

==> Pandas is used to analyze big data and make conclusions based on statistical theories

==> Pandas is typically used to clean messy data and make them readable and simple to access

==> Pandas can give co-relation between two or more columns

==> It can also help us in doing all statistical methodologies as well as plotting and creating graphs.

==> Pandas allows importing data from various file format such as JSON, SQL, Microsoft Excel etc.

==> Pandas allow various data manipulation operations such as merging, selecting as well as data cleaning

==> We can also handle missing null values and do various operations related with stats

==> We can also use various inbuilt graphs and charts available in Pandas and can generate reports.

In []:

```
1
```

pip install pandas

In []:

```
1
```

In [5]:

```
1 import pandas as pd #importing the library
```

In [6]:

```
1 pd.__version__
```

Out[6]:

'1.2.4'

In [8]:

```
1 import numpy as np
```

In [10]:

```
1 res = np.array([10,20,30,40,50])
```

In [11]:

```
1 res
```

Out[11]:

array([10, 20, 30, 40, 50])

Series

A series is a 1 Dimensional Array Like object containing sequence of values (Similar to that of Numpy) and an associated array of data label called as indexes

In [12]:

```
1 ser = pd.Series([10,20,30,40,50])
```

In [13]:

```
1 print(ser)
```

```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

In [17]:

```
1 ser1 = pd.Series([10,20,30,40,50],index=['a','b','c','d','e'])
```

In [18]:

```
1 ser1
```

Out[18]:

```
a    10
b    20
c    30
d    40
e    50
dtype: int64
```

In [19]:

```
1 ser[3]
```

Out[19]:

```
40
```

In [21]:

```
1 ser1['b']
```

Out[21]:

```
20
```

In [23]:

```
1 ser1[2]
```

Out[23]:

30

In [24]:

```
1 ser1['c']
```

Out[24]:

30

In []:

```
1
```

In [31]:

```
1 # to know all the indexes in the series  
2 ser3 = pd.Series([200,300,400,500],index=['Mumbai','Chennai','Delhi','Jaipur'])
```

In [27]:

```
1 ser3
```

Out[27]:

```
Mumbai      200  
Chennai     300  
Delhi       400  
Jaipur      500  
dtype: int64
```

In [29]:

```
1 ser3.index
```

Out[29]:

```
Index(['Mumbai', 'Chennai', 'Delhi', 'Jaipur'], dtype='object')
```

In []:

```
1  
2
```

In [33]:

```
1 ser3['Chennai'] = 900
```

In [34]:

```
1 ser3
```

Out[34]:

```
Mumbai      200
Chennai     900
Delhi       400
Jaipur      500
dtype: int64
```

In [35]:

```
1 ser3
```

Out[35]:

```
Mumbai      200
Chennai     900
Delhi       400
Jaipur      500
dtype: int64
```

In [36]:

```
1 ser3 > 500 #get the output using filteration in boolean
```

Out[36]:

```
Mumbai      False
Chennai      True
Delhi        False
Jaipur       False
dtype: bool
```

In [37]:

```
1 ser3[ser3 > 500] #get the output of the actual value, use this syntax
```

Out[37]:

```
Chennai     900
dtype: int64
```

In [38]:

```
1 ser3[ser3<500]
```

Out[38]:

```
Mumbai      200
Delhi       400
dtype: int64
```

In []:

```
1
```

In [39]:

```
1 ser3
```

Out[39]:

```
Mumbai      200
Chennai      900
Delhi        400
Jaipur       500
dtype: int64
```

In [46]:

```
1 ser3 = pd.Series([200,300,400,500],index=['Mumbai','Chennai','Delhi','Jaipur'])
```

In [47]:

```
1 ser3
```

Out[47]:

```
Mumbai      200
Chennai      300
Delhi        400
Jaipur       500
dtype: int64
```

In [50]:

```
1 res = ser3 * 2
```

In [51]:

```
1 res
```

Out[51]:

```
Mumbai      400
Chennai      600
Delhi        800
Jaipur      1000
dtype: int64
```

In [52]:

```
1 ser3
```

Out[52]:

```
Mumbai      200
Chennai      300
Delhi        400
Jaipur       500
dtype: int64
```

In []:

```
1 ser3 = ser3 * 2
```

creating series using dictionary

In [53]:

```
1 my_city = {  
2     'Pune':35000,  
3     'Nagpur':49000,  
4     'Mumbai':88000,  
5     'Nasik':30000  
6 }
```

In [54]:

```
1 print(my_city)
```

```
{'Pune': 35000, 'Nagpur': 49000, 'Mumbai': 88000, 'Nasik': 30000}
```

In [55]:

```
1 myseries = pd.Series(my_city)
```

In [56]:

```
1 print(myseries)
```

```
Pune      35000  
Nagpur     49000  
Mumbai     88000  
Nasik      30000  
dtype: int64
```

In []:

```
1
```

In [61]:

```
1 city_data = {'Pune': 35000, 'Nagpur': 49000, 'Mumbai': 88000, 'Nasik': 30000, "D
```

In [62]:

```
1 city = pd.Series(city_data)
```

In [63]:

```
1 city
```

Out[63]:

```
Pune      35000.0  
Nagpur     49000.0  
Mumbai     88000.0  
Nasik      30000.0  
Delhi         NaN  
dtype: float64
```

In []:

```
1
```

identifying not null values

In [64]:

```
1 pd.isnull(city)
```

Out[64]:

```
Pune      False
Nagpur     False
Mumbai     False
Nasik      False
Delhi      True
dtype: bool
```

In []:

```
1
```

In [65]:

```
1 pd.notnull(city)
```

Out[65]:

```
Pune      True
Nagpur     True
Mumbai     True
Nasik      True
Delhi      False
dtype: bool
```

In []:

```
1
```

In [81]:

```
1 ser1 = pd.Series([10,20,30,40,np.nan],index=['A','B','C','D','E'])
```

In [82]:

```
1 ser1
```

Out[82]:

```
A      10.0
B      20.0
C      30.0
D      40.0
E       NaN
dtype: float64
```

In [83]:

```
1 ser2 = pd.Series([100,200,300,400,500],index=['A','B','C','D','E'])
```

In [77]:

```
1 ser2
```

Out[77]:

```
A    100
B    200
C    300
D    400
E    500
dtype: int64
```

In []:

```
1
```

In [84]:

```
1 ser1 + ser2
```

Out[84]:

```
A    110.0
B    220.0
C    330.0
D    440.0
E         NaN
dtype: float64
```

In []:

```
1
```

In [85]:

```
1 city
```

Out[85]:

```
Pune      35000.0
Nagpur     49000.0
Mumbai     88000.0
Nasik      30000.0
Delhi             NaN
dtype: float64
```

In [86]:

```
1 city1 = pd.Series([1000,2000,3000,4000,5000],index=['Pune','Nagpur','Chennai',
```


In [87]:

```
1 city1
```

Out[87]:

```
Pune      1000
Nagpur     2000
Chennai    3000
Delhi      4000
Mumbai     5000
dtype: int64
```

In []:

```
1
```

In [88]:

```
1 city + city1
```

Out[88]:

```
Chennai      NaN
Delhi         NaN
Mumbai      93000.0
Nagpur       51000.0
Nasik         NaN
Pune        36000.0
dtype: float64
```

In []:

```
1
```

In [90]:

```
1 ser3 = city+city1
```

In [91]:

```
1 ser3
```

Out[91]:

```
Chennai      NaN
Delhi         NaN
Mumbai      93000.0
Nagpur       51000.0
Nasik         NaN
Pune        36000.0
dtype: float64
```

In [97]:

```
1 s1 = pd.Series(['Mum', 'Del', 'Hyd'], index=['A', 'B', 'C'])
```

In [99]:

```
1 s2 = pd.Series(['Mah', 'Ind', 'Ap'], index=['A', 'P', 'Q'])
```

In [100]:

```
1 s1
```

Out[100]:

```
A    Mum
B    Del
C    Hyd
dtype: object
```

In [101]:

```
1 s2
```

Out[101]:

```
A    Mah
P    Ind
Q    Ap
dtype: object
```

In [102]:

```
1 s1+s2
```

Out[102]:

```
A    MumMah
B         NaN
C         NaN
P         NaN
Q         NaN
dtype: object
```

In []:

```
1
```

In []:

```
1
```

Data Frame

A DataFrame represents a rectangular table of data and contains ordered collection of columns each of which can be a different data type (numeric, boolean, String, etc)

The DataFrame has both row and column index, it can be thought as a collection of series all sharing the same index. The data is stored in one or more 2-D block rather than a list or dictionary or other collection of 1-D Array

In []:

```
1
```

In [104]:

```
1 data = {  
2     'State': ['Mumbai', 'Pune', 'Chennai', 'Nagpur', 'Delhi'],  
3     'year': [2001, 2011, 2018, 2019, 2020],  
4     'pop': [1.5, 1.6, 1.7, 3.6, 3.2]  
5 }
```

In [105]:

```
1 data
```

Out[105]:

```
{'State': ['Mumbai', 'Pune', 'Chennai', 'Nagpur', 'Delhi'],  
 'year': [2001, 2011, 2018, 2019, 2020],  
 'pop': [1.5, 1.6, 1.7, 3.6, 3.2]}
```

In [106]:

```
1 df = pd.DataFrame(data)
```

In [108]:

```
1 df
```

Out[108]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [109]:

```
1 df['State']
```

Out[109]:

```
0    Mumbai  
1      Pune  
2    Chennai  
3    Nagpur  
4      Delhi  
Name: State, dtype: object
```

In [110]:

```
1 df['year']
```

Out[110]:

```
0    2001
```

```
1    2011
```

```
2    2018
```

```
3    2019
```

```
4    2020
```

Name: year, dtype: int64

In [111]:

```
1 df['pop']
```

Out[111]:

```
0    1.5
```

```
1    1.6
```

```
2    1.7
```

```
3    3.6
```

```
4    3.2
```

Name: pop, dtype: float64

In [112]:

```
1 df
```

Out[112]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [113]:

```
1 df
```

Out[113]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [116]:

```
1 df.head(3)
```

Out[116]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7

In [115]:

```
1 df.head()
```

Out[115]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In []:

```
1
```