

Pandas Pivot tables row subtotals

I'm using Pandas 0.10.1

37

Considering this Dataframe:

Date	State	City	SalesToday	SalesMTD	SalesYTD
20130320	stA	ctA	20	400	1000
20130320	stA	ctB	30	500	1100
20130320	stB	ctC	10	500	900
20130320	stB	ctD	40	200	1300
20130320	stC	ctF	30	300	800

How can i group subtotals per state?

State	City	SalesToday	SalesMTD	SalesYTD
stA	ALL	50	900	2100
stA	ctA	20	400	1000
stA	ctB	30	500	1100

I tried with a pivot table but i only can have subtotals in columns

```
table = pivot_table(df, values=['SalesToday', 'SalesMTD', 'SalesYTD'],\
                    rows=['State', 'City'], aggfunc=np.sum, margins=True)
```

I can achieve this on excel, with a pivot table.

python

pandas

pivot-table

asked Mar 22 '13 at 12:16



balsagoth

1,159 3 14 29

4 Answers

If you put State and City not both in the rows, you'll get separate margins. Reshape and you get the table you're after:

44

```
In [10]: table = pivot_table(df, values=['SalesToday', 'SalesMTD', 'SalesYTD'],\
                             rows=['State'], cols=['City'], aggfunc=np.sum, margins=True)
```

```
In [11]: table.stack('City')
Out[11]:
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

stB	All	700	50	2200
	ctC	500	10	900
	ctD	200	40	1300
stC	All	300	30	800
	ctF	300	30	800
All	All	1900	130	5100
	ctA	400	20	1000
	ctB	500	30	1100
	ctC	500	10	900
	ctD	200	40	1300
	ctF	300	30	800

I admit this isn't totally obvious.

answered Mar 22 '13 at 16:07



Wes McKinney

59.9k 21 116 94

-
- 3 this works if we have values=, if columns are created from columns=... there'll be only one "All" column. – Winand May 29 '15 at 10:31
-

▲ You can get the summarized values by using `groupby()` on the State column.

20

Let's make some sample data first:

```
import pandas as pd
import StringIO

incsv = StringIO.StringIO("""Date,State,City,SalesToday,SalesMTD,SalesYTD
20130320,stA,ctA,20,400,1000
20130320,stA,ctB,30,500,1100
20130320,stB,ctC,10,500,900
20130320,stB,ctD,40,200,1300
20130320,stC,ctF,30,300,800""")

df = pd.read_csv(incsv, index_col=['Date'], parse_dates=True)
```

Then apply the `groupby` function and add a column City:

```
dfsum = df.groupby('State', as_index=False).sum()
dfsum['City'] = 'All'

print dfsum
```

	State	SalesToday	SalesMTD	SalesYTD	City
0	stA	50	900	2100	All
1	stB	50	700	2200	All
2	stC	30	300	800	All

We can append the original data to the summed df by using `append`:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

		SalesMTD	SalesToday	SalesYTD
State	City			
stA	All	900	50	2100
	ctA	400	20	1000
	ctB	500	30	1100
stB	All	700	50	2200
	ctC	500	10	900
	ctD	200	40	1300
stC	All	300	30	800
	ctF	300	30	800

I added the `set_index` and `sort_index` to make it look more like your example output, its not strictly necessary to get the results.

answered Mar 22 '13 at 12:38



Rutger Kassies

31.9k 9 80 80

Another alternative function is `concat()` which allows me to have more control on index by setting parameter `ignore_index`. `pd.concat([x, y], ignore_index=True)`. [This article](#) also mentions that `concat()` is more efficient comparing with `pandas.append()`. – Michael Revlis Sep 3 '18 at 3:25

I Think this subtotal example code is what you want(similar to excel subtotal)

6

I assume that you want group by columns A, B, C, D, than count column value of E

```
main_df.groupby(['A', 'B', 'C']).apply(lambda sub_df: sub_df\
    .pivot_table(index=['D'], values=['E'], aggfunc='count', margins=True))
```

output:

```
A B C D E
    a 1
a a a b 2
    c 2
    all 5
    a 3
b b a b 2
    c 2
    all 7
    a 3
b b b b 6
    c 2
    d 3
    all 14
```

edited Apr 20 '17 at 21:49



Parfait

58.7k 10 55 75

answered Apr 20 '17 at 20:18



hs moon

61 1 2

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

@Parfait in your example is there a way to include a grand total? – Clickinaway Nov 12 '18 at 16:54

- 1 @Clickinaway ... this is not my answer, I only helped edit. But depending on [pivot_table](#) specification you can get row subtotals and column grand totals. – Parfait Nov 12 '18 at 17:16

@Parfait I'm currently failing at doing this and have been looking at other SE posts all morning. My thread is here stackoverflow.com/questions/53266032/... – Clickinaway Nov 12 '18 at 17:20

How about this one ?

1

```
table = pd.pivot_table(data, index=['State'], columns = ['City'], values=['SalesToday',  
'SalesMTD', 'SalesYTD'], \n                        aggfunc=np.sum, margins=True)
```

 [enter image description here](#)

answered May 13 '16 at 5:43



[Richard Mao](#)

124 1 9