g., meter)
<code>d2</code>	upper local distance bound (if <code>longlat = TRUE</code> , in kilometer; otherwise in the spatial unit of the dataset, e.g., meter)
<code>longlat</code>	<code>TRUE</code> if point coordinates are longitude-latitude
<code>method</code>	if <code>x</code> is <code>SpatialPolygons</code> , specifies the method to identify the neighbour polygons; see details
<code>...</code>	additional arguments; see details

## Details

The function is mostly based on `dnearneigh` (for points), and `poly2nb` (for polygons), implemented in the `spdep` package by Roger Bivand.

When `x` is `SpatialPolygons`, there is two methods (can be specified through `method`) to identify the neighbour polygons. The default method (`'bound'`) seeks the polygons that has one or more points in their boundaries within the specified distance (`d`), while the method `'centroid'` considers any polygon with a centroid within the given distance.

One additional argument is `queen` (default is `TRUE`), can be used only when `x` is `SpatialPolygons`, and `method='bound'`, if `TRUE`, a single shared boundary point meets the contiguity condition, if `FALSE`, more than one shared point is required.

`neighd` for `SpatialPolygons` returns distances of each polygon to centroids of neighbor polygons.

## Value

An object of class `neighbours`

## Author(s)

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

## References

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

## Examples

```
#
```

---

```
elsa
```

---

*Entropy-based Local indicator of Spatial Association*

---

## Description

Calculate ELSA statistic for a categorical or continuous spatial dataset.

## Usage

```
elsa(x,d,nc,categorical,dif,classes,stat,...)
```

## Arguments

<code>x</code>	a raster object ( <code>RasterLayer</code> or <code>SpatialPointsDataFrame</code> or <code>SpatialPolygonsDataFrame</code> )
<code>d</code>	numeric local distance, or an object of class <code>neighbours</code> created by <code>dneigh</code> when <code>x</code> is <code>SpatialPoints</code> or <code>SpatialPolygons</code>
<code>nc</code>	optional, for continuous data it specifies the number of classes through categorizing the variable. If missing, it is automatically calculated (recommended)
<code>categorical</code>	logical, specified whether <code>x</code> is a continuous or categorical. If missed the function tries to detect it
<code>dif</code>	the difference between categories, only for categorical
<code>classes</code>	Optional, only when <code>x</code> is categorical is a character vector contains classes; would be useful when the dataset is part of a bigger dataset or when it does not contain all the categories, then by specifying the full set of categories, they will be taken into account to calculate ELSA, and therefore, it would be comparable with the other dataset with the same list of classes (these classes may alternatively introduce by <code>dif</code> , as the classes considered to specify dissimilarities in <code>dif</code> list, would be used as classes)
<code>stat</code>	specifies which statistic should be calculated by the function; it can be "elsa" (default), or either of the two components of the statistic, "Ea", or "Ec"; ELSA is the product of Ea and Ec. (it is possible to select more than one statistic); this argument is ignored if <code>x</code> is <code>Spatial*</code> object as all the three statistics are returned (see details)



... additional arguments including:

- `cells` - a numeric vector to specify for which raster cells the ELSA statistic should be calculated; it works when `x` is `RasterLayer`, and if it is specified, ELSA is calculated only for the specified cells;
- `filename` - only if `x` is `RasterLayer`, specifies the name of the raster file to be written in the working directory.
- `zcol` - only if `x` is `SpatialPointsDataFrame` or `SpatialPolygonsDataFrame`, specifies the name of the column (variable) in the attribute table for which the elsa statistics are calculated (i.e., ELSA, Ea, Ec).
- `drop` - logical; only if `x` is `SpatialPointsDataFrame` or `SpatialPolygonsDataFrame`, specifies whether the output should be a `data.frame` or a `Spatial*` object.
- `method` - only if `SpatialPolygonsDataFrame`, specifies the method for identifying the neighbourhood polygons; default: 'centroid' (see `dneigh`).

## Details

`dif` can be used when categorical values are sorted into hierarchical system (e.g., CORINE land cover). This make it possible to difine different weights of similarity between each pairs of categories when the level of similarity is not the same between different classes in the variable. For example, two categories belong to two forest types are more similar than two categories, one a forest type and the other one an agriculture type. So, it can take this differences into account when the spatial autocorrelation for categorical variables is quantified.

the ELSA statistics has two terms, "Ea" and "Ec", in the reference. It can be specified in the `stat` argument if either of these terms should be returned from the function or ELSA ("E"), which is the product of these two terms,  $Ea * Ec$ . All three terms can also be selected.

## Value

`Raster*` if `x` is a `RasterLayer`  
`Spatial*` or `data.frame` if `x` is a `Spatial*`

## Author(s)

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

## References

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

## Examples

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)

plot(r, main='a continuous raster map')
```

```
e <- elsa(r,d=2000,categorical=FALSE)

plot(e)
```

---

elsa.test

*Elsa test for local spatial autocorrelation*


---

## Description

This function uses a non-parametric approach to test whether local spatial autocorrelation (characterised by ELSA) is significant. It generates a p-value at each spatial location (a raster cell or spatial point/polygon) that can be used to infer the significance of local spatial autocorrelation.

## Usage

```
elsa.test(x, d, n, method, null, nc, categorical, dif, classes, ...)
```

## Arguments

x	A Raster or Spatial* dataset
d	the local distance, or an object of class neighbours created by dneigh function
n	number of simulation, default is 999 for small datasets, and 99 for large datasets
null	Optional, a null distribution of data (a Raster if x is Raster or a numerical vector if x is either Raster or Spatial dataset ); if not provided, a null distribution is generated by the function
method	resampling method for nonparametric simulation, can be either 'boot' (bootstrapping; default) or 'perm' (permutation)
nc	number of classes (only if x is a continuous variable); if not specified, it is estimated using nclass function
categorical	logical, specifies whether x is a categorical; if not specified, it is guessed by the function
dif	the level of dissimilarities between different categories (only if x is a categorical variable); see <a href="#">dif2list</a> for more details
classes	Optional, only when x is categorical is a character vector contains classes; would be useful when the dataset is part of a bigger dataset or when it does not contain all the categories, then by specifying the full set of categories, they will be taken into account to calculate ELSA, and therefore, it would be comparable with the other dataset with the same list of classes (these classes may alternatively introduce by dif, as the classes considered to specify dissimilarities in dif list, would be used as classes)
...	Additional arguments passed to writeRaster function (applied only when x is Raster)

**Details**

This function test how significant the local spatial autocorrelation is at each location, so it generates a p-value at each location through a Monte Carlo simulation and a non-parametric approach. See the reference for the details about the method.

If null distribution is not provided, the function generates a null distribution by randomly shuffling the values in the dataset.

**Value**

An object same as the input (x)

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
file <- system.file('external/dem_example.grd',package='elsa')

r <- raster(file)

plot(r,main='a continuous raster map')

et <- elsa.test(r,d=2000,n=99, categorical=FALSE)

plot(et)
```

---

entrogram

*Entrogram for Spatial Data*


---

**Description**

Compute sample (empirical) entrogram from spatial data. The function returns a binned entrogram and an entrogram cloud.

**Usage**

```
entrogram(x, width, cutoff,...)
```

**Arguments**

<code>x</code>	a spatial object (RasterLayer or SpatialPoints or SpatialPolygons)
<code>width</code>	the lag size (width of subsequent distance intervals) into which cell pairs are grouped for ELSA estimates. If missing, the cell size (raster resolution) is assigned.
<code>cutoff</code>	spatial separation distance up to which cell pairs are included in ELSA estimates; as a default, the length of the diagonal of the box spanning the data is divided by three.
<code>...</code>	Additional arguments including <code>zcol</code> (when <code>x</code> is Spatial* object, specifies the name of the variable in the dataset; <code>longlat</code> (when <code>x</code> is Spatial* object, specifies whether the dataset has a geographic coordinate system); <code>s</code> (only when <code>x</code> is a Raster object, it would be useful when the dataset is big, so then by specifying <code>s</code> , the calculation would be based on a sample with size <code>s</code> drawn from the dataset, default is NULL means all cells should be contributed in the calculations)

**Details**

Entrogram is a variogram-like graph to explore spatial structure in a single variable. An entrogram summarizes the spatial relations in the data, and can be used to understand within what range (distance) the data is spatially autocorrelated.

**Value**

Entrogram	an object containing entrogram cloud and the entrogram within each distance interval
-----------	--

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)
plot(r, main='a continuous raster map')

en <- entrogram(r, width=2000)

plot(en)
```

---

Entrogram-class	<i>An S4 class for Entrogram</i>
-----------------	----------------------------------

---

**Description**

An S4 class representing Entrogram dataset generated by entrogram function

**Slots**

width the width of susequent distance intervals  
 cutoff spatial separation distance up to which point pairs are included in entrogram estimates  
 entrogramCloud A matrix containing the elsa for each location and the neighbourhood distance  
 entrogram A data.frame containing mean elsa values at each distance interval

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

---

lisa	<i>Local indicators of Spatial Associations</i>
------	---

---

**Description**

Calculate local indicators of spatial association (LISA) for a continuous (numeric) variable at each location in a Raster layer or a SpatialPointsDataFrame or a SpatialPolygonsDataFrame.

**Usage**

```
lisa(x,d1,d2,statistic,...)
```

**Arguments**

x	a raster object (RasterLayer or SpatialPointsDataFrame or SpatialPolygonsDataFrame)
d1	numeric lower bound of local distance (default=0), or an object of class neighbours created by dneigh when x is SpatialPoints or SpatialPolygons
d2	numeric upper bound of local distance, not needed if d1 is a neighbours object,

**statistic** a character string specifying the LISA statistic that should be calculated. This can be one of "I" (or "localmoran" or "moran"), "c" (or "localgeary" or "geary"), "G" (or "localG"), "G\*" (or "localG\*")

**...** additional arguments including filename (only when x is Raster, specifies the name of the raster file when the output should be written; additional arguments for `writeRaster` function can also be specified); mi (only when x is Raster and statistic='I', specifies whether raw Local Moran's I statistic (Ii) should be returned, or standardized value (Z.Ii). e.g., mi="I", mi='Z' (default)); zcol (only when x is a Spatial\* object specifies the name of the variable column in the data); longlat (logical, only when x is a Spatial\* object specifies whether the coordinate system is geographic); drop (logical, only when x is a Spatial\* object, if TRUE, the original data structure (Spatial\* object) is returned, otherwise a numeric vector is returned)

### Details

This function can calculate different LISA statistics at each location in the input dataset. The statistics, implemented in this function, include local Moran's I ("I"), local Geary's c ("c"), local G and G\* ("G" and "G\*"). This function returns standardized value (Z) for Moran, G and G\*.

### Value

RasterLayer if x is a RasterLayer  
 Spatial\* if x is a Spatial\* and drop=FALSE  
 numeric vector if x is a Spatial\* and drop=TRUE

### Author(s)

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

### References

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

Anselin, L. 1995. Local indicators of spatial association, *Geographical Analysis*, 27, 93–115;

Getis, A. and Ord, J. K. 1996 Local spatial statistics: an overview. In P. Longley and M. Batty (eds) *Spatial analysis: modelling in a GIS environment* (Cambridge: Geoinformation International), 261–277.

### Examples

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)

plot(r, main='a continuous raster map')

mo <- lisa(r, d2=2000, statistic='i') # local moran's I (Z.Ii value)
```

```

plot(mo, main="local Moran's I (Z.Ii)")

mo <- lisa(r,d2=2000,statistic='i',mi='I') # local moran's I (Ii value (non-standardized))

plot(mo, main="local Moran's I (Ii))")

gc <- lisa(r,d2=2000,statistic='c') # local Geary's c

plot(gc, main="local Geary's c")

g <- lisa(r,d2=2000,statistic='g') # local G

plot(g, main="local G")

```

---

moran

*Global Spatial Autocorrelation Statistics*


---

## Description

Functions to calculate Moran's I and Geary's c statistics.

## Usage

```

moran(x,d1,d2,...)
geary(x,d1,d2,...)

```

## Arguments

x	a raster object (RasterLayer or SpatialPointsDataFrame or SpatialPolygonsDataFrame)
d1	lower bound local distance, or an object of class neighbours created by dneigh when x is SpatialPoints or SpatialPolygons
d2	upper bound local distance
...	additional arguments including zcol (when x is Spatial* object, specifies the name of the variable in the dataset; longlat (when x is Spatial* object, specifies whether the dataset has a geographic coordinate system

## Details

moran and geary are two functions to measure global spatial autocorrelation within the range of distance specified through d1 and d2. It returns a single numeric value than can show the degree of spatial autocorrelation in the whole dataset.

## Value

A numeric value.

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)

moran(r, d1=0, d2=2000)

geary(r, d1=0, d2=2000)
```

---

nclass

*Best number of classes for categorizing a continuous variable*


---

**Description**

This function explores the best number of classes to categorize (discretize) a continuous variable.

**Usage**

```
nclass(x, th, ...)
```

**Arguments**

x	a RasterLayer or a numeric vector
th	A threshold (default = 0.005) used to find the best number of classes
...	Additional arguments; currently probs implemented that specifies which extreme values (outliers) should be ignored; specified as a percentile probabilities, e.g., c(0.005,0.995), default is NULL

**Details**

The function uses an approach introduced in Naimi et al. (under review), to find the best number of classes (categories) when a continuous variable is discretizing. The threshold is corresponding to the acceptable level of information loose through discretizing procedure. For the details, see the reference.

**Value**

An object with the same class as the input x



**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
file <- system.file('external/dem_example.grd', package='elsa')
r <- raster(file)
plot(r, main='a continuous raster map')

nclass(r)

nclass(r, th=0.01)

nclass(r, th=0.1)
```

---

neighbours-class	<i>An S4 class of neighbour features</i>
------------------	--

---

**Description**

An S4 class representing a list of neighbour features generated by dneigh function

**Slots**

distance1 the distance from which the neighbour features are sought  
 distance2 the distance up to which the neighbour features are sought  
 neighbours A list containing numeric vectors each specifies which the neighbours for the corresponding feature

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

## Variogram

*Empirical Variogram from Spatial Data***Description**

Compute sample (empirical) variogram from spatial data. The function returns a binned variogram and a variogram cloud.

**Usage**

```
Variogram(x, width, cutoff,...)
```

**Arguments**

x	a spatial object (RasterLayer or SpatialPointsDataFrame or SpatialPolygonsDataFrame)
width	the lag size (width of subsequent distance intervals) into which cell pairs are grouped for semivariance estimates. If missing, the cell size (raster resolution) is assigned.
cutoff	spatial separation distance up to which cell pairs are included in semivariance estimates; as a default, the length of the diagonal of the box spanning the data is divided by three.
...	Additional arguments including cloud that specifies whether a variogram cloud should be included to the output (default is FALSE), zcol (when x is Spatial* object, specifies the name of the variable in the dataset; longlat (when x is Spatial* object, specifies whether the dataset has a geographic coordinate system; s (only when x is a Raster object, it would be useful when the dataset is big, so then by specifying s, the calculation would be based on a sample with size s drawn from the dataset, default is NULL means all cells should be contributed in the calculations)

**Details**

Variogram is a graph to explore spatial structure in a single variable. A variogram summarizes the spatial relations in the data, and can be used to understand within what range (distance) the data is spatially autocorrelated.

**Value**

Variogram	an object containing variogram cloud and the variogram within each distance interval
-----------	--

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

Naimi, B., Hamm, N. A., Groen, T. A., Skidmore, A. K., Toxopeus, A. G., & Alibakhshi, S. (2019). ELSA: Entropy-based local indicator of spatial association. *Spatial statistics*, 29, 66-88.

**Examples**

```
file <- system.file('external/dem_example.grd',package='elsa')
r <- raster(file)
plot(r,main='a continuous raster map')

en <- Variogram(r, width=2000)

plot(en)
```

# Index

## \* **spatial**

- categorize, [2](#)
  - correlogram, [3](#)
  - dif2list, [4](#)
  - dneigh, [6](#)
  - elsa, [8](#)
  - elsa.test, [10](#)
  - entrogram, [11](#)
  - Entrogram-class, [13](#)
  - lisa, [13](#)
  - moran, [15](#)
  - nclass, [16](#)
  - neighbours-class, [17](#)
  - Variogram, [18](#)
- categorize, [2](#)
- categorize, list-method (categorize), [2](#)
- categorize, numeric-method (categorize), [2](#)
- categorize, RasterLayer-method (categorize), [2](#)
- categorize, RasterStackBrick-method (categorize), [2](#)
- correlogram, [3](#)
- correlogram, RasterLayer-method (correlogram), [3](#)
- correlogram, Spatial-method (correlogram), [3](#)
- Correlogram-class (Entrogram-class), [13](#)
- data.frameORmatrix-class (Entrogram-class), [13](#)
- dif2list, [4](#), [10](#)
- dif2list, data.frameORmatrix, ANY-method (dif2list), [4](#)
- dif2list, data.frameORmatrix-method (dif2list), [4](#)
- dneigh, [6](#)
- dneigh, data.frameORmatrix-method (dneigh), [6](#)
- dneigh, SpatialPoints-method (dneigh), [6](#)
- dneigh, SpatialPolygons-method (dneigh), [6](#)
- elsa, [8](#)
- elsa, RasterLayer-method (elsa), [8](#)
- elsa, SpatialPointsDataFrame-method (elsa), [8](#)
- elsa, SpatialPolygonsDataFrame-method (elsa), [8](#)
- elsa.test, [10](#)
- elsa.test, RasterLayer-method (elsa.test), [10](#)
- elsa.test, Spatial-method (elsa.test), [10](#)
- entrogram, [11](#)
- entrogram, RasterLayer-method (entrogram), [11](#)
- entrogram, SpatialPointsDataFrame-method (entrogram), [11](#)
- entrogram, SpatialPolygonsDataFrame-method (entrogram), [11](#)
- Entrogram-class, [13](#)
- geary (moran), [15](#)
- geary, RasterLayer-method (moran), [15](#)
- geary, Spatial-method (moran), [15](#)
- lisa, [13](#)
- lisa, RasterLayer-method (lisa), [13](#)
- lisa, Spatial-method (lisa), [13](#)
- matrixORnull-class (Entrogram-class), [13](#)
- moran, [15](#)
- moran, RasterLayer-method (moran), [15](#)
- moran, Spatial-method (moran), [15](#)
- nclass, [16](#)
- nclass, numeric-method (nclass), [16](#)
- nclass, RasterLayer-method (nclass), [16](#)
- neighbours-class, [17](#)
- neighd (dneigh), [6](#)

neighd,data.frameORmatrix-method  
    (dneigh), [6](#)  
neighd,SpatialPoints-method (dneigh), [6](#)  
neighd,SpatialPolygons-method (dneigh),  
    [6](#)  
  
Variogram, [18](#)  
Variogram,RasterLayer-method  
    (Variogram), [18](#)  
Variogram,Spatial-method (Variogram), [18](#)  
Variogram-class (Entrogram-class), [13](#)  
  
writeRaster, [2](#), [14](#)