



UNIVERSIDAD NACIONAL DE COLOMBIA

# Modelado de datos en áreas

Estadística espacial

**Daniela Arbeláez Montoya**  
**Jefferson Gamboa Betancur**  
**Jean Paul Piedrahita García**

Universidad Nacional de Colombia  
Ciencias, Escuela de Estadística  
Medellín, Colombia  
2021

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Vecinos espaciales y peso espacial</b>	<b>6</b>
2.1. Objetos vecinos . . . . .	6
2.2. Objetos de ponderaciones espaciales . . . . .	8
2.3. Manejo de objetos de ponderaciones espaciales . . . . .	9
2.4. Uso de pesos para simular la autocorrelación espacial . . . . .	9
<b>3. Prueba de autocorrelación espacial</b>	<b>9</b>
3.1. Pruebas globales . . . . .	9
3.2. Pruebas locales . . . . .	10
<b>4. Ajuste de modelos de datos de área</b>	<b>10</b>
4.1. Enfoques de estadística espacial . . . . .	10
4.1.1. Modelos autorregresivos simultáneos . . . . .	10
4.1.2. Modelos autorregresivos condicionales . . . . .	10
4.1.3. Ajuste de modelos de regresión espacial . . . . .	10

# 1. Introducción

A lo largo del desarrollo de este documento, se mostrará la construcción de vecinos y los pesos que se pueden aplicar a los vecindarios. Una vez que este importante y a menudo exigente prerequisite esté en su lugar, se procede a buscar formas de medir la autocorrelación espacial.

Si bien las pruebas se basan en modelos de procesos espaciales, primero se examinan las pruebas y solo posteriormente se pasa al modelado. También es interesante mostrar cómo se puede introducir la autocorrelación espacial en datos independientes, de modo que se puedan realizar simulaciones.

El conjunto de datos con el que se trabajará en esta ocasión, contiene 281 distritos censales para ocho condados centrales del estado de Nueva York complementado con límites de tramo. El área tiene una extensión de unos 160 km de norte a sur y de 120 km de este a oeste.

```
> library(rgdal)
> library(sf)
> library(spdep)
>
> NY8 <- readOGR("Base de datos", "NY8_utm18")

## OGR data source with driver: ESRI Shapefile
## Source: "G:\Mi unidad\Universidad Nacional de Colombia\13. Treceavo semestre\Estadística\NY8_utm18.shp"
## with 281 features
## It has 17 fields

> TCE <- readOGR("Base de datos", "TCE")

## OGR data source with driver: ESRI Shapefile
## Source: "G:\Mi unidad\Universidad Nacional de Colombia\13. Treceavo semestre\Estadística\TCE.shp"
## with 11 features
## It has 5 fields

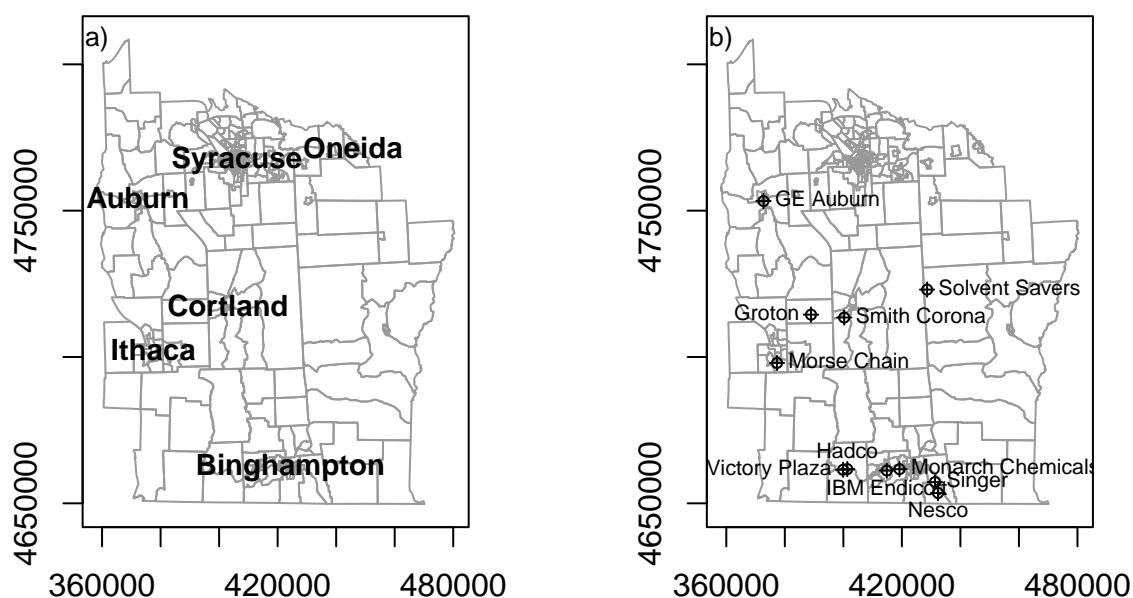
> cities <- readOGR("Base de datos", "NY8cities")

## OGR data source with driver: ESRI Shapefile
## Source: "G:\Mi unidad\Universidad Nacional de Colombia\13. Treceavo semestre\Estadística\NY8cities.shp"
## with 6 features
## It has 1 fields
```

```

> par(mfrow=c(1,2))
> plot(NY8, border="grey60", axes=TRUE)
> text(coordinates(cities), labels=as.character(cities$names), font=2, cex=0.9)
> text(bbox(NY8)[1,1], bbox(NY8)[2,2], labels="a)", cex=0.8)
> plot(NY8, border="grey60", axes=TRUE)
> points(TCE, pch=1, cex=0.7)
> points(TCE, pch=3, cex=0.7)
> text(coordinates(TCE), labels=as.character(TCE$name), cex=0.7,
+ font=1, pos=c(4,1,4,1,4,4,4,2,3,4,2), offset=0.3)
> text(bbox(NY8)[1,1], bbox(NY8)[2,2], labels="b)", cex=0.8)

```

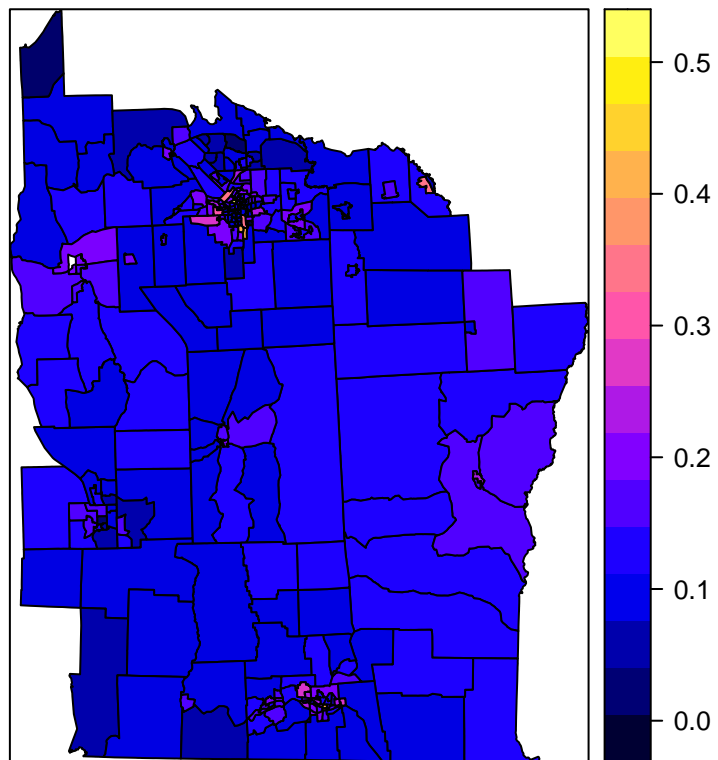


La figura a) muestra las principales ciudades en el área de estudio y b) la ubicación de 11 sitios de desechos peligrosos.

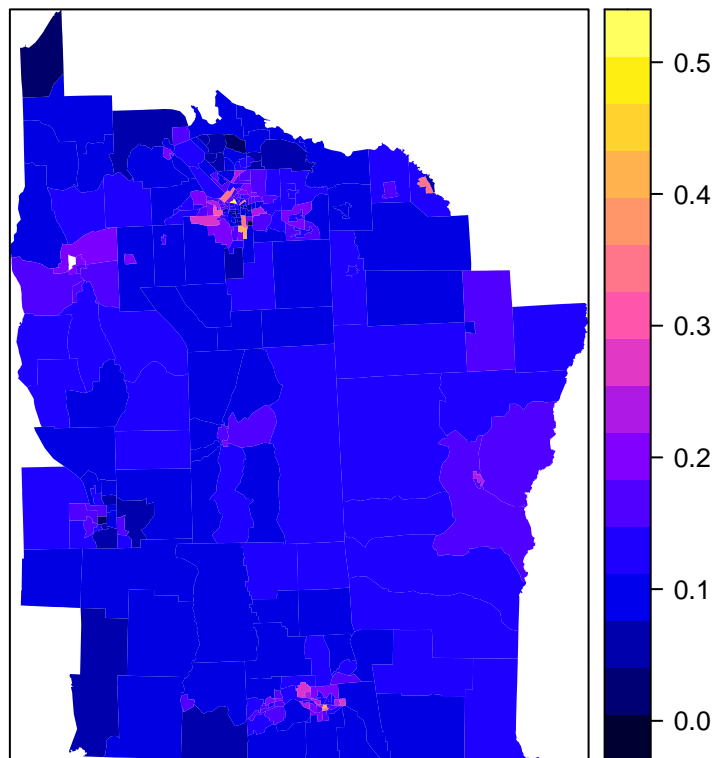
```

> spplot(NY8, c("PCTAGE65P"))

```



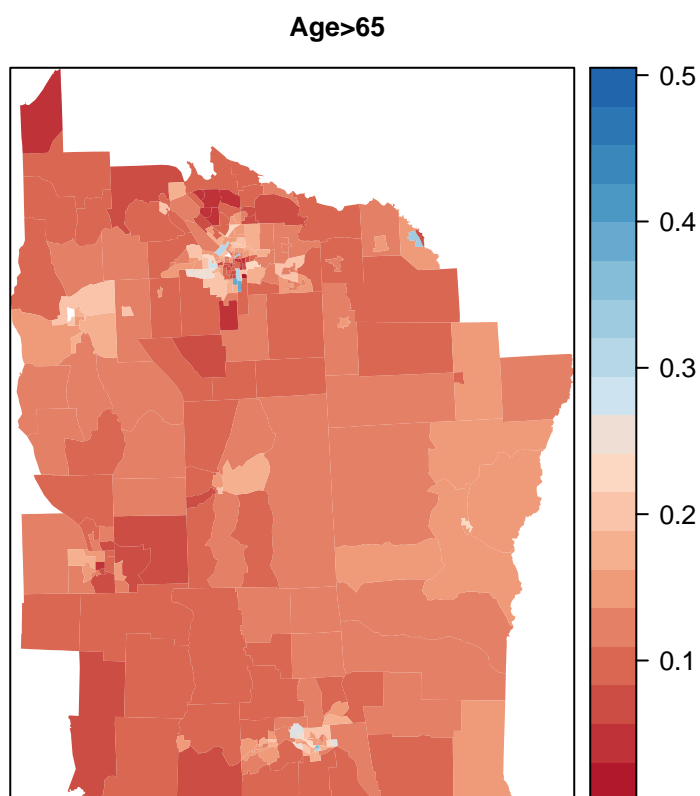
```
> spplot(NY8, c("PCTAGE65P"), col="transparent")
```



```

> library("RColorBrewer")
> #color palette creator function
> rds <- colorRampPalette(brewer.pal(8, "RdBu"))
> #get a range for the values
> tr_at <- seq(min(NY8$PCTAGE65P), max(NY8$PCTAGE65P), length.out=20)
> #create a color interpolating function taking the required
> #number of shades as argument
> tr_rds <- rds(20)
> #parameters
> # at - at which values colors change
> # col.regions - specify fill colors
> tr_pl <- spplot(NY8, c("PCTAGE65P"), at=tr_at, col="transparent",
+               col.regions=tr_rds, main=list(label="Age>65", cex=0.8))
> plot(tr_pl)

```



## 2. Vecinos espaciales y peso espacial

La creación de ponderaciones espaciales es un paso necesario en el uso de datos de área, quizás solo para verificar que no haya patrones espaciales restantes en los residuos. El primer paso es definir a qué relaciones entre observaciones se les dará un peso distinto de cero, es decir, elegir el criterio de vecino que se usará; el segundo es asignar pesos a los enlaces vecinos identificados.

Tratar de detectar patrones en mapas de residuos visualmente no es una opción aceptable, por lo que se incluyen un montón de funciones en el paquete **spdep** para ayudar.

Las viñetas “**nb**”, “**CO69**” y “**sids**” en **spdep** incluyen discusiones sobre la creación y el uso de ponderaciones espaciales, y se pueden acceder a ellas de la siguiente manera:

```
> vignette("nb", package = "spdep")
```

### 2.1. Objetos vecinos

En el paquete **spdep**, las relaciones vecinas entre **n** observaciones están representadas por un objeto de clase **nb**. Es una lista de longitud **n** con los números de índice de vecinos de cada componente registrados como un vector entero. Si alguna observación no tiene vecinos, el componente contiene un número entero cero. También contiene atributos, típicamente un vector de identificadores de región de caracteres y un valor lógico que indica si las relaciones son simétricas. Los identificadores de región pueden usarse para verificar la integridad entre los datos mismos y el objeto vecino.

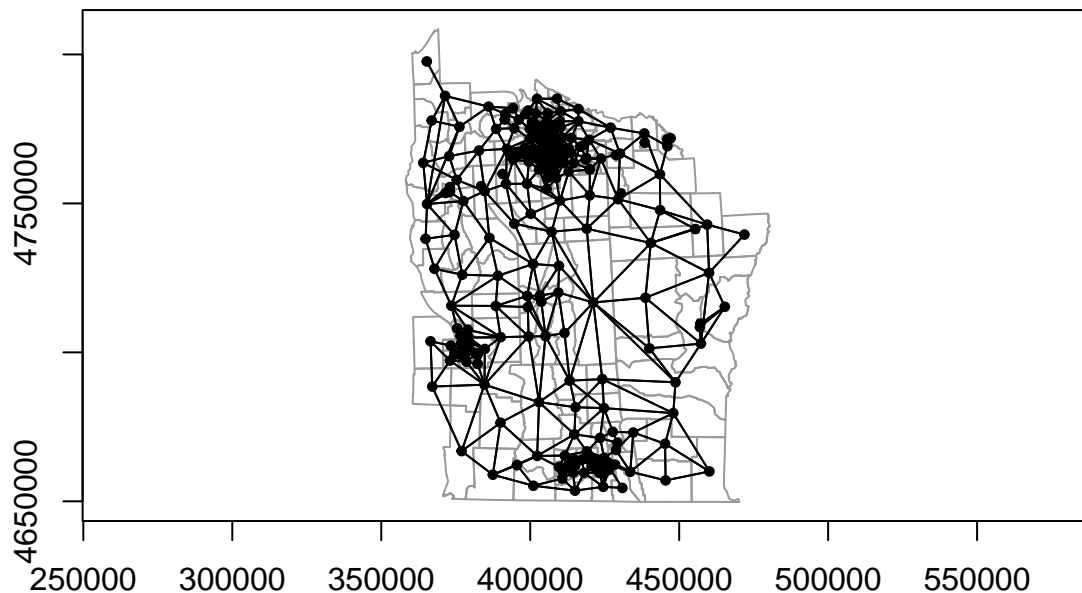
La función auxiliar **card** devuelve la cardinalidad del conjunto de vecinos para cada objeto, es decir, el número de vecinos.

```
> # reads a GAL lattice file into a neighbors list
> NY_nb <- read.gal("Base de datos/NY_nb.gal", region.id = row.names(NY8))
> summary(NY_nb)
```

```
## Neighbour list object:
## Number of regions: 281
## Number of nonzero links: 1522
## Percentage nonzero weights: 1.927534
## Average number of links: 5.41637
## Link number distribution:
##
```

```
## 1 2 3 4 5 6 7 8 9 10 11
## 6 11 28 45 59 49 45 23 10 3 2
## 6 least connected regions:
## 55 97 100 101 244 245 with 1 link
## 2 most connected regions:
## 34 82 with 11 links

> par(mfrow=c(1,1))
> plot(NY8, border="grey60", axes=TRUE)
> plot(NY_nb, coordinates(NY8), pch=19, cex=0.6, add=TRUE)
```



La figura muestra el gráfico vecino completo para el área de estudio de ocho condados.

Como ahora se tiene un objeto nb para examinar, se pueden presentar los métodos estándar para estos objetos. Hay métodos de impresión, resumen, diagrama y otros; el método de resumen presenta una tabla de la distribución del número de enlace, y tanto el método de impresión como el de resumen informan de la asimetría y la presencia de observaciones sin vecinos; la asimetría está presente cuando  $i$  es un vecino de  $j$  pero  $j$  no es un vecino de  $i$ .

Con motivos de simplicidad al mostrar como crear objetos vecinos, se trabaja con un subconjunto



del mapa que consta de los censos dentro de Syracuse, aunque los mismos principios se aplican al conjunto de datos completo.

```
> Syracuse <- NY8[NY8$AREANAME == "Syracuse city",]  
> Sy0_nb <- subset(NY_nb, NY8$AREANAME == "Syracuse city")  
> summary(Sy0_nb)
```

```
## Neighbour list object:  
## Number of regions: 63  
## Number of nonzero links: 346  
## Percentage nonzero weights: 8.717561  
## Average number of links: 5.492063  
## Link number distribution:  
##  
## 1 2 3 4 5 6 7 8 9  
## 1 1 5 9 14 17 9 6 1  
## 1 least connected region:  
## 164 with 1 link  
## 1 most connected region:  
## 136 with 9 links
```

## 2.2. Objetos de ponderaciones espaciales

```
> Sy0_lw_W <- nb2listw(Sy0_nb); Sy0_lw_W  
  
## Characteristics of weights list object:  
## Neighbour list object:  
## Number of regions: 63  
## Number of nonzero links: 346  
## Percentage nonzero weights: 8.717561  
## Average number of links: 5.492063  
##  
## Weights style: W  
## Weights constants summary:  
##      n   nn S0      S1      S2
```

```
## W 63 3969 63 24.78291 258.564

> names(Sy0_lw_W)

## [1] "style"      "neighbours" "weights"

> names(attributes(Sy0_lw_W))

## [1] "names"      "class"      "region.id" "call"      "GeoDa"

> 1/rev(range(card(Sy0_lw_W$neighbours)))

## [1] 0.1111111 1.0000000

> summary(unlist(Sy0_lw_W$weights))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1111 0.1429 0.1667 0.1821 0.2000 1.0000

> summary(sapply(Sy0_lw_W$weights, sum))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##          1          1          1          1          1          1

> Sy0_lw_B <- nb2listw(Sy0_nb, style = "B")
> summary(unlist(Sy0_lw_B$weights))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##          1          1          1          1          1          1

> summary(sapply(Sy0_lw_B$weights, sum))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1.000 4.500 6.000 5.492 6.500 9.000
```

## 2.3. Manejo de objetos de ponderaciones espaciales

## 2.4. Uso de pesos para simular la autocorrelación espacial

# 3. Prueba de autocorrelación espacial

## 3.1. Pruebas globales

### 3.2. Pruebas locales

## 4. Ajuste de modelos de datos de área

### 4.1. Enfoques de estadística espacial

#### 4.1.1. Modelos autorregresivos simultáneos

#### 4.1.2. Modelos autorregresivos condicionales

#### 4.1.3. Ajuste de modelos de regresión espacial