# 點空間型態
# 密度分析

空間分析 2019.04.29

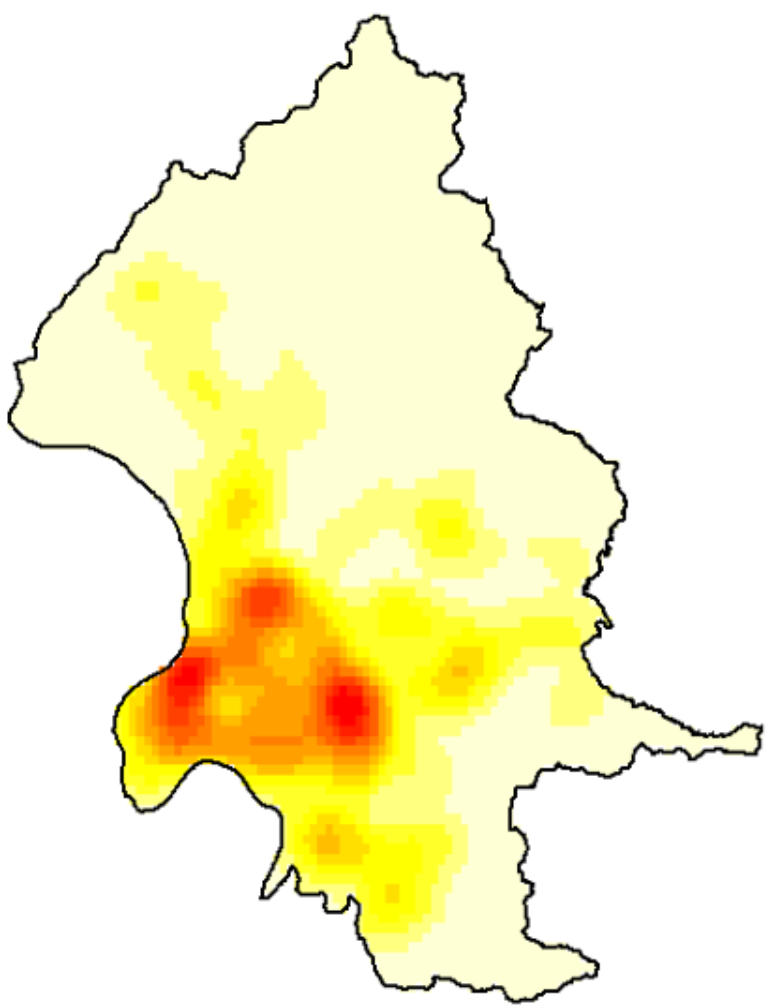
TA 杜承軒

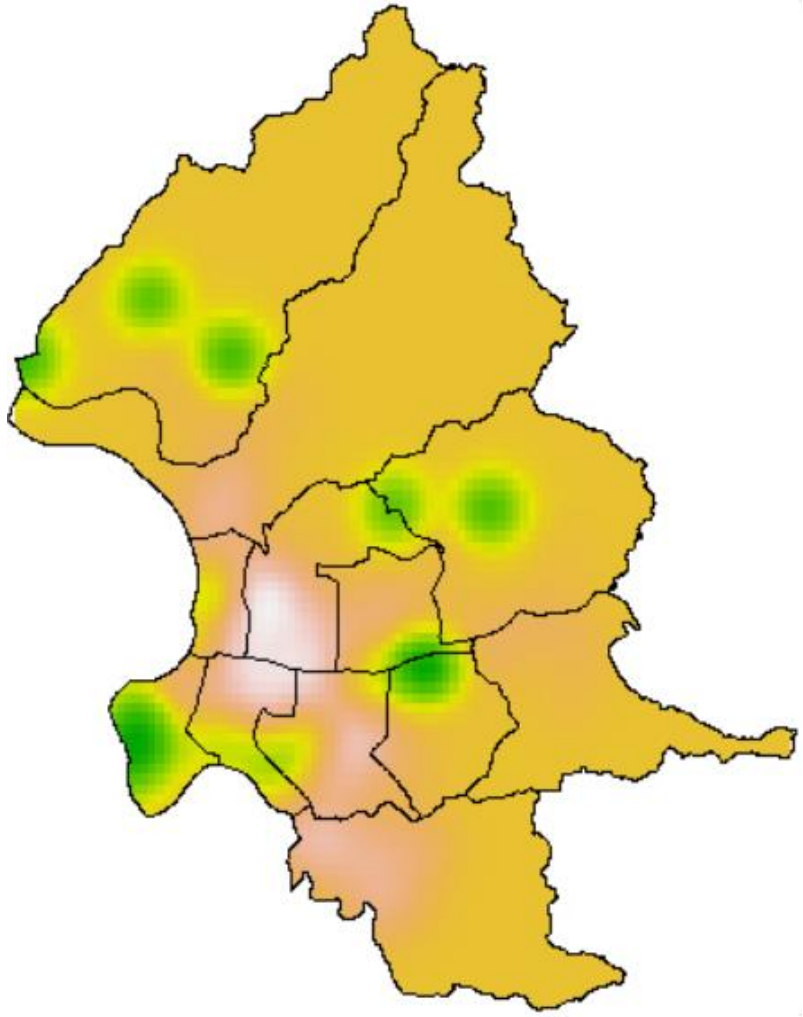
Lab 9

# 實作 Dual KDE：PTS1 – PTS2

利用 splancs 與 GISTools 的R套件繪製KDE地圖

實習八
參考





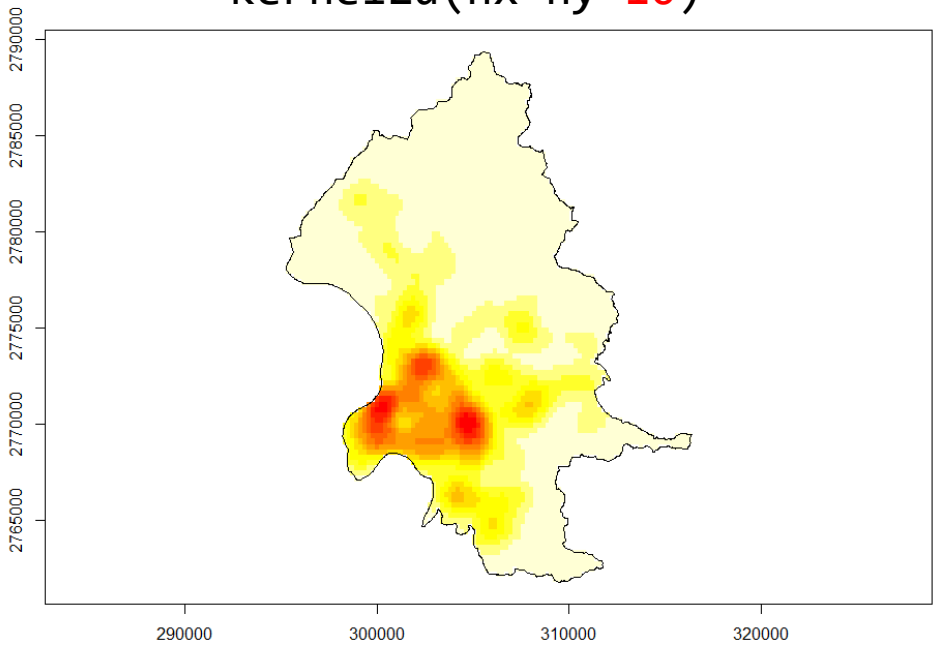
## KDE
## 核密度估計

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)

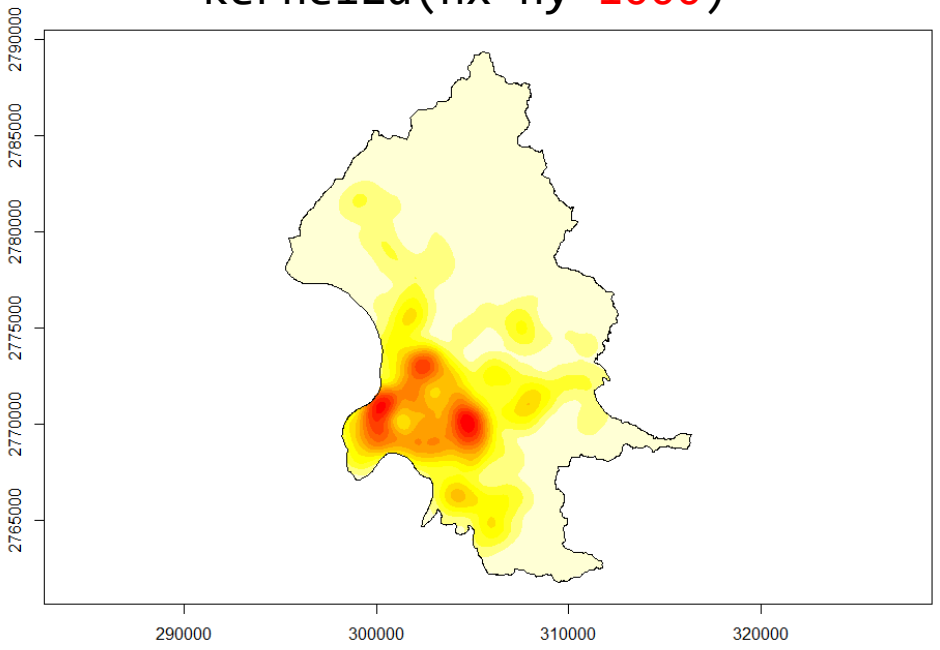
## 1.均勻網格

Q:網格要多細？
A:考慮呈現結果、計算量的大小……

kernel2d(nx=ny=10)




kernel2d(nx=ny=1000)



## 2.搜尋半徑

Q:搜尋半徑設多少？

A: (1)MISE (2)演算法 (3)影響半徑

↓

mse2d()

mse2d(pts,poly,nsmse,range)

-K階鄰近分析

-空間自相關分析

-**自訂**

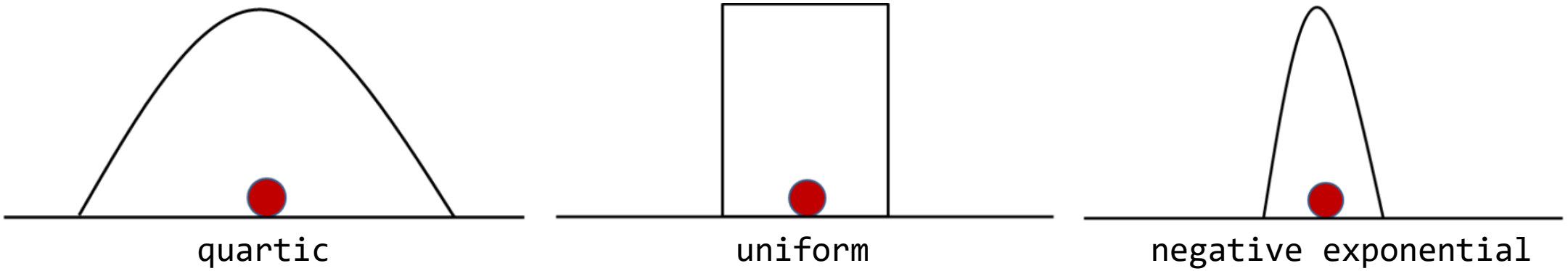
[補充]

→ 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



KDE

splancs

```
kernel2d(pts,poly,h0,nx=20,ny=20,kernel='quartic',quiet=FALSE)
```

**讀檔**

```
pts = as.points(x座標, y座標) #設定事件點
bnd = as.points(x座標, y座標) #設定邊界
```

**計算KDE**

```
kde.pts = kernel2d(pts, bnd,  1500,    50, 50)
```

1500：搜尋半徑

50, 50：網格數量(xy方向)

**繪圖**

```
polymap(bnd)                    #底圖
image(kde.pts, add=T)           #KDE圖
```

KDE

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
```

**計算KDE**

```
KDE.PTS = kde.points(PTS,  3000,   100,  lims=BND)
```

3000：搜尋半徑

100：網格數量(單方向)

**繪圖**

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

Dual KDE

splancs

讀檔

計算KDE

```
kde1 = kernel2d(pts1, bnd, 1500, 50, 50)
kde2 = kernel2d(pts2, bnd, 1500, 50, 50)
```

KDE相減

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

或

```
kde.diff = kde1
kde.diff$z = kde1$z-kde2$z
```

繪圖

Dual KDE

GISTools

raster

讀檔

**計算KDE**

```
KDE1 = kde.points(PTS1, 3000, 100, BND)
KDE2 = kde.points(PTS1, 3000, 100, BND)
```

**KDE相減**

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

繪圖