PYTHON機器學習入門

UNIT 7：CLUSTERING

授課教師：江尚瑀

APPLICATIONS OF CLUSTERING

- 群集分析(**Clustering**)，又稱作分群分析、集群分析、聚類分析，是典型的非監督式學習(**Unsupervised Learning**)，適合不知如何分群的資料集
  - **Customer Segmentation**
  - **Document Clustering**
  - **Recommendation Engines**
  - **Image Segmentation**

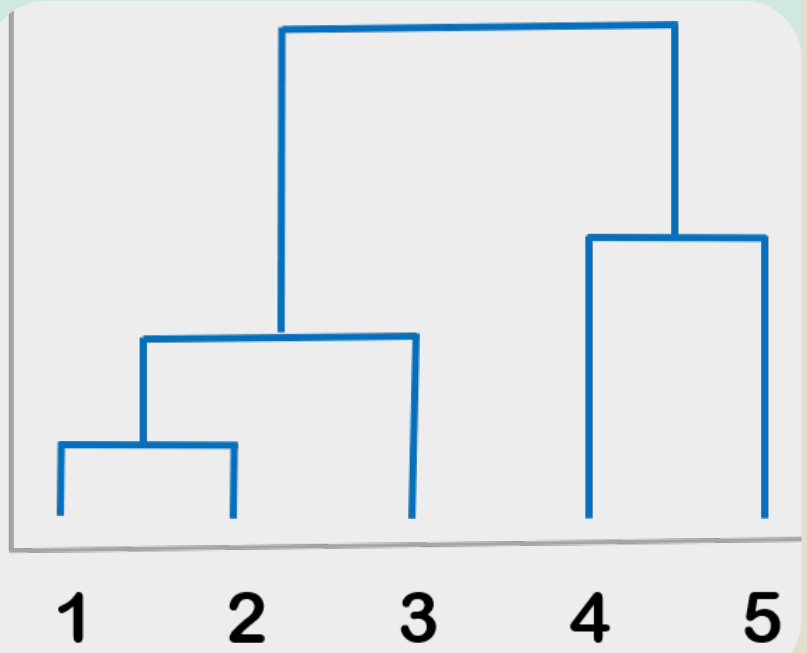
CLUSTERING

- 主要目地是分析資料集中的各資料點，找出資料點彼此間的相似程度，並依此將同性質之資料分為同個群集**(clusters)**。
  - 找出各群的代表點。代表點是群集中的中心點**(centroids)**或是原型**(prototypes)**。

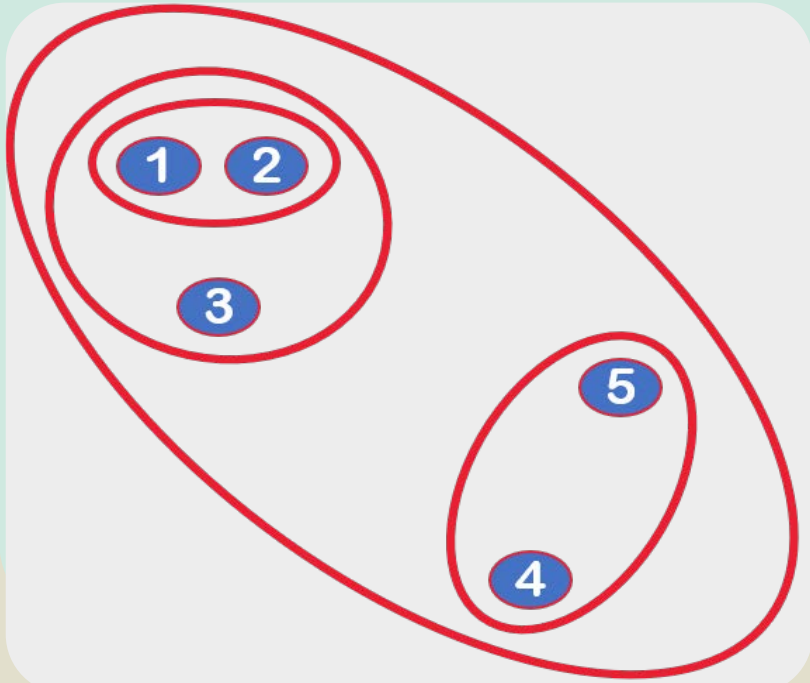
- 群集分析應遵守唯一原則：組內資料同值，組間資料異質
  - 每一資料點距離其所屬群集之中心點距離最小，距離其他群集之中心點距離較大
  - 每個資料點只能隸屬於一個群集

CLUSTERING CATEGORIES

階層式分群分析
(Hierarchical Clustering)



切割式分群分析
(Partitional Clustering)



CLUSTERING CATEGORIES

階層式分群分析
(Hierarchical Clustering)

聚合式
（Agglomerative）

分裂式（Divisive）

切割式分群分析
(Partitional Clustering)

k - 平均演算法
(*k*-means)

階層式分群分析

- 聚合式是透過一種階層架構的方式，將資料層層反覆地進行分裂或聚合，以產生最後的樹狀結構，常見的方式有兩種：

聚合式(Agglomerative, AGNES)

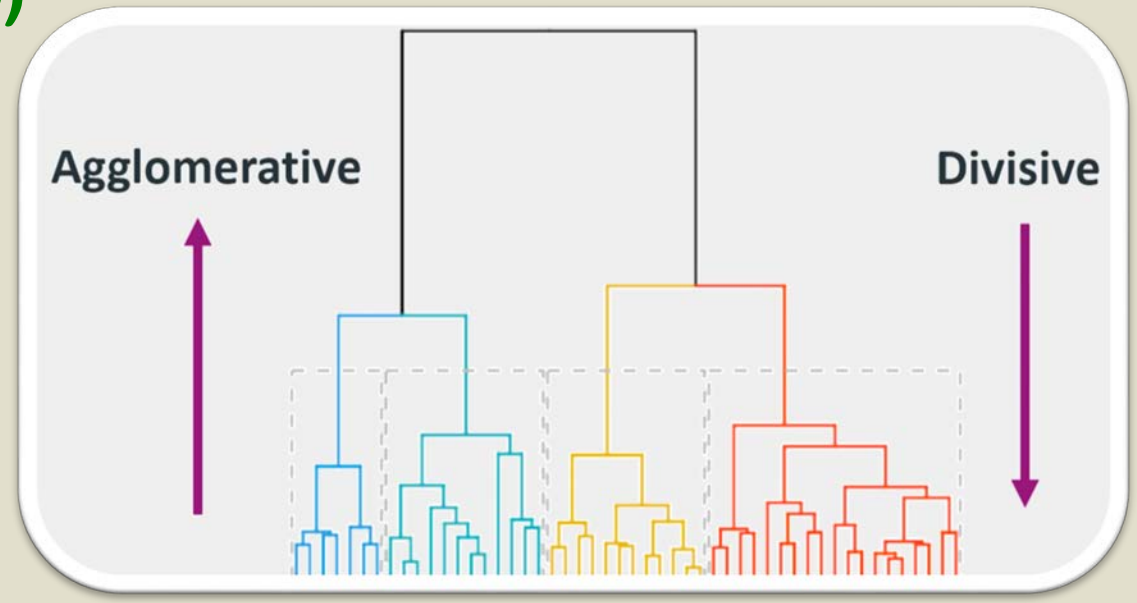
由樹狀結構的**底部**開始

由下往上將數據資料或群集逐次合併

分裂式(Divisive, DIANA)

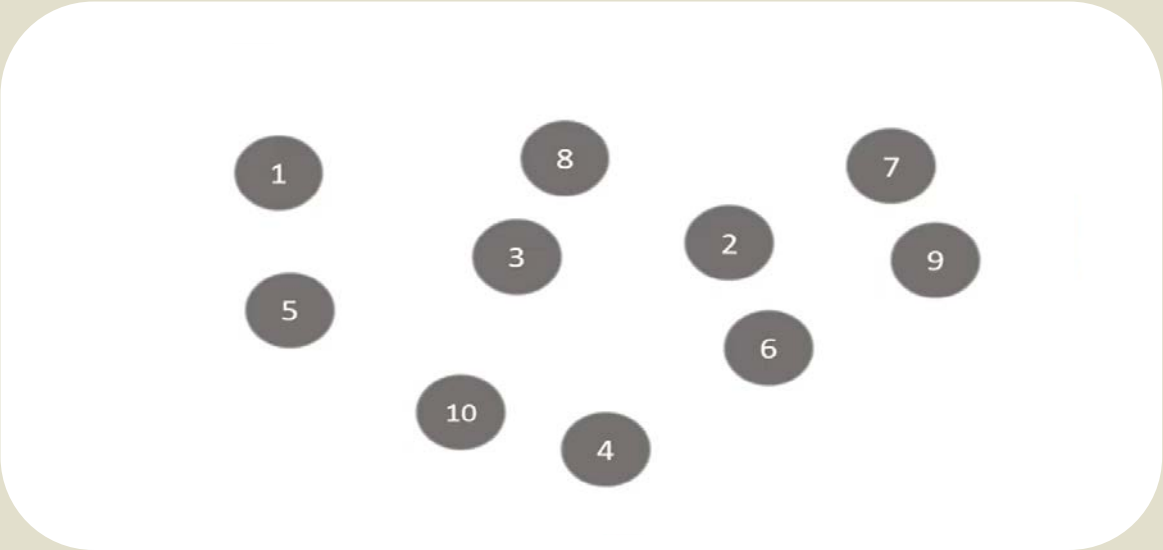
由樹狀結構的**頂端**開始

由上往下將樹聚群集逐次分裂。



聚合式分群法 AGGLOMERATIVE CLUSTERING

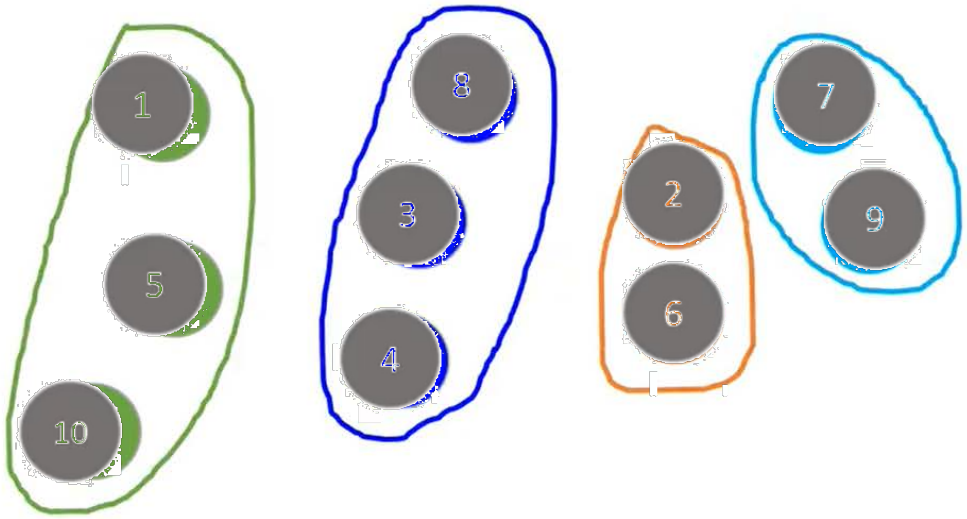
- 由樹狀結構的底部開始層層聚合，一開始將每一筆資料視為一個集群（**cluster**），假設現在擁有 n 筆資料，則初始即擁有 n 個集群：
  1. 將每筆資料視為一個群集 **Ci , i = 1 to n**
  2. 找出所有群集間，距離最接近的兩個群集 **Ci**、**Cj**
  3. 合併 **Ci**、**Cj** 成為一個新的群集
  4. 若目前的集群數多於預期群數，則反覆重複上述步驟，直到集群數降到預期群數



分裂式分群法 DIVISIVE CLUSTERING

- 由樹狀結構的頂部開始層層分裂，一開始將所有的資料視為一個集群（**cluster**），即初始的 n 筆資料點 $\{x_1,x_2,x_3,...,x_n\}$ 皆在同個集群 C 中：
  1. 找出所有的資料間，距離集群 C 中心最遠的資料點 x_i
  2. 將該筆資料自集群 C 分裂出來，形成新集群 E
  3. 繼續尋找下一個距離集群 C 中心最遠的資料點 x_j，並計算該資料點與原集群 C 之距離 **(**$d\ (C, j)$**)** 與新集群 E 距離**(**記為 $d(E, j)$**)**
  4. 若 $d(C, j) > d(E, j)$，則將該資料點 x_j 併入新集群 E 中
  5. 若目前集群數少於預期群數，則反覆上述步驟，直到集群數達到預期群數。

分裂式分群法 DIVISIVE CLUSTERING



第二回合

分群 K=4
接著各群找出較相近的距離，再分為4群

階層式群集分析優缺點

優點

- 使用樹狀結構表現分群過程，概念簡單
- 僅需資料點間的距離，即可建構分群結果

缺點

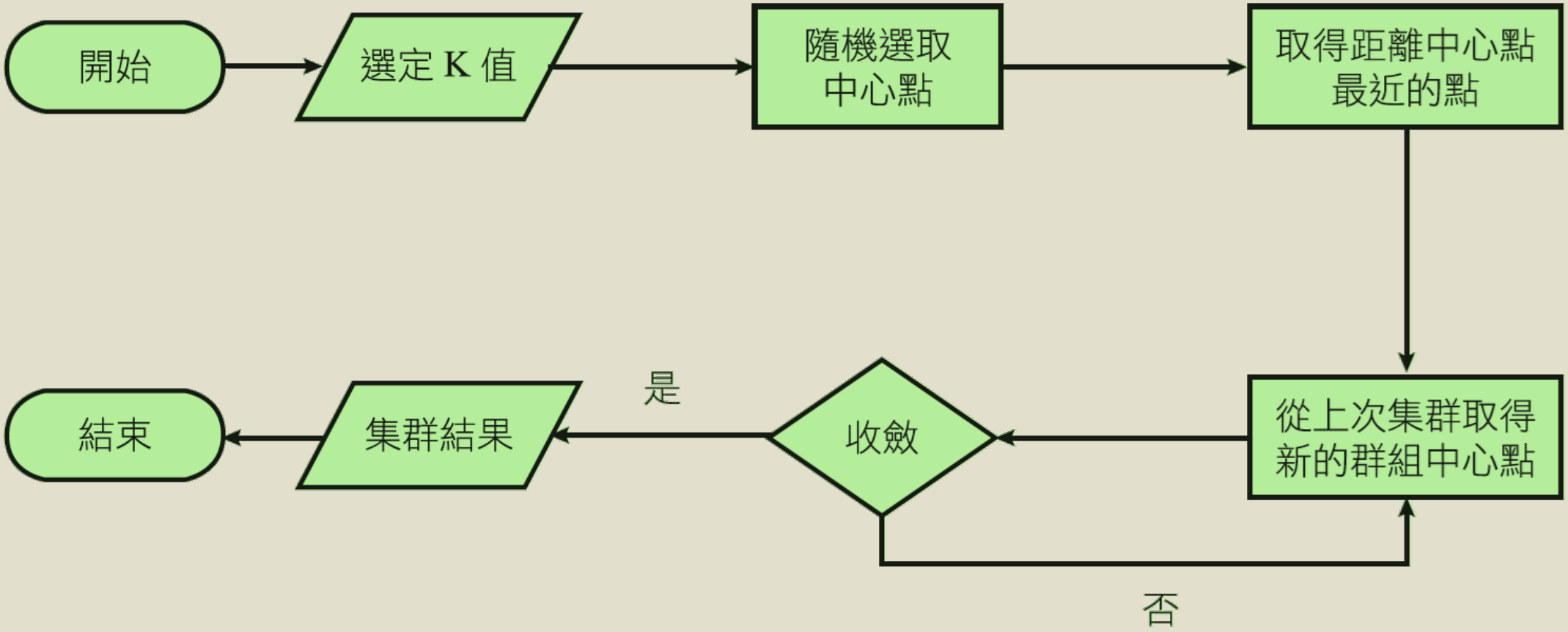
- 演算法過於簡單，一般只適用於少量資料分析，而不適合處理大量資料
- 運算速度慢

切割式分群分析

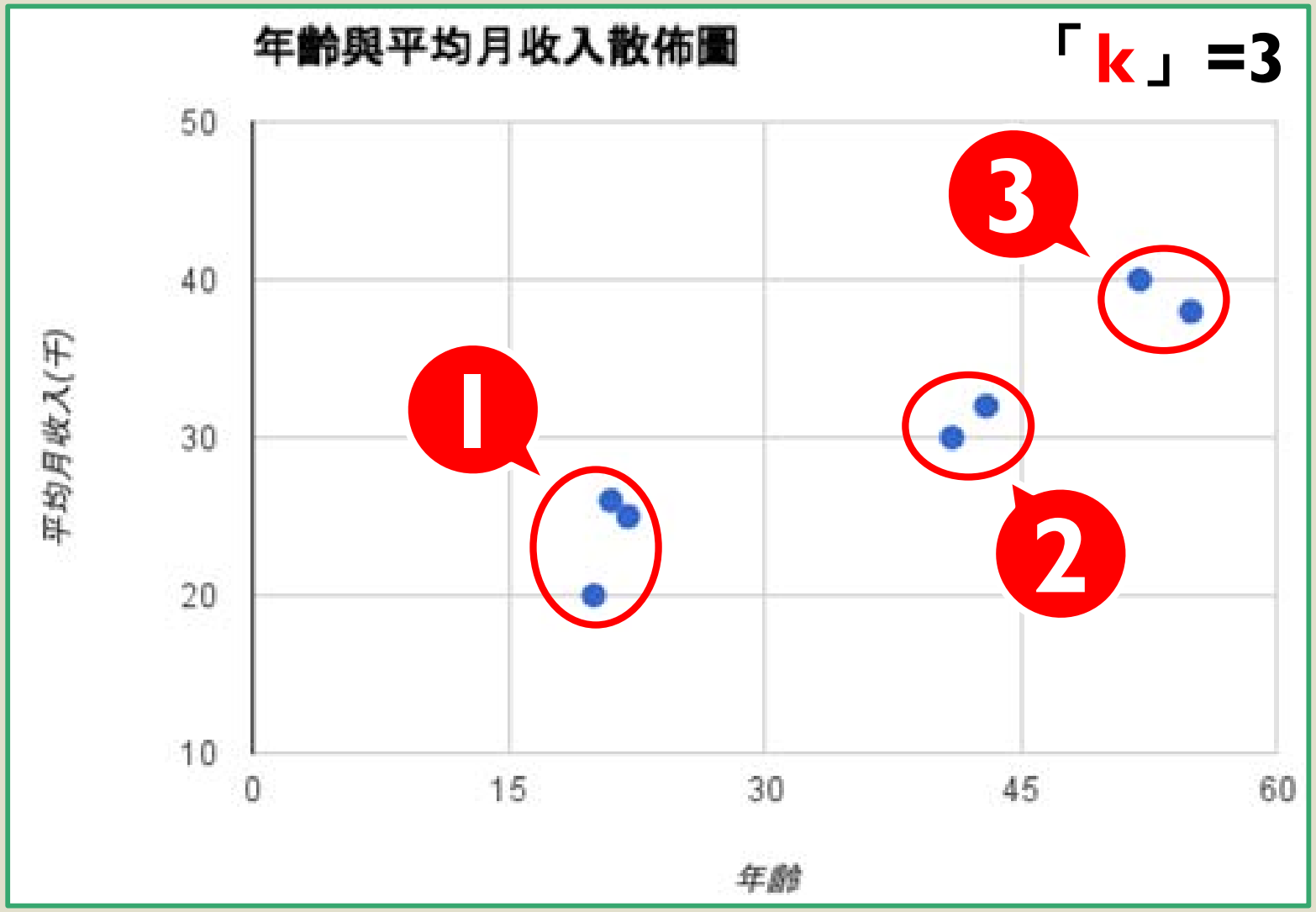
- ***k*-平均演算法（*k*-means clustering）**的目的是把 ***n*** 個資料劃分到 ***k*** 個集群中，每個資料皆可視為二維空間中的一資料點，且皆必定鄰近所屬集群之**中心點**(即該集群內資料屬性特徵的平均值) 。
- 使用***k*–means**演算法，在擁有 ***n*** 個資料點，且資料包含 ***d*** 個特徵之情境下，欲找出 ***k*** 個最佳解群集，此類問題的計算複雜度***(O)*** 非常高，計算公式為：

$$O = (n^{dk+1} \log n)$$

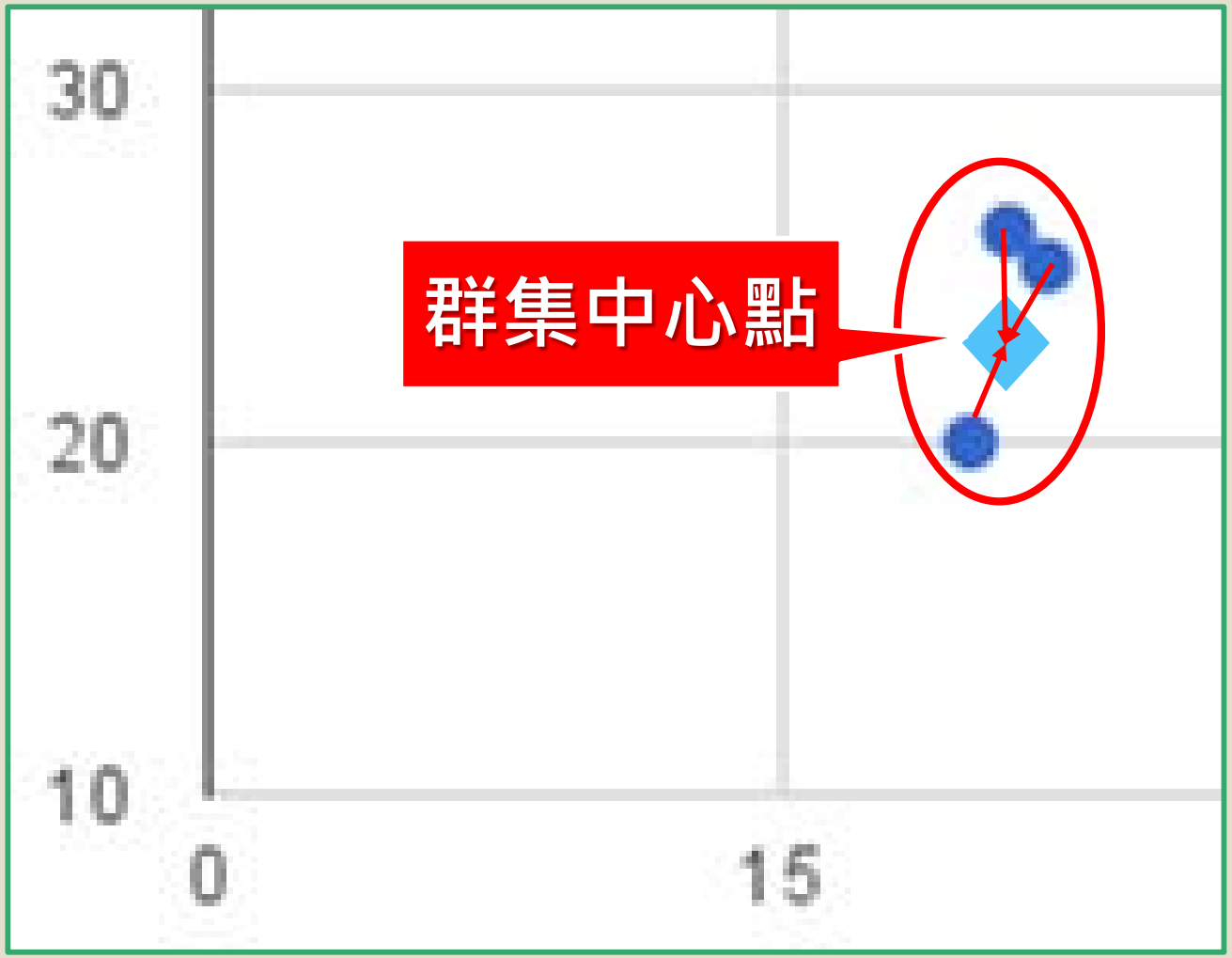
K-MEANS演算過程



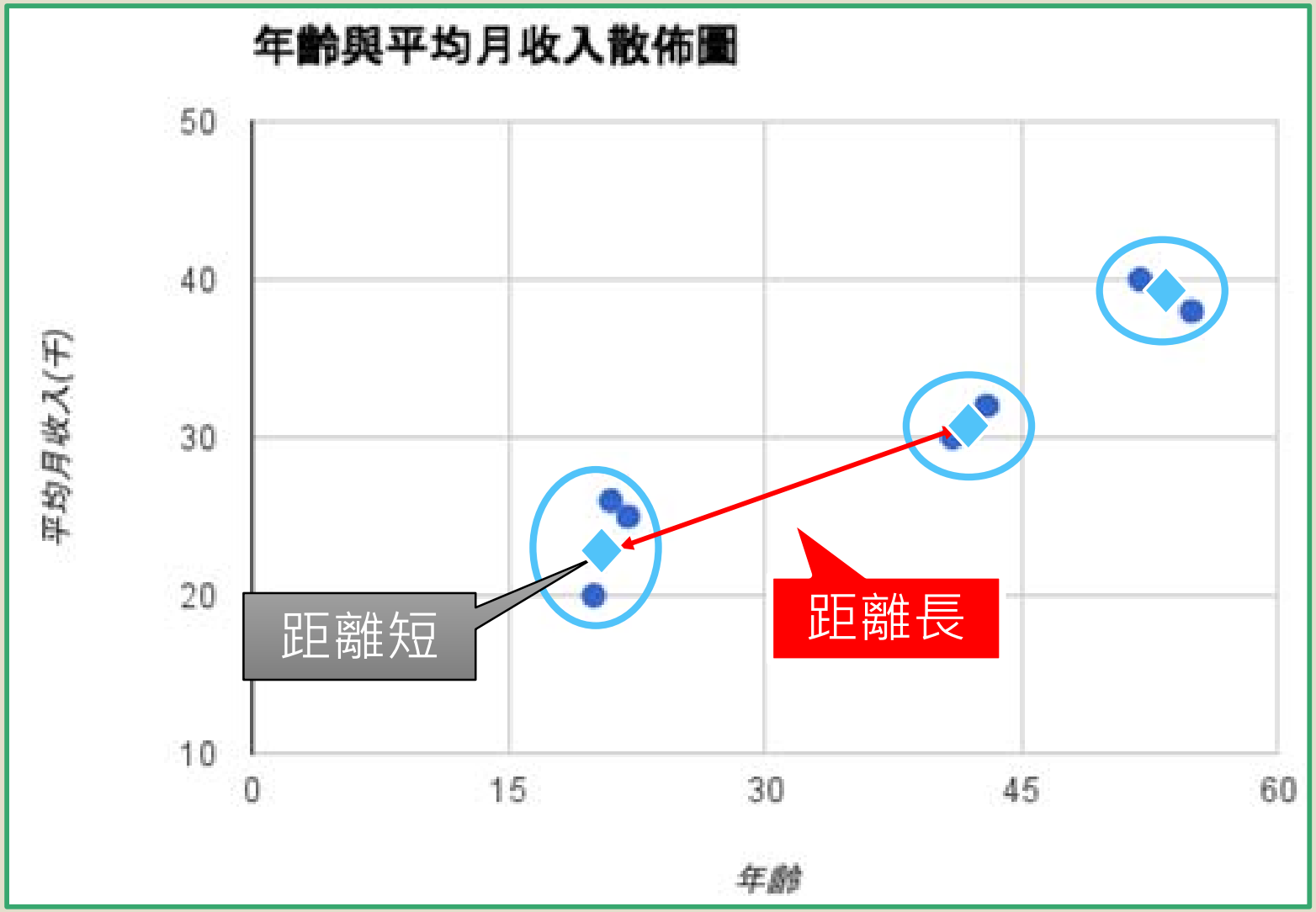
群集數K



資料平均數：群集中心點

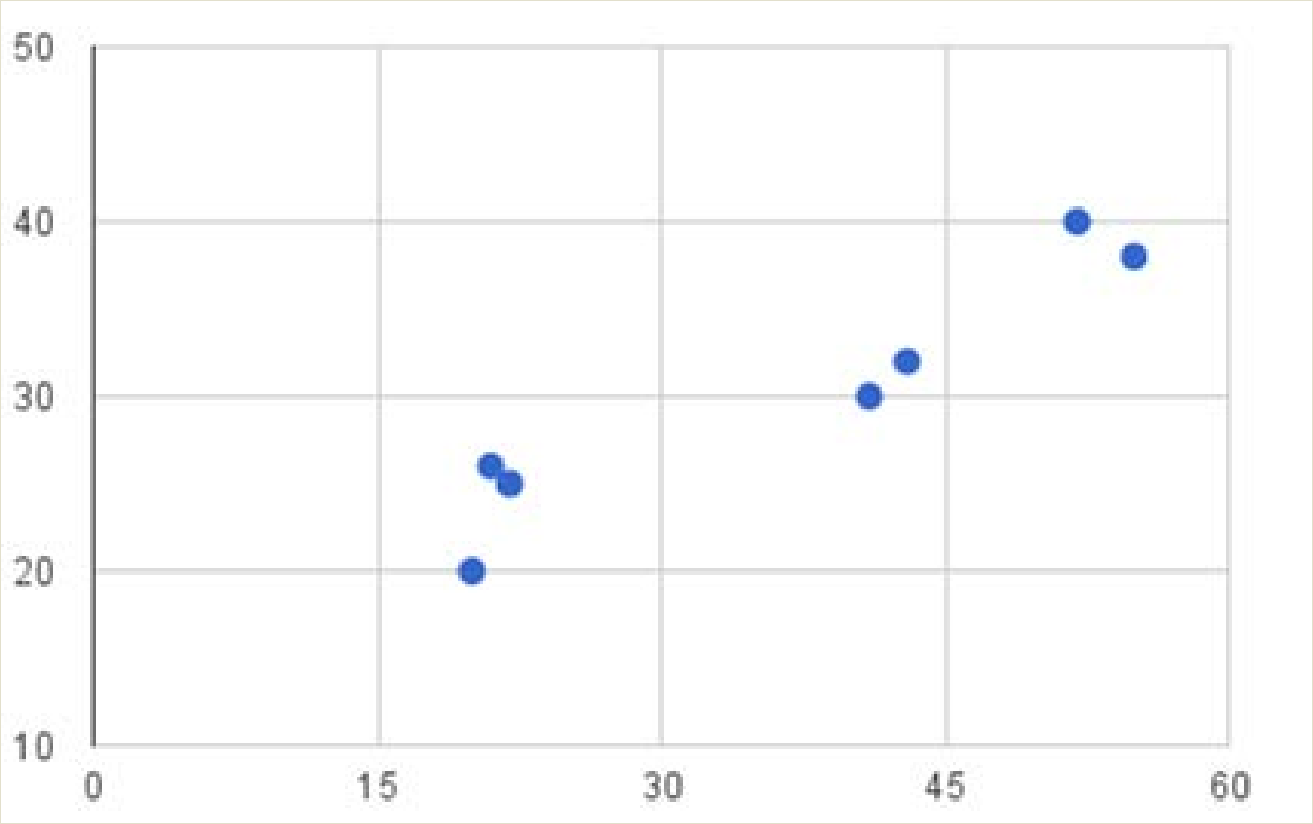


各群集與資料點間距離關係



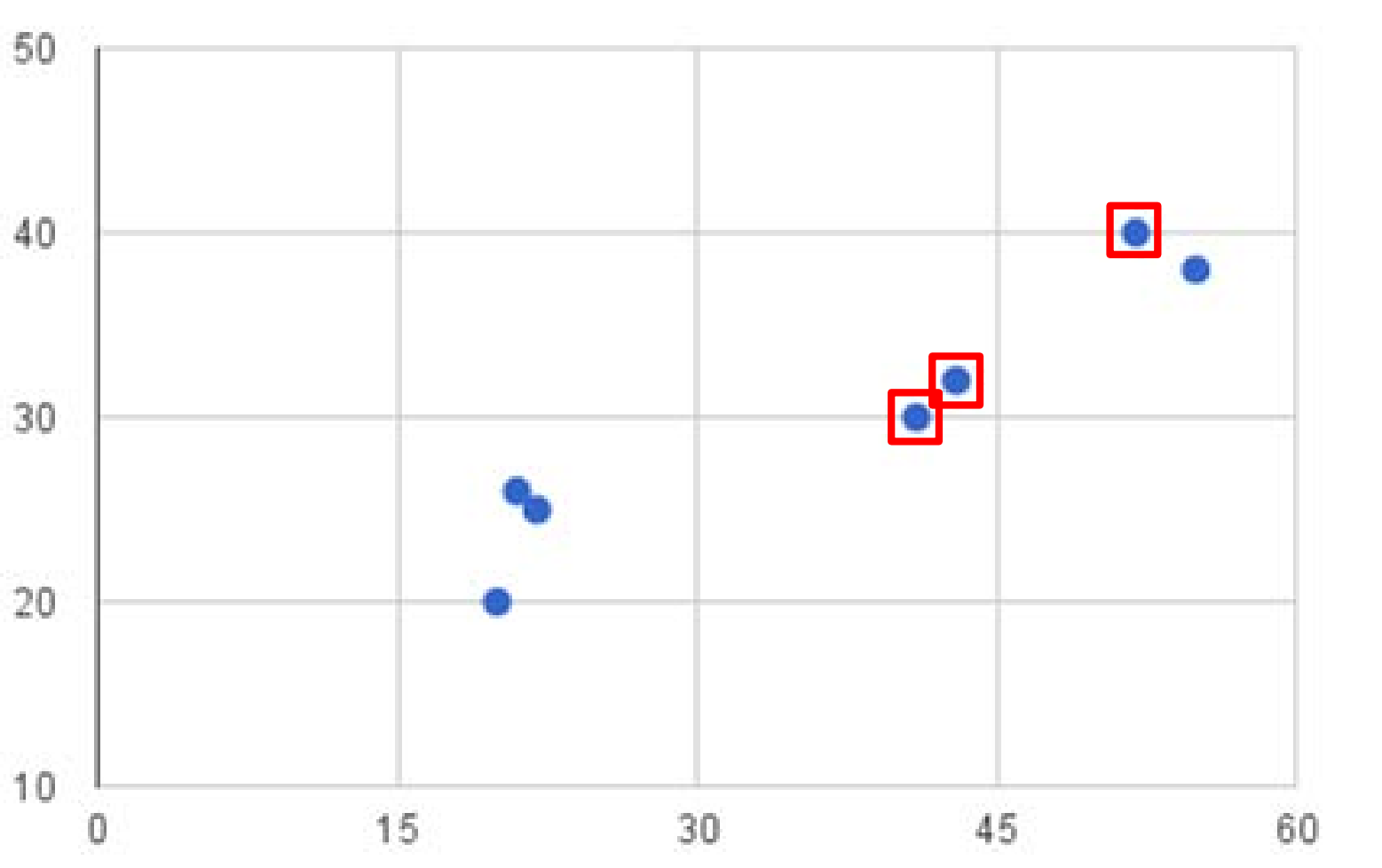
資料集散佈圖

- **X軸：年齡**
- **Y軸：月收入**
- 欲分析群集數→ ***k = 3***



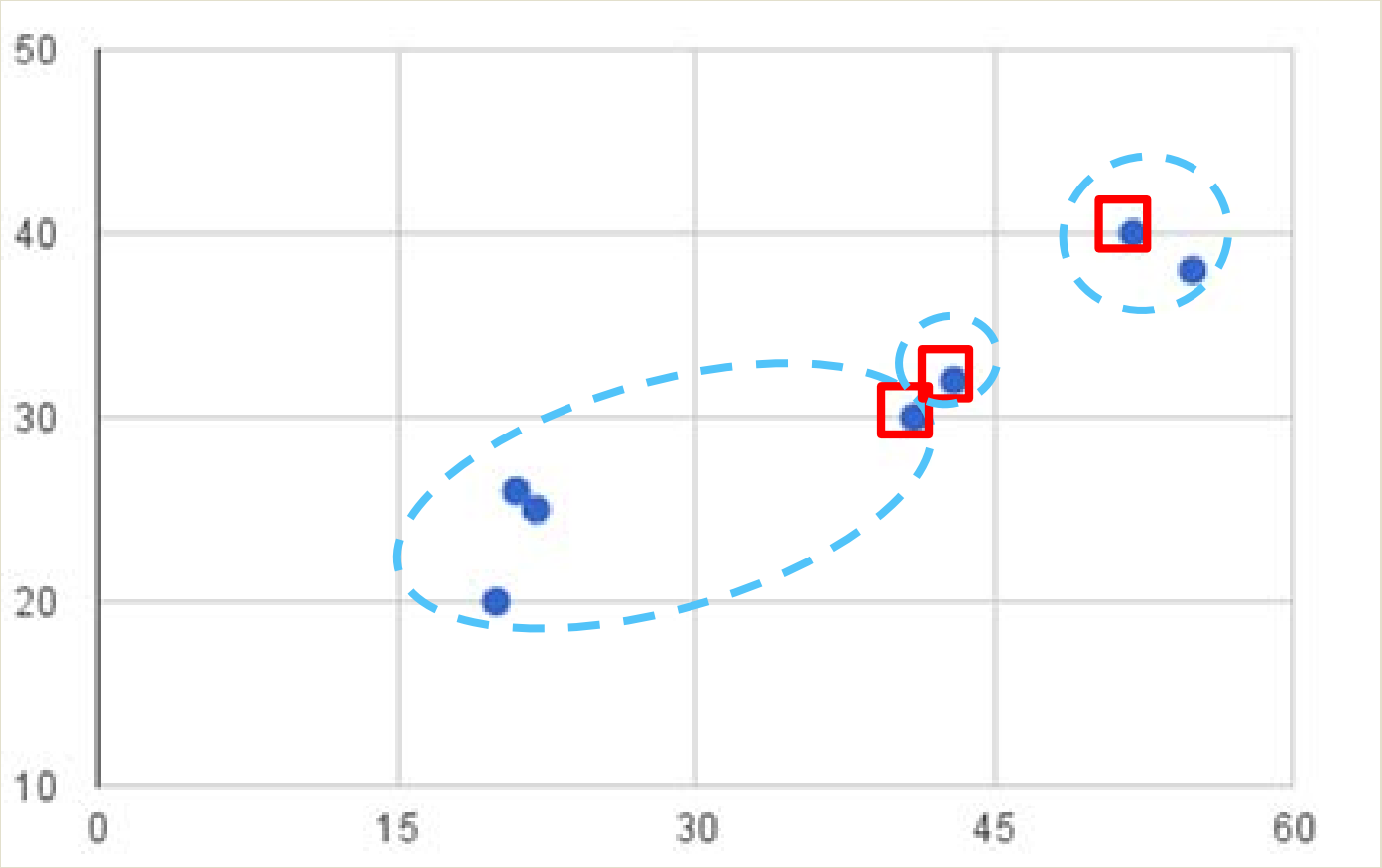
1-1. 起始設置

隨機選擇任 ***s*** 個資料點當作起始 ***k*** 群的**群集中心點**



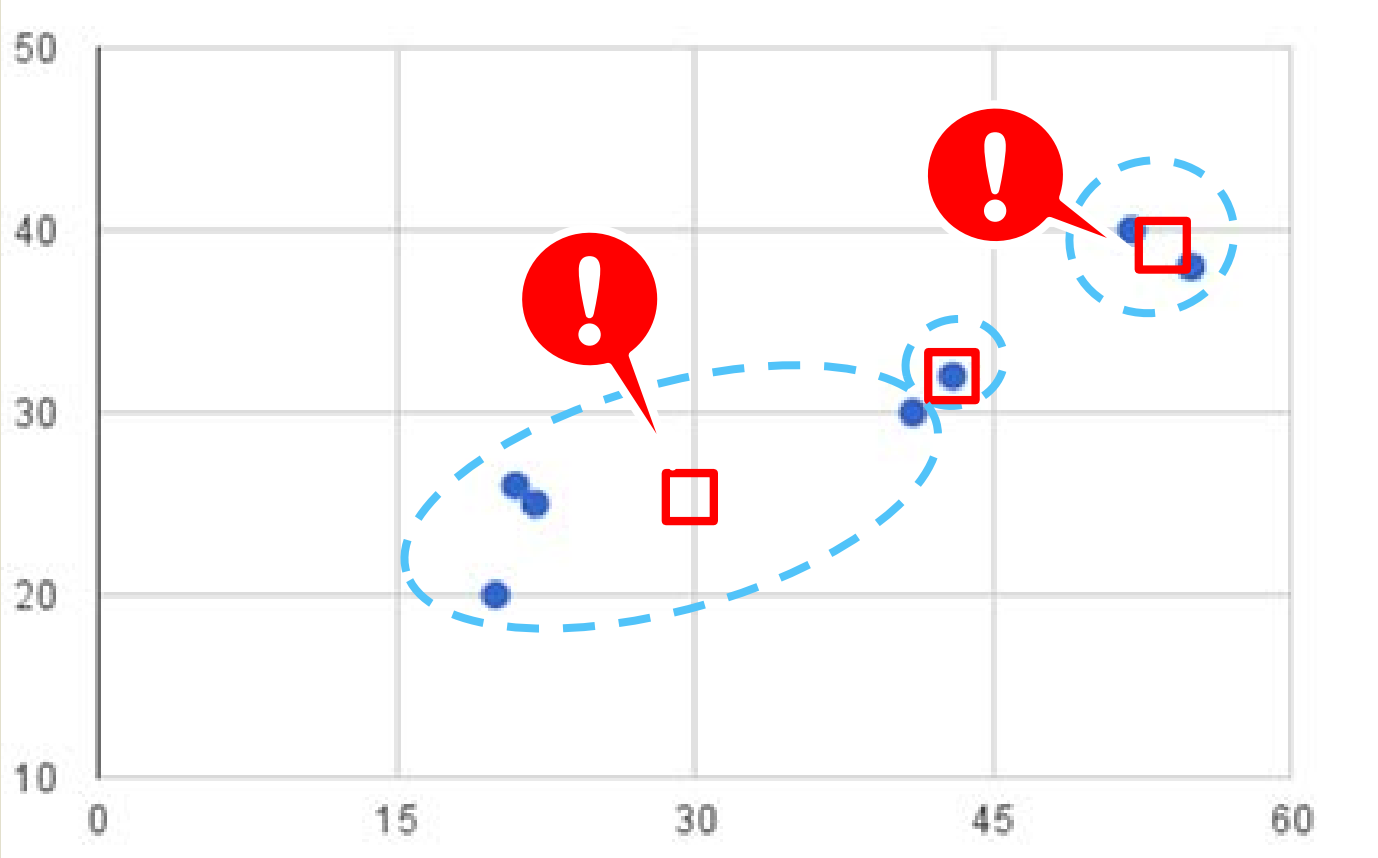
1-2. 形成群集

利用距離計算公式將資料點分別歸類到距其最近之**群集中心點**所屬之群集，形成 ***k=3*** 的群集。



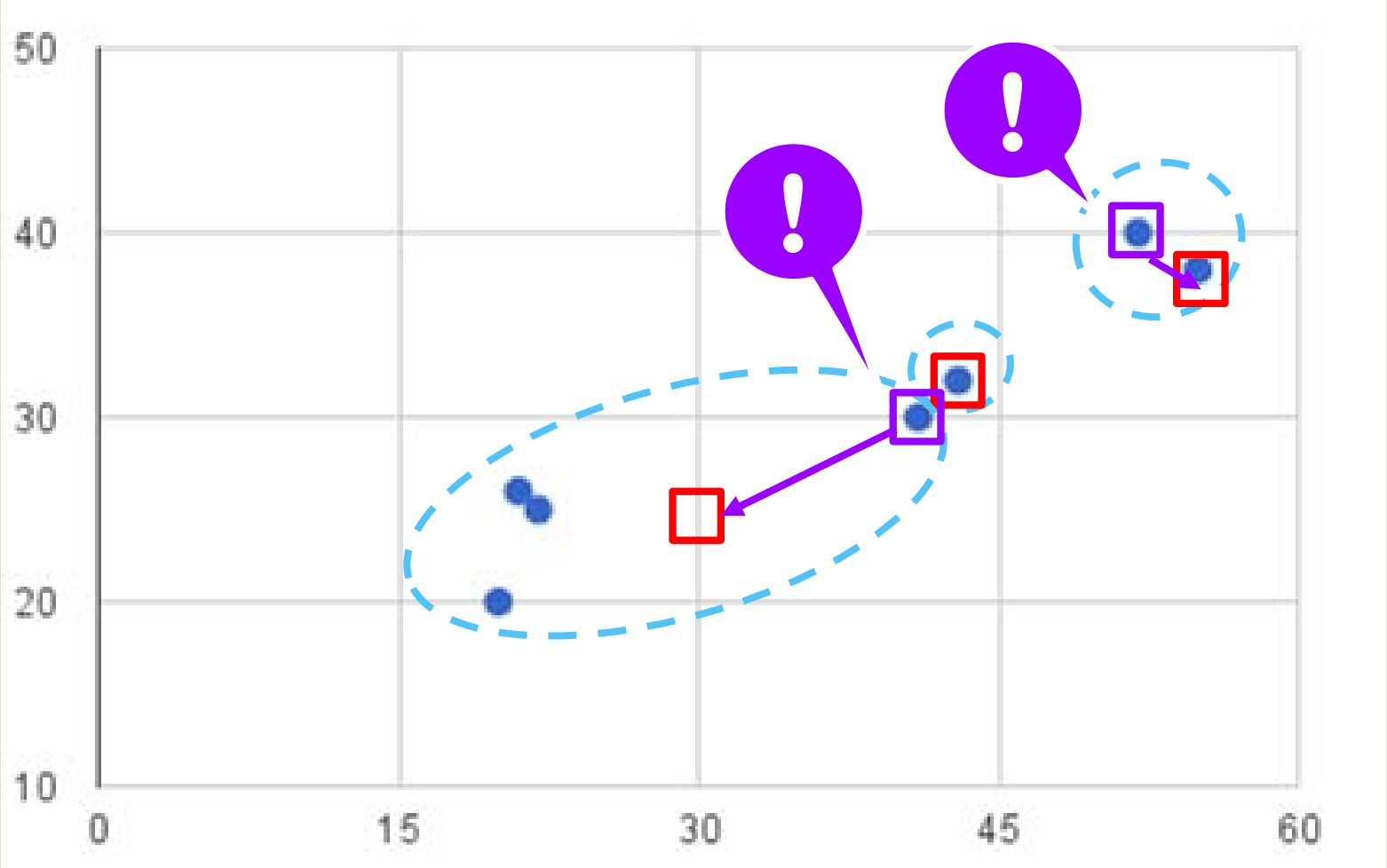
1-3. 計算群集中心

利用各群集中目前所包含的資料點，重新計算各群集之**群集中心點**



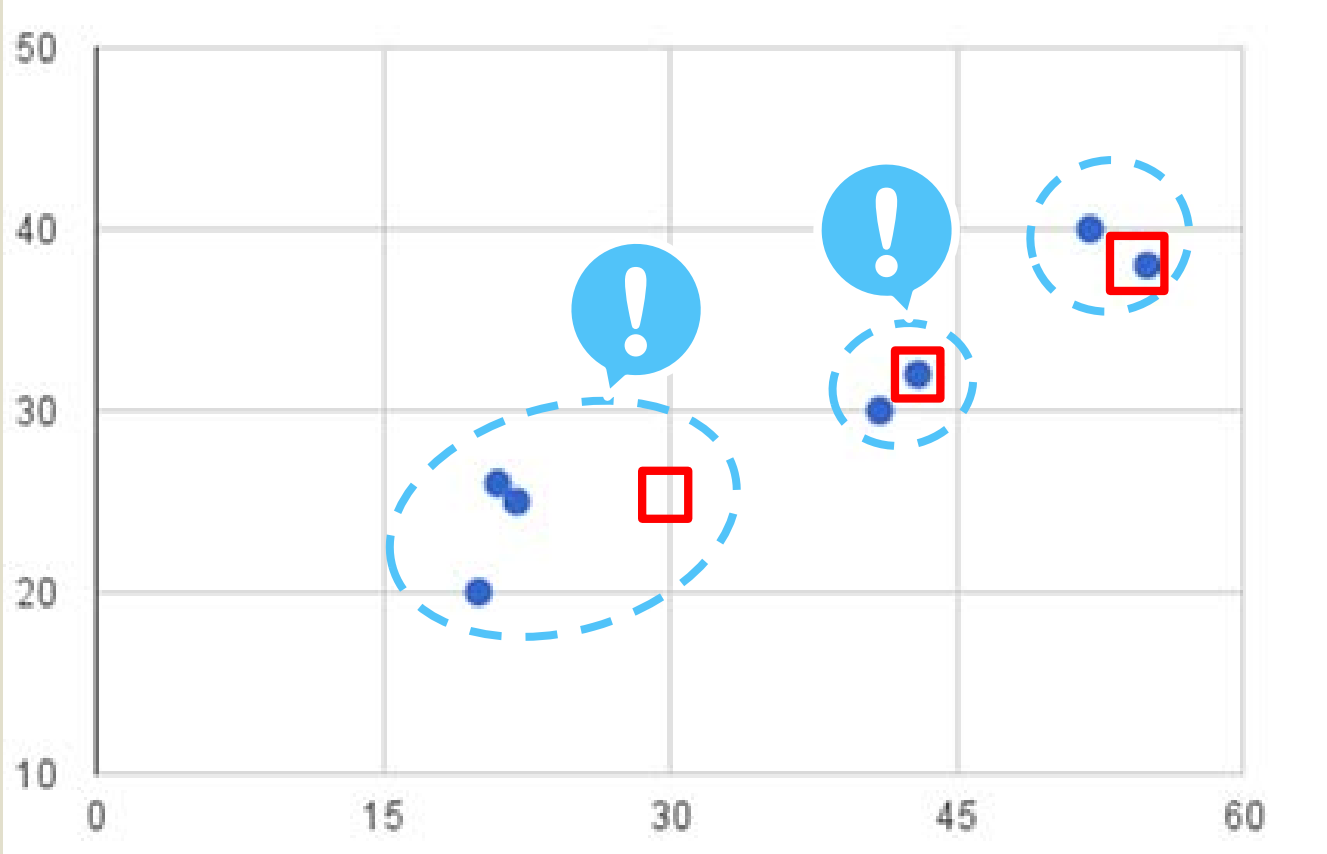
1-4. 結束第一輪分群

新的**群集中心點**與前一階段的**群集中心點**不同，故繼續執行分群



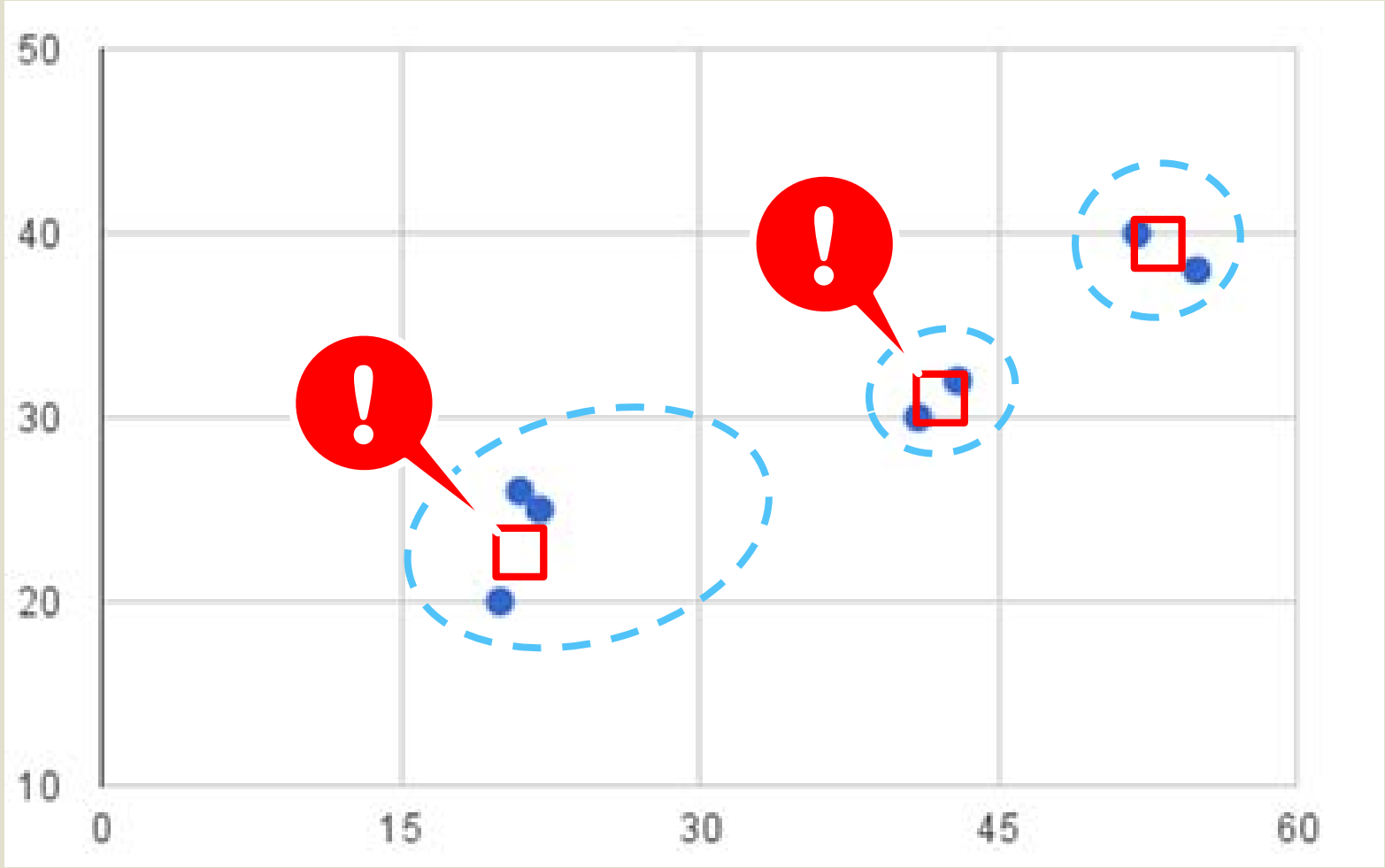
2-1. 形成群集

利用距離計算公式，將資料點分別歸類到距其最近之**群集中心點**所屬之**群集**，形成 ***k=3*** 的群集。



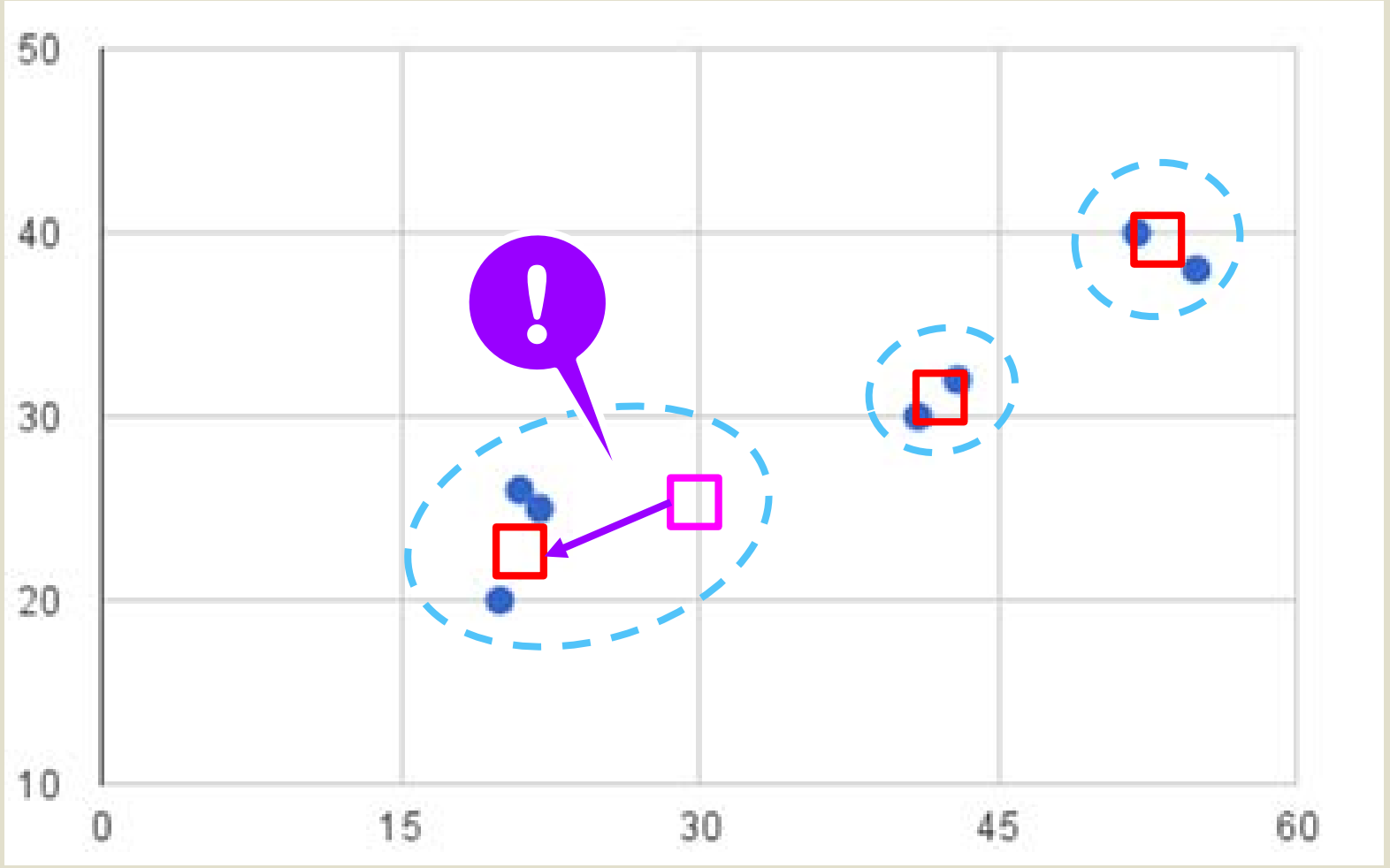
2-2. 計算群集中心

利用各群集中目前所包含的資料點，重新計算各群集之**群集中心點**



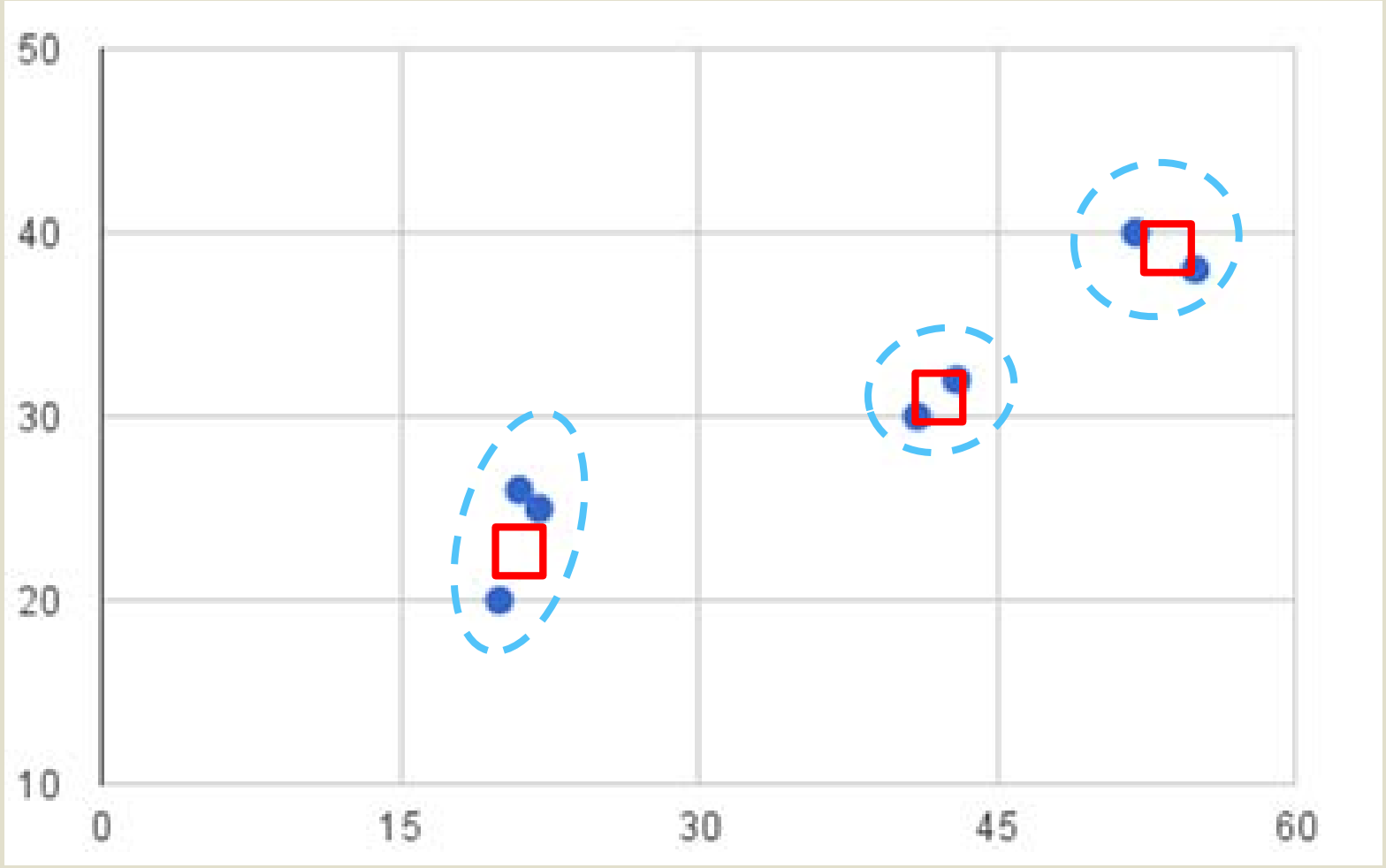
2-3. 結束第二輪分群

新的**群集中心點**與前一階段的**群集中心點**仍不同，故繼續執行分群



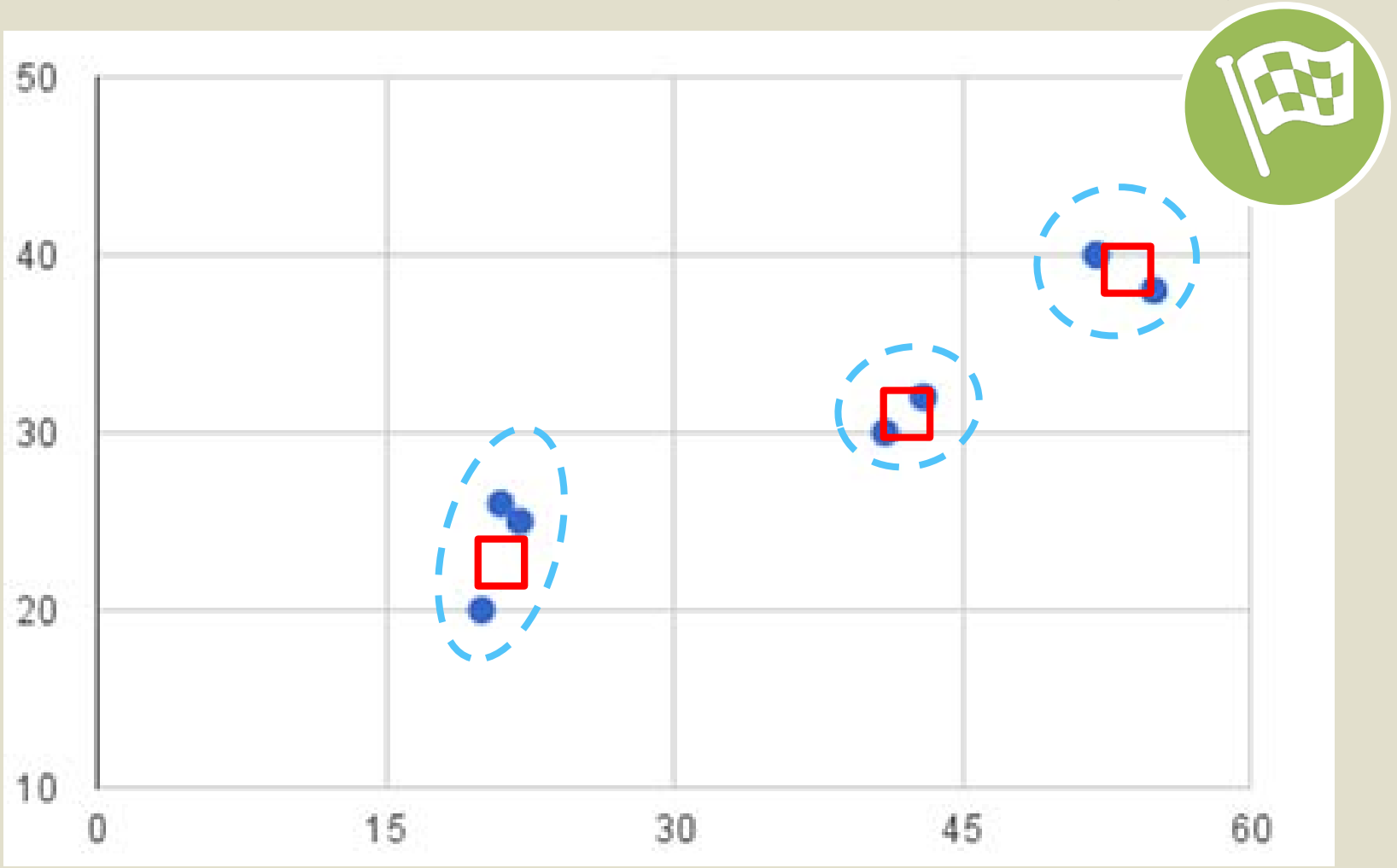
3-1. 計算群集中心

利用各群集中目前所包含的資料點，重新計算各群集之**群集中心點**



3-2. 產生最終分群結果

新的**群集中心點**與前一階段的**群集中心點**完全相同，故分群分析演算結束



如何計算距離？

資料點距離

- 歐氏距離
- 曼哈頓距離

集群間距離

- 單一連結
- 完整連結
- 平均連結

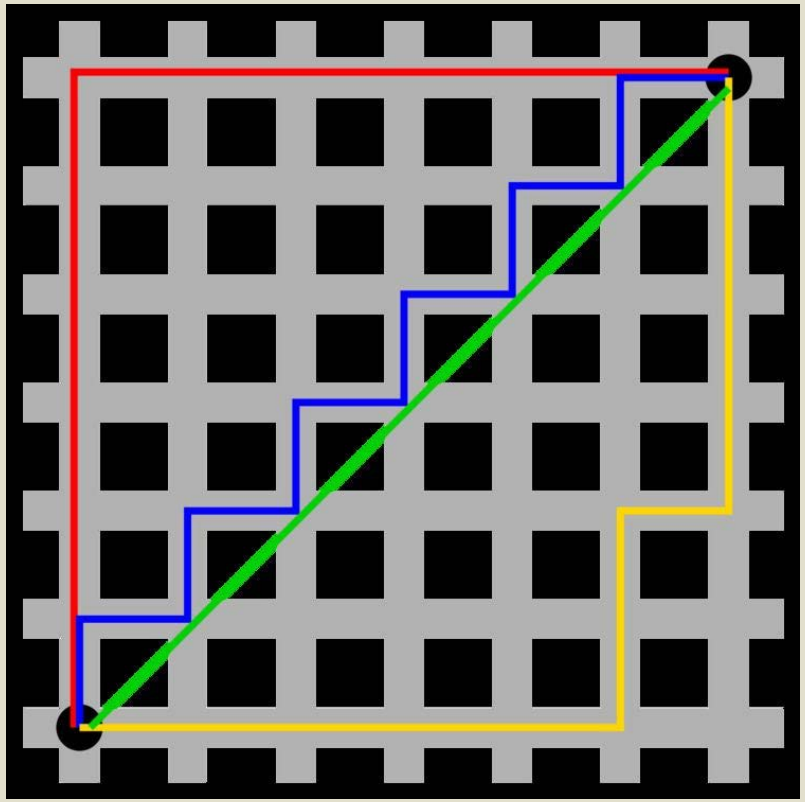
歐基里德距離

- **Euclidean distance**

計算兩個資料點在二維空間中的直線距離，計算公式為：

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



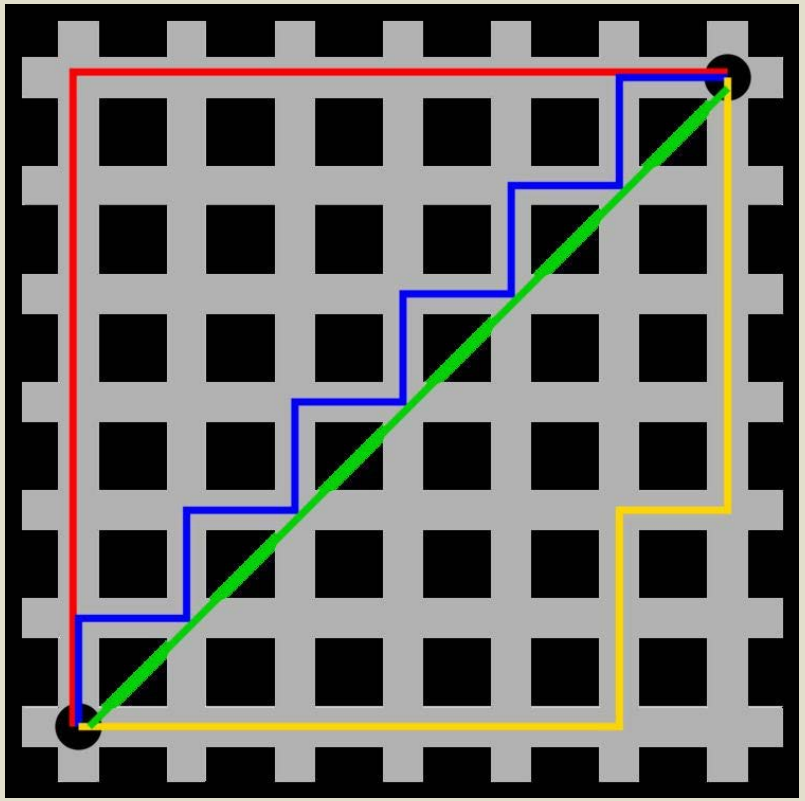
曼哈頓距離

- **Manhattan distance**

在曼哈頓街區要從一個點到達另一個點，距離顯然不是兩點間的直線距離，這個實際距離即「曼哈頓距離」，計算公式為：

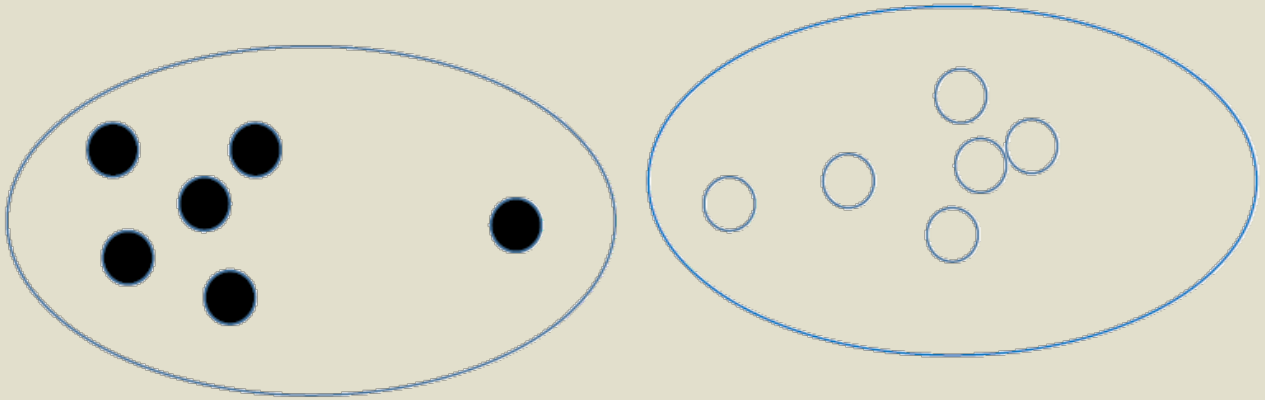
$$D = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$

$$= \sum_{i=1}^{n} |x_i - y_i|$$



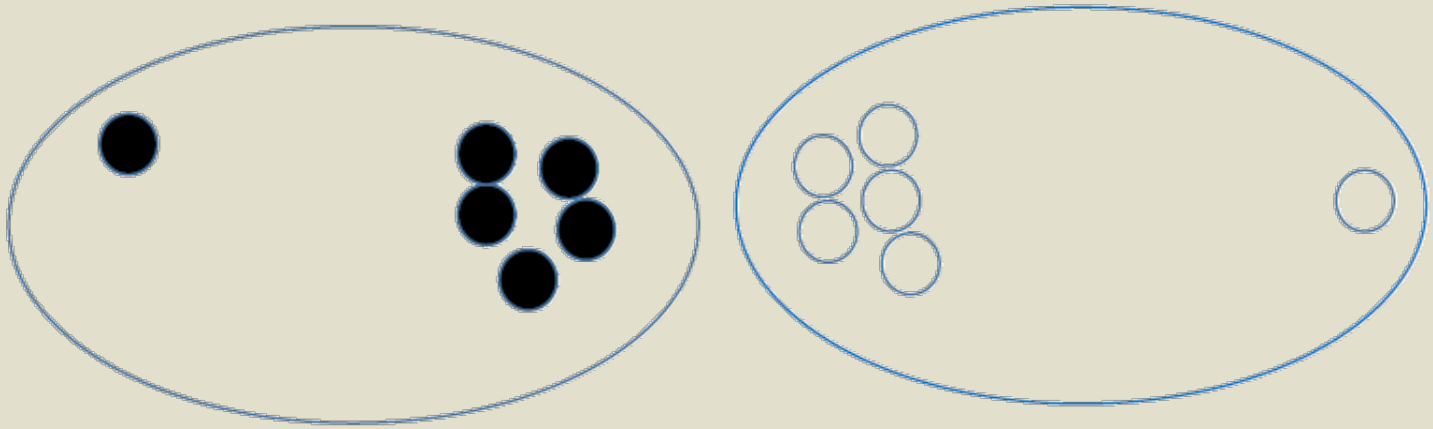
兩群集間距離：單一連結

- 單一連結聚合演算法**(single-linkage agglomerative algorithm)**：群集與群集間的距離可以定義為不同群集中最接近兩點間的距離。
- 集群間的距離定義為不同集群中最接近兩點間的距離，會在過程中產生「大者恆大」的效果。



兩群集間距離：完整連結

- 完整連結聚合演算法(**complete-linkage agglomerative algorithm**）：群集間的距離定義為不同群集中最遠兩點間的距離，這樣可以保證這兩個集合合併後，任何一對的距離不會大於 **d**。
- 集群間的距離定義為不同集群中最遠兩點間的距離，容易產生「齊頭並進」的效果。



兩群集間距離：平均連結

- 平均連結聚合演算法**(average-linkage agglomerative algorithm)**：群集間的距離定義為不同群集間各點與各點間距離總和的平均。
- 集群間的距離定義為：不同集群之間，各點間距離總和的平均，也就是把兩個集合中的點兩兩的距離全部放在一起求一個平均值，相對也能得到合適一點的結果。

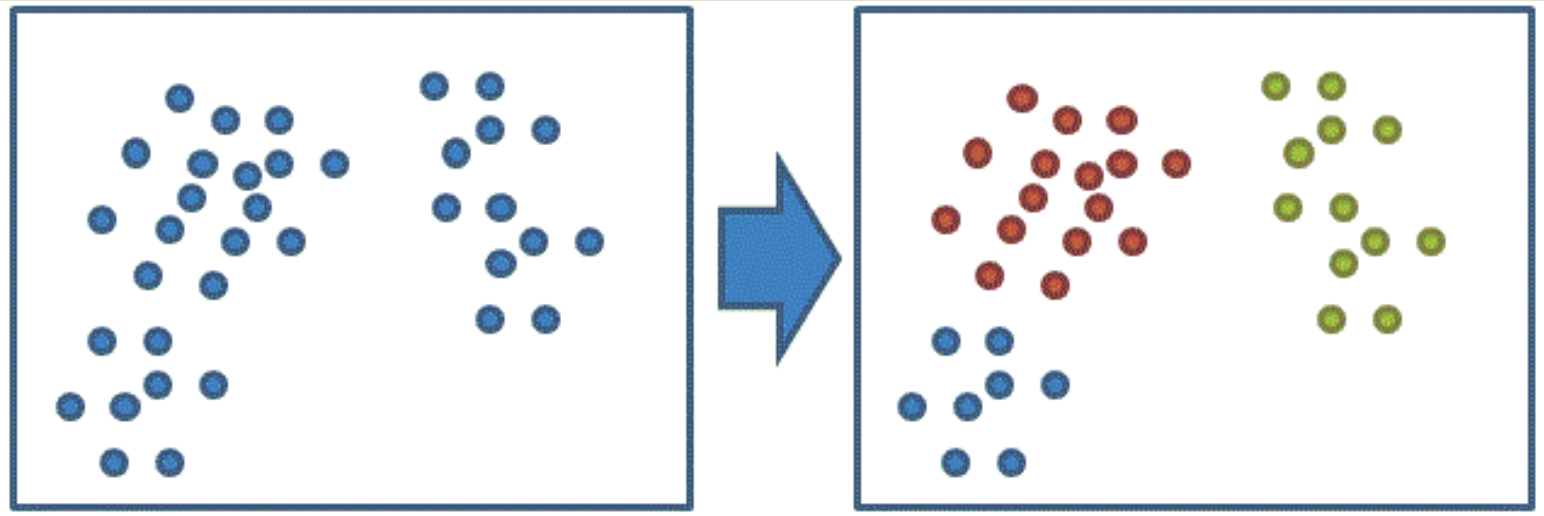
兩群集間距離：華德法

- 華德法（**Ward's method**）：

  群集間的距離定義為在將兩群合併後，各點到合併後群中心的距離平方和。

K - MEANS注意事項

- 各個Cluster的大小: 要確保Cluster具代表性，原則上5-10% 為最低門檻
- 總體Cluster的數量: 過多欄位會產生過多的Cluster，造成分析上的困難，建議2-12個
- 期望群內距離短群外距離長



K - MEANS優缺點

優點

- 演算法簡單且可以快速地完成分群任務
- 有效處理資料分布集中的大數據集

缺點

- 分群結果容易被雜訊與極端值影響
- 需要事先設定 **k** 值，若不恰當會無法有效分群
- 只能收斂到局部最小

LAB

- 7.1_Clustering-Hierarchical.ipynb

  from sklearn.cluster import AgglomerativeClustering

- 7.1_Clustering-Kmeans.ipynb

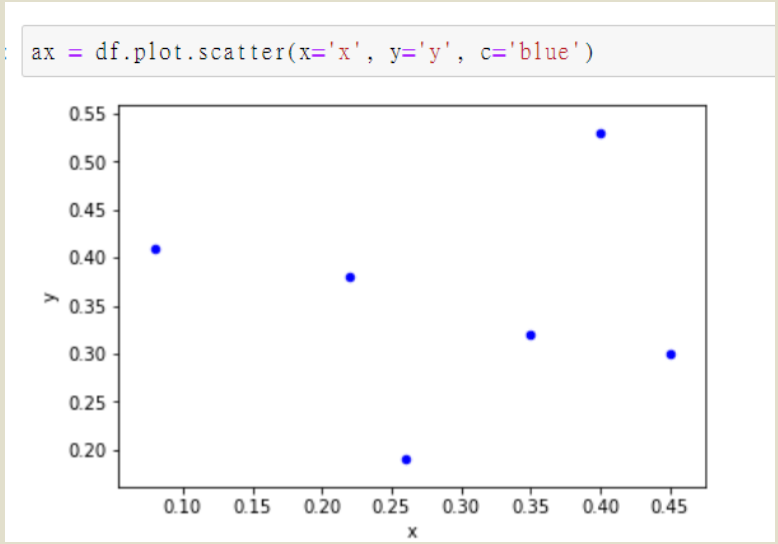
  from sklearn.cluster import Kmeans

- 7.2_Clustering實作.ipynb

LAB - HIERARCHICAL CLUSTERING

- 計算群集距離衡量方式**sklearn.metrics.pairwise_distances**
  7.3_Distance_Matrix.ipynb

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

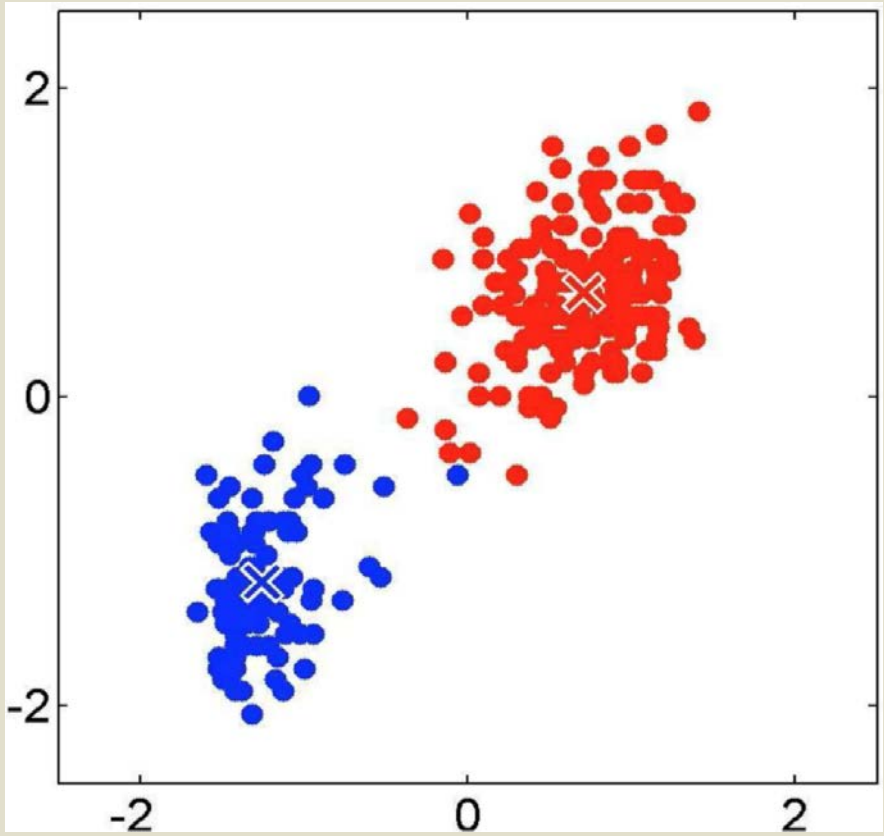


LAB - IMAGE SEGMENTATION

- Using K-Means Clustering

Perform K-means clustering as taught in the lectures

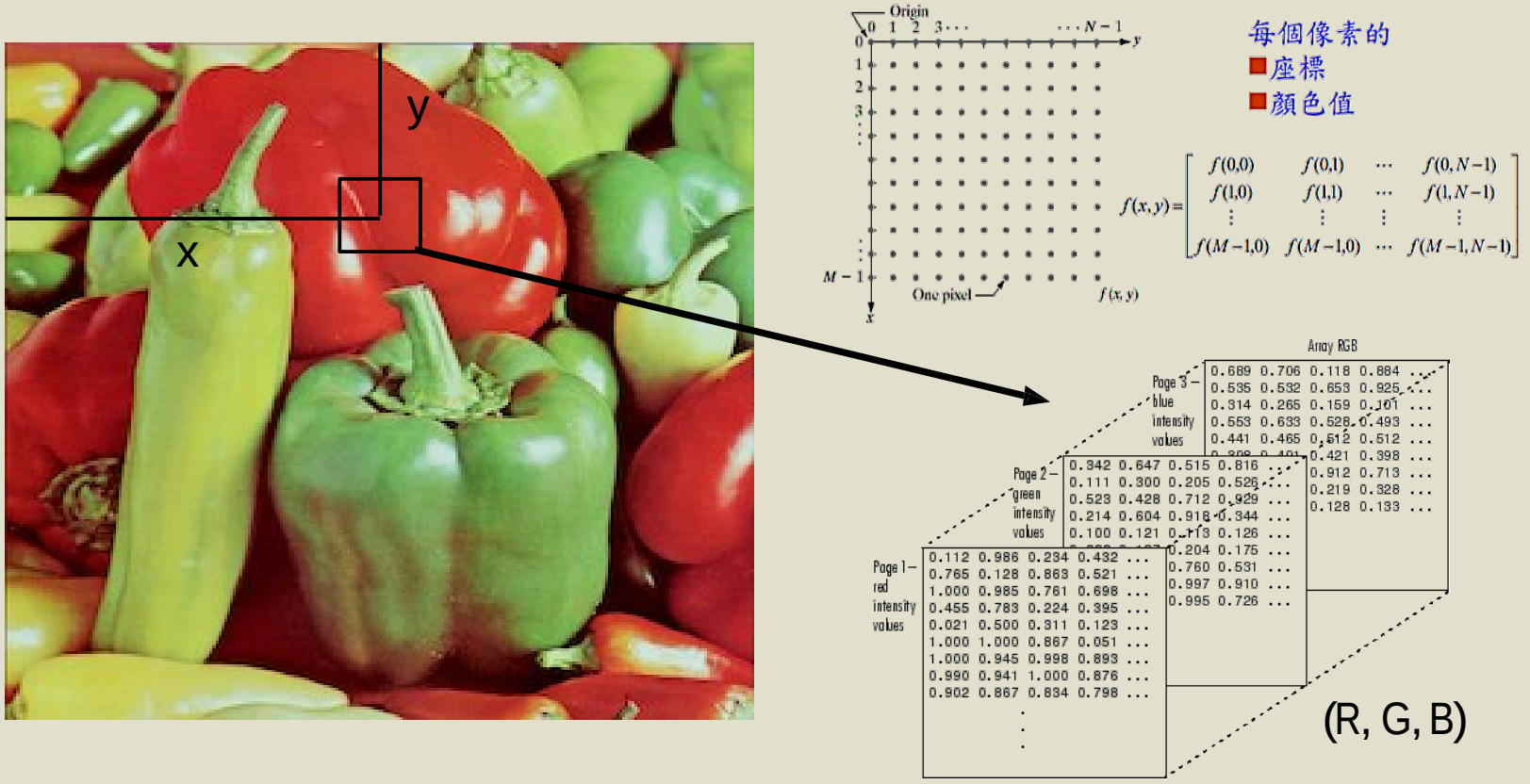
Example: K = 2



The lab practice referred to NTU Deep Learning for Computer Vision (DLCV) course website.

LAB - IMAGE SEGMENTATION

- Using K-Means Clustering
  - Perform K-means clustering on each pixel's color



The lab practice referred to NTU Deep Learning for Computer Vision (DLCV) course website.

LAB - IMAGE SEGMENTATION

- **Example:**

  K-means clustering on each pixel's color with different number of clusters

  For each clustering group, replace all pixels' RGB values with that of the cluster center

K = 2 K = 4 K = 8 K = 16 K = 32

The lab practice referred to NTU Deep Learning for Computer Vision (DLCV) course website.

LAB - IMAGE SEGMENTATION

Problem 1: K-Means Clustering (24%)

K-Means Clustering is an unsupervised learning algorithm for data grouping. In image segmentation, it can be applied to partition image pixels into different groups based on the associated pixel values or features. In this problem, you will learn how to segment the provided image by using K-means clustering.

1. (10%) For $K = 2, 4, 8, 16$, and 32, perform K-means clustering on the provided `bird.jpg` by taking the RGB values of each pixel as the feature of interest. Take a 64×64 pixel color image for example, we have a total of $64 \times 64 = 4096$ data points for K-means clustering, and each data point is described as a three dimensional vector (i.e., (R, G, B)). To visualize your image segmentation results, plot the clustering results by replacing all pixels' RGB value in each cluster with the that of the corresponding cluster center.
2. (6%) Repeat 1. but take both RGB values and the location (x and y) as a five dimensional vector as the feature for describing each pixel. Show the segmentation results.
3. (8%) Compare your results obtained in 1. and 2., and briefly explain the differences between the two methods under the same K. If further improved segmentation results would be desirable, please provide possible modification or extension to the above feature definition (and visualize your results).

The lab practice referred to NTU Deep Learning for Computer Vision (DLCV) course website.