

Pandas Tutorial

In [1]:

```
import pandas as pd
```

In [2]:

```
print(pd.__version__)
```

0.23.0

Series create,delete,query

In [3]:

```
#createing a series from a list
arr = [0,1,2,3,4]
s1 = pd.Series(arr)
s1
```

Out[3]:

```
0    0
1    1
2    2
3    3
4    4
dtype: int64
```

In [4]:

```
order = [1,2,3,4,5]
s2 = pd.Series(arr, index = order)
s2
```

Out[4]:

```
1    0
2    1
3    2
4    3
5    4
dtype: int64
```

In [7]:

```
import numpy as np
n = np.random.randn(5) #create random Ndarray
n
```

Out[7]:

```
array([-0.07788705,  0.95421712, -0.38592665,  0.07268258, -0.96966898])
```

In [8]:

```
n = np.random.randn(5)
index = ['a','b','c','d','e']
s2 = pd.Series(n,index = index)
s2
```

Out[8]:

```
a   -1.893366
b    0.560269
c   -1.768075
```

```
c    1.700075  
d   -0.041365  
e    0.900363  
dtype: float64
```

In [9]:

```
#create series from dictionary  
d = {'a':1, 'b':2, 'c':3, 'd':4, 'e':5}  
s3 = pd.Series(d)  
s3
```

Out[9]:

```
a    1  
b    2  
c    3  
d    4  
e    5  
dtype: int64
```

In [11]:

```
#you can modify the index of series  
print(s1)  
s1.index = ['A', 'B', 'C', 'D', 'E']  
s1
```

```
0    0  
1    1  
2    2  
3    3  
4    4  
dtype: int64
```

Out[11]:

```
A    0  
B    1  
C    2  
D    3  
E    4  
dtype: int64
```

In [13]:

```
#slicing  
a = s1[:3]  
a
```

Out[13]:

```
A    0  
B    1  
C    2  
dtype: int64
```

In [14]:

```
s1[:-1]
```

Out[14]:

```
A    0  
B    1  
C    2  
D    3  
dtype: int64
```

In [15]:

```
s1[:-2]
```

```
s1[1:2]
```

```
Out[15]:
```

```
A    0
B    1
C    2
dtype: int64
```

```
In [16]:
```

```
s1[2:]
```

```
Out[16]:
```

```
C    2
D    3
E    4
dtype: int64
```

```
In [17]:
```

```
s1[-1:]
```

```
Out[17]:
```

```
E    4
dtype: int64
```

```
In [18]:
```

```
s1[:]
```

```
Out[18]:
```

```
A    0
B    1
C    2
D    3
E    4
dtype: int64
```

```
In [19]:
```

```
s4 = s1.append(s3)
s4
```

```
Out[19]:
```

```
A    0
B    1
C    2
D    3
E    4
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

```
In [20]:
```

```
s4.drop('e')
```

```
Out[20]:
```

```
A    0
B    1
C    2
D    3
E    4
```

```
a    1
b    2
c    3
d    4
dtype: int64
```

In [21]:

```
s4
```

Out[21]:

```
A    0
B    1
C    2
D    3
E    4
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

Series Operations

In [22]:

```
arr1 = [0,1,2,3,4,5,7]
arr2 = [6,7,8,9,5]
```

In [23]:

```
s5 = pd.Series(arr2)
s5
```

Out[23]:

```
0    6
1    7
2    8
3    9
4    5
dtype: int64
```

In [24]:

```
s6 = pd.Series(arr1)
s6
```

Out[24]:

```
0    0
1    1
2    2
3    3
4    4
5    5
6    7
dtype: int64
```

In [25]:

```
s5.add(s6)
```

Out[25]:

```
0    6.0
1    8.0
2   10.0
3    12.0
```

```
3      12.0
4       9.0
5      NaN
6      NaN
dtype: float64
```

In [26]:

```
s5.sub(s6)
```

Out[26]:

```
0      6.0
1      6.0
2      6.0
3      6.0
4      1.0
5      NaN
6      NaN
dtype: float64
```

In [31]:

```
s5.mul(s6)
```

Out[31]:

```
0      0.0
1      7.0
2     16.0
3     27.0
4     20.0
5      NaN
6      NaN
dtype: float64
```

In [28]:

```
s5.div(s6)
```

Out[28]:

```
0      inf
1    7.000000
2    4.000000
3    3.000000
4    1.250000
5      NaN
6      NaN
dtype: float64
```

In [30]:

```
s6.median()
```

Out[30]:

```
3.0
```

Create Dataframe

In [32]:

```
dates = pd.date_range('today', periods = 6) # Define time sequence as index
dates
```

Out[32]:

```
DatetimeIndex(['2019-06-25 11:47:21.446865', '2019-06-26 11:47:21.446865',
               '2019-06-27 11:47:21.446865', '2019-06-28 11:47:21.446865',
               '2019-06-29 11:47:21.446865', '2019-06-30 11:47:21.446865'],
              dtype='datetime64[ns]', freq='D')
```

```
dtype='datetime64[ns]', freq='D')
```

```
In [33]:
```

```
dates = pd.date_range('today', periods = 6)
num_arr = np.random.randn(6,4) #import numpy random array
num_arr
```

```
Out[33]:
```

```
array([[ 0.82930431,  0.4811995 ,  1.31532364, -0.71012505],
       [ 0.14751642,  0.36503439, -0.33122858,  0.92140384],
       [ 0.18476184,  0.67979586,  0.75683623, -0.10789186],
       [-1.02795989, -1.16056354, -0.50625387, -0.73408948],
       [-0.79544592, -1.19454427,  0.88934828,  0.67229091],
       [ 1.23127779, -0.54049387, -1.43345136, -1.24953489]])
```

```
In [39]:
```

```
dates = pd.date_range('today', periods = 6)
num_arr = np.random.randn(6,4)
columns = ['A','B','C','D'] #use the table as the column name

df1 = pd.DataFrame(num_arr, index = dates, columns = columns)
df1
```

```
Out[39]:
```

	A	B	C	D
2019-06-25 12:08:41.949925	-0.696586	-0.404536	-1.164790	-0.290514
2019-06-26 12:08:41.949925	-0.664101	-0.713482	0.542591	0.268359
2019-06-27 12:08:41.949925	0.146754	-0.611483	1.512614	-0.317588
2019-06-28 12:08:41.949925	-0.149249	1.407380	-1.623165	-0.025980
2019-06-29 12:08:41.949925	-0.244226	0.906728	-0.220997	-2.303296
2019-06-30 12:08:41.949925	0.060454	1.881131	0.278896	-0.353786

```
In [40]:
```

```
#create DataFrame with dictionary
data = {'Animal': ['cat','cat','dog','snake','dog'], 'Age': [2.3,3.0,2,np.nan,4.0], 'Visits': [1,3,2,3,1], 'priority':['yes','yes','no','yes','no']}
labels = ['a','b','c','d','e']
df2 = pd.DataFrame(data,index = labels)
df2
```

```
Out[40]:
```

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	2.0	2	no
d	snake	NaN	3	yes
e	dog	4.0	1	no

```
In [41]:
```

```
#see datatypes of array
df2.dtypes
```

```
Out[41]:
```

```
Animal      object
```

```
Age          float64
Visits       int64
priority     object
dtype: object
```

In [43]:

```
df2.head(2)
```

Out[43]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes

In [45]:

```
df2.tail(3)
```

Out[45]:

	Animal	Age	Visits	priority
c	dog	2.0	2	no
d	snake	NaN	3	yes
e	dog	4.0	1	no

In [48]:

```
print(df2.index)
df2.columns
```

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

Out[48]:

```
Index(['Animal', 'Age', 'Visits', 'priority'], dtype='object')
```

In [49]:

```
df2.values
```

Out[49]:

```
array([[ 'cat', 2.3, 1, 'yes'],
       [ 'cat', 3.0, 3, 'yes'],
       [ 'dog', 2.0, 2, 'no'],
       [ 'snake', nan, 3, 'yes'],
       [ 'dog', 4.0, 1, 'no']], dtype=object)
```

In [50]:

```
df2.describe() #see statical data of dataframe
```

Out[50]:

	Age	Visits
count	4.000000	5.0
mean	2.825000	2.0
std	0.888351	1.0
min	2.000000	1.0

	Age	Visits
25%	2.225000	1.0
50%	2.650000	2.0
75%	3.250000	3.0
max	4.000000	3.0

In [51]:

```
df2.T
```

Out[51]:

	a	b	c	d	e
Animal	cat	cat	dog	snake	dog
Age	2.3	3	2	NaN	4
Visits	1	3	2	3	1
priority	yes	yes	no	yes	no

In [54]:

```
df2.sort_values(by = 'Age')
```

Out[54]:

	Animal	Age	Visits	priority
c	dog	2.0	2	no
a	cat	2.3	1	yes
b	cat	3.0	3	yes
e	dog	4.0	1	no
d	snake	NaN	3	yes

In [58]:

```
#slicing dataframe
df2[1:3]
```

Out[58]:

	Animal	Age	Visits	priority
b	cat	3.0	3	yes
c	dog	2.0	2	no

In [60]:

```
#query dataframe by tag
df2[['Age','Visits']]
```

Out[60]:

	Age	Visits
a	2.3	1
b	3.0	3
c	2.0	2
d	NaN	3

	Animal	Age	Visits
e	dog	4.0	1

In [61]:

```
df2.iloc[1:3] #query rows 2,3
```

Out[61]:

	Animal	Age	Visits	priority
b	cat	3.0	3	yes
c	dog	2.0	2	no

In [65]:

```
df3 = df2.copy()
df3
```

Out[65]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	2.0	2	no
d	snake	NaN	3	yes
e	dog	4.0	1	no

In [66]:

```
df3.isnull()
```

Out[66]:

	Animal	Age	Visits	priority
a	False	False	False	False
b	False	False	False	False
c	False	False	False	False
d	False	True	False	False
e	False	False	False	False

In [67]:

```
df3.loc['c', 'Age'] = 1.5
df3
```

Out[67]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	1.5	2	no
d	snake	NaN	3	yes
e	dog	4.0	1	no

In [70]:

```
df3[['Age']].mean()
```

Out[70]:

```
Age      2.7
dtype: float64
```

In [73]:

```
df3['Visits'].sum()
```

Out[73]:

```
10
```

In [74]:

```
df3.sum()
```

Out[74]:

```
Animal      catcatdogsnakedog
Age          10.8
Visits       10
priority     yesyesnoyesno
dtype: object
```

In [76]:

```
string = pd.Series(['A', 'B', 'C', 'D', 'np.nan', 'abab'])
string
```

Out[76]:

```
0      A
1      B
2      C
3      D
4  np.nan
5    abab
dtype: object
```

In [77]:

```
string = pd.Series(['A', 'B', 'C', 'D', 'np.nan', 'abab'])
string.str.lower()
```

Out[77]:

```
0      a
1      b
2      c
3      d
4  np.nan
5    abab
dtype: object
```

Operations for DataFrame missing values

In [80]:

```
df4 = df3.copy()
df4.fillna(4)
```

Out[80]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	1.5	2	no
d	snake	4.0	3	yes
e	dog	4.0	1	no

In [83]:

```
df4 = df3.copy()
df4.fillna(4)
```

Out[83]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	1.5	2	no
d	snake	4.0	3	yes
e	dog	4.0	1	no

In [85]:

```
df5 = df3.copy()
df5.dropna(how = 'any')
```

Out[85]:

	Animal	Age	Visits	priority
a	cat	2.3	1	yes
b	cat	3.0	3	yes
c	dog	1.5	2	no
e	dog	4.0	1	no

DataFrame file oprations

In [86]:

```
df3.to_csv('animal.csv')
```