Pandas Series -- Practice Code

In [1]:
```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from numpy.random import randn
```

In [2]:
```
S1 = Series( [10, 20, 30, 40 ])
S1
```

Out[2]:
```
0    10
1    20
2    30
3    40
dtype: int64
```

In [3]:
```
S2 = Series([10, 20, 30, 40 ], index=['A', 'B', 'C', 'D'])
S2
```

Out[3]:
```
A    10
B    20
C    30
D    40
dtype: int64
```

In [4]:
```
scores = [ 92, 88, 95, 85, 98 ]
students = ['Gary', 'Alex', 'Kris', 'Tom', 'Cathy']

S3 = Series(scores, index=students)
S3
```

Out[4]:
```
Gary     92
Alex     88
Kris     95
Tom      85
Cathy    98
dtype: int64
```

In [5]:
```
S3['Alex']
```

Out[5]:
```
88
```

In [6]:
```
'Gary'in S3
```

Out[6]:
```
True
```

In [7]:
```
'Mike' in S3
```

Out[7]:
```
False
```

In [8]:
```
# Convert to Dictionary

S3d = S3.to_dict()
S3d
```

Out[8]:
```
{'Alex': 88, 'Cathy': 98, 'Gary': 92, 'Kris': 95, 'Tom': 85}
```

In [9]:
```
S4 = Series( S3d )
```

In [10]:
```
S4
```

Out[10]:
```
Alex     88
Cathy    98
Gary     92
Kris     95
Tom      85
dtype: int64
```

In [11]:
```
pd.isnull(S4)
```

Out[11]:
```
Alex     False
Cathy    False
Gary     False
Kris     False
Tom      False
dtype: bool
```

In [12]:
```
pd.notnull( S4)
```

Out[12]:
```
Alex     True
Cathy    True
Gary     True
Kris     True
Tom      True
dtype: bool
```

In [13]:
```
S3
```

Out[13]:
```
Gary     92
Alex     88
Kris     95
Tom      85
Cathy    98
dtype: int64
```

In [14]:
```
S4
```

Out[14]:
```
Alex     88
Cathy    98
Gary     92
Kris     95
Tom      85
dtype: int64
```

In [15]:
```
S3 + S4
```

Out[15]:
```
Alex     176
Cathy    196
Gary     184
Kris     190
Tom      170
dtype: int64
```

In [16]:
```
S4
```

Out[16]:
```
Alex     88
Cathy    98
Gary     92
Kris     95
Tom      85
dtype: int64
```

In [17]:
```
S4.name = "Python 101 Students"
S4
```

Out[17]:
```
Alex     88
Cathy    98
Gary     92
Kris     95
Tom      85
Name: Pyhthon 101 Students, dtype: int64
```

Indexing

In [18]:
```
S4['Gary']
```

Out[18]:
```
92
```

In [19]:
```
S4.index
```

Out[19]:
```
Index([u'Alex', u'Cathy', u'Gary', u'Kris', u'Tom'], dtype='objec
t')
```

In [20]:
```
S4.values
```

Out[20]:
```
array([88, 98, 92, 95, 85])
```

In [21]:
```
S4
```

Out[21]:
```
Alex     88
Cathy    98
Gary     92
Kris     95
Tom      85
Name: Pyhthon 101 Students, dtype: int64
```

In [22]:
```
S4[1]
```

Out[22]:
```
98
```

In [23]:
```
S4[2:]
```

Out[23]:
```
Gary    92
Kris    95
Tom     85
Name: Pyhthon 101 Students, dtype: int64
```

Reindexing

In [24]:
```
x = list( range(11, 16))
ind = ['A', 'B', 'C', 'D', 'E']

S5 = Series( x, index=ind )
S5
```

Out[24]:
```
A    11
B    12
C    13
D    14
E    15
dtype: int64
```

In [25]:
```
ind2 = ['A', 'B', 'C', 'D', 'E', 'F', 'G']

S6 = S5.reindex( ind2 )
S6
```

Out[25]:
```
A    11
B    12
C    13
D    14
E    15
F   NaN
G   NaN
dtype: float64
```

In [27]:
```
ind3 = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']

S7 = S6.reindex( ind3, fill_value = 0 )
S7
```

Out[27]:
```
A    11
B    12
C    13
D    14
E    15
F   NaN
G   NaN
H     0
I     0
dtype: float64
```

Selections

In [28]:
```
S7
```

Out[28]:
```
A    11
B    12
C    13
D    14
E    15
F   NaN
G   NaN
H     0
I     0
dtype: float64
```

In [30]:
```
S7 * 2
```

Out[30]:
```
A    22
B    24
C    26
D    28
E    30
F   NaN
G   NaN
H     0
I     0
dtype: float64
```

In [31]:
```
S7
```

Out[31]:
```
A    11
B    12
C    13
D    14
E    15
F   NaN
G   NaN
H     0
I     0
dtype: float64
```

In [32]:
```
S7['D']
```

Out[32]:
```
14.0
```

In [33]:
```
S7[ ['C', 'E', 'F'] ]
```

Out[33]:
```
C    13
E    15
F   NaN
dtype: float64
```

In [34]:
```
S7
```

Out[34]:
```
A    11
B    12
C    13
D    14
E    15
F   NaN
G   NaN
H     0
I     0
dtype: float64
```

In [35]:
```
S7[ S7 > 13 ]
```

Out[35]:
```
D    14
E    15
dtype: float64
```

In [36]:
```
S7[ S7 == 0 ] = 100
```

In [37]:
```
S7
```

Out[37]:
```
A     11
B     12
C     13
D     14
E     15
F    NaN
G    NaN
H    100
I    100
dtype: float64
```

Data Alignment

In [39]:
```
S8 = Series( [0, 1, 2 ], index=['A', 'B', 'C'] )
S9 = Series( [3, 4, 5, 6 ], index=['A', 'B', 'C', 'D'] )
```

In [40]:
```
S8
```

Out[40]:
```
A    0
B    1
C    2
dtype: int64
```

In [41]:
```
S9
```

Out[41]:
```
A    3
B    4
C    5
D    6
dtype: int64
```

In [42]:
```
S8 + S9
```

Out[42]:
```
A     3
B     5
C     7
D   NaN
dtype: float64
```

In [43]:
```
S8 = S8 + S9
```

In [44]:
```
S8
```

Out[44]:
```
A     3
B     5
C     7
D   NaN
dtype: float64
```

Rank() and Sort()

In [45]:
```
ind = "a b c d e f g h i j".split()
ind
```

Out[45]:
```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

In [46]:
```
x = []
for i in range(10):
    x.append( np.random.randint( 1, 100))
x
```

Out[46]:
```
[46, 85, 76, 17, 97, 65, 80, 11, 55, 25]
```

In [47]:
```
S10 = Series( x, index=ind)
S10
```

Out[47]:
```
a    46
b    85
c    76
d    17
e    97
f    65
g    80
h    11
i    55
j    25
dtype: int64
```

In [48]:
```
S10.rank()
```

Out[48]:
```
a     4
b     9
c     7
d     2
e    10
f     6
g     8
h     1
i     5
j     3
dtype: float64
```

In [49]:
```
S10.sort()
S10
```

Out[49]:
```
h    11
d    17
j    25
a    46
i    55
f    65
c    76
g    80
b    85
e    97
dtype: int64
```

In [50]:
```
S10.rank()
```

Out[50]:
```
h     1
d     2
j     3
a     4
i     5
f     6
c     7
g     8
b     9
e    10
dtype: float64
```

Missing Data

In [51]:
```
S11 = Series([100, 200, np.nan, 400])
S11
```

Out[51]:
```
0    100
1    200
2    NaN
3    400
dtype: float64
```

In [52]:
```
S11.isnull()
```

Out[52]:
```
0    False
1    False
2     True
3    False
dtype: bool
```

In [53]:
```
S12 = S11.dropna()
S12
```

Out[53]:
```
0    100
1    200
3    400
dtype: float64
```

Multi-level Indexing

In [55]:
```
ind1 = [1,1,1, 2, 2, 2 ]
ind2 = "a b c a b c".split()

x = []
for i in range(6):
    x.append( np.random.randint( 101, 1000))
```

In [56]:
```
S13 = Series( x, index=[ ind1, ind2 ])
S13
```

Out[56]:
```
1  a    453
   b    902
   c    645
2  a    101
   b    701
   c    393
dtype: int64
```

In [57]:
```
S13[1]
```

Out[57]:
```
a    453
b    902
c    645
dtype: int64
```

In [61]:
```
S13[ :, 'a']
```

Out[61]:
```
1    453
2    101
dtype: int64
```

In [62]:
```
df1 = S13.unstack()
df1
```

Out[62]:

	a	b	c
1	453	902	645
2	101	701	393

In [ ]:
```

```