點型態分析：最鄰近分析法
Point Pattern Analysis: Nearest Neighbor Analysis

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

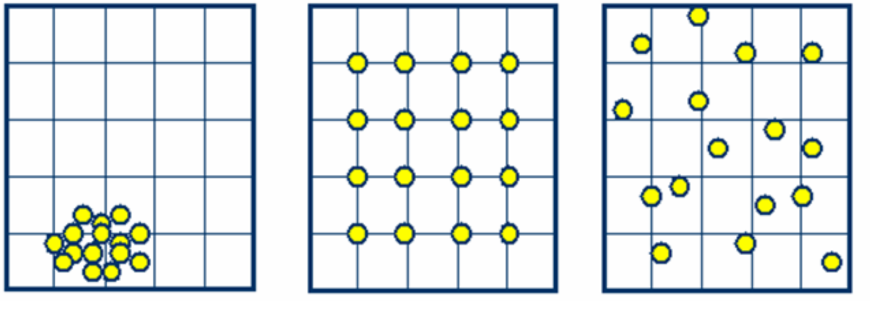
授課教師：溫在弘
E-mail: wenthung@ntu.edu.tw

Outline

- Analyzing global patterns: Overview
- Nearest Neighbor Analysis (NNA)
  - Test statistic: R scale
  - Statistical Significance Test
    - Comparing with **theoretical random pattern**
    - Monte Carlo significance test
  - K-order NNA
- G(d) Function

Point Pattern Analysis

- Analyzing *Global* Patterns (4 weeks)
  - Quadrat Analysis
  - Nearest Neighbor Analysis
  - *Distance*-based Functions
  - *Density*-based Methods



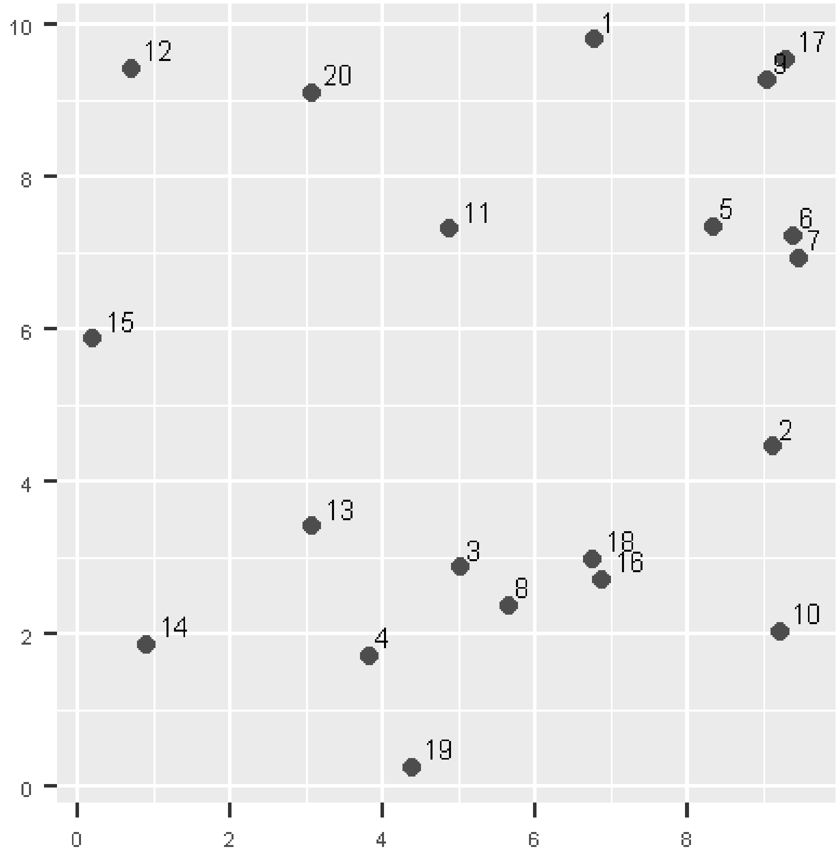
(a) Clustering (b) Dispersion/Uniform (c) Random

Properties of Spatial Point Patterns

- First-order Property (measuring the **intensity**)
  - **First order property** indicates the intensity of a process, mean number of events per unit area at point p.
  - The concept is similar to the **mean** as the first order statistics in statistical theory.
- Second-order Property (measuring the **dependency**)
  - **Second order property** of a spatial point pattern indicates spatial dependency of a process, mean number of paired events per unit area, like between point p and q.

Nearest Neighbor Analysis (NNA)

measures the average distance from each point in the study area to **its nearest point**.



From	To	Distance	From	To	Distance
1	9	2.32	11	20	2.55
2	10	2.43	12	20	2.39
3	8	0.81	13	4	1.85
4	19	1.56	14	13	2.67
5	6	1.05	15	12	3.58
6	7	0.3	16	18	0.29
7	6	0.3	17	9	0.37
8	3	0.81	18	16	0.29
9	17	0.37	19	4	1.56
10	2	2.43	20	12	2.39

Nearest Neighbor Analysis (NNA)

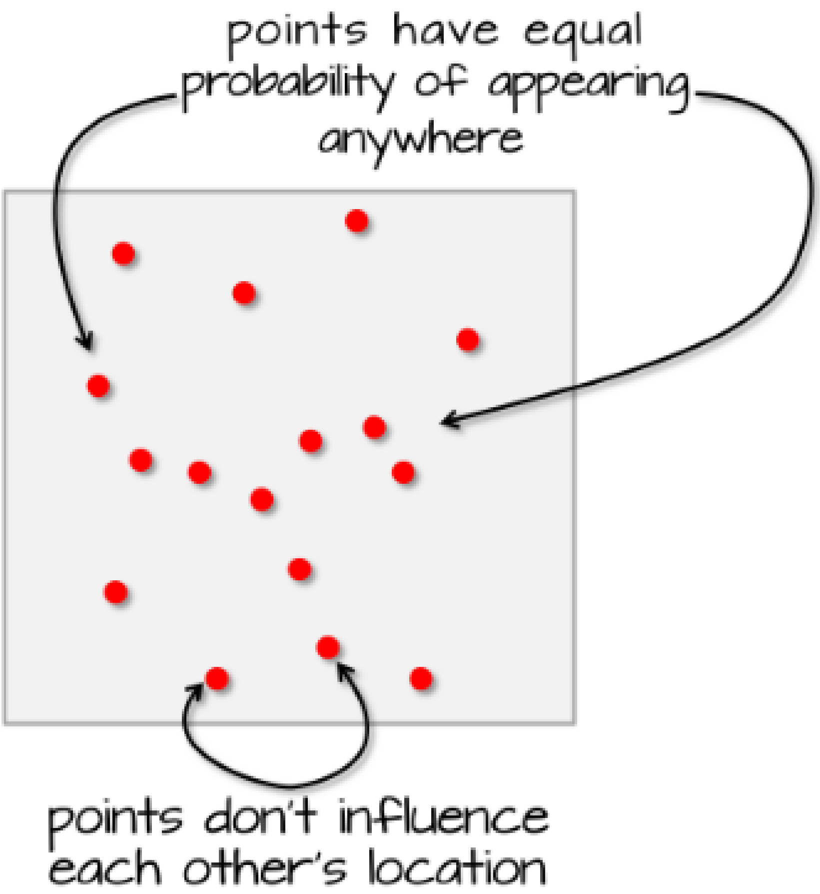
- The NNA compares the average distance between nearest neighbors to that of a random pattern
  - 大於 隨機分布的平均距離 : Dispersed
  - 小於 隨機分布的平均距離 : Clustered

- R Scale（ r_{obs} / r_{exp} ）
  - For theoretical random pattern: $r_{exp} = \frac{0.5}{\sqrt{n / A}}$

Theoretical Random Pattern

- Spatial analysis techniques compare observed point patterns to ones generated by an **independent random process** (IRP) also called **complete spatial randomness** (CSR). CSR/IRP satisfy two conditions:
- Any event has equal probability of being in any location, a **1st order effect**.
- The location of one event is independent of the location of another event, a **2nd order effect**.

Theoretical Random Pattern



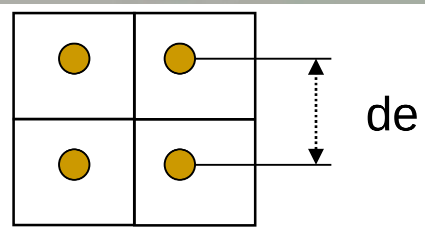
Average nearest distance of theoretical random pattern

Dispersed/ uniform

The expected mean distance for a completely dispersed distribution

$$\bar{d}_e = \frac{1}{\sqrt{n / A}}$$

Divide the number of features by the area of the study area, and take the square root...then divide the result into 1



Random

The expected mean distance for a random distribution

$$\bar{d}_e = \frac{0.5}{\sqrt{n / A}}$$

Divide 0.5 by the square root of the number of features divided by the area

Point	Nearest Neighbor	Distance
1	2	1
2	3	0.1
3	2	0.1
4	5	1
5	4	1
6	5	2
7	6	2.7
8	10	1
9	10	1
10	9	1
		10.9

r	1.09
Area of Region	50
Density	0.2
Expected Mean	1.118034
R	0.974926

Point	Nearest Neighbor	Distance
1	3	2.2
2	4	2.2
3	4	2.2
4	5	2.2
5	7	2.2
6	7	2.2
7	8	2.2
8	9	2.2
9	10	2.2
10	9	2.2
		22

r	2.2
Area of Region	50
Density	0.2
Expected Mean	1.118034
R	1.96774

Point	Nearest Neighbor	Distance
1	2	0.1
2	3	0.1
3	2	0.1
4	5	0.1
5	4	0.1
6	5	0.1
7	6	0.1
8	9	0.1
9	10	0.1
10	9	0.1
		1

r	0.1
Area of Region	50
Density	0.2
Expected Mean	1.118034
R	0.089443

$$\bar{r} = \frac{\sum r}{n}$$

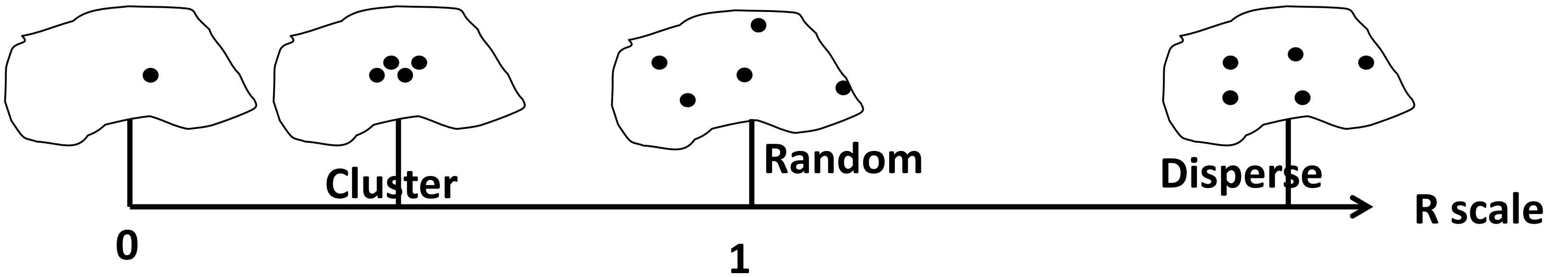
$$d = \frac{n}{area}$$

$$\bar{r}(e) = \frac{.5}{\sqrt{d}}$$

$$R = \frac{r}{\bar{r}(e)}$$

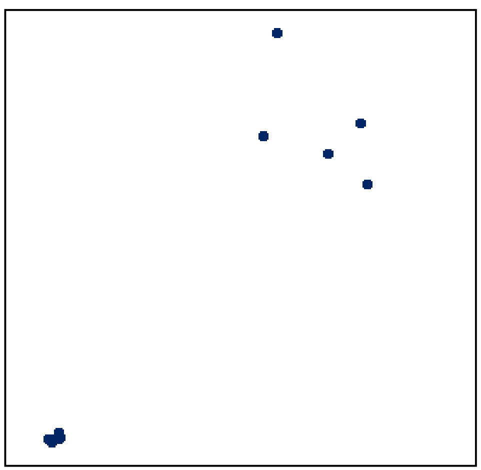
R Scale

- R = 0 (completely clustered)
- R = 1 (random)
- R = 2.149 (completely dispersed)



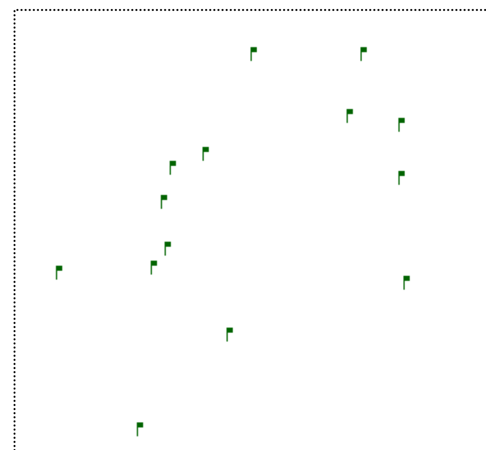
Average Nearest Neighbor Distance

	D.obs	D.exp	R	Result
情境 1	3.157	7.958	0.397	Cluster
情境 2	40	13.5	2.424	Disperse
情境 3	14.486	11.207	1.293	Random



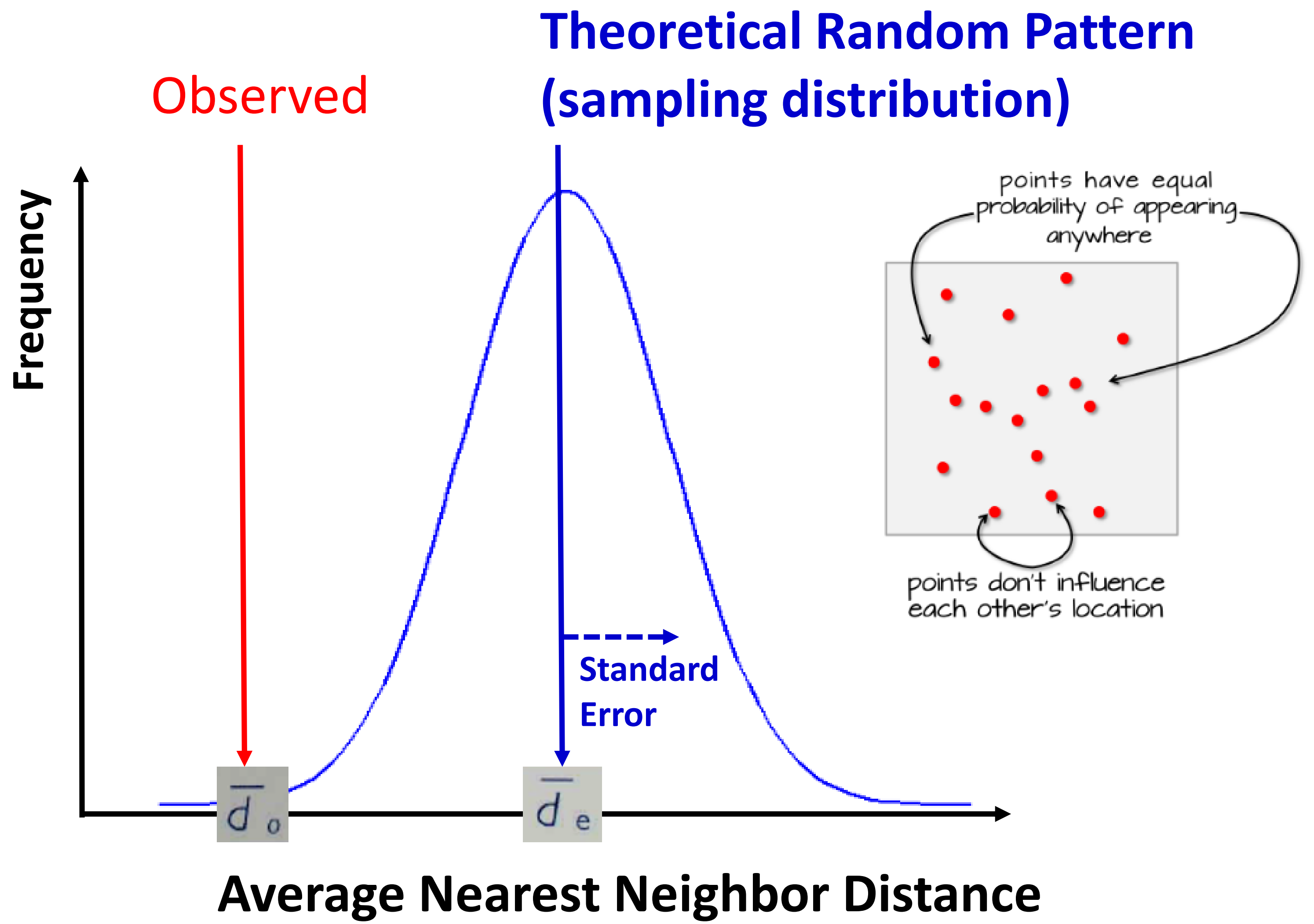
情境1

情境2



情境3

Significance Test of NNA:
1. Comparing with Theoretical Random Pattern



Significance Test of NNA

The expected mean distance is subtracted from the observed mean distance....

The Z-score

$$Z = \frac{\bar{d}_o - \bar{d}_e}{SE}$$

....and the difference divided by the standard error

The expected mean distance for a random distribution

$$\bar{d}_e = \frac{0.5}{\sqrt{n / A}}$$

The standard error

0.26136 (a mathematical constant) is divided by....

$$SE = \frac{0.26136}{\sqrt{n^2 / A}}$$

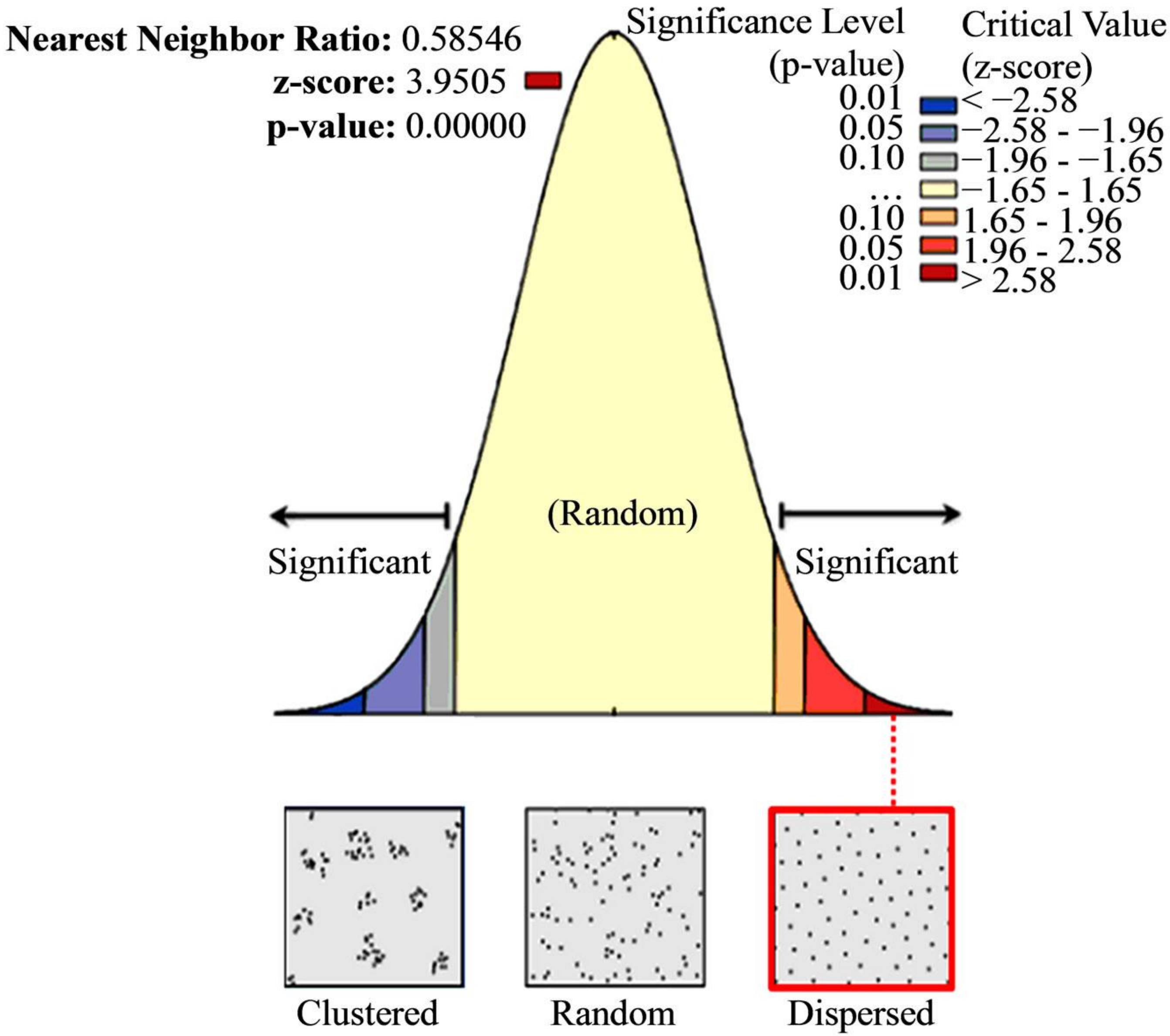
....the square root of the number of features squared (n^2) divided by the areal extent (A) of the study area

Standard Error of R-scale

- To describe the likelihood that any differences occur purely by chance
  - The calculated difference is relatively small compared to the std. error then the difference is statistically insignificant
  - The Std. Error for the obs. distance
    - $SE_r = 0.26136 / \sqrt{(n^2/A)}$
    - $Z_R = r_{obs} - r_{exp} / SE_r$
    - The difference will be statistically significant if

      $-1.96 < Z_R < 1.96$ (for $\alpha=0.05$) two tailed

      (For single tailed 1.96 will changed as 1.645)

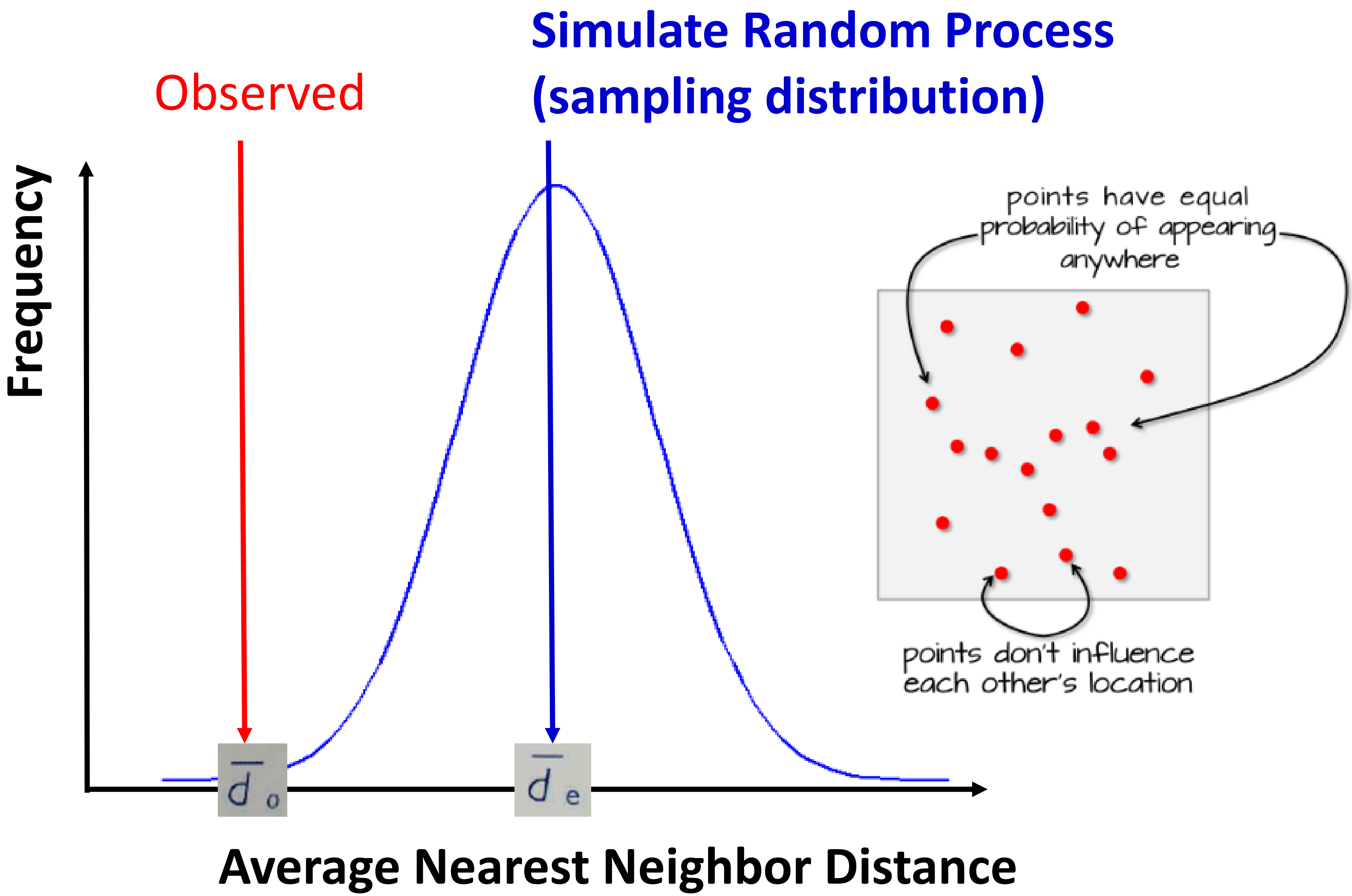


Nearest Neighbor Ratio: 0.58546
z-score: 3.9505
p-value: 0.00000
Significance Level (p-value)
Critical Value (z-score)
0.01 < −2.58
0.05 −2.58 - −1.96
0.10 −1.96 - −1.65
... −1.65 - 1.65
0.10 1.65 - 1.96
0.05 1.96 - 2.58
0.01 > 2.58
(Random)
Significant
Significant
Clustered
Random
Dispersed

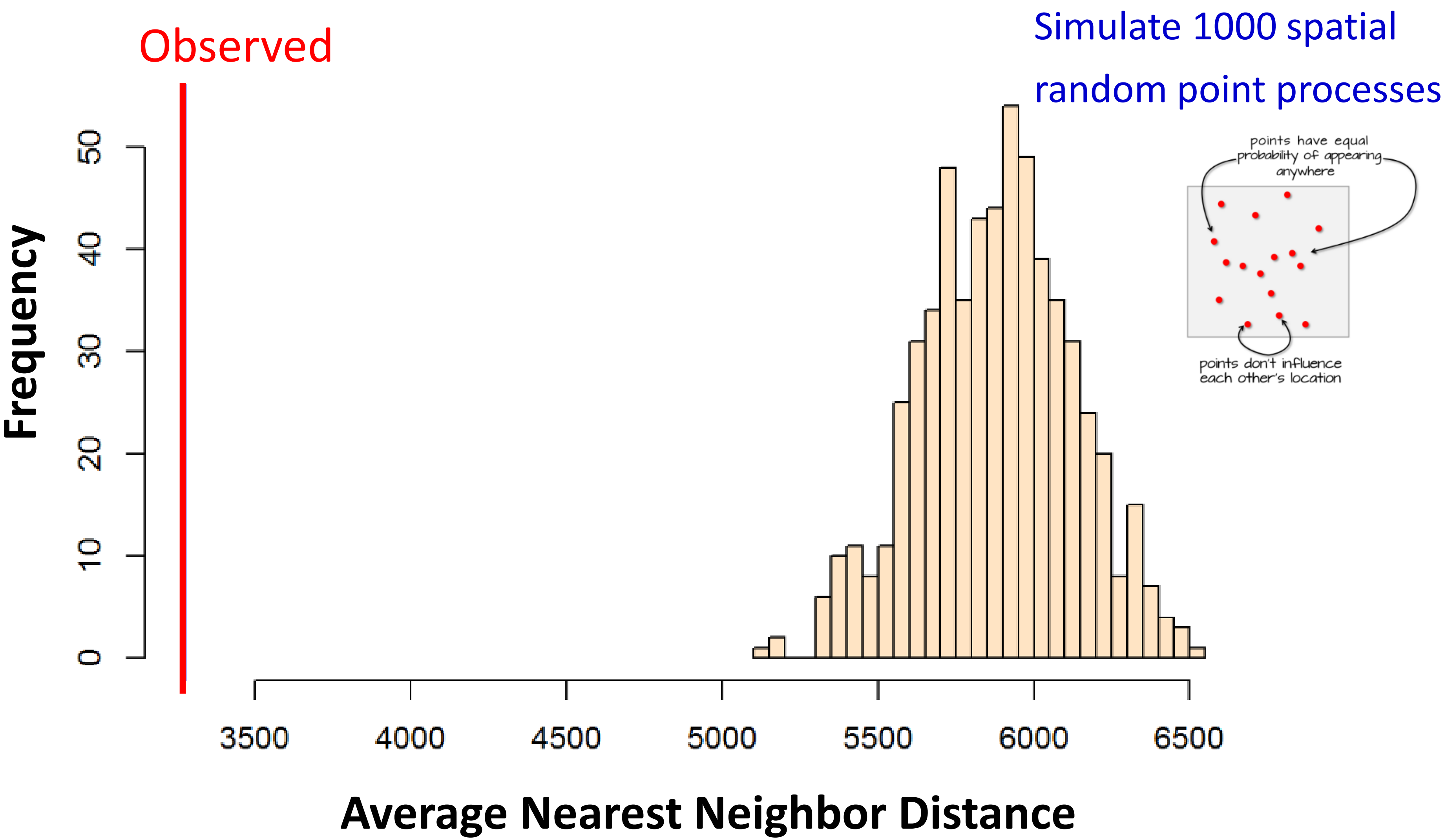
Significance Test of NNA: 2. Monte Carlo Significance Test

- The significance of any departures from CSR can be evaluated using simulated “confidence envelopes”
- Simulate many (eg. 100) spatial point processes
- Rank all the simulations
- Pull out the 5^{th} and 95^{th} values
- Plot these as the 95% confidence intervals

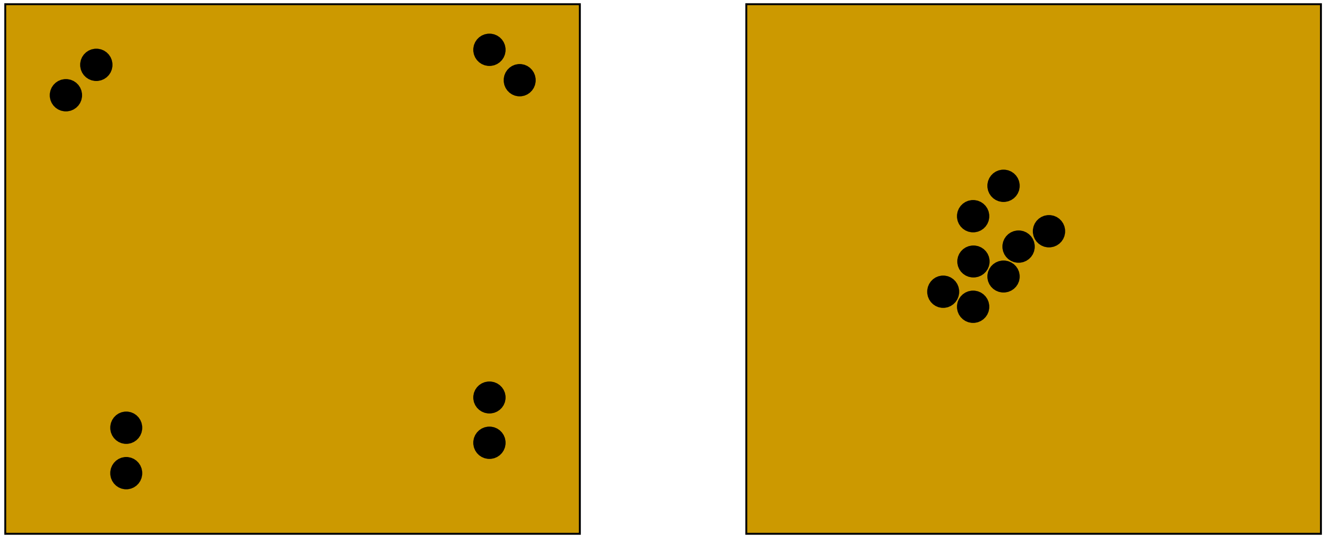
Monte Carlo Significance Test



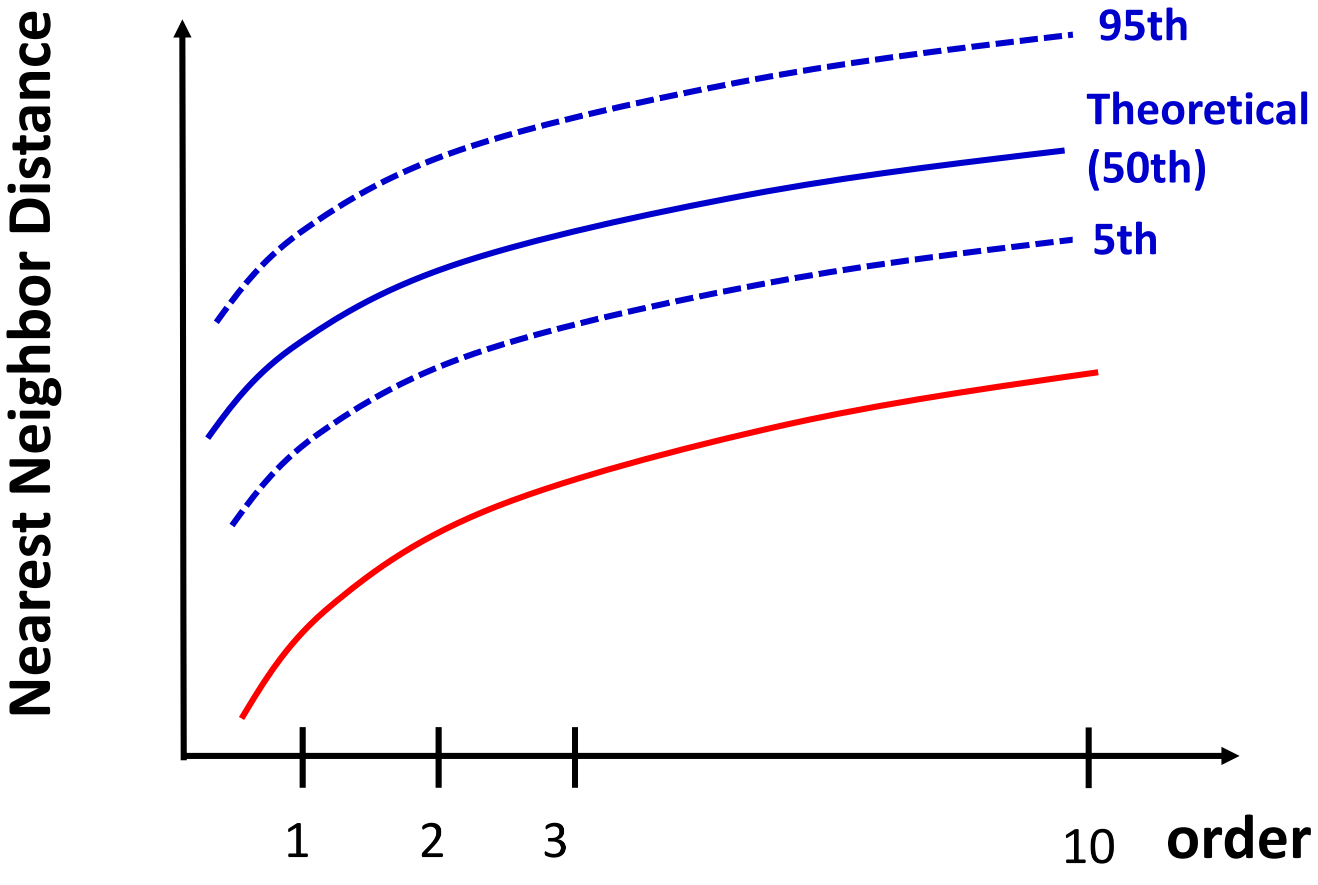
Monte Carlo Significance Test



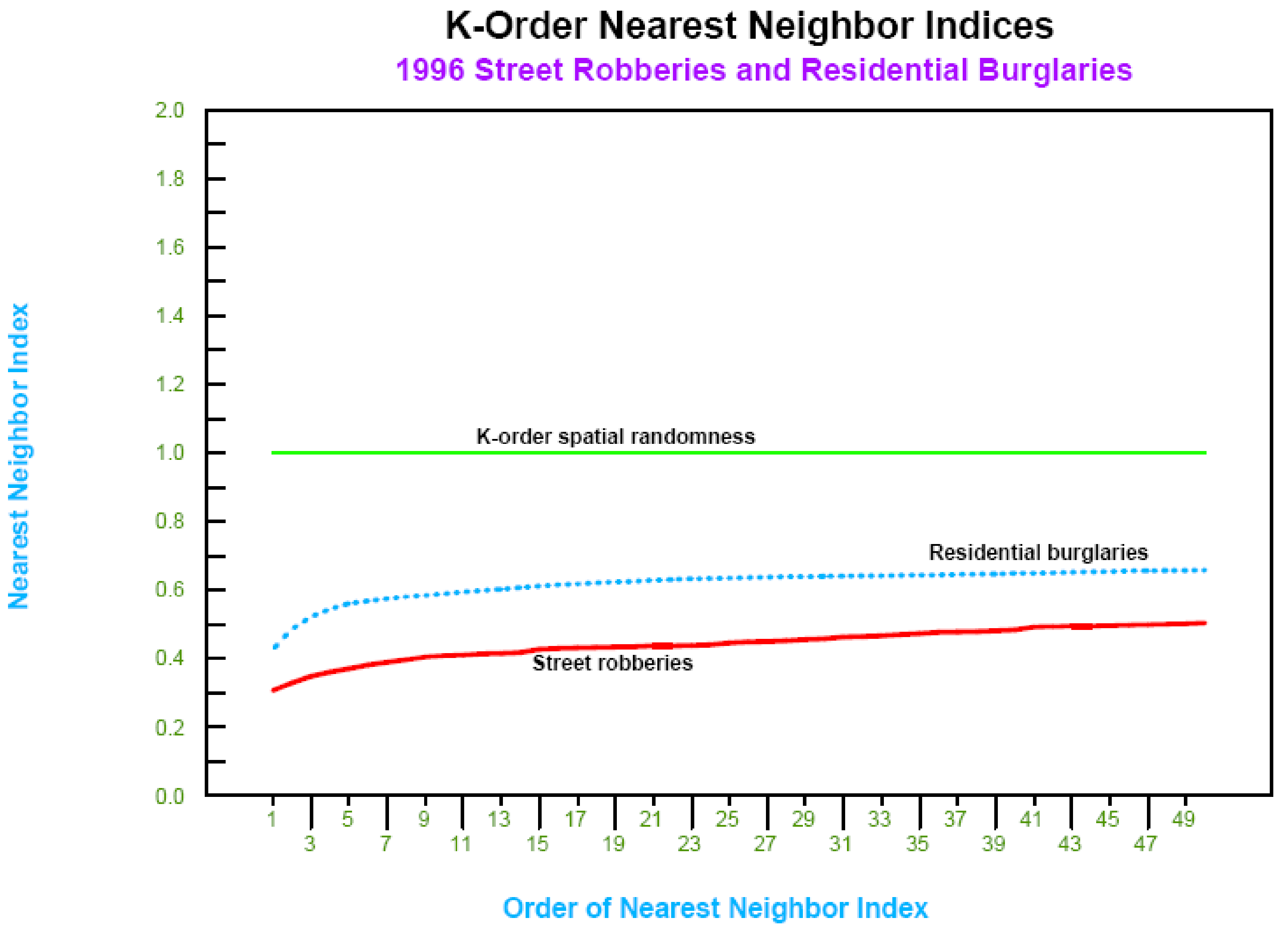
Why needs Higher Order NNA?



K-order Nearest Neighbor Distance



K-order Nearest Neighbor Indices: Example



研讀教科書教材 TEXT_Nearest.Neighbor.Analysis.pdf

(內容列入期末考範圍)

CALCULATING THE AVERAGE DISTANCE BETWEEN FEATURES

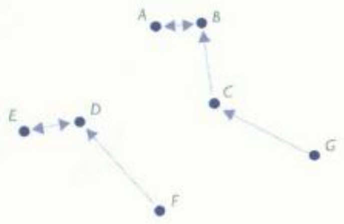
The nearest neighbor index is based on the research of ecologists Philip Clark and Frances Evans, who developed the method in the 1950s to quantify patterns in distributions of various plant species. In this method, the GIS finds the distance between each feature and its closest neighbor, then calculates the average (or mean) of these distances.

What the nearest neighbor index measures

The nearest neighbor index measures how similar the mean distance is to the expected mean distance for a hypothetical random distribution. The index is either the difference between the two or the ratio of the observed distance divided by the expected distance.

Calculating the observed mean distance

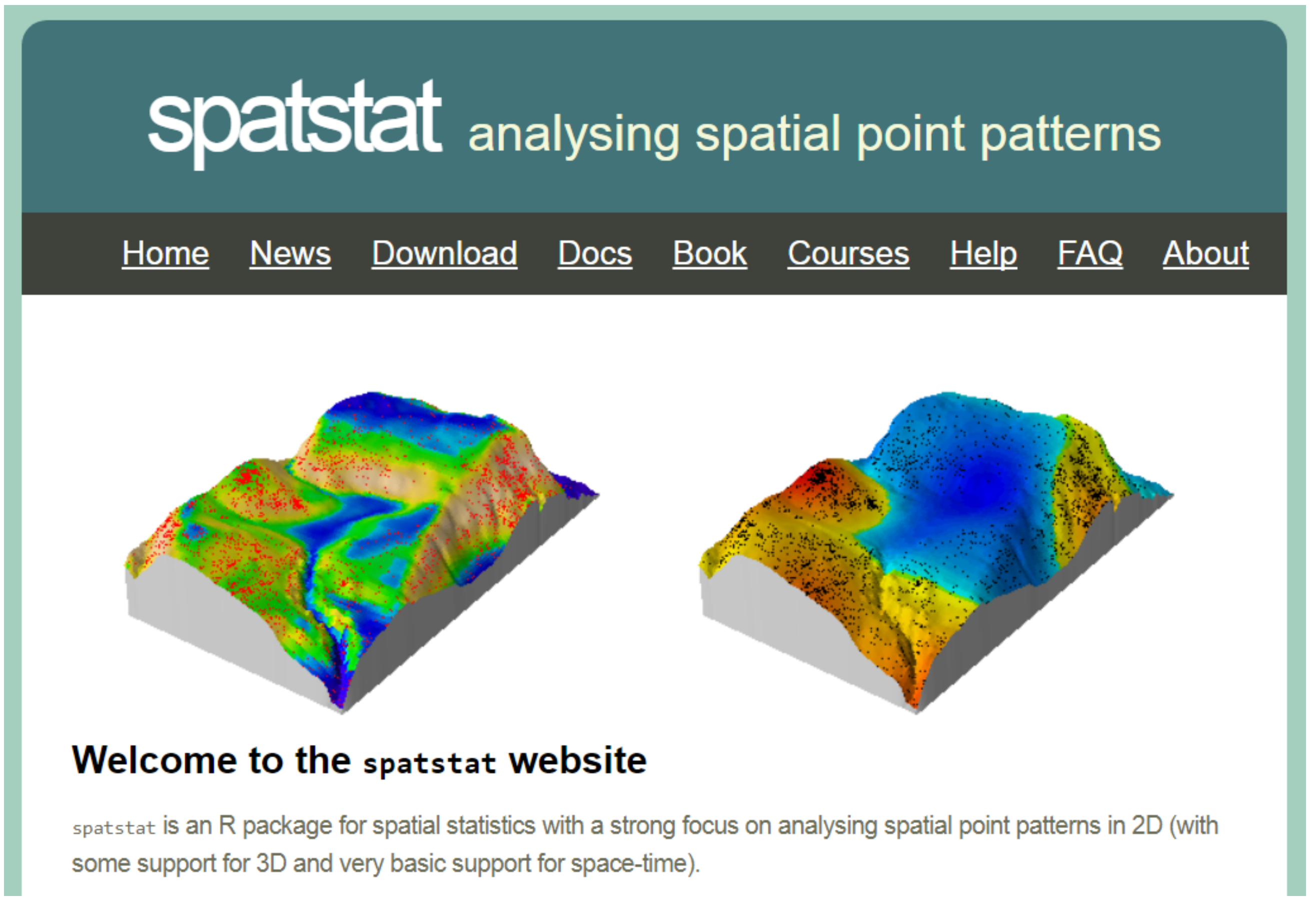
To get the distance from each feature to its nearest neighbor, the GIS essentially calculates the distance from each feature to all other features in the set, then finds the shortest distance, the nearest neighbor to the feature.



	A	B	C	D	E	F	G
A		988	2117	2494	3538	3858	4267
B	988		1725	3348	4308	4004	3601
C	2117	1725		2804	4034	2567	2309
D	2494	3348	2804		1196	2510	4897
E	3538	4308	4034	1196		3277	6034
F	3858	4004	2567	2510	3277		3433
G	4267	3601	2309	4897	6034	3433	

R Lab: Introducing spatstat package

http://spatstat.org/

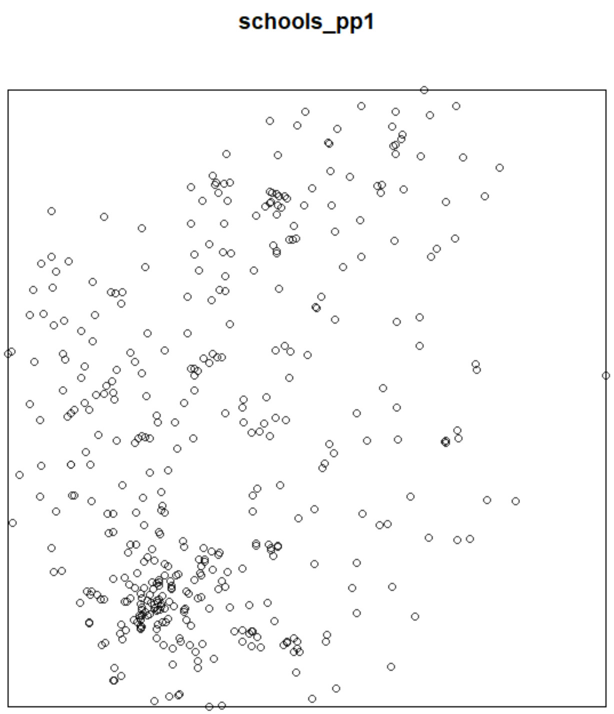


Spatstat: Introducing PPP format

schools_sf <- st_read("Schools.shp")

bnd <- st_bbox(schools_sf)

```
> bnd<-st_bbox(schools_sf)
> bnd
     xmin      ymin      xmax      ymax
 155883.0 2535016.0  207754.2 2588604.0
```



x.coor <- schools_sf$X_coor

y.coor <- schools_sf$Y_coor

x.range <- c(bnd[1],bnd[3])

y.range <- c(bnd[2],bnd[4])

schools_pp1 <- ppp(x.coor, y.coor, x.range, y.range)

Spatstat: Introducing PPP format

schools_sf <- st_read("Schools.shp")

bnd <- st_bbox(schools_sf)

```
> bnd<-st_bbox(schools_sf)
> bnd
     xmin      ymin      xmax      ymax 
 155883.0 2535016.0  207754.2 2588604.0 
```

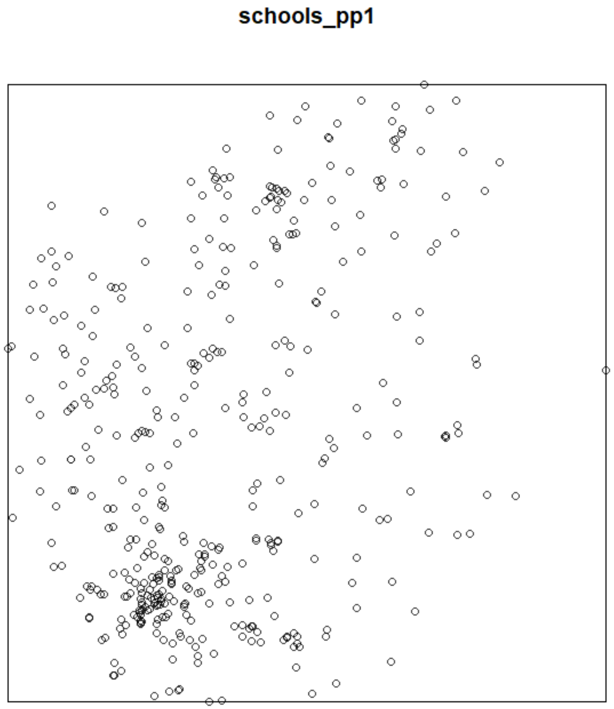
x.coor <- schools_sf$X_coor

y.coor <- schools_sf$Y_coor

x.range <- c(bnd[1],bnd[3])

y.range <- c(bnd[2],bnd[4])

schools_pp1 <- ppp(x.coor, y.coor, x.range, y.range)



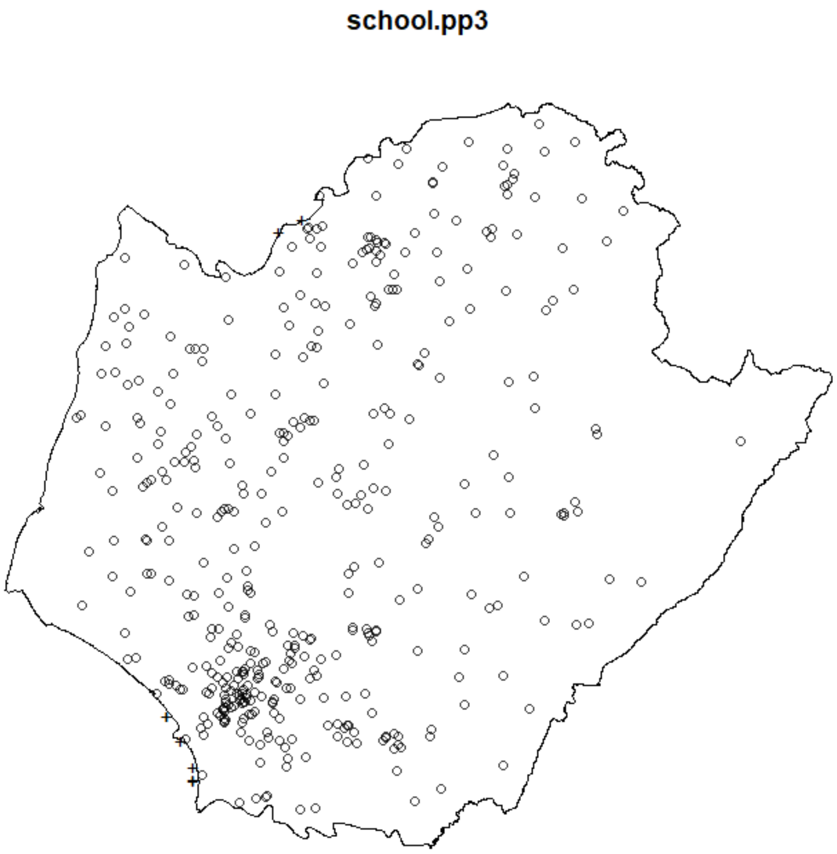
Spatstat: Introducing PPP format (2)

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

index<- county_sf$COUNTY_ID == "67000" #"台南市"
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)
```

PPP format: setting the boundary coordinates of a polygon



R Lab: Average nearest neighbor distance -- using nndist()

```
# calculating the area
x<-x.range[2]-x.range[1]
y<-y.range[2]-y.range[1]
sqr.area<- x*y

nnd<-nndist(schools_pp1, k=1)
d1<-mean(nnd) # Tainan School
rd<- 0.5/sqrt(424/sqr.area)  # theoretical random pattern
r.scale <- d1/rd
```

R code: K-order Nearest Neighbor Distance

nndist(schools_pp1, k=1:20)

```
> nndist(schools_pp1, k=1:20)
          dist.1    dist.2    dist.3    dist.4    dist.5
 [1,] 3293.07708 3906.5673 4060.5085 4248.9399 5542.0873
 [2,]  668.73292 2118.0747 2905.5489 3127.5107 3195.0453
 [3,]  327.91614  362.3026  914.1177 1146.8717 1333.7450
 [4,]  139.67384  306.3700  982.9686  984.7296 1329.9439
 [5,]  139.67384  176.0771  865.0065  961.6888 1466.0069
 [6,]  176.07710  306.3700  690.3840 1042.5882 1317.7148
 [7,]  666.12329  690.3840  865.0065  982.9686 1518.2910
 [8,] 1562.80945 1762.7531 1795.1202 2195.3224 2959.6642
 [9,]  756.48794 1028.4560 1287.6734 1462.6867 1504.4356
[10,]  673.42023  849.8465  997.4824 1028.4560 1322.5454
[11,]  211.31029 1504.4356 1544.2539 1953.0921 2064.9741
[12,]  240.39489 1795.1202 1922.6266 2634.6808 2710.5818
[13,]  240.39489 1562.8095 2095.2726 2657.0145 2735.3344
[14,]  246.63068  464.6477  978.4555 1100.1906 1333.7450
[15,]  246.63068  678.1678 1220.2986 1324.5051 1572.0850
[16,]   54.12947 1925.9305 2217.4034 2273.8727 2284.3634
[17,]   54.12947 1978.5555 2166.0443 2225.3922 2327.4399
[18,] 1146.87169 1183.3835 1400.9277 1660.7438 1854.3505
```

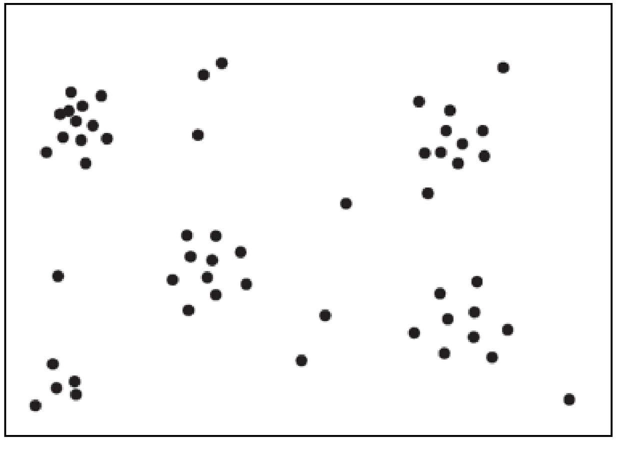
School id

k-order

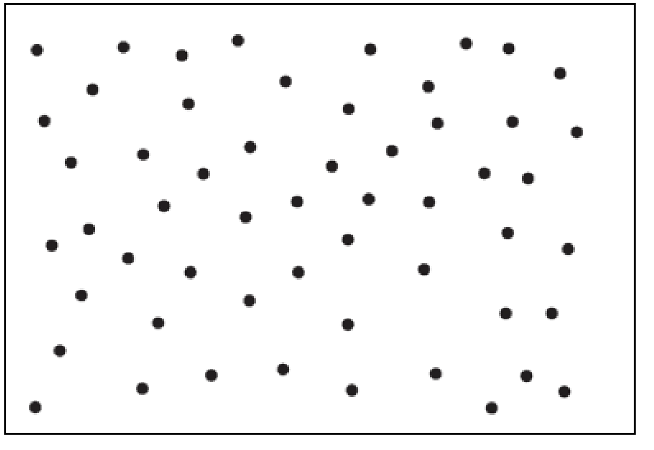
The Second-order Property: Distance-based Methods

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

(PP1)



(PP2)



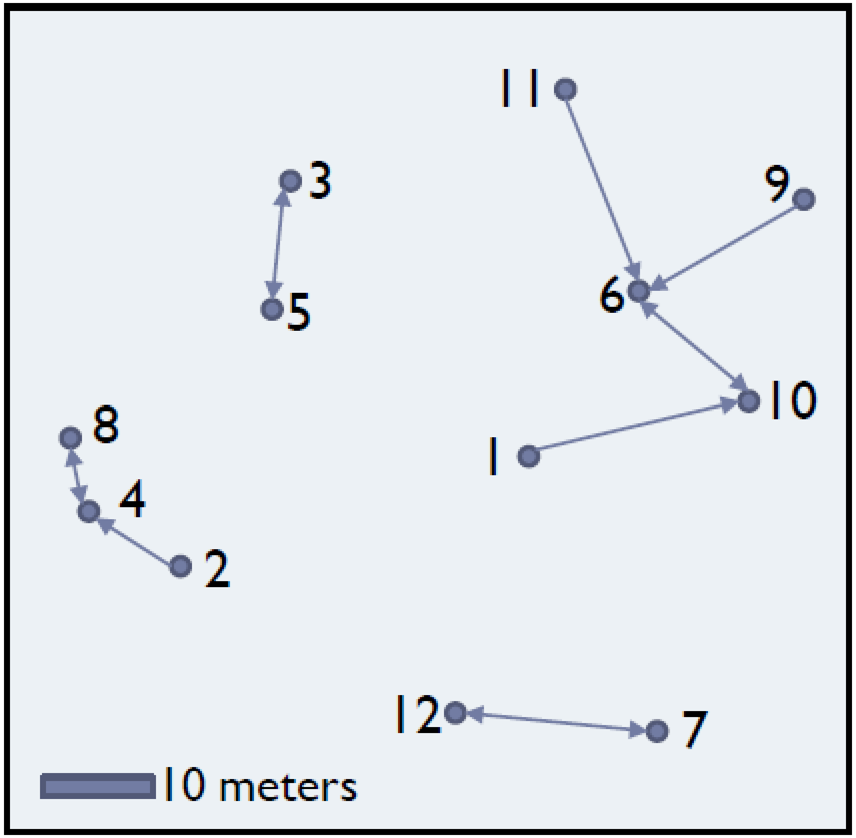
G Function: G(d)

- **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*.

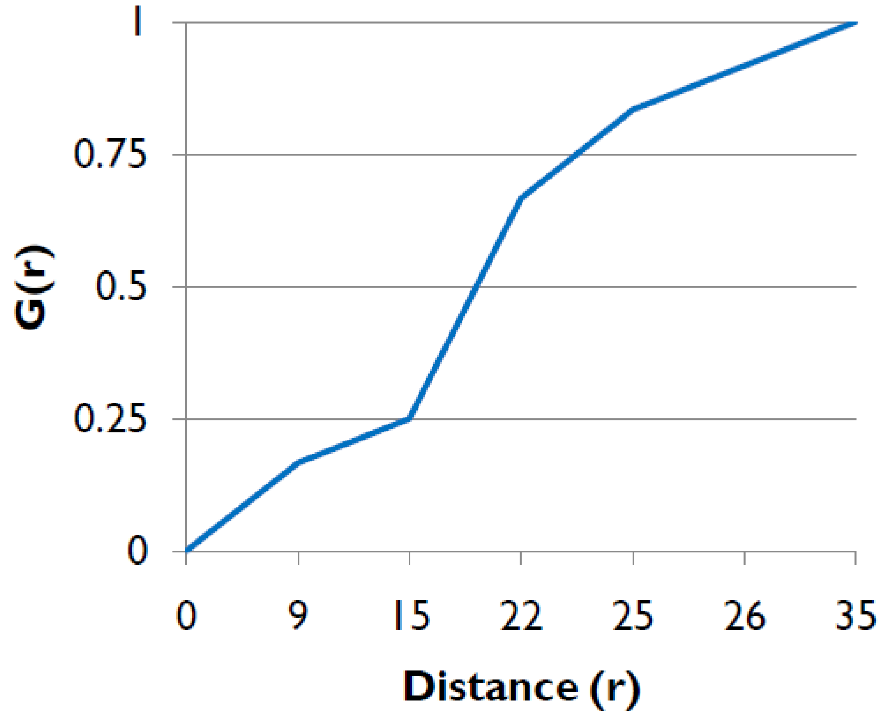
計算範例



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)

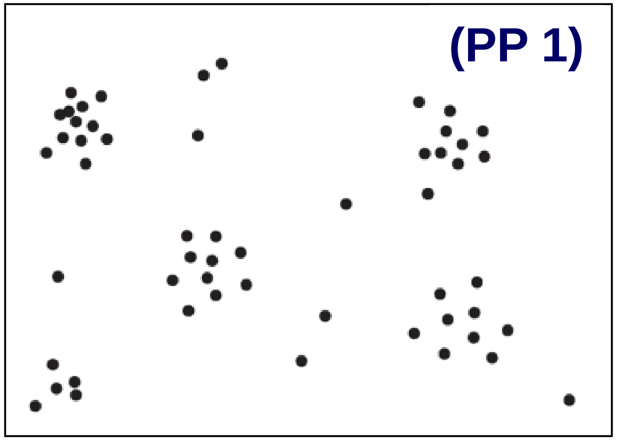
Event	x	y	Nearest neighbor	r_{min}
1	66.22	32.54	10	25.59
2	22.52	22.39	4	15.64
3	31.01	81.21	5	21.14
4	9.47	31.02	8	24.81
5	30.78	60.10	3	9.00
6	75.21	58.93	10	21.14
7	79.26	7.68	12	21.94
8	8.23	39.93	4	9.00
9	98.73	42.53	6	21.94
10	89.78	42.53	6	21.94
11	65.19	92.08	6	34.63
12	54.46	8.48	7	24.81

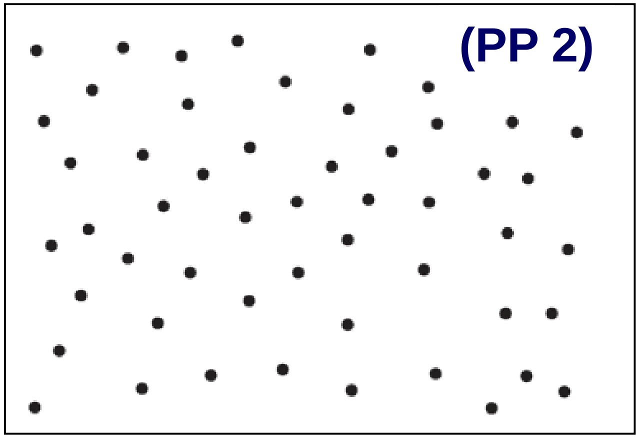


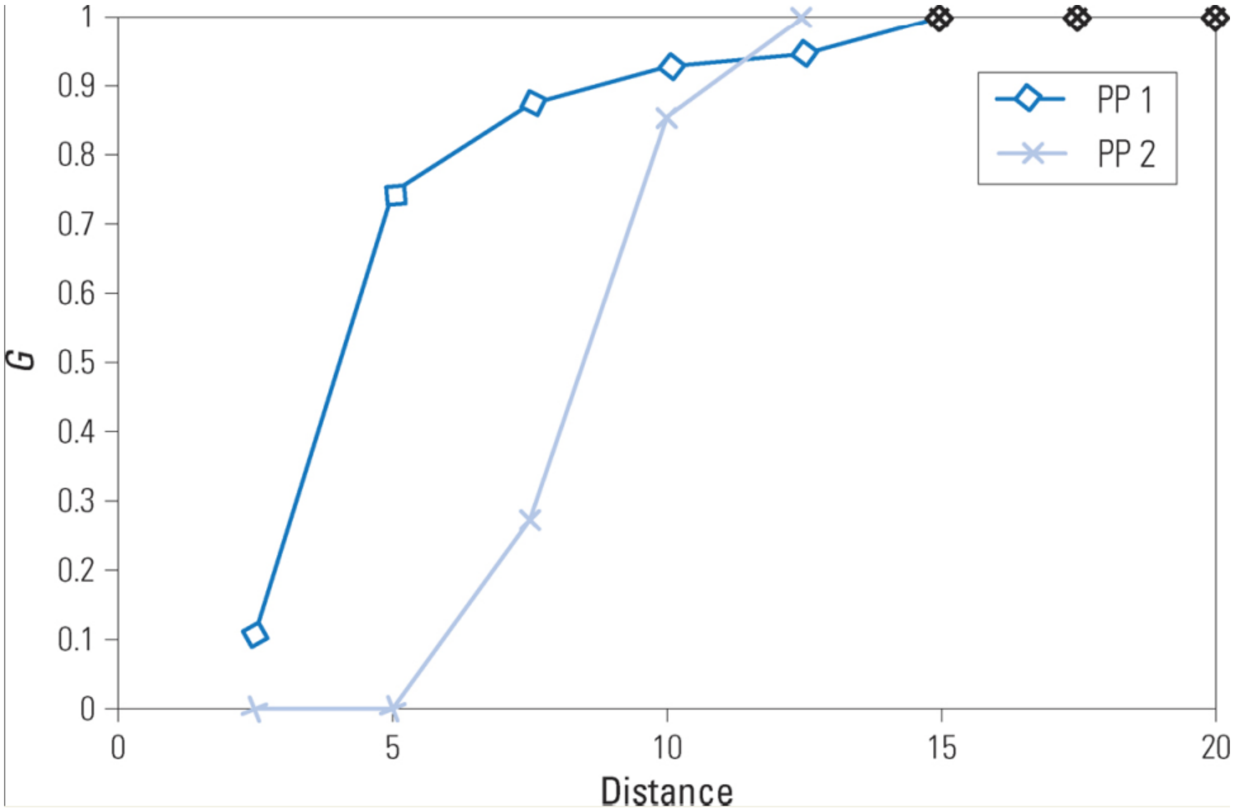
G Function的解讀

- 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





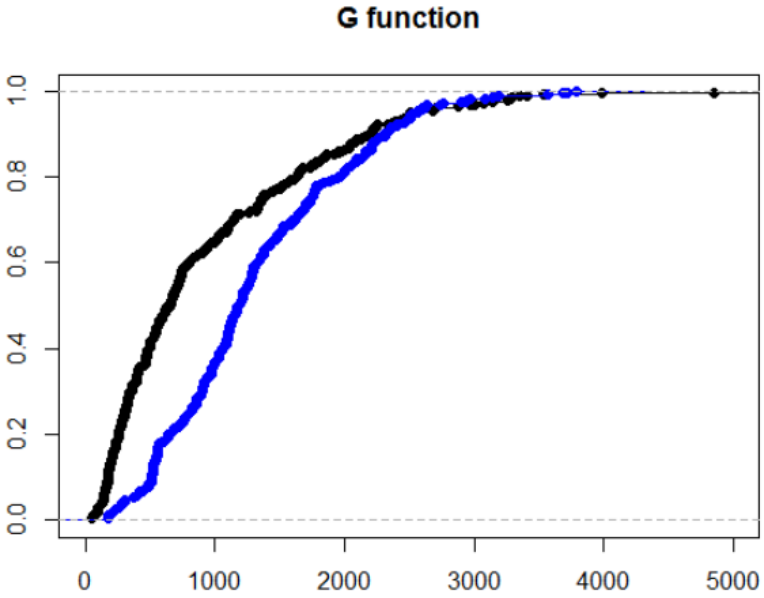


R Lab: G(d) Function

```
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')
```



How do we examine significance？

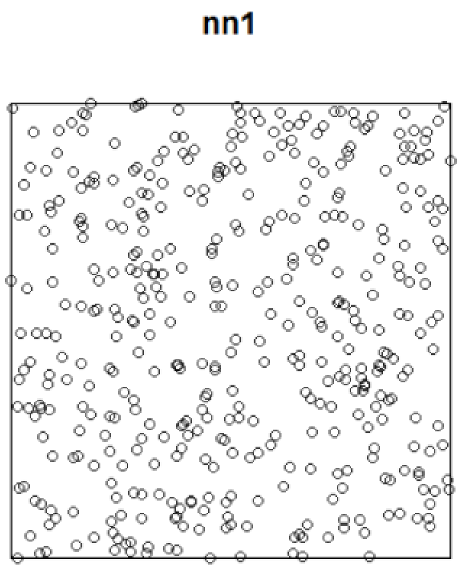
Significant departure from Complete Spatial Randomness (CSR)

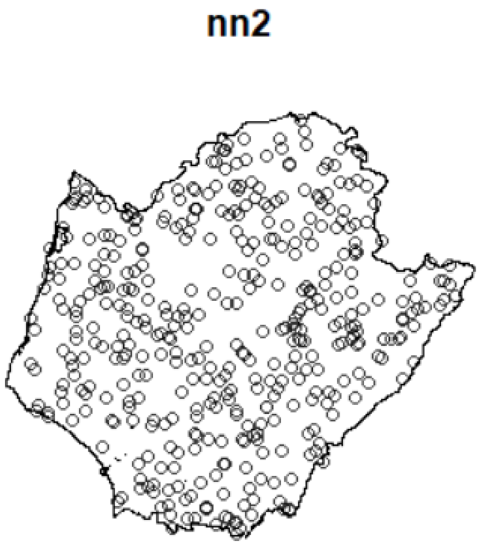
- The significance of any departures from CSR (either clustering or uniform) can be evaluated using simulated “confidence envelopes”
- Simulate many (eg. 1000) spatial point processes and estimate the G function for each of these
- Rank all the simulations
- Pull out the 5^{th} and 95^{th} G(r) values
- Plot these as the 90% confidence intervals

R Lab: Generating Random Points: rpoint()

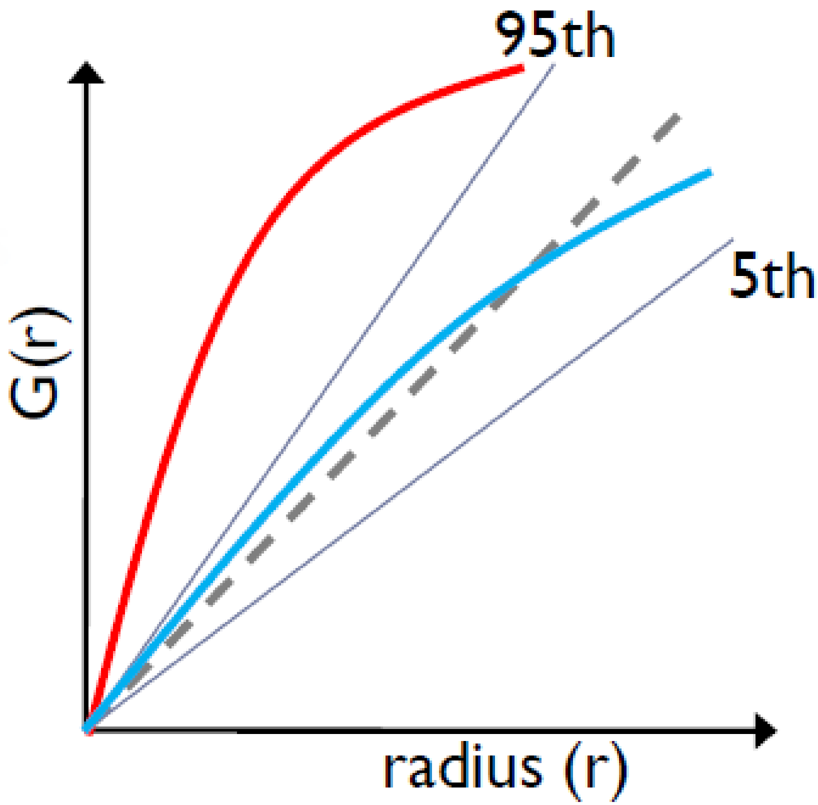
```
TN.Windows<-owin(xrange=x.range, yrange=y.range)
nn1<-rpoint(424, win=TN.Windows)
```

```
nn2<- rpoint(415, win = PTS_bnd)
plot(nn2, pch=16)
```





顯著性檢定的圖示



單尾檢定 (one-tailed test)

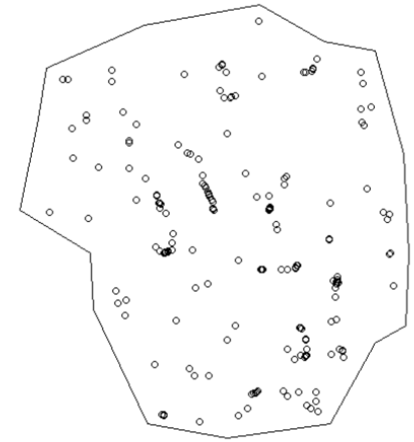
H0: G(r) <= Gr(r) of random 95th percentile

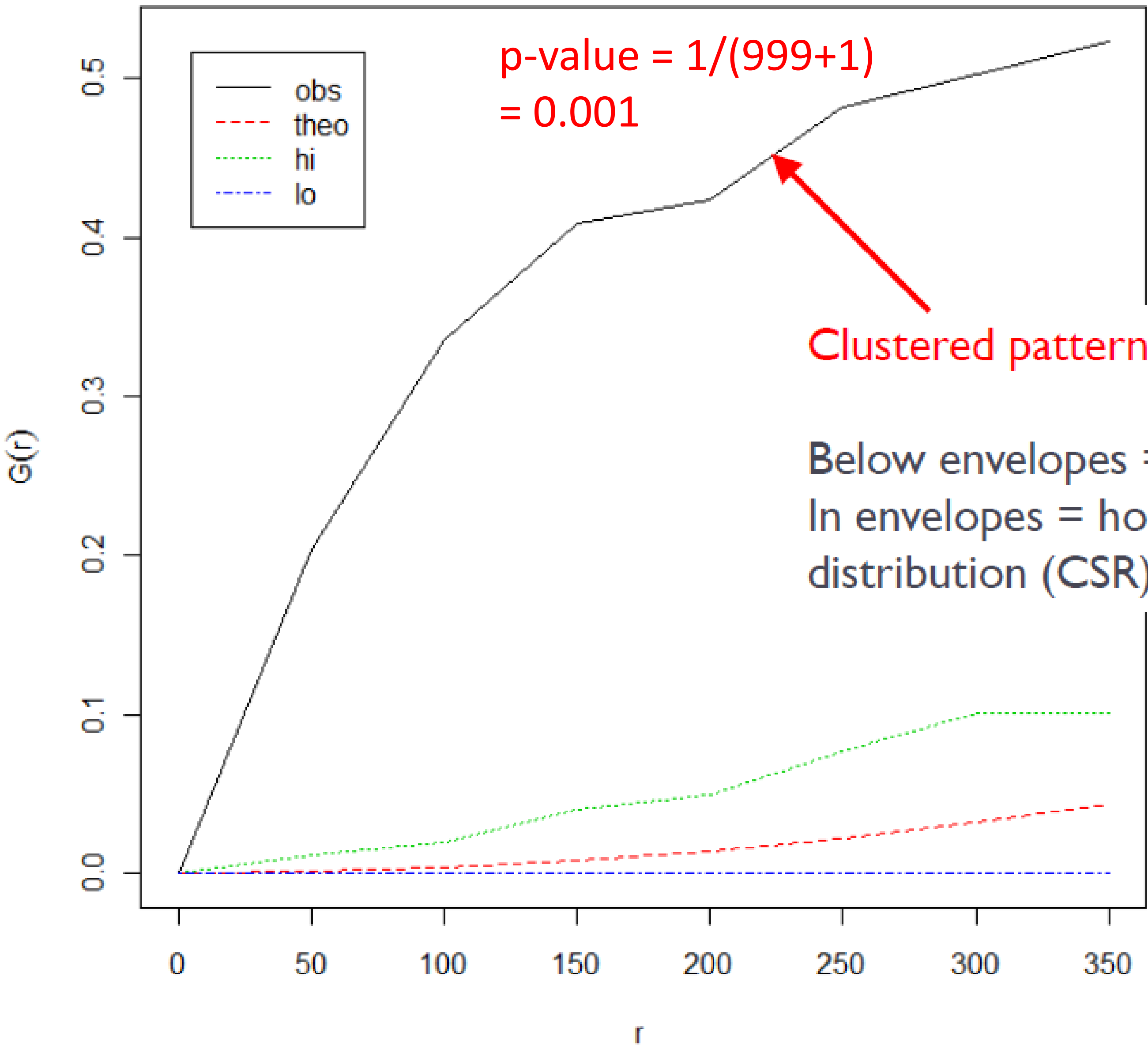
H1: G(r) > Gr(r) of random 95th percentile

If reject H0, it can conclude that

It shows a **statistically significant clustering** pattern (p-value < 0.05)

範例





p-value = 1/(999+1)
= 0.001

Clustered pattern (above the envelopes)

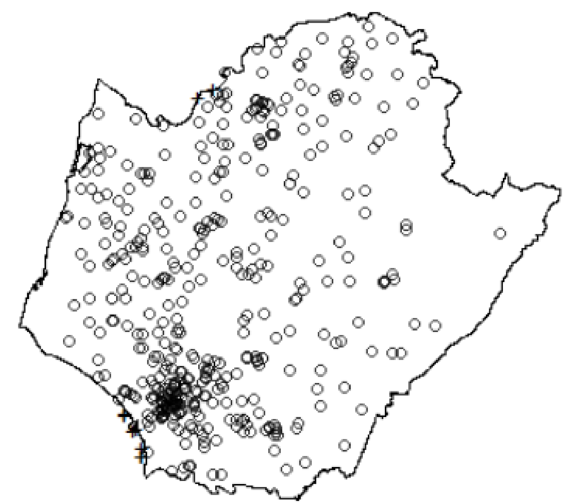
Below envelopes = regular pattern
In envelopes = homogeneous distribution (CSR)

999 simulation runs of random points

本週實習: Analysis of Nearest-Neighbor Distances

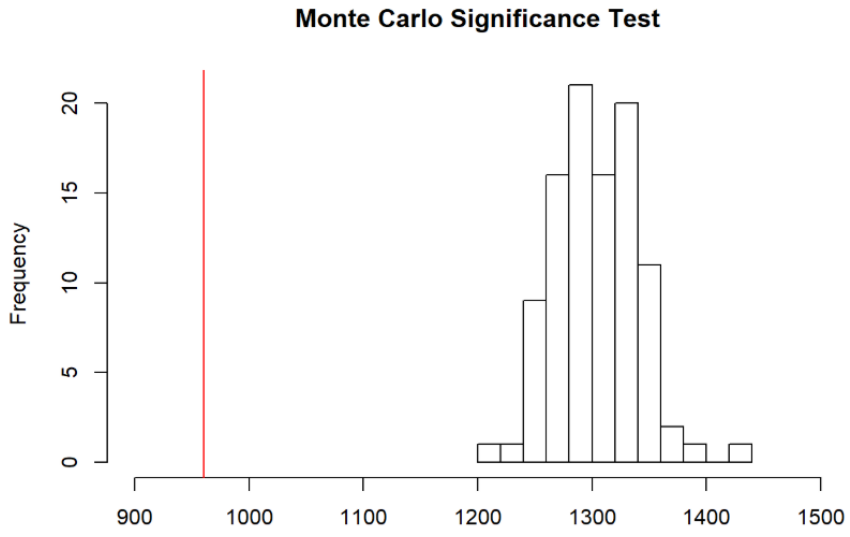
- 圖資：
  - 台南市學校 schools.shp
  - 縣市邊界圖層 taiwan_county.shp
- 分析方法：

  （以行政區範圍為研究區邊界）
  - 1. Nearest Neighbor Analysis
  - 2. K-order Nearest Neighbor Indices
  - 3. G Function
- 用Monte Carlo Simulation檢定統計顯著性

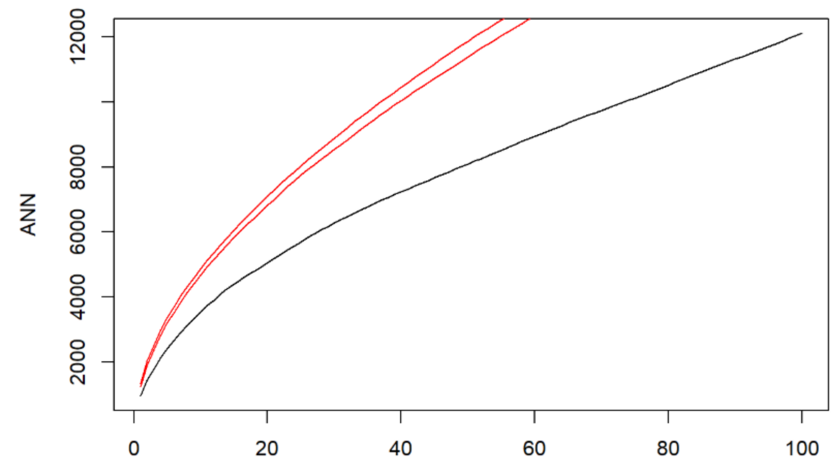


参考答案

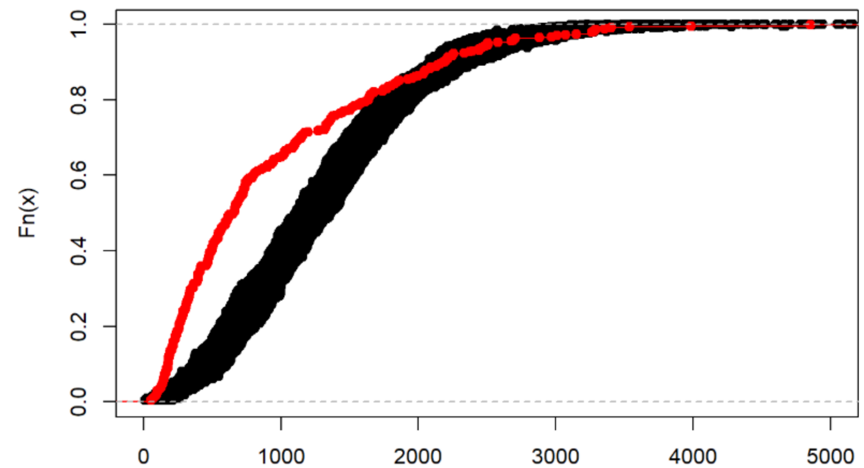
Nearest Neighbor Analysis



K-order Nearest Neighbor Indices



G Function



本週作業 1（分別上傳 html 與 pdf files）

- 利用課堂提供的資料，利用Nearest-Neighbor Distances，比較任兩個縣市信仰「觀音菩薩」的村落型祭祀圈的寺廟空間群聚特性，並討論之。（包括：Nearest Neighbor Analysis, K-order Nearest Neighbor Indices, and G(d) Function）

本週作業 2（分別上傳 html 與 pdf files）

- 論文研讀摘要與個人心得：Reading_Temporal.Land.Use.pdf
  Temporal land-use pattern analysis with the use of nearest neighbor and quadrat methods, *Annals of the Association of American Geographers*, 1964
- 心得內容應包括以下部分：
  - 該分析方法如何與就讀科系專業的連結
  - 分析方法有任何進一步精進或改善的想法