

Tutorial 10 - Pivot table

In [1]: `import pandas as pd`

In [2]: `df = pd.read_csv('sample_data_tutorial_10.csv')`
`df`

Out[2]:

	date	city	temperature	humidity
0	5/1/2017	new york	65	56
1	5/2/2017	new york	66	58
2	5/3/2017	new york	68	60
3	5/1/2017	mumbai	75	80
4	5/2/2017	mumbai	78	83
5	5/3/2017	mumbai	82	85
6	5/1/2017	beijing	80	26
7	5/2/2017	beijing	77	30
8	5/3/2017	beijing	79	35

In [3]: `# Vamos efetuar um pivot (index > rows; columns > colunas)`
`# Pivot and Pivot Table são comandos usados para agregar e resumir dados`
`df.pivot(index='date', columns='city')`

Out[3]:

	temperature			humidity		
city	beijing	mumbai	new york	beijing	mumbai	new york
date						
5/1/2017	80	75	65	26	80	56
5/2/2017	77	78	66	30	83	58
5/3/2017	79	82	68	35	85	60

In [4]: `# Imagine agora criar um pivot somente da temperature (usar a opção 'values')`
`df_temp = df.pivot(index='date', columns='city', values='temperature')`
`df_temp`

Out[4]:

city	beijing	mumbai	new york
date			
5/1/2017	80	75	65
5/2/2017	77	78	66
5/3/2017	79	82	68

```
In [5]: # Agora vamos carregar uma tabela que contem dados repetidos de um mesmo dia (manhã e tarde) e faremos a média do dia
df = pd.read_csv('sample_data_tutorial_10A.csv')
df
```

Out[5]:

	date	city	temperature	humidity
0	5/1/2017	new york	65	56
1	5/1/2017	new york	61	54
2	5/2/2017	new york	70	60
3	5/2/2017	new york	72	62
4	5/1/2017	mumbai	75	80
5	5/1/2017	mumbai	78	83
6	5/2/2017	mumbai	82	85
7	5/2/2017	mumbai	80	26

```
In [6]: df.pivot_table(index='city', columns='date')
```

Out[6]:

	humidity		temperature	
date	5/1/2017	5/2/2017	5/1/2017	5/2/2017
city				
mumbai	81.5	55.5	76.5	81.0
new york	55.0	61.0	63.0	71.0

```
In [7]: # Observe no caso anterior que a função agregada do 'pivot table' é média mas podem os efetuar a soma e outras funções:
# Google numpy mathematical functions or pandas pivot table
df.pivot_table(index='city', columns='date', aggfunc='sum')
```

Out[7]:

	humidity		temperature	
date	5/1/2017	5/2/2017	5/1/2017	5/2/2017
city				
mumbai	163	111	153	162
new york	110	122	126	142

In [8]: *# Usando a opção 'margins' ele realiza nas colunas e linhas a operação especificada:*
`df.pivot_table(index='city', columns='date', aggfunc='min', margins='True')`

Out[8]:

	humidity			temperature		
date	5/1/2017	5/2/2017	All	5/1/2017	5/2/2017	All
city						
mumbai	80	26	26	75	80	75
new york	54	60	54	61	70	61
All	54	26	26	61	70	61

In [9]: *# Agora vamos carregar uma tabela que contem dados somente de uma cidade mas em datas diferentes*
`df = pd.read_csv('sample_data_tutorial_10B.csv')`
`df`

Out[9]:

	date	city	temperature	humidity
0	5/1/2017	new york	65	56
1	5/2/2017	new york	61	54
2	5/3/2017	new york	70	60
3	12/1/2017	new york	30	50
4	12/2/2017	new york	28	52
5	12/3/2017	new york	25	51

In [10]: *# Vamos agora utilizar a função 'Grouper'*
Vemos na tabela anterior que a coluna 'date' é um string e que precisa ser transformada em 'date' para realizarmos
a operação de agrupar os dados por mês:
`df['date']=pd.to_datetime(df['date'])`
`df`

Out[10]:

	date	city	temperature	humidity
0	2017-05-01	new york	65	56
1	2017-05-02	new york	61	54
2	2017-05-03	new york	70	60
3	2017-12-01	new york	30	50
4	2017-12-02	new york	28	52
5	2017-12-03	new york	25	51

```
In [11]: df.pivot_table(index=pd.Grouper(freq='M', key='date'), columns='city')
```

Out[11]:

	humidity	temperature
city	new york	new york
date		
2017-05-31	56.666667	65.333333
2017-12-31	51.000000	27.666667