Tools for the exploratory analysis of two-dimensional spatial point patterns

An introduction to `spgrid` and `spkde`

Maurizio Pisati

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

14th UK Stata Users Group meeting
Cass Business School (London), September 8-9, 2008

Outline

The programs

- The purpose of this talk is to introduce `spgrid` and `spkde`, two novel user-written Stata programs for the exploratory analysis of two-dimensional spatial point patterns

The programs

- The purpose of this talk is to introduce `spgrid` and `spkde`, two novel user-written Stata programs for the exploratory analysis of two-dimensional spatial point patterns
- `spgrid` generates several kinds of two-dimensional grids covering rectangular or irregular study regions

The programs

- The purpose of this talk is to introduce `spgrid` and `spkde`, two novel user-written Stata programs for the exploratory analysis of two-dimensional spatial point patterns
- `spgrid` generates several kinds of two-dimensional grids covering rectangular or irregular study regions
- `spkde` implements a variety of nonparametric kernel-based estimators of the probability density function and the intensity function of two-dimensional spatial point patterns

Two-dimensional spatial point patterns

- A two-dimensional spatial point pattern **S** can be defined as a set of points $\mathbf{s}_i$ $(i = 1, ..., n)$ located in a two-dimensional study region $\mathcal{R}$ at coordinates (s_{i1}, s_{i2})

Two-dimensional spatial point patterns

- A two-dimensional spatial point pattern **S** can be defined as a set of points $\mathbf{s}_i$ $(i = 1, ..., n)$ located in a two-dimensional study region $\mathcal{R}$ at coordinates (s_{i1}, s_{i2})
- Each point $\mathbf{s}_i$ represents the location in $\mathcal{R}$ of an "object" of some kind: people, events, sites, buildings, plants, cases of a disease, etc.

Two-dimensional spatial point patterns

- A two-dimensional spatial point pattern **S** can be defined as a set of points $\mathbf{s}_i$ $(i = 1, ..., n)$ located in a two-dimensional study region $\mathcal{R}$ at coordinates (s_{i1}, s_{i2})
- Each point $\mathbf{s}_i$ represents the location in $\mathcal{R}$ of an "object" of some kind: people, events, sites, buildings, plants, cases of a disease, etc.
- Points $\mathbf{s}_i$ will be referred to as the *data points*

Two-dimensional spatial point patterns

- A two-dimensional spatial point pattern **S** can be defined as a set of points $\mathbf{s}_i$ $(i = 1, ..., n)$ located in a two-dimensional study region $\mathcal{R}$ at coordinates (s_{i1}, s_{i2})
- Each point $\mathbf{s}_i$ represents the location in $\mathcal{R}$ of an "object" of some kind: people, events, sites, buildings, plants, cases of a disease, etc.
- Points $\mathbf{s}_i$ will be referred to as the *data points*

Two-dimensional spatial point patterns

- A two-dimensional spatial point pattern **S** can be defined as a set of points $\mathbf{s}_i$ $(i = 1, ..., n)$ located in a two-dimensional study region $\mathcal{R}$ at coordinates (s_{i1}, s_{i2})
- Each point $\mathbf{s}_i$ represents the location in $\mathcal{R}$ of an "object" of some kind: people, events, sites, buildings, plants, cases of a disease, etc.
- Points $\mathbf{s}_i$ will be 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 $\mathcal{R}$

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 $\mathcal{R}$
- 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 $\lambda(\mathbf{s})$

Two-dimensional spatial point patterns

- The probability density function $p(\mathbf{s})$ defines the probability of observing an object per unit area at location $\mathbf{s} \in \mathcal{R}$, while the intensity function $\lambda(\mathbf{s})$ defines the expected number of objects per unit area at location $\mathbf{s} \in \mathcal{R}$

Two-dimensional spatial point patterns

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

Two-dimensional spatial point patterns

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

Two-dimensional spatial point patterns

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

Two-dimensional spatial point patterns

- Specifically, the intensity $\lambda(\mathbf{s}_g)$ at each grid point $\mathbf{s}_g$ is estimated by:

$$\hat{\lambda}(\mathbf{s}_g) = \frac{c}{A_g} \sum_{i=1}^{n} k\left(\frac{\mathbf{s}_i - \mathbf{s}_g}{h}\right) w_i$$

where $k(\cdot)$ is the *kernel function* – usually a unimodal symmetrical bivariate probability density function; h is the *kernel bandwidth*, i.e., the radius of the kernel function; w_i is the value taken on by an optional weighting variable W; A_g is the area of the subregion of $\mathcal{R}$ over which the kernel function is evaluated, possibly corrected for *edge effects*; and c is a constant of proportionality

spgrid

- The purpose of spgrid is to generate two-dimensional grids that can be subsequently used by other programs to carry out several kinds of spatial data analysis, e.g., kernel estimation of densities and intensities for two-dimensional spatial point patterns

spgrid

- The purpose of spgrid is to generate two-dimensional grids that can be subsequently used by other programs to carry out several kinds of spatial data analysis, e.g., kernel estimation of densities and intensities for two-dimensional spatial point patterns
- In the context of spatial data analysis, a *grid* is a regular tessellation of the study region $\mathcal{R}$ that divides it into a set of contiguous cells whose centers are referred to as the *grid points*

spgrid

- spgrid can generate both square and hexagonal grids, i.e., grids whose cells are either square or hexagonal

spgrid

- spgrid can generate both square and hexagonal grids, i.e., grids whose cells are either square or hexagonal
- spgrid can generate grids covering both rectangular and irregular study regions, possibly made up by more than one polygon

spgrid

- spgrid can generate both square and hexagonal grids, i.e., grids whose cells are either square or hexagonal
- spgrid can generate grids covering both rectangular and irregular study regions, possibly made up by more than one polygon
- spgrid is able to generate grids with gaps, i.e., grids from which one or more subareas of the study region are excluded from the analysis

spkde

- spkde implements a variety of nonparametric kernel-based estimators of the probability density function and the intensity function of two-dimensional spatial point patterns

spkde

- spkde implements a variety of nonparametric kernel-based estimators of the probability density function and the intensity function of two-dimensional spatial point patterns
- spkde allows to choose among eight different kernel functions: uniform, normal, truncated normal, negative exponential, truncated negative exponential, quartic, triangular, and epanechnikov

spkde

- spkde implements a variety of nonparametric kernel-based estimators of the probability density function and the intensity function of two-dimensional spatial point patterns
- spkde allows to choose among eight different kernel functions: uniform, normal, truncated normal, negative exponential, truncated negative exponential, quartic, triangular, and epanechnikov
- The kernel bandwidth can be fixed, variable (based on a minimum number of weighted or unweighted data points), or a combination of the two (adaptive)

spkde

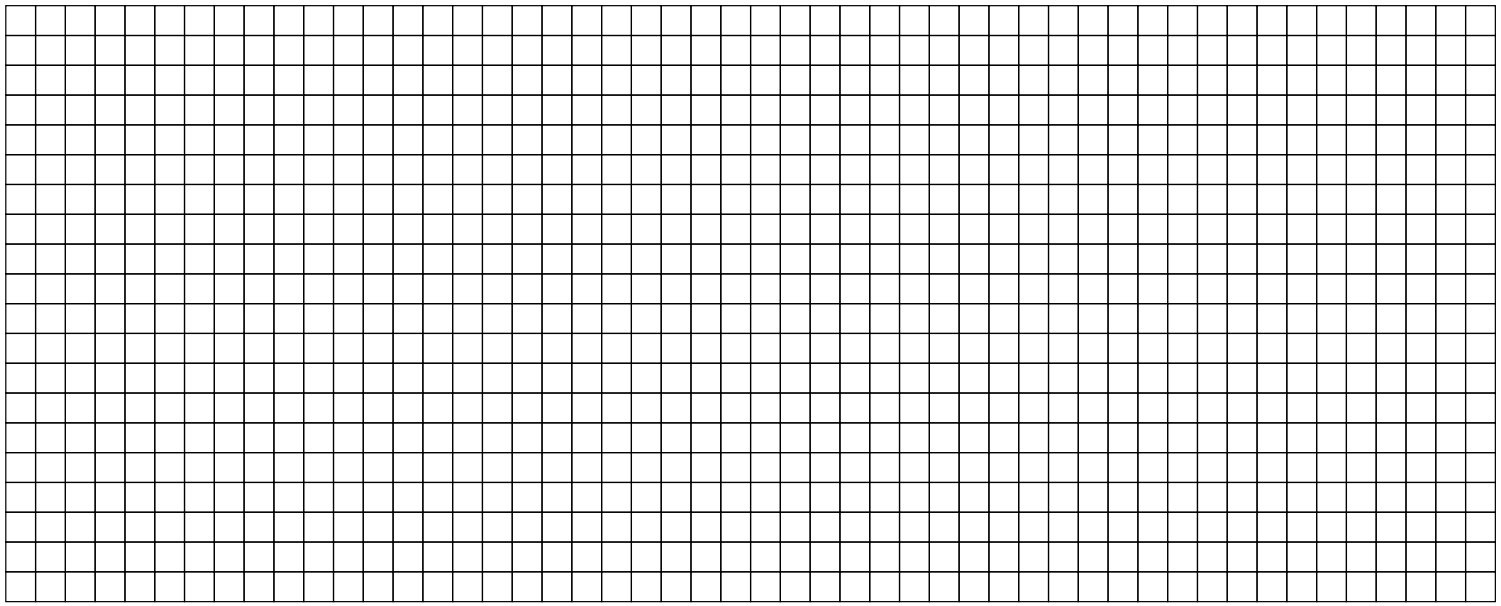
- spkde implements a variety of nonparametric kernel-based estimators of the probability density function and the intensity function of two-dimensional spatial point patterns
- spkde allows to choose among eight different kernel functions: uniform, normal, truncated normal, negative exponential, truncated negative exponential, quartic, triangular, and epanechnikov
- The kernel bandwidth can be fixed, variable (based on a minimum number of weighted or unweighted data points), or a combination of the two (adaptive)
- spkde applies an approximate edge correction to the estimates of the quantities of interest

Creating two-dimensional grids

- Let's see how `spgrid` can be used to generate several kinds of two-dimensional grids

Example 1

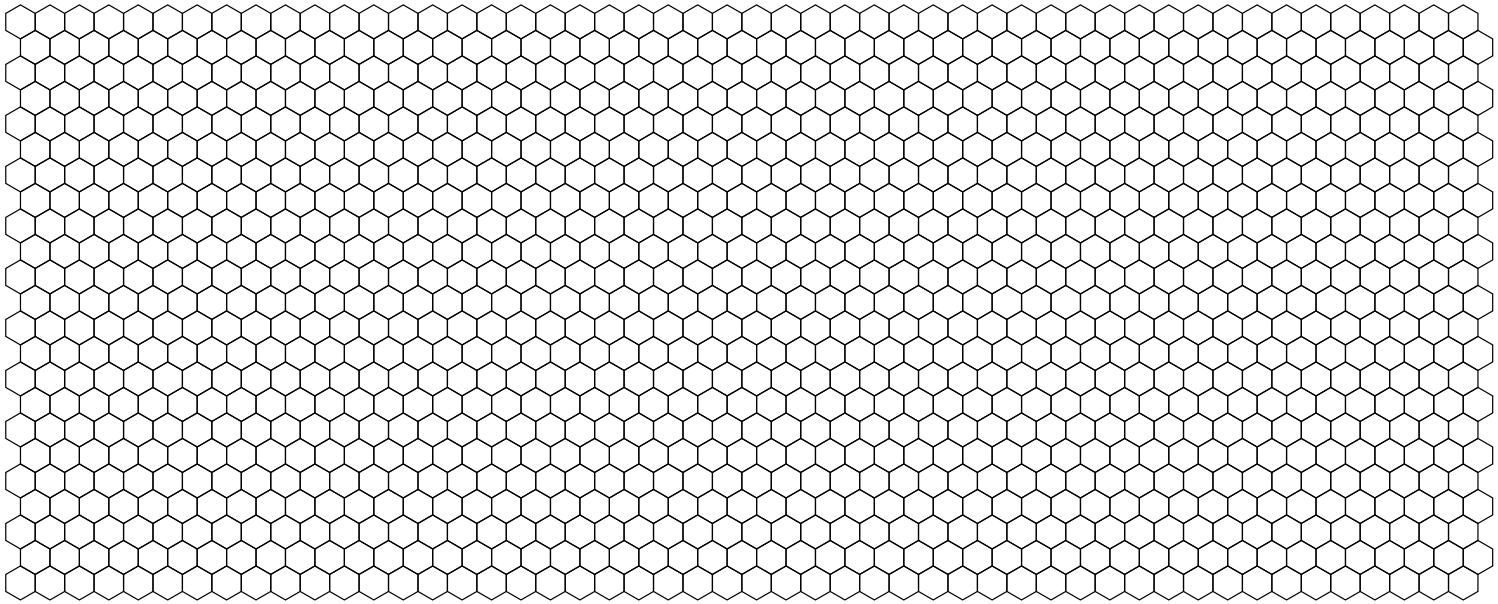
Rectangular study region - Square grid cells



```
. spgrid, shape(square) resolution(w10) xrange(0 500) yrange(0 200)   ///
     verbose replace cells("Rectangle-GridCells(Square).dta")         ///
     points("Rectangle-GridPoints(Square).dta")
. use "Rectangle-GridPoints(Square).dta", clear
. spmap using "Rectangle-GridCells(Square).dta", id(spgrid_id)
```

Example 2

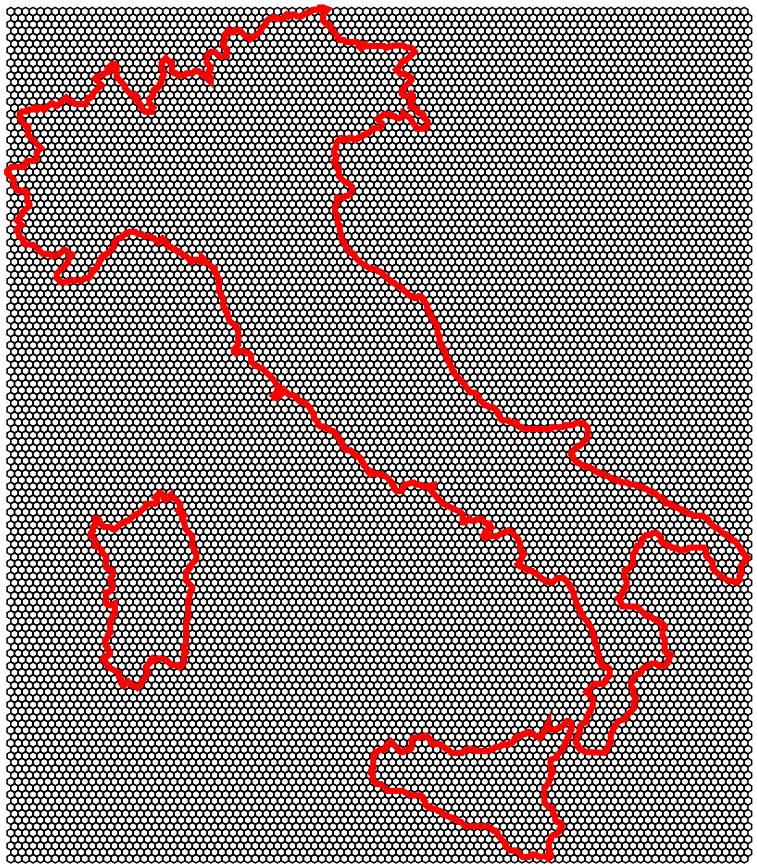
Rectangular study region - Hexagonal grid cells



```
. spgrid, shape(hexagonal) resolution(w10) xrange(0 500) yrange(0 200)   ///
    verbose replace cells("Rectangle-GridCells(Hexagonal).dta")          ///
    points("Rectangle-GridPoints(Hexagonal).dta")
. use "Rectangle-GridPoints(Hexagonal).dta", clear
. spmap using "Rectangle-GridCells(Hexagonal).dta", id(spgrid_id)
```

Example 3

Irregular study region - Hexagonal grid cells



```
. spgrid using "Italy-OutlineCoordinates.dta", ///
    shape(hexagonal) resolution(w10)           ///
    verbose replace                            ///
    cells("Italy-GridCells(Hexagonal).dta")    ///
    points("Italy-GridPoints(Hexagonal).dta")
. use "Italy-GridPoints(Hexagonal).dta", clear
. spmap using "Italy-GridCells(Hexagonal).dta", ///
    id(spgrid_id)                               ///
    poly(data("Italy-OutlineCoordinates.dta")   ///
    ocolor(red) osize(thick))
```

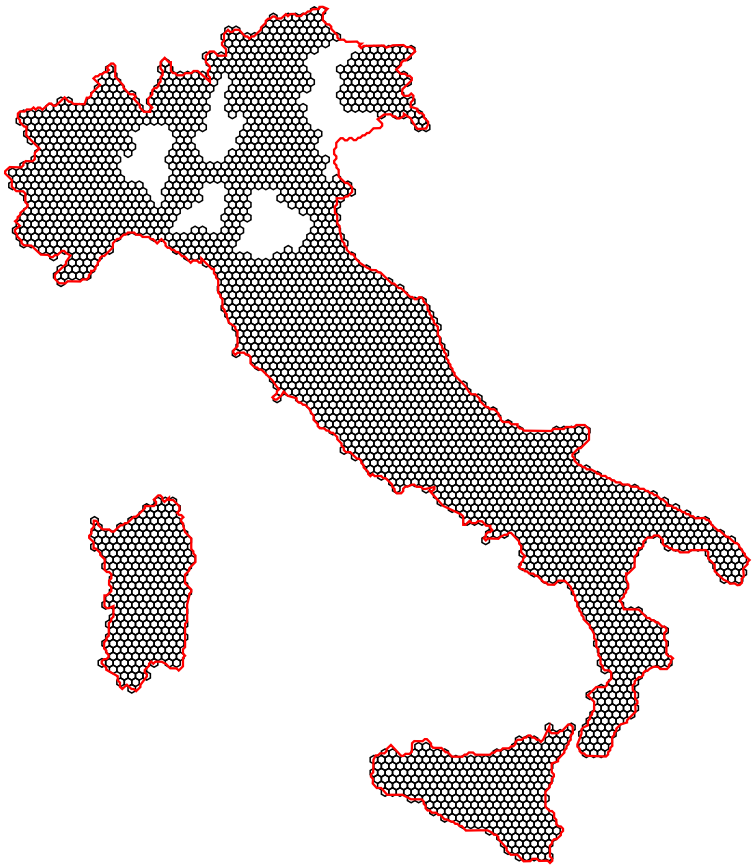
Example 4

Irregular study region - Hexagonal grid cells (valid cells only)

```
. spgrid using "Italy-OutlineCoordinates.dta", ///
    shape(hexagonal) resolution(w10)           ///
    verbose replace compress                   ///
    cells("Italy-GridCells(HexValid).dta")     ///
    points("Italy-GridPoints(HexValid).dta")
. use "Italy-GridPoints(HexValid).dta", clear
. spmap using "Italy-GridCells(HexValid).dta", ///
    id(spgrid_id)                              ///
    poly(data("Italy-OutlineCoordinates.dta") ///
    ocolor(red) osize(medium))
```

Example 5

Irregular study region with some areas excluded - Hexagonal grid cells (valid cells only)



```
. spgrid using "Italy-OutlineCoordinates.dta",   ///
     shape(hexagonal) resolution(w10)            ///
     mapexclude("Italy-Exclude.dta")             ///
     verbose replace compress                    ///
     cells("Italy2-GridCells(HexValid).dta")     ///
     points("Italy2-GridPoints(HexValid).dta")
. use "Italy2-GridPoints(HexValid).dta", clear
. spmap using "Italy2-GridCells(HexValid).dta", ///
     id(spgrid_id)                               ///
     poly(data("Italy-OutlineCoordinates.dta")   ///
     ocolor(red) osize(medium))
```

Estimating density and intensity functions

- Now, let's see how we can use `spkde` and the two-dimensional grids generated by `spgrid` to estimate the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of any given spatial point pattern

Estimating density and intensity functions

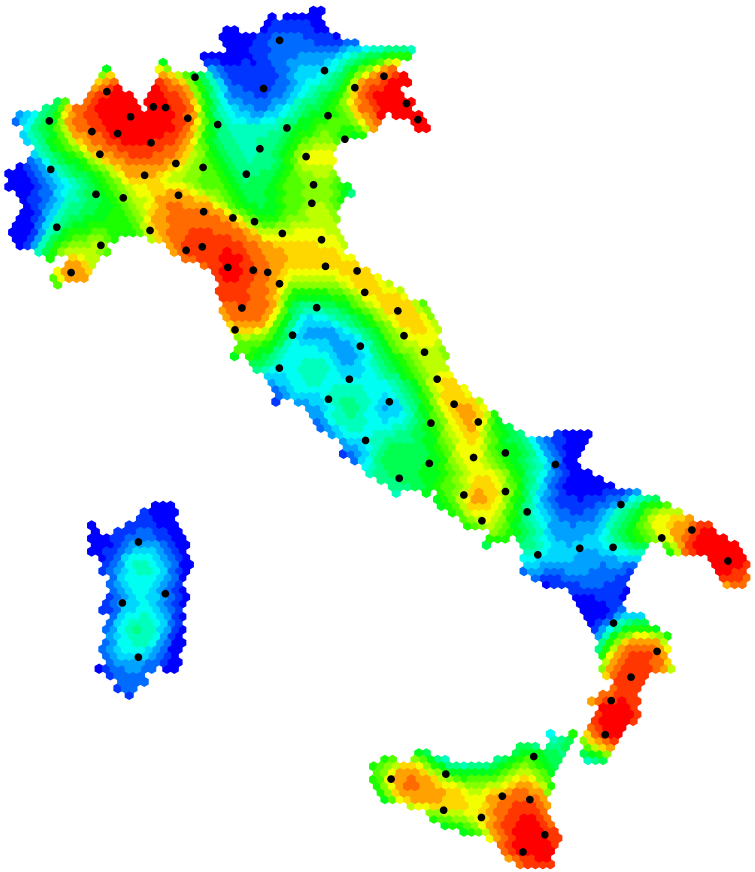
- Now, let's see how we can use `spkde` and the two-dimensional grids generated by `spgrid` to estimate the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of any given spatial point pattern
- To this aim, we will use data pertaining to the 103 Italian provinces, taking provinces centroids as the observed data points $\mathbf{s}_i$ $(i = 1, ..., 103)$

Estimating density and intensity functions

- Now, let's see how we can use `spkde` and the two-dimensional grids generated by `spgrid` to estimate the probability density function $p(\mathbf{s})$ and the intensity function $\lambda(\mathbf{s})$ of any given spatial point pattern
- To this aim, we will use data pertaining to the 103 Italian provinces, taking provinces centroids as the observed data points $\mathbf{s}_i$ $(i = 1, ..., 103)$
- $p(\mathbf{s})$ and $\lambda(\mathbf{s})$ will be estimated at each point $\mathbf{s}_g$ $(g = 1, ..., 3,483)$ of the grid generated in Example 4 above

Example 1: Simple point pattern

Quartic kernel function - Fixed bandwidth (100 km)

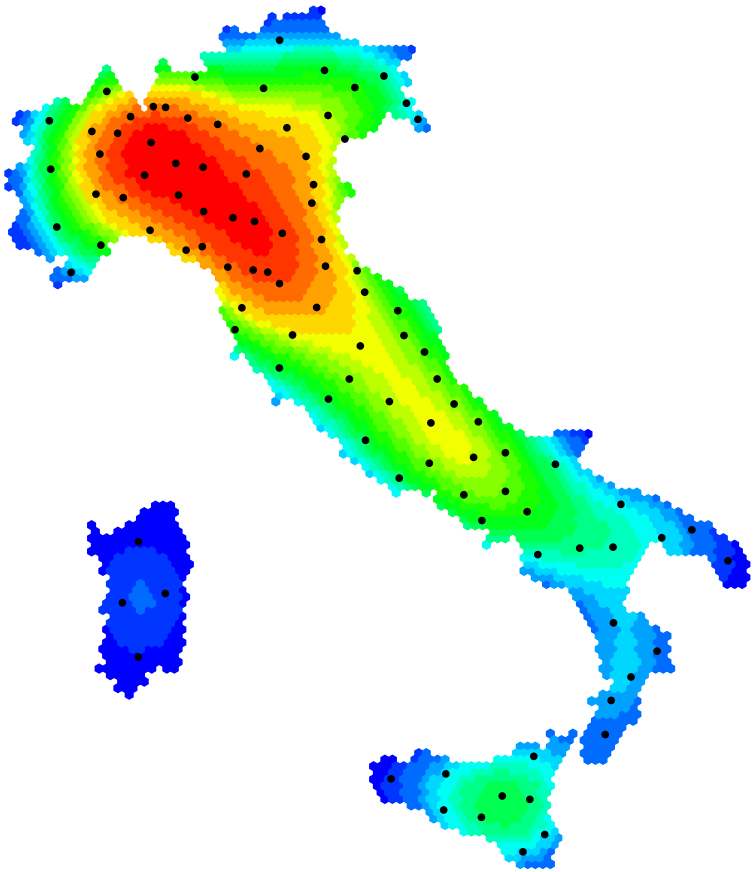


```
. use "Italy-DataPoints.dta", clear
. spkde using "Italy-GridPoints(HexValid).dta", ///
    xcoord(xcoord) ycoord(ycoord)                ///
    kernel(quartic) method(fixband)              ///
    bandwidth(100) verbose                       ///
    saving("Italy-Kde1.dta", replace)
. use "Italy-Kde1.dta", clear
. spmap density using                            ///
    "Italy-GridCells(HexValid).dta",             ///
    id(spgrid_id) clnum(20) fcolor(Rainbow)      ///
    ocolor(none ..) legend(off)                  ///
    point(data("Italy-DataPoints.dta")           ///
    x(xcoord) y(ycoord) size(*0.5))
```

Example 2: Simple point pattern

Normal kernel function - Fixed bandwidth (69.35 km)

The chosen bandwidth equals the average distance between each data point and its 5 nearest neighbors

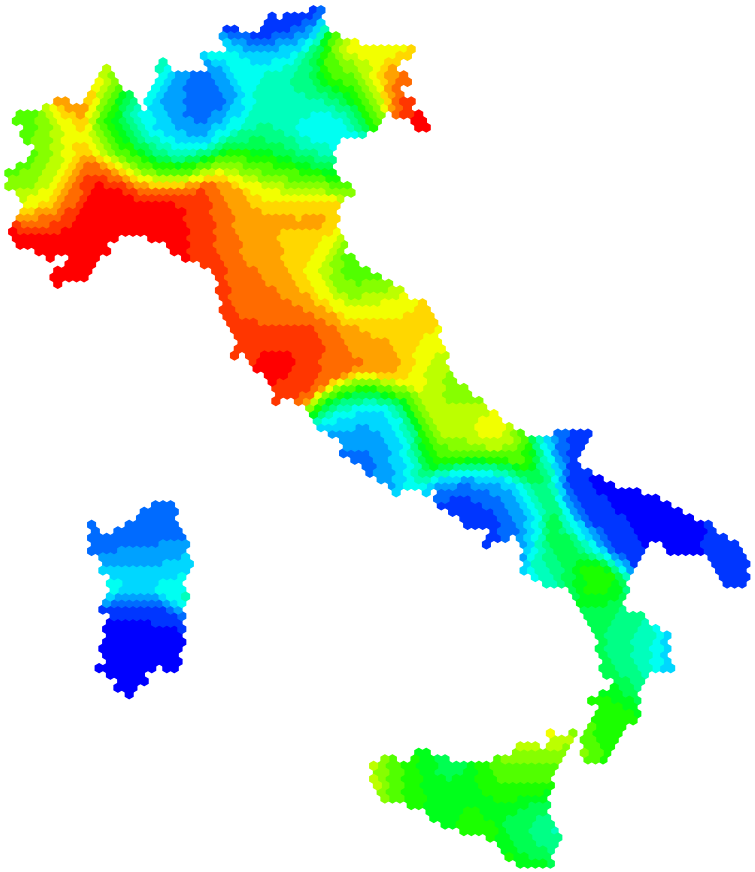


```
. use "Italy-DataPoints.dta", clear
. spkde using "Italy-GridPoints(HexValid).dta", ///
    xcoord(xcoord) ycoord(ycoord)                ///
    kernel(normal) method(fixband)               ///
    bandwidth(ad5) verbose                       ///
    saving("Italy-Kde2.dta", replace)
. use "Italy-Kde2.dta", clear
. spmap density using                            ///
    "Italy-GridCells(HexValid).dta",             ///
    id(spgrid_id) clnum(20) fcolor(Rainbow)      ///
    ocolor(none ..) legend(off)                  ///
    point(data("Italy-DataPoints.dta")           ///
    x(xcoord) y(ycoord) size(*0.5))
```

Example 3: Ratio of two intensities

Deaths for cardiovascular diseases / Total population

Quartic kernel function - Fixed bandwidth (100 km)



```
. use "Italy-DataPoints.dta", clear
. spkde dcvd95 pop95 using                    ///
    "Italy-GridPoints(HexValid).dta",         ///
    xcoord(xcoord) ycoord(ycoord)             ///
    kernel(quartic) method(fixband)           ///
    bandwidth(100) verbose                    ///
    saving("Italy-Kde3.dta", replace)
. use "Italy-Kde3.dta", clear
. generate ratio = dcvd95_intensity /   ///
    pop95_intensity * 1000
. spmap ratio using                                ///
    "Italy-GridCells(HexValid).dta",               ///
    id(spgrid_id) clnum(20) fcolor(Rainbow)        ///
    ocolor(none ..) legend(off)
```

Estimating bivariate densities for non-spatial data

- `spgrid` and `spkde` can be used to estimate the joint probability density function $p(x, y)$ of any pair of quantitative variables X and Y

Estimating bivariate densities for non-spatial data

- `spgrid` and `spkde` can be used to estimate the joint probability density function $p(x, y)$ of any pair of quantitative variables X and Y
- As an example, let's estimate and plot the bivariate probability density function for two of the variables included in the `auto` dataset: `mpg` and `price`

Example

Step 1: Normalize variables in the range $[0, 1]$

```
. sysuse "auto.dta", clear
. summarize price mpg
. clonevar x = mpg
. clonevar y = price
. replace x = (x-0) / (50-0)
. replace y = (y-0) / (20000-0)
. mylabels 0(10)50, myscale((@-0) / (50-0)) local(XLAB)
. mylabels 0(5000)20000, myscale((@-0) / (20000-0)) local(YLAB)
. keep x y
. save "xy.dta", replace
```

Example

Step 2: Generate a 100x100 grid

```
. spgrid, shape(hexagonal) xdim(100)    ///
     xrange(0 1) yrange(0 1)            ///
     verbose replace                    ///
     cells("2D-GridCells.dta")          ///
     points("2D-GridPoints.dta")
```

Example

Step 3: Estimate the bivariate probability density function

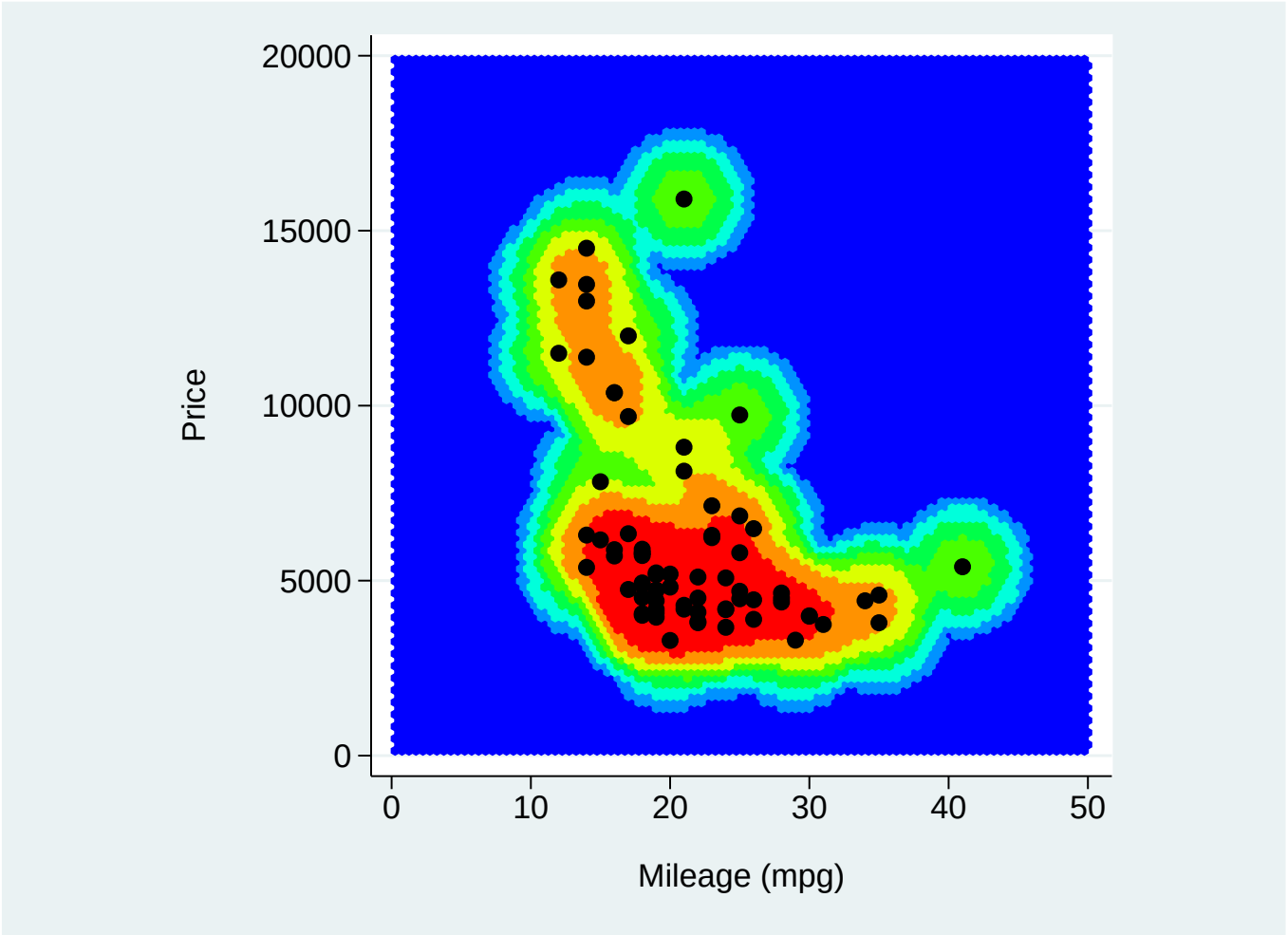
```
. spkde using "2D-GridPoints.dta", xcoord(x) ycoord(y)   ///
      kernel(quartic) method(fixband) bandwidth(0.1)     ///
      noedge verbose saving("2D-Kde.dta", replace)
```

Example

Step 4: Display the density plot

```
. use "2D-Kde.dta", clear
. recode density (.=0)
. spmap density using "2D-GridCells.dta",     ///
     id(spgrid_id) clnum(20) fcolor(Rainbow) ///
     ocolor(none ..) legend(off)             ///
     point(data("xy.dta") x(x) y(y))         ///
     freestyle aspectratio(1)                ///
     xtitle(" " "Mileage (mpg)")             ///
     xlab(‘XLAB’)                            ///
     ytitle("Price" " ")                     ///
     ylab(‘YLAB’, angle(0))
```

Example



Conclusion

- `spgrid` and `spkde` add to the growing set of commands for spatial data analysis available to Stata users

Conclusion

- `spgrid` and `spkde` add to the growing set of commands for spatial data analysis available to Stata users
- Both programs will be submitted to the SSC Archive as soon as their respective help files are ready

Conclusion

- `spgrid` and `spkde` add to the growing set of commands for spatial data analysis available to Stata users
- Both programs will be submitted to the SSC Archive as soon as their respective help files are ready
- I'm currently working on other Stata tools for exploratory spatial data analysis: ideas and suggestions are welcome