**Homework09**

**姓名：廖一旭**

**学号：SA23008212**

**日期：2024/5/4**

**Question1：**

Moran's I 和 Geary's C 是两种常用于空间数据分析的统计方法，用于检测空间属性值的集聚（热点）和异常值（离群点）。以下是对这两种指标的解释及其在识别热点和离群点中的应用：

**Moran's I：**Moran's I 是一种衡量空间自相关的指标，用来判断一个区域内的空间元素是否与其周围空间元素相似。其值范围从 -1（完全负自相关）到 +1（完全正自相关），0 表示随机分布。

hotspots热点识别：当 Moran's I 显著大于0时，表明空间属性值在该区域呈正相关，即相似或相关的值倾向于聚集在一起，形成热点。

outliers离群点识别：Moran's I 不常直接用于识别离群点，更多的是通过这个指标的映射（例如局部Moran's I图）来间接观察离群点。

**Geary's C：**Geary's C 也是衡量空间自相关的指标，但与 Moran's I 相比，它更强调空间单元之间的差异。Geary's C 的值范围从0（完全相关）到大于1（负相关），1表示随机分布。

hotspots热点识别：当 Geary's C 显著小于1时，表明相邻的观测值相似度高，指示这些区域可能是属性值的热点。

outliers离群点识别：Geary's C 值大于1表明相邻的观测值差异大，这可能表明这些区域存在离群点。

**应用场景：**局部Moran's I和局部Geary's C：这两种指标的局部版本特别适合用于识别具体的热点和离群点。局部Moran's I可以帮助识别空间聚类（高-高或低-低值聚集）和空间离群点（高-低或低-高值相邻）。而局部Geary's C则更加敏感于相邻数值之间的差异，适合用来发现离群点。

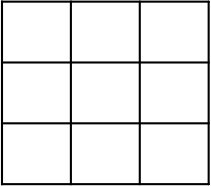
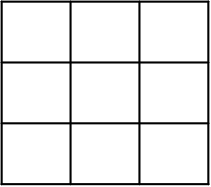
Spatial hotspots：使用局部Moran’s I

spatial outliers：使用局部Moran‘s I，局部Geary‘s C

**Question2：**

Moran's I 和 Geary's C 是两种可用于计算areal 数据空间相关性的技术。它们主要计算空间中邻近位置之间的相关性。这两种技术都可用于计算空间自相关性的全局和局部估计值。局部估计值通常用于识别空间聚类（即热点）和属性值中的异常值。一般来说，I 和 C 的正值表示该地物被具有相似值的地物所包围（即空间聚类），而负值表示该地物被具有不同值的地物所包围（即空间离群值）。

在处理多边形数据时，I 和 C 统计量是根据多边形中心点计算的，并采用邻近加权方案（空间权重矩阵），以考虑多边形之间的几何不规则性。在处理网格数据时，使用图像窗口来开发空间权重矩阵。用于权重矩阵的窗口通常有两种形式：Queen 过滤器和 Rook 过滤器。



空间权重矩阵：

Rook边相关的SWM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

Queen点相关的SWM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 4 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 5 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 7 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 8 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

创建权重矩阵代码如下：

Rook filter（图一）:

# 创建Rook邻接结构

rook\_neighbors <- dnearneigh(coords, d1 = 0, d2 = 1, row.names = NULL)

# 将邻接结构转换为权重矩阵

rook\_weights <- nb2listw(rook\_neighbors, style = "B")

# 提取权重矩阵的邻接矩阵形式

rook\_adj\_matrix <- listw2mat(rook\_weights)

# 打印邻接矩阵

print("Rook邻接规则的权重矩阵：")

print(rook\_adj\_matrix) Queen filter（图二）:

# 安装并加载spdep包

install.packages("spdep")

library(spdep)

# 创建一个3x3的网格坐标

coords <- expand.grid(x = 1:3, y = 1:3)

# 创建Queen邻接结构

queen\_neighbors <- dnearneigh(coords, d1 = 0, d2 = sqrt(2))

# 将邻接结构转换为权重矩阵

queen\_weights <- nb2listw(queen\_neighbors, style = "B")

# 提取权重矩阵的邻接矩阵形式

queen\_adj\_matrix <- listw2mat(queen\_weights)

# 打印邻接矩阵

print("Queen邻接规则的权重矩阵：")

print(queen\_adj\_matrix)