import Libraries

In [4]:

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

Check pandas version

In [3]:

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

```
1.2.4
```

Create empty series

In [5]:

```
a=pd.Series()
a
```

Out[5]:

```
Series([], dtype: float64)
```

Create series from numpy array

In [31]:

```
import pandas as pd
import numpy as np
a=np.array([1,2,3,4,5,6,7])
v=pd.Series(a)
v
```

Out[31]:

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
dtype: int32
```

Create series from list

In [11]:

```
s=pd.Series([1,2,3],index=['1','b','c'])
s
```

Out[11]:

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

Modifying Index

In [32]:

```
import pandas as pd
import numpy as np
a=np.array([1,2,3,4,5,6,7])
v=pd.Series(a)
v.index=['a','b','c','d','e','f','g']
v
```

Out[32]:

```
a    1
b    2
c    3
d    4
e    5
f    6
g    7
dtype: int32
```

Create series using random and range function

In [33]:

```
import pandas as pd
import numpy as np
v1=np.random.random(12)
v1
```

Out[33]:

```
array([0.15227061, 0.22354497, 0.40052103, 0.65282548, 0.00868966,
       0.78623224, 0.82786261, 0.99576551, 0.4563993 , 0.6094402 ,
       0.46764059, 0.66755303])
```

In [34]:

```
import pandas as pd
import numpy as np
v2=np.arange(2,14)
v2
```

Out[34]:

```
array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

Create series from dictionary

In [35]:

```
import pandas as pd
import numpy as np
d={'a':10,'b':20,'c':30}
s1=pd.Series(d)
s1
```

Out[35]:

```
a    10
b    20
c    30
dtype: int64
```

Series object Attributes

In [37]:

```
import pandas as pd
import numpy as np
a=np.array([10,22,31,42,55,36,47])
v=pd.Series(a,index=['a','b','c','d','e','f','g'])
v
```

Out[37]:

```
a    10
b    22
c    31
d    42
e    55
f    36
g    47
dtype: int32
```

Shape of the series

In [38]:

```
v.shape
```

Out[38]:

```
(7,)
```

Size of the series

In [39]:

```
v.size
```

Out[39]:

```
7
```

Number of the series

In [40]:

```
v.ndim
```

Out[40]:

```
1
```

Number of bytes

In [41]:

```
v.nbytes
```

Out[41]:

```
28
```

To check whether null is present in series

In [42]:

```
v.hasnans
```

Out[42]:

```
False
```

To check your series is empty or not

In [43]:

```
v.empty
```

Out[43]:

```
False
```

To print SeriesValues

In [44]:

```
v.values
```

Out[44]:

```
array([10, 22, 31, 42, 55, 36, 47])
```

series function

In [45]:

```
s=pd.Series([1,2,1,3,3,4,2,5])
s
```

Out[45]:

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

In [46]:

```
s.value_counts()
```

Out[46]:

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

Append Serie

In [48]:

```
s2=s1.copy()
s2
```

Out[48]:

```
a    10
b    20
c    30
dtype: int64
```

In [49]:

```
d={'e':50,'f':60,'g':70,'h':80}
s3=pd.Series(d)
s3
```

Out[49]:

```
e    50
f    60
g    70
h    80
dtype: int64
```

In [50]:

```
s4=s2.append(s3)
s4
```

Out[50]:

```
a    10
b    20
c    30
e    50
f    60
g    70
h    80
dtype: int64
```

Operation on series

In [51]:

```
s1
```

Out[51]:

```
a    10
b    20
c    30
dtype: int64
```

In [52]:

```
s2
```

Out[52]:

```
a    10
b    20
c    30
dtype: int64
```

In [54]:

```
s1.add(s2)
```

Out[54]:

```
a    20
b    40
c    60
dtype: int64
```

In [55]:

```
x=np.array([10,20,30,40])
y=np.array([1,2,3,4])
s1=pd.Series(x)
s2=pd.Series(y)
s1,s2
```

Out[55]:

```
(0    10
 1    20
 2    30
 3    40
 dtype: int32,
 0    1
 1    2
 2    3
 3    4
 dtype: int32)
```

Addition of two series

In [56]:

```
s1.add(s2)
```

Out[56]:

```
0    11
1    22
2    33
3    44
dtype: int32
```

Subtraction of two series

In [57]:

```
s1.subtract(s2)
```

Out[57]:

```
0     9
1    18
2    27
3    36
dtype: int32
```

Multiplication of two series

In [58]:

```
s1.mul(s2)
```

Out[58]:

```
0     10
1     40
2     90
3    160
dtype: int32
```

In [ ]:

```
### Division of two series
```

In [59]:

```
s1.div(s2)
```

Out[59]:

```
0    10.0
1    10.0
2    10.0
3    10.0
dtype: float64
```

Maximum number in a series

In [60]:

```
s1.max()
```

Out[60]:

```
40
```

minimum number in a series

In [63]:

```
s1.min()
```

Out[63]:

```
10
```

Average

In [64]:

```
s1.median()
```

Out[64]:

```
25.0
```

Standard deviation

In [65]:

```
s1.std()
```

Out[65]:

```
12.909944487358056
```

increment all numbers in a series by 9

In [66]:

```
s1.add(9)
```

Out[66]:

```
0    19
1    29
2    39
3    49
dtype: int32
```

Series Comparison

In [67]:

```
s1.equals(s2)
```

Out[67]:

```
False
```

In [69]:

```
s5=pd.Series([1,2,1,3,3,4,2,5])
s5
```

Out[69]:

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

Data frame

Create an empty DatFrame

In [3]:

```
import pandas as pd
import numpy as np
a=pd.DataFrame()
print(a)
```

```
Empty DataFrame
Columns: []
Index: []
```

Create DataFrame from List

In [5]:

```
import pandas as pd
import numpy as np
l=['lingesh','bhava','vino']
d=pd.DataFrame(l)
d
```

Out[5]:

	0
0	lingesh
1	bhava
2	vino

Add column in the data Frame

In [17]:

```
import pandas as pd
import numpy as np
l=['lingesh','bhava','vino']
marks=[67,54,89]
d=pd.DataFrame(l)
d[1]=marks
d
```

Out[17]:

	0	lingesh	bhava	vino
0	lingesh	67	54	89
1	bhava	67	54	89
2	vino	67	54	89

Create DataFrame from Dictionory

In [56]:

```
d1=pd.DataFrame({'ID':[11,22,33,44],"name":['karan','lingesh','bava','vino']})
d1
```

Out[56]:

	ID	name
0	11	karan
1	22	lingesh
2	33	bava
3	44	vino

Creating DataFrame from random values

In [57]:

```
r=np.random.random((7,7))
r
```

Out[57]:

```
array([[0.73593395, 0.06693997, 0.1689812 , 0.56138682, 0.37333172,
        0.96228273, 0.83055202],
       [0.37606478, 0.55200019, 0.19645179, 0.40212443, 0.41186171,
        0.20004046, 0.21376193],
       [0.93325286, 0.20145558, 0.86292302, 0.47827038, 0.68088715,
        0.80530703, 0.26104669],
       [0.86751398, 0.68957567, 0.85310338, 0.75041484, 0.00773862,
        0.96477139, 0.1487256 ],
       [0.87098116, 0.43786519, 0.51224732, 0.89623284, 0.97015428,
        0.48090437, 0.26737661],
       [0.06219461, 0.69812218, 0.28850359, 0.32687628, 0.55968557,
        0.47640219, 0.80817562],
       [0.48562019, 0.43185358, 0.26008507, 0.62844813, 0.89198269,
        0.71854643, 0.54027985]])
```

In [58]:

```
d2=pd.DataFrame(r)
d2
```

Out[58]:

	0	1	2	3	4	5	6
0	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
1	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047
3	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
4	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
5	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176
6	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280

In [59]:

```
date1=pd.date_range('today',periods=7)
date1
```

Out[59]:

```
DatetimeIndex(['2021-07-15 10:56:07.012608', '2021-07-16 10:56:07.012608',
               '2021-07-17 10:56:07.012608', '2021-07-18 10:56:07.012608',
               '2021-07-19 10:56:07.012608', '2021-07-20 10:56:07.012608',
               '2021-07-21 10:56:07.012608'],
              dtype='datetime64[ns]', freq='D')
```

In [19]:

```
date2=pd.date_range(start='2021-01-20',end='2021-01-25')
date2
```

Out[19]:

```
DatetimeIndex(['2021-01-20', '2021-01-21', '2021-01-22', '2021-01-23',
               '2021-01-24', '2021-01-25'],
              dtype='datetime64[ns]', freq='D')
```

In [60]:

```
date3=pd.date_range(start='2021-01-20',periods=7)
date3
```

Out[60]:

```
DatetimeIndex(['2021-01-20', '2021-01-21', '2021-01-22', '2021-01-23',
               '2021-01-24', '2021-01-25', '2021-01-26'],
              dtype='datetime64[ns]', freq='D')
```

In [61]:

```
d2
```

Out[61]:

	0	1	2	3	4	5	6
0	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
1	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047
3	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
4	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
5	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176
6	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280

In [62]:

```
d2=pd.DataFrame(r,date3)
d2
```

Out[62]:

	0	1	2	3	4	5	6
2021-01-20	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
2021-01-21	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2021-01-22	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047
2021-01-23	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
2021-01-24	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
2021-01-25	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176
2021-01-26	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280

In [63]:

```
#Changing column names
d2.columns=['A','B','C','D','E','F','G']
d2
```

Out[63]:

	A	B	C	D	E	F	G
2021-01-20	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
2021-01-21	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2021-01-22	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047
2021-01-23	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
2021-01-24	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
2021-01-25	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176
2021-01-26	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280

In [64]:

```
#List index
d2.index
```

Out[64]:

```
DatetimeIndex(['2021-01-20', '2021-01-21', '2021-01-22', '2021-01-23',
               '2021-01-24', '2021-01-25', '2021-01-26'],
              dtype='datetime64[ns]', freq='D')
```

In [65]:

```
#list column names
d2.columns
```

Out[65]:

```
Index(['A', 'B', 'C', 'D', 'E', 'F', 'G'], dtype='object')
```

In [66]:

```
#datatypr of each column
d2.dtypes
```

Out[66]:

```
A    float64
B    float64
C    float64
D    float64
E    float64
F    float64
G    float64
dtype: object
```

In [67]:

```
d2.sort_values(by='A',ascending=False)
```

Out[67]:

	A	B	C	D	E	F	G
2021-01-22	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047
2021-01-24	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
2021-01-23	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
2021-01-20	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
2021-01-26	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280
2021-01-21	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2021-01-25	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176

In [68]:

```
d2.sort_values(by='A')
```

Out[68]:

	A	B	C	D	E	F	G
2021-01-25	0.062195	0.698122	0.288504	0.326876	0.559686	0.476402	0.808176
2021-01-21	0.376065	0.552000	0.196452	0.402124	0.411862	0.200040	0.213762
2021-01-26	0.485620	0.431854	0.260085	0.628448	0.891983	0.718546	0.540280
2021-01-20	0.735934	0.066940	0.168981	0.561387	0.373332	0.962283	0.830552
2021-01-23	0.867514	0.689576	0.853103	0.750415	0.007739	0.964771	0.148726
2021-01-24	0.870981	0.437865	0.512247	0.896233	0.970154	0.480904	0.267377
2021-01-22	0.933253	0.201456	0.862923	0.478270	0.680887	0.805307	0.261047

Delete column in dataframe

In [76]:

```
d2.pop('A')
```

Out[76]:

```
2021-01-20    0.735934
2021-01-21    0.376065
2021-01-22    0.933253
2021-01-23    0.867514
2021-01-24    0.870981
2021-01-25    0.062195
2021-01-26    0.485620
Freq: D, Name: A, dtype: float64
```

In [77]:

```
d2.drop(['B'],axis=1) #axis=1 column axis=0 Row
```

Out[77]:

	D	E	F	G
2021-01-20	0.561387	0.373332	0.962283	0.830552
2021-01-21	0.402124	0.411862	0.200040	0.213762
2021-01-22	0.478270	0.680887	0.805307	0.261047
2021-01-23	0.750415	0.007739	0.964771	0.148726
2021-01-24	0.896233	0.970154	0.480904	0.267377
2021-01-25	0.326876	0.559686	0.476402	0.808176
2021-01-26	0.628448	0.891983	0.718546	0.540280

In [ ]:

```
### Delete rows in dataframe
```

In [80]:

```
r=np.random.random((10,10))
```

In [81]:

```
r
```

Out[81]:

```
array([[0.5851492 , 0.22686157, 0.49736343, 0.30759104, 0.06998271,
        0.29774646, 0.38150993, 0.54952246, 0.41601384, 0.69670761],
       [0.4442793 , 0.30154244, 0.93848347, 0.26152524, 0.34153239,
        0.52612934, 0.90679981, 0.97128581, 0.06110662, 0.91063126],
       [0.69276594, 0.22984093, 0.95595425, 0.68789311, 0.37204466,
        0.95663451, 0.30688614, 0.98066279, 0.78008831, 0.04175758],
       [0.91209564, 0.90550213, 0.40046344, 0.65198329, 0.63623888,
        0.71367404, 0.90318885, 0.97421629, 0.28519191, 0.09594647],
       [0.60629538, 0.70297879, 0.92183499, 0.35166536, 0.6154441 ,
        0.66121358, 0.79458395, 0.35619597, 0.76241675, 0.27141307],
       [0.97692686, 0.16038571, 0.88020947, 0.77457864, 0.13827866,
        0.69946154, 0.80251796, 0.00105814, 0.40975031, 0.27949456],
       [0.92845527, 0.34688981, 0.4710215 , 0.35744841, 0.66544908,
        0.40836073, 0.72681061, 0.88856428, 0.68498368, 0.49797085],
       [0.91573932, 0.3295838 , 0.72737515, 0.09785795, 0.84133392,
        0.81247263, 0.52630273, 0.5003011 , 0.48208062, 0.77047882],
       [0.32886666, 0.60796895, 0.33752266, 0.54317831, 0.77156032,
        0.10491056, 0.58444331, 0.30729772, 0.42536616, 0.00410018],
       [0.75653615, 0.02893232, 0.96378307, 0.4620447 , 0.7130324 ,
        0.23569088, 0.14585279, 0.28172053, 0.30265474, 0.26945696]])
```

In [87]:

```
d2=d2.iloc[2:,]
d2
```

Out[87]:

	B	D	E	F	G
2021-01-22	0.201456	0.478270	0.680887	0.805307	0.261047
2021-01-23	0.689576	0.750415	0.007739	0.964771	0.148726
2021-01-24	0.437865	0.896233	0.970154	0.480904	0.267377
2021-01-25	0.698122	0.326876	0.559686	0.476402	0.808176
2021-01-26	0.431854	0.628448	0.891983	0.718546	0.540280

In [96]:

```
d2=d2.iloc[:-2,]
d2
```

Out[96]:

B	D	E	F	G

Data Selection

In [98]:

```
d1=pd.DataFrame({'ID':[1,2,3,4],'name':['lingesh','bhava','karan','vino']})
d1
```

Out[98]:

	ID	name
0	1	lingesh
1	2	bhava
2	3	karan
3	4	vino

In [106]:

```
d1.index=['A','B','C','D']
d1
```

Out[106]:

	ID	name
A	1	lingesh
B	2	bhava
C	3	karan
D	4	vino

In [107]:

```
#data selection using roq lable
d1.loc['C']
```

Out[107]:

```
ID          3
name    karan
Name: C, dtype: object
```

In [110]:

```
#data selection using postion
d1.iloc[1]
```

Out[110]:

```
ID          2
name    bhava
Name: B, dtype: object
```

In [112]:

```
d1.iloc[1:2]
```

Out[112]:

	ID	name
B	2	bhava

In [113]:

```
d1.loc['A':'C']
```

Out[113]:

	ID	name
A	1	lingesh
B	2	bhava
C	3	karan

In [114]:

```
d1.loc[d1.ID>2]
```

Out[114]:

	ID	name
C	3	karan
D	4	vino

set value

In [115]:

```
date3=pd.date_range(start='2021-01-25',periods=7)
date3
```

Out[115]:

```
DatetimeIndex(['2021-01-25', '2021-01-26', '2021-01-27', '2021-01-28',
               '2021-01-29', '2021-01-30', '2021-01-31'],
              dtype='datetime64[ns]', freq='D')
```

In [116]:

```
r=np.random.random((7,7))
r
```

Out[116]:

```
array([[0.52529565, 0.6009938 , 0.3013396 , 0.48208897, 0.60279196,
        0.67662083, 0.17203643],
       [0.42214531, 0.58918674, 0.39837725, 0.09402338, 0.65609609,
        0.07907863, 0.09655488],
       [0.89050198, 0.3680993 , 0.83198866, 0.85096285, 0.64907354,
        0.60965739, 0.52213724],
       [0.91067376, 0.05534266, 0.16624702, 0.85739911, 0.14455935,
        0.37463935, 0.9452208 ],
       [0.54390841, 0.26519849, 0.49045849, 0.76602306, 0.5617203 ,
        0.53950934, 0.58584415],
       [0.0348189 , 0.86410701, 0.08322228, 0.95745177, 0.64191559,
        0.42335389, 0.31972798],
       [0.36962612, 0.47517   , 0.26737896, 0.35578518, 0.94472105,
        0.16699954, 0.06844808]])
```

In [117]:

```
d2=pd.DataFrame(r)
d2
```

Out[117]:

	0	1	2	3	4	5	6
0	0.525296	0.600994	0.301340	0.482089	0.602792	0.676621	0.172036
1	0.422145	0.589187	0.398377	0.094023	0.656096	0.079079	0.096555
2	0.890502	0.368099	0.831989	0.850963	0.649074	0.609657	0.522137
3	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
4	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
5	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
6	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [119]:

```
d2=pd.DataFrame(r,date3)
d2
```

Out[119]:

	0	1	2	3	4	5	6
2021-01-25	0.525296	0.600994	0.301340	0.482089	0.602792	0.676621	0.172036
2021-01-26	0.422145	0.589187	0.398377	0.094023	0.656096	0.079079	0.096555
2021-01-27	0.890502	0.368099	0.831989	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [121]:

```
d2.columns=['A','B','C','D','E','F','G']
d2
```

Out[121]:

	A	B	C	D	E	F	G
2021-01-25	0.525296	0.600994	0.301340	0.482089	0.602792	0.676621	0.172036
2021-01-26	0.422145	0.589187	0.398377	0.094023	0.656096	0.079079	0.096555
2021-01-27	0.890502	0.368099	0.831989	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [ ]:

```
d2.at[0:3,'C',]=90
```

In [127]:

```
d2
```

Out[127]:

	A	B	C	D	E	F	G
2021-01-25	0.525296	0.600994	90.000000	0.482089	0.602792	0.676621	0.172036
2021-01-26	0.422145	0.589187	90.000000	0.094023	0.656096	0.079079	0.096555
2021-01-27	0.890502	0.368099	90.000000	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [ ]:

```
d2.at[0:3,'A',]=80
```

In [130]:

```
d2
```

Out[130]:

	A	B	C	D	E	F	G
2021-01-25	80.000000	0.600994	90.000000	0.482089	0.602792	0.676621	0.172036
2021-01-26	80.000000	0.589187	90.000000	0.094023	0.656096	0.079079	0.096555
2021-01-27	80.000000	0.368099	90.000000	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [133]:

```
d2.iat[0,4]=100
d2
```

Out[133]:

	A	B	C	D	E	F	G
2021-01-25	80.000000	0.600994	90.000000	0.482089	100.000000	0.676621	0.172036
2021-01-26	80.000000	0.589187	90.000000	0.094023	0.656096	0.079079	0.096555
2021-01-27	80.000000	0.368099	90.000000	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [134]:

```
d2[d2.isin([100])]=88
d2
```

Out[134]:

	A	B	C	D	E	F	G
2021-01-25	80.000000	0.600994	90.000000	0.482089	88.000000	0.676621	0.172036
2021-01-26	80.000000	0.589187	90.000000	0.094023	0.656096	0.079079	0.096555
2021-01-27	80.000000	0.368099	90.000000	0.850963	0.649074	0.609657	0.522137
2021-01-28	0.910674	0.055343	0.166247	0.857399	0.144559	0.374639	0.945221
2021-01-29	0.543908	0.265198	0.490458	0.766023	0.561720	0.539509	0.585844
2021-01-30	0.034819	0.864107	0.083222	0.957452	0.641916	0.423354	0.319728
2021-01-31	0.369626	0.475170	0.267379	0.355785	0.944721	0.167000	0.068448

In [ ]:

```
###Display all rows where values of 'A' is 88
```

In [136]:

```
d2[d2['E']==88]
```

Out[136]:

	A	B	C	D	E	F	G
2021-01-25	80.0	0.600994	90.0	0.482089	88.0	0.676621	0.172036

In [3]:

```
r=np.random.random((7,7))
r
```

Out[3]:

```
array([[0.31619334, 0.61572773, 0.13849946, 0.29725678, 0.05110454,
        0.61010769, 0.64198968],
       [0.59478778, 0.55120288, 0.18201075, 0.70746043, 0.46694797,
        0.24195358, 0.12804741],
       [0.26127482, 0.07741765, 0.49703598, 0.20688676, 0.49337188,
        0.56503398, 0.70402835],
       [0.66784187, 0.66244165, 0.17039492, 0.61302391, 0.79473167,
        0.68089998, 0.5657938 ],
       [0.09034335, 0.941152  , 0.94658351, 0.77925655, 0.25209755,
        0.73017521, 0.74170978],
       [0.7003773 , 0.04886989, 0.79320954, 0.63075237, 0.30824291,
        0.85586016, 0.26689787],
       [0.46209453, 0.36599106, 0.18518399, 0.39036008, 0.29403508,
        0.76413182, 0.68384452]])
```

In [4]:

```
d2=pd.DataFrame(r)
d2
```

Out[4]:

	0	1	2	3	4	5	6
0	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
1	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
3	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
4	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
5	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
6	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [5]:

```
d2.columns=['A','B','C','D','E','F','G']
d2
```

Out[5]:

	A	B	C	D	E	F	G
0	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
1	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
3	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
4	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
5	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
6	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [6]:

```
d2.describe()
```

Out[6]:

	A	B	C	D	E	F	G
count	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000
mean	0.441845	0.466115	0.416131	0.517857	0.380076	0.635452	0.533187
std	0.228642	0.323788	0.335591	0.218976	0.234338	0.198586	0.239205
min	0.090343	0.048870	0.138499	0.206887	0.051105	0.241954	0.128047
25%	0.288734	0.221704	0.176203	0.343808	0.273066	0.587571	0.416346
50%	0.462095	0.551203	0.185184	0.613024	0.308243	0.680900	0.641990
75%	0.631315	0.639085	0.645123	0.669106	0.480160	0.747154	0.693936
max	0.700377	0.941152	0.946584	0.779257	0.794732	0.855860	0.741710

In [7]:

```
d2.sum()
```

Out[7]:

```
A    3.092913
B    3.262803
C    2.912918
D    3.624997
E    2.660532
F    4.448162
G    3.732311
dtype: float64
```

In [8]:

```
d2.quantile(0.75)-d2.quantile(0.25)
```

Out[8]:

```
A    0.342581
B    0.417380
C    0.468920
D    0.325298
E    0.207094
F    0.159583
G    0.277591
dtype: float64
```

In [9]:

```
#correlation
d2.corr()
```

Out[9]:

	A	B	C	D	E	F	G
A	1.000000	-0.378452	-0.357772	0.230023	0.474518	-0.028333	-0.725890
B	-0.378452	1.000000	-0.035855	0.499564	-0.058570	-0.157675	0.251514
C	-0.357772	-0.035855	1.000000	0.405939	-0.205229	0.465001	0.095212
D	0.230023	0.499564	0.405939	1.000000	0.203578	-0.036630	-0.463293
E	0.474518	-0.058570	-0.205229	0.203578	1.000000	-0.193420	-0.174331
F	-0.028333	-0.157675	0.465001	-0.036630	-0.193420	1.000000	0.444051
G	-0.725890	0.251514	0.095212	-0.463293	-0.174331	0.444051	1.000000

In [10]:

```
# covaiance
d2.cov()
```

Out[10]:

	A	B	C	D	E	F	G
A	0.052277	-0.028017	-0.027452	0.011517	0.025424	-0.001286	-0.039701
B	-0.028017	0.104839	-0.003896	0.035420	-0.004444	-0.010138	0.019480
C	-0.027452	-0.003896	0.112621	0.029831	-0.016140	0.030989	0.007643
D	0.011517	0.035420	0.029831	0.047951	0.010447	-0.001593	-0.024267
E	0.025424	-0.004444	-0.016140	0.010447	0.054914	-0.009001	-0.009772
F	-0.001286	-0.010138	0.030989	-0.001593	-0.009001	0.039436	0.021094
G	-0.039701	0.019480	0.007643	-0.024267	-0.009772	0.021094	0.057219

In [11]:

```
x=np.percentile(d2,0.25)
x
```

Out[11]:

```
0.04913804690879462
```

Standrand deviation

In [12]:

```
#sample standard deviation
import statistics as st
st.stdev(d2['B'])
```

Out[12]:

```
0.32378803135814394
```

In [13]:

```
#population standard deviation
st.pstdev(d2['B'])
```

Out[13]:

```
0.2997694674971549
```

Apply Function

In [14]:

```
d2
```

Out[14]:

	A	B	C	D	E	F	G
0	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
1	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
3	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
4	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
5	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
6	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [15]:

```
#finding max value in column
d2.apply(max)
```

Out[15]:

```
A    0.700377
B    0.941152
C    0.946584
D    0.779257
E    0.794732
F    0.855860
G    0.741710
dtype: float64
```

In [16]:

```
#finding min value in column
d2.apply(min)
```

Out[16]:

```
A    0.090343
B    0.048870
C    0.138499
D    0.206887
E    0.051105
F    0.241954
G    0.128047
dtype: float64
```

In [17]:

```
#sum of column values
d2.apply(sum)
```

Out[17]:

```
A    3.092913
B    3.262803
C    2.912918
D    3.624997
E    2.660532
F    4.448162
G    3.732311
dtype: float64
```

In [18]:

```
#sum of rows
d2.apply(np.sum,axis=1)
```

Out[18]:

```
0    2.670879
1    2.872411
2    2.805049
3    4.155128
4    4.481318
5    3.604210
6    3.145641
dtype: float64
```

In [19]:

```
#square root of all values in a data frame
import math
d2.applymap(math.sqrt)
```

Out[19]:

	A	B	C	D	E	F	G
0	0.562311	0.784683	0.372155	0.545213	0.226063	0.781094	0.801243
1	0.771225	0.742430	0.426627	0.841107	0.683336	0.491888	0.357837
2	0.511150	0.278240	0.705008	0.454848	0.702404	0.751687	0.839064
3	0.817216	0.813905	0.412789	0.782958	0.891477	0.825167	0.752193
4	0.300572	0.970130	0.972925	0.882755	0.502093	0.854503	0.861226
5	0.836885	0.221065	0.890623	0.794199	0.555196	0.925127	0.516622
6	0.679775	0.604972	0.430330	0.624788	0.542250	0.874146	0.826949

In [21]:

```
d2.apply(lambda x:min(x))
```

Out[21]:

```
A    0.090343
B    0.048870
C    0.138499
D    0.206887
E    0.051105
F    0.241954
G    0.128047
dtype: float64
```

In [22]:

```
d2.apply(lambda x:x*x)
```

Out[22]:

	A	B	C	D	E	F	G
0	0.099978	0.379121	0.019182	0.088362	0.002612	0.372231	0.412151
1	0.353772	0.303825	0.033128	0.500500	0.218040	0.058542	0.016396
2	0.068265	0.005993	0.247045	0.042802	0.243416	0.319263	0.495656
3	0.446013	0.438829	0.029034	0.375798	0.631598	0.463625	0.320123
4	0.008162	0.885767	0.896020	0.607241	0.063553	0.533156	0.550133
5	0.490528	0.002388	0.629181	0.397849	0.095014	0.732497	0.071234
6	0.213531	0.133949	0.034293	0.152381	0.086457	0.583897	0.467643

merge DataFrame

In [23]:

```
import pandas as pd
import numpy as np
d=pd.DataFrame({'id':[1,2,3,4],'Name':['lingesh','karan','vino','rishi']})
d
```

Out[23]:

	id	Name
0	1	lingesh
1	2	karan
2	3	vino
3	4	rishi

In [24]:

```
d1=pd.DataFrame({'id':[1,4,7,8],'Name':["riya","veena",'seenu','liya']})
d1
```

Out[24]:

	id	Name
0	1	riya
1	4	veena
2	7	seenu
3	8	liya

In [25]:

```
#inner join
pd.merge(d,d1,on='id',how='inner')
```

Out[25]:

	id	Name_x	Name_y
0	1	lingesh	riya
1	4	rishi	veena

In [26]:

```
#full outer join
pd.merge(d,d1,on='id',how='outer')
```

Out[26]:

	id	Name_x	Name_y
0	1	lingesh	riya
1	2	karan	NaN
2	3	vino	NaN
3	4	rishi	veena
4	7	NaN	seenu
5	8	NaN	liya

In [27]:

```
#Left outer join
pd.merge(d,d1,on='id',how='left')
```

Out[27]:

	id	Name_x	Name_y
0	1	lingesh	riya
1	2	karan	NaN
2	3	vino	NaN
3	4	rishi	veena

In [28]:

```
#right outer join
pd.merge(d,d1,on='id',how='right')
```

Out[28]:

	id	Name_x	Name_y
0	1	lingesh	riya
1	4	rishi	veena
2	7	NaN	seenu
3	8	NaN	liya

Dealing with null values

In [29]:

```
d2
```

Out[29]:

	A	B	C	D	E	F	G
0	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
1	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
3	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
4	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
5	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
6	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [31]:

```
date1=pd.date_range('today',periods=7)
date1
```

Out[31]:

```
DatetimeIndex(['2021-07-16 10:44:55.141559', '2021-07-17 10:44:55.141559',
               '2021-07-18 10:44:55.141559', '2021-07-19 10:44:55.141559',
               '2021-07-20 10:44:55.141559', '2021-07-21 10:44:55.141559',
               '2021-07-22 10:44:55.141559'],
              dtype='datetime64[ns]', freq='D')
```

In [32]:

```
date3=pd.date_range(start='2021-01-20',periods=7)
date3
```

Out[32]:

```
DatetimeIndex(['2021-01-20', '2021-01-21', '2021-01-22', '2021-01-23',
               '2021-01-24', '2021-01-25', '2021-01-26'],
              dtype='datetime64[ns]', freq='D')
```

In [33]:

```
d2=pd.DataFrame(r,date3)
d2
```

Out[33]:

	0	1	2	3	4	5	6
2021-01-20	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
2021-01-21	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2021-01-22	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
2021-01-23	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
2021-01-24	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
2021-01-25	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
2021-01-26	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [34]:

```
d2
```

Out[34]:

	0	1	2	3	4	5	6
2021-01-20	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990
2021-01-21	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047
2021-01-22	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028
2021-01-23	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794
2021-01-24	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710
2021-01-25	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898
2021-01-26	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845

In [ ]:

```
d2.at[0:8, 'E']=np.NaN
d2.at[0:2, 'A']=np.NaN
d2.at[1:5, 'C']=np.NaN
```

In [37]:

```
d2
```

Out[37]:

	0	1	2	3	4	5	6	E	A	C
2021-01-20	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990	NaN	NaN	NaN
2021-01-21	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047	NaN	NaN	NaN
2021-01-22	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028	NaN	NaN	NaN
2021-01-23	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794	NaN	NaN	NaN
2021-01-24	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710	NaN	NaN	NaN
2021-01-25	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898	NaN	NaN	NaN
2021-01-26	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845	NaN	NaN	NaN

In [38]:

```
#detect non missing values
#it will return true for not-null values and false for null values
d2.notna()
```

Out[38]:

	0	1	2	3	4	5	6	E	A	C
2021-01-20	True	True	True	True	True	True	True	False	False	False
2021-01-21	True	True	True	True	True	True	True	False	False	False
2021-01-22	True	True	True	True	True	True	True	False	False	False
2021-01-23	True	True	True	True	True	True	True	False	False	False
2021-01-24	True	True	True	True	True	True	True	False	False	False
2021-01-25	True	True	True	True	True	True	True	False	False	False
2021-01-26	True	True	True	True	True	True	True	False	False	False

In [39]:

```
d2.isna()
```

Out[39]:

	0	1	2	3	4	5	6	E	A	C
2021-01-20	False	False	False	False	False	False	False	True	True	True
2021-01-21	False	False	False	False	False	False	False	True	True	True
2021-01-22	False	False	False	False	False	False	False	True	True	True
2021-01-23	False	False	False	False	False	False	False	True	True	True
2021-01-24	False	False	False	False	False	False	False	True	True	True
2021-01-25	False	False	False	False	False	False	False	True	True	True
2021-01-26	False	False	False	False	False	False	False	True	True	True

In [41]:

```
d2=d2.fillna(102)
d2
```

Out[41]:

	0	1	2	3	4	5	6	E	A	
2021-01-20	0.316193	0.615728	0.138499	0.297257	0.051105	0.610108	0.641990	102.0	102.0	102
2021-01-21	0.594788	0.551203	0.182011	0.707460	0.466948	0.241954	0.128047	102.0	102.0	102
2021-01-22	0.261275	0.077418	0.497036	0.206887	0.493372	0.565034	0.704028	102.0	102.0	102
2021-01-23	0.667842	0.662442	0.170395	0.613024	0.794732	0.680900	0.565794	102.0	102.0	102
2021-01-24	0.090343	0.941152	0.946584	0.779257	0.252098	0.730175	0.741710	102.0	102.0	102
2021-01-25	0.700377	0.048870	0.793210	0.630752	0.308243	0.855860	0.266898	102.0	102.0	102
2021-01-26	0.462095	0.365991	0.185184	0.390360	0.294035	0.764132	0.683845	102.0	102.0	102

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: