*Spatial Statistics Lab 6*

**Onyedikachi J Okeke**

2024-02-23

### 0.0 To Load the library

library(sp)  
library(spatstat)  
library(sf)  
library(spatstat.geom)  
library(ctv)  
library(terra)  
library(spdep)  
library(rgdal)  
library(terra)  
library(RColorBrewer)  
library(classInt)

##### Library

## Loading required package: spatstat.data

## Loading required package: spatstat.geom

## spatstat.geom 3.2-8

## Loading required package: spatstat.random

## spatstat.random 3.2-2

## Loading required package: spatstat.explore

## Loading required package: nlme

## spatstat.explore 3.2-5

## Loading required package: spatstat.model

## Loading required package: rpart

## spatstat.model 3.2-8

## Loading required package: spatstat.linnet

## spatstat.linnet 3.1-3

##   
## spatstat 3.0-7   
## For an introduction to spatstat, type 'beginner'

## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf\_use\_s2() is TRUE

## terra 1.7.65

##   
## Attaching package: 'terra'

## The following objects are masked from 'package:spatstat.geom':  
##   
## area, delaunay, is.empty, rescale, rotate, shift, where.max,  
## where.min

## Loading required package: spData

## Please note that rgdal will be retired during October 2023,  
## plan transition to sf/stars/terra functions using GDAL and PROJ  
## at your earliest convenience.  
## See https://r-spatial.org/r/2023/05/15/evolution4.html and https://github.com/r-spatial/evolution  
## rgdal: version: 1.6-7, (SVN revision 1203)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 3.6.2, released 2023/01/02  
## Path to GDAL shared files: C:/Users/GIS/AppData/Local/R/win-library/4.3/rgdal/gdal  
## GDAL does not use iconv for recoding strings.  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ runtime: Rel. 9.2.0, March 1st, 2023, [PJ\_VERSION: 920]  
## Path to PROJ shared files: C:\Program Files\PostgreSQL\14\share\contrib\postgis-3.2\proj  
## PROJ CDN enabled: FALSE  
## Linking to sp version:2.1-0  
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,  
## use options("rgdal\_show\_exportToProj4\_warnings"="none") before loading sp or rgdal.

##   
## Attaching package: 'rgdal'

## The following object is masked from 'package:terra':  
##   
## project

### Loading Sids data

# Define the file path  
file\_path <- "C:/Spatial Statistics Labwork/Lab5Data/sids2.shp"  
  
# Import the shapefile  
sids <- st\_read(file\_path)  
  
class(sids)  
  
# Check the structure of the imported object  
str(sids)

##### Result

## Reading layer `sids2' from data source   
## `C:\Spatial Statistics Labwork\Lab5Data\sids2.shp' using driver `ESRI Shapefile'  
## Simple feature collection with 100 features and 18 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965  
## CRS: NA

## [1] "sf" "data.frame"

## Classes 'sf' and 'data.frame': 100 obs. of 19 variables:  
## $ AREA : num 0.114 0.061 0.143 0.07 0.153 0.097 0.062 0.091 0.118 0.124 ...  
## $ PERIMETER: num 1.44 1.23 1.63 2.97 2.21 ...  
## $ CNTY\_ : num 1825 1827 1828 1831 1832 ...  
## $ CNTY\_ID : num 1825 1827 1828 1831 1832 ...  
## $ NAME : chr "Ashe" "Alleghany" "Surry" "Currituck" ...  
## $ FIPS : chr "37009" "37005" "37171" "37053" ...  
## $ FIPSNO : num 37009 37005 37171 37053 37131 ...  
## $ CRESS\_ID : int 5 3 86 27 66 46 15 37 93 85 ...  
## $ BIR74 : num 1091 487 3188 508 1421 ...  
## $ SID74 : num 1 0 5 1 9 7 0 0 4 1 ...  
## $ NWBIR74 : num 10 10 208 123 1066 ...  
## $ BIR79 : num 1364 542 3616 830 1606 ...  
## $ SID79 : num 0 3 6 2 3 5 2 2 2 5 ...  
## $ NWBIR79 : num 19 12 260 145 1197 ...  
## $ SIDR74 : num 0.917 0 1.568 1.969 6.334 ...  
## $ SIDR79 : num 0 5.54 1.66 2.41 1.87 ...  
## $ NWR74 : num 9.17 20.53 65.24 242.13 750.18 ...  
## $ NWR79 : num 13.9 22.1 71.9 174.7 745.3 ...  
## $ geometry :sfc\_MULTIPOLYGON of length 100; first list element: List of 1  
## ..$ :List of 1  
## .. ..$ : num [1:27, 1:2] -81.5 -81.5 -81.6 -81.6 -81.7 ...  
## ..- attr(\*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"  
## - attr(\*, "sf\_column")= chr "geometry"  
## - attr(\*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA NA NA NA NA ...  
## ..- attr(\*, "names")= chr [1:18] "AREA" "PERIMETER" "CNTY\_" "CNTY\_ID" ...

### Check the projection of the shapefile

# Extract CRS information using st\_crs()  
crs\_info <- st\_crs(sids)  
  
# Print CRS information  
print(crs\_info)

##### Result

## Coordinate Reference System: NA

### Define the target CRS (NAD27 UTM Zone 17N

# Assign the coordinate system (WGS84)  
st\_crs(sids) <- st\_crs("+proj=longlat +ellps=WGS84")  
  
# Check the current CRS  
print(st\_crs(sids))  
  
# Define the target CRS (NAD27 UTM Zone 17N)  
target\_crs <- "+proj=utm +zone=17 +datum=NAD27"  
  
# Project the shapefile to the target CRS  
sids\_projected <- st\_transform(sids, target\_crs)  
  
# Check the CRS of the projected shapefile  
print(st\_crs(sids\_projected))

##### Result

## Coordinate Reference System:  
## User input: +proj=longlat +ellps=WGS84   
## wkt:  
## GEOGCRS["unknown",  
## DATUM["Unknown based on WGS 84 ellipsoid",  
## ELLIPSOID["WGS 84",6378137,298.257223563,  
## LENGTHUNIT["metre",1],  
## ID["EPSG",7030]]],  
## PRIMEM["Greenwich",0,  
## ANGLEUNIT["degree",0.0174532925199433],  
## ID["EPSG",8901]],  
## CS[ellipsoidal,2],  
## AXIS["longitude",east,  
## ORDER[1],  
## ANGLEUNIT["degree",0.0174532925199433,  
## ID["EPSG",9122]]],  
## AXIS["latitude",north,  
## ORDER[2],  
## ANGLEUNIT["degree",0.0174532925199433,  
## ID["EPSG",9122]]]]

## Coordinate Reference System:  
## User input: +proj=utm +zone=17 +datum=NAD27   
## wkt:  
## PROJCRS["unknown",  
## BASEGEOGCRS["unknown",  
## DATUM["North American Datum 1927",  
## ELLIPSOID["Clarke 1866",6378206.4,294.978698213898,  
## LENGTHUNIT["metre",1]],  
## ID["EPSG",6267]],  
## PRIMEM["Greenwich",0,  
## ANGLEUNIT["degree",0.0174532925199433],  
## ID["EPSG",8901]]],  
## CONVERSION["UTM zone 17N",  
## METHOD["Transverse Mercator",  
## ID["EPSG",9807]],  
## PARAMETER["Latitude of natural origin",0,  
## ANGLEUNIT["degree",0.0174532925199433],  
## ID["EPSG",8801]],  
## PARAMETER["Longitude of natural origin",-81,  
## ANGLEUNIT["degree",0.0174532925199433],  
## ID["EPSG",8802]],  
## PARAMETER["Scale factor at natural origin",0.9996,  
## SCALEUNIT["unity",1],  
## ID["EPSG",8805]],  
## PARAMETER["False easting",500000,  
## LENGTHUNIT["metre",1],  
## ID["EPSG",8806]],  
## PARAMETER["False northing",0,  
## LENGTHUNIT["metre",1],  
## ID["EPSG",8807]],  
## ID["EPSG",16017]],  
## CS[Cartesian,2],  
## AXIS["(E)",east,  
## ORDER[1],  
## LENGTHUNIT["metre",1,  
## ID["EPSG",9001]]],  
## AXIS["(N)",north,  
## ORDER[2],  
## LENGTHUNIT["metre",1,  
## ID["EPSG",9001]]]]

### #difine contiguity based neighbors

# Create Queen contiguity spatial weights matrix  
sids\_nbq <- poly2nb(sids, queen = TRUE)  
# Check the summary of the spatial weights matrix  
summary(sids\_nbq)  
  
# Create Queen (false) contiguity spatial weights matrix  
sids\_nbr <- poly2nb(sids, queen = FALSE)  
# Check the summary of the spatial weights matrix  
summary(sids\_nbr)  
  
coords <- st\_coordinates(sids)  
  
# Check the number of neighborhoods in sids\_nbq  
num\_neighborhoods <- length(sids\_nbq)  
  
# Check the number of coordinate pairs in coords  
num\_coords <- nrow(coords)  
  
# Print out the counts for verification  
print(paste("Number of neighborhoods:", num\_neighborhoods))  
print(paste("Number of coordinate pairs:", num\_coords))

##### Result

## Neighbour list object:  
## Number of regions: 100   
## Number of nonzero links: 490   
## Percentage nonzero weights: 4.9   
## Average number of links: 4.9   
## Link number distribution:  
##   
## 2 3 4 5 6 7 8 9   
## 8 15 17 23 19 14 2 2   
## 8 least connected regions:  
## 4 21 45 56 77 80 90 99 with 2 links  
## 2 most connected regions:  
## 39 67 with 9 links

## Neighbour list object:  
## Number of regions: 100   
## Number of nonzero links: 462   
## Percentage nonzero weights: 4.62   
## Average number of links: 4.62   
## Link number distribution:  
##   
## 2 3 4 5 6 7 8 9   
## 8 18 20 25 21 4 3 1   
## 8 least connected regions:  
## 4 21 45 56 77 80 90 99 with 2 links  
## 1 most connected region:  
## 39 with 9 links

## [1] "Number of neighborhoods: 100"

## [1] "Number of coordinate pairs: 2529"

### Row-standardized weights matrix

#Row-standardized weights matrix  
sids\_nbq\_w<- nb2listw(sids\_nbq)  
print(sids\_nbq\_w)  
sids\_nbq\_w$neighbours[1:5]

##### Result

## Characteristics of weights list object:  
## Neighbour list object:  
## Number of regions: 100   
## Number of nonzero links: 490   
## Percentage nonzero weights: 4.9   
## Average number of links: 4.9   
##   
## Weights style: W   
## Weights constants summary:  
## n nn S0 S1 S2  
## W 100 10000 100 44.65023 410.4746

## [[1]]  
## [1] 2 18 19  
##   
## [[2]]  
## [1] 1 3 18  
##   
## [[3]]  
## [1] 2 10 18 23 25  
##   
## [[4]]  
## [1] 7 56  
##   
## [[5]]  
## [1] 6 9 16 28

### Binary Weights

#Binary Weights  
sids\_nbq\_wb<-nb2listw(sids\_nbq, style="B")  
#examine the dataset  
summary (sids)

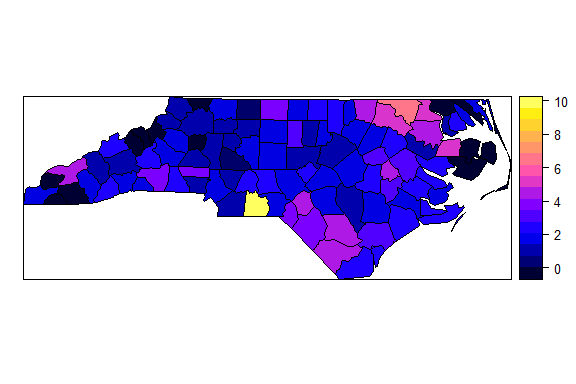
##### Result

## AREA PERIMETER CNTY\_ CNTY\_ID   
## Min. :0.0420 Min. :0.999 Min. :1825 Min. :1825   
## 1st Qu.:0.0910 1st Qu.:1.324 1st Qu.:1902 1st Qu.:1902   
## Median :0.1205 Median :1.609 Median :1982 Median :1982   
## Mean :0.1263 Mean :1.673 Mean :1986 Mean :1986   
## 3rd Qu.:0.1542 3rd Qu.:1.859 3rd Qu.:2067 3rd Qu.:2067   
## Max. :0.2410 Max. :3.640 Max. :2241 Max. :2241   
## NAME FIPS FIPSNO CRESS\_ID   
## Length:100 Length:100 Min. :37001 Min. : 1.00   
## Class :character Class :character 1st Qu.:37051 1st Qu.: 25.75   
## Mode :character Mode :character Median :37100 Median : 50.50   
## Mean :37100 Mean : 50.50   
## 3rd Qu.:37150 3rd Qu.: 75.25   
## Max. :37199 Max. :100.00   
## BIR74 SID74 NWBIR74 BIR79   
## Min. : 248 Min. : 0.00 Min. : 1.0 Min. : 319   
## 1st Qu.: 1077 1st Qu.: 2.00 1st Qu.: 190.0 1st Qu.: 1336   
## Median : 2180 Median : 4.00 Median : 697.5 Median : 2636   
## Mean : 3300 Mean : 6.67 Mean :1050.8 Mean : 4224   
## 3rd Qu.: 3936 3rd Qu.: 8.25 3rd Qu.:1168.5 3rd Qu.: 4889   
## Max. :21588 Max. :44.00 Max. :8027.0 Max. :30757   
## SID79 NWBIR79 SIDR74 SIDR79   
## Min. : 0.00 Min. : 3.0 Min. :0.000 Min. :0.000   
## 1st Qu.: 2.00 1st Qu.: 250.5 1st Qu.:1.084 1st Qu.:1.249   
## Median : 5.00 Median : 874.5 Median :1.855 Median :2.075   
## Mean : 8.36 Mean : 1352.8 Mean :2.046 Mean :2.039   
## 3rd Qu.:10.25 3rd Qu.: 1406.8 3rd Qu.:2.604 3rd Qu.:2.539   
## Max. :57.00 Max. :11631.0 Max. :9.554 Max. :6.114   
## NWR74 NWR79 geometry   
## Min. : 1.49 Min. : 3.24 MULTIPOLYGON :100   
## 1st Qu.:123.47 1st Qu.:120.20 epsg:NA : 0   
## Median :304.99 Median :307.62 +proj=long...: 0   
## Mean :312.50 Mean :310.45   
## 3rd Qu.:469.27 3rd Qu.:463.09   
## Max. :772.73 Max. :759.22

### Plot the SIDR74 from 1974-1978

# If `sids` is an sf object, convert it to SpatialPolygonsDataFrame or similar  
if (class(sids)[1] == "sf") {  
 library(sf)  
 sids <- as(sids, "Spatial") # Converts sf object to sp class  
}  
  
if (!"SIDR74" %in% names(sids)) {  
 stop("Column 'SIDR74' does not exist in 'sids'")  
}  
  
spplot(sids, "SIDR74")

##### Result

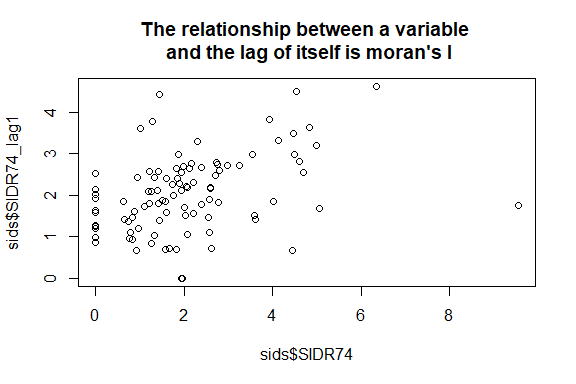


### Define a new lagged variable using row standardized weights

sids$SIDR74\_lag1 <- lag(sids\_nbq\_w, sids$SIDR74)  
lm2 <- lm(sids$SIDR74\_lag1~sids$SIDR74)  
lm2  
plot(y= sids$SIDR74\_lag1, x= sids$SIDR74)  
title("The relationship between a variable \n and the lag of itself is moran's I")

##### Result

##   
## Call:  
## lm(formula = sids$SIDR74\_lag1 ~ sids$SIDR74)  
##   
## Coefficients:  
## (Intercept) sids$SIDR74   
## 1.5626 0.2309



Q1. What is the coefficient for variable SIDR74. What does this coefficient mean?

In the context of this analysis, where **SIDR74\_lag1** is a lagged variable created using row standardized weights (presumably reflecting some spatial relationship or influence among observations), the coefficient can be interpreted as the strength and direction of the relationship between an area's original value of **SIDR74** and the spatially weighted average (lag) of **SIDR74** in its neighboring areas.

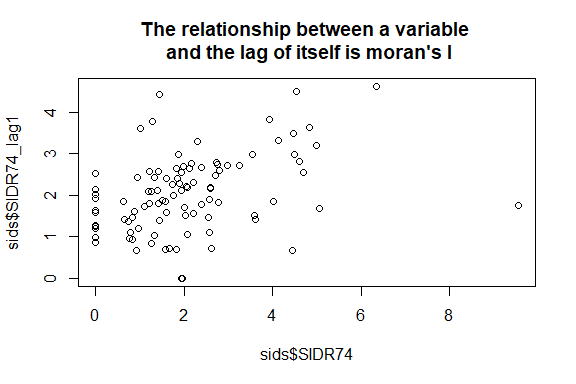
A coefficient of **0.2309** suggests a positive relationship between **SIDR74** and its spatial lag, **SIDR74\_lag1**. This means that areas with higher values of **SIDR74** tend to be surrounded by areas that also have, on average, higher values of **SIDR74**, after adjusting for the influence of spatial proximity as defined by the row standardized weights. This could imply spatial autocorrelation, where similar values cluster together in space. The specific context of **SIDR74** (which might refer to a rate or count of an event or attribute per area) would further determine the substantive interpretation of this finding.

### Define a new lagged variable using row standardized weights

sids$SIDR74\_lag1 <- lag(sids\_nbq\_w, sids$SIDR74)  
lm2 <- lm(sids$SIDR74\_lag1~sids$SIDR74)  
lm2  
plot(y= sids$SIDR74\_lag1, x= sids$SIDR74)  
title("The relationship between a variable \n and the lag of itself is moran's I")

##### Result

##   
## Call:  
## lm(formula = sids$SIDR74\_lag1 ~ sids$SIDR74)  
##   
## Coefficients:  
## (Intercept) sids$SIDR74   
## 1.5626 0.2309



### Global moran’s I test

moran.test (sids$SIDR74, sids\_nbq\_w, randomisation = FALSE) # Moran I test under normality  
moran.test(sids$SIDR74, sids\_nbq\_wb) # Moran I test under randomization using binary weight

##### Result

##   
## Moran I test under normality  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_w   
##   
## Moran I statistic standard deviate = 3.6957, p-value = 0.0001097  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic Expectation Variance   
## 0.230910472 -0.010101010 0.004252954

##   
## Moran I test under randomisation  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_wb   
##   
## Moran I statistic standard deviate = 3.6355, p-value = 0.0001387  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic Expectation Variance   
## 0.210046482 -0.010101010 0.003666802

Q2. What are the results (pattern)? Compare their test statistic, standard deviate, p-value, and variance.

The Moran I tests for **SIDR74** using row-standardized and binary weights reveal significant positive spatial autocorrelation, indicating that similar values tend to cluster together spatially. The first test, with row-standardized weights (**sids\_nbq\_w**), shows a Moran I statistic of 0.2309, a standard deviate of 3.6957, and a p-value of 0.0001097, suggesting a strong spatial linkage. The second test, using binary weights (**sids\_nbq\_wb**), reports a Moran I of 0.2100, a standard deviate of 3.6355, and a p-value of 0.0001387, also indicating significant autocorrelation but to a slightly lesser degree.

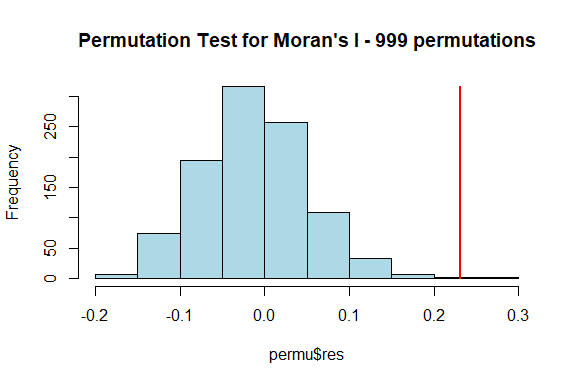
Comparatively, the row-standardized weights produce a higher Moran I statistic and variance, suggesting stronger autocorrelation than with binary weights. Both tests, however, firmly reject the null hypothesis of spatial randomness, confirming the presence of meaningful spatial patterns in **SIDR74**. The differences in Moran I statistics and variances between the tests highlight how weighting schemes can influence the perceived strength of spatial relationships.

### Monte-Carlo simulation of Moran’s I

set.seed(1234)  
permu<-moran.mc(sids$SIDR74, sids\_nbq\_w, nsim = 999)  
permu  
graph999 <- hist(permu$res,freq=TRUE,col="light blue",main="Permutation Test for Moran's I - 999 permutations")  
lines(permu$statistic,max(graph999$counts),type="h",col="red",lwd=2)

##### Result

##   
## Monte-Carlo simulation of Moran I  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_w   
## number of simulations + 1: 1000   
##   
## statistic = 0.23091, observed rank = 999, p-value = 0.001  
## alternative hypothesis: greater



Q3. Explain the histogram.

The histogram represents the distribution of Moran's I values obtained from a permutation test with 999 simulations, used to assess the significance of spatial autocorrelation in the **SIDR74** variable. The light blue bars show the frequency of different Moran's I statistics generated under the null hypothesis of spatial randomness. The red vertical line represents the observed Moran's I statistic from the actual data, which is positioned at the extreme right of the distribution, indicating a higher than expected value under the null hypothesis. With an observed rank of 999 out of 1000 (including the actual observation), the p-value is 0.001, suggesting that the observed spatial autocorrelation (Moran's I of 0.23091) is statistically significant and unlikely to be due to random chance. This implies a strong positive spatial autocorrelation in the **SIDR74** data, meaning that similar values tend to cluster together in space.

### local Empirical Bayes estimates

set.seed(1234)  
EBImoran.mc(n = sids$SID74, x = sids$BIR74, sids\_nbq\_wb, nsim = 999)

##### Result

## The default for subtract\_mean\_in\_numerator set TRUE from February 2016

##   
## Monte-Carlo simulation of Empirical Bayes Index (mean subtracted)  
##   
## data: cases: sids$SID74, risk population: sids$BIR74  
## weights: sids\_nbq\_wb  
## number of simulations + 1: 1000  
##   
## statistic = 0.24299, observed rank = 1000, p-value = 0.001  
## alternative hypothesis: greater

Q4. How is the result from this method different from previous results?

The result from the Monte-Carlo simulation of the Empirical Bayes Index (EBI) represents a different approach to assessing spatial autocorrelation, specifically within the context of an Empirical Bayes framework. Unlike the previous Moran's I test which directly measured spatial autocorrelation in the **SIDR74** variable, the EBI Moran's I accounts for underlying population risk by considering both the cases (**SID74**) and the risk population (**BIR74**). Here, the test statistic is 0.24299 with an observed rank of 1000 out of 1000, yielding a p-value of 0.001, indicating that the positive spatial autocorrelation is statistically significant.

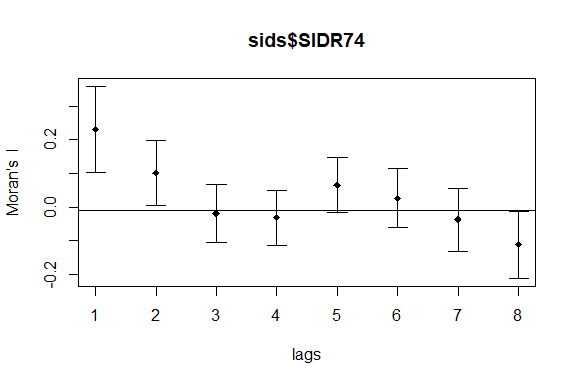
The primary difference from the previous results is that this method adjusts for the underlying population at risk, which is an important consideration in spatial epidemiology. By doing so, it potentially provides a more accurate measure of spatial autocorrelation by accounting for the variability due to the population at risk. This approach might reveal different patterns of spatial distribution that are not apparent when looking at raw counts alone, as it essentially standardizes the **SID74** variable by the **BIR74** variable before assessing spatial autocorrelation. The EBI Moran's I is particularly useful when dealing with rate data or when there is a need to control for underlying population heterogeneity.

### Explore the Correlogram

cor8<-sp.correlogram(sids\_nbq, sids$SIDR74, order=8, method="I")  
cor8  
plot(cor8)

##### Result

## Spatial correlogram for sids$SIDR74   
## method: Moran's I  
## estimate expectation variance standard deviate Pr(I) two sided   
## 1 (100) 0.2309105 -0.0101010 0.0040651 3.7801 0.0001568 \*\*\*  
## 2 (100) 0.1008680 -0.0101010 0.0023725 2.2782 0.0227118 \*   
## 3 (100) -0.0193919 -0.0101010 0.0018274 -0.2173 0.8279410   
## 4 (100) -0.0325159 -0.0101010 0.0016559 -0.5508 0.5817448   
## 5 (100) 0.0650216 -0.0101010 0.0017007 1.8216 0.0685140 .   
## 6 (100) 0.0257709 -0.0101010 0.0019062 0.8216 0.4112923   
## 7 (100) -0.0373522 -0.0101010 0.0022010 -0.5809 0.5613297   
## 8 (100) -0.1123651 -0.0101010 0.0024863 -2.0509 0.0402763 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



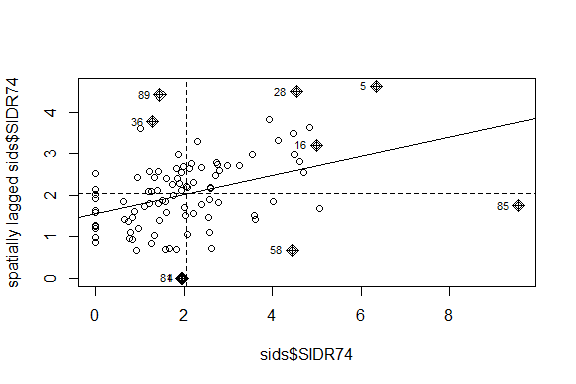
Q5.Explain the plot

The plot is a spatial correlogram depicting Moran's I statistic across eight spatial lags for the variable **SIDR74**. The correlogram shows that spatial autocorrelation is positive and significant at the first lag, decreases thereafter, becomes slightly positive at the fifth, and significantly negative at the eighth lag, suggesting a complex spatial structure where similarity between values decreases with distance.

### Explore the local Moran’ I

#First plot the moran scatterplot  
moran.plot(sids$SIDR74, sids\_nbq\_w)

##### Result



Q6.Explain the plot

### LISA

locm <- localmoran(sids$SIDR74, sids\_nbq\_w) #Calculate the local Moran’s I  
locm  
summary(locm)  
sids$sSIDR74<-scale(sids$SIDR74) #Standardize it  
sids$lag\_sSIDR74<-lag(sids\_nbq\_w, sids$sSIDR74) #Define a lag variable  
sids$sSIDR74  
sids$lag\_sSIDR74  
sids$quad\_sig <- NA #Define a new variable quad\_sig and set the initial value to be NA  
#Next define five types of results: High-High, Low-Low, High-Low, Low-High, and not significant  
sids@data[(sids$sSIDR74>= 0 & sids$lag\_sSIDR74>= 0) & (locm[,5]<= 0.05),"quad\_sig"] <- 1  
sids@data[(sids$sSIDR74<= 0 & sids$lag\_sSIDR74<= 0) & (locm[,5]<= 0.05),"quad\_sig"] <- 2  
sids@data[(sids$sSIDR74>= 0 & sids$lag\_sSIDR74<= 0) & (locm[,5]<= 0.05),"quad\_sig"] <- 3  
sids@data[(sids$sSIDR74<= 0 & sids$lag\_sSIDR74>= 0) & (locm[,5]<= 0.05),"quad\_sig"] <- 4  
sids@data[locm[,5]> 0.05, "quad\_sig"] <- 5

##### Result

## Ii E.Ii Var.Ii Z.Ii Pr(z != E(Ii))  
## 1 0.6310748786 -5.253842e-03 1.706527e-01 1.54036966 0.1234702732  
## 2 0.6623096739 -1.724742e-02 5.534677e-01 0.91343954 0.3610114160  
## 3 0.2611273135 -9.386694e-04 1.799022e-02 1.95385513 0.0507183520  
## 4 0.0643500578 -2.449646e-05 1.212295e-03 1.84888574 0.0644743138  
## 5 4.5018072121 -7.578589e-02 1.697456e+00 3.51348378 0.0004422714  
## 6 1.7850003736 -3.174806e-02 1.003759e+00 1.81334350 0.0697788582  
## 7 0.6501842893 -1.724742e-02 5.534677e-01 0.89714098 0.3696436947  
## 8 0.0296937560 -1.724742e-02 3.251623e-01 0.08231972 0.9343924700  
## 9 1.0811054665 -1.794640e-02 3.380994e-01 1.89014881 0.0587380603  
## 10 0.1095145237 -8.372708e-03 2.012111e-01 0.26280934 0.7926975396  
## 11 -0.0234643632 -5.284452e-05 1.280603e-03 -0.65421774 0.5129715318  
## 12 -0.3326221272 -9.911744e-03 1.882590e-01 -0.74376382 0.4570193714  
## 13 -0.0363897589 -4.996801e-04 9.580910e-03 -0.36666633 0.7138679229  
## 14 -0.0313095732 -1.136489e-03 2.751116e-02 -0.18191365 0.8556504911  
## 15 -0.0521339751 -1.830433e-04 5.975829e-03 -0.67203867 0.5015590787  
## 16 1.3937422098 -3.570745e-02 4.617760e-01 2.10355172 0.0354175694  
## 17 0.1185411250 -1.889268e-04 6.167872e-03 1.51179523 0.1305859620  
## 18 0.3832955417 -2.470151e-03 2.860057e-02 2.28105755 0.0225450418  
## 19 0.5699442407 -6.856263e-03 1.650202e-01 1.41989778 0.1556374432  
## 20 -0.0082999710 -1.735555e-06 5.667108e-05 -1.10231323 0.2703255195  
## 21 0.2950205870 -2.101489e-03 1.037837e-01 0.92229603 0.3563741842  
## 22 0.6412057897 -1.724742e-02 3.251623e-01 1.15471509 0.2482071173  
## 23 0.4874935375 -6.518558e-03 1.242348e-01 1.40157401 0.1610424940  
## 24 -0.1301779297 -1.564048e-03 2.094277e-02 -0.88873200 0.3741471299  
## 25 0.2834456297 -5.957964e-03 7.942667e-02 1.02688249 0.3044757992  
## 26 0.0597937618 -1.607180e-03 2.537883e-02 0.38542443 0.6999229861  
## 27 -0.0638277063 -2.238444e-03 3.532472e-02 -0.32769185 0.7431446668  
## 28 2.4869124672 -2.547597e-02 4.762720e-01 3.64048516 0.0002721248  
## 29 -0.0178686412 -2.516525e-03 4.815471e-02 -0.06995985 0.9442256168  
## 30 0.0052179951 -5.973172e-06 1.145867e-04 0.48801529 0.6255390112  
## 31 -0.0148394720 -1.294483e-05 1.736020e-04 -1.12528398 0.2604687163  
## 32 0.9925945877 -1.724742e-02 5.534677e-01 1.35739827 0.1746546984  
## 33 0.2114656908 -1.956046e-03 3.745076e-02 1.10282946 0.2701012333  
## 34 0.2078520056 -6.048909e-04 9.561365e-03 2.13184840 0.0330193115  
## 35 0.3758372032 -1.724742e-02 4.107768e-01 0.61331431 0.5396685341  
## 36 -0.5326215117 -2.346032e-03 3.701855e-02 -2.75608003 0.0058498683  
## 37 0.1203575976 -3.649204e-03 4.876117e-02 0.56157585 0.5744050352  
## 38 -0.3087884896 -1.333453e-03 4.348325e-02 -1.47441789 0.1403691106  
## 39 0.3729374976 -4.800334e-03 4.874786e-02 1.71085298 0.0871082567  
## 40 0.5489630746 -6.105681e-03 1.164142e-01 1.62683581 0.1037719740  
## 41 0.7122310666 -1.724742e-02 4.107768e-01 1.13817628 0.2550468830  
## 42 0.1555643871 -1.451507e-03 1.943802e-02 1.12620548 0.2600785533  
## 43 -0.0172359813 -1.721213e-03 2.304359e-02 -0.10220456 0.9185943088  
## 44 -0.4425998607 -3.721749e-02 6.873961e-01 -0.48894648 0.6248795857  
## 45 -0.4003776555 -1.724742e-02 8.388495e-01 -0.41831593 0.6757161457  
## 46 -0.2018287843 -1.130778e-03 1.786453e-02 -1.50157593 0.1332066629  
## 47 0.0364695019 -9.287233e-04 1.467535e-02 0.30871431 0.7575388541  
## 48 0.0820757784 -2.843113e-03 3.290659e-02 0.46812611 0.6396944080  
## 99 0.0374858053 -6.539265e-05 3.236057e-03 0.66010907 0.5091838390  
## 100 0.1266134045 -2.513229e-04 8.204400e-03 1.40061081 0.1613304873  
## attr(,"call")  
## localmoran(x = sids$SIDR74, listw = sids\_nbq\_w)  
## attr(,"class")  
## [1] "localmoran" "matrix" "array"   
## attr(,"quadr")  
## mean median pysal  
## 1 Low-Low Low-Low Low-Low  
## 2 Low-Low Low-Low Low-Low  
## 3 Low-Low Low-Low Low-Low  
## 4 Low-Low High-Low Low-Low  
## 5 High-High High-High High-High  
## 6 High-High High-High High-High  
## 7 Low-Low Low-Low Low-Low  
## 8 Low-Low Low-High Low-Low  
## 9 High-High High-High High-High  
## 10 Low-Low Low-Low Low-Low  
## 11 Low-High High-High Low-High  
## 12 High-Low High-Low High-Low  
## 13 High-Low High-Low High-Low  
## 14 High-Low High-Low High-Low  
## 15 Low-High Low-High Low-High  
## 16 High-High High-High High-High  
## 17 Low-Low Low-Low Low-Low  
## 18 Low-Low Low-Low Low-Low  
## 19 Low-Low Low-Low Low-Low  
## 20 High-Low High-Low High-Low  
## 21 Low-Low Low-Low Low-Low  
## 22 Low-Low Low-Low Low-Low  
## 23 Low-Low Low-Low Low-Low  
## 24 Low-High Low-High Low-High  
## 25 Low-Low Low-Low Low-Low  
## 26 Low-Low Low-Low Low-Low  
## 27 High-Low High-Low High-Low  
## 28 High-High High-High High-High  
## 29 Low-High Low-High Low-High  
## 30 Low-Low High-Low Low-Low  
## 31 Low-High High-High Low-High  
## 32 Low-Low Low-Low Low-Low  
## 33 High-High High-High High-High  
## 34 Low-Low Low-Low Low-Low  
## 35 Low-Low Low-Low Low-Low  
## 36 Low-High Low-High Low-High  
## 37 Low-Low Low-Low Low-Low  
## 38 High-Low High-Low High-Low  
## 39 Low-Low Low-Low Low-Low  
## 40 Low-Low Low-Low Low-Low  
## 41 Low-Low Low-Low Low-Low  
## 42 Low-Low Low-Low Low-Low  
## 92 High-High High-High High-High  
## 93 High-High High-High High-High  
## 94 High-High High-High High-High  
## 95 High-High High-High High-High  
## 96 High-High High-High High-High  
## 97 High-High High-High High-High  
## 98 High-High High-High High-High  
## 99 High-High High-High High-High  
## 100 High-High High-High High-High

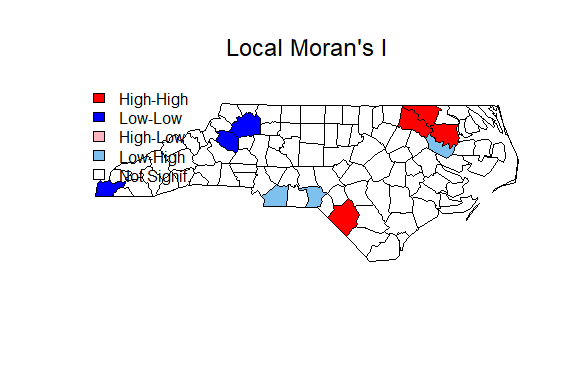
## Ii E.Ii Var.Ii Z.Ii   
## Min. :-1.34220 Min. :-2.324e-01 Min. :0.000004 Min. :-3.4278   
## 1st Qu.:-0.03206 1st Qu.:-1.625e-02 1st Qu.:0.009576 1st Qu.:-0.3400   
## Median : 0.06207 Median :-2.408e-03 Median :0.038541 Median : 0.5248   
## Mean : 0.23091 Mean :-1.010e-02 Mean :0.224349 Mean : 0.4336   
## 3rd Qu.: 0.36889 3rd Qu.:-5.786e-04 3rd Qu.:0.288689 3rd Qu.: 1.1423   
## Max. : 4.50181 Max. :-2.300e-07 Max. :4.322954 Max. : 3.6405   
## Pr(z != E(Ii))   
## Min. :0.0002721   
## 1st Qu.:0.1612585   
## Median :0.3681420   
## Mean :0.4190472   
## 3rd Qu.:0.6500945   
## Max. :0.9478855

## [,1]  
## [1,] -0.717585539  
## [2,] -1.300161466  
## [3,] -0.303313349  
## [4,] -0.048998958  
## [5,] 2.725394399  
## [6,] 1.763980455  
## [7,] -1.300161466  
## [8,] -1.300161466  
## [9,] 1.326245441  
## [10,] -0.905874785  
## [11,] -0.071967293  
## [12,] 0.985621669  
## [13,] 0.221299899  
## [14,] 0.333747305  
## [15,] -0.133940548  
## [16,] 1.870745046  
## [17,] -0.136076132  
## [18,] -0.492036068  
## [19,] -0.819745277  
## [64,] 0.11205358  
## [65,] -0.39462176  
## [66,] -0.03694816  
## [67,] 0.14971002  
## [68,] -0.34036307  
## [69,] -0.43577588  
## [70,] 0.39690040  
## [71,] 0.37313316  
## [72,] 0.04600467  
## [73,] 0.05404128  
## [74,] 0.35387349  
## [75,] -0.37255601  
## [76,] 0.33189795  
## [77,] 0.59173636  
## [78,] -0.26521097  
## [79,] 0.24563957  
## [80,] 0.23024615  
## [81,] -1.30016147  
## [82,] 0.59641770  
## [83,] 0.13410374  
## [84,] 0.99534303  
## [85,] -0.19104502  
## [86,] 0.31203248  
## [87,] -0.08446713  
## [88,] 0.22596874  
## [89,] 1.50408634  
## [90,] -0.68128061  
## [91,] 0.15971499  
## [92,] 0.59938458  
## [93,] 0.07742349  
## [94,] 1.12345774  
## [95,] 0.08638818  
## [96,] 0.58787699  
## [97,] 0.42168316  
## [98,] 0.91943429  
## [99,] 0.46355618  
## [100,] 0.79866314

### Plot the LISA Map

breaks <- seq(1, 5, 1)  
labels <- c("High-High", "Low-Low", "High-Low", "Low-High", "Not Signif.")  
np <- findInterval(sids$quad\_sig, breaks)  
colors <- c("red", "blue", "lightpink", "skyblue2", "white")  
plot(sids, col = colors[np]) #colors[np] manually sets the color for each county  
mtext("Local Moran's I", cex = 1.5, side = 3, line = 1)  
legend("topleft", legend = labels, fill = colors, bty = "n")

##### Result



Q7. Plot and explain the pattern.

The map shows Local Moran's I results, identifying clusters of similar values (high-high or low-low) and spatial outliers (high-low or low-high) for a given variable. Red areas indicate clusters of high values, while blue areas show clusters of low values. Light red and light blue denote spatial outliers. Areas in white are not statistically significant, suggesting no strong local spatial autocorrelation.

### explore the Getis G

globalG.test(sids$SIDR74, sids\_nbq\_wb)

##### Result

##   
## Getis-Ord global G statistic  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_wb   
##   
## standard deviate = 2.4526, p-value = 0.007092  
## alternative hypothesis: greater  
## sample estimates:  
## Global G statistic Expectation Variance   
## 5.710707e-02 4.949495e-02 9.633064e-06

Q8. What are your results and explain the pattern (clustered? Hotspot or cold spot?)

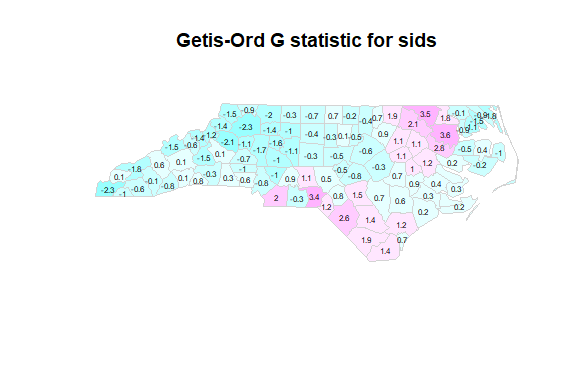
The Getis-Ord global G statistic analysis conducted on **sids$SIDR74** with binary weights **sids\_nbq\_wb** yields a standard deviate of 2.4526 and a p-value of 0.007092. This indicates that the spatial distribution of **SIDR74** values is statistically significantly different from what would be expected under spatial randomness.

Since the global G statistic is positive and the test's alternative hypothesis is 'greater', the results suggest a pattern of clustering within the dataset. Specifically, because the standard deviate is positive, it points towards the presence of hotspots – areas where high values of **SIDR74** are more concentrated than would be expected if underlying spatial processes were random. The significance of the G statistic means that these hotspots are not likely to be the result of random variation but rather reflect a meaningful spatial pattern.

### Local Getis G

G <- localG(sids$SIDR74, sids\_nbq\_wb)  
par.G <- par(mfrow=c(1,1))  
# set the graphics parameters to display to maps on top of each other  
# in the same window  
brks <- seq(-4,6,1)  
# creates 10 intervals between -4 and +5  
# summary(G) the variable to be plotted has a low of -3.6 and high of 5.7  
cm.col <- cm.colors(length(brks)-1)  
# picks colors corresponding for each interval out of the "cm" color palette  
plot(sids,border="lightgray",col=cm.col[findInterval(G,brks,all.inside=TRUE)])  
# maps the shape file and fills each polygon with a color corresponding to  
# the level of G at that location  
text(coordinates(sids)[,1],coordinates(sids)[,2], round(G, digits=1), cex=0.5)  
# writes the G-statistic in each polygon  
title("Getis-Ord G statistic for sids")

##### Result



Q9.Is it different from the LISA map in Q7? Why?

**Yes,** the Getis-Ord G statistic map differs from the Local Moran's I map. While the Local Moran's I identifies clusters of high or low values and outliers, the Getis-Ord G map focuses on the intensity and significance of spatial hotspots and cold spots. Values on the Getis-Ord map indicate where high or low values cluster spatially but do not necessarily reflect similarity among neighbors. This approach highlights areas with statistically significant high or low values relative to the entire study area, providing a different aspect of spatial patterns.

### Geary’s C

library(spdep)  
geary.test(sids$SIDR74, sids\_nbq\_wb, randomisation=FALSE)  
gc\_perm <- geary.mc(sids$SIDR74, sids\_nbq\_wb, nsim=99)  
gc\_perm

##### Result

##   
## Geary C test under normality  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_wb   
##   
## Geary C statistic standard deviate = 4.1465, p-value = 1.688e-05  
## alternative hypothesis: Expectation greater than statistic  
## sample estimates:  
## Geary C statistic Expectation Variance   
## 0.67796675 1.00000000 0.00603181

##   
## Monte-Carlo simulation of Geary C  
##   
## data: sids$SIDR74   
## weights: sids\_nbq\_wb   
## number of simulations + 1: 100   
##   
## statistic = 0.67797, observed rank = 1, p-value = 0.01  
## alternative hypothesis: greater

Q10. What is the pattern? Are these results different? Explain the differences if any.

The Geary C test shows a statistic of 0.67797, which is less than the expected value of 1 under the null hypothesis of no spatial autocorrelation. The low p-values from both the normality test and the Monte-Carlo simulation indicate a significant spatial autocorrelation. Geary's C is inversely related to Moran's I; hence, a value less than 1 suggests positive autocorrelation. The results from the normality test and the Monte-Carlo simulation are consistent, reinforcing the presence of spatial autocorrelation within the **sids$SIDR74** data.