Spatial Autocorrelation

Dan N.

2024-03-07

Moran’s I

We can assess spatial autocorrelation using this statistic. Specifically, we are interested in knowing if a measure is similar in value for locations in close proximity. We can determine if the pattern of a measure for all locations is random, dispersed or clustered. We will explore Global Moran’s I as well as Local Moran’s I using the same data from the geographic weighted regression example.

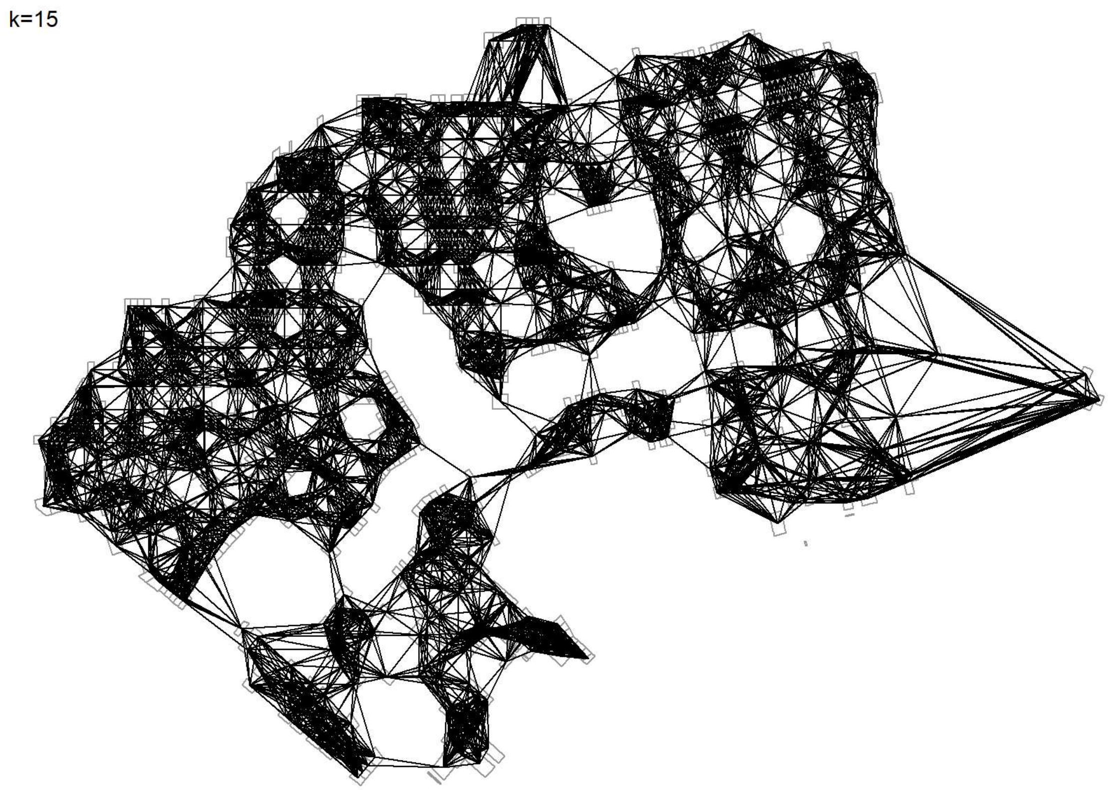
Global Moran’s I

The global test returns a single value describing our entire sample area by averaging each comparison (each location to its neighbors). Moran’s I can be interpreted as follows:

- -1 = perfect dispersion
- 0 = random dispersion
- 1 = perfect clustering

We need to define what a neighbor is. There are different methods such as rook weights, queen weights or k-nearest neighbors. Here I use k-nearest neighbors since some polygons do not share a vertex and would fail to define neighbors using, i.e. queen weights.

```
knear15 <- knn2nb(knearneigh(coordinates(riverside), k=15))
plot(riverside, border="grey60")
plot(knear15, riverside, add=TRUE, pch=".")
text(bbox(riverside)[1,1], bbox(riverside)[2,2], labels="k=15", cex=1)
```



With k = 15, there will be some/many links that cross each other.

But lets move forward and run the test.

```
weights15 <- nb2listw(knear15)
moran.test(riverside$inf_price, weights15)
```

```
##
##  Moran I test under randomisation
##
## data:  riverside$inf_price
## weights: weights15
##
## Moran I statistic standard deviate = 27.741, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      2.618183e-01     -7.390983e-04      8.957901e-05
```

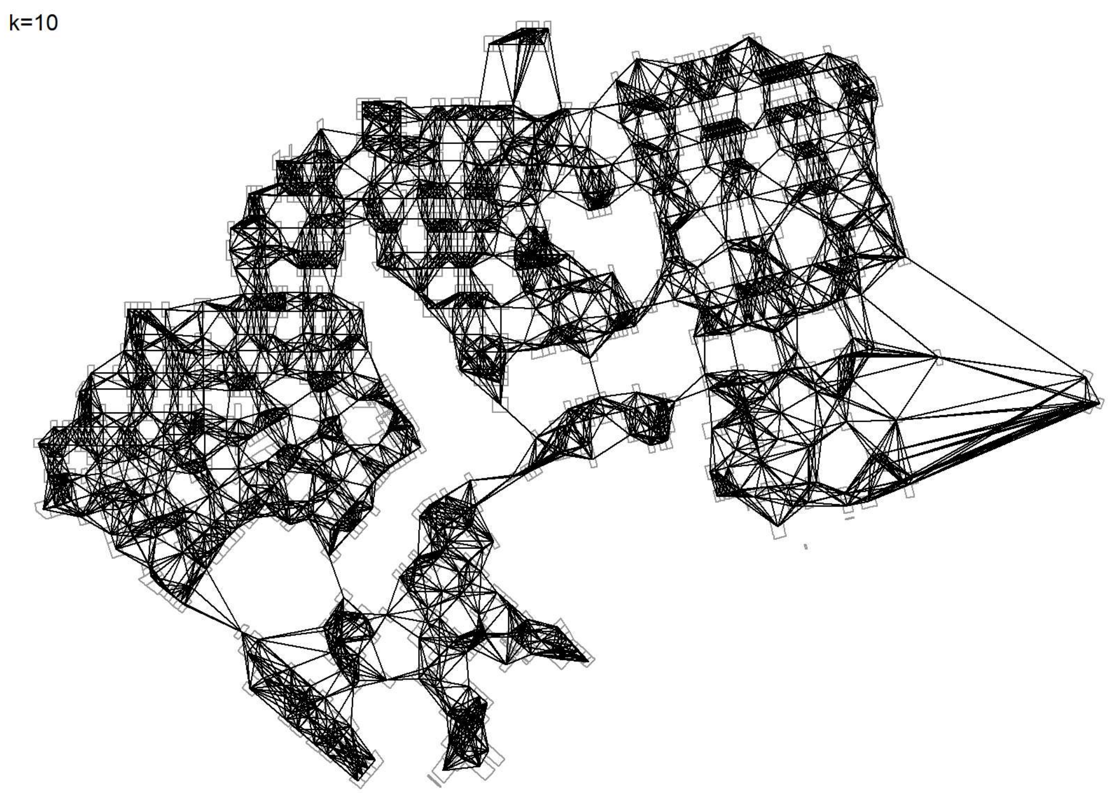
From the output above, we can reject the H0: the data is randomly dispersed in favor of the H1: the data is *not* randomly dispersed. Because Moran's I = 0.2618, we know that there is some clustering happening (neighbors of a particular location have similar values to that location).

Local Moran’s I and Local Indicator of Spatial Association (LISA Clusters)

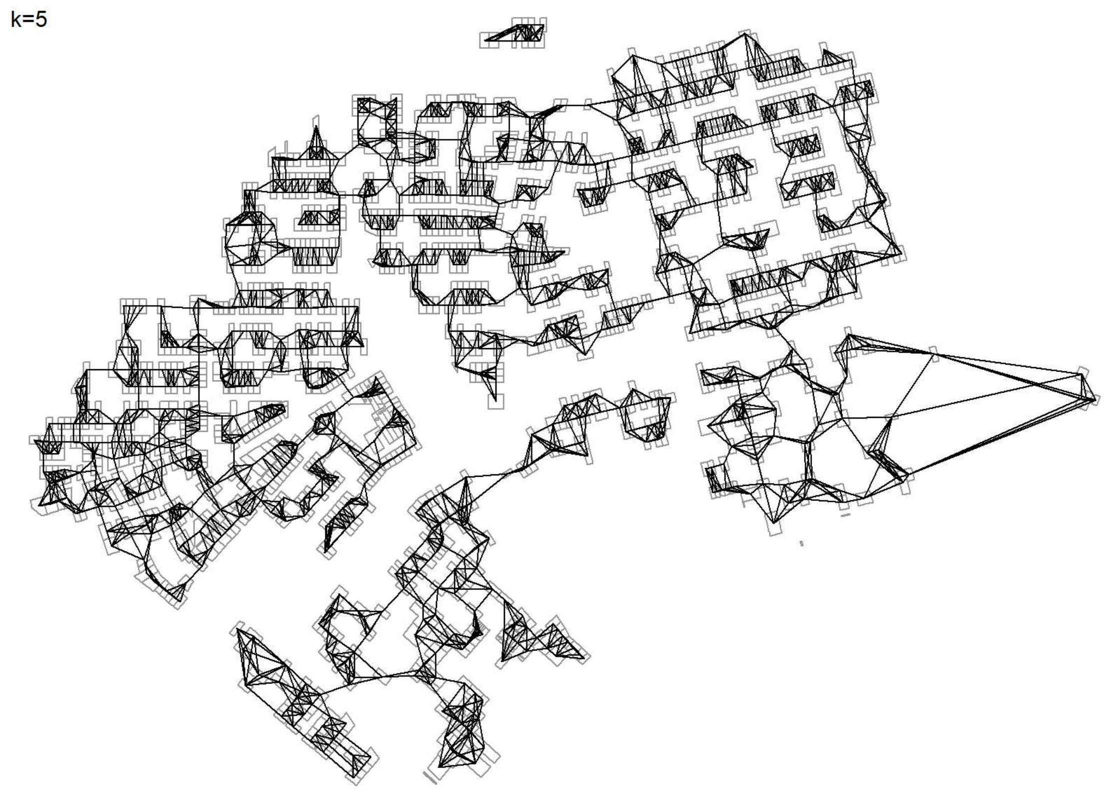
Now that we know there is some clustering happening, it could be of interest to know where these clusters are, and if high sales price values are clutered to other high sales values **or** if low sales values are clustered to other low sales values. Lets try this out, but with a smaller definition for a “neighbor” (k=10 and k=5).

```
knear10 <- knn2nb(knearneigh(coordinates(riverside), k = 10))
```

```
plot(riverside, border="grey60")
plot(knear10, riverside, add=TRUE, pch=".")
text(bbox(riverside)[1,1], bbox(riverside)[2,2], labels="k=10", cex=1)
```




```
knear5 <- knn2nb(knearneigh(coordinates(riverside), k = 5))
plot(riverside, border="grey60")
plot(knear5, riverside, add=TRUE, pch=".")
text(bbox(riverside)[1,1], bbox(riverside)[2,2], labels="k=5", cex=1)
```



Comparing the two plots, when k = 5, we have some groups of polygons that are not linked to others. Because this is one neighborhood, I would prefer not to have isolated groups of polygons. So we will move forward with k = 10.

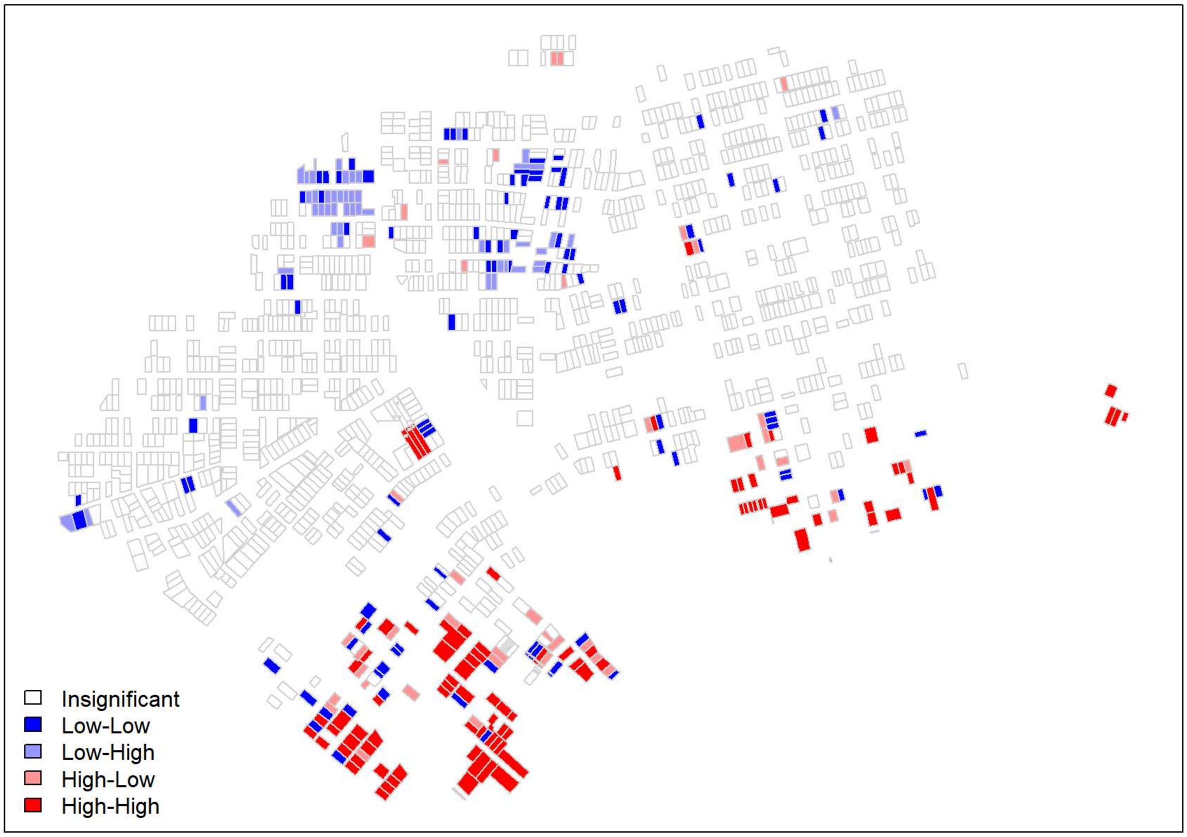
```
weights10 <- nb2listw(knear10)
local_m <- localmoran(riverside$inf_price, weights10)
head(local_m, n = 5)
```

```
##             Ii          E.Ii      Var.Ii        Z.Ii Pr(z != E(Ii))
## 1  0.200331100 -1.724005e-04 0.023183635  1.31683448      0.1878941
## 2 -0.021593130 -1.013559e-04 0.013630854 -0.18408175      0.8539493
## 3  0.183637753 -2.489530e-04 0.033475515  1.00504776      0.3148739
## 4 -0.003675363 -4.253279e-05 0.005720360 -0.04803231      0.9616905
## 5  0.023392174 -9.840329e-06 0.001323498  0.64326746      0.5200506
```

The output returns local I, expected I, variance, Z statistic, and p value. With this output we can create our LISA clusters. We will relax the significance to 0.10 (as opposed to the 0.05 convention)

```
m.price <- riverside$inf_price - mean(riverside$inf_price)
m.local <- local_m[,1] - mean(local_m[,1])

quadrant[m.price < 0 & m.local < 0] <- 1
quadrant[m.price < 0 & m.local > 0] <- 2
quadrant[m.price > 0 & m.local < 0] <- 3
quadrant[m.price > 0 & m.local > 0] <- 4
quadrant[local_m[,5] > signif] <- 0
```



Now we can visualize where the clusters are happening. Remember the global statistic with k = 15 revealed that some clustering was happening, but it was not a strong indicator (0.2618). We echo this with the plot above, where we see some clustering happening. More so in the High-High, indicating high sales prices are surrounded by high sales prices.