



程式設計概論 PROGRAMMING 101 —PANDAS進行資料科學分析

授課老師：邱淑怡



2 OUTLINE

- Pandas 資料結構
- DataFrame 建立
- DataFrame 的基本資訊
- DataFrame 讀取外部檔案或外部連結(URL)
- DataFrame 資料選取、篩選
- DataFrame 的合併
- DataFrame 資料分群

3 PANDAS

- Pandas (powerful Python data analysis toolkit) (https://pandas.pydata.org/pandas-docs/stable/getting_started/index.html)
 - pandas為第三方套件，需安裝
- Pandas模組的操作

4 PANDAS 資料結構

- Series (序列)
- DataFrame (資料框)
- Panel

Python提供的資料結構: list, dictionary, tuple, set
Python Numpy提供的資料結構: array
Python pandas提供的資料結構: dataframe, series

5 SERIES

- **Series** 是一個類似一維陣列的資料結構，它由兩個部分組成：
 - 第一部分是**label**或**index**
 - 第二部分是對應的數值

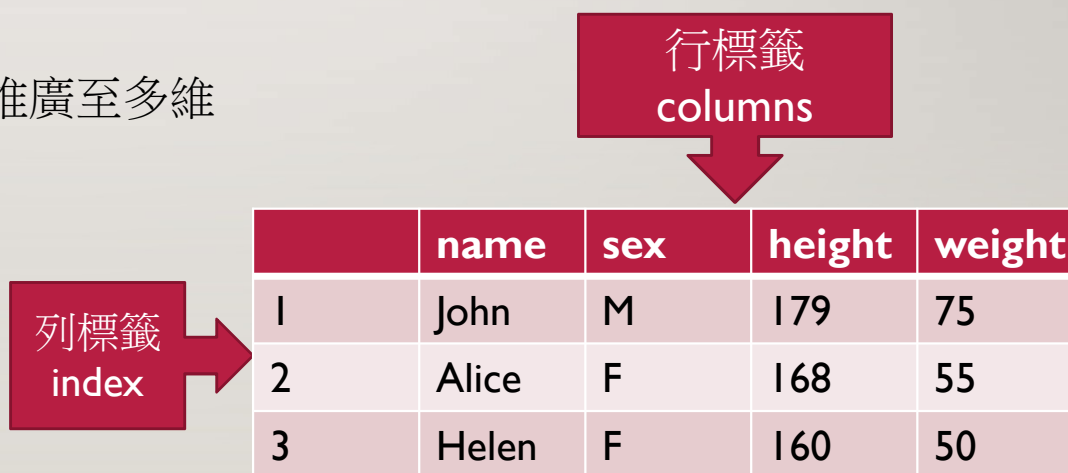
```
import pandas as pd
Series1=pd.Series([0.003731, -0.001838,-0.003087,-0.024112],
index=['中國石油','工商銀行','上汽集團','寶鋼股份'])
print(Series1)
```

6 SERIES缺點

- 只能有兩列，一列為索引列；另一列為數值列，若有更多數值列，則需要使用 `dataframe` 處存資料

7 DATAFRAME

- 可用來存放整數、字串、浮點數、Python物件等資料型別的二維陣列
- 可以將它想像成試算表
- DataFrame設計初衷就是將序列從一維推廣至多維
- DataFrame由三部分組成:
 - 第一部分是列 (index)
 - 第二部分是行標籤(columns):變數名稱
 - 第三部分是設定值



The diagram illustrates the structure of a DataFrame. A red box labeled '行標籤' (row labels) and 'columns' points down to the first column of the table. Another red box labeled '列標籤' (column labels) and 'index' points right to the first row of the table. The table itself has a header row with columns 'name', 'sex', 'height', and 'weight', and a first column with indices 1, 2, and 3.

	name	sex	height	weight
1	John	M	179	75
2	Alice	F	168	55
3	Helen	F	160	50

8 DATAFRAME

- `import pandas as pd`
- `Df=pd.DataFrame(data,=None, index=None, columns=None,其他選擇性參數)`, `data`用來指定**DataFrame**的資料, `index`用來指定資料的列標籤, `column`用來指定資料的行標籤(變數名稱)
- **DataFrame**的資料來源包括:
 - Python list, tuple, dict, Series, 一維的ndarray
 - 二維的ndarray, Series, 或其他**DataFrame**

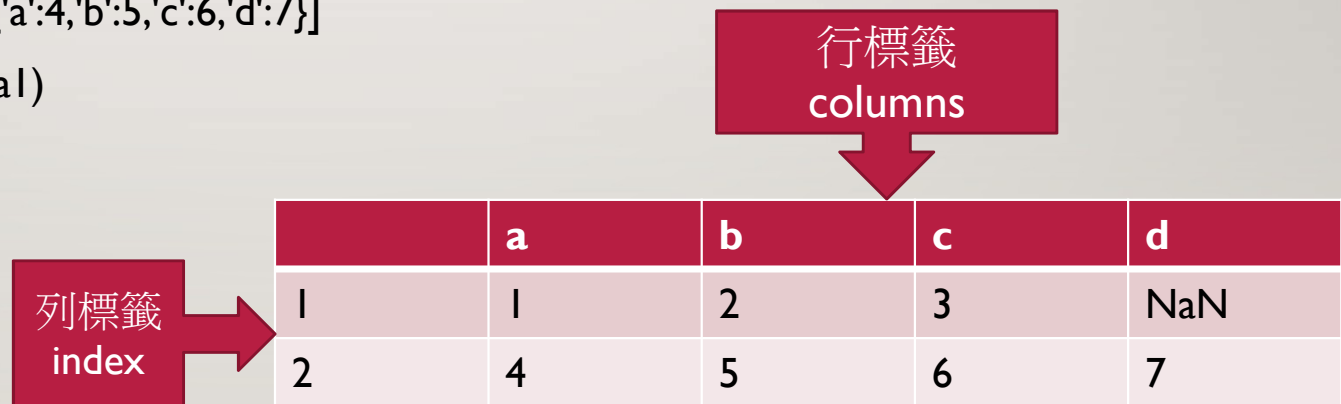
The diagram illustrates a DataFrame table. A red box labeled '行標籤 columns' with a downward arrow points to the column headers. Another red box labeled '列標籤 index' with a rightward arrow points to the row indices. The table itself has a header row with columns 'name', 'sex', 'height', and 'weight', and three data rows with indices 1, 2, and 3.

	name	sex	height	weight
1	John	M	179	75
2	Alice	F	168	55
3	Helen	F	160	50

9 DATAFRAME操作: 利用DICT所組成的串列建立一個DATAFRAME

- 透過dict所組成的串列建立一個DataFrame並指派給Df1，若沒有對應的值就給NaN

- Data1=[{'a':1,'b':2,'c':3}, {'a':4,'b':5,'c':6,'d':7}]
- Df1=pd.DataFrame(Data1)
- print(Df1)
- print(Df1.index)
- print(Df1.columns)
- print(Df1[['a','c']])
- print(Df1['a'][1])
- print(Df1[:1])



The diagram illustrates the DataFrame structure. A red box labeled "行標籤" (row labels) and "columns" points to the column headers. Another red box labeled "列標籤" (column labels) and "index" points to the row indices.

	a	b	c	d
1	1	2	3	NaN
2	4	5	6	7

10 DATAFRAME操作: 利用DICT所組成的串列建立一個DATAFRAME，並建立INDEX和COLUMNS

- `Data1=[{'a':1,'b':2,'c':3}, {'a':4,'b':5,'c':6,'d':7}]`
- `Df2=pd.DataFrame(Data1,index=['first','second'])`
- `print(Df2)`
- `Df3=pd.DataFrame(Data1,index=[1,2], columns=['d','b','a'])`
- `print(Df3)`

II DATAFRAME基本資訊

- Df2.info()
- Df2.describe()
- Df2.columns
- Df2.sort_values(by='a').head()



Column name

範例說明

13 建立DATAFRAME

```
import pandas as pd

per_df = pd.DataFrame() # 產生一個空的dataframe

col = ['class','name','Birthday','salary','height','weight']

data = [['class0', 'John', '1993-10-01', 36000, 177, 76], ['class0', 'Bob', '1992-10-02', 52000, 173, 68],
['class1', 'Helen', '1990-10-01', 43000, 167, 55], ['class2', 'Alice', '1983-10-03', 27000, 169, 56], ['class1',
'Justin', '1991-10-02', 22000, 180, 78], ['class0', 'David', '2001-10-03', 15000, 170, 69]]

per_df = pd.DataFrame(data, columns=col)

#per_df = pd.DataFrame(data, columns=col, index=['1','2','3','4','5','6'])

print(per_df)
```

14 DATAFRAME 資料選取

- 欲選出第幾筆資料的某些欄位有兩種方式:
 1. 用中括號篩選: 先篩選第幾筆到第幾筆資料，再選欄位
◆ `per_df.iloc[0:3,2:5]`
 2. 用`loc(location)`中括號裡面第一個放`index`的範圍，第二個放`column`的名稱
◆ `per_df.loc['2':'4','name':'weight']`

15 PANDAS 索引的運用

- 透過pandas提供的索引參照屬屬性取得series 或DataFrame的部分

索引參照屬性	說明
at	透過行/列標籤取得單一值，和loc屬性類似，若只取得單一值，可以使用at
iat	透過行/列索引編號取得單一值，和iloc屬性類似，若只取得單一值，可以使用iat
loc	透過行/列標籤取得一組列或行; loc[]的方法，用index的標籤來取出資料
iloc	透過行/列索引編號取得一組列或行

16 PANDAS 索引的運用(CONT.)

下面兩行意義相同

```
print(per_df.at['I', 'name'])
```

```
print(per_df.iat[0, 1])
```

17 PANDAS 索引的運用(CONT.)

```
print(per_df.loc['l'])  
print(per_df.loc['2':'4', 'name': 'weight'])  
print(per_df.iloc[1:3, 1:])  
# 上述兩行意義相同
```

18 PANDAS 索引: 可透過直接索引|參照取的DATAFRAME的部分資料

```
print(per_df['class'])  
print(per_df['name']['l']) # 注意順序: 先行標籤(columns)，再列標籤(index)  
print(per_df[['name','weight']])  
print(per_df[:2])  
print(per_df[per_df['salary']>50000])
```

19 DATAFRAME讀取外部檔案

- 外部資料匯入並直接產生資料框，包含txt, excel, csv等格式的檔案均可匯入至Python的資料結構

1. `read_table(“檔案路徑”,index_col, delimiter)`: index_col=0表示檔案第一列為索引列， delimiter為分隔符號
2. `read_excel(“檔案路徑”, sheetname, header, index_col)`: header=0表示用第一行作為指定列名稱行
3. `read_csv(“檔案路徑”, sep, delimiter, header, index_col)`

20 讀取檔案之範例程式: 可以直接讀取檔案或透過網址(URL)讀取

1. Excel file: `read_df = pd.read_excel("data2.xlsx", sheet_name='工作表2', header=0, index_col=0)`
2. csv file: `US2020_df = pd.read_csv("D:\\temp\\governors_county.csv", header=0)`
3. Txt file:
 - `txt_url = 'http://people.apache.org/~edwardyoon/kmeans.txt'`
 - `iris_df = pd.read_table(txt_url, sep = "\\t")`
4. Json file:
 - `json_url = "https://pkgstore.datahub.io/machine-learning/iris/iris_json/data/23a7b3de91da915b506f7ca23f6d1141/iris_json.json"`
 - `iris_json_df = pd.read_json(json_url)`

JSON (JavaScript Object Notation) 格式的資料是網站資料傳輸

21 DATAFRAME 依據"CLASS"分組(分群)

```
per_df.plot()  
frame_class = per_df.groupby('class')  
print(frame_class.groups)
```

22 DATAFRAME 依據"CLASS"分組(CONT.)

```
frame_class = per_df.groupby('class')
```

```
frame_class.get_group("class2")
```

```
frame_class.sum()
```

```
frame_class.median()
```

```
frame_class.boxplot()
```

```
frame_class.boxplot()
```

某一個class_name

23 合併DATAFRAME(資料框)

- 合併資料框的四個常用函數與方法，分別是 `concat`、`append`、`merge` 與 `join`
 - `concat` 與 `merge` 是簡單合併以及聯結的常規函數
 - `append` 與 `join` 則是建構於資料框的方法，目的是簡化合併的語法。
 - `pd.concat()`
 - `pd.append()`
 - `pd.merge()`
 - `pd.join()`

<https://medium.com/datainpoint/pandas105-83870f2a665b>

24 串接兩個DATAFRAME：水平、垂直合併

- 水平合併、垂直合併：pd.concat()
- 垂直合併：~~df1.append(df2, ignore_index=True)~~

水平
合併

df1.ch	df1.ca	df2.ch	df2.ca

垂直
合併

df1.ch	df1.ca
df2.ch	df2.ca

25 串接兩個DATAFRAME：CONCAT

```
import pandas as pd
df1 = pd.DataFrame()
df2 = pd.DataFrame()
df1["character"] = ["Rachel Green", "Monica Geller", "Phoebe Buffay"]
df1["cast"] = ["Jennifer Aniston", "Courteney Cox", "Lisa Kudrow"]
df2["character"] = ["Joey Tribbiani", "Chandler Bing", "Ross Geller"]
df2["cast"] = ["Matt LeBlanc", "Matthew Perry", "David Schwimmer"]
df3 = pd.concat([df1, df2], ignore_index=True)
# axis=0 (垂直合併) as default(預設值); axis=1 (水平合併)
# ignore_index=True means 重設列索引
```

df1.ch	df1.ca
df2.ch	df2.ca

垂直
合併

df1.ch	df1.ca	df2.ch	df2.ca

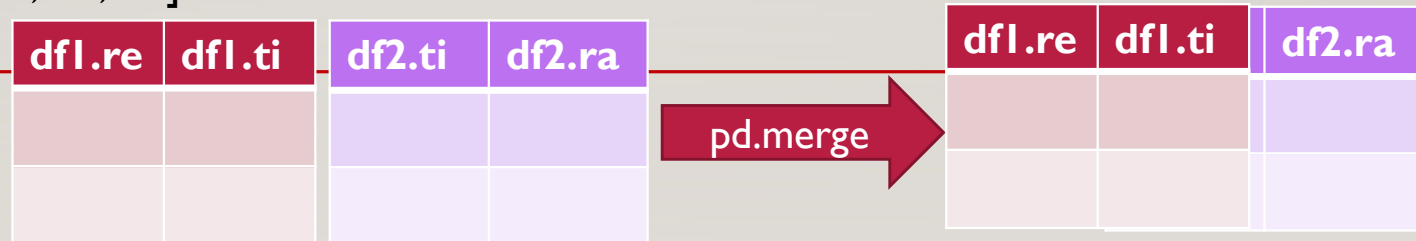
水平
合併

26 聯接兩個DATAFRAME

- 依據某個欄位名稱進行結合，該欄位只會出現一欄 → 水平合併

```
import pandas as pd
df1 = pd.DataFrame()
df2 = pd.DataFrame()
```

```
df1["title"] = ["The Avengers", "Avengers:Age of Ultron", "Avengers: Infinity War", "Avengers: Endgame"]
df1["release_year"] = [2012, 2015, 2018, 2019]
df2["title"] = ["Avengers: Infinity War", "Avengers: Endgame", "The Avengers", "Avengers:Age of Ultron"]
df2["rating"] = [8.5, 8.6, 8.5, 7.3]
pd.merge(df1, df2)
```



27

各組討論 練習匯入檔案於**DATAFRAME** 並做前處理

擷取需要的資料、多個**dataframe**的整合、分組加總....

28 PANDAS 索引: 可透過直接索引|參照取的DATAFRAME的部分資料

```
print(per_df['class'])  
print(per_df['name']['l']) # 注意順序: 先行標籤(columns)，再列標籤(index)  
print(per_df[['name','weight']])  
print(per_df[:2])
```

29 PANDAS : DATAFRAME 依據條件擷取部分資料

□ 單一條件

```
print(per_df[per_df['salary']>50000])
```

□ 多個條件

```
c1 = per_df['class']=='class0'
```

```
c2 = per_df['height'] > 170
```

```
temp_df=per_df[(c1 & c2)]
```

```
display(temp_df)
```

30 PANDAS讀取外部檔案

- 外部資料匯入並直接產生資料框，包含txt, excel, csv等格式的檔案均可匯入至Python的資料結構

1. `read_table(“檔案路徑”,index_col, delimiter)`: index_col=0表示檔案第一列為索引列， delimiter為分隔符號
2. `read_excel(“檔案路徑”, sheetname, header, index_col)`: header=0表示用第一行作為指定列名稱行
3. `read_csv(“檔案路徑”, sep, delimiter, header, index_col)`

31 PANDAS : 刪除部分DATAFRAME的部分資料

```
new_df1 = per_df.drop(["class"], axis=1)
```

```
display(new_df1)
```

```
new_df2= per_df.drop(['3','5']) # axis預設值為0
```

```
display(new_df2)
```

```
# 刪除空值(nan)
```

```
new_df3 = per_df.dropna()
```

32 讀取檔案之範例程式: 可以直接讀取檔案或透過網址(URL)讀取

1. Excel file: `read_df = pd.read_excel("data2.xlsx", sheet_name='工作表2', header=0, index_col=0)`
2. csv file: `US2020_df = pd.read_csv("D:\\temp\\governors_county.csv", header=0)`
3. Txt file:
 - `txt_url = 'http://people.apache.org/~edwardyoon/kmeans.txt'`
 - `iris_df = pd.read_table(txt_url, sep = "\\t")`
4. Json file:
 - `json_url = "https://pkgstore.datahub.io/machine-learning/iris/iris_json/data/23a7b3de91da915b506f7ca23f6d1141/iris_json.json"`
 - `iris_json_df = pd.read_json(json_url)`

JSON (JavaScript Object Notation) 格式的資料是網站資料傳輸

33 練習題

1. 匯入 `countries.csv`
2. 擷取 "Taiwan" 各年度的人口數量
3. 擷取 2000 年以後且人口數大於 250,000,000