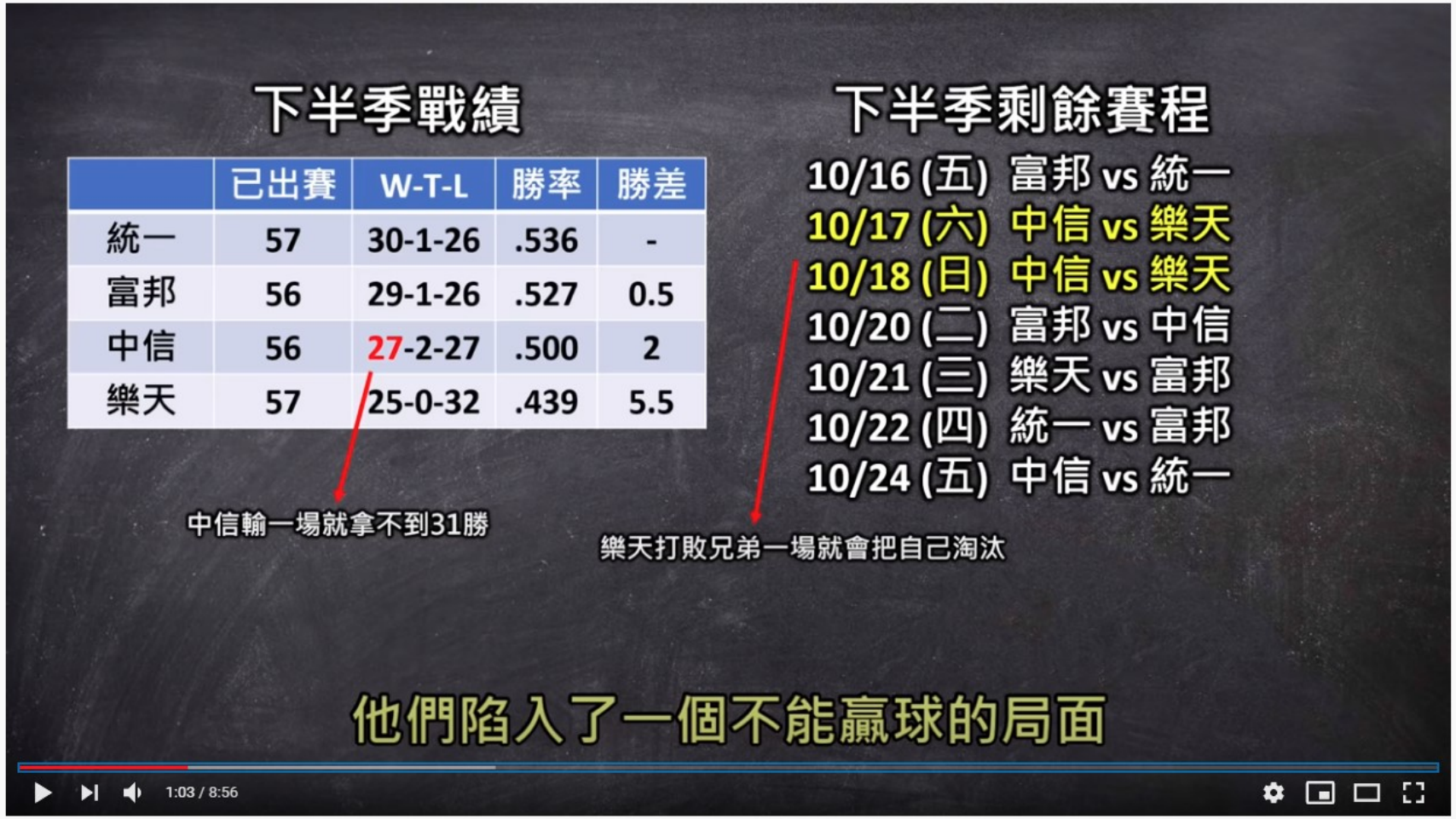
運動數據分析專題 I

中華職棒季後賽晉級分析

2020 中華職棒季後賽晉級分析

- 2020 年中華職棒下半季剩下最後 7 場比賽，當時 YouTuber 台南Josh 發布了一部影片，影片中提到一個有趣的狀態，就是樂天隊剩下兩場對中信隊的比賽「不能贏」，否則就會讓自己無緣晉級季後賽來爭取年度總冠軍。
- 職業運動中，在年度賽事的末期經常需要針對剩餘賽事進行分析，以掌握晉級與否和對戰隊伍。
- 本專題就讓我們撰寫一個程式，分析中職 2020 年最後 7 場比賽的勝負將如何影響各隊晉級。

2020 中華職棒季後賽晉級分析



	已出賽	W-T-L	勝率	勝差
統一	57	30-1-26	.536	-
富邦	56	29-1-26	.527	0.5
中信	56	27-2-27	.500	2
樂天	57	25-0-32	.439	5.5

#中華職棒 #CPBL #不要想著贏要想著不能贏
故意輸球才能打季後賽？中華職棒的賽制是在鼓勵放水？【Josh聊棒球】
175,926 views • Oct 15, 2020

https://youtu.be/6Vgn82anQhs

4.1K 89 SHARE SAVE

2020 中華職棒賽制

新聞 賽程 成績看板 球隊戰績 數據統計 球隊 球員 票務 商城 Eng 線上文物館 CPBL TV APP

ABOUT 關於中職

聯盟簡介 | 聯盟組織架構 | 職棒大事記 | 球迷成長數

聯盟組織架構

聯盟目前的架構是由各球團共同組成，各球團們對職棒事務的推動有絕對的自主參與權，且保有各自企業體的特色，在比賽上彼此競爭全力爭勝，以呈現球迷一個「真正的」職業競賽。目前推動發展的「責任場制」，將賦予各球團更多的責任，包括票房的推廣、球迷的經營，強調地域屬性，與母企業體結合，將企業文化融合呈現於球賽之中，未來將朝向「主客場制」的目標邁進。

聯盟的角色，未來將以辦理比賽相關業務為主，站在推廣棒球運動的立場而言，「職業棒球」的蓬勃發展，勢將成為所有業餘球員努力奮鬥的目標，且亦將帶動三級學生棒球的穩定成長。中華職業棒球大聯盟因扛負著身為棒球發展火車頭的重大責任，所以更應努力地辦妥每一場比賽，以便將場場精彩且細緻的職業競賽，呈現予所有的球迷共賞。

一. 賽制

一. 賽制

目前中華職棒共有四支球隊，每年球季約自三月至十月間，賽程進行三十週，每隊每球季各打一百二十場，全年共計二百四十場的例行賽，賽程採上、下半球季。當年度上、下半球季第一名為不同隊時，由上、下半球季第一名兩隊進行年度總冠軍賽，採七戰四勝制。當年度上、下半球季第一名為同一隊時，由全季勝率第二名與第三名球隊進行季後賽，採五戰三勝制。季後賽獲勝球隊再與年度勝率第一名球隊進行總冠軍賽，採七戰四勝制，但第一名先勝一場。若全年度勝率第一名球隊未取得上下半季任一季第一時，由上下半球季第一之兩隊中，全季勝率較低之球隊，與全季勝率第一之球隊，進行季後賽，採五戰三勝制。季後賽獲勝球隊再與上下半球季第一之兩隊中，全季勝率較高之球隊，進行總冠軍賽，採七戰四勝制。每年上、下半球季中並舉辦一場紅白明星對抗賽，如此的賽制勢將帶給球迷們一整年精采刺激的比賽。

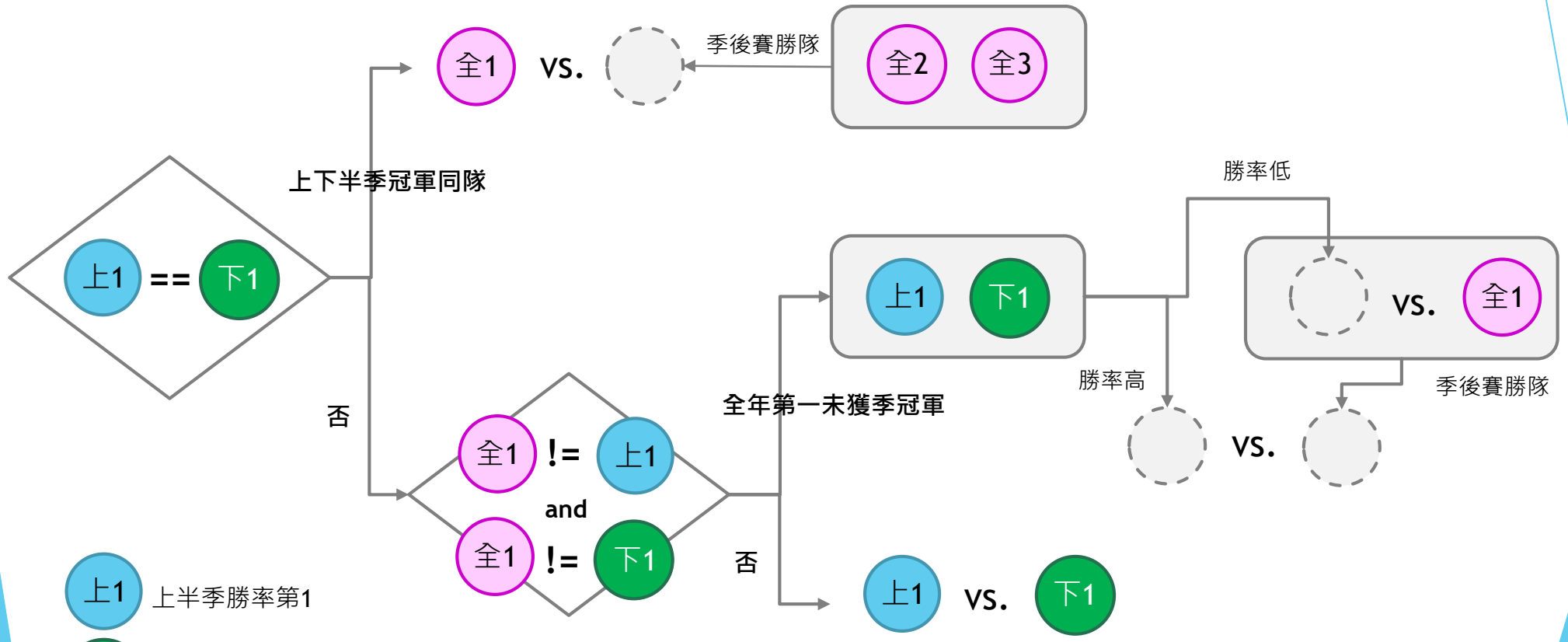
2020 中華職棒賽制

- 關鍵數據：勝率
  - 全年勝率、上半季勝率、下半季勝率

一. 賽制

目前中華職棒共有四支球隊，每年球季約自三月至十月間，賽程進行三十週，每隊每球季各打一百二十場，全年共計二百四十場的例行賽，賽程採上、下半球季。當年度上、下半球季第一名為不同隊時，由上、下半球季第一名兩隊進行年度總冠軍賽，採七戰四勝制。當年度上、下半球季第一名為同一隊時，由全季勝率第二名與第三名球隊進行季後賽，採五戰三勝制。季後賽獲勝球隊再與年度勝率第一名球隊進行總冠軍賽，採七戰四勝制，但第一名先勝一場。若全年度勝率第一名球隊未取得上下半季任一季第一時，由上下半球季第一之兩隊中，全季勝率較低之球隊，與全季勝率第一之球隊，進行季後賽，採五戰三勝制。季後賽獲勝球隊再與上下半球季第一之兩隊中，全季勝率較高之球隊，進行總冠軍賽，採七戰四勝制。每年上、下半球季中並舉辦一場紅白明星對抗賽，如此的賽制勢將帶給球迷們一整年精采刺激的比賽。

2020 中華職棒賽制



上1 上半季勝率第1

下1 下半季勝率第1

全1 全年度勝率第1

一. 賽制

目前中華職棒共有四支球隊，每年球季約自三月至十月間，賽程進行三十週，每隊每球季各打一百二十場，全年共計二百四十場的例行賽，賽程採上、下半球季。當年度上、下半球季第一名為不同隊時，由上、下半球季第一名兩隊進行年度總冠軍賽，採七戰四勝制。當年度上、下半球季第一名為同一隊時，由全季勝率第二名與第三名球隊進行季後賽，採五戰三勝制。季後賽獲勝球隊再與年度勝率第一名球隊進行總冠軍賽，採七戰四勝制，但第一名先勝一場。若全年度勝率第一名球隊未取得上下半季任一季第一時，由上下半球季第一之兩隊中，全季勝率較低之球隊，與全季勝率第一之球隊，進行季後賽，採五戰三勝制。季後賽獲勝球隊再與上下半球季第一之兩隊中，全季勝率較高之球隊，進行總冠軍賽，採七戰四勝制。每年上、下半球季中並舉辦一場紅白明星對抗賽，如此的賽制勢將帶給球迷們一整年精采刺激的比賽。

專題子任務說明

- 任務一
  - 給定比賽勝/和/負結果，計算各球隊勝率，並排序輸出。
- 任務二
  - 依據各球隊勝率及中職規則，輸出季後賽結果。
- 任務三
  - 給定比賽勝/和/負結果以及最後數場對戰組合，分析球隊晉級總冠軍戰的可能性。

程式先備知識

- 範例程式中會使用到的程式知識
  - `if` 和 `for` 句型
  - 列表 (`list`) 資料型態
    - 一維、二維、list comprehension
    - `sort()` 操作與 lambda
  - 字典 (`dict`) 資料型態
  - 函式
    - 函式的撰寫與呼叫

子任務一：勝率、季冠軍與年度冠軍

- 已知資訊：
  - 目前各球隊的 [勝, 和, 負] 場數
    - 以一個字典儲存，鍵為球隊名稱，值為列表，內存 [勝, 和, 負] 場數。

```
上半季 = {'中信':[37, 0, 23], '富邦':[23, 0, 37], '統一':[26, 0, 34], '樂天':[34, 0, 26]}
下半季 = {'中信':[30, 2, 28], '富邦':[31, 1, 28], '統一':[32, 1, 27], '樂天':[25, 0, 35]}
```

- 任務：
  - 計算各隊上半季、下半季和全年度的勝率。
  - 依勝率排序後輸出。

Python 支援中文的變數與函式名稱，此處為了方便對照規則，將變數和函式以中文命名。

子任務一：勝率、季冠軍與年度冠軍

```
勝場, 平局, 敗場 = 0, 1, 2

上半季 = {'中信':[37, 0, 23], '富邦':[23, 0, 37], '統一':[26, 0, 34], '樂天':[34, 0, 26]}
下半季 = {'中信':[30, 2, 28], '富邦':[31, 1, 28], '統一':[32, 1, 27], '樂天':[25, 0, 35]}


上半季勝率, 下半季勝率, 全年勝率 = [], [], []
for 球隊 in 上半季:
    上半勝, 上半場數 = 上半季[球隊][勝場], 上半季[球隊][勝場]+上半季[球隊][敗場]
    下半勝, 下半場數 = 下半季[球隊][勝場], 下半季[球隊][勝場]+下半季[球隊][敗場]
    上半季勝率.append([球隊, round(上半勝/上半場數,3)])
    下半季勝率.append([球隊, round(下半勝/下半場數,3)])
    全年勝率.append([球隊, round((上半勝+下半勝)/(上半場數+下半場數),3)])

for e in [上半季勝率, 下半季勝率, 全年勝率]:
    e.sort(key=lambda x: x[1], reverse=True)
    for t in e:
        print(t[0], t[1])
    print()
```

上 [['中信', 0.617], ['富邦', 0.383], ['統一', 0.433], ['樂天', 0.567]]
下 [['中信', 0.517], ['富邦', 0.525], ['統一', 0.542], ['樂天', 0.417]]
全 [['中信', 0.568], ['富邦', 0.454], ['統一', 0.487], ['樂天', 0.492]]

排序後

上 [['中信', 0.617], ['樂天', 0.567], ['統一', 0.433], ['富邦', 0.383]]
下 [['統一', 0.542], ['富邦', 0.525], ['中信', 0.517], ['樂天', 0.417]]
全 [['中信', 0.568], ['樂天', 0.492], ['統一', 0.487], ['富邦', 0.454]]

```
中信 0.617
樂天 0.567
統一 0.433
富邦 0.383

統一 0.542
富邦 0.525
中信 0.517
樂天 0.417

中信 0.568
樂天 0.492
統一 0.487
富邦 0.454
```

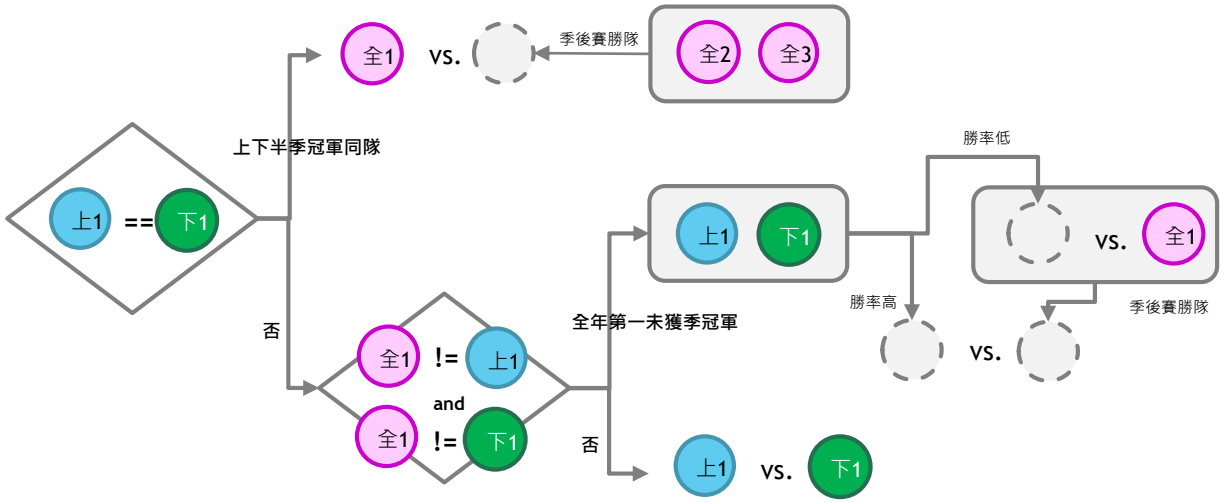
子任務二：晉級規則

- 若已知上半季、下半季和全年勝率，可依規則得出晉級結果。

```
第一, 第二, 第三 = 0, 1, 2
球隊名稱, 勝率值 = 0, 1

def 列印晉級結果(上半季勝率, 下半季勝率, 全年勝率):
    上1 = 上半季勝率[第一][球隊名稱]
    下1 = 下半季勝率[第一][球隊名稱]
    下2 = 下半季勝率[第二][球隊名稱]
    全1 = 全年勝率[第一][球隊名稱]
    全2 = 全年勝率[第二][球隊名稱]
    全3 = 全年勝率[第三][球隊名稱]

    print('季後賽:', end=' ')
    if 上1 == 下1:
        print(上1, 'vs. (', 全2, ',', 全3, ')')
    else:
        if 全1!=上1 and 全1!=下1:

            上1全年勝率 = 下1全年勝率 = 0
            for e in 全年勝率:
                if e[球隊名稱] == 上1:
                    上1全年勝率 = e[勝率值]
                elif e[球隊名稱] == 下1:
                    下1全年勝率 = e[勝率值]

            if 上1全年勝率>下1全年勝率:
                print(上1, 'vs. (', 全1, ',', 下1, ')')
            else:
                print(下1, 'vs. (', 全1, ',', 上1, ')')
        else:
            if 下半季勝率[第一][勝率值]!=下半季勝率[第二][勝率值]:
                print(上1, 'vs.', 下1)
            else:
                print(上1, 'vs. 加賽(', 下1, ',', 下2, ')')

main()
```

排序後

```
上 [['中信', 0.617], ['樂天', 0.567], ['統一', 0.433], ['富邦', 0.383]]
下 [['統一', 0.542], ['富邦', 0.525], ['中信', 0.517], ['樂天', 0.417]]
全 [['中信', 0.568], ['樂天', 0.492], ['統一', 0.487], ['富邦', 0.454]]
```



子任務二：晉級規則

```
勝場, 平局, 敗場 = 0, 1, 2
第一, 第二, 第三 = 0, 1, 2
球隊名稱, 勝率值 = 0, 1

def main():

    上半季 = {'中信':[37, 0, 23], '富邦':[23, 0, 37], '統一':[26, 0, 34], '樂天':[34, 0, 26]}
    下半季 = {'中信':[30, 2, 28], '富邦':[31, 1, 28], '統一':[32, 1, 27], '樂天':[25, 0, 35]}

    上半季勝率, 下半季勝率, 全年勝率 = [], [], []
    for 球隊 in 上半季:
        上半勝, 上半場數 = 上半季[球隊][勝場], 上半季[球隊][勝場]+上半季[球隊][敗場]
        下半勝, 下半場數 = 下半季[球隊][勝場], 下半季[球隊][勝場]+下半季[球隊][敗場]
        上半季勝率.append([球隊, round(上半勝/上半場數,3)])
        下半季勝率.append([球隊, round(下半勝/下半場數,3)])
        全年勝率.append([球隊, round((上半勝+下半勝)/(上半場數+下半場數),3)])

    for e in [上半季勝率, 下半季勝率, 全年勝率]:
        e.sort(key=lambda x: x[1], reverse=True)
        for t in e:
            print(t[0], t[1])
        print()

    列印晉級結果(上半季勝率, 下半季勝率, 全年勝率)

main()
```

```
中信 0.617
樂天 0.567
統一 0.433
富邦 0.383

統一 0.542
富邦 0.525
中信 0.517
樂天 0.417

中信 0.568
樂天 0.492
統一 0.487
富邦 0.454

季後賽: 中信 vs. 統一
```

若下半季改為
'中信':[35, 2, 23]

```
中信 0.617
樂天 0.567
統一 0.433
富邦 0.383

中信 0.603
統一 0.542
富邦 0.525
樂天 0.417

中信 0.61
樂天 0.492
統一 0.487
富邦 0.454

季後賽: 中信 vs. ( 樂天 , 統一 )
```

子任務三：季後賽晉級分析

- 如何窮舉 7 場比賽的勝負結果？
  - 每場比賽有兩支球隊參與。
  - 假設以 0 代表客場隊伍，1 代表主場隊伍。

```
剩餘比賽 = [['富邦', '統一'],['中信', '樂天'],['中信', '樂天'],
            ['富邦', '中信'],['樂天', '富邦'],['統一', '富邦'],
            ['中信', '統一']]
```

  - 000000 代表 7 場比賽都是客隊獲勝，即勝隊為：
    - 富邦、中信、中信、富邦、樂天、統一、中信
  - 101001 代表 7 場比賽的勝隊為：
    - 統一、中信、樂天、富邦、樂天、統一、統一

子任務三：季後賽晉級分析

- 如何窮舉 7 場比賽的勝負結果？
  - 由前頁可知，7 場比賽的勝負結果為 7 個 {0, 1} 所組成。
    - 為簡化可能情況，這 7 場比賽我們假設不會和局。
  - 借用二進位數字系統的知識，我們知道數字 0, 1, ..., 127 這 128 個數字的二進位表示法，恰好就是 7 個 {0, 1} 所能構成的所有組合 ($2^7 = 128$)。

```
for d in range(2**7):
    結果 = [int(e) for e in '{:07b}'.format(d)]
    print(d, 結果)
```

```
0 [0, 0, 0, 0, 0, 0, 0]
1 [0, 0, 0, 0, 0, 0, 1]
2 [0, 0, 0, 0, 0, 1, 0]
3 [0, 0, 0, 0, 0, 1, 1]

[省略]

125 [1, 1, 1, 1, 1, 0, 1]
126 [1, 1, 1, 1, 1, 1, 0]
127 [1, 1, 1, 1, 1, 1, 1]
```

另一種作法是利用 itertools 中的 permutations。

子任務三：季後賽晉級分析

```
def main():
    場數 = len(剩餘比賽)
    for d in range(2**場數):
        結果 = [int(e) for e in f'{{:0{場數}b}}'.format(d)]

        下半季預測 = deepcopy(下半季)
        比賽勝隊列表 = []
        for g in range(場數):
            w = 結果[g]
            勝隊, 敗隊 = 剩餘比賽[g][w], 剩餘比賽[g][(w+1)%2]
            比賽勝隊列表.append(勝隊)
            下半季預測[勝隊][勝場] += 1
            下半季預測[敗隊][敗場] += 1

        上半季勝率, 下半季勝率, 全年勝率 = [], [], []
        for 球隊 in 上半季:
            上半勝, 上半場數 = 上半季[球隊][勝場], 上半季[球隊][勝場]+上半季[球隊][敗場]
            下半勝, 下半場數 = 下半季預測[球隊][勝場], 下半季預測[球隊][勝場]+下半季預測[球隊][敗場]
            上半季勝率.append([球隊, round(上半勝/上半場數,3)])
            下半季勝率.append([球隊, round(下半勝/下半場數,3)])
            全年勝率.append([球隊, round((上半勝+下半勝)/(上半場數+下半場數),3)])

        for e in [上半季勝率, 下半季勝率, 全年勝率]:
            e.sort(key=lambda x: x[1], reverse=True)

        print('可能性', d+1, 比賽勝隊列表, end=' => ')
        列印晉級結果(上半季勝率, 下半季勝率, 全年勝率)
```

```
from copy import deepcopy

勝場, 平局, 敗場 = 0, 1, 2
第一, 第二, 第三 = 0, 1, 2
球隊名稱, 勝率值 = 0, 1

上半季 = {'中信':[37, 0, 23], '富邦':[23, 0, 37], '統一':[26, 0, 34], '樂天':[34, 0, 26]}
下半季 = {'中信':[27, 2, 27], '富邦':[29, 1, 26], '統一':[30, 1, 26], '樂天':[25, 0, 32],}

剩餘比賽 = [['富邦', '統一'],['中信', '樂天'],['中信', '樂天'],['富邦', '中信'],
            ['樂天', '富邦'],['統一', '富邦'],['中信', '統一']]
```

deepcopy() 函式用來將字典 '下半季' 完整複製一份。

子任務三：季後賽晉級分析

程式執行結果

```
可能性 1 ['富邦', '中信', '中信', '富邦', '樂天', '統一', '中信'] => 季後賽: 中信 vs. 加賽( 富邦 , 統一 )
可能性 2 ['富邦', '中信', '中信', '富邦', '樂天', '統一', '統一'] => 季後賽: 中信 vs. 統一
可能性 3 ['富邦', '中信', '中信', '富邦', '樂天', '富邦', '中信'] => 季後賽: 中信 vs. 富邦
可能性 4 ['富邦', '中信', '中信', '富邦', '樂天', '富邦', '統一'] => 季後賽: 中信 vs. 富邦
可能性 5 ['富邦', '中信', '中信', '富邦', '富邦', '統一', '中信'] => 季後賽: 中信 vs. 富邦
可能性 6 ['富邦', '中信', '中信', '富邦', '富邦', '統一', '統一'] => 季後賽: 中信 vs. 加賽( 富邦 , 統一 )
可能性 7 ['富邦', '中信', '中信', '富邦', '富邦', '富邦', '中信'] => 季後賽: 中信 vs. 富邦
… (以下略，總共 128 種可能性)
```

子任務三：季後賽晉級分析

	已出賽	W-T-L	勝率	勝差
統一	57	30-1-26	.536	-
富邦	56	29-1-26	.527	0.5
中信	56	27-2-27	.500	2
樂天	57	25-0-32	.439	5.5

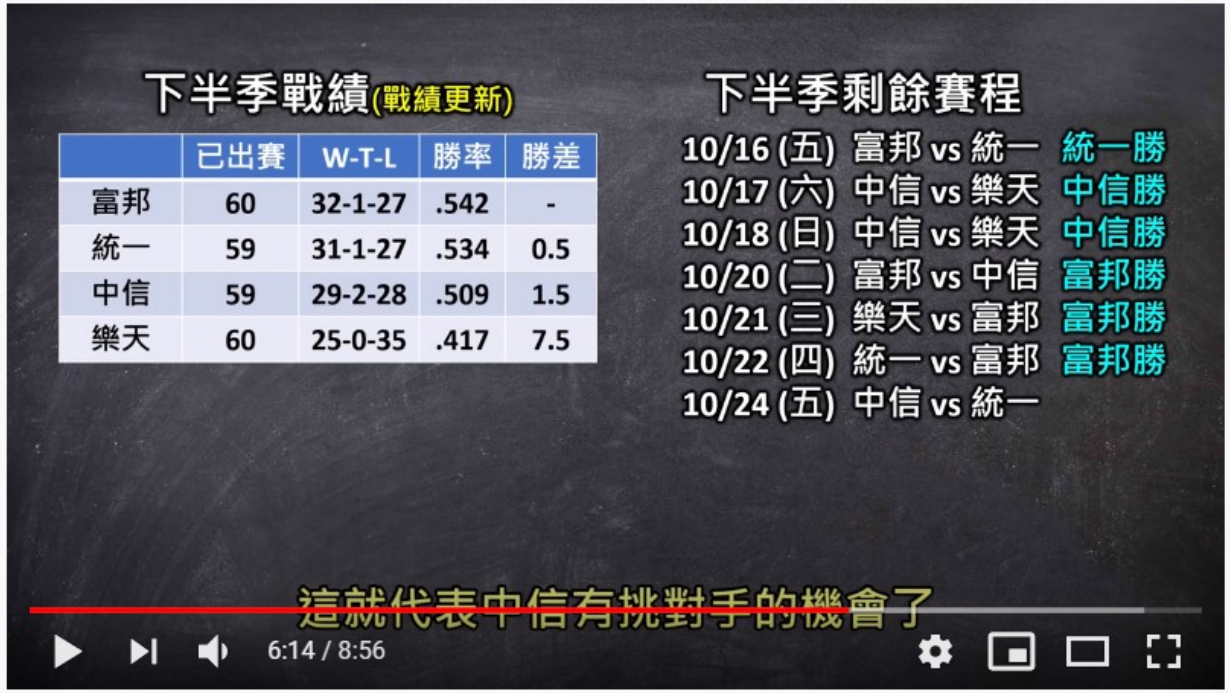
10/16 (五) 富邦 vs 統一 富邦勝
10/17 (六) 中信 vs 樂天 中信勝
10/18 (日) 中信 vs 樂天 中信勝
10/20 (二) 富邦 vs 中信 中信勝
10/21 (三) 樂天 vs 富邦 樂天勝
10/22 (四) 統一 vs 富邦 富邦勝
10/24 (五) 中信 vs 統一

#中華職棒 #CPBL #不要想著贏要想著不能贏
故意輸球才能打季後賽？中華職棒的賽制是在鼓勵放水？【Josh聊棒球】
177,577 views • Oct 15, 2020 4.1K 89 SHARE SAVE

如果是這個戰況，直到最後一場，四隊都還有機會爭取年度總冠軍。

可能性 11 ['富邦', '中信', '中信', '中信', '樂天', '富邦', '中信'] => 季後賽: 中信 vs. (樂天 , 統一)
可能性 12 ['富邦', '中信', '中信', '中信', '樂天', '富邦', '統一'] => 季後賽: 中信 vs. 加賽(富邦 , 統一)

子任務三：季後賽晉級分析



	已出賽	W-T-L	勝率	勝差
富邦	60	32-1-27	.542	-
統一	59	31-1-27	.534	0.5
中信	59	29-2-28	.509	1.5
樂天	60	25-0-35	.417	7.5

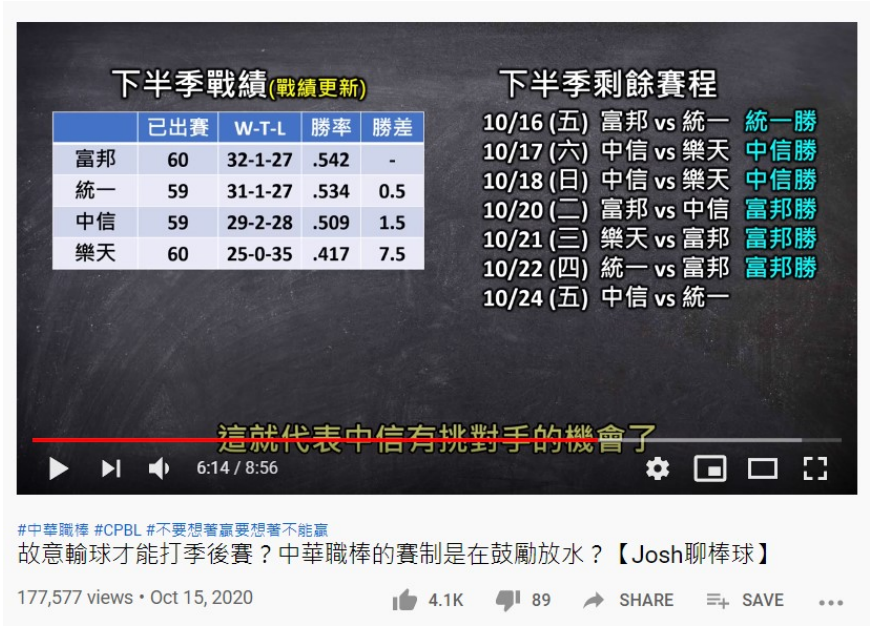
#中華職棒 #CPBL #不要想著贏要想著不能贏
故意輸球才能打季後賽？中華職棒的賽制是在鼓勵放水？【Josh聊棒球】
177,577 views • Oct 15, 2020 4.1K 89 SHARE SAVE

如果是這個戰況，而且中信輸掉最後一場，就有機會讓富邦和統一加賽。（如果中信不希望對上富邦，可採取此策略。）

可能性 71 ['統一', '中信', '中信', '富邦', '富邦', '富邦', '中信'] => 季後賽: 中信 vs. 富邦
可能性 72 ['統一', '中信', '中信', '富邦', '富邦', '富邦', '統一'] => 季後賽: 中信 vs. 加賽(富邦 , 統一)

子任務三：季後賽晉級分析

- 在 128 種比賽可能性中，請問：
  - 樂天隊有幾種可能性保有爭奪總冠軍的機會？
  - 如果 10/16 富邦輸了，富邦還有機會不和統一加賽就打進總冠軍賽嗎？



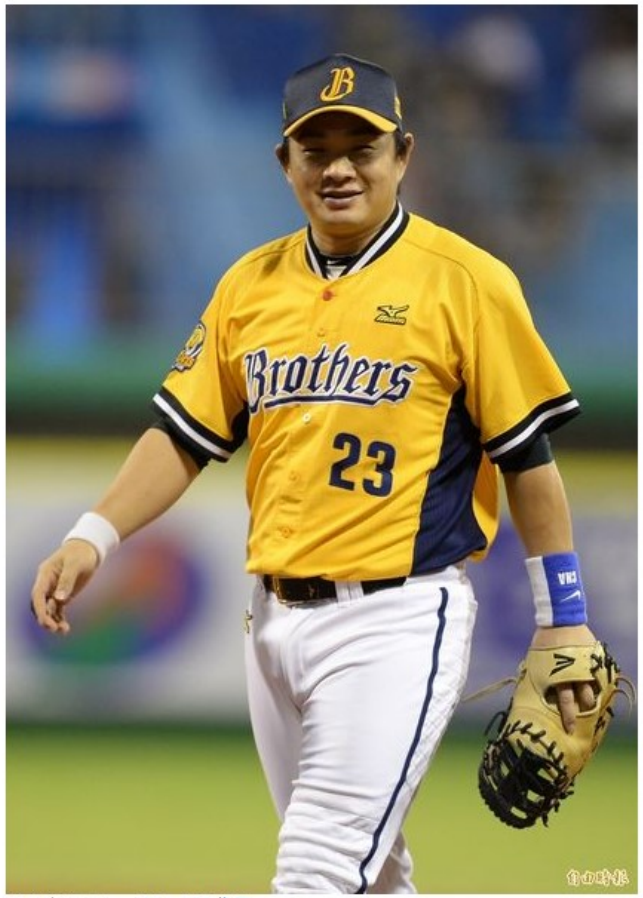
	已出賽	W-T-L	勝率	勝差
富邦	60	32-1-27	.542	-
統一	59	31-1-27	.534	0.5
中信	59	29-2-28	.509	1.5
樂天	60	25-0-35	.417	7.5

運動數據分析專題 II

中華職棒球員紀錄側寫

球員紀錄與側寫

- 中華職棒球員彭政閔
  - 19 年職業生涯
  - 生涯打擊率三成三
  - 保送數 953，聯盟第一
  - 連續 16 年打擊率三成以上
  - 兩千安、兩百盜、近兩百轟



此預覽的大小：426 × 600 像素.
完整解析度（800 × 1,126 像素，檔案大小：169 KB，MIME類型：image/jpeg）
內容說明：中信兄弟隊彭政閔
所有人：自由時報
攝影者：林正堃

本上傳圖片採用CC-BY-NC授權，引用時請「註明出處」、「勿做為商業用途」。

球員紀錄與側寫：常用棒球數據縮寫

縮寫	意義	縮寫	意義
G	出賽數	BB	保送數
PA	打席	SF	犧牲飛球
AB	打數 (扣除保送、犧觸、犧飛)	TB	壘打數
RBI	打點	SO	被三振數
H	安打數	SB	盜壘成功數
1B/2B/3B	一/二/三壘安打數	AVG	打擊率 (H/AB)
HR	全壘打數	OBP	上壘率 (H+BB)/(AB+BB+SF)
GIDP	雙殺打	SLG	長打率 (TB/AB)
SAC	犧牲短打	SF	犧牲飛球
IBB	故意四壞	CS	盜壘失敗數

資料檔內容

cpbl-player-batter-1990-2019

以 Excel 開啟

cpbl-player-batter-1990-2019 - Excel

A1 STATE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1	STATE	YEAR	NAME	G	PA	AB	RBI	R	H	1B	2B	3B	HR	TB	SO	SB	AVG	OBP	SLG	GIDP	SAC	SF	BB	IBB	HBP	CS
2	退役	1990	鷹俠	90	378	322	58	58	94	52	21	3	18	175	58	7	0.292	0.376	0.543	3	0	4	41	6	5	
3	退役	1990	瑞克	90	381	342	45	48	110	68	33	2	7	168	37	7	0.322	0.38	0.491	11	1	3	30	1	4	
4	退役	1990	洪一中	90	404	365	31	52	104	78	22	4	0	134	17	5	0.285	0.338	0.367	10	4	4	27	0	4	
5	退役	1990	王俊郎	90	348	313	24	32	71	57	12	1	1	88	40	3	0.227	0.288	0.281	6	8	0	25	0	2	
6	退役	1990	李居明	88	371	335	42	43	97	70	15	4	8	144	34	3	0.29	0.338	0.43	6	0	5	27	4	0	
7	退役	1990	林易增	87	383	350	34	66	116	94	15	3	4	149	20	34	0.331	0.378	0.426	7	2	3	24	0	4	
8	退役	1990	何良志	86	330	278	15	47	67	58	5	4	0	80	33	12	0.241	0.342	0.288	8	8	1	43	0	0	
9	退役	1990	林百亨	86	338	310	34	31	82	63	14	0	5	111	28	4	0.265	0.311	0.358	10	5	0	21	2	0	
10	退役	1990	陳金茂	86	364	315	49	59	93	64	21	2	6	136	40	7	0.295	0.366	0.432	4	3	7	32	0	7	

以 Notepad++ 開啟

C:\Users\User\Desktop\cpbl-player-batter-1990-2019.csv - Notepad++ [Administrator]

```
STATE,YEAR,NAME,G,PA,AB,RBI,R,H,1B,2B,3B,HR,TB,SO,SB,AVG,OBP,SLG,GIDP,SAC,SF,BB,IBB,HBP,CS,GO,AO,G/F,SB%,TA,SSA
退役,1990,鷹俠,90,378,322,58,58,94,52,21,3,18,175,58,7,0.292,0.376,0.543,3,0,4,41,6,5,5,58,116,0.5,0.583,0.992,1116.93
退役,1990,瑞克,90,381,342,45,48,110,68,33,2,7,168,37,7,0.322,0.38,0.491,11,1,3,30,1,4,2,104,94,1.106,0.778,0.857,854.64
退役,1990,洪一中,90,404,365,31,52,104,78,22,4,0,134,17,5,0.285,0.338,0.367,10,4,4,27,0,4,3,130,118,1.102,0.625,0.62,573.93
退役,1990,王俊郎,90,348,313,24,32,71,57,12,1,1,88,40,3,0.227,0.288,0.281,6,8,0,25,0,2,6,137,65,2.108,0.333,0.465,454.84
退役,1990,李居明,88,371,335,42,43,97,70,15,4,8,144,34,3,0.29,0.338,0.43,6,0,5,27,4,0,5,87,122,0.713,0.375,0.715,803.55
退役,1990,林易增,87,383,350,34,66,116,94,15,3,4,149,20,34,0.331,0.378,0.426,7,2,3,24,0,4,18,96,121,0.793,0.654,0.815,730.43
退役,1990,何良志,86,330,278,15,47,67,58,5,4,0,80,33,12,0.241,0.342,0.288,8,8,1,43,0,0,6,123,56,2.196,0.667,0.6,396.01
```

專題子任務說明

- 任務一
  - 單年條件查詢：例如「打擊率三成以上」。
- 任務二
  - 指定球員數據統計：例如「彭政閔生涯保送次數」。
- 任務三
  - 指定球員連續紀錄查詢：例如「彭政閔連續三成打擊率」紀錄。

程式先備知識

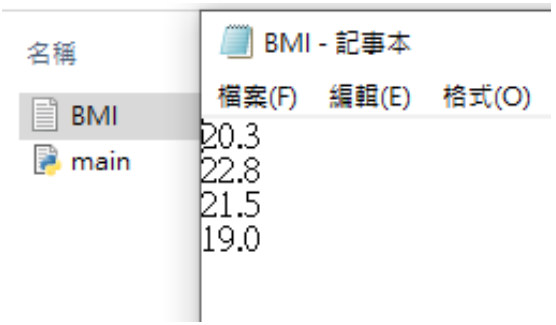
- 範例程式中會使用到的程式知識
  - if 和 for 句型
  - 列表 (list) 資料型態
  - 字典 (dict) 資料型態
  - 函式
    - 函式的撰寫與呼叫

讀取檔案資料

- 假設我們有一個資料檔案叫 *BMI.txt*。
  - 只要三行程式就可輕鬆讀出檔案中的內容。

```
f = open('BMI.txt')
for line in f:
    print(line)
```

```
20.3

22.8

21.5

19.0
```



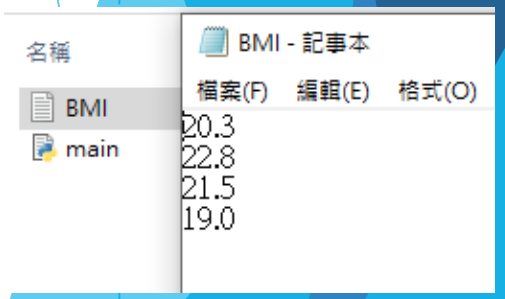
檔案中每一行結尾有一個未顯示的換行字元。因此line 的內容依序為 '20.3\n'、'22.8\n'、'21.5\n'和'19.0\n' 。

讀取檔案資料：基本字串處理

- 假設我們有一個資料檔案叫 *BMI.txt*。
  - 讀出檔案中的字串資料，接下來就可以運用之前所學來處理囉！

```
f = open('BMI.txt')

data = []
for line in f:
    bmi = float(line.strip())
    data.append(bmi)

print(data)
if len(data)>0:
    print('共', len(data), '筆資料，平均 BMI = ', sum(data)/len(data))
else:
    print('無資料')
```

[20.3, 22.8, 21.5, 19.0]
共 4 筆資料，平均 BMI = 20.9



字串的 `strip()` 函式可以清除頭尾的連續空白字元與換行字元。

讀取檔案資料：基本字串處理

- 字串的 strip() 函式
  - 清除頭尾的連續空白字元與換行字元。

```
s = 'abc'
print(s)

s = 'def\n'
print(s)
print('---')

s = s.strip()
print(s)
print('---')

s = '          xyz        \n        '
print(s.strip())
```

```
abc
def

---
def
---
xyz
```

讀取檔案資料：基本字串處理

- 讀入每行並使用 strip() 去除頭尾的白字元

```
f = open('BMI.txt')
for line in f:
    print(line)
```

```
20.3

22.8

21.5

19.0
```

```
f = open('BMI.txt')
for line in f:
    line = line.strip()
    print(line)
```

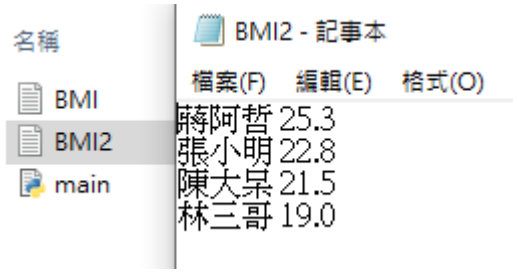
```
20.3
22.8
21.5
19.0
```

讀取檔案資料：基本字串處理

- 如果資料檔中每一行有不只一個資料項，我們可以利用字串的 split() 函式來切割。

```
f = open('BMI2.txt', encoding='utf8')
for line in f:
    line = line.split()
    print(line)
```

```
['蔣阿哲', '25.3']
['張小明', '22.8']
['陳大呆', '21.5']
['林三哥', '19.0']
```



字串的 split() 預設會以空白 (' ')、跳位 ('\t')和換行字元 ('\n') 來切開子字串。

讀取檔案資料：基本字串處理

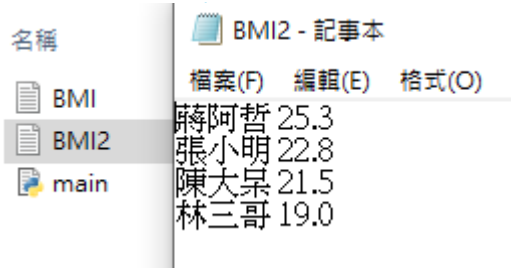
- 更多的檔案讀取方式
  - read() 一次讀完整個檔案，回傳一個字串。
  - readline() 一次讀一行，回傳字串。
  - readlines() 一次讀完檔案，回傳字串的列表。

  - 以前述方式讀取資料時，會將換行字元當作資料的一部份。若要去除（頭）尾部的換行字元，別忘了利用字串的 strip() 函式。

讀取檔案資料：建立列表並處理

- 我們可以將檔案內容先儲存成一個二維列表，以利後續處理與分析。

```
f = open('BMI2.txt', encoding='utf8')

data = []
for line in f:
    line = line.split()
    data.append(line)

print(data)
```

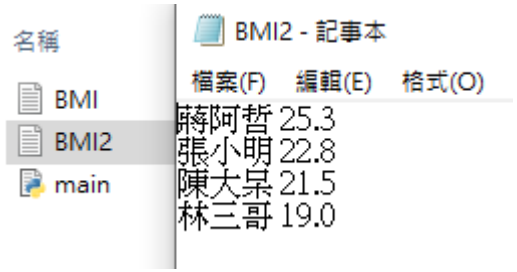



```
[['蔣阿哲', '25.3'], ['張小明', '22.8'], ['陳大呆', '21.5'], ['林三哥', '19.0']]
```

讀取檔案資料：建立列表並處理

- 此例示範讀入使用者的姓名和 BMI 指數，並輸出診斷結果。

```
f = open('BMI2.txt', encoding='utf8')
data = []
for line in f:
    line = line.split()
    data.append(line)

for person in data:
    name = person[0]
    bmi = float(person[1])
    if bmi <= 0:
        print(name, 'BMI 不在本程式接受範圍')
    elif 0 < bmi < 18.5:
        print(name, 'BMI:', bmi, '體重過輕')
    elif 18.5 <= bmi < 24:
        print(name, 'BMI:', bmi, '體重健康')
    elif 24 <= bmi < 27:
        print(name, 'BMI:', bmi, '體重過重')
    else:
        print(name, 'BMI:', bmi, '體重肥胖')
```




```
蔣阿哲 BMI: 25.3 體重過重
張小明 BMI: 22.8 體重健康
陳大呆 BMI: 21.5 體重健康
林三哥 BMI: 19.0 體重健康
```

讀取檔案資料：中文編碼

- 如果在讀取資料時發生錯誤，通常是因為沒有指定正確的中文編碼。

```
f = open('BMI2.txt')
for line in f:
    line = line.split()
    print(line)
```

UnicodeDecodeError: 'cp950' codec can't decode byte 0xb8 in position 4: illegal multibyte sequence

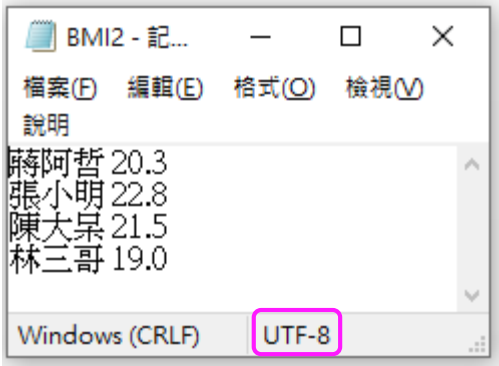
- 此時可嘗試在 open() 傳入 encoding 參數。
  - 中文編碼通常有 'utf8' 和 'ANSI' 兩種，兩種都試試看。('ANSI' 中文編碼也可傳入 'big5')

```
f = open('BMI2.txt', encoding='utf8')
```

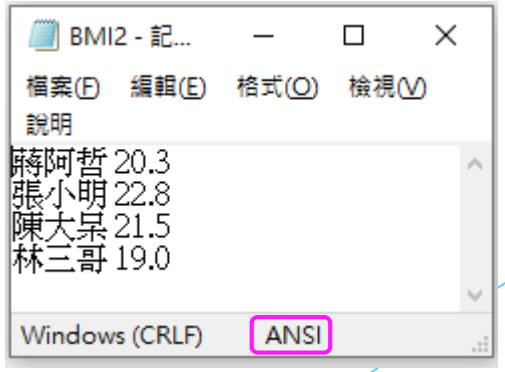
讀取檔案資料：中文編碼

- 指定中文編碼就沒問題囉！
  - 只有英數字的話，可以不必特別指定編碼方式。

```
f = open('BMI2.txt', encoding='utf8')
```



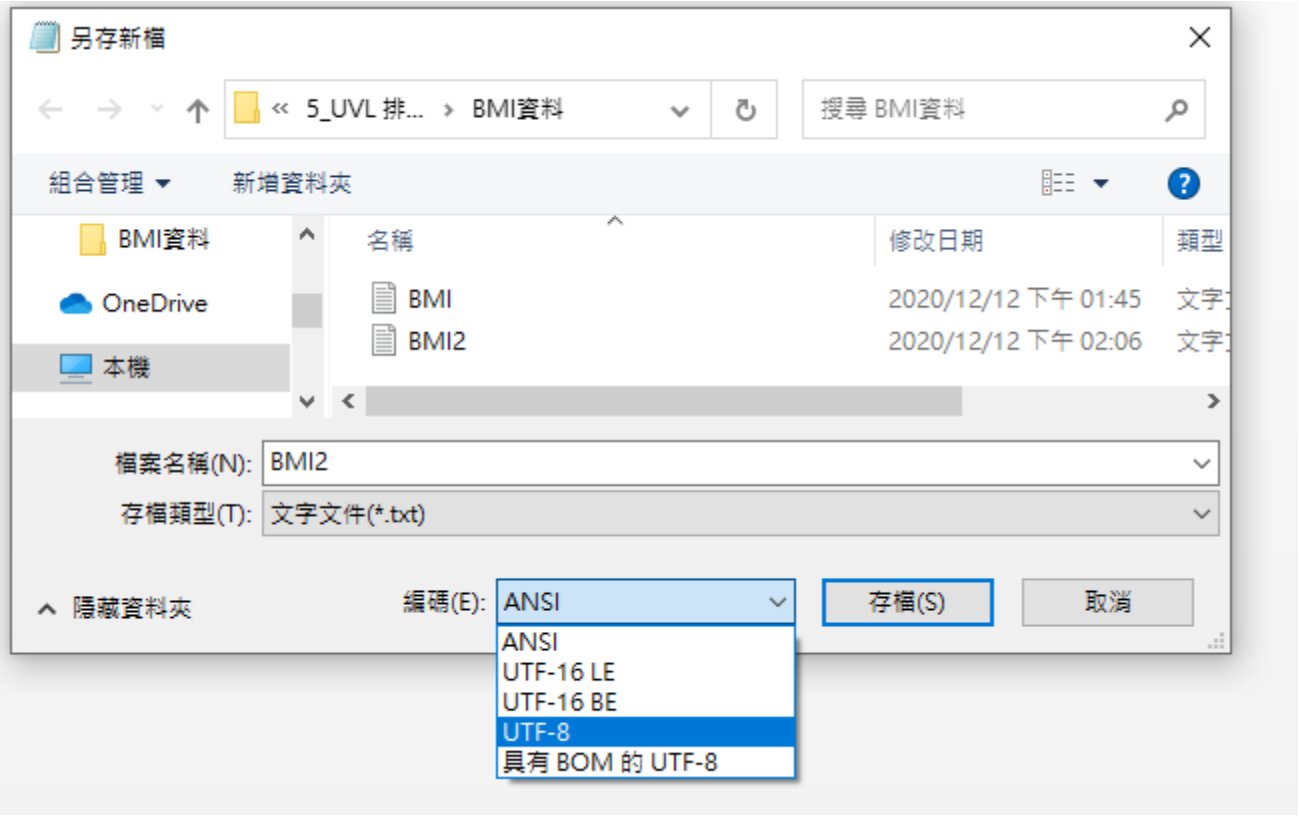

```
f = open('BMI2.txt', encoding='ANSI')
```



有些版本的記事本可以看到編碼方式。

讀取檔案資料：中文編碼

- 用文書軟體儲存文字檔時，通常可以選擇編碼方式。



球員紀錄與側寫：資料檔內容

以 Excel 開啟

cpbl-player-batter-1990-2019 - Excel

A1 STATE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1	STATE	YEAR	NAME	G	PA	AB	RBI	R	H	1B	2B	3B	HR	TB	SO	SB	AVG	OBP	SLG	GIDP	SAC	SF	BB	IBB	HBP	CS
2	退役	1990	鷹俠	90	378	322	58	58	94	52	21	3	18	175	58	7	0.292	0.376	0.543	3	0	4	41	6	5	
3	退役	1990	瑞克	90	381	342	45	48	110	68	33	2	7	168	37	7	0.322	0.38	0.491	11	1	3	30	1	4	
4	退役	1990	洪一中	90	404	365	31	52	104	78	22	4	0	134	17	5	0.285	0.338	0.367	10	4	4	27	0	4	
5	退役	1990	王俊郎	90	348	313	24	32	71	57	12	1	1	88	40	3	0.227	0.288	0.281	6	8	0	25	0	2	
6	退役	1990	李居明	88	371	335	42	43	97	70	15	4	8	144	34	3	0.29	0.338	0.43	6	0	5	27	4	0	
7	退役	1990	林易增	87	383	350	34	66	116	94	15	3	4	149	20	34	0.331	0.378	0.426	7	2	3	24	0	4	
8	退役	1990	何良志	86	330	278	15	47	67	58	5	4	0	80	33	12	0.241	0.342	0.288	8	8	1	43	0	0	
9	退役	1990	林百亨	86	338	310	34	31	82	63	14	0	5	111	28	4	0.265	0.311	0.358	10	5	0	21	2	0	
10	退役	1990	陳金茂	86	364	315	49	59	93	64	21	2	6	136	40	7	0.295	0.366	0.432	4	3	7	32	0	7	

以 Notepad++ 開啟

C:\Users\User\Desktop\cpbl-player-batter-1990-2019.csv - Notepad++ [Administrator]

```
STATE,YEAR,NAME,G,PA,AB,RBI,R,H,1B,2B,3B,HR,TB,SO,SB,AVG,OBP,SLG,GIDP,SAC,SF,BB,IBB,HBP,CS,GO,AO,G/F,SB%,TA,SSA
退役,1990,鷹俠,90,378,322,58,58,94,52,21,3,18,175,58,7,0.292,0.376,0.543,3,0,4,41,6,5,5,58,116,0.5,0.583,0.992,1116.93
退役,1990,瑞克,90,381,342,45,48,110,68,33,2,7,168,37,7,0.322,0.38,0.491,11,1,3,30,1,4,2,104,94,1.106,0.778,0.857,854.64
退役,1990,洪一中,90,404,365,31,52,104,78,22,4,0,134,17,5,0.285,0.338,0.367,10,4,4,27,0,4,3,130,118,1.102,0.625,0.62,573.93
退役,1990,王俊郎,90,348,313,24,32,71,57,12,1,1,88,40,3,0.227,0.288,0.281,6,8,0,25,0,2,6,137,65,2.108,0.333,0.465,454.84
退役,1990,李居明,88,371,335,42,43,97,70,15,4,8,144,34,3,0.29,0.338,0.43,6,0,5,27,4,0,5,87,122,0.713,0.375,0.715,803.55
退役,1990,林易增,87,383,350,34,66,116,94,15,3,4,149,20,34,0.331,0.378,0.426,7,2,3,24,0,4,18,96,121,0.793,0.654,0.815,730.43
退役,1990,何良志,86,330,278,15,47,67,58,5,4,0,80,33,12,0.241,0.342,0.288,8,8,1,43,0,0,6,123,56,2.196,0.667,0.6,396.01
```

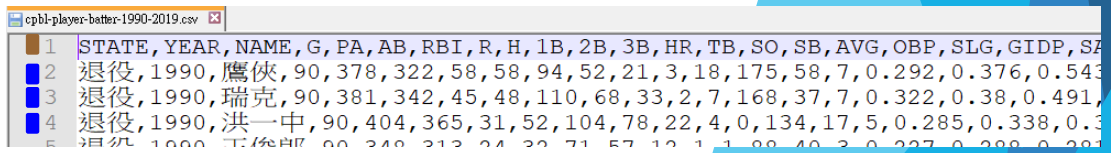
球員紀錄與側寫：讀取資料

- 讀取逐年資料
  - ReadDataFromFile() 從 *cpbl-player-batter-1990-2019.csv* 檔案中讀出資料。

```
def ReadDataFromFile(filename):
    year_data = []
    with open(filename) as inf:
        texts = inf.readlines()
        items = texts[0].strip().split(',')
        for line in texts[1:]:
            raw = line.strip().split(',')
            res = []
            for e in raw:
                if '.' in e: #資料為小數
                    res.append(float(e))
                elif e.isdigit(): #資料為整數
                    res.append(int(e))
                else:
                    res.append(e)
            year_data.append(dict(zip(items, res))) #將欄位名稱 items 和欄位內容 res 合併成字典
    return items, year_data
```

資料檔是 CSV 格式，欄位間以逗號隔開，因此以 split(',') 來分割資料。

- items 代表資料檔中第一段的資料項目，包含 'STATE', 'YEAR', 'NAME', 等等。
- year_data 是由多個字典組成的列表，每個字典代表一位球員在一個年度的資料。



本函式之讀取功能亦可透過 csv 模組的 DictReader() 函式達成，但讀取後仍需轉換資料型態。

球員紀錄與側寫：讀取資料

- 讀取逐年資料
  - ReadDataFromFile() 從 *cpbl-player-batter-1990-2019.csv* 檔案中讀出資料。
  - 列印前三筆逐年資料

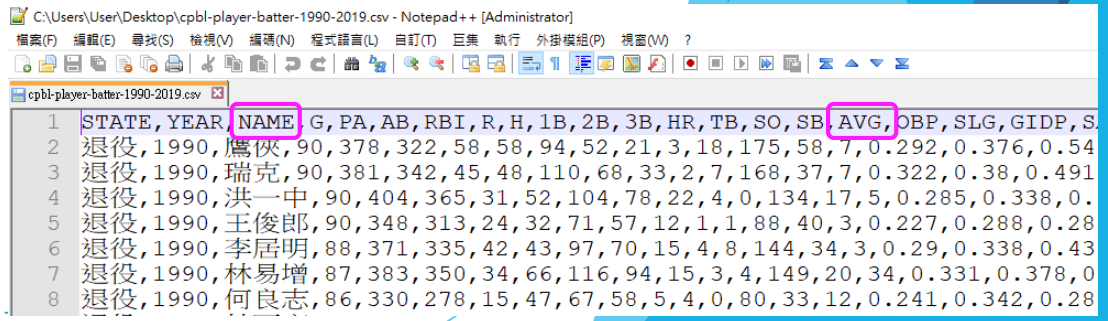
```
def main():
    items, year_data = ReadDataFromFile('cpbl-player-batter-1990-2019.csv')
    for e in year_data[:3]:
        print(e)
    return
```

```
{'現況': '退役', 'YEAR': 1990, 'NAME': '鷹俠', 'G': 90, 'PA': 378, 'AB': 322, 'RBI': 58, 'R': 58, 'H': 94, '1B': 52, '2B': 21,
'3B': 3, 'HR': 18, 'TB': 175, 'SO': 58, 'SB': 7, 'OBP': 0.376, 'SLG': 0.543, 'AVG': 0.292, 'GIDP': 3, 'SAC': 0, 'SF': 4, 'BB': 41,
'IBB': 6, 'HBP': 5, 'CS': 5, 'GO': 58, 'AO': 116, 'G/F': 0.5, 'SB%': 0.583, 'TA': 0.992, 'SSA': 1116.93}
{'現況': '退役', 'YEAR': 1990, 'NAME': '瑞克', 'G': 90, 'PA': 381, 'AB': 342, 'RBI': 45, 'R': 48, 'H': 110, '1B': 68, '2B': 33,
'3B': 2, 'HR': 7, 'TB': 168, 'SO': 37, 'SB': 7, 'OBP': 0.38, 'SLG': 0.491, 'AVG': 0.322, 'GIDP': 11, 'SAC': 1, 'SF': 3, 'BB': 30,
'IBB': 1, 'HBP': 4, 'CS': 2, 'GO': 104, 'AO': 94, 'G/F': 1.106, 'SB%': 0.778, 'TA': 0.857, 'SSA': 854.64}
{'現況': '退役', 'YEAR': 1990, 'NAME': '洪一中', 'G': 90, 'PA': 404, 'AB': 365, 'RBI': 31, 'R': 52, 'H': 104, '1B': 78, '2B': 22,
'3B': 4, 'HR': 0, 'TB': 134, 'SO': 17, 'SB': 5, 'OBP': 0.338, 'SLG': 0.367, 'AVG': 0.285, 'GIDP': 10, 'SAC': 4, 'SF': 4, 'BB': 27,
'IBB': 0, 'HBP': 4, 'CS': 3, 'GO': 130, 'AO': 118, 'G/F': 1.102, 'SB%': 0.625, 'TA': 0.62, 'SSA': 573.93}
```

子任務一：單年資料查詢

- 以 for 句型依序檢視每一列資料，並以 if 條件查詢。
  - p 代表一列資料。
  - p['NAME'] 取出球員姓名，p['AVG'] 取出打擊率 (AVG)，以此類推。
  - fprint() 是老師寫好的一個格式化函式，將一列資料儘量以整齊一致的格式輸出到畫面上。

```
def main():
    items, year_data = ReadDataFromFile('cpbl-player-batter-1990-2019.csv')
    #-----------------------------------------------------------------
    # 查詢某種特定條件：
    #-----------------------------------------------------------------
    print(' '*3,end='')
    fprint(items)
    rec = []
    for p in year_data:
        if p['NAME']=='彭政閔' and p['AVG']>=0.3:
            fprint(p.values())
            rec += [p['NAME']]
    print(Counter(rec))
```



Counter 需由 collections 模組匯入。

子任務一：單年資料查詢

- 以 for 句型依序檢視每一列資料，並以 if 條件查詢。

```
def main():
    items, year_data = ReadDataFromFile('cpbl-player-batter-1990-2019.csv')

    #-----------------------------------------------------------------
    # 查詢某種特定條件：
    #-----------------------------------------------------------------
    print(' '*3,end='')
    fprint(items)
    rec = []
    for p in year_data:
        if p['NAME']=='彭政閔' and p['AVG']>=0.3:
            fprint(p.values())
            rec += [p['NAME']]
    print(Counter(rec))
```

```
STATE YEAR NAME    G   PA   AB  RBI    R    H   1B   2B   3B   HR   TB   SO   SB   AVG   OBP   SLG GIDP  SAC   SF   BB  IBB  HBP   CS   GO   AO   G/F   SB%    TA     SSA
退役 2001 彭政閔   59  169  136   19   20   43   32    5    1    5   65   33    5 0.316 0.433 0.478    4    5    0   25    0    3    3   32   28 1.143 0.625 0.980 576.180
退役 2002 彭政閔   88  358  277   71   66   86   48   17    3   18  163   82   16 0.310 0.442 0.588    5    3    6   62    2    8    6   53   62 0.855 0.727 1.243 1188.470
退役 2003 彭政閔  100  443  369   83   83  131   85   25    3   18  216   81   22 0.355 0.446 0.585    3    1    3   54    7    9    9   86   74 1.162 0.710 1.232 1346.010
退役 2004 彭政閔   99  425  338   66   90  127   92   14    3   18  201   64   26 0.376 0.484 0.595    8    1    4   69    7    6   11   97   54 1.796 0.703 1.343 1266.740
退役 2005 彭政閔   76  331  274   47   57   93   68    8    3   14  149   49   13 0.339 0.442 0.544    3    3    1   47    2    4    5   94   39 2.410 0.722 1.138 1003.420

[中間省略]

退役 2014 彭政閔  118  488  399   56   57  120   95   15    1    9  164   69   18 0.301 0.411 0.411   12    1    6   73    3    6    7  114  102 1.118 0.720 0.886 924.750
退役 2015 彭政閔  108  484  409   57   62  131  110   16    1    4  161   81    3 0.320 0.413 0.394   14    1    5   65    1    3    4   97  105 0.924 0.429 0.787 846.290
退役 2016 彭政閔  105  403  343   67   58  117   90   19    0    8  160   47    4 0.341 0.417 0.466   13    0    9   47    0    4    2   88  100 0.880 0.667 0.892 996.110
退役 2019 彭政閔   92  328  288   34   29   87   67   18    0    2  111   40    1 0.302 0.367 0.385    7    2    4   29    2    3    0   79   86 0.919     1 0.702 623.080
Counter({'彭政閔': 17})
```

子任務一：單年資料查詢

- 以 for 句型依序檢視每一列資料，並以 if 條件查詢。

```
def main():
    items, year_data = ReadDataFromFile('cpbl-player-batter-1990-2019.csv')

    #-----------------------------------------------------------------
    # 查詢某種特定條件：
    #-----------------------------------------------------------------
    print(' '*3,end='')
    fprint(items)
    rec = []
    for p in year_data:
        if p['NAME']=='彭政閔' and p['AVG']>=0.3:
            fprint(p.values())
            rec += [p['NAME']]
    print(Counter(rec))
```

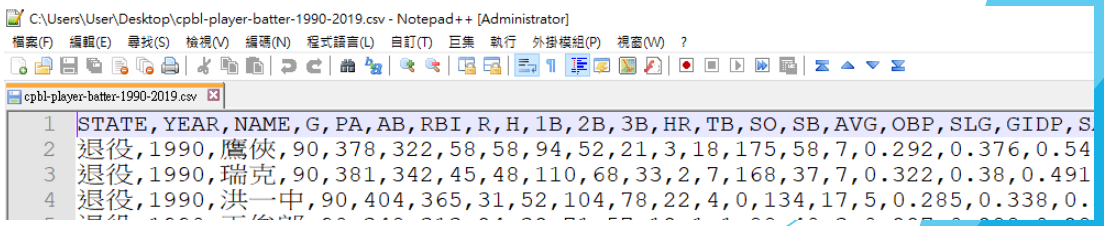
試著寫出你想查詢的條件，例如：

- 年度全壘打數 20 支以上且盜壘數 20 次以上的紀錄。(有打擊有腳程)
- 年度打席 150 以上且三振數 10 以上的紀錄。(打擊纏鬥性高)
- 年度出賽數 50 以上但打席 50 以下的紀錄。(守備組)

子任務二：指定球員數據統計

- 整併球員資料
  - GetPlayerRecord() 從 year_data 中將每位球員的資料整理在一起。

```
def GetPlayerRecord(items, year_data):
    player_data = {}
    for p in year_data: #year_data 是一個由字典組成的列表。
        name = p['NAME']
        if name not in player_data:
            player_data[name] = {}
            for it in items[3:]:
                player_data[name][it] = [p[it]]
        else:
            for it in items[3:]:
                player_data[name][it] += [p[it]]
    return player_data
```



子任務二：指定球員數據統計

- 整併球員資料
  - GetPlayerRecord() 從 year_data 中將每位球員的資料整理在一起。

```
def main():
    items, year_data = ReadDataFromFile('cpbl-player-batter-1990-2019.csv')
    player_data = GetPlayerRecord(items, year_data)
    print('彭政閔歷年四壞數', player_data['彭政閔']['BB'])
    print('張泰山歷年全壘打數', player_data['張泰山']['HR'])
```

```
彭政閔歷年四壞數 [25, 62, 54, 69, 47, 25, 48, 45, 66, 58, 37, 69, 60, 73, 65, 47, 40, 34, 29]
張泰山歷年全壘打數 [16, 9, 14, 17, 11, 13, 10, 28, 21, 15, 24, 19, 11, 17, 16, 14, 17, 9, 5, 3]
```

子任務二：指定球員數據統計

- player_data[姓名] 取出一位球員的歷年紀錄，其為一字典。
  - 例如 p1 = player_data['彭政閔']。
- player[姓名][數據] 取出該球員歷年的特定數據，其為一列表。
  - 例如 data = p1['SB']，data 可以代入內建函式 sum() 和 len()。

```
name = '彭政閔'
pl = player_data[name]

# 求總數
item = 'SB'
data = pl[item]
print(name, item, '總數', sum(data))

# 求平均
item = 'AVG'
data = pl[item]
print(name, len(data), '年生涯', item, '平均', sum(data)/len(data))
```

```
彭政閔 SB 總數 231
彭政閔 19 年生涯 AVG 平均 0.3330526315789474
```

子任務三：指定球員連續紀錄

- player_data[姓名] 取出一位球員的歷年紀錄，其為一字典。
  - 例如 p1 = player_data['彭政閔']。
- player[姓名][數據] 取出該球員歷年的特定數據，其為一列表。
  - 例如 data = p1['SB']。

```
name = '彭政閔'
pl = player_data[name]
item = 'AVG'
data = pl[item]

max_cont = 0
cont = 0
for e in data:
    if e>=0.3:
        cont += 1
        if cont>max_cont:
            max_cont = cont
    else:
        cont = 0
print(name, item, '連續保持三成', max_cont, '年')
```

彭政閔 AVG 連續保持三成 16 年

子任務三：指定球員連續紀錄

- player_data[姓名] 取出一位球員的歷年紀錄，其為一字典。
  - 例如 p1 = player_data['彭政閔']。
- player[姓名][數據] 取出該球員歷年的特定數據，其為一列表。
  - 例如 data = p1['SB']。

```
name = '彭政閔'
pl = player_data[name]
item = 'AVG'
data = pl[item]

max_cont = 0
cont = 0
for i in range(1, len(data)):
    if data[i]>data[i-1]:
        cont += 1
        if cont>max_cont:
            max_cont = cont
    else:
        cont = 0
print(name, item, '最長連續進步', max_cont, '年')
```

彭政閔 AVG 最長連續進步 3 年

球員紀錄與側寫

- 網路上有很多關於球員的紀錄與故事，現在你可以透過你的程式有效率地挖掘出更多有趣的紀錄和故事囉！

運動數據分析專題 III

108 UVL 大專女排冠軍戰

2020-04-20 UVL》臺師大女子排球隊二連霸 羅儀璟榮獲MVP

賽前被對方互稱為最熟悉對手的臺師大與北市大，在冠軍賽正面交鋒，尋求連霸的臺師大面對北市大頑強的挑戰，克服二、三兩局的落後困境後來居上，以3:0 (25:15、25:19、25:20) 連續兩年在冠軍賽中擊敗北市大拿下冠軍。

http://pr.ntnu.edu.tw/news/index.php?mode=data&id=19205&keywords=UVL

排球的基本認識

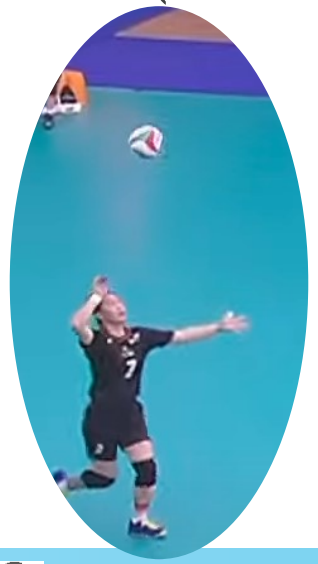
YouTube TW
Search
國立臺灣師範大學 - 先發球員
FOX SPORTS
We The Fans
UVL
17 邱雅慧
11 葉于雯
4 羅儀璟
6 童琦芳
10 劉美菁
7 陳巧聆
2 林其蓉
LIBERO
WE THE FANS
3:52 / 2:02:00
#訂閱SSUtvSports #WeTheFans #UVL
HD決賽::冠軍賽::臺灣師大vs臺北市大::女一級 108UVL大專排球聯賽 網路直播
96,479 views • Streamed live on Apr 19, 2020
338
11
SHARE
SAVE

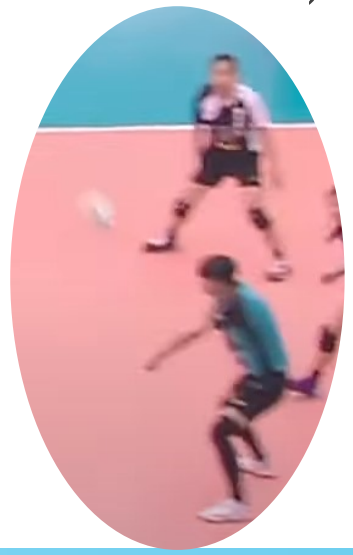
截圖來自 https://www.youtube.com/watch?v=3DjkhWQupkc&t=5879s

排球的基本認識

- 基本流程
  - A 隊發球
  - B 隊接球、舉球、攻擊 (也可以一擊或二擊就擊回對方球場)
  - A 隊接球、舉球、攻擊

  …… (循環直到球掉到地上)





排球的基本認識

- 得分
  - 球掉在 A 隊的界內，B 隊得一分。
  - A 隊把球碰到界外，B 隊得一分。

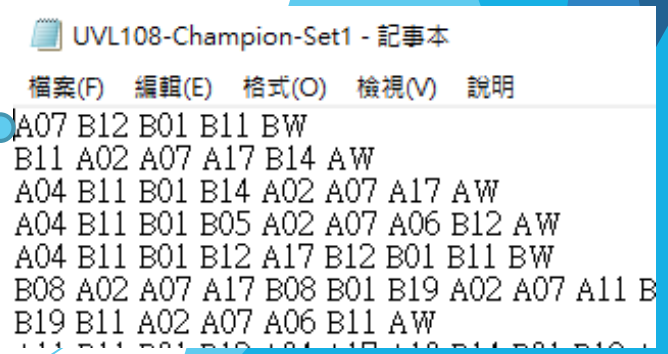
- 五戰三勝制
  - 前四局 25 分制，第五局 15 分制。

資料集說明

- 108UVL大專排球聯賽冠軍賽::臺灣師大vs臺北市大
  - https://www.youtube.com/watch?v=3DjkhWQupkc&t=5879s
  - 由老師觀看比賽過程並逐一記錄每一球碰球球員以及該球得分隊伍。
    - A 代表臺師大，B 代表北市大。A07 代表臺師大 7 號。
    - 每一列代表得到一分的過程。第一個資料項代表發球的球員，後續依序為碰球的球員。最後的 AW (BW) 代表 A (B) 隊得分。
    - 例如下面這一列代表的過程是

      A07 發球，B12 接球，B01 舉球，B11 殺球，B 隊得分。（B11 殺球成功。）

```
A07 B12 B01 B11 BW
```



專題子任務說明

- 任務一
  - 計算兩隊比數。
- 任務二
  - 計算球員的發球次數。
- 任務三
  - 自行發想。

程式先備知識

- 範例程式中會使用到的程式知識
  - if 和 for 句型
  - 字串 (str) 資料型態與 join() 操作
  - 列表 (list) 資料型態
  - 檔案處理
    - 開啟文字檔與讀檔

子任務一：計算兩隊比數

- 任務說明
  - 讀入資料檔 *UVL108-Champion-Set1.txt* 的內容。
  - 計算 AW 和 BW 出現的次數，即為兩隊比數。

```
oneset = []
with open('UVL108-Champion-Set1.txt') as f:
    for line in f:
        oneset.append(line.split())

for ball in oneset[:3]: #印出前三筆資料
    print(ball)

score_A = 0
for ball in oneset:
    if ball[-1]=='AW':
        score_A += 1

print('A 隊得', score_A, '分')
```

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

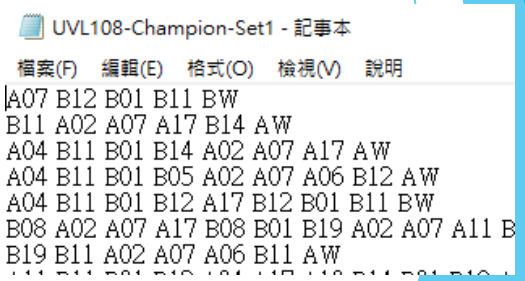
```
['A07', 'B12', 'B01', 'B11', 'BW']
['B11', 'A02', 'A07', 'A17', 'B14', 'AW']
['A04', 'B11', 'B01', 'B14', 'A02', 'A07', 'A17', 'AW']
A 隊得 25 分
```

子任務一：計算兩隊比數

- 任務說明
  - 讀入資料檔 *UVL108-Champion-Set1.txt* 的內容。
  - 計算 AW 和 BW 出現的次數，即為兩隊比數。

參考程式

```
oneset = []
with open('UVL108-Champion-Set1.txt') as f:
    for line in f:
        oneset.append(line.split())

score_A = score_B = 0
for ball in oneset:
    if ball[-1]=='AW':
        score_A += 1
    else:
        score_B += 1

print('臺師大 vs. 北市大:', score_A, ':', score_B)
```




```
臺師大 vs. 北市大: 25 : 15
```

子任務一：計算兩隊比數

- 任務說明
  - 利用迴圈和字串處理產生檔名，便可輕鬆計算多局的比數。

參考程式

```
for i in range(1, 4):
    oneset = []
    with open(f'UVL108-Champion-Set{i}.txt') as f:
        for line in f:
            oneset.append(line.split())

    score_A = score_B = 0
    for ball in oneset:
        if ball[-1]=='AW':
            score_A += 1
        else:
            score_B += 1

    print('第', i, '局，臺師大 vs. 北市大:', score_A, ':', score_B)
```

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

```
第 1 局，臺師大 vs. 北市大: 25 : 15
第 2 局，臺師大 vs. 北市大: 25 : 19
第 3 局，臺師大 vs. 北市大: 25 : 20
```

子任務一：計算兩隊比數

- 任務說明
  - 利用 Python 內建的 Counter 型態，可以簡便地計算一維列表中每個資料出現的次數。

```
from collections import Counter

data = [1, 2, 1, 2, 2, 4, 3, 4, 4, 3]
c = Counter(data)
print('1 出現', c[1], '次')

data = ['AW', 'BW', 'AW', 'AW', 'BW']
c = Counter(data)
print('AW 出現', c['AW'], '次')
```

```
1 出現 2 次
AW 出現 3 次
```

因為時間有限，這裡就不解釋第一行了。從實用的角度，照著打就對了。
(很簡略的解說：從 collections 這個程式模組中匯入 Counter 這個資料型態。)

子任務一：計算兩隊比數

- 任務說明
  - 有了 Counter，只要把想記錄的結果（此處為字串 AW 和 BW）存入到一維列表，存好後傳給 Counter()，就可以得到其對應的出現次數了。

```
from collections import Counter

for i in range(1, 4):
    oneset = []
    with open(f'UVL108-Champion-Set{i}.txt') as f:
        for line in f:
            oneset.append(line.split())

    results = [ball[-1] for ball in oneset]
    ct = Counter(results)

    print('第', i, '局，臺師大 vs. 北市大:', ct['AW'], ':', ct['BW'])
```

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

```
第 1 局，臺師大 vs. 北市大: 25 : 15
第 2 局，臺師大 vs. 北市大: 25 : 19
第 3 局，臺師大 vs. 北市大: 25 : 20
```

子任務二：發球次數統計

- 在任務一中，我們關注每一球紀錄的最後一個項目，可以得到兩隊的得分比數。
- 如果我們要計算每一位球員的發球次數，該關注哪一個項目？該修改程式的哪些部份？

UVL108-Champion-Set1 - 記事本

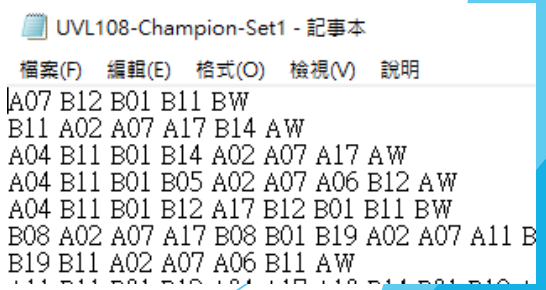
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

子任務二：發球次數統計

- 如果我們要計算每一位球員的發球次數，該關注哪一個項目？該修改程式的哪些部份？
  - 首先，我們要記錄每一球的第一個項目，即發球者。
  - 另外，我們要累計所有局的結果，所以要移動 results = [] 到迴圈之外。

```
from collections import Counter

results = []
for i in range(1, 4):
    oneset = []
    with open(f'UVL108-Champion-Set{i}.txt') as f:
        for line in f:
            oneset.append(line.split())

    for ball in oneset:
        results.append(ball[0])

ct = Counter(results)
```



子任務二：發球次數統計

- 如果不只是要查詢單一個資料的次數，而是要列出所有資料的次數，同樣可以用 for … in … 句型來操作 Counter。

參考程式

```
from collections import Counter

results = []
for i in range(1, 4):
    oneset = []
    with open(f'UVL108-Champion-Set{i}.txt') as f:
        for line in f:
            oneset.append(line.split())

    for ball in oneset:
        results.append(ball[0])

ct = Counter(results)
for e in ct:
    print('球員', e, '發球', ct[e], '次')
```

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

```
球員 A07 發球 9 次
球員 B11 發球 8 次
球員 A04 發球 24 次
球員 B08 發球 2 次
球員 A11 發球 14 次
球員 B05 發球 8 次
球員 A17 發球 13 次
球員 B12 發球 10 次

... [略]
```

子任務二：發球次數統計

- 如果要依次數排序，Counter 有個 most_common() 函式可以用，它會回傳排序好的二維列表。
  - 二維列表中的每個一維列表是由 (資料, 次數) 組成的。

參考程式

```
from collections import Counter

results = []
for i in range(1, 4):
    oneset = []
    with open(f'UVL108-Champion-Set{i}.txt') as f:
        for line in f:
            oneset.append(line.split())

    for ball in oneset:
        results.append(ball[0])

ct = Counter(results)
for e in ct.most_common():
    print('球員', e[0], '發球', e[1], '次')
```

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

```
球員 A04 發球 24 次
球員 A11 發球 14 次
球員 A17 發球 13 次
球員 B12 發球 10 次
球員 A07 發球 9 次
球員 B10 發球 9 次
球員 B11 發球 8 次
球員 B05 發球 8 次

... [略]
```

子任務二：發球次數統計

```
球員 A04 發球 24 次
球員 A11 發球 14 次
球員 A17 發球 13 次
球員 B12 發球 10 次
球員 A07 發球 9 次
球員 B10 發球 9 次
球員 B11 發球 8 次
球員 B05 發球 8 次

... [略]
```

照片來源
https://volsports.co/blog/2020/04/27/lo/

更多專題任務 ...

- 為什麼羅儀璟發球時師大特別會得分？
  - 她發球會直接得分嗎？
    - 如何判斷直接得分呢？

      發球者球隊 ball[0][0] 和得分球隊 ball[-1][0] 相同
      且該筆紀錄沒有發球者的隊友碰過球

```
for ball in oneset:
    if ball[0][0] == ball[-1][0] and ''.join(ball).count(ball[0][0])==2:
        results.append(ball[0])
```

  - 她發球會壓制對方的攻擊嗎？
    - 如果對方的第二位接球員不是 'B01' 也不是 'B02'，就算是壓制。

北市大先發舉球
陳佳蔓

北市大替補舉球
陳莉鎔

(UVL108 最佳舉球)

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

```
球員 A11 發球得分 4 次
球員 A04 發球得分 3 次
球員 B10 發球得分 3 次
球員 A17 發球得分 2 次
球員 B12 發球得分 2 次
球員 B05 發球得分 1 次
球員 B19 發球得分 1 次
球員 A10 發球得分 1 次
球員 B01 發球得分 1 次
球員 B02 發球得分 1 次
```

```
球員 A04 發球壓制 3 次
球員 A17 發球壓制 2 次
球員 A11 發球壓制 2 次
球員 A10 發球壓制 1 次
球員 B10 發球壓制 1 次
```

更多專題任務 …

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

- 想一想，還能作些什麼分析呢？
  - 發球失分？
    - 如何判斷發球失分呢？

      發球後沒有己方球隊沒再碰到球且是對方得分

```
for ball in oneset:
    if ''.join(ball[1:]).count(ball[0][0])==0:
        results.append(ball[0])
```

    - 其實上面包含兩種情況：

      發球直接失分；
      發球後對方扣殺得分

再加上 `len(ball)==2` 這個條件就可以過濾出直接失分了。

```
球員 B12 發球失分 5 次
球員 A07 發球失分 3 次
球員 A06 發球失分 3 次
球員 A04 發球失分 3 次
球員 B19 發球失分 3 次
球員 A17 發球失分 3 次
球員 B11 發球失分 2 次
球員 B10 發球失分 2 次
球員 A10 發球失分 2 次
球員 B02 發球失分 2 次
球員 B01 發球失分 1 次
球員 B05 發球失分 1 次
球員 A11 發球失分 1 次
```

更多專題任務 ...

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

- 想一想，還能作些什麼分析呢？
  - 碰球與接發球次數

```
for ball in oneset:
    touch += ball
    if ball[1][1] != 'W':
        receive.append(ball[1])

... (以 Counter 計數)
```

```
球員 A07 碰球 126 次
球員 A04 碰球 112 次
球員 B01 碰球 100 次
球員 B11 碰球 90 次
球員 B05 碰球 81 次
球員 B12 碰球 75 次
球員 A06 碰球 72 次
球員 A02 碰球 68 次
球員 A17 碰球 60 次

... [略] ...
```

A07 和 B01 是主力舉球員，每波都要過手，碰球次數自然很多。

A04、B11、B05 都是主攻手，所以碰球次數也多。

但 A04 明顯比該隊碰球數第三的 A06 多出 40 次，是接球特別好還是攻勢特別多呢？

```
球員 B05 接發球 19 次
球員 B11 接發球 18 次
球員 A02 接發球 16 次
球員 A04 接發球 16 次
球員 B14 接發球 14 次
球員 B12 接發球 12 次
球員 A06 接發球 10 次
球員 B15 接發球 3 次
球員 A17 接發球 3 次
球員 A?? 接發球 2 次
球員 B02 接發球 1 次
球員 B?? 接發球 1 次
球員 A11 接發球 1 次
球員 B01 接發球 1 次
```

A02 是自由球員，接發球全隊最多很正常，但 A04 也一樣多，表示 A 隊會安排由她來負責接球。

B05 和 B11 都比自由球員 B14 接球次數還多，是否表示她們比自由球員更會接發球？或者是 A 隊故意發給她們？

更多專題任務 ...

UVL108-Champion-Set1 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
A07 B12 B01 B11 BW
B11 A02 A07 A17 B14 AW
A04 B11 B01 B14 A02 A07 A17 AW
A04 B11 B01 B05 A02 A07 A06 B12 AW
A04 B11 B01 B12 A17 B12 B01 B11 BW
B08 A02 A07 A17 B08 B01 B19 A02 A07 A11 B
B19 B11 A02 A07 A06 B11 AW
```

- 想一想，還能作些什麼分析呢？
  - 計算接發球後得失分數

```
for ball in oneset:
    if ball[1][1] != 'W':
        if ball[-1][0] == ball[1][0]:
            receive.append('+'+ball[1])
        else:
            receive.append('-'+ball[1])
```

```
球員 +A04 接發球 14 次
球員 +B05 接發球 14 次
球員 +A02 接發球 12 次
球員 -B11 接發球 12 次
球員 -B14 接發球 9 次
球員 +B12 接發球 6 次
球員 +B11 接發球 6 次
球員 +B14 接發球 6 次
球員 -B12 接發球 6 次
球員 -A06 接發球 5 次
球員 -B05 接發球 5 次
球員 +A06 接發球 5 次
球員 -A02 接發球 4 次
球員 -A04 接發球 2 次
球員 +A?? 接發球 2 次
球員 -A17 接發球 2 次
球員 +B15 接發球 2 次
球員 -B?? 接發球 1 次
球員 -B15 接發球 1 次
球員 -A11 接發球 1 次
球員 +A17 接發球 1 次
球員 -B02 接發球 1 次
```

A02 是自由球員，接發球全隊最多很正常，但 A04 也一樣多，表示 A 隊會安排由她來負責接球。
A04 接球後球隊攻勢最順暢，其次是 A02，因此 A 隊安排正確。

B05 和 B11 都比自由球員 B14 接球次數還多，是否表示 她們比自由球員更會接發球？或者是 A 隊故意發給她們？
B05 接球後球隊攻勢順暢，但 B11 和 B14 接球後失分多過得分。

此外，A 隊保護舉球員 A07 100% 成功， B 隊也保護得很好，B02 未接過發球，B02 接過 1 次，但那次是意外。

YouTube TW

運動數據分析專題 IV

圖表繪製

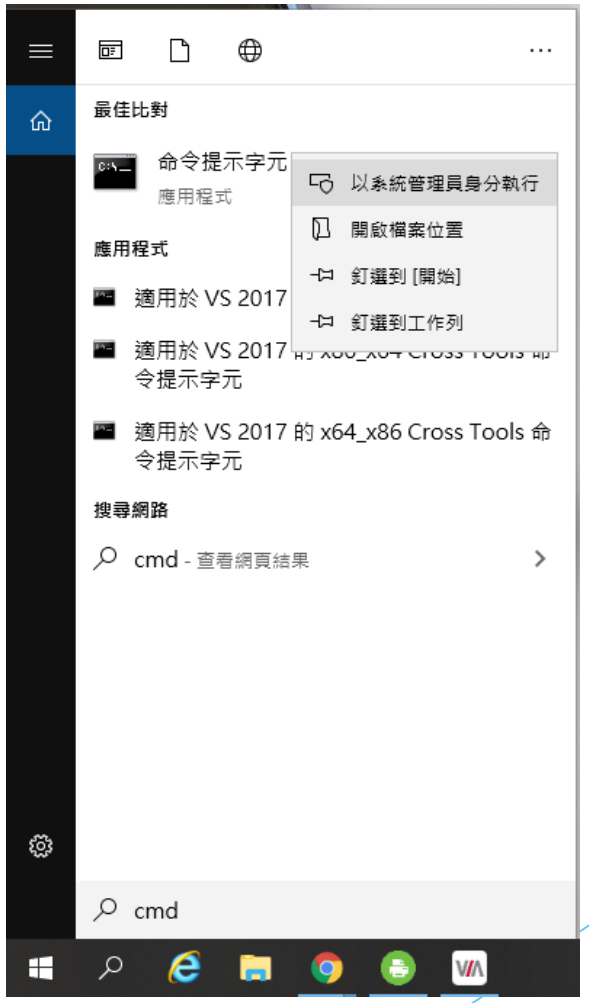
matplotlib

- 今天我們要練習使用 matplotlib 這個 Python 的繪圖套件。

- 師父領進門，修行在個人。
  - matplotlib 的繪圖功能很多，課堂上先帶大家瀏覽基本功能。
  - 若有進階需求，詢問 Google 大神通常也都會有答案。

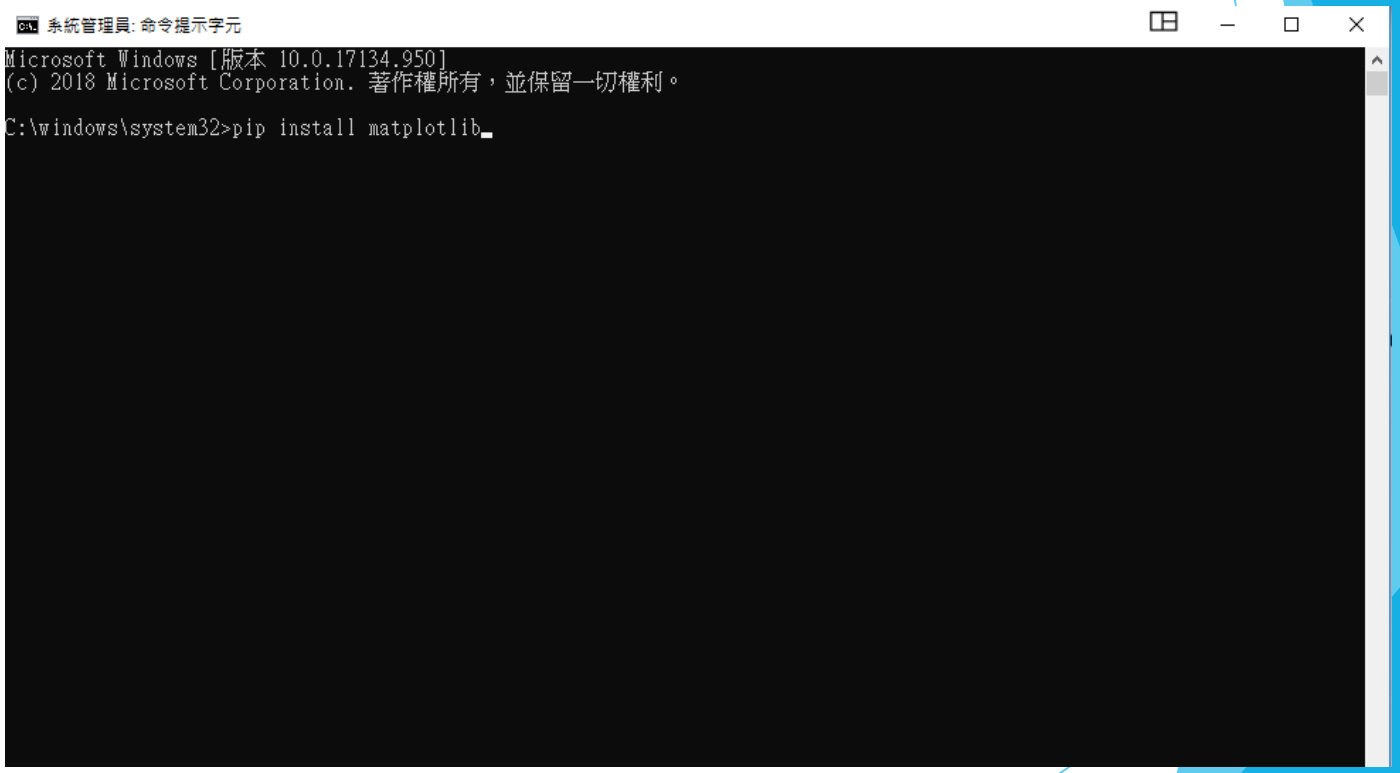
安裝步驟1 - 開啟命令提示字元

- 點擊「搜尋」
- 輸入「**cmd**」
- 對「**命令提示字元**」右鍵，「**以系統管理員身分執行**」。



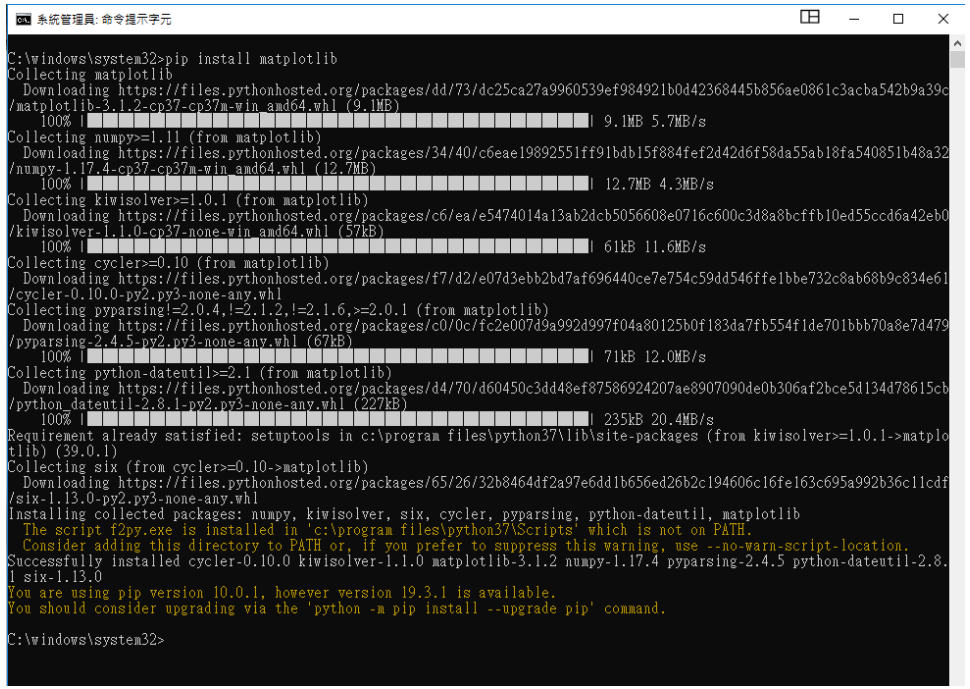
安裝步驟2 - 輸入安裝指令

- 輸入：**pip install matplotlib**
- 按下 Enter 鍵。



安裝步驟3 - 下載過程

- 安裝過程如圖：
  - 下載需花一點點時間，若出現紅字代表安裝失敗
  - 安裝完成後不要關掉此視窗，等等後面還會用到。



安裝步驟4 - 下載中文字型及設定程式

- 自 Google Noto Fonts 頁面下載
  - Noto Sans CJK TC 字型包
- 自本教材指定頁面下載
  - one_click_to_move_font.py
- 將兩個檔案放在同一個資料夾
  - 建議放在「下載」資料夾，以接續下一頁投影片的操作。

NotoSansCJKtc-Light

one_click_to_move_font

安裝步驟5 - 執行設定程式

回到「命令提示字元」視窗。

- 先輸入： **cd C:\Users\User\Downloads**
- 再輸入： **one_click_to_move_font.py**

程式會自動執行完成中文字型路徑及相關設定

```
C:\windows\system32>cd C:\Users\User\Downloads

C:\Users\User\Downloads>one_click_to_move_font.py

C:\Users\User\Downloads>
```

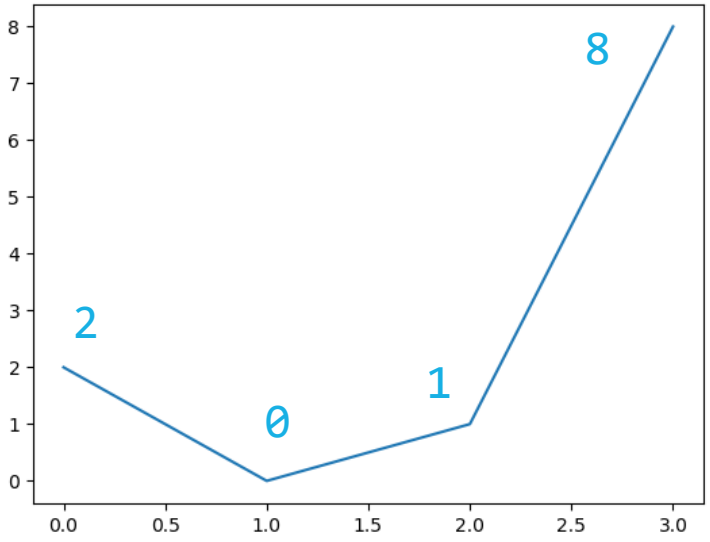
專題子任務說明

- 任務一
  - 繪製中職球員歷年數據折線圖。
- 任務二
  - 繪製中職球員數據散布圖。
- 任務三
  - 繪製 108 UVL 球員球據長條圖。

折線圖：plot()

- 就這樣，三行程式碼給你畫出個折線圖。

```
import matplotlib.pyplot as plt

plt.plot([2, 0, 1, 8])      #繪製 2, 0, 1, 8 四個資料
plt.show()                  #沒有呼叫 show() 就不會秀出來喔
```



最簡單的傳法，把資料點 (就是 y) 傳入 plot()。

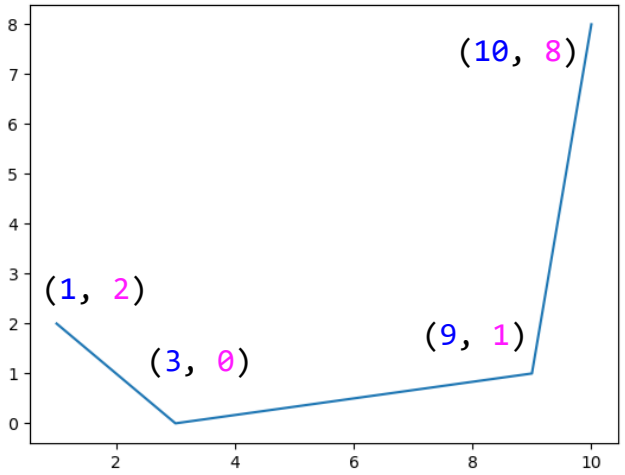
x 座標點預設是 0, 1, 2, ...。

如果只想快速看一個趨勢，咻一下就有得看了。

折線圖：plot()

- 指定 x 座標值。

```
import matplotlib.pyplot as plt

x = [1, 3, 9, 10]
y = [2, 0, 1, 8]
plt.plot(x, y)    #若 x 和 y 長度不同會發生執行錯誤
plt.show()
```



想想看，如果資料是這樣：
data = [1, 2, 3, 0, 9, 1, 10, 8]
該怎麼拿出 x 和 y 資料值？

如果是這樣呢？
data = [[1, 2], [3, 0], [9, 1], [10, 8]]

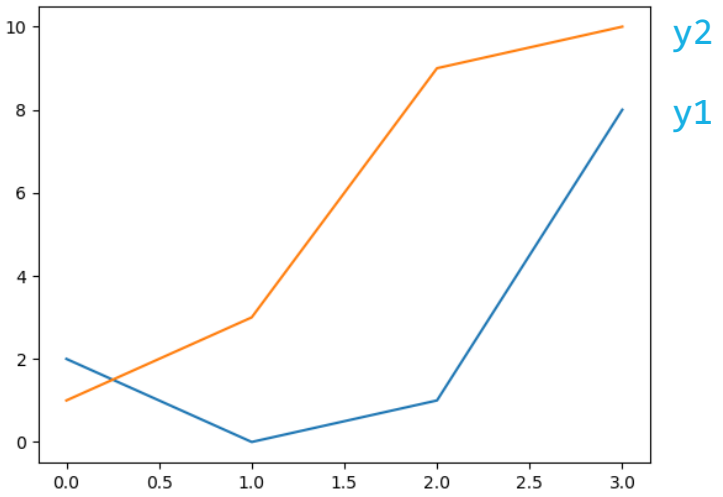
參考答案1 x = data[::2]

參考答案2 x = [e[0] for e in data]

折線圖：plot()

- 多組資料也沒問題。

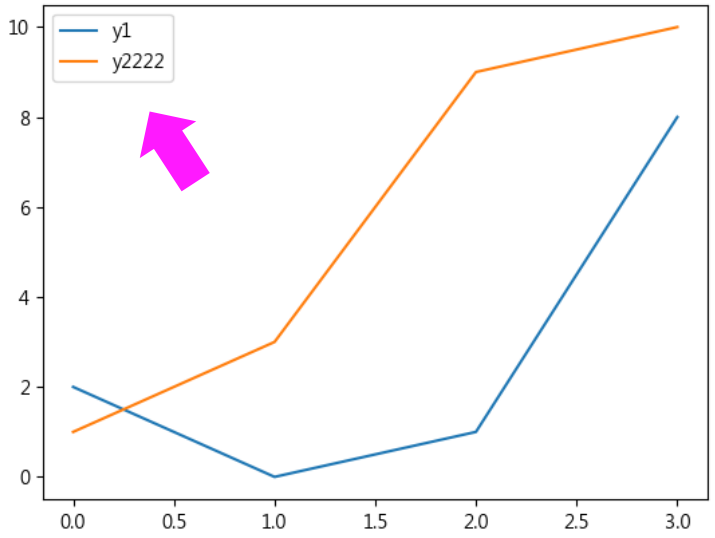
```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
y2 = [1, 3, 9, 10]
plt.plot(y1)
plt.plot(y2)
plt.show()
```



通用：資料標籤

- 使用 plot() 的參數 label。
- 呼叫 plt.legend()。
  - 沒有呼叫 legend() 就不會顯示出左上那塊囉。

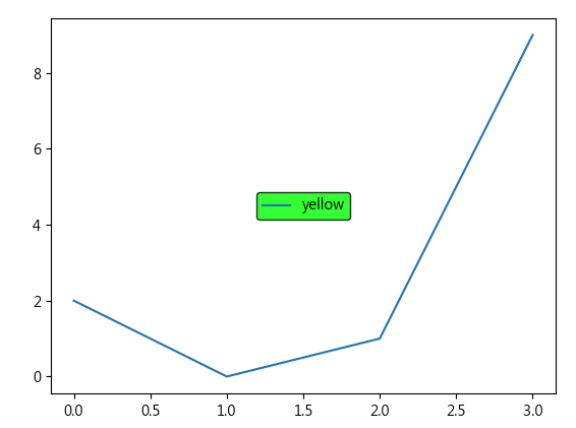
```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
y2 = [1, 3, 9, 10]
plt.plot(y1, label='y1')
plt.plot(y2, label='y2222')
plt.legend()
plt.show()
```



通用：資料標籤

- 圖例 legend() 參數

參數名稱	可用值與範例
ncol	圖例以幾欄顯示，預設值為 1。
fontsize	圖例字型大小。
facecolor / edgecolor	圖例字型/邊框顏色。
loc	圖例位置：'best', 'upper/lower right/left', 'right', 'center left/right', 'lower/upper center', 'center'



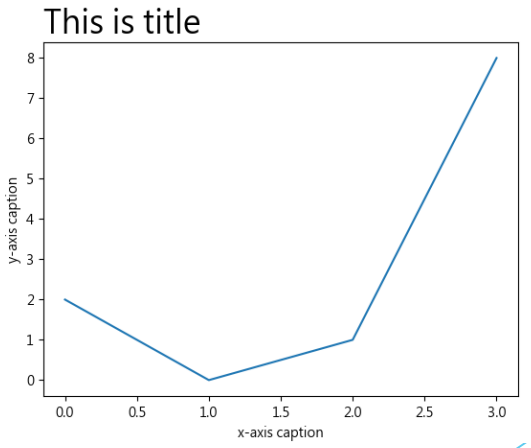

```
import matplotlib.pyplot as plt

plt.plot([2, 0, 1, 9], label='yellow')
plt.legend(fontsize=10, facecolor='#00ff00',
           edgecolor='black', loc='center')
plt.show()
```

通用：標題

- 呼叫 plt 模組的函式

函式名稱	說明	參數（還有很多問 google 大神）
title()	圖表標題	fontsize 字型大小 loc 文字位置：left center right
xlabel()	x 軸標題	fontsize 字型大小
ylabel()	y 軸標題	fontsize 字型大小

```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
plt.plot(y1, label='y1')

plt.title('This is title', fontsize=24, loc='left')
plt.xlabel('x-axis caption', fontsize=10)
plt.ylabel('y-axis caption', fontsize=10)

plt.show()
```



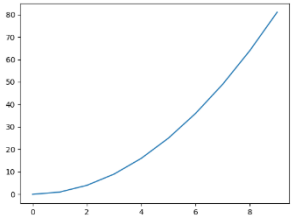
通用：圖表刻度

- 呼叫 plt 模組的函式

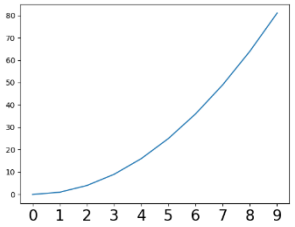
函式名稱	說明
xticks()	x 軸刻度，傳入 x 軸要顯示的數值。
yticks()	y 軸刻度，傳入 y 軸要顯示的數值。

```
import matplotlib.pyplot as plt

plt.plot(range(10), [e**2 for e in range(10)])
plt.show()
```

0, 1, 2, …, 9　　0, 1, 4, …, 81

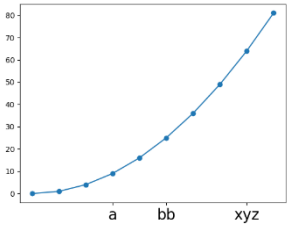


```
import matplotlib.pyplot as plt

plt.plot(range(10), [e**2 for e in range(10)])
plt.xticks(range(10), fontsize=20)
plt.show()
```




```
import matplotlib.pyplot as plt

plt.plot(range(10), [e**2 for e in range(10)], 'o-')
plt.xticks([3, 5, 8], ['a', 'bb', 'xyz'], fontsize=20)
plt.show()
```

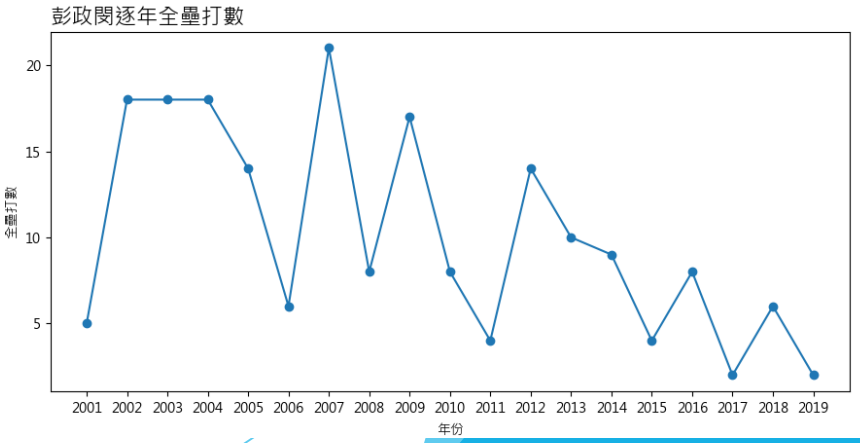
在 3, 5, 8 的刻度顯示 'a', 'bb', 'xyz'



子任務一：中職逐年全壘打數折線圖

請下載 *中職全壘打資料範例.txt*。

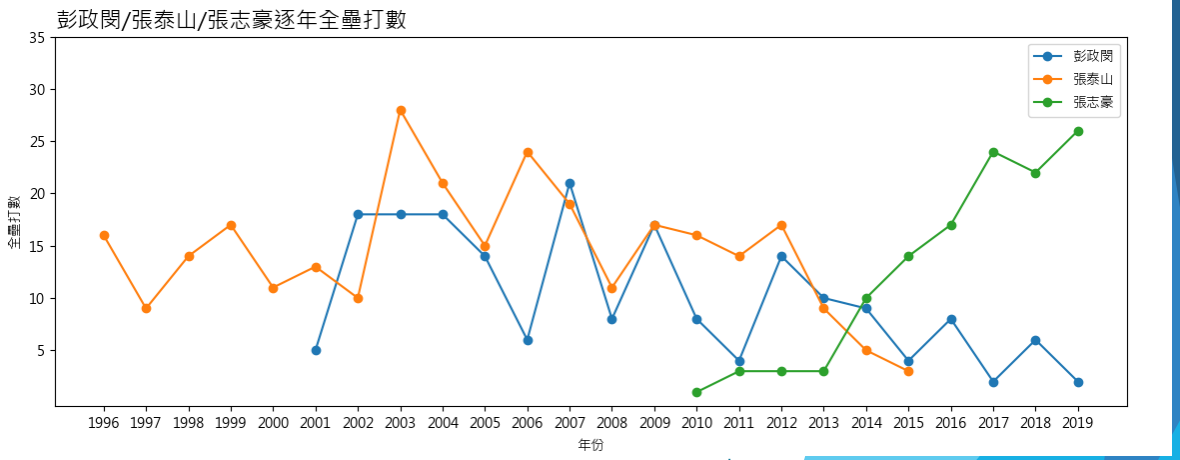
```
import matplotlib.pyplot as plt

data = []
with open('中職全壘打資料範例.txt', encoding='utf8') as f:
    for line in f:
        data.append(line.split())

year = []
HR = []
for e in data:
    if e[1] == '彭政閔':
        year.append(int(e[0]))
        HR.append(int(e[2]))

plt.figure(figsize=(12, 6))
plt.plot(year, HR, marker='o')
plt.xticks(year)
plt.xlabel('年份')
plt.yticks([5, 10, 15, 20])
plt.ylabel('全壘打數')
plt.title('彭政閔逐年全壘打數', loc='left', fontsize=16)
plt.show()
```



子任務一：中職逐年全壘打數折線圖

請下載 *中職全壘打資料範例.txt*。

```
import matplotlib.pyplot as plt

data = []
with open('中職全壘打資料範例.txt', encoding='utf8') as f:
    for line in f:
        data.append(line.split())

plt.figure(figsize=(12, 6))
players = ['彭政閔', '張泰山', '張志豪']
for name in players:
    year = []
    HR = []
    for e in data:
        if e[1] == name:
            year.append(int(e[0]))
            HR.append(int(e[2]))
    plt.plot(year, HR, marker='o', label=name)

plt.xticks(range(1996, 2020))
plt.xlabel('年份')
plt.yticks(range(5, 40, 5))
plt.ylabel('全壘打數')
plt.legend()
plt.title('/'.join(players)+'逐年全壘打數', loc='left', fontsize=16)
plt.show()
```

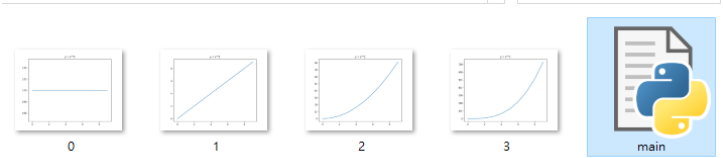


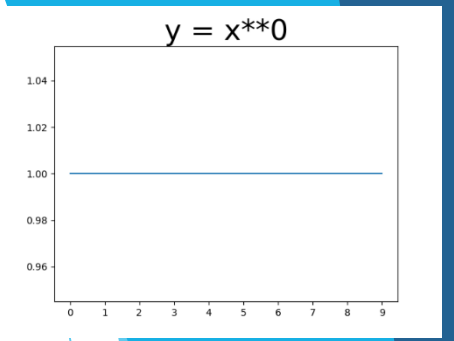
通用：儲存圖檔

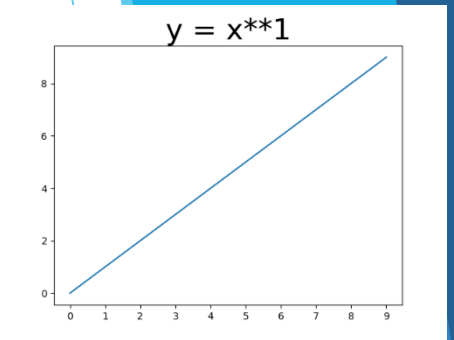
- 呼叫 plt 模組的函式

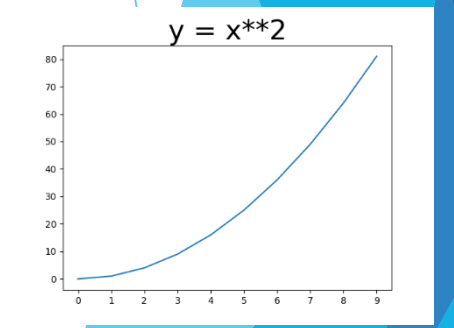
函式名稱	說明
clf()	清除圖表內容 (clear figure)。
savefig()	傳入圖檔檔名，支援 png, pdf, eps, pgf, ps, raw, rgba, svg, svgz 格式。

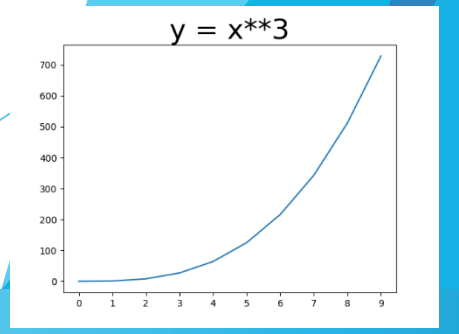
```
import matplotlib.pyplot as plt

for i in range(4):
    x = range(10)
    y = [e**i for e in x]
    plt.clf()
    plt.title('y = x**' + str(i))
    plt.plot(x, y)
    plt.savefig(str(i) + '.png')
```











撰寫程式快速繪製多個圖檔來放進書面報告或投影片簡報，是不是很過癮呢！

以前要針對 Excel 裡的資料一組一組慢慢拉，學會程式後，一切更有效率了！

子任務一：中職逐年全壘打數折線圖

```
import matplotlib.pyplot as plt

data = []
with open('中職全壘打資料範例.txt', encoding='utf8') as f:
    for line in f:
        data.append(line.split())

players = ['彭政閔', '張泰山', '張志豪']
for name in players:
    year = []
    HR = []
    for e in data:
        if e[1] == name:
            year.append(int(e[0]))
            HR.append(int(e[2]))

    plt.figure(figsize=(12,5))
    plt.plot(year, HR, marker='o', label=name)
    plt.xticks(range(1996, 2020))
    plt.xlabel('年份')
    plt.yticks(range(5, 40, 5))
    plt.ylabel('全壘打數')
    plt.legend()
    plt.title(name+'逐年全壘打數', loc='left', fontsize=16)
    plt.savefig(name+'.png')
    plt.show()
```

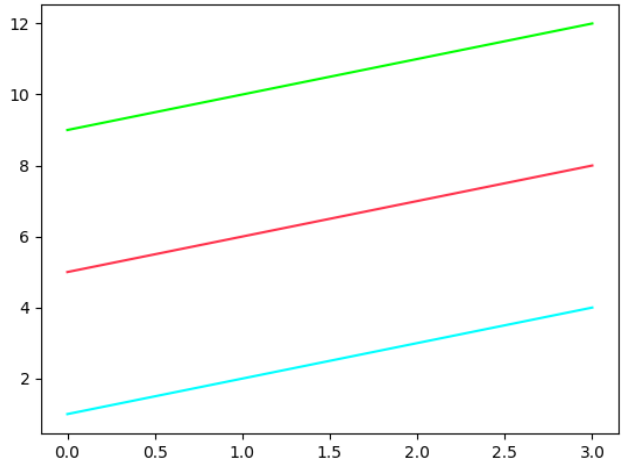
請下載 *中職全壘打資料範例.txt*。

注意如果要 show()，要放在 savefig() 之後。否則關閉視窗後再 savefig() 會存到一張空白圖表。

微調：折線圖線條

- 線條顏色

參數名稱	可用值與範例
color	字串：blue, green, red, cyan, magenta, yellow, black, white 三原色浮點數： (1.0, 0.2, 0.3) 三原色十六進位： #000000 ~ #FFFFFF plt.plot([9, 10, 11, 12], color='#00FF00') plt.plot([5, 6, 7, 8], color=(1.0, 0.2, 0.3)) plt.plot([1, 2, 3, 4], color='cyan')

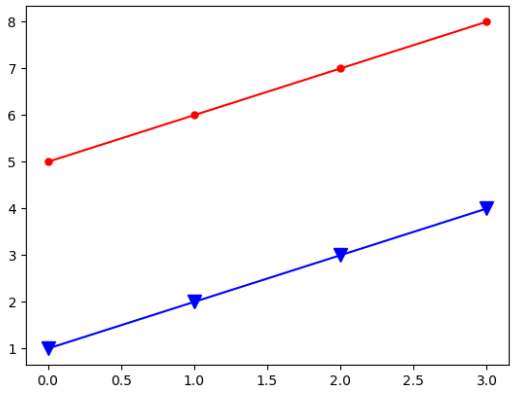


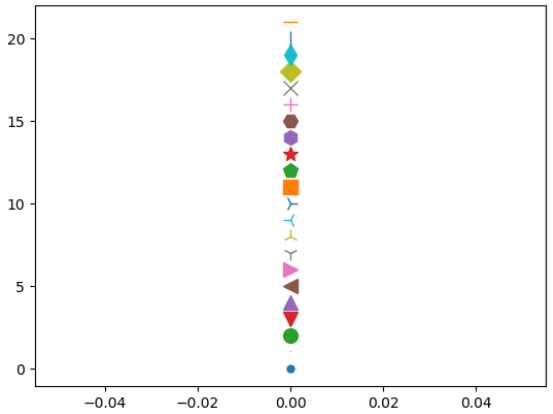
微調：折線圖線條

- 線條符號

參數名稱	可用值與範例
marker	字串：. , o v ^ < > 1 2 3 4 s p * h H + x D d | _

markersize	`plt.plot([5, 6, 7, 8], color='red', marker='o', markersize=5)`
	`plt.plot([1, 2, 3, 4], color='blue', marker='v', markersize=10)`
	`plt.show()`
	
	`mstr = '.,ov^<>1234sp*hH+xDd|_'`
	`for i in range(len(mstr)):`
	`  plt.plot([i], marker=mstr[i], markersize=10)`
	`plt.show()`

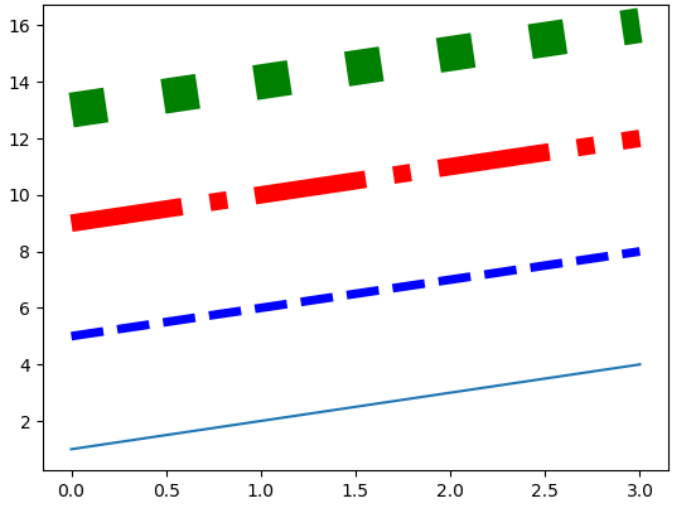




微調：折線圖線條

- 線條樣式

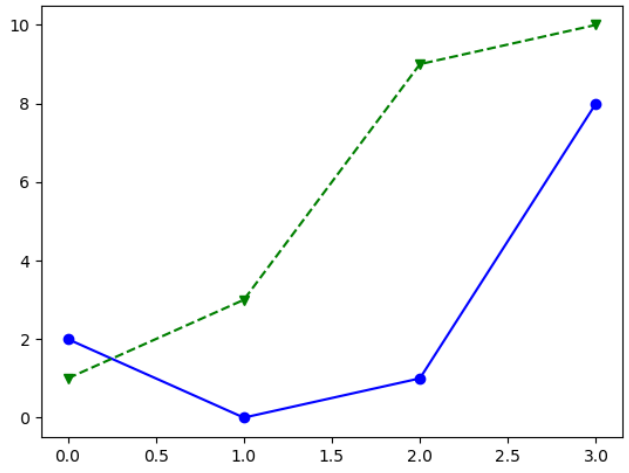
參數名稱	可用值與範例
linestyle	字串：- -- -. :
linewidth	plt.plot([13, 14, 15, 16], color='green', linestyle=':', linewidth=20) plt.plot([9, 10, 11, 12], color='red', linestyle='-.', linewidth=10) plt.plot([5, 6, 7, 8], color='blue', linestyle='--', linewidth=5) plt.plot([1, 2, 3, 4])



微調：折線圖線條

- 線條簡易版字串傳法
  - 顏色 (color)、符號 (marker) 和線條樣式 (linestyle) 三位一體！

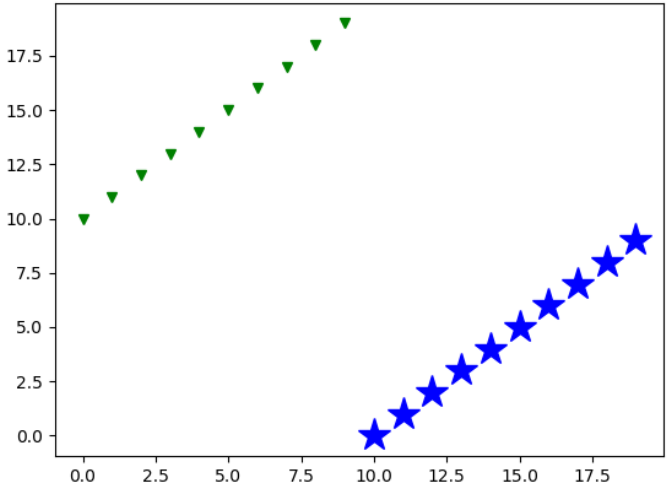
```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
y2 = [1, 3, 9, 10]
plt.plot(y1, 'bo-')
plt.plot(y2, 'gv--')
plt.show()
```



散佈圖：plot()

- 散佈圖可用於觀察兩個維度資料的關係。
- 使用 plot() 便可繪製。

```
import matplotlib.pyplot as plt

x = range(10)
y = range(10, 20)

plt.plot(x, y, 'gv')
plt.plot(y, x, 'b*', markersize=20)
plt.show()
```

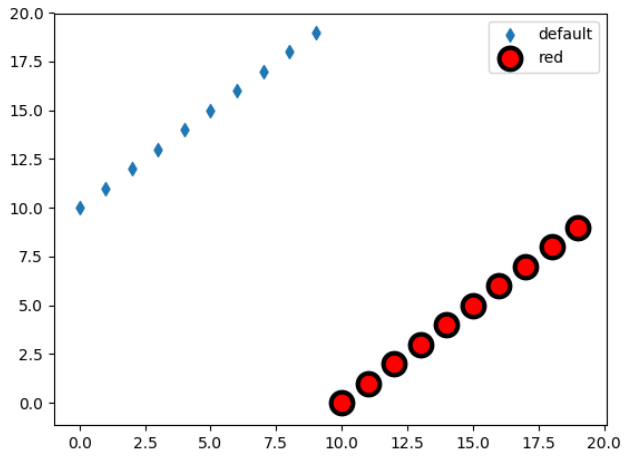


不要傳入線條樣式，plot() 就不會連線，自然而然就變成散佈圖的樣子囉。

散佈圖：scatter()

- scatter() 是一個專門繪製散佈圖的函式。

參數名稱	可用值與範例
marker	字串：. , o v ^ < > 1 2 3 4 8 s p * h H + x D d | _
s	符號大小
c	符號顏色
linewidths	符號邊框寬度
edgecolors	符號邊框顏色
label	符號標籤

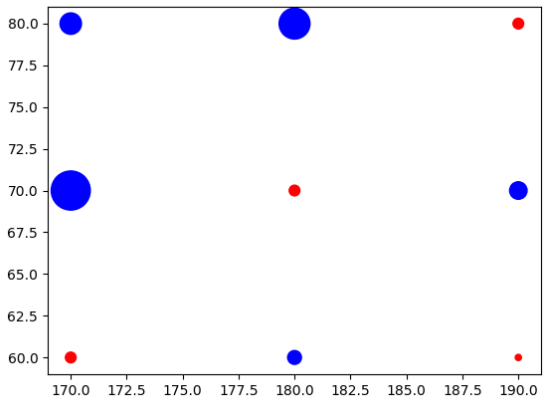



```
import matplotlib.pyplot as plt
x = range(10)        #0, 1, ..., 9
y = range(10, 20)    #10, 11, ..., 19

plt.scatter(x, y, marker='d', label='default')
plt.scatter(y, x, s=200, c='red', linewidths=3, edgecolors='black', label='red')
plt.legend()
plt.show()
```

散佈圖：scatter()

- scatter() 可以指定個別資料點的顏色與大小。

```
import matplotlib.pyplot as plt

#[身高, 體重, 人數]
h_w_c = [[170, 80, 12], [180, 80, 25], [190, 80, 3],
         [170, 70, 40], [180, 70, 3], [190, 70, 8],
         [170, 60, 3], [180, 60, 5], [190, 60, 1]]

x = [e[0] for e in h_w_c]      #以身高作為 x軸
y = [e[1] for e in h_w_c]      #以體重作為 y軸
s = [e[2]*20 for e in h_w_c]  #以人數作為符號大小
c = ['blue' if e[2]>3 else 'red' for e in h_w_c ]

plt.scatter(x, y, s, c)
plt.show()
```




```
c = ['blue' if e[2]>3 else 'red' for e in h_w_c ]
這一行也可寫作
for e in h_w_c:
    if e[2]>3:
        c.append('blue')
    else:
        c.append('red')
```

alpha 透明度：試試在上面的 scatter 呼叫多加 alpha=0.3。

子任務二：中職球員能力散佈圖

```
import csv
import matplotlib.pyplot as plt

AVG = []
OBP = []
year = '2019'
with open('cpbl-player-batter-1990-2019.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for e in reader:
        if e['YEAR']==year:
            AVG.append(float(e['AVG']))
            OBP.append(float(e['OBP']))
```

請下載 *cpbl-player-batter-1990-2019.csv*。

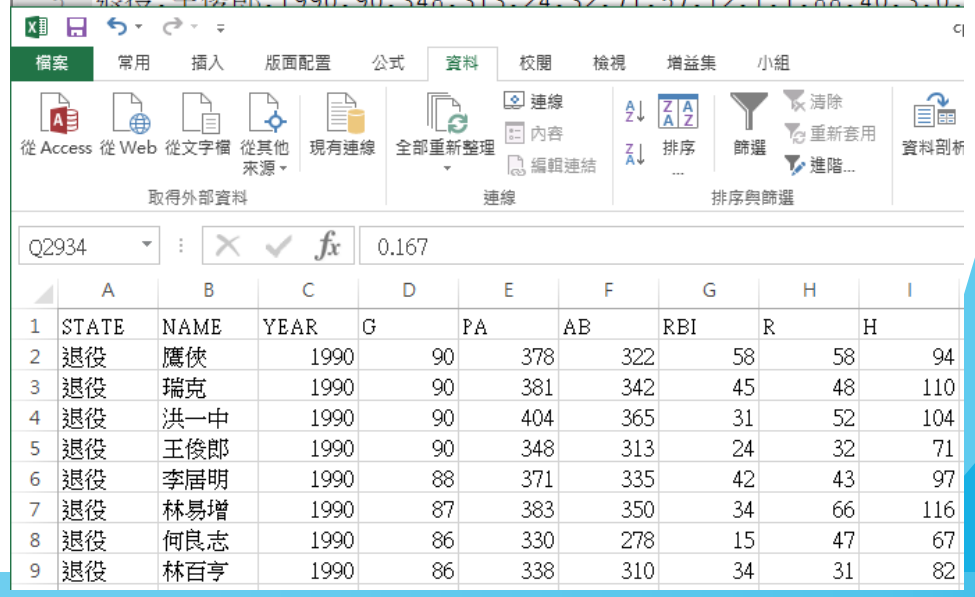
此例中使用到專門用來讀取 csv 檔案的 csv 模組。請到 cmd 視窗下 **pip install csv** 命令即可安裝。

csv 模組的 DictReader() 函式會將檔案以 csv 檔式讀入，並以第一列作為索引。

例如檔案中第一列有 STATE、NAME、YEAR 等等，我們就可以對第二列開始的每一列資料取出該欄位的值，像程式中的 e['YEAR'] 就會取出一列中的 YEAR 欄位值 – 以右邊檔案為例，第二列的 'NAME' 欄位是鷹俠，'YEAR' 欄是 1990。

cpbl-player-batter-1990-2019.csv

```
STATE,NAME,YEAR,G,PA,AB,RBI,R,H,1B,2B,3B,HR,TB,SO,SB,AVG,
退役,鷹俠,1990,90,378,322,58,58,94,52,21,3,18,175,58,7,0.
退役,瑞克,1990,90,381,342,45,48,110,68,33,2,7,168,37,7,0.
退役,洪一中,1990,90,404,365,31,52,104,78,22,4,0,134,17,5,
退役,王俊郎,1990,90,348,313,24,32,71,57,12,1,1,88,40,3,0,
```

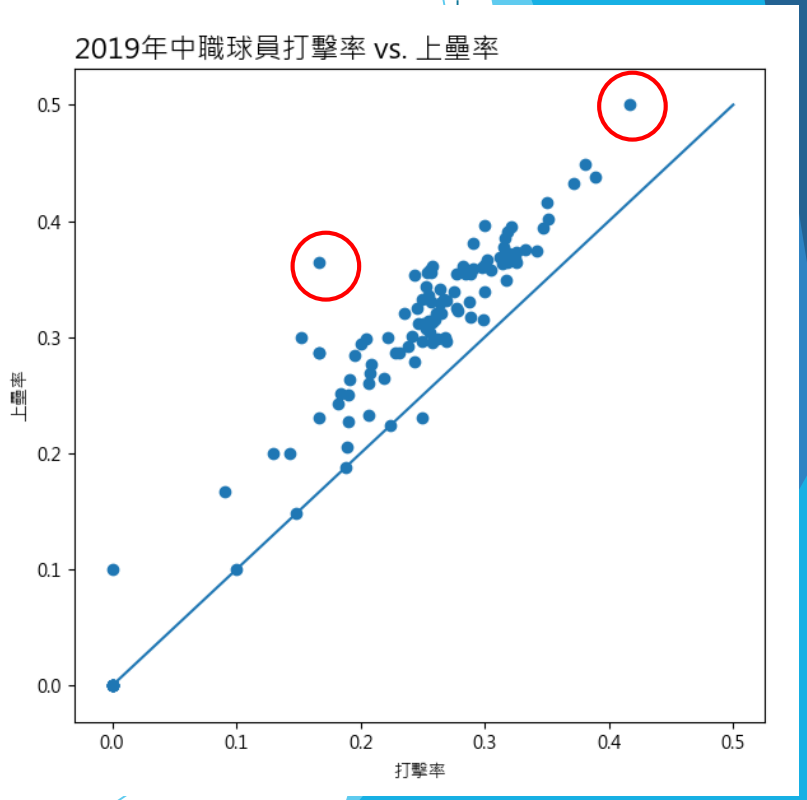


	A	B	C	D	E	F	G	H	I
1	STATE	NAME	YEAR	G	PA	AB	RBI	R	H
2	退役	鷹俠	1990	90	378	322	58	58	94
3	退役	瑞克	1990	90	381	342	45	48	110
4	退役	洪一中	1990	90	404	365	31	52	104
5	退役	王俊郎	1990	90	348	313	24	32	71
6	退役	李居明	1990	88	371	335	42	43	97
7	退役	林易增	1990	87	383	350	34	66	116
8	退役	何良志	1990	86	330	278	15	47	67
9	退役	林百亨	1990	86	338	310	34	31	82

子任務二：中職球員能力散佈圖

```
import csv
import matplotlib.pyplot as plt

AVG = []
OBP = []
year = '2019'
with open('cpbl-player-batter-1990-2019.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for e in reader:
        if e['YEAR']==year:
            AVG.append(float(e['AVG']))
            OBP.append(float(e['OBP']))

plt.figure(figsize=(7,7))
plt.scatter(AVG, OBP)
plt.plot([0, 0.5], [0, 0.5])
plt.xlabel('打擊率')
plt.ylabel('上壘率')
plt.title(year+'年中職球員打擊率 vs. 上壘率', fontsize=16, loc='left')
plt.show()
```

請下載 *cpbl-player-batter-1990-2019.csv*。

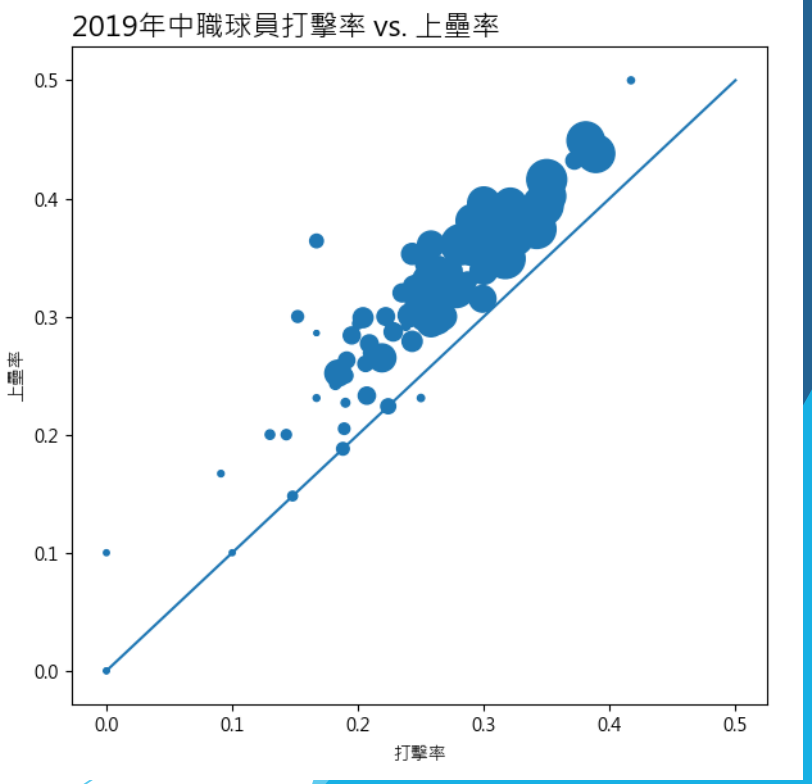


有兩個很特別的人，他們是誰呀？
原來是打席只有 14 席的柯育民和 56 席的朱元勤。

子任務二：中職球員能力散佈圖

請下載 *cpbl-player-batter-1990-2019.csv*。

```
import csv
import matplotlib.pyplot as plt

AVG = []
OBP = []
PA = []
year = '2019'
with open('cpbl-player-batter-1990-2019.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for e in reader:
        if e['YEAR']==year:
            AVG.append(float(e['AVG']))
            OBP.append(float(e['OBP']))
            PA.append(int(e['PA']))

plt.figure(figsize=(7,7))
plt.scatter(AVG, OBP, s=PA)
plt.plot([0, 0.5], [0, 0.5])
plt.xlabel('打擊率')
plt.ylabel('上壘率')
plt.title(year+'年中職球員打擊率 vs. 上壘率', fontsize=16, loc='left')
plt.show()
```

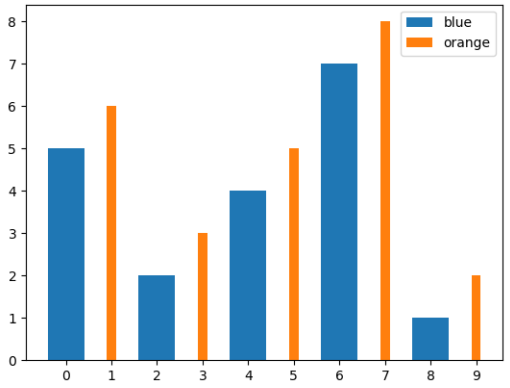


如果我們把打席作為圓圈大小，就可以很快得知那是因為打席太少的偏誤資料。
你還可以再想想要以什麼作為顏色，呈現更多的資訊。

長條圖：bar()

- 長條圖用來比較多組資料的數值大小。
- bar() 函式可以繪製長條圖。

參數名稱	可用值與範例
x	x軸座標值
height	y軸座標值
width	長條寬度（預設 0.8）
label	標籤

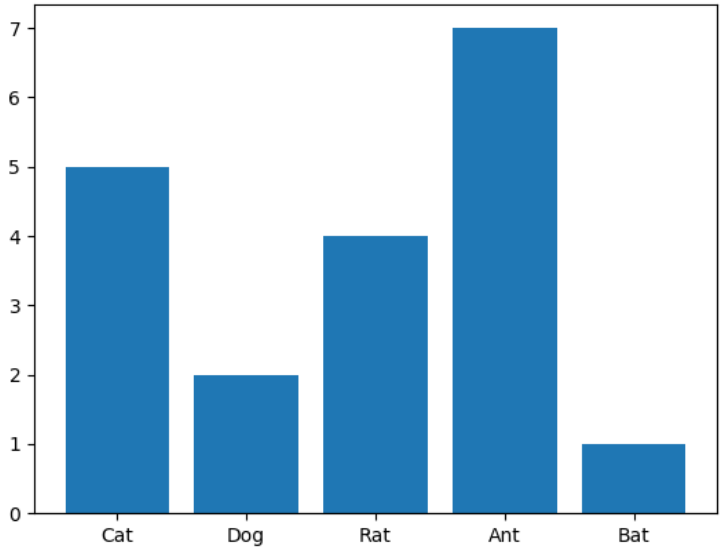



```
import matplotlib.pyplot as plt

counts = [5, 2, 4, 7, 1]
x = range(0, 10, 2) #0, 2, 4, 6, 8

plt.bar(x, counts, label='blue')
plt.bar(x=[e+1 for e in x], height=[e+1 for e in counts], width=0.2, label='orange')
plt.xticks(range(10))
plt.legend()
plt.show()
```

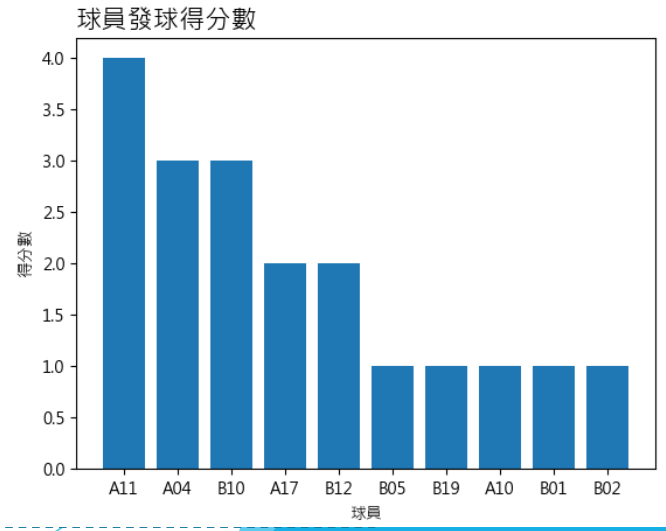
長條圖：bar()

- 使用 xticks() 來指定位置上的文字。

```
import matplotlib.pyplot as plt

data = [['Cat', 5], ['Dog', 2], ['Rat', 4], ['Ant', 7], ['Bat', 1]]
plt.bar(range(len(data)), [e[1] for e in data])
plt.xticks(range(len(data)), [e[0] for e in data])
plt.show()
```



子任務三：排球發球得分長條圖

請下載 UVL108-Champion-Set[1,2,3].txt。

```
import matplotlib.pyplot as plt
from collections import Counter

results = []
for i in range(1, 4):
    oneset = []
    with open('UVL108-Champion-Set'+str(i)+'.txt') as f:
        for line in f:
            oneset.append(line.split())

    for ball in oneset:
        if ball[0][0] == ball[-1][0] and ''.join(ball).count(ball[0][0])==2:
            results.append(ball[0])

ct = Counter(results)
data = ct.most_common()
for e in data:
    print('球員', e[0], '發球得分', e[1], '次')

plt.title('球員發球得分數', fontsize=16, loc='left')
plt.xlabel('球員')
plt.ylabel('得分數')
plt.bar(range(len(data)), [e[1] for e in data])
plt.xticks(range(len(data)), [e[0] for e in data])
plt.show()
```