Lecture 5

Spatial point pattern data analysis

MSc in Epidemiology @ Imperial College London

March 18, 2019

Lecture outline

Analysing point pattern data in R

- The two main packages used for spatial point pattern analysis are spatstat and splancs.
- spatstat is an R library for the statistical analysis of spatial data, mainly spatial point patterns.
- spatstat contains facilities for data manipulation, tools for exploratory data, analysis, convenient graphical facilities, tools to simulate a wide range of point pattern models, versatile model-fitting capabilities, and model diagnostics.
- spatstat contains 43 datasets. The online manual is over 1000 pages.
- The basic data types in spatstat are Point Patterns, Windows, and Pixel Images.

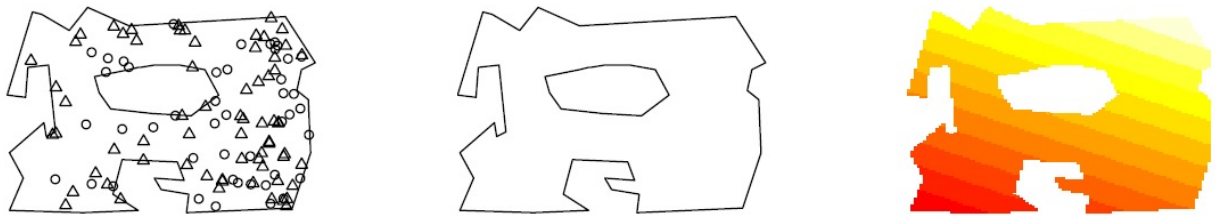
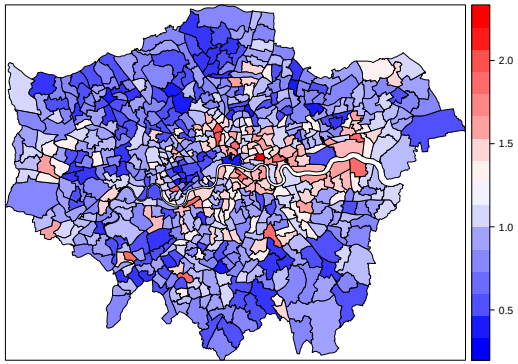


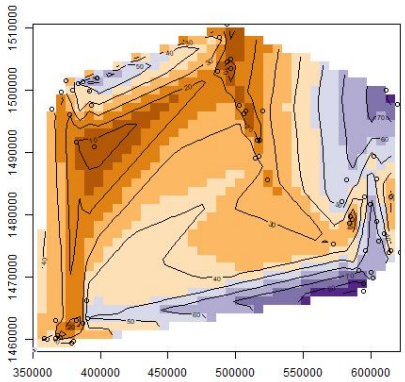
Figure: A point pattern, a window, and a pixel image. http://www3.uji.es/~mateu/badturn.pdf

Analysing point pattern data in R

- A point pattern is represented in spatstat by an object of the class ppp. To obtain it you can:
  - use one of the datasets supplied with the package
  - create one from data in R, using `ppp()`
  - create one from data in a text file, using `scanpp()`
  - convert data from other R libraries, using `as.ppp()`

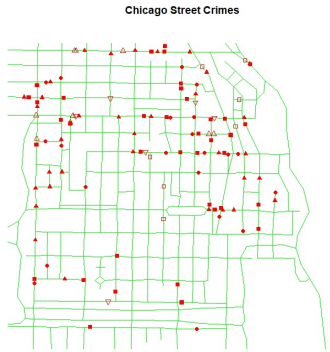
Recap of spatial data





(a) **Areal data**
A spatial process is observed on a regular or irregular grid. Most of the time this arises due to aggregation of some sort

(b) **Geostatistical data**
A spatial process that varies continuously is observed only at a few points



(c) **Point pattern data**
A spatial process is observed at a set of locations and the locations themselves are of interest

Point patter data analysis

- A point pattern data set gives the locations of all events or individuals observed in a certain region. Examples:
  - locations of trees in a forest stand
  - locations of crimes
  - accidents
  - locations of persons with a disease
  - galaxies in the universe
- Typically, a point data sets consists of a set of observed (x, y) coordinates, say $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where n is the number of observations. As alternative notation, each point could be denoted by a vector $\boldsymbol{s}_i$, where $\boldsymbol{s}_i = (x_i, y_i)$.

Overview

- A point pattern is a collection of points in some area or set and is typically interpreted as a sample from (or realisation of) a point process.
- The statistical models describing them are called point process models (PPM).
- PPMs treat both the numbers and the locations of discrete points as random quantities governed by an underlying, continuous intensity field.
- The intensity is the expected number of points per unit area in some study region and is the modelled parameter.

Questions and Objectives

- Is the pattern random or structured in some fashion?
- What kind of attraction/repulsion exists in the process?
- Is there regular spacing between locations or do locations show a tendency to cluster?
- Is there a measurement associated with points (*marked patterns*)?
- Does the probability of observing the event vary according to some *covariates*? (Need to relate predictors to observations in a regression type setting)

Point pattern terminology

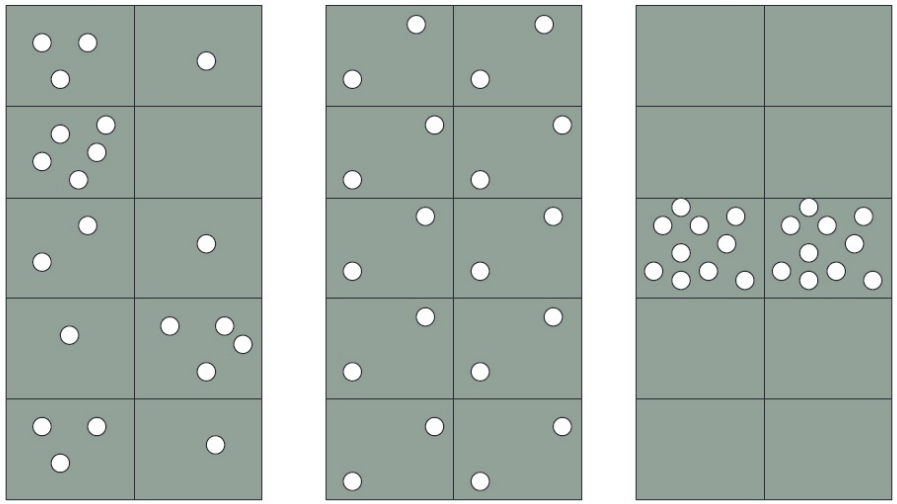
We distinguish between:

- *Point* is any location in the study area where an event could occur
- *Event location* is a particular location in the study area where an event did occur
- *Window* is our study region, in two-dimensional space. The word *window* implies that we are looking through glass at a piece of a process
- *Mapped point pattern* is a map of points, assumes all relevant events within study area have been recorded (i.e. the Cartesian coordinates of points are determined either for all points in the whole window, resulting in mapped data)
- *Sampled point pattern* events are recorded from a sample of different areas within a region

Types of relationships among points

Three general patterns:

- Random - any point is equally likely to occur at any location and the position of any point is not affected by the position of any other point
- Uniform - every point is as far from all of its neighbors as possible (repulsion).
- Clustered - many points are concentrated close together, and large areas that contain very few, if any, points (attraction).



Credit: Seth Spielman, http://www.sethspielman.org

Marks

- The points may have extra information called marks attached to them
- The mark represents an *attribute* of the point
- The mark variable could be:
  - continuous: the mark attached to each point is a single real number (e.g. tree height);
  - categorical: points are classified into several types; the mark attached to each point is a level of a factor (e.g. tree species)

Covariates

- Covariates are explanatory variables
- The analysis may involve assessing the role of covariates in determining intensity (e.g. does disease incidence depend on elevation above sea level?)
- The analysis may control for the covariate effects when assessing interaction between points (e.g. is there clustering of disease outcomes after controlling for elevation?)

Point pattern properties

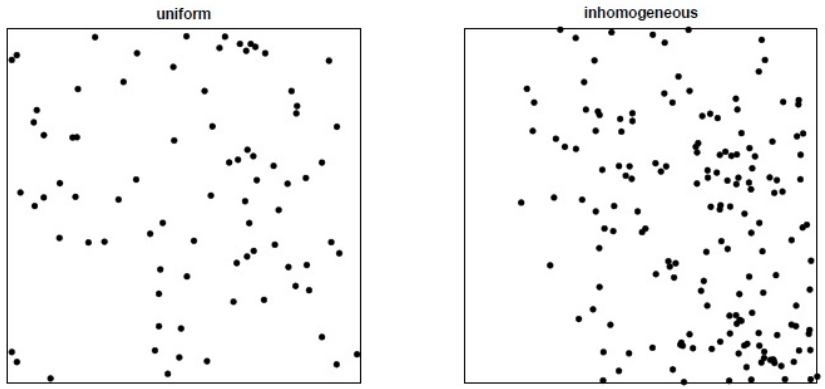
Point patterns have first and second order properties:

- First-order properties measure the distribution of events in a study region: intensity of a spatial process and the observed density of a pattern under study

- Second-order properties measure the tendency of events to appear clustered, independently, or regularly-spaced

Intensity

- The Intensity λ is the mean number of events per unit area:
  - Intensity may be constant $\rightarrow$ uniform or homogeneous
  - Intensity may vary from location to location $\rightarrow$ non-uniform inhomogeneous



- First step in analysing a point pattern

Intensity estimation

- If the point process is known to be *homogeneous* an unbiased estimate of the intensity is

$$\hat{\lambda} = \frac{n}{|A|}$$

where n is the number of points observed in region A, and $|A|$ is the area of region A.

- If the point process is known to be *inhomogeneous*, the intensity function or intensity measure can be estimated non-parametrically by techniques such as kernel smoothing (e.g. Diggle 1985).

Complete Spatial Randomness (CSR)

- Complete Spatial Randomness (CSR): the events are distributed *independently* and *homogeneously* over a domain $\mathcal{D}$.
- CSR means an event is equally likely to occur at any location or region within $\mathcal{D}$.
- A point process which is CSR point process is formally defined as a homogeneous Poisson process (HPP).
  - Under HPP, the location of one point in space does not affect the probabilities of other points' appearing nearby. The intensity of the point process in $\mathcal{D}$ is a constant.
- A generalization of HPP which allows for non-constant intensity $\lambda(s)$ is called an inhomogeneous Poisson process (IPP).

Homogeneous Poisson Process

The homogeneous Poisson process with intensity λ is a particular set of spatial point processes. It is defined by two properties (Waller and Gotway, 2004, p. 123):

1. The number of events occurring within a finite region A is a random variable following a *Poisson distribution* with mean $\lambda|A|$, where λ is the intensity parameter of the process and $|A|$ is the area of A.
2. Given the total number of events n occurring within an area A, the locations of the n events represent an independent random sample of n locations, where each point (location where an event could occur) is equally likely to be chosen as an event.

Criterion 2 represents the general concept of CSR (events uniformly distributed across the study area) and criterion 1 introduces the idea of an intensity λ representing the number of events expected per unit area.

Homogeneous Poisson Process

Here we illustrate two scenarios that result in the same thing:

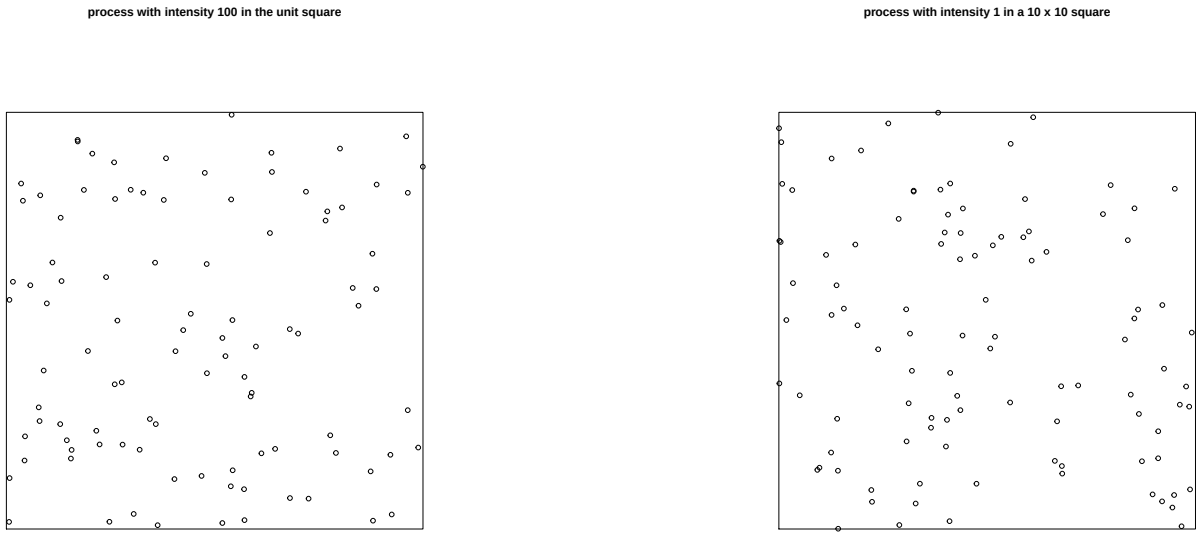


Figure: Generation of Poisson Point Pattern

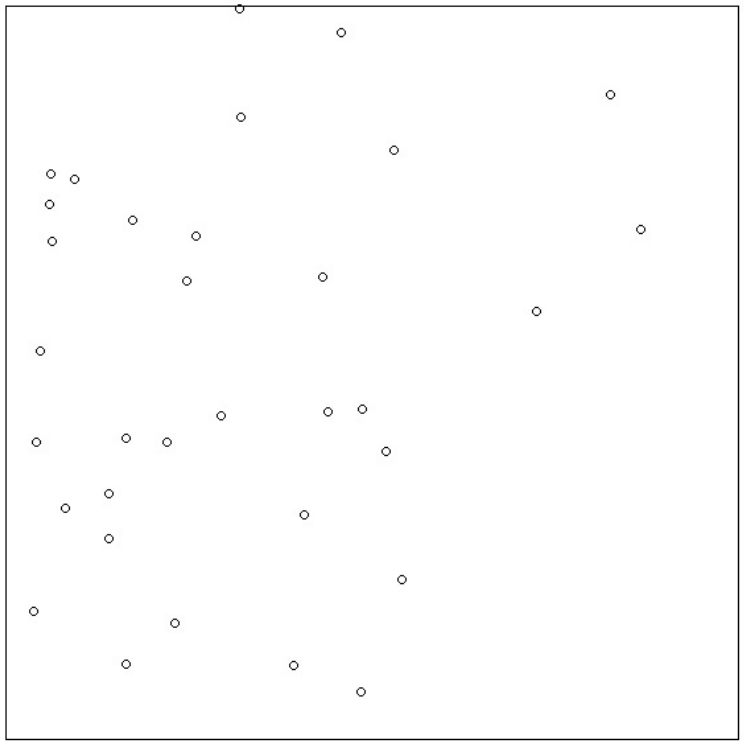
```
# uniform Poisson process with intensity 100 in the unit square
pp1 <- rpoispp(100)
plot(pp1, "process with intensity 100 in the unit square")

# uniform Poisson process with intensity 1 in a 10 x 10 square
pp2 <- rpoispp(1, win=owin(c(0,10),c(0,10)))
plot(pp2, main="process with intensity 1 in a 10 x 10 square")
```

Homogeneous Poisson Process

- The Poisson distribution allows the total number of observed events to vary from realization to realization while maintaining a fixed but unknown number of events per unit area.
- The expected number of events per unit area is the intensity, that, as we saw previously, is $\hat{\lambda} = \frac{n}{|A|}$
- Under a HPP, the location of one point in space does not affect the probabilities of other points appearing nearby. The intensity of the point process in area A is a *constant* $\lambda > 0$.
- To investigate this in practice, plots of the data are typically a good place to start, but the tendency of the human eye to see clustering or other structure in virtually every point pattern renders a strictly graphical approach unreliable.

Inhomogeneous Poisson process

- Inhomogeneous Poisson process (IPP) occurs when the intensity λ is not constant over the region
- The number of events occurring within a finite region A is a random variable following a Poisson distribution with mean $\int_A \lambda(\boldsymbol{s})d\boldsymbol{s}$
- In many cases, homogeneity in intensity is not realistic (for example the locations of trees in a forest may be irregular due to geographic features such as soil, rock, slope or other terrain irregularities.)
- This spatial variation can be diverse, with events appearing more likely in some areas than others, thus the *intensity is a function that varies spatially* $\lambda(\boldsymbol{s})$
- For an IPP point process, intensity can be estimated non-parametrically with kernel smoothing




```
# Inhomogeneous Poisson process in unit square
# with intensity lambda(x,y) = 100 * exp(-3*x)
# Intensity is bounded by 100

pp <- rpoispp(function(x,y) {100 * exp(-3*x)}, 100)
plot(pp, "Inhomogeneous Poisson process")
```

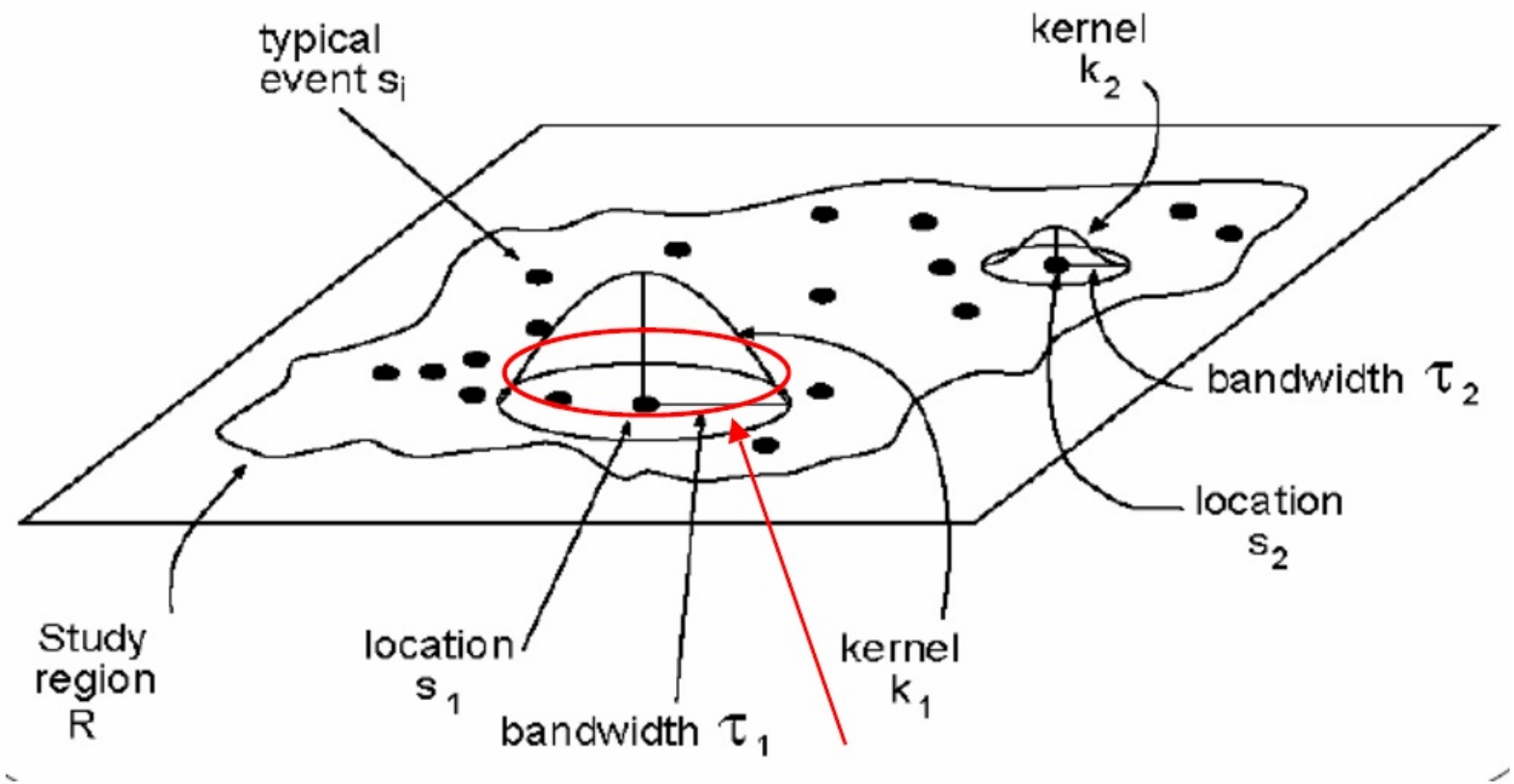
Kernel Density estimation

- Kernel smoothing is a popular non-parametric way of estimating the intensity at a given point s
- Here, the intensity is represented by a generic function for λ, called the kernel that allows it to vary spatially

$$\hat{\lambda}(s) = \frac{1}{h^2} \sum_i k \frac{\|s - s_i\|}{h} / q(\|s\|)$$

- s is any location
- k is the kernel function (an arbitrary probability density)
- h is the bandwidth for smoothing (small values of h result in spiky estimates and large values of h result in smoother functions)
- $q(\|s\|)$ is border correction to compensate for observations missing due to edge effects

Kernel Density estimation



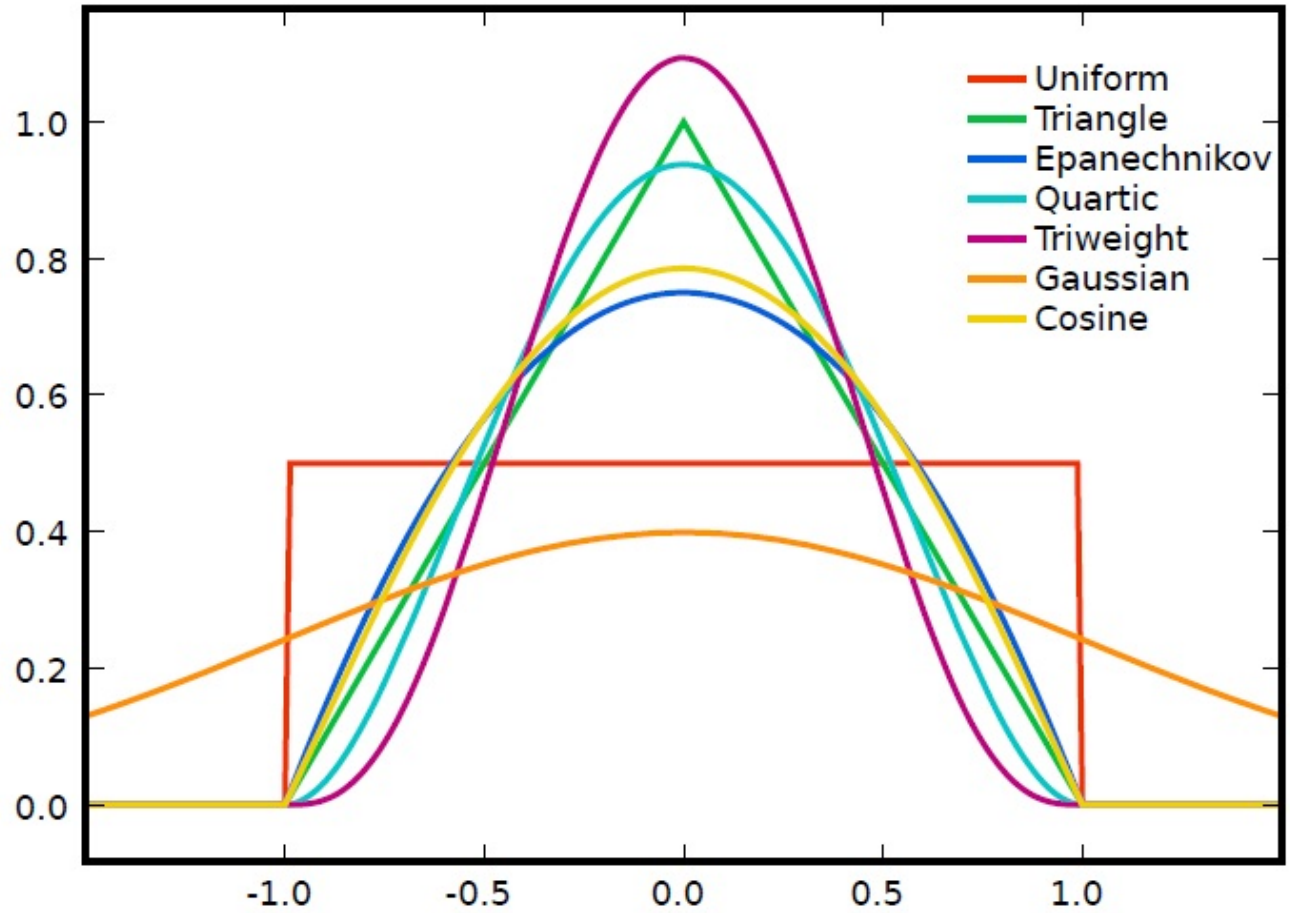
- Creating a smooth surface for each kernel
- Surface value highest in the center (point location) and diminishes with distance…reaches 0 at radius distance

The kernel (k) is basically a mathematical function that calculates how the surface value "falls off" as it reaches the radius.

Credit: Seth Spielman

Examples of kernel functions

There are various kernel functions:

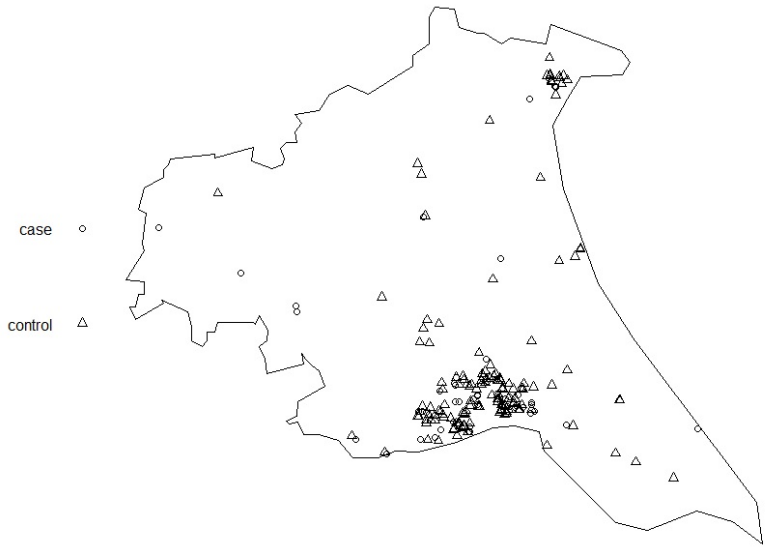


From http://upload.wikimedia.org/wikipedia/commons/4/47/Kernels.svg

Kernel density surfaces in spatstat

- In spatstat the density function creates a kernel density surface which is stored in R as spatstat class im or image.
- The example is obtained by using the dataset humberside from the package spatstat that includes the spatial locations of cases of childhood leukaemia and lymphoma, and randomly-selected controls, in North Humberside.

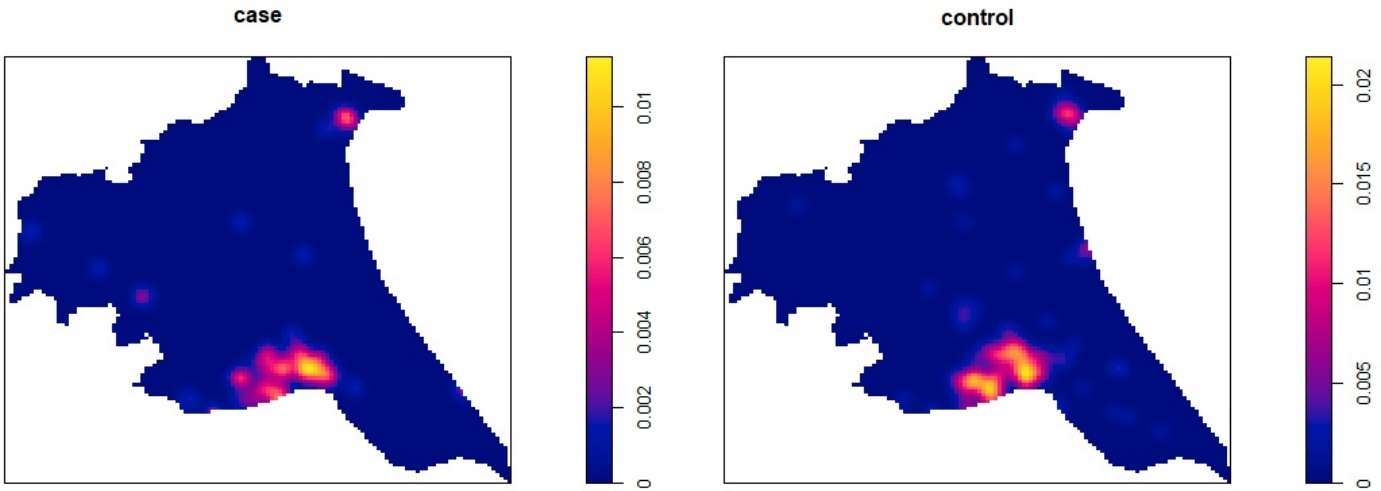
```
data(humberside)
x <- humberside
summary(x)
plot(x, main="")
```



The next 2 plots show kernel density estimates for the Humberside data at different values of the bandwidth h.

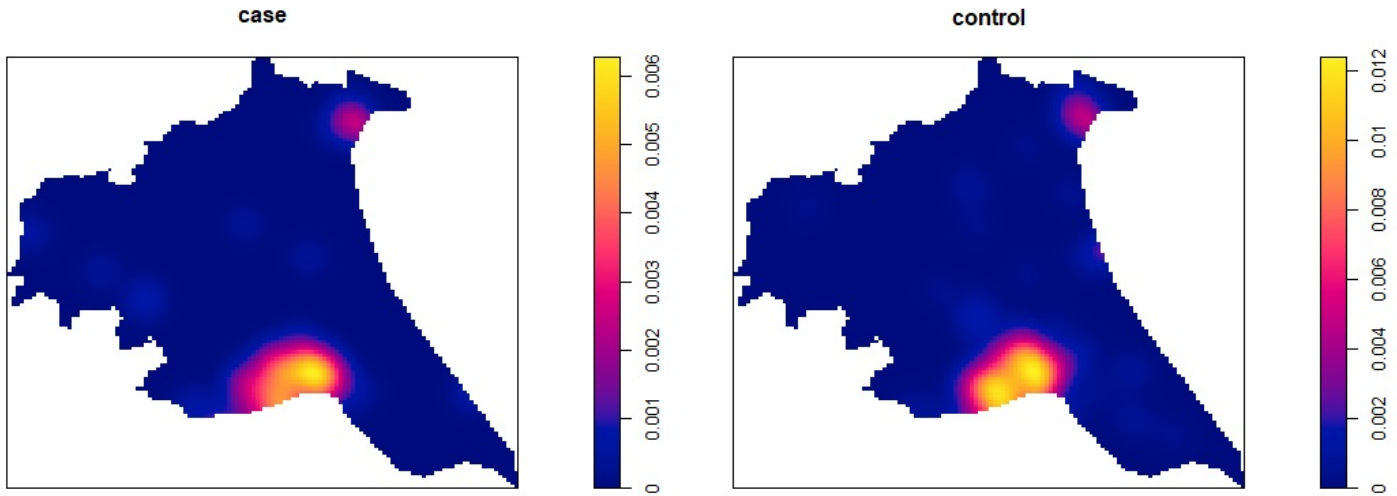
Kernel density surfaces in spatstat

```
plot(density(split(x), sigma=10))
# sigma determines the bandwidth of the kernel
```



Kernel density surfaces in spatstat

```
plot(density(split(x), sigma=20))
```



Notes on bandwidth selection

- The kernel bandwidth `sigma` controls the degree of smoothing
- How we saw, small value of `sigma` produces an irregular intensity surface, while a large value of sigma appears to oversmooth the intensity.
- There is no general rule for selecting it. The default in `spatstat` is to take `sigma` equal to one-eighth of the shortest side length of the enclosing rectangle
- Several algorithms are available for automatically selecting the bandwidth `sigma` by minimising a measure of error (e.g. `bw.diggle` from Diggle (1985) implements a mean square error cross-validation method)

Tests for CSR

- Statistics that measure clustering, and perhaps even associated significance tests, are often used
- The statistical tests for studying point distributions rely on the comparison between an observed spatial pattern and a random theoretical pattern (i.e. the null hypothesis is that the observed pattern is random and is produced by the CSR process.)
- Several tests of CSR use Monte Carlo methods. Simulate a large number of CSR processes and compare the test statistic from N_{sim} to test statistic from observed
- In the next slides, we are going to use:
  - Quadrat counting
  - the *nearest neighbour distances* (in particular the $\mathcal{G}$ function)
  - the Ripley's K function

Quadrat counting test of homogeneity

- The quadrat counting method determines the point distribution by examining its density over the study area.
- Analysis is based on quadrats (or grid cells) that are constructed over the observation window (the most commonly used surfaces in quadrat analysis are square grids, but other surfaces can be used depending on the analytical objectives of the study and the nature of the spatial phenomena under investigation).
- Then, the next step is the quantification of the number of events per cell (quadrat) and the intensity of events in each quadrat.
- The end goal is to compare the observed distribution of points to a theoretical random pattern to assess whether it is clustered, dispersed, or random.

Quadrat counting in spatstat

To illustrate the method we use the data set `swedishpines` in `spatstat` that represents the positions of 71 trees in a Swedish forest.

```
data(swedishpines)

plot(swedishpines,
     main="Swedish Pines point pattern")
```

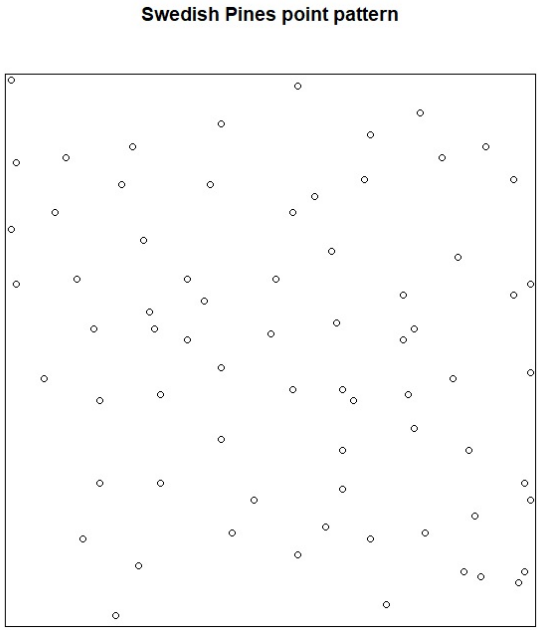
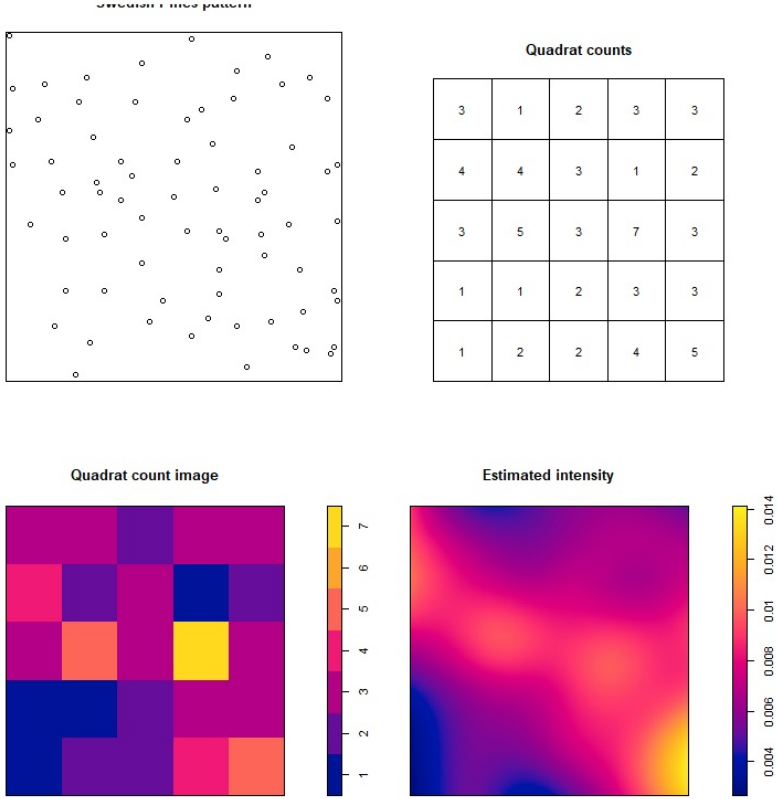


Figure: Locations of trees

Plotting

A series of plots can be performed by using an object of class `solist`, that represents a list of spatial objects (such as point patterns, line segment patterns, pixel images)

```
X <- swedishpines
QC <- quadratcount(X)
QCI <- as.im(X,dimyx=5)
DI <- density(X)
L <- solist(X, QC, QCI, DI)
names(L) <- c("Swedish Pines pattern",
              "Quadrat counts",
              "Quadrat count image",
              "Estimated intensity")
```



Quadrat counting in spatstat

- The null hypothesis is that the intensity is homogeneous, and the alternative hypothesis is that the intensity is inhomogeneous in some unspecified fashion.

- We divide the window into say m quadrats and count the numbers of points $n_1, \ldots, n_m$ in each quadrat

- If the null hypothesis is true, the n_j are realisations of independent Poisson random variables with expected values $\mu_j = \lambda a_j$ where λ is the unknown intensity and a_j is the area of j-th quadrat. If the quadrats all have equal area, then the counts n_j are independent Poisson random variables with equal mean λa.

Quadrat counting in spatstat

- The χ^2 (chi-squared) test could be used to test homogeneity assuming independence.

- The χ^2 *test of uniformity* is:

$$\chi^2 = \sum_j \frac{(\text{observed} - \text{expected})^2}{\text{expected}} = \sum_j \frac{(n_j - e_j)^2}{e_j}$$

where the estimated intensity is $\bar{\lambda} = n/a$ and the expected count in quadrat j are $e_j = \bar{\lambda} a_j = n a_j / a$

Quadrat counting in spatstat

```
swetest <- quadrat.test(swedishpines, nx = 3, ny = 3)
swetest

plot(swedishpines, main = "")
plot(swetest, add = TRUE, cex = 2)
```

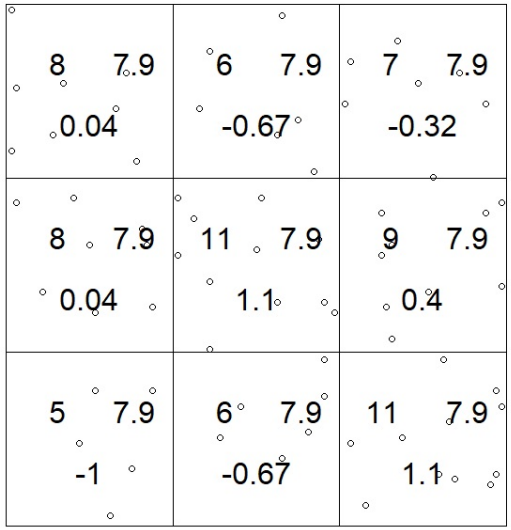


Figure: Quadrat counting test of CSR

Note: Plotting the object will display the quadrats, annotated by their observed and expected counts and the Pearson residuals.

Weaknesses of quadrat counting method

- The power of the quadrat test depends on the size of quadrats:
  - If too small, they may contain only a few points
  - If too large, they may contain too many points
- It results in a single measure for the entire distribution, so variations within the region are not recognized
- Broadly speaking, it is a measure of dispersion, and not really pattern, because it is based primarily on the density of points, and not their arrangement in relation to one another

Ripley's K function

- The Ripley's K function is an effective method for seeing whether the process is completely random in space
- The K-function is a function of the *distance* (d) and is defined as:

$$K(d) = \lambda^{-1} E(N_d)$$

  - E is the expectation
  - N_d is the number of points within a distance d of an arbitrary point
  - λ is the intensity of the process, i.e. the mean number of points per unit area
  - d is the radius of a circle centered on an arbitrary event
- $K(d)$ tests the expected number of events within distance d from an arbitrary event (excluding the chosen event itself) divided by the average number of events per unit area
- $K(d)$ is equivalent to showing the variance of the number of events occurring in subregion A (Ripley 1977) so is a second order property of the point process.

Ripley's K-function

- The theoretical value of K is known for certain spatial point process models.
- For instance, the K-function for an homogenous Poisson process is $K(d) = \pi d^2$, since in this case the number of points within d of an arbitrary point should be proportional to the area of a circle of radius d;
- Values of $K(d)$ higher than πd^2 indicate *clustering*, while smaller values indicate a regular pattern
- What should K look like for violations of CSR?
  - K above expectations
    - ★ More points than expected
    - ★ Evidence of clustering
  - K below expectations
    - ★ Evidence of repulsion

K function in R

- The K function estimation may be estimated in spatstat package using the Kest function
- We use the data set bramblecane that contain the location of living bramble canes

```
data(bramblecanes)
plot(bramblecanes)
```

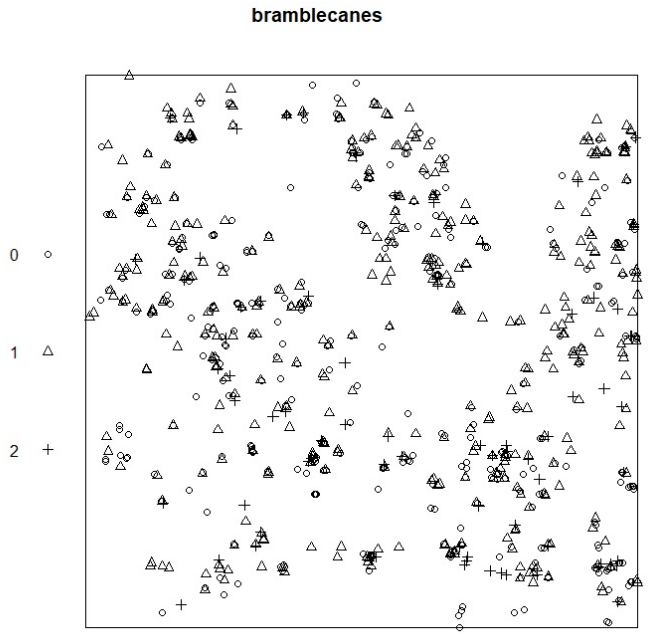


Figure: Bramble cane locations

K function in R

```
kf <- Kest(bramblecanes, correction='border')
plot(kf)
```

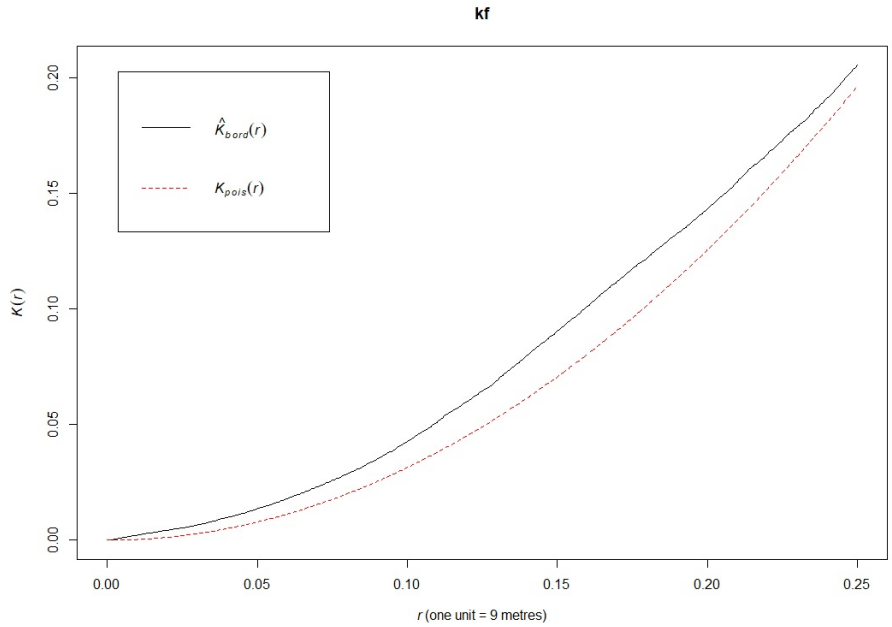


Figure: K function

- The red line (labelled $\hat{K}_{pois}$) represents the theoretical K function under the null hypothesis that the points are completely randomly distributed (CSR), while the black line is the estimated empirical K function (labelled $\hat{K}_{bord}$).
- Empirical values less than our theoretical ones suggest a regular pattern. Empirical values greater than theoretical values suggest clustering.

K function in R

- Fore a more rigorous investigation, we can allow for sampling variation via simulations

- This simulation approach is sometime referred to as *envelope* analysis, the evelope being the highest and lowest value of $\hat{K}(d)$ for a value of d

- The envelop function for the `bramblecane` can be used following the syntax:

```
kf.env <- envelope(bramblecanes,  fun=Kest, correction='border')
```

The $\mathcal{G}$ function

- The $\mathcal{G}$ Function measures the distribution of distances from an arbitrary event to its nearest neighbors.

$$\hat{\mathcal{G}}(r) = \frac{\sum_{i=1}^{n} I_i}{n}$$

$$I_i = \begin{cases} 1 & \text{if } d_i \in \{d_i : d_i \leq r, \forall i\} \\ 0 & \text{otherwise} \end{cases}$$

- where $d_i = \min_j\{d_{ij}, \forall j \neq i \in S\}, i = 1, \ldots, n$.
- So, the $\mathcal{G}$ function represents the number of elements in the set of distances up to some threshold r, normalized by the total number of points n in point pattern S.
- Under CSR, the value of the $\mathcal{G}$ function becomes:

$$\mathcal{G}(r) = 1 - e^{\lambda \pi r^2}$$

- where λ is the mean number of events per unit (intensity).

The $\mathcal{G}$ function

- The comparability of a point process with CSR can be assessed by plotting the empirical function $\hat{\mathcal{G}}(r)$ against the theoretical expectation $\mathcal{G}(r)$.

- For a *clustered pattern*, observed locations should be closer to each other than expected under CSR. A *regular pattern* should have higher nearest-neighbor distances than expected under CSR.

- Note that the $\mathcal{G}$ function is a first-order spatial function, that tells us nothing about the relationship between the points, just that there is one.

The $\mathcal{G}$ function

```
gf <- Gest(bramblecanes, correction='border')
plot(gf)

# with envelope
gf.env <- envelope(bramblecanes,
       fun=Gest, correction='border')
plot(gf.env)
```

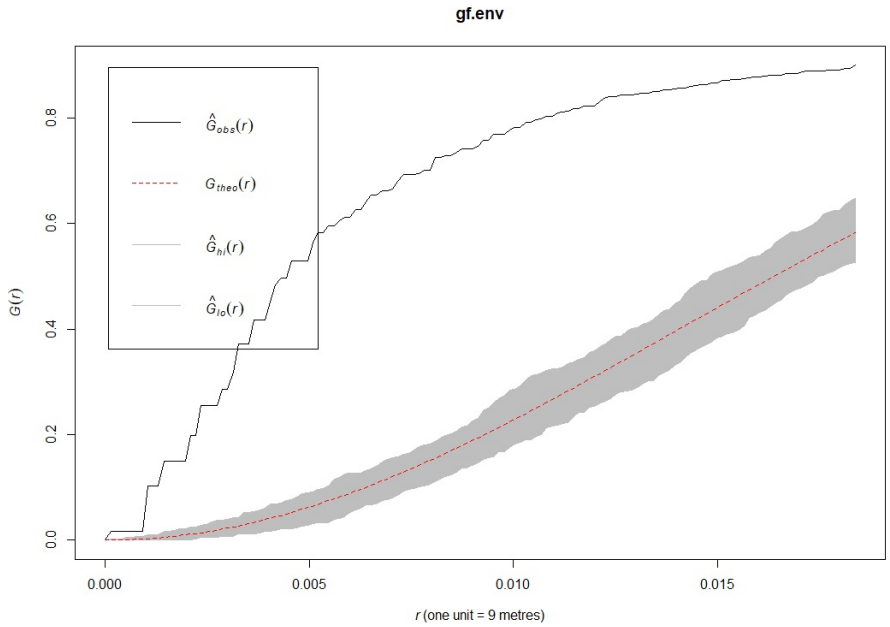


Figure: $\mathcal{G}$ function with envelope

If there is no clustering, we should get a straight diagonal line

Practical 5: Bayesian analysis of the relationship between a putative source of environmental pollution (an asbestos cement plant) and the occurrence of pleural malignant mesothelioma in the area of Casale Monferrato (Italy) in 1987-1993

The authors assumed that cases and controls are a random sample from a *marked point process*, the mark indicating case-control status.

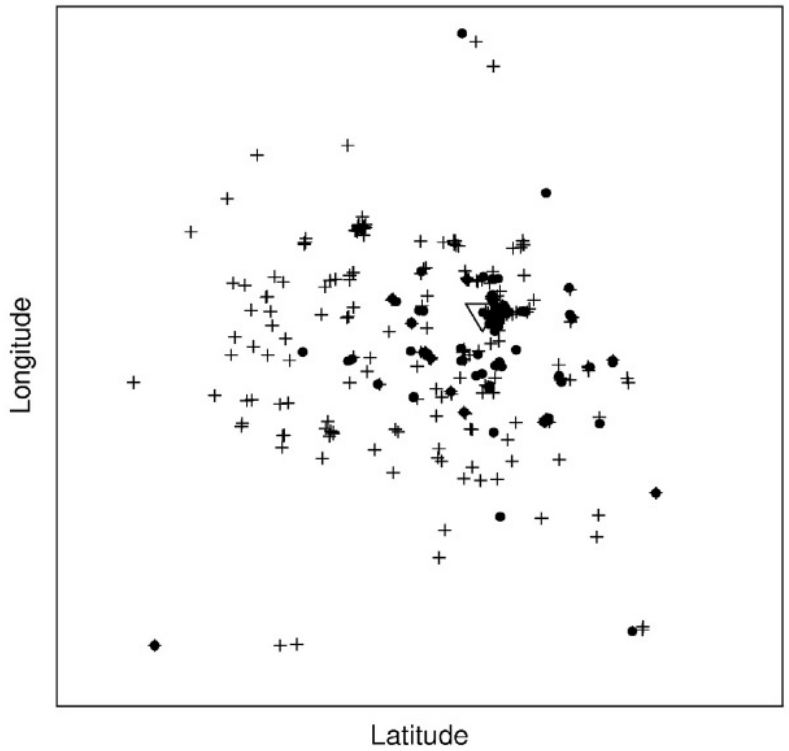


Fig. 1. Cases (•), controls (+) of residence locations and asbestos cement plant source (∇).

- This case-control sampling design aimed to assess the risk gradient as function of the distance from a putative source, constituted by an asbestos cement (AC) plant (active from 1907 to 1985)

- The data are an example of a heterogeneous Poisson point process in the plane, in which the intensity of cases in a given location depends by:

  - the number of cases
  - the distance from source
  - a series of individual covariates

- The sources of asbestos exposure are identified as:
  - *Occupational exposure* in the AC plant
  - *Domestic exposure*, that refers to either the indoor presence of asbestos materials such as asbestos fabrics of ironing tables, fireproof sheets for stoves and ovens, or AC materials and roofing in very close proximity to the house (e.g. garden, courtyard)
  - *Occupation in the AC industry of relatives and cohabitants*
- Different model for the distance from the AC plant were considered. In the practical, we will assume an exponential decay with threshold, following Diggle and Rowlingson, 1994: $f(d_i) = \alpha \exp(-\beta d^2)$ where α is the parameter capturing the excess relative risk at source, d is the distance from the source and β is the parameter of the exponential decrease function.

References and further reading

- Baddeley A, et al., (2016). *Spatial Point Patterns. Methodology and Applications with R*. Chapman & Hall.
- Baddeley A, *Analysing spatial point patterns in R*: https://research.csiro.au/software/wp-content/uploads/sites/6/2015/02/Rspatialcourse_CMIS_PDF-Standard.pdf
- DiMaggio C, *Spatial Epidemiology Notes*, http://www.columbia.edu/~cjd11/charles_dimaggio/DIRE/resources/spatialEpiBook.pdf
- Diggle PJ., (2014). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC Press.
- Illian J. et al., (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley.
- Banerjee S. et al., (2014). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall. (Chapter 8)
- Waller L. and Gotway CA., (2004). *Applied spatial statistics for public health data*. Wiley. (Chapter 5)
- Diggle P.J., (1985) A kernel method for smoothing point process data. Journal of the Royal Statistical Society, Series C (Applied Statistics), 34:138-147