In [3]:

```
import pandas as pd
s=pd.Series([1,2,3])
print(s)
```

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

In [4]:

```
import pandas as pd
s=pd.Series([1,2,3],index=['a','b','c'])
print(s)
```

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

In [9]:

```
import pandas as pd
s=pd.Series([1,2,3],index=['a','b','c'])
print(s.values)
print(s.index)
print(s['c'],s[1],s[2])
```

```
[1 2 3]
Index(['a', 'b', 'c'], dtype='object')
3 2 3
```

In [10]:

```
import pandas as pd
s=pd.Series([15,-2,3,1])
print(s[0:2])
```

```
0    15
1    -2
dtype: int64
```

In [12]:

```
import pandas as pd
s=pd.Series([15,-2,3,1],index=['x','y','z','w'])
print(s[0:2])
print(s[['y','w']])
```

```
x    15
y    -2
dtype: int64
y   -2
w    1
dtype: int64
```

In [13]:

```
import pandas as pd
s=pd.Series([15,-2,3,1],index=['x','y','z','w'])
s[1]=0
print(s)
```

```
x    15
y     0
z     3
w     1
dtype: int64
```

In [15]:

```
import pandas as pd
s=pd.Series([15,-2,3,1],index=['x','y','z','w'])
ser=s
print(ser)
```

```
x    15
y    -2
z     3
w     1
dtype: int64
```

In [16]:

```
import pandas as pd
s=pd.Series([15,-2,3,1],index=['x','y','z','w'])
ser=pd.Series(s)
print(ser)
```

```
x    15
y    -2
z     3
w     1
dtype: int64
```

In [18]:

```
import numpy as np
arr=np.array([1,2,3,4])
s1=pd.Series(arr)
print(s1)
arr[2]=-5
print(arr)
print(s1)
```

```
0    1
1    2
2    3
3    4
dtype: int32
[ 1  2 -5  4]
0    1
1    2
2   -5
3    4
dtype: int32
```

In [19]:

```
import pandas as pd
ser=pd.Series([5,-2,3,4])
print(ser>2)
print(ser[ser>2])
```

```
0     True
1    False
2     True
3     True
dtype: bool
0    5
2    3
3    4
dtype: int64
```

In [20]:

```
ser=pd.Series([10,11,5,8,3],index=['a','b','c','d','e'])
print(ser>6)
print(ser[ser>6])
```

```
a     True
b     True
c    False
d     True
e    False
dtype: bool
a    10
b    11
d     8
dtype: int64
```

In [21]:

```
ser=pd.Series([10,11,5,8,3],index=['a','b','c','d','e'])
print(ser+4)
```

```
a    14
b    15
c     9
d    12
e     7
dtype: int64
```

In [25]:

```
ser=pd.Series([10,11,-5,8,3],index=['a','b','c','d','e'])
print(np.log(ser))
```

```
a    2.302585
b    2.397895
c         NaN
d    2.079442
e    1.098612
dtype: float64

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\series.py:853: Runt
imeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

In [23]:

```
ser=pd.Series([10,11,5,8,3],index=['a','b','c','d','e'])
print(ser/2)
```

```
a    5.0
b    5.5
c    2.5
d    4.0
e    1.5
dtype: float64
```

In [24]:

```
ser=pd.Series([1,0,2,1,2,3],index=['white','white','red','green','yellow','blue'])
print(ser)
```

```
white     1
white     0
red       2
green     1
yellow    2
blue      3
dtype: int64
```

In [4]:

```
import pandas as pd
ser=pd.Series([1,0,2,1,2,3],index=['white','white','red','green','yellow','blue'])
print(ser)
print(ser.unique())
print(ser.value_counts())
print(ser.isin([1,3]))
```

```
white     1
white     0
red       2
green     1
yellow    2
blue      3
dtype: int64
[1 0 2 3]
2    2
1    2
3    1
0    1
dtype: int64
white      True
white     False
red       False
green      True
yellow    False
blue       True
dtype: bool
```

In [5]:

```
#not a number method
import numpy as np
import pandas as pd
ser=pd.Series([5,-2,np.nan,4,0])
print(ser)
```

```
0    5.0
1   -2.0
2    NaN
3    4.0
4    0.0
dtype: float64
```

In [8]:

```
ser=pd.Series([8,-3,np.nan,4,9])
print(ser)
print(ser.isnull())
print(ser.notnull())
#filtering the values
print(ser[ser.notnull()])
```

```
0    8.0
1   -3.0
2    NaN
3    4.0
4    9.0
dtype: float64
0    False
1    False
2     True
3    False
4    False
dtype: bool
0     True
1     True
2    False
3     True
4     True
dtype: bool
0    8.0
1   -3.0
3    4.0
4    9.0
dtype: float64
```

In [9]:

```
#series as dictionary
mydict={'red':2000,'yellow':500,'blue':300}
ser1=pd.Series(mydict)
print(ser1)
```

```
red       2000
yellow     500
blue       300
dtype: int64
```

In [14]:

```
mydict={'red':2000,'yellow':500,'blue':300}
colors=['red','yellow','blue','green']
ser1=pd.Series(mydict,index=colors)
print(ser1)
print(ser1.notnull())
print(ser1[ser1.notnull()])
```

```
red       2000.0
yellow     500.0
blue       300.0
green        NaN
dtype: float64
red        True
yellow     True
blue       True
green     False
dtype: bool
red       2000.0
yellow     500.0
blue       300.0
dtype: float64
```

In [20]:

```
#operations between series
mydict={'red':2000,'yellow':500,'blue':300}
mydict1={'red':500,'yellow':700,'blue':500,'green':400}
ser=pd.Series(mydict)
ser1=pd.Series(mydict1)
print(ser)
print(ser1)
print("ser1+ser",ser1+ser,sep='\n')
```

```
red       2000
yellow     500
blue       300
dtype: int64
red       500
yellow    700
blue      500
green     400
dtype: int64
ser1+ser
blue       800.0
green        NaN
red       2500.0
yellow    1200.0
dtype: float64
```

In [17]:

```
import numpy as np
frame1=pd.DataFrame([[6,3,2,1],[2,1,4,6],[1,2,0,3],[4,0,1,2]],
                    index=['red','blue','green','yellow'],
                    columns=['pen','ball','paper','pencil'])
print(frame1)
#sorting
print(frame1.sort_values(by='ball'))
```

```
        pen  ball  paper  pencil
red       6     3      2       1
blue      2     1      4       6
green     1     2      0       3
yellow    4     0      1       2
        pen  ball  paper  pencil
yellow    4     0      1       2
blue      2     1      4       6
green     1     2      0       3
red       6     3      2       1
```

In [28]:

```
#data frame
import pandas as pd
mydict={'color':['red','yellow','green','blue','orange'],
        'object':['ball','pen','book','scale','mug'],
        'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(mydict)
print(frame)
print('changing index values')
frame=pd.DataFrame(mydict,index=['one','two','three','four','five'])
print(frame)
print('specifying the columns')
frame=pd.DataFrame(mydict,index=['one','two','three','four','five'],
                   columns=['object','price'])
print(frame)
```

```
    color object  price
0     red   ball    1.5
1  yellow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
changing index values
        color object  price
one       red   ball    1.5
two    yellow    pen    2.4
three   green   book    1.3
four     blue  scale    3.6
five   orange    mug    4.8
specifying the columns
      object  price
one     ball    1.5
two      pen    2.4
three   book    1.3
four   scale    3.6
five     mug    4.8
```

In [46]:

```
mydict={'color':['red','yellow','green','blue','orange'],
       'object':['ball','pen','book','scale','mug'],
       'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(mydict,index=['one','two','three','four','five'])
print(frame)
print()
print(frame['color'],end="\n")
print(frame.price)
print(frame.columns,frame.index,sep="\n")
print(frame.values)
```

```
        color object  price
one       red   ball    1.5
two    yellow    pen    2.4
three   green   book    1.3
four     blue  scale    3.6
five   orange    mug    4.8

one         red
two      yellow
three     green
four       blue
five     orange
Name: color, dtype: object
one      1.5
two      2.4
three    1.3
four     3.6
five     4.8
Name: price, dtype: float64
Index(['color', 'object', 'price'], dtype='object')
Index(['one', 'two', 'three', 'four', 'five'], dtype='object')
[['red' 'ball' 1.5]
 ['yellow' 'pen' 2.4]
 ['green' 'book' 1.3]
 ['blue' 'scale' 3.6]
 ['orange' 'mug' 4.8]]
```

In [41]:

```
frame1=pd.DataFrame([[4,'fox'],[2,'kangaroo'],[4,'deer'],[8,'spider'],[np.nan,'snake'
]],
                    columns=['no_oflegs','animal'],
                    index=[0,1,2,3,4])
print(frame1)
```

```
   no_oflegs    animal
0        4.0       fox
1        2.0  kangaroo
2        4.0      deer
3        8.0    spider
4        NaN     snake
```

In [47]:

```
frame3=pd.DataFrame(np.arange(16).reshape(4,4),
                    index=['red','yellow','green','blue'],
                    columns=['ball','pen','book','mug'])
print(frame3)
```

```
        ball  pen  book  mug
red        0    1     2    3
yellow     4    5     6    7
green      8    9    10   11
blue      12   13    14   15
```

In [50]:

```
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
print(frame.loc[2])
print(frame.loc[[1,3]])
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
colors    green
object     book
price       1.3
Name: 2, dtype: object
  colors object  price
1  yelow    pen    2.4
3   blue  scale    3.6
```

In [54]:

```
#slicing frame object
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
print(frame[0:1])
print(frame[1:3])
print(frame['colors'][3])
print(frame['object'][2])
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
  colors object  price
0    red   ball    1.5
  colors object  price
1  yelow    pen    2.4
2  green   book    1.3
blue
book
```

In [63]:

```
#assigning values to the frame
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
print()
frame.index.names=['idx']
frame.columns.names=['item']
print(frame)
#adding new column to the existing data frames
frame['count']=10
print(frame)
frame['count']=[10,12,15,16,18]
print(frame)
ser=pd.Series(np.arange(10,16))
frame['newser']=ser
print(frame)
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8

item  colors object  price
idx
0        red   ball    1.5
1      yelow    pen    2.4
2      green   book    1.3
3       blue  scale    3.6
4     orange    mug    4.8
item  colors object  price  count
idx
0        red   ball    1.5     10
1      yelow    pen    2.4     10
2      green   book    1.3     10
3       blue  scale    3.6     10
4     orange    mug    4.8     10
item  colors object  price  count
idx
0        red   ball    1.5     10
1      yelow    pen    2.4     12
2      green   book    1.3     15
3       blue  scale    3.6     16
4     orange    mug    4.8     18
item  colors object  price  count  newser
idx
0        red   ball    1.5     10      10
1      yelow    pen    2.4     12      11
2      green   book    1.3     15      12
3       blue  scale    3.6     16      13
4     orange    mug    4.8     18      14
```

In [66]:

```
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
frame['count']=[10,12,15,16,18]
print(frame)
del frame['count']
print(frame)
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
   colors object  price  count
0     red   ball    1.5     10
1   yelow    pen    2.4     12
2   green   book    1.3     15
3    blue  scale    3.6     16
4  orange    mug    4.8     18
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
```

In [72]:

```
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
print(frame.isin([1.5,'pen']))
print(frame[frame.isin([1.5,'pen'])])
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
   colors  object  price
0   False   False   True
1   False    True  False
2   False   False  False
3   False   False  False
4   False   False  False
  colors object  price
0    NaN    NaN    1.5
1    NaN    pen    NaN
2    NaN    NaN    NaN
3    NaN    NaN    NaN
4    NaN    NaN    NaN
```

In [4]:

```
#filtering values
import pandas as pd
import numpy as np
data={'colors':['red','yelow','green','blue','orange'],
     'object':['ball','pen','book','scale','mug'],
     'price':[1.5,2.4,1.3,3.6,4.8]}
frame=pd.DataFrame(data)
print(frame)
#print(frame[frame > 2.5])
```

```
   colors object  price
0     red   ball    1.5
1   yelow    pen    2.4
2   green   book    1.3
3    blue  scale    3.6
4  orange    mug    4.8
```

In [11]:

```
#nested dictionary
nestdict={'red':{2011:12,2013:15},
          'blue':{2011:14,2012:13,2013:16},
          'green':{2011:10,2012:17,2013:18}}
frame1=pd.DataFrame(nestdict)
print(frame1)
#transposition of frame
print(frame1.T)
print(frame1.index.is_unique)
```

```
       red  blue  green
2011  12.0    14     10
2013  15.0    16     18
2012   NaN    13     17
       2011  2013  2012
red    12.0  15.0   NaN
blue   14.0  16.0  13.0
green  10.0  18.0  17.0
True
```

In [6]:

```
ser1=pd.Series([5,0,3,8,4],index=['red','blue','yellow','white','green'])
print(ser1)
print(ser1.idxmin())
print(ser1.idxmax())
```

```
red       5
blue      0
yellow    3
white     8
green     4
dtype: int64
blue
white
```

In [10]:

```
ser2=pd.Series(np.arange(6),index=['white','white','blue','green','green','yellow'])
print(ser2)
print(ser2['white'])
print(ser2.index.is_unique)
```

```
white     0
white     1
blue      2
green     3
green     4
yellow    5
dtype: int32
white    0
white    1
dtype: int32
False
```

In [12]:

```
#other functionality on indexing
#reindexing
ser3=pd.Series([2,5,7,4],index=['one','two','three','four'])
print(ser3)
ser3.reindex(['three','four','five','one'])
```

```
one      2
two      5
three    7
four     4
dtype: int64
```

Out[12]:

```
three    7.0
four     4.0
five     NaN
one      2.0
dtype: float64
```

In [13]:

```
ser4=pd.Series([1,5,6,3],index=[0,3,5,6])
print(ser4)
print(ser4.reindex(range(6)))
#using ffill or bfill
print(ser4.reindex(range(6),method='ffill'))
print(ser4.reindex(range(6),method='bfill'))
```

```
0    1
3    5
5    6
6    3
dtype: int64
0    1.0
1    NaN
2    NaN
3    5.0
4    NaN
5    6.0
dtype: float64
0    1
1    1
2    1
3    5
4    5
5    6
dtype: int64
0    1
1    5
2    5
3    5
4    6
5    6
dtype: int64
```

In [1]:

```
#on dataframes
import pandas as pd
data={'color':['red','green','blue'],
     'object':['pen','ball','pencil'],
     'price':[20,15,14]}
frame3=pd.DataFrame(data)
print(frame3)
print(frame3.reindex(range(3),columns=['color','count','object','price']))
print(frame3.reindex(range(3),method='ffill',columns=['color','count','object','price'
]))
```

```
   color  object  price
0    red     pen     20
1  green    ball     15
2   blue  pencil     14
   color  count  object  price
0    red    NaN     pen     20
1  green    NaN    ball     15
2   blue    NaN  pencil     14
   color  count  object  price
0    red    red     pen     20
1  green  green    ball     15
2   blue   blue  pencil     14
```

In [3]:

```
#dropping
import pandas as pd
import numpy as np
ser5=pd.Series(np.arange(4),index=['red','green','blue','white'])
print(ser5)
print('dropping index green')
print(ser5.drop('green'))
print(ser5.drop(['red','blue']))
```

```
red      0
green    1
blue     2
white    3
dtype: int32
dropping index green
red      0
blue     2
white    3
dtype: int32
green    1
white    3
dtype: int32
```

In [4]:

```
#dropping in dataframes
frame4=pd.DataFrame(np.arange(16).reshape(4,4),
                    index=['r','b','g','w'],
                    columns=['ball','pen','pencil','book'])
print(frame4)
print('dropping indexes blue , green')
print(frame4.drop(['b','g']))
print('dropping columns')
print(frame4.drop(['pen','pencil'],axis=1))
```

```
   ball  pen  pencil  book
r     0    1       2     3
b     4    5       6     7
g     8    9      10    11
w    12   13      14    15
dropping indexes blue , green
   ball  pen  pencil  book
r     0    1       2     3
w    12   13      14    15
dropping columns
   ball  book
r     0     3
b     4     7
g     8    11
w    12    15
```

In [7]:

```
#alignment
s1=pd.Series([3,2,5,1],index=['white','yellow','green','blue'])
s2=pd.Series([1,4,7,2,1],index=['white','yellow','black','green','brown'])
print('s1 series are:',s1,sep='\n')
print('s2 series are:',s2,sep='\n')
print('s1+s2',s1+s2,sep='\n')
```

```
s1 series are:
white     3
yellow    2
green     5
blue      1
dtype: int64
s2 series are:
white     1
yellow    4
black     7
green     2
brown     1
dtype: int64
s1+s2
black     NaN
blue      NaN
brown     NaN
green     7.0
white     4.0
yellow    6.0
dtype: float64
```

In [2]:

```
import pandas as pd
import numpy as np

frame5=pd.DataFrame(np.arange(16).reshape(4,4),index=['red','blue','yellow','green'],co
lumns=['ball','pen','pencil','book'])
print('frame5:',frame5,sep='\n')
frame6=pd.DataFrame(np.arange(12).reshape(4,3),index=['blue','green','white','yellow'],
columns=['pen','ball','mug'])
print('frame6:',frame6,sep='\n')
print('frame5+frame6:',(frame5+frame6),sep='\n')
```

```
frame5:
        ball  pen  pencil  book
red        0    1       2     3
blue       4    5       6     7
yellow     8    9      10    11
green     12   13      14    15
frame6:
        pen  ball  mug
blue      0     1    2
green     3     4    5
white     6     7    8
yellow    9    10   11
frame5+frame6:
        ball  book  mug   pen  pencil
blue     5.0   NaN  NaN   5.0     NaN
green   16.0   NaN  NaN  16.0     NaN
red      NaN   NaN  NaN   NaN     NaN
white    NaN   NaN  NaN   NaN     NaN
yellow  18.0   NaN  NaN  18.0     NaN
```

In [3]:

```
print(frame5.add(frame6))
```

```
        ball  book  mug   pen  pencil
blue     5.0   NaN  NaN   5.0     NaN
green   16.0   NaN  NaN  16.0     NaN
red      NaN   NaN  NaN   NaN     NaN
white    NaN   NaN  NaN   NaN     NaN
yellow  18.0   NaN  NaN  18.0     NaN
```

In [4]:

```
s5=pd.Series(np.arange(4),index=['ball','pen','pencil','book'])
print('s5:',s5,sep='\n')
print('frame5:',frame5,sep='\n')
print('frame5+s5:',(frame5+s5),sep='\n')
```

```
s5:
ball      0
pen       1
pencil    2
book      3
dtype: int32
frame5:
        ball  pen  pencil  book
red        0    1       2     3
blue       4    5       6     7
yellow     8    9      10    11
green     12   13      14    15
frame5+s5:
        ball  pen  pencil  book
red        0    2       4     6
blue       4    6       8    10
yellow     8   10      12    14
green     12   14      16    18
```

In [8]:

```
def fun(x):
    return x.max()-x.min()
frame=pd.DataFrame(np.arange(16).reshape(4,4))
print(frame)
print(frame.apply(fun,axis=0))#row wise
#print(frame)
```

```
    0   1   2   3
0   0   1   2   3
1   4   5   6   7
2   8   9  10  11
3  12  13  14  15
0    12
1    12
2    12
3    12
dtype: int64
```

In [12]:

```
fun=lambda.x : x.max()-x.min()
print(frame.apply(fun))
```

```
  File "<ipython-input-12-cb0b16fc1950>", line 1
    fun=lambda(x) : x.max()-x.min()
               ^
SyntaxError: invalid syntax
```

In [1]:

```
import pandas as pd
import numpy as np
frame1=pd.DataFrame([[1,3,5,7],[2,4,6,8],[1,5,6,2],[3,6,8,9]],
                    columns=['ball','pen','pencil','book'],
                    index=['red','green','yellow','blue'])
def create(x):
    return pd.Series([x.max(),x.min()],index=['max','min'])
frame1.apply(create)
```

Out[1]:

	ball	pen	pencil	book
max	3	6	8	9
min	1	3	5	2

In [4]:

```
def Findsum(y):
    return pd.Series(y.sum(),index=['sum'])
print(frame1.apply(Findsum))
```

```
     ball  pen  pencil  book
sum     7   18      25    26
```

In [3]:

```
import pandas as pd
import numpy as np
frame1=pd.DataFrame([[1,3,5,7],[2,4,6,8],[1,5,6,2],[3,6,8,9]],
                    columns=['ball','pen','pencil','book'],
                    index=['red','green','yellow','blue'])
print(frame1.sum())
print(frame1.mean())
print(frame1.describe())
```

```
ball       7
pen       18
pencil    25
book      26
dtype: int64
ball      1.75
pen       4.50
pencil    6.25
book      6.50
dtype: float64
           ball       pen    pencil      book
count  4.000000  4.000000  4.000000  4.000000
mean   1.750000  4.500000  6.250000  6.500000
std    0.957427  1.290994  1.258306  3.109126
min    1.000000  3.000000  5.000000  2.000000
25%    1.000000  3.750000  5.750000  5.750000
50%    1.500000  4.500000  6.000000  7.500000
75%    2.250000  5.250000  6.500000  8.250000
max    3.000000  6.000000  8.000000  9.000000
```

In [8]:

```
#sorting and ranking
ser1=pd.Series([5,0,3,8,4],index=['red','green','yellow','blue','white'])
print(ser1)
print(ser1.sort_index(ascending=True))
print(ser1.sort_index(ascending=False))
print(ser1.sort_values(ascending=True))
```

```
red       5
green     0
yellow    3
blue      8
white     4
dtype: int64
blue      8
green     0
red       5
white     4
yellow    3
dtype: int64
yellow    3
white     4
red       5
green     0
blue      8
dtype: int64
green     0
yellow    3
white     4
red       5
blue      8
dtype: int64
```

In [13]:

```
frame1=pd.DataFrame(np.arange(16).reshape(4,4),index=['red','blue','green','yellow'],
                    columns=['pen','ball','paper','pencil'])
print(frame1.sort_index(ascending=True))
print(frame1.sort_index(axis=1))
print(frame1.sort_values(by='ball'))
```

```
        pen  ball  paper  pencil
blue      4     5      6       7
green     8     9     10      11
red       0     1      2       3
yellow   12    13     14      15
        ball  paper  pen  pencil
red        1      2    0       3
blue       5      6    4       7
green      9     10    8      11
yellow    13     14   12      15
        pen  ball  paper  pencil
red       0     1      2       3
blue      4     5      6       7
green     8     9     10      11
yellow   12    13     14      15
```

In [15]:

```
#ranking
ser1=pd.Series([5,0,3,8,4],index=['red','green','yellow','blue','white'])
print(ser1)
print(ser1.rank())
print(ser1.rank(ascending=False))
```

```
red       5
green     0
yellow    3
blue      8
white     4
dtype: int64
red       4.0
green     1.0
yellow    2.0
blue      5.0
white     3.0
dtype: float64
red       2.0
green     5.0
yellow    4.0
blue      1.0
white     3.0
dtype: float64
```

In [25]:

```
frame=pd.DataFrame([[4,'fox'],[2,'kangaroo'],[4,'deer'],[8,'spider'],[np.nan,'snake']],
                   columns=['number_legs','animal'],index=[0,1,2,3,4])
print(frame)
frame['default_rank']=frame['number_legs'].rank()
print(frame)
```

```
   number_legs    animal
0          4.0       fox
1          2.0  kangaroo
2          4.0      deer
3          8.0    spider
4          NaN     snake
   number_legs    animal  default_rank
0          4.0       fox           2.5
1          2.0  kangaroo           1.0
2          4.0      deer           2.5
3          8.0    spider           4.0
4          NaN     snake           NaN
```

In [8]:

```
#covariance and correlation
import pandas as pd
import numpy as np
ser1=pd.Series([5,2,3],index=['red','green','blue'])
print(ser1)
print(ser1.var())
frame1=pd.DataFrame([[10,15,7,2,16],[13,0,7,4,1]],
                    index=['commercial watched','product purchase'])
print(frame1)
print(frame1.var())
print('covariance:',frame1.cov(),sep='\n')
print('correlation:',frame1.corr(),sep='\n')
```

```
red      5
green    2
blue     3
dtype: int64
2.3333333333333335
                     0   1  2  3   4
commercial watched  10  15  7  2  16
product purchase    13   0  7  4   1
0      4.5
1    112.5
2      0.0
3      2.0
4    112.5
dtype: float64
covariance:
      0      1    2     3      4
0   4.5  -22.5  0.0   3.0  -22.5
1 -22.5  112.5  0.0 -15.0  112.5
2   0.0    0.0  0.0   0.0    0.0
3   3.0  -15.0  0.0   2.0  -15.0
4 -22.5  112.5  0.0 -15.0  112.5
correlation:
     0    1   2    3    4
0  1.0 -1.0 NaN  1.0 -1.0
1 -1.0  1.0 NaN -1.0  1.0
2  NaN  NaN NaN  NaN  NaN
3  1.0 -1.0 NaN  1.0 -1.0
4 -1.0  1.0 NaN -1.0  1.0
```

In [12]:

```
import pandas as pd
import numpy as np
ser1=pd.Series([6,7,np.nan,4],index=['red','green','blue','yellow'])
print('series 1:',ser1,sep='\n')
ser1['green']=None
print(ser1)
print(ser1.dropna())
#print(ser1)
```

```
series 1:
red       6.0
green     7.0
blue      NaN
yellow    4.0
dtype: float64
red       6.0
green     NaN
blue      NaN
yellow    4.0
dtype: float64
red       6.0
yellow    4.0
dtype: float64
```

In [17]:

```
# using how option
frame1=pd.DataFrame([[6,np.nan,4.0],[np.nan,np.nan,np.nan],[2.2,np.nan,5]],index=['red'
,'green','blue'],
                    columns=['ball','pen','pencil'])
print('frame1:',frame1,sep='\n')
#print(frame1.dropna(how=all))
#filling the nan position values using fillna()
frame1.fillna(0)
```

```
frame1:
       ball  pen  pencil
red     6.0  NaN     4.0
green   NaN  NaN     NaN
blue    2.2  NaN     5.0
```

Out[17]:

	ball	pen	pencil
red	6.0	0.0	4.0
green	0.0	0.0	0.0
blue	2.2	0.0	5.0

In [21]:

```
#hierarchical indexing and levelling
mser=pd.Series(np.random.random(8),
               index=[['w','w','w','b','b','r','r','r'],
                      ['up','down','left','up','down','up','down','left']])
print(mser)
print('indexes:',mser.index,sep='\n')
print(mser['w'])
print(mser['w','up'])
print(mser[:,'up'])
print('converting series into dataframe using unstack() method')
print(mser.unstack())
```

```
w  up      0.694212
   down    0.405391
   left    0.351115
b  up      0.200439
   down    0.154843
r  up      0.728985
   down    0.644616
   left    0.625199
dtype: float64
indexes:
MultiIndex([('w',   'up'),
            ('w', 'down'),
            ('w', 'left'),
            ('b',   'up'),
            ('b', 'down'),
            ('r',   'up'),
            ('r', 'down'),
            ('r', 'left')],
           )
up      0.694212
down    0.405391
left    0.351115
dtype: float64
0.6942119950056037
w    0.694212
b    0.200439
r    0.728985
dtype: float64
converting series into dataframe using unstack() method
       down      left        up
b  0.154843       NaN  0.200439
r  0.644616  0.625199  0.728985
w  0.405391  0.351115  0.694212
```

In [22]:

```
mframe=pd.DataFrame(np.arange(16).reshape(4,4),
                    index=['red','blue','yellow','white'],
                    columns=['ball','pen','pencil','book'])
print('mframe:',mframe,sep='\n')
print('converting dataframe into series using stack() method')
print(mframe.stack())
```

```
mframe:
        ball  pen  pencil  book
red        0    1       2     3
blue       4    5       6     7
yellow     8    9      10    11
white     12   13      14    15
converting dataframe into series using stack() method
red     ball       0
        pen        1
        pencil     2
        book       3
blue    ball       4
        pen        5
        pencil     6
        book       7
yellow  ball       8
        pen        9
        pencil    10
        book      11
white   ball      12
        pen       13
        pencil    14
        book      15
dtype: int32
```

In [30]:

```
#creating hierachical indexing for dataframe
import pandas as pd
import numpy as np
mframe=pd.DataFrame(np.arange(16).reshape(4,4),
                    index=[['white','white','red','red'],['up','down','up','down']],
                    columns=[['pen','pen','pencil','pencil'],[1,2,1,2]])
print('mframe:',mframe,sep='\n')
#adding headers to the indexes
mframe.index.names=['colors','direction']
mframe.columns.names=['object','id']
print(mframe)
#swapping of indexes
print(mframe.swaplevel('colors','direction'))
#sorting values
mframe.sort_index(level='colors')  #error coming
#summation
print(mframe.sum(level='colors'))
```

```
mframe:
           pen     pencil
             1   2      1   2
white up     0   1      2   3
      down   4   5      6   7
red   up     8   9     10  11
      down  12  13     14  15
object           pen     pencil
id                 1   2      1   2
colors direction
white  up          0   1      2   3
       down        4   5      6   7
red    up          8   9     10  11
       down       12  13     14  15
object           pen     pencil
id                 1   2      1   2
direction colors
up        white    0   1      2   3
down      white    4   5      6   7
up        red      8   9     10  11
down      red     12  13     14  15
object pen     pencil
id       1   2      1   2
colors
white    4   6      8  10
red     20  22     24  26
```

In [17]:

```
csvfile=pd.read_csv('data.csv')
print(csvfile)
```

```
     id      name  marks
0  3001    afreen    9.0
1  3002  keerthi     9.5
2  3003     priya    9.3
```

In [18]:

```
csvfile=pd.read_csv('data.csv',sep=',')
print(csvfile)
```

```
     id     name  marks
0  3001   afreen    9.0
1  3002  keerthi    9.5
2  3003    priya    9.3
```

In [19]:

```
csvfile=pd.read_csv('data.csv',names=['sid','snames'])
print(csvfile)
```

```
          sid snames
id       name  marks
3001   afreen      9
3002  keerthi    9.5
3003    priya    9.3
```

In [21]:

```
import pandas as pd
csvfile=pd.read_csv('sample2.csv',index_col=['color','direction'])
print(csvfile)
```

```
                it1  it2
color direction
red   up          1    2
      down        2    3
white up          3    4
      left        5    6
      down        7    8
```

In [22]:

```
txtfile=pd.read_table('mydata.txt',sep='\s+')
print(txtfile)
```

```
   rollno name  percent
0    3001    X       93
1    3002    Y       95
2    3003    Z       91
```

In [23]:

```
txtfile=pd.read_table('mydata.txt',sep='\D+')
print(txtfile)
```

```
   Unnamed: 0  Unnamed: 1
0        3001          93
1        3002          95
2        3003          91

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: Parser
Warning: Falling back to the 'python' engine because the 'c' engine does n
ot support regex separators (separators > 1 char and different from '\s+'
are interpreted as regex); you can avoid this warning by specifying engine
='python'.
  """Entry point for launching an IPython kernel.
```

In [ ]: