Pandas

- It is a open source library and most popular library
- Mainly use in data analysis (EDA)
- pip install pandas
- This is a high level data manipulation tool.
- It deals with data structure.(Series and DataFrame)

Series

- Called as One Dimensional Data

DataFrame

- Called as Multi Dimensional Data

- These data structure are build on the numpy package
- The key data structure is DataFrame (Tabular data)
- Data in pandas often used to feed statistical analysis on plotting funtions from matplotlib
- pip install pandas
- import pandas as pd
- Wes Mckinney founder of the Pandas.
- It is used to handle missing data, merging, concatinate and reshaping the data and etc.
- Pandas stands for panel data.
- It was orignated with this idea of panel data which means mathmatical methods for multi dimensional data.
- Panel is a 3-d labeled array, this is also one of the data structure in pandas but rarely used.
- pandas contains data structure and manipulation tool design to make data analysis and cleaning fast and easy in python.

- It has two data structures - Series , DataFrame

In [1]:
```
import pandas as pd
```

In [2]:
```
# It is a one dimensional array like object(string) containing a sequence of values and n-Index

data=[1,2,3,4]
s1=pd.Series(data)
print(s1)
```

```
0    1
1    2
2    3
3    4
dtype: int64
```

In [3]:
```
s1.values
```

Out[3]:
```
array([1, 2, 3, 4])
```

In [4]:
```
s1.index
```

Out[4]:
```
RangeIndex(start=0, stop=4, step=1)
```

In [5]:
```
#index changing
s1.index=["a","b","c","d"] #a,b,c,d called as labels
print(s1)
```

```
a    1
b    2
c    3
d    4
dtype: int64
```

In [6]:
```
s1.index
```

Out[6]:
```
Index(['a', 'b', 'c', 'd'], dtype='object')
```

In [7]:
```
#indexing in series within pandas
s1["a"]
```

Out[7]:
```
1
```

In [8]:
```
#iloc vs loc
#iloc queries by position and loc by label name
```

In [9]:
```
print(s1.iloc[2])
print(s1.loc["c"])
```

```
3
3
```

In [10]:
```
#mutability (adding new values in index)
s1["e"]=5
print(s1)
```

```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

In [11]:
```
s1.loc["f"]=6
print(s1)
```

```
a    1
b    2
c    3
d    4
e    5
f    6
dtype: int64
```

In [12]:
```
6 in s1.values #membership operators
```

Out[12]:
```
True
```

In [13]:
```
10 in s1.values
```

Out[13]:
```
False
```

In [14]:
```
"e" in s1.index #membership
```

Out[14]: True

In [15]:
```
s1[1:4]
```

Out[15]:

	0
b	2
c	3
d	4

dtype: int64

In [16]:
```
data1={"Jaipur":"Rajasthan","Mumbai":"Maharastra","kolkata":"West Bengal","Banglore":"Karnataka","Chandigarh":"Punjab"}
s2=pd.Series(data1)
print(s2)
```

```
Jaipur          Rajasthan
Mumbai         Maharastra
kolkata       West Bengal
Banglore        Karnataka
Chandigarh         Punjab
dtype: object
```

In [17]:
```
s2.name="states"
print(s2)
```

```
Jaipur          Rajasthan
Mumbai         Maharastra
kolkata       West Bengal
Banglore        Karnataka
Chandigarh         Punjab
Name: states, dtype: object
```

In [18]:
```
s2.index.name="Capital"
```

In [19]:
```
print(s2)
```

```
Capital
Jaipur          Rajasthan
Mumbai         Maharastra
kolkata       West Bengal
Banglore        Karnataka
Chandigarh         Punjab
Name: states, dtype: object
```

In [20]:
```
s2[0:3] #slicing
```

Out[20]:

	states
Capital	
Jaipur	Rajasthan
Mumbai	Maharastra
kolkata	West Bengal

dtype: object

In [21]:
```
s2[["Jaipur","Mumbai","kolkata"]]
```

Out[21]:

	states
Capital	
Jaipur	Rajasthan
Mumbai	Maharastra
kolkata	West Bengal

dtype: object

In [22]:
```
Capitals=["Jaipur","Mumbai","delhi","kanpur"]
s3=pd.Series(data1,index=Capitals)
print(s3)
```

```
Jaipur     Rajasthan
Mumbai    Maharastra
delhi            NaN
kanpur           NaN
dtype: object
```

In [23]:
```
#identify the missing data if it is present in the data
s3.isnull()
```

Out[23]:

	0
Jaipur	False
Mumbai	False
delhi	True
kanpur	True

dtype: bool

In [24]:
```
s3.notnull() #reverse of isnull command
```

Out[24]:

	0
Jaipur	True
Mumbai	True
delhi	False
kanpur	False

dtype: bool

In [25]:
```
s=pd.Series(["India","Pakistan","Australia","New zealand"],index=["cricket","cricket","cricket","cricket"])
print(s)
```

```
cricket          India
cricket       Pakistan
cricket      Australia
cricket    New zealand
dtype: object
```

In [26]:
```
s.loc["cricket"] #index labels can be nonunique
```

Out[26]:

	0
cricket	India
cricket	Pakistan
cricket	Australia
cricket	New zealand

dtype: object

In [27]:
```
colors=["blue","blue","pink","white",None]
pd.Series(colors)
```

Out[27]:

	0
0	blue
1	blue
2	pink
3	white
4	None

dtype: object

In [28]:
```
num=[1,2,3,None]
pd.Series(num)
```

Out[28]:

	0
0	1.0
1	2.0
2	3.0
3	NaN

dtype: float64

DataFrame

- It represents a rectangular table of data and contains a collection of columns.
- The DataFrame has both a row and colum index.

In [30]:
```
# Create a dataframe with student details
import pandas as pd
student1=pd.Series({"Name":"Utkarsh","ID":1})
student2=pd.Series({"Name":"Sanad","ID":2})
student3=pd.Series({"Name":"Himanshu","ID":3})
student4=pd.Series({"Name":"Shivam","ID":4})
student5=pd.Series({"Name":"Raj","ID":5})
student6=pd.Series({"Name":"Virendar","ID":6})
```

In [31]:
```
df1=pd.DataFrame([student1,student2,student3,student4,student5,student6],index=((101,102,103,104,105,106)))
```

In [32]:
```
print(df1)
```

```
         Name  ID
101   Utkarsh   1
102     Sanad   2
103  Himanshu   3
104    Shivam   4
105       Raj   5
106  Virendar   6
```

In [33]:
```
df1
```

Out[33]:

	Name	ID
101	Utkarsh	1
102	Sanad	2
103	Himanshu	3
104	Shivam	4
105	Raj	5
106	Virendar	6

In [34]:
```
data={"Name":["sanad","himanshu","virendar","Raj","shivam","utkarsh"],"Id":[1,2,3,4,5,6]}
df2=pd.DataFrame(data,index=[101,102,103,104,105,106])
df2
```

Out[34]:

	Name	Id
101	sanad	1
102	himanshu	2
103	virendar	3
104	Raj	4
105	shivam	5
106	utkarsh	6

In [36]:
```
# #fetch top 5 data from all data
df2.head()
```

Out[36]:

	Name	Id
101	sanad	1
102	himanshu	2
103	virendar	3
104	Raj	4
105	shivam	5

In [37]:
```
df2.tail() #fetch all 5 bottom data from all above
```

Out[37]:

	Name	Id
102	himanshu	2
103	virendar	3
104	Raj	4
105	shivam	5
106	utkarsh	6

In [38]:
```
#adding new column in the DataFrame
df3=pd.DataFrame(data,index=[101,102,103,104,105,106],columns=["Name","Id","Age"])
df3
```

Out[38]:

	Name	Id	Age
101	sanad	1	NaN
102	himanshu	2	NaN
103	virendar	3	NaN
104	Raj	4	NaN
105	shivam	5	NaN
106	utkarsh	6	NaN

In [39]:
```
# Inserting data in Age column
df3.Age=21
df3
```

Out[39]:

	Name	Id	Age
101	sanad	1	21
102	himanshu	2	21
103	virendar	3	21
104	Raj	4	21
105	shivam	5	21
106	utkarsh	6	21

In [40]:
```
df3.index
```

Out[40]:
```
Index([101, 102, 103, 104, 105, 106], dtype='int64')
```

In [41]:
```
df3.columns # All column names
```

Out[41]:
```
Index(['Name', 'Id', 'Age'], dtype='object')
```

In [42]:
```
df3.values #most important
```

Out[42]:
```
array([['sanad', 1, 21],
       ['himanshu', 2, 21],
       ['virendar', 3, 21],
       ['Raj', 4, 21],
       ['shivam', 5, 21],
       ['utkarsh', 6, 21]], dtype=object)
```

In [43]:
```
df3["Name"]
```

Out[43]:

	Name
101	sanad
102	himanshu
103	virendar
104	Raj
105	shivam
106	utkarsh

dtype: object

In [44]:
```
df3.Name
```

Out[44]:

	Name
101	sanad
102	himanshu
103	virendar
104	Raj
105	shivam
106	utkarsh

dtype: object

In [45]:
```
df3.loc[104]
```

Out[45]:

	104
Name	Raj
Id	4
Age	21

dtype: object

In [46]:
```
df3.iloc[0]
```

Out[46]:

	101
Name	sanad
Id	1
Age	21

dtype: object

In [47]:
```
df3.loc[107]=["Satvamev",7,23] # Adding new Row along with data
```

In [48]:
```
df3
```

Out[48]:

	Name	Id	Age
101	sanad	1	21
102	himanshu	2	21
103	virendar	3	21
104	Raj	4	21
105	shivam	5	21
106	utkarsh	6	21
107	Satvamev	7	23

In [49]:
```
df3["Name"][101] #slicing
```

Out[49]:
```
'sanad'
```

In [50]:
```
# Inserting the new data within the age column

val=pd.Series([29.5,30,20,24,25,36,22],index=[101,102,103,104,105,106,107])
df3["Age"]=val
df3
```

Out[50]:

	Name	Id	Age
101	sanad	1	29.5
102	himanshu	2	30.0
103	virendar	3	20.0
104	Raj	4	24.0
105	shivam	5	25.0
106	utkarsh	6	36.0
107	Satvamev	7	22.0

In [51]:
```
df3["Weight"]=df3.Age # Creating the new column
df3
```

Out[51]:

	Name	Id	Age	Weight
101	sanad	1	29.5	29.5
102	himanshu	2	30.0	30.0
103	virendar	3	20.0	20.0
104	Raj	4	24.0	24.0
105	shivam	5	25.0	25.0
106	utkarsh	6	36.0	36.0
107	Satvamev	7	22.0	22.0

In [52]:
```
# Creating the height column and inserting the data within the specific index number
val2=pd.Series([6,5],index=[101,103])
df3["Height"]=val2
df3
```

Out[52]:

	Name	Id	Age	Weight	Height
101	sanad	1	29.5	29.5	6.0
102	himanshu	2	30.0	30.0	NaN
103	virendar	3	20.0	20.0	5.0
104	Raj	4	24.0	24.0	NaN
105	shivam	5	25.0	25.0	NaN
106	utkarsh	6	36.0	36.0	NaN
107	Satvamev	7	22.0	22.0	NaN

In [53]:
```
# Delete the data temporarily using drop function

df3.drop([102,104,105,106,107])
```

Out[53]:

	Name	Id	Age	Weight	Height
101	sanad	1	29.5	29.5	6.0
103	virendar	3	20.0	20.0	5.0

In [ ]:
```
#Delete all missing data along with rows and column using inplace=True Command

#df3.drop([102,104,105,106,107],inplace=True)
```

In [54]:
```
#only remove column
del df3["Weight"]
df3
```

Out[54]:

	Name	Id	Age	Height
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN

In [55]:
```
#row to column or visa-versa transpose temporarily

df3.T
```

Out[55]:

	101	102	103	104	105	106	107
Name	sanad	himanshu	virendar	Raj	shivam	utkarsh	Satvamev
Id	1	2	3	4	5	6	7
Age	29.5	30.0	20.0	24.0	25.0	36.0	22.0
Height	6.0	NaN	5.0	NaN	NaN	NaN	NaN

In [56]:
```
df3.index.name="trainee"
df3.columns.name="details"
```

In [57]:
```
df3
```

Out[57]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN

In [58]:
```
df4=df3.reindex([101,202,103,204,205,206]) # reindex the existing index
df4
```

Out[58]:

details	Name	Id	Age	Height
trainee				
101	sanad	1.0	29.5	6.0
202	NaN	NaN	NaN	NaN
103	virendar	3.0	20.0	5.0
204	NaN	NaN	NaN	NaN
205	NaN	NaN	NaN	NaN
206	NaN	NaN	NaN	NaN

In [59]:
```
df3 # Original Data
```

Out[59]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN

In [60]:
```
df5=df3.reindex([101,202,103,204,205,206],method="ffill") #forward filling
df5
```

Out[60]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
202	Satvamev	7	22.0	NaN
103	virendar	3	20.0	5.0
204	Satvamev	7	22.0	NaN
205	Satvamev	7	22.0	NaN
206	Satvamev	7	22.0	NaN

In [61]:
```
df6=df3.reindex([101,202,103,204,205,206],method="bfill") #backward filling
df6
```

Out[61]:

details	Name	Id	Age	Height
trainee				
101	sanad	1.0	29.5	6.0
202	NaN	NaN	NaN	NaN
103	virendar	3.0	20.0	5.0
204	NaN	NaN	NaN	NaN
205	NaN	NaN	NaN	NaN
206	NaN	NaN	NaN	NaN

In [62]:
```
# Create new dataframe

df = pd.DataFrame({"A":[1,None,None,4], "B":[None,5,None,7]})
df
```

Out[62]:

	A	B
0	1.0	NaN
1	NaN	5.0
2	NaN	NaN
3	4.0	7.0

In [63]:
```
df.bfill()
```

Out[63]:

	A	B
0	1.0	5.0
1	4.0	5.0
2	4.0	7.0
3	4.0	7.0

In [64]:
```
df.bfill(limit=1)
```

Out[64]:

	A	B
0	1.0	5.0
1	NaN	5.0
2	4.0	7.0
3	4.0	7.0

In [65]:
```
#missing data handling commands like, drop, bfill,ffill,reindex
df5.dropna()
```

Out[65]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
103	virendar	3	20.0	5.0

In [66]:
```
df5
```

Out[66]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
202	Satvamev	7	22.0	NaN
103	virendar	3	20.0	5.0
204	Satvamev	7	22.0	NaN
205	Satvamev	7	22.0	NaN
206	Satvamev	7	22.0	NaN

In [67]:
```
# #Replace NaN values with 0

df5.fillna(0)
```

Out[67]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
202	Satvamev	7	22.0	0.0
103	virendar	3	20.0	5.0
204	Satvamev	7	22.0	0.0
205	Satvamev	7	22.0	0.0
206	Satvamev	7	22.0	0.0

In [68]:
```
df5.set_index(["Name"]) # to make any column as index
```

Out[68]:

details	Id	Age	Height
Name			
sanad	1	29.5	6.0
Satvamev	7	22.0	NaN
virendar	3	20.0	5.0
Satvamev	7	22.0	NaN
Satvamev	7	22.0	NaN
Satvamev	7	22.0	NaN

In [69]:
```
df5.reset_index() # Resetting the Index
```

Out[69]:

details	trainee	Name	Id	Age	Height
0	101	sanad	1	29.5	6.0
1	202	Satvamev	7	22.0	NaN
2	103	virendar	3	20.0	5.0
3	204	Satvamev	7	22.0	NaN
4	205	Satvamev	7	22.0	NaN
5	206	Satvamev	7	22.0	NaN

In [70]:
```
# Indexing and shorting a DataFrame

df3["Name"] #got all details from Name column
```

Out[70]:

trainee	Name
101	sanad
102	himanshu
103	virendar
104	Raj
105	shivam
106	utkarsh
107	Satvamev

dtype: object

In [71]:
```
df3[["Name","Id"]] #due to 2d data
```

Out[71]:

details	Name	Id
trainee		
101	sanad	1
102	himanshu	2
103	virendar	3
104	Raj	4
105	shivam	5
106	utkarsh	6
107	Satvamev	7

In [72]:
```
# Getting all data according to the label

df3.loc[101]
```

Out[72]:

	101
details	
Name	sanad
Id	1
Age	29.5
Height	6.0

dtype: object

In [73]:
```
df3.loc[105]["Name"] #use of slicing
```

Out[73]:
```
'shivam'
```

In [74]:
```
df3
```

Out[74]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN

In [75]:
```
df3.loc[[101,102,103],["Name","Id"]]
```

Out[75]:

details	Name	Id
trainee		
101	sanad	1
102	himanshu	2
103	virendar	3

In [76]:
```
df3.iloc[[0,1],[1,2]] #[0,1] are rows and [1,2] are column
```

Out[76]:

details	Id	Age
trainee		
101	1	29.5
102	2	30.0

In [77]:
```
df3.loc[:,["Name","Age"]] #for all rows and specific columns
```

Out[77]:

details	Name	Age
trainee		
101	sanad	29.5
102	himanshu	30.0
103	virendar	20.0
104	Raj	24.0
105	shivam	25.0
106	utkarsh	36.0
107	Satvamev	22.0

In [78]:
```
df3.sort_index() #for sorting all the data in ascending order or increment order
```

Out[78]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN

In [79]:
```
df3.sort_index(axis=1,ascending=True) #in pandas axis=1 column and axis = 0 are rows
```

Out[79]:

details	Age	Height	Id	Name
trainee				
101	29.5	6.0	1	sanad
102	30.0	NaN	2	himanshu
103	20.0	5.0	3	virendar
104	24.0	NaN	4	Raj
105	25.0	NaN	5	shivam
106	36.0	NaN	6	utkarsh
107	22.0	NaN	7	Satvamev

In [80]:
```
df3.sort_values(by="Name") # Sorting the data with respect Name Data
```

Out[80]:

details	Name	Id	Age	Height
trainee				
104	Raj	4	24.0	NaN
107	Satvamev	7	22.0	NaN
102	himanshu	2	30.0	NaN
101	sanad	1	29.5	6.0
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
103	virendar	3	20.0	5.0

In [82]:
```
df3.sort_values(by="Age") # Sorting the data with respect Age
```

Out[82]:

details	Name	Id	Age	Height
trainee				
103	virendar	3	20.0	5.0
107	Satvamev	7	22.0	NaN
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
106	utkarsh	6	36.0	NaN

In [83]:
```
# Inserting new Data

df3.loc[108]=["sanad",1,29.5,6.0]
```

In [84]:
```
df3
```

Out[84]:

details	Name	Id	Age	Height
trainee				
101	sanad	1	29.5	6.0
102	himanshu	2	30.0	NaN
103	virendar	3	20.0	5.0
104	Raj	4	24.0	NaN
105	shivam	5	25.0	NaN
106	utkarsh	6	36.0	NaN
107	Satvamev	7	22.0	NaN
108	sanad	1	29.5	6.0

Descriptive statistics

In [85]:
```
import numpy as np
data={"a":np.arange(11,21),"b":np.arange(21,31),"c":np.arange(31,41)} #numpy is more efficeient then pandas
df=pd.DataFrame(data)
print(df)
```

```
    a   b   c
0  11  21  31
1  12  22  32
2  13  23  33
3  14  24  34
4  15  25  35
5  16  26  36
6  17  27  37
7  18  28  38
8  19  29  39
9  20  30  40
```

In [86]:
```
df.sum()
```

Out[86]:

	0
a	155
b	255
c	355

dtype: int64

In [87]:
```
df.sum(axis=1)
```

Out[87]:

	0
0	63
1	66
2	69
3	72
4	75
5	78
6	81
7	84
8	87
9	90

dtype: int64

In [88]:
```
df.mean()
```

Out[88]:

	0
a	15.5
b	25.5
c	35.5

dtype: float64

In [89]:
```
df.mean(skipna=True) # this command is use when we have NaN data present in the array
```

Out[89]:

	0
a	15.5
b	25.5
c	35.5

dtype: float64

In [90]:
```
df.min()
```

Out[90]:

	0
a	11
b	21
c	31

dtype: int64

In [91]:
```
df.var() # Variance
```

Out[91]:

	0
a	9.166667
b	9.166667
c	9.166667

dtype: float64

In [92]:
```
df.max() # Max Value
```

Out[92]:

	0
a	20
b	30
c	40

dtype: int64

In [93]:
```
df.std() # Standard Deviation
```

Out[93]:

	0
a	3.02765
b	3.02765
c	3.02765

dtype: float64

In [94]:
```
df.median()
```

Out[94]:

	0
a	15.5
b	25.5
c	35.5

dtype: float64

In [95]:
```
df.mode()
```

Out[95]:

	a	b	c
0	11	21	31
1	12	22	32
2	13	23	33
3	14	24	34
4	15	25	35
5	16	26	36
6	17	27	37
7	18	28	38
8	19	29	39
9	20	30	40

In [96]:
```
df.describe() # Generates descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distrib
```

Out[96]:

	a	b	c
count	10.00000	10.00000	10.00000
mean	15.50000	25.50000	35.50000
std	3.02765	3.02765	3.02765
min	11.00000	21.00000	31.00000
25%	13.25000	23.25000	33.25000
50%	15.50000	25.50000	35.50000
75%	17.75000	27.75000	37.75000
max	20.00000	30.00000	40.00000

In [ ]:
```

```