Pandas rank vs transform('rank')

Ask Question

2

1

I'm not sure if this is a bug or a feature, but I really want to understand how it works.

I have a very simple dataset

```
In [0]: data
Out[0]:
  group  value  data
0     A      1     1
1     A      2     1
2     B      3     1
3     B      4     1
```

And then I have some transformations:

```
In [1]: data.groupby('group').transform('rank')
Out[1]:
   value  data
0    1.0   1.5
1    2.0   1.5
2    1.0   1.5
3    2.0   1.5

In [2]: data.groupby('group').value.transform('rank')
Out[2]:
0    1
1    1
2    2
3    2

In [3]: data.groupby('group').data.transform('rank')
Out[3]:
0    1.5
1    1.5
2    1.5
3    1.5

In [4]: data.groupby('group').transform('rank').value
Out[4]:
0    1.0
1    2.0
2    1.0
3    2.0

In [5]: data.groupby('group').value.rank()
Out[5]:
0    1.0
1    2.0
2    1.0
3    2.0

In [6]: data.groupby('group').cumcount()
```

```
2    0
3    1
```

The things I find strange:

- The 1st one. While I seem to understand what happened to `value` column (similar to the 5th and 6th ones) I can't understand what happened to `data` column. Where did the value `1.5` come from?
- The 2nd one. Instead of selecting only specified column and applying logic similar to the 1st one it completely changed the output. I can presume that now in enumerates groups instead of rows inside of groups, but I still don't understand why it's done in such a strange way?
- The 3rd one. It behaves exactly as expected in terms that it just selected specified column and applied the same logic from the 1st one (in contrast to previous one). But I'm still missing the origin of this value;
- The 4th one. Shouldn't this one be the same as the 2nd one?
- The 5th and 6th ones looks exactly the same, but the latter one starts from 0. Is it correct?

I would very appreciate if someone could explain it to me.

Thanks.

python pandas

asked Nov 4 '16 at 11:45

Viktor Ershov
61 7

1 Answer

1

Let me add bit more confusion - rank() method has a `method` parameter ...

default: `method='average'`

```
In [70]: data.groupby('group').tra
Out[70]:
   value  data
0    1.0   1.5
```

```
In [71]: data.groupby('group').tra
Out[71]:
   value  data
0    1.0   1.5
1    2.0   1.5
2    1.0   1.5
3    2.0   1.5
```

method: `min`

```
In [72]: data.groupby('group').tra
Out[72]:
   value  data
0    1.0   1.0
1    2.0   1.0
2    1.0   1.0
3    2.0   1.0
```

method: `max`

```
In [73]: data.groupby('group').tra
Out[73]:
   value  data
0    1.0   2.0
1    2.0   2.0
2    1.0   2.0
3    2.0   2.0
```

method: `first`

```
In [74]: data.groupby('group').tra
Out[74]:
   value  data
0    1.0   1.0
1    2.0   2.0
2    1.0   1.0
3    2.0   2.0
```

method: `dense`

```
In [75]: data.groupby('group').tra
Out[75]:
   value  data
0    1.0   1.0
1    2.0   1.0
2    1.0   1.0
3    2.0   1.0
```

from docs:

> **method** : { `average` , `min` , `max` , `first` , `dense` }
>
> average: average rank of group
>
> min: lowest rank in group
>
> max: highest rank in group
>
> first: ranks assigned in order they appear in the array
>
> dense: like 'min', but rank always increases by 1 between groups

and there is yet another parameter:

> Computes percentage rank of data

answered Nov 4 '16 at 12:41

MaxU
125k 13 128 183

That definitely makes things even more confusing to me. But I still don't understand two things: 1) How `1.5` is calculated for `data` column? How can be average for 1 and 1 be 1.5? 2) Why it seems like a different logic as applied to `value` and `data` columns? – Viktor Ershov Nov 4 '16 at 15:15