

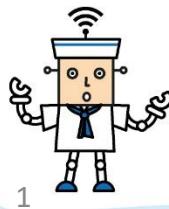


第4章

初探資料科學（一）：

用 pandas 做資料前處理

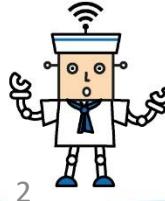
在第三章我們學會了如何取得資料、轉成 pandas 的資料框，並儲存和讀取 CSV 檔。在本章中，將以學會處理 pandas 資料框結構內的資料為目標，持續介紹利用資料框的功能進行資料前處理。



4-1 常見的資料處理工作

4-1-1 資料觀察

- 搜集到的資料難免會有些不完整，通常拿到資料之後要先進行**檢視**，若察覺有誤就需加以處理。
- 常見的資料處理工作有**觀察**、**過濾**、**刪除重複資料**和**補缺失值**。
- pandas提供如：**head()**、**info()**、**describe()**、**duplicated()**等觀察函式，避免人為因素造成的遺漏。



4-1-1 資料觀察

info()
函式

印出資料框各行(欄)所包含的內容
資料框名稱.info()



可以了解是否需要
進行資料修補

describe()
函式

印出資料框相關的統計數據
資料框名稱.describe()

- 贝壳 icon **count**：行的資料個數。
- 贝壳 icon **mean**：行的資料平均值。
- 贝壳 icon **std**：行的資料標準差。
- 贝壳 icon **min**：行的資料最小值。
- 贝壳 icon **max**：行的資料最大值。
- 贝壳 icon **25%**：行的資料由小到大排名前 25% 的值。
- 贝壳 icon **50%**：行的資料由小到大排名前 50% 的值，即中位數。
- 贝壳 icon **75%**：行的資料由小到大排名前 75% 的值。



4-1-1 資料觀察

duplicated()
函式

檢測重複的記錄(列資料)

資料框名稱.duplicated()

執行這個敘述得到True(真)時，代表那一筆編號(列索引)
的資料和另一筆資料重複。



drop_duplicates()
函式

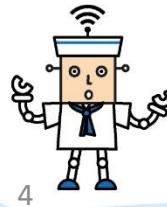
刪除重複的記錄(列資料)

資料框名稱2 = 資料框名稱1.drop_duplicates()

重複的刪除!



1. 資料框名稱相同時，會將刪除後的結果直接更新到原資料框。
2. 二者不相同則將結果更新到資料框2，而原資料框1並不會改變。





觀察資料框和刪除重複記錄

EX4-1.1.ipynb

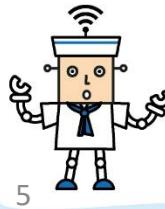
01

首先讀取 Google 雲端硬碟中的 CSV 檔案「學生成績檔 -4-1.1.csv」並轉成資料框型別，呼叫 head() 函式顯示前 5 筆記錄，也可以使用 print(df) 敘述顯示所有的記錄。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-1.1.csv')  
5 df.head()
```

Drive already mounted at /content/MyGoogleDrive; to attempt to forcibly remount, call drive.mount

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
2	1080003	何宜敏	女	1080003@sun.tc.edu.tw	82	87.0	86.0	82	82.0
3	1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0
4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	NaN	92.0	97	98.0





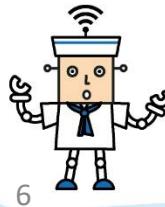
觀察資料框和刪除重複記錄

EX4-1.1.ipynb

02

接著呼叫 `info()` 函式印出各行所包含的內容，仔細觀察之後發現以下的問題：

- (1) 全班共有 40 位同學，但 ID、name、sex、email、第 1 次平時考、第 4 次平時考，怎麼多出一筆記錄變 41 筆呢？
- (2) 第 2 次平時考：以表中 41 筆記錄來說，少了兩筆記錄？
- (3) 第 3 次平時考、第 5 次平時考：以表中有 41 筆記錄來說，也各少了一筆記錄？





實作

觀察資料框和刪除重複記錄

EX4-1.1.ipynb



1 df.info()

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 0 to 40
Data columns (total 9 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   ID        41 non-null      int64  
 1   name      41 non-null      object  
 2   sex       41 non-null      object  
 3   email     41 non-null      object  
 4   第1次平時考 41 non-null    float64
 5   第2次平時考 39 non-null    float64
 6   第3次平時考 40 non-null    float64
 7   第4次平時考 41 non-null    int64  
 8   第5次平時考 40 non-null    float64
dtypes: float64(3), int64(3), object(3)
memory usage: 3.0+ KB
```

多一個分數

多出一筆

以表中 41 筆
記錄來說，少
兩個分數

少一個分數



資料科學 × 機器學習

實戰探索

Practical Exploration





觀察資料框和刪除重複記錄

EX4-1.1.ipynb

03

呼叫 `describe()` 函式後，也可以從下圖中 `count` 的數據發現如 `info()` 函式顯示的問題，另外還有一些統計上的數據，例如：從 `mean` 可以發現全班成績逐漸進步中。



1 df.describe()

	ID	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
count	4.100000e+01	41.000000	39.000000	40.000000	41.000000	40.000000
mean	1.080020e+06	80.341463	82.564103	83.375000	83.487805	84.200000
std	1.176342e+01	21.096931	17.784965	18.077735	20.975130	21.367601
min	1.080001e+06	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
25%	1.080010e+06	81.000000	81.000000	80.750000	85.000000	83.750000
50%	1.080020e+06	86.000000	87.000000	86.500000	87.000000	88.500000
75%	1.080030e+06	91.000000	90.500000	93.500000	93.000000	94.000000
max	1.080040e+06	97.000000	100.000000	100.000000	100.000000	100.000000

有學生缺考？

全班 40 位，怎有 41 筆？

平均而言，成績逐漸進步？



資料科學 × 機器學習

實戰探索

Practical Exploration





觀察資料框和刪除重複記錄

EX4-1.1.ipynb

04

呼叫 `duplicated()` 函式檢查資料有無重複的情形，結果發現內定列索引（編號）6 和別的資料重複了。

1 `df.duplicated()`

```
0    False  
1    False  
2    False  
3    False  
4    False  
5    False  
6    True  
7    False  
8    False  
9    False  
10   False
```

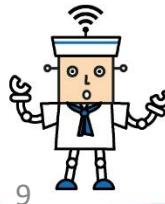
表示此筆記錄
和其他重複



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

觀察資料框和刪除重複記錄

EX4-1.1.ipynb

05

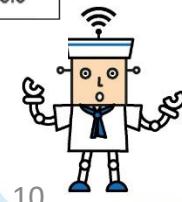
呼叫 `drop_duplicates()` 函式將重複的列資料刪除，接著再次呼叫 `info()` 函式查看，發現資料個數已符合。

```
1 df = df.drop_duplicates()  
2 df.info()
```

```
<class 'pandas.core  
Int64Index: 40 ent  
Data columns (total  
#   Column Non-  
---  
0   ID      40 no  
1   name    40 no  
2   sex     40 no  
3   email   40 no  
4   第1次平時考 40  
5   第2次平時考 38  
6   第3次平時考 39  
7   第4次平時考 40  
8   第5次平時考 39  
dtypes: float64(3)  
memory usage: 3.1+  
  
1 df
```

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
1080003	何宜敏	女	1080003@sun.tc.edu.tw	82	87.0	86.0	82	82.0
1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0
1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	NaN	92.0	97	98.0
1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0
1080007	李宇綸	女	1080007@sun.tc.edu.tw	82	76.0	89.0	87	85.0
1080008	李憲勝	男	1080008@sun.tc.edu.tw	-1	-1.0	-1.0	-1	-1.0
					57.0	54	53.0	
					84.0	86	83.0	

刪除重複列資料後，列索引（編號）並不會自動遞補更新，在後面 4-14 頁會教您如何解決這一點

已刪除 1 筆
重複記錄



觀察資料框和刪除重複記錄

EX4-1.1.ipynb

06

儲存結果「學生成績檔 -4-1.1-ANS.csv」。

```
1 o_filepath = '/content/MyGoogleDrive/My Drive/Colab Notebooks/資料檔/完成檔/Ch04/'  
2 GooglePath = o_filepath  
3 filename='學生成績檔-4-1.1-ANS.csv'  
4 df.to_csv(GooglePath + filename, index=False)
```

↑
將結果存成 CSV 檔



資料科學 × 機器學習

實戰探索

Practical Exploration





df[df.duplicated()] 的運算過程

df.duplicated(): 逐筆列出是否重複，會傳回各筆是否重複的布林值

0	False
1	False
2	False
3	False
4	False
5	False
6	True
7	False
8	False
9	False
10	False
11	False
12	False
13	False
14	False
15	False

將 df.duplicated() 的
結果傳入 df[] 內

df[df.duplicated()]: 只列出重複 (True) 的記錄

運算如下

df[[False, False, False, ..., True, ...]]
第 0 列 第 1 列 第 2 列 第 6 列

列出重複為 True 的記錄

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
6	1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0



加深
知識

pandas常用的資料觀察函式

shape
函式

印出資料框的列數及行數

資料框名稱.shape

結果是用一對數值 (m, n)
表示資料框有 m 列、 n 行。

▶ 1 df.shape

⇨ (41, 9)

columns
函式

印出資料框的行索引

資料框名稱.columns[N]

[N] 若省略，會印出所有的
行索引。

▶ 1 df.columns

```
⇨ Index(['ID', 'name', 'sex', 'email', '第1次平時考', '第2次平時考', '第3次平時考', '第4次平時考',  
         '第5次平時考'],  
        dtype='object')
```



1 df.columns[4]

⇨ '第1次平時考'



加深
知識

pandas常用的資料觀察函式

index
函式

印出資料框的列索引
資料框名稱.index

執行結果用
(start=X, stop=Y, step=Z)
表示列索引的開始值X、結束值Y
及間隔值Z。



1 df.index

⇨ RangeIndex(start=0, stop=41, step=1)

列出所有性別為「男」的列索引



1 df[df['sex'] == '男'].index

⇨ 54Index([0, 3, 5, 6, 8, 10, 12, 14, 17, 19, 20, 22, 27, 28, 29, 31, 32,
33, 34, 35, 37, 39, 40],
dtype='int64')

4-1-2 資料篩選、刪除列資料與行資料

資料
篩選

篩選資料框資料

運算子

串列名稱 = 資料框名稱['行索引'] == 值

- 針對該行索引的元素逐一檢查，得到結果為「True/False」的序列。
- 搭配使用邏輯運算子可以篩選符合多個條件的記錄。

比較運算子

$==$ (等於)	$>$ (大於)	$>=$ (大於等於)
$!=$ (不等於)	$<$ (小於)	$<=$ (小於等於)

邏輯運算子

$\&$ (且) $|$ (或) \sim (非)

運算子



例 ➤ `filter = df['第1次平時考'] == -1` 篩選第1次平時考缺考(分數是-1)

例 ➤ `x = (df['第1次平時考'] >= 90) & (df['第2次平時考'] < 60)`

篩選「第1次平時考大於等於90分」且「第2次平時考小於60分」



4-1-2 資料篩選、刪除列資料與行資料

drop()
函式

刪除資料框的列資料(記錄)/行資料(欄)

資料框名稱2 = 資料框名稱1.drop(索引, axis = 0或1)

1. 資料框名稱相同時，會將刪除後的結果直接更新到原資料框。
2. 一次要刪除多列(行)資料時，索引要寫成串列。

axis = 0或省略：刪除列資料
axis = 1：刪除行資料



例 ➤ `df = df.drop([1, 3], axis = 0)` 刪除列索引「1」和「3」的兩列資料

也可
寫成 ↗

`a = [1, 3]`

`df = df.drop(a)`

例 ➤ `df = df.drop('第1次平時考', axis = 1)` 刪除行索引為「第1次平時考」的整行資料





實作

資料篩選、刪除列 / 行資料

EX4-1.2.ipynb

01

首先讀取「學生成績檔 -4-1.2.csv」並轉成資料框型別，接著篩選出「第1次平時考」分數被標註為「-1」（缺考）的同學。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-1.2.csv')  
5 filter = df['第1次平時考'] == -1 ← 進行篩選  
6 print(filter)
```

```
Drive already mounted at /content/MyGoogleDrive; to attempt  
0 False  
1 False  
2 False  
3 False  
4 False  
5 False  
6 False  
7 True ← True : 表示此筆記錄的「第1次平  
8 False 時考」分數被標註為「-1」(缺考)  
9 False  
10 False  
11 True ←  
12 False  
13 False
```



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

資料篩選、刪除列 / 行資料

EX4-1.2.ipynb

02

印出所有符合 01 篩選出的同學資料，分別是列索引 7 及 11 二位同學。



1 df[filter]

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
7	1080008	李憲勝	男	1080008@sun.tc.edu.tw	-1	-1.0	-1.0	-1	-1.0
11	1080012	林絜峰	男	1080012@sun.tc.edu.tw	-1	75.0	66.0	-1	-1.0

↑
缺考



資料科學 X 機器學習

實戰探索

Practical Exploration





實作

資料篩選、刪除列 / 行資料

EX4-1.2.ipynb

03

呼叫 `drop()` 函式刪除篩選出來的列索引 7、11 的資料。

```
1 i = df[filter].index #找出所有符合條件的記錄索引  
2 df = df.drop(i, axis = 0)  
3 df
```

列索引 7 及 11 的資料已被刪除

usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will
""Entry point for launching an IPython kernel.

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
2	1080003	何宜敏	女	1080003@sun.tc.edu.tw	82	87.0	86.0	82	82.0
3	1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0
4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	NaN	92.0	97	98.0
5	1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0
6	1080007	李宇綸	女	1080007@sun.tc.edu.tw	82	76.0	89.0	87	85.0
8	1080009	杜以潔	女	1080009@sun.tc.edu.tw	53	56.0	57.0	54	53.0
9	1080010	沈程隆	男	1080010@sun.tc.edu.tw	81	77.0	84.0	86	83.0
10	1080011	沈慈惠	女	1080011@sun.tc.edu.tw	87	84.0	90.0	87	92.0
12	1080013	林保苓	女	1080013@sun.tc.edu.tw	92	94.0	99.0	96	94.0
13	1080014	林宏銘	男	1080014@sun.tc.edu.tw	60	56.0	58.0	85	88.0



資料科學 × 機器學習

實戰探索

Practical Exploration





列索引重新編號

reset_index()
函式

將列索引重新編號

資料框2 = 資料框1.reset_index(drop=True/False)



drop=True : 用來更新原本的內定索引。
drop=False或者省略 : 重新編號內定索引，
並將原索引(有缺漏)
保留下來新增成一行。



列索引重新編號



```
1 df1 = df.drop(2)  
2 df1.head()
```

列索引 2 的記錄被刪除，內定索引（列編號）並不會自動遞補更新



	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
3	1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0
4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	NaN	92.0	97	98.0
5	1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0



列索引重新編號

呼叫 `reset_index()` 將列索引重新編號

```
1 df1 = df1.reset_index()  
2 df1.head()
```

	index	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1	1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
2	3	1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0
3	4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	NaN	92.0	97	98.0
4	5	1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0

`reset_index()` 函式會將列索引自動重新編號

4-1-3 補上缺失值

isnull()
函式

檢查資料框空值的資料格

資料框名稱2 = pd.isnull(資料框名稱1)

1. 檢查「資料框名稱1」後會產生一個由「True/False」
組成的資料框(資料框名稱2)
2. 資料格「True」表示為空值「NaN」



4-1-3 補上缺失值

fillna()
函式

資料框空值資料格的補值

資料框名稱2['行索引'] = 資料框名稱1['行索引'].fillna(值)

1. 將資料框內指定「行索引」中所有的空值「NaN」以fillna()指定的值補值，並產生新的資料框(資料框名稱2)。
2. 資料框名稱相同時，會將補值後的結果直接更新到原資料框。

例 ➤ `df['第2次平時考'] = df['第2次平時考'].fillna(60)`

`df`資料框「第2次平時考」那一行的空值資料格皆給予60





實作

空值資料檢查及補值

EX4-1.3.ipynb

01

首先讀取「學生成績檔 -4-1.3.csv」並轉成資料框型別，呼叫 `isnull()` 函式可以快速檢查出資料框中何處有「NaN」的資料格，例如：「第 2 次平時考」的列索引 4。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-1.3.csv')  
5 df.info()
```

```
Drive already mounted at /content/MyGoogleDrive; to attempt to forcibly remount,  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 38 entries, 0 to 37  
Data columns (total 9 columns):  
 #   Column   Non-Null Count  Dtype     
---  --      --           --  
 0   ID       38 non-null    int64  
 1   name     38 non-null    object  
 2   sex      38 non-null    object  
 3   email    38 non-null    object  
 4   第1次平時考 38 non-null  int64  
 5   第2次平時考 36 non-null  float64  
 6   第3次平時考 37 non-null  float64  
 7   第4次平時考 38 non-null  int64  
 8   第5次平時考 37 non-null  float64  
dtypes: float64(3), int64(3), object(3)  
memory usage: 2.8+ KB
```

原始資料中缺了 2 筆分數



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

空值資料檢查及補值

EX4-1.3.ipynb

```
1 df1 = pd.isnull(df)  
2 df1
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	True	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False

缺的其中一筆是列索列 4，在 df 內的數據為「NaN」



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

空值資料檢查及補值

EX4-1.3.ipynb

02

針對成績是「NaN」的資料格，本例中採用該次考試全班同學的平均分數來補上缺值。`mean()` 函式可用來計算平均，在本章稍後會再做詳細的說明。

```
1 x = df['第2次平時考'].mean()  
2 df['第2次平時考'] = df['第2次平時考'].fillna(x)  
3 df
```

計算第2次平時考的平均

以平均來補值

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0
1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0
2	1080003	何宜敏	女	1080003@sun.tc.edu.tw	82	87.000000	86.0	82	82.0
3	1080004	何志陞	男	1080004@sun.tc.edu.tw	91	92.000000	99.0	91	92.0
4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0
5	1080006	宋緯挺	男	1080006@sun.tc.edu.tw	84	81.000000	83.0	90	88.0
6	1080007	李宇綸	女	1080007@sun.tc.edu.tw	88	70.000000	88.0	87	85.0
7	1080009	杜以潔	女	1080009@sun.tc.edu.tw	81	77.000000	84.0	54	53.0
8	1080010	沈程隆	男	1080010@sun.tc.edu.tw	87	84.000000	90.0	86	83.0
9	1080011	沈蕙惠	女	1080011@sun.tc.edu.tw	81	77.000000	84.0	87	92.0

以平均分數補上空值



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

空值資料檢查及補值

EX4-1.3.ipynb

03

同理，將其他次平時考（每行）缺值的資料格分別補上該次全班的平均分數。

```
1 x = df['第3次平時考'].mean()  
2 df['第3次平時考'] = df['第3次平時考'].fillna(x)  
3 x = df['第5次平時考'].mean()  
4 df['第5次平時考'] = df['第5次平時考'].fillna(x)  
5 df.info()
```

#	Column	Non-Null Count	Dtype
0	ID	38 non-null	int64
1	name	38 non-null	object
2	sex	38 non-null	object
3	email	38 non-null	object
4	第1次平時考	38 non-null	int64
5	第2次平時考	38 non-null	float64
6	第3次平時考	38 non-null	float64
7	第4次平時考	38 non-null	int64
8	第5次平時考	38 non-null	float64

dtypes: float64(3), int64(3), object(3)
memory usage: 2.8+ KB

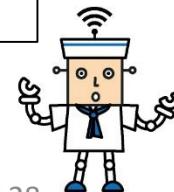
所有缺值資料已全部補上該次的平均分數



資料科學 × 機器學習

實戰探索

Practical Exploration





刪除「NaN」資料格

dropna()
函式

刪除資料框中包含空值(NaN)的列/行資料

資料框名稱 = 資料框名稱.dropna(axis = 0或1)

axis=0或省略：刪除所有包含「NaN」的「列」資料。

axis = 1：刪除所有包含「NaN」的「行」資料。



例▶ `df = df.dropna()` 將df資料框有「NaN」的「列」資料全部刪除

例▶ `df = df.dropna(axis = 1)` 將df資料框有「NaN」的「行」資料全部刪除

4-1-4 資料轉換

對應關係

變數名稱 = {原值:新值, 原值:新值, 原值:新值,...}

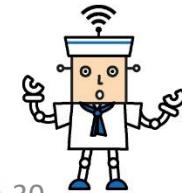
1. { } 括號內可以放一個以上的對應關係。
2. 每個對應關係之間以逗號「,」分隔。



例

`s = {'男': 1, '女': 0}`

將'男'對應為1、「女」對應為0



4-1-4 資料轉換

map()
函式

轉換行資料

資料框名稱2['行索引'] =

資料框名稱1['行索引'].map(對應關係的變數名稱)

- 針對資料框整行資料內的值進行轉換。
- 可將對應關係先儲存於變數，也可以直接寫入。
- 資料框名稱相同時，會將轉換後的結果直接更新到原資料框。

例

$s = \{'男': 1, '女': 0\}$

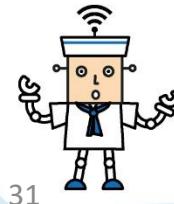
$df['sex'] = df['sex'].map(s)$

將 df 資料框 sex 行 資料 (欄)

'男' → 1、'女' → 0

也可
寫成

$df['sex'] = df['sex'].map(\{'男': 1, '女': 0\})$





實作

資料轉換

EX4-1.4.ipynb

01

首先讀取「學生成績檔 -4-1.4.csv」並轉成資料框型別。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-1.4.csv')  
5 df.head()
```

Drive already mounted at /content/MyGoogleDrive; to attempt to forcibly remount, call drive.mount(")

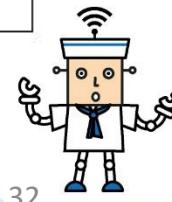
	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0
1	1080002	王倫樺	女	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0
2	1080003	何宜敏	女	1080003@sun.tc.edu.tw	7.000000	86.0	82	82.0	
3	1080004	何志陞	男	1080004@sun.tc.edu.tw	2.000000	99.0	91	92.0	
4	1080005	吳一歌	女	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

資料轉換

EX4-1.4.ipynb

02

進行資料轉換之前，首先要建立一個對應關係，再將建好的對應關係透過 `map()` 函式進行轉換。

```
1 s = {'男': 1, '女':0}
2 df['sex']=df['sex'].map(s)
3 df.head()
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	1	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0
2	1080003	何宜敏	0	1080003@sun.tc.edu.tw	88	87.000000	86.0	82	82.0
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	90	99.000000	99.0	91	92.0
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0



資料科學 × 機器學習

實戰探索

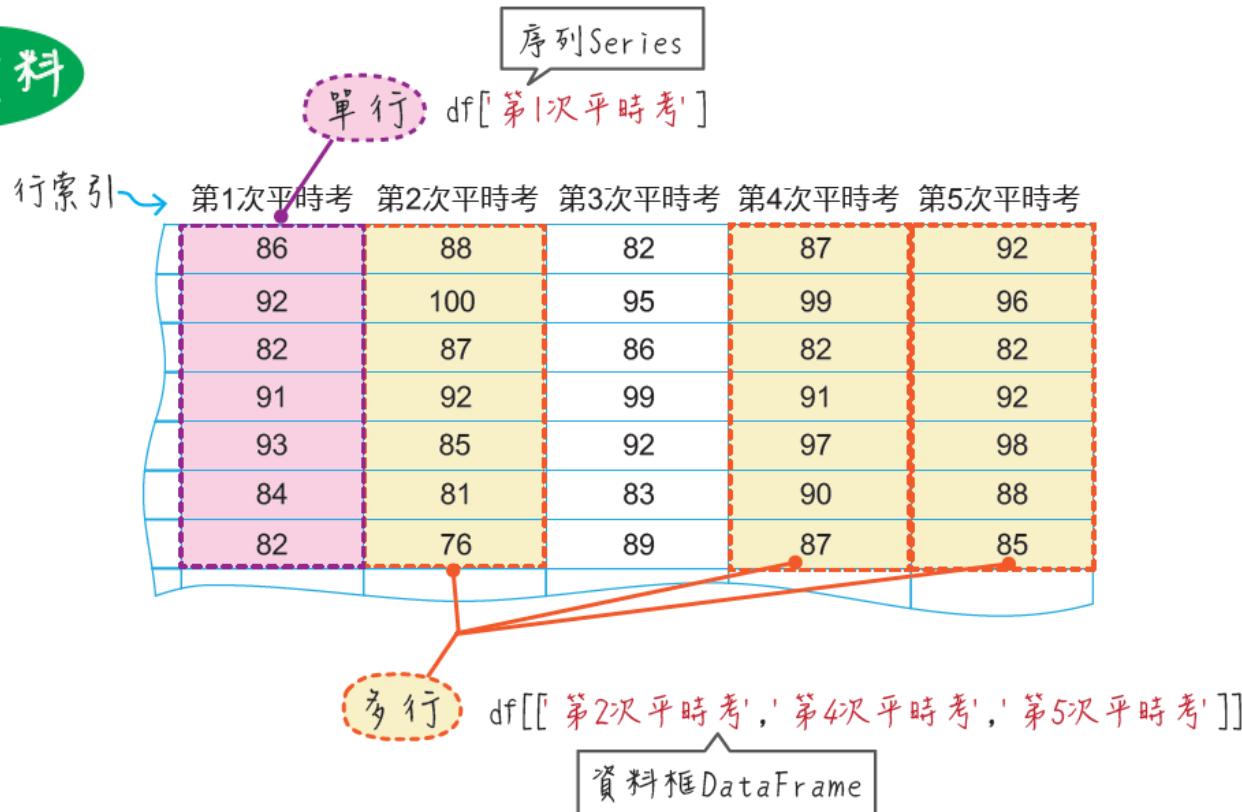
Practical Exploration



4-2 資料框的資料處理

4-2-1 選取行資料

選取行資料





以「自定索引」選取行資料

EX4-2.1.ipynb

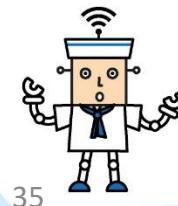
01

首先讀取「學生成績檔 -4-2.1.csv」並轉成資料框型別，選取「第1次平時考」的單行資料，並印出前 5 筆資料。

```
1 from google.colab import drive
2 drive.mount('/content/MyGoogleDrive')
3 import pandas as pd
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.1.csv')
5 df['第1次平時考'].head()
```

印出第1次時平時考的前5筆資料

```
In [1]: Drive already mounted at /content/MyGoogleDrive; to attempt to for
        0    86
        1    92
        2    82
        3    91
        4    93
Name: 第1次平時考, dtype: int64
```





以「自定索引」選取行資料

EX4-2.1.ipynb

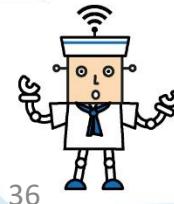
02

底下則是選取所有平時考的多行資料，並印出前 5 筆資料。

```
1 df[['第1次平時考', '第2次平時考', '第3次平時考', '第4次平時考', '第5次平時考']].head()
```

注意前後都是兩個中括號

	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	86	88.000000	82.0	87	92.0
1	92	100.000000	95.0	99	96.0
2	82	87.000000	86.0	82	82.0
3	91	92.000000	99.0	91	92.0
4	93	85.138889	92.0	97	98.0



4-2-2 選取列資料

選取列資料

列索引
(由 0 開始)

單列 $df[1:2]$

選取「列索引 1」資料時，不可只寫 $df[1]$ ，會發生錯誤！

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88	82
1	1080002	王倫樺	女	1080001@sun.tc.edu.tw	92	100	95
2	1080003	何宜敏	女	1080001@sun.tc.edu.tw	82	87	86
3	1080004	何志陞	男	1080001@sun.tc.edu.tw	91	92	99
4	1080005	吳一歌	女	1080001@sun.tc.edu.tw	93	85	92
5	1080006	宋緯挺	男	1080001@sun.tc.edu.tw	84	81	83

多列 $df[3:6]$

選取「列索引 3~5」資料，要寫 3:6 呢！





選取單列 / 多列

EX4-2.2.ipynb

01

首先讀取「學生成績檔 -4-2.2.csv」並轉成資料框型別，選取資料框的第 0~3 列，印出所有選取到的資料。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.2.csv')  
5 df[0:4] ←  
    ↓  
    列出 0~3 列
```

Drive already mounted at /content/MyGoogleDrive; to attempt to forcibly remount, call drive.mount("

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
0	1080001	丁軒軒	1 1080001@sun.tc.edu.tw	86	88.0	82.0	87	92.0
1	1080002	王倫樺	0 1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
2	1080003	何宜敏	0 1080003@sun.tc.edu.tw	82	87.0	86.0	82	82.0
3	1080004	何志陞	1 1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0





選取單列 / 多列

EX4-2.2.ipynb

02

選取第 5 列，印出選取到的資料。



1 df[5:6]



不可只寫 `df[5]` 喔！會出錯！

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
5	1080006	宋緯挺	1	1080006@sun.tc.edu.tw	84	81.0	83.0	90	88.0



4-2-3 選取指定的資料格

選取資料格

		：全部的行				5~7列索引			先列後行	
		df.iloc[3, :]				df.iloc[5:8, :]			df.iloc[1, 4]	
		單列				多列			單格	
行索引	列索引	0	1	2	3	4	5	6	7	8
		ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考		
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88	82			
1	1080002	王倫樺	女	1080001@sun.tc.edu.tw	92	100	95			
2	1080003	何宜敏	女	1080001@sun.tc.edu.tw	82	87	86			
3	1080004	何志陞	男	1080001@sun.tc.edu.tw	91	92	99			
4	1080005	吳一歌	女	1080001@sun.tc.edu.tw	93	85	92			
5	1080006	宋緯挺	男	1080001@sun.tc.edu.tw	84	81	83			
6	1080007	李宇綸	女	1080001@sun.tc.edu.tw	82	76	89			
7	1080008	李憲勝	男	1080001@sun.tc.edu.tw	90	98	96			
8	1080009	杜以潔	女	1080001@sun.tc.edu.tw	53	56	57			
9	1080010	沈程隆	男	1080001@sun.tc.edu.tw	81	77	84			
10	1080011									

多行 df.iloc[:, 1:4] 1~3行索引

單行 df.iloc[:, 6] 全部的列



4-2-3 選取指定的資料格

選取資料格

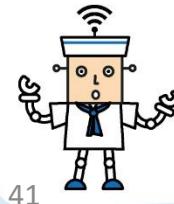
列索引	0	1	2	3	4	5	6	7
行索引	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考
0	1080001	丁軒軒	男	1080001@sun.tc.edu.tw	86	88	82	87
1	1080002	王倫樺	女	1080001@sun.tc.edu.tw	92	100	95	99
2	1080003	何宜敏	女	1080001@sun.tc.edu.tw	82	87	86	82
3	1080004	何志陞	男	1080001@sun.tc.edu.tw	91	92	99	91
4	1080005	吳一歌	女	1080001@sun.tc.edu.tw	93	85	92	97
5	1080006	宋緯挺	男	1080001@sun.tc.edu.tw	84	81	83	90
6	1080007	李宇綸	女	1080001@sun.tc.edu.tw	82	76	89	87
7	1080008	李憲勝	男	1080001@sun.tc.edu.tw	90	98	96	93
8	1080009	杜以潔	女	1080001@sun.tc.edu.tw	53	56	57	54
9	1080010	沈程隆	男	1080001@sun.tc.edu.tw	81	77	84	86
10	1080011							

範圍 df. iloc[3:7, 0:4]
3~6列索引、0~3行索引

多格 df. iloc[[4, 6], [5, 7]]
先列後行

也可寫成
 $x = [1, 4, 6]$
df. iloc[1, x]

也可寫成
 $y = [3, 5, 7]$
df. iloc[y, 4]



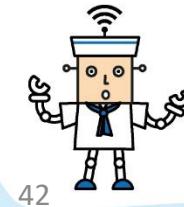
4-2-3 選取指定的資料格

設定指定
資料格的值
`df.iloc`

`df.iloc[列索引, 行索引] = 值`

例▶ `df.iloc[1, 4] = 90` 將列索引1、行索引4的資料格設定為90

例▶ `df.iloc[1, 5:8] = 0` 將列索引1、行索引5、6、7的三個資料格設定為0





實作

使用 iloc 函式選取和修改指定資料格的值

EX4-2.3.ipynb

01

首先讀取「學生成績檔 -4-2.3.csv」並轉成資料框型別，選取並印出第 1 列（列索引為 1，即王倫樺）的各項資料。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.3.csv')  
5 df.head()  
6 df.iloc[1,:]
```

```
↳ Drive already mounted at /content/MyGoogleDrive; to attempt to  
ID 1080002  
name 王倫樺  
sex 0  
email 1080002@sun.tc.edu.tw  
第1次平時考 92  
第2次平時考 100  
第3次平時考 95  
第4次平時考 99  
第5次平時考 96  
Name: 1, dtype: object
```





實作

使用 iloc 函式選取和修改指定資料格的值 EX4-2.3.ipynb

02

選取並印出第 1 列到第 3 列 (列索引 1~3，即王倫樺～何志陞) 的各項資料。



1 df.iloc[1:4,:]

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0
2	1080003	何宜敏	0	1080003@sun.tc.edu.tw	82	87.0	86.0	82	82.0
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.0	99.0	91	92.0



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

使用 iloc 函式選取和修改指定資料格的值

EX4-2.3.ipynb

03

選取並印出第 1 列第 4 行的資料格。(即王倫樺的第 1 次平時考成績)

```
1 df.iloc[1,4]
```

```
2 92
```

04

選取並印出第 1 列第 1、4、6、8 行共 4 個資料格。(即王倫樺的 name、第 1 次平時考成績、第 3 次平時考成績、第 5 次平時考成績)

```
1 df.iloc[1,[1,4,6,8]]
```

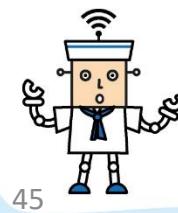
```
2
  name      王倫樺
  第1次平時考    92
  第3次平時考    95
  第5次平時考    96
Name: 1, dtype: object
```



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

使用 iloc 函式選取和修改指定資料格的值

EX4-2.3.ipynb

05

將王倫樺的第 1 次平時考成績由「92」修改成「90」。



```
1 df.iloc[1,4] = 90  
2 df.iloc[1,:]
```

ID	1080002
name	王倫樺
sex	0
email	1080002@sun.tc.edu.tw
第1次平時考	90
第2次平時考	100
第3次平時考	95
第4次平時考	99
第5次平時考	96
Name:	1, dtype: object

「92」修改成「90」



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

使用 iloc 函式選取和修改指定資料格的值

EX4-2.3.ipynb

06

將王倫樺的第 2 次平時考 ~ 第 5 次平時考成績都修改成「0」。

```
1 df.iloc[1,5:9] = 0  
2 df.iloc[1,:]
```

```
ID           1080002  
name        王倫樺  
sex            0  
email      1080002@sun.tc.edu.tw  
第1次平時考       90  
第2次平時考       0  
第3次平時考       0  
第4次平時考       0  
第5次平時考       0  
Name: 1, dtype: object
```



這些成績都修改成「0」



資料科學 × 機器學習

實戰探索

Practical Exploration



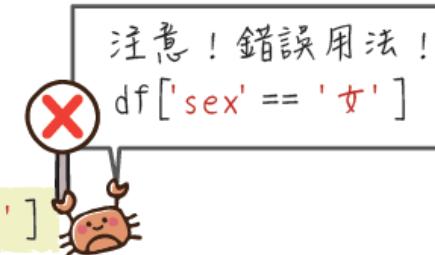
4-2-4 條件式選取

條件式
篩選資料格

資料框名稱[條件式]

例 篩選出行索引為「女」的資料

`df[df['sex'] == '女']` 或 `df[df.sex == '女']`



例 篩選出行索引「第1次平時考」90分以上的資料

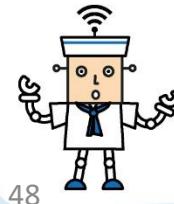
`df[df['第1次平時考'] >= 90]` 或 `df[df.第1次平時考 >= 90]`

例 篩選出「第1次平時考」90分以上且「性別」為女的資料

`df[(df['sex'] == '女') & (df['第1次平時考'] >= 90)]`

或

`df[(df.sex == '女') & (df.第1次平時考 >= 90)]`





實作

篩選符合條件的資料

EX4-2.4.ipynb

01

首先讀取「學生成績檔 -4-2.4.csv」並轉成資料框型別，篩選並印出所有第 1 次平時考成績大於等於 90 分的資料。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.4.csv')  
5 df[df['第1次平時考']>=90].head()
```

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.000000	99.0	91	92.0
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0
10	1080013	林保苓	0	1080013@sun.tc.edu.tw	92	94.000000	99.0	96	94.0
21	1080024	許雅均	0	1080024@sun.tc.edu.tw	92	89.000000	96.0	97	98.0

成績 >= 90



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

篩選符合條件的資料

EX4-2.4.ipynb

02

篩選到的資料除了可以整列印出之外，還能只印出部分行，例如：
第 1 次平時考成績小於 60 分的同學姓名。

```
▶ 1 df[df['第1次平時考'] < 60].name
```

```
↳ 7 杜以潔  
      22 許銘婷  
Name: name, dtype: object
```

03

篩選並印出每一次平時考成績都大於 95 分的同學姓名。

```
▶ 1 df[(df['第1次平時考']>95) & (df['第2次平時考']>95) & 接下行  
          (df['第3次平時考']>95) & (df['第4次平時考']>95) & 接下行  
          (df['第5次平時考']>95)].name
```

```
↳ 24 陳生貞  
Name: name, dtype: object
```

由於本書版面無法呈現完整的敘述，請在同一列書寫程式敘述並且不要斷列，以免產生錯誤。





實作

篩選符合條件的資料

EX4-2.4.ipynb

04

篩選並印出陳生貞同學的各項資料。



```
1 df[df['name'] == '陳生貞']
```

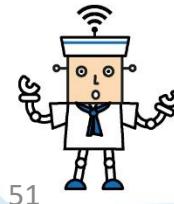
	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
24	1080027	陳生貞	1	1080027@sun.tc.edu.tw	97	100.0	100.0	100	100.0



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

篩選符合條件的資料

EX4-2.4.ipynb

05

篩選並印出第 1 次平時考 90 以上，或第 3 次平時考 95 分以上的同學。



```
1 df[(df['第1次平時考']>=90) | (df['第3次平時考']>=95)]
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.000000	95.000000	99
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.000000	99.000000	91
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.000000	97
10	1080013	林保苓	0	1080013@sun.tc.edu.tw	92	94.000000	99.000000	96
21	1080024	許雅均	0	1080024@sun.tc.edu.tw	92	89.000000	96.000000	97
24	1080027	陳生貞	1	1080027@sun.tc.edu.tw	97	100.000000	100.000000	100
25	1080028	陳宇愷	1	1080028@sun.tc.edu.tw	94	92.000000	93.000000	94
26	1080029	陳杰育	1	1080029@sun.tc.edu.tw	90	98.000000	96.000000	93
27	1080030	陳一潔	0	1080030@sun.tc.edu.tw	94	90.000000	100.000000	97



資料科學 × 機器學習

實戰探索

Practical Exploration



4-2-5 排序

sort_index()
函式

以內定列索引為key(鍵值)進行排序

資料框2 = 資料框1.sort_index(ascending=True/False)



1. *ascending=True* 或省略：表示由小到大排列。
2. *ascending=False*：表示由大到小排列。

sort_values()
函式

以指定行索引為key(鍵值)進行排序

資料框2 = 資料框1.sort_values('行索引', ascending=True/False)





實作

排序

EX4-2.5.ipynb

01

首先讀取「學生成績檔 -4-2.5.csv」並轉成資料框型別，將 df 資料框依內定列索引由大到小排序，完成後印出排序的結果。

```
1 from google.colab import drive
2 drive.mount('/content/MyGoogleDrive')
3 import pandas as pd
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.5.csv')
5 df = df.sort_index(ascending=False) ←
6 df.head()
```

依內定列索引由大到小排序

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考
37	1080040	謝穎安	1 1080040@sun.tc.edu.tw	78	82.0	81.000000	79
36	1080039	蔡凌嘉	1 1080039@sun.tc.edu.tw	94	100.0	100.000000	94
35	1080038	劉二婕	0 1080038@sun.tc.edu.tw	94	100.0	86.135135	92
34	1080037	劉隆霖	1 1080037@sun.tc.edu.tw	91	87.0	89.000000	93
33	1080036	廖安軒	0 1080036@sun.tc.edu.tw	91	92.0	97.000000	91

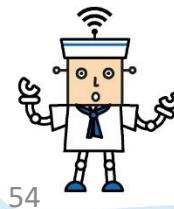
列索引由大到小排序



資料科學 × 機器學習

實戰探索

Practical Exploration





實作

排序

EX4-2.5.ipynb

02

以第 1 次平時考成績由小到大排序，並將排序結果存到另一個資料框 df1，完成後印出資料框 df1。

```
1 df1 = df.sort_values('第1次平時考')
2 df1.head()
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考
22	1080025	許銘婷	0	1080025@sun.tc.edu.tw	52	50.0	49.0	74
7	1080009	杜以潔	0	1080009@sun.tc.edu.tw	53	56.0	57.0	54
11	1080014	林宏銘	1	1080014@sun.tc.edu.tw	60	56.0	58.0	85
14	1080017	胡祥傑	1	1080017@sun.tc.edu.tw	77	75.0	78.0	80
37	1080040	謝穎安	1	1080040@sun.tc.edu.tw	78	82.0	81.0	79

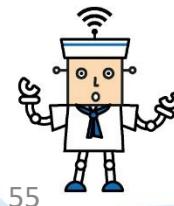
第 1 次平時考由小到大排序



資料科學 X 機器學習

實戰探索

Practical Exploration





加廣 知識

如果想要使用多行排序，例如：先以第 3 次平時考成績由大到小排列，若成績相同則再以第 1 次平時考由大到小排列，這樣當成鍵值的兩個行索引就要設定成串列。

```
1 df = df.sort_values(['第3次平時考', '第1次平時考'], ascending=False)  
2 df.head()
```

先以第 3 次平時考由大到小排序

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
24	1080027	陳生貞	1	1080027@sun.tc.edu.tw	97	100.0	100.0	100 100.000000
36	1080039	蔡凌嘉	1	1080039@sun.tc.edu.tw	94	100.0	100.0	94 95.000000
27	1080030	陳一潔	0	1080030@sun.tc.edu.tw	94	90.0	100.0	97 88.702703
10	1080013	林保苓	0	1080013@sun.tc.edu.tw	92	94.0	99.0	96 94.000000
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.0	99.0	91 92.000000

分數相同再以第 1 次平時考由大到小排序

4-2-6 計算最大/最小/平均數/總和

資料
統計

資料框名稱['行索引'].函式名稱(axis='columns')



常用函式

`max` (最大值)

`min` (最小值)

`sum` (總和)

`mean` (平均值)

1. 分別計算各列的數據

2. 若省略不寫或改為
`axis='index'`，則是
分別計算各行的數據





實作

max, min, mean, sum 的計算

EX4-2.6.ipynb

01

首先讀取「學生成績檔 -4-2.6.csv」並轉成資料框型別，計算並印出全班同學第 1 次平時考的最高分 (max) 和最低分 (min) 。

```
1 from google.colab import drive  
2 drive.mount('/content/MyGoogleDrive')  
3 import pandas as pd  
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.6.csv')  
5 df['第1次平時考'].max() ← 最高分
```

```
↳ Drive already mounted at /content/MyGoogleDrive; to attempt to for  
97
```

```
[ ] 1 df['第1次平時考'].min() ← 最低分
```

52





實作

max, min, mean, sum 的計算

EX4-2.6.ipynb

02

計算並印出全班同學第 1 次平時考～第 5 次平時考的各次成績總和 (sum) 及平均成績 (mean)。

```
1 c = ['第1次平時考', '第2次平時考', '第3次平時考', '第4次平時考', '第5次平時考']
2 df[c].sum()
```

```
↳ 第1次平時考      3212.000000
    第2次平時考      3235.277778
    第3次平時考      3273.135135
    第4次平時考      3335.000000
    第5次平時考      3370.702703
    dtype: float64
```

```
[ ] 1 df[c].mean()
```

```
第1次平時考      84.526316
第2次平時考      85.138889
第3次平時考      86.135135
第4次平時考      87.763158
第5次平時考      88.702703
dtype: float64
```



資料科學 × 機器學習

實戰探索

Practical Exploration





max, min, mean, sum 的計算

EX4-2.6.ipynb

03

計算並印出每位同學自己 5 次平時考的平均。



```
1 df[c].mean(axis='columns')
```

分別計算各列的平均

```
↳ 0      87.000000
    1      96.400000
    2      83.800000
    3      93.000000
    4      93.027778
    5      85.200000
    6      83.800000
    7      54.600000
    8      82.200000
    9      88.000000
   10     95.000000
```



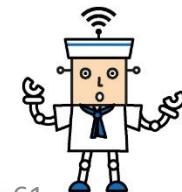
4-2-7 新增行資料

新增
行資料

以**行索引**增加一行到資料框的最右端

資料框名稱['**行索引**'] = 序列/串列

以**序列或串列**來
設定該行的資料





實作

新增行資料

EX4-2.7.ipynb

01

首先讀取「學生成績檔 -4-2.7.csv」並轉成資料框型別，計算每位同學 5 次平時考的總分，並將結果增添一行到 df 資料框中。

```
1 from google.colab import drive
2 drive.mount('/content/MyGoogleDrive')
3 import pandas as pd
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.7.csv')
5 c = ['第1次平時考','第2次平時考','第3次平時考','第4次平時考','第5次平時考']
6 df['總分'] = df[c].sum(axis='columns') ←
7 df.head()
```

計算總分並增添一行「總分」

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	總分	
0	1080001	丁軒軒	1	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0	435.000000
1	1080002	王倫樟	0	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0	482.000000
2	1080003	何宜敏	0	1080003@sun.tc.edu.tw	82	87.000000	86.0	82	82.0	419.000000
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.000000	99.0	91	92.0	465.000000
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0	465.138889

在資料框中新增一行「總分」





新增行資料

EX4-2.7.ipynb

02

計算每位同學 5 次平時考的平均，並將結果增添一行到 df 資料框中。

```
1 df['平均'] = df[c].mean(axis='columns')
2 df.head()
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	總分	平均
0	1080001	丁軒軒	1	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0	435.000000	87.000000
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0	482.000000	96.400000
2	1080003	何宜敏	0	1080003@sun.tc.edu.tw	82	87.000000	86.0	82	82.0	419.000000	83.800000
3	1080004	何志墾	1	1080004@sun.tc.edu.tw	91	92.000000	99.0	91	92.0	465.000000	93.000000
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0	465.138889	93.027778



在資料框中新增一行「平均」





實作

新增行資料

EX4-2.7.ipynb

03

以平均成績由大到小排序之後，再利用串列依序填入名次，並將結果增添一行到 df 資料框中。

```
1 df = df.sort_values('平均', ascending=False)
2 df['排名'] = list(range(1, len(df)+1))
3 df.head()
```

這段敘述會產生一個 [1, 2, 3..., df
資料框總列數] 連續的整數串列

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	總分	平均	排名
24	1080027	陳生貞	1	1080027@sun.tc.edu.tw	97	100.0	100.0	100	100.0	497.0	99.4	1
36	1080039	蔡凌嘉	1	1080039@sun.tc.edu.tw	94	100.0	100.0	94	95.0	483.0	96.6	2
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0	482.0	96.4	3
10	1080013	林保苓	0	1080013@sun.tc.edu.tw	92	94.0	99.0	96	94.0	475.0	95.0	4
25	1080028	陳宇愷	1	1080028@sun.tc.edu.tw	94	92.0	93.0	94	100.0	473.0	94.6	5

依排序結果後加入名次





實作

新增行資料

EX4-2.7.ipynb

04

利用條件篩選印出前 3 名同學的資料。



```
1 df[df.排名 <= 3]
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	總分	平均	排名
24	1080027	陳生貞	1	1080027@sun.tc.edu.tw	97	100.0	100.0	100	100.0	497.0	99.4	1
36	1080039	蔡凌嘉	1	1080039@sun.tc.edu.tw	94	100.0	100.0	94	95.0	483.0	96.6	2
1	1080002	王倫樺	0	1080002@sun.tc.edu.tw	92	100.0	95.0	99	96.0	482.0	96.4	3

列出符合條件的資料



資料科學 × 機器學習

實戰探索

Practical Exploration



4-2-8 新增列資料

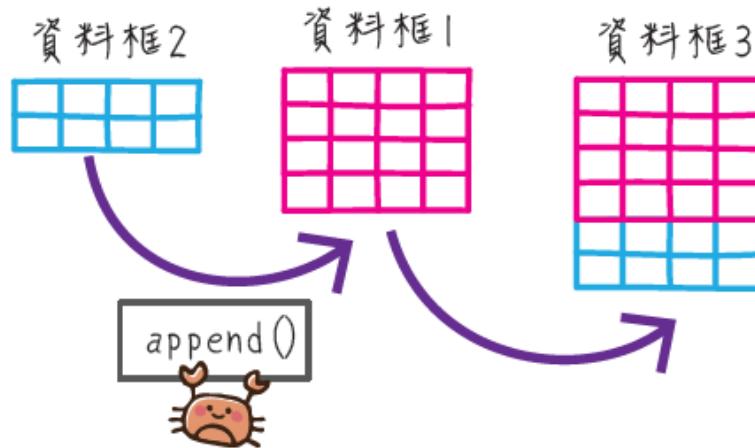
append()
函式

新增列資料

ignore_index=True :

將內定列索引重新由0開始編號

資料框3 = 資料框1.append(資料框2,ignore_index=True/False)



1. 將 **資料框2** 資料增加到 **資料框1** 的尾端，再把增加後的 **資料框1** 結果放到 **資料框3**。
2. **資料框3** 的名稱若和 **資料框1** 相同，會將新增後的結果直接更新到 **資料框1**。





實作

新增列資料

EX4-2.8.ipynb

01

首先讀取「學生成績檔 -4-2-8-ANS.csv」並轉成資料框型別，呼叫 `tail()` 函式印出最後 5 列的資料，結果顯示最後 1 列是「謝穎安」。

```
1 from google.colab import drive
2 drive.mount('/content/MyGoogleDrive')
3 import pandas as pd
4 df=pd.read_csv(i_filepath + '學生成績檔-4-2.8.csv')
5 df.tail()
```

ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	
33	1080036	廖安軒	0	1080036@sun.tc.edu.tw	91	92.0	97.000000	91	94.0
34	1080037	劉隆霖	1	1080037@sun.tc.edu.tw	91	87.0	89.000000	93	95.0
35	1080038	劉二婕	0	1080038@sun.tc.edu.tw	94	100.0	86.135135	92	95.0
36	1080039	蔡凌嘉	1	1080039@sun.tc.edu.tw	94	100.0	100.000000	94	95.0
37	1080040	謝穎安	1	1080040@sun.tc.edu.tw	78	82.0	81.000000	79	80.0

最後 1 列是「謝穎安」





實作

新增列資料

EX4-2.8.ipynb

2

將新同學「張三」的各項資料轉成資料框 df1，並呼叫 `append()` 函式新增到資料框 df 的尾端，再以 `tail()` 函式印出最後 5 列的資料查看結果。

s 是串列(一維)，[s] 則會形成二維串列

```
1 s = ['1080041', '張三', '1', '1080041@sun.tc.edu.tw',
       90, 85, 90, 95, 95]
2 df1 = pd.DataFrame(data=[s], columns=df.columns)
3 df = df.append(df1, ignore_index=True)
4 df.tail()
```

添加資料

新資料 df1 的行
索引和 df 一樣

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考
34	1080037	劉隆霖	1	1080037@sun.tc.edu.tw	91	87.0	89.000000	93	95.0
35	1080038	劉二婕	0	1080038@sun.tc.edu.tw	94	100.0	86.135135	92	95.0
36	1080039	蔡凌嘉	1	1080039@sun.tc.edu.tw	94	100.0	100.000000	94	95.0
37	1080040	謝穎安	1	1080040@sun.tc.edu.tw	78	82.0	81.000000	79	80.0
38	1080041	張三	1	1080041@sun.tc.edu.tw	90	85.0	90.000000	95	95.0

將新資料加至最後 1 列





實作

新增列資料

EX4-2.8.ipynb

03

重新計算並新增全班的總分、平均及排名，再依「ID」由小到大排列。

「ID」需先轉換資料型別後才可以正確排序

```
1 c = ['第1次平時考', '第2次平時考', '第3次平時考', '第4次平時考', '第5次平時考']
2 df['總分'] = df[c].sum(axis='columns')
3 df['平均'] = df[c].mean(axis='columns')
4 df = df.sort_values('平均', ascending=False)
5 df['排名'] = list(range(1, len(df)+1))
6 df['ID'] = df.ID.astype(str) ←
7 df = df.sort_values(['ID'], ascending=True)
8 df.head()
```

	ID	name	sex	email	第1次平時考	第2次平時考	第3次平時考	第4次平時考	第5次平時考	總分	平均	排名
0	1080001	軒軒	1	1080001@sun.tc.edu.tw	86	88.000000	82.0	87	92.0	435.000000	87.000000	22
1	1080002	王倫禪	0	1080002@sun.tc.edu.tw	92	100.000000	95.0	99	96.0	482.000000	96.400000	3
2	1080003	何宣敏	0	1080003@sun.tc.edu.tw	82	87.000000	86.0	82	82.0	419.000000	83.800000	27
3	1080004	何志陞	1	1080004@sun.tc.edu.tw	91	92.000000	99.0	91	92.0	465.000000	93.000000	12
4	1080005	吳一歌	0	1080005@sun.tc.edu.tw	93	85.138889	92.0	97	98.0	465.138889	93.027778	11

依「ID」由小到大排列



資料科學 × 機器學習

實戰探索

Practical Exploration

