pandas??

August 9, 2019

```
In [2]: import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
```

```
In [3]: df1=DataFrame(np.arange(12.).reshape((3,4)),columns=list('abcd'))
```

```
In [4]: df2=DataFrame(np.arange(20.).reshape((4,5)),columns=list('abcde'))
```

```
In [5]: df1
```

```
Out[5]:      a    b     c     d
        0  0.0  1.0   2.0   3.0
        1  4.0  5.0   6.0   7.0
        2  8.0  9.0  10.0  11.0
```

```
In [6]: df2
```

```
Out[6]:       a     b     c     d     e
        0   0.0   1.0   2.0   3.0   4.0
        1   5.0   6.0   7.0   8.0   9.0
        2  10.0  11.0  12.0  13.0  14.0
        3  15.0  16.0  17.0  18.0  19.0
```

```
In [7]: df1+df2
```

```
Out[7]:       a     b     c     d   e
        0   0.0   2.0   4.0   6.0 NaN
        1   9.0  11.0  13.0  15.0 NaN
        2  18.0  20.0  22.0  24.0 NaN
        3   NaN   NaN   NaN   NaN NaN
```

```
In [8]: df1.add(df2,fill_value=0)
```

```
Out[8]:       a     b     c     d     e
        0   0.0   2.0   4.0   6.0   4.0
        1   9.0  11.0  13.0  15.0   9.0
        2  18.0  20.0  22.0  24.0  14.0
        3  15.0  16.0  17.0  18.0  19.0
```

```
In [9]: df1.reindex(columns=df2.columns,fill_value=0)

Out[9]:      a    b     c     d  e
        0  0.0  1.0   2.0   3.0  0
        1  4.0  5.0   6.0   7.0  0
        2  8.0  9.0  10.0  11.0  0

In [11]: arr=np.arange(12.).reshape((3,4))

In [12]: arr

Out[12]: array([[ 0.,  1.,  2.,  3.],
                [ 4.,  5.,  6.,  7.],
                [ 8.,  9., 10., 11.]])

In [13]: arr[0]

Out[13]: array([0., 1., 2., 3.])

In [15]: arr-arr[0]

Out[15]: array([[0., 0., 0., 0.],
                [4., 4., 4., 4.],
                [8., 8., 8., 8.]])

In [16]: frame=DataFrame(np.arange(12.).reshape((4,3)),columns=list('bde'),index=['Utah','Ohio

In [17]: series=frame.ix[0]

/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarn
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.


In [18]: frame

Out[18]:           b     d     e
         Utah    0.0   1.0   2.0
         Ohio    3.0   4.0   5.0
         Texas   6.0   7.0   8.0
         Oregon  9.0  10.0  11.0

In [19]: series
```

```
Out[19]: b    0.0
         d    1.0
         e    2.0
         Name: Utah, dtype: float64
```

```
In [20]: frame-series
```

```
Out[20]:           b    d    e
         Utah    0.0  0.0  0.0
         Ohio    3.0  3.0  3.0
         Texas   6.0  6.0  6.0
         Oregon  9.0  9.0  9.0
```

```
In [21]: series2=Series(range(3),index=['b','e','f'])
```

```
In [22]: frame+series2
```

```
Out[22]:           b   d     e   f
         Utah    0.0 NaN   3.0 NaN
         Ohio    3.0 NaN   6.0 NaN
         Texas   6.0 NaN   9.0 NaN
         Oregon  9.0 NaN  12.0 NaN
```

```
In [23]: series3=frame['d']
```

```
In [24]: frame
```

```
Out[24]:           b     d     e
         Utah    0.0   1.0   2.0
         Ohio    3.0   4.0   5.0
         Texas   6.0   7.0   8.0
         Oregon  9.0  10.0  11.0
```

```
In [25]: series3
```

```
Out[25]: Utah       1.0
         Ohio       4.0
         Texas      7.0
         Oregon    10.0
         Name: d, dtype: float64
```

```
In [26]: frame.sub(series3,axis=0)
```

```
Out[26]:           b    d    e
         Utah   -1.0  0.0  1.0
         Ohio   -1.0  0.0  1.0
         Texas  -1.0  0.0  1.0
         Oregon -1.0  0.0  1.0
```

```
In [29]: frame=DataFrame(np.random.randn(4,3),columns=list('bde'),index=['Utah','Ohio','Texas'
In [30]: frame
```

```
Out[30]:                b         d         e
         Utah    0.955887  0.008501  1.775859
         Ohio   -0.395190  0.280049  0.732160
         Texas   1.422932  0.221347  1.352449
         Oregon -0.743218 -0.020887 -0.515879
```

```
In [31]: np.abs(frame)
```

```
Out[31]:                b         d         e
         Utah    0.955887  0.008501  1.775859
         Ohio    0.395190  0.280049  0.732160
         Texas   1.422932  0.221347  1.352449
         Oregon  0.743218  0.020887  0.515879
```

```
In [33]: f=lambda x: x.max()-x.min()
```

```
In [34]: frame.apply(f)
```

```
Out[34]: b    2.166150
         d    0.300936
         e    2.291738
         dtype: float64
```

```
In [35]: frame.apply(f,axis=1)
```

```
Out[35]: Utah      1.767357
         Ohio      1.127349
         Texas     1.201585
         Oregon    0.722331
         dtype: float64
```

```
In [36]: def f(x):
             return Series([x.min(),x.max()],index=['min','max'])
```

```
In [37]: frame.apply(f)
```

```
Out[37]:             b         d         e
         min -0.743218 -0.020887 -0.515879
         max  1.422932  0.280049  1.775859
```

```
In [38]: format=lambda x: '%.2f' % x
```

```
In [40]: frame.applymap(format)
```

```
Out[40]:             b      d      e
         Utah     0.96   0.01   1.78
         Ohio    -0.40   0.28   0.73
         Texas    1.42   0.22   1.35
         Oregon  -0.74  -0.02  -0.52
```

```
In [42]: #
```

```
In [43]: frame['e'].map(format)
```

```
Out[43]: Utah       1.78
         Ohio       0.73
         Texas      1.35
         Oregon    -0.52
         Name: e, dtype: object
```

```
In [44]: obj=Series(range(4),index=['d','a','b','c'])
```

```
In [45]: obj.sort_index()
```

```
Out[45]: a    1
         b    2
         c    3
         d    0
         dtype: int64
```

```
In [46]: frame=DataFrame(np.arange(8).reshape((2,4)),index=['three','one'],columns=['d','a','b
```

```
In [47]: frame.sort_index()
```

```
Out[47]:        d  a  b  c
         one    4  5  6  7
         three  0  1  2  3
```

```
In [49]: frame.sort_index(axis=1)
```

```
Out[49]:        a  b  c  d
         three  1  2  3  0
         one    5  6  7  4
```

```
In [50]: frame.sort_index(axis=1,ascending=False)
```

```
Out[50]:        d  c  b  a
         three  0  3  2  1
         one    4  7  6  5
```

```
In [51]: obj=Series([4,7,-3,2])
```

```
In [55]: obj.sort_values()
```

```
Out[55]: 2   -3
         3    2
         0    4
         1    7
         dtype: int64
```

```
In [56]: obj=Series([4,np.nan,7,np.nan,-3,2])
In [57]: obj.sort_values()

Out[57]: 4   -3.0
         5    2.0
         0    4.0
         2    7.0
         1    NaN
         3    NaN
         dtype: float64

In [58]: frame=DataFrame({'b':[4,7,-3,2],'a':[0,1,0,1]})

In [59]: frame

Out[59]:    b  a
         0  4  0
         1  7  1
         2 -3  0
         3  2  1

In [63]: frame.sort_values(by='b')

Out[63]:    b  a
         2 -3  0
         3  2  1
         0  4  0
         1  7  1

In [65]: frame.sort_values(by=['a','b'])

Out[65]:    b  a
         2 -3  0
         0  4  0
         3  2  1
         1  7  1

In [66]: obj=Series([7,-5,7,4,2,0,4])

In [67]: obj.rank()

Out[67]: 0    6.5
         1    1.0
         2    6.5
         3    4.5
         4    3.0
         5    2.0
         6    4.5
         dtype: float64

In [68]: obj.rank(method='first')
```

```
Out[68]: 0    6.0
         1    1.0
         2    7.0
         3    4.0
         4    3.0
         5    2.0
         6    5.0
         dtype: float64
```

```
In [69]: obj.rank(ascending=False, method='max')
```

```
Out[69]: 0    2.0
         1    7.0
         2    2.0
         3    4.0
         4    5.0
         5    6.0
         6    4.0
         dtype: float64
```

```
In [70]: frame=DataFrame({'b':[4.3,7,-3,2],'a':[0,1,0,1],'c':[-2,5,8,-2.5]})
```

```
In [71]: frame
```

```
Out[71]:      b  a    c
         0  4.3  0 -2.0
         1  7.0  1  5.0
         2 -3.0  0  8.0
         3  2.0  1 -2.5
```

```
In [72]: frame.rank(axis=1)
```

```
Out[72]:      b    a    c
         0  3.0  2.0  1.0
         1  3.0  1.0  2.0
         2  1.0  2.0  3.0
         3  3.0  2.0  1.0
```

```
In [73]: obj=Series(range(5),index=['a','a','b','b','c'])
```

```
In [74]: obj
```

```
Out[74]: a    0
         a    1
         b    2
         b    3
         c    4
         dtype: int64
```

```
In [75]: obj.index.is_unique
```

```
Out[75]: False

In [76]: obj['a']

Out[76]: a    0
         a    1
         dtype: int64

In [78]: obj['c']

Out[78]: 4

In [79]: df=DataFrame(np.random.randn(4,3),index=['a','a','b','b'])

In [80]: df

Out[80]:           0         1         2
         a -0.034490 -0.417726 -0.886925
         a  0.094167  0.284157  0.860532
         b  0.425268  2.094657 -0.186384
         b -0.336301  1.191337  1.208016

In [83]: df.loc['b']

Out[83]:           0         1         2
         b  0.425268  2.094657 -0.186384
         b -0.336301  1.191337  1.208016

In [84]: df=DataFrame([[1.4,np.nan],[7.1,-4.5],[np.nan,np.nan],[0.75,-1.3]],index=['a','b','c'

In [85]: df

Out[85]:     one  two
         a  1.40  NaN
         b  7.10 -4.5
         c   NaN  NaN
         d  0.75 -1.3

In [86]: df.sum()

Out[86]: one    9.25
         two   -5.80
         dtype: float64

In [87]: df.sum(axis=1)

Out[87]: a    1.40
         b    2.60
         c    0.00
         d   -0.55
         dtype: float64
```

```
In [88]: df.mean(axis=1,skipna=False)

Out[88]: a       NaN
         b     1.300
         c       NaN
         d    -0.275
         dtype: float64

In [89]: df.idxmax()

Out[89]: one    b
         two    d
         dtype: object

In [90]: df.cumsum()

Out[90]:     one  two
         a  1.40  NaN
         b  8.50 -4.5
         c   NaN  NaN
         d  9.25 -5.8

In [91]: df.describe()

Out[91]:             one       two
         count  3.000000  2.000000
         mean   3.083333 -2.900000
         std    3.493685  2.262742
         min    0.750000 -4.500000
         25%    1.075000 -3.700000
         50%    1.400000 -2.900000
         75%    4.250000 -2.100000
         max    7.100000 -1.300000

In [92]: obj=Series(['a','a','b','c']*4)

In [93]: obj.describe()

Out[93]: count      16
         unique      3
         top         a
         freq        8
         dtype: object

In [100]: # page 145 !!!

In [101]: obj=Series(['c','a','d','a','a','b','b','c','c'])

In [102]: uniques=obj.unique()
In [103]: uniques

Out[103]: array(['c', 'a', 'd', 'b'], dtype=object)

In [109]: obj.value_counts()

Out[109]: a    3
          c    3
          b    2
          d    1
          dtype: int64

In [110]: pd.value_counts(obj.values, sort=False)

Out[110]: c    3
          a    3
          b    2
          d    1
          dtype: int64

In [111]: mask=obj.isin(['b','c'])

In [112]: mask

Out[112]: 0     True
          1    False
          2    False
          3    False
          4    False
          5     True
          6     True
          7     True
          8     True
          dtype: bool

In [113]: obj[mask]

Out[113]: 0    c
          5    b
          6    b
          7    c
          8    c
          dtype: object

In [114]: data=DataFrame({'Qu1':[1,3,4,3,4],'Qu2':[2,3,1,2,3],'Qu3':[1,5,2,4,4]})

In [115]: data
```

```
Out[115]:    Qu1  Qu2  Qu3
          0    1    2    1
          1    3    3    5
          2    4    1    2
          3    3    2    4
          4    4    3    4
```

```
In [116]: result=data.apply(pd.value_counts).fillna(0)
```

```
In [117]: result
```

```
Out[117]:    Qu1  Qu2  Qu3
          1  1.0  1.0  1.0
          2  0.0  2.0  1.0
          3  2.0  2.0  0.0
          4  2.0  0.0  2.0
          5  0.0  0.0  1.0
```

```
In [118]: from numpy import nan as NA
```

```
In [119]: data=Series([1,NA, 3.5, NA, 7])
```

```
In [120]: data.dropna()
```

```
Out[120]: 0    1.0
          2    3.5
          4    7.0
          dtype: float64
```

```
In [121]: data[data.notnull()]
```

```
Out[121]: 0    1.0
          2    3.5
          4    7.0
          dtype: float64
```

```
In [122]: df=DataFrame(np.random.randn(7,3))
```

```
In [124]: df.ix[:4,1]=NA
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarn
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

```
In [125]: df.ix[:2,2]=NA
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarn
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

```
In [126]: df
```

```
Out[126]:           0         1         2
          0 -0.292331       NaN       NaN
          1 -1.035102       NaN       NaN
          2 -1.467619       NaN       NaN
          3 -0.102500       NaN  0.362828
          4  1.141697       NaN  0.029871
          5 -1.685245  1.826811  0.207340
          6  0.355074  0.225247 -0.809508
```

```
In [127]: df.dropna(thresh=3)
```

```
Out[127]:           0         1         2
          5 -1.685245  1.826811  0.207340
          6  0.355074  0.225247 -0.809508
```

```
In [128]: # 3nah
```

```
In [129]: df.fillna(0)
```

```
Out[129]:           0         1         2
          0 -0.292331  0.000000  0.000000
          1 -1.035102  0.000000  0.000000
          2 -1.467619  0.000000  0.000000
          3 -0.102500  0.000000  0.362828
          4  1.141697  0.000000  0.029871
          5 -1.685245  1.826811  0.207340
          6  0.355074  0.225247 -0.809508
```

```
In [132]: df.fillna({1: 0.5, 3:-1})
```

```
Out[132]:           0         1         2
          0 -0.292331  0.500000       NaN
          1 -1.035102  0.500000       NaN
          2 -1.467619  0.500000       NaN
          3 -0.102500  0.500000  0.362828
          4  1.141697  0.500000  0.029871
          5 -1.685245  1.826811  0.207340
          6  0.355074  0.225247 -0.809508
```

```
In [133]: _=df.fillna(0,inplace=True)
```

```
In [134]: df
```

```
Out[134]:           0         1         2
          0 -0.292331  0.000000  0.000000
          1 -1.035102  0.000000  0.000000
          2 -1.467619  0.000000  0.000000
          3 -0.102500  0.000000  0.362828
          4  1.141697  0.000000  0.029871
          5 -1.685245  1.826811  0.207340
          6  0.355074  0.225247 -0.809508
```

```
In [135]: df=DataFrame(np.random.randn(6,3))
```

```
In [136]: df.ix[2:, 1]=NA
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarni
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

```
In [137]: df.ix[4:, 2]=NA
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarni
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

```
In [138]: df
```

```
Out[138]:           0         1         2
          0 -0.883780 -0.619053 -1.990344
          1 -0.350460  0.257555  1.015083
          2 -0.113104       NaN  0.048670
          3 -1.008698       NaN  0.257753
          4  1.227487       NaN       NaN
          5 -0.421137       NaN       NaN
```

```
In [139]: df.fillna(method='ffill')
```

```
Out[139]:           0         1         2
          0 -0.883780 -0.619053 -1.990344
          1 -0.350460  0.257555  1.015083
          2 -0.113104  0.257555  0.048670
          3 -1.008698  0.257555  0.257753
          4  1.227487  0.257555  0.257753
          5 -0.421137  0.257555  0.257753
```

```
In [140]: df.fillna(method='ffill',limit=2)
```

```
Out[140]:           0         1         2
          0 -0.883780 -0.619053 -1.990344
          1 -0.350460  0.257555  1.015083
          2 -0.113104  0.257555  0.048670
          3 -1.008698  0.257555  0.257753
          4  1.227487       NaN  0.257753
          5 -0.421137       NaN  0.257753
```

```
In [141]: data=Series([1., NA, 3.5, NA, 7])
```

```
In [142]: data.fillna(data.mean())
```

```
Out[142]: 0    1.000000
          1    3.833333
          2    3.500000
          3    3.833333
          4    7.000000
          dtype: float64
```

```
In [143]: data=Series(np.random.randn(10),index=[['a','a','a','b','b','b','c','c','d','d'],[1,
```

```
In [144]: data
```

```
Out[144]: a  1   -0.385105
             2   -1.691653
             3   -0.468179
          b  1    0.254043
             2    0.223226
             3   -0.660269
          c  1   -0.475831
             2    0.231813
          d  2    1.772386
             3   -1.006470
          dtype: float64
```

```
In [145]: data.index
```

```
Out[145]: MultiIndex(levels=[['a', 'b', 'c', 'd'], [1, 2, 3]],
                     codes=[[0, 0, 0, 1, 1, 1, 2, 2, 3, 3], [0, 1, 2, 0, 1, 2, 0, 1, 1, 2]])
```

```
In [146]: data['b']
```

```
Out[146]: 1    0.254043
          2    0.223226
          3   -0.660269
          dtype: float64
```

```
In [147]: data['b':'c']
```

```
Out[147]: b  1    0.254043
             2    0.223226
             3   -0.660269
          c  1   -0.475831
             2    0.231813
          dtype: float64
```

```
In [148]: data.ix[['b','d']]
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarn
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

```
Out[148]: b  1    0.254043
             2    0.223226
             3   -0.660269
          d  2    1.772386
             3   -1.006470
          dtype: float64
```

```
In [149]: data[:,2]
```

```
Out[149]: a   -1.691653
          b    0.223226
          c    0.231813
          d    1.772386
          dtype: float64
```

```
In [150]: data.unstack()
```

```
Out[150]:          1         2         3
          a -0.385105 -1.691653 -0.468179
          b  0.254043  0.223226 -0.660269
          c -0.475831  0.231813       NaN
          d       NaN  1.772386 -1.006470
```

```
In [151]: # unstack dataframe stacking

In [152]: data.unstack().stack()

Out[152]: a  1   -0.385105
             2   -1.691653
             3   -0.468179
          b  1    0.254043
             2    0.223226
             3   -0.660269
          c  1   -0.475831
             2    0.231813
          d  2    1.772386
             3   -1.006470
          dtype: float64

In [153]: frame=DataFrame(np.arange(12).reshape((4,3)),index=[['a','a','b','b'],[1,2,1,2]],colu

In [154]: frame

Out[154]:      Ohio     Colorado
              Green Red    Green
          a 1     0   1        2
            2     3   4        5
          b 1     6   7        8
            2     9  10       11

In [155]: frame.index.names=['key1','key2']

In [156]: frame.columns.names=['state','color']

In [157]: frame

Out[157]: state      Ohio     Colorado
          color     Green Red    Green
          key1 key2
          a    1        0   1        2
               2        3   4        5
          b    1        6   7        8
               2        9  10       11

In [158]: frame['Ohio']

Out[158]: color      Green  Red
          key1 key2
          a    1         0    1
               2         3    4
          b    1         6    7
               2         9   10
```

```
In [163]: frame.swaplevel('key1','key2')

Out[163]: state      Ohio      Colorado
          color     Green Red    Green
          key2 key1
          1    a        0   1        2
          2    a        3   4        5
          1    b        6   7        8
          2    b        9  10       11

In [166]: frame.sum(level='key2')

Out[166]: state  Ohio     Colorado
          color Green Red    Green
          key2
          1         6   8       10
          2        12  14       16

In [167]: frame.sum(level='color',axis=1)

Out[167]: color      Green  Red
          key1 key2
          a    1         2    1
               2         8    4
          b    1        14    7
               2        20   10

In [168]: frame=DataFrame({'a':range(7), 'b':range(7,0,-1),'c':['one','one','one','two','two',

In [169]: frame

Out[169]:    a  b    c  d
          0  0  7  one  0
          1  1  6  one  1
          2  2  5  one  2
          3  3  4  two  0
          4  4  3  two  1
          5  5  2  two  2
          6  6  1  two  3

In [170]: frame2=frame.set_index(['c','d'])

In [171]: frame2

Out[171]:        a  b
          c   d
          one 0  0  7
              1  1  6
              2  2  5
          two 0  3  4
              1  4  3
              2  5  2
              3  6  1
```

```
In [172]: frame.set_index(['c','d'],drop=False)

Out[172]:        a  b    c  d
          c   d
          one 0  0  7  one  0
              1  1  6  one  1
              2  2  5  one  2
          two 0  3  4  two  0
              1  4  3  two  1
              2  5  2  two  2
              3  6  1  two  3

In [173]: frame2.reset_index()

Out[173]:       c  d  a  b
          0  one  0  0  7
          1  one  1  1  6
          2  one  2  2  5
          3  two  0  3  4
          4  two  1  4  3
          5  two  2  5  2
          6  two  3  6  1

In [174]: ser=Series(np.arange(3.))

In [176]: ser

Out[176]: 0    0.0
          1    1.0
          2    2.0
          dtype: float64

In [177]: ser2=Series(np.arange(3.),index=['a','b','c'])

In [179]: ser2[-1]

Out[179]: 2.0

In [ ]:
```