Spatial Exploratory Data Analysis: Social Vulnerability Index

Thesis for a Master of Public Health, Epidemiology

Nathan Garcia-Diaz

Brown University, School of Public Health

August 02, 2024

Purpose Statement

This fill acts as an extension to the `Exploratory Data Analysis: Social Vulnerability Index` by examining the spatial components. Given the nature of the data, an examination of the spatial components is required. Specifically, this document will pull from two sources to help with the writing of the code: Ch 7.6 and 7.7 in Analyzing US Census Data: Methods, Maps, and Models in R (link) and Manny Gimond's A basic introduction to Moran's I analysis in R (link).

The examination of the spatial components include: Moran's I calculation with Monte Carlos Simulation, Moran's Scatterplot, Local Spatial Autocorrelation with Getis-Ord local G_i^*, and Hot/Cold Spot Identification.

Definitions:

- **Moran's I**: Moran's I is a measure of spatial autocorrelation, quantifying the degree to which a variable is similarly distributed across neighboring geographic areas. It ranges from -1 (indicating perfect dispersion) to +1 (indicating perfect clustering), with values around 0 suggesting a random spatial pattern. It is used to detect and measure the presence of spatial autocorrelation, helping analysts understand whether the spatial distribution of a variable is clustered, dispersed, or random.
- **Monte Carlo Simulations**: Monte Carlo simulation is a computational technique that uses repeated random sampling to estimate the statistical properties of a system. It is used in tandem with Moran's I calculation to assess the significance of observed spatial autocorrelation by comparing it to the distribution of Moran's I values generated under the null hypothesis of spatial randomness. This is preformed as suggested by Gimond.
- **Moran's Scatter Plot**: Moran's scatterplot is a graphical representation that illustrates the relationship between a variable's values and the spatially lagged values of the same variable, used to visualize spatial autocorrelation. The plot typically includes a 45-degree reference line and divides the data points into four quadrants to help identify patterns of clustering or dispersion. It is used to diagnose and visualize spatial autocorrelation, helping to identify patterns of spatial clustering or dispersion in a dataset.
- **Local Spatial Autocorrelation**: Local measures of spatial autocorrelation, like the Getis-Ord local G_i^*, are used to identify clusters or "hot spots" of similar values within a spatial dataset. The Getis-Ord local G_i^* statistic specifically measures the degree of clustering of high or low values around each point, indicating areas with significant local spatial association.
  - *Positive Gi Values*: indicate areas where high values of rpl_themes are surrounded by other high values, or low values are surrounded by other low values. This suggests clustering of similar values.
  - *Negative Gi Values*: Indicate areas where high values of rpl_themes are surrounded by low values, or vice versa. This suggests spatial outliers or contrast.

Preparation

```
### importing packages
# define desired packages
library(tidyverse)  # general data manipulation
library(knitr)      # Rmarkdown interactions
library(here)       # define top level of project folder
                        # this allows for specification of where
                        # things live in relation to the top level
# spatial tasks
library(tigris)     # obtain shp files
library(spdep)      # exploratory spatial data analysis


### loading data
svi_df = read_csv(here::here("01_Data", "svi_df.csv")) %>%
  mutate(fips = as.character(fips))

## obtaining SPH files for RI tracts
tracts = tracts(state = "RI", year = 2022, cb = TRUE)

## joining data
svi_map = inner_join(tracts, svi_df, by = c("GEOID" = "fips"))

## define neighbors
nb_list = poly2nb(svi_map, queen=TRUE)
summary(nb_list)  # check neighbors
```

```
## Neighbour list object:
## Number of regions: 246
## Number of nonzero links: 1336
## Percentage nonzero weights: 2.207681
## Average number of links: 5.430894
## 2 regions with no links:
## 84 233
## 4 disjoint connected subgraphs
## Link number distribution:
##
##  0  1  2  3  4  5  6  7  8  9 10 11 12
##  2  1  5 25 44 62 42 32 21  4  5  2  1
## 1 least connected region:
## 85 with 1 link
## 1 most connected region:
## 211 with 12 links
## assign weights
weights = nb2listw(nb_list, style="W", zero.policy=TRUE)
summary(weights)   # check weights
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 246
## Number of nonzero links: 1336
## Percentage nonzero weights: 2.207681
## Average number of links: 5.430894
## 2 regions with no links:
## 84 233
## 4 disjoint connected subgraphs
## Link number distribution:
##
##  0  1  2  3  4  5  6  7  8  9 10 11 12
##  2  1  5 25 44 62 42 32 21  4  5  2  1
## 1 least connected region:
## 85 with 1 link
## 1 most connected region:
## 211 with 12 links
##
## Weights style: W
## Weights constants summary:
##     n    nn  S0       S1       S2
## W 244 59536 244 96.64698 1001.215
```

Morans I Calculation and Scatter Plot

At $\alpha = 0.05$, all variables are statistically significant. Variables that had a statistic close to zero include the estimated number of unemployed individuals (16 yrs +) (`e_unemp`) and the estimated number of persons in group quarters (`e_groupq`). Meanwhile, the variables with the top 3 highest moran's i value include the number of minorities (`e_minrty`), the number of individuals who self report having limited english (`e_limeng`), and the SVI Summary Values (`rpl_themes`). All other statistically significant variables demonstrated weaker presence of spatial autocorrelation, and the only variable to be statistically insignificant (i.e., demonstrate no spatial autocorrelation) is the proportion of individuals who are unemployed (`e_unemp`).

Additionally, a moran's scatterplot was only preformed for the outcome of interest since creating graphs for all variables would provide redundant information. However, these graphs can be made available upon request. In support with the Moran's I calculation with Monte Carlo Simulations, the scatterplot suggests a positive correlation between the SVI Summary Value and its spatial lag, representative of spatial autocorrelation in the data.

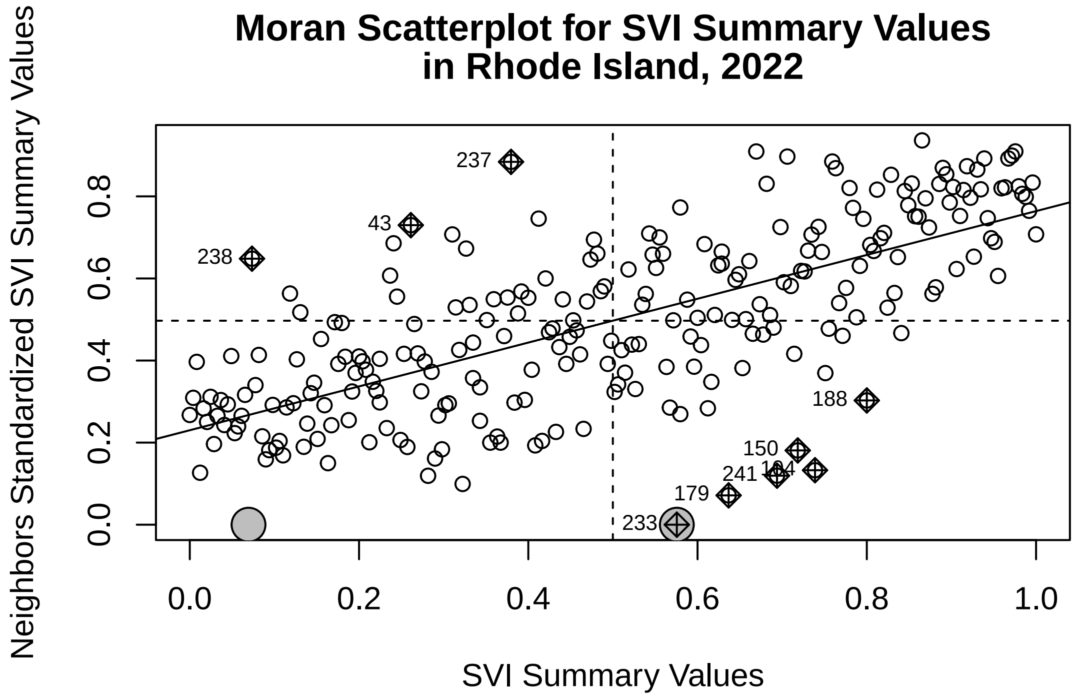
```r
## calculating test statistic - moran's i via monte carlo simulation
columns = as.data.frame(svi_map) %>%
  select(rpl_themes, starts_with("e_")) %>%
  colnames()

moran_df = tibble(NULL)

## creates calculations for all variables
for (col in columns){
  # calculate the moran's object
  res = moran.mc(pull(svi_map, !!col),
                 listw = weights,
                 nsim = 999,
                 zero.policy = TRUE)

  # use the res object to create a new row in moran_df
  moran_df = moran_df %>%
    bind_rows(tibble(variable = col,
                     statistic = res$statistic,
                     pvalue = res$p.value))
}

# creating moran scatter plot
moran.plot(x = svi_map$rpl_themes,
           listw = weights,
           xlab = "SVI Summary Values",
           ylab = "Neighbors Standardized SVI Summary Values",
           main = c("Moran Scatterplot for SVI Summary Values",
                    "in Rhode Island, 2022"))
```



```r
# creating summary table for moran's i values for all variables
moran_df = moran_df %>% arrange(desc(statistic))
knitr::kable(moran_df, caption = c("Moran's I with Monte Carlo Simulations"),
             digits = c(0, 3, 3))
```

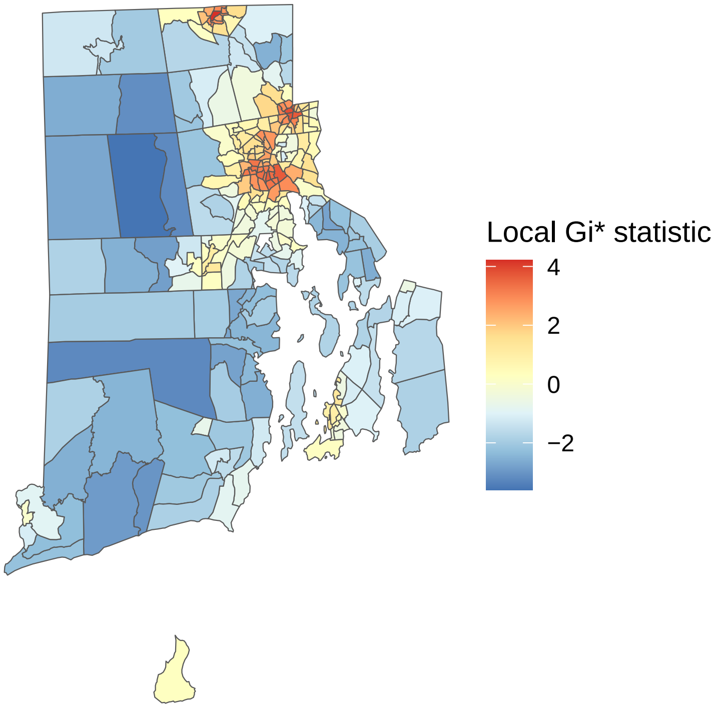
Table 1: Moran's I with Monte Carlo Simulations

variable	statistic	pvalue
e_minrty	0.586	0.001
e_limeng	0.579	0.001
rpl_themes	0.525	0.001
e_nohsdp	0.400	0.001
e_pov150	0.346	0.001
e_uninsur	0.316	0.001
e_noveh	0.287	0.001
e_age65	0.279	0.001
e_crowd	0.228	0.001
e_age17	0.203	0.001
e_hburd	0.193	0.001
e_munit	0.180	0.001
e_mobile	0.156	0.002
e_sngpnt	0.142	0.001
e_disabl	0.126	0.001
e_groupq	0.069	0.042
e_unemp	0.057	0.062

Local Spatial Autocorrelation with G_i^* & Hot/Cold Spot Identification

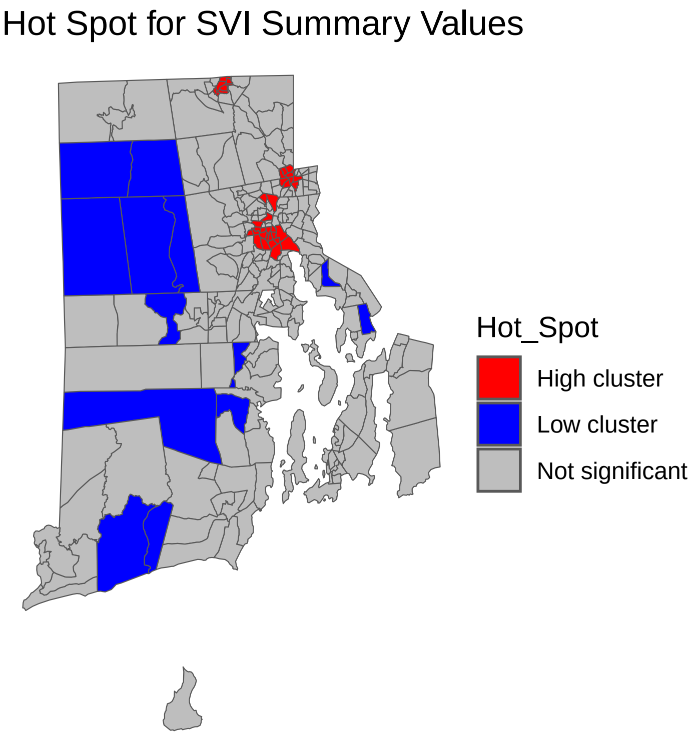
```
# For Gi*, re-compute the weights with `include.self()`
localg_weights = nb2listw(include.self(nb_list), style="W", zero.policy=TRUE)
svi_map = svi_map %>%
  mutate(localG = as.numeric(localG(svi_map$rpl_themes, localg_weights)))

# Local spatial autocorrelation
ggplot(svi_map, aes(fill = localG)) +
  geom_sf() +
  scale_fill_distiller(palette = "RdYlBu") +
  theme_void() +
  labs(fill = "Local Gi* statistic")
```



```r
# hotspot identification
svi_map = svi_map %>%
  mutate(Hot_Spot = case_when(
    localG >= 2.576 ~ "High cluster",
    localG <= -2.576 ~ "Low cluster",
    TRUE ~ "Not significant"
  ))

ggplot(svi_map, aes(fill = Hot_Spot), color = "grey90", size = 0.001) +
  geom_sf() +
  scale_fill_manual(values = c("red", "blue", "grey")) +
  theme_void() +
  labs(title = "Hot Spot for SVI Summary Values")
```


Hot Spot for SVI Summary Values
Hot_Spot
🟥 High cluster
🟦 Low cluster
⬜ Not significant