



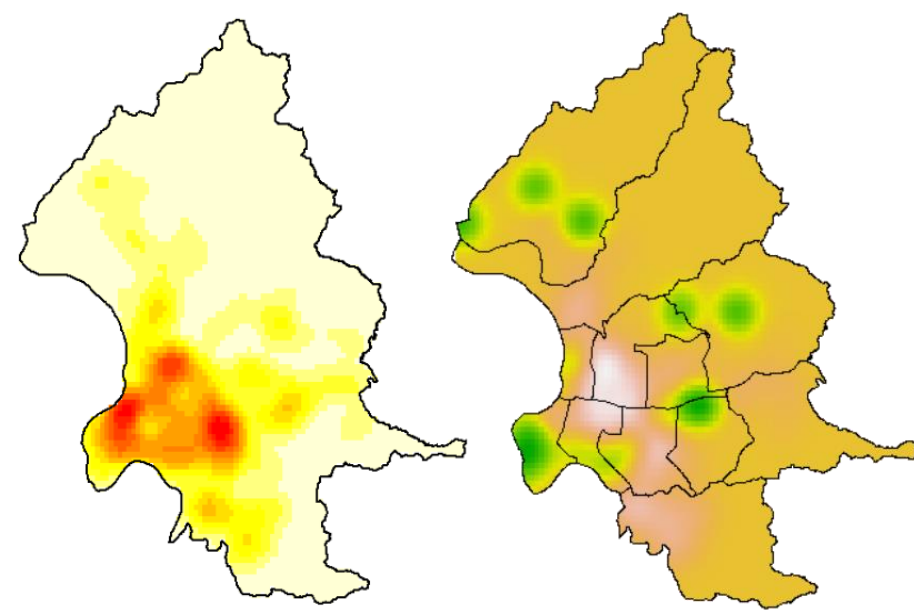
點空間型態 密度分析

空間分析 2020.05.11
TA 杜承軒

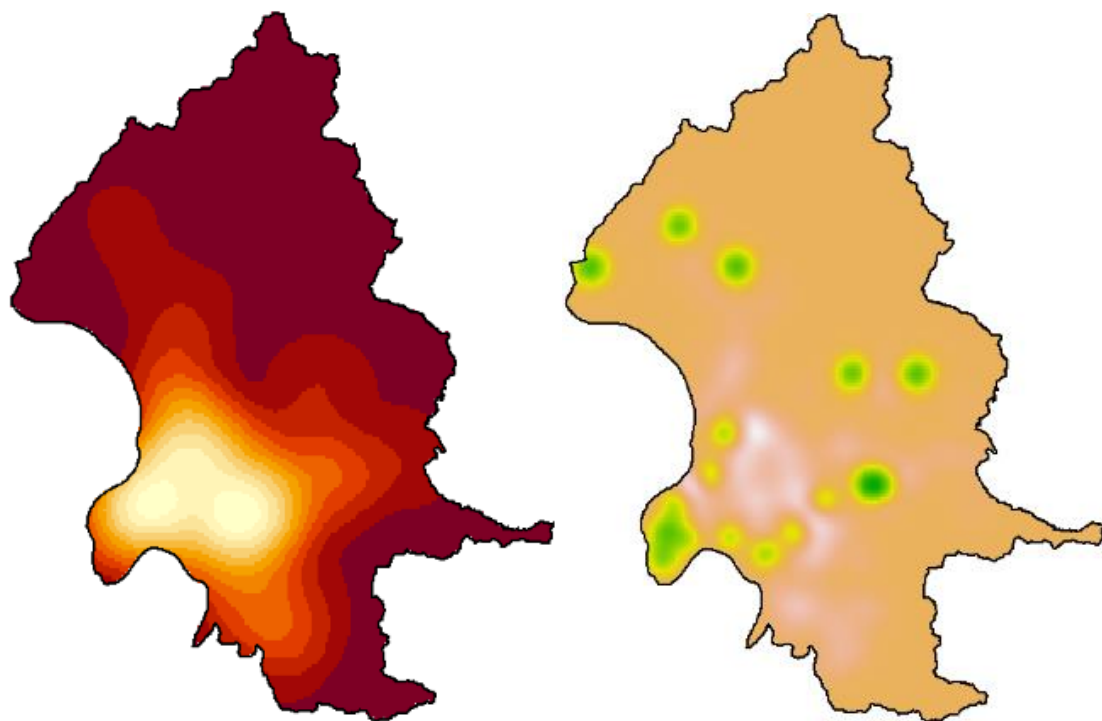
實作 Dual KDE : PTS1 - PTS2

利用 **splancs** 與 **GISTools** 的R套件繪製KDE地圖

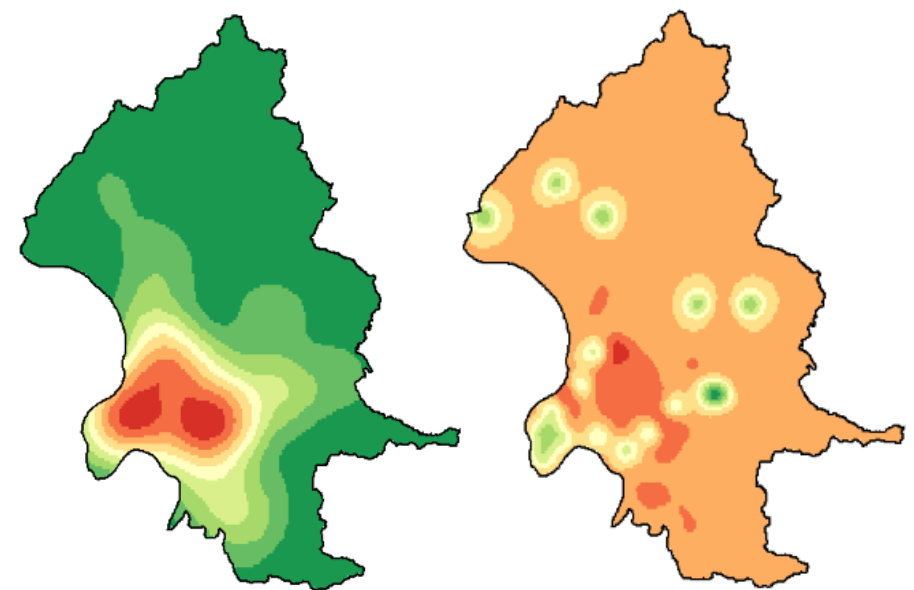
- 搜尋半徑 2000 公尺
- 網格數量 200×200 個
- **splancs** - tpe_sqr_bnd.csv 作為臺北市邊界
- **GISTools** - Taipei_BND.shp 作為臺北市邊界



實習範例



套色後



KDE 核密度估計

- Step 1: 研究區域建立均勻**網格**
- Step 2: 設定**搜尋半徑** (bandwidth)
- Step 3: 選擇**核密度函數** (Kernel function)

1. 均勻網格

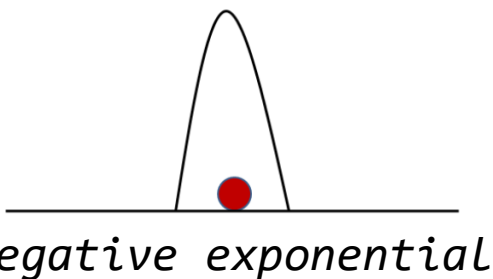
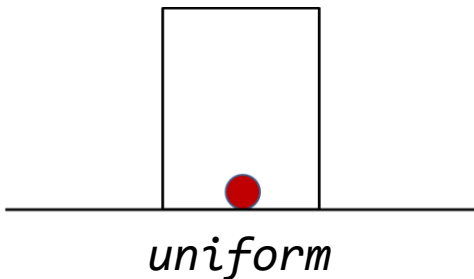
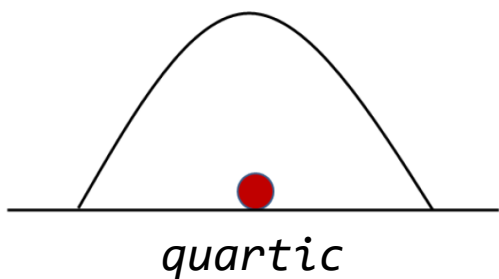
- Q: 網格要多細？
A: 考慮呈現結果、計算量的大小.....

2. 搜尋半徑

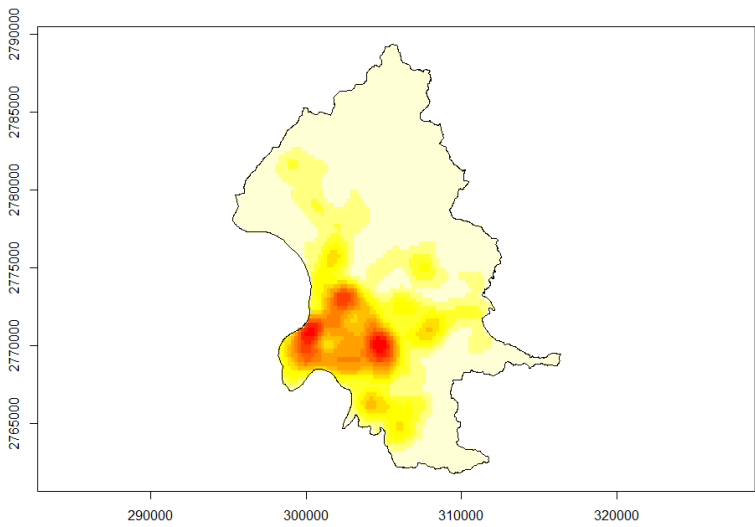
- Q: 搜尋半徑設多少？
A: (1) MISE → `mse2d(pts,poly,nsmse,range)`
(2) 演算法
(3) 影響半徑: κ階鄰近分析 / 空間自相關分析 / 自訂
→ `kNNdistplot()`

3. 核密度函數

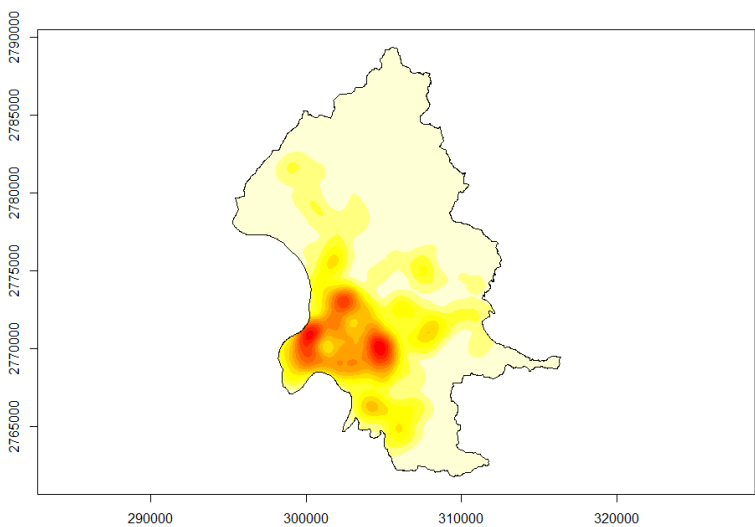
- Q: 設定核密度函數意義？
A: 隨著距離增加，相關性遞減的效果



kernel2d(nx=ny=10)



kernel2d(nx=ny=1000)



splancs

```
kernel12d(pts,poly,h0,nx=20,ny=20,kernel='quartic',quiet=FALSE)
```

讀檔

```
pts = as.points(xcoord, ycoord) #設定事件點  
bnd = as.points(xcoord, ycoord) #設定邊界
```

KDE

計算 KDE

搜尋半徑 網格數量(xy方向)

```
kde = kernel12d(pts, bnd, 2000, 200, 200)
```

Dual KDE

KDE 相減

```
diff = kde$z-kde2$z  
kde.diff = list(x=kde$x, y=kde1$y, z=diff)
```

或

```
kde.diff = kde  
kde.diff$z = kde$z-kde2$z
```

繪圖

```
polymap(bnd)            #底圖  
image(kde, add=T)    #KDE圖(可用col設定顏色)
```

GISTools

```
kde.points(pts,h,n=200,lims=NULL)
```

讀檔

```
PTS = SpatialPoints(pts, proj) #設定事件點  
BND = readOGR(.....) #設定邊界，直接讀取shp檔
```

KDE

計算 KDE

搜尋半徑 網格數量(單方向)

```
KDE = kde.points(PTS, 2000, 200, lims=BND)
```

Dual KDE

KDE 相減

+ raster

```
KDE1.R = raster(KDE)  
KDE2.R = raster(KDE2)  
KDE.DIFF = KDE1.R-KDE2.R #raster可直接相減
```

繪圖

```
plot(KDE) #KDE圖  
masker=poly.outer(KDE, BND) #建立遮罩  
add.masking(masker, col="blue") #覆蓋遮罩  
plot(BND, add=T) #加邊框
```

ggtern

```
kde2d(x, y, h, n = 25, lims = c(range(x), range(y)))
```

讀檔

```
#劃定邊界  
lims=c(TPE@bbox[1,], TPE@bbox[2,])
```

KDE

計算 KDE

搜尋半徑 網格數量(xy方向)

```
Kde = kde2d(xcoord, ycoord, 2000, 200, lims)
```

Dual KDE

KDE 相減

同splancs的方法

```
Kde.diff = Kde  
Kde.diff$z = Kde$z-Kde2$z
```

繪圖

遮罩同GISTools的方法

```
image(Kde, asp=1) #xy比例一樣  
masker=poly.outer(as.points(TPE@bbox[1,],TPE@bbox[2,]),TPE) #建立遮罩  
add.masking(masker, col="white") #覆蓋遮罩  
plot(TPE,add=T) #加邊框
```

kde2d.weighted
(X,Y,1000,50,W)

X	Y	W
301905	2779551	1.26
301692	2769025	3.25
304126	2769248	3.95
300076	2769041	5.45
306460	2775395	1.63
303025	2768395	3.86
299044	2768237	2.96
308974	2775350	4.23

Weighted KDE

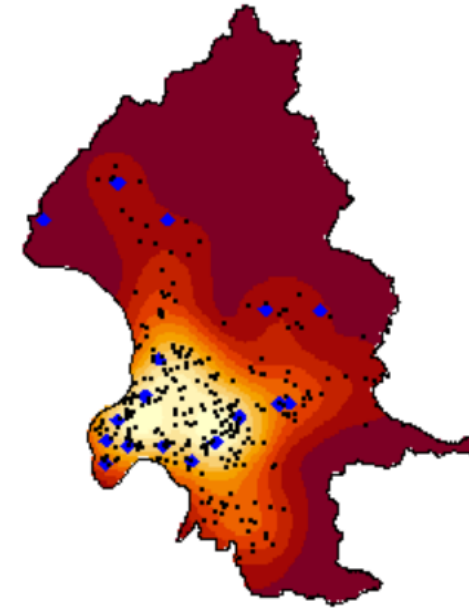
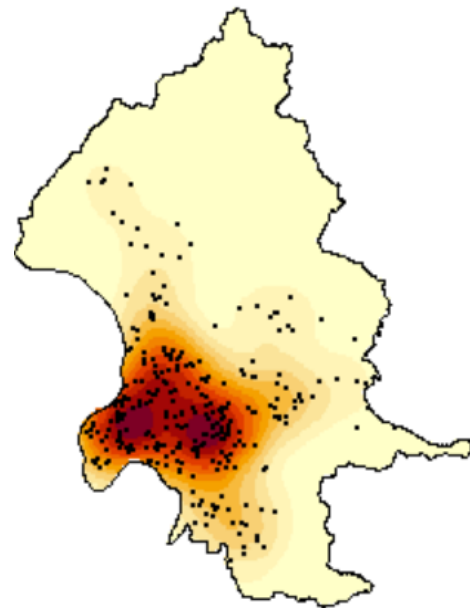
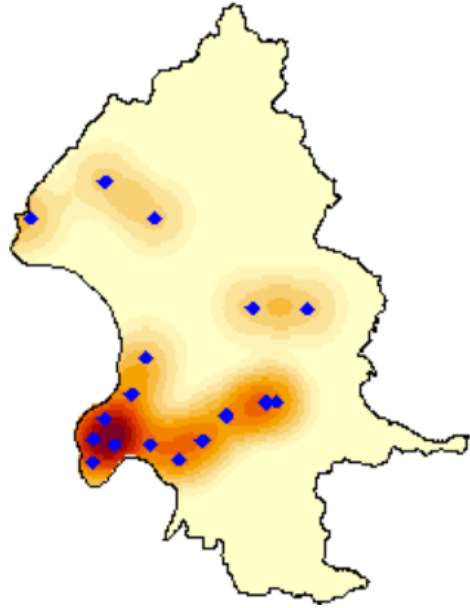
```
kde2d.weighted(x, y, h, n = 25, lims = c(range(x), range(y)),w)
```

KDE 1

KDE 2

Dual KDE

splancs



GISTools

