PANDAS SESSION - BY NAVEEN GUPTA

M-9971999404 , E-mail:gupta.naveen1@gmail.com

youtube.com/c/NaveenGupta , https://wakelet.com/@NaveenGupta429 (https://wakelet.com/@NaveenGupta429)

CBSE Syllabus Class - XII 2020-21 Marks - 30

Pandas - 1 Session Introduction to Python libraries- Pandas, Matplotlib. Data structures in Pandas - Series and Data Frames. Series: Creation of Series from – ndarray, dictionary, scalar value; mathematical operations; Head and Tail functions; Selection, Indexing and Slicing. Data Frames: creation - from dictionary of Series, list of dictionaries, Text/CSV files; display; iteration; Operations on rows and columns: add, select, delete, rename; Head and Tail functions; Indexing using Labels, Boolean Indexing; Joining, Merging and Concatenation. Importing/Exporting Data between CSV files and Data Frames

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
%matplotlib inline
```

In [2]:
```
#Series creation - One dimensional array
```

In [2]:
```
l=['a','b','c','d','e'] #List creation
a=pd.Series(l)  #List passed to series method
a
```

Out[2]:
```
0    a
1    b
2    c
3    d
4    e
dtype: object
```

In [3]:
```
#Creation of series with multiple list
sub=['Eng','CSC','IP','Maths','Act']
name=['Geeta','Naveen','Vaishali','Seba','Jyotika']
Sch=pd.Series(name,index=sub)
Sch
```

Out[3]:
```
Eng         Geeta
CSC        Naveen
IP       Vaishali
Maths        Seba
Act       Jyotika
dtype: object
```

In [4]:
```
#•      NumPy Array
a=pd.Series(np.arange(1,80,10))
a
```

Out[4]:
```
0     1
1    11
2    21
3    31
4    41
5    51
6    61
7    71
dtype: int32
```

In [5]:
```
a=pd.Series(np.array([3,4,5,6,7]))
a
```

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

In [6]:
```
#Using Dictionary
d={1:'a',2:'b',3:'c',4:'d'}
a=pd.Series(d)
a
```

Out[6]:
```
1    a
2    b
3    c
4    d
dtype: object
```

In [7]:
```
#•      Scalar value or constant
a=pd.Series(10,index=['a','b','c','d'])
a
```

Out[7]:
```
a    10
b    10
c    10
d    10
dtype: int64
```

In [8]:
```
a=pd.Series(('a','b','c','d'),index=[18,2,3,40])
a
```

Out[8]:
```
18    a
2     b
3     c
40    d
dtype: object
```

In [9]:
```
#Creation of series from CSV file
df=pd.read_csv("e:\\dataset\\gross-enrollment-ratio-2013-2016_1.csv")
df
```

Out[9]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Upper
0	Andaman & Nicobar Islands	2013-14	95.88	91.97	93.93	94.70	
1	Andhra Pradesh	2013-14	96.62	96.87	96.74	82.81	
2	Arunachal Pradesh	2013-14	129.12	127.77	128.46	112.64	
3	Assam	2013-14	111.77	115.16	113.43	87.85	
4	Bihar	2013-14	95.03	101.15	97.96	80.60	
...	...	...	...	...	...	...	
105	Tripura	2014-15	109.50	110.49	109.98	118.80	
106	Uttar Pradesh	2014-15	91.54	98.93	95.00	68.14	
107	Uttaranchal	2014-15	100.11	101.02	100.54	84.20	
108	West Bengal	2014-15	101.78	102.91	102.33	96.16	
109	All India	2014-15	98.85	101.43	100.08	87.71	

110 rows × 14 columns

In [10]:
```
type(df)
```

Out[10]:
```
pandas.core.frame.DataFrame
```

In [11]:
```
df.columns
```

Out[11]:
```
Index(['State_UT', 'Year', 'Primary_Boys', 'Primary_Girls', 'Primary_Total',
       'Upper_Primary_Boys', 'Upper_Primary_Girls', 'Upper_Primary_Total',
       'Secondary_Boys', 'Secondary_Girls', 'Secondary_Total',
       'Higher_Secondary_Boys', 'Higher_Secondary_Girls',
       'Higher_Secondary_Total'],
      dtype='object')
```

In [12]:
```
a=pd.Series(df['Primary_Boys'])
a
```

Out[12]:
```
0       95.88
1       96.62
2      129.12
3      111.77
4       95.03
        ...
105    109.50
106     91.54
107    100.11
108    101.78
109     98.85
Name: Primary_Boys, Length: 110, dtype: float64
```

In [13]:
```
type(a)
```

Out[13]:
```
pandas.core.series.Series
```

In [14]:
```
#Naming the Series
b=pd.Series([23,45,12,32])
b
```

Out[14]:
```
0    23
1    45
2    12
3    32
dtype: int64
```

In [15]:
```
b.name="Age"
b
```

Out[15]:
```
0    23
1    45
2    12
3    32
Name: Age, dtype: int64
```

In [16]:
```
b.index.name="Yrs"
b
```

Out[16]:
```
Yrs
0    23
1    45
2    12
3    32
Name: Age, dtype: int64
```

In [17]:
```
#Series Object

print("Index Values",Sch.index)
print("Values", Sch.values)
print("Data Type",Sch.dtype)
print("Dimensions",Sch.ndim)
print("Size of the Series",Sch.size)
print("Checking nans",Sch.hasnans)
print("Checking if any position is empty or not",Sch.empty)
```

```
Index Values Index(['Eng', 'CSC', 'IP', 'Maths', 'Act'], dtype='object')
Values ['Geeta' 'Naveen' 'Vaishali' 'Seba' 'Jyotika']
Data Type object
Dimensions 1
Size of the Series 5
Checking nans False
Checking if any position is empty or not False
```

In [18]:
```
#Series Fucntion
#Will display first 5 records
Sch.head()
```

Out[18]:
```
Eng         Geeta
CSC        Naveen
IP       Vaishali
Maths        Seba
Act       Jyotika
dtype: object
```

In [19]:
```
#will display first 3 records
Sch.head(3)
```

Out[19]:
```
Eng       Geeta
CSC      Naveen
IP     Vaishali
dtype: object
```

In [21]:
```
#Tail Function
Sch.tail()
```

Out[21]:
```
Eng         Geeta
CSC        Naveen
IP       Vaishali
Maths        Seba
Act       Jyotika
dtype: object
```

In [22]:
```
Sch.tail(2)
```

Out[22]:
```
Maths       Seba
Act      Jyotika
dtype: object
```

In [23]:
```
#Will count the values
Sch.count()
```

Out[23]:
```
5
```

In [24]:
```
a=pd.Series(np.array([1,2,3,np.nan,4]))
print(a)
a.count()
```

```
0    1.0
1    2.0
2    3.0
3    NaN
4    4.0
dtype: float64
```

Out[24]:
```
4
```

In [25]:
```
#Accessing the elements of Series
```

In [26]:
```
animal=pd.Series(['Lion','Bear','Elephant','Tiger','Wolf'],index=['L','B','E',
'T','W'])
animal
```

Out[26]:
```
L        Lion
B        Bear
E    Elephant
T       Tiger
W        Wolf
dtype: object
```

In [27]:
```
#attribute access
animal.B
```

Out[27]:
```
'Bear'
```

In [28]:
```
animal.L
```

Out[28]:
```
'Lion'
```

In [29]:
```
#Index Value
animal['E']
```

Out[29]:
```
'Elephant'
```

In [30]:
```
animal[['L','E','W']]
```

Out[30]:
```
L        Lion
E    Elephant
W        Wolf
dtype: object
```

In [31]:
```
#accessing through position
animal[3]
```

Out[31]:
```
'Tiger'
```

In [32]:
```
#Slicing of Series
```

In [33]:
```
animal[1:4]
```

Out[33]:
```
B        Bear
E    Elephant
T       Tiger
dtype: object
```

In [34]:
```
animal[-3:-1]
```

Out[34]:
```
E    Elephant
T       Tiger
dtype: object
```

In [35]:
```
animal[:2]
```

Out[35]:
```
L    Lion
B    Bear
dtype: object
```

In [36]:
```
animal[::-1] #Reverse
```

Out[36]:
```
W        Wolf
T       Tiger
E    Elephant
B        Bear
L        Lion
dtype: object
```

In [37]:
```
#Boolean Indexing
```

In [38]:
```
animal=='L'
```

Out[38]:
```
L    False
B    False
E    False
T    False
W    False
dtype: bool
```

In [39]:
```
animal>'Lion'
```

Out[39]:
```
L    False
B    False
E    False
T     True
W     True
dtype: bool
```

In [20]:
```
#Number Series
a=pd.Series(np.random.randint(12,50,(10)))
a
```

Out[20]:
```
0    14
1    34
2    39
3    31
4    32
5    27
6    20
7    30
8    15
9    30
dtype: int32
```

In [21]:
```
a>10
```

Out[21]:
```
0    True
1    True
2    True
3    True
4    True
5    True
6    True
7    True
8    True
9    True
dtype: bool
```

In [22]:
```
a[a>30]
```

Out[22]:
```
1    34
2    39
3    31
4    32
dtype: int32
```

In [43]:
```
#Mathematical / Arthmetic Operatioms
s=pd.Series([1,2,3,4,5,6])
print(s)
s1=pd.Series([8,7,5,6,np.nan,4])
print(s1)
s2=pd.Series([11,12,13,14,6])
print(s2)
```

```
0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
0    8.0
1    7.0
2    5.0
3    6.0
4    NaN
5    4.0
dtype: float64
0    11
1    12
2    13
3    14
4     6
dtype: int64
```

In [44]:
```
#Addtion +
s+s1
```

Out[44]:
```
0     9.0
1     9.0
2     8.0
3    10.0
4     NaN
5    10.0
dtype: float64
```

In [45]:
```
s+s2# if the value will not match ot will generate NaN
```

Out[45]:
```
0    12.0
1    14.0
2    16.0
3    18.0
4    11.0
5     NaN
dtype: float64
```

In [46]:
```
s1**s
```

Out[46]:
```
0       8.0
1      49.0
2     125.0
3    1296.0
4       NaN
5    4096.0
dtype: float64
```

In [47]:
```
s-s1
```

Out[47]:
```
0   -7.0
1   -5.0
2   -2.0
3   -2.0
4    NaN
5    2.0
dtype: float64
```

In [48]:
```
#Vector Operation
```

In [49]:
```
df
```

Out[49]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Upper
0	Andaman & Nicobar Islands	2013-14	95.88	91.97	93.93	94.70	
1	Andhra Pradesh	2013-14	96.62	96.87	96.74	82.81	
2	Arunachal Pradesh	2013-14	129.12	127.77	128.46	112.64	
3	Assam	2013-14	111.77	115.16	113.43	87.85	
4	Bihar	2013-14	95.03	101.15	97.96	80.60	
...	...	...	...	...	...	...	
105	Tripura	2014-15	109.50	110.49	109.98	118.80	
106	Uttar Pradesh	2014-15	91.54	98.93	95.00	68.14	
107	Uttaranchal	2014-15	100.11	101.02	100.54	84.20	
108	West Bengal	2014-15	101.78	102.91	102.33	96.16	
109	All India	2014-15	98.85	101.43	100.08	87.71	

110 rows × 14 columns

In [50]:
```
noboy=pd.Series(df['Primary_Boys'])
noboy
```

Out[50]:
```
0       95.88
1       96.62
2      129.12
3      111.77
4       95.03
        ...
105    109.50
106     91.54
107    100.11
108    101.78
109     98.85
Name: Primary_Boys, Length: 110, dtype: float64
```

In [51]:
```
print(type(noboy))
noboy.dtype
```

```
<class 'pandas.core.series.Series'>
```

Out[51]:
```
dtype('float64')
```

In [52]:
```
noboy-2
```

Out[52]:
```
0       93.88
1       94.62
2      127.12
3      109.77
4       93.03
        ...
105    107.50
106     89.54
107     98.11
108     99.78
109     96.85
Name: Primary_Boys, Length: 110, dtype: float64
```

In [53]:
```
noboy**2
```

Out[53]:
```
0       9192.9744
1       9335.4244
2      16671.9744
3      12492.5329
4       9030.7009
          ...
105    11990.2500
106     8379.5716
107    10022.0121
108    10359.1684
109     9771.3225
Name: Primary_Boys, Length: 110, dtype: float64
```

In [54]:
```
noboy<20
```

Out[54]:
```
0      False
1      False
2      False
3      False
4      False
       ...
105    False
106    False
107    False
108    False
109    False
Name: Primary_Boys, Length: 110, dtype: bool
```

In [55]:
```
noboy[noboy>60]
```

Out[55]:
```
0       95.88
1       96.62
2      129.12
3      111.77
4       95.03
        ...
105    109.50
106     91.54
107    100.11
108    101.78
109     98.85
Name: Primary_Boys, Length: 110, dtype: float64
```

In [56]:
```
#Modifying/Changing Series values
```

In [57]:
```
a
```

Out[57]:
```
0    29
1    30
2    16
3    27
4    46
5    30
6    35
7    25
8    46
9    28
dtype: int32
```

In [58]:
```
a[4]=456  #Changed the fourth index
a
```

Out[58]:
```
0     29
1     30
2     16
3     27
4    456
5     30
6     35
7     25
8     46
9     28
dtype: int32
```

In [59]:
```
a[2:4]=45,67    #2 and 3 value changed
a
```

Out[59]:
```
0     29
1     30
2     45
3     67
4    456
5     30
6     35
7     25
8     46
9     28
dtype: int32
```

In [60]:
```
a[6:8]=np.mean(a)
a
```

Out[60]:
```
0     29.0
1     30.0
2     45.0
3     67.0
4    456.0
5     30.0
6     79.1
7     79.1
8     46.0
9     28.0
dtype: float64
```

'''Data Frame. The Data frame is the two-dimensional data structure, for example, the data is aligned in the tabular fashion in rows and columns'''

In [61]:
```
#Creation of data frame
```

In [23]:
```
#Creating Empty Data frame
df=pd.DataFrame()
print(df)
```

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

In [24]:
```
type(df)
```

Out[24]:
```
pandas.core.frame.DataFrame
```

In [64]:
```
#Creating Pandas DataFrame from lists of lists.
Teacher=[["Naveen",50000],["Raj",90000],["Niyati",45000]]
df=pd.DataFrame(Teacher,columns=['Name','Salary'])
df
```

Out[64]:

	Name	Salary
0	Naveen	50000
1	Raj	90000
2	Niyati	45000

In [25]:
```
#Creating Data Frame from dict of narray/lists
Teacher={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','CHARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUNJABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45000,34000]}
df=pd.DataFrame(Teacher)
df
```

Out[25]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [27]:
```
#Creating Data frame from list of dicts
data = [{'No': 1, 'Name': "Rama", 'Age':3}, {'No':3, 'Name': 20, 'Age': 30}]
df=pd.DataFrame(data)
df
```

Out[27]:

	No	Name	Age
0	1	Rama	3
1	3	20	30

In [67]:
```
#Specifying the column Sequence
df=pd.DataFrame(data,columns=['No','Name','Age'])
df
```

Out[67]:

	No	Name	Age
0	1	Rama	3
1	3	20	30

In [68]:
```
#Creating DataFrame from list of tuples
students = [ ('Jacky', 34, 'Chandigarh' , 'India') ,
             ('Riti', 30, 'Delhi' , 'India' ) ,
             ('Vikas', 31, 'Mumbai' , 'India' ) ,
             ('Neelu', 32, 'Bangalore' , 'India' ) ,
             ('Rajesh', 16, 'New York' , 'US') ,
             ('Mickey', 17, 'las vegas' , 'US')  ]
df=pd.DataFrame(students, columns = ['Name' , 'Age', 'City' , 'Country'])
df
```

Out[68]:

	Name	Age	City	Country
0	Jacky	34	Chandigarh	India
1	Riti	30	Delhi	India
2	Vikas	31	Mumbai	India
3	Neelu	32	Bangalore	India
4	Rajesh	16	New York	US
5	Mickey	17	las vegas	US

In [69]:
```
#Creating Data frame from CSV (Comma Separated Values)
df=pd.read_csv("E:\\dataset\\flight.csv")
df
```

Out[69]:

	FNO	Airline	Noofpassenger
0	1	Indigo	230000
1	2	SpiceJet	12000
2	3	Indian Airlines	240000
3	4	Lufthansa	245000
4	5	Air Asia	210000

In [70]:
```
#specfying the column header and ignoring the file header
df=pd.read_csv("E:\\dataset\\flight.csv",names=["Flight No","Airline","Passeng
er"],header=0)
df
```

Out[70]:

	Flight No	Airline	Passenger
0	1	Indigo	230000
1	2	SpiceJet	12000
2	3	Indian Airlines	240000
3	4	Lufthansa	245000
4	5	Air Asia	210000

In [71]:
```
#Dataframe Objects

#Transpose
df.T
```

Out[71]:

	0	1	2	3	4
Flight No	1	2	3	4	5
Airline	Indigo	SpiceJet	Indian Airlines	Lufthansa	Air Asia
Passenger	230000	12000	240000	245000	210000

In [73]:
```
#Axes
df.axes
```

Out[73]:
```
[RangeIndex(start=0, stop=5, step=1),
 Index(['Flight No', 'Airline', 'Passenger'], dtype='object')]
```

In [74]:
```
#columns
df.columns
```

Out[74]:
```
Index(['Flight No', 'Airline', 'Passenger'], dtype='object')
```

In [75]:
```
#Showing Sata Type
df.dtypes
```

Out[75]:
```
Flight No     int64
Airline      object
Passenger     int64
dtype: object
```

In [76]:
```
#iat - Access a single value for a row/column pair by integer position
df.iat[3,2]
```

Out[76]:
```
245000
```

In [77]:
```
df.iat[0,1]
```

Out[77]:
```
'Indigo'
```

In [78]:
```
#.iloc - Purely integer-location based indexing for selection by position.
df.iloc[1:3]
```

Out[78]:

	Flight No	Airline	Passenger
1	2	SpiceJet	12000
2	3	Indian Airlines	240000

In [29]:
```
#loc - Access a group of rows and columns by label(s) or a boolean array.
df.loc[1:3]
```

Out[29]:

	No	Name	Age
1	3	20	30

In [80]:
```
#Displaying index values
df.index
```

Out[80]:
```
RangeIndex(start=0, stop=5, step=1)
```

In [81]:
```
#Displaying the dimensions
df.ndim
```

Out[81]:
```
2
```

In [82]:
```
#Shape
df.shape
```

Out[82]:
```
(5, 3)
```

In [83]:
```
#Size
df.size
```

Out[83]:
```
15
```

In [84]:
```
#Displaying all the values as tuples
df.values
```

Out[84]:
```
array([[1, 'Indigo', 230000],
       [2, 'SpiceJet', 12000],
       [3, 'Indian Airlines', 240000],
       [4, 'Lufthansa', 245000],
       [5, 'Air Asia', 210000]], dtype=object)
```

In [85]:
```
#DataFrame functions

df.head()
```

Out[85]:

	Flight No	Airline	Passenger
0	1	Indigo	230000
1	2	SpiceJet	12000
2	3	Indian Airlines	240000
3	4	Lufthansa	245000
4	5	Air Asia	210000

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

Out[86]:

	Flight No	Airline	Passenger
0	1	Indigo	230000
1	2	SpiceJet	12000
2	3	Indian Airlines	240000
3	4	Lufthansa	245000
4	5	Air Asia	210000

In [87]:
```
df.count()
```

Out[87]:
```
Flight No    5
Airline      5
Passenger    5
dtype: int64
```

In [88]:
```
#Displaying DataFrame - Select Statement
```

In [30]:
```
Teacher={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','CHARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUNJABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45000,34000]}
df=pd.DataFrame(Teacher)
df
```

Out[30]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [90]:
```
print(df)
```

```
   TNO       TNAME                 TADD  SALARY
0  T01        AMIT     123 PASCHIM VIHAR   23000
1  TO2      RAJESH     6/11 RAMESH NAGAR   34000
2  T03       BINNY  5 WEST PUNJABHI BAGH   12000
3  T04       CHARU       23 MALVIYA NAGAR   45000
4  TO5   MEENAKSHI          19 MEERA BAGH   34000
```

In [91]:
```
#Loops - Iteration
```

In [31]:
```
#iterrows()
sum1=0
for (index,values) in df.iterrows():
    print("Row index:",index,"\nRows =",values)
    sum1+=values[3]     #Iteration on Salaries
print("Sum of the Salary",sum1)
```

```
Row index: 0
Rows = TNO                    T01
TNAME                   AMIT
TADD      123 PASCHIM VIHAR
SALARY                23000
Name: 0, dtype: object
Row index: 1
Rows = TNO                    TO2
TNAME                 RAJESH
TADD      6/11 RAMESH NAGAR
SALARY                34000
Name: 1, dtype: object
Row index: 2
Rows = TNO                       T03
TNAME                    BINNY
TADD      5 WEST PUNJABHI BAGH
SALARY                   12000
Name: 2, dtype: object
Row index: 3
Rows = TNO                     T04
TNAME                 CHARU
TADD      23 MALVIYA NAGAR
SALARY                45000
Name: 3, dtype: object
Row index: 4
Rows = TNO                  TO5
TNAME          MEENAKSHI
TADD      19 MEERA BAGH
SALARY             34000
Name: 4, dtype: object
Sum of the Salary 148000
```

In [93]:
```
#iteritems  - Columns
for (index) in df.iteritems():
    print("Column Name:",index,"\n")
```

```
Column Name: ('TNO', 0    T01
1    TO2
2    T03
3    T04
4    TO5
Name: TNO, dtype: object)

Column Name: ('TNAME', 0         AMIT
1       RAJESH
2        BINNY
3        CHARU
4    MEENAKSHI
Name: TNAME, dtype: object)

Column Name: ('TADD', 0       123 PASCHIM VIHAR
1       6/11 RAMESH NAGAR
2    5 WEST PUNJABHI BAGH
3        23 MALVIYA NAGAR
4           19 MEERA BAGH
Name: TADD, dtype: object)

Column Name: ('SALARY', 0    23000
1    34000
2    12000
3    45000
4    34000
Name: SALARY, dtype: int64)
```

In [32]:
```
#Projections - Displaying the columns
#We will use given DataFrame
df=pd.read_csv("e:\\dataset\\gross-enrollment-ratio-2013-2016_1.csv")
df
```

Out[32]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Upper
0	Andaman & Nicobar Islands	2013-14	95.88	91.97	93.93	94.70	
1	Andhra Pradesh	2013-14	96.62	96.87	96.74	82.81	
2	Arunachal Pradesh	2013-14	129.12	127.77	128.46	112.64	
3	Assam	2013-14	111.77	115.16	113.43	87.85	
4	Bihar	2013-14	95.03	101.15	97.96	80.60	
...	...	...	...	...	...	...	
105	Tripura	2014-15	109.50	110.49	109.98	118.80	
106	Uttar Pradesh	2014-15	91.54	98.93	95.00	68.14	
107	Uttaranchal	2014-15	100.11	101.02	100.54	84.20	
108	West Bengal	2014-15	101.78	102.91	102.33	96.16	
109	All India	2014-15	98.85	101.43	100.08	87.71	

110 rows × 14 columns

In [95]:
```
df.columns
```

Out[95]:
```
Index(['State_UT', 'Year', 'Primary_Boys', 'Primary_Girls', 'Primary_Total',
       'Upper_Primary_Boys', 'Upper_Primary_Girls', 'Upper_Primary_Total',
       'Secondary_Boys', 'Secondary_Girls', 'Secondary_Total',
       'Higher_Secondary_Boys', 'Higher_Secondary_Girls',
       'Higher_Secondary_Total'],
      dtype='object')
```

In [96]:
```
#attribute access
print(df.Year)
df.Year.unique()
```

```
0      2013-14
1      2013-14
2      2013-14
3      2013-14
4      2013-14
        ...
105    2014-15
106    2014-15
107    2014-15
108    2014-15
109    2014-15
Name: Year, Length: 110, dtype: object
```

Out[96]:
```
array(['2013-14', '2015-16', '2014-15'], dtype=object)
```

In [97]:
```
#Through Index value
#accessing through column index label
df['State_UT'].unique
```

Out[97]:
```
0      Andaman & Nicobar Islands
1                 Andhra Pradesh
2              Arunachal Pradesh
3                          Assam
4                          Bihar
                 ...
105                      Tripura
106                Uttar Pradesh
107                  Uttaranchal
108                  West Bengal
109                    All India
Name: State_UT, Length: 110, dtype: object
```

In [98]:
```
#Selecting Multiple Columns
df[['State_UT', 'Year', 'Primary_Boys']]
```

Out[98]:

	State_UT	Year	Primary_Boys
0	Andaman & Nicobar Islands	2013-14	95.88
1	Andhra Pradesh	2013-14	96.62
2	Arunachal Pradesh	2013-14	129.12
3	Assam	2013-14	111.77
4	Bihar	2013-14	95.03
...	...	...	...
105	Tripura	2014-15	109.50
106	Uttar Pradesh	2014-15	91.54
107	Uttaranchal	2014-15	100.11
108	West Bengal	2014-15	101.78
109	All India	2014-15	98.85

110 rows × 3 columns

In [99]:
```
#Data Frame Slicing
df[2:4]
```

Out[99]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Upper_Pri
2	Arunachal Pradesh	2013-14	129.12	127.77	128.46	112.64	
3	Assam	2013-14	111.77	115.16	113.43	87.85	

In [100]:
```
#Selecting Particualr columns based on slicing
df[['State_UT', 'Year', 'Primary_Boys']][2:4]
```

Out[100]:

	State_UT	Year	Primary_Boys
2	Arunachal Pradesh	2013-14	129.12
3	Assam	2013-14	111.77

In [101]:
```
df[['State_UT', 'Year', 'Primary_Boys']][1::3]
```

Out[101]:

	State_UT	Year	Primary_Boys
1	Andhra Pradesh	2013-14	96.62
4	Bihar	2013-14	95.03
7	Dadra & Nagar Haveli	2013-14	89.50
10	Goa	2013-14	104.97
13	Himachal Pradesh	2013-14	99.80
16	Karnataka	2013-14	101.18
19	Madhya Pradesh	2013-14	111.85
22	Meghalaya	2013-14	132.89
25	Odisha	2013-14	107.15
28	Rajasthan	2013-14	102.35
31	Tripura	2013-14	112.70
34	West Bengal	2013-14	103.16
37	Andhra Pradesh	2015-16	84.88
40	Bihar	2015-16	104.35
43	Dadra & Nagar Haveli	2015-16	84.69
46	Goa	2015-16	100.89
49	Himachal Pradesh	2015-16	97.97
52	Karnataka	2015-16	102.93
55	MADHYA PRADESH	2015-16	95.35
58	Meghalaya	2015-16	138.75
61	Odisha	2015-16	104.91
64	Rajasthan	2015-16	101.27
67	Telangana	2015-16	103.13
70	Uttarakhand	2015-16	98.87
73	Andaman & Nicobar Islands	2014-15	92.91
76	Assam	2014-15	113.39
79	Chhattisgarh	2014-15	103.30
82	Delhi	2014-15	109.38
85	Haryana	2014-15	95.92
88	Jharkhand	2014-15	107.77
91	Lakshadweep	2014-15	78.76
94	Manipur	2014-15	132.47
97	Nagaland	2014-15	99.32
100	Punjab	2014-15	103.40
103	Tamil Nadu	2014-15	102.79

	State_UT	Year	Primary_Boys
106	Uttar Pradesh	2014-15	91.54
109	All India	2014-15	98.85

In [102]:
```
#Reversing the dataframe
df[['State_UT', 'Year', 'Primary_Boys']][::-1]
```

Out[102]:

	State_UT	Year	Primary_Boys
109	All India	2014-15	98.85
108	West Bengal	2014-15	101.78
107	Uttaranchal	2014-15	100.11
106	Uttar Pradesh	2014-15	91.54
105	Tripura	2014-15	109.50
...	...	...	...
4	Bihar	2013-14	95.03
3	Assam	2013-14	111.77
2	Arunachal Pradesh	2013-14	129.12
1	Andhra Pradesh	2013-14	96.62
0	Andaman & Nicobar Islands	2013-14	95.88

110 rows × 3 columns

In [103]:
```
#Boolean Indexing on DataFrame
df['Primary_Boys']<10
```

Out[103]:
```
0      False
1      False
2      False
3      False
4      False
       ...
105    False
106    False
107    False
108    False
109    False
Name: Primary_Boys, Length: 110, dtype: bool
```

In [104]:
```
df['Primary_Boys']==1
```

Out[104]:
```
0      False
1      False
2      False
3      False
4      False
       ...
105    False
106    False
107    False
108    False
109    False
Name: Primary_Boys, Length: 110, dtype: bool
```

In [105]:
```
df['Primary_Boys']<10
```

Out[105]:
```
0      False
1      False
2      False
3      False
4      False
       ...
105    False
106    False
107    False
108    False
109    False
Name: Primary_Boys, Length: 110, dtype: bool
```

In [106]:
```
df[df['Primary_Boys']<90]
```

Out[106]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Uppe
5	Chandigarh	2013-14	88.42	96.09	91.85	99.93	
7	Dadra & Nagar Haveli	2013-14	89.50	81.89	85.78	100.76	
8	Daman & Diu	2013-14	87.80	89.78	88.69	84.06	
14	Jammu And Kashmir	2013-14	84.03	85.88	84.90	71.64	
18	Lakshadweep	2013-14	83.42	77.76	80.59	100.53	
26	Pondicherry	2013-14	89.53	95.38	92.29	92.38	
37	Andhra Pradesh	2015-16	84.88	84.05	84.48	81.12	
41	Chandigarh	2015-16	77.42	86.57	81.44	90.42	
43	Dadra & Nagar Haveli	2015-16	84.69	80.21	82.53	93.71	
44	Daman & Diu	2015-16	79.68	84.95	82.03	74.86	
48	Haryana	2015-16	89.96	93.21	91.41	87.39	
50	Jammu And Kashmir	2015-16	84.86	87.24	85.98	68.77	
54	Lakshadweep	2015-16	77.90	69.90	73.80	92.53	
62	Puducherry	2015-16	80.20	90.23	84.79	82.41	
69	Uttar Pradesh	2015-16	88.63	96.16	92.15	68.24	
74	Andhra Pradesh	2014-15	88.31	88.11	88.21	79.34	
78	Chandigarh	2014-15	83.62	91.49	87.11	94.68	
80	Dadra & Nagar Haveli	2014-15	85.41	78.78	82.18	97.19	
81	Daman & Diu	2014-15	83.93	87.37	85.47	80.28	
87	Jammu And Kashmir	2014-15	85.06	87.00	85.97	69.82	
91	Lakshadweep	2014-15	78.76	74.21	76.45	96.20	
99	Pondicherry	2014-15	84.49	92.40	88.16	86.87	

In [107]:
```
df[df['Primary_Boys']>=100]
```

Out[107]:

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Uppe
2	Arunachal Pradesh	2013-14	129.12	127.77	128.46	112.64	
3	Assam	2013-14	111.77	115.16	113.43	87.85	
6	Chhattisgarh	2013-14	104.06	103.93	103.99	100.35	
9	Delhi	2013-14	108.78	112.95	110.67	117.76	
10	Goa	2013-14	104.97	105.32	105.14	106.29	
11	Gujarat	2013-14	100.32	102.06	101.13	91.82	
15	Jharkhand	2013-14	109.57	110.92	110.23	91.70	
16	Karnataka	2013-14	101.18	100.73	100.96	91.72	
19	Madhya Pradesh	2013-14	111.85	111.09	111.49	96.04	
21	Manipur	2013-14	145.68	152.81	149.15	111.62	
22	Meghalaya	2013-14	132.89	137.89	135.35	102.24	
23	Mizoram	2013-14	127.88	123.89	125.96	119.38	
24	Nagaland	2013-14	116.66	121.05	118.78	99.06	
25	Odisha	2013-14	107.15	104.48	105.84	86.44	
27	Punjab	2013-14	104.33	107.20	105.61	94.04	
28	Rajasthan	2013-14	102.35	100.60	101.53	86.83	
29	Sikkim	2013-14	128.15	120.58	124.42	132.96	
30	Tamil Nadu	2013-14	102.40	102.72	102.56	97.16	
31	Tripura	2013-14	112.70	113.95	113.31	113.20	
34	West Bengal	2013-14	103.16	104.88	104.00	92.84	
35	All India	2013-14	100.20	102.65	101.36	86.31	
38	Arunachal Pradesh	2015-16	127.61	125.88	126.76	127.14	
39	Assam	2015-16	104.70	107.59	106.11	87.65	

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Uppe
40	Bihar	2015-16	104.35	111.30	107.67	98.21	
42	Chhattisgarh	2015-16	100.17	99.87	100.02	101.62	
45	Delhi	2015-16	108.04	113.93	110.71	118.86	
46	Goa	2015-16	100.89	104.45	102.57	96.83	
51	Jharkhand	2015-16	108.56	109.92	109.22	97.75	
52	Karnataka	2015-16	102.93	103.04	102.98	92.43	
57	Manipur	2015-16	128.91	132.90	130.85	127.00	
58	Meghalaya	2015-16	138.75	143.12	140.90	126.00	
59	Mizoram	2015-16	124.91	121.00	122.99	135.90	
61	Odisha	2015-16	104.91	102.50	103.73	94.86	
64	Rajasthan	2015-16	101.27	99.48	100.43	91.46	
65	Sikkim	2015-16	107.27	98.32	102.87	143.72	
66	Tamil Nadu	2015-16	103.39	104.43	103.89	92.55	
67	Telangana	2015-16	103.13	102.90	103.02	88.61	
68	Tripura	2015-16	107.58	108.36	107.96	125.75	
71	West Bengal	2015-16	103.13	104.26	103.68	97.90	
75	Arunachal Pradesh	2014-15	128.78	127.45	128.13	120.11	
76	Assam	2014-15	113.39	116.60	114.96	90.10	
79	Chhattisgarh	2014-15	103.30	102.85	103.08	100.87	
82	Delhi	2014-15	109.38	114.61	111.75	118.08	
83	Goa	2014-15	103.03	105.01	103.97	99.23	
88	Jharkhand	2014-15	107.77	109.08	108.40	95.81	
89	Karnataka	2014-15	101.89	101.84	101.86	92.53	

	State_UT	Year	Primary_Boys	Primary_Girls	Primary_Total	Upper_Primary_Boys	Uppe
92	Madhya Pradesh	2014-15	101.94	100.20	101.11	92.69	
94	Manipur	2014-15	132.47	136.38	134.37	115.39	
95	Meghalaya	2014-15	136.19	140.69	138.40	113.28	
96	Mizoram	2014-15	124.80	120.44	122.66	127.63	
98	Odisha	2014-15	106.88	104.12	105.53	90.47	
100	Punjab	2014-15	103.40	107.26	105.11	94.56	
102	Sikkim	2014-15	116.95	108.05	112.57	133.96	
103	Tamil Nadu	2014-15	102.79	103.45	103.11	93.37	
104	Telangana	2014-15	103.55	103.59	103.57	89.94	
105	Tripura	2014-15	109.50	110.49	109.98	118.80	
107	Uttaranchal	2014-15	100.11	101.02	100.54	84.20	
108	West Bengal	2014-15	101.78	102.91	102.33	96.16	

In [108]:
```
df['State_UT'][df['Primary_Boys']>=110]
```

Out[108]:
```
2      Arunachal Pradesh
3                  Assam
19        Madhya Pradesh
21               Manipur
22             Meghalaya
23               Mizoram
24              Nagaland
29                Sikkim
31               Tripura
38     Arunachal Pradesh
57               Manipur
58             Meghalaya
59               Mizoram
75     Arunachal Pradesh
76                 Assam
94               Manipur
95             Meghalaya
96               Mizoram
102               Sikkim
Name: State_UT, dtype: object
```

In [109]:
```
df[['State_UT','Year']][(df['Primary_Boys']>=120) & (df['Primary_Girls']>=120
)]
```

Out[109]:

	State_UT	Year
2	Arunachal Pradesh	2013-14
21	Manipur	2013-14
22	Meghalaya	2013-14
23	Mizoram	2013-14
29	Sikkim	2013-14
38	Arunachal Pradesh	2015-16
57	Manipur	2015-16
58	Meghalaya	2015-16
59	Mizoram	2015-16
75	Arunachal Pradesh	2014-15
94	Manipur	2014-15
95	Meghalaya	2014-15
96	Mizoram	2014-15

In [33]:
```
#Adding a column

emp={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','C
HARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUN
JABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45
000,34000]}
df=pd.DataFrame(emp,columns=['TNO','TNAME','TADD','SALARY'])
df
```

Out[33]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [34]:
```
#Using Column index
df['DESIG']=['Manager','Salesman','Manager','HR','Director']
df
```

Out[34]:

	TNO	TNAME	TADD	SALARY	DESIG
0	T01	AMIT	123 PASCHIM VIHAR	23000	Manager
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000	Salesman
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000	Manager
3	T04	CHARU	23 MALVIYA NAGAR	45000	HR
4	TO5	MEENAKSHI	19 MEERA BAGH	34000	Director

In [112]:
```
#Calculating field
df['TAX']=df.SALARY*20/100
df
```

Out[112]:

	TNO	TNAME	TADD	SALARY	DESIG	TAX
0	T01	AMIT	123 PASCHIM VIHAR	23000	Manager	4600.0
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000	Salesman	6800.0
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000	Manager	2400.0
3	T04	CHARU	23 MALVIYA NAGAR	45000	HR	9000.0
4	TO5	MEENAKSHI	19 MEERA BAGH	34000	Director	6800.0

In [113]:
```
#Inserting column using Series
Comm=np.array([2000,3000,1000,np.nan,2000])
df['COM']=Comm
df
```

Out[113]:

	TNO	TNAME	TADD	SALARY	DESIG	TAX	COM
0	T01	AMIT	123 PASCHIM VIHAR	23000	Manager	4600.0	2000.0
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000	Salesman	6800.0	3000.0
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000	Manager	2400.0	1000.0
3	T04	CHARU	23 MALVIYA NAGAR	45000	HR	9000.0	NaN
4	TO5	MEENAKSHI	19 MEERA BAGH	34000	Director	6800.0	2000.0

In [114]:
```
#insert Function

emp={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','C
HARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUN
JABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45
000,34000]}
df=pd.DataFrame(emp,columns=['TNO','TNAME','TADD','SALARY'])
df
```

Out[114]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [115]:
```
df.insert(2,"DESIG",['Manager','Salesman','Manager','HR','Director'])
df
```

Out[115]:

	TNO	TNAME	DESIG	TADD	SALARY
0	T01	AMIT	Manager	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	Salesman	6/11 RAMESH NAGAR	34000
2	T03	BINNY	Manager	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	HR	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	Director	19 MEERA BAGH	34000

In [116]:
```
id1=3
tax=df.SALARY*20/100
df.insert(id1,"TAX",tax)
df
```

Out[116]:

	TNO	TNAME	DESIG	TAX	TADD	SALARY
0	T01	AMIT	Manager	4600.0	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	Salesman	6800.0	6/11 RAMESH NAGAR	34000
2	T03	BINNY	Manager	2400.0	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	HR	9000.0	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	Director	6800.0	19 MEERA BAGH	34000

In [117]:
```
#Assign Function
df.assign(Comm=[2300,3400,np.nan,4000,4500])
```

Out[117]:

	TNO	TNAME	DESIG	TAX	TADD	SALARY	Comm
0	T01	AMIT	Manager	4600.0	123 PASCHIM VIHAR	23000	2300.0
1	TO2	RAJESH	Salesman	6800.0	6/11 RAMESH NAGAR	34000	3400.0
2	T03	BINNY	Manager	2400.0	5 WEST PUNJABHI BAGH	12000	NaN
3	T04	CHARU	HR	9000.0	23 MALVIYA NAGAR	45000	4000.0
4	TO5	MEENAKSHI	Director	6800.0	19 MEERA BAGH	34000	4500.0

In [118]:
```
#Using Dictionary
state={'DELHI':'BINNY','KERALA':'RAJESH','PUNJAB':'CHARU','HARYANA':'MEENAKSH
I','Delhi':'AMIT'}
df['State']=state
df
```

Out[118]:

	TNO	TNAME	DESIG	TAX	TADD	SALARY	State
0	T01	AMIT	Manager	4600.0	123 PASCHIM VIHAR	23000	DELHI
1	TO2	RAJESH	Salesman	6800.0	6/11 RAMESH NAGAR	34000	KERALA
2	T03	BINNY	Manager	2400.0	5 WEST PUNJABHI BAGH	12000	PUNJAB
3	T04	CHARU	HR	9000.0	23 MALVIYA NAGAR	45000	HARYANA
4	TO5	MEENAKSHI	Director	6800.0	19 MEERA BAGH	34000	Delhi

In [119]:
```
#. Using loc() function
df.loc[:,"Desig"]=['Manager','Salesman','Manager','HR','Director']
df
```

Out[119]:

	TNO	TNAME	DESIG	TAX	TADD	SALARY	State	Desig
0	T01	AMIT	Manager	4600.0	123 PASCHIM VIHAR	23000	DELHI	Manager
1	TO2	RAJESH	Salesman	6800.0	6/11 RAMESH NAGAR	34000	KERALA	Salesman
2	T03	BINNY	Manager	2400.0	5 WEST PUNJABHI BAGH	12000	PUNJAB	Manager
3	T04	CHARU	HR	9000.0	23 MALVIYA NAGAR	45000	HARYANA	HR
4	TO5	MEENAKSHI	Director	6800.0	19 MEERA BAGH	34000	Delhi	Director

In [120]:
```
df
```

Out[120]:

	TNO	TNAME	DESIG	TAX	TADD	SALARY	State	Desig
0	T01	AMIT	Manager	4600.0	123 PASCHIM VIHAR	23000	DELHI	Manager
1	TO2	RAJESH	Salesman	6800.0	6/11 RAMESH NAGAR	34000	KERALA	Salesman
2	T03	BINNY	Manager	2400.0	5 WEST PUNJABHI BAGH	12000	PUNJAB	Manager
3	T04	CHARU	HR	9000.0	23 MALVIYA NAGAR	45000	HARYANA	HR
4	TO5	MEENAKSHI	Director	6800.0	19 MEERA BAGH	34000	Delhi	Director

In [121]:
```
#Adding a row in a Data Frame
emp={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','C
HARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUN
JABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45
000,34000]}
df=pd.DataFrame(emp,columns=['TNO','TNAME','TADD','SALARY'])
df
```

Out[121]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [122]:
```
#Using .loc() or iloc () and list - Insert Command
df.loc[5]=['T05','NAVEEN','97 TAGORE GARDEN',89900]
df
```

Out[122]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000
5	T05	NAVEEN	97 TAGORE GARDEN	89900

In [123]:
```
#Add row in the dataframe using dataframe.append() and Dictionary
#df.append({"TNAME":"Ritika","TADD":"Texas"},ignore_index=True)
#df
```

In [124]:
```
s=pd.Series(['T07','Anita','Jaipur',20000],index=df.columns)
df.append(s,ignore_index=True)
```

Out[124]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000
5	T05	NAVEEN	97 TAGORE GARDEN	89900
6	T07	Anita	Jaipur	20000

In [125]:
```
multiple = [pd.Series(['T08','Raju','Bangalore',28000], index=df.columns ) ,
                pd.Series(['T09','Sam', 'Tokyo', 39000], index=df.columns ) ,
                pd.Series(['T010','Rocky', 'Las Vegas',35000], index=df.column
s ) ]
multiple
```

Out[125]:
```
[TNO             T08
 TNAME          Raju
 TADD      Bangalore
 SALARY        28000
 dtype: object,
 TNO          T09
 TNAME        Sam
 TADD       Tokyo
 SALARY     39000
 dtype: object,
 TNO            T010
 TNAME         Rocky
 TADD      Las Vegas
 SALARY        35000
 dtype: object]
```

In [126]:
```
df=df.append(multiple,ignore_index=True)
df
```

Out[126]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000
5	T05	NAVEEN	97 TAGORE GARDEN	89900
6	T08	Raju	Bangalore	28000
7	T09	Sam	Tokyo	39000
8	T010	Rocky	Las Vegas	35000

In [127]:
```
#Adding new columns using concat method

d={'TNO':['T07','TO8'],'TNAME':['MANOJ','BHAWNA'],'TADD':['Dwarka','Rohini'],
'SALARY':[45000,30000]}
df1=pd.DataFrame(d)
df1
```

Out[127]:

	TNO	TNAME	TADD	SALARY
0	T07	MANOJ	Dwarka	45000
1	TO8	BHAWNA	Rohini	30000

In [128]:
```
pd.concat((df,df1),ignore_index=True)
df
```

Out[128]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000
5	T05	NAVEEN	97 TAGORE GARDEN	89900
6	T08	Raju	Bangalore	28000
7	T09	Sam	Tokyo	39000
8	T010	Rocky	Las Vegas	35000

In [129]:
```
#Adding a Column
tax=df.SALARY*20/100
dtax=pd.DataFrame({"Tax":tax})
pd.concat((df,dtax),axis=1)
```

Out[129]:

	TNO	TNAME	TADD	SALARY	Tax
0	T01	AMIT	123 PASCHIM VIHAR	23000	4600.0
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000	6800.0
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000	2400.0
3	T04	CHARU	23 MALVIYA NAGAR	45000	9000.0
4	TO5	MEENAKSHI	19 MEERA BAGH	34000	6800.0
5	T05	NAVEEN	97 TAGORE GARDEN	89900	17980.0
6	T08	Raju	Bangalore	28000	5600.0
7	T09	Sam	Tokyo	39000	7800.0
8	T010	Rocky	Las Vegas	35000	7000.0

In [130]:
```
#DataFrame.Drop
#column Dropping

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

Out[130]:

	TNO	TNAME	SALARY
0	T01	AMIT	23000
1	TO2	RAJESH	34000
2	T03	BINNY	12000
3	T04	CHARU	45000
4	TO5	MEENAKSHI	34000
5	T05	NAVEEN	89900
6	T08	Raju	28000
7	T09	Sam	39000
8	T010	Rocky	35000

In [131]:
```
#Dropping with index value
df.drop(df.columns[1],axis=1)
```

Out[131]:

	TNO	SALARY
0	T01	23000
1	TO2	34000
2	T03	12000
3	T04	45000
4	TO5	34000
5	T05	89900
6	T08	28000
7	T09	39000
8	T010	35000

In [132]:
```
#Dropping by multiple columns names
#df.drop(columns=['TNAME','SALARY'],axis=1)
```

In [133]:
```
#Dropping Rows
emp={'TNO':['T01','TO2','T03','T04','TO5'],'TNAME':['AMIT','RAJESH','BINNY','C
HARU','MEENAKSHI'],'TADD':['123 PASCHIM VIHAR','6/11 RAMESH NAGAR','5 WEST PUN
JABHI BAGH','23 MALVIYA NAGAR','19 MEERA BAGH'],'SALARY':[23000,34000,12000,45
000,34000]}
df=pd.DataFrame(emp,columns=['TNO','TNAME','TADD','SALARY'])
df
```

Out[133]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [134]:
```
#a.     Dropping by a row number
df.drop([1,3])
```

Out[134]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [135]:
```
#b.     Dropping by an index number
df.drop([3])
```

Out[135]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [136]:
```
#c.     Dropping rows by position
df.drop(df.index[0:2])
```

Out[136]:

	TNO	TNAME	TADD	SALARY
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [137]:
```
# this will delete all the rows from -5 to -2, only -1 row will be available
df.drop(df.index[:-1])
```

Out[137]:

	TNO	TNAME	TADD	SALARY
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [138]:
```
#       Dropping rows based on the conditions
df.drop(df[df['SALARY'] < 25000].index)
```

Out[138]:

	TNO	TNAME	TADD	SALARY
1	TO2	RAJESH	6/11 RAMESH NAGAR	34000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [139]:
```
#Multiple Condtions
df.drop(df[(df['TNAME'] == 'RAJESH') & (df['SALARY']>25000)].index,inplace=True)
df
```

Out[139]:

	TNO	TNAME	TADD	SALARY
0	T01	AMIT	123 PASCHIM VIHAR	23000
2	T03	BINNY	5 WEST PUNJABHI BAGH	12000
3	T04	CHARU	23 MALVIYA NAGAR	45000
4	TO5	MEENAKSHI	19 MEERA BAGH	34000

In [140]:
```
d={'TNO':['T07','TO8'],'TNAME':['MANOJ','BHAWNA'],'TADD':['Dwarka','Rohini'],
'SALARY':[45000,30000]}
df1=pd.DataFrame(d)
df1
```

Out[140]:

	TNO	TNAME	TADD	SALARY
0	T07	MANOJ	Dwarka	45000
1	TO8	BHAWNA	Rohini	30000

In [141]:
```
df1.to_csv("E:\\teacher.csv")
print("Data Transfered")
```

```
Data Transfered
```

PANDAS SESSION - BY NAVEEN GUPTA

M-9971999404 , E-mail:gupta.naveen1@gmail.com

youtube.com/c/NaveenGupta , https://wakelet.com/@NaveenGupta429 (https://wakelet.com/@NaveenGupta429)