用 pandas 撰寫樞紐分析表

簡報閱讀 範例與作業 問題討論 學習心得(完成)

重要知識點

- 在資料中的索引轉欄位、欄位轉索引
- 欄位名稱轉為欄位值
- 重新組織資料

在前天學到畫圖，x 軸為索引(index)，y 軸為欄位(column)，此時 x 軸與 y 軸被索引以及欄位所限制，想畫出各種不同的圖做分析，就必須用到**欄位轉索引**或是**索引轉欄位**。

欄位轉索引

以下表資料為例，欄位為 subject、type 組成，索引為 year、visit 組成

- 欄位轉索引：將一欄位(column)轉成一索引(index)，使用.stack()即可，可以將 type 這個欄位轉成了索引，所以索引變成了 year、visit、type。
- 注意 .stack() 會由最外層的欄位開始轉換，原欄位為 subject、type，會先由 type 轉換過去索引，如果在做一次才會把 subject 也轉換過去索引，如左表。

year	visit						
2013	1	1.2	-1.7	0.3	0.4	0.8	0.2
	2	-1.3	-0.2	-1.9	-0.3	-0.7	-1.4
2014	1	0.7	-0.9	1.5	-0.5	-0.6	-0.1
	2	0.6	-0.6	-0.2	0.2	0.3	-1.9

```
df.stack()
```

		subject	Bob	Guido	Sue
year	visit	type			
2013	1	HR	1.2	0.3	0.8
		Temp	-1.7	0.4	0.2
	2	HR	-1.3	-1.9	-0.7
		Temp	-0.2	-0.3	-1.4
2014	1	HR	0.7	1.5	-0.6
		Temp	-0.9	-0.5	-0.1
	2	HR	0.6	-0.2	0.3
		Temp	-0.6	0.2	-1.9

```
2013  1      HR    Bob     -0.4
                   Guido   -1.3
                   Sue     -1.6
             Temp  Bob      0.0
                   Guido   -0.7
                   Sue     -1.4
      2      HR    Bob     -0.1
                   Guido    1.1
                   Sue      1.5
             Temp  Bob      1.9
                   Guido    2.3
                   Sue     -1.0
2014  1      HR    Bob     -3.3
                   Guido    0.4
                   Sue     -0.1
             Temp  Bob      0.7
                   Guido   -1.0
                   Sue     -1.0
      2      HR    Bob     -0.7
                   Guido    0.4
                   Sue     -0.0
             Temp  Bob      1.3
                   Guido   -0.4
                   Sue      1.0
dtype: float64
```

索引轉欄位

以下表資料為例，欄位為 subject、type 組成，索引為 year、visit 組成

- 索引轉欄位：將一索引(index)轉成一欄位(column) ，使用.unstack()即可，可以將 visit 這個索引轉成了欄位，所以欄位變成了 subject、type 、visit。
- 注意與 .stack() 相同會由最外層開始轉換，原索引為 year、visit，會先由 visit 轉換過去欄位

year	visit						
2013	1	1.2	-1.7	0.3	0.4	0.8	0.2
	2	-1.3	-0.2	-1.9	-0.3	-0.7	-1.4
2014	1	0.7	-0.9	1.5	-0.5	-0.6	-0.1
	2	0.6	-0.6	-0.2	0.2	0.3	-1.9

```
df.unstack()
```

subject	Bob				Guido				Sue			
type	HR		Temp		HR		Temp		HR		Temp	
visit	1	2	1	2	1	2	1	2	1	2	1	2
year												
2013	-0.4	-0.1	0.0	1.9	-1.3	1.1	-0.7	2.3	-1.6	1.5	-1.4	-1.0
2014	-3.3	-0.7	0.7	1.3	0.4	0.4	-1.0	-0.4	-0.1	-0.0	-1.0	1.0

欄位名稱轉為欄位值

數據分析的時候經常要規寬數據變成長數據，有點像你們用 excel 做透視跟逆透視的過程。
例如下表格（紅框），要將欄位轉成欄位值，也就是說將 Name、Course、Age 轉成欄位值，如下圖，使用.melt() 就可以做到。

參數

- id_vars：不需要被轉換的列名
- value_vars：需要轉換的列名，如果剩下的列全部都要轉換，就不用寫了。

0	John	Masters	27
1	Bob	Graduate	23
2	Shiela	Graduate	21

```
df.melt()
```

	variable	value
0	Name	John
1	Name	Bob
2	Name	Shiela
3	Course	Masters
4	Course	Graduate
5	Course	Graduate
6	Age	27
7	Age	23
8	Age	21

保留 Name 欄位其餘轉成欄位值

```
df.melt(id_vars='Name')
```

	Name	variable	value
0	John	Course	Masters
1	Bob	Course	Graduate
2	Shiela	Course	Graduate
3	John	Age	27
4	Bob	Age	23
5	Shiela	Age	21

```
df.melt(value_vars='Name')
```

	variable	value
0	Name	John
1	Name	Bob
2	Name	Shiela

重新組織資料

在做資料分析時很常要重新組織資料，在裡面最靈活好用的就是 .pivot() 函數

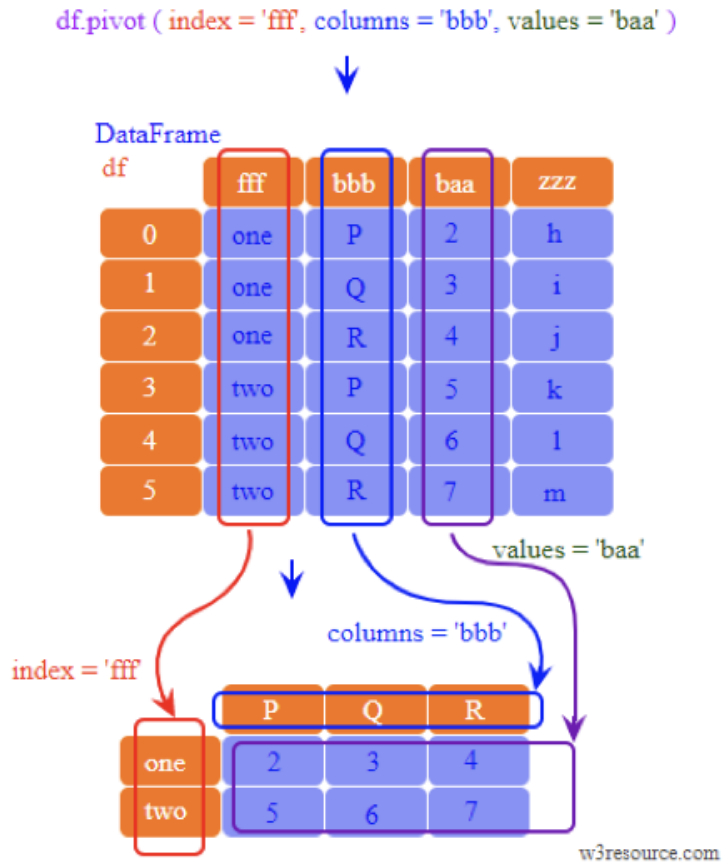
- .pivot()函數根據給定的索引/列值重新組織給定的 DataFrame，接下來以右表為例做介紹

參數

- index：新資料的索引名稱
- columns：新資料的欄位名稱
- values：新資料的值名稱

	fff	bbb	baa	zzz
0	one	P	2	h
1	one	Q	3	i
2	one	R	4	j
3	two	P	5	k
4	two	Q	6	l
5	two	R	7	m

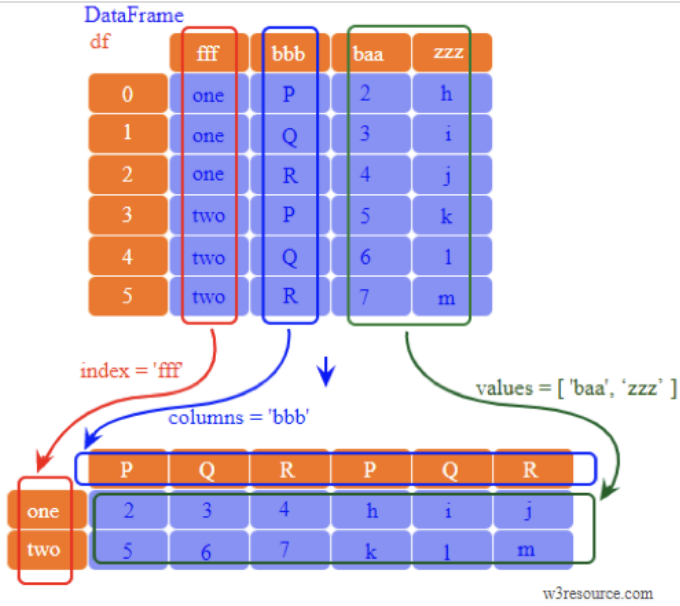
值轉成 baa 欄位



索引轉成f ff 欄位

欄位轉成 bbb 欄位

值轉成 baa、zzz 欄位



---

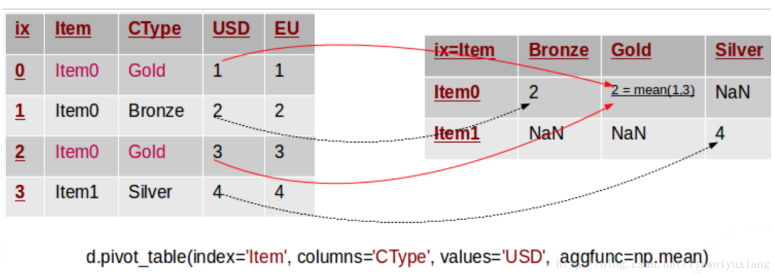
pivot_table 方法實現了類似 pivot 方法的功能，它可以在指定的列和行有重複的情況下使用，我們可以使用均值、中值或其他的聚合函式來計算重複條目中的單個值。

參數

- index : 新資料的索引名稱
- columns : 新資料的欄位名稱
- values : 新資料的值名稱
- aggfunc : 重複數字的函數邏輯(均值、中值或其他的聚合函式)

---

table.pivot_table(index='Item',columns='CType',values='USD', aggfunc=np.mean)



aggfunc=np.mean，所以針對這兩筆資料做平均

知識點回顧

索引轉欄位 .unstack()、欄位轉索引 .stack()，注意都是由最外層開始轉換。

欄位名稱轉為欄位值.melt()，其中參數

- id_vars：不需要被轉換的列名
- value_vars：需要轉換的列名，如果剩下的列全部都要轉換，就不用寫了。

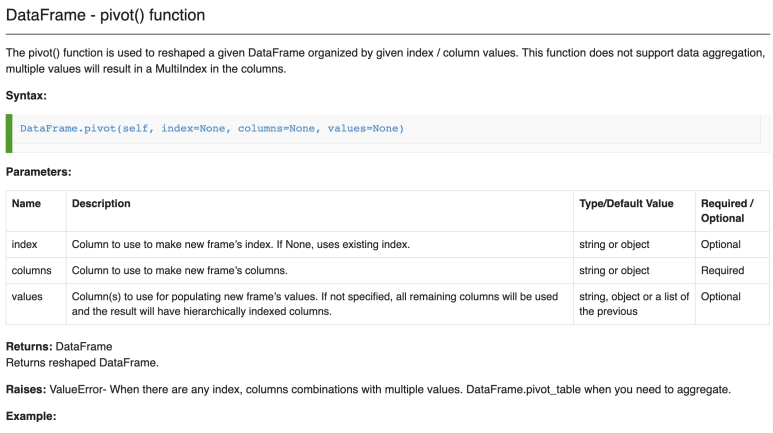
重新組織資料.pivot()，其中參數

- index：新資料的索引名稱
- columns：新資料的欄位名稱
- values：新資料的值名稱

參考資料

pivot

網站：[Pandas DataFrame：pivot() function]



DataFrame - pivot() function

The pivot() function is used to reshaped a given DataFrame organized by given index / column values. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns.

Syntax:

```
DataFrame.pivot(self, index=None, columns=None, values=None)
```

Parameters:

Name	Description	Type/Default Value	Required / Optional
index	Column to use to make new frame's index. If None, uses existing index.	string or object	Optional
columns	Column to use to make new frame's columns.	string or object	Required
values	Column(s) to use for populating new frame's values. If not specified, all remaining columns will be used and the result will have hierarchically indexed columns.	string, object or a list of the previous	Optional

Returns: DataFrame
Returns reshaped DataFrame.

Raises: ValueError- When there are any index, columns combinations with multiple values. DataFrame.pivot_table when you need to aggregate.

Example:

stack & unstack

摘要

前面給大家分享了pandas做資料合併的兩篇[pandas.merge]和[pandas.cancat]的用法。今天這篇主要講的是pandas的DataFrame的軸旋轉操作，stack和unstack的用法。

首先，要知道以下五點：
1.stack：將資料的列“旋轉”為行
2.unstack：將資料的行“旋轉”為列
3.stack和unstack預設操作為最內層
4.stack和unstack預設旋轉軸的級別將會成果結果中的最低級別（最內層）
5.stack和unstack為一組逆運算操作
第一點和第二點以及第五點比較好懂，可能乍看第三點和第四點會不太理解，沒關係，看看具體下面的例子，你就懂了。

1. 建立DataFrame,行索引名為state，列索引名為number

```
import pandas as pd
import numpy as np
data = pd.DataFrame(np.arange(6).reshape((2,3)),index=pd.Index(['Ohio','Colorado'],name='sta
                  ,columns=pd.Index(['one','two','three'],name='number'))
data
```

---

melt

網站：[Python Pandas.melt（）使用及代碼示例]

為了簡化表中數據的分析，我們可以使用Python中的Pandas將數據重塑為更計算機友好的形式。Pandas.melt（）是創建的功能之一
。Pandas.melt（）取消將DataFrame從寬格式轉換為長格式。melt
（）函數很有用，可以將DataFrame壓縮為一種格式，其中一列或多列是標識符變量，而所有其他列（被視為測量變量）都不會旋轉到行軸，僅留下兩個非標識符列，變量和值。
用法：

```
pandas.melt（frame，id_vars = None，value_vars = None，
 var_name = None，value_name ='value'，col_level = None）
```

參數：

- **框架：** DataFrame

  id_vars [元組，列表或ndarray，可選]：使用標識符變量的列

  。**value_vars [元組，列表或ndarray，可選]：**要取消透視的列。如果未指定，則使用未。設置為id_vars的所有列

  VAR_NAME [標量]：用於“變量”列的名稱如果為無，則使用frame.columns.name或“可變的”。

  值名稱[標量，默認為“值”]：用於“值”列的名稱。

  col_level [INT或字符串，可選]：如果列是多指標，則使用此級別進行融合。

延伸閱讀

快速瞭解 Pivot Table 與應用

網站：[Pandas 好好用系列 | 快速瞭解 Pivot Table 與應用]

```
大熊貓。數據透視表（data，values = None，index = None，columns = None，
aggfunc ='mean'，fill_value = None，margins = False，dropna = True，
margins_name ='All'，觀察到的= False）
```

常用參數：

- **data**：讀取你要使用的 DataFrame
- **index**：必要參數。此處輸入不想要變動的數據，作為想要比較的欄位基礎，該數據會成為第一欄的索引（index），此處能以 list、array 等方式輸入多個 index，則結果會以巢狀的方式呈現。
- **values**：可選。可以對需要計算的數據做篩選，如果以 list、array 等方式輸入多個 value，則能夠分別獲得該欄位的不同數值。
- **columns**：可選。用以分割數據，去選出想比較的特定欄位。
- **aggfunc**：function 參數。是 Pivot Table 裡最厲害的功能，能夠引入 max、min 等內建參數，甚至能自訂 function 使用。

選用參數：

- **fill_value**：用特定值取代 NULL 的欄位。
- **margins**：布林值，用來確認是否顯示該欄位的加總。
- **margins_name**：字串，用來顯示上面 margin 增加的列或欄的名稱。

一文看懂透視表 pivot_table

網站：[Pandas | 一文看懂透視表pivot_table]

為什麼要使用pivot_table?

- 靈活性高，可以隨意定制你的分析計算要求
- 脈絡清晰易於理解數據
- 操作性強，報表神器

如何使用pivot_table?

首先讀取數據，作為一個老火密，本文將火箭隊當家吉祥物James_Harden本賽季比賽數據作為數據集進行講解，就是下面這個大鬍子。