

# 會員輪廓自動化分群 說明文件

數據中心 數據策略

20230810

# Outlines

- 1** 資料索取與Python安裝
- 2** 初階－使用步驟與結果產出
- 3** 進階－檔案與選項設定細節
- 4** 整體程式流程

# Request Data in SQL environment

Take aws-redshift as example

## ■ *Program*

- 00\_MasterStore.sql
- 00\_Request.sql

\*Files in blue are manually set by users

Temporary Table for 時間區段  
後續串接與此表有關




```
1 -----
2 --Time: 2023-08-08; Jing-Hui Tong
3 --Request the dataset to classify member based on PMA
4 --Input Table:
5 --  setopdata.member_member_x,
6 --  datacenter.festival_config,
7 --  setopdata.pos_d_m_member,
8 --  setopdata.mstbda_store_m
9 --Output Table:
10 --  analysis.kmean_clustering_sample_v6 (L202)
11 -----
12
13
14 ---- Setting variables
15 --DROP TABLE #variables;
16
17 create temporary table variables AS (
18     select '2023-07-17'::date as startdate,      -- you need to set
19           '7':: int as howlong_day,             -- you need to set
20           (howlong_day-1)::int as inputday,
21           DATEADD(day, inputday, startdate)::date as enddate
22 );
```

# Basic Python Installation

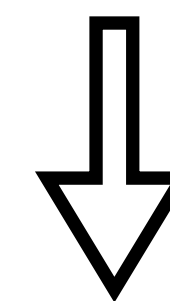
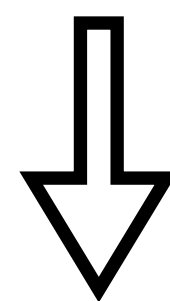
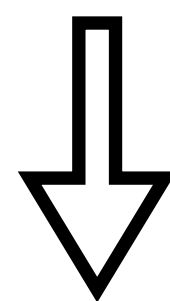
Website to download Anaconda: <https://www.anaconda.com/download/>  
(滑到最下面)

**According to your system**

## Anaconda Installers

 Windows	 Mac	 Linux
<b>Python 3.11</b> ↓ 64-Bit Graphical Installer (898.6 MB)	<b>Python 3.11</b> ↓ 64-Bit Graphical Installer (610.5 MB) ↓ 64-Bit Command Line Installer (612.1 MB) ↓ 64-Bit (M1) Graphical Installer (643.9 MB) ↓ 64-Bit (M1) Command Line Installer (645.6 MB )	<b>Python 3.11</b> ↓ 64-Bit (x86) Installer (1015.6 MB) ↓ 64-Bit (Power8 and Power9) Installer (473.8 MB) ↓ 64-Bit (AWS Graviton2 / ARM64) Installer (727.4 MB) ↓ 64-bit (Linux on IBM Z & LinuxONE) Installer (340.8 MB)

*\*Mac請選擇Graphical Installer*



**websites  
to refer**

<https://walker-a.com/archives/6260>

<https://www.datacamp.com/tutorial/installing-anaconda-mac-os-x>

<https://ivonblog.com/posts/linux-anaconda/>

# Outlines

- 1** 資料索取與Python安裝
- 2** 初階－使用步驟與結果產出
- 3** 進階－檔案與選項設定細節
- 4** 整體程式流程



# Step 0. Putting these files in the same folder

## ■ *Program*

- 01\_Clustering.py
- arguments.py
- autoclassify.py
- plotfigure.py
- preprocess.py
- checkfonts.py

## ■ *Config*

- category\_config.json
- pma\_config.csv
- store\_area\_config.csv

## ■ *Fonts*

- fonts/TaipeiSansTCBeta-Bold.ttf
- fonts/TaipeiSansTCBeta-Light.ttf
- fonts/TaipeiSansTCBeta-Regular.ttf

## ■ *Data*

- KmeanClustering\_sampledata\_v6.csv

\*Files in blue are manually set by users

# Step 1. The modules you need to install in Anaconda Prompt(for Windows)/Terminal (for MacOS/Linux)

- kmodes ➡ pip install kmodes
- kneed ➡ pip install kneed
- matplotlib ➡ pip -m pip install -U matplotlib
- sklearn ➡ pip install -U scikit-learn
- sklearn\_extra ➡ pip install scikit-learn-extra
- pandas ➡ pip install pandas
- 中文字體 ➡ *python checkfonts.py*

# Step 3. Execute Program

1. Open terminal/Anaconda Prompt
2. Enter the folder where you put the code
3. %*python 01\_Clustering.py -h*

Tips for entering the folder:

Asking ChatGPT →

- Windows: 在cmd 底下，如何進入特定的資料夾
- MacOS/Linux: 在terminal 底下，如何進入特定的資料夾

執行

選項說明與預設值

```
usage: python 01_Clustering.py --file [filename] --pma [no.] -N [number]/-A [options]
```

--file --pma -N/-A 為必選選項

options:

```
-h, --help                show this help message and exit
-f FILENAME, -F FILENAME, --file FILENAME
                           Specify the file you input
--pma PMA, --PMA PMA      Specify the PMA number to classify
-c CATEGORY, -C CATEGORY, --category CATEGORY
                           Select one or more features you want to include to classify. Options include: 'all', 'personal',
                           'rfm', 'area', 'purchasetime', 'prefer', 'calculated'. [Default is None]
-o, -O, --overwrite       Force the overwriting of pre-existing results. Default behaviour prompts for those that already
                           exist. Selecting overwrite and skip (ie, both flags) negate each other, and both are set to false
                           (every repeat is prompted). [Default is False]
-v, -V, --verbose         Specify to increase verbosity. [Default is False]
```

Cluster method:

分群方法選擇

Auto Classify Settings:

指定群數/自動分群選項

RFM level:

若分析特徵包含RFM，可選擇如何分等級方式、RFM是否只選兩個特徵(i.e. f & m)

Path:

檔案路徑設定

# Step 3. Execute Program (Continued)

4. %*python 01\_Clustering.py --file KmeanClustering\_sampledata\_v5.csv --pma 41 -A -V*

指令邏輯正確畫面

程式開始

```
|=====|
|           Executing Program           |
|=====|
|=====|
|           Data Preprocessing           |
|=====|
*****  Read columns: ['mid', 'pma_no_fin', 'qty', 'avg_qty']
*****  PMA: 41
*****  Begin to read the file.....
*****  Deal with object mid -- LabelEncoder
*****  Deal with object pma_no_fin -- LabelEncoder
```

程式結束

```
|=====|
|           Saving descriptive statistics of dataset           |
|=====|
*****  filename path : 分群結果路徑

A total of 2314868 records were processed in 135.112791 seconds in PMA: 41
```

指令邏輯錯誤畫面

Example 1

```
% python 01_Clustering.py --file KmeanClustering_sampledata_v5.csv --pma 41 -V -C qqk --method kmeans
usage: python 01_Clustering.py --file [filename] --pma [no.] -N [number]/-A [options]
01_Clustering.py: error: Method: kmeans is not in our options
```

>> Method should follow the description (—method); 此例為大小寫錯誤

Example 2

```
% python 01_Clustering.py --file KmeanClustering_sampledata_v5.csv --pma 41-A -V -C qqk
usage: python 01_Clustering.py --file [filename] --pma [no.] -N [number]/-A [options]
01_Clustering.py: error: argument --pma/--PMA: invalid int value: '41-A'
```

Example 3

```
% python 01_Clustering.py --file KmeanClustering_sampledata_v5.csv --pma 41 -A -V -C qqk,eee
usage: python 01_Clustering.py --file [filename] --pma [no.] -N [number]/-A [options]
01_Clustering.py: error: The category 'qqk' is not in category_config.json
```

>> 每個選項間應有空格; 正確: --pma[空格]41[空格]-A

\*orange line: error message

>> 欄位qqk 不在設定檔內



# Step 4. Results

程式放置的資料夾

\_\_pycache\_\_

00\_Documents

00\_MasterStore.sql

00\_Request.sql

01\_Clustering.ipynb

01\_Clustering.py

99\_Clustering-test.ipynb

arguments.py

autoclassify.py

category\_config.json

KmeanCluster...ledata\_v2.csv

KmeanCluster...ledata\_v3.csv

KmeanCluster...ledata\_v4.csv

KmeanCluster...ledata\_v5.csv

KmeanCluster...mpledata.csv

KmeanCluster...\_schema.xlsx

plotfigure.py

pma\_config.csv

PMA-38

PMA-41

preprocess.py

store\_area\_config.csv

PMA-41

Group\_00

Group\_01

Group\_02

Group\_03

Group\_04

Group\_05

Group\_06

Group\_07

Group\_08

Group\_09

Group\_10

pma\_41\_KMe...to\_cluster.png

pma\_41\_Overall.png

pma\_41.log

Group

00.Descriptiv...istics\_G02.csv

01.plot\_ratio.png

02.Detail\_info\_mid.csv

紅框檔案為自動儲存結果

Default path：程式存放位置。可藉由選項 --savepath [yourpath]更改

## [PMA-41] 各分類結果

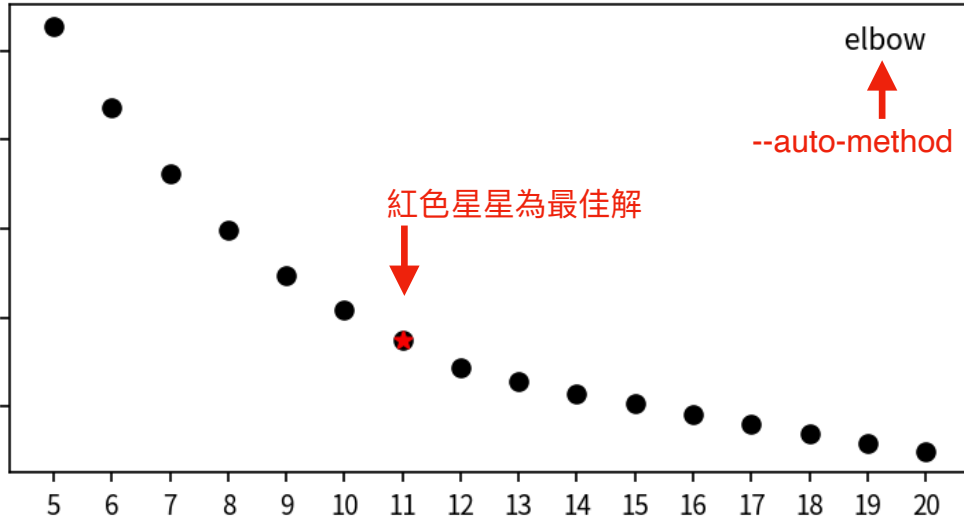
○ pma\_41.log

> log檔，紀錄程式執行選項設定與各群數量與比例

```
=====
Executing time: 2023-07-28 17:03:09.400491
PMA: 41
Datalength: 2314868
Features: [personal, purchasetime, prefer, rfm, calculated, area]
Cluster method: KMeans/Auto/elbow
Num. of Group: 11
=====
Group 0: 445884 (19.26%)
Group 1: 115931 (5.01%)
Group 2: 20123 (0.87%)
Group 3: 516943 (22.33%)
Group 4: 207066 (8.95%)
Group 5: 15900 (0.69%)
Group 6: 401715 (17.35%)
Group 7: 62880 (2.72%)
Group 8: 392092 (16.94%)
Group 9: 129973 (5.61%)
Group 10: 6361 (0.27%)
=====
```

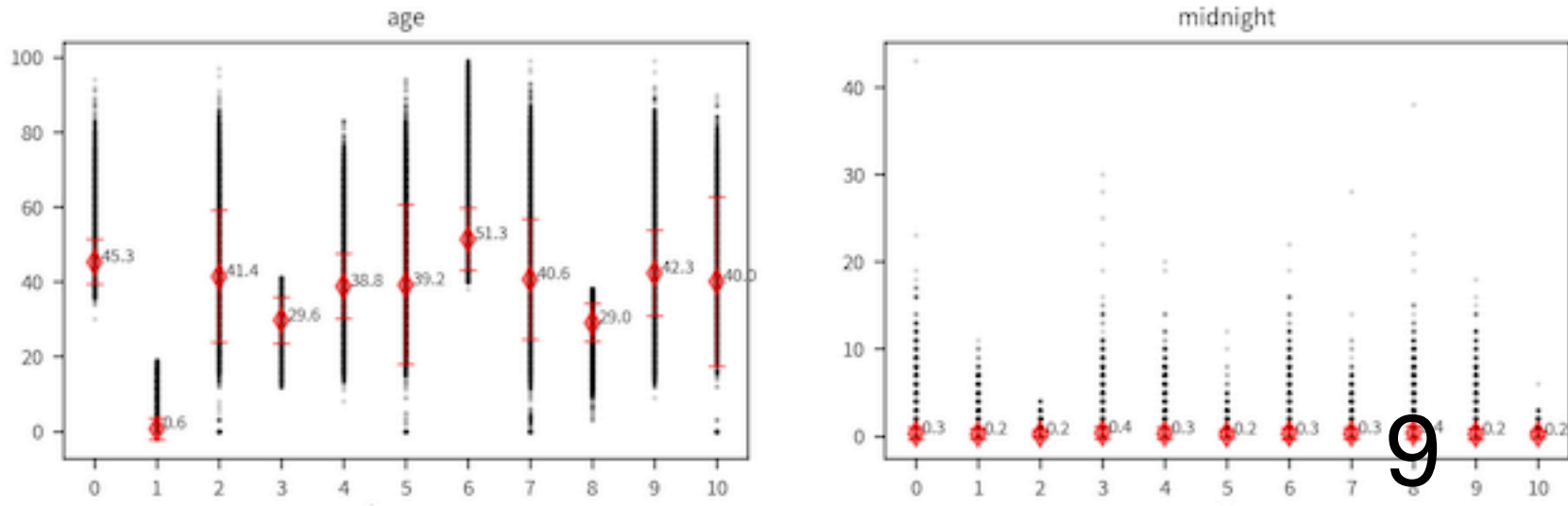
○ pma\_41\_KMeans\_auto\_cluster.png

> 自動決定分群分佈圖  
> 若為指定分群數目，則無此檔案



○ pma\_41\_Overall.png

> 各分析欄位分佈圖 (format: values)  
> 紅點為平均值



# Step 4. Results (Continued)

程式放置的資料夾

\_\_pycache\_\_

00\_Documents

00\_MasterStore.sql

00\_Request.sql

01\_Clustering.ipynb

01\_Clustering.py

99\_Clustering-test.ipynb

arguments.py

autoclassify.py

category\_config.json

KmeanCluster...ledata\_v2.csv

KmeanCluster...ledata\_v3.csv

KmeanCluster...ledata\_v4.csv

KmeanCluster...ledata\_v5.csv

KmeanCluster...mpledata.csv

KmeanCluster...\_schema.xlsx

plotfigure.py

pma\_config.csv

PMA-38

PMA-41

preprocess.py

store\_area\_config.csv

PMA-41

Group\_00

Group\_01

Group\_02

Group\_03

Group\_04

Group\_05

Group\_06

Group\_07

Group\_08

Group\_09

Group\_10

pma\_41\_KMe...to\_cluster.png

pma\_41\_Overall.png

pma\_41.log

Group

00.Descriptiv...istics\_G02.csv

01.plot\_ratio.png

02.Detail\_info\_mid.csv

紅框檔案為自動儲存結果

Default path：程式存放位置。可藉由選項 --savepath [yourpath]更改

## [Group\_02] 各群結果

00.Descriptive\_statistics\_G02.csv

> 各欄位敘述性統計值

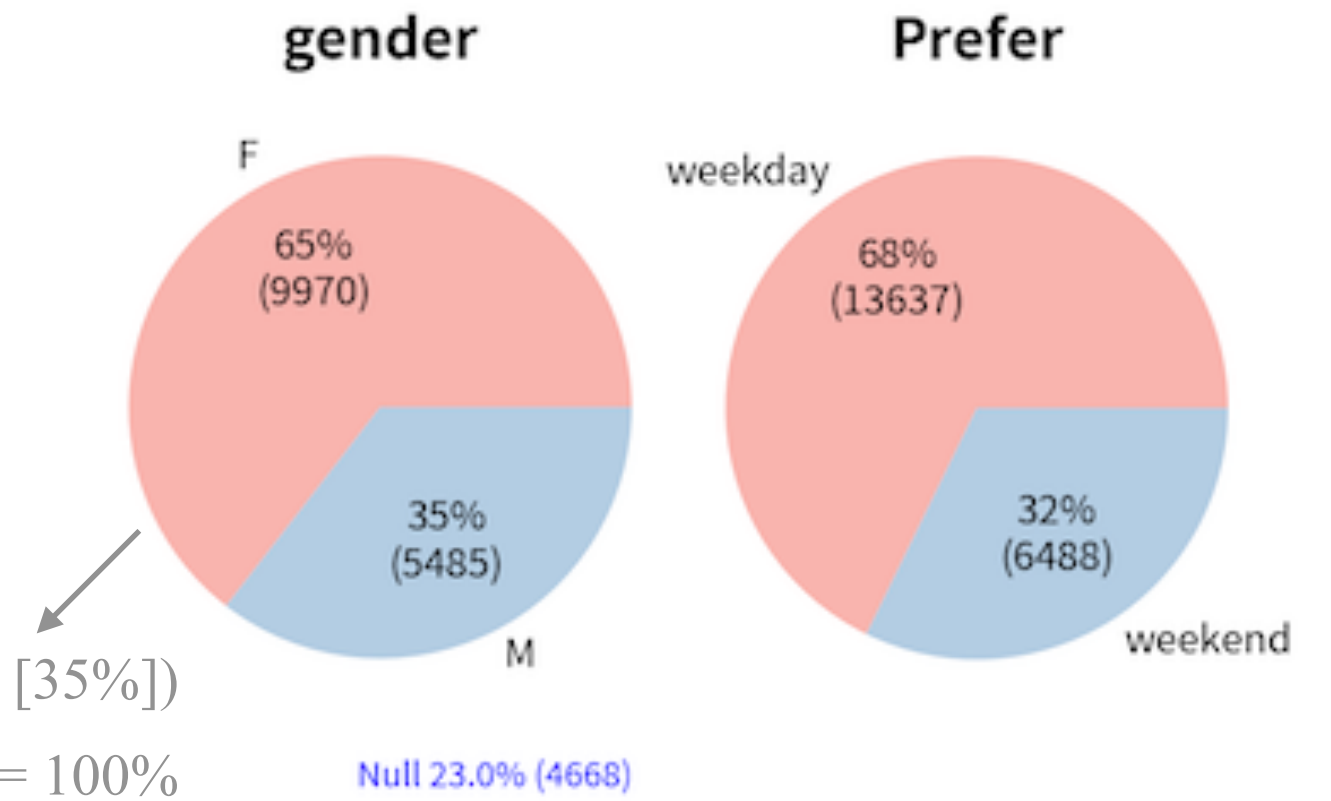
	A	B	C	D	E
1		pma_no_fin	qty	avg_qty	aov
2	mean	41	1.22511554	1.0512036	178.811079
3	std	0	0.70136344	0.3543205	13.8533934
4	min	41	1	1	129.33
5	25%	41	1	1	179
6	50%	41	1	1	179
7	75%	41	1	1	179
8	max	41	36	36	499

01.plot\_ratio.png

> 各分析欄位圓餅圖  
(format: object)

> Null數量以藍字標記

Null [23%]+gender [77%]\*(F [65%]+M [35%])  
= 100%



02.Detail\_info\_mid.csv

> 完整欄位串mid

	A	B	C	D
mid		pma_no_fin	qty	avg_qty
000c41a	la733c9	41	1	1
003357a	1172af3	41	1	1
000ebd1	fa5877	41	1	1
00180a1	a5fb20d	41	1	1
0018cb1	l535771	41	1	1
0046441	256dc8d	41	1	1





# Outlines

- 1** 資料索取與Python安裝
- 2** 初階－使用步驟與結果產出
- 3** 進階－檔案與選項設定細節
- 4** 整體程式流程



# Detail of Input File

(((!!))) Your analysis files need to be set up with the columns in category\_config.json.

## ○ KmeanClustering\_schema.xlsx

必選欄位名稱

不可缺失與更改



A	B	C	D	E
特徵	欄位名稱	說明	範例	每週更新
必選	mid	會員欄位	00000f5f157245539b3e8eb3f72ca7e9	
	pma_no_fin	大分類	23	
	qty	購買總數	1	
	avg_qty	平均購買數量( $qty/rfm\_frequency$ )	1	
calculated	aov	該類別平均客單價( $sum(mm\_sales)/qty$ )	89	
	cv_ratio	平均購買頻率	0.1	
	cv	顧客價值	8.9	
rfm	rfm_recency	最近一次消費天數	4	
	rfm_frequency	消費頻率( $count\ rec\_no$ )	1	
	rfm_monetary_sum	消費總額	89	
	rfm_monetary	消費金額	89	
prefer	weekday	偏好平日購買	1	
	weekend	偏好假日購買	0	
purchasetime	midnight	購買時段計數(21-04)	0	
	morning	購買時段計數(05-10)	0	
	noon	購買時段計數(11-14)	0	
	afternoon	購買時段計數(15-16)	0	
	night	購買時段計數(17-20)	1	
area	area_01	母商圈計數(住宅區)	0	
	area_02	母商圈計數(商業區)	0	
	area_03	母商圈計數(工業區)	0	
	area_04	母商圈計數(幹道型)	1	
	area_05	母商圈計數(文教區)	0	
	area_06	母商圈計數(醫院)	0	
	area_07	母商圈計數(交通轉運站)	0	
	area_09	母商圈計數(風景區)	0	
	area_10	母商圈計數(辦公商圈)	0	
	area_11	母商圈計數(外島)	0	
active	active_city	活躍縣市	新北市	
	active_region	活躍區域	北部	
personal	age	年齡	26	
	home_city	居住地	新北市	
	sex	性別	F	

# Detail of Config Setting

- category\_config.json  只接受json 格式

[default]

不可更改

最外層括弧

-C 輸入的特徵名稱

```
{
  "personal": {
    "key"      : ["gender", "age", "home_city"],
    "format"   : ["object", "values", "object"]
  },
  "purchasetime": {
    "key"      : ["midnight", "morning", "noon", "afternoon", "night"],
    "format"   : ["values", "values", "values", "values", "values"]
  },
  "prefer": {
    "key"      : ["weekday", "weekend"],
    "format"   : ["object", "object"]
  },
  "rfm" : {
    "key"      : ["rfm_recency", "rfm_frequency", "rfm_monetary"],
    "format"   : ["values", "values", "values"]
  },
  "calculated" : {
    "key"      : ["cv_ratio", "aov", "cv"],
    "format"   : ["values", "values", "values"]
  },
  "area" : {
    "key"      : ["area_01", "area_02", "area_03", "area_04", "area_05", "area_06", "area_07", "area_09", "area_10", "area_11"],
    "format"   : ["object", "object", "object", "object", "object", "object", "object", "object", "object", "object"]
  }
}
```



執行程式選了-C personal

分析時就會讀gender, age, home\_city這三個欄位

欲分析欄位

分析欄位格式，values(會有統計值)/object(以圓餅圖表示)



- RFM名稱必須包含"recency", "frequency", "monetary"

- 且按此順序"R" > "F" > "M"

- 會自動生成rfm\_level(object)



須以area為開頭，各商圈代號見[store\\_area\\_config.csv](#)



欲新增特徵，務必保持此資料格式(json)

# Detail of Config Setting

 檔名不要更改

## ○ pma\_config.csv

1	pma_no,pma_name
2	08,國際精品
3	09,i預購
4	11,生鮮蔬果
5	12,預購商品
6	13,地瓜類
7	14,麵食
8	15,健美機能品
9	16,食材調味品
10	17,霜淇淋
11	18,商品預售
12	19,紙、生理用品
13	23,糖果
14	25,熟料理
15	27,零食
16	28,鮮食麵
17	29,甜點
18	30,沙拉
19	31,玩具
20	33,百貨


 header不要更改

## ○ store\_area\_config.csv

1	master_area_no, master_area_name, cnt
2	01,住宅區,3246
3	02,商業區,471
4	03,工業區,320
5	04,幹道型,1826
6	05,文教區,307
7	06,醫院,109
8	07,交通轉運站,167
9	09,風景區,128
10	10,辦公商圈,226
11	11,外島,81

 header不要更改

Download on  
2023/07/26

 若分析的分類編號沒有在此檔案，會報錯！

```
% python 01_Clustering.py --file KmeanClustering_sampledat
a_v5.csv --pma 01 -A -V -O -C all
usage: python 01_Clustering.py --file [filename] --pma [no.] -N [number]/-A [options]
01_Clustering.py: error: PMA: 1 is not efficient pma.
```



# Detail of Parameters Setting: Method

```
-m METHOD, -M METHOD, --method METHOD
    Choose the method to classify. Options include:
    'KMeans', 'KPrototypes', 'KMedoids', 'KModes',
    'WKMeans'. For large dataset, 'KMeans' is faster than
    other methods. [Default is 'KMeans']

-w WEIGHT, -W WEIGHT, --weight WEIGHT
    When selecting 'WKMeans', one column in the dataset
    must be chosen as the weighting factor. [Default is
    None]
```

資料數量: 2,314,868  
指定分群數量: 10

## 執行時間測試

Method	Time
KMeans	2m06s
KModes	1h53m
KPrototypes	6h6m
KMedoids	Killed
Weight-KMeans	1m18s

- KMeans: 利用歐式距離計算算數平均值，資料形式只限Values，在此程式會把Object部分做LabelEncoder
- KModes: 只接受資料形式為Object
- KPrototypes: KMeans+KModes，可接受混合型數據(Values+Object)，執行時間久
- KMedoids: 利用歐式距離計算群內距離和最小的方式(i.e. 中位數)
- WKMeans: 對於每個欄位給予權重，權重須藉由-w 給予

## 執行時間測試 分群方法: KMeans

分群	PMA	Records	Auto/Manual	Time (s)
	38 (香菸)	774,397	Auto	217.428405
			Manual(10)	66.472927
	65 (米飯)	1,604,118	Auto	382.838203
			Manual(10)	113.588311
	41 (冷藏飲品)	2,314,868	Auto	395.461441
			Manual(10)	124.532646



# Detail of Parameters Setting: Select-Method

```
--auto-method SELECTMETHOD
```

Choose the method to find the best number of cluster. Options include: 'elbow', 'calinski-harabasz', 'davies-bouldin', 'silhouette'. Detail setting from website: <https://reurl.cc/M8mr9K> [Default is 'elbow']

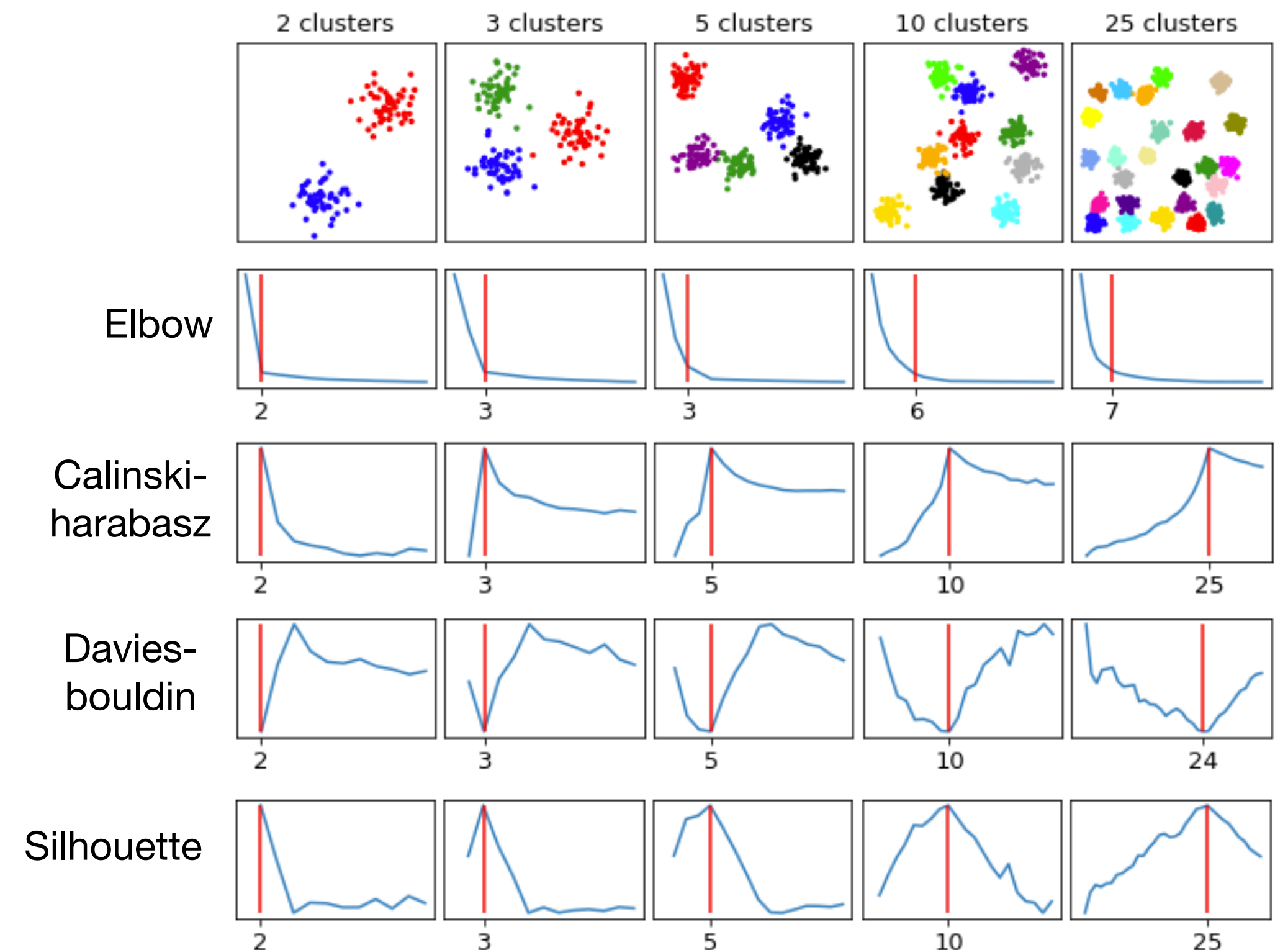
- Elbow: 尋找各群的距離平方和的轉折點
- Calinski-harabasz: 群與群間的離散度 / 群內點跟點的離散度，分數越高越好
- Davies-bouldin: 計算群與群之間的相似度，分數越低(相似度越低)越好
- Silhouette: 計算A點與同群樣本點的距離與其他群樣本點距離，分數越高越好，計算複雜度高(計算時間很久)

## Reference:

<https://towardsdatascience.com/are-you-still-using-the-elbow-method-5d271b3063bd>

<https://medium.com/@haataa/how-to-measure-clustering-performances-when-there-are-no-ground-truth-db027e9a871c>

## Method Performance



[https://github.com/smazzanti/are\\_you\\_still\\_using\\_elbow\\_method/blob/main/are-you-still-using-elbow-method.ipynb](https://github.com/smazzanti/are_you_still_using_elbow_method/blob/main/are-you-still-using-elbow-method.ipynb)

# Detail of Parameters Setting: RFM level

**R**ecency 最近一次消費

**F**requency 消費頻率

**M**onetary 消費金額

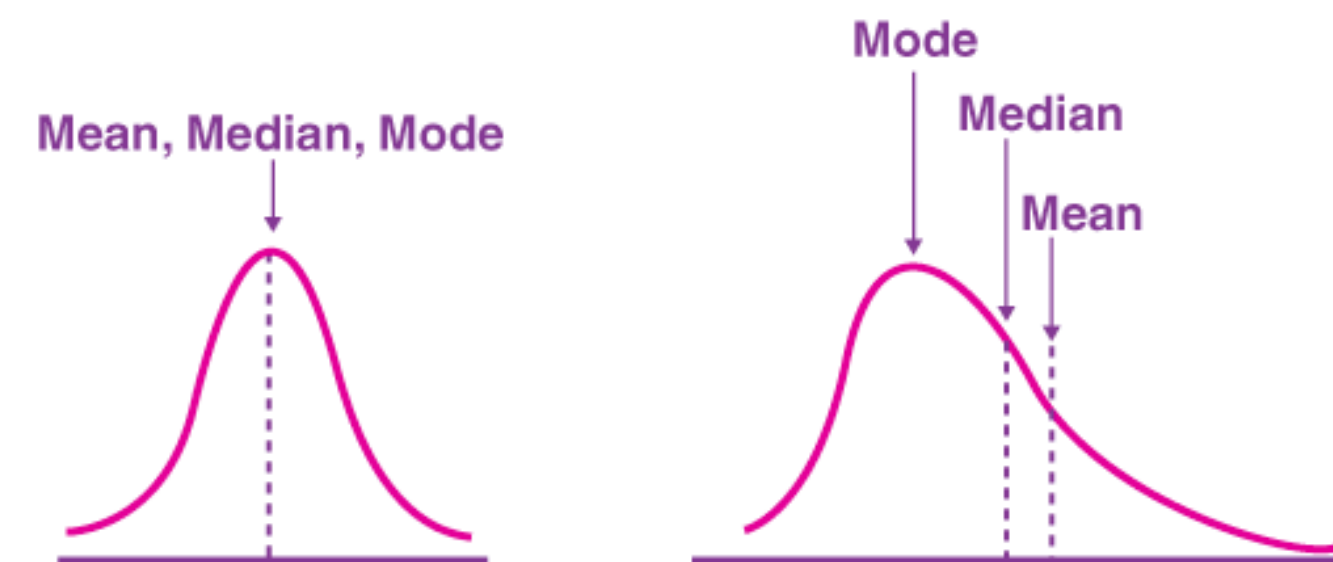
```
RFM level:
***** If 'rfm' in category, settings how to distinguish the RFM level

--rfm-level RFM_LEVEL
    Selecting how to distinguish the level. Options include:
    'median', 'mode', 'average'. [Default is 'mode']

--rfm-select
    Choosing whether three elements to classify or not. [Default
    is False]

--rfm-select-two SELECT_TWO
    Choosing whether two elements to classify. Options include:
    'r', 'f', 'm'. [Default is None]
```

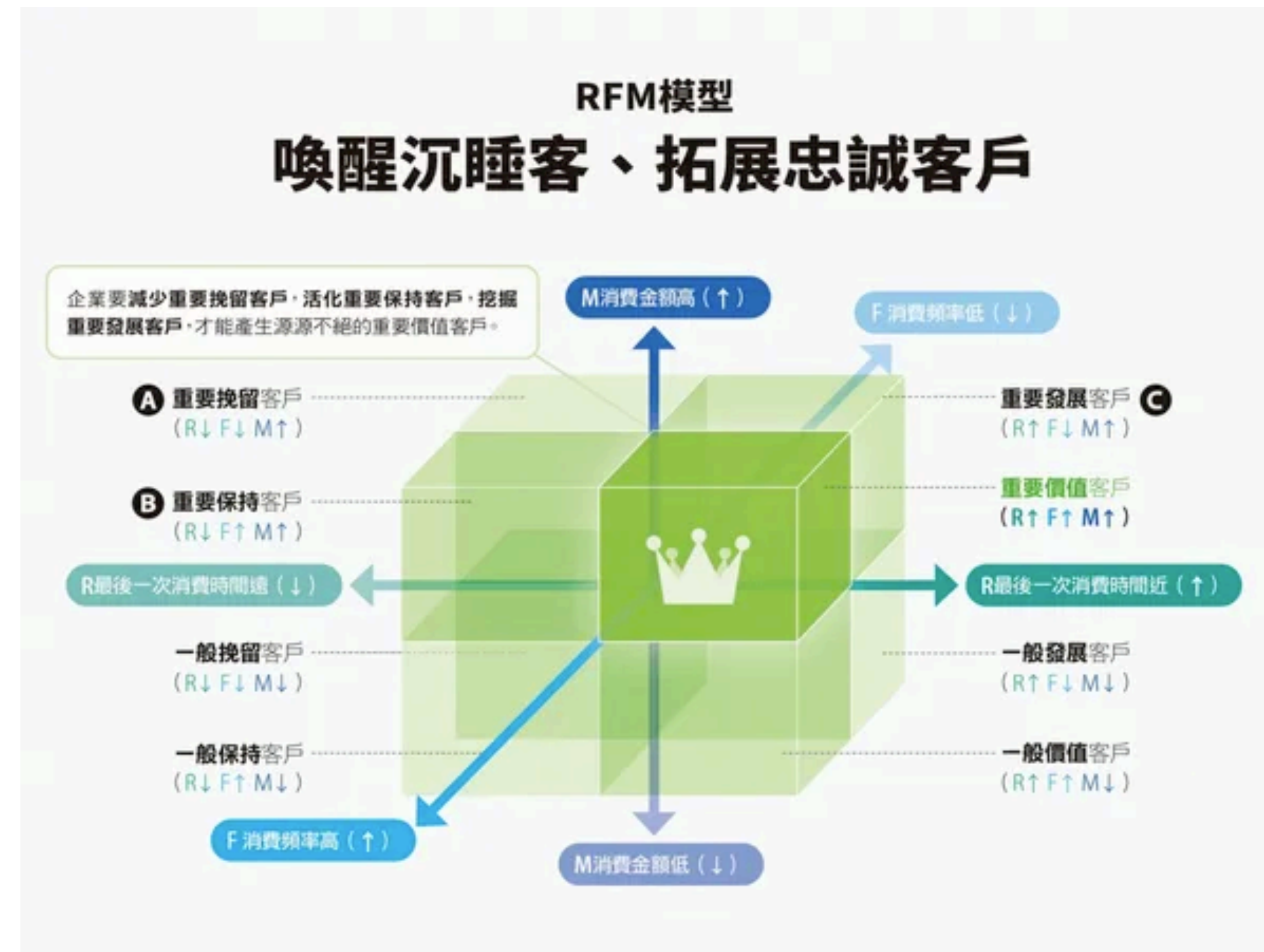
- rfm-level: 以平均(average)、中位數(median)或眾數(mode)來區分RFM等級



- rfm-select-two: 若僅分析兩項指標，必選 --rfm-select。原先三個指標取其中兩個，使用方法為: --rfm-select-two f,m，以避免誤下指令

Reference:

<https://www.managertoday.com.tw/articles/view/60050>

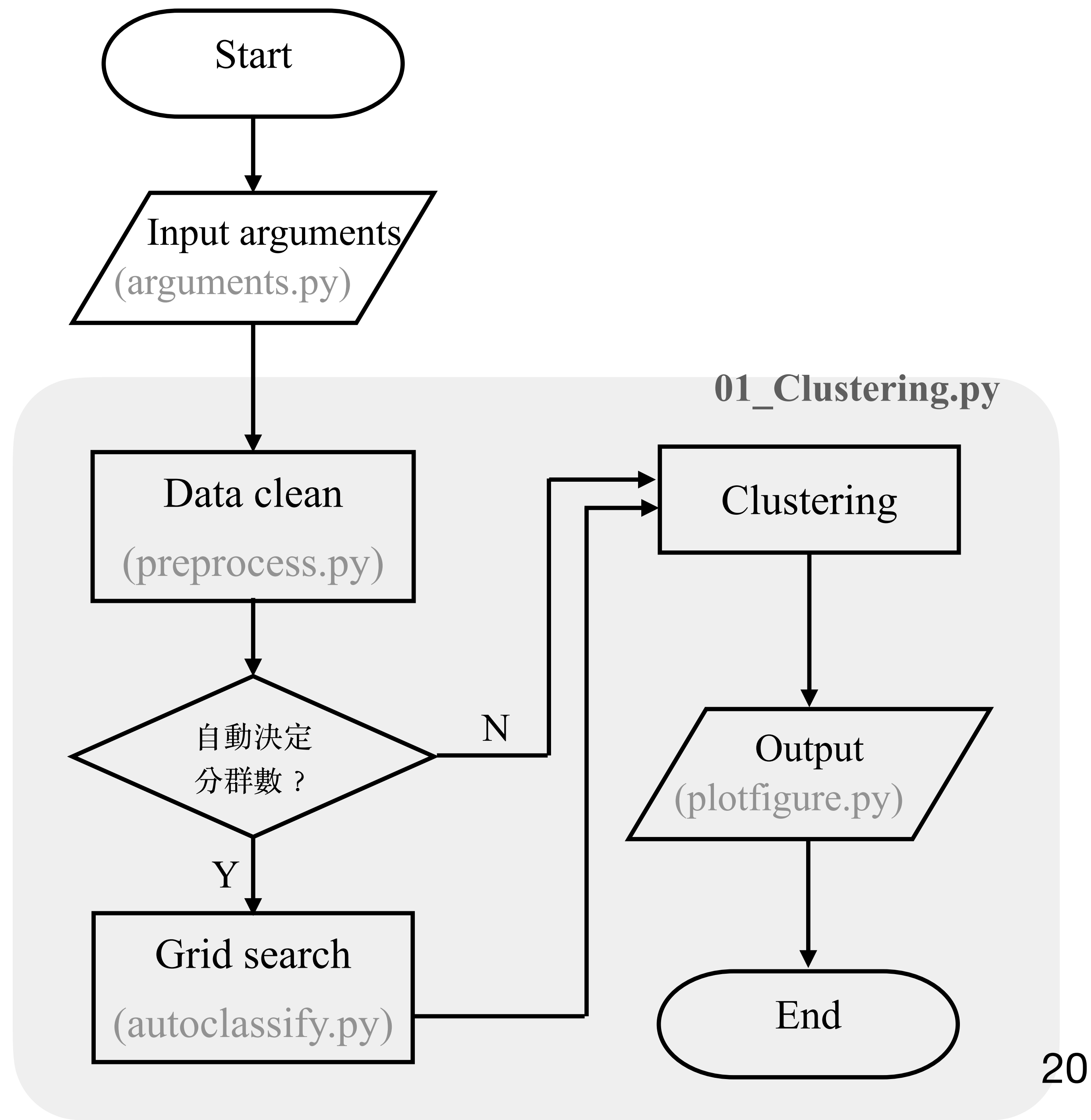




# Outlines

- 1** 資料索取與Python安裝
- 2** 初階－使用步驟與結果產出
- 3** 進階－檔案與選項設定細節
- 4** 整體程式流程

# Detail of Program Setting



- 專案開發環境: MacOS v12.6.4, Python 3.11.3
- 使用套件版本: kmodes v0.12.2, kneed v0.8.5, matplotlib v3.7.1  
numpy v1.24.2, pandas v1.5.3, scikit-learn v1.2.2,  
scikit-learn-extra v0.3.0, pip v23.1.2