**Homework-09 田晓妍**

**The questions:**

1. Local estimates of Moran’s I and Geary’s C are often used to identify spatial clustering/hotspots) and outliers in attribute values. Please point out which conditions indicate hotspots and which ones indicate outliers.
2. When working with lattice data, there are typically two forms of filters used for the weight matrix. Please points the following filters, and give their spatial weights matrices, respectively. Also, please write a line code for create the weighting matrix according to any one of the weight filters.

**Answers:**

1. Local estimates of Moran’s I and Geary’s C are spatial autocorrelation statistics that can be used to detect spatial clustering or hotspots, as well as outliers, in the distribution of attribute values across a region.

1.1\*\*Hotspots:\*\*

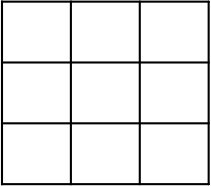
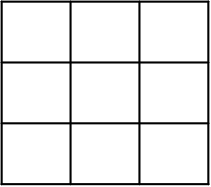
- Moran’s I: A positive Moran’s I value at a particular location indicates clustering or positive spatial autocorrelation, i.e., similar values tend to be clustered together. This clustering can be interpreted as a hotspot if the surrounding areas have significantly similar values. A statistically significant positive Moran’s I suggests that the observed clustering is unlikely to be random.

- Geary’s C: A Geary’s C value less than 1 at a particular location suggests positive spatial autocorrelation, indicating clustering. As with Moran’s I, a statistically significant Geary’s C less than 1 can be interpreted as a hotspot.

1.2\*\*Outliers:\*\*

- Moran’s I and Geary’s C can also indicate outliers when they show negative values or Geary’s C is greater than 1. These values indicate negative spatial autocorrelation, meaning dissimilarity in values between a location and its neighbors. If a location has a statistically significant negative Moran’s I or Geary’s C greater than 1, it is likely to be an outlier in terms of its attribute value compared to its neighbors.

2.



2.1 The left one is Rook Contiguity Filter:

- This filter considers two grid cells as neighbors if they share a common edge (but not a corner).

- Spatial weights matrix as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- the rook contiguity matrix can be created with the “rook\_from\_lattice” function using “spdeg” package :

library(spdep)

rook\_matrix <- rook\_from\_lattice(lattice\_data)

- or choosing the “poly2nb” function which returns an adjacency list, containing the adjacency relationships between polygons to compute a spatial weight matrix:

library(spdep)

nb\_rock <- poly2nb(spdf, row.names = NULL, snap=sqrt(.Machine$double.eps),

queen=TRUE,queen=FALSE)

listw <- nb2listw(nb\_rock)

print(listw)

#

# spdf: A SpatialPolygons or SpatialPolygonsDataFrame object representing the polygons.

# row.names: Optionally provide row names for the neighbour list. If not provided, the row names from 'pl' will be used.

# snap: A numerical value defining the snapping distance. When the distance between two polygon vertices is less than this value,

# they are considered to be coincident or adjacent.

# By default, it is set to the square root of the machine epsilon for double precision floating point numbers

# to account for minor differences in coordinates due to computational precision or rounding errors.

# queen: A logical value indicating whether to use queen contiguity (TRUE) or rook contiguity (FALSE).

# Queen contiguity considers both shared boundaries and vertices as adjacency,

# while rook contiguity only considers shared boundaries.

# useC: A logical value indicating whether to use C language-based underlying code for faster computation.

# foundInBox: An optional function that can be provided to accelerate the neighbour search process for large datasets.

2.2 The right one is Queen Contiguity Filter:\*\*

- This filter considers two grid cells as neighbors if they share a common edge or corner.

- Spatial weights matrix is shown:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- a line code for create the weighting matrix:

library(spdep)

queen\_matrix <- queen\_from\_lattice(lattice\_data)

- or the another method for the same purpose:

library(spdep)

nb\_queen <- poly2nb(spdf, queen=TRUE)

listw <- nb2listw(nb\_queen) print(listw)