點型態分析
鄰近分析

空間分析 2021.05.03

TA 杜承軒

Lab 6

以行政區範圍為研究區邊界，使用以下方法進行點型態分析，並用 Monte Carlo Simulation 檢定統計顯著性

1. ***Nearest Neighbor Analysis***
2. ***K-order Nearest Neighbor Indices***
3. ***G Function***

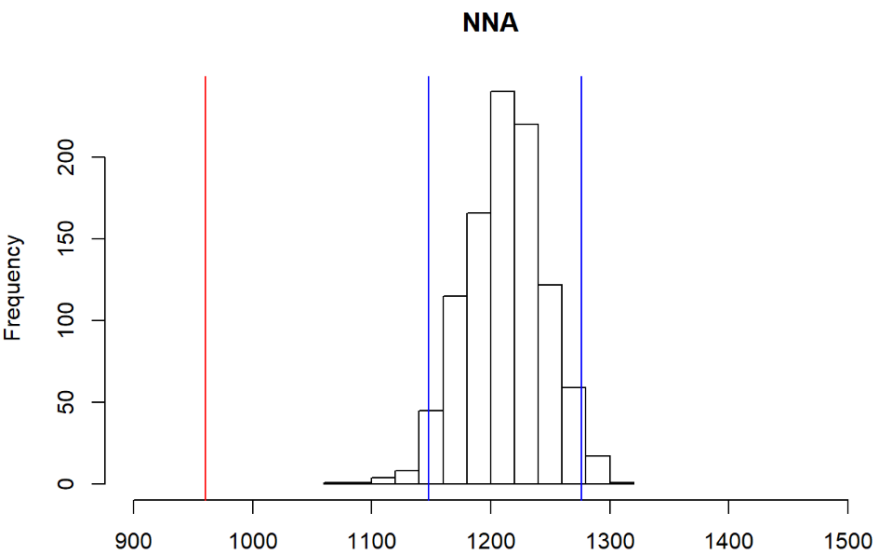
■ 圖資：

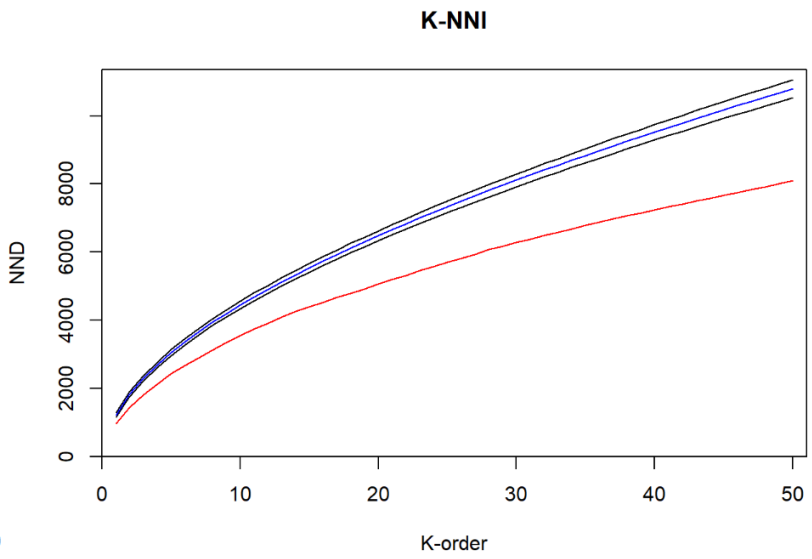
- 臺南學校點位

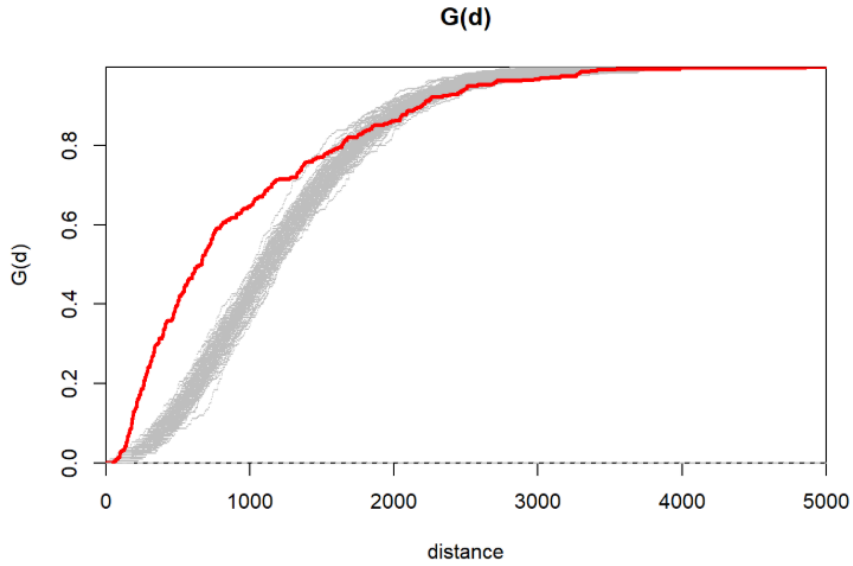
  schools.shp

- **臺南市行政區**

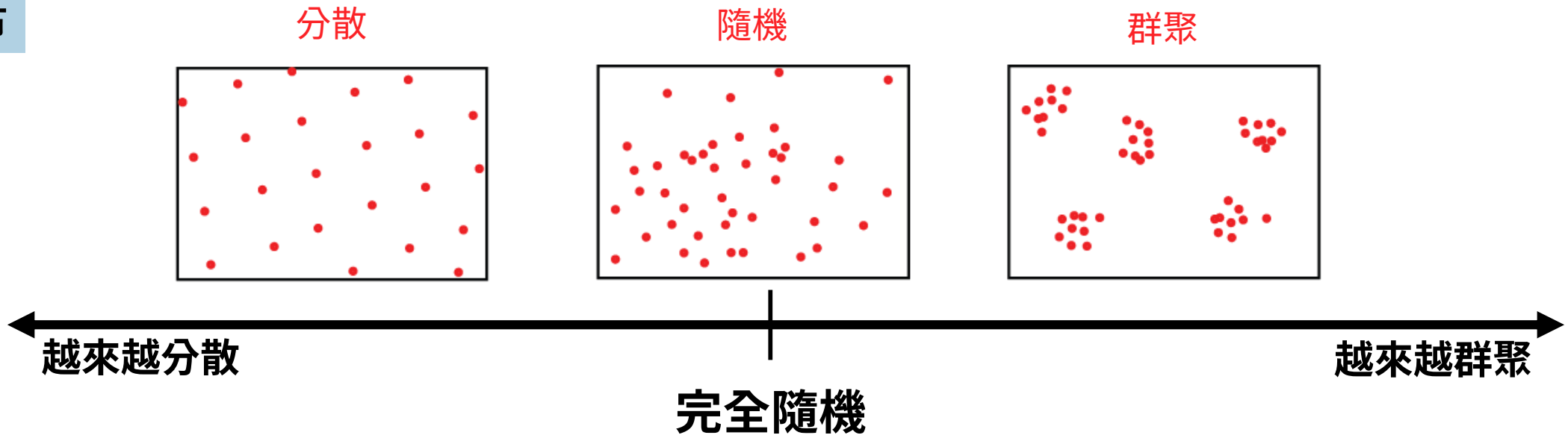
  TainanCounty.shp







點型態分布



	H_0		H_a		p-value	$p \leq \alpha$	$p > \alpha$
雙尾	隨機	$VMR = 1$	非隨機	$VMR \neq 1$	`2*pt(t,k-1,lower.tail=t<0)`	非隨機	隨機
單尾 (右尾)	非群聚	$VMR = 1$ $(VMR \leq 1)$	群聚	$VMR > 1$	`pt(t,k-1,lower.tail=F)`	群聚	非群聚
單尾 (左尾)	非均勻	$VMR = 1$ $(VMR \geq 1)$	均勻	$VMR < 1$	`pt(t,k-1)`	均勻	非均勻

※以VMR舉例

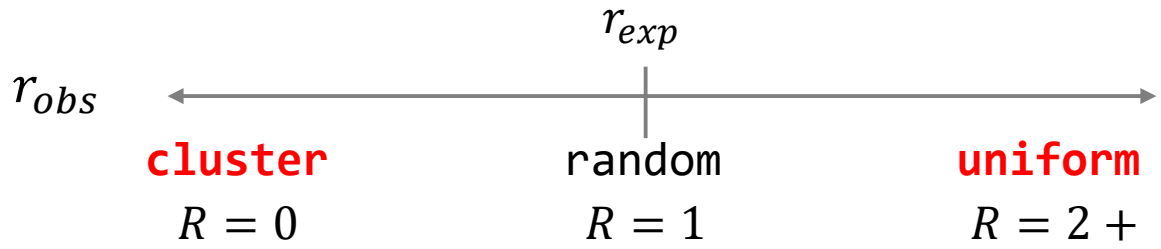
NNA & K-order NNI

NNA　　k-NNI

#1 - 每一個點，找**最近的點**的距離／找**前k近的點**的距離

#2 - 所有距離的**平均**，得到r_{obs}

(#3 - **觀察值**與**理論隨機值**的比值：$R = {r_{obs}}/{r_{exp}}$)



顯著性檢定

1. 理論隨機分布(NNA)

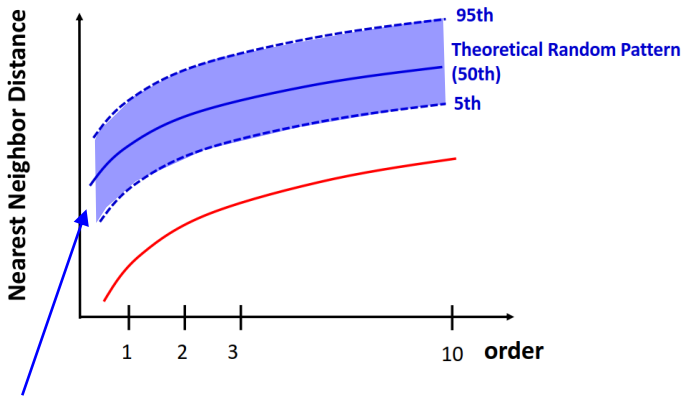
$$Z = \frac{r_{obs} - r_{exp}}{s.e.}$$

$$r_{exp} = \frac{0.5}{\sqrt{{n}/{A}}}$$

$$s.e. = \frac{0.26136}{\sqrt{{n^2}/{A}}}$$

2. Monte Carlo 顯著性檢定

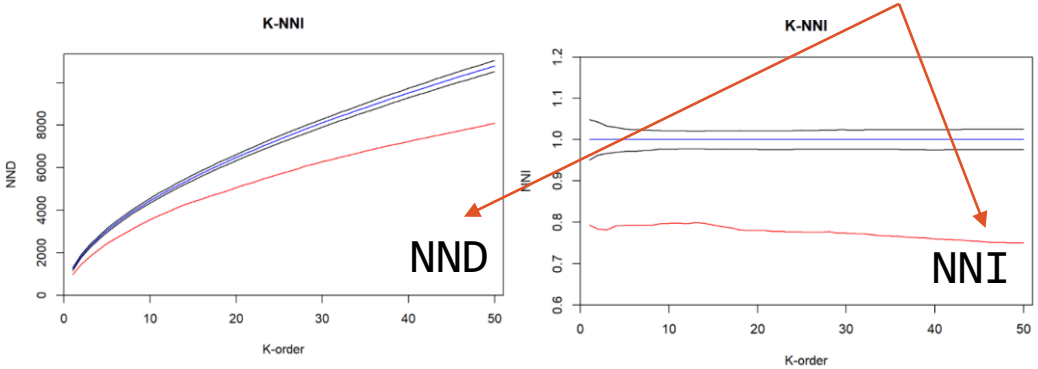
※ 蒙地卡羅方法：隨機模擬方法的統稱，用大量隨機抽樣來計算



信賴包絡(confidence envelope)

模擬隨機分布 (i.e.1000次)
判斷是否落在隨機的信賴包絡之中
(i.e.1000次中，排序前後50名的數值)

概念相同
(只差有沒有除以隨機值)



sf 實作
NNA & K-NNI

Q：如何用過去已經會的函數來實作？

#1 - 每一個點，找**前k近的點**的距離

#2 - 所有距離的**平均**，得到r_{obs}

NNA

```
D = st_distance(schools)
near.dist=apply(D,1,function(x) sort(x)[1+1])
mean(near.dist) → 960.67
```

k-NNI 1~10

```
near.dists=apply(D,1,function(x) sort(x)[1:10+1])
rowMeans(near.dists) → 961 1434 1802 2138 2419 2671 2904 3120 3344 3549
```

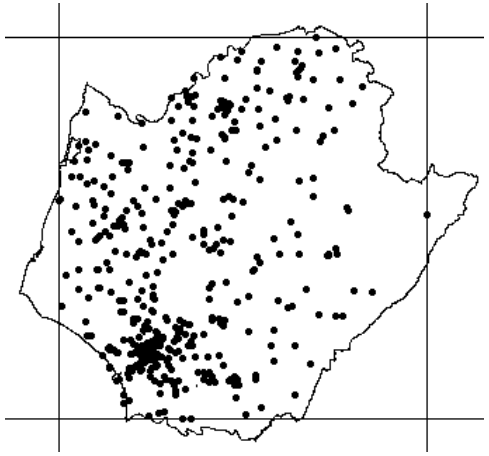
spatstat 實作

→ 設定邊界，可以創隨機點，進行蒙地卡羅顯著性檢定

→ ppp格式：包含「點位」、「邊界」

ppp格式

ppp(x座標,y座標,邊界範圍)



點

```
coord = st_coordinates(schools)
x.coor = coord$X
y.coor = coord$Y
```

邊界

(1)點邊界矩形 (boundary box)

```
rect = st_bbox(schools)
x.range = c(rect[1], rect[3])
y.range = c(rect[2], rect[4])
Windows = owin(xrange=x.range, yrange=y.range)
```

(2)多邊形sf

```
Windows = as.owin(TN)
```

(3)多邊形座標點

```
xy=st_coordinates(TN)
xp=rev(xy[,1]);yp=rev(xy[,2])
newxy=cbind(xp,yp)
Windows=owin(poly=newxy)
```

※逆時針順序

轉換成ppp

ppp(x.coor,y.coor,Windows)

P.S.矩形邊界可以直接寫成

ppp(x.coor,y.coor,x.range,y.range)

或

as.ppp(coord, Windows)

ppp 實作
NNA & k-NNI

#1 - 每一個點，找**前k近的點**的距離

#2 - 所有距離的**平均**，得到r_{obs}

NNA

```
near.dist = nndist(schools.ppp, k=1)
mean(near.dist) → 960.67
```

k-NNI 1~10

```
near.dists =nndist(schools.ppp, k=1:10)
colMeans(near.dists) 或 apply(near.dists,2,mean)
→ 961 1434 1802 2138 2419 2671 2904 3120 3344 3549
```

產生隨機點(ppp格式)

```
RandomPts = rpoint(n, win=Windows)
# n = 隨機點個數
# n = schools.ppp$n
```

重複模擬概念

※ 模擬1000次 or 999次

NNA

```
mean(nndist(points.ppp, k=1))
```

MC(NNA)

```
mean(nndist(RandomPts, k=1))
```

→ 重複1000次找前後2.5%

K-NNI

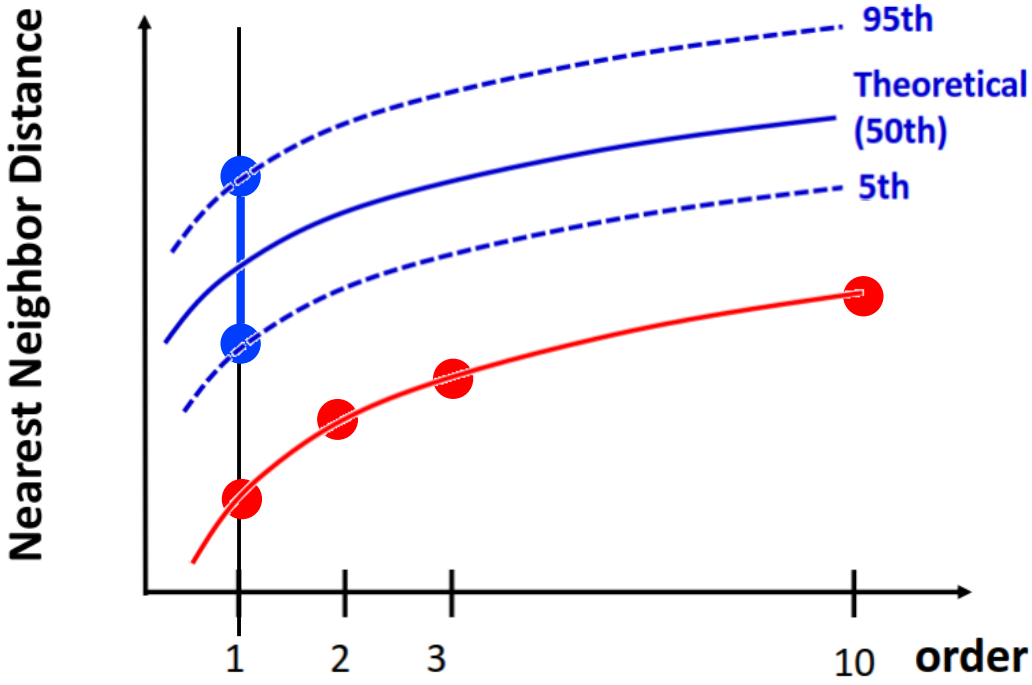
```
apply(nndist(points.ppp, k=1:10),2,mean)
```

MC(K-NNI)

```
apply(nndist(RandomPts, k=1:10),2,mean)
```

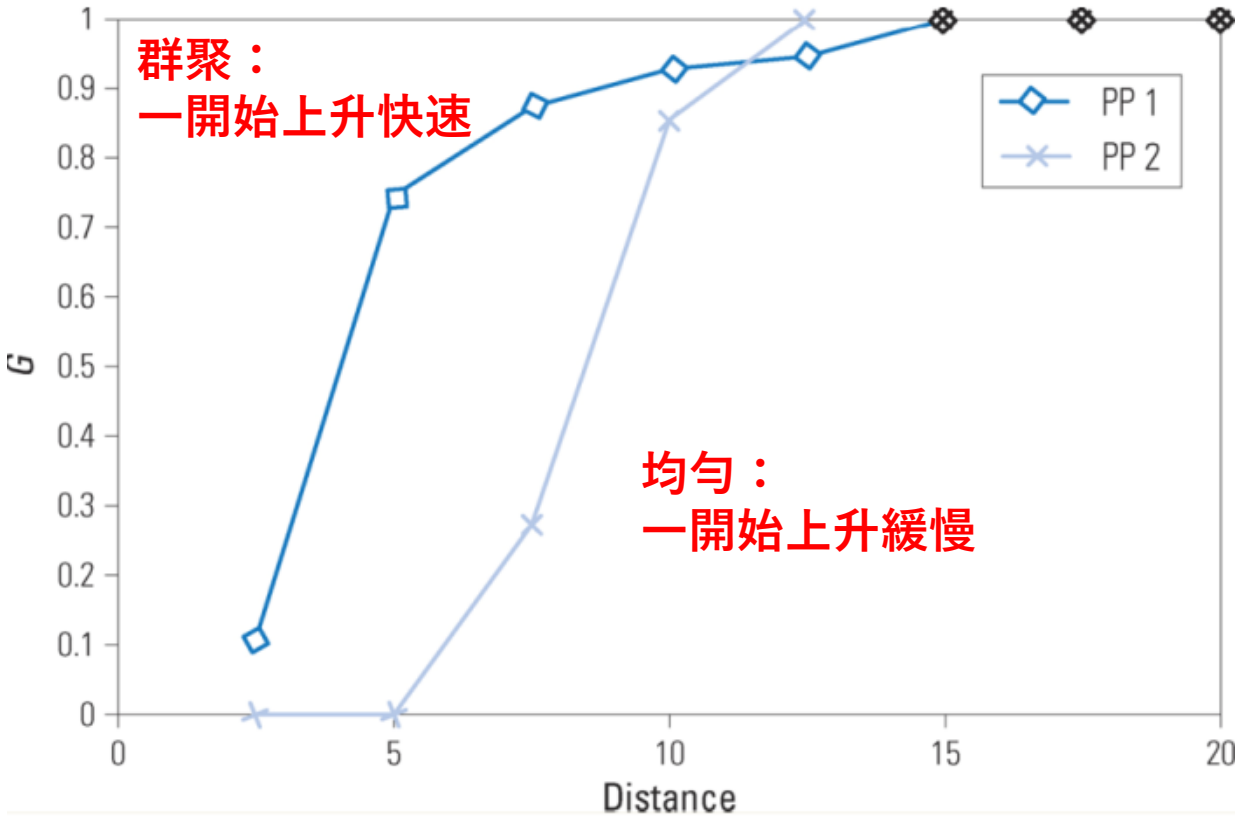
→ 重複1000次找前後2.5%

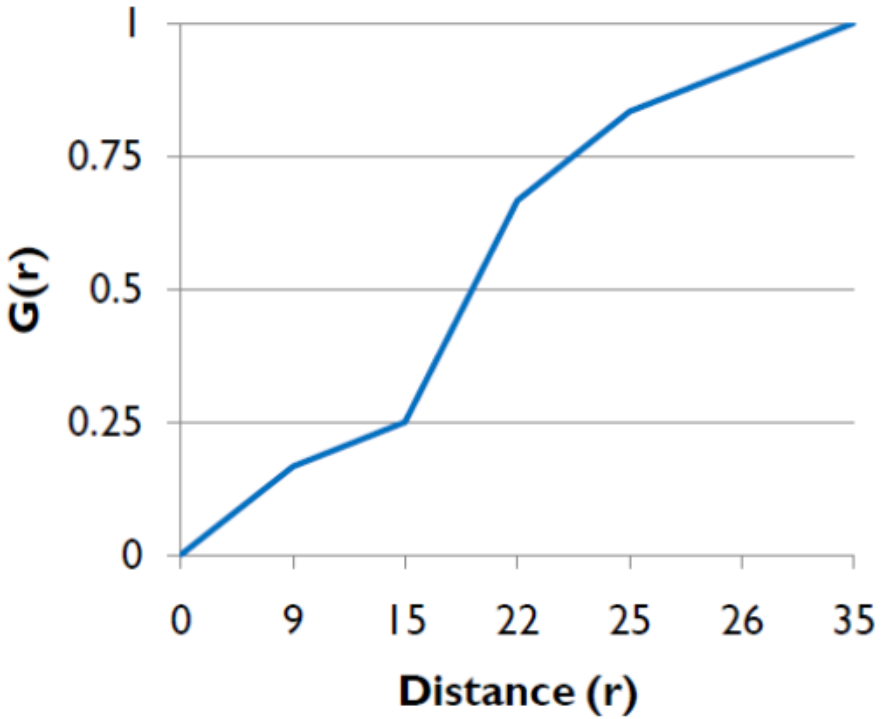
※ 排序函數：sort()
※ 善用sapply重複計算



G Function

「最近點的距離」的累積頻率分布



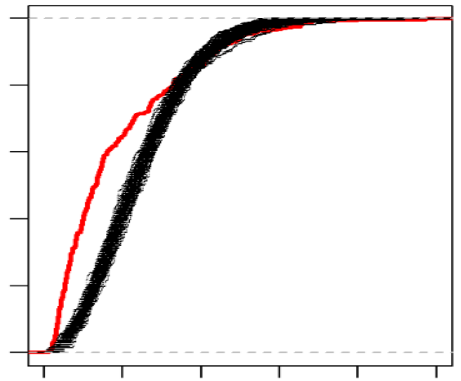


G 實作

```
near.dist = nndist(schools.ppp, k=1)
G = ecdf(near.dist)
```

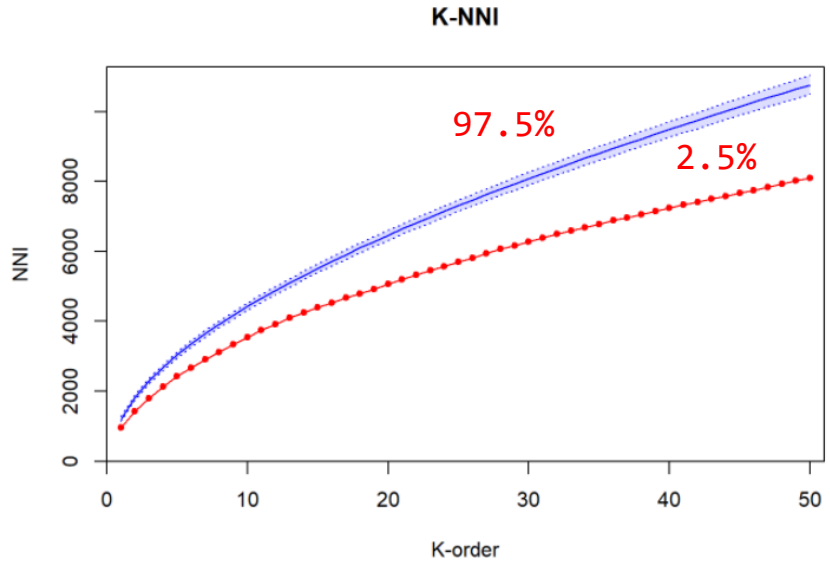
→累積頻率分布

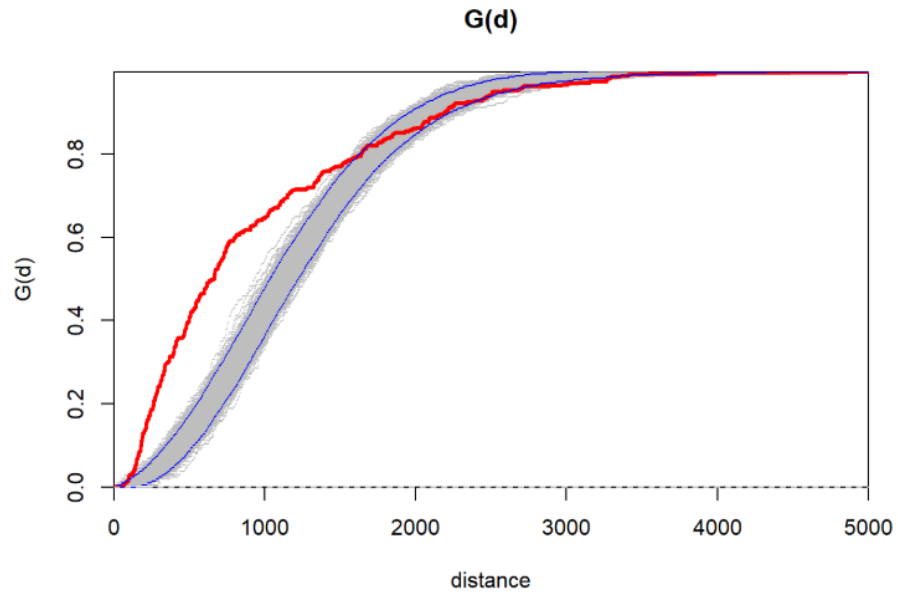
Monte Carlo
顯著性檢定



實習Hint

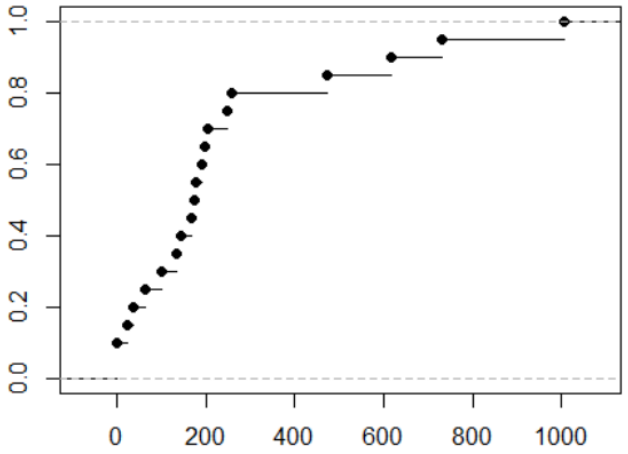
- $\alpha = ?$
  單尾 or 雙尾？
- K-NNI
  y軸為距離(NND)即可
- G(d)
  列出所有的模擬值即可



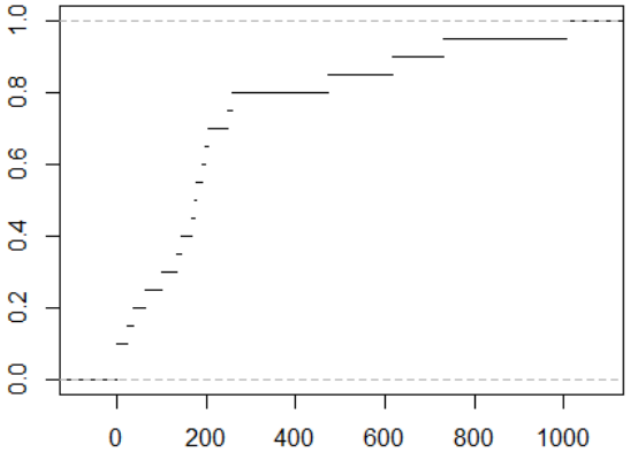


ecdf() 繪圖

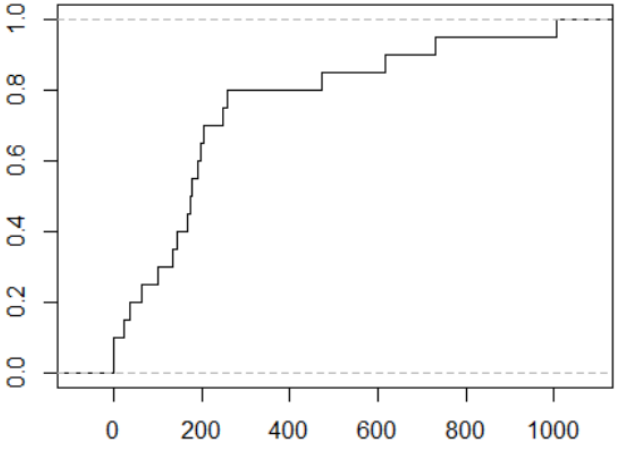
plot(ecdf(x))



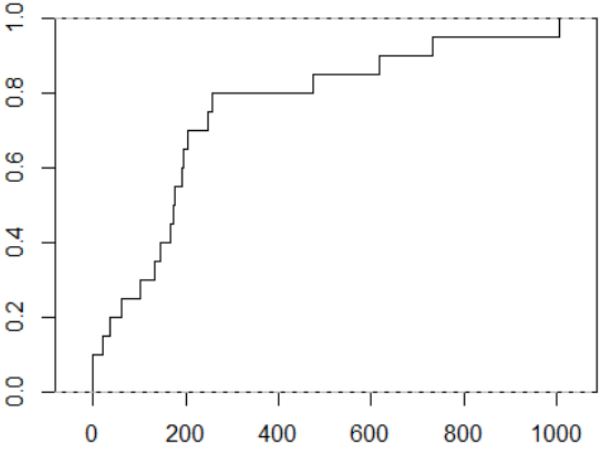
+ cex=0

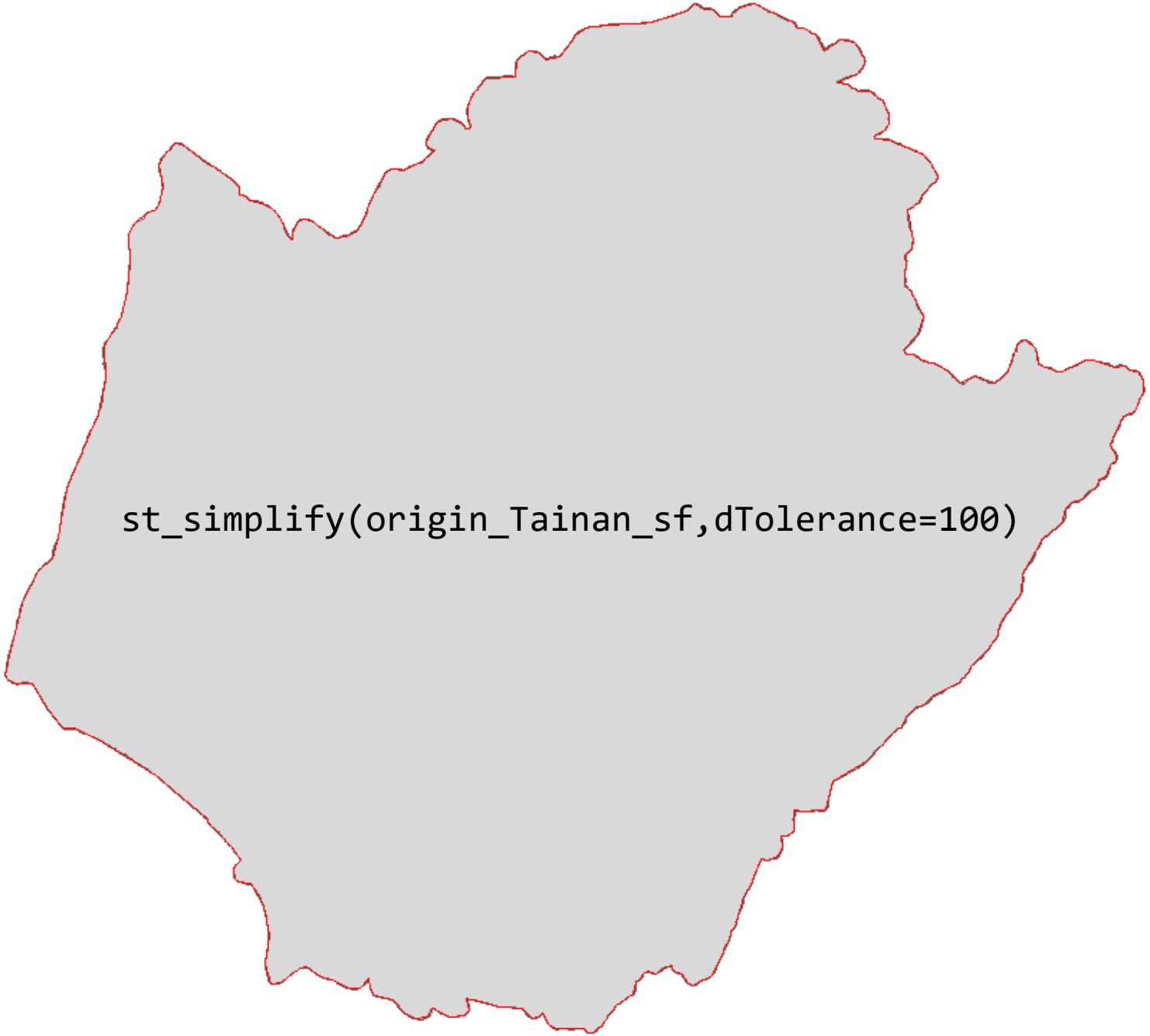


+ verticals=T



+ yaxs="i"




st_simplify(origin_Tainan_sf,dTolerance=100)