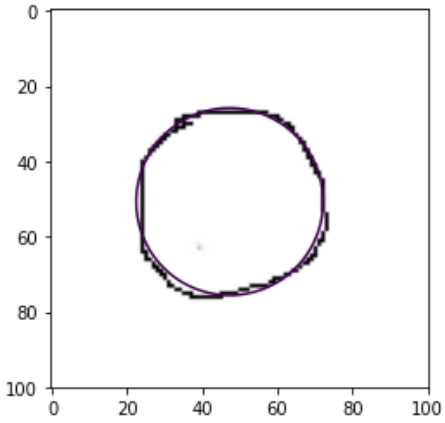
1.

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)



(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 $c1\ x^2 + c2\ xy + c3\ y^2 + c4\ x + c5\ y = 1$ 套入即可。

4.

```
[x[i*19] for i in range(len(x)//19)]
```

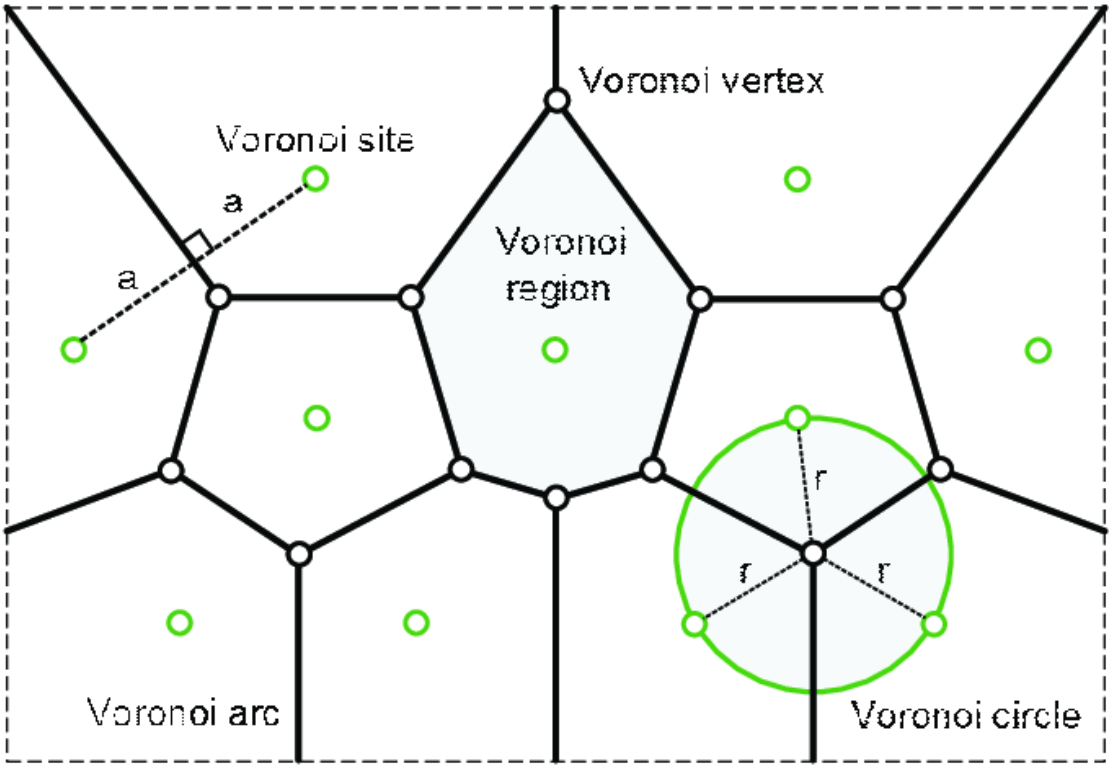
我在所有的點中，從第 0 個點開始，每隔 19 點就挑一個點出來當作 sample data，因此會挑出[0],[19],[38],…，我發現 19 這個數字可以讓每張圖有較高的平均分數，17~20 之間的得分都差不多。時間複雜度為 O(n/19)=O(n)，n=所有的點。

```
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)