14. Tổ chức lại bảng dữ liệu (phần 2)

Bài học gồm có:

+ melting dataframe

+ pivot table

Melting

Cho hai dạng biểu diễn dữ liệu sau:

Dạng 1

year	product	turnover
2011	Iphone	10000
2011	SS	30000
2011	LG	123000
2012	SS	134500
2012	LG	90000
2012	Iphone	23400
2013	Iphone	56000
2013	LG	234000
2013	SS	1234567

Dạng 2

product	year	Iphone	LG	SS
0	2011	10000	123000	30000
1	2012	23400	90000	134500
2	2013	56000	234000	1234567

Dạng 1 có thể chuyển qua dạng 2 thông qua pivot() và reset_index() như đã học ở bài trước. Cụ thể như sau:

```
>>> import pandas as pd
>>> df = pd.DataFrame([[2011,'Iphone',10000],[2011,'SS',30000],[2011,'LG',123000],[2012,'SS',134500],[2012,'LG',90000],[2012,'Iphone',23400],[2013,'Iphone',56000],[2013,'LG',234000],[201
>>> df
   year product  turnover
0  2011  Iphone     10000
1  2011      SS     30000
2  2011      LG    123000
3  2012      SS    134500
4  2012      LG     90000
5  2012  Iphone     23400
6  2013  Iphone     56000
7  2013      LG    234000
8  2013      SS   1234567
>>> df = df.pivot(index='year', columns='product',values = 'turnover').reset_index()
>>> df
product  year  Iphone      LG       SS
0        2011   10000  123000    30000
1        2012   23400   90000   134500
2        2013   56000  234000  1234567
>>>
```

Câu hỏi đặt ra là: làm thế nào để chuyển đổi dữ liệu từ dạng 2 về dạng 1?

Pandas đã cung cấp phương thức để làm việc này (ta gọi đó là unpivot), đó là:**pandas.melt(frame, id_vars=None, value_vars=None, var_name=None, value_name='value', col_level=None).**

Chức năng này rất hữu ích để chuyển một dataframe vào một định dạng mà một hoặc nhiều cột là các biến số định danh (id_vars), trong khi tất cả các cột khác, được coi là các biến đo được (value_vars), được chuyển đổi từ cột qua hàng (unpivoted), chỉ để lại hai không định danh cột, 'variable' và 'value'.

Ví dụ sau sẽ chuyển mọi cột thành hàng.

```
>>> pd.melt(df)
   product    value
0     year     2011
1     year     2012
2     year     2013
3   Iphone    10000
4   Iphone    23400
5   Iphone    56000
6       LG   123000
7       LG    90000
8       LG   234000
9       SS    30000
10      SS   134500
11      SS  1234567
```

Tuy nhiên ta thấy rằng ví dụ trên không thể đưa dạng 2 về dạng 1. Ta có thể giải quyết vấn đề này qua cách sau sử dụng id_var = ‘year’ để không chuyển cột ‘year’ thành hàng và đặt tên cho hai cột không định danh tương ứng là var_name=’product’ và value_name = ‘turnover’.

```
>>> pd.melt(df,id_vars=['year'],var_name='product',value_name='turnover')
   year product  turnover
0  2011  Iphone     10000
1  2012  Iphone     23400
2  2013  Iphone     56000
```

```
3  2011     LG    123000
4  2012     LG     90000
5  2013     LG    234000
6  2011     SS     30000
7  2012     SS    134500
8  2013     SS   1234567
>>>
```

Pivot Table

Trong phần này các bạn cần download file. Và tạo dataframe như sau:

```
>>> df = pd.read_csv("sales-funnel.csv",sep=",")
>>> df
    Account                       Name           Rep        Manager  \
0    714466              Trantow-Barrows   Craig Booker   Debra Henley
1    714466              Trantow-Barrows   Craig Booker   Debra Henley
2    714466              Trantow-Barrows   Craig Booker   Debra Henley
3    737550  Fritsch, Russel and Anderson   Craig Booker   Debra Henley
4    146832                 Kiehn-Spinka  Daniel Hilton   Debra Henley
5    218895                   Kulas Inc   Daniel Hilton   Debra Henley
6    218895                   Kulas Inc   Daniel Hilton   Debra Henley
7    412290                Jerde-Hilpert     John Smith   Debra Henley
8    740150                  Barton LLC     John Smith   Debra Henley
9    141962                  Herman LLC    Cedric Moss  Fred Anderson
10   163416                  Purdy-Kunde   Cedric Moss  Fred Anderson
11   239344                  Stokes LLC    Cedric Moss  Fred Anderson
12   239344                  Stokes LLC    Cedric Moss  Fred Anderson
13   307599   Kassulke, Ondricka and Metz    Wendy Yule  Fred Anderson
14   688981                 Keeling LLC     Wendy Yule  Fred Anderson
15   729833                   Koepp Ltd     Wendy Yule  Fred Anderson
16   729833                   Koepp Ltd     Wendy Yule  Fred Anderson

       Product  Quantity   Price     Status
0          CPU         1   30000  presented
1     Software         1   10000  presented
2  Maintenance         2    5000    pending
3          CPU         1   35000   declined
4          CPU         2   65000        won
5          CPU         2   40000    pending
6     Software         1   10000  presented
7  Maintenance         2    5000    pending
8          CPU         1   35000   declined
9          CPU         2   65000        won
10         CPU         1   30000  presented
11 Maintenance         1    5000    pending
12    Software         1   10000  presented
13 Maintenance         3    7000        won
14         CPU         5  100000        won
15         CPU         2   65000   declined
16     Monitor         2    5000  presented
>>>
```

Chúng ta đã học ở các phần trước pivot và melt như là tool để reshape dữ liệu bảng. Tuy nhiên pivot() không phải lúc nào cũng làm được. Cùng xem xét ví dụ phía trên: có khá nhiều dòng là kết hợp giữa “Manager” và “Rep” và mong muốn của chúng ta là muốn tổng kết giá thành ‘price’ cho sự kết hợp giữa “Manager” và “Rep”. Nếu ta dùng pivot() thì sẽ không thể làm việc được với untidy data như trên.

```
>>> df.pivot(index='Manager',columns='Rep',values = 'Price')
    raise ValueError('Index contains duplicate entries, '
ValueError: Index contains duplicate entries, cannot reshape
>>>
```

Lỗi bên trên là do pivot yêu cầu cặp (index, columns) phải là unique gắn với một value trong bảng mới. Và đó là lí do ra đời của pivot_table() nó giải quyết nhiều values cho mỗi cặp (index, columns) sử dụng một reduction thường thì reduction là giá trị trung bình (average). Chúng ta có thể sử dụng các reduction khác thông qua từ khóa aggfunc, ví dụ, aggfunc=’sum’.

```
>>> df.pivot_table(index='Manager',columns='Rep',values = 'Price')
Rep           Cedric Moss  Craig Booker  Daniel Hilton  John Smith  \
Manager
Debra Henley          NaN       20000.0   38333.333333     20000.0
Fred Anderson     27500.0           NaN           NaN         NaN

Rep           Wendy Yule
Manager
Debra Henley         NaN
Fred Anderson    44250.0
>>>
```

Đôi khi khá là hữu ích khi ta muốn biết tổng của tất cả các kết hợp qua từ khóa margins=True như ví dụ sau:

```
>>> df.pivot_table(index='Manager',columns='Rep',values = 'Price',margins=True)
Rep           Cedric Moss  Craig Booker  Daniel Hilton  John Smith  \
Manager
Debra Henley          NaN       20000.0   38333.333333     20000.0
Fred Anderson     27500.0           NaN           NaN         NaN
All               27500.0       20000.0   38333.333333     20000.0

Rep           Wendy Yule           All
Manager
Debra Henley         NaN  26111.111111
Fred Anderson    44250.0  35875.000000
All              44250.0  30705.882353
>>>
```

Tham khảo dữ liệu tại http://pbpython.com/pandas-pivot-table-explained.html

Chi tiết hơn về phương thức pivot_table(), các bạn có thể xem thêm tại web

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.pivot_table.html

Lưu ý: Những giá trị được gán vào id_vars không được giữ vị trí là index trong DataFrame đầu vào. Cần reset_index() trước khi sử dụng pivot_table().

Kết Luận

Trong bài này chúng tôi tiếp tục giới thiệu thêm hai phương pháp nữa trong pandas hỗ trợ rearranging và reshaping dữ liệu. Trước tiên phải kể đến là pd.melt() được xem như một phương thức “Unpivot” cho phép chúng ta chuyển một DataFrame từ wide format to long format mang ý nghĩa tương tự như .stack(), bên cạnh đó nó còn hỗ trợ tùy chọn biến số định danh. Nếu như .pivot() mang một yếu điểm là nó yêu cầu cặp (index, columns) phải là duy nhất gắn với một value trong bảng (hay gọi là tidy data) thì .pivot_table() ra đời để giải quyết yếu điểm trên qua việc kết hợp với một phương thức reduction qua đối số aggfunc.