Spatial Data and Analysis

Discussion 4

Felipe González

UC Berkeley

September 25*th*

Outline

Display text

- Until now, we have been using the command `disp('Hi')` to display text in the command window
- However, this command does not interact in a great way with numbers. This is a major disadvantage, specially when we are trying to compress the output of our code
- Sometimes you might want to display actual calculations. To achieve more control over the display we will use the command `fprintf`

Display text — `fprintf`

- The command `fprintf` has the following structure:

```
fprintf('Number %f displayed', x)
```

- The symbol % tells MATLAB that there is a number in that position. The letter after % specifies the format

- For example, the following would produce the mean of 100 draws from a normal distribution with 2 decimal numbers:

```
fprintf('The mean is %.2f', mean(randn(100,1)))
```

Display text — useful commands

`%d`	:	integer
`%s`	:	string
`%f`	:	number with decimals
`%.1f`	:	number with 1 decimal
`n`	:	print a new line
`t`	:	print a tab

Interactive screening

- Draw N numbers from a normal distribution. Repeat this S times and answer the following questions:
  1. What is the mean of the distribution of minimums?
  2. Plot the distribution of minimums

Interactive screening

- Draw N numbers from a normal distribution. Repeat this S times and answer the following questions:
  1. What is the mean of the distribution of minimums?
  2. Plot the distribution of minimums
- Write a program that does the following:
  1. Allows a random user to select N and S
  2. Prints output in a friendly way

Interactive screening — example

- Show m-file with written program
- You can download it from the bCourse website in the `Files/Resources` folder

Polygons, structures and shapefiles

- Theoretically, a polygon is a closed shape where the first and last vertex are the same:

$$\widetilde{P} \;=\; \{\vec{X}, \vec{Y}\} \qquad \text{where } \vec{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \text{ and } \vec{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

  i.e., $x_1 = x_N$ and $y_1 = y_N$

- However, in reality many units of interest are composed by multiple polygons (e.g., Hawaii)

Polygons, structures and shapefiles

- In that case, multiple polygons are separated by a missing value. Example of two polygons defining a state:

$$\widetilde{P} = \{\vec{X}, \vec{Y}\} \quad \text{where } \vec{X} = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1N} \\ . \\ x_{21} \\ \vdots \\ x_{2N} \end{bmatrix} \text{ and } \vec{Y} = \begin{bmatrix} y_{11} \\ \vdots \\ y_{1N} \\ . \\ y_{21} \\ \vdots \\ y_{2N} \end{bmatrix}$$

in which case, $x_{11} = x_{1N}$, $x_{21} = x_{2N}$, and $y_{11} = y_{1N}$, $y_{21} = y_{2N}$

Shapefiles

- A shapefile is a set of four files that define structures (`s`) and attributes (`a`):
  1. `NAME.shp` → two vectors defining polygons
  2. `NAME.dbf` → attributes
  3. `NAME.shx`
  4. `NAME.prj`

Shapefiles

- A shapefile is a set of four files that define structures (`s`) and attributes (`a`):
  1. `NAME.shp` → two vectors defining polygons
  2. `NAME.dbf` → attributes
  3. `NAME.shx`
  4. `NAME.prj`
- 3 types of shapefiles: points, lines, and polygons

Shapefiles

- A shapefile is a set of four files that define structures (`s`) and attributes (`a`):
  1. `NAME.shp` → two vectors defining polygons
  2. `NAME.dbf` → attributes
  3. `NAME.shx`
  4. `NAME.prj`
- 3 types of shapefiles: points, lines, and polygons
- A structure is a set of polygons (e.g., U.S. is a structure composed by states)
- An attribute is a characteristic of a structure (e.g., population)

Shapefiles — lat lon variables

- Two variables in `NAME.shp` define polygons:
  1. Latitude
  2. Longitude

Shapefiles — lat lon variables

- Two variables in `NAME.shp` define polygons:
  1. Latitude
  2. Longitude
- Latitude and longitude are nothing more than numbers in a vector, separated by missing values (`NaN`) that define the end of a closed polygon

Shapefiles — lat lon variables

- Two variables in `NAME.shp` define polygons:
  1. Latitude
  2. Longitude
- Latitude and longitude are nothing more than numbers in a vector, separated by missing values (`NaN`) that define the end of a closed polygon
- How to open a shapefile:

```
[s,a] = shaperead('NAME.shp','UseGeoCoords',true);
```

Shapefiles — lat lon variables

- Let's count the number of closed polygons in a structure:

```
1       disp('Number of closed polygons:')
2       disp(sum(isnan(s(1).Lon)))
```

Shapefiles — lat lon variables

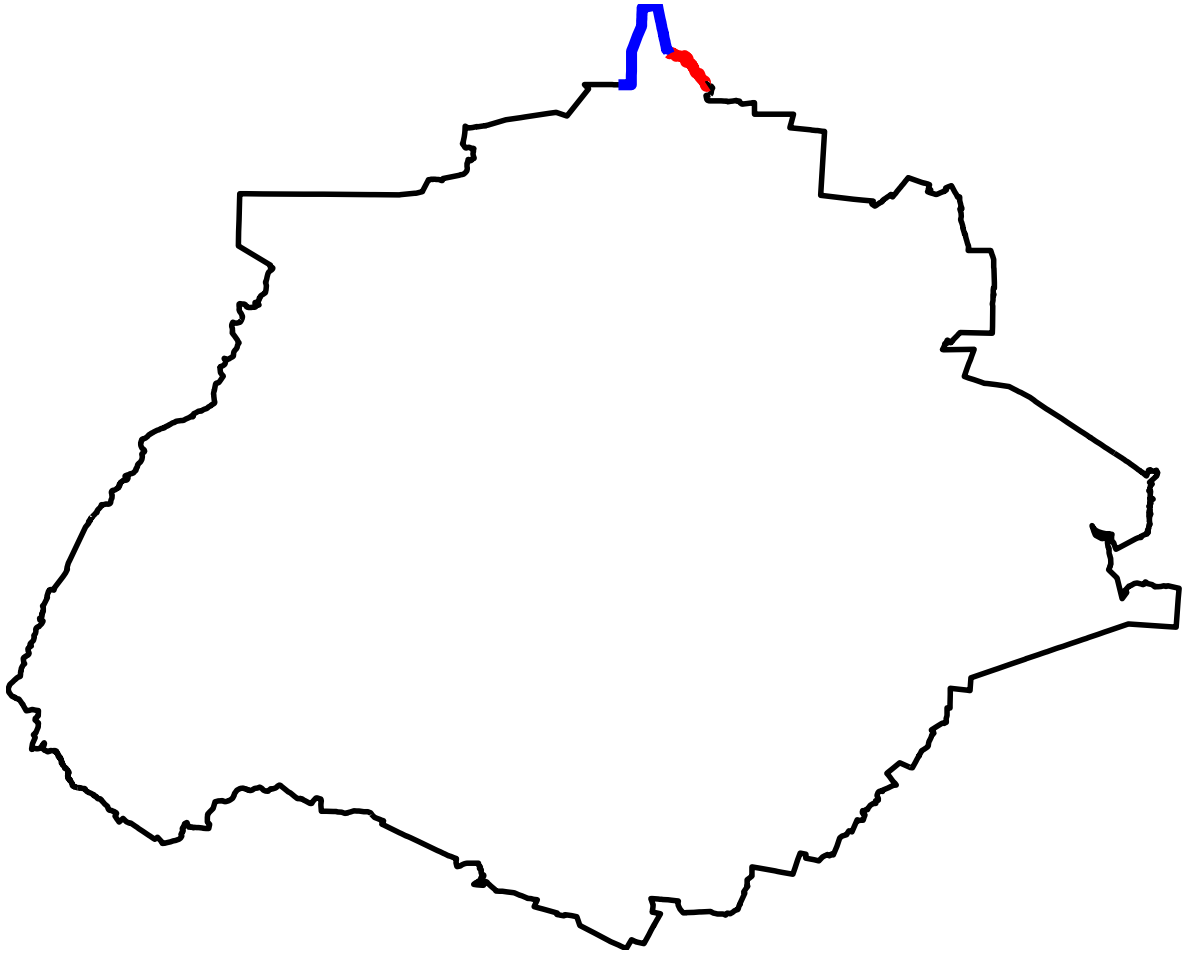
- Let's count the number of closed polygons in a structure:

```
disp('Number of closed polygons:')
disp(sum(isnan(s(1).Lon)))
```

- Now let's explore values in lat lon vectors:

```
x = 100;
hold on
plot(s(1).Lon, s(1).Lat, '-k', 'LineWidth',2)
plot(s(1).Lon(1,1:x), s(1).Lat(1,1:x),...
    '-r','LineWidth',4)
plot(s(1).Lon(1,end-x:end),...
     s(1).Lat(1,end-x:end),'-b','LineWidth',4)
hold off
```

Shapefiles — lat lon variables



Shapefiles — multiple polygons

- How do you plot multiple polygons into a single figure?

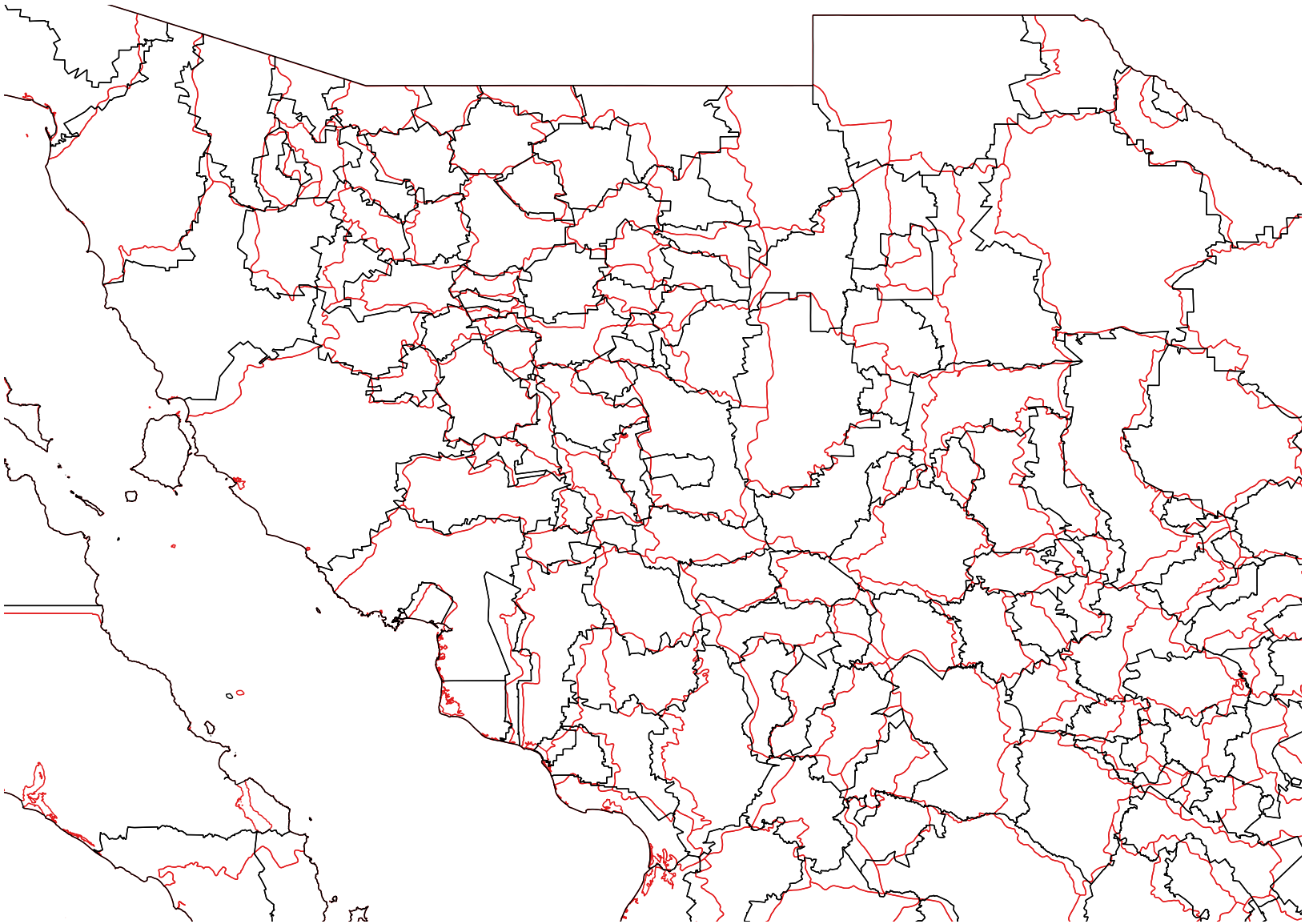
```
1       hold on
2       for i = 1:size(s,1)
3           plot(s(i).Lon, s(i).Lat, 'k')
4               set(gcf,'color','w')
5               box off
6               axis equal
7               axis off
8       end
9       hold off
```

Shapefiles — multiple polygons

Shapefiles — where to find them?

- There are multiple sources in the internet where you can get shapefiles for free. These are usually `.zip` files with previously discussed files (example: DIVA-GIS)
- However, careful is needed because of two different reasons:
  1. Not all shapefiles use the same projection
  2. Check shapefiles make sense (attributes and geometry)
- Official sources are always preferred

Official source and downloaded from website



Occupied zones

- In many cities there are zones in which the state does not have control. For simplicity we will call these "occupied zones"
- This means lots of people don't benefit from public programs and don't have access to basic infrastructure such as police stations, hospitals, schools, etc
- In the following application we will empirically analyze with MATLAB the real case of occupied zones in a large Latin American city

Occupied zones

- In particular, we are interested in two things when it comes to ocuppied zones and information about people living in there:

  1. How many people actually live in occupied zones?
     - This will be an estimate with a confidence interval

  2. How are aggregate statistics biased because of the existence of occupied zones that don't appear in surveys?
     - How to guess statistics of occupied zones?

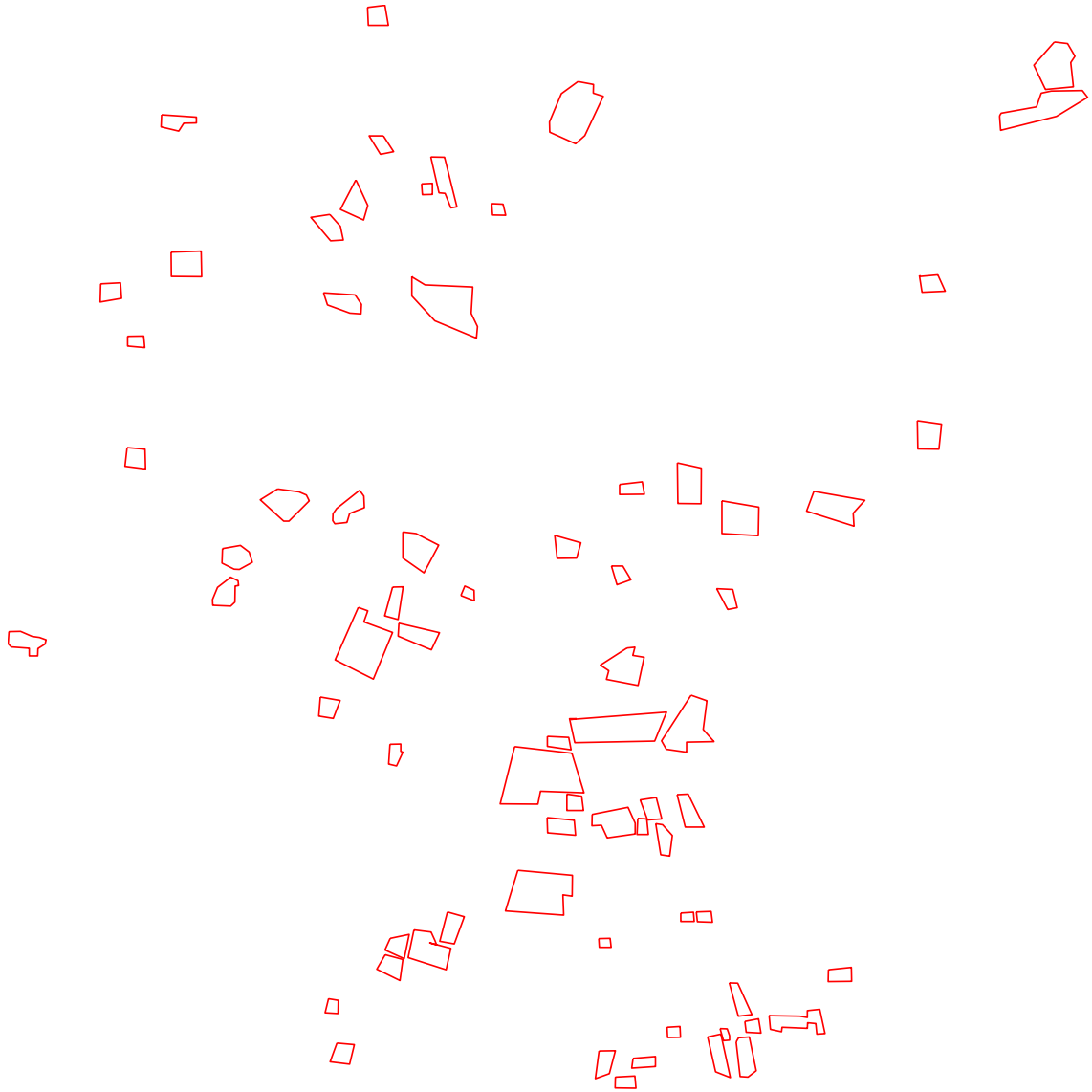
Occupied zones according to news story

Digitizing occupied zones

- Digitizing occupied zones in an image into a shapefile is easier in a different software which we will review in another discussion
- There are 71 polygons in the previous figure.
- Let's open that shapefile:

```
[z,b] = shaperead('Zonas Ocupadas.shp',...
                  'UseGeoCoords', true) ;
```

Occupied zones in a shapefile

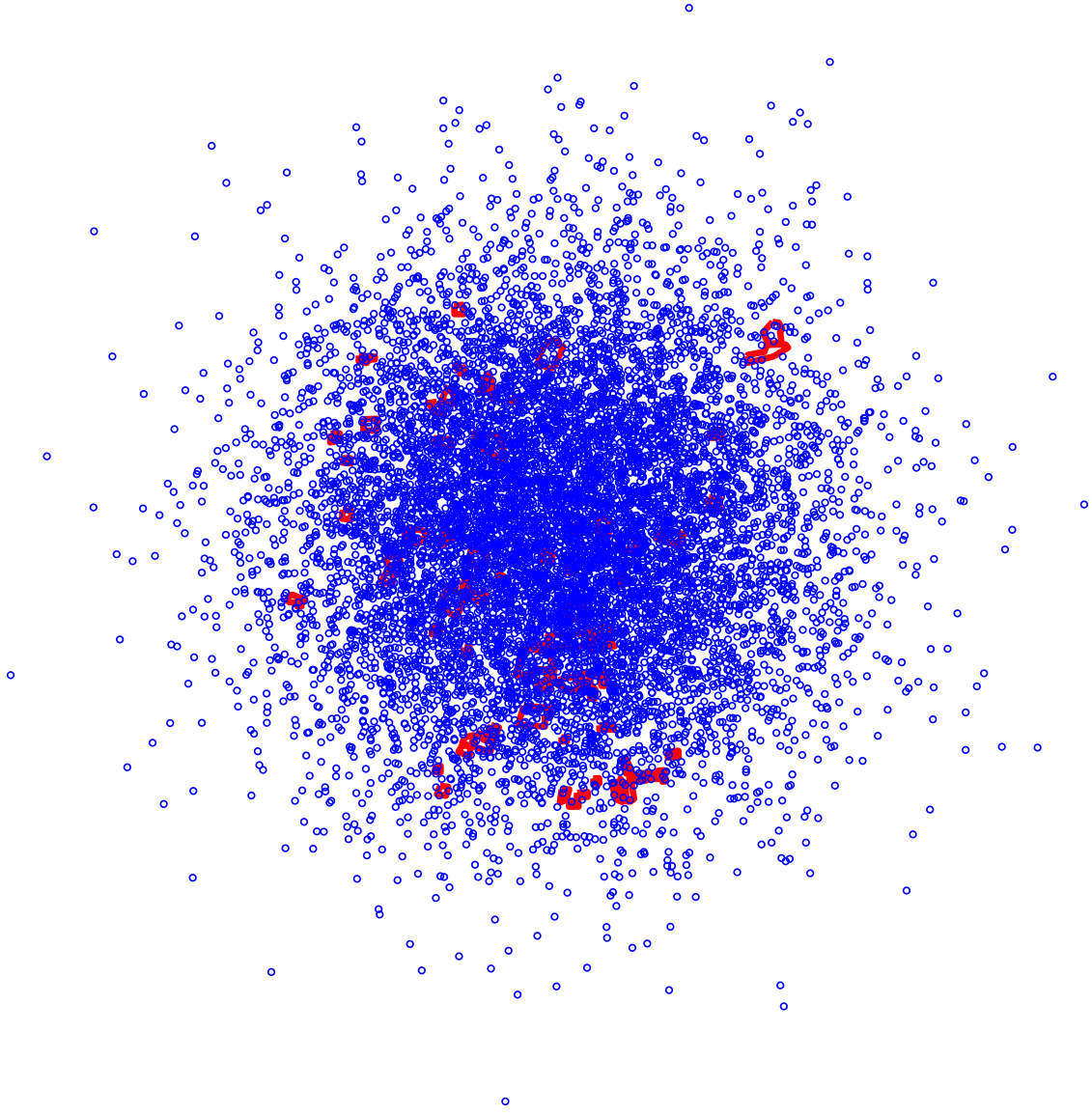


Random locations in the city

```
% Random locations in Santiago
    clear ; clc
    rng(1234)

% Generates 'N' random locations in Santiago
    N     = 10000                          ;
    loc_x = -70.63 + .08 * randn(N, 1) ;
    loc_y = -33.47 + .08 * randn(N, 1) ;
```

Simulating 10,000 random locations in the city



Inside and outside occupy zones

```
% Loop over 71 occupied zones
IN = NaN(N, 71);
     for i = 1:71
         IN(:,i) = inpolygon(loc_x, loc_y,...
                   z(i).Lon', z(i).Lat');
     end
disp('Number of individuals living in occupy zones:')
disp(sum(sum(IN)))
```

Answer: with random geographic sampling there are $\sim$4% of individuals living inside occupied zones. A Montecarlo exercise in the 'appropriate dimension' can give you a sense of uncertainty

Bias in aggregate statistics

- How are aggregate statistics biased because of the existence of occupied zones that don't appear in surveys?

Bias in aggregate statistics

- How are aggregate statistics biased because of the existence of occupied zones that don't appear in surveys?
- The key to answer this question is to find a similar group of individuals and assign their observables to people living inside occupied zones

Bias in aggregate statistics

- How are aggregate statistics biased because of the existence of occupied zones that don't appear in surveys?
- The key to answer this question is to find a similar group of individuals and assign their observables to people living inside occupied zones
- One alternative is to use observables of people living in areas closed to occupied zones. Let's define 'close' as approximately 1 kilometer

Buffer zones for occupied zones

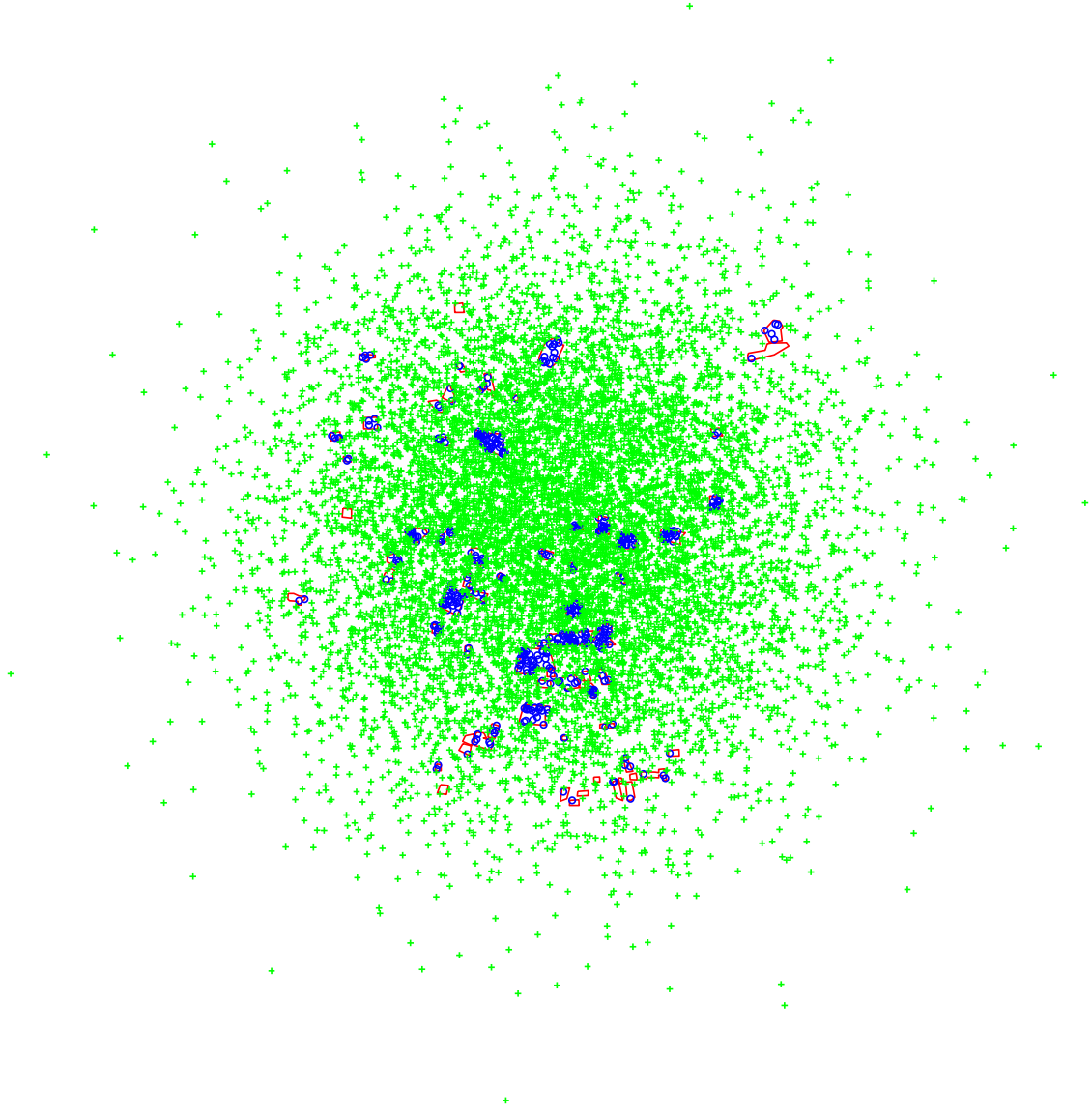
In order to find people living close to ocuppied zones we will create buffer zones around occupied zones

Buffer zones for occupied zones

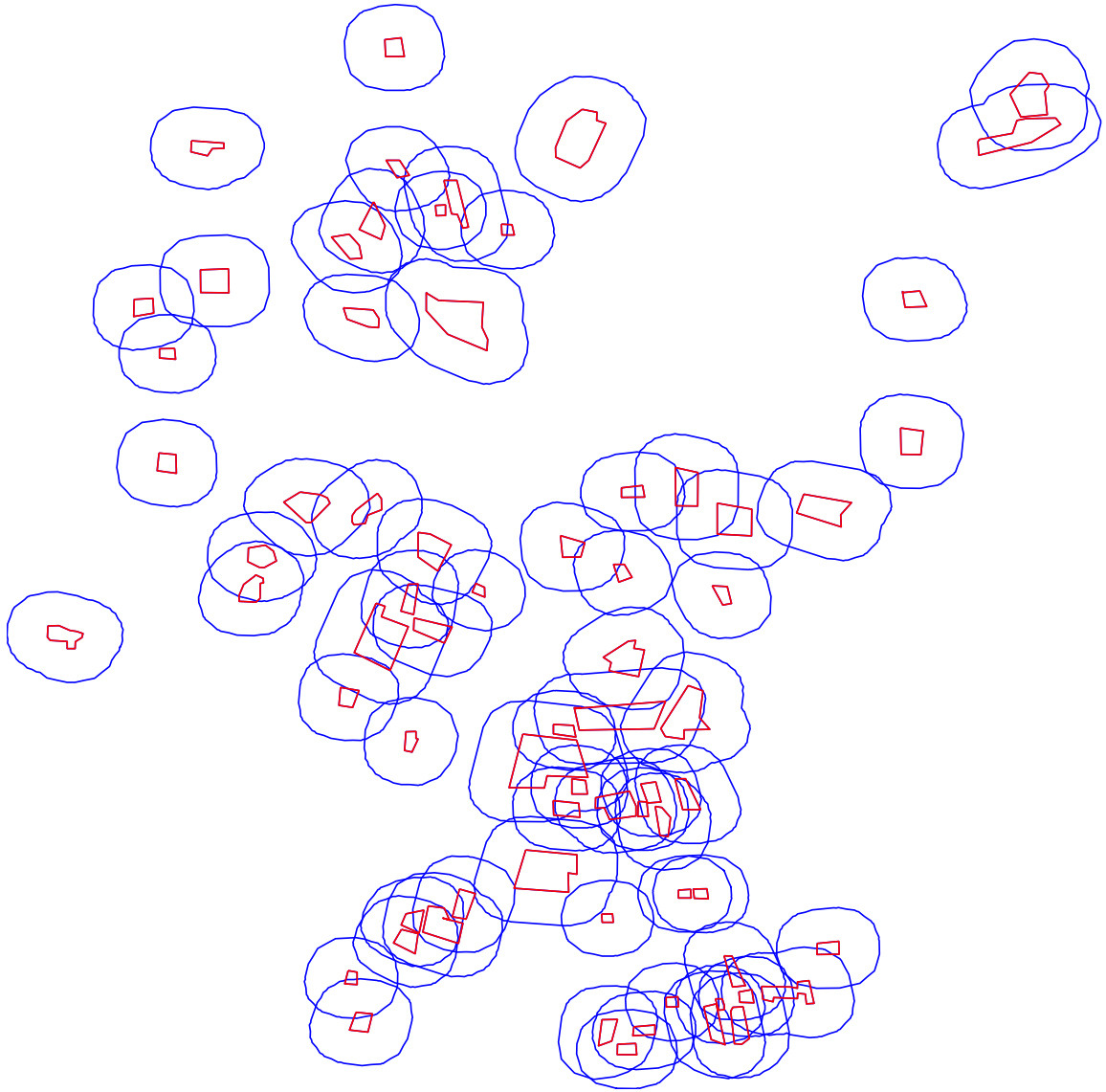
In order to find people living close to ocuppied zones we will create buffer zones around occupied zones

```
1       B(71).Lon = '' ;
2       B(71).Lat = '' ;
3       disp('Generates structure with buffers:')
4       for i = 1 : 71
5           [buff_lat, buff_lon] = bufferm(z(i).Lat',...
6                                          z(i).Lon',.01);
7           B(i) = struct('Lon', {buff_lon},...
8                  'Lat', {buff_lat}) ;
9           disp(i)
10      end
```

Households inside and outside of occupied zones



Buffer zones of .01 degrees ($\approx$ 1 km.) around each zone



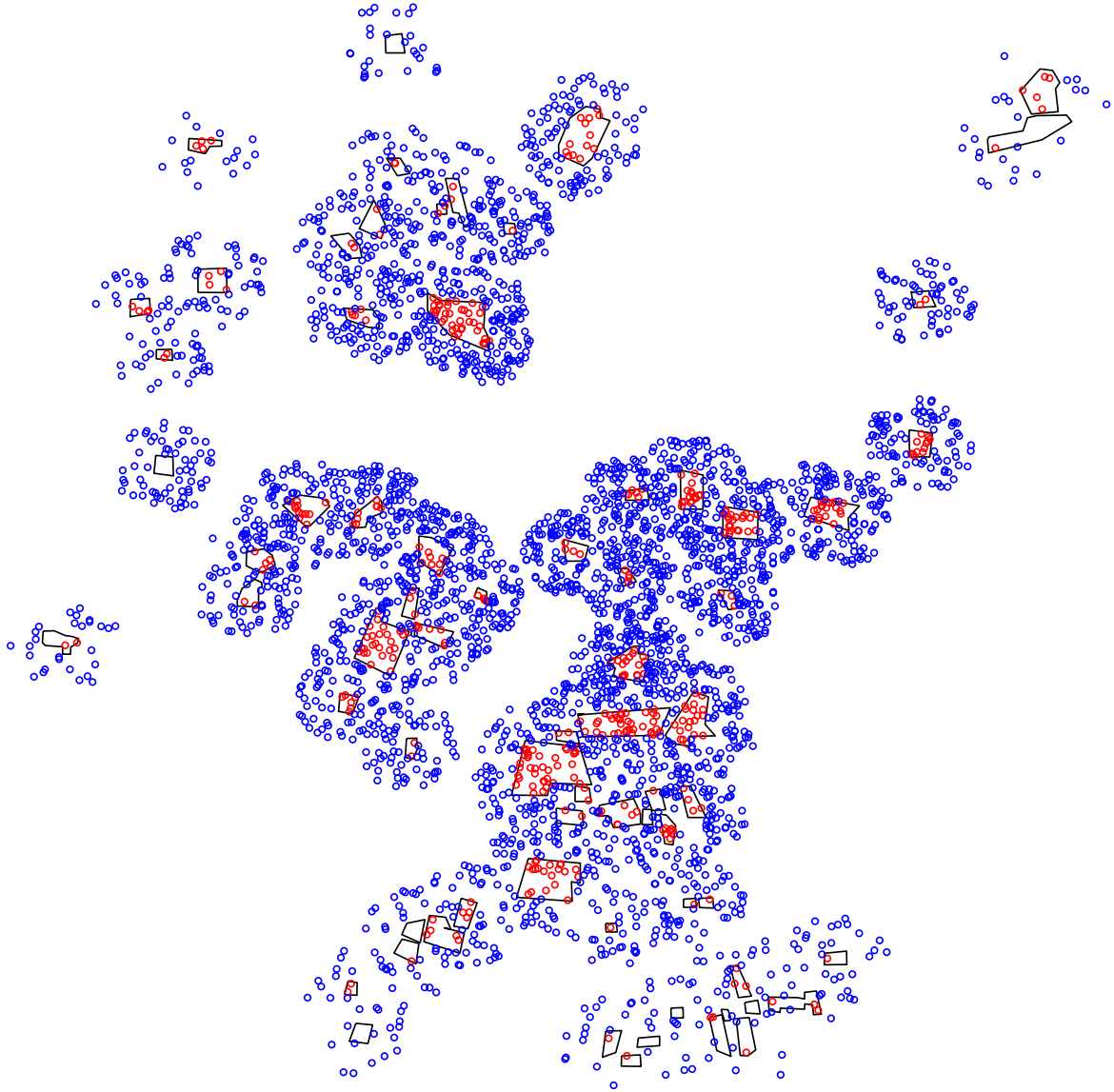
Group of interest

```
% Living in buffers outside occupy zones
% Loop over 71 occupy locations
Control = NaN(N, 71);
    for i = 1:71
        Control(:,i) = inpolygon(loc_x, loc_y, B(i).Lon'
    end

% Individuals can be located in more than 1 buffer
A   = sum(Control, 2)  ;
OUT = (A > 0 & IN == 0);

% Group of interest
    disp('Number of individuals living in buffers:')
    disp(sum(sum(OUT)))
```

Inside and outside households



Additional resources

- Where to download shapefiles:
  - DIVA-GIS
- Check some MATLAB code at Prof. Hsiang's website