Spatial Data and Analysis

Discussion 2

Felipe González

UC Berkeley

September 11*th*

Admin stuff

1. Check the discussion syllabus in the website
2. Remember to use "`[spatial]`" in the subject header
3. Check discussion slides, there is useful stuff

Outline

Miscellaneous #1: vector and locations

- For simplicity think of a vector as a $n \times 1$ matrix: $\vec{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$

- N locations can be described by 2 vectors of size N:

$$(\vec{x}, \vec{y}) \;=\; \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

  where (x_i, y_i) is a single location with $i = 1, \ldots, N$

- Random locations: $(\vec{x}, \vec{y})$ is `[randn(N, 1) randn(N,1)]`

Miscellaneous #2: matlab v/s stata

- Matlab reads the code `plot(A,B)` and plots A in the x-axis and B in the y-axis $\sim$ `plot(x,y)`
- Stata reads the code `scatter A B` and plots A in the y-axis and B in the x-axis $\sim$ `scatter y x`
- In a way, matlab thinks more mathematically and stata more statistically.
- When plotting locations in matlab: `plot(lon,lat)`, not the perhaps more intuitive `plot(lat,lon)`

Logical operators

- Basic logical operators we will use repeatedly:

$$\begin{aligned} \text{EQUAL} &: \quad = \\ \text{GREATER} &: \quad > \\ \text{GREATER OR EQUAL} &: \quad >= \\ \text{AND} &: \quad \& \\ \text{OR} &: \quad | \\ \text{IS NOT} &: \quad \sim= \end{aligned}$$

- Useful to create indicator variables and others

Example

- Examples of logical operators:

```
% Parameters
    A = [1 0 1];
    B = [0 0 1];
    C = [0 4 0];

% Operations
    C = (A==1 & B==1) ;
    D = (A==1 | B==1) ;
    E = A(A >1)       ;
    F = C(C >1)       ;
```

Example (cont.)

- You very rarely have to loop through a whole matrix to get only specific parts of it
- Exercise: find an approximate value for the mean of negative numbers in a normal distribution with 1,000 values

```
X = randn(1000,1) ;
mean(X(X<0));
```

if

- Sometimes `if` statements can be useful:

```matlab
% Parameters
    N = 100             ;
    A = ceil(randn(N,1));
    B = NaN(N,1)        ;

% If statement
    if   A(10, 1) == 1
         disp('Hi everyone')
    else disp('Its hot')
    end
```

Comments

- Note that the command `==` is used when using a statement, and `=` when assigning values
- Short circuit behavior:

$$\begin{aligned} \texttt{A}\ \&\ \texttt{B} \quad &: \quad \texttt{A} \text{ and } \texttt{B} \text{ are evaluated} \\ \texttt{A}\ \&\&\ \texttt{B} \quad &: \quad \texttt{B} \text{ only evaluated if } \texttt{A} \text{ is true} \end{aligned}$$

- Multiple statements need to be separated by parentheses:

```
% Statement 1
    (x == 3 & x > 4) | x < 4
% Statement 2
     x == 3 & (x > 4 | x < 4)
```

Loops

- Loops allow you to execute a block of code many times. For example, create a vector x with ten ones using a loop:

```
for i = 1 : 10
      x(i, 1) = 1 ;
end
```

- However, this is obviously inefficient since we can create the same vector simply typing `x=ones(10,1)`.

- Knowing when to use a loop is an important part of being efficient at coding. Indentation is important for readability

Example

- Create a fake dataset with panel structure where variables are 100 ID's and 10 years:

ID	Year
1	1
$\vdots$	$\vdots$
1	10
$\vdots$	$\vdots$
100	1
$\vdots$	$\vdots$
100	10

Solution

```matlab
% Parameters
    ID = 100          ;
    t  = 10           ;
    x  = NaN(ID*t, 2) ;

% Loop
    for i = 1:ID
        x((i-1)*10+1:i*10, 1) = i    ;
        x((i-1)*10+1:i*10, 2) = 1:10 ;
    end
```

Monte Carlo

- Monte Carlo is an algorithm that uses random sampling to obtain numerical results
- This method is useful in applied statistics to obtain quick answers to a wide range of questions
- Consider $y_i = 5 + 0.5x_i + \varepsilon_i$. Then:
  1. what is the distribution of an OLS estimator using (y, x)?
  2. what are the consequences of classical measurement error in x?
- Assume $x \sim N(0,1)$, $\varepsilon \sim N(0,1)$, $x \perp \varepsilon$ and different magnitudes for the measurement error

Structure and parameters

- Think about the following structure to answer these questions: define parameters, procedure, output

```
% Parameters
    clear ; clc
    s = 1000    ;
    N = 100     ;
    b = NaN(s,1);
    g = NaN(s,1);
```

- Always useful to think about what is the output of the process you're writing

```
% Loop
    for i = 1:s
    % Variables
        x = randn(N, 1) ;
        z = x - randn(N,1) ;
        y = 5 + .5 * x + randn(N,1);
    % OLS coefficients
        X = [ones(N,1) x] ;
        beta = (X' * X) \ (X' * y) ;
        b(i,1) = beta(2, 1);
    % Attenuated OLS coefficients
        Z = [ones(N,1) z] ;
        gamma = (Z' * Z) \ (Z' * y) ;
        g(i,1) = gamma(2, 1);
    end
```

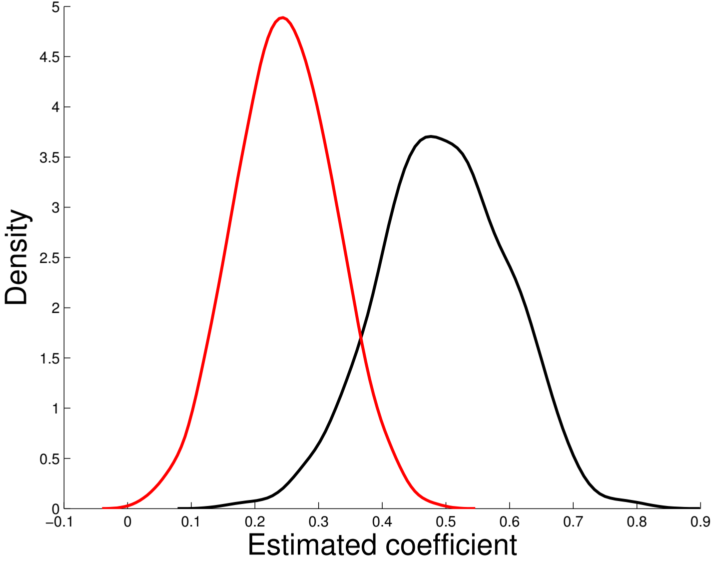
Output

```
% Figure
    [a,b] = ksdensity(b) ;
    [c,d] = ksdensity(g) ;

    hold on
    plot(b, a, 'k', 'LineWidth', 2)
    plot(d, c, 'r', 'LineWidth', 2)
        ylabel('Density'    , 'FontSize' , 20)
        xlabel('Coefficient', 'FontSize' , 20)
        box off
        set(gcf, 'Color', 'w')
    hold off

    export_fig 'ols.pdf'
```

Figure: *Distribution of OLS coefficient and attenuation bias*

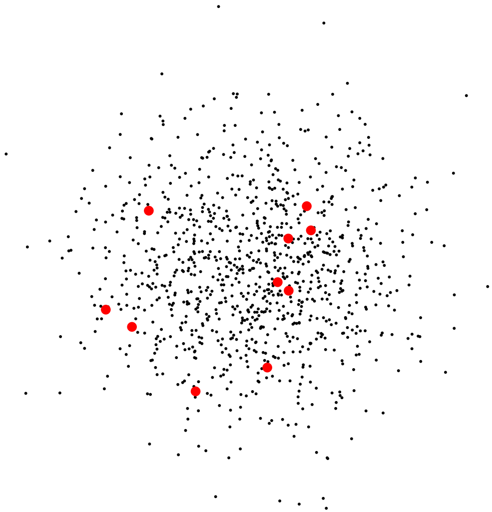


Application: school choice

- Consider a simple model of school choice in which students enroll in the school that is closer to their home
- As researcher you are worried that students' locations could be measured with error
- This is crucial if we want to predict how many students will enroll in each school (e.g., construction of schools)
- Use point processes, distances, Monte Carlo and the model of school choice to explore the consequences of measurement error

```
% Parameters
    N     = 1000     ; % students
    S     = 10       ; % schools
    x      = 0:.01:1 ; % size of noise
    ERROR = NaN(N, size(x, 2)) ;
% Location of schools
    lat_s = randn(S,1)    ;
    lon_s = randn(S,1)    ;
    LOC_s = [lon_s lat_s] ;
% Location of students
    lat_n = randn(N,1)    ;
    lon_n = randn(N,1)    ;
    LOC_n = [lon_n lat_n] ;
% Distance and school choice
    D = pdist2(LOC_n, LOC_s, 'euclidean') ;
    [d, c] = min(D,[],2) ;
```

Figure: *Schools and students in space*



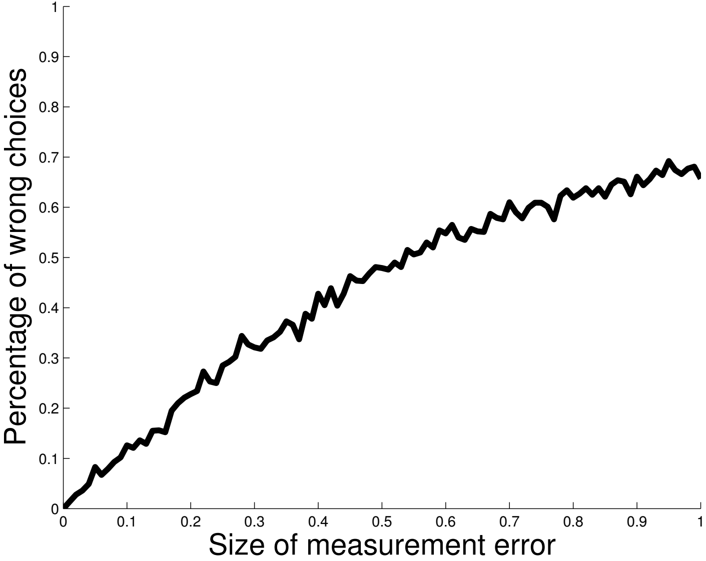
```
% Loop for different types of noise
    for i = 1:size(x,2)
    % Noisy location of students
        lat_n_n = lat_n + x(1,i)*randn(N,1) ;
        lon_n_n = lon_n + x(1,i)*randn(N,1) ;
        LOC_n_n = [lon_n_n lat_n_n]    ;
    % Noisy distance from students to schools
        D_n = pdist2(LOC_n_n, LOC_s, 'euclidean') ;
    % Noisy school choice
        [d_n, c_n] = min(D_n,[],2) ;
    % Actual and noisy school choice
        ERROR(:, i) = (c ~= c_n) ;
    end
% Percentage of errors in choices
    result = sum(ERROR, 1)./N ;
```

Output

```
% Figure
plot(x, result, 'k', 'LineWidth', 4)
  ylabel('Percentage of wrong choices', 'FontSize', 20)
  xlabel('Size of measurement error'  , 'FontSize', 20)
  axis([0 1 0 1])
  box off
  set(gcf, 'Color', 'w')

  export_fig 'error_1.pdf'
```

Figure: *Mistakes due to measurement error*



Incorporating uncertainty

```
% Loop for different location of students
    for j = 1:M
    % Location of students
        lat_n = randn(N,1)    ;
        lon_n = randn(N,1)    ;
        LOC_n = [lon_n lat_n] ;
    % Distance and school choice
        D = pdist2(LOC_n, LOC_s, 'euclidean') ;
        [d, c] = min(D,[],2) ;
    % Loop for different types of noise
        for i = 1 : size(x, 2)
        % Noisy location of students
            lat_n_n = lat_n + x(1,i)*randn(N,1) ;
```

Incorporating uncertainty

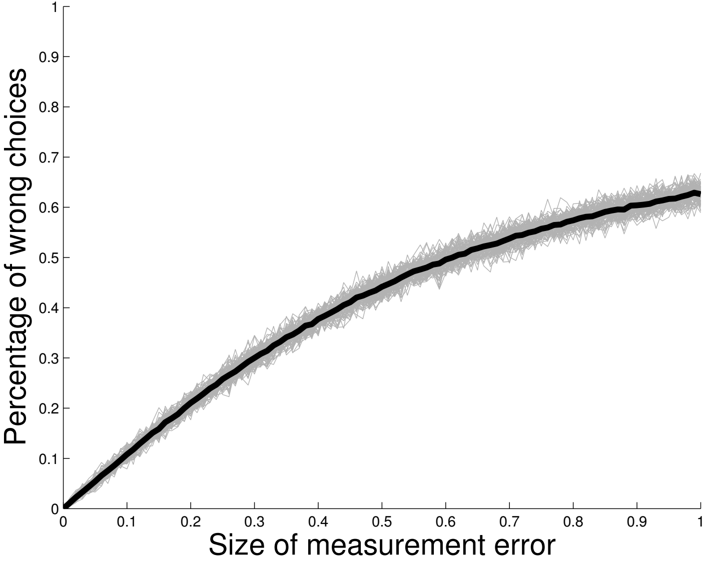
```
                lon_n_n = lon_n + x(1,i)*randn(N,1) ;
                LOC_n_n = [lon_n_n lat_n_n]    ;
            % Noisy distance and noisy school choice
                D_n = pdist2(LOC_n_n, LOC_s, 'euclidean') ;
                [d_n, c_n] = min(D_n,[],2) ;
            % Actual and noisy school choice
                ERROR(:, i) = (c ~= c_n) ;
            end
        % Percentage of errors in choices
            result(j,:) = sum(ERROR, 1)./N ;
        end
```

Output

```
% Plot error
hold on
plot(x, result, 'Color', [.7 .7 .7], 'LineWidth', .5)
plot(x, mean(result), 'k', 'LineWidth', 4 )
 ylabel('Percentage of wrong choices', 'FontSize' , 20)
 xlabel('Size of measurement error'  , 'FontSize' , 20)
 axis([0 1 0 1])
 box off
 set(gcf, 'Color', 'w')
hold off

export_fig 'error_2.pdf'
```

Figure: *Incorporating uncertainty using Montecarlo*



Additional resources

1. Explore `export_fig`
2. Explore `mcode`
3. Explore options of commands we have used (e.g, `plot`, `pdist2`)