

PROGRAM NO:4

AIM:Programs to handle data using pandas

PANDAS

In []:

```
import numpy as np
import pandas as pd
```

In []:

```
labels=['a','b','c']
my_list=[10,20,30]
arr=np.array([10,20,30])
d={'a':10,'b':20,'c':30}
```

In []:

```
pd.Series(data=my_list)
```

Out[]:

```
0    10
1    20
2    30
dtype: int64
```

In []:

```
pd.Series(data=my_list,index=labels)
```

Out[]:

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

In []:

```
pd.Series(my_list,labels)
```

Out[]:

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

In []:

```
pd.Series(arr)
```

Out[]:

```
0     10
1     20
2     30
dtype: int64
```

In []:

```
pd.Series(arr, labels)
```

Out[]:

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

In []:

```
pd.Series(d)
```

Out[]:

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

In []:

```
ser1=pd.Series([1,2,3,4],index=['USA','Germany','Japan','Italy'])
```

In []:

```
ser2=pd.Series([1,2,3,4],index=['USA','Germany','Newyork','London'])
```

In []:

```
ser1['USA']
```

Out[]:

```
1
```

In []:

```
ser1+ser2
```

Out[]:

```
Germany    4.0
Italy      NaN
Japan      NaN
London     NaN
Newyork    NaN
USA        2.0
dtype: float64
```

DATAFRAMES

In []:

```
from numpy.random import randn
np.random.seed(101)
```

In []:

```
df=pd.DataFrame(randn(5,4),index='A B C D E'.split(),columns='W X Y Z'.split())
```

In []:

```
df
```

Out[]:

W	X	Y	Z
---	---	---	---

A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df['X']
```

Out[]:

A 0.628133
B -0.319318
C 0.740122
D -0.758872
E 1.978757
Name: X, dtype: float64

In []:

```
df[['X','Y','Z']]
```

Out[]:

	X	Y	Z
A	0.628133	0.907969	0.503826
B	-0.319318	-0.848077	0.605965
C	0.740122	0.528813	-0.589001
D	-0.758872	-0.933237	0.955057
E	1.978757	2.605967	0.683509

Adding two Columns and Form new column

In []:

```
df['new']=df['Y']+df['X']
```

In []:

```
df
```

Out[]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	1.536102
B	0.651118	-0.319318	-0.848077	0.605965	-1.167395
C	-2.018168	0.740122	0.528813	-0.589001	1.268936
D	0.188695	-0.758872	-0.933237	0.955057	-1.692109
E	0.190794	1.978757	2.605967	0.683509	4.584725

In []:

```
df.drop('new',axis=1)
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

	W	X	Y	Z
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df
```

Out[]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	1.536102
B	0.651118	-0.319318	-0.848077	0.605965	-1.167395
C	-2.018168	0.740122	0.528813	-0.589001	1.268936
D	0.188695	-0.758872	-0.933237	0.955057	-1.692109
E	0.190794	1.978757	2.605967	0.683509	4.584725

In []:

```
df.drop('new',axis=1,inplace=True)
```

In []:

```
df
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df.drop('E',axis=0)
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In []:

```
df.loc['A']
```

Out[]:

W 2.706850
X 0.628133
Y 0.907969
Z 0.503826
Name: A, dtype: float64

In []:

```
df.iloc[0]
```

Out[]:

```
W      2.706850
X      0.628133
Y      0.907969
Z      0.503826
Name: A, dtype: float64
```

In []:

```
df.loc['B','Y']
```

Out[]:

```
-0.8480769834036315
```

In []:

```
df.loc[['A','B'],['W','Y']]
```

Out[]:

	W	Y
A	2.706850	0.907969
B	0.651118	-0.848077

In []:

```
df
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df[df>0]
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	NaN	NaN	0.605965
C	NaN	0.740122	0.528813	NaN
D	0.188695	NaN	NaN	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df[df['W']>0]
```

Out[]:

	W	X	Y	Z
--	---	---	---	---

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df[df['W']>0]['Y']
```

Out[]:

A 0.907969
B -0.848077
D -0.933237
E 2.605967
Name: Y, dtype: float64

In []:

```
df[df['W']>0][['X','Y']]
```

Out[]:

	X	Y
A	0.628133	0.907969
B	-0.319318	-0.848077
D	-0.758872	-0.933237
E	1.978757	2.605967

In []:

```
df[(df['W']>0) & (df['Y']>1)]
```

Out[]:

	W	X	Y	Z
E	0.190794	1.978757	2.605967	0.683509

In []:

```
df[(df['W']>0) | (df['Y']>1)]
```

Out[]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

RESULT:Program executed sucessfully and output is obtained

In []:

