Package ‘spnaf’

August 25, 2022

Type Package

Title Spatial Network Autocorrelation for Flow Data

Version 0.2.1

Description Identify statistically significant flow clusters using the local spatial network autocorrelation statistic $G_{ij}$ proposed by 'Berglund' and 'Karlström' (1999) <doi:10.1007/s101090050013>. The metric, an extended statistic of 'Getis/Ord' G ('Getis' and 'Ord' 1992) <doi:10.1111/j.1538-4632.1992.tb00261.x>, detects a group of flows having similar traits in terms of directionality. You provide OD data and the associated polygon to get results with several parameters, some of which are defined by spdep package.

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports dplyr, magrittr, sf, spdep, tidyr, rlang

Suggests knitr, rmarkdown, tmap

VignetteBuilder knitr

NeedsCompilation no

Author Youngbin Lee [aut, cre] (<https://orcid.org/0000-0003-3466-6495>), Hui Jeong Ha [aut] (<https://orcid.org/0000-0002-5229-4528>), Sohyun Park [aut] (<https://orcid.org/0000-0002-1231-5662>), Kyusik Kim [aut] (<https://orcid.org/0000-0003-3753-3196>), Jinhyung Lee [aut] (<https://orcid.org/0000-0003-1859-3441>)

Maintainer Youngbin Lee <youngbin@snu.ac.kr>

Repository CRAN

Date/Publication 2022-08-25 08:20:02 UTC
R topics documented:

- CA ................................................................. 2
- CA_polygon .................................................. 3
- Gij_polygon ................................................... 3

Description

A dataframe containing migration of CA counties with origins and destinations from US Census

Usage

CA

Format

A dataframe object with 2580 rows and 12 variables

- State.Code.of.Geography.A  Destinations’ State code
- FIPS.County.Code.of.Geography.A  Destinations’ FIPS County code
- FIPS.County.Code.of.Geography.B  Origins’ FIPS County code
- State.Name.of.Geography.A  Destinations’ State name
- County.Name.of.Geography.A  Destinations’ County name
- County.Name.of.Geography.B  Origins’ County name
- Flow.from.Geography.B.to.Geography.A  Flow count from the origin to the destination
- Counterflow.from.Geography.A.to.Geography.B  Counterflow count from the destination to the origin
- Net.Migration.from.Geography.B.to.Geography.A  Net migration count from the origin to the destination

Source

**Examples**

CA

---

**CA_polygon**

Sample polygon data of California counties.

**Description**

A sf(simple feature) containing geometric boundaries of CA counties with their codes.

**Usage**

CA_polygon

**Format**

A sf object with 58 rows and 2 variables

- **id** FIPS County code of geography
- **geometry** the geometry column for counties (CRS: NAD83)

---

**Gij.polygon**

Calculate spatial autocorrelation with OD data and corresponding polygons.

**Description**

Calculate spatial autocorrelation with OD data and corresponding polygons.

**Usage**

Gij.polygon(df, shape, queen = TRUE, snap = 1, method = "t", R = 1000)

**Arguments**

- **df** A data.frame that contains your Origin-Destination data. The df must consist of "oid" (origin id), "did" (destination id), "n" (flow weight).
- **shape** A shapefile (in a polygon type) that matches to your OD dataframe. The shape must have an "id" column to match your ids in df.
- **queen** A TRUE/FALSE input that is used to calculate spdep’s spatial contingency (Please view documents of poly2nb for more information).
- **snap** A parameter that is also used to calculate spdep’s spatial contingency (Please view documents of poly2nb for more information).
- **method** A string value among "o" (origin based), "d" (destination based), and "t" (both way) which determines the way to generate Spatial Weights. The default value is "t".
- **R** An integer value to define how many times you want to execute bootstrapping.
**Value**

The result is in the form of a list which includes a dataframe and a sf object. Both contain Gij statistics and p-value columns merged to your input df. The geometry type of the latter is linestring.

**References**


**Examples**

```r
# Data manipulation
CA <- spnaf::CA
OD <- cbind(OD, CA$Flow.from.Geography.B.to.Geography.A)
OD <- data.frame(OD)
names(OD) <- c("oid", "did", "n")
OD$n <- as.numeric(OD$n)
OD <- OD[order(OD[,1], OD[,2]),]
head(OD) # check the input df’s format

# Load sf polygon
CA_polygon <- spnaf::CA_polygon
head(CA_polygon) # it has a geometry column

# Execution of Gij.polygon with data above and given parameters
result <- Gij.polygon(df = OD, shape = CA_polygon, queen = TRUE, snap = 1,
method = "t", R = 1000)

# check the results
head(result[[1]])
head(result[[2]])
```
Index

* datasets
  CA, 2
  CA_polygon, 3
CA, 2
CA_polygon, 3
Gij_polygon, 3
poly2nb, 3