Pandas

January 1, 2020

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
pd.__version__ #  Pandas
```

[2]:
```
'0.25.3'
```

[3]:
```
ser = pd.Series([1, 3, 5.1, np.nan, ' '], index = list(range(1, 6)))
ser
```

[3]:
```
1      1
2      3
3    5.1
4    NaN
5
dtype: object
```

[4]:
```
#  2    []
ser[2]
```

[4]:
```
3
```

[5]:
```
#  2    []
ser[2] = 'Jack'
ser
```

[5]:
```
1       1
2    Jack
3     5.1
4     NaN
5
dtype: object
```

[7]:
```
#  1 3 5   []
ser[[1, 3, 5]]
```

[7]:
```
1      1
3    5.1
5
dtype: object
```

[8]:
```
print(ser.values)
type(ser.values) # Series
```

```
[1 'Jack' 5.1 nan ' ']
```

[8]:
```
numpy.ndarray
```

[10]:
```
#   Series    1
ser2 = pd.Series([18, 19, 17], index = range(1, 4))
ser2 + 1 #    1 ser2
```

[10]:
```
1    19
2    20
3    18
dtype: int64
```

[11]:
```
# ser2
ser2
```

[11]:
```
1    18
2    19
3    17
dtype: int64
```

[14]:
```
ser = pd.Series([1, 3, 5.1, 6, 9], index = list(range(1, 6)))
ser[ser % 2 == 1] #   []
```

[14]:
```
1    1.0
2    3.0
5    9.0
dtype: float64
```

[16]:
```
#    Series
# beijing shanghai guangzhou   9240 8960 7400
dic = {'beijing':9240, 'shanghai':8960, 'guangzhou': 7400}
ser3 = pd.Series(dic)
ser3
```

[16]:
```
beijing      9240
shanghai     8960
guangzhou    7400
dtype: int64
```

[17]:
```
ser3[['beijing','guangzhou']]
```

[17]:
```
beijing      9240
guangzhou    7400
dtype: int64
```

[20]:
```
print('beijing' in ser3)
print(ser3.to_dict()) #
print(ser3.tolist()) #
print(ser3.to_json()) # json
```

```
True
{'beijing': 9240, 'shanghai': 8960, 'guangzhou': 7400}
[9240, 8960, 7400]
{"beijing":9240,"shanghai":8960,"guangzhou":7400}
```

[21]:
```
ser3.to_frame() # DataFrame
```

[21]:

	0
beijing	9240
shanghai	8960
guangzhou	7400

1 DataFrame

1.0.1 NumPy DataFrame

[23]:
```
datas = pd.date_range('20190101', periods=6)
datas
```

[23]:
```
DatetimeIndex(['2019-01-01', '2019-01-02', '2019-01-03', '2019-01-04',
               '2019-01-05', '2019-01-06'],
              dtype='datetime64[ns]', freq='D')
```

[25]:
```
df = pd.DataFrame(np.random.randn(6, 4), index = datas, columns = list('ABCD'))
df
```

[25]:

	A	B	C	D
2019-01-01	0.732546	-0.611456	0.672733	0.325307
2019-01-02	-0.855926	-0.781192	-0.957641	0.384169
2019-01-03	-1.863531	0.039663	-0.486457	-2.928965
2019-01-04	2.583917	-0.621555	2.694782	0.577375
2019-01-05	0.281635	-0.064713	1.057266	-0.252323
2019-01-06	-0.668788	3.076215	-0.503385	-0.638602

1.0.2 Series DataFrame

[26]:
```
df2 = pd.DataFrame({'A': 1.,
                    'B': pd.Timestamp('20130102'),
                    'C': pd.Series(1, index=list(range(4)), dtype='float32'),
                    'D': np.array([3] * 4, dtype='int32'),
                    'E': pd.Categorical(["test", "train", "test", "train"]),
                    'F': 'foo'})
df2
```

[26]:

	A	B	C	D	E	F
0	1.0	2013-01-02	1.0	3	test	foo
1	1.0	2013-01-02	1.0	3	train	foo
2	1.0	2013-01-02	1.0	3	test	foo
3	1.0	2013-01-02	1.0	3	train	foo

1.0.3 DataFrame

[27]:
```
df2.dtypes
```

[27]:
```
A           float64
B    datetime64[ns]
C           float32
D             int32
E          category
F            object
dtype: object
```

1.0.4 IPython tab

1.1

[29]:
```
df.head()
```

[29]:

	A	B	C	D
2019-01-01	0.732546	-0.611456	0.672733	0.325307
2019-01-02	-0.855926	-0.781192	-0.957641	0.384169
2019-01-03	-1.863531	0.039663	-0.486457	-2.928965
2019-01-04	2.583917	-0.621555	2.694782	0.577375
2019-01-05	0.281635	-0.064713	1.057266	-0.252323

[35]:
```
df.tail(3)
```

[35]:

	A	B	C	D
2019-01-04	2.583917	-0.621555	2.694782	0.577375
2019-01-05	0.281635	-0.064713	1.057266	-0.252323
2019-01-06	-0.668788	3.076215	-0.503385	-0.638602

[32]:
```
df.index
```

[32]:
```
DatetimeIndex(['2019-01-01', '2019-01-02', '2019-01-03', '2019-01-04',
               '2019-01-05', '2019-01-06'],
              dtype='datetime64[ns]', freq='D')
```

[34]:
```
df.columns
```

[34]:
```
Index(['A', 'B', 'C', 'D'], dtype='object')
```

1.1.1 DataFrame.to_numpy() NumPy

- NumPy DataFrame
- DataFrame Pandas NumPy
- Pandas DataFrame NumPy object DataFrame Python
- DataFrame.to_numpy()

df DataFrame DataFrame.to_numpy()

[5]:
```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(6, 4), index = pd.date_range('20190101',␣
↪periods=6), columns = list('ABCD'))
df.to_numpy()
```

[5]:
```
array([[-1.71499051,  0.94256846,  0.538325  , -0.66895854],
       [-0.54364367, -1.42667186,  0.7317733 ,  1.40133949],
       [ 3.61197637, -0.56630825,  0.81489105, -0.29826305],
       [-1.56784327, -1.08047619,  0.09456628,  0.15996781],
       [-0.26885637,  0.88649103, -0.16920089, -0.29695971],
       [ 0.86889926,  0.9945544 , -0.14170551, -1.92196584]])
```

df2 DataFrame DataFrame.to_numpy()

[6]:
```
df2 = pd.DataFrame({'A': 1.,
                    'B': pd.Timestamp('20130102'),
                    'C': pd.Series(1, index=list(range(4)), dtype='float32'),
                    'D': np.array([3] * 4, dtype='int32'),
                    'E': pd.Categorical(["test", "train", "test", "train"]),
                    'F': 'foo'})
df2.to_numpy()
```

[6]:
```
array([[1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'test', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'train', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'test', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'train', 'foo']],
      dtype=object)
```

describe()

[7]:
```
df.describe()
```

[7]:
```
              A         B         C         D
count  6.000000  6.000000  6.000000  6.000000
mean   0.064257 -0.041640  0.311442 -0.270807
std    1.977056  1.111446  0.439320  1.083322
min   -1.714991 -1.426672 -0.169201 -1.921966
25%   -1.311793 -0.951934 -0.082638 -0.576285
50%   -0.406250  0.160091  0.316446 -0.297611
75%    0.584460  0.928549  0.683411  0.045736
max    3.611976  0.994554  0.814891  1.401339
```

[8]:
```
df.T
```

[8]:
```
   2019-01-01  2019-01-02  2019-01-03  2019-01-04  2019-01-05  2019-01-06
A   -1.714991   -0.543644    3.611976   -1.567843   -0.268856    0.868899
B    0.942568   -1.426672   -0.566308   -1.080476    0.886491    0.994554
C    0.538325    0.731773    0.814891    0.094566   -0.169201   -0.141706
D   -0.668959    1.401339   -0.298263    0.159968   -0.296960   -1.921966
```

[9]:
```
df.sort_index(axis=1, ascending=False)
```

[9]:
```
                   D         C         B         A
2019-01-01 -0.668959  0.538325  0.942568 -1.714991
2019-01-02  1.401339  0.731773 -1.426672 -0.543644
2019-01-03 -0.298263  0.814891 -0.566308  3.611976
2019-01-04  0.159968  0.094566 -1.080476 -1.567843
2019-01-05 -0.296960 -0.169201  0.886491 -0.268856
2019-01-06 -1.921966 -0.141706  0.994554  0.868899
```

[11]:
```
df.sort_values(by='B')
```

[11]:
```
                   A         B         C         D
2019-01-02 -0.543644 -1.426672  0.731773  1.401339
2019-01-04 -1.567843 -1.080476  0.094566  0.159968
2019-01-03  3.611976 -0.566308  0.814891 -0.298263
2019-01-05 -0.268856  0.886491 -0.169201 -0.296960
2019-01-01 -1.714991  0.942568  0.538325 -0.668959
2019-01-06  0.868899  0.994554 -0.141706 -1.921966
```

2

2.1

Series df.A

[12]:
```
df['A']
```

[12]:
```
2019-01-01   -1.714991
2019-01-02   -0.543644
2019-01-03    3.611976
2019-01-04   -1.567843
2019-01-05   -0.268856
2019-01-06    0.868899
Freq: D, Name: A, dtype: float64
```

[13]:
```
df[0:3]
```

[13]:

	A	B	C	D
2019-01-01	-1.714991	0.942568	0.538325	-0.668959
2019-01-02	-0.543644	-1.426672	0.731773	1.401339
2019-01-03	3.611976	-0.566308	0.814891	-0.298263

[15]:
```
df['20190102':'20190104']
```

[15]:

	A	B	C	D
2019-01-02	-0.543644	-1.426672	0.731773	1.401339
2019-01-03	3.611976	-0.566308	0.814891	-0.298263
2019-01-04	-1.567843	-1.080476	0.094566	0.159968

[22]:
```
df[['A','C']]
```

[22]:

	A	C
2019-01-01	-1.714991	0.538325
2019-01-02	-0.543644	0.731773
2019-01-03	3.611976	0.814891
2019-01-04	-1.567843	0.094566
2019-01-05	-0.268856	-0.169201
2019-01-06	0.868899	-0.141706

[26]:
```
df[2:3]
```

[26]:

	A	B	C	D
2019-01-03	3.611976	-0.566308	0.814891	-0.298263

2.1.1 .loc

[28]:
```
df.loc['20190101']
```

[28]:
```
A   -1.714991
B    0.942568
C    0.538325
D   -0.668959
Name: 2019-01-01 00:00:00, dtype: float64
```

[29]:
```
df.loc[:, ['A', 'C']]
```

[29]:
```
                   A         C
2019-01-01 -1.714991  0.538325
2019-01-02 -0.543644  0.731773
2019-01-03  3.611976  0.814891
2019-01-04 -1.567843  0.094566
2019-01-05 -0.268856 -0.169201
2019-01-06  0.868899 -0.141706
```

[30]:
```
df.loc['20190102':'20190104',['A', 'C']]
```

[30]:
```
                   A         C
2019-01-02 -0.543644  0.731773
2019-01-03  3.611976  0.814891
2019-01-04 -1.567843  0.094566
```

[31]:
```
df.loc['20190101', ['A', 'B']]
```

[31]:
```
A   -1.714991
B    0.942568
Name: 2019-01-01 00:00:00, dtype: float64
```

[32]:
```
df.loc['20190101', 'A']
```

[32]:
```
-1.7149905142046848
```

[33]:
```
df.at['20190101', 'A']
```

[33]:
```
-1.7149905142046848
```

2.1.2 .iloc

[35]:
```
df.iloc[3]
```

[35]:
```
A   -1.567843
B   -1.080476
C    0.094566
D    0.159968
Name: 2019-01-04 00:00:00, dtype: float64
```

[36]:
```
df.iloc[3:5, 0:2]
```

[36]:
```
                   A         B
2019-01-04 -1.567843 -1.080476
2019-01-05 -0.268856  0.886491
```

[37]:
```
df.iloc[[1, 2, 4], [0, 2]]
```

[37]:
```
                   A         C
2019-01-02 -0.543644  0.731773
2019-01-03  3.611976  0.814891
2019-01-05 -0.268856 -0.169201
```

[38]:
```
df.iloc[1:3, :]
```

[38]:
```
                   A         B         C         D
2019-01-02 -0.543644 -1.426672  0.731773  1.401339
2019-01-03  3.611976 -0.566308  0.814891 -0.298263
```

[39]:
```
df.iloc[:, 1:3]
```

[39]:
```
                   B         C
2019-01-01  0.942568  0.538325
2019-01-02 -1.426672  0.731773
2019-01-03 -0.566308  0.814891
2019-01-04 -1.080476  0.094566
2019-01-05  0.886491 -0.169201
2019-01-06  0.994554 -0.141706
```

[40]:
```
df.iloc[1, 1]
```

[40]:
```
-1.4266718551281794
```

[41]:
```
df.iat[1, 1]
```

[41]:
```
-1.4266718551281794
```

3

[42]:
```
df[df.A > 0]
```

[42]:

	A	B	C	D
2019-01-03	3.611976	-0.566308	0.814891	-0.298263
2019-01-06	0.868899	0.994554	-0.141706	-1.921966

DataFrame

[43]:
```
df[df > 0]
```

[43]:

	A	B	C	D
2019-01-01	NaN	0.942568	0.538325	NaN
2019-01-02	NaN	NaN	0.731773	1.401339
2019-01-03	3.611976	NaN	0.814891	NaN
2019-01-04	NaN	NaN	0.094566	0.159968
2019-01-05	NaN	0.886491	NaN	NaN
2019-01-06	0.868899	0.994554	NaN	NaN

[46]:
```
df2 = df.copy()
df2['E'] = ['one', 'one', 'two', 'three', 'four', 'three']
df2
```

[46]:

	A	B	C	D	E
2019-01-01	-1.714991	0.942568	0.538325	-0.668959	one
2019-01-02	-0.543644	-1.426672	0.731773	1.401339	one
2019-01-03	3.611976	-0.566308	0.814891	-0.298263	two
2019-01-04	-1.567843	-1.080476	0.094566	0.159968	three
2019-01-05	-0.268856	0.886491	-0.169201	-0.296960	four
2019-01-06	0.868899	0.994554	-0.141706	-1.921966	three

[47]:
```
df2[df2['E'].isin(['one', 'two'])]
```

[47]:

	A	B	C	D	E
2019-01-01	-1.714991	0.942568	0.538325	-0.668959	one
2019-01-02	-0.543644	-1.426672	0.731773	1.401339	one
2019-01-03	3.611976	-0.566308	0.814891	-0.298263	two

4

[49]:
```
s1 = pd.Series([1, 2, 3, 4, 5, 6], index = pd.date_range('20190102', periods=6))
s1
```

[49]:
```
2019-01-02    1
2019-01-03    2
2019-01-04    3
2019-01-05    4
2019-01-06    5
2019-01-07    6
Freq: D, dtype: int64
```

[50]:
```
df['F'] = s1
df
```

[50]:

	A	B	C	D	E	F
2019-01-01	-1.714991	0.942568	0.538325	-0.668959	one	NaN
2019-01-02	-0.543644	-1.426672	0.731773	1.401339	one	1.0
2019-01-03	3.611976	-0.566308	0.814891	-0.298263	two	2.0
2019-01-04	-1.567843	-1.080476	0.094566	0.159968	three	3.0
2019-01-05	-0.268856	0.886491	-0.169201	-0.296960	four	4.0
2019-01-06	0.868899	0.994554	-0.141706	-1.921966	three	5.0

[51]:
```
df.loc[:, 'D'] = np.array([5] * len(df))
df
```

[51]:

	A	B	C	D	E	F
2019-01-01	-1.714991	0.942568	0.538325	5	one	NaN
2019-01-02	-0.543644	-1.426672	0.731773	5	one	1.0
2019-01-03	3.611976	-0.566308	0.814891	5	two	2.0
2019-01-04	-1.567843	-1.080476	0.094566	5	three	3.0
2019-01-05	-0.268856	0.886491	-0.169201	5	four	4.0
2019-01-06	0.868899	0.994554	-0.141706	5	three	5.0

[53]:
```
df3 = df.copy()
```

[54]:
```

```

```
␣
↪---------------------------------------------------------------------------

TypeError                                 Traceback (most recent call␣
↪last)

<ipython-input-54-db17e19f0a2e> in <module>
----> 1 df3[df3 > 0] = -df3
      2 df3


d:\python3.7.5\lib\site-packages\pandas\core\generic.py in __neg__(self)
```

```
   1514             or is_object_dtype(values)
   1515         ):
-> 1516             arr = operator.neg(values)
   1517         else:
   1518             raise TypeError(


TypeError: bad operand type for unary -: 'str'
```

[ ]: