Spatial Autocorrelation and Areal Data

HES 505 Fall 2023: Session 21

Matt Williamson

Objectives

By the end of today you should be able to:

- Use the `spdep` package to identify the neighbors of a given polygon based on proximity, distance, and minimum number
- Understand the underlying mechanics of Moran's I and calculate it for various neighbors
- Distinguish between global and local measures of spatial autocorrelation
- Visualize neighbors and clusters

Revisiting Spatial Autocorrelation

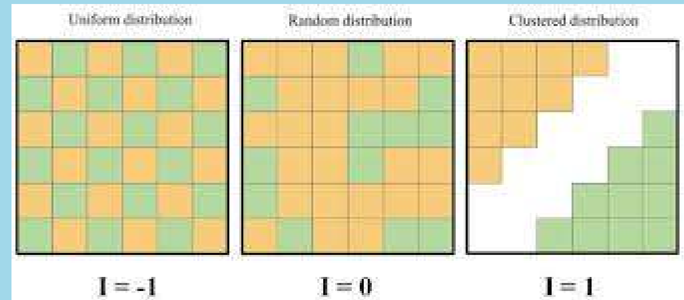
Spatial Autocorrelation

- Attributes (features) are often non-randomly distributed
- Especially true with aggregated data
- Interest is in the relationship between proximity and the feature
- Difference from kriging and semivariance

Moran’s I

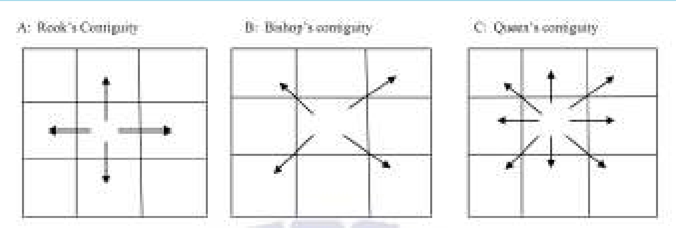
- Moran’s I

$$I(d) = \frac{\sum_{i}^{n} \sum_{j \neq i}^{n} w_{ij} (x_i - \bar{x})(x_j - \bar{x})}{S^2 \sum_{i}^{n} \sum_{j \neq i}^{n} w_{ij}}$$



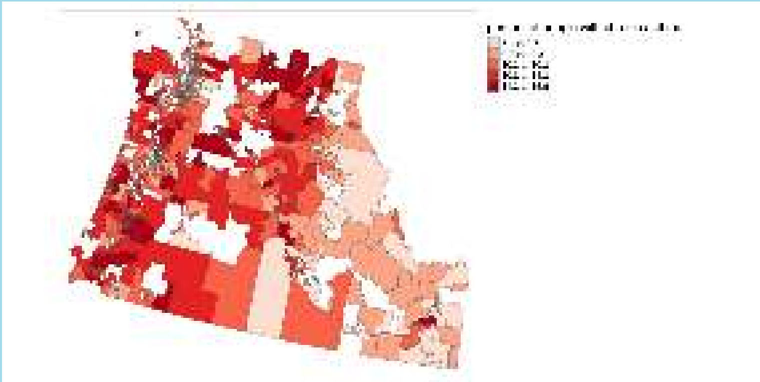
Finding Neighbors

- How do we define I(d) for areal data?
- What about w_{ij}?
- We can use `spdep` for that!!



Using spdep

```
cdc <- read_sf("data/opt/data/2023/vectore
  select(stateabbr, countyname, countyfips
```



Finding Neighbors

- Queen, rook, (and bishop) cases impose neighbors by contiguity
- Weights calculated as a 1/num. of neighbors

```r
nb.qn <- poly2nb(cdc, queen=TRUE)
nb.rk <- poly2nb(cdc, queen=FALSE)
```

Finding Neighbors

Getting Weights

::: columns ::: {.column width="60%"}

```
lw.qn <- nb2listw(nb.qn, style="W", zero.policy=TRUE)
lw.qn$weights[1:5]
```

```
[[1]]
[1] 0.5 0.5

[[2]]
[1] 0.25 0.25 0.25 0.25

[[3]]
[1] 0.2 0.2 0.2 0.2 0.2

[[4]]
[1] 0.3333333 0.3333333 0.3333333

[[5]]
[1] 1
```

```
asthma.lag <- lag.listw(lw.qn, cdc$casthma_cr)
```

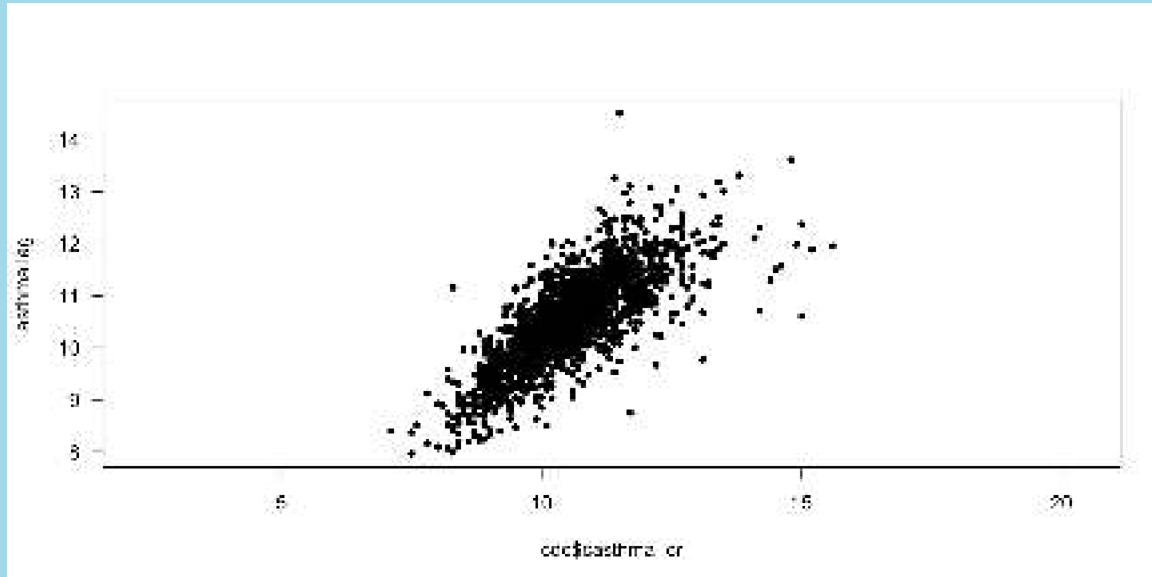
::: :::{.column width="40%"}

```
                        asthma.lag
[1,] "Camas"       "9.9"  "10.3"
[2,] "Kootenai"    "10.4" "9.575"
[3,] "Kootenai"    "10"   "9.88"
[4,] "Kootenai"    "9.5"  "10.2666666666667"
[5,] "Twin Falls"  "10.2" "9.5"
[6,] "Twin Falls"  "10.4" "9.9"
```

Fit a model

- Moran's I is just the slope of the regression!

```
M <- lm(asthma.lag ~ cdc$c
```




```
cdc$casthma_cr
     0.6467989
```

Comparing observed to expected

```r
n <- 400L   # Define the number of simulat
I.r <- vector(length=n)  # Create an empty

for (i in 1:n){
  # Randomly shuffle income values
  x <- sample(cdc$casthma_cr, replace=FALS
  # Compute new set of lagged values
  x.lag <- lag.listw(lw.qn, x)
  # Compute the regression slope and store
  M.r    <- lm(x.lag ~ x)
  I.r[i] <- coef(M.r)[2]
}
```