Spatial Data Analysis in Stata
An Overview

Maurizio Pisati

Department of Sociology and Social Research
University of Milano-Bicocca (Italy)
maurizio.pisati@unimib.it

2012 Italian Stata Users Group meeting
Bologna
September 20-21, 2012

Outline

Outline

Outline

Outline

4 Measuring spatial proximity

Outline

4. Measuring spatial proximity

5. Detecting spatial autocorrelation
   - Overview
   - Measuring spatial autocorrelation
   - Global indices of spatial autocorrelation
   - Local indices of spatial autocorrelation

Outline

Introduction

Spatial data analysis in Stata

- Stata users can perform spatial data analysis using a variety of user-written commands published in the *Stata Technical Bulletin*, the *Stata Journal*, or the SSC Archive

Spatial data analysis in Stata

- Stata users can perform spatial data analysis using a variety of user-written commands published in the *Stata Technical Bulletin*, the *Stata Journal*, or the SSC Archive
- In this talk, I will briefly illustrate the use of six such commands: `spmap`, `spgrid`, `spkde`, `spatwmat`, `spatgsa`, and `spatlsa`

Spatial data analysis in Stata

- Stata users can perform spatial data analysis using a variety of user-written commands published in the *Stata Technical Bulletin*, the *Stata Journal*, or the SSC Archive
- In this talk, I will briefly illustrate the use of six such commands: `spmap`, `spgrid`, `spkde`, `spatwmat`, `spatgsa`, and `spatlsa`
- I will also mention a pair of Stata commands/suites for fitting spatial regression models: `spatreg` and `sppack`

Spatial data: a discrete view

- For simplicity, let us represent **space** as a plane, i.e., as a flat two-dimensional surface

Spatial data: a discrete view

- For simplicity, let us represent **space** as a plane, i.e., as a flat two-dimensional surface
- In spatial data analysis, we can distinguish two conceptions of space (Bailey and Gatrell 1995: 18):
  - *Entity view*: Space as an area filled with a set of discrete objects
  - *Field view*: Space as an area covered with essentially continuous surfaces

Spatial data: a discrete view

- For simplicity, let us represent **space** as a plane, i.e., as a flat two-dimensional surface
- In spatial data analysis, we can distinguish two conceptions of space (Bailey and Gatrell 1995: 18):
  - *Entity view*: Space as an area filled with a set of discrete objects
  - *Field view*: Space as an area covered with essentially continuous surfaces
- Here we take the former view and define **spatial data** as information regarding a given set of discrete spatial objects located within a study area $\mathcal{A}$

Attributes of spatial objects

- Information about spatial objects can be classified into two categories:
  - Spatial attributes
  - Non-spatial attributes

Attributes of spatial objects

- Information about spatial objects can be classified into two categories:
  - Spatial attributes
  - Non-spatial attributes
- The **spatial attributes** of a spatial object consist of one or more pairs of coordinates that represent its shape and/or its location within the study area

Attributes of spatial objects

- Information about spatial objects can be classified into two categories:
  - Spatial attributes
  - Non-spatial attributes
- The **spatial attributes** of a spatial object consist of one or more pairs of coordinates that represent its shape and/or its location within the study area
- The **non-spatial attributes** of a spatial object consist of its additional features that are relevant to the analysis at hand

Types of spatial objects

- According to their spatial attributes, **spatial objects** can be classified into several types

Types of spatial objects

- According to their spatial attributes, **spatial objects** can be classified into several types
- Here, we focus on two basic types:
  - Points (point data)
  - Polygons (area data)

Points

- A point $\mathbf{s}_i$ is a zero-dimensional spatial object located within study area $\mathcal{A}$ at coordinates (s_{i1}, s_{i2})

Points

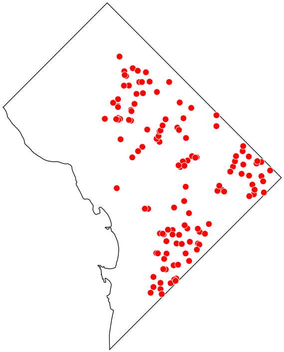
- A point $\mathbf{s}_i$ is a zero-dimensional spatial object located within study area $\mathcal{A}$ at coordinates (s_{i1}, s_{i2})
- Points can represent several kinds of real entities, e.g., dwellings, buildings, places where specific events took place, pollution sources, trees

Points

- A point $\mathbf{s}_i$ is a zero-dimensional spatial object located within study area $\mathcal{A}$ at coordinates (s_{i1}, s_{i2})
- Points can represent several kinds of real entities, e.g., dwellings, buildings, places where specific events took place, pollution sources, trees

Points

- A point $\mathbf{s}_i$ is a zero-dimensional spatial object located within study area $\mathcal{A}$ at coordinates (s_{i1}, s_{i2})
- Points can represent several kinds of real entities, e.g., dwellings, buildings, places where specific events took place, pollution sources, trees



Polygons

- A polygon $\mathbf{r}_i$ is a *region* of study area $\mathcal{A}$ bounded by a closed polygonal chain whose $M \geq 4$ vertices are defined by the coordinate set $\{(r_{i1(1)}, r_{i2(1)}), (r_{i1(2)}, r_{i2(2)}), \ldots, (r_{i1(m)}, r_{i2(m)}), \ldots, (r_{i1(M)}, r_{i2(M)})\}$, where $r_{i1(1)} = r_{i1(M)}$ and $r_{i2(1)} = r_{i2(M)}$

Polygons

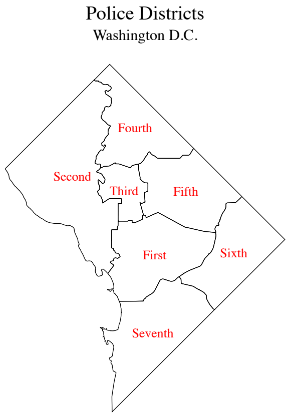
- A polygon $\mathbf{r}_i$ is a *region* of study area $\mathcal{A}$ bounded by a closed polygonal chain whose $M \geq 4$ vertices are defined by the coordinate set $\{(r_{i1(1)}, r_{i2(1)}), (r_{i2(2)}, r_{i2(2)}), \ldots, (r_{i1(m)}, r_{i2(m)}), \ldots, (r_{i1(M)}, r_{i2(M)})\}$, where $r_{i1(1)} = r_{i1(M)}$ and $r_{i2(1)} = r_{i2(M)}$
- Polygons can represent several kinds of real entities, e.g., states, provinces, counties, census tracts, electoral districts, parks, lakes

Polygons

- A polygon $\mathbf{r}_i$ is a *region* of study area $\mathcal{A}$ bounded by a closed polygonal chain whose $M \geq 4$ vertices are defined by the coordinate set $\{(r_{i1(1)}, r_{i2(1)}), (r_{i1(2)}, r_{i2(2)}), \ldots, (r_{i1(m)}, r_{i2(m)}), \ldots, (r_{i1(M)}, r_{i2(M)})\}$, where $r_{i1(1)} = r_{i1(M)}$ and $r_{i2(1)} = r_{i2(M)}$
- Polygons can represent several kinds of real entities, e.g., states, provinces, counties, census tracts, electoral districts, parks, lakes

Polygons

- A polygon $\mathbf{r}_i$ is a *region* of study area $\mathcal{A}$ bounded by a closed polygonal chain whose $M \geq 4$ vertices are defined by the coordinate set $\{(r_{i1(1)}, r_{i2(1)}), (r_{i1(2)}, r_{i2(2)}), \ldots, (r_{i1(m)}, r_{i2(m)}), \ldots, (r_{i1(M)}, r_{i2(M)})\}$, where $r_{i1(1)} = r_{i1(M)}$ and $r_{i2(1)} = r_{i2(M)}$
- Polygons can represent several kinds of real entities, e.g., states, provinces, counties, census tracts, electoral districts, parks, lakes



Visualizing spatial data

Thematic maps

- Most analyses of spatial data have their natural starting point in displaying the information of interest by one or more **maps**
- If properly designed, maps can help the analyst to detect interesting patterns in the data, spatial relationships between two or more phenomena, unusual observations, and so on
- **Thematic maps** represent the spatial distribution of a phenomenon of interest within a given study area (Slocum *et al.* 2005)

Thematic maps in Stata

- Stata users can generate thematic maps using `spmap`, a user-written command freely available from the SSC Archive (latest version: 1.2.0)
- `spmap` is a very flexible command that allows for creating a large variety of thematic maps, from the simplest to the most complex
- While providing sensible defaults for most options and supoptions, `spmap` gives the user full control over the formatting of almost every map element, thus allowing the production of highly customized maps

Thematic maps in Stata

- In the following, I will show some examples on using `spmap` for creating common types of thematic maps:
  - Dot maps
  - Proportional symbol maps
  - Diagram maps
  - Choropleth maps
  - Multivariate maps

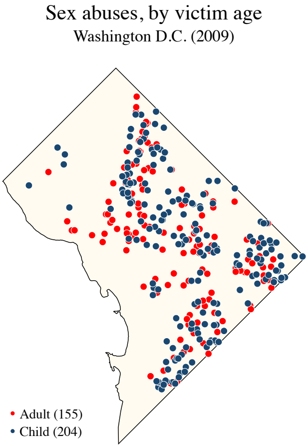
Dot maps

- A **dot map** shows the spatial distribution of a set of point spatial objects $\mathbf{S} \equiv \{\mathbf{s}_i; i = 1, \ldots, N\}$, i.e., their location within a given study area $\mathcal{A}$
- If the point spatial objects have variable attributes, it is possible to represent this information using symbols of different colors and/or of different shape

Dot maps: example 1

Spatial distribution of 359 cases of sex abuse, Washington D.C. (2009). Different colors are used to distinguish adult victims from child victims

```
use "Crime2009.dta", clear
generate _ID = _n
generate victim = method
recode victim (4/7=1)(17/18=2)(*=.)
label define victim 1 "Adult" 2 "Child"
label values victim victim
spmap using "Boundaries.dta", id(_ID) fcolor(eggshell)      ///
   point(x(x_coord) y(y_coord) select(keep if offense==6)   ///
     by(victim) size(*1.2) fcolor(red navy)                 ///
     ocolor(white ..) osize(*0.5 ..) legenda(on)            ///
     legcount)                                              ///
   legend(size(*1.8) rowgap(1.2))                           ///
   title("Sex abuses, by victim age", size(*1.2))           ///
   subtitle("Washington D.C. (2009)" " ", size(*1.2))
```



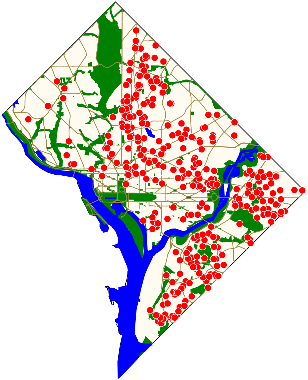
Dot maps: example 2

Spatial distribution of 359 cases of sex abuse, Washington D.C. (2009). Major roads, watercourses and parks are added to the map for reference

```
use "Crime2009.dta", clear
generate _ID = _n
spmap using "Boundaries.dta", id(_ID) fcolor(eggshell)     ///
   point(x(x_coord) y(y_coord) select(keep if offense==6)  ///
     size(*1.2) fcolor(red) ocolor(white) osize(*0.5))     ///
   polygon(data("Water&Parks.dta") by(type)                ///
     ocolor(none ..) fcolor(green blue))                   ///
   line(data("MajorRoads.dta") color(brown))               ///
   title("Sex abuses", size(*1.2))                         ///
   subtitle("Washington D.C. (2009)" " ", size(*1.2))
```

Sex abuses

Washington D.C. (2009)



Proportional symbol maps

- A **proportional symbol map** represents the values taken by a numeric variable of interest Y on a set of point spatial objects $\mathbf{S}$ located within a given study area $\mathcal{A}$
- Proportional symbol maps can be used with two types of point data (Slocum *et al.* 2005: 310):
  - **True point data** are measured at actual point locations
  - **Conceptual point data** are collected over a set of regions $\mathbf{R} \equiv \{\mathbf{r}_i; i = 1, \ldots, N\}$, but are conceived as being located at representative points within the regions, typically at their centroids
- The area of each point symbol is sized in direct proportion to the corresponding value of Y

Proportional symbol maps: example

Mean family income in the seven Police Districts of Washington D.C. (2000)

```
use "PoliceDistricts-Data.dta", clear
generate Y = income_ma/1000
format Y %4.1f
spmap using "PoliceDistricts-Coordinates.dta", id(id)        ///
   fcolor(eggshell)                                          ///
   point(x(x_coord) y(y_coord) proportional(Y) fcolor(red)   ///
     ocolor(white) size(*3.5))                               ///
   label(x(x_coord) ycoord(y_coord) label(Y) color(white)    ///
     size(*1.4))                                             ///
   title("Mean family income (in thousands of US dollars)") ///
   subtitle("Washington D.C. (2000)" " ")
```

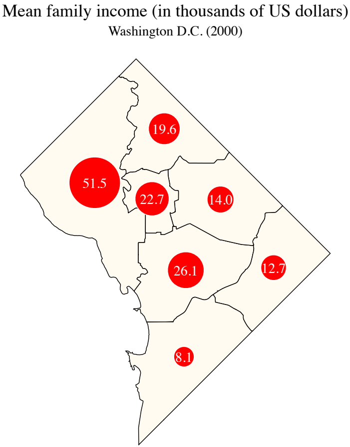


Diagram maps

- A **diagram map** follows the same logic as a proportional symbol map, but represents the values of the variable of interest using bar charts, pie charts, or other types of diagram
- The use of pie charts allows to display the spatial distribution of compositional data, i.e., of two or more numeric variables that represent parts of a whole

Diagram maps: example 1

Mean family income in the seven Police Districts of Washington D.C. (2000). Data are represented by framed-rectangle charts, with the overall mean income as the reference value

```
use "PoliceDistricts-Data.dta", clear
spmap using "PoliceDistricts-Coordinates.dta", id(id)      ///
   fcolor(eggshell)                                        ///
   diagram(var(income_ma) refweight(poptot) fcolor(green)  ///
      x(x_coord) y(y_coord) size(1.3))                     ///
   title("Mean family income")                             ///
   subtitle("Washington D.C. (2000)" " ")
```

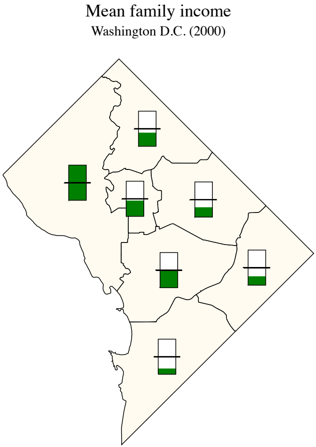
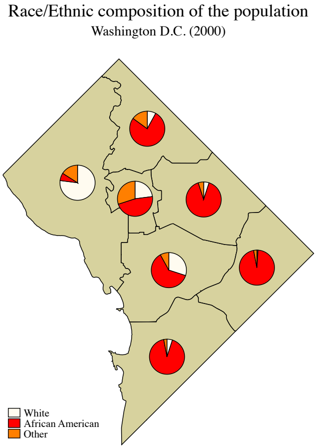


Diagram maps: example 2

Race/Ethnic composition of the population of the seven Police Districts of Washington D.C. (2000). Data are represented by pie charts

```
use "PoliceDistricts-Data.dta", clear
generate white_pct = pop_white/poptot*100
generate afroam_pct = pop_afroam/poptot*100
generate other_pct = pop_other/poptot*100
label variable white_pct "White"
label variable afroam_pct "African American"
label variable other_pct "Other"
spmap using "PoliceDistricts-Coordinates.dta", id(id)       ///
   fcolor(stone)                                             ///
   diagram(var(white_pct afroam_pct other_pct) x(x_coord)   ///
     y(y_coord) fcolor(eggshell red orange) size(1.3)        ///
     legenda(on))                                            ///
   legend(size(*1.4))                                        ///
   title("Race/Ethnic composition of the population")       ///
   subtitle("Washington D.C. (2000)" " ")
```



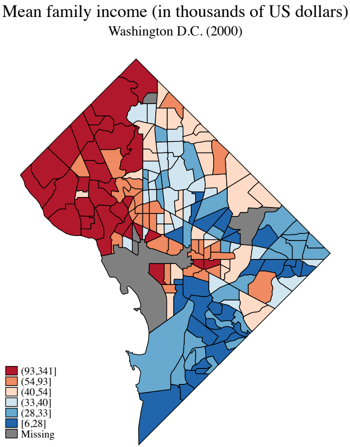
Choropleth maps

- A **choropleth map** displays the values taken by a variable of interest Y on a set of regions $\mathbf{R}$ within a given study area $\mathcal{A}$
- When Y is numeric, each region is colored or shaded according to a discrete scale based on its value on Y
- The number of classes k that make up the discrete scale, and the corresponding class breaks, can be based on several different criteria – e.g., quantiles, equal intervals, boxplot, standard deviates

Choropleth maps: example 1

Mean family income in the 188 Census Tracts of Washington D.C. (2000). Income is divided into six classes based on the *quantiles* method

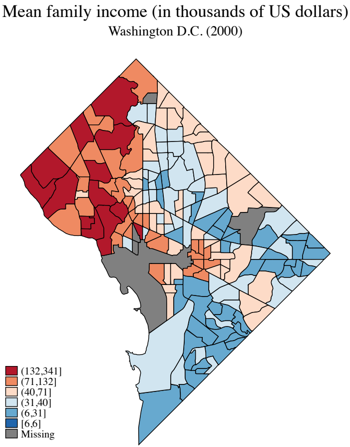
```
use "Census2000-Data.dta", clear
generate Y = income_ma/1000
format Y %3.0f
spmap Y using "Census2000-Coordinates.dta", id(id)          ///
   clnumber(6) clmethod(quantile) fcolor(BuRd)              ///
   ndfcolor(gs8) ndlab("Missing")                           ///
   legend(size(*1.4))                                       ///
   title("Mean family income (in thousands of US dollars)") ///
   subtitle("Washington D.C. (2000)" " ")
```



Choropleth maps: example 2

Mean family income in the 188 Census Tracts of Washington D.C. (2000). Income is divided into six classes based on the *boxplot* method

```
use "Census2000-Data.dta", clear
generate Y = income_ma/1000
format Y %3.0f
spmap Y using "Census2000-Coordinates.dta", id(id)           ///
   clnumber(6) clmethod(boxplot) fcolor(BuRd)                ///
   ndfcolor(gs8) ndlab("Missing")                            ///
   legend(size(*1.4))                                        ///
   title("Mean family income (in thousands of US dollars)") ///
   subtitle("Washington D.C. (2000)" " ")
```



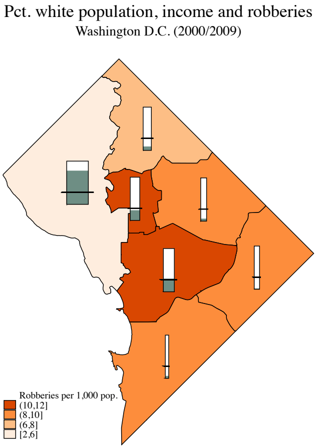
Multivariate maps

- A **multivariate map** combines several types of thematic mapping to simultaneously display the spatial distribution of multiple phenomena within a given study area $\mathcal{A}$

Multivariate maps: example

The map shows the relationship between pct. white population (represented by framed-rectangle charts), mean family income (represented by the width of framed-rectangle charts) and robbery rate (represented by shades of color) across the seven Police Districts of Washington D.C. (2000/2009)

```
use "PoliceDistricts-Data.dta", clear
generate Y = pop_white/poptot*100
format Y %2.0f
spmap Y using "PoliceDistricts-Coordinates.dta", id(id)      ///
   clmethod(custom) clbreaks(0 25 50 75 100) fcolor(YlGn)    ///
   legtit("Pct. white population")                           ///
   diagram(var(income_ma) refweight(poptot) fcolor(red)      ///
     x(x_coord) y(y_coord) size(1.3))                        ///
   legend(size(*1.4))                                        ///
   title("Mean family income and pct. white population")     ///
   subtitle("Washington D.C. (2000)" " ")
```



Exploring spatial point patterns

Two-dimensional spatial point patterns

- A **two-dimensional spatial point pattern** can be defined as a set of N point spatial objects $\mathbf{S}$ located within a given study area $\mathcal{A}$
- Usually, each point $\mathbf{s}_i \in \mathbf{S}$ represents a real entity of some kind: people, events, sites, buildings, plants, cases of a disease, etc.
- Alternatively, each point $\mathbf{s}_i$ represents the centroid of a region
- Points $\mathbf{s}_i$ are referred to as the *data points*

Two-dimensional spatial point patterns

- In the analysis of spatial point patterns, we are often interested in determining whether the observed data points exhibit some form of *clustering*, as opposed to being distributed uniformly within $\mathcal{A}$

Two-dimensional spatial point patterns

- In the analysis of spatial point patterns, we are often interested in determining whether the observed data points exhibit some form of *clustering*, as opposed to being distributed uniformly within $\mathcal{A}$
- To explore the possibility of point clustering, it may be useful to describe the spatial point pattern of interest by means of its probability density function $p(\mathbf{s})$ and/or its intensity function $\lambda(\mathbf{s})$ (Waller and Gotway 2004)

Two-dimensional spatial point patterns

- The **probability density function** $p(\mathbf{s})$ defines the probability of observing an object per unit area at location $\mathbf{s} \in \mathcal{A}$
- The **intensity function** $\lambda(\mathbf{s})$ defines the expected number of objects per unit area at location $\mathbf{s} \in \mathcal{A}$
- The probability density function and the intensity function differ only by a constant of proportionality

Kernel estimators

- Both the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of a two-dimensional spatial point pattern can be estimated by means of nonparametric estimators, e.g., kernel estimators (Waller and Gotway 2004)

Kernel estimators

- Both the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of a two-dimensional spatial point pattern can be estimated by means of nonparametric estimators, e.g., kernel estimators (Waller and Gotway 2004)
- **Kernel estimators** are used to generate a spatially smooth estimate of $p(\mathbf{s})$ and/or $\lambda(\mathbf{s})$ at a fine grid of points $\mathbf{s}_g$ $(g = 1, ..., G)$ covering the study area $\mathcal{A}$

Kernel estimators

- Both the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of a two-dimensional spatial point pattern can be estimated by means of nonparametric estimators, e.g., kernel estimators (Waller and Gotway 2004)
- **Kernel estimators** are used to generate a spatially smooth estimate of $p(\mathbf{s})$ and/or $\lambda(\mathbf{s})$ at a fine grid of points $\mathbf{s}_g$ $(g = 1, ..., G)$ covering the study area $\mathcal{A}$
- In the context of spatial data analysis, a **grid** is a regular tessellation of the study area $\mathcal{A}$ that divides it into a set of G contiguous cells whose centers are referred to as the *grid points* and denoted by $\mathbf{s}_g$

Kernel estimation in Stata

- Stata users can generate kernel estimates of the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ using two user-written commands freely available from the SSC Archive: `spgrid` and `spkde`

Kernel estimation in Stata

- Stata users can generate kernel estimates of the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ using two user-written commands freely available from the SSC Archive: `spgrid` and `spkde`
- `spgrid` (latest version: 1.0.1) generates several kinds of two-dimensional grids covering rectangular or irregular study areas

Kernel estimation in Stata

- Stata users can generate kernel estimates of the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ using two user-written commands freely available from the SSC Archive: `spgrid` and `spkde`
- `spgrid` (latest version: 1.0.1) generates several kinds of two-dimensional grids covering rectangular or irregular study areas
- `spkde` (latest version: 1.0.0) implements a variety of kernel estimators of $p(\mathbf{s})$ and $\lambda(\mathbf{s})$

Kernel estimation in Stata

- Stata users can generate kernel estimates of the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ using two user-written commands freely available from the SSC Archive: `spgrid` and `spkde`
- `spgrid` (latest version: 1.0.1) generates several kinds of two-dimensional grids covering rectangular or irregular study areas
- `spkde` (latest version: 1.0.0) implements a variety of kernel estimators of $p(\mathbf{s})$ and $\lambda(\mathbf{s})$
- `spmap` can then be used to visualize the kernel estimates generated by `spgrid` and `spkde`

Kernel estimation: example

Our purpose is to estimate the probability density function of a set of 139 points representing the **homicides** committed in Washington D.C. in 2009

Kernel estimation: example

Step 1

We use **spgrid** to generate a grid covering the area of Washington D.C. We choose a relatively fine grid resolution (grid cell width = 200 meters). **spmap** is used to display the grid

Kernel estimation: example

Step 1

We use **spgrid** to generate a grid covering the area of Washington D.C. We choose a relatively fine grid resolution (grid cell width = 200 meters). **spmap** is used to display the grid

```
spgrid using "Boundaries.dta", resolution(w200)    ///
   dots compress unit(meters) cells("ctemp.dta")   ///
   points("ptemp.dta") replace

use "ptemp.dta", clear
spmap using "ctemp.dta", id(spgrid_id)
```

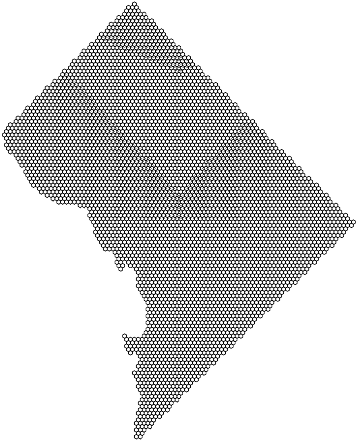
Kernel estimation: example

Step 1

We use **spgrid** to generate a grid covering the area of Washington D.C. We choose a relatively fine grid resolution (grid cell width = 200 meters). **spmap** is used to display the grid

```
spgrid using "Boundaries.dta", resolution(w200)    ///
   dots compress unit(meters) cells("ctemp.dta")   ///
   points("ptemp.dta") replace

use "ptemp.dta", clear
spmap using "ctemp.dta", id(spgrid_id)
```



Kernel estimation: example

Step 2

We use **`spkde`** to generate kernel estimates of the probability distribution of homicides in Washington D.C. We choose a quartic kernel function with fixed bandwidth equal to 1,000 meters and edge correction. **`spmap`** is used to display the results

Kernel estimation: example

Step 2

We use **spkde** to generate kernel estimates of the probability distribution of homicides in Washington D.C. We choose a quartic kernel function with fixed bandwidth equal to 1,000 meters and edge correction. **spmap** is used to display the results

```
use "Crime2009.dta", clear
keep if offense==4
spkde using "ptemp.dta", x(x_coord) y(y_coord)   ///
   kernel(quartic) bandwidth(fbw) fbw(1000)      ///
   edgecorrect dots saving("kde.dta", replace)

use "kde.dta", clear
spmap p using "ctemp.dta", id(spgrid_id) clmethod(quantile) ///
   clnumber(20) fcolor(Rainbow) ocolor(none ..) legend(off) ///
   title("Homicides", size(*1.2))                           ///
   subtitle("Washington D.C. (2009)" " ", size(*1.2))
```

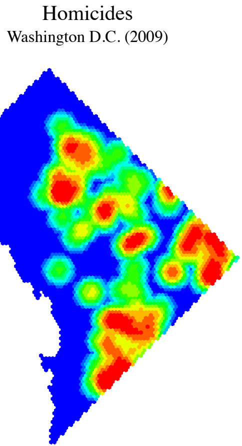
Kernel estimation: example

Step 2

We use **spkde** to generate kernel estimates of the probability distribution of homicides in Washington D.C. We choose a quartic kernel function with fixed bandwidth equal to 1,000 meters and edge correction. **spmap** is used to display the results

```
use "Crime2009.dta", clear
keep if offense==4
spkde using "ptemp.dta", x(x_coord) y(y_coord)   ///
   kernel(quartic) bandwidth(fbw) fbw(1000)      ///
   edgecorrect dots saving("kde.dta", replace)

use "kde.dta", clear
spmap p using "ctemp.dta", id(spgrid_id) clmethod(quantile) ///
   clnumber(20) fcolor(Rainbow) ocolor(none ..) legend(off) ///
   title("Homicides", size(*1.2))                        ///
   subtitle("Washington D.C. (2009)" " ", size(*1.2))
```



Kernel estimation: example

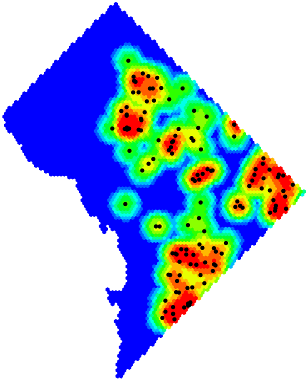
Step 2

We use **spkde** to generate kernel estimates of the probability distribution of homicides in Washington D.C. We choose a quartic kernel function with fixed bandwidth equal to 1,000 meters and edge correction. **spmap** is used to display the results

```
use "Crime2009.dta", clear
keep if offense==4
spkde using "ptemp.dta", x(x_coord) y(y_coord)   ///
   kernel(quartic) bandwidth(fbw) fbw(1000)      ///
   edgecorrect dots saving("kde.dta", replace)

use "kde.dta", clear
spmap p using "ctemp.dta", id(spgrid_id) clmethod(quantile) ///
   clnumber(20) fcolor(Rainbow) ocolor(none ..) legend(off) ///
   title("Homicides", size(*1.2))                           ///
   subtitle("Washington D.C. (2009)" " ", size(*1.2))
```

Homicides
Washington D.C. (2009)



Measuring spatial proximity

Spatial weights matrix

- Most spatial data analyses require that the degree of **spatial proximity** among the spatial objects of interest be expressed in some way

Spatial weights matrix

- Most spatial data analyses require that the degree of **spatial proximity** among the spatial objects of interest be expressed in some way
- Typically, the degree of spatial proximity among a given set of N spatial objects is represented by a $N \times N$ matrix called **spatial weights matrix** and denoted by $\mathbf{W}$

Spatial weights matrix

- Most spatial data analyses require that the degree of **spatial proximity** among the spatial objects of interest be expressed in some way
- Typically, the degree of spatial proximity among a given set of N spatial objects is represented by a $N \times N$ matrix called **spatial weights matrix** and denoted by $\mathbf{W}$
- Each element (i, j) of $\mathbf{W}$ – which we denote by w_{ij} – expresses the degree of spatial proximity between the pair of objects i and j

Spatial weights matrix

- Most spatial data analyses require that the degree of **spatial proximity** among the spatial objects of interest be expressed in some way
- Typically, the degree of spatial proximity among a given set of N spatial objects is represented by a $N \times N$ matrix called **spatial weights matrix** and denoted by $\mathbf{W}$
- Each element (i, j) of $\mathbf{W}$ – which we denote by w_{ij} – expresses the degree of spatial proximity between the pair of objects i and j
- Depending on the application, the N main diagonal elements of $\mathbf{W}$ are assigned value $w_{ii} = 0$ or value $w_{ii} > 0$

Spatial weights matrix

- A common variant of **W** is the **row-standardized spatial weights matrix $\mathbf{W}_{std}$**, whose elements are defined as follows:

$$w_{ij}^{std} = \frac{w_{ij}}{\sum_{j=1}^{N} w_{ij}}$$

Spatial weights matrices in Stata

- Stata users can generate several kinds of spatial weights matrices using `spatwmat`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)

Spatial weights matrices in Stata

- Stata users can generate several kinds of spatial weights matrices using `spatwmat`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)
- `spatwmat` (latest version: 1.0) imports or generates from scratch the spatial weights matrices required by other commands for spatial data analysis (see below)

Detecting spatial autocorrelation

Spatial autocorrelation

- Forty years ago, the geographer and statistician Waldo Tobler formulated the *first law of geography*: "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970: 234)

Spatial autocorrelation

- Forty years ago, the geographer and statistician Waldo Tobler formulated the *first law of geography*: "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970: 234)
- This "law" defines the statistical concept of (positive) **spatial autocorrelation**, according to which two or more objects that are spatially close tend to be more similar to each other – with respect to a given attribute Y – than are spatially distant objects

Spatial autocorrelation

- Forty years ago, the geographer and statistician Waldo Tobler formulated the *first law of geography*: "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970: 234)
- This "law" defines the statistical concept of (positive) **spatial autocorrelation**, according to which two or more objects that are spatially close tend to be more similar to each other – with respect to a given attribute Y – than are spatially distant objects
- In general, spatial autocorrelation implies **spatial custering**, i.e., the existence of sub-areas of the study area where the attribute of interest Y takes higher than average values (*hot spots*) or lower than average values (*cold spots*)

Indices of spatial autocorrelation

- We consider measures of spatial autocorrelation that apply to *area data*

Indices of spatial autocorrelation

- We consider measures of spatial autocorrelation that apply to *area data*
- Measures of spatial autocorrelation can be classified into two broad categories:

Indices of spatial autocorrelation

- We consider measures of spatial autocorrelation that apply to *area data*
- Measures of spatial autocorrelation can be classified into two broad categories:
  - Global indices of spatial autocorrelation

Indices of spatial autocorrelation

- We consider measures of spatial autocorrelation that apply to *area data*
- Measures of spatial autocorrelation can be classified into two broad categories:
  - Global indices of spatial autocorrelation
  - Local indices of spatial autocorrelation

Global indices of spatial autocorrelation

- A **global index of spatial autocorrelation** expresses the overall degree of similarity between spatially close regions observed in a given study area $\mathcal{A}$ with respect to a numeric variable Y (Pfeiffer *et al.* 2008)

Global indices of spatial autocorrelation

- A **global index of spatial autocorrelation** expresses the overall degree of similarity between spatially close regions observed in a given study area $\mathcal{A}$ with respect to a numeric variable Y (Pfeiffer *et al.* 2008)
- Since global indices of spatial autocorrelation summarize the phenomenon of interest in a single value, they are intended not so much for identifying specific spatial clusters, as for detecting the presence of a general tendency to clustering within the study area

Global indices of spatial autocorrelation in Stata

- Stata users can compute global indices of spatial autocorrelation using `spatgsa`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)
- `spatgsa` (latest version: 1.0) computes three global indices of spatial autocorrelation: Moran's I, Getis and Ord's G, and Geary's c. For each index and each numeric variable of interest, `spatgsa` computes and displays in tabular form the value of the index itself, the expected value of the index under the null hypothesis of no global spatial autocorrelation, the standard deviation of the index, the z-value, and the corresponding one- or two-tailed p-value

Global indices of spatial autocorrelation: example

- **Study area**: Ohio
- **Regions**: 88 counties
- **Variables of interest**:
  - Pct. population aged 18+ with poor-to-fair health status (`pct_poorhealth`)
  - Pct. population aged 18+ currently smoking (`pct_currsmoker`)
  - Pct. population aged 18+ ever diagnosed with high blood pressure (`pct_hibloodprs`)
  - Pct. population aged 18+ obese (`pct_obese`)

Global indices of spatial autocorrelation: example

Step 1

We use `spatwmat` to import an existing binary spatial weights matrix – stored in the Stata dataset `Counties-Contiguity.dta` – and convert it into a properly formatted row-standardized spatial weights matrix `Ws`

Global indices of spatial autocorrelation: example

Step 1

We use `spatwmat` to import an existing binary spatial weights matrix – stored in the Stata dataset `Counties-Contiguity.dta` – and convert it into a properly formatted row-standardized spatial weights matrix `Ws`

```
spatwmat using "Counties-Contiguity.dta",   ///
   name(Ws) standardize
```

Global indices of spatial autocorrelation: example

```
The following matrix has been created:

1. Imported binary weights matrix Ws (row-standardized)
   Dimension: 88x88
```

Global indices of spatial autocorrelation: example

Step 2

We use `spatgsa` with the spatial weights matrix `Ws` to compute Moran's I on the variables of interest

Global indices of spatial autocorrelation: example

Step 2

We use `spatgsa` with the spatial weights matrix `Ws` to compute Moran's I on the variables of interest

```
use "Counties-Data.dta", clear
spatgsa pct_poorhealth pct_currsmoker pct_hibloodprs   ///
   pct_obese, w(Ws) moran
```

Global indices of spatial autocorrelation: example

Measures of global spatial autocorrelation

Weights matrix

Name: **Ws**
Type: **Imported (binary)**
Row-standardized: **Yes**

Moran's I

Variables	I	E(I)	sd(I)	z	p-value*
pct_poorhealth	**0.399**	**-0.011**	**0.065**	**6.337**	**0.000**
pct_currsmoker	**0.339**	**-0.011**	**0.065**	**5.367**	**0.000**
pct_hibloodprs	**0.126**	**-0.011**	**0.065**	**2.119**	**0.017**
pct_obese	**0.167**	**-0.011**	**0.065**	**2.730**	**0.003**

*1-tail test

Local indices of spatial autocorrelation

- A **local index of spatial autocorrelation** expresses, for each region $\mathbf{r}_i$ of a given study area $\mathcal{A}$, the degree of similarity between that region and its neighboring regions with respect to a numeric variable Y (Pfeiffer *et al.* 2008)

Local indices of spatial autocorrelation

- A **local index of spatial autocorrelation** expresses, for each region $\mathbf{r}_i$ of a given study area $\mathcal{A}$, the degree of similarity between that region and its neighboring regions with respect to a numeric variable Y (Pfeiffer *et al.* 2008)
- The local indices of spatial autocorrelation are derived from the corresponding global indices and share their fundamental properties

Local indices of spatial autocorrelation in Stata

- Stata users can compute local indices of spatial autocorrelation using `spatlsa`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)

Local indices of spatial autocorrelation in Stata

- Stata users can compute local indices of spatial autocorrelation using `spatlsa`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)
- `spatlsa` (latest version: 1.0) computes four indices of spatial autocorrelation: Moran's I_i, Getis and Ord's G_i and $G_i^{\star}$, and Geary's c_i. For each index and each region in the analysis, `spatlsa` computes and displays in tabular form the value of the index itself, the expected value of the index under the null hypothesis of no local spatial autocorrelation, the standard deviation of the index, the z-value, and the corresponding one- or two-tailed p-value

Local indices of spatial autocorrelation: example

- **Study area**: Ohio
- **Regions**: 88 counties
- **Variable of interest**: Pct. population aged 18+ with poor-to-fair health status (`pct_poorhealth`)

Local indices of spatial autocorrelation: example

We use `spatlsa` with the standardized spatial weights matrix `Ws` – previously generated by `spatwmat` – to compute Moran's I_i on the variable of interest. In the output, counties are sorted by z-value

Local indices of spatial autocorrelation: example

We use `spatlsa` with the standardized spatial weights matrix `Ws` – previously generated by `spatwmat` – to compute Moran's I_i on the variable of interest. In the output, counties are sorted by z-value

```
spatwmat using "Counties-Contiguity.dta", name(Ws) standardize
use "Counties-Data.dta", clear
spatlsa pct_poorhealth, w(Ws) moran id(name) sort
```

Local indices of spatial autocorrelation: example

Measures of local spatial autocorrelation

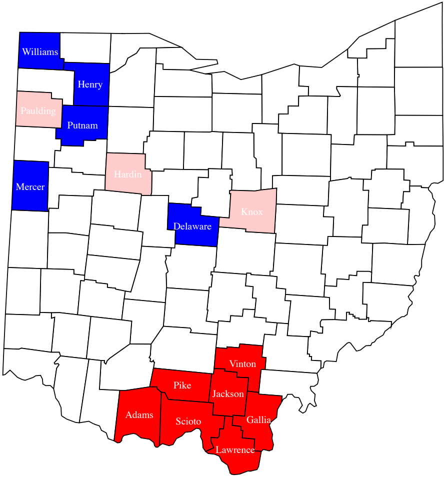
(*Output omitted*)

Moran's Ii (**Poor-to-fair health status (pct. pop 18+)**)

name	Ii	E(Ii)	sd(Ii)	z	p-value*
Knox	-0.816	-0.011	0.358	-2.246	0.012
Hardin	-0.760	-0.011	0.358	-2.090	0.018
Paulding	-1.089	-0.011	0.560	-1.924	0.027
Licking	-0.457	-0.011	0.358	-1.244	0.107
(*Output omitted*)					
Hancock	0.545	-0.011	0.358	1.555	0.060
Williams	0.927	-0.011	0.560	1.675	0.047
Delaware	0.677	-0.011	0.389	1.769	0.038
Mercer	0.908	-0.011	0.482	1.906	0.028
Putnam	0.949	-0.011	0.358	2.682	0.004
Henry	1.087	-0.011	0.358	3.067	0.001
Vinton	1.246	-0.011	0.389	3.230	0.001
Gallia	2.433	-0.011	0.482	5.069	0.000
Pike	2.197	-0.011	0.429	5.150	0.000
Adams	3.578	-0.011	0.482	7.442	0.000
Lawrence	5.503	-0.011	0.560	9.844	0.000
Jackson	3.911	-0.011	0.389	10.077	0.000
Scioto	5.400	-0.011	0.482	11.220	0.000

*1-tail test

Local indices of spatial autocorrelation: example



Fitting spatial regression models

Spatial regression

- The aim of **spatial regression** is to estimate the relationship between an outcome variable of interest Y and one or more predictors X, taking into proper account the spatial dependence among observations

Spatial regression

- The aim of **spatial regression** is to estimate the relationship between an outcome variable of interest Y and one or more predictors X, taking into proper account the spatial dependence among observations
- Two types of spatial dependence are most commonly considered (Ward and Gleditsch 2008):
  - A spatial autoregressive process in the error term
  - A spatial autoregressive process in the outcome variable

Spatial error model

- The first type of spatial dependence is represented by the **spatial error model**:

$$Y = \mathbf{X}\beta + \lambda\mathbf{W}\xi + \epsilon$$

where Y denotes an $N \times 1$ vector of observations on the outcome variable; $\mathbf{X}$ denotes an $N \times j$ matrix of observations on the predictor variables; β denotes a $j \times 1$ vector of regression coefficients; λ denotes the spatial autoregressive parameter; $\mathbf{W}$ denotes the $N \times N$ spatial weights matrix; ξ denotes an $N \times 1$ vector of spatial errors; and ϵ denotes an $N \times 1$ vector of normally distributed, homoskedastic, and uncorrelated errors

Spatial lag model

- The second type of spatial dependence is represented by the **spatial lag model**:

$$Y = \mathbf{X}\beta + \rho\mathbf{W}Y + \epsilon$$

where ρ denotes the spatial autoregressive parameter; and all the other terms are defined as above

Spatial error vs. spatial lag models

- The *spatial lag model* treats spatial dependence as substance, assuming that the value taken by Y in each region is affected by the values taken by Y in the neighboring regions

Spatial error vs. spatial lag models

- The *spatial lag model* treats spatial dependence as substance, assuming that the value taken by Y in each region is affected by the values taken by Y in the neighboring regions
- On the other hand, the *spatial error model* treats spatial dependence as nuisance

Spatial regression in Stata

- Stata users can fit spatial error and spatial lag models using `spatreg`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)

Spatial regression in Stata

- Stata users can fit spatial error and spatial lag models using `spatreg`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)
- An excellent alternative to `spatreg` is represented by `sppack`, a suite of Stata commands – freely available from the SSC Archive – written by David M. Drukker, Hua Peng, Ingmar Prucha, and Rafal Raciborski

Spatial regression in Stata

- Stata users can fit spatial error and spatial lag models using `spatreg`, a user-written command published in the *Stata Technical Bulletin* (Pisati 2001)
- An excellent alternative to `spatreg` is represented by `sppack`, a suite of Stata commands – freely available from the SSC Archive – written by David M. Drukker, Hua Peng, Ingmar Prucha, and Rafal Raciborski
- `sppack` is faster and more flexible than `spatreg`. Moreover, while `spatreg` is limited to the analysis of small sets of observations, `sppack` can deal with very large Ns

References

References

- Bailey, T.C. and A.C. Gatrell. 1995. *Interactive Spatial Data Analysis*. Harlow: Longman.
- Pfeiffer, D., Robinson, T., Stevenson, M., Stevens, K., Rogers, D. and A. Clements. 2008. *Spatial Analysis in Epidemiology*. Oxford: Oxford University Press.
- Pisati, M. 2001. sg162: Tools for spatial data analysis. *Stata Technical Bulletin* 60: 21–37. In *Stata Technical Bulletin Reprints*, vol. 10, 277–298. College Station, TX: Stata Press.
- Slocum, T.A., McMaster, R.B., Kessler, F.C. and H.H. Howard. 2005. *Thematic Cartography and Geographic Visualization*. 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall.
- Tobler, W.R. 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography* 46: 234–240.
- Waller, L.A. and C.A. Gotway. 2004. *Applied Spatial Statistics for Public Health Data*. Hoboken, NJ: Wiley.
- Ward, M.D. and K.S. Gleditsch. 2008. *Spatial Regression Models*. Thousand Oaks, CA: Sage.