

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

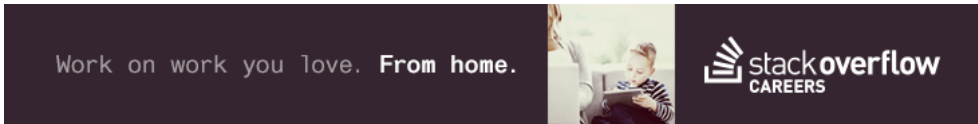
Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

## How can I round values in Pandas DataFrame containing mixed datatypes for further data comparison?



I have a dataframe df\_left:

	IDX1	IDX2	IDX3	IDX4	ValueType	Value
0	A	A1	Q	1983 Q4	W	10.123
1	A	A1	Q	1983 Q4	X	A
2	A	A1	Q	1983 Q4	Y	F
3	A	A1	Q	1983 Q4	Z	NaN
4	A	A1	Q	1984 Q1	W	110.456
...						

created from a previous post:

[Background information](#)

AND dataframe df\_right:

	IDX1	IDX2	IDX3	IDX4	ValueType	Value
0	A	A1	Q	1983 Q4	W	10
1	A	A1	Q	1983 Q4	X	A
2	A	A1	Q	1983 Q4	Y	F
3	A	A1	Q	1983 Q4	Z	NaN
4	A	A1	Q	1984 Q1	W	110

I compare and reconcile the data both values and text of which the following works:

```
df_compare = pd.merge(df_Left, df_Right, how='outer', on = ['IDX1', 'IDX2', 'IDX3', 'IDX4', 'ValueType'])
df_compare.columns = ['IDX1', 'IDX2', 'IDX3', 'IDX4', 'ValueType', 'From', 'To']
df_compare = df_compare[df_compare.From!=df_compare.To]
```

Whilst the results are as expected, before the comparison I would like to round the data in the value coulmn.

I have tried:

```
df.apply(np.round)
```

and also:

```
df.round(decimals=0, out=None)
```

but both as expected thow an error:


```
AttributeError: ("str' object has no attribute 'rint'", u'occurred at index Code')
```

python

numpy

pandas

asked May 15 '15 at 11:15

 toasteez

238 14

### 2 Answers

A custom method for rounding just the floats may solve rounding a mixed dtype column

```
In [238]: def round_float(s):
          ''1. if s is float, round it to 0 decimals
          2. else return s as is
```

```

...
import re
m = re.match("(\\d+\\.\\d+)", s.__str__())
try:
    r = round(float(m.groups(0)[0]),0)
except:
    r = s
return r

```

```

In [239]: s = u'''  IDX1 IDX2 IDX3      IDX4 ValueType Value
0  A  A1  Q  1983 Q4      W  10.23
1  A  A1  Q  1983 Q4      X  A
2  A  A1  Q  1983 Q4      Y  F
3  A  A1  Q  1983 Q4      Z  NaN
4  A  A1  Q  1984 Q1      W  110.15'''

```

```

In [240]: df1 = pd.read_csv(StringIO(s), delimiter="\s+")

```

```

In [241]: df1["Value2"] = df1.Value.apply(round_float)

```

```

In [242]: df1

```

```

Out[242]:
  IDX1 IDX2  IDX3  IDX4  ValueType  Value  Value2
0  A  A1  Q  1983  Q4      W  10.23    10
1  A  A1  Q  1983  Q4      X  A      A
2  A  A1  Q  1983  Q4      Y  F      F
3  A  A1  Q  1983  Q4      Z  NaN    NaN
4  A  A1  Q  1984  Q1      W  110.15   110

```

answered May 15 '15 at 19:05



UNagaswamy

309 8 21

USE STACK OVERFLOW TO  
FIND THE BEST DEVELOPERS



Here's a fairly general solution you can apply to multiple columns. The 'To' column doesn't need to be rounded, I just included it for the generality of two columns rather than one:

```
df
```

```

  IDX1  IDX2  IDX3  IDX4  ValueType  From  To
0  A1  Q  1983  Q4      W  10.123  10
3  A1  Q  1983  Q4      Z  NaN  NaN
4  A1  Q  1984  Q1      W  110.456  110

```

```

In [399]: df[['From','To']].astype(float).apply(np.round)

```

```

  From  To
0   10   10
3  NaN NaN
4  110  110

```

That's the safest way in that it won't let you accidentally wipe out non-numeric values, but if you have truly mixed types in there, you can do this:

```
df[['From','To']].convert_objects(convert_numeric=True).apply(np.round)
```

```

  From  To
0   10   10
3  NaN NaN
4  110  110

```

But since this will convert any non-numeric values to NaN, just make sure that's what you want before you over-write anything.

edited May 15 '15 at 12:57

answered May 15 '15 at 12:36



JohnE

4,862 1 7 32

Thanks. I still seem to be getting decimals df\_compare[['From','To']].astype(float).apply(np.round)  
df\_compare['Result'] = pd.Series(df\_compare.From.astype(float).apply(np.round)  
df\_compare.To.astype(float).apply(np.round), index=df\_compare.index) df\_compare =  
df\_compare.query('Result != 0') – toasteez May 15 '15 at 13:31

Can you show the data/values for which this happens? I just used the data/code you provided and didn't see this. Also, you may want to add this to the question rather than as a comment so it's easier to read. – JohnE May 15 '15 at 13:40

I can't quite tell what you are doing in the comment, but you don't need the Series, you can just store result like this: df\_compare['Result'] = df\_compare.From.astype(float).apply(np.round) – JohnE May 15 '15 at 13:46