Lecture 11:
Intro to Spatial Data

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

September 14, 2021

Announcement

- **Problem Set 2 is due next Thursday September 15 at 1:00pm**
- At some point before the class I'll send what points everyone should present. You should be prepared. Your grade will be impacted if you are not ready.
- I expect good presentations. You are encouraged to consider it as a mini-seminar, just 2-5 minutes using one or two slides
- Attempt to make a concise interpretation of the relevant material, making effective use of supporting numerical and graphical evidence.

Agenda

Motivation

- In Big Data volume was only a part of the story
- Big Data are data of high complexity: anarchic and spontaneous
- They are the by product of an action: pay with credit card, tweet, move from point A to point B, buy a house, etc.
- Now we are going to focus on spatial data

Motivation

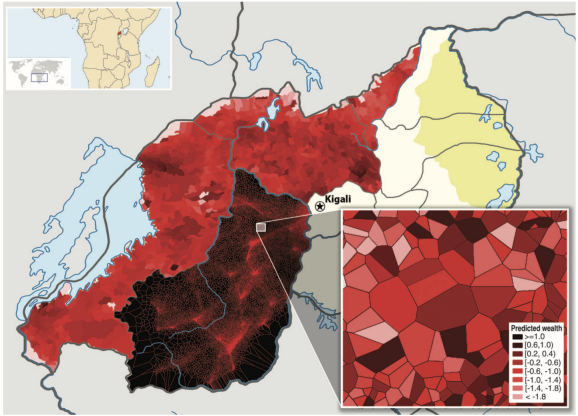
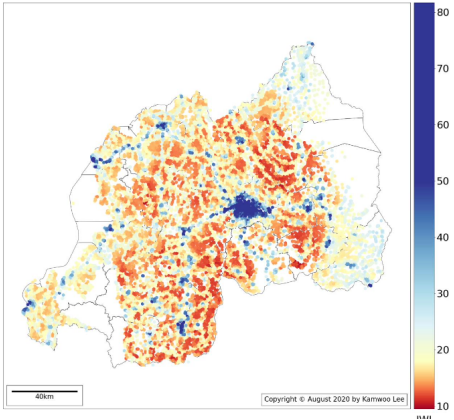


Fig. 2. Construction of high-resolution maps of poverty and wealth from call records. Information derived from the call records of 1.5 million subscribers is overlaid on a map of Rwanda. The northern and western provinces are divided into cells (the smallest administrative unit of the country), and the cell is shaded according to the average (predicted) wealth of all mobile subscribers in that cell. The southern province is overlaid with a Voronoi division that uses geographic identifiers in the call data to segment the region into several hundred thousand small partitions. (**Bottom right inset**) Enlargement of a 1-km^2 region near Kiyonza, with Voronoi cells shaded by the predicted wealth of small groups (5 to 15 subscribers) who live in each region.

Blumenstock et al (2015)

Motivation



Lee, K., & Braithwaite, J. (2020)

Types of Spatial Data

Spatial data comes in many “shapes” and “sizes”, the most common types of spatial data are:

- Points are the most basic form of spatial data. Denotes a single point location, such as cities, a GPS reading, or any other discrete object defined in space.
- Lines are a set of ordered points, connected by straight line segments
- Polygons denote an area, and can be thought as a sequence of connected points, where the first point is the same as the last
- Grid (Raster) are a collection of points or rectangular cells, organized in a regular lattice.

Types of Spatial Data: Points

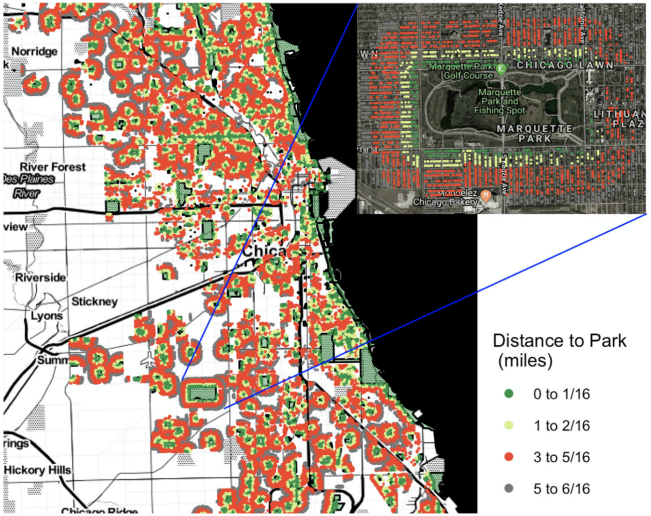


Fig. 1. Housing transactions around parks: neighborhood distance intervals. Notes: The following figure shows transactions within 3/8 miles of the nearest park in Chicago. The

Types of Spatial Data: Lines

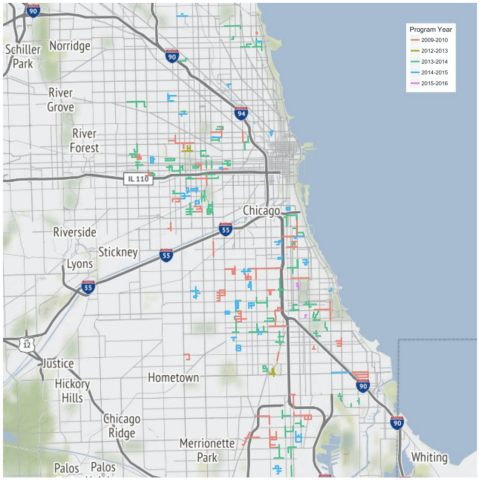
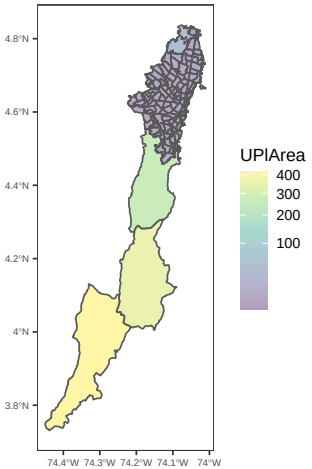


Fig. 1. Safe Passage Routes, by year of program adoption.
Note: Shapefiles with Safe Passage shape and location where obtained from the Chicago Data Portal and year that the program was launched at each location through a FOIA request.

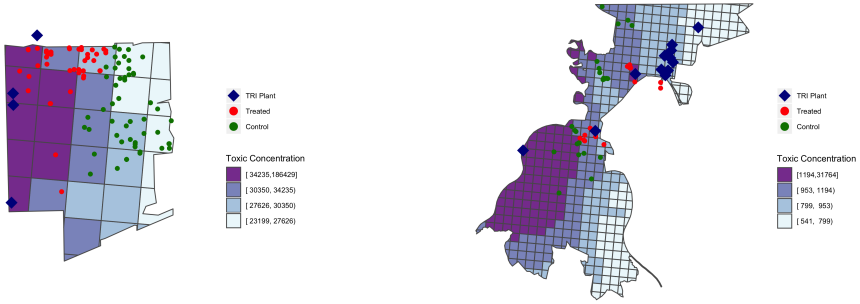
McMillen, Sarmiento-Barbieri & Singh, 2019

Types of Spatial Data: Polygons



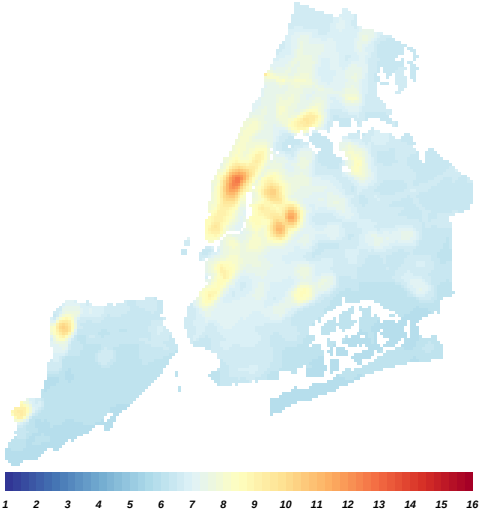
Source:https://datosabiertos.bogota.gov.co/dataset/unidad-de-planeamiento-bogota-d-c

Types of Spatial Data: Combination



Christensen,Sarmiento-Barbieri, & Timmins (2020)

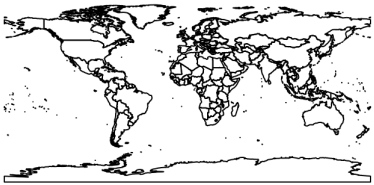
Types of Spatial Data: Rasters

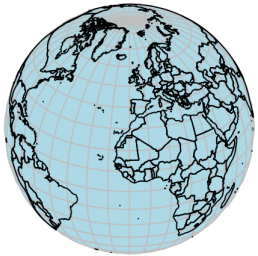


Source:https://data.cityofnewyork.us/Environment/NYCCAS-Air-Pollution-Rasters/q68s-8qxv

The earth ain't flat

- The world is an irregularly shaped ellipsoid, but plotting devices are flat
- But if you want to show it on a flat map you need a map projection
- This will determine how to transform and distort latitudes and longitudes to preserve some of the map properties: area, shape, distance, direction, or bearing





The earth ain't flat

- For example, sailors use Mercator projection where meridians and parallels cross each other always at the same 90 degrees angle.
- It allows to easy locate yourself on the line showing direction in which you sail
- But the projection do not preserve distances

Source: https://www.geoawesomeness.com/all-map-projections-in-compared-and-visualized/

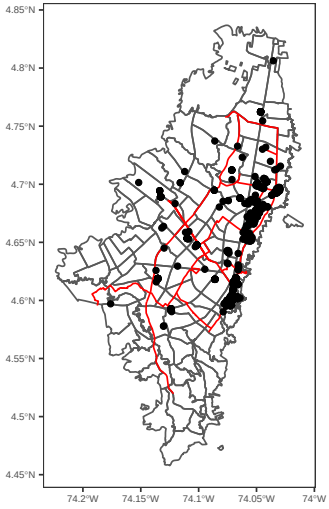
Which projection should I choose?

- "There exist no all-purpose projections, all involve distortion when far from the center of the specified frame" (Bivand, Pebesma, and Gómez-Rubio 2013)
- Geographic coordinate systems: coordinate systems that span the entire globe (e.g. latitude / longitude).
  - For geographic CRSs, the answer is often WGS84
  - WGS84 is the most common CRS in the world, EPSG code: 4326.
- Projected coordinate systems: coordinate systems that are localized to minimize visual distortion in a particular region (e.g. Robinson, UTM, State Plane)
  - In some cases, it is not something that we are free to decide: "often the choice of projection is made by a public mapping agency" (Bivand, Pebesma, and Gómez-Rubio 2013).
  - This means that when working with local data sources, it is likely preferable to work with the CRS in which the data was provided.
  - For Bogotá the IGAC promotes the adoption of MAGNA-SIRGAS. EPSG code: 4626

Spatial Econometrics: Motivation

$$y = X\beta + u$$

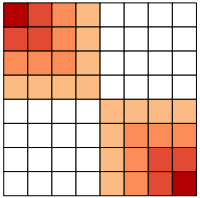
- Independence assumption between observation is no longer valid
- Attributes of observation i may influence the attributes of observation j.
- We will consider various alternatives to model spatial dependence

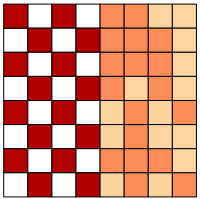


Spatial Econometrics: Motivation

$$y = X\beta + u$$

- Independence assumption between observation is no longer valid
- Attributes of observation i may influence the attributes of observation j.
- Positive Spatial correlation arises when units that are *close* to one another are more similar than units that are far apart
- Similarly spatial heterogeneity arises when some areas present more variability than others





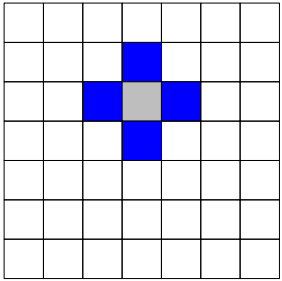
Spatial Econometrics: Closeness

"Everything is related to everything else, but close things are more related than things that are far apart" (Tobler, 1979).

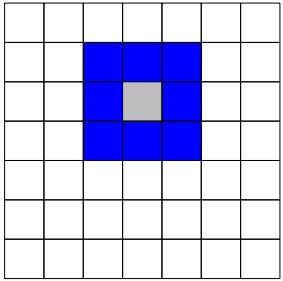
- One of the major differences between standard econometrics and standard spatial econometrics lies, in the fact that, in order to treat spatial data, we need to use two different sets of information
  1. Observed values of the economic variables
  2. Particular location where those variables are observed and to the various links of proximity between all spatial observations

Spatial Econometrics: Closeness

Rook criterion: two units are close to one another if they share a side



Queen criterion: two units are close if they share a side or an edge.



Spatial Econometrics: Weights Matrix

- At the heart of traditional spatial econometrics is the definition of the *weights matrix*:

$$W = \begin{pmatrix} w_{11} & \dots & \dots & w_{n1} \\ \vdots & w_{ij} & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{n_1} & \dots & \dots & w_{nn} \end{pmatrix}_{n \times n} \tag{1}$$

with generic element:

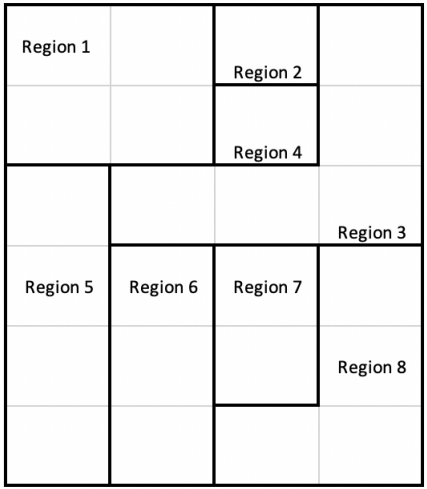
$$w_{ij} = \begin{cases} 1 & if\ j \in N(i) \\ 0 & o.w \end{cases} \tag{2}$$

$N(i)$ being the set of neighbors of location j. By convention, the diagonal elements are set to zero, i.e. $w_{ii} = 0$.

Spatial Econometrics: Weights Matrix

- The specification of the neighboring set ($N(i)$) is quite arbitrary and there's a wide range of suggestions in the literature.
  - Rook criterion
  - Queen criterion
  - Two observations are neighbors if they are within a certain distance, i.e., $j \in N(j)$ if $d_{ij} < d_{max}$ where d is the distance between location i and j.
  - Closest neighbor, ties can be solved randomly
  - More general matrices can also be specified by considering entries of w_{ij} as functions of geographical, economic or social distances between areas rather than simply characterized by dichotomous entries

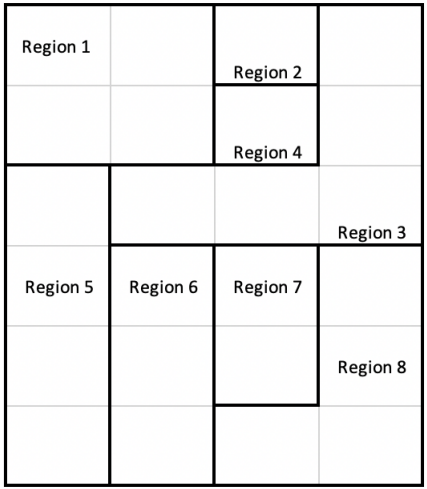
Some Examples of Weights Matrices



Adjacency Criterion

$$W = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}_{8\times 8}$$

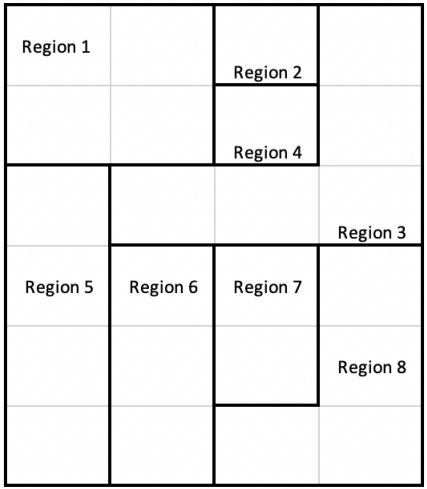
Some Examples of Weights Matrices



Nearest Neighbor

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}_{8\times 8}$$

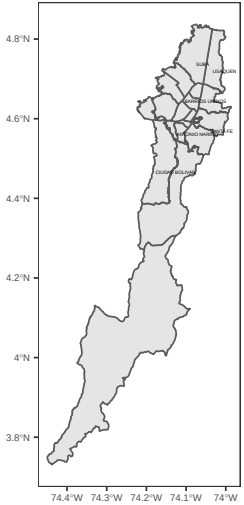
Some Examples of Weights Matrices



Distance < 2

$$W = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}_{8\times 8}$$

Some Examples of Weights Matrices



Some Examples of Weights Matrices

	ANTONIO NARIÑO	TUNJUELITO	RAFAEL URIBE URIBE	CANDELARIA	BARRIOS UNIDOS	TEUSAQUILLO	PUENTE ARANDA	LOS MARTIRES	SUMAPAZ	USAQUEN	CHAPINERO	SANTA FE	SAN CRISTOBAL	USME	CIUDAD BOLIVAR	BOSA	KENNEDY	FONTIBON	ENGATIVA	SUBA
ANTONIO NARIÑO	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0
TUNJUELITO	1	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0
RAFAEL URIBE URIBE	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
CANDELARIA	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
BARRIOS UNIDOS	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	1
TEUSAQUILLO	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0	0	0	1	1	0
PUENTE ARANDA	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0
LOS MARTIRES	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
SUMAPAZ	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
USAQUEN	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
CHAPINERO	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
SANTA FE	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0
SAN CRISTOBAL	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
USME	0	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0
CIUDAD BOLIVAR	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0
BOSA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
KENNEDY	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0
FONTIBON	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0
ENGATIVA	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
SUBA	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0

Some Examples of Weights Matrices

Quite often the W matrices are standardized to sum to one in each row

$$w_{ij}^{*} = \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}} w_{ij} \tag{3}$$

This can be quite useful since $L(y) = W^{*}y$ in which each single element is equal to

$$L(y_i) = \sum_{j=1}^{n} w_{ij}^{*} y_j \tag{4}$$

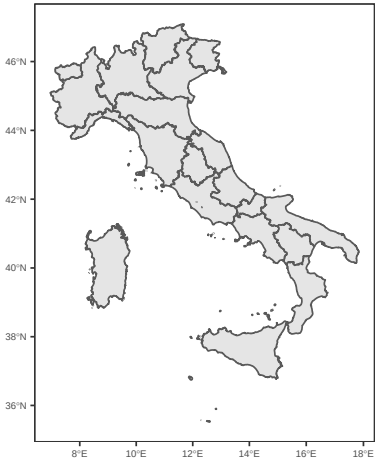
$$= \sum_{j=1}^{n} \frac{w_{ij} y_j}{\sum_{j=1}^{n} w_{ij}}$$

$$= \frac{\sum_{j \in N(i)} y_j}{\#N(i)} \tag{5}$$

Some Examples of Weights Matrices

	ANTONIO NARIÑO	TUNJUELITO	RAFAEL URIBE URIBE	CANDELARIA	BARRIOS UNIDOS	TEUSAQUILLO	PUENTE ARANDA	LOS MARTIRES	SUMAPAZ	USAQUEN	CHAPINERO	SANTA FE	SAN CRISTOBAL	USME	CIUDAD BOLIVAR	BOSA	KENNEDY	FONTIBON	ENGATIVA	SUBA
ANTONIO NARIÑO	0.0000000	0.1666667	0.1666667	0.0000000	0.0000000	0.0000000	0.1666667	0.1666667	0.0	0.00	0.0000000	0.1666667	0.1666667	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
TUNJUELITO	0.1666667	0.0000000	0.1666667	0.0000000	0.0000000	0.0000000	0.1666667	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.1666667	0.1666667	0.00	0.1666667	0.0000000	0.0000000	0.0000000
RAFAEL URIBE URIBE	0.2500000	0.2500000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.2500000	0.2500000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
CANDELARIA	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
BARRIOS UNIDOS	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.2000000	0.0000000	0.0000000	0.0	0.20	0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.2000000	0.2000000
TEUSAQUILLO	0.0000000	0.0000000	0.0000000	0.0000000	0.1428571	0.0000000	0.1428571	0.1428571	0.0	0.00	0.1428571	0.1428571	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.1428571	0.1428571	0.0000000
PUENTE ARANDA	0.1666667	0.1666667	0.0000000	0.0000000	0.0000000	0.1666667	0.0000000	0.1666667	0.0	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.1666667	0.1666667	0.0000000	0.0000000
LOS MARTIRES	0.2500000	0.0000000	0.0000000	0.0000000	0.0000000	0.2500000	0.2500000	0.0000000	0.0	0.00	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
SUMAPAZ	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
USAQUEN	0.0000000	0.0000000	0.0000000	0.0000000	0.3333333	0.0000000	0.0000000	0.0000000	0.0	0.00	0.3333333	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.3333333
CHAPINERO	0.0000000	0.0000000	0.0000000	0.0000000	0.2000000	0.2000000	0.0000000	0.0000000	0.0	0.20	0.0000000	0.2000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.2000000
SANTA FE	0.1666667	0.0000000	0.0000000	0.1666667	0.0000000	0.1666667	0.0000000	0.1666667	0.0	0.00	0.1666667	0.0000000	0.1666667	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
SAN CRISTOBAL	0.2500000	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	0.2500000	0.0000000	0.2500000	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
USME	0.0000000	0.2000000	0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.2	0.00	0.0000000	0.0000000	0.2000000	0.0000000	0.2000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000
CIUDAD BOLIVAR	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.2500000	0.0000000	0.25	0.2500000	0.0000000	0.0000000	0.0000000
BOSA	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.5000000	0.0000000	0.0000000	0.0000000
KENNEDY	0.0000000	0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.2000000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.20	0.0000000	0.2000000	0.0000000	0.0000000
FONTIBON	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.2500000	0.2500000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.2500000	0.0000000	0.2500000	0.0000000
ENGATIVA	0.0000000	0.0000000	0.0000000	0.0000000	0.2500000	0.2500000	0.0000000	0.0000000	0.0	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.2500000	0.0000000	0.2500000
SUBA	0.0000000	0.0000000	0.0000000	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000	0.0	0.25	0.2500000	0.0000000	0.0000000	0.0000000	0.0000000	0.00	0.0000000	0.0000000	0.2500000	0.0000000

Some Examples of Weights Matrices



Some Examples of Weights Matrices

	Piemonte	Valle D'Aosta	Lombardia	Trentino-Alto Adige	Veneto	Friuli Venezia Giulia	Liguria	Emilia-Romagna	Toscana	Umbria	Marche
Piemonte	0.0000000	0.25	0.2500000	0.00	0.0000000	0.00	0.2500000	0.2500000	0.0000000	0.0000000	0.0000000
Valle D'Aosta	1.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Lombardia	0.2500000	0.00	0.0000000	0.25	0.2500000	0.00	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000
Trentino-Alto Adige	0.0000000	0.00	0.5000000	0.00	0.5000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Veneto	0.0000000	0.00	0.2500000	0.25	0.0000000	0.25	0.0000000	0.2500000	0.0000000	0.0000000	0.0000000
Friuli Venezia Giulia	0.0000000	0.00	0.0000000	0.00	1.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Liguria	0.3333333	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.3333333	0.3333333	0.0000000	0.0000000
Emilia-Romagna	0.1666667	0.00	0.1666667	0.00	0.1666667	0.00	0.1666667	0.0000000	0.1666667	0.0000000	0.1666667
Toscana	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.2000000	0.2000000	0.0000000	0.2000000	0.2000000
Umbria	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.3333333	0.0000000	0.3333333
Marche	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.2000000	0.2000000	0.2000000	0.0000000
Lazio	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.1666667	0.1666667	0.1666667
Abruzzo	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.3333333
Molise	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Campania	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Puglia	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Basilicata	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Calabria	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Sicilia	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Sardegna	0.0000000	0.00	0.0000000	0.00	0.0000000	0.00	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

	Lazio	Abruzzo	Molise	Campania	Puglia	Basilicata	Calabria	Sicilia	Sardegna
Piemonte	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Valle D'Aosta	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Lombardia	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Trentino-Alto Adige	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Veneto	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Friuli Venezia Giulia	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Liguria	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Emilia-Romagna	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Toscana	0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Umbria	0.3333333	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Marche	0.2000000	0.2000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Lazio	0.0000000	0.1666667	0.1666667	0.1666667	0.0000000	0.0000000	0.0000000	0	0
Abruzzo	0.3333333	0.0000000	0.3333333	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Molise	0.2500000	0.2500000	0.0000000	0.2500000	0.2500000	0.0000000	0.0000000	0	0
Campania	0.2500000	0.0000000	0.2500000	0.0000000	0.2500000	0.2500000	0.0000000	0	0
Puglia	0.0000000	0.0000000	0.3333333	0.3333333	0.0000000	0.3333333	0.0000000	0	0
Basilicata	0.0000000	0.0000000	0.0000000	0.3333333	0.3333333	0.0000000	0.3333333	0	0
Calabria	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.0000000	0	0
Sicilia	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0
Sardegna	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0	0

Spatial Regressions

Spatial Autoregressive (SAR) Models

- Spatial lag dependence in a regression setting can be modeled similar to an autoregressive process in time series. Formally,

$$y = \rho W y + X\beta + \epsilon$$

- Wy induces a nonzero correlation with the error term, similar to the presence of an endogenous variable (OVB).
- Unlike to time series, Wy_i is always correlated with ϵ_i
- OLS estimates in the non spatial model will be biased and inconsistent. (Anselin and Bera, 1998)
- The estimation of the SAR model can be approached in two ways.
  1. Assume normality of the error term and use maximum likelihood.
  2. Use 2SLS
- In `R` the function `lagsarlm` uses MLE

Spatial Regresion in R

- Example crime, foreclosures, and unemployment
- Load Packages

```
require("spdep")
require("spatialreg")
require("stargazer")
```

- OLS

```
ols<-lm(violent~est_fcs_rt+bls_unemp, data=chi.poly)
```

- SAR model

```
list.queen<-poly2nb(chi.poly, queen=TRUE)
W<-nb2listw(list.queen, style="W", zero.policy=TRUE)
W
sar.chi<-lagsarlm(violent~est_fcs_rt+bls_unemp, data=chi.poly, W)
```

Spatial Regresion in R

```
stargazer(ols,sar.chi, header=FALSE, type="latex")
```

	Dependent variable:	
	Violent Crime	
	OLS	*SAR*
	(1)	(2)
Foreclosures	28.298***	15.682***
	(1.435)	(1.560)
Unemployment	−0.308	8.895*
	(5.770)	(5.245)
Constant	−18.627	−93.789**
	(45.366)	(41.316)
Observations	897	897
Note:	*p<0.1; **p<0.05; ***p<0.01	

Review & Next Steps

- Today:
  - Closeness
  - Weights Matrix
  - Examples of Weight Matrices Weights Matrix in R
  - Spatial Regressions (SAR Models)
- Next class: More on Spatial Regressions

Further Readings (I)

- Albouy, D., Christensen, P., & Sarmiento-Barbieri, I. (2020). Unlocking amenities: Estimating public good complementarity. Journal of Public Economics, 182, 104110.
- Anselin, Luc, & Anil K Bera. 1998. "Spatial Dependence in Linear Regression Models with an Introduction to Spatial Econometrics." Statistics Textbooks and Monographs 155. MARCEL DEKKER AG: 237–90.
- Arbia, G. (2014). A primer for spatial econometrics with applications in R. Palgrave Macmillan. (Chapters 2 and 3)
- Bivand, R. S., & Pebesma, E. J. (2020). Spatial Data Science https://keen-swartz-3146c4.netlify.app/ (Chapter 8)
- Bivand, R. S., Gómez-Rubio, V., & Pebesma, E. J. (2008). Applied spatial data analysis with R (Vol. 747248717, pp. 237-268). New York: Springer.
- Blumenstock, J., Cadamuro, G., & On, R. (2015). Predicting poverty and wealth from mobile phone metadata. Science, 350(6264), 1073-1076.
- Christensen, P., Sarmiento-Barbieri, I., Timmins C. (2020). Housing Discrimination and the Pollution Exposure Gap in the United States. NBER WP No. 26805
- Lee, K., & Braithwaite, J. (2020). High-Resolution Poverty Maps in Sub-Saharan Africa. arXiv preprint arXiv:2009.00544.

Further Readings (II)

- Lovelace, R., Nowosad, J., & Muenchow, J. (2019). Geocomputation with R. CRC Press. (Chapters 2 & 6)
- McMillen, D., Sarmiento-Barbieri, I., & Singh, R. (2019). Do more eyes on the street reduce Crime? Evidence from Chicago's safe passage program. Journal of urban economics, 110, 1-25.
- Sarmiento-Barbieri, I. (2016). An Introduction to Spatial Econometrics in R. http://www.econ.uiuc.edu/~lab/workshop/Spatial_in_R.html
- Tobler, WR. 1979. "Cellular Geography." In Philosophy in Geography, 379–86. Springer.
- Wasser, L. GIS With R: Projected vs Geographic Coordinate Reference Systems https://www.earthdatascience.org/courses/earth-analytics/spatial-data-r/geographic-vs-projected-coordinate-reference-systems-UTM/ Last Access September 10, 2020

Reading and Mapping spatial data in R

- Spatial data comes in various formats.
- One of the most used format are `shapefiles`
- This type of files stores non topological geometry and attribute information for the spatial features in a data set
  - Main file: file.shp
  - Index file: file.shx
  - dBASE table: file.dbf
- Data comes from `https://datosabiertos.bogota.gov.co`

Reading shapefiles in R

- Basic Packages
  - Read and handle spatial data

```
require("sf")
```

  - Plotting and data wrangling

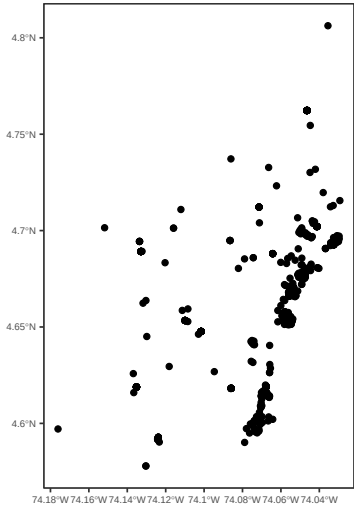
```
require("ggplot2")
require("dplyr")
```

```
bars<-st_read("egba/EGBa.shp")
```

```
## Reading layer 'EGBa' from data source 'egba/EGBa.shp' using driver
## 'ESRI Shapefile'
## Simple feature collection with 515 features and 7 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: -74.17607 ymin: 4.577897 xmax: -74.02929 ymax: 4.806253
## CRS:            4686
```

Visualizing Points

```
ggplot()+
  geom_sf(data=bars) +
  theme_bw() +
  theme(axis.title =element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
axis.text = element_text(size=6))
```



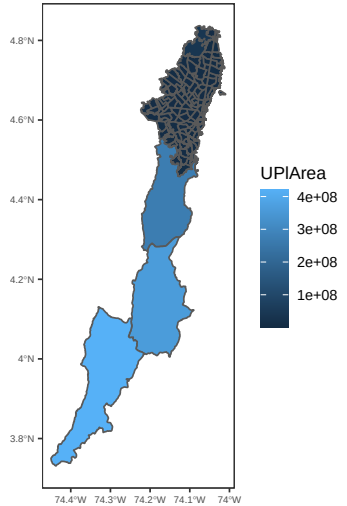
Visualizing Lines

```
ciclovias<-read_sf("Ciclovia/Ciclovia.shp")
ggplot()+
  geom_sf(data=ciclovias) +
  theme_bw() +
  theme(axis.title =element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size=6))
```

Visualizing Polygons

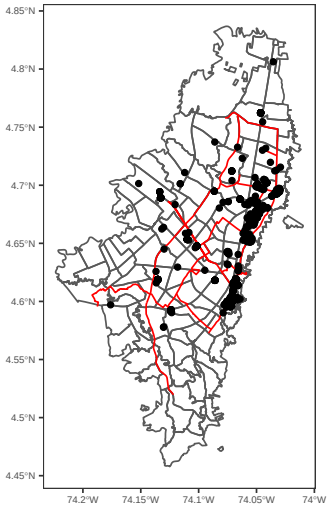
```
upla<-read_sf("upla/UPla.shp")

ggplot()+
  geom_sf(data=upla, aes(fill = UPlArea)) +
  theme_bw() +
  theme(axis.title =element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size=6))
```



Visualizing Points, Lines, and Polygons

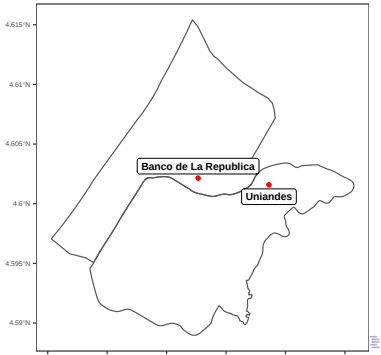
```
ggplot()+
  geom_sf(data=upla
%>% filter(grepl("RIO",UPlNombre)==FALSE),
fill = NA) +
  geom_sf(data=ciclovias, col="red") +
  geom_sf(data=bars) +
  theme_bw() +
  theme(axis.title =element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size=6))
```



Creating Spatial Objects

```
db<-data.frame(place=c("Uniandes","Banco de La Republica"),
        lat=c(4.601590,4.602151),
        long=c(-74.066391,-74.072350),
        nudge_y=c(-0.001,0.001))
db<-db %>% mutate(latp=lat,longp=long)
db<-st_as_sf(db,coords=c('longp','latp'),crs=4326)
```

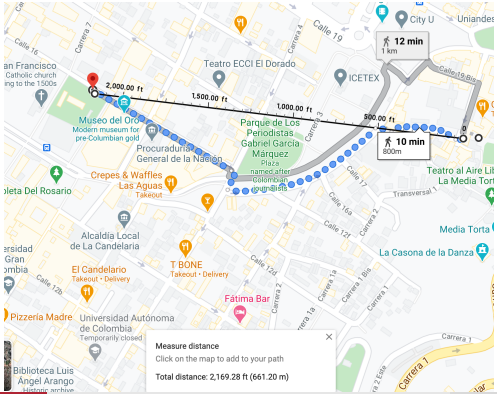
```
ggplot()+
  geom_sf(data=upla
    %>% filter(UPlNombre
    %in%c("LA CANDELARIA","LAS NIEVES")), fill = NA) +
  geom_sf(data=db, col="red") +
  geom_label(data = db, aes(x = long, y = lat,
                label = place),
       size = 3, col = "black", fontface = "bold",
       nudge_y =db$nudge_y) +
  theme_bw() +
  theme(axis.title =element_blank(),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       axis.text = element_text(size=6))
```



Measuring Distances

```
st_distance(db)
```

```
## Units: [m]
##           [,1]     [,2]
## [1,]   0.0000 664.1323
## [2,] 664.1323   0.0000
```



Measuring Distances

```
st_distance(db,ciclovias)
Error in st_distance(db, ciclovias) : st_crs(x) == st_crs(y) is not TRUE
st_crs(ciclovias)
```

```
## Coordinate Reference System:
##   User input: 3857
##   wkt:
## PROJCS["WGS 84 / Pseudo-Mercator",
##     GEOGCS["WGS 84",
##         DATUM["WGS_1984",
##             SPHEROID["WGS 84",6378137,298.257223563,
##                 AUTHORITY["EPSG","7030"]],
##             AUTHORITY["EPSG","6326"]],
##         PRIMEM["Greenwich",0,
##             AUTHORITY["EPSG","8901"]],
##         UNIT["degree",0.0174532925199433,
##             AUTHORITY["EPSG","9122"]],
##         AUTHORITY["EPSG","4326"]],
##     PROJECTION["Mercator_1SP"],
##     PARAMETER["central_meridian",0],
##     PARAMETER["scale_factor",1],
##     PARAMETER["false_easting",0],
##     PARAMETER["false_northing",0],
##     UNIT["metre",1,
##         AUTHORITY["EPSG","9001"]],
##     AXIS["X",EAST],
##     AXIS["Y",NORTH],
##     EXTENSION["PROJ4","+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +n
##     AUTHORITY["EPSG","3857"]]
```

Measuring Distances

```
ciclovias<-st_transform(ciclovias, 4686)
st_crs(ciclovias)
```

```
## Coordinate Reference System:
##   User input: EPSG:4686
##   wkt:
## GEOGCS["MAGNA-SIRGAS",
##     DATUM["Marco_Geocentrico_Nacional_de_Referencia",
##         SPHEROID["GRS 1980",6378137,298.257222101,
##             AUTHORITY["EPSG","7019"]],
##         TOWGS84[0,0,0,0,0,0,0],
##         AUTHORITY["EPSG","6686"]],
##     PRIMEM["Greenwich",0,
##         AUTHORITY["EPSG","8901"]],
##     UNIT["degree",0.0174532925199433,
##         AUTHORITY["EPSG","9122"]],
##     AUTHORITY["EPSG","4686"]]
```

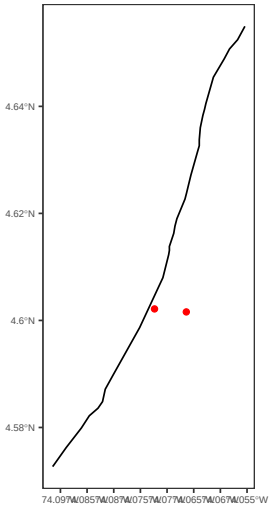
```
db<-st_transform(db, 4326)
st_distance(db,ciclovias)
```

```
## Units: [m]
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] 9514.617 10789.90 6035.283 12855.90 6025.017 8311.922 4579.450 741.6047
## [2,] 9221.998 10686.39 6143.960 13004.84 5871.073 7656.183 4014.993 116.5939
##          [,9]     [,10]    [,11]    [,12]    [,13]    [,14]
## [1,] 1002.8751 6255.692 2385.125 8402.580 8669.030 3788.265
```

Measuring Distances

```
ciclovias_sp<-ciclovias[8,]

ggplot()+
  geom_sf(data=ciclovias[8,], fill = NA) +
  geom_sf(data=db, col="red") +
  theme_bw() +
  theme(axis.title =element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size=6))
```



Weights Matrix in R

```
require("sf")
require("spdep")
require("dplyr")
```

```
chi.poly<-read_sf("foreclosures/foreclosures.shp")
st_crs(chi.poly) #doesn't have a projection
```

```
## Coordinate Reference System: NA
```

```
st_crs(chi.poly)<-4326 #WGS84 set it in the map
```

Weights Matrix in R

```
chi.poly<-st_transform(chi.poly,26916) #reproject planarly
#NAD83 UTM Zone 16N
st_crs(chi.poly)
```

```
## Coordinate Reference System:
##   User input: EPSG:26916
##   wkt:
## PROJCS["NAD83 / UTM zone 16N",
##     GEOGCS["NAD83",
##         DATUM["North_American_Datum_1983",
##             SPHEROID["GRS 1980",6378137,298.257222101,
##                 AUTHORITY["EPSG","7019"]],
##             TOWGS84[0,0,0,0,0,0,0],
##             AUTHORITY["EPSG","6269"]],
##         PRIMEM["Greenwich",0,
##             AUTHORITY["EPSG","8901"]],
##         UNIT["degree",0.0174532925199433,
##             AUTHORITY["EPSG","9122"]],
##         AUTHORITY["EPSG","4269"]],
##     PROJECTION["Transverse_Mercator"],
##     PARAMETER["latitude_of_origin",0],
##     PARAMETER["central_meridian",-87],
##     PARAMETER["scale_factor",0.9996],
##     PARAMETER["false_easting",500000],
##     PARAMETER["false_northing",0],
##     UNIT["metre",1,
##         AUTHORITY["EPSG","9001"]],
##     AXIS["Easting",EAST],
##     AXIS["Northing",NORTH],
##     AUTHORITY["EPSG","26916"]]
```

Weights Matrix in R

```
str(chi.poly)
```

```
## tibble [897 x 17] (S3: sf/tbl_df/tbl/data.frame)
##  $ SP_ID     : chr [1:897] "1" "2" "3" "4" ...
##  $ fips      : chr [1:897] "17031010100" "17031010200" "17031010300" "17031010400" ...
##  $ est_fcs   : int [1:897] 43 129 55 21 64 56 107 43 7 51 ...
##  $ est_mtgs  : int [1:897] 904 2122 1151 574 1427 1241 1959 830 208 928 ...
##  $ est_fcs_rt: num [1:897] 4.76 6.08 4.78 3.66 4.48 4.51 5.46 5.18 3.37 5.5 ...
##  $ res_addr  : int [1:897] 2530 3947 3204 2306 5485 2994 3701 1694 443 1552 ...
##  $ est_90d_va: num [1:897] 12.61 12.36 10.46 5.03 8.44 ...
##  $ bls_unemp : num [1:897] 8.16 8.16 8.16 8.16 8.16 8.16 8.16 8.16 8.16 8.16 ...
##  $ county    : chr [1:897] "Cook County" "Cook County" "Cook County" "Cook County" ...
##  $ fips_num  : num [1:897] 1.7e+10 1.7e+10 1.7e+10 1.7e+10 1.7e+10 ...
##  $ totpop    : int [1:897] 5391 10706 6649 5325 10944 7178 10799 5403 1089 3634 ...
##  $ tothu     : int [1:897] 2557 3981 3281 2464 5843 3136 3875 1768 453 1555 ...
##  $ huage     : int [1:897] 61 53 56 60 54 58 48 57 61 48 ...
##  $ oomedval  : int [1:897] 169900 147000 119800 151500 143600 145900 153400 170500 215900 114700 ...
##  $ property  : num [1:897] 646 914 478 509 641 612 678 332 147 351 ...
##  $ violent   : num [1:897] 433 421 235 159 240 266 272 146 78 84 ...
##  $ geometry  :sfc_POLYGON of length 897; first list element: List of 1
##   ..$ : num [1:15, 1:2] 443923 444329 444814 444839 444935 ...
##   ..- attr(*, "class")= chr [1:3] "XY" "POLYGON" "sfg"
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA NA NA NA NA ...
##   ..- attr(*, "names")= chr [1:16] "SP_ID" "fips" "est_fcs" "est_mtgs" ...
```

Weights Matrix in R

```
plot(chi.poly['violent'])
```

Weights Matrix in R

```
list.queen<-poly2nb(chi.poly, queen=TRUE)
W<-nb2listw(list.queen, style="W", zero.policy=TRUE)
W
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 897
## Number of nonzero links: 6140
## Percentage nonzero weights: 0.7631036
## Average number of links: 6.845039
##
## Weights style: W
## Weights constants summary:
##     n      nn  S0       S1       S2
## W 897 804609 897 274.4893 3640.864
```

Weights Matrix in R

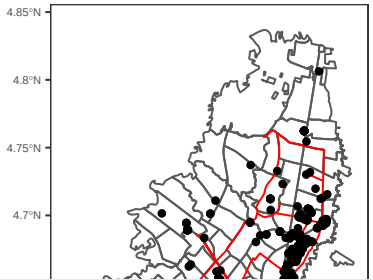
```
plot(W,st_geometry(st_centroid(chi.poly)))
```

Weights Matrix in R

```
coords <- st_centroid(st_geometry(chi.poly), of_largest_polygon=TRUE)
W_dist<-dnearneigh(coords,0,1000)
W_dist
```

```
## Neighbour list object:
## Number of regions: 897
## Number of nonzero links: 5448
## Percentage nonzero weights: 0.6770991
## Average number of links: 6.073579
## 55 regions with no links:
## 141 142 143 145 153 154 155 158 462 631 637 638 642 643 644 645 655 656 657 658 659 758 759 769 820 821 822 823 824 855 856 857 861 862 8
```

```
plot(W_dist, coords)
```



Weights Matrix in R

```
W_dist<-dnearneigh(coords,0,4300)
W_dist
```

```
## Neighbour list object:
## Number of regions: 897
## Number of nonzero links: 87988
## Percentage nonzero weights: 10.9355
## Average number of links: 98.09142
```

```
plot(W_dist, coords)
```