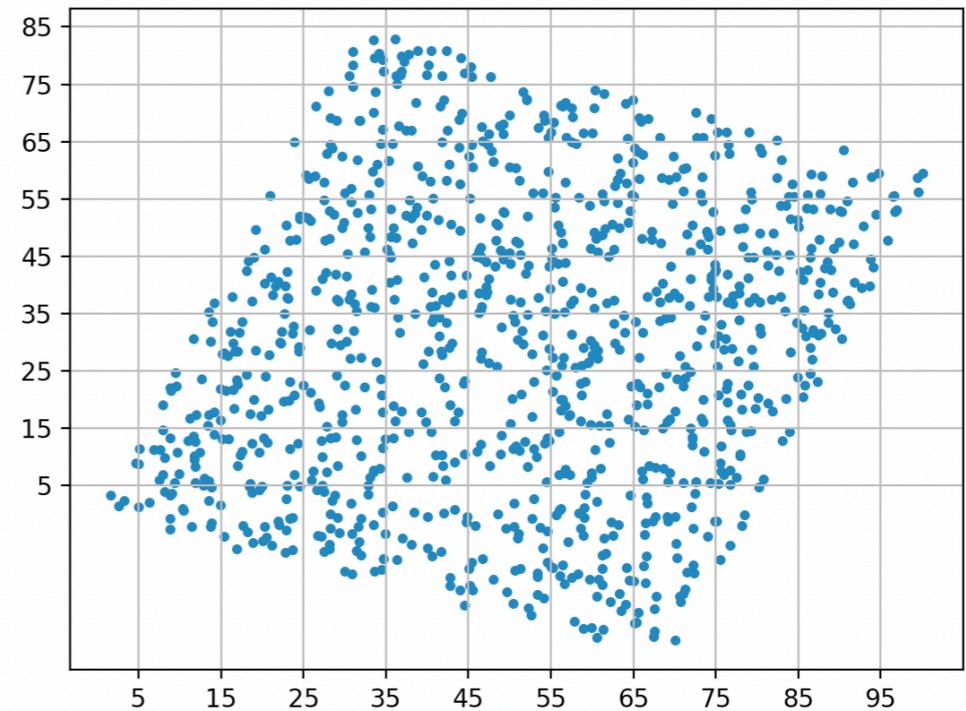
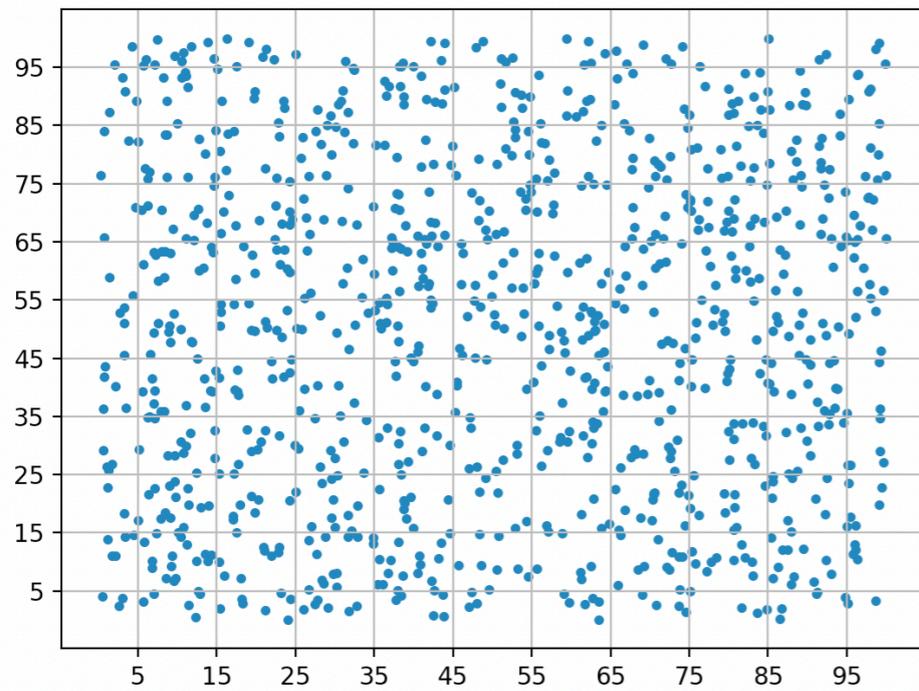


幾何圖形樣本資料分布的交叉熵 (亂度) **Cross Entropy**計算



實驗29、產生二維正方形資料點並繪製資料點與棋盤格子

實驗30、二維正方形資料點在x座標軸投影的累計分布與y座標軸投影的累計分布

實驗31、計算正方形資料點、旋轉資料點、菱形資料點的**Cross Entropy**

實驗29、產生二維正方形資料點並繪製資料點與棋盤格子

步驟一、匯入套件

```
import matplotlib.pyplot as plt  
import random  
import numpy as np
```

步驟二

在 $[0, 100] \times [0, 100]$ 產生
1000二維資料點

```
def produce_sig(N): # 1A
```

```
    x_list = []
```

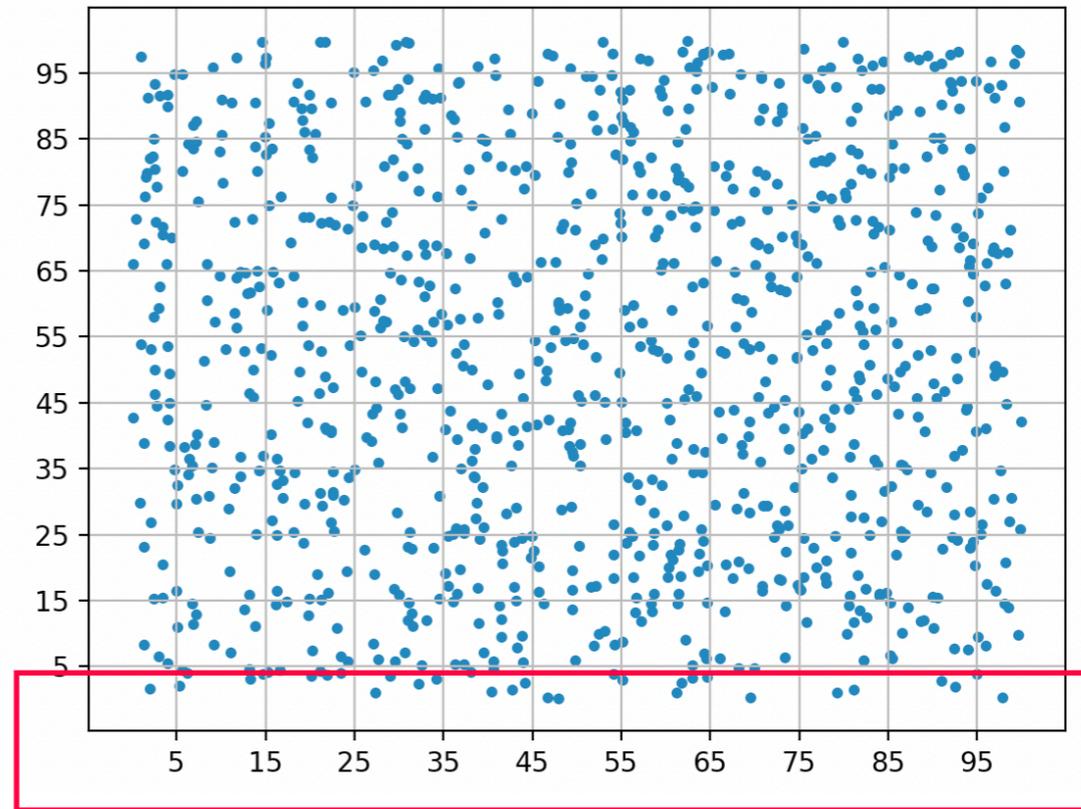
```
    y_list = []
```

請以for迴圈產生資料點的x座標與y座標
，分別附加在x_list、y_list串列

```
return x_list, y_list
```

步驟三、在串列最小值與
最大值間產生n個格點

令max_v代表data_list的最小值，min_v代表data_list的最小值，請在區間I = [max_v,min_v]取n個點，將區間I等量切割為n個不重疊的子區間，串列bin的元素代表n個切割點



```
def bin_points_list(data_list,n):
```

請完成此程式

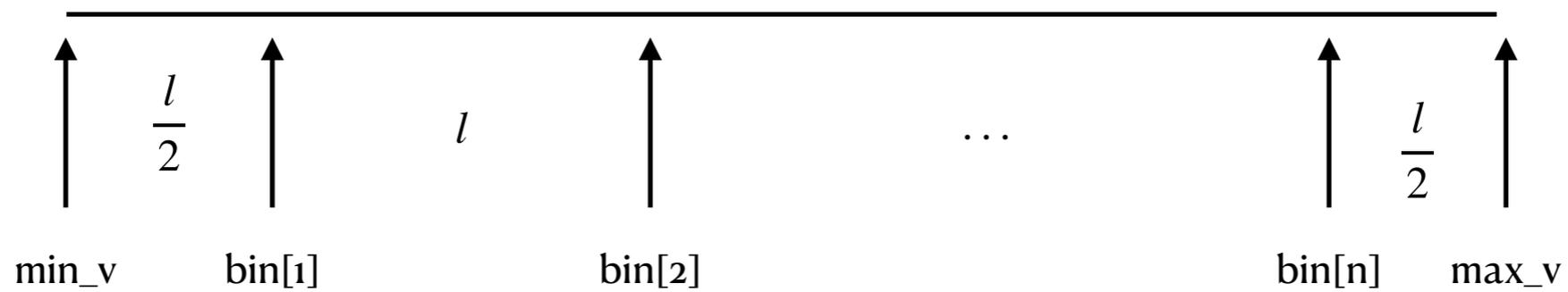
```
return bin
```

$$\text{Interval length } l = \frac{\text{max_v} - \text{min_v}}{n}$$

$$\text{bin}[1] = \text{min_v} + \frac{l}{2}$$

$$\text{bin}[2] = \text{bin}[1] + l$$

$$\text{Check point : } \text{bin}[n] = \text{max_v} - \frac{l}{2}$$

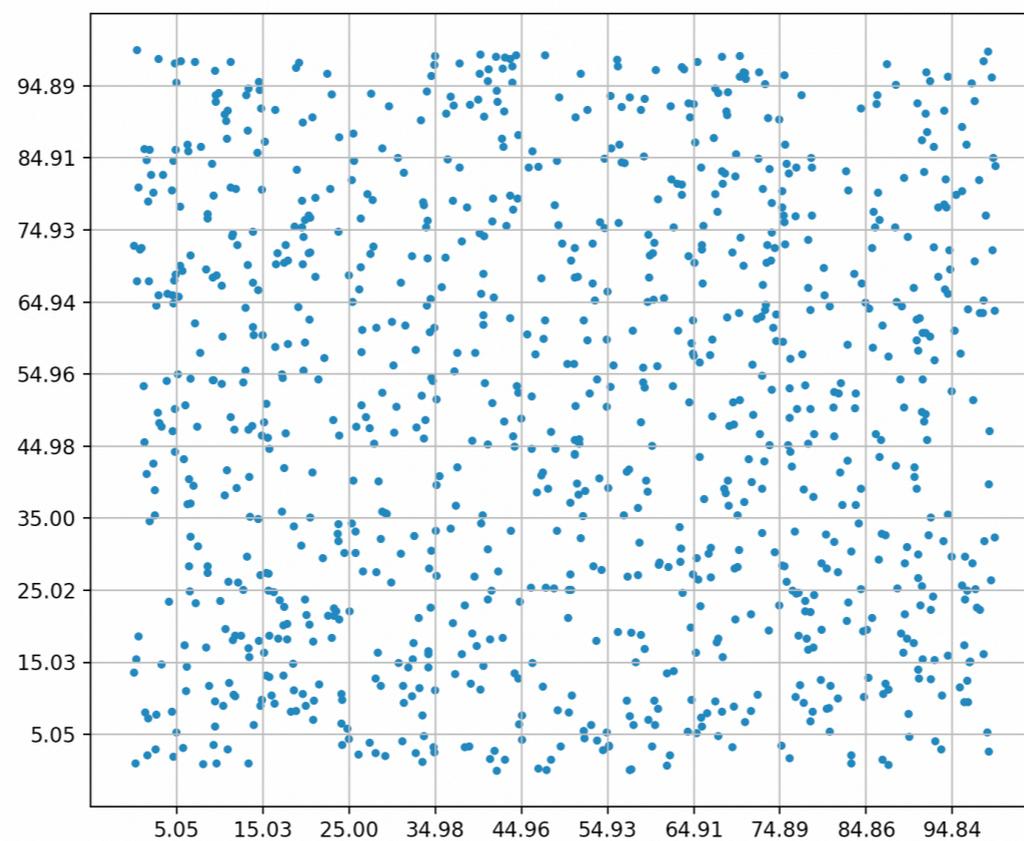


步驟四、繪製有格點的二 維資料點圖

```
def plot_points(x_list,y_list,xticks, yticks): # 1D  
    fig = plt.figure()  
    ax = fig.gca()
```

完成函數plot_points繪製資料點與棋盤格子

步驟五、產生二維正方習 資料點並繪製資料點與棋 盤格子

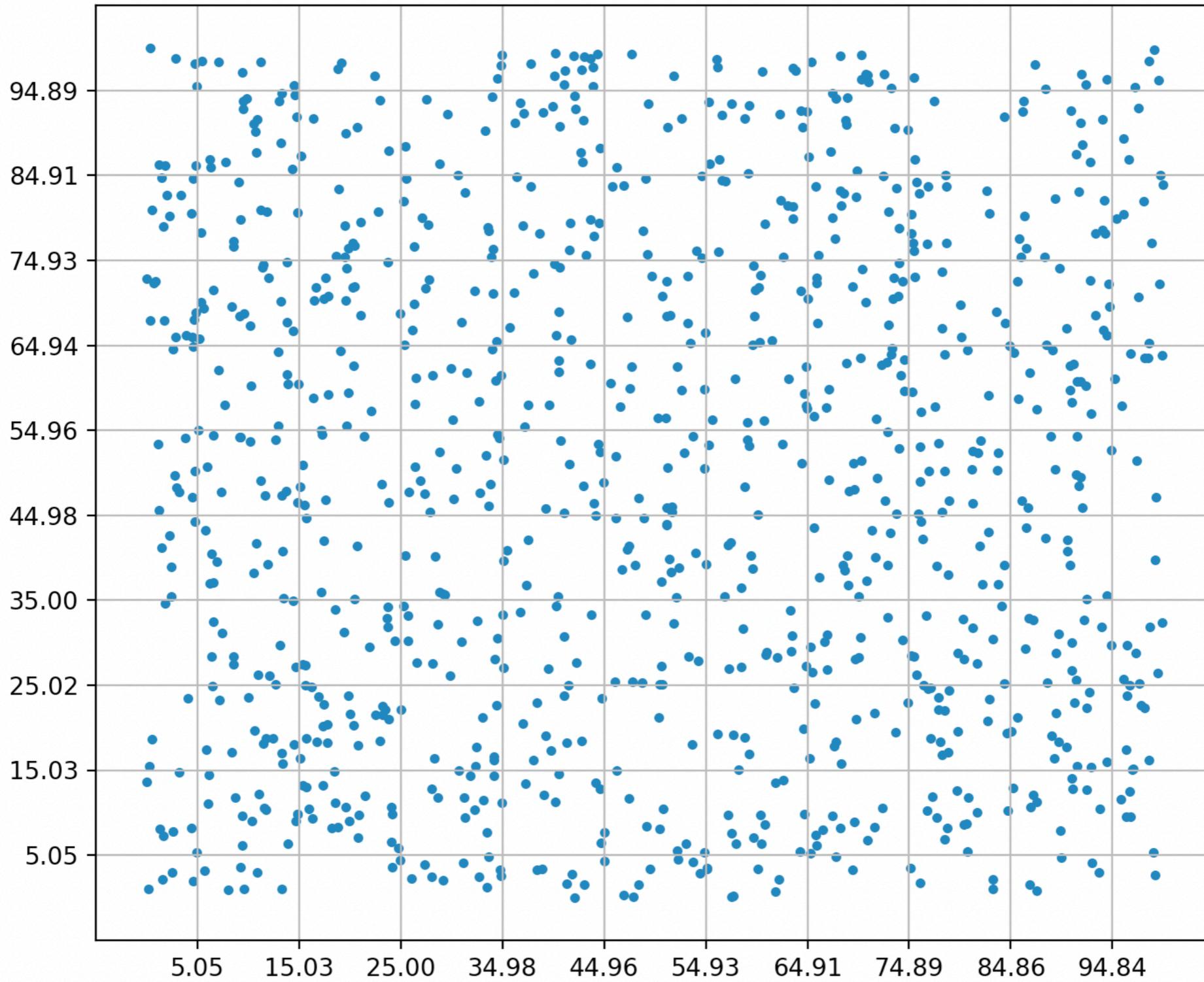


完成測試程式test1

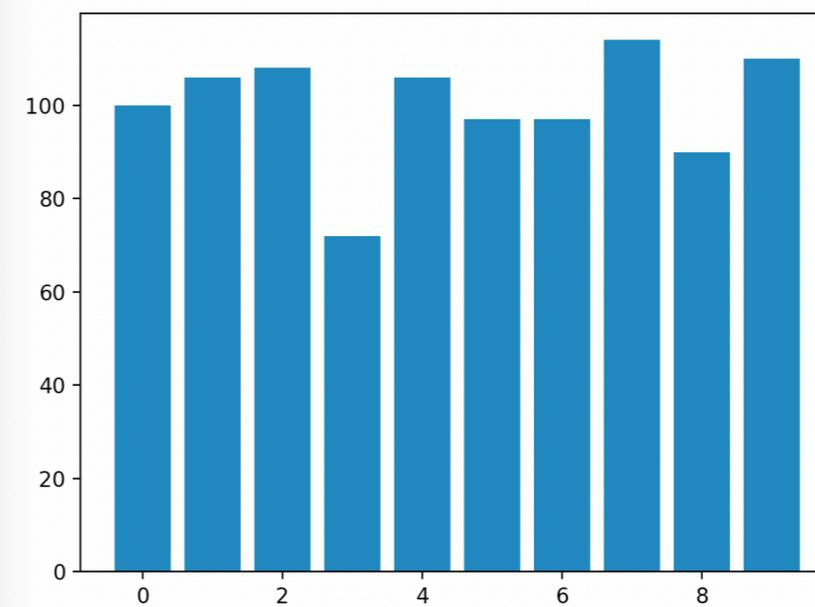
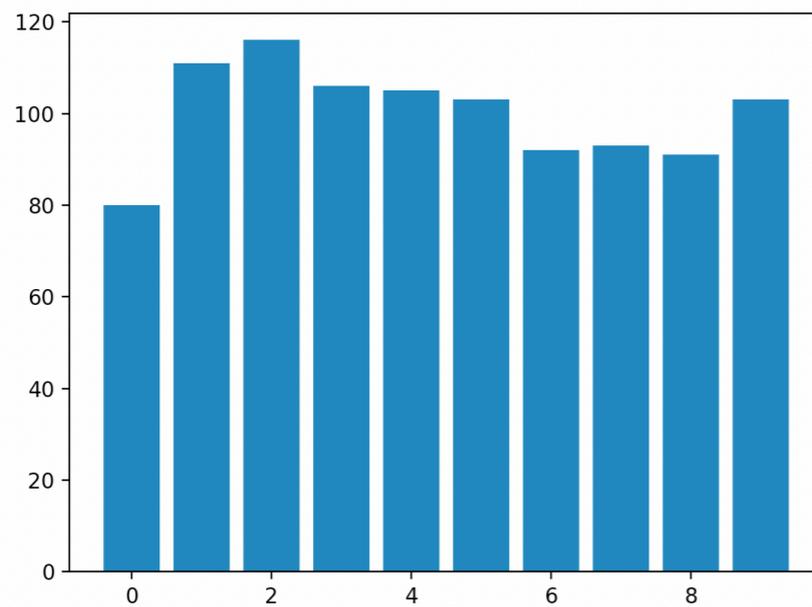
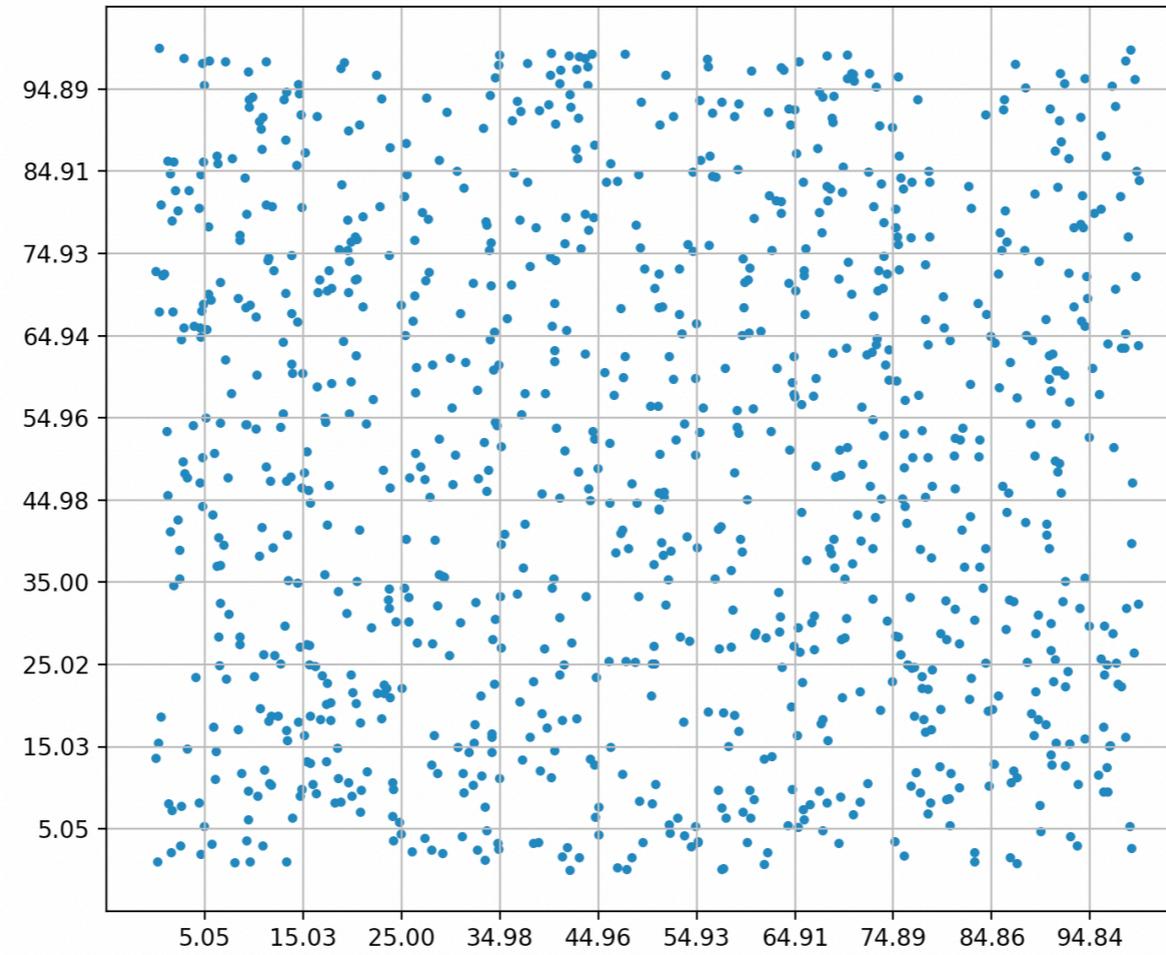
```
from shape2D2 import *
N = 1000
bin_size = 10
x_list, y_list = produce_sig(N)
x_bin = bin_points_list(x_list, bin_size)
y_bin = bin_points_list(y_list, bin_size)
plot_points(x_list, y_list, x_bin, y_bin)

plt.show()
```

shape2D2.py中的函數
produce_sig
bin_points_list
plot_points



**實驗30、二維正方形資料點
在x座標軸投影的累計分布
與y座標軸投影的累計分布**



二維資料點在x軸投影的累計直方圖 二維資料點在y軸投影的累計直方圖

步驟一、my_histogram 橫座標與縱座標的 直方累計圖

完成my_histogram函數

```
def my_histogram(x,n): #3B
    bin = bin_points_list(x,n)
    x2bin = np.zeros(n)
    for i in range(len(x)):
```

計算ind，並將x2bin[ind]的值加1

```
    plt.figure()
    plt.bar(range(n),x2bin)
    return x2bin, bin
```

$$bin[ind] = \min_j \|x[i] - bin[j]\|$$

求ind使得
bin[ind]成為串列bin的n個元
素中與x[i]最靠近的值

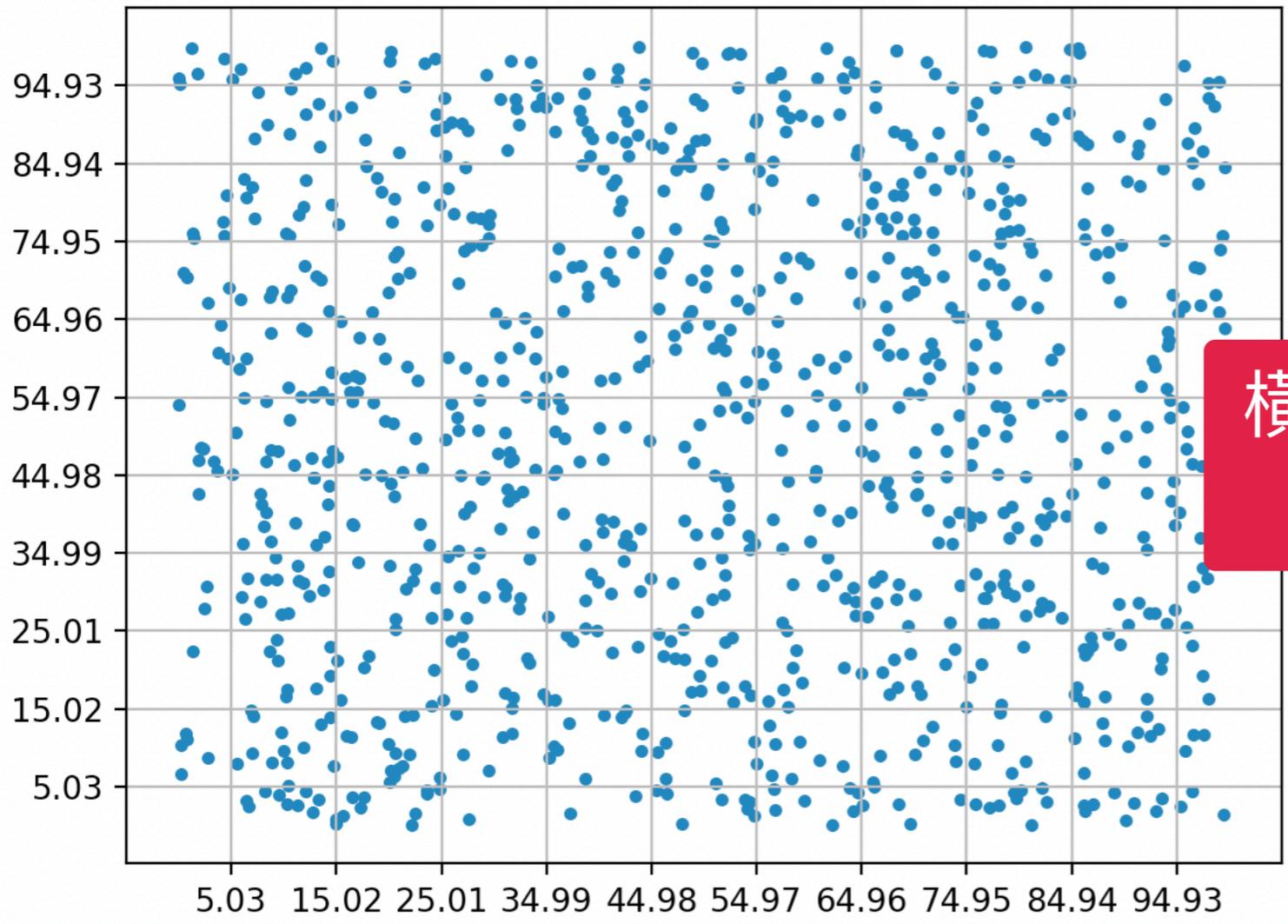
步驟二、繪製在x座標軸投影的累計分布直方圖與y座標軸投影的累計分布直方圖

完成測試程式test2

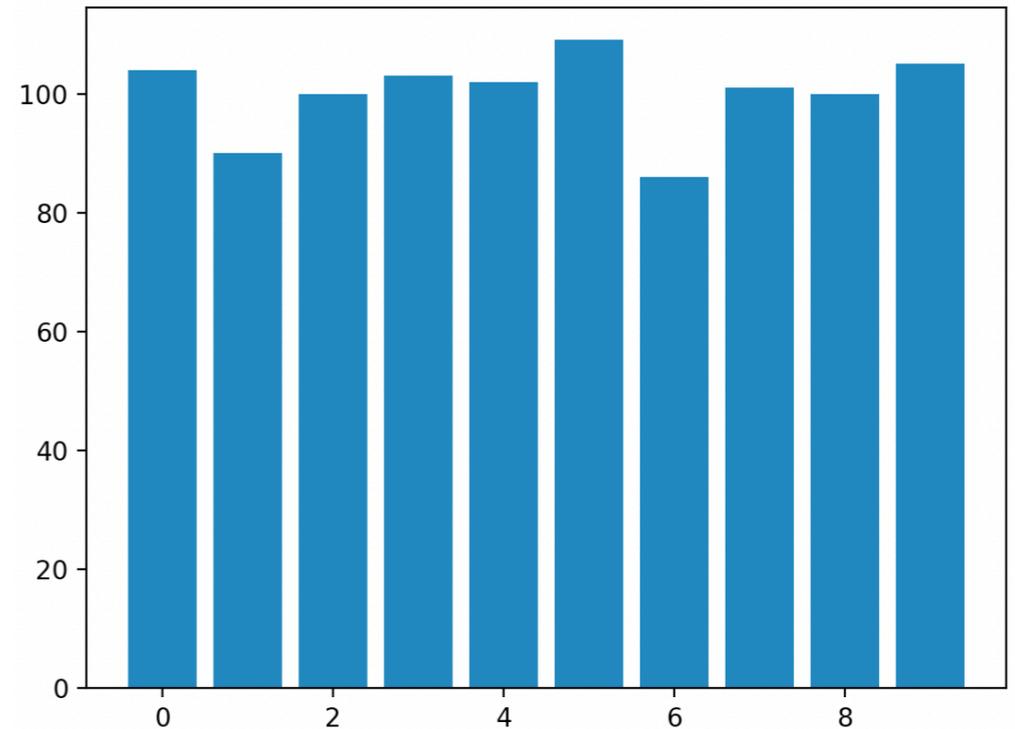
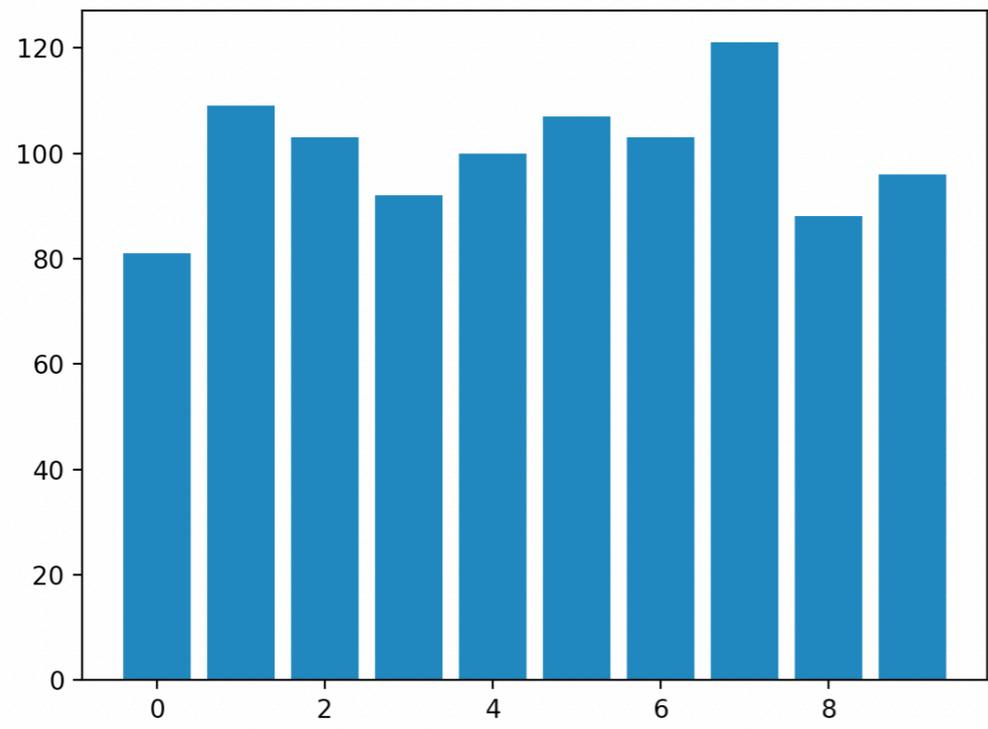
```
from shape2D2 import *  
N = 1000  
bin_size = 10  
x_list, y_list = produce_sig(N)  
bin_x = bin_points_list(x_list, bin_size)  
bin_y = bin_points_list(x_list, bin_size)  
plot_points(x_list, y_list, bin_x, bin_y)
```

繪製在x座標軸投影的累計分布直方圖與y座標軸投影的累計分布直方圖

```
plt.show()
```



橫坐標與縱座標的直方累
計圖



步驟四、

計算 `my_joint_histogram`

聯合累計矩陣

決定xind，使得x_bin[xind]代表x_bin中與x[i]最近的格點
決定yind，使得y_bin[yind]代表y_bin與y[i]最近的格點

```
def my_joint_histogram(x,y,n,x_bin,y_bin): #3A
    xybin = np.zeros((n,n))
    for i in range(len(x)):
```

計算xind與yind，並將xybin[xind,yind] 的值加1

```
return xybin
```

xybin是nxn的陣列，它的內容
代表落在相對應方格的資料點
總數

步驟五、

計算在 x 座標軸投影的累計分布

計算在 y 座標軸投影的累計分布

計算聯合累計矩陣計算

完成測試程式test3

```
from shape2D2 import *
N = 1000
bin_size = 10
x_list, y_list = produce_sig(N)
bin_x = bin_points_list(x_list, bin_size)
bin_y = bin_points_list(x_list, bin_size)
plot_points(x_list, y_list, bin_x, bin_y)
x2bin =
y2bin =
joint =
print(joint)
plt.show()
```

計算在x座標軸投影的累計分布，計算在y座標軸投影的累計分布，計算
聯合累計矩陣

**實驗31A、
計算正方形資料點的
Cross Entropy**

$x2bin[i]/N$ 代表資料點在x軸投影落在第*i*個子區間的機率

$y2bin[j]/N$ 代表資料點在y軸投影落在第*j*個子區間的機率

$$\text{令 } p[i,j] = x2bin[i]/N * y2bin[j]/N$$

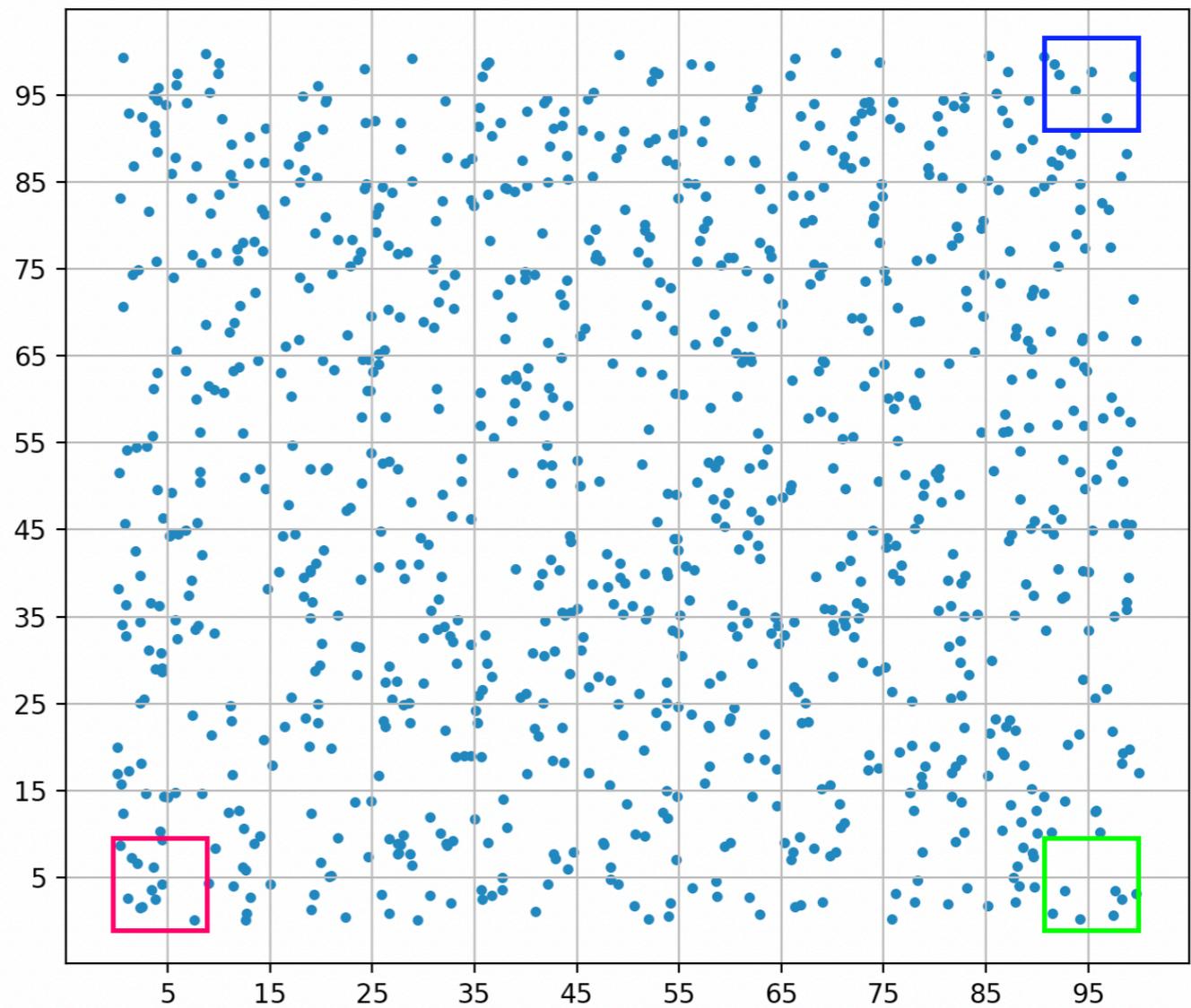
令 q 為 $joint/N$ ， $q[i,j]$ 代表資料點在x軸第*i*個子區間，y軸第*j*個子區間的聯合機率

以 $D = \sum_{i=1}^n \sum_{j=1}^n D_{ij}$ 代表交錯亂度，其中

$$D_{ij} = \begin{cases} 0 & \text{if } q[i,j] < 10^{-6} \\ q[i,j] \log \frac{q[i,j]}{p[i,j]} & \text{otherwise} \end{cases}$$

```
[ 14. 11.  8. 17. 11.  8.  8.  8. 10. 16.]  
[12.  6. 11.  5.  8.  5. 10. 10. 13.  7.]  
[16.  4. 11.  7.  9.  9. 13. 11.  9.  8.]  
[11.  9.  9. 10.  5.  8. 10. 11. 12.  8.]  
[11.  7. 11. 18.  6.  9.  8. 10.  9. 12.]  
[11.  8. 12. 10. 14.  7. 11. 12. 12.  9.]  
[11.  7.  9. 14.  9.  7. 12. 11. 12.  8.]  
[ 6. 12.  7. 12. 12. 11. 11.  7. 12. 11.]  
[12. 16. 12. 12.  8. 10.  9. 11.  9. 12.]  
[ 7. 10.  6.  8. 13. 12. 10.  7. 12.  8.]
```

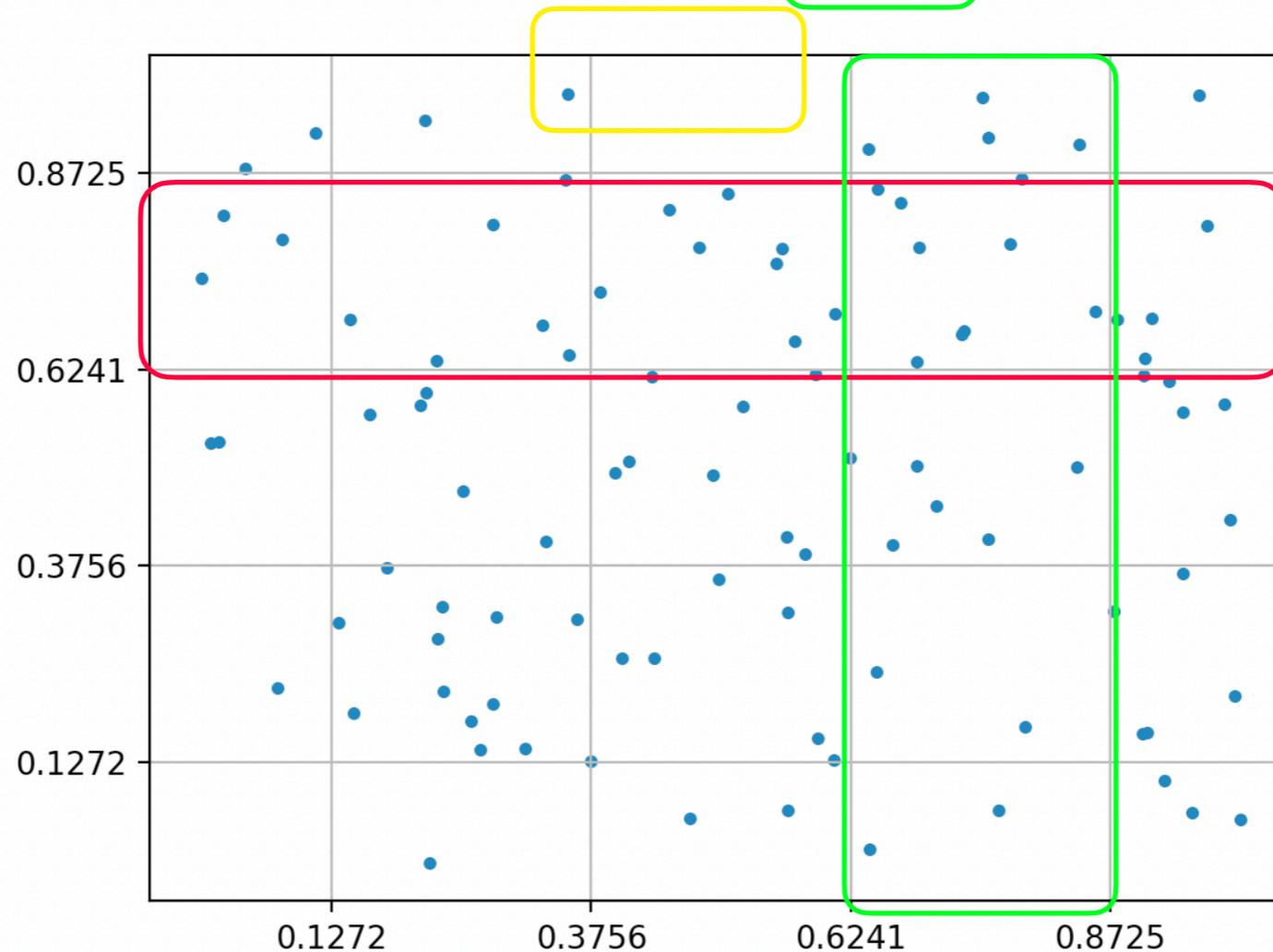
joint_histogram
聯合累計圖



將聯合累
計除以N
就是
聯合
機率

	[8	27	25	22	18]	
[2	3	0	6	1]
[3	5	10	7	5]
[2	6	7	5	4]
[1	12	6	2	5]
[0	1	2	2	3]
						[
						12
						30
						24
						25
						8
]

將個別累
計除以N
就是
個別
機率

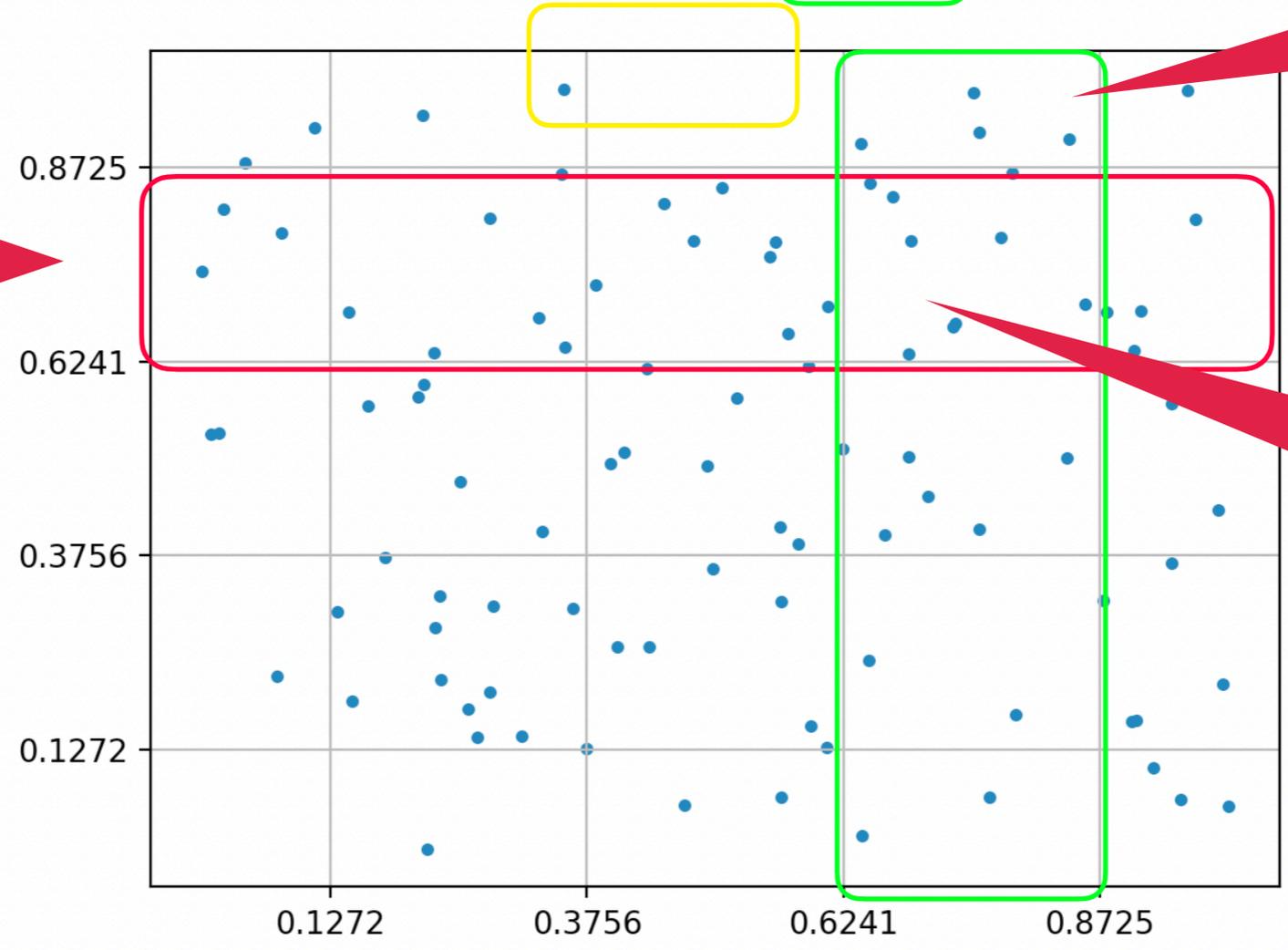


[8 27 25 22 18] x2bin

y2bin	[12	2	3	0	6	1]
	[31	3	5	10	7	5]
	[24	2	6	7	5	4]
	[25	1	12	6	2	5]
	[8	0	1	2	2	3]

個別機率
22/100

個別機率
31/100



聯合機率
7/100

計算所有 D_{ij} 與 D

$$D = \sum_{i,j} D_{ij}$$

```
def calculate_D(joint,x2bin,y2bin): # 4
    m = len(x2bin)
    n = len(y2bin)
    N = np.sum(x2bin)
    D = 0
    for i in range(m):
        for j in range(n):
            p = 
            q = 
            if abs(q) > 10**-6:
                D += 
    return D
```

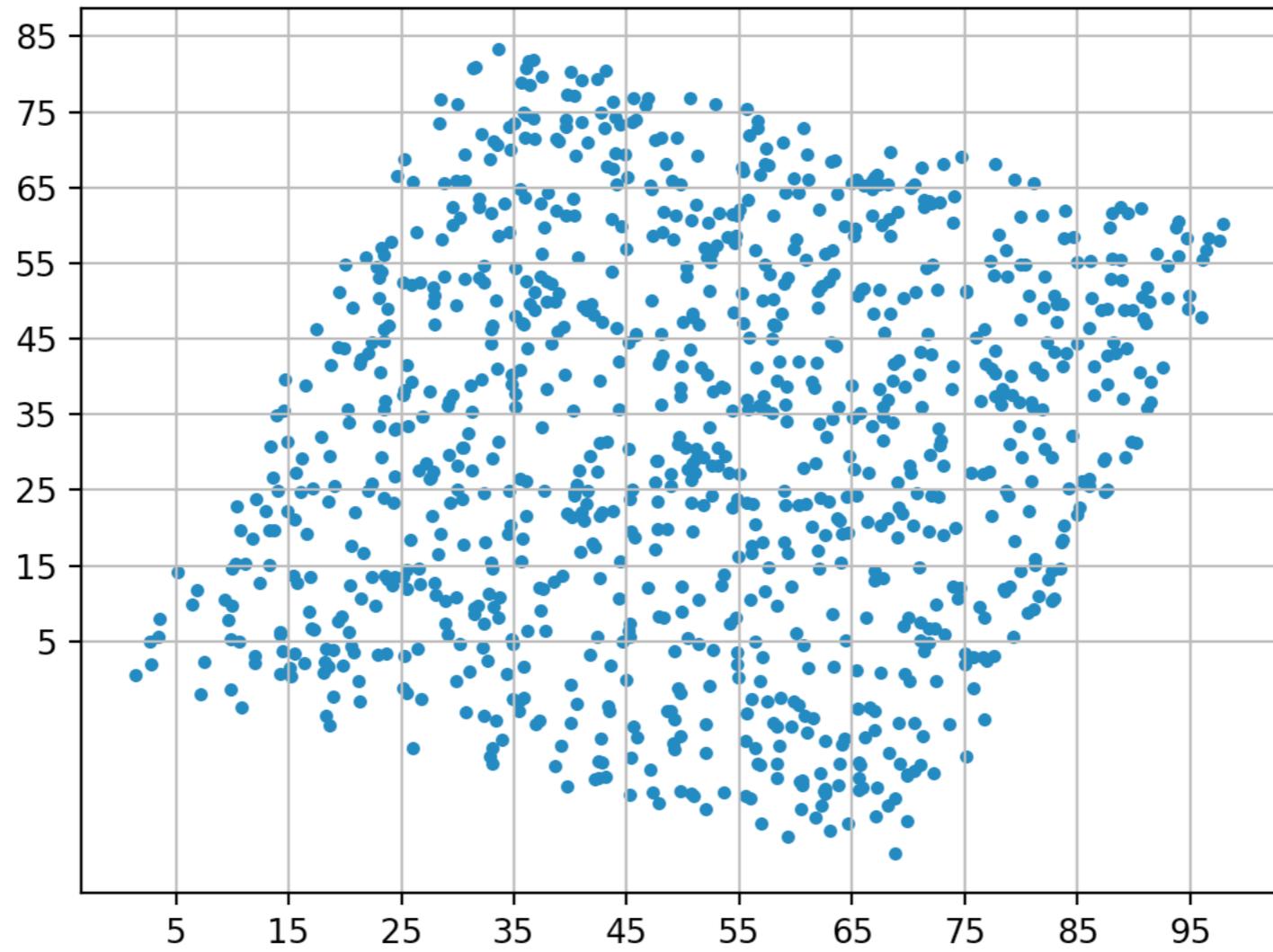
$$D_{ij} = q[i,j] \log \frac{q[i,j]}{p[i,j]}$$

**將N設為20000，n設為
20完成測試程式
計算Cross Entropy**

```
from shape2D2 import *
N = 2000
bin_size = 20
x_list, y_list = produce_sig(N)
x_bin = bin_points_list(x_list, bin_size)
y_bin = bin_points_list(y_list, bin_size)
plot_points(x_list, y_list, x_bin, y_bin)
x2bin, x_bin = my_histogram(x_list, bin_size)
y2bin, y_bin = my_histogram(y_list, bin_size)
joint =
my_joint_histogram(x_list, y_list, bin_size, x_bin, y_bin)
D =                 
print('cross entropy:', D)
plt.show()
```

**實驗31B、
計算旋轉資料的Cross
Entropy**

步驟一、將資料旋轉



```
def A_sig(A,x_list,y_list): # 2
    N = len(x_list)
    D = np.zeros((N,2))
    for i in range(N):
        D[i,0] = x_list[i]
        D[i,1] = y_list[i]
    DA = A @ np.transpose(D)
    return np.transpose(DA)
```

步驟二、完成測試程式 計算Cross Entropy

```
from shape2D2 import *
N = 2000
bin_size = 20
x_list, y_list = produce_sig(N)
A = np.matrix([[0.7, 0.31], [-0.25, 0.85]])
DA = A_sig(A, x_list, y_list)
x_list = 
y_list = 
x_bin = bin_points_list(x_list, bin_size)
y_bin = bin_points_list(y_list, bin_size)
plot_points(x_list, y_list, x_bin, y_bin)
x2bin, x_bin = my_histogram(x_list, bin_size)
y2bin, y_bin = my_histogram(y_list, bin_size)
joint =
my_joint_histogram(x_list, y_list, bin_size, x_bin, y_bin)
D = calculate_D(joint, x2bin, y2bin)
print('cross entropy:', D)
plt.show()
```

**實驗31C、將N設為2000，
n設為20，計算菱形資料的
Cross Entropy**

```
from shape2D2 import *
N = 2000
bin_size = 20
x_list, y_list = produce_sig(N)
A = np.matrix([[1/np.sqrt(2), 1/np.sqrt(2)], [1/
np.sqrt(2), -1/np.sqrt(2)]])
DA = A_sig(A,x_list,y_list)
x_list = list(DA[:,0])
y_list = list(DA[:,1])
x_bin = bin_points_list(x_list,bin_size)
y_bin = bin_points_list(y_list,bin_size)
plot_points(x_list,y_list,x_bin, y_bin)
x2bin, x_bin = my_histogram(x_list,bin_size)
y2bin, y_bin = my_histogram(y_list,bin_size)
joint =
my_joint_histogram(x_list,y_list,bin_size,x_bin,y_bin)
D = calculate_D(joint,x2bin,y2bin)
print('cross entropy:', D)
plt.show()
```