

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

```
In [2]: import numpy as np
```

```
In [3]: import matplotlib as plt
```

```
In [6]: import seaborn as sns
```

```
In [7]: df=pd.read_csv('Academic-Performance-Dataset.csv')
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Rollno          20 non-null    int64
 1   Name            18 non-null    object
 2   Gender          20 non-null    object
 3   Branch          20 non-null    object
 4   Attendance      20 non-null    int64
 5   Phy_marks       19 non-null    float64
 6   Che_marks       17 non-null    float64
 7   EM1_marks       18 non-null    float64
 8   PPS_marks       19 non-null    float64
 9   SME_marks       20 non-null    int64
10   Total Marks     20 non-null    int64
11   Percentage      20 non-null    float64
dtypes: float64(5), int64(4), object(3)
memory usage: 2.0+ KB
```

```
In [9]: df.describe()
```

```
Out[9]:
```

	Rollno	Attendance	Phy_marks	Che_marks	EM1_marks	PPS_marks	SME_marks
count	20.00000	20.000000	19.000000	17.000000	18.000000	19.000000	20.000000
mean	10.50000	75.100000	63.421053	80.764706	83.444444	60.736842	62.700000
std	5.91608	14.660724	34.940133	13.690916	11.078449	43.598983	24.891554
min	1.00000	53.000000	-34.000000	51.000000	63.000000	-99.000000	23.000000
25%	5.75000	63.750000	59.000000	70.000000	75.750000	56.000000	42.750000
50%	10.50000	73.000000	67.000000	83.000000	83.000000	66.000000	60.500000
75%	15.25000	87.000000	88.000000	93.000000	93.000000	87.500000	80.250000
max	20.00000	98.000000	99.000000	100.000000	100.000000	99.000000	99.000000

In [10]: df

Out[10]:

	Rollno	Name	Gender	Branch	Attendance	Phy_marks	Che_marks	EM1_marks	F
0	1	Mohammed	M	Comp	72	62.0	98.0	63.0	
1	2	Reyansh	M	IT	58	62.0	83.0	83.0	
2	3	Aarav	M	IT	57	-20.0	100.0	NaN	
3	4	Atharv	M	IT	60	89.0	83.0	70.0	
4	5	Vivaan	M	Comp	85	90.0	NaN	78.0	
5	6	Advik	M	ENTC	94	99.0	84.0	100.0	
6	7	Ansh	M	ENTC	98	88.0	95.0	81.0	
7	8	Ishaan	M	ENTC	75	66.0	51.0	83.0	
8	9	Dhruv	M	ENTC	63	NaN	NaN	97.0	
9	10	Siddharth	M	ENTC	96	67.0	78.0	95.0	
10	11	Vihaan	M	ENTC	82	54.0	70.0	88.0	
11	12	NaN	M	IT	75	64.0	67.0	71.0	
12	13	Aarush	M	IT	67	56.0	81.0	NaN	
13	14	Leo	M	IT	98	-34.0	70.0	94.0	
14	15	Maryam	F	IT	64	87.0	60.0	90.0	
15	16	Saanvi	F	Comp	66	90.0	95.0	67.0	
16	17	Zaranew	F	Comp	93	54.0	NaN	75.0	
17	18	Inaya	F	Comp	74	67.0	93.0	93.0	
18	19	Aarya	F	Comp	72	88.0	84.0	81.0	
19	20	NaN	F	Comp	53	76.0	81.0	93.0	

```
In [ ]: male_female=student.groupby('Gender')['Gender'].count()
print(male_female)
```

```
In [12]: M_F=df.groupby('Gender')['Gender'].count()
print(M_F)
```

```
Gender
F      6
M     14
Name: Gender, dtype: int64
```

```
In [14]: df.Branch.unique()
```

```
Out[14]: array(['Comp', 'IT', 'ENTC'], dtype=object)
```

```
In [15]: df.Attendance.unique()
```

```
Out[15]: array([72, 58, 57, 60, 85, 94, 98, 75, 63, 96, 82, 67, 64, 66, 93,
       74, 53])
```

```
In [22]: mean_Phy_marks = df.groupby('Gender').Phy_marks.mean()
print(mean_Phy_marks)
```

```
Gender
F    77.000000
M    57.153846
Name: Phy_marks, dtype: float64
```

```
In [20]: mean_SME_marks=df.groupby('Gender').SME_marks.mean()
print(mean_SME_marks)
```

```
Gender
F    66.500000
M    61.071429
Name: SME_marks, dtype: float64
```

```
In [ ]:
```

```
In [29]: mean_Phy_marks_Branch=df.groupby(['Gender', 'Branch']).SME_marks.mean()
print(mean_Phy_marks_Branch)
```

```
Gender  Branch
F        Comp    61.800000
         IT      90.000000
M        Comp    46.000000
         ENTC    77.000000
         IT      50.166667
Name: SME_marks, dtype: float64
```

```
In [30]: df.Phy_marks.describe()
```

```
Out[30]: count    19.000000
mean      63.421053
std       34.940133
min      -34.000000
25%       59.000000
50%       67.000000
75%       88.000000
max       99.000000
Name: Phy_marks, dtype: float64
```

```
In [31]: df.Phy_marks.unique()
```

```
Out[31]: array([ 62., -20.,  89.,  90.,  99.,  88.,  66.,  nan,  67.,  54.,
        64.,
        56., -34.,  87.,  76.]
```

```
In [35]:
```

```
print(df.groupby('Gender').Phy_marks.describe())
```

	count	mean	std	min	25%	50%	75%	max
Gender								
F	6.0	77.000000	14.282857	54.0	69.25	81.5	87.75	90.0
M	13.0	57.153846	40.143173	-34.0	56.00	64.0	88.00	99.0

```
In [34]:
```

```
print(df.groupby('Gender').SME_marks.describe())
```

	count	mean	std	min	25%	50%	75%	max
Gender								
F	6.0	66.500000	28.549956	23.0	50.00	71.0	86.75	99.0
M	14.0	61.071429	24.135720	23.0	40.75	56.0	77.50	99.0

```
In [37]: group=pd.cut(df['SME_marks'],bins=4)
print(group)
```

```
0    (22.924, 42.0]
1    (22.924, 42.0]
2    (22.924, 42.0]
3    (22.924, 42.0]
4     (42.0, 61.0]
5     (80.0, 99.0]
6     (61.0, 80.0]
7     (61.0, 80.0]
8     (42.0, 61.0]
9     (80.0, 99.0]
10    (42.0, 61.0]
11    (80.0, 99.0]
12    (42.0, 61.0]
13    (61.0, 80.0]
14    (80.0, 99.0]
15    (61.0, 80.0]
16    (61.0, 80.0]
17    (80.0, 99.0]
18    (42.0, 61.0]
19    (22.924, 42.0]
Name: SME_marks, dtype: category
Categories (4, interval[float64, right]): [(22.924, 42.0] < (42.0,
61.0] < (61.0, 80.0] < (80.0, 99.0]]
```

```
In [38]: group=pd.cut(df['Phy_marks'],bins=4)
print(group)
```

```
0      (32.5, 65.75]
1      (32.5, 65.75]
2      (-34.133, -0.75]
3      (65.75, 99.0]
4      (65.75, 99.0]
5      (65.75, 99.0]
6      (65.75, 99.0]
7      (65.75, 99.0]
8      NaN
9      (65.75, 99.0]
10     (32.5, 65.75]
11     (32.5, 65.75]
12     (32.5, 65.75]
13     (-34.133, -0.75]
14     (65.75, 99.0]
15     (65.75, 99.0]
16     (32.5, 65.75]
17     (65.75, 99.0]
18     (65.75, 99.0]
19     (65.75, 99.0]
Name: Phy_marks, dtype: category
Categories (4, interval[float64, right]): [(-34.133, -0.75] < (-0.7
5, 32.5] < (32.5, 65.75] < (65.75, 99.0]]
```

```
In [39]: df.groupby(group)['Phy_marks'].count()
```

```
Out[39]: Phy_marks
(-34.133, -0.75]      2
(-0.75, 32.5]         0
(32.5, 65.75]         6
(65.75, 99.0]        11
Name: Phy_marks, dtype: int64
```

```
In [40]: pd.crosstab(group,df['Gender'])
```

```
Out[40]:
```

	Gender	
	F	M
Phy_marks		
(-34.133, -0.75]	0	2
(32.5, 65.75]	1	5
(65.75, 99.0]	5	6

```
In [ ]:
```

```
In [41]: df = pd.read_csv("Iris.csv")
df.head()
```

```
Out[41]:
```

	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 [42]: df = pd.read_csv("Iris.csv")
df.tail()
```

```
Out[42]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
In [43]: print('Iris-setosa')
setosa = df['Species'] == 'Iris-setosa'
print(df[setosa].describe())
```

```
Iris-setosa
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalW
idthCm					
count	50.00000	50.00000	50.000000	50.000000	5
0.00000					
mean	25.50000	5.00600	3.418000	1.464000	
0.24400					
std	14.57738	0.35249	0.381024	0.173511	
0.10721					
min	1.00000	4.30000	2.300000	1.000000	
0.10000					
25%	13.25000	4.80000	3.125000	1.400000	
0.20000					
50%	25.50000	5.00000	3.400000	1.500000	
0.20000					
75%	37.75000	5.20000	3.675000	1.575000	
0.30000					
max	50.00000	5.80000	4.400000	1.900000	
0.60000					

```
In [44]: print('\nIris-versicolor')
setosa = df['Species'] == 'Iris-versicolor'
print(df[setosa].describe())
```

Iris-versicolor					
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	Petal
WidthCm					
count	50.00000	50.000000	50.000000	50.000000	5
0.000000					
mean	75.50000	5.936000	2.770000	4.260000	
1.326000					
std	14.57738	0.516171	0.313798	0.469911	
0.197753					
min	51.00000	4.900000	2.000000	3.000000	
1.000000					
25%	63.25000	5.600000	2.525000	4.000000	
1.200000					
50%	75.50000	5.900000	2.800000	4.350000	
1.300000					
75%	87.75000	6.300000	3.000000	4.600000	
1.500000					
max	100.00000	7.000000	3.400000	5.100000	
1.800000					

```
In [45]: print('\nIris-virginica')
setosa = df['Species'] == 'Iris-virginica'
print(df[setosa].describe())
```

Iris-virginica					
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	Petal
WidthCm					
count	50.00000	50.000000	50.000000	50.000000	5
0.000000					
mean	125.50000	6.58800	2.974000	5.552000	
2.02600					
std	14.57738	0.63588	0.322497	0.551895	
0.27465					
min	101.00000	4.90000	2.200000	4.500000	
1.40000					
25%	113.25000	6.22500	2.800000	5.100000	
1.80000					
50%	125.50000	6.50000	3.000000	5.550000	
2.00000					
75%	137.75000	6.90000	3.175000	5.875000	
2.30000					
max	150.00000	7.90000	3.800000	6.900000	
2.50000					

```
In [ ]:
```

