DataFrame Revision

1. Basic Operations

```
import pandas as pd
d={ 'Admno': [123,514,981,265,225,122],
    'FirstName':['Sumit','Nidhi','Reena','Ravi','Sanjay','Vivek'],
    'LastName':['Roy','Singh','Kumari','Kumar','Das','Kumar'],
    'Class':['X' , 'XII' , 'XI' , 'XI' , 'XII' , 'X']
  }
n=pd.DataFrame(d)
print(n)

# accessing columns
print("------O/P 1--------")
print(n.FirstName)
print("------O/P 2--------")
print(n[['FirstName']])
# accessing rows & columns using loc
print("------O/P 3--------")
print(n.loc[:,'Admno':'LastName'])
print("------O/P 4--------")
print(n.loc[:,['Admno','LastName']])
print("------O/P 5--------")
print(n.loc[0:2,:])
print("------O/P 6--------")
print(n.loc[[0,2],:])
# accessing rows & columns using iloc
print("------O/P 7--------")
print(n.iloc[0:2,:])
print("------O/P 8--------")
print(n.iloc[0:2,1:2])
print("------O/P 9--------")
print(n.iloc[[0,2],:])
# accessing values
print("------O/P 10--------")
print(n.FirstName[2])
print("------O/P 11--------")
print(n.iat[2,1])
#adding columns
print("------O/P 12--------")
n['Marks']=20
print(n)
print("------O/P 13--------")
n.loc[:,'Marks1']=[50,60,70,80,90,80]
print(n)
#adding rows
print("------O/P 14--------")
n.loc[6,:]= [12 , 20 , 30, 40, 50, 60]
print(n)
```

```
#deleting rows and columns
print("------O/P 15--------")
n=n.drop([6])
print(n)
n=n.drop(['Marks','Marks1'], axis=1)
print(n)
#properties of Dataframe
print("------O/P 16--------")
print(n.index)
print("------O/P 17--------")
print(n.columns)
print("------O/P 18--------")
print(n.size)
print("------O/P 19--------")
print(n.shape)
print("------O/P 20--------")
print(n.dtypes)
print("------O/P 21--------")
print(n.ndim)
print("------O/P 22--------")
print(len(n))
print("------O/P 23--------")
print(len(n))
print("------O/P 24--------")
print(n.count(0))
print(n.count(1))
```

Output:

```
  Admno FirstName LastName Class
0   123    Sumit      Roy     X
1   514    Nidhi    Singh   XII
2   981    Reena   Kumari    XI
3   265     Ravi    Kumar    XI
4   225   Sanjay      Das   XII
5   122    Vivek    Kumar     X
------O/P 1--------
0    Sumit
1    Nidhi
2    Reena
3     Ravi
4   Sanjay
5    Vivek
Name: FirstName, dtype: object
------O/P 2--------
  FirstName
0    Sumit
1    Nidhi
2    Reena
3     Ravi
4   Sanjay
5    Vivek
```

------O/P 3--------

	Admno	FirstName	LastName
0	123	Sumit	Roy
1	514	Nidhi	Singh
2	981	Reena	Kumari
3	265	Ravi	Kumar
4	225	Sanjay	Das
5	122	Vivek	Kumar

------O/P 4--------

	Admno	LastName
0	123	Roy
1	514	Singh
2	981	Kumari
3	265	Kumar
4	225	Das
5	122	Kumar

------O/P 5--------

	Admno	FirstName	LastName	Class
0	123	Sumit	Roy	X
1	514	Nidhi	Singh	XII
2	981	Reena	Kumari	XI

------O/P 6--------

	Admno	FirstName	LastName	Class
0	123	Sumit	Roy	X
2	981	Reena	Kumari	XI

------O/P 7--------

	Admno	FirstName	LastName	Class
0	123	Sumit	Roy	X
1	514	Nidhi	Singh	XII

------O/P 8--------

	FirstName
0	Sumit
1	Nidhi

------O/P 9--------

	Admno	FirstName	LastName	Class
0	123	Sumit	Roy	X
2	981	Reena	Kumari	XI

------O/P 10--------

Reena

------O/P 11--------

Reena

------O/P 12--------

	Admno	FirstName	LastName	Class	Marks
0	123	Sumit	Roy	X	20
1	514	Nidhi	Singh	XII	20
2	981	Reena	Kumari	XI	20
3	265	Ravi	Kumar	XI	20
4	225	Sanjay	Das	XII	20
5	122	Vivek	Kumar	X	20

------O/P 13--------

	Admno	FirstName	LastName	Class	Marks	Marks1
0	123	Sumit	Roy	X	20	50
1	514	Nidhi	Singh	XII	20	60

2 981 Reena Kumari XI 20 70
3 265 Ravi Kumar XI 20 80
4 225 Sanjay Das XII 20 90
5 122 Vivek Kumar X 20 80

------O/P 14--------

	Admno	FirstName	LastName	Class	Marks	Marks1
0	123.0	Sumit	Roy	X	20.0	50.0
1	514.0	Nidhi	Singh	XII	20.0	60.0
2	981.0	Reena	Kumari	XI	20.0	70.0
3	265.0	Ravi	Kumar	XI	20.0	80.0
4	225.0	Sanjay	Das	XII	20.0	90.0
5	122.0	Vivek	Kumar	X	20.0	80.0
6	12.0	20	30	40	50.0	60.0

------O/P 15--------

	Admno	FirstName	LastName	Class	Marks	Marks1
0	123.0	Sumit	Roy	X	20.0	50.0
1	514.0	Nidhi	Singh	XII	20.0	60.0
2	981.0	Reena	Kumari	XI	20.0	70.0
3	265.0	Ravi	Kumar	XI	20.0	80.0
4	225.0	Sanjay	Das	XII	20.0	90.0
5	122.0	Vivek	Kumar	X	20.0	80.0

	Admno	FirstName	LastName	Class
0	123.0	Sumit	Roy	X
1	514.0	Nidhi	Singh	XII
2	981.0	Reena	Kumari	XI
3	265.0	Ravi	Kumar	XI
4	225.0	Sanjay	Das	XII
5	122.0	Vivek	Kumar	X

------O/P 16--------

Int64Index([0, 1, 2, 3, 4, 5], dtype='int64')

------O/P 17--------

Index(['Admno', 'FirstName', 'LastName', 'Class'], dtype='object')

------O/P 18--------

24

------O/P 19--------

(6, 4)

------O/P 20--------

Admno float64
FirstName object
LastName object
Class object
dtype: object

------O/P 21--------

2

------O/P 22--------

6

------O/P 23--------

6

------O/P 24--------

Admno 6
FirstName 6
LastName 6
Class 6
dtype: int64
0 4

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

2. DataFrame Functions

```
import pandas as pd
import numpy as np
d={2016: {'Qtr1':34500 , 'Qtr2':56000  , 'Qtr3':47000  , 'Qtr4':49000},
    2017 : {'Qtr1': 44900 , 'Qtr2' : 46100  , 'Qtr3': 57000  , 'Qtr4' : 59000},
    2018 : {'Qtr1' : 54500  , 'Qtr2' : 51000  , 'Qtr3' : 57000  , 'Qtr4' : 58500},
    2019 : {'Qtr1': 61000}
  }
sal=pd.DataFrame(d)
print(sal)
print("-------- O/P 1 --------")
print(sal.min(axis=0))
print("-------- O/P 2 --------")
print(sal.max(axis=1))
print("-------- O/P 3 --------")
print(sal.mode(axis=1))
print("-------- O/P 4 --------")
print(sal.mean())
print("-------- O/P 5 --------")
print(sal.median())
print("-------- O/P 6 --------")
print(sal.count(axis=1))
print("-------- O/P 7 --------")
print(sal.sum())
print("-------- O/P 8 --------")
print(sal.var())
#renaming index and column labels (inplace attribute)
print("-------- O/P 9 --------")
print(sal.rename(index={'Qtr1':'Q1' , 'Qtr2' : 'Q2'} ))
print("-------- O/P 10 --------")
print(sal.rename({2016 : 16 }, axis=1))
#Changing order of index of existing labels/indices (fill-value attribute)
print(sal)
print("-------- O/P 11 --------")
print(sal.reindex(index=['Qtr3', 'Qtr2','Qtr1','Qtr4','Qtr5'], fill_value=1000))
print("-------- O/P 12 --------")
print(sal.reindex(columns=[2019,2018,2017,2016,2015],  fill_value=1000))
print(sal)

#pipe( ) function
print("-------- O/P 13 --------")
print(sal.multiply(sal.add(30),3 ))
print("-------- O/P 14 --------")
print(sal.pipe(np.add, 30).pipe(np.multiply , 3))
```

#apply( ) function - series function - one row and one column

```
print("--------O/P 15 --------")
print(sal.apply(np.mean))
```

#applymap( ) function - element function - each individual element

```
print("--------O/P 16 --------")
print(sal.applymap(np.mean))
```

output:

```
2016   2017   2018    2019
Qtr1  34500  44900  54500  61000.0
Qtr2  56000  46100  51000      NaN
Qtr3  47000  57000  57000      NaN
Qtr4  49000  59000  58500      NaN
-------- O/P 1 --------
2016    34500.0
2017    44900.0
2018    51000.0
2019    61000.0
dtype: float64
-------- O/P 2 --------
Qtr1    61000.0
Qtr2    56000.0
Qtr3    57000.0
Qtr4    59000.0
dtype: float64
-------- O/P 3 --------
          0       1       2       3
Qtr1  34500.0  44900.0  54500.0  61000.0
Qtr2  46100.0  51000.0  56000.0      NaN
Qtr3  57000.0      NaN      NaN      NaN
Qtr4  49000.0  58500.0  59000.0      NaN
-------- O/P 4 --------
2016    46625.0
2017    51750.0
2018    55250.0
2019    61000.0
dtype: float64
-------- O/P 5 --------
2016    48000.0
2017    51550.0
2018    55750.0
2019    61000.0
dtype: float64
-------- O/P 6 --------
Qtr1    4
Qtr2    3
Qtr3    3
Qtr4    3
```

dtype: int64
-------- O/P 7 --------
2016 186500.0
2017 207000.0
2018 221000.0
2019 61000.0
dtype: float64
-------- O/P 8 --------
2016 8.022917e+07
2017 5.299000e+07
2018 1.075000e+07
2019 NaN
dtype: float64
-------- O/P 9 --------

	2016	2017	2018	2019
Q1	34500	44900	54500	61000.0
Q2	56000	46100	51000	NaN
Qtr3	47000	57000	57000	NaN
Qtr4	49000	59000	58500	NaN

-------- O/P 10 --------

	16	2017	2018	2019
Qtr1	34500	44900	54500	61000.0
Qtr2	56000	46100	51000	NaN
Qtr3	47000	57000	57000	NaN
Qtr4	49000	59000	58500	NaN

	2016	2017	2018	2019
Qtr1	34500	44900	54500	61000.0
Qtr2	56000	46100	51000	NaN
Qtr3	47000	57000	57000	NaN
Qtr4	49000	59000	58500	NaN

-------- O/P 11 --------

	2016	2017	2018	2019
Qtr3	47000	57000	57000	NaN
Qtr2	56000	46100	51000	NaN
Qtr1	34500	44900	54500	61000.0
Qtr4	49000	59000	58500	NaN
Qtr5	1000	1000	1000	1000.0

-------- O/P 12 --------

	2019	2018	2017	2016	2015
Qtr1	61000.0	54500	44900	34500	1000
Qtr2	NaN	51000	46100	56000	1000
Qtr3	NaN	57000	57000	47000	1000
Qtr4	NaN	58500	59000	49000	1000

	2016	2017	2018	2019
Qtr1	34500	44900	54500	61000.0
Qtr2	56000	46100	51000	NaN
Qtr3	47000	57000	57000	NaN

```
Qtr4  49000  59000  58500     NaN
-------- O/P 13 --------
        2016       2017       2018        2019
Qtr1  1191285000  2017357000  2971885000  3.722830e+09
Qtr2  3137680000  2126593000  2602530000         NaN
Qtr3  2210410000  3250710000  3250710000         NaN
Qtr4  2402470000  3482770000  3424005000         NaN
-------- O/P 14 --------
       2016     2017     2018     2019
Qtr1  103590.0  134790.0  163590.0  183090.0
Qtr2  168090.0  138390.0  153090.0       NaN
Qtr3  141090.0  171090.0  171090.0       NaN
Qtr4  147090.0  177090.0  175590.0       NaN
-------- O/P 15 --------
2016    46625.0
2017    51750.0
2018    55250.0
2019    61000.0
dtype: float64
-------- O/P 16 --------
      2016    2017    2018    2019
Qtr1  34500.0  44900.0  54500.0  61000.0
Qtr2  56000.0  46100.0  51000.0     NaN
Qtr3  47000.0  57000.0  57000.0     NaN
Qtr4  49000.0  59000.0  58500.0     NaN
```

3. DataFrame advanced Operations

```
#Advance operations in DataFrame
import pandas as pd
import matplotlib.pyplot as pl
d={'Year': [2009, 2009, 2009 , 2009 , 2009 ],
   'Month' : ['January' , 'February' , 'March' , 'April' , 'May'],
   'Passengers' : [112 , 118 , 132 , 129 , 121]}
n=pd.DataFrame(d)
print(n)
#use of pivot()
print(n.pivot(index='Year' , columns='Month', values='Passengers'))
d1={'Year': [2009, 2009, 2009 , 2009 , 2009 ],
   'Month' : ['January' , 'February' , 'March' , 'April' , 'May'],
   'Passengers' : [112 , 118 , 132 , 129 , 121]}
n1=pd.DataFrame(d1)
print(n1)

#use of pivot_table( )
#total passengers per year
print("-------- O/P 1 --------")
print(n1.pivot_table(index='Year',values='Passengers', aggfunc='sum'))
```

```
#average passengers per month
print("-------- O/P 2 --------")
print(n1.pivot_table(index='Month',values='Passengers',aggfunc='mean'))
#aggfunc -- count, sum , mean , median , min , max , std , var , mad


#sorting , inplace attribute
print("-------- O/P 3 --------")
print(n1.sort_values('Passengers')) # ascending order
print("-------- O/P 4 --------")
print(n1.sort_values('Passengers', ascending=False)) # descending order
print("-------- O/P 5 --------")
print(n1.sort_values(['Month', 'Passengers'])) # ascending order

#Histogram
a=n1.hist(column="Passengers")
pl.show( )

#groupby( ) function
g=n1.groupby('Month')
print("-------- O/P 5 --------")
print(g.groups)
print("-------- O/P 6 --------")
print(g.get_group('January')) #list group created for the passed value
print("-------- O/P 7 --------")
print(g.size()) #list the size of the groups created
print("-------- O/P 8 --------")
print(g.count()) #list the count of non-NA values for each column in the groups created
print("-------- O/P 9 --------")
print(g['Year'].head()) #lists the specified column from the grouped object created.
```

output :

```
  Year    Month  Passengers
0  2009   January       112
1  2009  February       118
2  2009     March       132
3  2009     April       129
4  2009       May       121
Month  April  February  January  March  May
Year
2009     129       118      112    132  121
  Year    Month  Passengers
0  2009   January       112
1  2009  February       118
2  2009     March       132
3  2009     April       129
4  2009       May       121
-------- O/P 1 --------
     Passengers
Year
```

2009 612
-------- **O/P 2** --------
Passengers
Month
April 129
February 118
January 112
March 132
May 121
-------- **O/P 3** --------
Year Month Passengers
0 2009 January 112
1 2009 February 118
4 2009 May 121
3 2009 April 129
2 2009 March 132
-------- **O/P 4** --------
Year Month Passengers
2 2009 March 132
3 2009 April 129
4 2009 May 121
1 2009 February 118
0 2009 January 112
-------- **O/P 5** --------
Year Month Passengers
3 2009 April 129
1 2009 February 118
0 2009 January 112
2 2009 March 132
4 2009 May 121
-------- **O/P 5** --------
{'April': Int64Index([3], dtype='int64'), 'February': Int64Index([1], dtype='int64'), 'January': Int64Index([0], dtype='int64'), 'March': Int64Index([2], dtype='int64'), 'May': Int64Index([4], dtype='int64')}
Year Month Passengers
0 2009 January 112
Month
April 1
February 1
January 1
March 1
May 1
dtype: int64
Year Passengers
Month
April 1 1
February 1 1
January 1 1
March 1 1
May 1 1
0 2009

1 2009
2 2009
3 2009
4 2009
Name: Year, dtype: int64