Pandas

March 28, 2018

1 Pandas

2

2.1 Pandas

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

3 python

```
In [2]: list = [i for i in range(3)]
        print(list)
```

```
[0, 1, 2]
```

```
In [3]: list.append('seven')
        print(list)
```

```
[0, 1, 2, 'seven']
```

```
In [4]: list.pop()
        print(list)
```

```
[0, 1, 2]
```

```
In [5]: list.insert(3, "tri")
        print(list)
```

```
[0, 1, 2, 'tri']
```

3.1

```
In [6]: squares = []
        for x in range(10):
            squares.append(x**2)
        print(squares)
        squares_true_way = [i**2 for i in range(10)]
        print(squares_true_way)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
In [7]: combs = [(x, y) for x in [1,2,3] for y in [3,1,4] if x != y]
        #
        combs = []
        for x in [1,2,3]:
            for y in [3,1,4]:
                if x != y:
                    combs.append((x, y))
        print(combs)
```

```
[(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)]
```

3.2

```
In [8]: kortej = 12345, 54321, 'hello!'

        toje_kortej = [i for i in range(4)], 12345, 54321, 'hello!'

        print(kortej)
        print(toje_kortej)
```

```
(12345, 54321, 'hello!')
([0, 1, 2, 3], 12345, 54321, 'hello!')
```

```
In [9]: toje_kortej[0].append("DOBAVIM_ELEMENT")
        print(toje_kortej)
```

```
([0, 1, 2, 3, 'DOBAVIM_ELEMENT'], 12345, 54321, 'hello!')
```

```
In [10]: madness = combs, kortej
         print(madness)
         combs.append(5)
         print(combs)
         print(madness)
```

```
([(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)], (12345, 54321, 'hello!'))
[(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4), 5]
([(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4), 5], (12345, 54321, 'hello!'))
```

3.3

```
In [11]: basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
         print(basket)

{'pear', 'banana', 'orange', 'apple'}


In [12]: 'orange' in basket

Out[12]: True

In [13]: a = set('abracadabra')
         b = set('alacazam')
         a - b

Out[13]: {'b', 'd', 'r'}
```

3.4

```
In [14]: tel = {'A': 1, 'B': 2, 'C': 3}

In [15]: tel.keys()

Out[15]: dict_keys(['C', 'A', 'B'])

In [16]: tel.values()

Out[16]: dict_values([3, 1, 2])

In [17]: 'D'in tel

Out[17]: False

In [18]: 'A'in tel

Out[18]: True
```

3.5 >>> s = pd.Series(data, index=index)

```
In [19]: s = pd.Series(np.random.randn(5), index=['a', 'b', 'c', 'd', 'e'])
         s
```

```
Out[19]: a    1.202065
         b   -0.668953
         c    0.953328
         d   -1.180525
         e   -0.284584
         dtype: float64

In [20]: d = {'a' : 0., 'b' : 1., 'c' : 2.}

         pd.Series(d)

Out[20]: a    0.0
         b    1.0
         c    2.0
         dtype: float64

In [21]: pd.Series(d, index=['b', 'c', 'd', 'a'])

Out[21]: b    1.0
         c    2.0
         d    NaN
         a    0.0
         dtype: float64

In [22]: s[0]

Out[22]: 1.2020649596548885

In [23]: s[:3]

Out[23]: a    1.202065
         b   -0.668953
         c    0.953328
         dtype: float64

In [24]: s[3:]

Out[24]: d   -1.180525
         e   -0.284584
         dtype: float64

In [25]: s[s>0.1]

Out[25]: a    1.202065
         c    0.953328
         dtype: float64
```

3.6

3.6.1 NumPy ,

3.6.2 , .

```
In [26]: import random
         a = np.random.random((2))
         b = [random.random() for i in range(0,2)]
         g = pd.Series([a,b], index=['one', 'two'])
         g
```

```
Out[26]: one    [0.09106196736781291, 0.462293550087279]
         two     [0.9421234764073749, 0.307740989603843]
         dtype: object
```

```
In [27]: a = np.insert(a,2,5,axis = 0)
         b.append(4)
```

```
In [28]: g
```

```
Out[28]: one      [0.09106196736781291, 0.462293550087279]
         two    [0.9421234764073749, 0.307740989603843, 4]
         dtype: object
```

```
In [29]: g = pd.Series([a,b], index=['one', 'two'])
```

```
In [30]: g
```

```
Out[30]: one    [0.09106196736781291, 0.462293550087279, 5.0]
         two       [0.9421234764073749, 0.307740989603843, 4]
         dtype: object
```

```
In [31]: np.sin(g['one'])
```

```
Out[31]: array([ 0.09093617,  0.44600208, -0.95892427])
```

```
In [32]: np.sin(g['two'])
```

```
Out[32]: array([ 0.80880868,  0.30290653, -0.7568025 ])
```

4 DataFrame

```
In [33]:  d = {'one' : pd.Series([1., 2., 3.], index=['a', 'b', 'c']),'two' : pd.Series([1., 2.,
```

```
In [34]: df = pd.DataFrame(d)
```

```
In [35]: df
```

```
Out[35]:    one  two
         a  1.0  1.0
         b  2.0  2.0
         c  3.0  3.0
         d  NaN  4.0
```

4.1

4.1.1 NumPy ,

4.1.2 , .

4.1.3 .

5 (ř ř)

```
In [36]: df['one']['d']=4.0

In [37]: df
```

Out[37]:

	one	two
a	1.0	1.0
b	2.0	2.0
c	3.0	3.0
d	4.0	4.0

```
In [38]: data1 = {'A':np.random.random(3),'B':np.random.random(3),'C':np.random.random(3)}
         df1 = pd.DataFrame(data1)
         df1.assign(e=s[:3].values)
```

Out[38]:

	A	B	C	e
0	0.028844	0.201677	0.435285	1.202065
1	0.285689	0.288644	0.281650	-0.668953
2	0.567317	0.343014	0.820218	0.953328

```
In [39]: df1['A'] = df1['B'] * df1['C']
         df1['flag'] = df1['A'] > 0.1
         df1
```

Out[39]:

	A	B	C	flag
0	0.087787	0.201677	0.435285	False
1	0.081296	0.288644	0.281650	False
2	0.281347	0.343014	0.820218	True

```
In [40]: df1.insert(2, 'copy_of_A', df1['A'][:2])

In [41]: df1
```

Out[41]:

	A	B	copy_of_A	C	flag
0	0.087787	0.201677	0.087787	0.435285	False
1	0.081296	0.288644	0.081296	0.281650	False
2	0.281347	0.343014	NaN	0.820218	True

```
In [43]: json_df = pd.read_json('example.json')

In [44]: json_df
```

```
Out[44]:    a  b
         0  1  A
         1  2  B
         2  3  C
         3  4  D
         4  5  E
         5  6  F

In [45]: np.sin(json_df['a'])

Out[45]: 0    0.841471
         1    0.909297
         2    0.141120
         3   -0.756802
         4   -0.958924
         5   -0.279415
         Name: a, dtype: float64
```

5.1

5.2 task-[1-2].json

5.2.1

5.2.2 // ,