程式設計概論
Programming 101
—pandas進行資料科學分析

授課老師：邱淑怡

DATE:5/12/2023

Outline

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

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

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

DataFrame

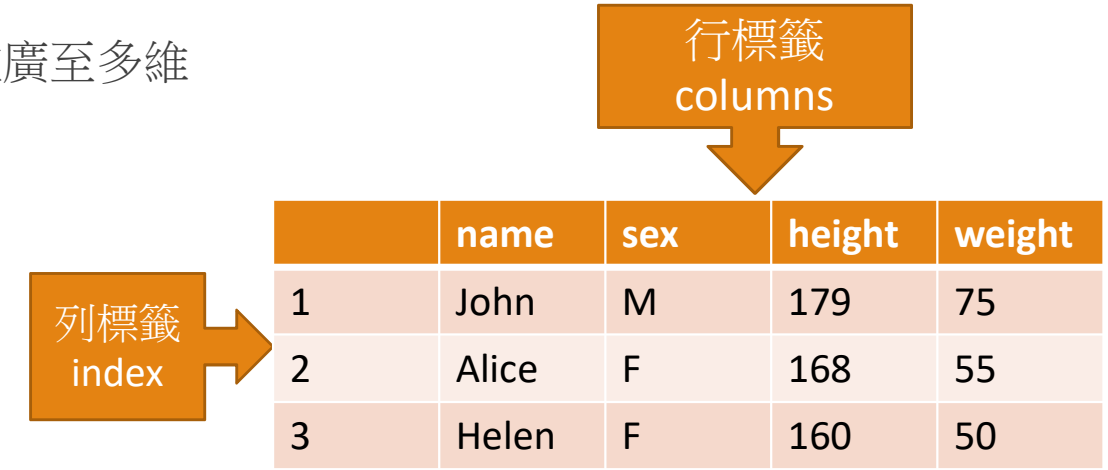
可用來存放整數、字串、浮點數、Python物件等資料型別的二維陣列

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



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

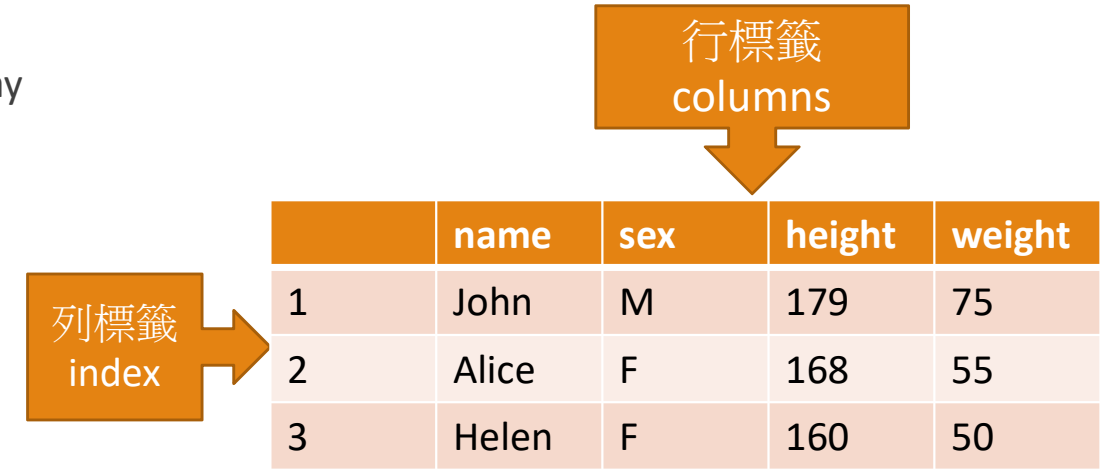
DataFrame

import pandas as pd

Df=pd.DataFrame(data, 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



dataframe基本資訊

display(Df2)

Df2.info()

Df2.describe()

Df2.columns

Df2.sort_values(by='a').head()

範例說明

建立dataframe

```
import pandas as pd

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

col = ['class','name','Birthdate','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, index=['1','2','3','4','5','6'])

print(per_df)
```

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

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'])

Pandas 索引的運用(cont.)

```
# 下面兩行意義相同
print(per_df.at['1', 'name'])
print(per_df.iat[0,1])
```

Pandas 索引的運用(cont.)

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

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

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)
```

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
new_df2=new_df2.drop(new_df2.index[0])  #remove first row
new_df2=new_df2.drop(new_df2.index[-1])  #remove last row
display(new_df2)
# 刪除空值(nan)
new_df3 = per_df.dropna()
```

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

練習題一

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

合併DataFrame(資料框)

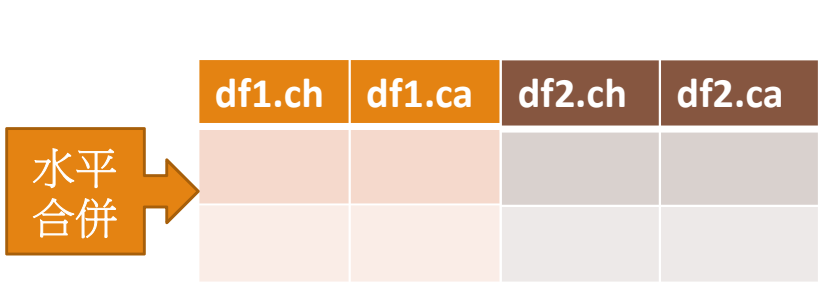
合併資料框的四個常用函數與方法，分別是 concat、append、merge 與 join

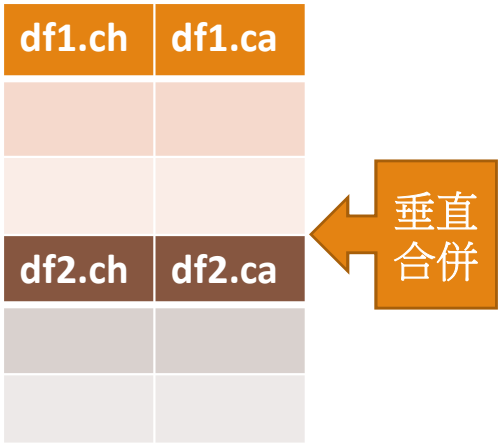
- concat 與 merge 是簡單合併以及聯結的常規函數
- append 與 join 則是建構於資料框的方法，目的是簡化合併的語法。
- pd.concat()
- pd.append()
- pd.merge()
- pd.join()

串接兩個dataframe：水平、垂直合併

水平合併、垂直合併：pd.concat()

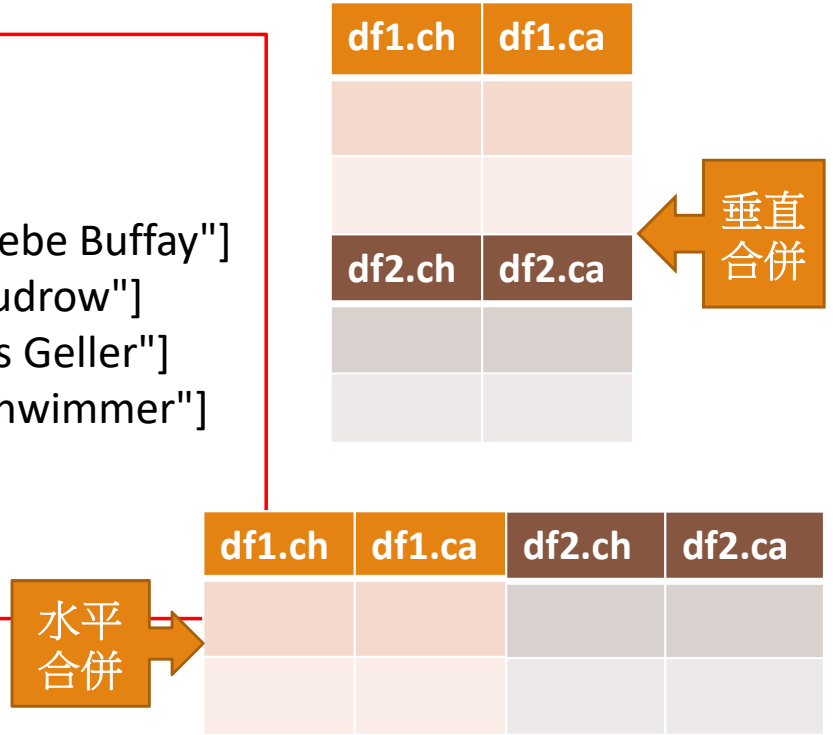
垂直合併: df.append(): df1.append(df2, ignore_index=True)





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
```

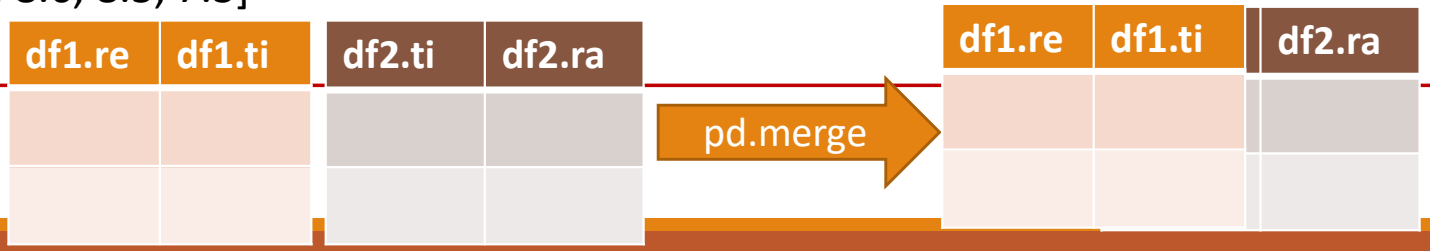


聯接兩個dataframe

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)
```



補充說明pandas 讀取外部檔案

- 外部資料匯入並直接產生資料框，包含excel, csv等格式的檔案均可匯入至Python的資料框(dataframe)。
- 若讀取檔案為excel檔案需額外安裝pip install openpyxl
  - openpyxl操作Excel的相依性套件

```
import pandas as pd
1. pd.read_csv(“檔案路徑”)
2. pd.read_excel(“檔案路徑”, sheet_name=“工作表名稱”)
```

補充資料

1. DataFrame 某個欄位(column_name: c) 轉成list

df1[‘c’].tolist()

2. 欲將index的資料納入欄位

df1.reset_index(inplace=True)

df1 = df1.rename(columns = {‘index’:’number’})

display(df1.head())