

11 Pandas

黃彬華編撰

- ❖ Pandas 概論
- ❖ Pandas 安裝
- ❖ Series
- ❖ DataFrame 建立
- ❖ DataFrame 資料處理
- ❖ 處理 CSV 文件
- ❖ 處理 JSON 文件
- ❖ 資料清洗
- ❖ 移除空值
- ❖ 填補空值
- ❖ 轉換格式
- ❖ 移除重複資料
- ❖ 移除錯誤資料

Pandas 概論

黃彬華編撰

- ❖ Pandas 是 Python 語言的一個擴充函式庫，用於資料處理與分析
 - Pandas 衍生自計量經濟學術用語「panel data」
 - Pandas 開放原始碼並由社群共同開發維護
- ❖ 初始版本於 2008 年發布，原作者為 Wes McKinney

Pandas 安裝

黃彬華編撰

- ❖ 安裝方式

- conda install pandas

- ❖ 其他安裝方式

- 參看 Getting started

- ❖ 程式引用

- import pandas as pd

Series

黃彬華編撰

- ❖ Series 類似 1 維陣列
- ❖ Series 內容可以來自 list, dictionary 或 NumPy 的 ndarray；「pd.Series()」是呼叫建構式而非一般函式
 - `pd.Series(["Python", "Java", "JS", "Swift", "C#"])`
 - `pd.Series({"name": "Python", "price": 500, "author": "Paul"})`
 - `pd.Series(np.array(["Python", "Java", "JS", "Swift", "C#"]))`
- ❖ 透過 index 取值
 - `pd.Series(seriesList, index=[0, 1])`
 - `pd.Series(seriesDic, index=['name', 'price'])`
- ❖ Series 也可以轉成 list, dictionary 或 ndarray
 - `series.tolist()`
 - `series.to_dict()`
 - `seriesDic.to_numpy()`

範例

黃彬華編撰

❖ SeriesDemo

DataFrame 建立

黃彬華編撰

- ❖ DataFrame 類似 2 維陣列
- ❖ DataFrame 內容可以來自 list、dictionary
 - 內容來自 2 維 list：1 維 list 長度可以不同，缺值則為 NaN
 - 內容來自 dict-list：list 長度要相同，否則產生 ValueError
 - 內容來自 list-dict：dict 長度可以不同，缺值則為 NaN
 - 內容來自 ndarray：1 維長度要相同，否則產生 ValueError
- ❖ DataFrame 可以轉成 ndarray、dictionary 或 list

The diagram illustrates a DataFrame table with the following structure:

	Column Label/ Header	0	1	2	3	4
Index Label		Name	Age	Marks	Grade	Hobby
0	S1	Joe	20	85.10	A	Swimming
1	S2	Nat	21	77.80	B	Reading
2	S3	Harry	19	91.54	A	Music
3	S4	Sam	20	88.78	A	Painting
4	S5	Monica	22	60.55	B	Dancing

Annotations in the diagram:

- Column Index:** Points to the header row (0-4).
- Row Index:** Points to the index column (0-4).
- Column:** Points to the 'Marks' column.
- Row:** Points to the 'S4' row.
- Element/ Value/ Entry:** Points to the value '88.78' at the intersection of row 'S4' and column 'Marks'.

範例

黃彬華編撰

❖ DFCreateDemo

DataFrame 資料處理

黃彬華編撰

- ❖ 可以對 DataFrame 做增刪改查 (CRUD) 操作

範例

黃彬華編撰

❖ DFManipulateDemo

處理 CSV 文件

黃彬華編撰

- ❖ CSV (Comma-Separated Values) 文件

- 屬於純文字，大多以逗號將資料值區隔

- ❖ 讀取 CSV 文件

- 預設分隔符號為 ",", 可使用正規表示式設定分隔符號
- 建議設定 engine="python" ; 因為預設為 engine="c" , 沒有支援正規表示式
 - ✦ `df = pd.read_csv("data.csv", delimiter="\s*,\s*", engine="python")`

- ❖ 轉存 CSV 文件

- header=False 不會有欄位名稱 , index=False 不會有 index
 - ✦ `df.to_csv("data.csv", header=False, index=False)`

範例

黃彬華編撰

❖ CsvDemo

處理 JSON 文件

黃彬華編撰

- ❖ JSON (JavaScript Object Notation) 文件

- 屬於純文字

- ❖ 讀取 JSON 文件

- ✦ `df = pd.read_json("data.json")`

- ❖ 轉存 JSON 文件

- 如果不希望產生的結果帶有 index，可以加上「orient="records"」

- ✦ `df.to_json("data.json", orient="records")`

範例

黃彬華編撰

❖ JsonDemo

❖ 常見的資料清洗方式

- 移除空值
- 填補空值
- 轉換格式
- 移除重複資料
- 移除錯誤資料

移除空值

黃彬華編撰

- ❖ 原始資料如果沒有值，讀進來會轉成 NaN
 - 例如「C#,,2020-3-9」讀進來會轉成「C# NaN NaN 2020-3-9」
- ❖ 檢查哪些資料被視為空值，被視為空值為 True
 - `df.isna()`
- ❖ 刪除帶有空值的該列資料
 - `df.dropna()`
- ❖ 可以自訂哪些值視為空值
 - 例如：`naCustom = ["na", "--"]`

範例

黃彬華編撰

❖ DropNaDemo

填補空值

黃彬華編撰

- ❖ 以指定值來填補指定欄位內的空值
 - `df["price"].fillna(0)`

範例

黃彬華編撰

❖ FillNaDemo

轉換格式

黃彬華編撰

- ❖ 將指定欄位內容轉換成日期格式
 - `df["date"] = pd.to_datetime(df["date"], format="ISO8601")`

範例

黃彬華編撰

❖ ToDatetimeDemo

移除重複資料

黃彬華編撰

- ❖ 檢查是否有重複資料
 - `df.duplicated()`
- ❖ 移除重複資料 (所有欄位值相同才會被視為重複資料)
 - `df.drop_duplicates()`

範例

黃彬華編撰

❖ DropDuplicatesDemo

移除錯誤資料

黃彬華編撰

- ❖ 自訂錯誤條件當作移除依據
 - `df.drop(df[df['price'] < 0].index)`

範例

黃彬華編撰

❖ DropConditionDemo