Group By và Pivot Table

06-01-2021

1 Giới thiệu

Group By (nhóm dữ liệu) và Pivot table (bảng tổng hợp) là hai công cụ để tổng hợp dữ liệu thường dùng.

```
In [2]: # dữ liệu mẫu
        import numpy as np
        import pandas as pd
        import seaborn as sns

        data = sns.load_dataset('planets')
        data.head()
```

Out[2]:

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300	7.10	77.40	2006
1	Radial Velocity	1	874.774	2.21	56.95	2008
2	Radial Velocity	1	763.000	2.60	19.84	2011
3	Radial Velocity	1	326.030	19.40	110.62	2007
4	Radial Velocity	1	516.220	10.50	119.47	2009

2 Group By

2.1 Giới thiệu

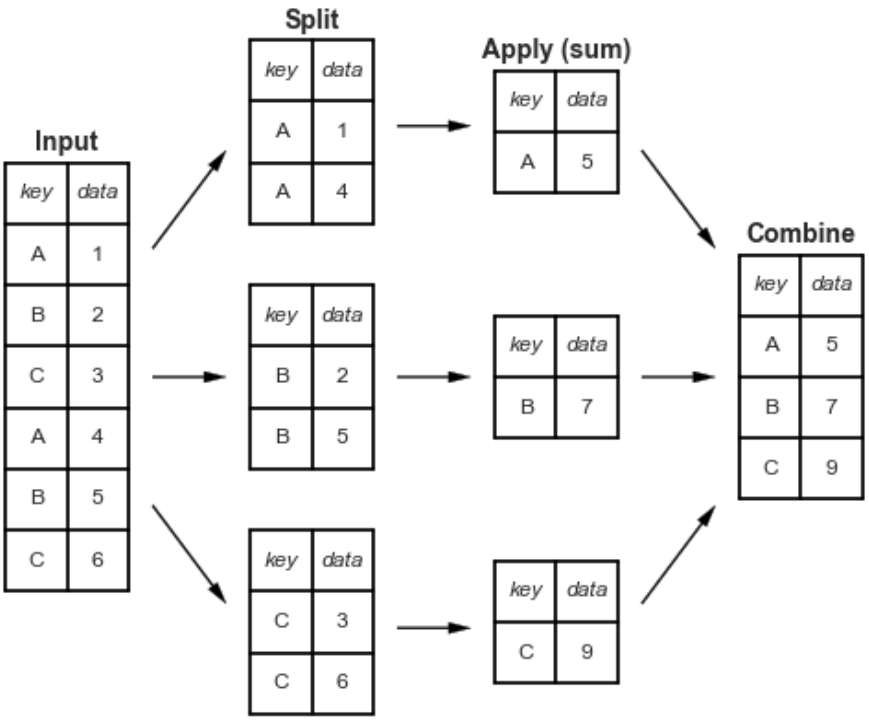
Các hàm tổng hợp thông tin cho người dùng cái nhìn tổng quan với dữ liệu. Tuy nhiên, chúng ta thường có nhu cầu tổng hợp thông tin dựa theo label của dữ liệu, chẳng hạn:

- Thống kê doanh thu theo tháng.
- Thống kê số lượng học sinh từng phân loại.
- ...

Trong những tình huống như thế, chúng ta cần dùng đến phương pháp là **Group By** (gộp dữ liệu).

Group By có thể hiểu là 3 thao tác liên tiếp:

- Split (chia nhỏ): chia dữ liệu thành từng nhóm dựa trên điều kiện nào đó.
- Apply (tính toán): sử dụng một hàm nào đó để tổng hợp dữ liệu.
- Combine (kết hợp): kết hợp kết quả tính toán của từng nhóm thành kết quả chung.



Group By

2.2 Split

Bước đầu tiên chính là chia tách dữ liệu thành từng nhóm theo một điều kiện nào đó.

Để thực hiện việc này, chúng ta sẽ sử dụng phương thức `.groupby()` của `DataFrame`. Phương thức này cung cấp khá nhiều cách thức chia dữ liệu, tuy nhiên, chúng ta sẽ chỉ tìm hiểu cách chia dữ liệu đơn giản nhất, đó là chia theo giá trị của một hay nhiều cột.

Để chia `DataFrame` theo dữ liệu của một hay nhiều cột, chúng ta dùng như sau:

```
.groupby(<danh_sách_cột_cần_chia>)
```

Kết quả trả về là một đối tượng `GroupBy`.

```
In [3]: g = data.groupby('method')
        print(g)
        print(type(g))
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001B67B657B38>
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

Chúng ta có thể lặp qua đối tượng này bằng cách:

```
for label, data in <groups>:
    <hành_động>
```

Trong đó:

- `<groups>` là đối tượng `GroupBy`.
- `label` là đại diện cho các giá trị chia tách.

- data là đại diện cho dữ liệu được chia tách.

```
In [3]: for l, d in data.groupby('method'):
            s = d.shape[0] # lấy số dòng trong mỗi nhóm
            print(f'Nhóm {l} có {s} dòng dữ liệu.')
```

```
Nhóm Astrometry có 2 dòng dữ liệu.
Nhóm Eclipse Timing Variations có 9 dòng dữ liệu.
Nhóm Imaging có 38 dòng dữ liệu.
Nhóm Microlensing có 23 dòng dữ liệu.
Nhóm Orbital Brightness Modulation có 3 dòng dữ liệu.
Nhóm Pulsar Timing có 5 dòng dữ liệu.
Nhóm Pulsation Timing Variations có 1 dòng dữ liệu.
Nhóm Radial Velocity có 553 dòng dữ liệu.
Nhóm Transit có 397 dòng dữ liệu.
Nhóm Transit Timing Variations có 4 dòng dữ liệu.
```

2.3 Apply và Combine

2.3.1 Tổng hợp thông tin cơ bản

Khi đã có đối tượng GroupBy, bạn có thể sử dụng các phương thức tổng hợp thông tin cơ bản. pandas sẽ tự động sử dụng các phương thức lên từng nhóm và kết hợp chúng lại thành một kiểu dữ liệu phù hợp.

```
In [4]: g = data.groupby('method')
        g.mean()
```

Out[4]:

method	number	orbital_period	mass	distance	year
Astrometry	1.000000	631.180000	NaN	17.875000	2011.500000
Eclipse Timing Variations	1.666667	4751.644444	5.125000	315.360000	2010.000000
Imaging	1.315789	118247.737500	NaN	67.715937	2009.131579
Microlensing	1.173913	3153.571429	NaN	4144.000000	2009.782609
Orbital Brightness Modulation	1.666667	0.709307	NaN	1180.000000	2011.666667
Pulsar Timing	2.200000	7343.021201	NaN	1200.000000	1998.400000
Pulsation Timing Variations	1.000000	1170.000000	NaN	NaN	2007.000000
Radial Velocity	1.721519	823.354680	2.630699		
Transit	1.954660	21.102073	1.470000		
Transit Timing Variations	2.250000	79.783500	NaN		

```
Radial Velocity                  51.600208  2007.518987
Transit                         599.298080  2011.236776
Transit Timing Variations      1104.333333  2012.500000
```

```
In [5]: g[['mass', 'distance']].max()
```

```
Out[5]:                                 mass  distance
        method
        Astrometry                       NaN     20.77
        Eclipse Timing Variations       6.05    500.00
        Imaging                          NaN    165.00
        Microlensing                     NaN   7720.00
        Orbital Brightness Modulation    NaN   1180.00
        Pulsar Timing                    NaN   1200.00
        Pulsation Timing Variations      NaN       NaN
        Radial Velocity                25.00    354.00
        Transit                         1.47   8500.00
        Transit Timing Variations        NaN   2119.00
```

2.3.2 `.agg()`

`.agg()` là một phương thức của đối tượng `GroupBy` cho phép người dùng:

1. Dùng nhiều phương thức tổng hợp khác nhau.
2. Chỉ ra phương thức tổng hợp cụ thể cho mỗi cột.

Đối với công dụng thứ nhất, bạn dùng cú pháp:

```
.agg(<danh_sách_hàm_tổng_hợp>)
```

Lưu ý: hàm tổng hợp là tên gọi chung cho tất cả các hàm mà đầu vào là một danh sách và đầu ra là một giá trị vô hướng.

```
In [4]: # áp dụng min và max cho cột mass
        g[['mass', 'distance']].agg([min, max])
```

```
Out[4]:                                   mass          distance
                                           min    max       min      max
        method
        Astrometry                         NaN    NaN     14.98    20.77
        Eclipse Timing Variations       4.2000   6.05    130.72   500.00
        Imaging                            NaN    NaN      7.69   165.00
        Microlensing                       NaN    NaN   1760.00  7720.00
        Orbital Brightness Modulation      NaN    NaN   1180.00  1180.00
        Pulsar Timing                      NaN    NaN   1200.00  1200.00
        Pulsation Timing Variations        NaN    NaN       NaN      NaN
        Radial Velocity                 0.0036  25.00      1.35   354.00
        Transit                         1.4700   1.47     38.00  8500.00
        Transit Timing Variations          NaN    NaN    339.00  2119.00
```

Đối với công dụng thứ hai, bạn có cú pháp:

```
.agg(
    tên_1 = (cột_muốn_tổng_hợp_1, hàm_tổng_hợp_1),
    tên_2 = (cột_muốn_tổng_hợp_2, hàm_tổng_hợp_2),
    ...
)
```

```
In [7]: # ví dụ, dùng hàm min, hàm np.std cho cột mass, dùng hàm max cho cột distance
        g.agg(
            mass_min = ('mass', min),
            mass_std = ('mass', np.std),
            distance_max = ('distance', max)
        )
```

Out[7]:

method	mass_min	mass_std	distance_max
Astrometry	NaN	NaN	20.77
Eclipse Timing Variations	4.2000	1.308148	500.00
Imaging	NaN	NaN	165.00
Microlensing	NaN	NaN	7720.00
Orbital Brightness Modulation	NaN	NaN	1180.00
Pulsar Timing	NaN	NaN	1200.00
Pulsation Timing Variations	NaN	NaN	NaN
Radial Velocity	0.0036	3.825883	354.00
Transit	1.4700	NaN	8500.00
Transit Timing Variations	NaN	NaN	2119.00

3 Pivot Table

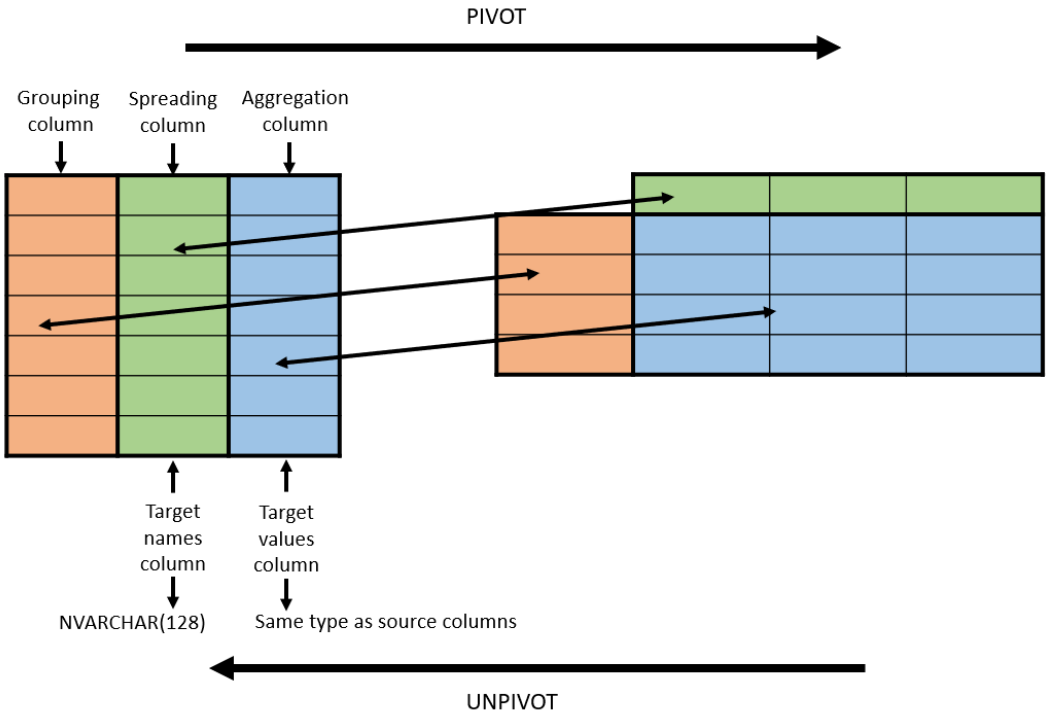
Pivot table (bảng tổng hợp) là một phương thức tổng hợp thông tin thường được sử dụng trong các phần mềm bảng tính (ví dụ Excel). Đối với `DataFrame`, chúng ta có phương thức `.pivot_table()` với cú pháp:

```
.pivot_table(
    values = <cột_cần_tổng_hợp>,
    index = <cột_làm_label_dòng>,
    columns = <cột_làm_label_cột>,
    aggfunc = <hàm_tổng_hợp>,
    margins
)
```

Khi gọi `.pivot_table()`, kết quả trả về là một `DataFrame` với:

- Label dòng là các giá trị khác nhau của `index`.
- Label cột là các giá trị khác nhau của `columns`.
- Giá trị của một ô là kết quả của hàm tổng hợp `aggfunc` của những giá trị trong `DataFrame` gốc, các giá trị này được nhóm theo `index` và `columns`.
- Tham số `margins` sẽ thêm phần tổng hợp theo cột, dòng. Có 2 giá trị là `True` và `False`. Mặc định là `False`.

```
In [45]: # thêm cột mới là thập kỷ phát hiện
         data = data.assign(decade = data.year.astype(str).str[0:3] + '0s')
```



Pivot

```
# tạo bảng tổng hợp với index = 'method', columns = 'decade'
data.pivot_table(
    values = 'mass',
    index = 'method',
    columns = 'decade',
    aggfunc = np.mean
)
```

Out[45]:

decade	1980s	1990s	2000s	2010s
method				
Eclipse Timing Variations	NaN	NaN	6.050000	4.200000
Radial Velocity	11.68	2.727128	3.204472	1.674492
Transit	NaN	NaN	NaN	1.470000

In [39]:
```
# tạo bảng tổng hợp với margins = True
data.pivot_table(
    values = 'distance',
    index = 'method',
    columns = 'decade',
    aggfunc = np.mean,
    margins = True
)
```

Out[39]:

decade	1980s	1990s	2000s	2010s	\
method					
Astrometry	NaN	NaN	NaN	17.875000	
Eclipse Timing Variations	NaN	NaN	130.720000	500.000000	
Imaging	NaN	NaN	59.801875	75.630000	

```
Microlensing                      NaN        NaN         NaN  4144.000000
Orbital Brightness Modulation     NaN        NaN         NaN  1180.000000
Pulsar Timing                     NaN        NaN         NaN  1200.000000
Radial Velocity                 40.57  25.846786   50.502193    56.913350
Transit                           NaN        NaN  723.257045   568.997000
Transit Timing Variations         NaN        NaN         NaN  1104.333333
All                             40.57  25.846786  132.900110   395.098462
```

```
decade                                 All
method
Astrometry                       17.875000
Eclipse Timing Variations       315.360000
Imaging                          67.715937
Microlensing                   4144.000000
Orbital Brightness Modulation  1180.000000
Pulsar Timing                  1200.000000
Radial Velocity                  51.600208
Transit                         599.298080
Transit Timing Variations      1104.333333
All                             264.069282
```

Về bản chất, `.pivot_table(values, index, columns, aggfunc)` chính là việc dùng cú pháp sau:

```
.groupby([index + columns])[values].agg(aggfunc).unstack(columns)
```

```
In [33]: data.groupby(['method', 'decade'])['mass'].agg(np.mean).unstack('decade')
```

```
Out[33]: decade                          1980s     1990s     2000s     2010s
         method
         Astrometry                        NaN       NaN       NaN       NaN
         Eclipse Timing Variations         NaN       NaN  6.050000  4.200000
         Imaging                           NaN       NaN       NaN       NaN
         Microlensing                      NaN       NaN       NaN       NaN
         Orbital Brightness Modulation     NaN       NaN       NaN       NaN
         Pulsar Timing                     NaN       NaN       NaN       NaN
         Pulsation Timing Variations       NaN       NaN       NaN       NaN
         Radial Velocity                 11.68  2.727128  3.204472  1.674492
         Transit                           NaN       NaN       NaN  1.470000
         Transit Timing Variations         NaN       NaN       NaN       NaN
```

Tuy nhiên, `.pivot_table()` cũng có những điểm tiện lợi của nó:

- Cú pháp dễ hơn, quen thuộc với những người sử dụng các phần mềm bảng tính.
- Tự động bỏ qua các dòng, cột mà chỉ chứa `NaN`.
- Tham số `margins`.