1.

資料夾中的circle1,2,3 / eclipse1,2,3是自己手畫的圓 / 橢圓。

2.

(a)我發現用normal equation跟numpy.linalg.lstsq解出來的x是一樣的，然後draw\_circle()也只有用到x，沒有用到lstsq回傳的residual或是rank，所以產生出來的fitted result一樣，皆如下圖所示。

(我在main()裡面實作normal equation)

一張含有 運動 的圖片

自動產生的描述

(b)

Numpy.linalg.lstsq:可以從其中一個回傳值residual(residuals=|b-Ax|平方過後的總和)的大小知道該圖跟圓的相似程度，residual越大代表越不像，越小代表越像。

Normal equation:這個方法只能得知圓心座標c1,c2與常數c3，不過這些資訊足以得知半徑長度，進而畫出預測的圓，再跟原圖做比較。

(c)我自己example的rank=3。要畫出rank<3的圖只需要pixel不超過3點即可，所以我在圖上點2點(rank=2)就完成了。(下圖的檔名為rank=2.png)



3.

仿造circle的方式，橢圓需要5個變數，分別是c1~c5；A、C的公式就直接將一般式c1 x2 + c2 xy + c3 y2 + c4 x + c5 y = 1套入即可。

4.

[x[i\*19] for i in range(len(x)//19)]

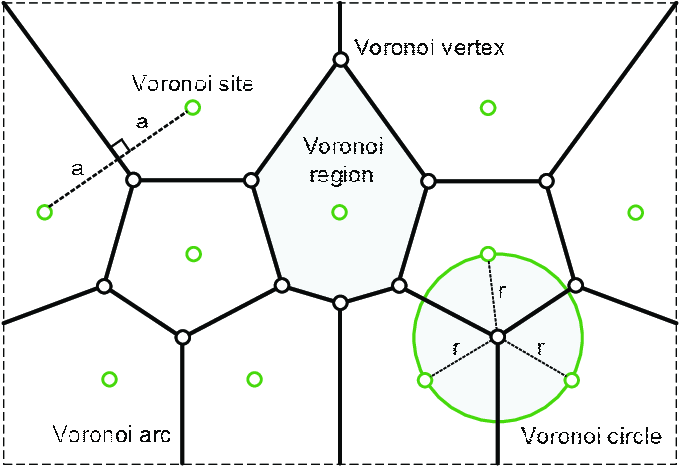
我在所有的點中，從第0個點開始，每隔19點就挑一個點出來當作sample data，因此會挑出[0],[19],[38],…，我發現19這個數字可以讓每張圖有較高的平均分數，17~20之間的得分都差不多。時間複雜度為O(n/19)=O(n)，n=所有的點。

sp = points if len(points)<50 else dataSampling(points)

但如果少於50個點的話，就直接讓sp=points，不需要經過sampling，多於50個點才需要sample。

5.

圖片來源: <https://www.researchgate.net/figure/Voronoi-diagram-in-a-plane_fig1_325582898>



Vonoroi diagram就是如上的圖，其中黑色邊上的任何一點到兩相鄰區塊內點的距離相同(如左上角a,a線段所示)；而由三條黑線交會所產生的交點，到相鄰三格內點的距離相同(如右下角半徑為r的圓)