空間自相關

空間分析 2021.05.24
TA 杜承軒

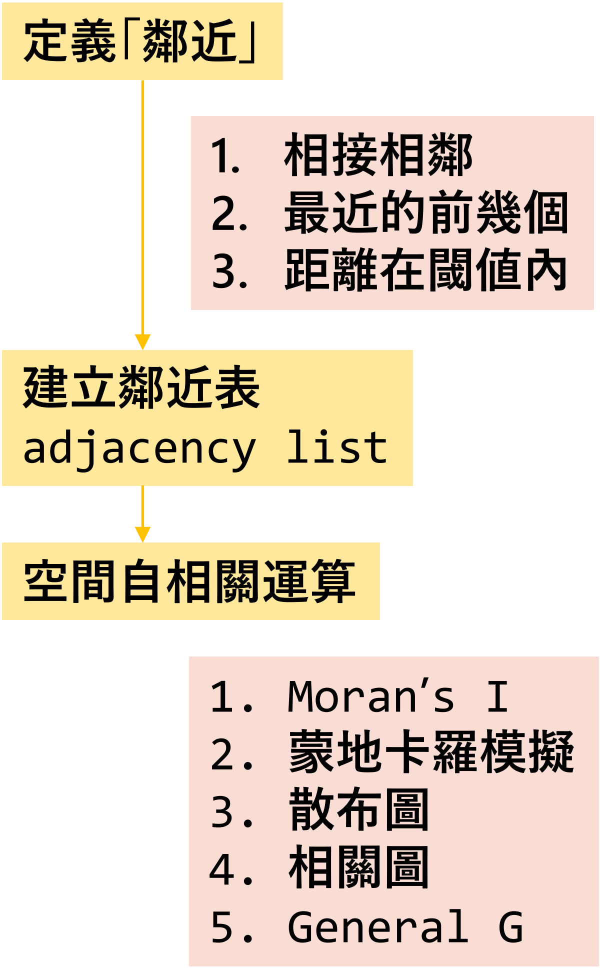
Lab 9

台灣鄉鎮市區人口密度的空間型態分析（資料：Popn_TWN2.shp）

1. **計算以下統計量與繪製圖表，說明其參數設定，並解釋其意義。**
   鄰近：Contiguity(Queen)
   列標準化Row-standardized：TRUE
   (1) Moran’s I coefficient
   (2) Monte-Carlo simulation
   (3) Moran scatter plot
   (4) Correlogram
   (5) General G statistic

2. **利用以下三種不同的空間鄰近定義，計算Moran’s I coefficient，比較其數值的差異，並討論可能的原因。**
   Spatial Neighbors:
   (1) Contiguity
   (2) K-nearest Neighbors (KNN)
   (3) Distance-based

實作



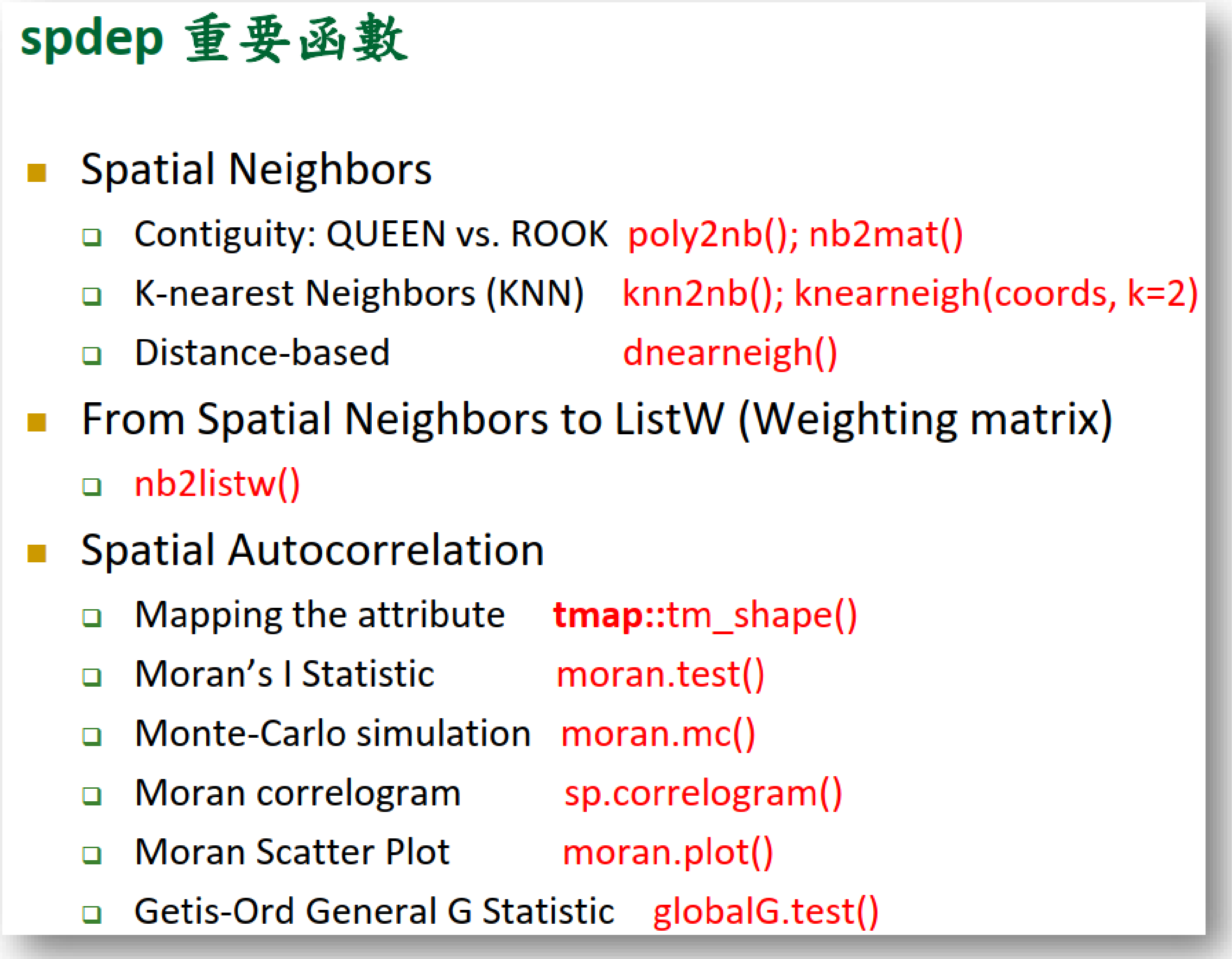
定義「鄰近」

1. 相接相鄰
2. 最近的前幾個
3. 距離在閾值內

建立鄰近表
adjacency list

空間自相關運算

1. Moran's I
2. 蒙地卡羅模擬
3. 散布圖
4. 相關圖
5. General G



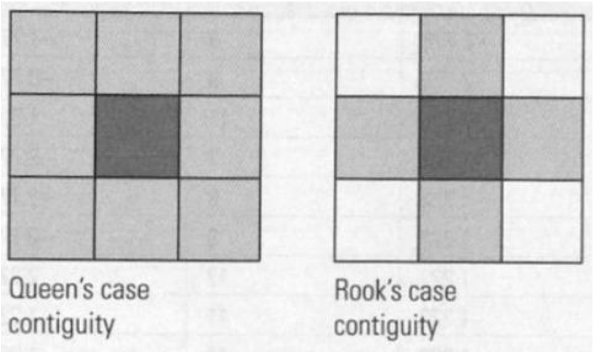
spdep 重要函數

- Spatial Neighbors
  - ❑ Contiguity: QUEEN vs. ROOK poly2nb(); nb2mat()
  - ❑ K-nearest Neighbors (KNN) knn2nb(); knearneigh(coords, k=2)
  - ❑ Distance-based dnearneigh()
- From Spatial Neighbors to ListW (Weighting matrix)
  - ❑ nb2listw()
- Spatial Autocorrelation
  - ❑ Mapping the attribute **tmap::**tm_shape()
  - ❑ Moran's I Statistic moran.test()
  - ❑ Monte-Carlo simulation moran.mc()
  - ❑ Moran correlogram sp.correlogram()
  - ❑ Moran Scatter Plot moran.plot()
  - ❑ Getis-Ord General G Statistic globalG.test()

鄰近

1. 相接相鄰

```
TW.nb    = poly2nb(TW) #預設queen=T
TW.nb    = poly2nb(TW,queen=F)
```

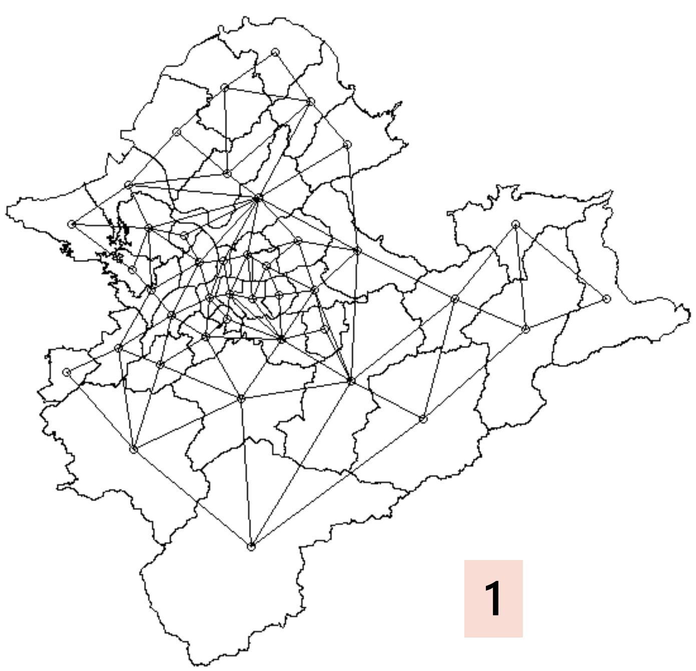


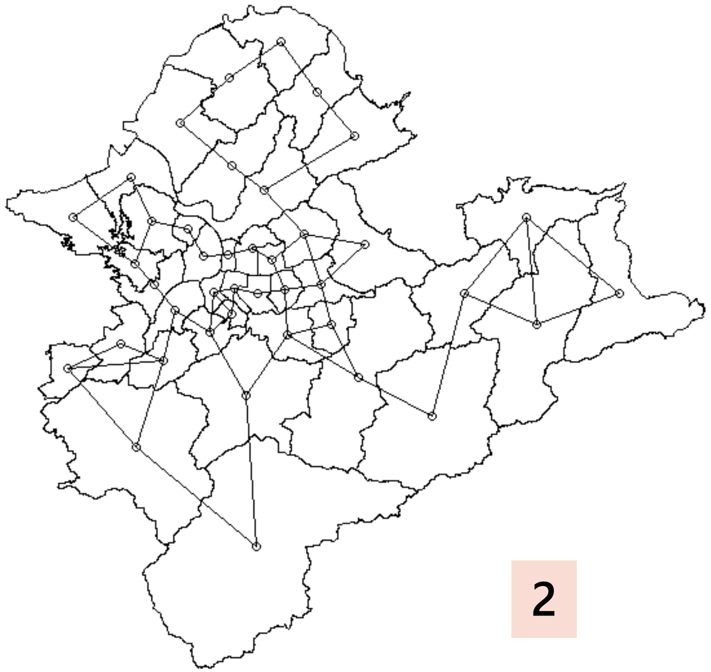
2. 最近的前幾個

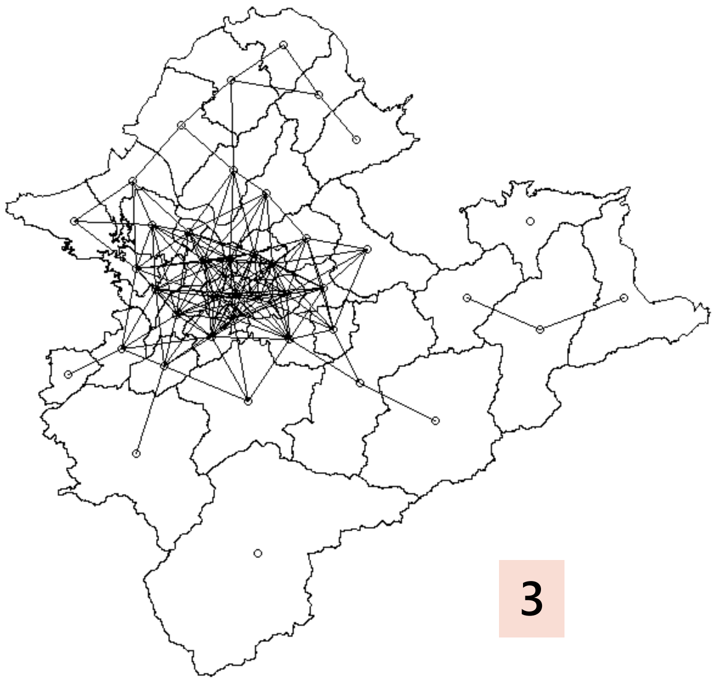
```
TW.cent = st_centroid(TW)
coords  = st_coordinates(TW.cent)
TW.nb   = knn2nb(knearneigh(coords, k=2))
                                          #前兩鄰近
```

3. 距離在閾值內

```
TW.nb   = dnearneigh(coords, d1=0, d2=10000)
```


1


2


3

鄰近表

鄰近目錄

```
TW.nb.w =  nb2listw(TW.nb,zero.policy=T) #預設style="W"(列標準化)
TW.nb.w =  nb2listw(TW.nb, style="B" ,zero.policy=T)
```

概念一樣
格式不一樣

鄰近矩陣

```
TW.nb.WM = nb2mat(TW.nb,zero.policy=T) #預設style="W"
```

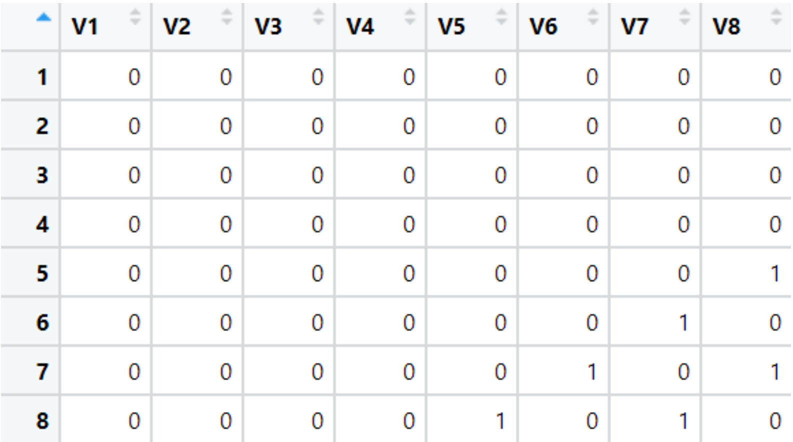
zero.policy=T

如果有些圖徵沒有鄰居，要打上 zero.policy=T

```
> TW.nb
Neighbour list object:
Number of regions: 368
Number of nonzero links: 1936
Percentage nonzero weights: 1.429584
Average number of links: 5.26087
11 regions with no links:
1 2 3 4 9 10 165 207 208 367 368
> TW.nb[8]
[[1]]
[1] 5 7
```

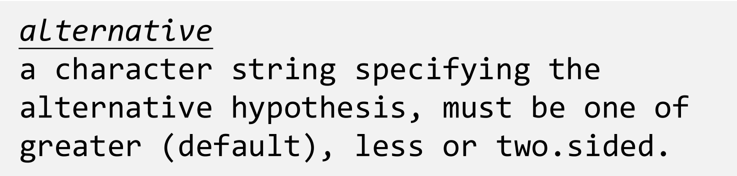
```
> TW.nb.w$neighbours
Neighbour list object:
Number of regions: 368
Number of nonzero links: 1936
Percentage nonzero weights: 1.429584
Average number of links: 5.26087
11 regions with no links:
1 2 3 4 9 10 165 207 208 367 368
> TW.nb.w$neighbours[8]
[[1]]
[1] 5 7
```

```
> TW.nb.WM (style="B")
```



	V1	V2	V3	V4	V5	V6	V7	V8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	1	0	1
8	0	0	0	0	1	0	1	0

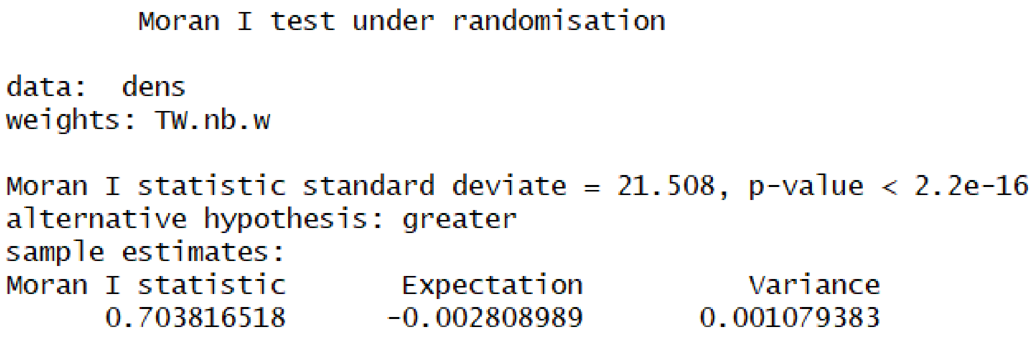
空間自相關運算



```
alternative
a character string specifying the
alternative hypothesis, must be one of
greater (default), less or two.sided.
```

1. Moran’s I coefficient

```
M = moran.test(dens, TW.nb.w, zero.policy=T) #randomisation
M = moran.test(dens, TW.nb.w, randomisation=F, zero.policy=T) #normalization
```

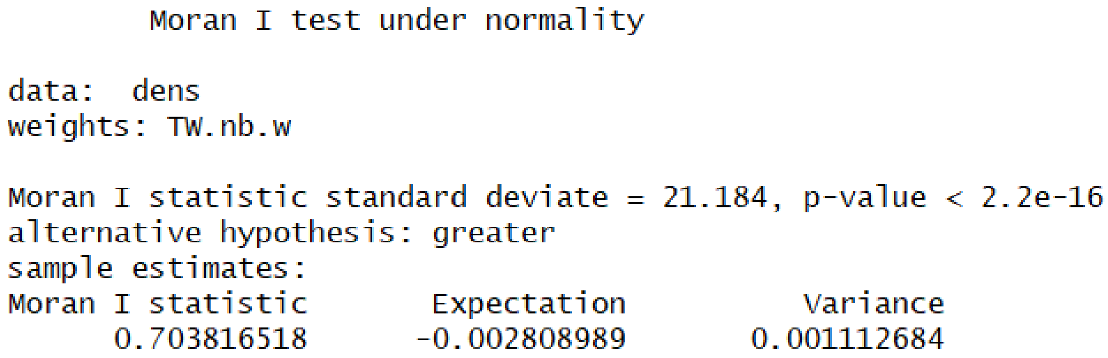


```
	Moran I test under randomisation

data:  dens  
weights: TW.nb.w  

Moran I statistic standard deviate = 21.508, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance 
      0.703816518      -0.002808989       0.001079383 
```

```
M$estimate[1]
```



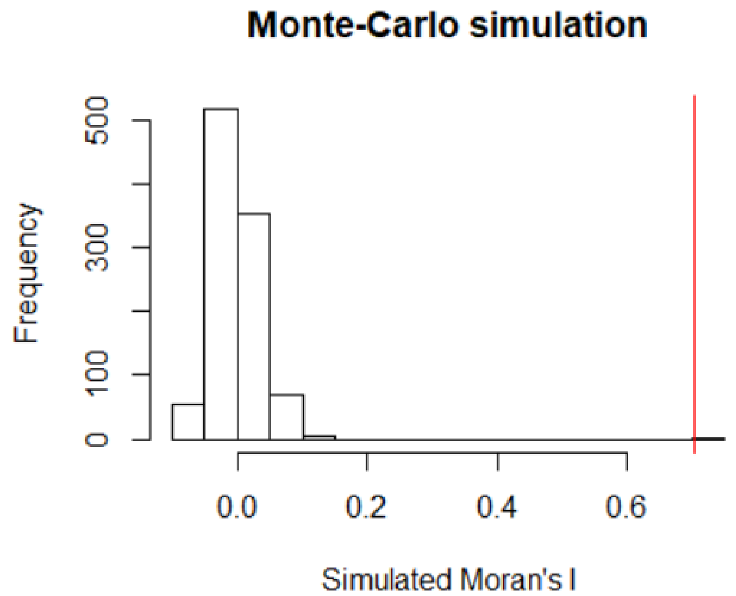
```
	Moran I test under normality

data:  dens  
weights: TW.nb.w  

Moran I statistic standard deviate = 21.184, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance 
      0.703816518      -0.002808989       0.001112684 
```

2. Monte-Carlo simulation

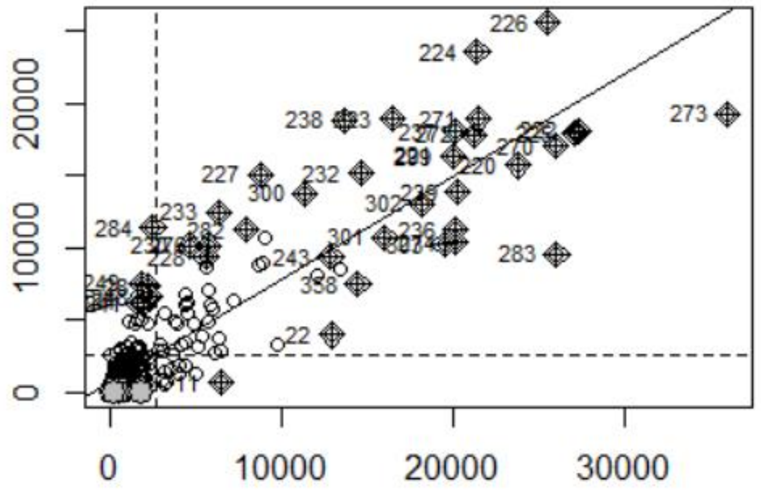
```
mc = moran.mc(dens, TW.nb.w,
              nsim=999,zero.policy=T)
#畫圖
hist(mc$res)
abline(v=mc$statistic, col="red")
```



空間自相關運算

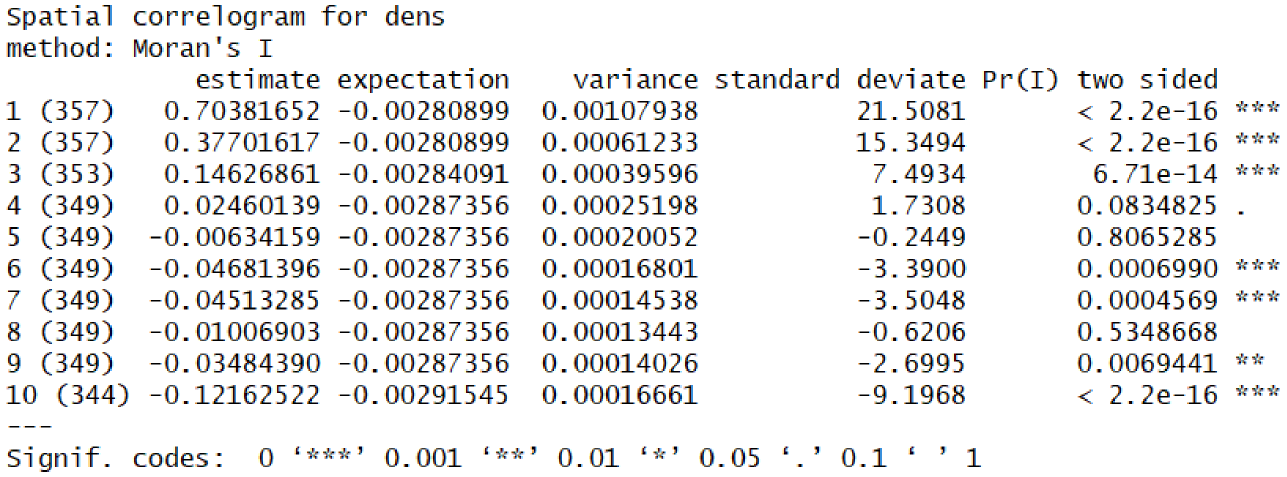
3. Moran scatter plot

```
moran.plot (dens, TW.nb.w, zero.policy=T)
```

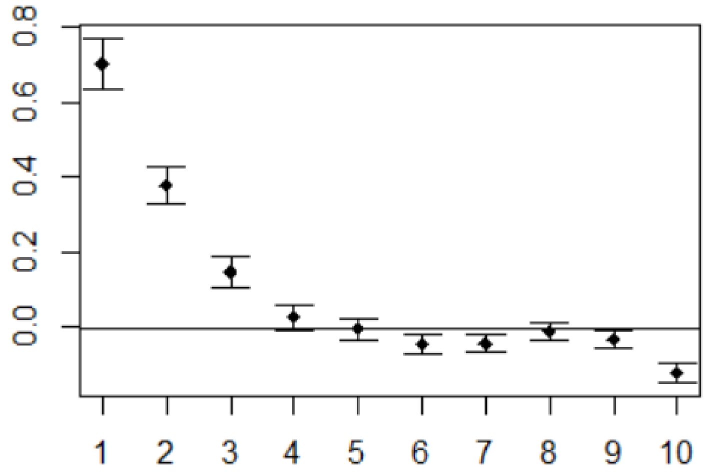


4. Correlogram

```
cor = sp.correlogram(TW.nb, dens, order=10, method="I", style="W",zero.policy=T)
print(cor); plot(cor)
```



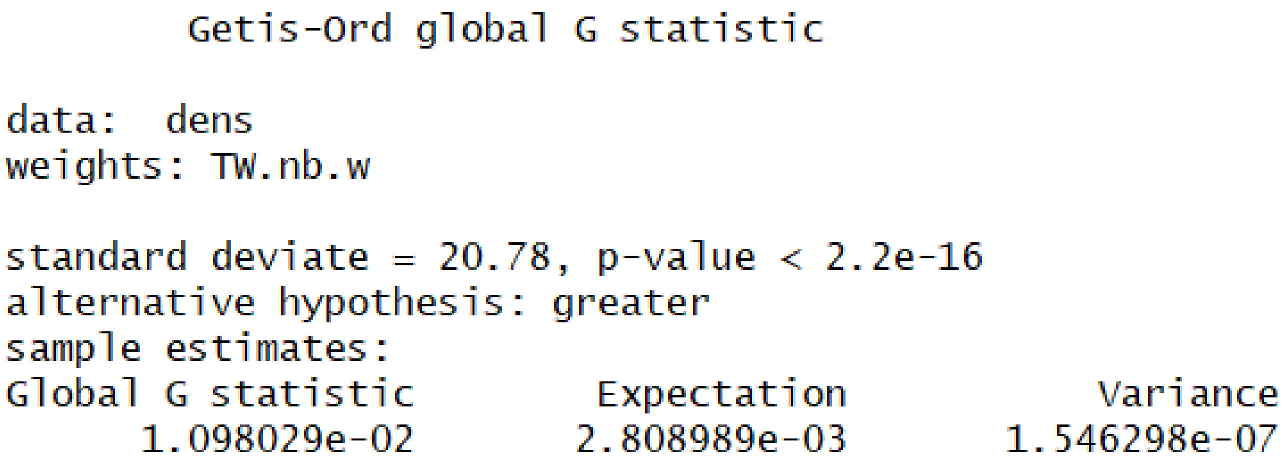
```
Spatial correlogram for dens
method: Moran's I
           estimate expectation   variance standard deviate Pr(I) two sided
1 (357)   0.70381652 -0.00280899 0.00107938          21.5081      < 2.2e-16 ***
2 (357)   0.37701617 -0.00280899 0.00061233          15.3494      < 2.2e-16 ***
3 (353)   0.14626861 -0.00284091 0.00039596           7.4934       6.71e-14 ***
4 (349)   0.02460139 -0.00287356 0.00025198           1.7308      0.0834825 .
5 (349)  -0.00634159 -0.00287356 0.00020052          -0.2449      0.8065285
6 (349)  -0.04681396 -0.00287356 0.00016801          -3.3900      0.0006990 ***
7 (349)  -0.04513285 -0.00287356 0.00014538          -3.5048      0.0004569 ***
8 (349)  -0.01006903 -0.00287356 0.00013443          -0.6206      0.5348668
9 (349)  -0.03484390 -0.00287356 0.00014026          -2.6995      0.0069441 **
10 (344) -0.12162522 -0.00291545 0.00016661          -9.1968      < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



空間自相關運算

5. General G statistic

```
G = globalG.test(dens, TW.nb.w, zero.policy=T)
```



```
        Getis-Ord global G statistic

data:   dens
weights: TW.nb.w

standard deviate = 20.78, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Global G statistic        Expectation          Variance
      1.098029e-02       2.808989e-03      1.546298e-07
```

Neighborhood Definition

$$G_i(d) = \frac{\sum_j w_{ij}(d)x_j}{\sum_j x_j}; j \neq i$$