

In [41]: import pandas as pd
df=pd.read_csv('Addata.csv')

Initial lookup

In [42]: df.head()

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000

In [43]: df.tail()

	ID	Experience_Years	Age	Gender	Salary
30	31	10	34	Male	80000
31	32	15	54	Male	900000
32	33	20	55	Female	1540000
33	34	19	53	Female	9300000
34	35	16	49	Male	7600000

In [44]: df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 35 entries, 0 to 34  
Data columns (total 5 columns):  
#   Column      Non-Null Count  Dtype    
...  ...          ...            
0    ID           35 non-null     int64    
1    Experience_Years  35 non-null     int64    
2    Age          35 non-null     int64    
3    Gender       35 non-null     object   
4    Salary       35 non-null     int64    
dtypes: int64(4), object(1)  
memory usage: 1.4+ KB
```

In [45]: df.describe()

	ID	Experience_Years	Age	Salary
count	35.000000	35.000000	35.000000	3.500000e+01
mean	18.000000	9.200000	35.485714	2.059147e+06
std	10.246991	7.55295	14.643552	3.170124e+06
min	1.000000	1.000000	17.000000	3.000000e+03
25%	9.500000	2.500000	22.500000	2.250000e+03
50%	18.000000	6.000000	29.000000	2.500000e+05
75%	26.500000	15.000000	53.500000	3.270000e+06
max	35.000000	27.000000	62.000000	1.000000e+07

Finding mean of all columns

In [46]: df[['Experience_Years','Age','Salary']].mean()

Experience_Years	9.200000e+00
Age	3.548571e+01
Salary	2.059147e+06
dtype:	float64

Finding median of all columns

In [47]: df[['Experience_Years','Age','Salary']].median()

Experience_Years	6.0
Age	29.0
Salary	250000.0
dtype:	float64

Finding minimum value of all columns

In [48]: df[['Experience_Years','Age','Salary']].min()

Experience_Years	1
Age	17
Salary	3000
dtype:	int64

Finding maximum value of all columns

In [49]: df[['Experience_Years','Age','Salary']].max()

Experience_Years	27
Age	62
Salary	10800080
dtype:	int64

Finding standard deviation value of all columns

In [50]: df[['Experience_Years','Age','Salary']].std()

Experience_Years	7.552950e+00
Age	1.464355e+01
Salary	3.170124e+06
dtype:	float64

Statistics according to categorical value

In [51]: df.groupby(['Gender']).count()

	ID	Experience_Years	Age	Salary
Gender				
Female	18	18	18	18
Male	17	17	17	17

In [52]: df.groupby(['Gender']).max()

	ID	Experience_Years	Age	Salary
Gender				
Female	34	27	62	10000000
Male	35	25	62	7600000

In [53]: df.groupby(['Gender']).min()

	ID	Experience_Years	Age	Salary
Gender				
Female	1	1	21	6000
Male	2	1	17	3000

In [54]: df.groupby(['Gender']).mean()

	ID	Experience_Years	Age	Salary
Gender				
Female	18.277778	10.111111	37.111111	2.054917e+06
Male	17.705882	8.235294	33.764706	2.063626e+06

In [55]: df.groupby(['Gender']).median()

	ID	Experience_Years	Age	Salary
Gender				
Female	17.5	7.5	31.0	250000.0
Male	19.0	6.0	29.0	220100.0

In [56]: df2=df.groupby('Gender')

Another way for seeing statistics for categorical value

In [57]: for i in df2:
print(i)

('Female',	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
2	3	3	23	Female	170000
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500
9	10	15	54	Female	650000
10	11	4	20	Female	250000
15	16	4	27	Female	87000
16	17	10	34	Female	930000
17	18	15	54	Female	7980000
23	24	1	21	Female	6000
24	25	4	23	Female	8900
25	26	3	22	Female	28000
27	28	27	62	Female	10000000
28	29	19	54	Female	5080000
29	30	2	21	Female	6100
32	33	20	55	Female	1540000
33	34	19	53	Female	9300000
('Male',	ID	Experience_Years	Age	Gender	Salary
1	2	1	21	Male	50000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5003000
11	12	6	29	Male	1400000
12	13	14	39	Male	6900950
13	14	11	40	Male	250100
14	15	2	23	Male	7500
19	19	2	21	Male	35000
19	20	10	36	Male	330000
20	21	15	54	Male	6570000
21	22	4	20	Male	25000
22	23	5	29	Male	6845000
26	27	1	18	Male	3000
30	31	10	34	Male	80000
31	32	15	54	Male	960000
34	35	16	49	Male	7600000

In [58]: df2.get_group('Male').head()

	ID	Experience_Years	Age	Gender	Salary
1	2	1	21	Male	50000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000
11	12	6	29	Male	1400000

In [59]: df2.get_group('Male').describe()

	ID	Experience_Years	Age	Salary
count	17.000000	17.000000	17.000000	1.700000e+01
mean	17.705882	8.235294	33.764706	2.063626e+06
std	10.122778	7.019642	13.967899	2.950974e+06
min	2.000000	1.000000	17.000000	3.000000e+03
25%	12.000000	2.000000	22.000000	2.500000e+04
50%	19.000000	6.000000	29.000000	2.201000e+05
75%	23.000000	14.000000	40.000000	6.001000e+06
max	35.000000	25.000000	62.000000	7.600000e+06

In [60]: df2.get_group('Female').head()

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
2	3	3	23	Female	170000
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500

In [61]: df2.get_group('Female').describe()

	ID	Experience_Years	Age	Salary
count	18.000000	18.000000	18.000000	1.800000e+01
mean	18.277778	10.111111	37.111111	2.054917e+06
std	18.649422	8.123234	15.466886	3.450120e+06
min	1.000000	1.000000	21.000000	6.000000e+03
25%	9.250000	3.250000	23.000000	3.037500e+04
50%	17.500000	7.500000	31.000000	2.500000e+05
75%	27.500000	19.000000	54.000000	1.387500e+06
max	34.000000	27.000000	62.000000	1.000000e+07

Python program for displaying statstical details like percentile, mean,standard deviation for the data set

In [62]: iris=pd.read_csv('Iris.csv')
iris.head()

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [63]: iris.describe()

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [64]: iris.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 6 columns):  
#   Column      Non-Null Count  Dtype    
...  ...          ...            
0    Id           150 non-null     int64    
1    SepalLengthCm  150 non-null     float64   
2    SepalWidthCm   150 non-null     float64   
3    PetalLengthCm  150 non-null     float64   
4    PetalWidthCm   150 non-null     float64   
5    Species        150 non-null     object   
dtypes: float64(4), int64(1), object(1)  
memory usage: 7.2+ KB
```

In [65]: iris2=iris.groupby('Species')

In [66]: iris.setosa=iris2.get_group('Iris-setosa')
iris.setosa.head()

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [67]: iris.verisicolor=iris2.get_group('Iris-versicolor')
iris.verisicolor.head()

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
50	51	7.0	3.2	4.7	1.4	Iris-versicolor
51	52	6.4	3.2	4.5	1.5	Iris-versicolor
52	53	6.9	3.1	4.9	1.5	Iris-versicolor
53	54	5.5	2.3	4.0	1.3	Iris-versicolor
54	55	6.5	2.8	4.6	1.5	Iris-versicolor

In [68]: iris.virginica=iris2.get_group('Iris-virginica')
iris.virginica.head()

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
100	101	6.3	3.3	6.0	2.5	Iris-virginica
101	102	5.8	2.7	5.1	1.9	Iris-virginica
102	103	7.1	3.0	5.9	2.1	Iris-virginica
103	104	6.3	2.9	5.6	1.8	Iris-virginica
104	105	6.5	3.0	5.8	2.2	Iris-virginica

Program

```
def Cal_Mean(df):  
    x = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].mean()  
    print(x)  
  
def Cal_Perc(df):  
    x = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].quantile()  
    print(x)  
  
def Cal_Median(df):  
    x = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].median()  
    print(x)  
  
def Cal_Max(df):  
    x = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].max()  
    print(x)  
  
def Cal_Min(df):  
    x = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].min()  
    print(x)  
  
def Menu(df):  
    print('Menu')  
    Cal_Mean(df)  
    print(" ")  
    print('Percentile')  
    Cal_Perc(df)  
    print(" ")  
    print('Median')  
    Cal_Median(df)  
    print(" ")  
    print('Max')  
    Cal_Max(df)  
    print(" ")  
    print('Min')  
    Cal_Min(df)  
    print(" ")  
  
def MainMenu():  
    while True:  
        print('1:For Iris-setosa')  
        print('2:For Iris-versicolor')  
        print('3:For Iris-virginica')  
        b = int(input('Enter your choice: '))  
        if b == 1:  
            print('Iris-setosa')  
            Menu(iris.setosa)  
            elif b == 2:  
                print('Iris-versicolor')  
                Menu(iris.verisicolor)  
            elif b == 3:  
                print('Iris-virginica')  
                Menu(iris.virginica)  
            else:  
                print('Choose correct option')  
        b = input('Do you want to continue?(Y/N)')  
        if b == 'N':  
            break  
    MainMenu()  
  
1:For Iris-setosa  
2:For Iris-versicolor  
3:For Iris-virginica  
Enter your choice: 1  
Iris-setosa  
Mean  
SepalLengthCm 5.886  
SepalWidthCm 3.418  
PetalLengthCm 1.464  
PetalWidthCm 0.244  
dtype: float64  
  
Percentile  
SepalLengthCm 5.9  
SepalWidthCm 3.4  
PetalLengthCm 1.5  
PetalWidthCm 0.2  
Name: 0.5, dtype: float64  
  
Median  
SepalLengthCm 5.9  
SepalWidthCm 3.4  
PetalLengthCm 1.5  
PetalWidthCm 0.2  
dtype: float64  
  
Max  
SepalLengthCm 5.8  
SepalWidthCm 4.4  
PetalLengthCm 1.9  
PetalWidthCm 0.6  
dtype: float64  
  
Min  
SepalLengthCm 4.3  
SepalWidthCm 2.3  
PetalLengthCm 1.0  
PetalWidthCm 0.1  
dtype: float64  
  
Do you want to continue?(Y/N)Y  
1:For Iris-setosa  
2:For Iris-versicolor  
3:For Iris-virginica  
Enter your choice: 2  
Iris-versicolor  
Mean  
SepalLengthCm 5.936  
SepalWidthCm 2.779  
PetalLengthCm 4.269  
PetalWidthCm 1.226  
dtype: float64  
  
Percentile  
SepalLengthCm 5.98  
SepalWidthCm 2.88  
PetalLengthCm 4.35  
PetalWidthCm 1.38  
Name: 0.5, dtype: float64  
  
Median  
SepalLengthCm 5.98  
SepalWidthCm 2.88  
PetalLengthCm 4.35  
PetalWidthCm 1.38  
dtype: float64  
  
Max  
SepalLengthCm 7.9  
SepalWidthCm 3.4  
PetalLengthCm 5.1  
PetalWidthCm 1.8  
dtype: float64  
  
Min  
SepalLengthCm 4.9  
SepalWidthCm 2.9  
PetalLengthCm 3.0  
PetalWidthCm 1.0  
dtype: float64  
  
Do you want to continue?(Y/N)Y  
1:For Iris-setosa  
2:For Iris-versicolor  
3:For Iris-virginica  
Enter your choice: 3  
Iris-virginica  
Mean  
SepalLengthCm 6.588  
SepalWidthCm 2.974  
PetalLengthCm 5.552  
PetalWidthCm 2.026  
dtype: float64  
  
Percentile  
SepalLengthCm 6.59  
SepalWidthCm 3.08  
PetalLengthCm 5.55  
PetalWidthCm 2.08  
Name: 0.5, dtype: float64  
  
Median  
SepalLengthCm 6.58  
SepalWidthCm 3.08  
PetalLengthCm 5.55  
PetalWidthCm 2.08  
dtype: float64  
  
Max  
SepalLengthCm 7.9  
SepalWidthCm 3.4  
PetalLengthCm 6.9  
PetalWidthCm 2.5  
dtype: float64  
  
Min  
SepalLengthCm 4.9  
SepalWidthCm 2.2  
PetalLengthCm 4.5
```

