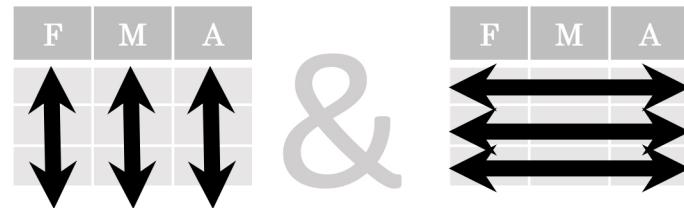


Pandas

資料角力快查表

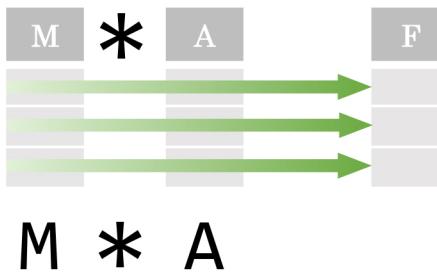
<http://pandas.pydata.org>

整齊資料 - 使用Pandas執行資料角力的基礎



在一個整齊的資料框架中：
每個變數被以欄的方式儲存。 每個觀測被以列的方式儲存。

整齊資料要歸功於Pandas向量式的運算模式。
Pandas會在你操作變數時自動持續觀測。沒有
其他格式能比Pandas更加直覺。



語法 - 創建資料框架

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a" : [4, 5, 6],
     "b" : [7, 8, 9],
     "c" : [10, 11, 12]},
    index = [1, 2, 3])
```

指定每欄的值。

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

指定每列的值。

		a	b	c
n	v			
d	1	4	7	10
e	2	5	8	11
		6	9	12

```
df = pd.DataFrame(
    {"a" : [4, 5, 6],
     "b" : [7, 8, 9],
     "c" : [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d',1),('d',2),('e',2)],
        names=['n','v']))
```

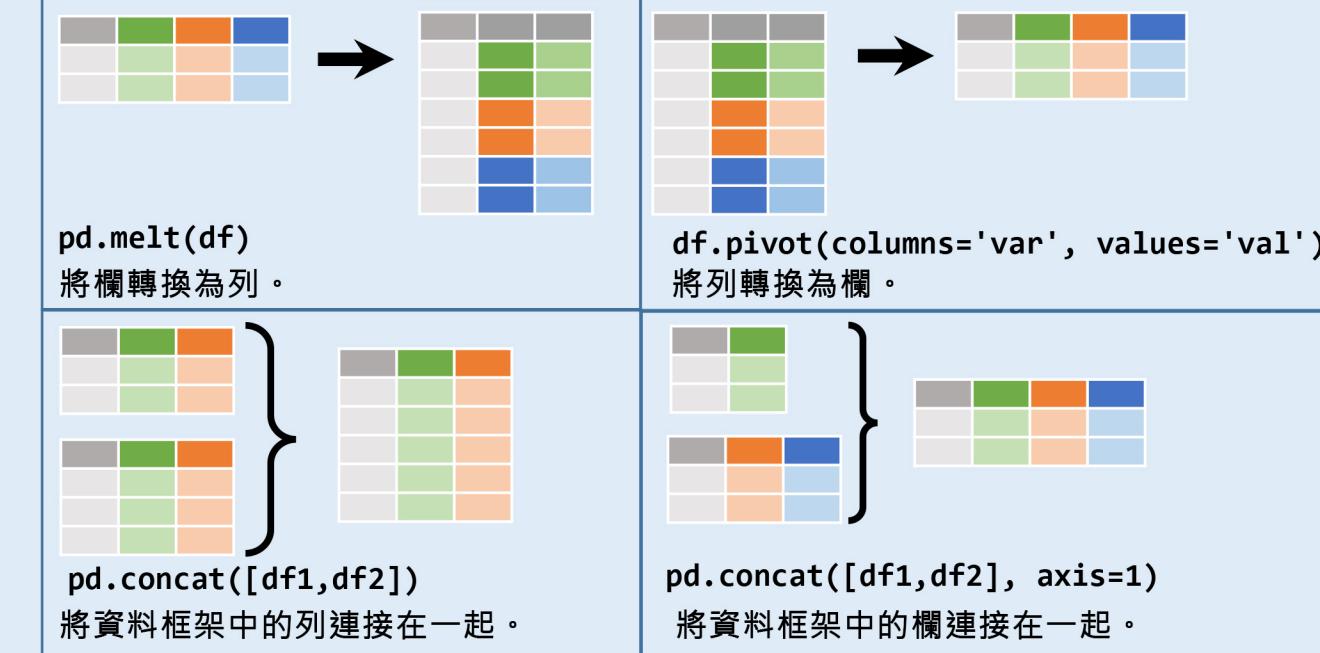
創建多索引的資料框架。

方法鍊 (鏈接編程)

多數Pandas中的函數都是回傳一個資料框架，
下一個函數便可直接運行在回傳的結果上。
同時提升程式碼閱讀性。

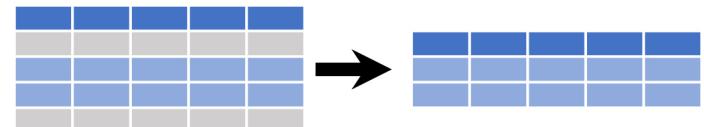
```
df = (pd.melt(df)
      .rename(columns={
          'variable' : 'var',
          'value' : 'val'})
      .query('val >= 200'))
```

資料塑形 - 改變資料的排版



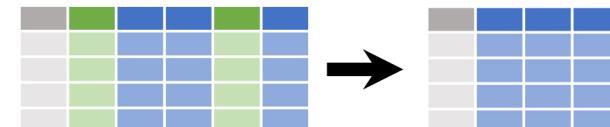
- `df.sort_values('mpg')`
依據列將某欄排序 (數值由低至高)。
- `df.sort_values('mpg', ascending=False)`
依據列將某欄排序 (數值由高至低)。
- `df.rename(columns = {'y':'year'})`
重新命名某資料框架。
- `df.sort_index()`
排序資料框架的索引。
- `df.reset_index()`
將索引重置為列數，並將索引移至欄。
- `df.drop(columns=['Length', 'Height'])`
從資料框架捨棄某欄。

子集觀測 (列)



- `df[df.Length > 7]`
取出符合篩選條件的列。
- `df.drop_duplicates()`
移除重複列 (僅考慮欄)。
- `df.head(n)`
取出前 n 列。
- `df.tail(n)`
取出末 n 列。
- `df.sample(frac=0.5)`
隨機取出指定百分比的列。
- `df.sample(n=10)`
隨機取出 n 列。
- `df.iloc[10:20]`
取出指定範圍的列。
- `df.nlargest(n, 'value')`
取出前 n 列並排序。
- `df.nsmallest(n, 'value')`
取出末 n 列並排序。

子集變數 (欄)



- `df[['width', 'length', 'species']]`
以欄位名稱取出數欄。
- `df['width'] or df.width`
以欄位名稱取出指定欄。
- `df.filter(regex='regex')`
取出欄位名稱符合正規表達式的欄。

正規表達式範例

'.'	包含句號 '!' 的字串
'Length\$'	結尾包含 'Length' 的字串
'^Sepal'	開頭包含 'Sepal' 的字串
'^x[1-5]\$'	開頭為 'x' 且結尾包含 1, 2, 3, 4, 5 的字串
'^(?!Species\$).*''	除了有包含 'Species' 以外的字串。

Python條件邏輯符號說明 (Pandas也是)

<	小於	!=	不等於
>	大於	df.column.isin(values)	組群關係
==	等於	pd.isnull(obj)	是 NaN
<=	小於等於	pd.notnull(obj)	不是NaN
>=	大於等於	&, , ~, df.any(), df.all()	邏輯判斷 and, or, not, xor, any, all

- `df.loc[:, 'x2':'x4']`
取出 'x2' 至 'x4' 的欄 (包含)。
- `df.iloc[:, [1,2,5]]`
取出第 1, 2, 5 位的欄 (首欄為第 0 位)。
- `df.loc[df['a'] > 10, ['a', 'c']]`
取出特定欄中符合篩選條件的列。

概述欄位

`df['w'].value_counts()`

計算欄中有多少不同列值。

`len(df)`

計算資料框架的列數。

`df['w'].nunique()`

計算欄中有多少不同值。

`df.describe()`

欄的基本統計資訊 (分組)。



Pandas提供大量概述函數可以針對不同的Pandas物件做運算 (資料框架欄、序列、分組、擴充和滾動，見下方)並產生個別的值。使用這些函數時，針對每欄回傳的物件會是Pandas序列。範例：

`sum()`

每個物件的總和。

`count()`

計算每個物件非

NA/null 值的數量。

`median()`

計算每個物件的中位數。

`quantile([0.25,0.75])`

計算每個物件的分位數。

`apply(function)`

在每個物件上執行函數。

`min()`

每個物件的最小值。

`max()`

每個物件的最大值。

`mean()`

每個物件的平均值。

`var()`

每個物件的變異數。

`std()`

每個物件的標準差。

資料分組

`df.groupby(by="col")`

回傳一個欄名稱 'col' 中依值分組的分組物件。

`df.groupby(level="ind")`

回傳一個索引名稱 'ind' 中依值分組的分組物件。

所有前列的概述函數都可執行於組。其他分組函數有：

`size()`

每組的尺寸。

`agg(function)`

使用函數集合組。

視窗

`df.expanding()`

回傳擴充物件。

`df.rolling(n)`

滾動視窗，按指定週期計算。

處理缺失值

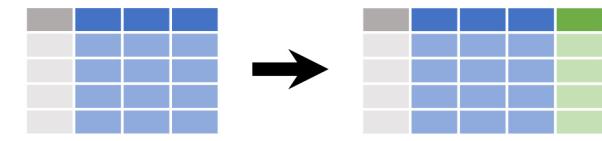
`df.dropna()`

捨棄任何包含 NA/null 欄的列。

`df.fillna(value)`

以某值取代所有 NA/null。

創建新欄



`df.assign(Area=lambda df: df.Length*df.Height)`

運算並新增一或多欄。

`df['Volume'] = df.Length*df.Height*df.Depth`

增加一欄。

`pd.qcut(df.col, n, labels=False)`

將欄分為 n 箱。



Pandas提供大量能操作一 (Pandas序列) 或所有欄的向量函數，這些函數為每欄產生個別的數值向量，或為每個序列產生個別序列。範例：

`max(axis=1)`

每個元素的最大值。

`clip(lower=-10,upper=10)`

依照輸入閥值修剪向量。

`min(axis=1)`

每個元素的最小值。

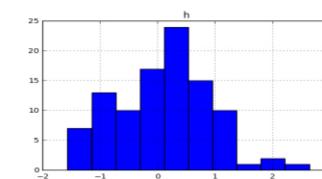
`abs()`

絕對值。

繪圖

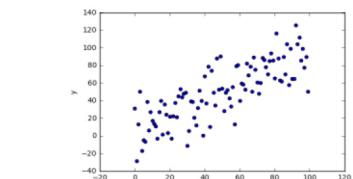
`df.plot.hist()`

累積計算依每欄繪製長條圖。



`df.plot.scatter(x='w',y='h')`

使用點組繪製散點圖。



合併資料組

`adf`

x1	x2
A	1
B	2
C	3

`bdf`

x1	x3
A	T
B	F
D	T



標準接合

x1	x2	x3
A	1	T
B	2	F
C	3	NaN

`pd.merge(adf, bdf, how='left', on='x1')`
依照相對應的列把 bdf 加入 adf。

x1	x2	x3
A	1.0	T
B	2.0	F
D	NaN	T

`pd.merge(adf, bdf, how='right', on='x1')`
依照相對應的列把 adf 加入 bdf。

x1	x2	x3
A	1	T
B	2	F
C	NaN	

`pd.merge(adf, bdf, how='inner', on='x1')`
合併資料，只保留兩者都有值的列。

x1	x2	x3
A	1	T
B	2	F
C	NaN	
D	NaN	T

`adf[adf.x1.isin(bdf.x1)]`
篩選出 adf 中所有與 bdf 有所對應的列。

x1	x2
C	3

`adf[~adf.x1.isin(bdf.x1)]`
篩選出 adf 中所有與 bdf 不對應的列。

`ydf`

x1	x2
A	1
B	2
C	3

`zdf`

x1	x2
B	2
C	3
D	4

集合操作

x1	x2
B	2
C	3
D	4

`pd.merge(ydf, zdf)`
ydf 與 zdf 中皆有出現的列 (交集)。

x1	x2
A	1

`pd.merge(ydf, zdf, how='outer')`
ydf 與 zdf 中所有有出現過的列 (並集)。

`pd.merge(ydf, zdf, how='outer', indicator=True).query('_merge == "left_only"').drop(columns=['_merge'])`
ydf 有出現但不在 zdf 中的列 (差集)。