R Module 9

Harrison Brown

2021-11-10

Contents

Data Setup

```
# Reproject the shapefile into UTM, a projected CRS
NC_utm <- st_transform(NC, st_crs(26917))
# Find centroids of each polygon
centroids <- NC_utm %>%
  st_geometry() %>%
  st_centroid()
```

```
# Calculate neighbors using Queen's rule
NC_queen <- poly2nb(pl = NC_utm,
                    row.names = NC_utm$NAME,
                    queen = TRUE)
NC_rook <- poly2nb(pl = NC_utm,
                   row.names = NC_utm$NAME,
                   queen = FALSE)
NC_k4 <- knn2nb(knearneigh(centroids, k = 4))
# Calculate max distance for dnearneigh
max.dist <- knn2nb(knearneigh(centroids, k = 1)) %>%
  nbdists(coords = centroids) %>%
  unlist() %>% max()
NC_d100 <- dnearneigh(x = centroids, d1 = 0, d2 = max.dist)
# Convert to weights list object
NC_queen_listw <- nb2listw(NC_queen)
NC_rook_listw <- nb2listw(NC_rook)
NC_k4_listw <- nb2listw(NC_k4)
NC_d100_listw <- nb2listw(NC_d100)
```

Analysis

```r
moran_MNEM2000 <- moran.test(
  x = NC_utm$MNEM2000,
  listw = NC_queen_listw,
  alternative = "two.sided",
  zero.policy = TRUE
)
```

Measuring Moran's I for Manufacturing Jobs gives a statistic of **0.3884704** and a p-value of $3.2387706 \times 10^{-10}$, indicating that there is somewhat strong positive spatial autocorrelation.

1. Moran's I for Numeric Variables in NC

The variables in this analysis are `MNEM1990`, `MNEM2000`, `TOTJOB1990`, and `TOTJOB2000`.

```r
vars <- list(MNEM1990 = "MNEM1990",
             MNEM2000 = "MNEM2000",
             TOTJOB1990 = "TOTJOB1990",
             TOTJOB2000 = "TOTJOB2000")
moran.list <- lapply(vars, function(x) {
  data <- NC_utm %>% dplyr::pull(x)
  moran.test(
    x = data,
    listw = NC_queen_listw,
    zero.policy = T,
    alternative = 'two.sided'
  )
})
```

```r
list.statistics <- data.frame(matrix(ncol = 0, nrow = 4))

list.statistics <- list.statistics %>%
  mutate(
    Rule = names(vars),
    `Moran's I` = sapply(moran.list, function(x)
      {x$estimate["Moran I statistic"]}),
    `p-value` = sapply(moran.list, function(x)
      {x$p.value}
  ))
```

Rule	Moran's I	p-value
MNEM1990	0.4283699	0.0000000
MNEM2000	0.3884704	0.0000000
TOTJOB1990	0.1642235	0.0042649
TOTJOB2000	0.1631958	0.0035879

2. Moran's I for different Neighbor Rules

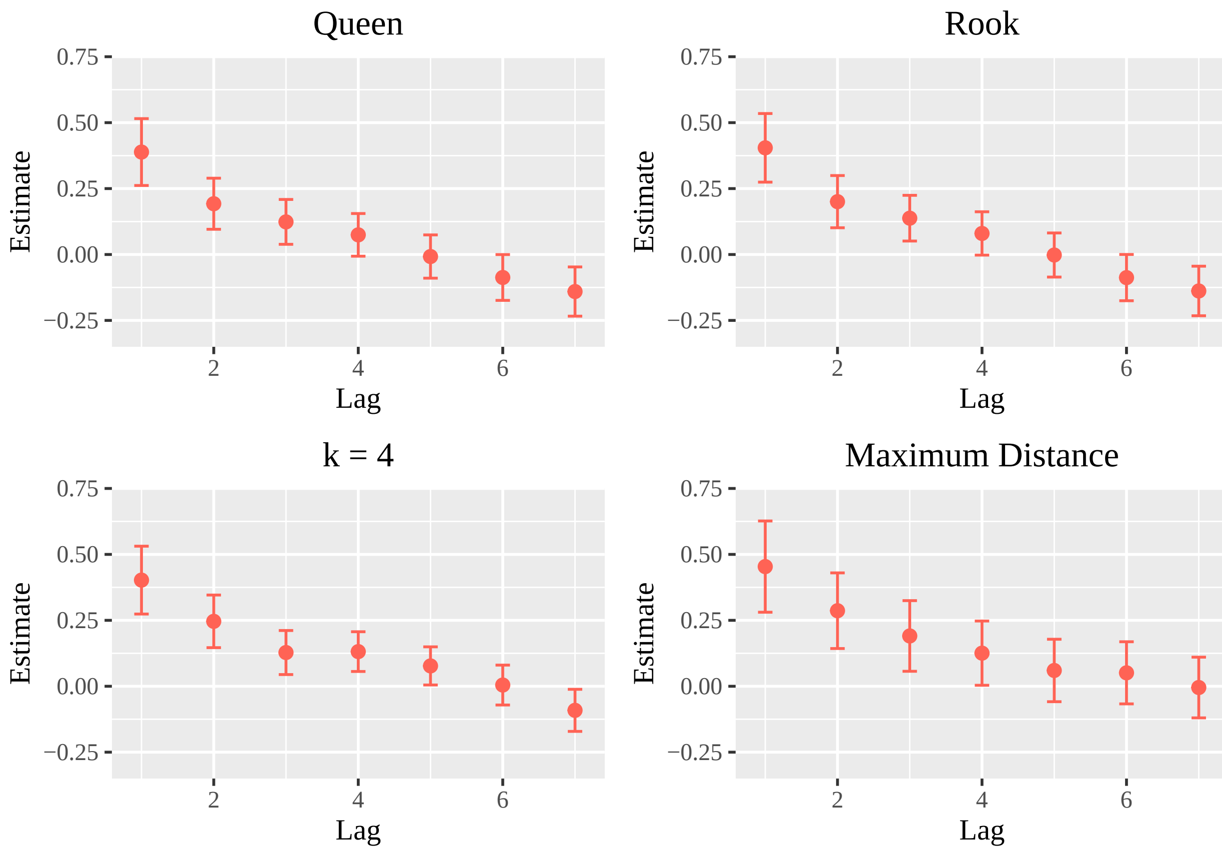
```
w_list <- list(
  Queen = NC_queen_listw,
  Rook = NC_rook_listw,
  `k = 4` = NC_k4_listw,
  `Maximum Distance` = NC_d100_listw
)
moran_diff_w <- lapply(w_list, function(w) {
  moran.test(
    x = NC_utm$MNEM2000,
    listw = w,
    alternative = "two.sided",
    zero.policy = TRUE
  )
})

diff_w_stats <- data.frame(matrix(ncol = 0, nrow = 4))
diff_w_stats <- diff_w_stats %>%
  mutate(
    Rule = names(w_list),
    `Moran's I` = sapply(moran_diff_w, function(x) {
      x$estimate["Moran I statistic"]
    }),
    `p-value` = sapply(moran_diff_w, function(x) {
      x$p.value
    }),
    `Avg. Neighbors` = sapply(w_list, function(x) {
      y <- x$neighbours
      len <- sapply(y, length)
      mean(len)
    })
  )
```

Rule	Moran's I	p-value	Avg. Neighbors
Queen	0.3884704	0e+00	4.90
Rook	0.4044841	0e+00	4.62
k = 4	0.4024844	0e+00	4.00
Maximum Distance	0.4536758	1e-07	2.80

I found it interesting that the Moran's I for the Rook and k = 4 cases were so similar; this might be because Rook searches for neighbors in four directions.

Spatial Correlogram Plots



The general spatial pattern with MNEM2000 is that counties become less similar the further out, but around order ~ 4, there is a small "bump", showing that clusters have a significant distance between clusters. Interestingly, the estimate drops below 0 for most of the models, indicating there is a slight negative spatial autocorrelation at large scales (lags) – this could represent clusters of manufacturing jobs being surrounded by areas with far fewer jobs, and shows how industrial centers "clump" together. From a business standpoint, it would not make sense to build factories very close together, but rather provide access to a greater geographic area.

4. LISA

```
localMoran.queen <- localmoran(NC_utm$MNEM2000, NC_queen_listw)

sig <- c(.5, .1, .05, .01)


NC_utm <- NC_utm %>%
  mutate(Significance = localMoran.queen[,5],
         Quadrant_name = attr(localMoran.queen, "quadr")$mean,
         Quadrant = data.table::fifelse((Significance > sig),
                                       "Insignificant",
                                       Quadrant_name %>% as.character()))
```