

統計套裝軟體應用 分群分析 (CLUSTERING ANALYSIS)

B11201015 王瑀晴

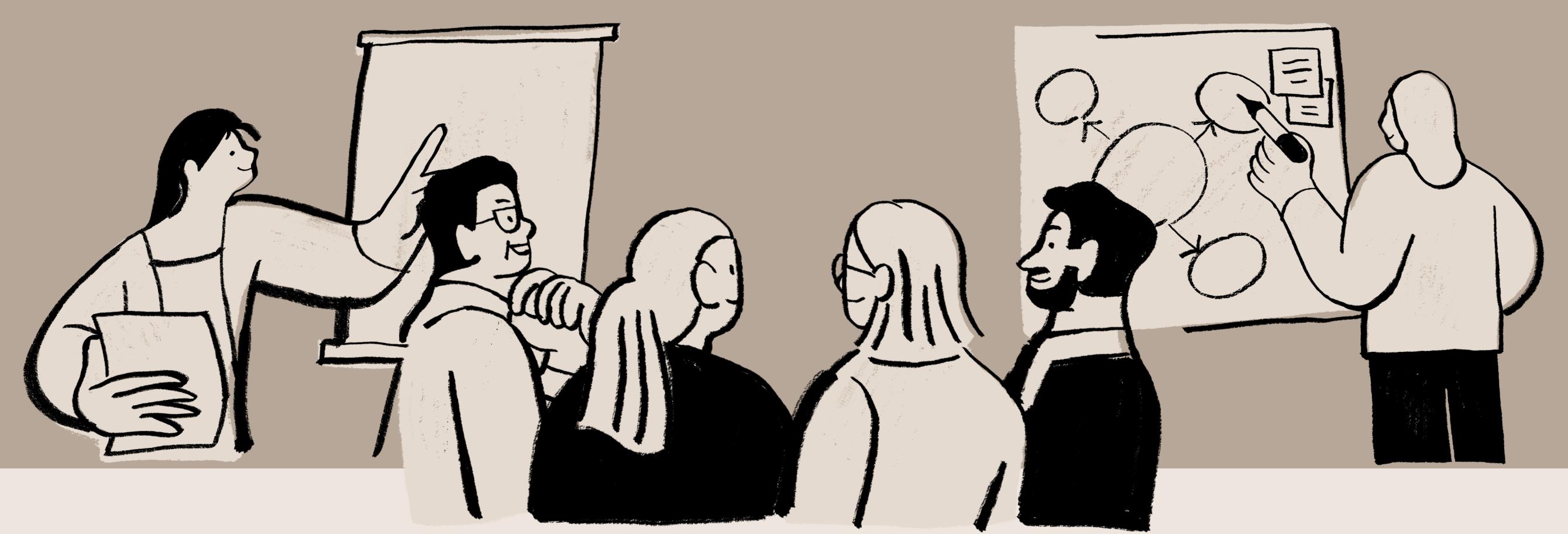
B11201021 程潛威

B11201106 張妮

B11201108 盧宜婷

B11201109 蔡碧芸

B11201110 王思樺



大綱

- 分群概念介紹
- 套件介紹
- 核心演算法與原理
- 實作演練
- 應用領域分析



什麼是 CLUSTERING (分群) ？

定義：

- 資料探勘 (Data Mining) 中的一種「非監督式學習」方法。
- 將數據依「相似度」分群



比較項目	監督式學習	非監督式學習
標籤	有	無
學習目標	「輸入 → 輸出」的對應關係	找出資料的內在架構
模型學習什麼	預測結果	資料本身的結構與規律
人為介入程度	高	低

什麼是 CLUSTERING (分群) ？

常見分群方法

- K-means
- Hierarchical Clustering
階層式分群



生活案例

- 找出尖峰時段
- 分析新聞類型
- 發現學生學習模式



二、演算法原理

k-Means 分群法

Hierarchical Clustering 階層式分群



K-MEANS 集群分析

什麼是集群分析？

核心直覺：「物以類聚」

生活比喻：

- 就像學生在操場集合，男生自然聚成一群，女生聚成一群。
- 雖然資料（如身高、體重）本身不會動，但我們可以透過演算法找到分界的基準。

K-means 的字面意義：

- K：你想將資料分成幾群（Clusters）？
- Means：每一群的「平均值」，也就是群心（Center）。



關鍵角色 - 「會動的群心」

Key-man (群心)

- 想像群心就像是團體中的 Key-man (核心人物)。
- 在分群尚未完成前，Key-man 是會變動的。

動態過程：

- 一開始隨機選 Key-man。
- 大家往最近的 Key-man 靠攏。
- 因為成員變了，團體的重心（平均位置）跟著變，Key-man 的位置也隨之移動。
- 一直換到位置穩定為止。



運作原理



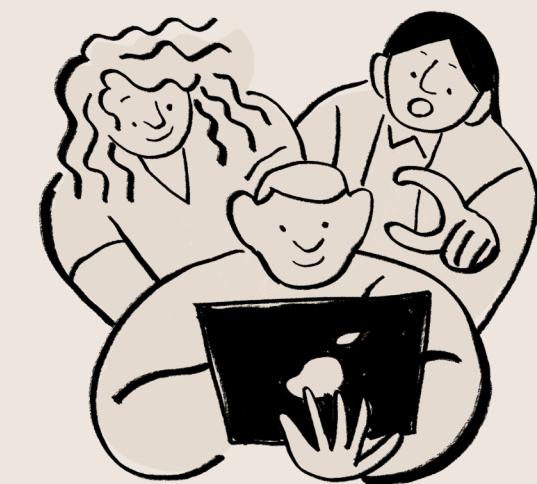
1. 初始化 (Initialization) :

- 隨機設定 K 個點作為初始群心
- (初始點的選擇會影響結果)



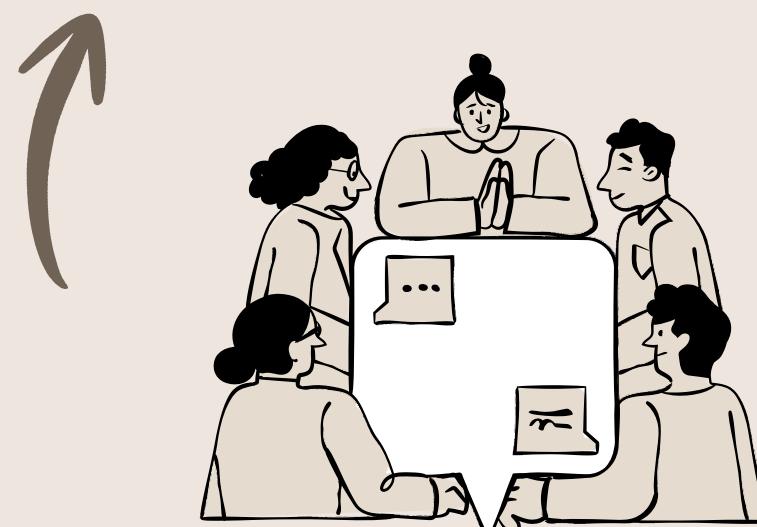
2. 分類 (Assignment) :

- 計算每一個資料點到 K 個群心的距離
- 將資料點指派給「距離最近」的那個群心



3. 更新 (Update) :

- 重新計算每一群的中心點
(取群內所有資料座標的平均值)
- 群心位置發生移動



4. 重複 (Loop) :

- 重複步驟 2 和 3，直到群心不再變動 (收斂)

數學原理

目標函數：最小化群內誤差平方和 (Minimize Sum of Squared Errors)

讓同一群的人靠得越緊密越好。

距離公式：歐基里德距離 (Euclidean Distance)

★ $d(x, y) = \sqrt{\sum (x_i - y_i)^2}$

用來計算資料點與群心之間的「直線距離」



手肘法 Elbow method



手肘法是以誤差平方和 (sum of the squared errors, SSE) 為指標，計算每一群中的每一個資料點到群中心的距離，找出 SSE 相對平緩的資料點作為拐點 (Inflection point)，並以此拐點選為群數。

SSE 計算方式如下：

$$\star \text{SSE} = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

其中，

- SSE 代表集群的好壞，也就是所有資料的誤差
- 總共有 K 個群
- C_i 代表其中一群，也就是第 i 個群
- p 代表 C_i 中的資料點
- m_i 代表該群心，也就是 C_i 中所有資料的平均值

階層式分群法 (hclust)

方法：

1. 各自為群

若有 N 筆資料，
就有 N 個群

2. 合併最相似者

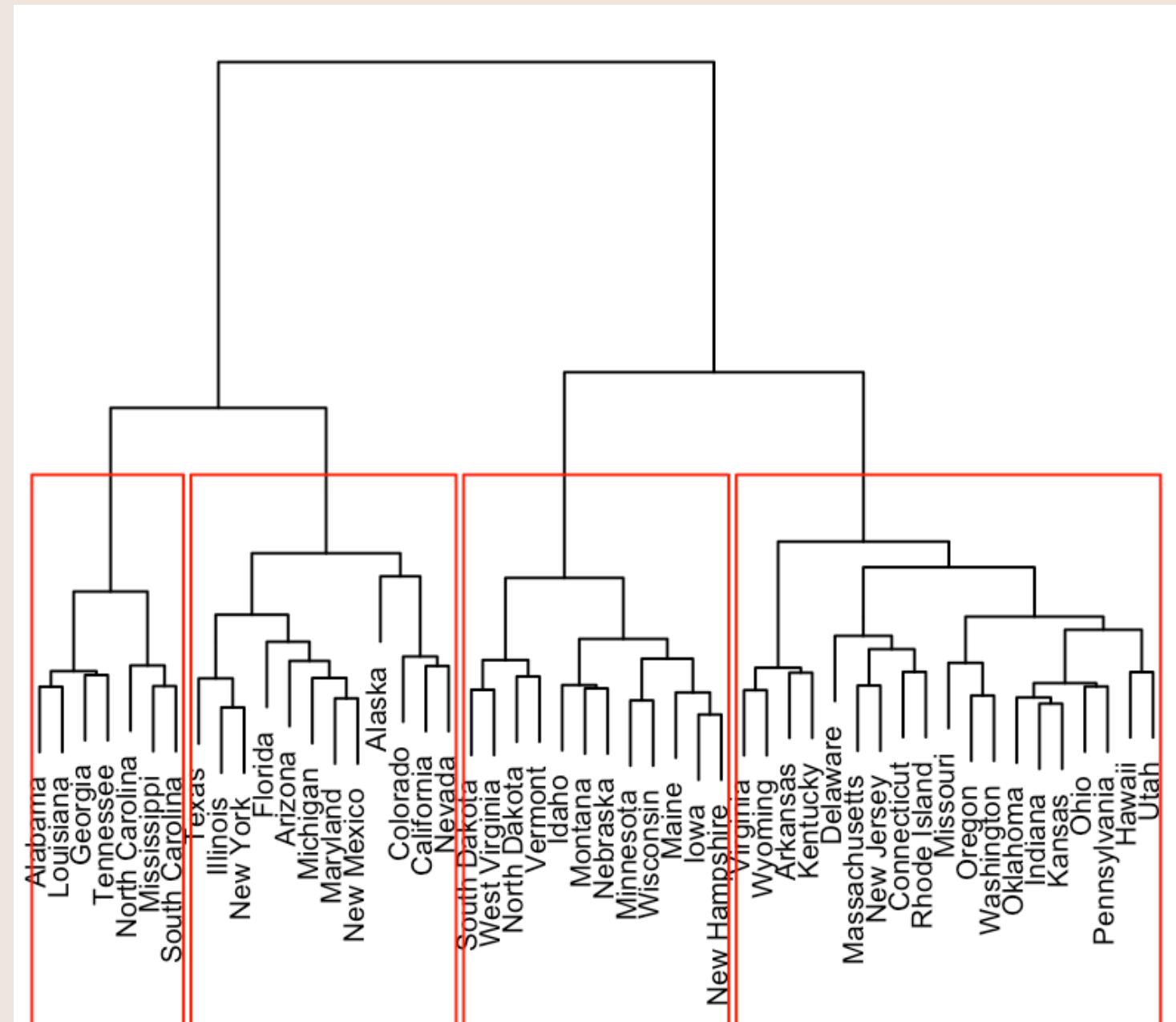
將距離最近的 兩個群聚合併

3. 重複合併

重複步驟 2，不斷將 小群合併成大群

4. 形成樹狀圖

直到所有資料點都合併成一個大群聚為止，最後再決定分成幾群



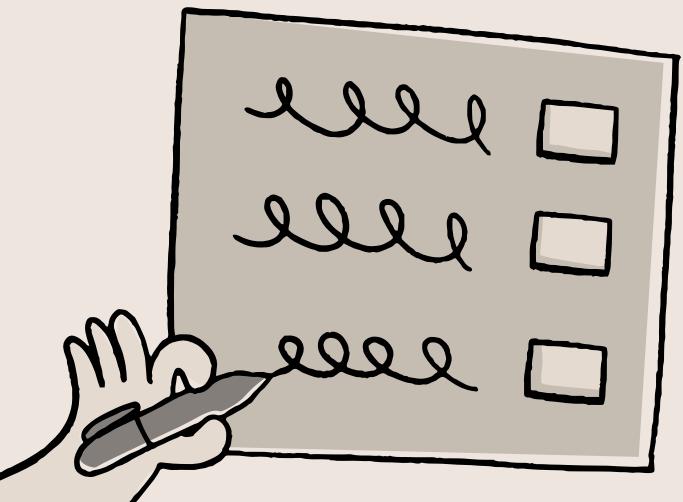
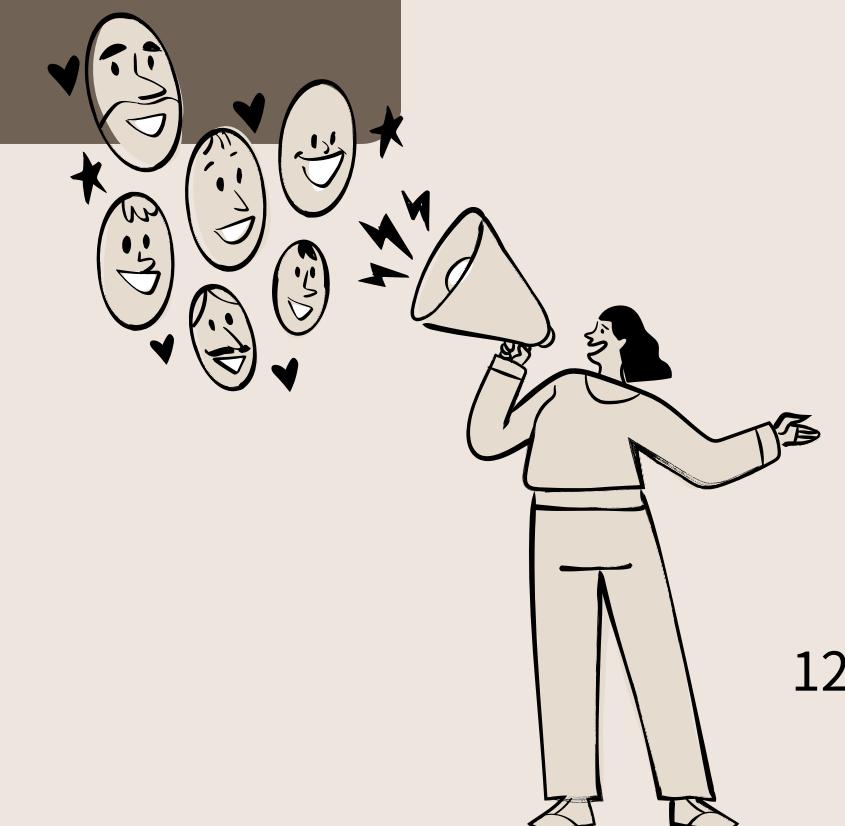
Ward's Method

目標：最小化誤差平方和(ESS)

$$(1) ESS_C = \sum_{x_i \in C} (x_i - \bar{x}_C)^2 \quad (2) \Delta(A, B) = ESS_{A \cup B} - (ESS_A + ESS_B)$$

$$(3) D(A, B) = \frac{n_A \cdot n_B}{n_A + n_B} \|\bar{x}_A - \bar{x}_B\|^2$$

合併時會考慮兩個因素：
1. 距離 2. 權重



PR

AC

TI

CE

三、實作演練

實作案例背景

Demo 1 - k-Means 實作

Demo 2 - 階層式分群與評估



實作案例背景



金融業競爭激烈，需深入了解顧客行為以提升營運績效
信用卡顧客行為多樣，難以用傳統方式有效分類

Credit Card Dataset for Clustering

大量顧客使用數據（消費額、繳款行為、交易次數等）

資料無標籤 → 無法直接使用監督式模型

目標：從顧客行為中找出隱含群體與行為模式

可協助企業

- 辨識高價值客群
 - 提供客製化行銷
- 找出可能流失的客戶
 - 提前挽留
- 作為 CRM 與風險管理策略基礎
- 提升行銷效率與整體獲利能力

Demo 1 - k-Means 實作



dplyr

cluster

factoextra

ggplot2

資料前處理 → 移除不需要欄位、資料欄位操作與補值

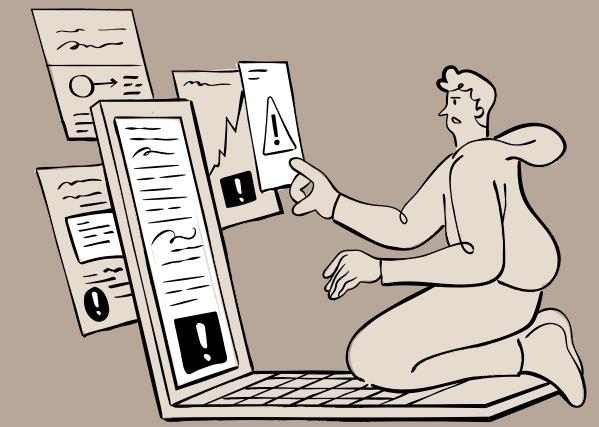
- **select()**：選取或移除資料框中的特定欄位
- **pipe (%>%)**：將前一步的結果「傳遞」給下一個函數

```
12 # -----
13 # 資料前處理
14 #
15
16 # 移除不需要的欄位
17 df = data %>% select(-CUST_ID)
18
19 # 檢查缺失值狀況
20 colSums(is.na(df))
21
22 # 用中位數補值(刪除整列可能造成資料大量流失)
23 for (col in names(df)) {
24   df[[col]][is.na(df[[col]])] = median(df[[col]], na.rm = TRUE)
25 }
26
27 # 標準化 (z-score)
28 df_scaled = scale(df)
29
```

消除單位與數量級差異

避免數值大的特徵（金額）掩蓋了數值小的特徵（頻率）

Demo 1 - k-Means 實作



dplyr

cluster

factoextra

ggplot2

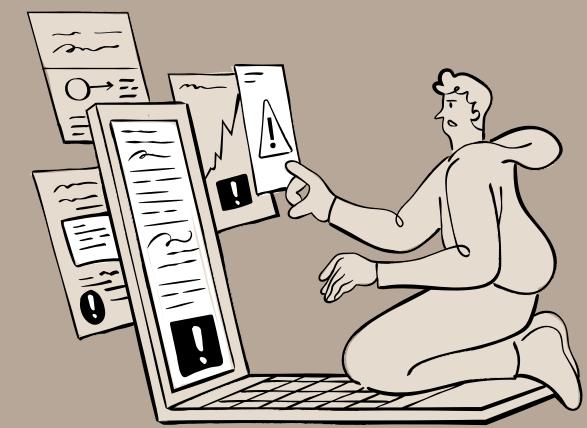
分群演算法核心工具 → 執行 K-means 與分群運算

- **kmeans()**：用於執行資料的分群分析

```
30 # -----
31 # K-means 分群
32 #
33
34 set.seed(118)
35 km = kmeans(df_scaled,
36               centers = 4,    # 可調整群數
37               nstart = 25)   # 多次初始化避免 local optimum
38 km
39
```

- **centers = 4 (群數設定)**：即為人為設定將客戶分成 4 堆
- **nstart = 25 (嘗試次數)**：確保結果穩定
 - 因為一開始是隨機選中心點的，如果運氣不好，選到的起點很爛，分出來的結果就會很爛（掉進局部最佳解 Local Optimum）

Demo 1 - k-Means 實作



dplyr

cluster

factoextra

ggplot2

分群結果視覺化 & PCA 降維

- **fviz_cluster()** : 直接把高維分群結果以 PCA 降維成 2D 圖

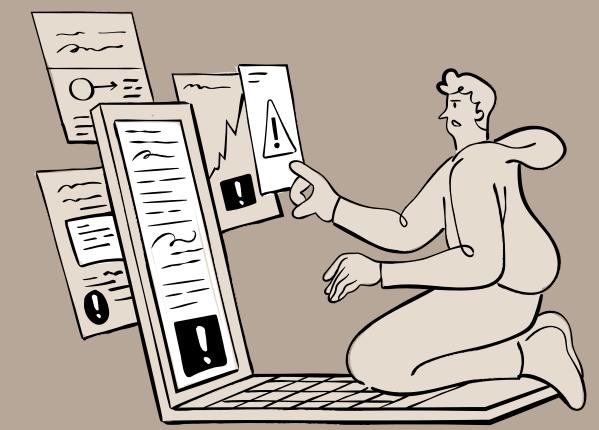
供 factoextra 套件底層運作 → 美化並建立分群視覺化的主題樣式

- **theme_minimal()** : 圖表背景設成簡潔的白色，去掉灰底網格

```
43 # -----
44 # 視覺化
45 # -----
46
47 # PCA 降維 + 分群結果視覺化
48 fviz_cluster(
49   km,
50   data = df_scaled,
51   geom = "point",
52   ellipse.type = "convex",
53   ggtheme = theme_minimal(),
54   main = "Credit Card Customer Clustering Result"
55 )
56
```

取得原始座標數據 → 17個維度 (欄位)

Demo 1 - k-Means 實作



dplyr

cluster

factoextra

ggplot2

分群結果視覺化 & PCA 降維

- **fviz_cluster()** : 直接把高維分群結果以 PCA 降維成 2D 圖

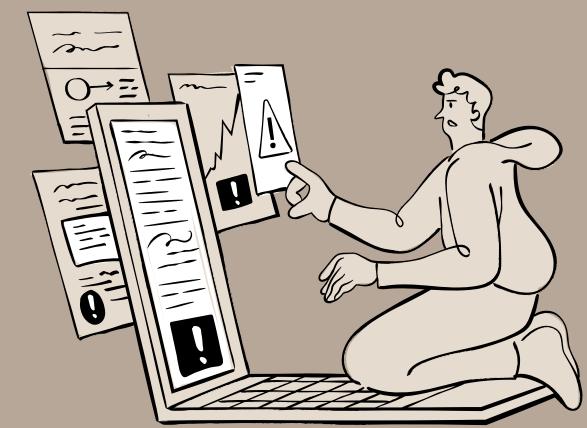
供 factoextra 套件底層運作 → 美化並建立分群視覺化的主題樣式

- **theme_minimal()** : 圖表背景設成簡潔的白色，去掉灰底網格

```
43 # -----
44 # 視覺化
45 # -----
46
47 # PCA 降維 + 分群結果視覺化
48 fviz_cluster(
49   km,
50   data = df_scaled,
51   geom = "point",
52   ellipse.type = "convex",
53   ggtheme = theme_minimal(),
54   main = "Credit Card Customer Clustering Result"
55 )
56
```

繪圖幾何元素 → 「點」代表每個客戶

Demo 1 - k-Means 實作



dplyr

cluster

factoextra

ggplot2

分群結果視覺化 & PCA 降維

- **fviz_cluster()** : 直接把高維分群結果以 PCA 降維成 2D 圖

供 factoextra 套件底層運作 → 美化並建立分群視覺化的主題樣式

- **theme_minimal()** : 圖表背景設成簡潔的白色，去掉灰底網格

```
43 # -----
44 # 視覺化
45 # -----
46
47 # PCA 降維 + 分群結果視覺化
48 fviz_cluster(
49   km,
50   data = df_scaled,
51   geom = "point",
52   ellipse.type = "convex",
53   ggtheme = theme_minimal(),
54   main = "Credit Card Customer Clustering Result"
55 )
56
```

畫出每一群的範圍框線 → 多邊形：有稜有角

> km

K-means clustering with 4 clusters of sizes 3989, 1197, 3363, 401

各群人數

Cluster means:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	
1	-0.2659988	-0.3707411	-0.3427854	-0.230659248	-0.3865510	-0.1829484	
2	1.4592898	0.3849408	-0.2350014	-0.164145135	-0.2541801	1.6872281	
3	-0.3186789	0.2474748	0.1129728	0.004276753	0.2590371	-0.3649444	
4	0.9626347	0.4634767	3.1639402	2.748625306	2.4315805	-0.1559175	
	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY		
1	-0.7939753	-0.3897275		-0.7102531	-0.1035736		
2	-0.5064753	-0.2137049		-0.4515378	1.7485690		
3	0.9868072	0.3233029		0.8767322	-0.4607102		
4	1.1341286	1.8033921		1.0604485	-0.3254707		
	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
1	-0.1658335	-0.4740003	-0.33463907	-0.2623140	-0.12512753	-0.2582916	-0.05424328
2	1.6192911	-0.2840458	0.83873996	0.6032523	0.49668253	-0.4062667	-0.09752305
3	-0.3591435	0.3036958	-0.07303821	-0.1357102	-0.08676555	0.3948791	0.06049096
4	-0.1720256	3.0161122	1.43773324	1.9468103	0.48976875	0.4704433	0.32339259

Clustering vector:

[1]	1	2	3	1	1	3	4	3	1	1	3	1	3	1	2	1	1	3	1	1	2	3	1
[51]	2	3	1	3	1	1	4	1	2	1	1	3	1	3	2	1	1	1	3	1	1	1	1
[101]	1	3	4	3	1	1	3	2	3	3	1	1	1	4	1	2	4	1	3	1	2	2	4
[151]	2	3	3	4	3	1	2	1	4	1	1	3	4	3	1	2	3	1	4	3	1	1	3
[201]	1	1	3	2	4	1	1	2	2	1	1	3	1	2	4	3	1	3	1	3	3	2	4
[251]	2	3	1	3	1	1	1	3	3	1	4	4	2	3	4	2	1	1	1	3	1	3	1
[301]	1	3	1	3	2	3	3	3	1	3	4	1	1	1	2	3	1	3	2	1	1	3	3
[351]	1	3	2	1	1	4	3	2	1	1	3	4	1	3	2	2	1	4	1	1	3	1	2
[401]	3	3	2	3	2	1	1	2	3	1	3	4	2	4	3	3	2	1	3	3	3	1	1
[451]	3	3	3	2	3	3	3	1	4	3	3	3	3	1	2	3	4	2	3	2	1	2	3
[501]	3	4	3	3	3	3	4	1	3	3	4	2	3	3	3	1	3	4	2	1	2	3	3
[551]	4	4	2	3	2	3	1	2	2	1	4	1	4	3	1	2	2	1	4	4	1	1	1
[601]	2	1	2	3	3	1	2	3	2	1	3	4	1	2	2	3	3	1	3	1	2	1	4
[651]	3	3	4	1	1	4	1	1	3	3	3	1	4	1	2	1	2	3	3	1	3	2	3
[701]	1	1	1	1	3	1	1	2	4	4	2	1	4	1	3	3	1	1	4	3	2	1	1
[751]	2	3	2	1	1	2	3	3	2	2	1	1	1	2	3	1	3	1	4	3	3	1	1
[801]	2	4	1	2	3	1	3	3	2	3	2	1	1	3	1	2	3	3	3	1	3	2	2
[851]	1	2	1	2	4	1	3	1	3	3	3	1	2	1	1	2	4	1	3	3	3	3	1
[901]	1	1	2	2	1	3	1	1	1	2	2	1	3	3	1	2	3	3	1	3	1	1	1
[951]	3	3	3	1	3	1	1	1	1	3	2	3	3	1	1	4	3	1	2	3	1	3	1
[reached 'max' / getOption("max.print") -- omitted 7950 entries]																							

Within cluster sum of squares by cluster:

[1] 24690.55 23398.25 29463.26 21498.56
(between_SS / total_SS = 34.9 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss"
[7] "size" "iter" "ifault"

每一群的「平均長相」

因為數據經過標準化 (Z-score)，所以數字的含義如下：

- 0：代表平均水準
- 正數：代表高於平均
- 負數：代表低於平均

特徵	群組 1 (大眾/低活)	群組 2 (預借現金族)	群組 3 (一般刷卡族)	群組 4 (超級 VIP)
BALANCE (餘額)	-0.26 (低)	+1.45 (很高)	-0.31 (低)	+0.96 (高)
PURCHASES (刷卡金額)	-0.34 (很少)	-0.23 (少)	+0.11 (普通)	+3.16 (極高!)
CASH_ADVANCE (預借現金)	-0.18 (沒借)	+1.68 (借爆!)	-0.36 (沒借)	-0.15 (沒借)
CREDIT_LIMIT (額度)	-0.33 (低)	+0.83 (高)	-0.07 (普通)	+1.43 (最高)
PAYMENTS (還款金額)	-0.26 (少)	+0.60 (多)	-0.13 (普通)	+1.94 (還很多)

模型解釋力：

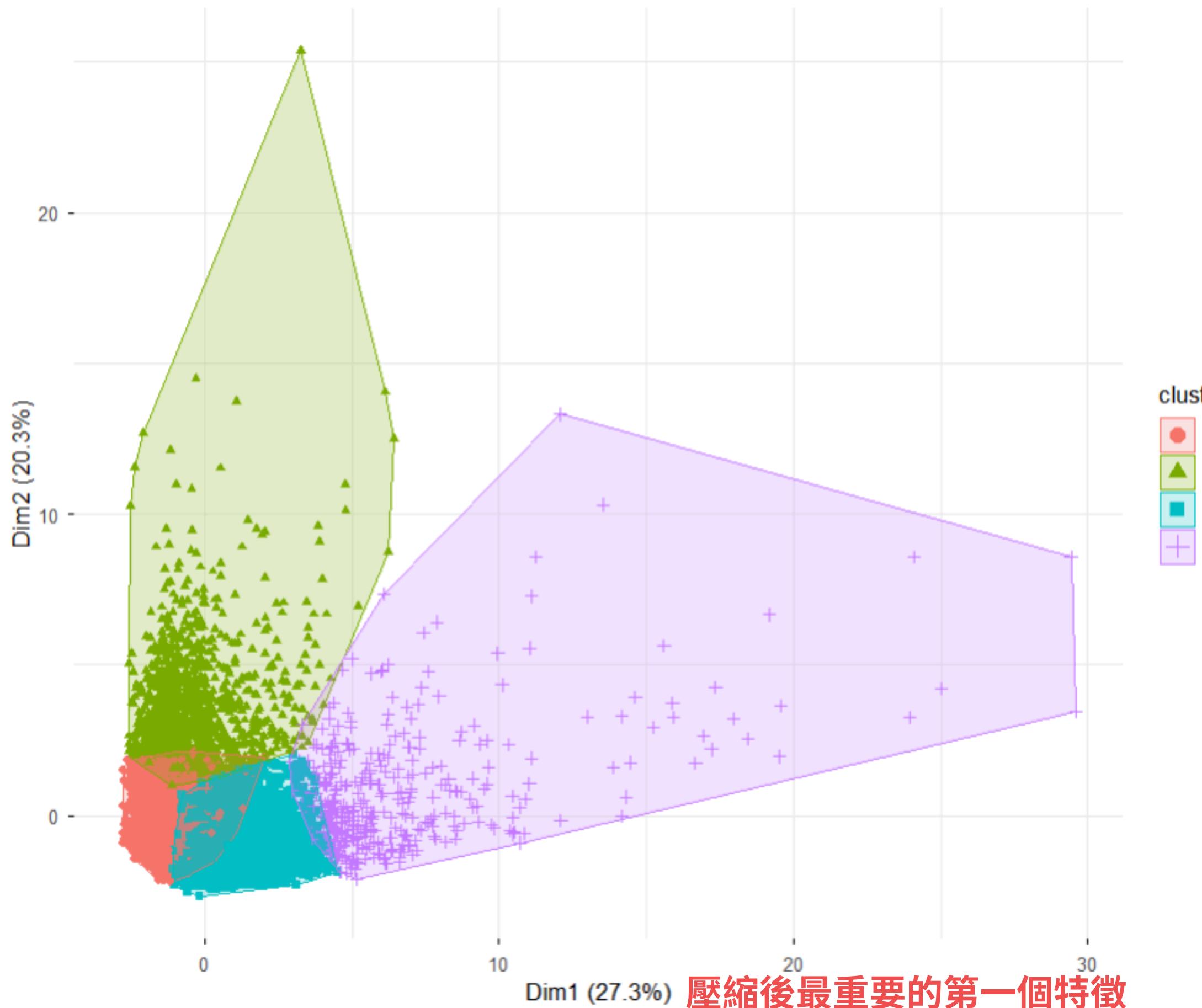
解釋了資料中約 35% 的差異性

是可以接受的範圍，但通常希望 > 50%

若要提高分數，可能需要：

- 嘗試不同的 K 值 (分 5 群或 6 群)
- 增加更多有用的欄位

Credit Card Customer Clustering Result



■ **VIP 大戶 (401人)**

高頻率 / 金額大 / 分期多 / 有能力付款
→ 高消費 + 高還款能力的 VIP
→ 策略：尊榮服務

■ **預借現金族 (1197人)**

現金預借頻率高 / 少全額繳清 / 餘額高 /
還款不穩定
→ 現金預借族、循環信用使用者
→ 策略：風險控管

■ **活躍小資族 (3363人)**

高頻率 / 金額中等或少 / 多全額繳清
→ 穩健型常用客戶
→ 策略：促銷激勵

■ **低活躍族群 (3989人)**

各項指標皆低
→ 輕度使用者、邊緣客群
→ 策略：挽留喚醒

Credit Card Customer Clustering Result



■ VIP 大戶 (401人)

高頻率 / 金額大 / 分期多 / 有能力付款

→ 高消費 + 高還款能力的 VIP

→ 策略：尊榮服務

■ 預借現金族 (1197人)

現金預借頻率高 / 少全額繳清 / 餘額高 /
還款不穩定

→ 現金預借族、循環信用使用者

→ 策略：風險控管

■ 活躍小資族 (3363人)

高頻率 / 金額中等或少 / 多全額繳清

→ 穩健型常用客戶

→ 策略：促銷激勵

■ 低活躍族群 (3989人)

各項指標皆低

→ 輕度使用者、邊緣客群

→ 策略：挽留喚醒

Demo 2 - 階層式分群實作



dplyr

cluster

factoextra

```
9 # -----
10 # 2：讀取資料與填補缺失值
11 #
12 df <- read_csv("CC GENERAL.csv")
13
14 # [dplyr] 源頭處理：填補缺失值 (NA)
15 # 說明：針對 MINIMUM_PAYMENTS 和 CREDIT_LIMIT 用中位數補值
16 df_clean <- df %>%
17   mutate(
18     MINIMUM_PAYMENTS = ifelse(is.na(MINIMUM_PAYMENTS), median(MINIMUM_PAYMENTS, na.rm = TRUE), MINIMUM_PAYMENTS)
19     CREDIT_LIMIT = ifelse(is.na(CREDIT_LIMIT), median(CREDIT_LIMIT, na.rm = TRUE), CREDIT_LIMIT)
20   )
21
22 # -----
23 # 3：建模前處理（資料變形）
24 #
25 # [dplyr] 資料變形：移除 ID -> 取 Log 對數 -> Scale 標準化
26 data_model <- df_clean %>%
27   select(-CUST_ID) %>%
28   mutate_all(~log1p(.)) %>%
29   scale()
30
```

Demo 2 - 階層式分群實作



dplyr

cluster

factoextra

建立階層式分群模型與評估 → 判斷k值(分幾群)

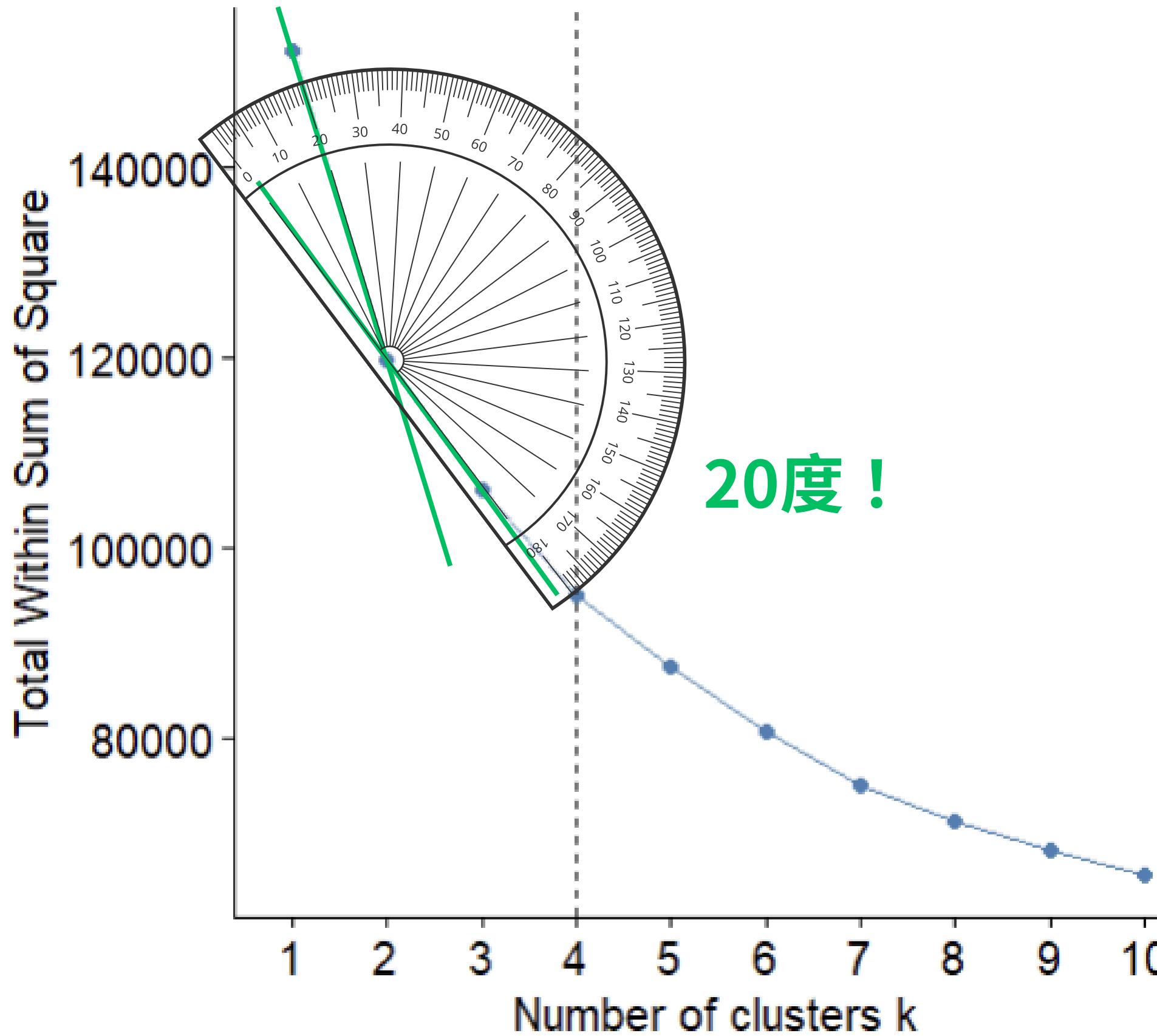
- **dist()**：計算客戶彼此之間在數學空間上的距離（歐式距離）
- **hclust()**：將距離近的資料點聚合，建立出完整的階層樹狀結構模型

利用 factoextra視覺化評估

- **fviz_nbclust()**：手肘法繪圖

```
31 # -----
32 # 4：建立階層式分群模型與評估
33 # -----
34 # [cluster] 計算距離與建立模型
35 # 1. 計算歐式距離
36 dist_mat <- dist(data_model, method = "euclidean")
37 # 2. 建立階層式分群 (Ward's Method)
38 hc_model <- hclust(dist_mat, method = "ward.D2")
39
40 # [factoextra] 繪製評估圖表
41 # 畫手肘圖 (Elbow Method) 來確認 k=4 是正確的
42 fviz_nbclust(data_model, FUN = hcut, method = "wss") +
43   geom_vline(xintercept = 4, linetype = 2) +
44   labs(title = "Elbow Method (Checking k=4)")
45
```

Elbow Method (Checking k=4)

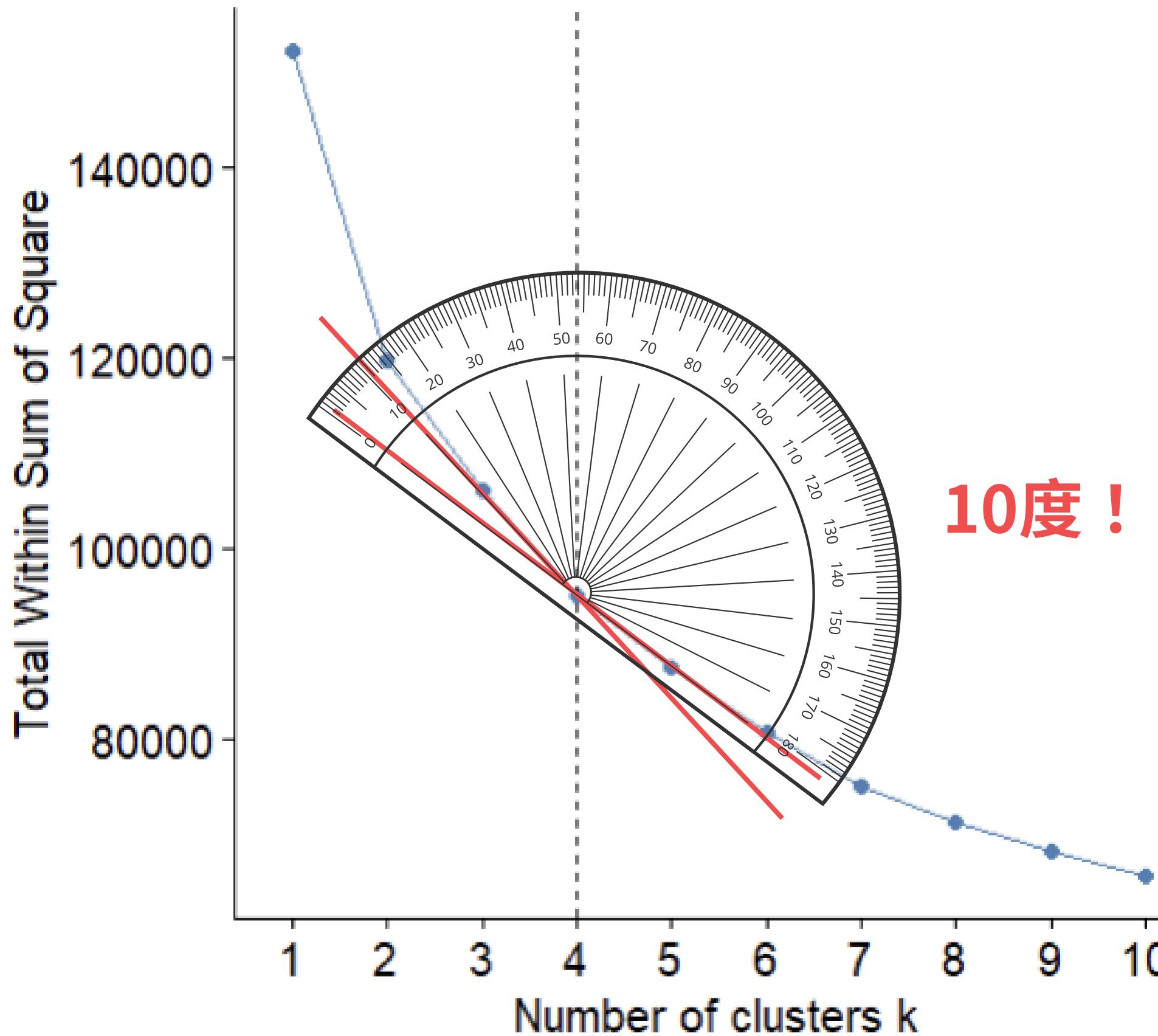


找下降速度開始明顯變慢的點

曲線在 $k=4$ 的位置出現明顯的轉折
(手肘)，這代表分成 4 群是 CP 值最
高的選擇，因此 $k = 4$

分成4群既能區分出明顯差異，
又不會讓群組變得過於破碎難以管理

Elbow Method (Checking k=4)

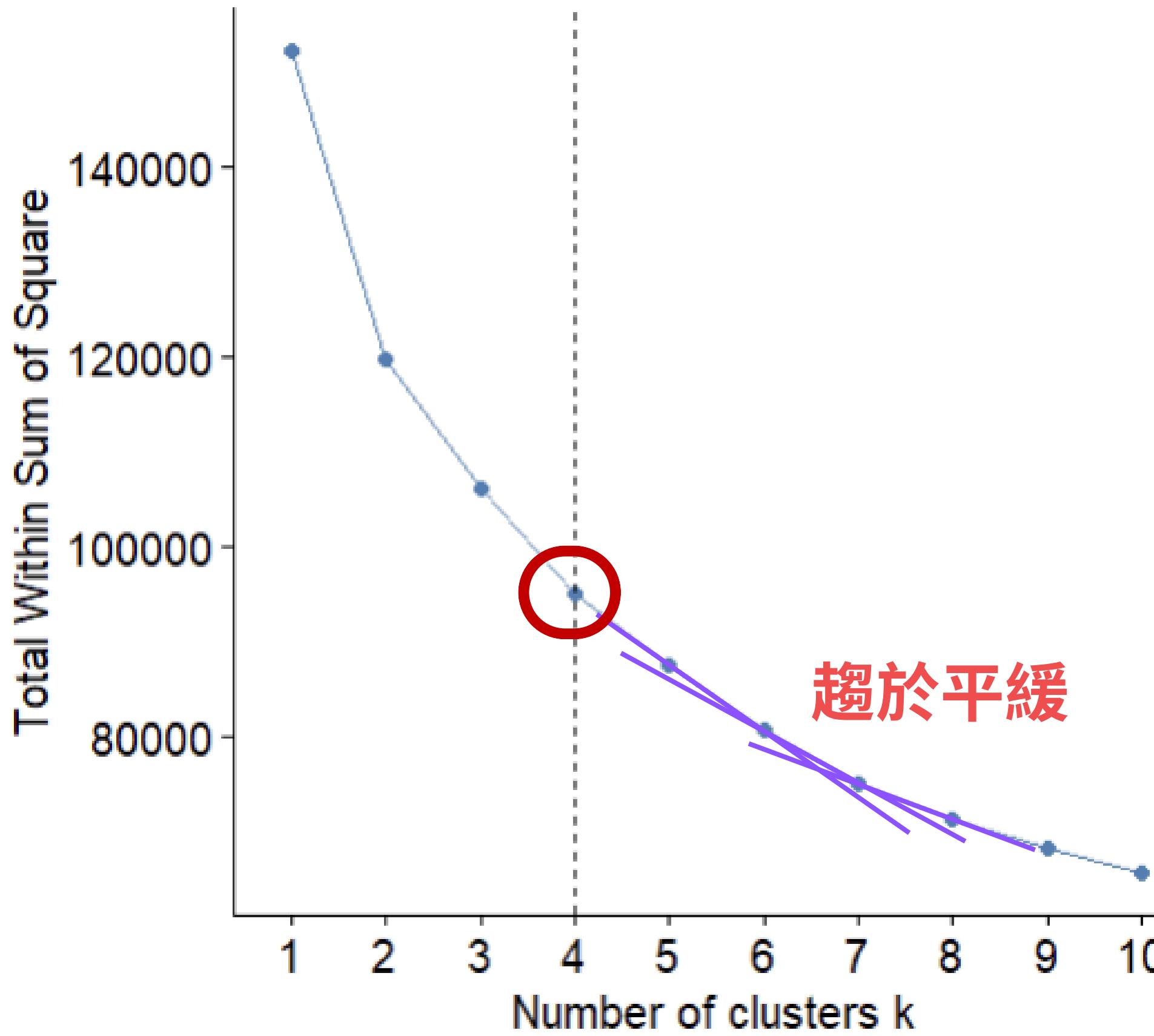


找下降速度開始明顯變慢的點

曲線在 $k=4$ 的位置出現明顯的轉折
(手肘)，這代表分成 4 群是 CP 值最
高的選擇，因此 $k = 4$

分成4群既能區分出明顯差異，
又不會讓群組變得過於破碎難以管理。

Elbow Method (Checking k=4)



找下降速度開始明顯變慢的點

曲線在 $k=4$ 的位置出現明顯的轉折
(手肘)，這代表分成 4 群是 CP 值最
高的選擇，因此 $k = 4$

分成4群既能區分出明顯差異，
又不會讓群組變得過於破碎難以管理。

Demo 2 - 階層式分群實作



dplyr

cluster

factoextra

結果分析 → 帶入k值，產出最終業務報表

```
46 # =====
47 # 5：產出最終業務報表
48 # =====
49 # [cluster] 根據模型進行切割 (k=4)
50 groups <- cutree(hc_model, k = 4)
51
52 # [dplyr] 將分群結果 (1,2,3,4) 貼回乾淨的原始資料
53 final_data <- df_clean %>%
54   select(-CUST_ID) %>%
55   mutate(Cluster = as.factor(groups))
56
57 # [dplyr] 製作業務報表 (Summarise)
58 cluster_profile <- final_data %>%
59   group_by(Cluster) %>%
60   summarise(
61     人數 = n(),
62     平均餘額 = mean(BALANCE),
63     平均消費金額 = mean(PURCHASES),
64     平均預借現金 = mean(CASH_ADVANCE),
65     平均分期消費 = mean(INSTALLMENTS_PURCHASES),
66     平均信用額度 = mean(CREDIT_LIMIT),
67     平均付款 = mean(PAYMENTS),
68     全額還款率 = mean(PRC_FULL_PAYMENT)
69   ) %>%
70   mutate(
71     # [dplyr] 金額類：取整數 (round 0)
72     平均餘額 = round(平均餘額, 1),
73     平均消費金額 = round(平均消費金額, 1),
74     平均預借現金 = round(平均預借現金, 1),
75     平均分期消費 = round(平均分期消費, 1),
76     平均信用額度 = round(平均信用額度, 1),
77     平均付款 = round(平均付款, 1),
78
79     # [dplyr] 百分比類：保留 2 位小數，避免變成 0
80     全額還款率 = round(全額還款率, 2)
81   )
82
```

- **cutree()** : 切割k組群組
- **group_by()** : 依照分群標籤將客戶分類
- **summarise()** : 計算各群的平均行為（消費、餘額等）

依據 k=4 將階層樹狀結構進行水平切割，取得最終分群結果

	Cluster	人數	平均餘額	平均消費金額	平均預借現金	平均分期消費	平均信用額度	平均付款	全額還款率
1	1	3385	950.3	1624.1	137.2	777.1	4454.0	1627.3	0.27
2	2	1999	2198.7	2.7	1995.2	0.3	4058.5	1743.0	0.05
3	3	2613	2367.0	1192.4	1582.0	351.2	5105.3	2275.9	0.08
4	4	953	215.1	377.7	183.0	136.5	3876.3	600.1	0.14

低風險

Cluster 1 – 穩定日常型客群

- 消費與餘額屬中低水準、使用模式穩定
- 預借現金低，還款紀律佳，全額還款率最高
- 多為一般上班族、日常生活型用卡者

最高風險

Cluster 2 – 高預借現金依賴族群

- 一般刷卡消費極低
- 大量依賴預借現金，全額還款率最低
- 多為短期資金需求高、財務壓力較大的客群

中高風險

Cluster 3 – 高消費但現金流緊縮族群

- 一般消費金額高，預借現金也偏高
- 還款行為較弱，全額還款率偏低
- 多為高消費力但現金流不穩定的客群

低中風險

Cluster 4 – 低活躍／低消費族群

- 整體用卡頻率低
- 消費、分期、預借現金皆不高
- 多為備用卡、偶爾使用卡的客戶

為什麼Demo 1和Demo 2的族群分析結果有些許差異呢？

主因：k-means與階層式分群的「運算邏輯」與「對群集形狀的假設」不同

	K-means 分群	階層式分群
策略	迭代優化：不斷修正中心點	貪婪合併：一步步合成，不可逆
群集形狀	傾向球狀、大小相近	取決於 Linkage (可長條/球狀)
隨機性	有 (受初始值影響)	無 (結果固定)
計算效率	快	慢

為什麼Demo 1和Demo 2的族群分析結果有些許差異呢？

且k-means在結果視覺化時，factoextra 會自動在後台對原始資料執行 **PCA (主成分分析)** 找出保留最多資訊的**兩個新軸**，也就是第21頁圖片中看到的 Dim1 (第一主成分) 和 Dim2 (第二主成分) 導致了視覺誤差，例如：

- **重疊 (Overlap)**：綠色群和藍色群的邊界好像有點重疊，實際上在高維空間中可能分得很開（例如在 Dim3 或 Dim4 的方向上是分開的），但因為被壓扁到 2D 平面上，看起來就像疊在一起
- **形狀扭曲**：就像把地球儀的3D壓成地圖的2D，邊緣的形狀會變形

ps. 想確認這兩個群集在統計上的相似度，而不依賴視覺，可使用 `adjustedRandIndex` (調整後蘭德指數) 來量化比較兩種分群結果的一致性 (此方法為Gemini提供)

```
library(mclust)
# 比較 K-means 分群結果與階層式分群結果的一致性 (0為完全不同，1為完全相同)
adjustedRandIndex(kmeans_result$cluster, hierarchical_result_clusters)
```

四、商業應用與結論

分析後的商業應用

結論與 Q&A



分析後的商業應用

客戶細分(customer segmentation)

- 透過分群分析，將原本龐大且複雜的客戶資料→轉化為具體的客群類型

行銷策略差異化 (Marketing Strategy)

- 穩定日常型客群
用卡與還款行為穩定，適合提供日常回饋以維持黏著度。
- 高預借現金依賴族群
預借現金比例高，應以風險控管與還款輔導為主。
- 高價值客群 (VIP 大戶)
消費金額高、貢獻度大，提供尊榮服務以提升忠誠度。
- 低活躍／低消費族群
使用頻率低，可透過喚醒優惠或降低行銷投入。

分析後的商業應用

風險控管與信用管理 (Risk Management)

- 調整信用額度
- 加強風險監控
- 提供理財或還款輔導

決策輔助與資源配置

- 管理層可依不同客群：
設定不同 KPI
制定差異化經營策略

建立長期客群監控機制

- 觀察客戶在不同群組間
的轉移

總結

1. 透過 k-means 與階層式分群
 - 辨識出不同顧客行為模式
 - 建立具商業意義的客群輪廓
2. 實作過程顯示：
 - 資料前處理與群數選擇非常關鍵
 - 分群結果具備實務可解釋性
3. 分群分析不只是理論工具，而是：
 - 行銷決策
 - 風險管理
 - 客戶經營的重要基礎

分工表

姓名	資料搜集 + 討論	製作簡報	程式實作	修正簡報
B11201015 王瑀晴				
B11201021 程潛威				
B11201106 張妮				
B11201108 盧宜婷				
B11201109 蔡碧芸				
B11201110 王思樺				



謝謝聆聽！

參考資料：

<https://chih-sheng-huang821.medium.com/機器學習-集群分析-k-means-clustering-e608a7fe1b43>

<https://kiwi-half.medium.com/k-means-自動選擇群數的方法-手肘法-輪廓係數法-clustering-半熟奇異果-kiwi-tech-fc2d882da917>