Pandas – pivot()

Return reshaped DataFrame organized by given index / column values.

DataFrame.pivot(index=None, **columns**=None, **values**=None)

index : str or object or a list of str, optional

Column to use to make new frame's index. If None, uses existing index.

columns : str or object or a list of str

Column to use to make new frame's columns.

values : str, object or a list of the previous, optional

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.

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

```
df.pivot(index='foo',
         columns='bar',
         values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

Pandas – pivot() - example

pivot, reshape DataFrame by given index / column / values

In [18]:
```
df4 = pd.DataFrame({'class':['cycle','cycle','cycle', 'pilates', 'pilates', 'pilates'],
                    'day':['0-Mon', '2-Wed', '5-Sat', '0-Mon', '1-Tue', '4-Fri'],
                    'hour': [12, 10, 15, 9, 10, 14],})
df4
```

Out[18]:

	class	day	hour
0	cycle	0-Mon	12
1	cycle	2-Wed	10
2	cycle	5-Sat	15
3	pilates	0-Mon	9
4	pilates	1-Tue	10
5	pilates	4-Fri	14

In [19]:
```
df_pivot = df4.pivot(index = 'class', columns = 'day', values = 'hour')
df_pivot
```

Out[19]:

day	0-Mon	1-Tue	2-Wed	4-Fri	5-Sat
class					
cycle	12.0	NaN	10.0	NaN	15.0
pilates	9.0	10.0	NaN	14.0	NaN

Pandas – stack() and unstack()

stack : "pivot" a level of the (possibly hierarchical) column labels, returning a DataFrame with an index with a new inner-most level of row labels

df2

		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8

MultiIndex

stacked = df2.stack()

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

MultiIndex

unstack : (inverse operation of stack) "pivot" a level of the (possibly hierarchical) row index to the column axis, producing a reshaped DataFrame with a new inner-most level of column labels

stacked

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

MultiIndex

stacked.unstack()

		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8

MultiIndex

Pandas – stack() and unstack() - example

Stack and unstack

In [20]:
```
df_stack=df_pivot.stack()
df_stack
```

Out[20]:
```
class     day
cycle     0-Mon    12.0
          2-Wed    10.0
          5-Sat    15.0
pilates   0-Mon     9.0
          1-Tue    10.0
          4-Fri    14.0
dtype: float64
```

In [21]:
```
df_unstack=df_stack.unstack()
df_unstack
```

Out[21]:

day	0-Mon	1-Tue	2-Wed	4-Fri	5-Sat
class					
cycle	12.0	NaN	10.0	NaN	15.0
pilates	9.0	10.0	NaN	14.0	NaN

Pandas – melt()

The top-level **melt**() function and the corresponding **DataFrame.melt**() are useful to massage a DataFrame into a format where one or more columns are identifier variables, while all other columns, considered measured variables, are "unpivoted" to the row axis, leaving just two non-identifier columns, "variable" and "value". The names of those columns can be customized by supplying the **var_name** and **value_name** parameters.

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

Pandas – melt() - example

Melt function

```
In [22]: df_melt = df4.melt(id_vars='class', value_vars='day')
         df_melt
```

Out[22]:

	class	variable	value
0	cycle	day	0-Mon
1	cycle	day	2-Wed
2	cycle	day	5-Sat
3	pilates	day	0-Mon
4	pilates	day	1-Tue
5	pilates	day	4-Fri