點型態分析：距離分析方法

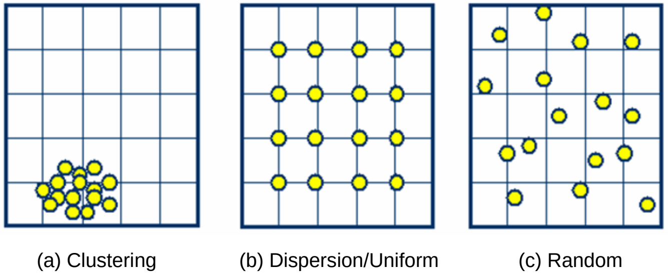
Point Pattern Analysis: Distance-based Methods

https://ceiba.ntu.edu.tw/1092Geog2017_

授課教師：溫在弘
E-mail: wenthung@ntu.edu.tw

Spatial Point Pattern Analysis

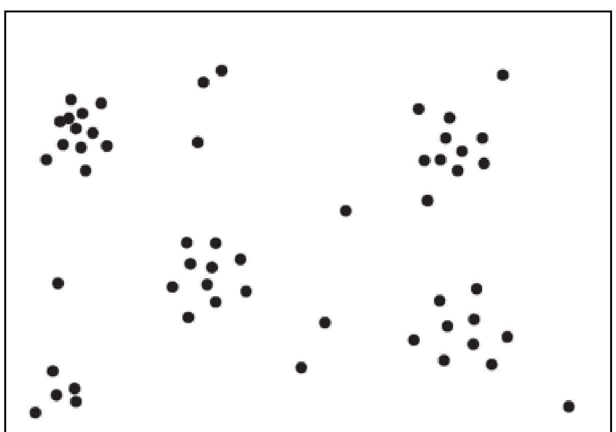
- Analyzing global patterns: Overview
  - Quadrat Analysis
  - Nearest Neighbor Analysis: K-order NNA
  - Distance-based Methods
    - G(d) and F(d) Functions
    - Ripley's K Functions
    - R codes



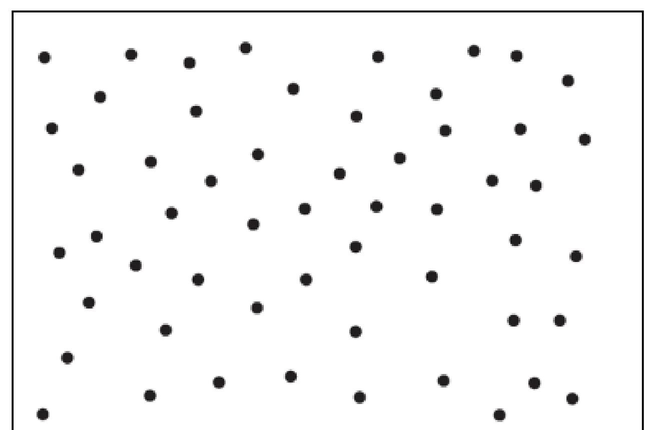
The Second-order Property: Distance-based Methods

- Nearest Neighbor Analysis (NNA)
- G Function: G(d)
- F Function: F(d)
- Ripley’s K Functions: K (d) and L(d)

(PP1)



(PP2)



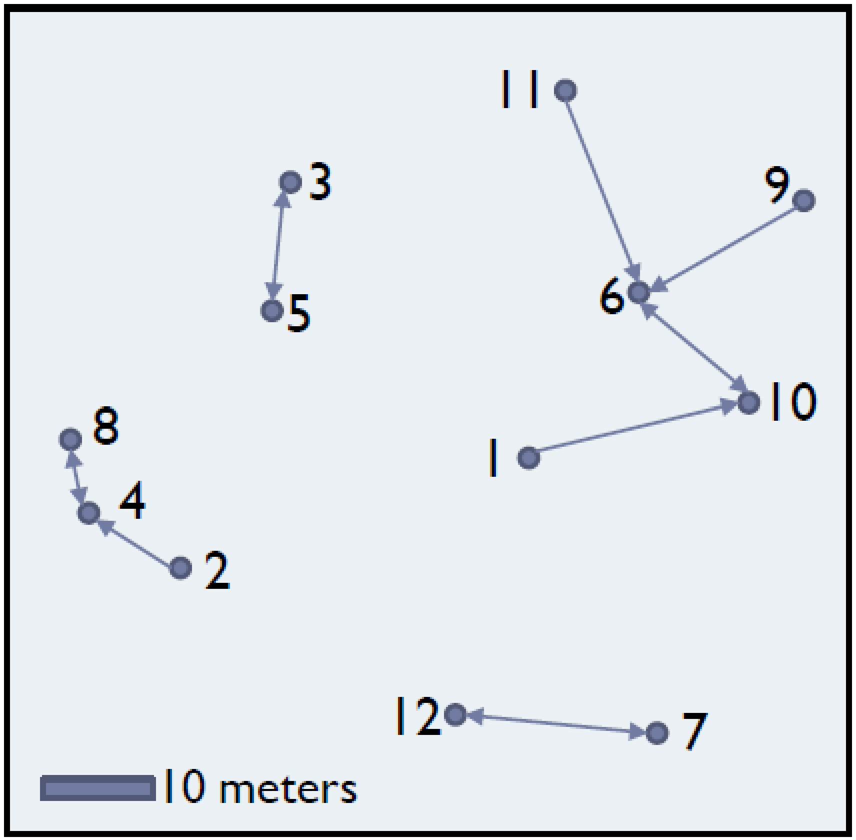
複習 G Function: G(d)

- **The G function** is defined as the *cumulative frequency distribution* of the nearest-neighbor distances

$$G(d) = \frac{\#(d_{\min}(\mathbf{x}_i) < d)}{n}$$

G(d) gives the proportion (since the count is divided by n) of nearest-neighbor distances that are less than distance *d*.

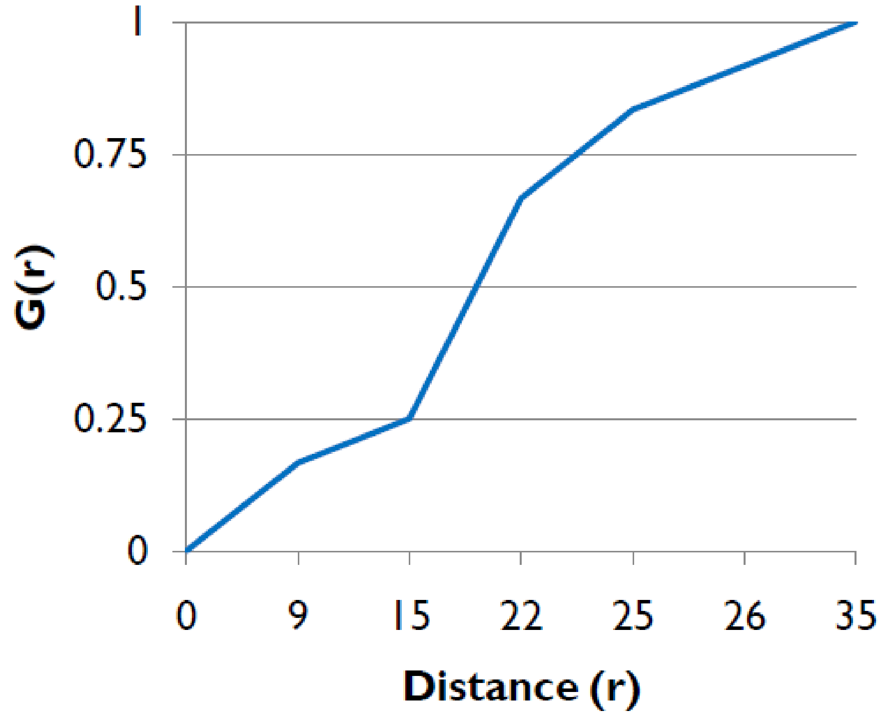
計算範例



Event	x	y	Nearest neighbor	r_{min}
1	66.22	32.54	10	25.59
2	22.52	22.39	4	15.64
3	31.01	81.21	5	21.14
4	9.47	31.02	8	24.81
5	30.78	60.10	3	9.00
6	75.21	58.93	10	21.14
7	79.26	7.68	12	21.94
8	8.23	39.93	4	9.00
9	98.73	42.53	6	21.94
10	89.78	42.53	6	21.94
11	65.19	92.08	6	34.63
12	54.46	8.48	7	24.81

計算範例：產生 G Function, G(r)

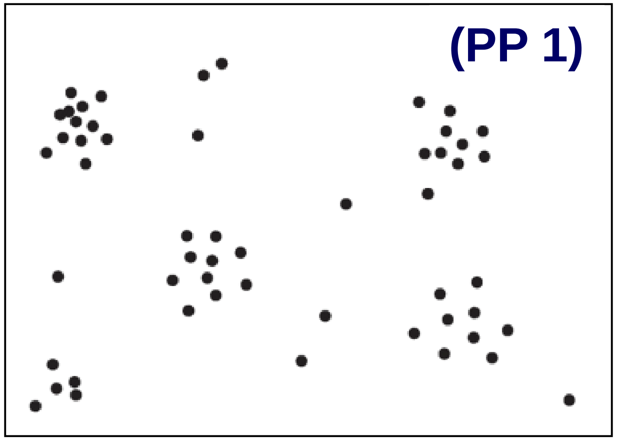
Event	x	y	Nearest neighbor	r_{min}
1	66.22	32.54	10	25.59
2	22.52	22.39	4	15.64
3	31.01	81.21	5	21.14
4	9.47	31.02	8	24.81
5	30.78	60.10	3	9.00
6	75.21	58.93	10	21.14
7	79.26	7.68	12	21.94
8	8.23	39.93	4	9.00
9	98.73	42.53	6	21.94
10	89.78	42.53	6	21.94
11	65.19	92.08	6	34.63
12	54.46	8.48	7	24.81

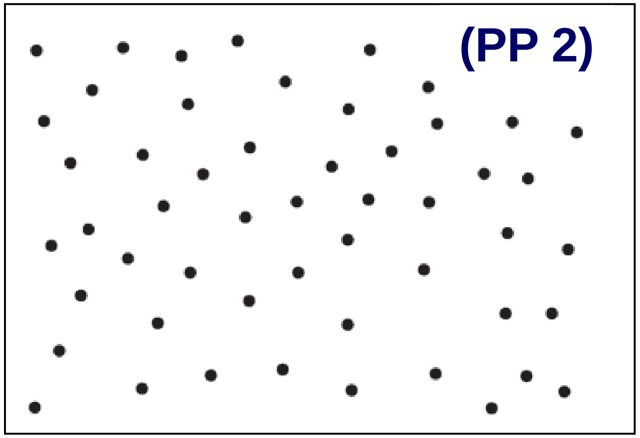


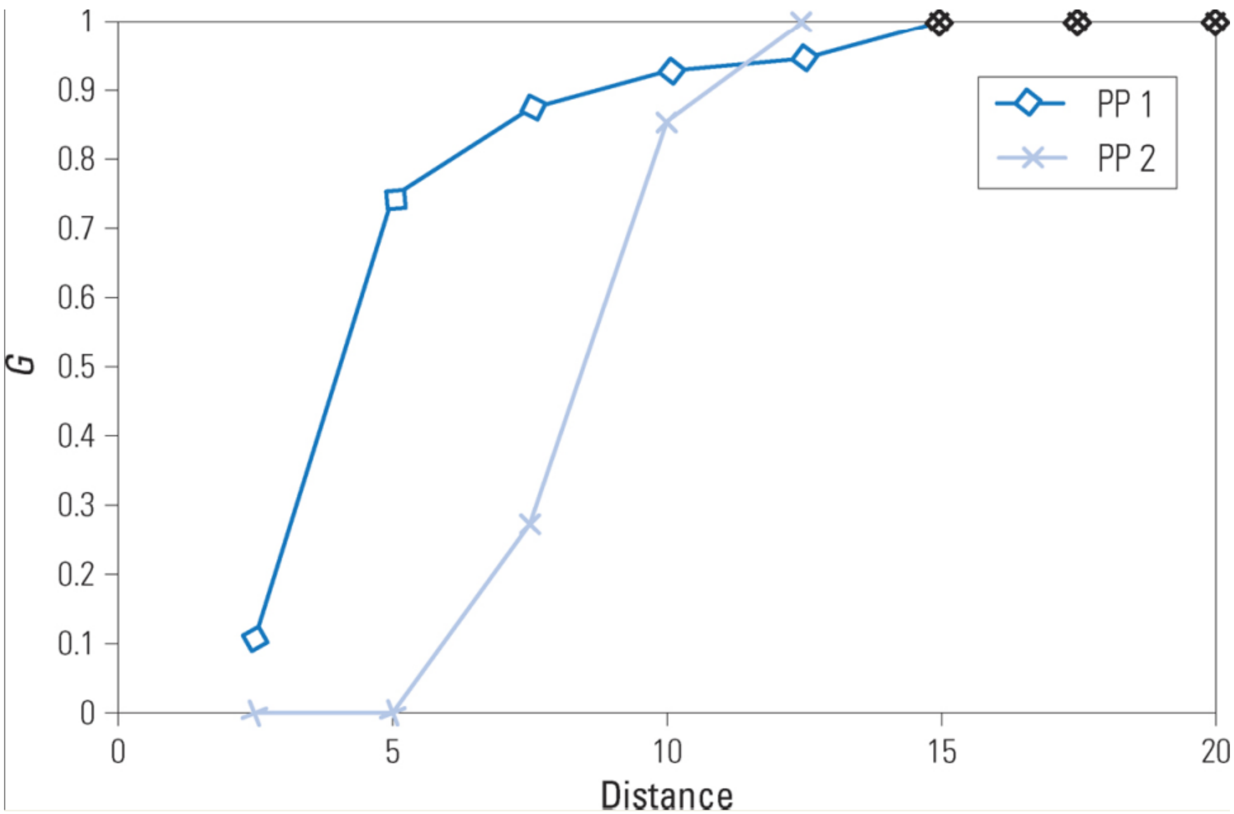
G Function的解讀

- The shape of G-function tells us the way the events are spaced in a point pattern
- **Clustered**：G increases rapidly at short distance
- **Uniform**：G increases slowly up to distance where most events spaced, then increases rapidly

範例





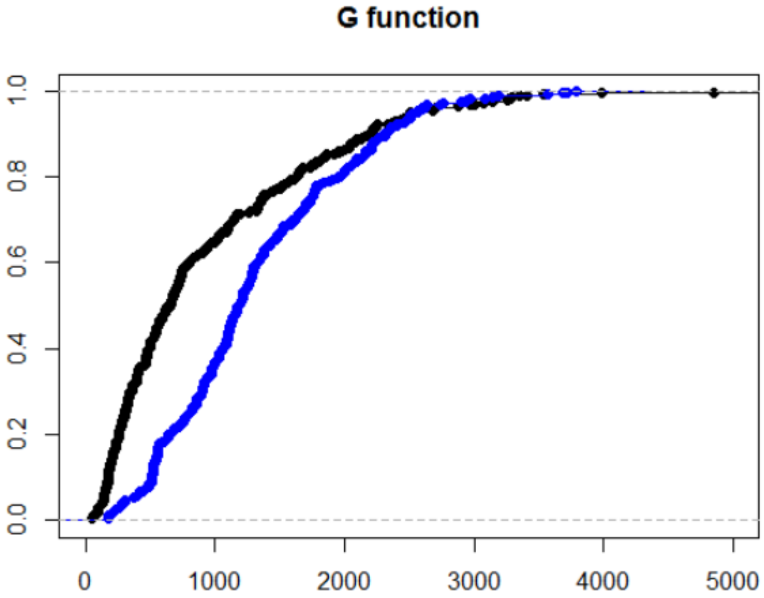


R Lab: G(d) Function

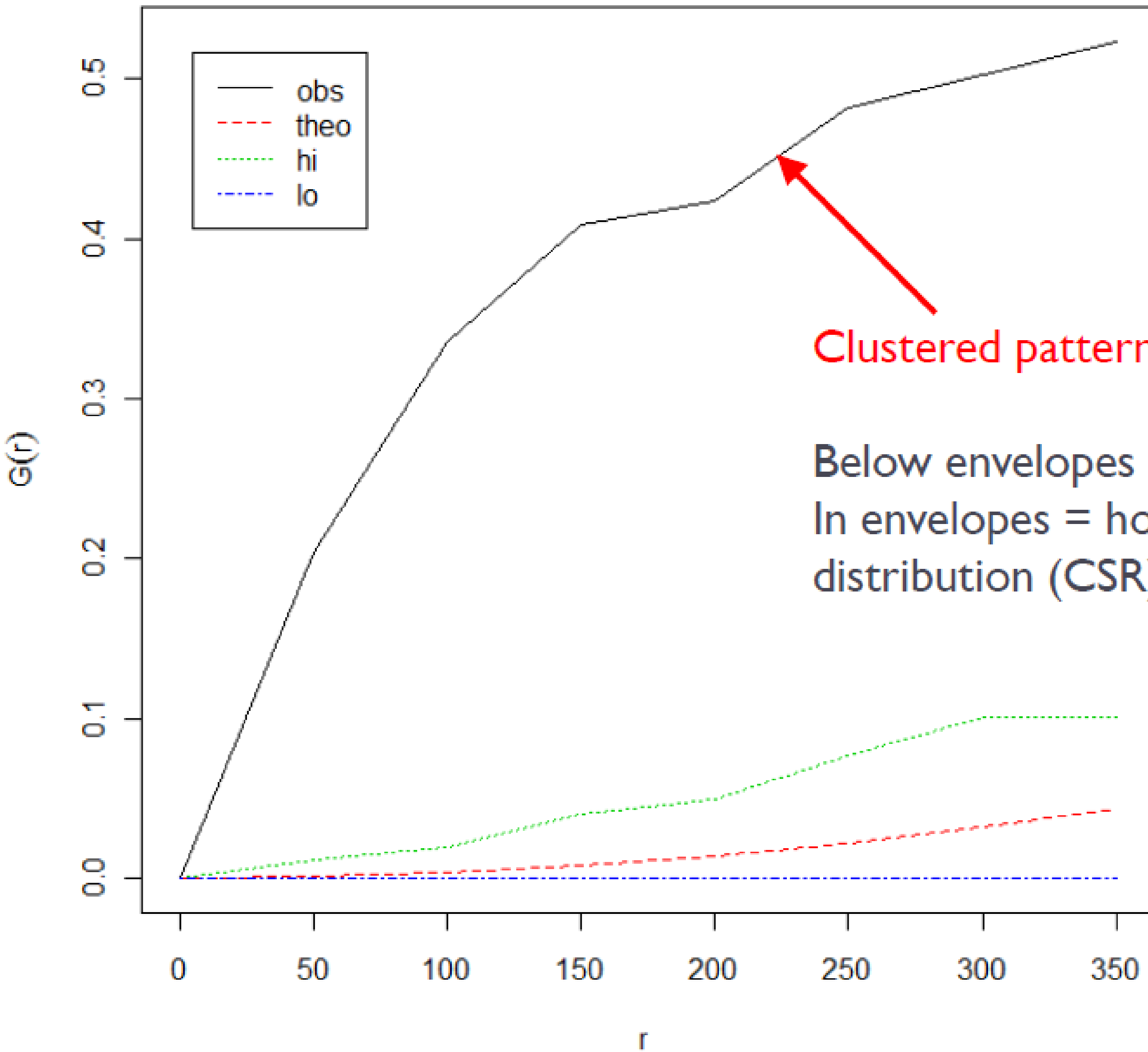
```
nnd<-nndist(School.ppp, k=1)
G = ecdf(nnd)
plot(G, main="G function", xlim=c(0,5000))
```

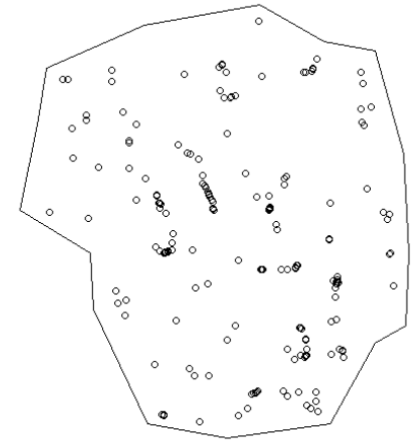
```
TN.Windows<-owin(xrange=x.range, yrange=y.range)
nn1<-rpoint(424, win=TN.Windows)
plot(nn1)
```

```
nnd1<-nndist(nn1, k=1)
G1 = ecdf(nnd1)
lines(G1,col='blue')
```



範例





Clustered pattern (above the envelopes)

Below envelopes = regular pattern
In envelopes = homogeneous distribution (CSR)

F Function: F(d)

- The *F function* is similar to the *G function*, but instead of **the events a sample of point locations is selected randomly from anywhere in the study area**

$$F(d) = \frac{\#(d_{\min}(\mathbf{x}_i, X) < d)}{m}$$

where $d_{min}(x_i,X)$ is the minimum distance from the location x_i in the randomly selected set of locations to the nearest event in the point pattern, X, and m is the number of randomly selected locations.

The nearest-neighbor distances are computed for randomly selected locations and not for the point event locations.

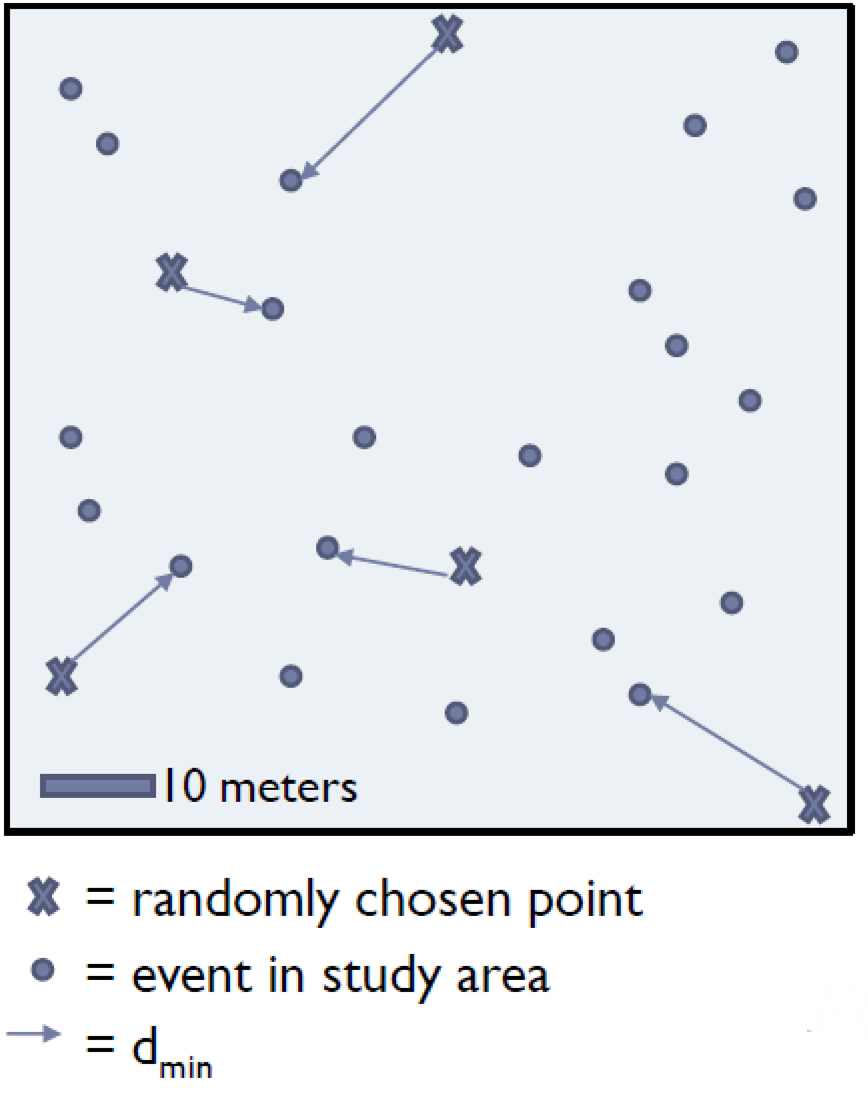
F Function 計算步驟

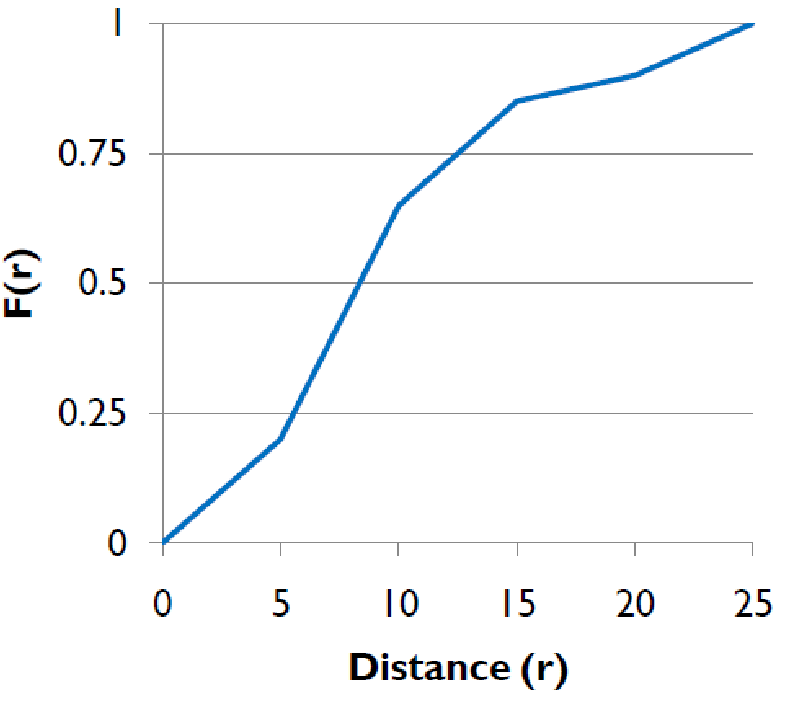
- Randomly select m points (p1, p2, …, pn)
- Calculate d_{min}(pi, s) as the minimum distance from location pi to any event in the point patterns
- Calculate F(d)

$$F(d) = \frac{\#[d_{\min}(p_i, s) < d]}{m}$$

$$= \frac{\#\text{ of point pairs where } r_{\min} \leq r}{\#\text{ sample points}}$$

F Function 計算範例

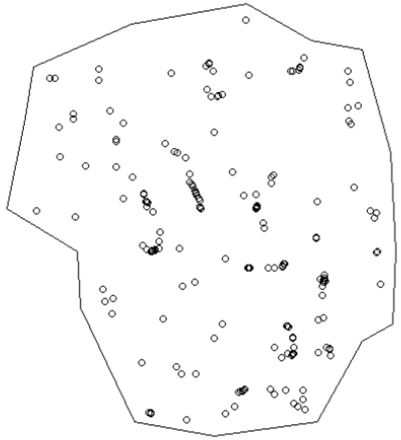


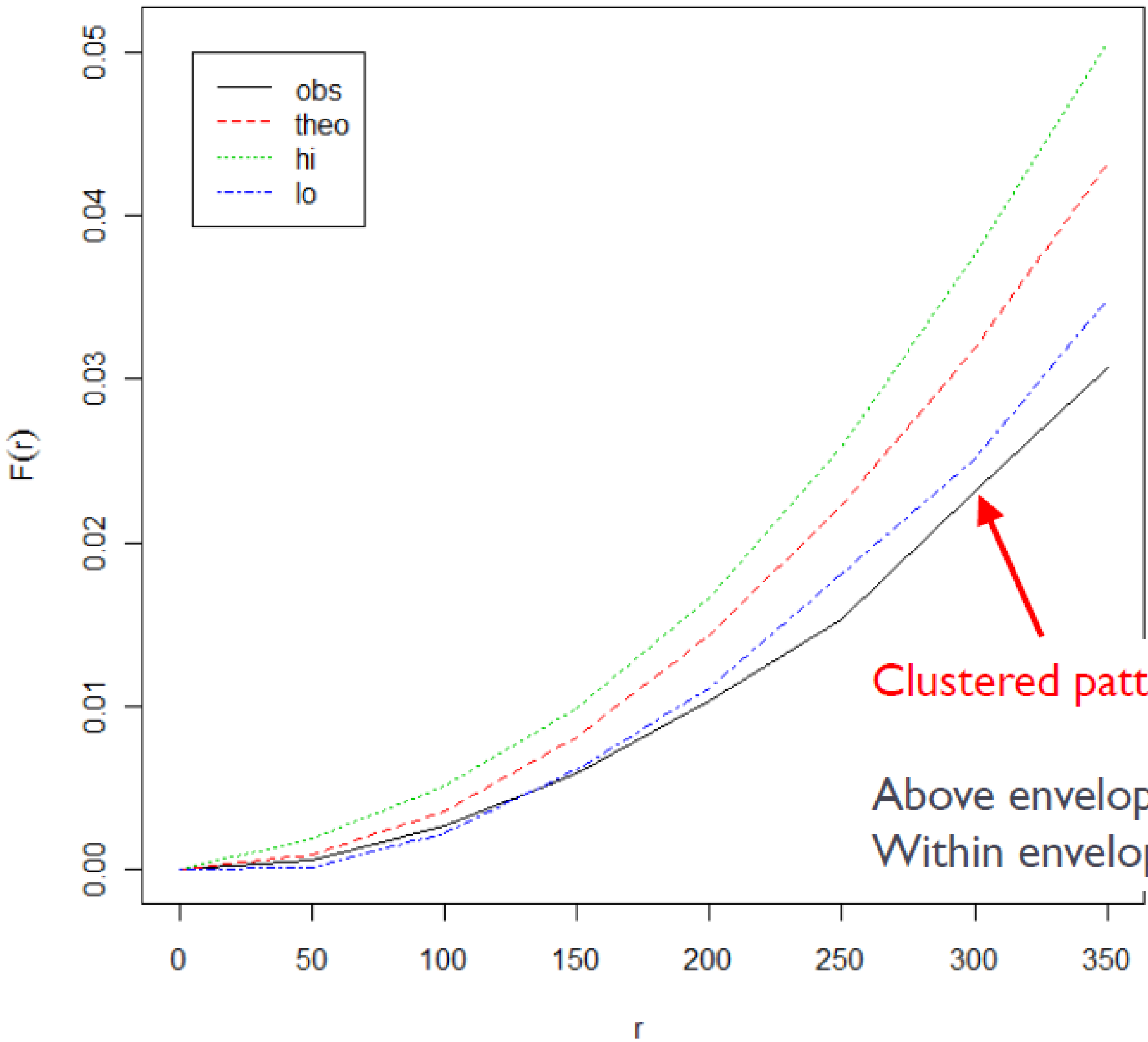


F Function的解讀

- **Clustered**： F(r) rises slowly at first, but more rapidly at longer distances
- **Uniform**： F(r) rises rapidly at first, then slowly at longer distances
- Examine significance by simulating “envelopes”

範例

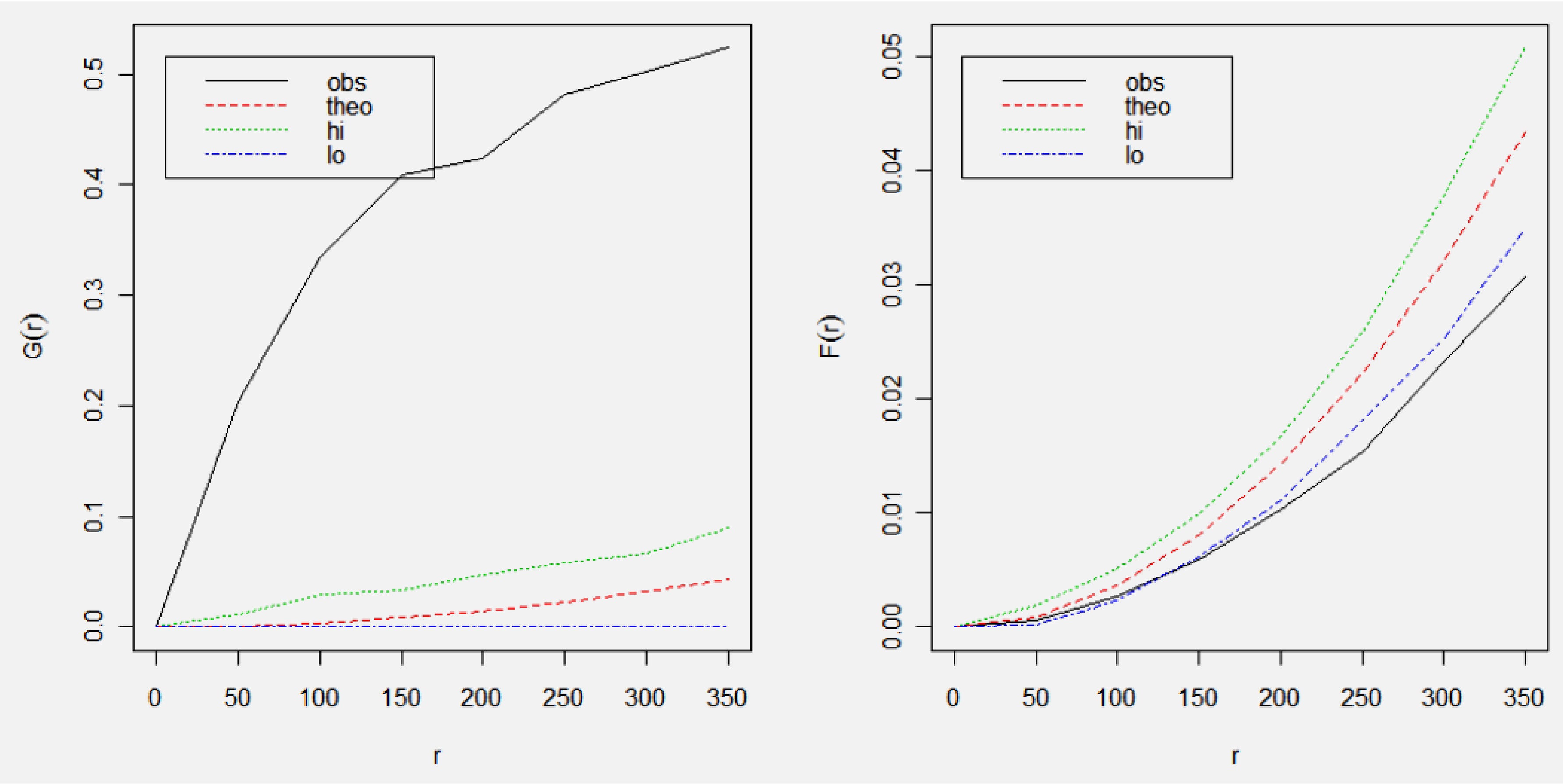




Clustered pattern (below the envelopes)

Above envelopes = regular pattern
Within envelopes = CSR

Comparison between G and F functions (spatial clustering tendency)



本週實作 F(d) Function

- Step 1: Loading school.shp
- Step 2: Generating Random Points: rpoint()

  (p1, p2, …, pn)
- Step 3: Calculate d_{min}(pi, s): st_distance() or nncross()
- Step 4: Calculate F(d): ecdf()
- Step 5: Monte Carlo Significance Test: for-loop
- Step 6: plotting the CDF curve: plot()
- **Final:** comparing with the result of **envelope** (school.ppp, fun=Fest)

簡介 nncross() function

nncross

Nearest Neighbour In Another Point Pattern

Given two point patterns `X` and `Y`, finds the nearest neighbour in `Y` of each point of `X`.

Keywords spatial, math

Usage

```
nncross(X, Y, iX=NULL, iY=NULL)
```

Arguments

X, Y Two point patterns (objects of class `"ppp"`).

iX, iY Optional identifiers, used to test whether a point in `X` is identical to a point in `Y`. See Details

R code

```
schools_sf <- st_read("Schools.shp")
county_sf <- st_read("TaiwanCounty.shp")

x.coor <- schools_sf$X_coor
y.coor <- schools_sf$Y_coor

index<- county_sf$COUNTY_ID == "67000" # Tainan city
TN_BND<- county_sf[index,]
xy <- st_coordinates(TN_BND)
x1<-rev(xy[,1]) # reverse the vector of X coord
y1<-rev(xy[,2]) # reverse the vector of Y coord
newxy<-cbind(x1, y1)
PTS_bnd <- owin(poly=newxy)
school.pp3<-ppp(x.coor, y.coor, window = PTS_bnd)
```

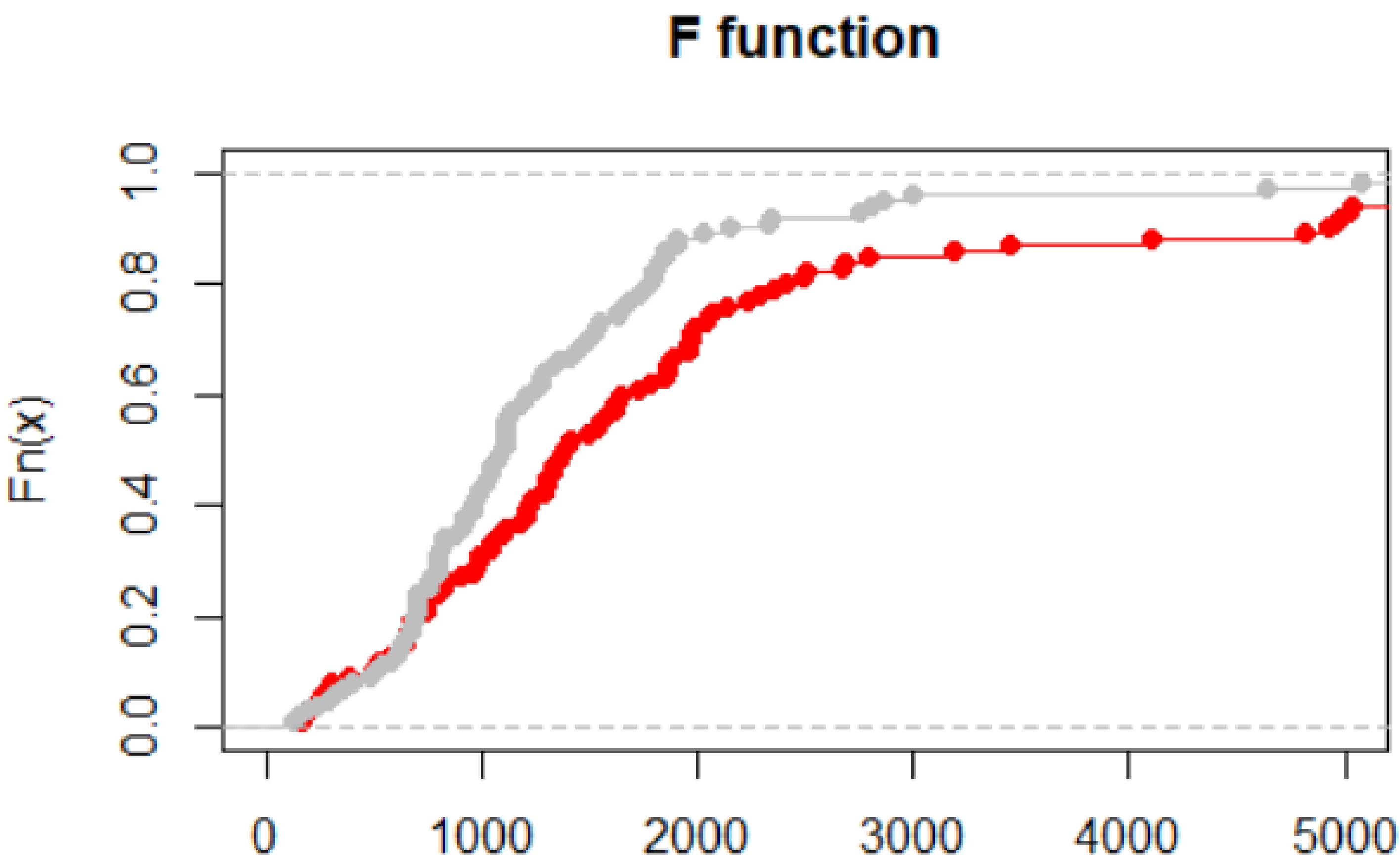
R code (cont'd)

```
RND <- rpoint(100, win= PTS_bnd)
nndist1 <- nncross(RND, school.pp3)
F <- ecdf(nndist1[,1])

RND2 <- rpoint(400, win= PTS_bnd)
nndist2 <- nncross(RND, RND2)
F_RND <- ecdf(nndist2[,1])

plot(F, main="F function", col = "red", xlim=c(0,5000))
lines(F_RND, col = "gray")
```

R code (cont'd)：預期結果



Ripley's K function

- Ripley's K function is a statistical method for point pattern analysis.
- summary of local dependence of spatial process → **second order** process
- expresses **number of expected events** within given distance of randomly chosen event

Ripley's K function

$$\widehat{K}(t) = \hat{\lambda}^{-1} \sum_{i} \sum_{j \neq i} w(l_i, l_j)^{-1} \frac{I(d_{ij} < t)}{N}$$

$$L(d) = \sqrt{\frac{K(d)}{\pi}} - d$$

Ripley's K function

$$\widehat{K}(t) = \hat{\lambda}^{-1} \sum_{i} \sum_{j \neq i} w(l_i, l_j)^{-1} \frac{I(d_{ij} < t)}{N}$$

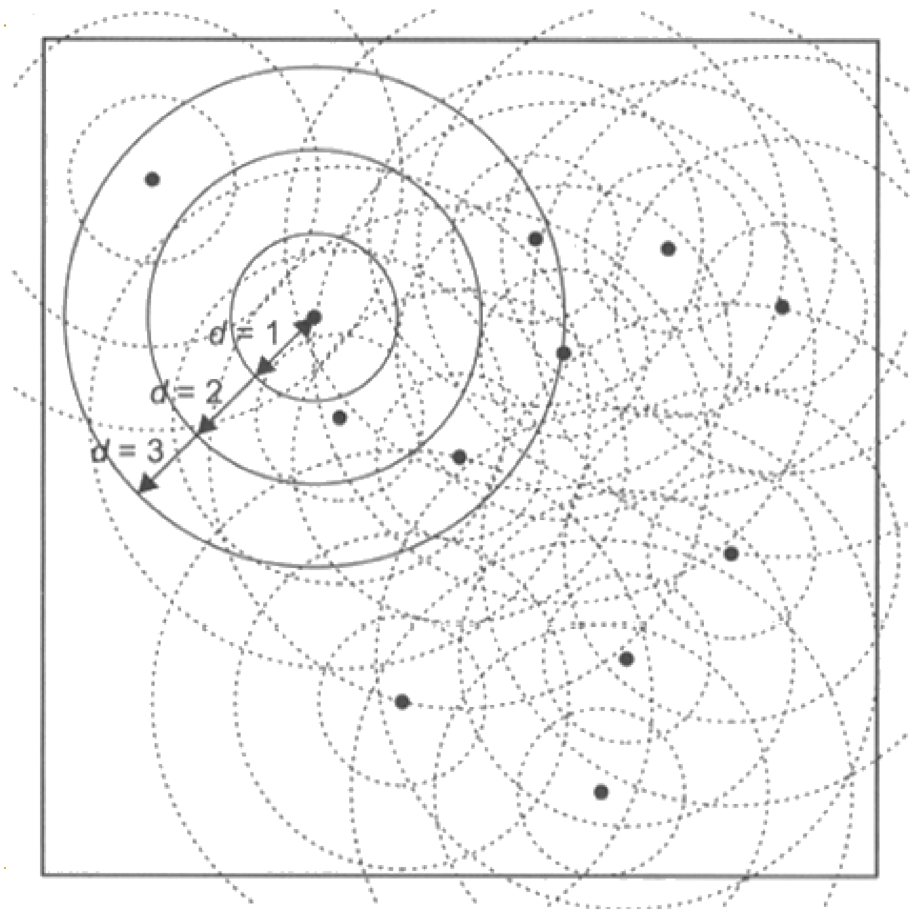
- *K(t)* describes characteristics of the point processes at many distance scales.

The K function is

$$K(t) = \lambda^{-1} \mathrm{E}[\text{number of extra events within distance } t \text{ of a randomly chosen event}]$$

λ is the density (number per unit area) of events. (N/A)

Steps of Estimating K-function



Four steps

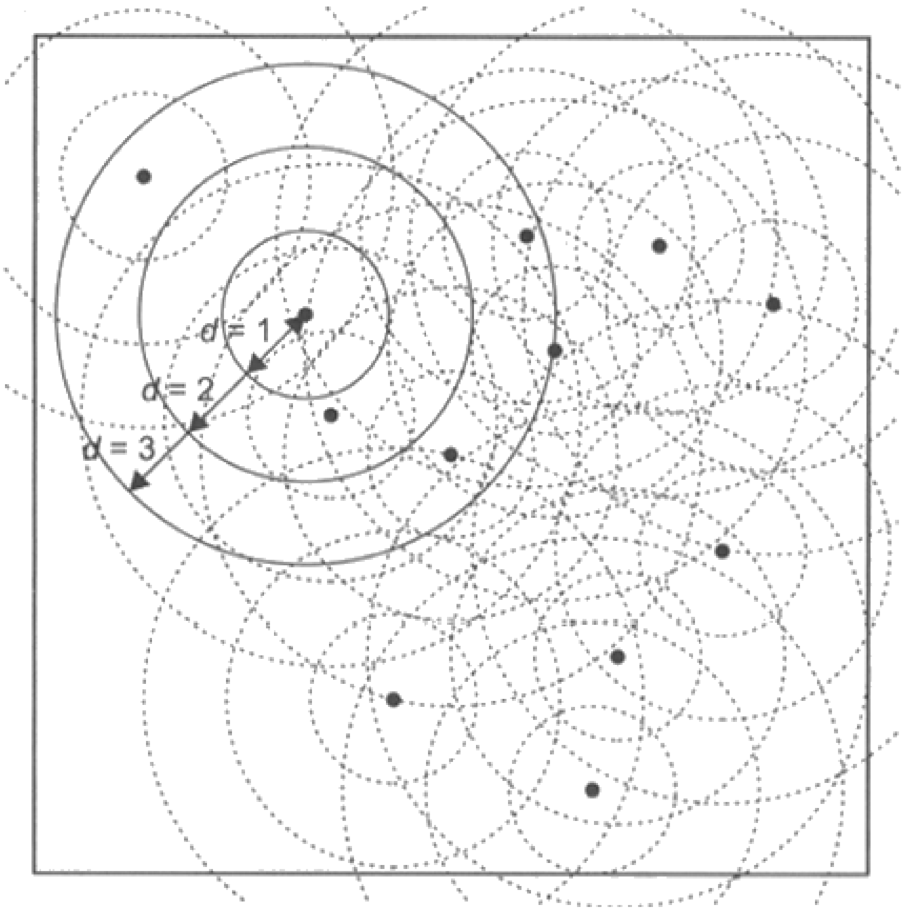
1) For a particular event, draw a circle centered at the event (***s**$_i$*) and with a radius of ***d***

2) Count the number of **<u>other events</u>** within the circle
$no.[S \in C(s_i, d)]$

3) Calculate the mean count of all events $\frac{\sum_{i=1}^{n} no.[S \in C(s_i, d)]}{n}$

4) This mean count is divided by the overall study area event density

Steps of Estimating K-function (cont'd)



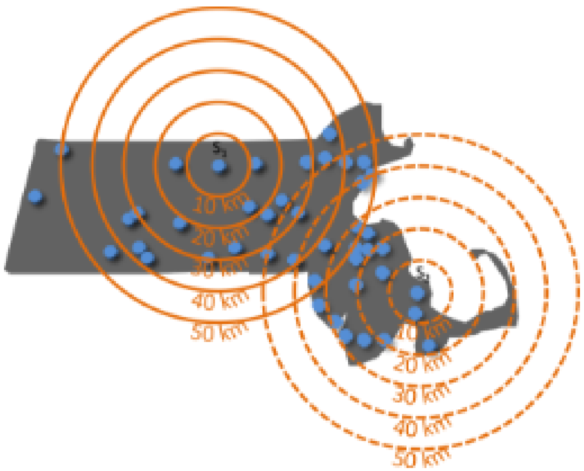
$$K(d) = \frac{\sum_{i=1}^{n} no.[S \in C(s_i, d)]}{n\lambda}$$

$$= \frac{a}{n} \bullet \frac{1}{n} \sum_{i=1}^{n} no.[S \in C(s_i, d)]$$

$\lambda = \frac{n}{a}$ is the study area event density

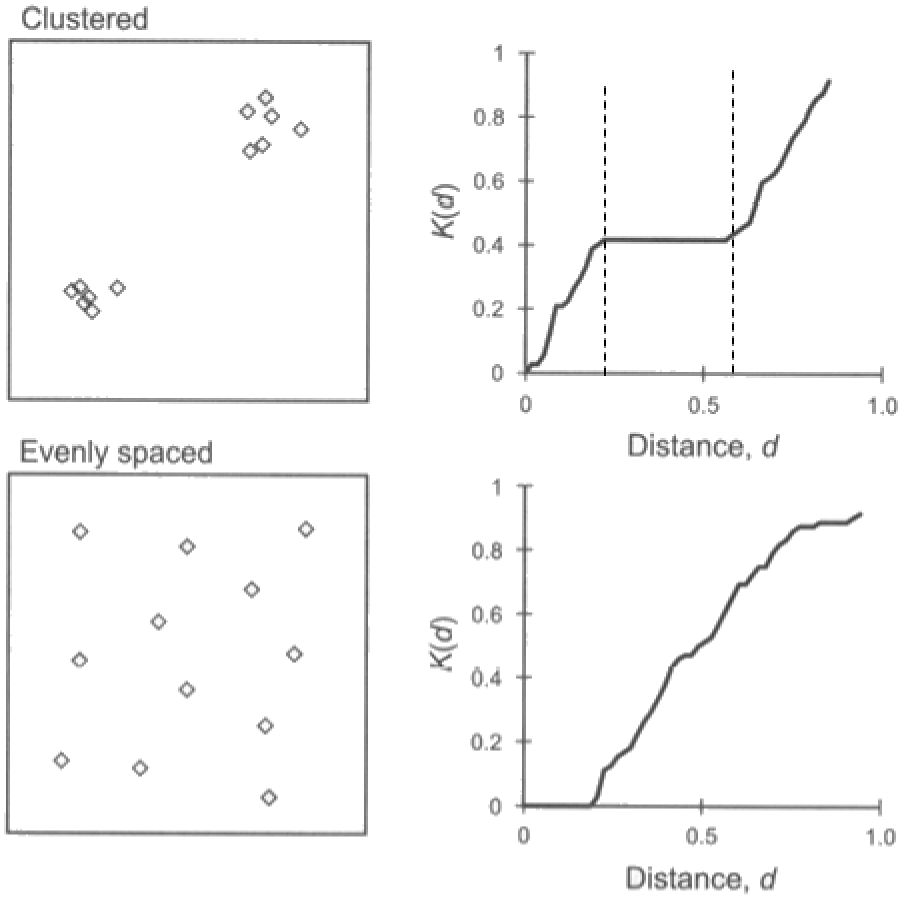
$$\widehat{K}(t) = \hat{\lambda}^{-1} \sum_{i} \sum_{j \neq i} w(l_i, l_j)^{-1} \frac{I(d_{ij} < t)}{N}$$

Estimating K-function: an Example



Distance band (km)	# events from s_1	# events from s_2	# events from s_i	K
10	0	1	...	0.012
20	3	5	...	0.067
30	9	14	...	0.153
40	17	17	...	0.269
50	25	23	...	0.419

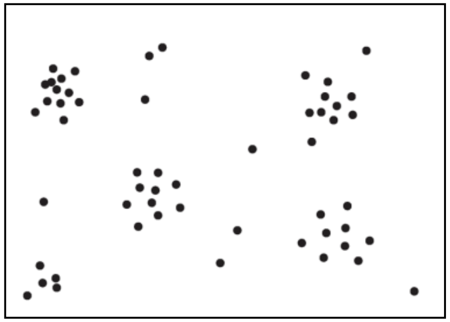
Interpretations of K-function



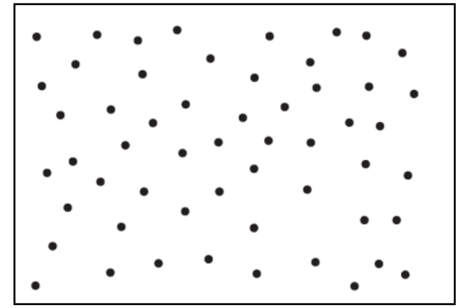
Clustered?

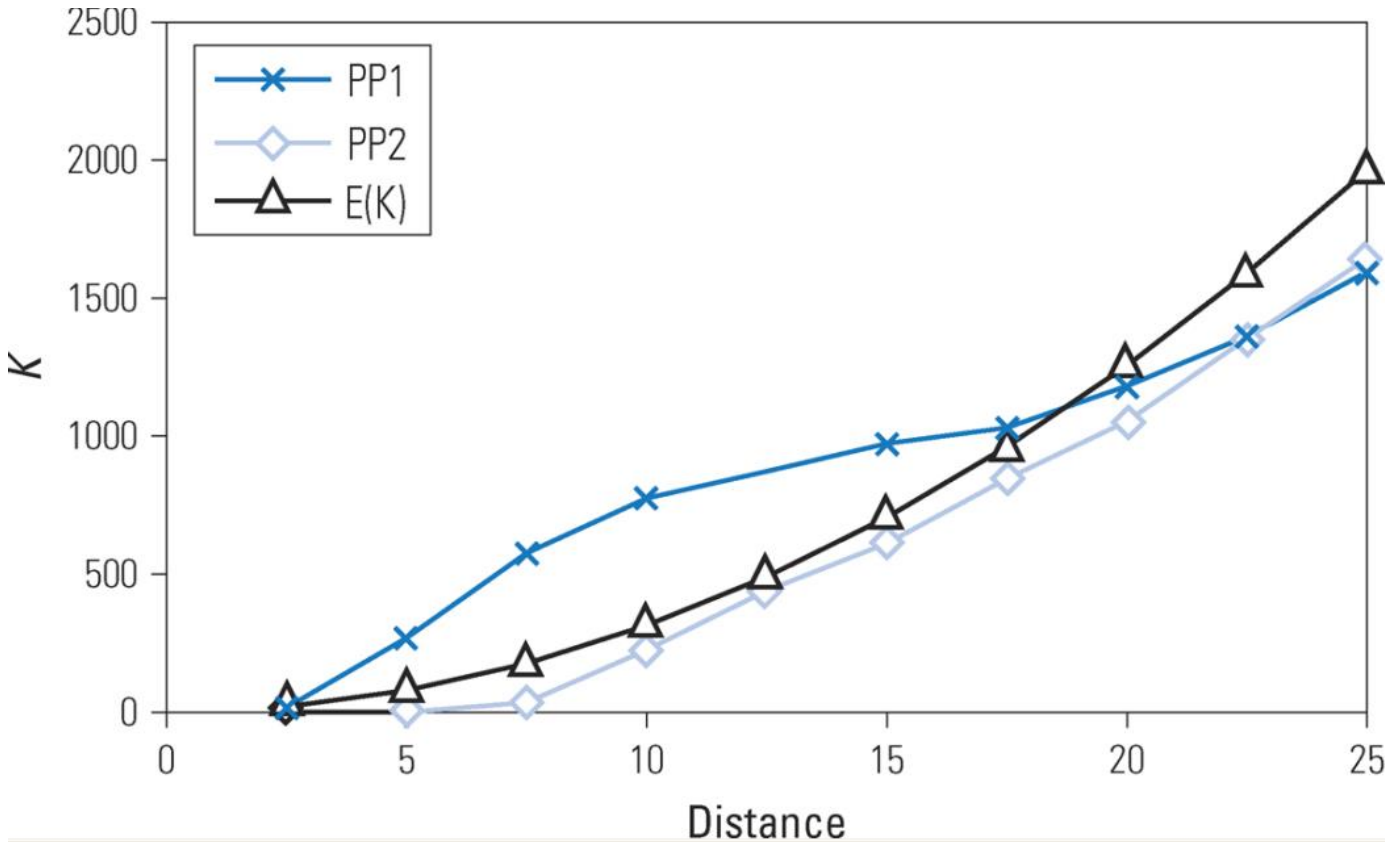
Evenly distributed?

(PP1)

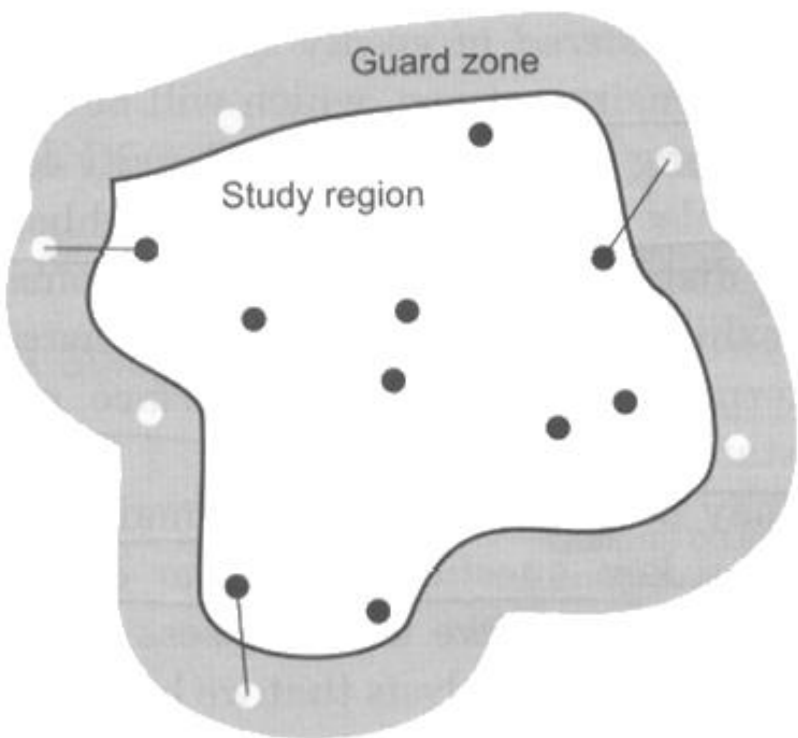


(PP2)





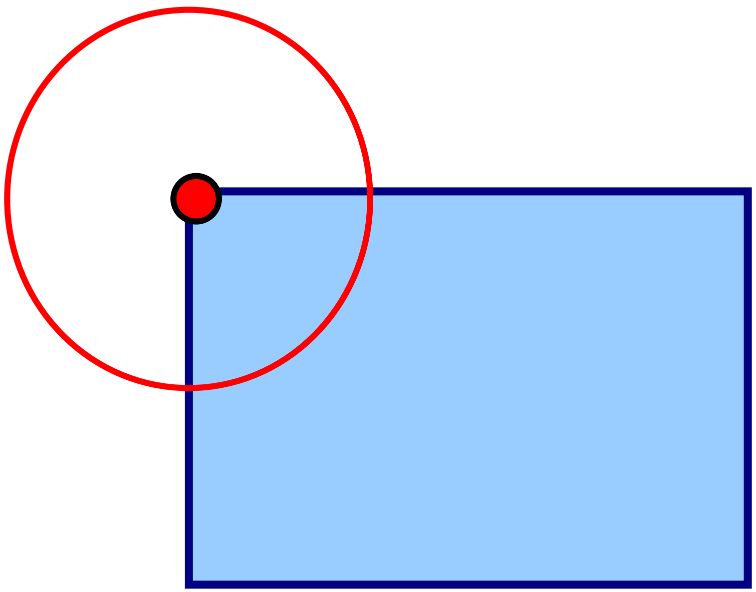
Edge effects



- Edge effects arise from the fact that events near the edge of the study area tend to have higher nearest-neighbor distances, even though they might have neighbors outside of the study area that are closer than any inside it.

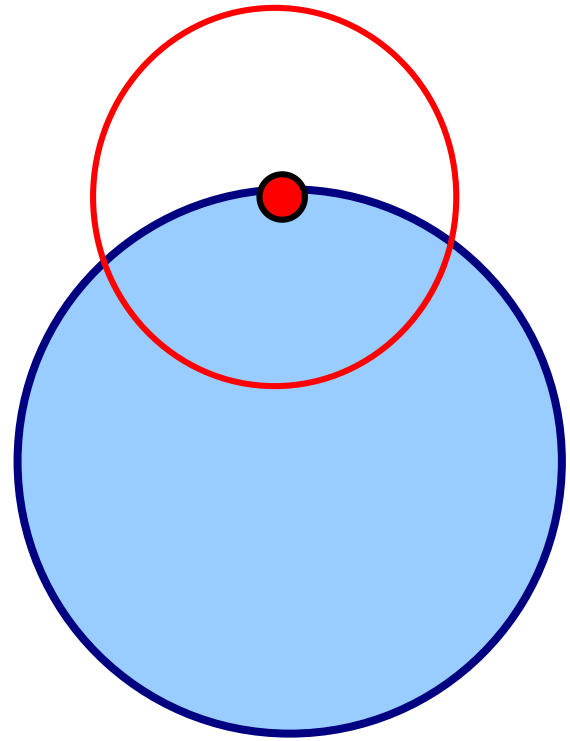
$$\widehat{K}(t) = \hat{\lambda}^{-1} \sum_{i} \sum_{j \neq i} \boxed{w(l_i, l_j)^{-1}} \frac{I(d_{ij} < t)}{N}$$

進行邊緣校正的示意圖與權重計算



proportion = 0.25;

the weight varies from 1 to 4



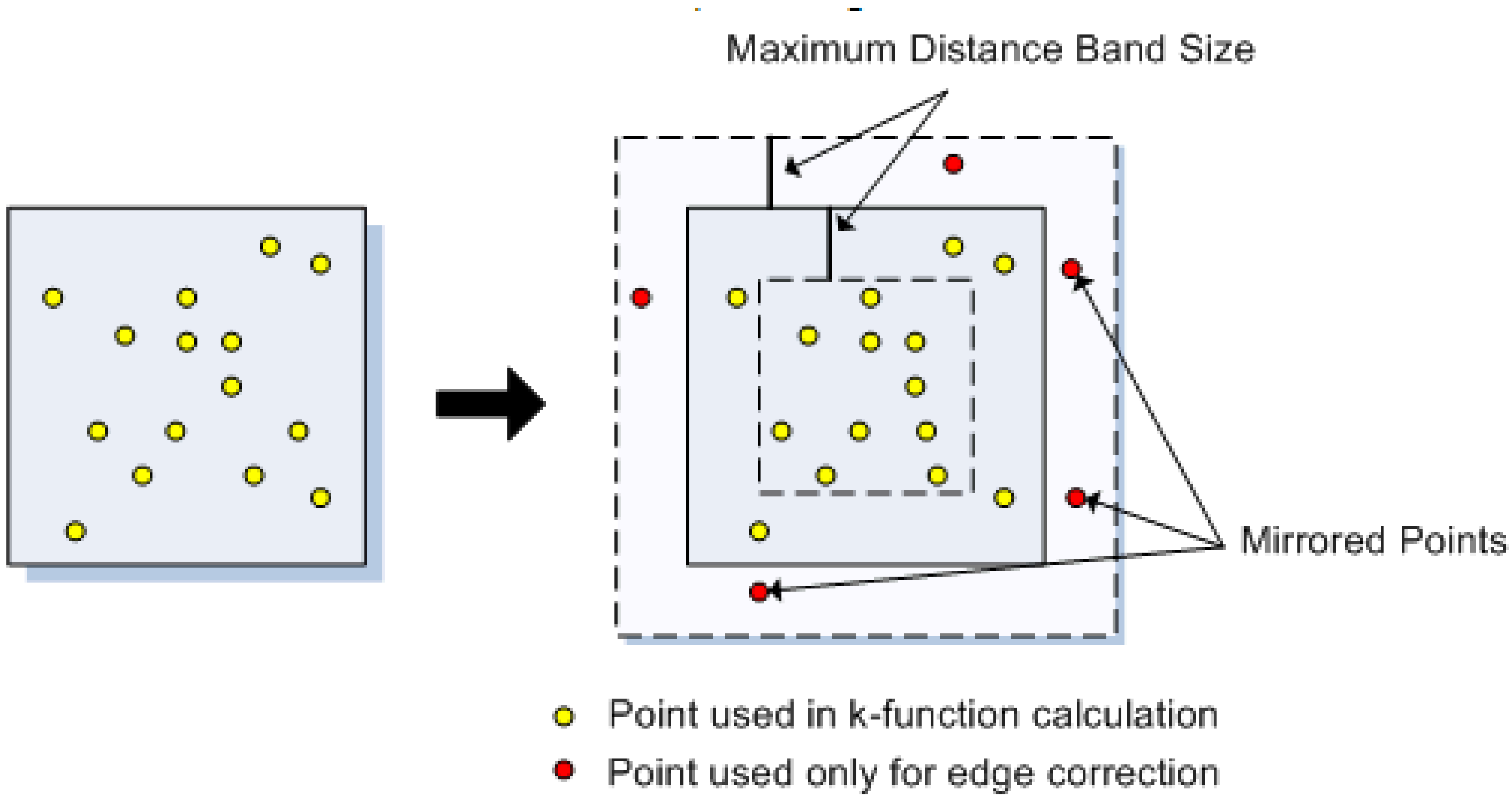
the weight varies

from 1 to 2.3834

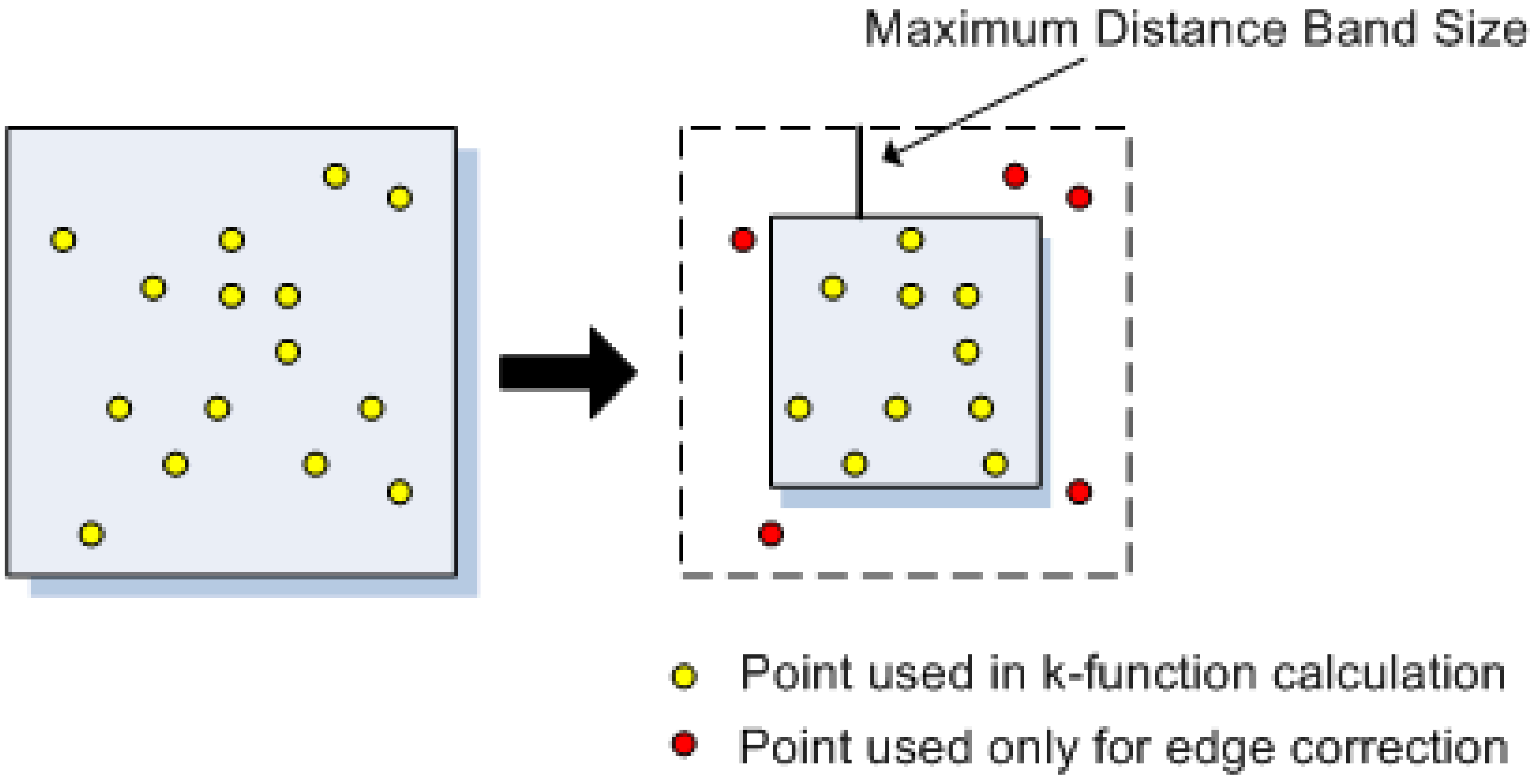
常見的邊緣校正方法
SIMULATE_OUTER_BOUNDARY_VALUES

- This method **creates points outside the study area boundary** that mirror those found inside the boundary in order to correct for underestimates near the edges.
- Points that are within a distance equal to **the maximum distance band of an edge of the study area** are mirrored.
- The mirrored points are used so that edge points will have more accurate neighbor estimates.

SIMULATE_OUTER_BOUNDARY_VALUES（圖示）



REDUCE_ANALYSIS_AREA（圖示）



K-function的公式說明

$$\widehat{K}(t) = \hat{\lambda}^{-1} \sum_{i} \sum_{j \neq i} w(l_i, l_j)^{-1} \frac{I(d_{ij} < t)}{N}$$

λ = N/A

$$\hat{K}(h) = \frac{R}{n^2} \sum \sum_{i \neq j} \frac{I_h(d_{ij})}{w_{ij}}$$

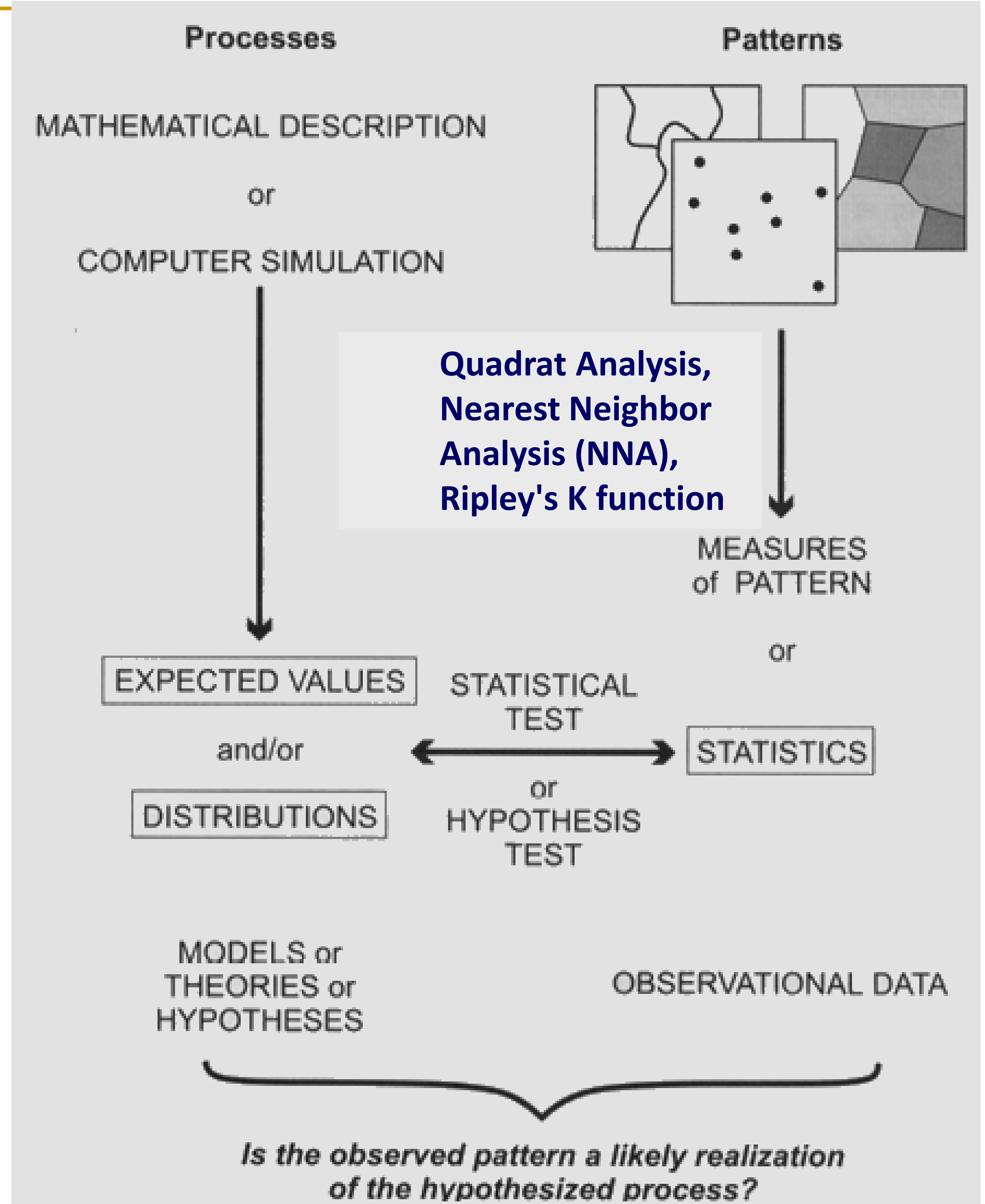
area of R → R

number of points → n

$I_h(d_{ij})$: dummy variable
1 if $d_{ij} \leq h$
0 otherwise

w_{ij}: edge correction
the proportion of circumference of circle
(centered on point i, containing point j)
=1 if whole circle in the study area

Assessing Point Patterns Statistically



Assessing Point Patterns Statistically

- The common use of Ripley's K(t) function is to test **C**omplete **S**patial **R**andomness (CSR)
  - test whether the observed events are consistent with a **homogeneous Poisson process**.

$$L(d) = \sqrt{\frac{K(d)}{\pi}} - d$$

Under CSR, $\sqrt{\frac{K(d)}{\pi}} = d$

→ L(d) = 0.

推導 Expected value of K(d)

- 設 λ(intensity) = a (每單位面積有a個點)

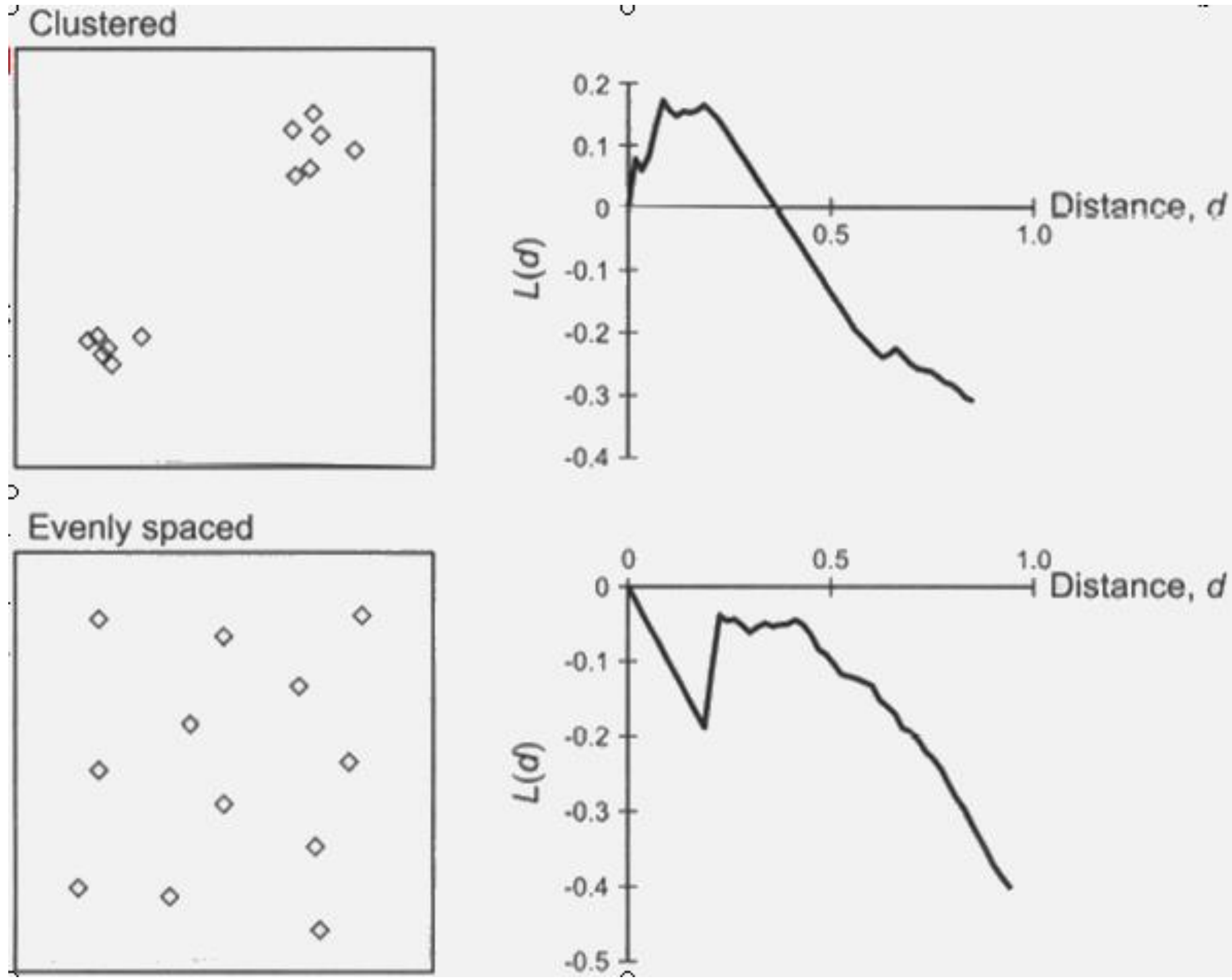
$$K(d) = \frac{\sum_{i=1}^{n} no.[S \in C(s_i, d)]}{n\lambda}$$

- **Expected value** of K(d) = (n *a* πd^2) / (n*a) = πd^2

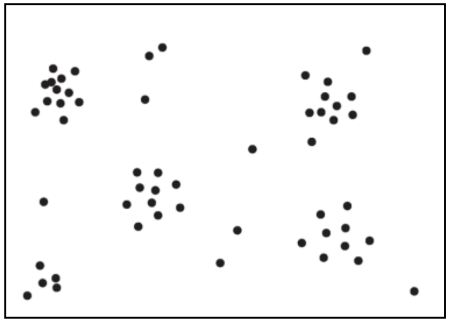
$$\sqrt{\frac{E(K(d))}{\pi}} = d \qquad E(L(d)) = \sqrt{\frac{E(K(d))}{\pi}} - d = 0$$

$$L(d) = \sqrt{\frac{K(d)}{\pi}} - d$$

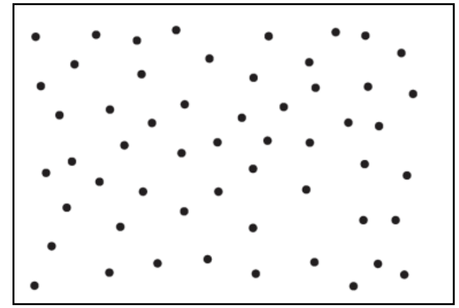
Under CSR, L(d) = 0.

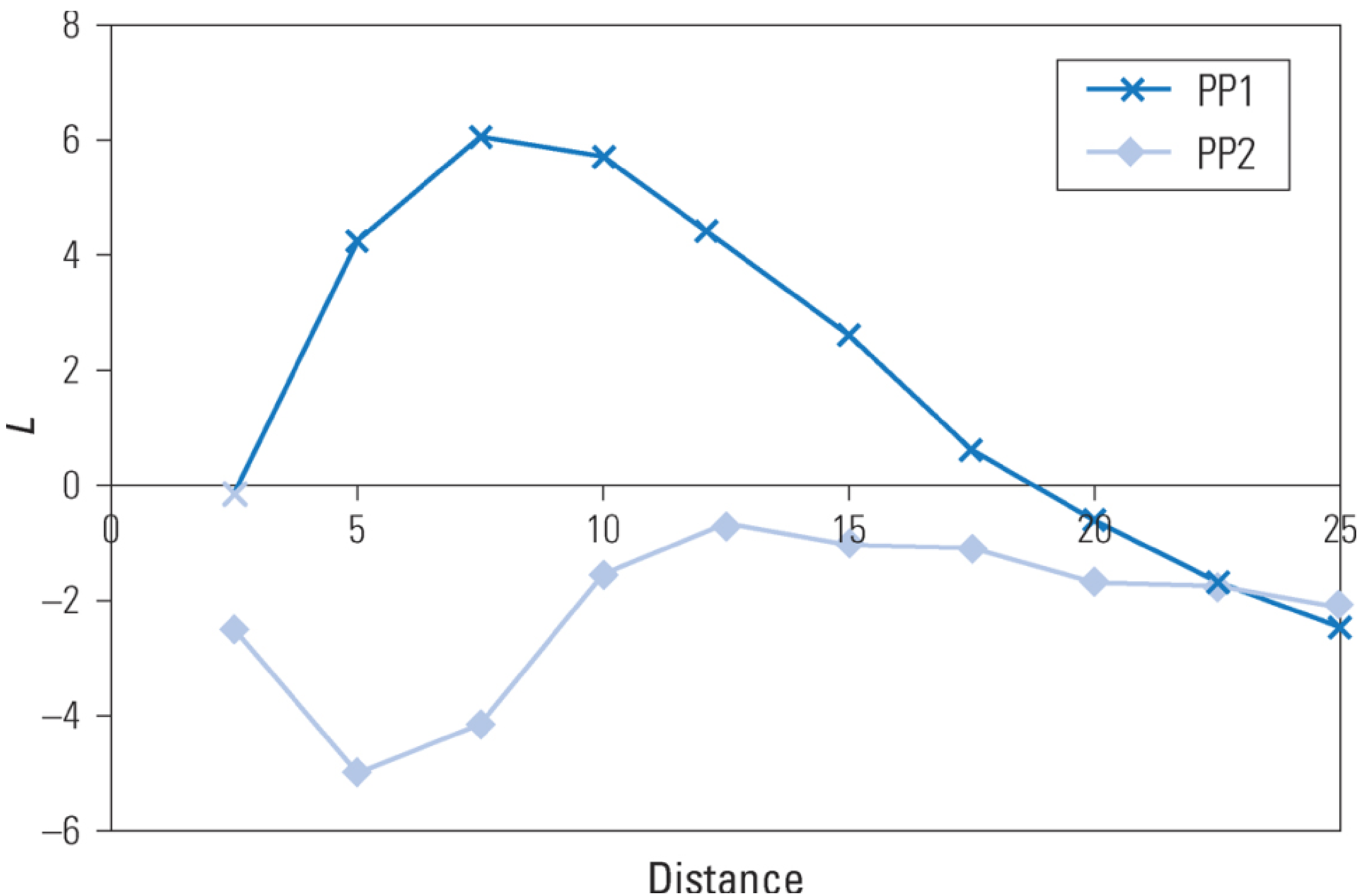


(PP1)

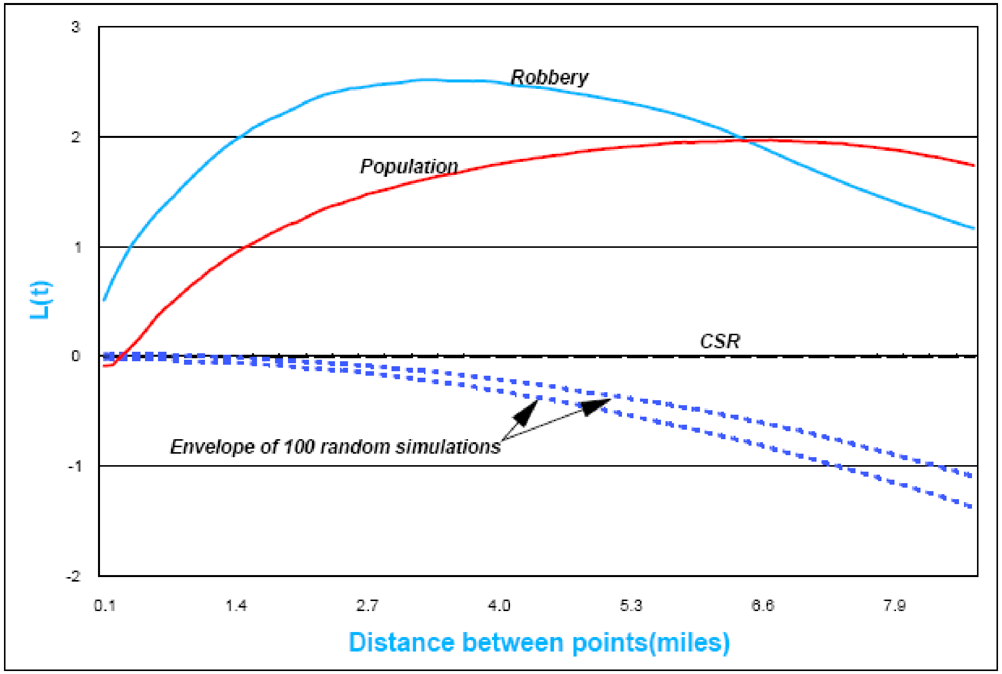


(PP2)

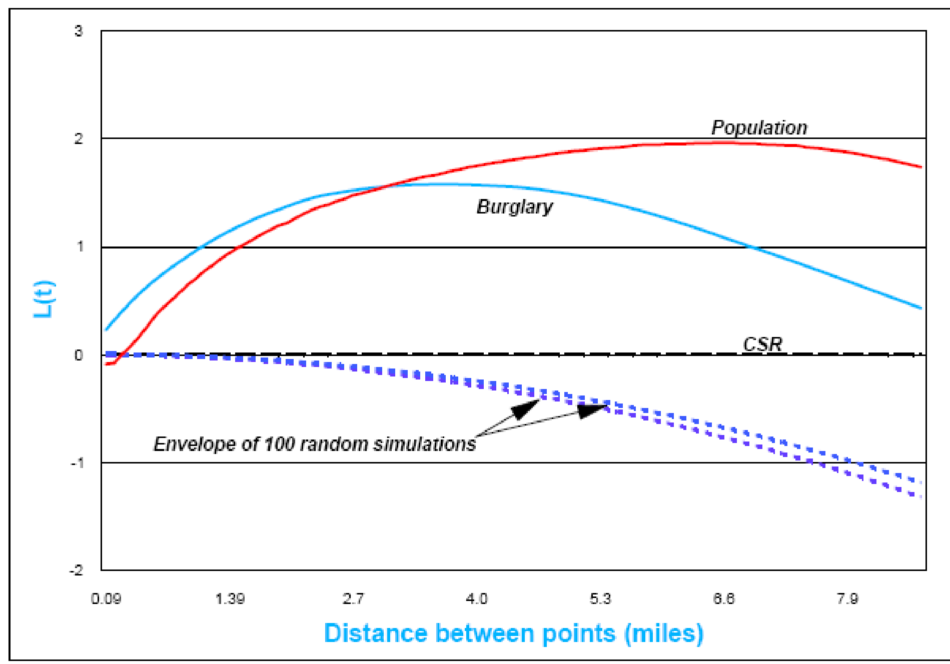




Robbery Cases



Burglary Cases



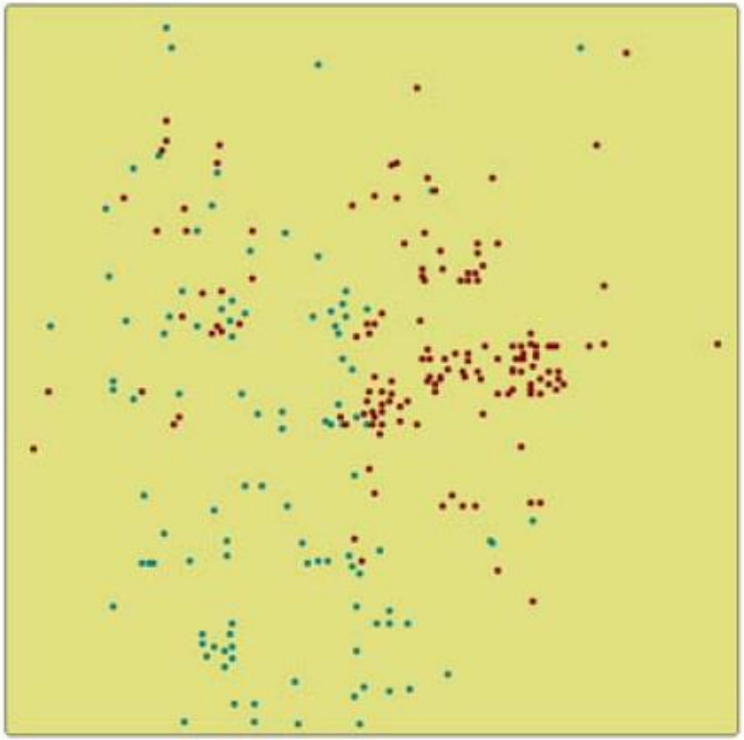
延伸的分析議題

如何利用 K-function 比較Red dots與Blue dots的空間群聚結構?

Bivariate K-function（加分題）

Is the distribution of one set of events related to the distribution of the other?

Black (red dots) and white (blue dots) crimes in Oklahoma



R code: K(d) and L(d)

Kest

K-Function

Estimates Ripley's reduced second moment function $K(r)$ from a point pattern in a window of arbitrary shape.

Keywords spatial, nonparametric

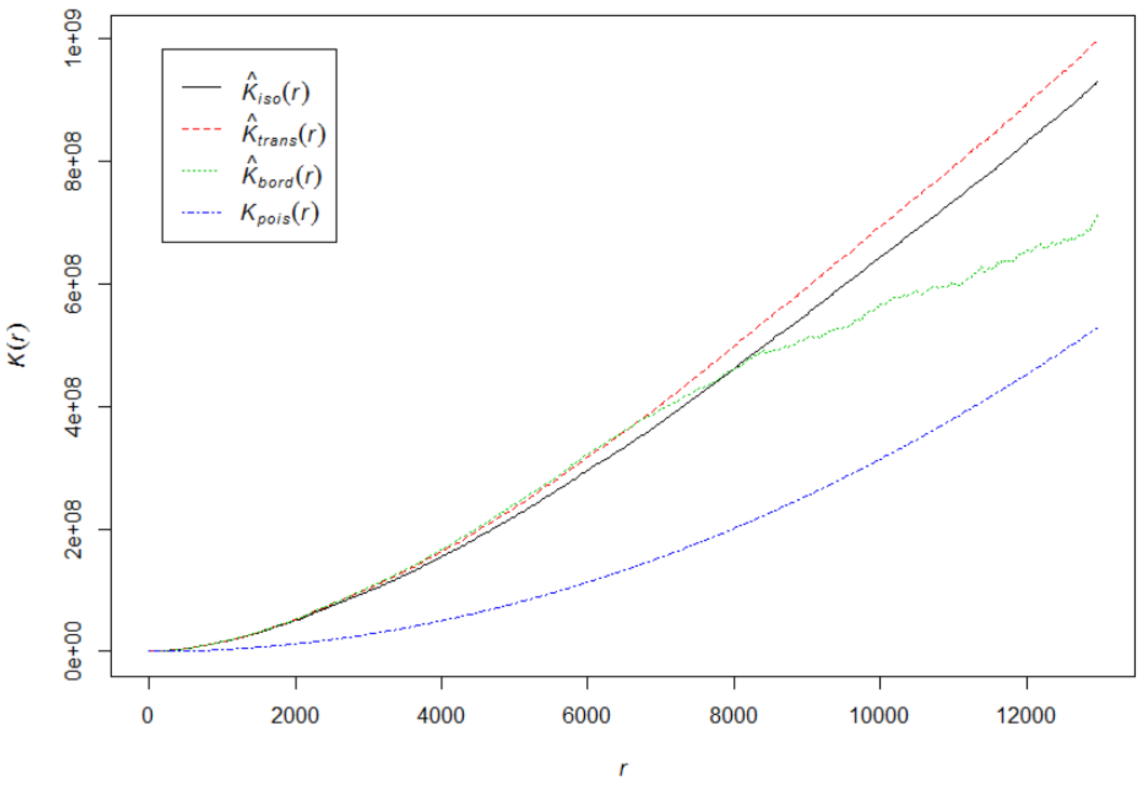
Usage

```
Kest(X, …, r=NULL, rmax=NULL, breaks=NULL,
     correction=c("border", "isotropic", "Ripley", "translate"),
    nlarge=3000, domain=NULL, var.approx=FALSE, ratio=FALSE)
```

```
> K <- Kest(School.ppp)
> plot(K, main=NULL)
```

R code: K(d)

```
> K <- Kest(School.ppp)
> plot(K, main=NULL)
```



Border Correction Methods

the border method or “reduced sample” estimator (see Ripley, 1988). This is the least efficient (statistically) and the fastest to compute. It can be computed for a window of arbitrary shape.

isotropic/Ripley

Ripley's isotropic correction (see Ripley, 1988; Ohser, 1983). This is implemented for rectangular and polygonal windows.

translate/translation

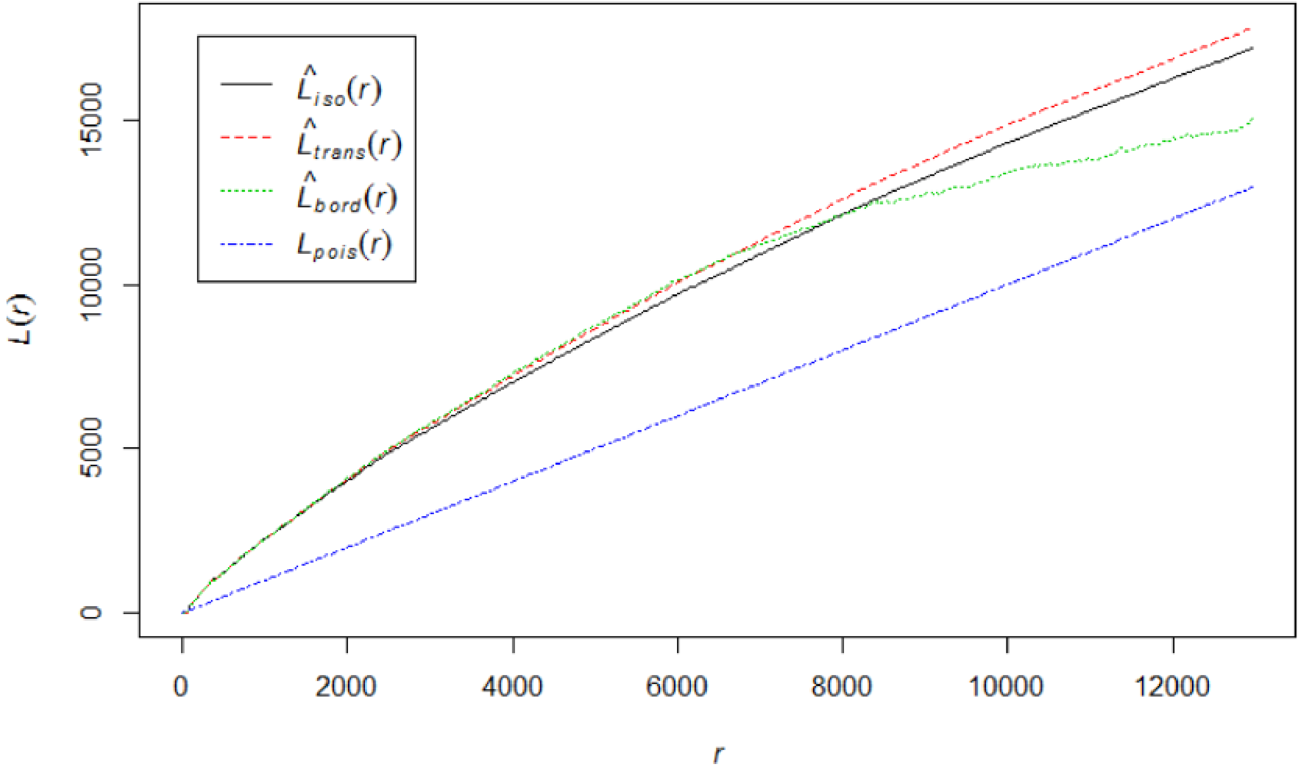
Translation correction (Ohser, 1983). Implemented for all window geometries, but slow for complex windows.

rigid

Rigid motion correction (Ohser and Stoyan, 1981). Implemented for all window geometries, but slow for complex windows.

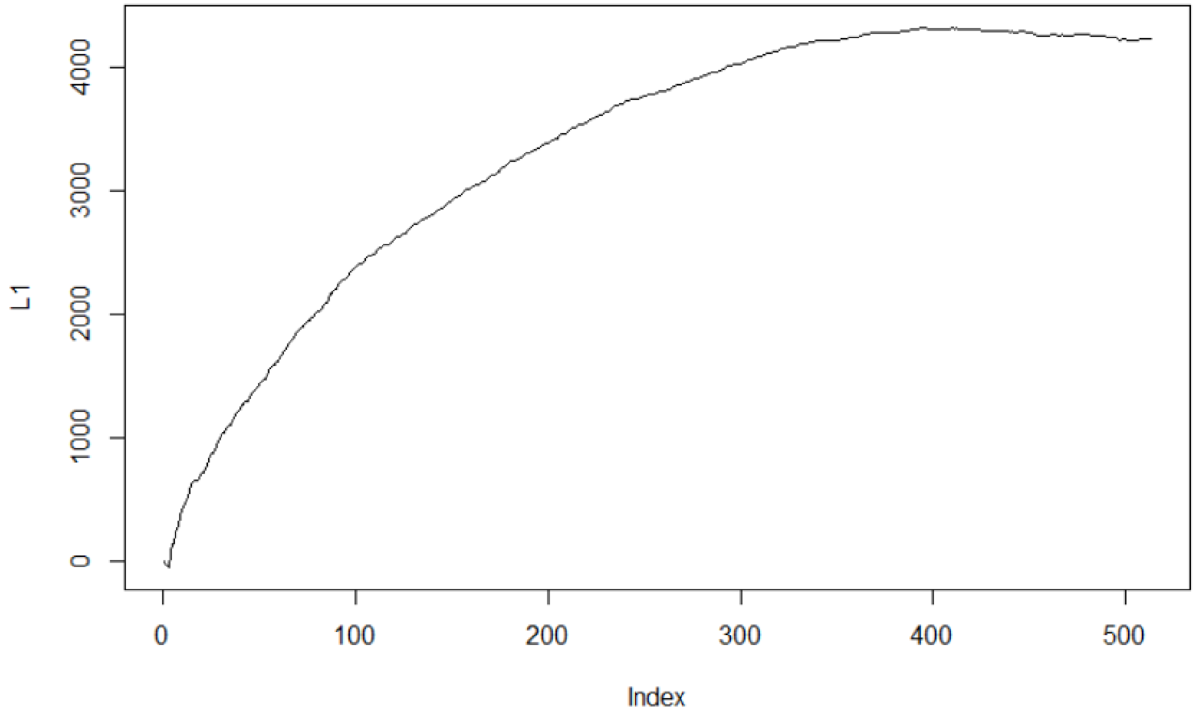
R code: L(d)

```
> L <- Lest(School.ppp)
```



R code: L(d)

```
> L <- Lest(School.ppp)
> L1<-L$iso - L$r
> plot(L1, main=NULL,type="l")
```



R code: Monte Carlo Significance Test

envelope

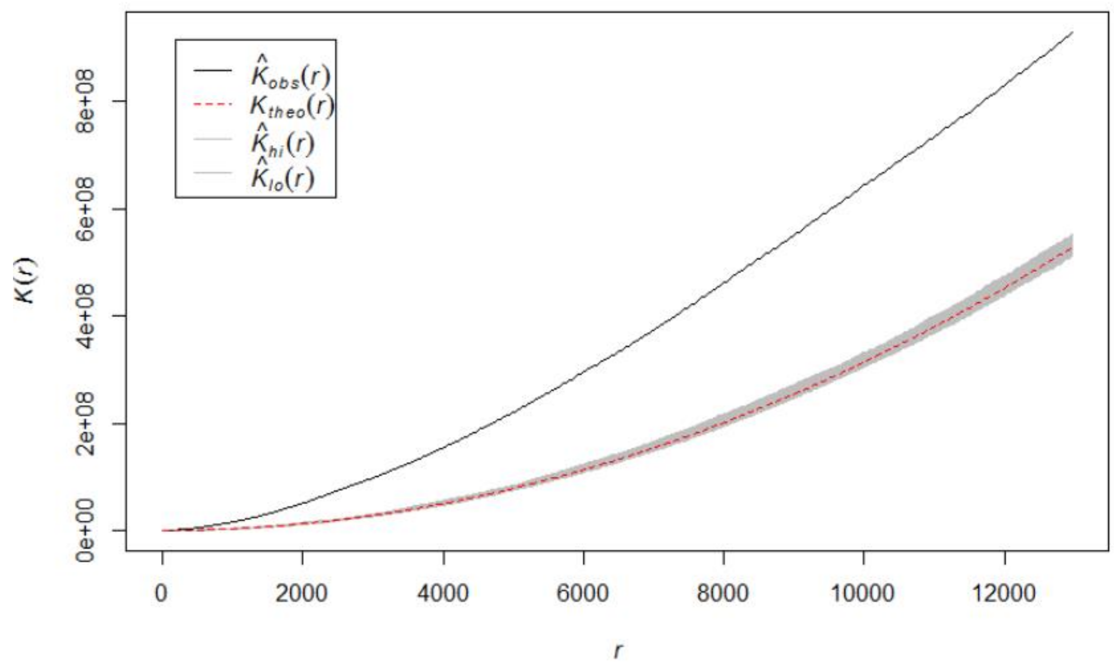
Simulation Envelopes Of Summary Function

Computes simulation envelopes of a summary function.

Keywords hplot, htest, spatial, iteration

Significant level = 1/(99+1) = 0.01

```
> CI<-envelope(School.ppp, fun=Kest, nsim=99, nrank=1)
Generating 99 simulations of CSR ...
```

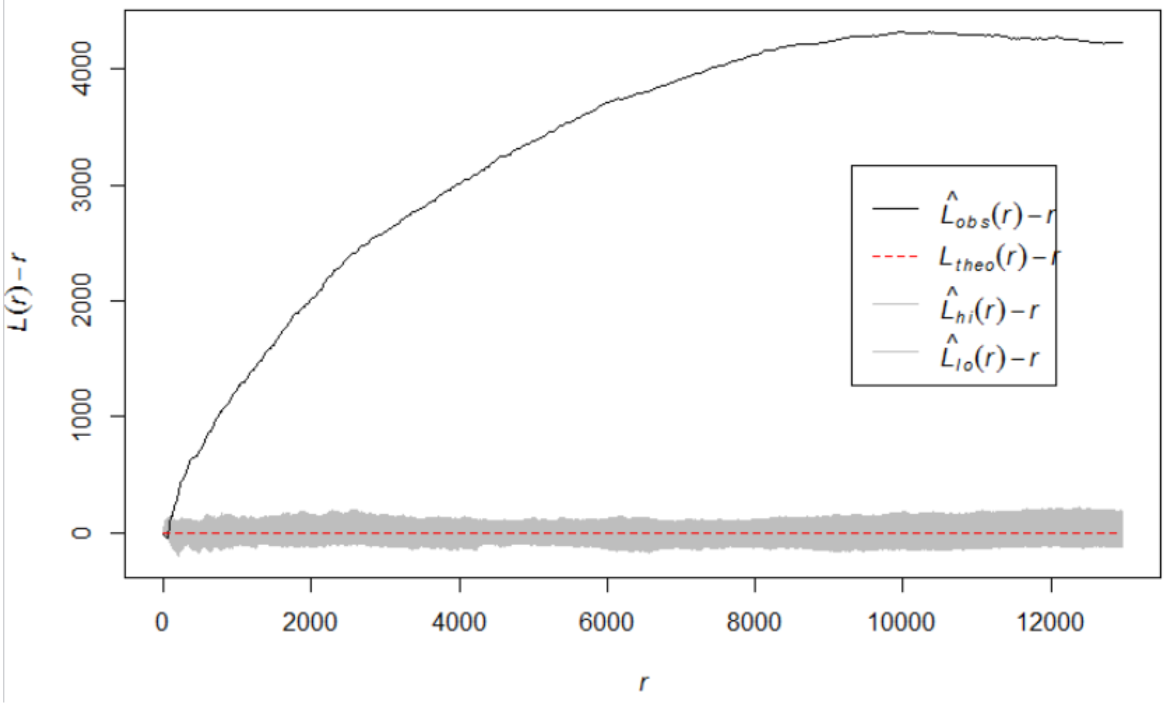


Rank of the envelope value amongst the nsim simulated values. A rank of 1 means that the minimum and maximum simulated values will be used.

R code: Monte Carlo Significance Test

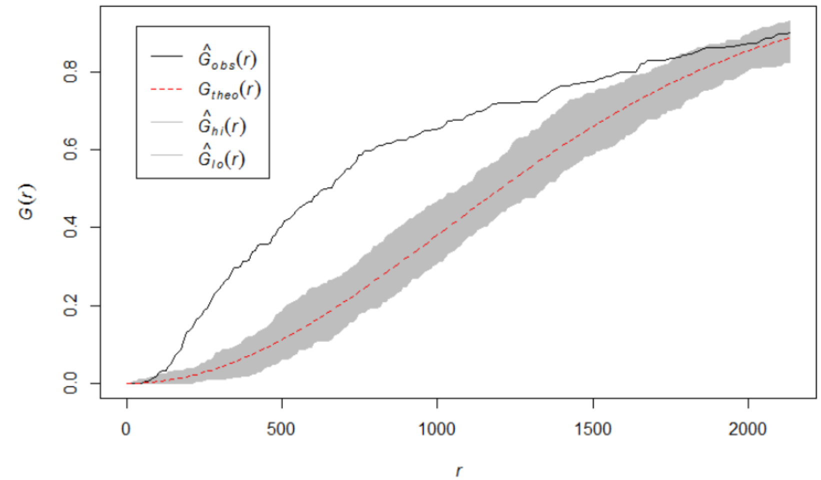
```
> CI_L<-envelope(School.ppp, fun=Lest, nsim=99, nrank=1)
Generating 99 simulations of CSR  ...
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 
 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45
54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 9

Done.
> plot(CI_L)
> plot(CI_L, .-r~r)
```

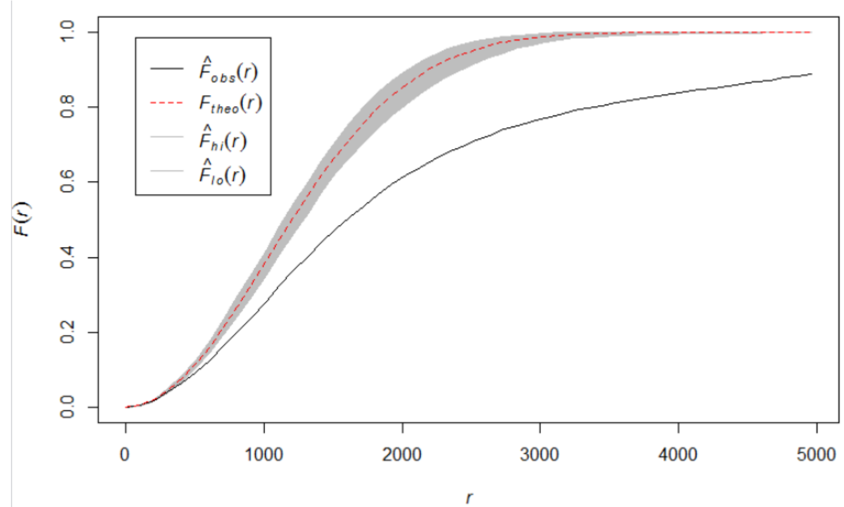


R code: Envelope for G(d) and F(d)

```
CI_G<-envelope(School.ppp, fun=Gest, nsim=99, nrank=1)
```




```
CI_F<-envelope(School.ppp, fun=Fest, nsim=99, nrank=1)
```



整理與複習

COMPARING METHODS FOR MEASURING THE PATTERN OF FEATURE LOCATIONS

Method	Statistic	Significance test	Advantages	Disadvantages
Overlaying areas of equal size	Quadrat analysis	Kolmogorov-Smirnov Chi-square Variance-mean ratio	Can be used when there are multiple features at a single location	Doesn't consider the distance between features; results are influenced by the size of the quadrats
Calculating the average distance between features	Nearest neighbor index	Z-score	Considers the distance between features	Results may be biased if there are many features near edge of study area
Counting the number of features within defined distances	K-function	Uses multiple simulations to create a random distribution envelope	Calculates the concentration of features at a range of scales or distances, simultaneously	Patterns are suspect at larger distances due to edge effects

本週實習

- 實作 Procedures of F(d) Function
  - Step 1: Loading school.shp
  - Step 2: Generating Random Points: rpoint()

    (p1, p2, …, pn)
  - Step 3: Calculate d_{min}(pi, s): st_distance() or nncross()
  - Step 4: Calculate F(d): ecdf()
  - Step 5: Monte Carlo Significance Test: for-loop
  - Step 6: plotting the CDF curve: plot()
  - **Final:** comparing with the result of **envelope** (school.ppp, fun=Fest)

本週作業

- 實作 Procedures of K(d) Function
  - comparing with the result of **envelope** (school.ppp , fun=Lest)

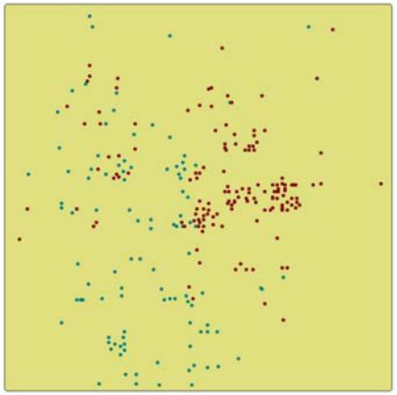
- 研讀期刊論文 **Reading_Geodemographics.pdf**

  Geodemographic analysis and the identification of potential business partnerships enabled by transit smart cards
  - 內容說明：用中文解釋每一個 figure and table的意義
  - 評論與心得 (字數與格式不限)

期末考加分題：
Bivariate K function

Is the distribution of one set of events related to the distribution of the other?

Black (red dots) and white (blue dots) crimes in Oklahoma



Tpe_Fastfood.shp

1. 台北市KFC是否顯著群聚在MIC地點的附近?（實作）＋5%

提示：套件 **spatstat** 的 Kcross 函數

2. 如何進行蒙地卡羅顯著性檢定？（實作＋說明）＋10%