Package: bispdep (via r-universe)

August 30, 2024

Type Package

Title Statistical Tools for Bivariate Spatial Dependence Analysis

Version 1.0-1

Date 2024-08-20

Author Carlos Melo [aut, cre]
(<https://orcid.org/0000-0002-5598-1913>), Oscar Melo [ctb]
(<https://orcid.org/0000-0002-0296-4511>), Sandra Melo [ctb]
(<https://orcid.org/0000-0002-4875-7657>)

Depends R (>= 3.3.0), methods, spData, sf

Imports spdep, spatialreg, stats, combinat, boot (>= 1.3-1), graphics, grDevices, ggplot2, sp (>= 1.0), RColorBrewer

Suggests parallel, spam

Maintainer Carlos Melo <cmelo@udistrital.edu.co>

Description A collection of functions to test spatial autocorrelation between variables, including Moran I, Geary C and Getis G together with scatter plots, functions for mapping and identifying clusters and outliers, functions associated with the moments of the previous statistics that will allow testing whether there is bivariate spatial autocorrelation, and a function that allows identifying (visualizing neighbours) on the map, the neighbors of any region once the scheme of the spatial weights matrix has been established.

License GPL (>= 2)

URL https://github.com/carlosm77/bispdep

Encoding UTF-8

RoxygenNote 7.2.3

Repository https://carlosm77.r-universe.dev

RemoteUrl https://github.com/carlosm77/bispdep

RemoteRef HEAD

RemoteSha 0421c220ed43dec820a10eb6b2fa42a648941fab

Contents

`connectivity.map` *Map of spatial connectivity*

Description

Generates a map of spatial connectivity according to a predefined neighborhood scheme, around a region selected by the user by clicking on the screen.

Usage

```
connectivity.map(nb,polygons,var.label,obs,col,cex, ...)
```

Arguments

`nb`	an object of class `nb` with a list of integer vectors containing neighbour region number ids
`polygons`	the spatial dataset: sf or SpatialPolygonsDataFrame (spdep)
`var.label`	variable to label in the connectivity neighborhood
`obs`	number of the observation around which neighborhood connectivity will be performed. by default is NULL, the user can press the (first) mouse button over the map region for which he wishes to identify his neighborhood.
`col`	color to assign in the regions of the defined neighborhood
`cex`	text size on neighborhood labels
`...`	further specifications, see plot_sf and plot and details.

Value

around a region selected by the user by clicking on the screen A spatial connectivity map showing the neighbors according to the weighting scheme.

See Also

poly2nb, knn2nb, graph2nb

Examples

```
library(spdep)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
#Queen neighbours
col_nbq1 <- poly2nb(columbus)             # queen, order 1
col.lags10 <- nblag(col_nbq1, 10)         # queen, up to order 10
connectivity.map(col.lags10[[1]],columbus,"COLUMBUS_I",obs=25,col="green",cex=0.5)

sf_obj <- st_centroid(st_geometry(columbus), of_largest_polygon)
sp_obj <- as(sf_obj, "Spatial")
coords <- st_coordinates(sf_obj)

col.k4 <- knn2nb(knearneigh(coords,4))
connectivity.map(col.k4,columbus,"COLUMBUS_I",obs=25,col="green",cex=0.5)

suppressMessages(trinb <- tri2nb(coords))
connectivity.map(trinb,columbus,"COLUMBUS_I",obs=25,col="green",cex=0.5)

gabrielnb=graph2nb(gabrielneigh(coords),sym=TRUE)
connectivity.map(gabrielnb,columbus,"COLUMBUS_I",obs=25,col="green",cex=0.5)
```

---

correlogram *Computes Moran's or Geary's coefficients on distance classes*

Description

Computes Moran's or Geary's coefficients on distance classes from a set of spatial coordinates and corresponding z values

Usage

```
correlogram(coords, z, method="Moran", nbclass = NULL, zero.policy=NULL, ...)
```

Arguments

coords	a two columns array, data.frame or matrix of spatial coordinates. Column 1 = X, Column 2 = Y.
z	a vector for the values at each location. Must have the same length as the row number of coords

method	the method used. Must be "Moran" (default) or "Geary"
nbclass	number of bins. If NULL Sturges method is used to compute an optimal number
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
...	further arguments to pass to e.g. moran.test or geary.test

Details

Uses the library spdep including moran.test or geary.test. Distances are euclidian and in the same unit as the spatial coordinates. Moran's Ho: I values larger than 0 due to chance; Geary's Ho: C values lesser than 1 due to chance. correlogram has print and plot methods; statistically significant values (p<0.05) are plotted in red.

Value

An object of class "correlogram", a matrix including:

class	bin centers
I	the coefficient values
p.value	probability of Ho
n	the number of pairs
low.l	bin lower limit
up.l	bin upper limit

Warning

Computing can take a long time for large data sets

References

see library spdep

See Also

geary.test, moran.test

Examples

```
library(spdep)
data(oldcol)
attach(COL.OLD)
coords<-cbind(X,Y)
corM <- correlogram(coords,CRIME,zero.policy=TRUE)
corM
plot(corM)

corG <- correlogram(coords,CRIME,method="Geary",zero.policy=TRUE)
corG
plot(corG)
```

correlogram.bi *Computes Bivariate Moran's or Geary's coefficients on distance classes*

Description

Computes Bivariate Moran's or Geary's coefficients on distance classes from a set of spatial coordinates and values of the two corresponding variables varX and varY.

Usage

```
correlogram.bi(coords, varX, varY, method="Moran", nbclass = NULL, zero.policy=NULL, ...)
```

Arguments

coords	a two columns array, data.frame or matrix of spatial coordinates. Column 1 = X, Column 2 = Y.
varX	a vector for the values of the variable x at each location. Must have the same length as the row number of coords
varY	a vector for the values of the variable y at each location. Must have the same length as the row number of coords
method	the method used. Must be "Moran" (default) or "Geary"
nbclass	number of bins. If NULL Sturges method is used to compute an optimal number
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
...	further arguments to pass to e.g. moranbi.test or gearybi.test

Details

Uses the library bispdep including moranbi.test or gearybi.test. Distances are euclidian and in the same unit as the spatial coordinates. Moran's Ho: I values larger than 0 due to chance; Geary's Ho: C values lesser than 1 due to chance, correlogrambi has print and plot methods; statistically significant values (p<0.05) are plotted in red.

Value

An object of class "correlogrambi", a matrix including:

class	bin centers
I	the coefficient values
p.value	probability of Ho
n	the number of pairs
low.l	bin lower limit
up.l	bin upper limit

Warning

Computing can take a long time for large data sets

References

see library bispdep

See Also

gearybi.test, moranbi.test

Examples

```
library(spdep)
library(sf)
data(oldcol)
attach(COL.OLD)
coords<-cbind(X,Y)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))


corbiM <- correlogram.bi(coords,columbus$CRIME,columbus$INC,zero.policy=TRUE)
corbiM
plot(corbiM)

corbiG <- correlogram.bi(coords,columbus$CRIME,columbus$INC,method="Geary",zero.policy=TRUE)
corbiG
plot(corbiG)
```

---

geary.bi *Compute Bivariate Geary's Cxy*

Description

A simple function to compute Bivariate Geary's Cxy;

$$C_{xy} = \frac{(n-1)}{2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - y_j)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Usage

```
geary.bi(varX, varY, listw, zero.policy=NULL, adjust.n = TRUE,
         alternative = "greater")
```

Arguments

varX	a numeric vector the same length as the neighbours list in listw with the values of the variable x
varY	a numeric vector the same length as the neighbours list in listw with the values of the variable y
listw	a listw object created for example by nb2listw
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.

Value

a list with

C	Bivariate Geary's C
Kx	sample kurtosis of x
Ky	sample kurtosis of y

References

Wartenberg, Daniel. 2010. Multivariate Spatial Correlation: A Method for Exploratory Geographical Analysis. Geographical Analysis. 17. 263 - 283. 10.1111/j.1538-4632.1985.tb00849.x.

See Also

gearybi.test

Examples

```
library(spdep)
data(oldcol)
col.W <- nb2listw(COL.nb, style="W")
geary.bi(COL.OLD$CRIME, COL.OLD$INC, col.W, zero.policy = TRUE)
```

---

gearybi.test *Bivariate Geary's Cxy test for spatial autocorrelation*

Description

Bivariate Geary's test for spatial autocorrelation using a spatial weights matrix in weights list form. The assumptions underlying the test are sensitive to the form of the graph of neighbour relationships and other factors.

Usage

```
gearybi.test(varX, varY, listw, randomisation=TRUE, zero.policy=NULL,
    alternative="greater", spChk=NULL, adjust.n=TRUE)
```

Arguments

varX	a numeric vector the same length as the neighbours list in listw with the values of the variable X
varY	a numeric vector the same length as the neighbours list in listw with the values of the variable Y
listw	a listw object created for example by nb2listw
randomisation	variance of I calculated under the assumption of randomisation, if FALSE normality
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
alternative	a character string specifying the alternative hypothesis, must be one of "greater" (default), "less" or "two.sided".
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted

Value

A list with class htest containing the following components:

statistic	the value of the standard deviate of Geary's Cxy, in the order given in Cliff and Ord 1973, p. 21, which is (ECxy - Cxy) / sqrt(VCxy), that is with the sign reversed with respect to the more usual (Cxy - ECxy) / sqrt(VCxy); this means that the "greater" alternative for the Bivariate Geary Cxy test corresponds to the "greater" alternative for Bivariate Moran's Ixy test.
p.value	the p-value of the test.
estimate	the value of the observed BIvariate Geary's Cxy, its expectation and variance under the method assumption.
alternative	a character string describing the alternative hypothesis.
method	a character string giving the assumption used for calculating the standard deviate.
data.name	a character string giving the name(s) of the data.

Note

The derivation of the test (Cliff and Ord, 1981, p. 18) assumes that the weights matrix is symmetric. For inherently non-symmetric matrices, such as k-nearest neighbour matrices, listw2U() can be used to make the matrix symmetric. In non-symmetric weights matrix cases, the variance of the test statistic may be negative (thanks to Franz Munoz I for a well documented bug report). Geary's C is affected by non-symmetric weights under normality much more than Moran's I. From 0.4-35, the sign of the standard deviate of C is changed to match Cliff and Ord (1973, p. 21).

References

Cliff, A. D., Ord, J. K. 1981 Spatial processes, Pion, p. 21, Cliff, A. D., Ord, J. K. 1973 Spatial Autocorrelation, Pion, pp. 15-16, 21; Bivand RS, Wong DWS 2018 Comparing implementations of global and local indicators of spatial association. TEST, 27(3), 716–748 doi:10.1007/s11749018-0599x

See Also

geary.bi, listw2U

Examples

```
library(spdep)
data(oldcol)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(COL.nb,style="W"),zero.policy=TRUE)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(COL.nb,style="W"),zero.policy=TRUE,
             randomisation=FALSE)
colold.lags <- nblag(COL.nb, 3)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(colold.lags[[2]],style="W"),
             zero.policy=TRUE)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(colold.lags[[3]],style="W"),
             zero.policy=TRUE,alternative="greater")
print(is.symmetric.nb(COL.nb))
coords.OLD <- cbind(COL.OLD$X, COL.OLD$Y)
COL.k4.nb <- knn2nb(knearneigh(coords.OLD, 4))
print(is.symmetric.nb(COL.k4.nb))
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(COL.k4.nb,style="W"),zero.policy=TRUE)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,nb2listw(COL.k4.nb,style="W"),zero.policy=TRUE,
             randomisation=FALSE)
cat("Note non-symmetric weights matrix - use listw2U()\n")
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,listw2U(nb2listw(COL.k4.nb,style="W")),
             zero.policy=TRUE)
gearybi.test(COL.OLD$CRIME,COL.OLD$INC,listw2U(nb2listw(COL.k4.nb,style="W")),
             zero.policy=TRUE,randomisation=FALSE)
```

`getis.cluster` *Getis and Ord's Gi* Cluster and Significance Map*

Description

Create the Getis Gi* Cluster Map and the corresponding Significance Map. Maps are done calculating the Local Gi* (localG - spdep) for each spatial unit and testing its significance.

Usage

```
getis.cluster(x, listw, zero.policy = NULL, polygons, significant = TRUE, pleg, ...)
```

Arguments

x	variable to create cluster and significance map
listw	a neighbours list with spatial weights. From package spdep: a listw object. Use poly2nb (class nb) and nb2listw (class listw, nb) from package spdep. Can be any type of listw object, for instance, rook contiguity (common edge) or queen contiguity (common edge or common vertex)
zero.policy	by default = NULL, if FALSE stop with error for any empty neighbour sets, if TRUE permit the weights list to be formed with zero-length weights vectors. Parameter inherited from the spdep package.
polygons	SpatialPolygons, SpatialPolygonsDataFrame or sfc_POLYGON object
significant	by default is TRUE, if FALSE the significant map is not created
pleg	the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by xy.coords
...	other graphical parameters as in par(..)

Details

Using the function localG (spdep) create the Getis Gi* Cluster Map and the corresponding Significance Map. The significance map is done testing the null hypothesis (Ho) of zero spatial autocorrelation for each spatial unit, then plotting a choropleth map with this legend values: (Not Significant, p-value=0.05, p-value= 0.01, p-value=0.001, p-value=0.0001, and Neighborless). Most significant clustered spatial units are those with p-values smaller than 0.0001. Not significant clustered spatial units are those with p-values grather than 0.05. Gi* Cluster Map is done based on the significance map, but the choropleth legend is different (Not - Significant, High-High, Low-Low, Low-High, High-Low, and Neighborless).

Value

one or two maps

Links

1. Spatial Autocorrelation

See Also

- Bivariate Moran's Ixy: moran.bi
- Plot Bivariate Moran's Ixy: moranbi.plot
- Bivariate Moran's Ixy Test: moranbi.test
- Bivariate Local Moran's Ixy and Test: localmoran.bi
- Create object "nb": poly2nb
- Create object "listw"/"nb": nb2listw

Examples

```
library(spdep)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")
getis.cluster(columbus$CRIME, a.lw, zero.policy = FALSE, st_geometry(columbus),
              significant=TRUE, pleg = "topleft")
```

`localmoran.bi` *BiLISA - Bivariate Local Indicators of Spatial Association Moran's Ixy statistic*

Description

The bivariate local spatial statistic Moran's I is calculated for each zone based on the spatial weights object used. The values returned include a Z-value, and may be used as a diagnostic tool. The statistic is:

$$I(i)_{xy} = \frac{(x_i - \bar{x})}{\sum_{k=1}^{n}(x_k - \bar{x})^2/(n-1)} \sum_{j=1}^{n} w_{ij}(y_j - \bar{y})$$

, and its expectation and variance were given in Anselin (1995), but those from Sokal et al. (1998) are implemented here.

Usage

```
localmoran.bi(varX,varY,listw,zero.policy=NULL,na.action=na.fail,conditional=TRUE,
              alternative="two.sided",mlvar=TRUE,spChk=NULL,adjust.x=FALSE)
```

Arguments

`varX`	a numeric vector the same length as the neighbours list in listw and `y`, with the values of the primary variable `x`
`varY`	a numeric vector the same length as the neighbours list in listw and `x`, with the values of the secundary variable `y`
`listw`	a `listw` object created for example by `nb2listw` with spatial weights
`zero.policy`	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA and stop with error for any empty neighbours sets.
`na.action`	a function (default `na.fail`), can also be `na.omit` or `na.exclude` - in these cases the weights list will be subsetted to remove NAs in the data. Similar meaning and values than parameter `na.action` of `localmoran`
`conditional`	default TRUE: expectation and variance are calculated using the conditional randomization null (Sokal 1998 Eqs. A7 & A8). Elaboration of these changes available in Sauer et al. (2021). If FALSE: expectation and variance are calculated using the total randomization null (Sokal 1998 Eqs. A3 & A4).

alternative	type of specifying the alternative hypothesis test, must be one of greater, less or two.sided (default).
mlvar	default TRUE: values of local Moran's I are reported using the variance of the variable of interest (sum of squared deviances over n), but can be reported as the sample variance, dividing by (n-1) instead; both are used in other implementations.
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
adjust.x	default FALSE, if TRUE, x values of observations with no neighbours are omitted in the mean of x

Details

The values of local Moran's I are divided by the variance (or sample variance) of the variable of interest to accord with Table 1, p. 103, and formula (12), p. 99, in Anselin (1995), rather than his formula (7), p. 98. The variance of the local Moran statistic is taken from Sokal et al. (1998) p. 334, equations 4 & 5 or equations 7 & 8 located depending on user specification. By default, the implementation divides by n, not (n-1) in calculating the variance and higher moments. Conditional code contributed by Jeff Sauer and Levi Wolf.

Value

Iyxi	local moran statistic
E.Iyxi	expectation of local moran statistic; for localmoran_permthe permutation sample means
Var.Iyxi	variance of local moran statistic; for localmoran_permthe permutation sample standard deviations
Z.Iyxi	standard deviate of local moran statistic; for localmoran_perm based on permutation sample means and standard deviations
Pr()	p-value of local moran statistic using pnorm(); for localmoran_perm using standard deviatse based on permutation sample means and standard deviations
Pr() Sim	For localmoran_perm, rank() and punif() of observed statistic rank for [0, 1] p-values using alternative=
Pr(folded) Sim	the simulation folded [0, 0.5] range ranked p-value (based on https://github.com/pysal/esda/blob/4a63e0b5df1e754b17b5f1205b8cadcbecc5e061/esda/crand.py#L211-L213)
Skewness	For localmoran_perm, the output of e1071::skewness() for the permutation samples underlying the standard deviates
Kurtosis	For localmoran_perm, the output of e1071::kurtosis() for the permutation samples underlying the standard deviates

In addition, an attribute data frame "quadr" with mean and median quadrant columns, and a column splitting on the demeaned variable and lagged demeaned variable at zero.

Note

Conditional permutations added for comparative purposes; permutations are over the whole data vector omitting the observation itself. For p-value adjustment, use `p.adjust()` or `p.adjustSP()` on the output vector.

References

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; Sokal, R. R, Oden, N. L. and Thomson, B. A. 1998. Local Spatial Autocorrelation in a Biological Model. Geographical Analysis, 30. 331–354; Czaplewski, R.L., Reich, R.M. 1993. Expected value and variance of Moran's bivariate spatial autocorrelation statistic for a permutation test, Research paper RM-309, Fort Collins, CO U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station, p, 13.; Anselin, Luc, Ibnu Syabri, and Oleg Smirnov. 2002. "Visualizing Multivariate Spatial Correlation with Dynamically Linked Windows." In New Tools for Spatial Data Analysis: Proceedings of the Specialist Meeting, edited by Luc Anselin and Sergio Rey. University of California, Santa Barbara: Center for Spatially Integrated Social Science (CSISS)

See Also

`localG`

Examples

```
library(spdep)
# load columbus data
data(columbus)
data(oldcol)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")
localmoran.bi(columbus$CRIME, columbus$HOVAL, a.lw, zero.policy=TRUE,
              alternative="two.sided")
```

`moran.bi` *Compute Bivariate Moran's Ixy*

Description

A simple function to compute bivariate Moran's Ixy, called by `moranbi.test` and `moranbi.mc`;

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - \bar{x})(y_j - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Usage

```
moran.bi(varX, varY, listw, zero.policy=NULL, adjust.n = TRUE, NAOK=FALSE)
```

Arguments

varX	a numeric vector of the same length as the neighbours list in listw with the values of the variable x
varY	a numeric vector of the same length as the neighbours list in listw with the values of the variable y
listw	a listw object created for example by nb2listw
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted
NAOK	if 'TRUE' then any 'NA' or 'NaN' or 'Inf' values in x are passed on to the foreign function. If 'FALSE', the presence of 'NA' or 'NaN' or 'Inf' values is regarded as an error

Value

a list of

I	Bivariate Moran's I
Kx	sample kurtosis of the variable X
Ky	sample kurtosis of the variable Y

References

Czaplewski, R.L., Reich, R.M. 1993. Expected value and variance of Moran's bivariate spatial autocorrelation statistic for a permutation test, Research paper RM-309, Fort Collins, CO U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station, p, 13.

See Also

moranbi.test, moranbi.mc

Examples

```
library(spdep)
data(oldcol)
col.W <- nb2listw(COL.nb, style="W")
crime <- COL.OLD$CRIME
inc <- COL.OLD$INC
str(moran.bi(crime, inc, col.W, zero.policy = TRUE, NAOK=FALSE))
set.seed(123)
is.na(crime) <- sample(1:length(crime), 5)
is.na(inc) <- sample(1:length(inc), 10)
str(moran.bi(crime, inc, col.W, zero.policy = TRUE, NAOK=TRUE))
```

`moran.cluster` *Maps of LISA clusters and statistical significance associated with LISA*

Description

Using the `localmoran` function, create a Local Indicators of Spatial Association (LISA) cluster map and corresponding significance map. The maps are made by calculating the LISAs from `localmoran` for each of the spatial units and then the statistical significance is evaluated to determine the spatial clusters and outliers.

Usage

```
moran.cluster(x, listw, zero.policy = NULL, polygons, conditional=TRUE,
                significant = TRUE, alternative = "two.sided", pleg, ...)
```

Arguments

`x`	variable to create cluster and significance map
`listw`	a neighbours list with spatial weights. From package spdep: a listw object. Use `poly2nb` (class `nb`) and `nb2listw` (class `listw`, `nb`) from package spdep. Can be any type of listw object, for instance, rook contiguity (common edge) or queen contiguity (common edge or common vertex)
`zero.policy`	by default = NULL, if FALSE stop with error for any empty neighbour sets, if TRUE permit the weights list to be formed with zero-length weights vectors. Parameter inherited from the `spdep` package.
`polygons`	SpatialPolygons, SpatialPolygonsDataFrame or sfc_POLYGON object
`conditional`	default TRUE: expectation and variance are calculated using the conditional randomization null (Sokal 1998 Eqs. A7 & A8). Elaboration of these changes available in Sauer et al. (2021). If FALSE: expectation and variance are calculated using the total randomization null (Sokal 1998 Eqs. A3 & A4).
`significant`	by default is TRUE, if FALSE the significant map is not created
`alternative`	by default is "two.sided". Type of alternative hypothesis test. Other values are "less" or "greater".
`pleg`	the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by `xy.coords`
`...`	other graphical parameters as in `par(..)`

Details

Using the function `localmoran` create the Local Indicators of Spatial Association - LISA Cluster Map and the corresponding Significance Map. The significance map is done testing the null hypothesis (Ho) of zero spatial autocorrelation for each spatial unit and then plotting a choropleth map with this legend values: (Not Significant, p-value=0.05, p-value= 0.01, p-value=0.001, p-value=0.0001,

and Neighborless). Maps can represent concentrations of similar (cluster) or dissimilar values (spatial outliers). Most significant clustered spatial units are those with p-values smaller than 0.0001. Not significant clustered spatial units are those with p-values greater than 0.05. LISA Cluster Map is done based on the significance map but the choropleth legend is different (Not - Significant, High-High, Low-Low, Low-High, High-Low, and Neighborless).

Value

two maps

Links

1. Indicators of Spatial Association

See Also

- Bivariate Moran's Ixy: `moran.bi`
- Plot Bivariate Moran's Ixy: `moranbi.plot`
- Bivariate Moran's Ixy Test: `moranbi.test`
- Create object "nb": `poly2nb`
- Create object "listw"/"nb": `nb2listw`

Examples

```
library(spdep)
data(columbus)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")
moran.cluster(columbus$CRIME, a.lw, zero.policy = FALSE, st_geometry(columbus),
              significant=TRUE,pleg = "topleft")
moran.cluster(columbus$CRIME, a.lw, zero.policy = FALSE, st_geometry(columbus),
              significant=TRUE,alternative="greater",pleg = "topleft")
```

`moranbi.cluster` *Maps of BiLISA clusters and statistical significance associated with BiLISA*

Description

Using the `localmoran.bi` function, create a Bivariate Local Indicators of Spatial Association (BiLISA) cluster map and corresponding significance map. The maps are made by calculating the BiLISAs from `localmoran.bi` for each of the spatial units and then the statistical significance is evaluated to determine the spatial clusters and outliers.

Usage

```
moranbi.cluster(varY, varX, listw, zero.policy = NULL, polygons, conditional=TRUE,
                significant = TRUE, alternative = "two.sided", pleg, ...)
```

Arguments

varY	a numeric vector the same length as the neighbours list in listw and x, with the values of the secundary variable y
varX	a numeric vector the same length as the neighbours list in listw and y, with the values of the primary variable x
listw	a neighbours list with spatial weights. From package spdep: a listw object. Use poly2nb (class nb) and nb2listw (class listw, nb) from package spdep. Can be any type of listw object, for instance, rook contiguity (common edge) or queen contiguity (common edge or common vertex)
zero.policy	by default = NULL, if FALSE stop with error for any empty neighbour sets, if TRUE permit the weights list to be formed with zero-length weights vectors. Parameter inherited from the spdep package.
polygons	SpatialPolygons, SpatialPolygonsDataFrame or sfc_POLYGON object
conditional	default TRUE: expectation and variance are calculated using the conditional randomization null (Sokal 1998 Eqs. A7 & A8). Elaboration of these changes available in Sauer et al. (2021). If FALSE: expectation and variance are calculated using the total randomization null (Sokal 1998 Eqs. A3 & A4).
significant	by default is TRUE, if FALSE the significant map is not created
alternative	by default is "two.sided". Type of alternative hypothesis test. Other values are "less" or "greater".
pleg	the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by xy.coords
...	other parameters similar to internal function moran and other graphical parameters as in par(..)

Details

Using the function localmoran.bi create the Bivariate Local Indicators of Spatial Association - BiLISA Cluster Map and the corresponding Significance Map. The significance map is done testing the null hypothesis (Ho) of zero spatial autocorrelation for each spatial unit and then plotting a choropleth map with this legend values: (Not Significant, p-value=0.05, p-value= 0.01, p-value=0.001, p-value=0.0001, and Neighborless). Maps can represent concentrations of similar (cluster) or dissimilar values (spatial outliers). Most significant clustered spatial units are those with p-values smaller than 0.0001. Not significant clustered spatial units are those with p-values greater than 0.05. BiLISA Cluster Map is done based on the significance map but the choropleth legend is different (Not - Significant, High-High, Low-Low, Low-High, High-Low, and Neighborless).

Value

one or two maps

Links

1. Indicators of Spatial Association

See Also

- Bivariate Moran's Ixy: moran.bi
- Plot Bivariate Moran's Ixy: moranbi.plot
- Bivariate Moran's Ixy Test: moranbi.test
- Create object "nb": poly2nb
- Create object "listw"/"nb": nb2listw

Examples

```
library(spdep)
data(columbus)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")
moranbi.cluster(columbus$CRIME, columbus$HOVAL, a.lw, zero.policy = FALSE,
                conditional=TRUE, st_geometry(columbus), significant=TRUE,
                pleg = "topleft")
moranbi.cluster(columbus$CRIME, columbus$HOVAL, a.lw, zero.policy = FALSE,
                st_geometry(columbus), significant=TRUE, alternative="greater",
                pleg = "topleft")
```

---

moranbi.mc *Permutation test for bivariate Moran's Ixy statistic*

Description

A permutation test for bivariate Moran's Ixy statistic calculated by using nsim random permutations of x and y for the given spatial weighting scheme, to establish the rank of the observed statistic in relation to the nsim simulated values.

Usage

```
moranbi.mc(varX, varY, listw, nsim, zero.policy=NULL, alternative="greater",
       na.action=na.fail, spChk=NULL, return_boot=FALSE, adjust.n=TRUE, parallel="no",
           ncpus = getOption("boot.ncpus", 1L), cl = NULL)
```

Arguments

varX	a numeric vector of the variable x the same length as the neighbours list in listw
varY	a numeric vector of the variable y the same length as the neighbours list in listw
listw	a listw object created for example by nb2listw
nsim	number of permutations
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
alternative	a character string specifying the alternative hypothesis, must be one of "greater" (default), "two.sided", or "less".
na.action	a function (default na.fail), can also be na.omit or na.exclude - in these cases the weights list will be subsetted to remove NAs in the data. It may be necessary to set zero.policy to TRUE because this subsetting may create no-neighbour observations. Note that only weights lists created without using the glist argument to nb2listw may be subsetted. na.pass is not permitted because it is meaningless in a permutation test.
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
return_boot	return an object of class boot from the equivalent permutation bootstrap rather than an object of class htest
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted
parallel	The type of parallel operation to be used (if any). If missing, the default is "no". for more details see boot
ncpus	integer: number of processes to be used in parallel operation: number of available CPUs. By default it is set to 1 with getOption("boot.ncpus", 1L)
cl	An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the boot call (by default is NULL).

Value

A list with class htest and mc.sim containing the following components:

statistic	the value of the observed bivariate Moran's I.
parameter	the rank of the observed bivariate Moran's I.
p.value	the pseudo p-value of the test.
alternative	a character string describing the alternative hypothesis.
method	a character string giving the method used.
data.name	a character string giving the name(s) of the data, and the number of simulations.
res	nsim simulated values of statistic, final value is observed statistic

References

Cliff, A. D., Ord, J. K. 1981 Spatial processes, Pion, p. 63-5.

See Also

moran.bi, moranbi.test

Examples

```
library(spdep)
data(oldcol)
colw <- nb2listw(COL.nb, style="W")
nsim <- 999
set.seed(123)
sim1 <- moranbi.mc(COL.OLD$CRIME, COL.OLD$INC, listw=colw, nsim=nsim, zero.policy=TRUE)
sim1
mean(sim1$res[1:nsim])
var(sim1$res[1:nsim])
summary(sim1$res[1:nsim])


colold.lags <- nblag(COL.nb, 3)
set.seed(1234)
sim2 <- moranbi.mc(COL.OLD$CRIME, COL.OLD$INC, nb2listw(colold.lags[[2]],style="W"),
                   nsim=nsim, zero.policy=TRUE)
summary(sim2$res[1:nsim])
sim3 <- moranbi.mc(COL.OLD$CRIME, COL.OLD$INC, nb2listw(colold.lags[[3]],style="W"),
                   nsim=nsim, zero.policy=TRUE)
summary(sim3$res[1:nsim])
sim4 <- moranbi.mc(COL.OLD$CRIME, COL.OLD$INC, colw, nsim=nsim, zero.policy=TRUE,
                   return_boot=TRUE,parallel="multicore")
sim4
```

---

moranbi.plot *Bivariate Moran scatterplot*

Description

A plot of spatial data of the variable "varY" against the spatially lagged values of the variable "varX", augmented by reporting the summary of influence measures for the linear relationship between the data of "varY" and the lag of "varX" ". If policy zero is TRUE, such observations are also flagged if they occur.

Usage

```
moranbi.plot(varY, varX, listw, zero.policy=NULL, spChk=NULL, labels=NULL,
 xlab=NULL, ylab=NULL, quiet=NULL, plot=TRUE, return_df=TRUE, ...)
```

Arguments

varX	a numeric vector of the same length as the neighbours list in listw with the values of the variable x
varY	a numeric vector of the same length as the neighbours list in listw with the values of the variable y
listw	a listw object created for example by nb2listw
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
labels	character labels for points with high influence measures, if set to FALSE, no labels are plotted for points with large influence
xlab	label for x axis
ylab	label for x axis
quiet	default NULL, use !verbose global option value; if TRUE, output of summary of influence object suppressed
plot	default is TRUE, to suppress the plotting use FALSE
return_df	default TRUE, invisibly return a data.frame object; if FALSE invisibly return an influence measures object
...	other graphical parameters as in par(..)

Value

The function returns a data.frame object with coordinates and influence measures if return_df is TRUE, or an influence object from influence.measures.

References

Matkan, A.A., Shahri, M. and Mirzaie, M., 2013, September. Bivariate Moran's I and LISA to explore the crash risky locations in urban areas. In Proceedings of the Conference of Network-Association of European Researchers on Urbanisation in the South, Enschede, The Netherlands (pp. 12-14).

See Also

localmoran.bi, influence.measures

Examples

```
library(spdep)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")

# Editing axis labels
CRIME <- as.vector(scale(columbus$CRIME))
```

```
INCOME <- as.vector(scale(columbus$INC))
moranbi.plot(CRIME,INCOME,quiet =FALSE,zero.policy =FALSE,listw=a.lw)
# Without editing the label of the axes
moranbi.plot(as.vector(scale(columbus$CRIME)),as.vector(scale(columbus$INC)),
             quiet =FALSE,zero.policy =FALSE,listw=a.lw)

# Moran scatterplot
mp <- moranbi.plot(CRIME,INCOME,quiet=FALSE,zero.policy=FALSE,listw=a.lw,
                   label=columbus$POLYID, plot = FALSE)

# Plot Moran Scatterplot in ggplot
if (require(ggplot2, quietly=TRUE)) {
# xname <- attr(mp, "xname")
ggplot2::ggplot(mp, aes(x=varY, y=wx)) + geom_point(shape=1) +
  geom_smooth(formula=y ~ x, method="lm") +
  geom_hline(yintercept=mean(mp$wx), lty=2) +
  geom_vline(xintercept=mean(mp$varY), lty=2) + theme_minimal() +
  geom_point(data=mp[mp$is_inf,], aes(x=varY, y=wx), shape=9) +
  geom_text(data=mp[mp$is_inf,], aes(x=varY, y=wx, label=labels, vjust=1.5)) +
  # xlab(xname) + ylab(paste0("Spatially lagged ", xname))
  xlab("Crime") + ylab("Spatially Lagged Income")
}
```

`moranbi.test` *Moran's Ixy test for bivariate spatial autocorrelation*

Description

Moran's Ixy test for bivariate spatial autocorrelation using a spatial weights matrix in weights list form. The assumptions underlying the test are sensitive to the form of the graph of neighbour relationships and other factors, and results may be checked against those of `moranbi.mc` permutations.

Usage

```
moranbi.test(varX,varY,listw,randomisation=TRUE,zero.policy=NULL,
             alternative="greater",rank=FALSE,spChk=NULL,adjust.n=TRUE,
             drop.EI2=FALSE)
```

Arguments

`varX`	a numeric vector of the same length as the neighbours list in listw with the values of the variable x
`varY`	a numeric vector of the same length as the neighbours list in listw with the values of the variable y
`listw`	a `listw` object created for example by `nb2listw`
`randomisation`	variance of I calculated under the assumption of randomisation, if FALSE normality

zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE assign NA
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.
rank	logical value - default FALSE for continuous variables, if TRUE, uses the adaptation of Moran's I for ranks suggested by Cliff and Ord (1981, p. 46)
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted
drop.EI2	default FALSE, if TRUE, emulate CrimeStat <= 4.02

Value

A list with class htest containing the following components:

statistic	the value of the standard deviate of Moran's Ixy.
p.value	the p-value of the test.
estimate	the value of the observed Moran's Ixy, its expectation and variance under the method assumption.
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.
method	a character string giving the assumption used for calculating the standard deviate.
data.name	a character string giving the name(s) of the data.

Note

Var(Ixy) is taken from Cliff and Ord (1969, p. 28), and Goodchild's CATMOG 47 (1986), see also Upton & Fingleton (1985) p. 171; it agrees with SpaceStat, see Tutorial workbook Chapter 22; VIxy is the second crude moment minus the square of the first crude moment. The derivation of the test (Cliff and Ord, 1981, p. 18) assumes that the weights matrix is symmetric. For inherently non-symmetric matrices, such as k-nearest neighbour matrices, listw2U() can be used to make the matrix symmetric.

References

Cliff, A. D., Ord, J. K. 1981 Spatial processes, Pion, p. 21; Bivand RS, Wong DWS 2018 Comparing implementations of global and local indicators of spatial association. TEST, 27(3), 716–748 doi:10.1007/s117490180599x

See Also

moran, moran.mc, listw2U

Examples

```
library(spdep)
data(columbus)
data(oldcol)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))
col_nbq <- poly2nb(columbus)
set.seed(123)
BMCrime <- moranbi.test(columbus$CRIME,columbus$INC,nb2listw(COL.nb, style="W"),
           zero.policy =TRUE, alternative = "two.sided")
BMCrime
moranbi.test(columbus$CRIME,columbus$INC,nb2listw(COL.nb,style="B"),
             zero.policy =TRUE,alternative = "two.sided",randomisation=FALSE)
colold.lags <- nblag(col_nbq, 3)
moranbi.test(columbus$CRIME,columbus$INC,nb2listw(colold.lags[[2]],style="W"),
             zero.policy =TRUE, alternative = "two.sided",randomisation=FALSE)
print(is.symmetric.nb(COL.nb))

COL.k4.nb <- knn2nb(knearneigh(coords, 4))
print(is.symmetric.nb(COL.k4.nb))
cat("Note: non-symmetric weights matrix, use listw2U()")
moranbi.test(columbus$CRIME,columbus$INC,listw2U(nb2listw(COL.k4.nb,style="W")),
             zero.policy =TRUE,adjust.n = TRUE)
moranbi.test(columbus$CRIME,columbus$INC,listw2U(nb2listw(COL.k4.nb,style="W")),
             zero.policy =TRUE, randomisation=FALSE)

ranksX <- rank(columbus$CRIME)
ranksY <- rank(columbus$INC)
names(ranksX) <- rownames(columbus)
names(ranksY) <- rownames(columbus)
moranbi.test(ranksX,ranksY,nb2listw(COL.k4.nb,style="W"),rank=TRUE,
             zero.policy=TRUE,alternative="two.sided")

crime <- columbus$CRIME
income <- columbus$INC
set.seed(123)
is.na(crime) <- sample(1:length(crime), 5)
is.na(income) <- sample(1:length(income), 4)
DF <- data.frame(crime,income)
col.na <- moranbi.test(DF$crime, DF$income, nb2listw(COL.nb, style="W"),
           zero.policy =TRUE)
col.na
```

`moranbir.test` *Moran's Ixy test using Monte Carlo permutations for bivariate spatial autocorrelation*

Description

Moran's Ixy test for bivariate spatial autocorrelation using a spatial weights matrix in weights list form. Constructs a plot of the empirical density from the simulations (Monte Carlo permutations)

and superimposes Moran's bivariate Ixy index.

Usage

```
moranbir.test(varX,varY,listw,zero.policy=NULL,adjust.n=TRUE,N,graph=FALSE,
              alternative="greater", spChk=NULL, print.results=TRUE, ...)
```

Arguments

varX	a numeric vector of the same length as the neighbours list in listw with the values of the variable x
varY	a numeric vector of the same length as the neighbours list in listw with the values of the variable y
listw	a listw object created for example by nb2listw
zero.policy	by default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE stop with error for any empty neighbours sets and assign NA.
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbour observations, if TRUE, the number of observations is adjusted
N	set the number of spatial permutations for the calculation.
graph	by default = FALSE. Use TRUE to create test's graphic.
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
print.results	by default = TRUE. Use FALSE to hide test results (table). Results are: observed, expected and p-value.
...	other parameters similar to original

Details

Compare the observed bivariate Moran's Ixy (moran.bi function) with the expected value empirical density. The expected value is -cor(x,y)/(n-1), where n is the number of rows/samples (number of polygons), and represents the null hyphothesis (Ho) of no spatial Autocorrelation (bivariate Moran's Ixy coefficient around zero). This expected value density is constructed with Monte Carlo simulations. Values significant below of -cor(x,y)/(n-1) represents negative spatial autocorrelation (generally negative values of observed bivariate Moran's Ixy), and values significant above of -cor(x,y)/(n-1) represents positive spatial autocorrelation (generally positive values of observed bivariate Moran's Ixy). For hypothesis testing the sample values are compared with empirical density, and p-value is calculated. For significant values of p-value (reject Ho), the conclusion of the test could be: "given the value of p-value, there is less than alpha (1%, or 5%, or 10%) likelihood that the pattern (clustered or dispersed) could be the result of random change".

Value

A list with class `htest` containing the following components:

`Observed`	the value of the observed Moran's Ixy.
`Expected`	the expected value of Moran's Ixy.
`p.value`	the p-value of the test.
`Values`	corresponds to the N simulated values of Moran's Ixy coefficient for bivariate autocorrelation.

Links

Adapted from:

1. Spatial Autocorrelation (Moran's I) Test
2. Moran's I Test

References

Cliff, A. D., Ord, J. K. 1981 Spatial processes, Pion, p. 21; Bivand RS, Wong DWS 2018 Comparing implementations of global and local indicators of spatial association. TEST, 27(3), 716–748 doi:10.1007/s117490180599x

See Also

`moran.bi`, `moranbi.mc`, `listw2U`

Examples

```
library(spdep)
data(columbus)
data(oldcol)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))
col_nbq <- poly2nb(columbus)
a.lw <- nb2listw(col_nbq, style="W")
set.seed(123)
RCrimeInc.Q <- moranbir.test(columbus$CRIME,columbus$INC,a.lw,graph=TRUE,
                             zero.policy =TRUE,N=1000)
RCrimeInc.Q$Values


COL.k4.nb <- knn2nb(knearneigh(coords, 4))
RCrimeInc.Knn <- moranbir.test(columbus$CRIME, columbus$INC, listw2U(nb2listw(COL.k4.nb,
                               style="W")),graph=TRUE,zero.policy =TRUE,N=1000)
RCrimeInc.Knn
```

randomize_vector *Generate a random sample from other vector*

Description

Use function 'sample' from base R library to generate a random sample. Function taken from Edzer Pebesma package.

Usage

```
randomize_vector(X, N)
```

Arguments

X	vector to choose from
N	number of random elements to select from X

Value

a list, a vector

Examples

```
library(spdep)
example(columbus)
#col_nbq <- poly2nb(columbus)
#a.lw <- nb2listw(col_nbq, style="W")
#set.seed(123)
DF <- data.frame(1:length(columbus$CRIME),columbus$CRIME,columbus$INC)
X1<-randomize_vector(DF$Obs,999)
```

spcorrelogram.bi *Bivariate spatial correlogram*

Description

Bivariate spatial correlograms for Moran's Ixy and the autocorrelation coefficient, with print and plot helper functions.

Usage

```
spcorrelogram.bi(neighbours, varX, varY, order = 1, method = "corr",
                 style = "W", randomisation = TRUE, zero.policy = NULL,
                 spChk=NULL, alternative = "greater", drop.EI2=FALSE)
## S3 method for class 'spcorbi'
plot(x, main, ylab, ylim, ...)
## S3 method for class 'spcorbi'
print(x, p.adj.method="none", ...)
```

Arguments

neighbours	an object of class nb
varX	a numeric vector of the variable x
varY	a numeric vector of the variable y
order	maximum lag order
method	"corr" for correlation, "I" for Moran's Ixy, "C" for Geary's Cxy
style	style can take values W, B, C, and S
randomisation	variance of I or C calculated under the assumption of randomisation, if FALSE normality
zero.policy	default NULL, use global option value; if FALSE stop with error for any empty neighbour sets, if TRUE permit the weights list to be formed with zero-length weights vectors
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.
drop.EI2	default FALSE, if TRUE, emulate CrimeStat <= 4.02
x	an object from spcorrelogram.bi() of class spcorbi
p.adj.method	correction method as in p.adjust
main	an overall title for the plot
ylab	a title for the y axis
ylim	the y limits of the plot
...	further arguments passed through

Details

The print function also calculates the standard deviates of Bivariate Moran's Ixy or Geary's Cxy and a two-sided probability value, optionally using p.adjust to correct by the nymber of lags. The plot function plots a bar from the estimated value of Bivariate Moran's Ixy, or Geary's Cxy to +/- twice the square root of its variance (in previous releases only once, not twice). The table includes the count of included observations in brackets after the lag order. Care must be taken when interpreting the results, since increasing the order of the lag tends to include fewer observations.

Value

returns a list of class spcorbi:

res	for "corr" a vector of values; for "I", a matrix of estimates of "I", expectations, and variances
method	"I" or "corr"
cardnos	list of tables of neighbour cardinalities for the lag orders used
var	variable name

References

Czaplewski, R.L., Reich, R.M. 1993. Expected value and variance of Moran's bivariate spatial autocorrelation statistic for a permutation test, Research paper RM-309, Fort Collins, CO U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station, p, 13.

See Also

nblag, moran.bi, p.adjust

Examples

```
library(spdep)
data(columbus)
data(oldcol)
columbus <- st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
plot(st_geometry(columbus))
col_nbq <- poly2nb(columbus)

Cspcb <- spcorrelogram.bi(col_nbq,columbus$CRIME,columbus$INC,order=7,
                          method="corr",zero.policy=TRUE,alternative="two.sided")
print(Cspcb)
plot(Cspcb)

Ispcb <- spcorrelogram.bi(col_nbq,columbus$CRIME,columbus$INC,order=7,
                          method="I",zero.policy=TRUE,alternative="two.sided")
print(Ispcb)
plot(Ispcb)

Cspcb <- spcorrelogram.bi(col_nbq,columbus$CRIME,columbus$INC,order=7,
                          method="C",zero.policy=TRUE,alternative="two.sided")
print(Ispcb)
plot(Ispcb)
```

Index