

Pivot basics

```
import pandas as pd
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
```

```
df = pd.read_csv("/content/drive/My Drive/pandas/pandas/10_pivot/weather.csv")
df
```

☞ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

	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

```
df.pivot(index='city', columns='date')
```

☞

	temperature			humidity		
date	5/1/2017	5/2/2017	5/3/2017	5/1/2017	5/2/2017	5/3/2017
city						
beijing	80	77	79	26	30	35
mumbai	75	78	82	80	83	85
new york	56	58	60	52	50	55

```
df.pivot(index='city',columns='date',values="humidity")
```

↳

date	5/1/2017	5/2/2017	5/3/2017
city			
beijing	26	30	35
mumbai	80	83	85
new york	56	58	60

```
df_val=df.pivot(index='city',columns='date',values="humidity")
```

```
df_val.index.name=None
```

```
df_val
```

↳

date	5/1/2017	5/2/2017	5/3/2017
beijing	26	30	35
mumbai	80	83	85
new york	56	58	60

```
df_val.columns.name = None
```

```
df_val
```

```
↳
```

	5/1/2017	5/2/2017	5/3/2017
beijing	26	30	35
mumbai	80	83	85
new york	56	58	60

```
df_val.loc['mumbai']
```

```
↳ 5/1/2017    80
   5/2/2017    83
   5/3/2017    85
   Name: mumbai, dtype: int64
```

```
df.pivot(index='date',columns='city')
```

```
↳
```

	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

```
df.pivot(index='humidity',columns='city')
```

```
↳
```

	date			temperature		
city	beijing	mumbai	new york	beijing	mumbai	new york
humidity						
26	5/1/2017	NaN	NaN	80.0	NaN	NaN
30	5/2/2017	NaN	NaN	77.0	NaN	NaN
35	5/3/2017	NaN	NaN	79.0	NaN	NaN
56	NaN	NaN	5/1/2017	NaN	NaN	65.0
58	NaN	NaN	5/2/2017	NaN	NaN	66.0
60	NaN	NaN	5/3/2017	NaN	NaN	68.0
80	NaN	5/1/2017	NaN	NaN	75.0	NaN
83	NaN	5/2/2017	NaN	NaN	78.0	NaN

Pivot Table

```
df = pd.read_csv("/content/drive/My Drive/pandas/pandas/10_pivot/weather2.csv")
df
```



	date	city	temperature	humidity
0	5/1/2017	new york	65	56

```
#df.pivot(index='city',columns='date',values="humidity")
```



```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-62-18b20adb1c3> in <module>()
----> 1 df.pivot(index='city',columns='date',values="humidity")
```



5 frames

```
/usr/local/lib/python3.6/dist-packages/pandas/core/reshape/reshape.py in _make_selectors(self)
    177
    178         if mask.sum() < len(self.index):
--> 179             raise ValueError("Index contains duplicate entries, cannot reshape")
    180
    181         self.group_index = comp_index
```

ValueError: Index contains duplicate entries, cannot reshape

```
df.pivot_table(index="city",columns="date")
```



	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

Margins

```
df.pivot_table(index="city",columns="date",aggfunc="sum")
```



	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

```
df.pivot_table(index="city",columns="date", margins=True,aggfunc=np.sum) # "sum","diff"
```

```
# see the all column - aggregate column and row
```



	humidity			temperature		
date	5/1/2017	5/2/2017	All	5/1/2017	5/2/2017	All
city						
mumbai	163	111	274	153	162	315
new york	110	122	232	126	142	268
All	273	233	506	279	304	583

Grouper

```
df = pd.read_csv("/content/drive/My Drive/pandas/pandas/10_pivot/weather3.csv")
df
```



	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

```
df['date'] = pd.to_datetime(df['date'])
```

```
df.pivot_table(index=pd.Grouper(freq='M',key='date'),columns='city') # average temp mon
# frequency monthly
```

↳

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

```
df = pd.DataFrame({"A": ["foo", "foo", "foo", "foo", "foo",
...                        "bar", "bar", "bar", "bar"],
...                "B": ["one", "one", "one", "two", "two",
...                        "one", "one", "two", "two"],
...                "C": ["small", "large", "large", "small",
...                        "small", "large", "small", "small",
...                        "large"]})
```

```
...
...
...
```

```
...
...
"D": [1, 2, 2, 3, 3, 4, 5, 6, 7],
"E": [2, 4, 5, 5, 6, 6, 8, 9, 9]})
```

df



	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9

```
table = pd.pivot_table(df, values='D', index=['A', 'B'],
...                      columns=['C'], aggfunc=np.sum)
```

table




```

        C large small
A    B
table = pd.pivot_table(df, values='D', index=['A', 'B'],
...                      columns=['C'], aggfunc=np.sum, fill_value=0)

```

```

foo one    4.0    1.0

```

table

```

↳      C large small

```

	A	B		
	bar	one	4	5
		two	7	6
foo	one		4	1
	two		0	6

```

table = pd.pivot_table(df, values=['D', 'E'], index=['A', 'C'],
...                      aggfunc={'D': np.mean,
...                                'E': np.mean})

```

table

```

↳

```

```

                D          E
table = pd.pivot_table(df, values=['D', 'E'], index=['A', 'C'],
...                     aggfunc={'D': np.mean,
...                               'E': [min, max, np.mean]})

```

```

    foo  large  2.000000  4.500000
table

```

↗

		D		E	
		mean	max	mean	min
A	C				
bar	large	5.500000	9.0	7.500000	6.0
	small	5.500000	9.0	8.500000	8.0
foo	large	2.000000	5.0	4.500000	4.0
	small	2.333333	6.0	4.333333	2.0

```

import pandas as pd
import numpy as np

```

```

d = pd.DataFrame({
    "Item": ['Item0', 'Item0', 'Item1', 'Item1'],
    "CType": ['Gold', 'Bronze', 'Gold', 'Silver'],
    "USD": ['1$', '2$', '3$', '4$'],
    "EU": ['1€', '2€', '3€', '4€']
})

```

d



	Item	CType	USD	EU
0	Item0	Gold	1\$	1€
1	Item0	Bronze	2\$	2€
2	Item1	Gold	3\$	3€
3	Item1	Silver	4\$	4€

<u>ix</u>	<u>Item</u>	<u>CType</u>	<u>USD</u>	<u>EU</u>
<u>0</u>	Item0	Gold	1\$	1€
<u>1</u>	Item0	Bronze	2\$	2€
<u>2</u>	Item1	Gold	3\$	3€
<u>3</u>	Item1	Silver	4\$	4€

<u>ix=Item</u>	<u>Bronze</u>	<u>Gold</u>	<u>Silver</u>
<u>Item0</u>	2\$	1\$	NaN
<u>Item1</u>	NaN	3\$	4\$

```
d.pivot(index='Item', columns='CType', values='USD')
```

```
p = d.pivot(index='Item', columns='CType', values='USD')
```

p

```
↳
```

CType	Bronze	Gold	Silver
Item			
Item0	2\$	1\$	NaN
Item1	NaN	3\$	4\$

```
# Original DataFrame: Access the USD cost of Item0 for Gold customers
print (d[(d.Item=='Item0') & (d.CType=='Gold')].USD.values)
#print (d[(d.Item=='Item0') & (d.CType=='Gold')]['USD'])
```

```
↳ ['1$']
```

```
# Pivoted DataFrame: Access the USD cost of Item0 for Gold customers
print (p[p.index=='Item0'].Gold.values)
```

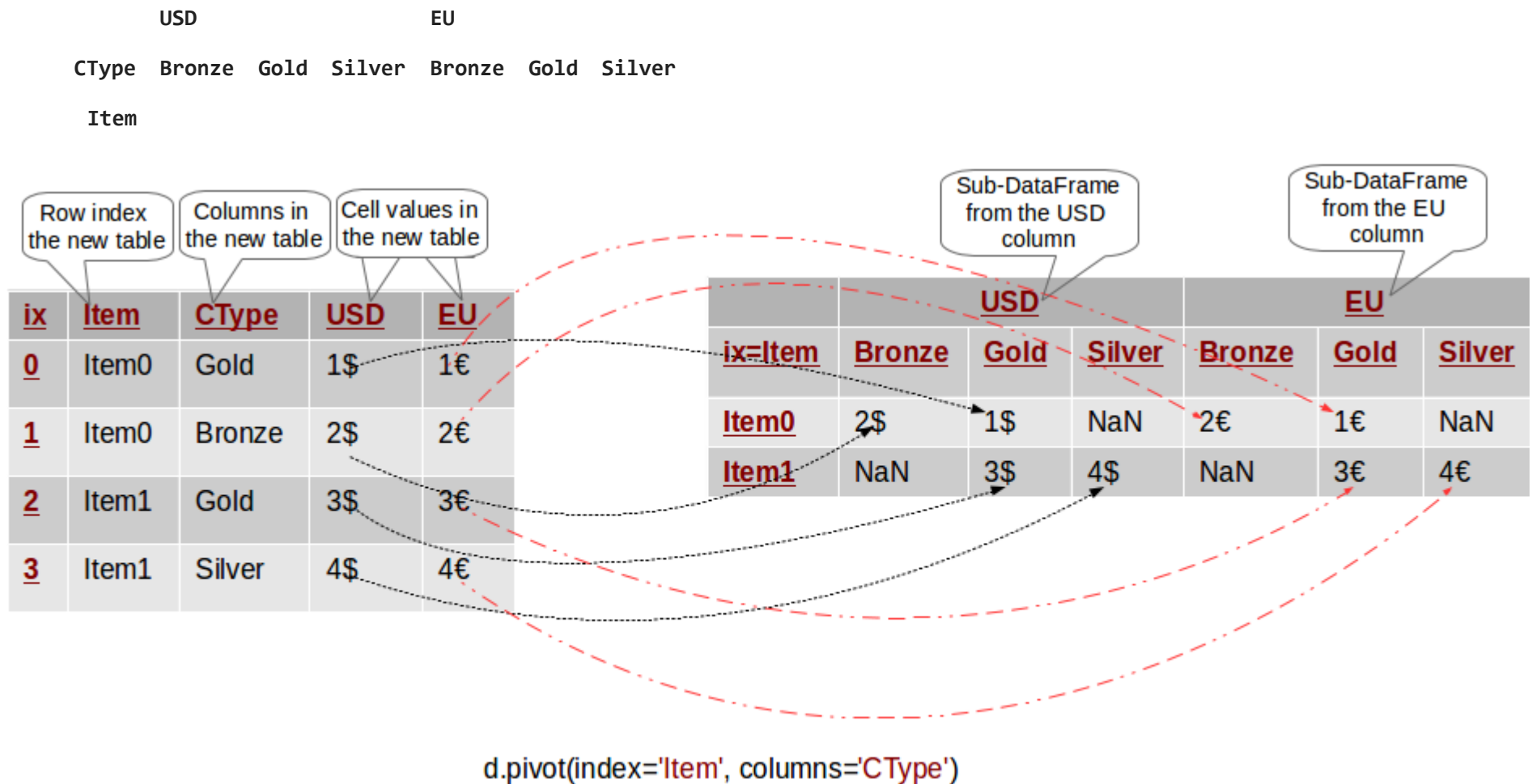
```
↳ ['1$']
```

▼ Pivoting By Multiple Columns

```
p = d.pivot(index='Item', columns='CType')
```

```
p
```

```
↳
```



```
# Original DataFrame: Access the USD cost of Item0 for Gold customers
print(d[(d.Item=='Item0') & (d.CType=='Gold')].USD.values)
```

```
↳ ['1$']
```

```
# Pivoted DataFrame: p.USD gives a "sub-DataFrame" with the USD values only
```

```
print(p.USD[p.USD.index=='Item0'].Gold.values)
```

```
↳ ['1$']
```

▼ Common Mistake in Pivoting

ix	Item	CType	USD	EU
0	Item0	Gold	1\$	1€
1	Item0	Bronze	2\$	2€
2	Item0	Gold	3\$	3€
3	Item1	Silver	4\$	4€

ix=Item	Bronze	Gold	Silver
Item0	2\$	1 or 3\$?	NaN
Item1	NaN	NaN	4\$

```
d.pivot(index='Item', columns='CType', values='USD')
```

```
d = pd.DataFrame({
    "Item": ['Item0', 'Item0', 'Item0', 'Item1'],
    "CType": ['Gold', 'Bronze', 'Gold', 'Silver'],
    "USD": ['1$', '2$', '3$', '4$'],
    "EU": ['1€', '2€', '3€', '4€']
})
```

```
p = d.pivot(index='Item', columns='CType', values='USD')
```



```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-76-415069b2054a> in <module>()
----> 1 p = d.pivot(index='Item', columns='CType', values='USD')
```

5 frames

```
/usr/local/lib/python3.6/dist-packages/pandas/core/reshape/reshape.py in _make_selectors(self)
177
178     if mask.sum() < len(self.index):
--> 179         raise ValueError("Index contains duplicate entries, cannot reshape")
180
181     self.group_index = comp_index
```

ValueError: Index contains duplicate entries, cannot reshape

SEARCH STACK OVERFLOW

▼ Pivot Table

<u>ix</u>	<u>Item</u>	<u>CType</u>	<u>USD</u>	<u>EU</u>	
<u>0</u>	Item0	Gold	1	1	
<u>1</u>	Item0	Bronze	2	2	
<u>2</u>	Item0	Gold	3	3	
<u>3</u>	Item1	Silver	4	4	

<u>ix=Item</u>	<u>Bronze</u>	<u>Gold</u>	<u>Silver</u>
<u>Item0</u>	2	2 = mean(1,3)	NaN
<u>Item1</u>	NaN	NaN	4

```
d.pivot_table(index='Item', columns='CType', values='USD', aggfunc=np.mean)
```

```
d = pd.DataFrame({
    "Item": ["Item0", "Item0", "Item0", "Item1"],
    "CType": ["Gold", "Bronze", "Gold", "Silver"],
    "USD": [1, 2, 3, 4],
    "EU": [1, 2, 3, 4]
})
```

```
item : [ item0 , item0 , item0 , item1 ],  
"CType":['Gold', 'Bronze', 'Gold', 'Silver'],  
"USD":  [1, 2, 3, 4],  
"EU":  [1.1,2.2,3.3,4.4]  
})
```

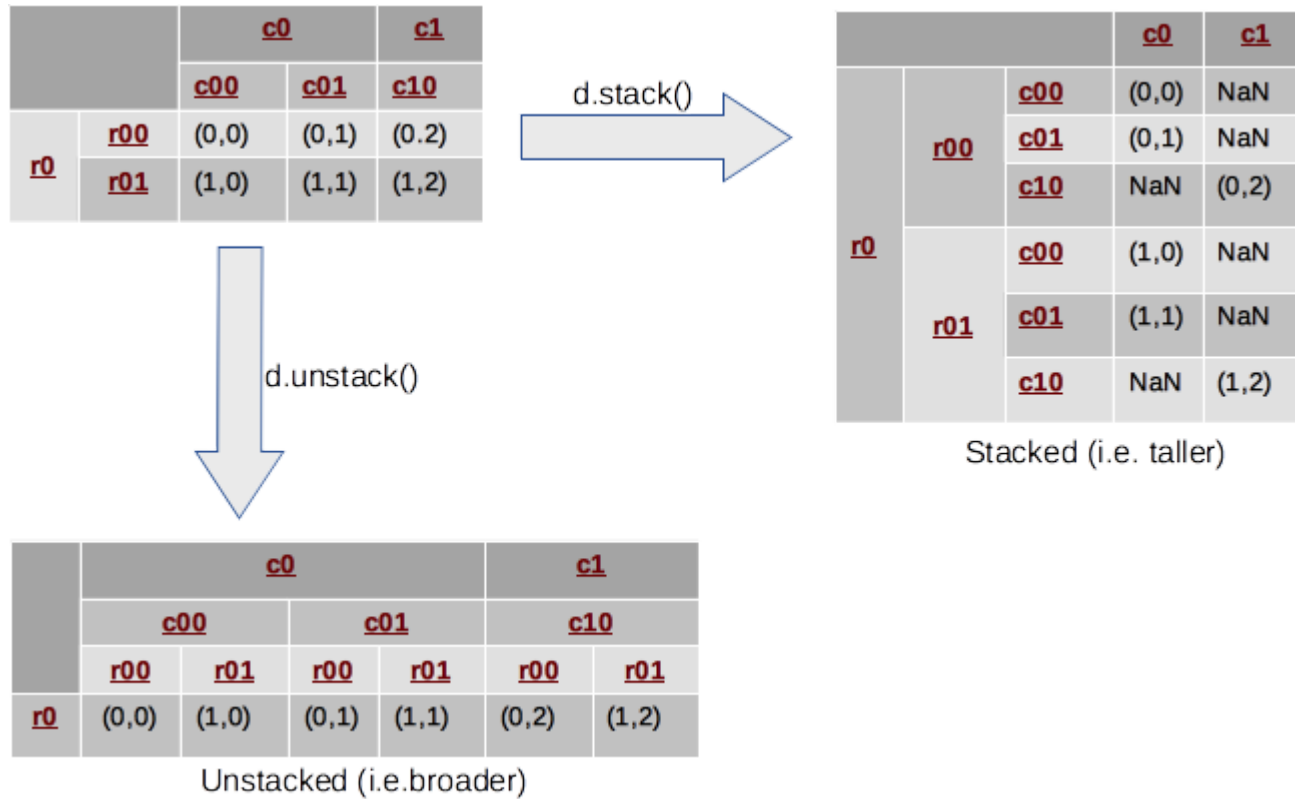
```
p = d.pivot_table(index='Item', columns='CType', values='USD', aggfunc=np.min)
```

p

```
↳
```

	CType	Bronze	Gold	Silver
Item				
Item0		2.0	1.0	NaN
Item1		NaN	NaN	4.0

▼ Stack/Unstack



Row Multi-Index

```
row_idx_arr = list(zip(['r0', 'r0'], ['r-00', 'r-01']))
```

```
row_idx = pd.MultiIndex.from_tuples(row_idx_arr)
```

Column Multi-Index

```
col_idx_arr = list(zip(['c0', 'c0', 'c1'], ['c-00', 'c-01', 'c-10']))
```

```
col_idx = pd.MultiIndex.from_tuples(col_idx_arr)
```

```
from pandas import DataFrame
# Create the DataFrame
d = DataFrame(np.arange(6).reshape(2,3), index=row_idx, columns=col_idx)
d = d.applymap(lambda x: (x // 3, x % 3))
d
```



```
# Stack/Unstack
s = d.stack()
s
```



```
u = d.unstack()
```

```
u
```

