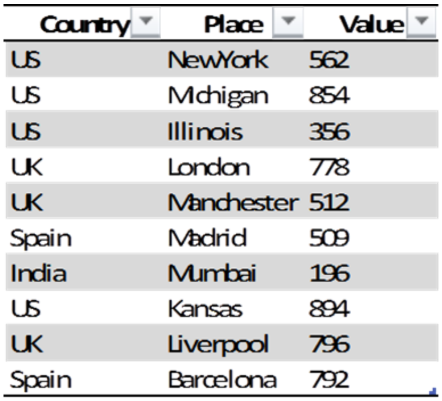
Find maximum value of a column and return the corresponding row values using Pandas

Ask Question

67

25



Country	Place	Value
US	NewYork	562
US	Michigan	854
US	Illinois	356
UK	London	778
UK	Manchester	512
Spain	Madrid	509
India	Mumbai	196
US	Kansas	894
UK	Liverpool	796
Spain	Barcelona	792

Using Python Pandas I am trying to find the 'Country' & 'Place' with the maximum value.

This returns the maximum value:

```
data.groupby(['Country','Place'])['Value'].max()
```

But how do I get the corresponding 'Country' and 'Place' name?

python pandas

edited Apr 1 '13 at 10:38

jamylak
85.3k 18 181 199

asked Apr 1 '13 at 10:31

richie
5,435 8 35 56

105

Assuming `df` has a unique index, this gives the row with the maximum value:

```
In [34]: df.loc[df['Value'].idxmax
Out[34]:
Country        US
Place      Kansas
Value         894
Name: 7
```

Note that `idxmax` returns index *labels*. So if the DataFrame has duplicates in the index, the label may not uniquely identify the row, so `df.loc` may return more than one row.

Therefore, if `df` does not have a unique index, you must make the index unique before proceeding as above. Depending on the DataFrame, sometimes you can use `stack` or `set_index` to make the index unique. Or, you can simply reset the index (so the rows become renumbered, starting at 0):

```
df = df.reset_index()
```

edited May 11 '18 at 21:01

Joseph Dasenbrock
127 2 9

answered Apr 1 '13 at 10:58

unutbu
561k 105 1209
1262

Thank you. That was exactly what I was looking for. – richie Apr 1 '13 at 11:03

```
df[df['Value']==df['Value'].max()]
```

edited May 5 '18 at 17:58

eyllanesc
85.5k 10 35 64

answered Apr 30 '18 at 17:07

Gaurav
311 4 3

Explanation :- The inner expression does a boolean check throughout the length of the dataFrame & that index which satisfies the right hand side of the expression(.max()) returns the index, which in turn calls the complete row of that dataFrame – penta Feb 22 at 5:28

8

The country and place is the index of the series, if you don't need the index, you can set `as_index=False` :

```
df.groupby(['country','place'], as_
```

Edit:

It seems that you want the place with max value for every country, following code will do what you want:

```
df.groupby("country").apply(lambda
```

edited Apr 1 '13 at 11:04

answered Apr 1 '13 at 10:50

HYRY
66.1k 13 126 148

that would only return the column names and the dtypes – richie Apr 1 '13 at 10:54

5

Use the `index` attribute of `DataFrame`. Note that I don't type all the rows in the example.

```
In [15]: df.index
Out[15]:
MultiIndex
[Spain  Manchester, UK     London

In [16]: df.index[0]
Out[16]: ('Spain', 'Manchester')

In [17]: df.index[1]
Out[17]: ('UK', 'London')
```

You can also get the value by that index:

```
In [21]: for index in df.index:
    print index, df[index]
   ....:
('Spain', 'Manchester') 512
('UK', 'London') 778
('US', 'Mchigan') 854
('US', 'NewYork') 562
```

Edit

Sorry for misunderstanding what you want, try followings:

```
In [52]: s=data.max()

In [53]: print '%s, %s, %s' % (s['C
US, NewYork, 854
```

edited Apr 1 '13 at 11:02

answered Apr 1 '13 at 10:44

waitingkuo
38.2k 17 87 100

correct. But I'm looking for a one line output that says, 'US, Kansas, 894' – richie Apr 1 '13 at 10:51

Sorry for misunderstanding, have updated :> – waitingkuo Apr 1 '13 at 11:11

Thanks. This would solve the problem for the current dataset where there is just 1 column with values. When there are more columns with values @unutbu's solution would work better. Thanks anyway. – richie Apr 1 '13 at 11:19

4

Define your DataFrame object, say **df**, and read the file.

In order to print the Country and Place with maximum value, use the following line of code.

```
print(df[['Country', 'Place']][
```

edited Feb 20 '18 at 7:08

Pang
6,988 16 66 105

answered Feb 20 '18 at 6:53

Arpit Sharma
85 12

4

I think the easiest way to return a row with the maximum value is by getting its index. `argmax()` can be used to return the index of the row with the largest value.

```
index = df.Value.argmax()
```

Now the index could be used to get the features for that particular row:

```
df.iloc[df.Value.argmax(), 0:2]
```

edited Jan 29 at 21:32

lenz
3,124 4 18 32

answered May 9 '18 at 10:48

sharad kakran
41 1

0

My solution for finding maximum values in columns:

```
df.ix[df.idxmax()]
```

, also minimum:

```
df.ix[df.idxmin()]
```

Marcin Lentner
33 5