

82.06 Pandas.melt()_规整数据_转换数据

原创 [mingkoukou](#) 最后发布于2018-11-21 17:33:38 阅读数 5297

[展开](#)

[收藏](#)

转换数据

`df.melt()` 是 `df.pivot()` 逆转操作函数

将列名转换为列数据(columns name → column values), **重构DataFrame:看下面的代码例子**

如果说 `df.pivot()` 将长数据集转换成宽数据集, `df.melt()` 则是将宽数据集变成长数据集

| 参数 | 类型 | 说明 |
|---------|--------------------------|---|
| frame | dataframe | 被 melt 的数据集名称 在 <code>pd.melt()</code> 中使用 |
| id_vars | tuple list ndarray | 可选项 不需要被转换的列名, 在转换后作为标识符列 (不是索引列) |
| | | 可选项 |

| | | |
|--|--------------------------|--|
| value_vars | tuple list ndarray | 需要被转换的现有列 如果未指明，除 id_vars 之外的其他列都被转换 |
| var_name | string | variable 默认值 自定义列名名称 设置 由 ' value_vars ' 组成的新的 column name |
| value_name | string | value 默认值 自定义列名名称 设置 由 ' value_vars ' 的数据组成的新的 column name |
| col_level | int string | 可选项 如果列是 MultilIndex ，则使用此级别 |
| <i># id_vars 标识列，不是索引</i> <i># value_vars 被转换的列名</i> <i># var_name "被转换的列名"组成的新列的 name</i> <i># value_name "被转换的列名"下的数据 组成的新列的 name</i> | | |

```

>>> import pandas as pd
>>> df = pd.DataFrame({'A': {0: 'a', 1: 'b', 2: 'c'},
...                    'B': {0: 1, 1: 3, 2: 5},
...                    'C': {0: 2, 1: 4, 2: 6}})
>>> df

```

```

    A  B  C
0  a  1  2
1  b  3  4
2  c  5  6

```

#保留 B 列

```
>>> df.melt(id_vars=['A'], value_vars=['B'])
```

```

   A variable  value
0  a         B      1
1  b         B      3
2  c         B      5

```

#保留 B C 列, 在B、C条件下的A

```
>>> df.melt(id_vars=['A'], value_vars=['B', 'C'])
```

```

   A variable  value
0  a         B      1
1  b         B      3
2  c         B      5
3  a         C      2
4  b         C      4
5  c         C      6

```

#自定义列名

```
>>> df.melt(id_vars=['A'], value_vars=['B'], var_name='myVarname')
```

```

   A myVarname  myValname
0  a         B          1
1  b         B          3
2  c         B          5

```

#如果 columns 是MultiIndex

```
>>> df.columns = [list('ABC'), list('DEF')]
```

```
>>> df
```

```

   A  B  C
   D  E  F
0  a  1  2
1  b  3  4
2  c  5  6

```

```
>>> df.melt(col_level=0, id_vars=['A'], value_vars=['B'])
  A variable  value
0  a        B      1
1  b        B      3
2  c        B      5

>>> df.melt(id_vars=['A', 'D'], value_vars=[('B', 'E')])
(A, D) variable_0 variable_1  value
0      a          B          E      1
1      b          B          E      3
2      c          B          E      5
```

Melt

df3

| | first | last | height | weight |
|---|-------|------|--------|--------|
| 0 | John | Doe | 5.5 | 130 |
| 1 | Mary | Bo | 6.0 | 150 |

➔

df3.melt(id_vars=['first', 'last'])

| | first | last | variable | value |
|---|-------|------|----------|-------|
| 0 | John | Doe | height | 5.5 |
| 1 | Mary | Bo | height | 6.0 |
| 2 | John | Doe | weight | 130 |
| 3 | Mary | Bo | weight | 150 |

<https://blog.csdn.net/mingyouou>

df.pivot <--> df.melt() 之间的转换操作

```
>>> df
  A  B  C
0  a  1  2
```

```
1  b  3  4
2  c  5  6
```

```
>>> melted=df.melt('A')
```

```
>>> melted
```

| | A | variable | value |
|---|---|----------|-------|
| 0 | a | B | 1 |
| 1 | b | B | 3 |
| 2 | c | B | 5 |
| 3 | a | C | 2 |
| 4 | b | C | 4 |
| 5 | c | C | 6 |

```
>>> reshaped =melted.pivot('A','variable','value')
```

```
>>> reshaped
```

| variable | B | C |
|----------|---|---|
| A | | |
| a | 1 | 2 |
| b | 3 | 4 |
| c | 5 | 6 |

```
>>> reshaped.reset_index()
```

| variable | A | B | C |
|----------|---|---|---|
| 0 | a | 1 | 2 |
| 1 | b | 3 | 4 |
| 2 | c | 5 | 6 |