

# Package ‘spnaf’

August 7, 2022

**Type** Package

**Title** Spatial Network Autocorrelation for Flow Data

**Version** 0.2.0

**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](https://doi.org/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](https://doi.org/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, rmarkdown, tmap

**Suggests** knitr

**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>>),  
Kyungho 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](mailto:youngbin@snu.ac.kr)>

**Repository** CRAN

**Date/Publication** 2022-08-07 17:20:02 UTC

## R topics documented:

CA . . . . .	2
CA_polygon . . . . .	3
Gij.polygon . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

CA	<i>Sample migration data by counties in California.</i>
----	---

---

### Description

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

### Usage

CA

### Format

A data.frame 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

**State.U.S..Island.Area.Foreign.Region.Code.of.Geography.B** Destinations' State U.S. Island Area Foreign Region 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

**State.U.S..Island.Area.Foreign.Region.of.Geography.B** Origins' State U.S. Island Area Foreign Region Code

**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

**Gross.Migration.between.Geography.A.and.Geography.B** Gross migration count between counties

### Source

Census.gov > Population > Migration/Geographic Mobility > Guidance for Data Users > County-to-County Migration Flows <https://www.census.gov/topics/population/migration/guidance/county-to-county-migration-flows.html>

**Examples**

CA

---

CA_polygon	<i>Sample polygon data of California counties.</i>
------------	--

---

**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	<i>Calculate spatial autocorrelation with OD data and corresponding polygons.</i>
-------------	---

---

**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 <a href="#">poly2nb</a> for more information).
snap	A parameter that is also used to calculate spdep's spatial contingency (Please view documents of <a href="#">poly2nb</a> 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**

Berglund, S., & Karlström, A. (1999). Identifying local spatial association in flow data, *Journal of Geographical Systems*, 1(3), 219-236. <https://doi.org/10.1007/s101090050013>

**Examples**

```
# Data manipulation
CA <- sfnaf::CA
OD <- cbind(CA$FIPS.County.Code.of.Geography.B, CA$FIPS.County.Code.of.Geography.A)
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 <- sfnaf::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](#)