Pandas

```
import pandas as pd

mydata=["Megha","Nandhitha","Hemabhinaya","Purushottam","Shivakumar","
Rajendra","Praveen"]
Ser=pd.Series(mydata)
print(Ser)
```

```
0          Megha
1      Nandhitha
2    Hemabhinaya
3    Purushottam
4     Shivakumar
5       Rajendra
6        Praveen
dtype: object
```

```
mydata=["Megha","Nandhitha","Hemabhinaya","Purushottam","Shivakumar","
Rajendra","Praveen"]
roll=[2,4,7,3,1,8,7]
Ser=pd.Series(mydata)
print(Ser)
```

```
0          Megha
1      Nandhitha
2    Hemabhinaya
3    Purushottam
4     Shivakumar
5       Rajendra
6        Praveen
dtype: object
```

```
Ser[4]
```

```
'Shivakumar'
```

```
Ser[2]
```

```
'Hemabhinaya'
```

```
mydata=["Megha","Nandhitha","Hemabhinaya","Purushottam","Shivakumar","
Rajendra","Praveen"]
roll=[2,4,7,3,1,8,7]
Ser1=pd.Series(mydata,index=roll)
print(Ser1)
```

```
2          Megha
4      Nandhitha
7    Hemabhinaya
```

```
3    Purushottam
1     Shivakumar
8       Rajendra
7        Praveen
dtype: object
```

```
Ser1.to_csv(r"C:\Mypythonfiles\Mydata.csv")
```

Dataframes

```
MyDict={"Names":
["Megha","Nandhitha","Hemabhinaya","Purushottam","Shivakumar","Rajendr
a","Praveen"],
        "Age":[20,19,20,19,20,19,20],
        "City":
["Haveri","Gangavathi","Koppal","Tumkur","Tumkur","Chitradurga","Tumku
r"]
       }
print(MyDict)
```

```
{'Names': ['Megha', 'Nandhitha', 'Hemabhinaya', 'Purushottam',
'Shivakumar', 'Rajendra', 'Praveen'], 'Age': [20, 19, 20, 19, 20, 19,
20], 'City': ['Haveri', 'Gangavathi', 'Koppal', 'Tumkur', 'Tumkur',
'Chitradurga', 'Tumkur']}
```

```
Dict_data=pd.DataFrame(MyDict)
print(Dict_data)
```

```
         Names  Age         City
0        Megha   20       Haveri
1    Nandhitha   19   Gangavathi
2  Hemabhinaya   20       Koppal
3  Purushottam   19       Tumkur
4   Shivakumar   20       Tumkur
5     Rajendra   19  Chitradurga
6      Praveen   20       Tumkur
```

```
Dict_data.to_csv(r"C:\Mypythonfiles\Dict-data.csv")
```

Load Data

```
import pandas as pd
Ddf=pd.read_csv(r"C:\Mypythonfiles\diabetcsvsmall.csv")
Ddf.head()
```

```
   preg  plas  pres  skin  insu  mass   pedi  age            class
0   6.0   148  72.0  35.0     0  33.6  0.627   50  tested_positive
```

```
1   1.0    85  66.0  29.0     0  26.6  0.351   31  tested_negative
2   8.0   183  64.0   0.0     0  23.3  0.672   32  tested_positive
3   1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative
4   0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive
```

```
Ddf.tail()
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
97    1.0    71  48.0   NaN    76  20.4  0.323   22  tested_negative
98    6.0    93  50.0  30.0    64  28.7  0.356   23  tested_negative
99    NaN   122  90.0  51.0   220  49.7  0.325   31  tested_positive
100   1.0   163  72.0   0.0     0  39.0  1.222   33  tested_positive
101   1.0   151  60.0   0.0     0  26.1  0.179   22  tested_negative
```

Accessing

```
Ddf.loc[12:19,"age"]
```

```
12    57
13    59
14    51
15    32
16    31
17    31
18    33
19    32
Name: age, dtype: int64
```

```
Ddf.loc[12:19]
```

```
    preg  plas  pres  skin  insu  mass   pedi  age            class
12  10.0   139  80.0   0.0     0  27.1  1.441   57  tested_negative
13   1.0   189  60.0  23.0   846  30.1  0.398   59  tested_positive
14   5.0   166  72.0  19.0   175  25.8  0.587   51  tested_positive
15   7.0   100   0.0   0.0     0  30.0  0.484   32  tested_positive
16   0.0   118  84.0  47.0   230  45.8  0.551   31  tested_positive
17   7.0   107  74.0   0.0     0  29.6  0.254   31  tested_positive
18   1.0   103  30.0  38.0    83  43.3  0.183   33  tested_negative
19   1.0   115  70.0  30.0    96  34.6  0.529   32  tested_positive
```

```
Ddf.loc[12:19,"class"]
```

```
12    tested_negative
13    tested_positive
14    tested_positive
15    tested_positive
16    tested_positive
17    tested_positive
18    tested_negative
```

```
19    tested_positive
Name: class, dtype: object
```

```
Ddf.iloc[12:19,3:8] #dataframe.iloc[row_range,column_range]
```

```
    skin  insu  mass   pedi  age
12   0.0     0  27.1  1.441   57
13  23.0   846  30.1  0.398   59
14  19.0   175  25.8  0.587   51
15   0.0     0  30.0  0.484   32
16  47.0   230  45.8  0.551   31
17   0.0     0  29.6  0.254   31
18  38.0    83  43.3  0.183   33
```

Feature Engineering

insu,mass,preg,plas,age,pres,skin==>Independent(Feature) class==> dependent ==> Target(depends on features)

```
Ddf.rename(columns={"plas":"Glucose"},inplace=True)
#dataframe.rename(columns={"old":"new"},inplace=True)

sDdf.head()
```

```
   preg  Glucose  pres  skin  insu  mass   pedi  age            class
0   6.0      148  72.0  35.0     0  33.6  0.627   50  tested_positive
1   1.0       85  66.0  29.0     0  26.6  0.351   31  tested_negative
2   8.0      183  64.0   0.0     0  23.3  0.672   32  tested_positive
3   1.0       89  66.0  23.0    94  28.1  0.167   21  tested_negative
4   0.0      137  40.0  35.0   168  43.1  2.288   33  tested_positive
```

```
Ddf["Glucose_in_mmol"]=Ddf["Glucose"]/18.018 #dataframe[new column name]=Content
#Converting Glucose from mg to mmol and creating new column

Ddf.head(10)
```

```
   preg  Glucose  pres  skin  insu  mass   pedi  age            class
\
0   6.0      148  72.0  35.0     0  33.6  0.627   50  tested_positive

1   1.0       85  66.0  29.0     0  26.6  0.351   31  tested_negative

2   8.0      183  64.0   0.0     0  23.3  0.672   32  tested_positive

3   1.0       89  66.0  23.0    94  28.1  0.167   21  tested_negative

4   0.0      137  40.0  35.0   168  43.1  2.288   33  tested_positive
```

```
5   5.0      116  74.0   0.0     0  25.6  0.201   30  tested_negative

6   3.0       78  50.0  32.0    88  31.0  0.248   26  tested_positive

7  10.0      115   0.0   0.0     0  35.3  0.134   29  tested_negative

8   2.0      197  70.0  45.0   543  30.5  0.158   53  tested_positive

9   8.0      125  96.0   0.0     0   0.0  0.232   54  tested_positive


   Glucose_in_mmol
0         8.214008
1         4.717505
2        10.156510
3         4.939505
4         7.603508
5         6.438006
6         4.329004
7         6.382506
8        10.933511
9         6.937507
```

Filter and Groups

```
fil_age_30less=Ddf[Ddf["age"]<30] #new df=your df[condition]
fil_age_30less.head(5)
```

```
    preg  Glucose  pres  skin  insu  mass   pedi  age            class
\
3    1.0       89  66.0  23.0    94  28.1  0.167   21  tested_negative

6    3.0       78  50.0  32.0    88  31.0  0.248   26  tested_positive

7   10.0      115   0.0   0.0     0  35.3  0.134   29  tested_negative

20   3.0      126  88.0  41.0   235  39.3  0.704   27  tested_negative

23   9.0      119  80.0  35.0     0  29.0  0.263   29  tested_positive


    Glucose_in_mmol
3          4.939505
6          4.329004
7          6.382506
20         6.993007
23         6.604507
```

```
Glucose_below_100=Ddf[Ddf["Glucose"]<100]
Glucose_below_100.head(5)
```

```
    preg  Glucose  pres  skin  insu  mass   pedi  age            class
\
1    1.0       85  66.0  29.0     0  26.6  0.351   31  tested_negative

3    1.0       89  66.0  23.0    94  28.1  0.167   21  tested_negative

6    3.0       78  50.0  32.0    88  31.0  0.248   26  tested_positive

21   8.0       99  84.0   0.0     0  35.4  0.388   50  tested_negative

27   1.0       97  66.0  15.0   140  23.2  0.487   22  tested_negative


    Glucose_in_mmol
1          4.717505
3          4.939505
6          4.329004
21         5.494505
27         5.383505

Glucose_above_100=Ddf[Ddf["Glucose"]>100]
Glucose_above_100.head(5)

   preg  Glucose  pres  skin  insu  mass   pedi  age            class
\
0   6.0      148  72.0  35.0     0  33.6  0.627   50  tested_positive

2   8.0      183  64.0   0.0     0  23.3  0.672   32  tested_positive

4   0.0      137  40.0  35.0   168  43.1  2.288   33  tested_positive

5   5.0      116  74.0   0.0     0  25.6  0.201   30  tested_negative

7  10.0      115   0.0   0.0     0  35.3  0.134   29  tested_negative


   Glucose_in_mmol
0         8.214008
2        10.156510
4         7.603508
5         6.438006
7         6.382506
```

Create a filtered data set which has only the rows with age between 20 and 30

```
fil_age=Ddf[(Ddf["age"]>20) & (Ddf["age"]<30)]
fil_age.head(10)

    preg  Glucose  pres  skin  insu  mass   pedi  age            class
\
3    1.0       89  66.0  23.0    94  28.1  0.167   21  tested_negative
```

```
6    3.0       78  50.0  32.0    88  31.0  0.248   26  tested_positive
7   10.0      115   0.0   0.0     0  35.3  0.134   29  tested_negative
20   3.0      126  88.0  41.0   235  39.3  0.704   27  tested_negative
23   9.0      119  80.0  35.0     0  29.0  0.263   29  tested_positive
27   1.0       97  66.0  15.0   140  23.2  0.487   22  tested_negative
31   3.0      158  76.0  36.0   245  31.6  0.851   28  tested_positive
32   3.0       88  58.0  11.0    54  24.8  0.267   22  tested_negative
33   6.0       92  92.0   0.0     0  19.9  0.188   28  tested_negative
40   3.0      180  64.0  25.0    70  34.0  0.271   26  tested_negative
```

```
    Glucose_in_mmol
3          4.939505
6          4.329004
7          6.382506
20         6.993007
23         6.604507
27         5.383505
31         8.769009
32         4.884005
33         5.106005
40         9.990010
```

```
# Group by clas and calculate average age
grouped_by_class_age=Ddf.groupby("class")["age"].mean()
grouped_by_class_age
#Results:
# The average age of diabetic tested_positive is 40.5
# The avrerage age of non-diabetic tested_negative is 31.23
```

```
class
tested_negative    31.238095
tested_positive    40.589744
Name: age, dtype: float64
```

```
group_class_ins=Ddf.groupby("class")["insu"].mean()
group_class_ins
#Results:
# The average insulin level of diabetic people is 114.69
# The average insulin level of non-diabetic people is 52.57
```

```
class
tested_negative     52.571429
tested_positive    114.692308
Name: insu, dtype: float64
```

```
grouped_by_class_age=Ddf.groupby("class")["age"].min()
grouped_by_class_age
```

```
class
tested_negative    21
tested_positive    25
Name: age, dtype: int64
```

```
grouped_by_class_age=Ddf.groupby("class")["age"].max()
grouped_by_class_age
```

```
class
tested_negative    60
tested_positive    60
Name: age, dtype: int64
```

```
grouped_by_class_age=Ddf.groupby("class")["age"].sum()
grouped_by_class_age
```

```
class
tested_negative    1968
tested_positive    1583
Name: age, dtype: int64
```

Cleaning Data

Handling Nulls

```
Ddf.isnull().sum()
```

```
preg               1
Glucose            0
pres               1
skin               1
insu               0
mass               1
pedi               1
age                0
class              0
Glucose_in_mmol    0
dtype: int64
```

```
Ddf.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             101 non-null    float64
 1   Glucose          102 non-null    int64  
 2   pres             101 non-null    float64
 3   skin             101 non-null    float64
 4   insu             102 non-null    int64  
 5   mass             101 non-null    float64
 6   pedi             101 non-null    float64
 7   age              102 non-null    int64  
 8   class            102 non-null    object 
 9   Glucose_in_mmol  102 non-null    float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.1+ KB

Ddf.isnull()

      preg  Glucose   pres   skin   insu   mass   pedi    age  
class  \
0    False    False  False  False  False  False  False  False  False

1    False    False  False  False  False  False  False  False  False

2    False    False  False  False  False  False  False  False  False

3    False    False  False  False  False  False  False  False  False

4    False    False  False  False  False  False  False  False  False

..     ...      ...    ...    ...    ...    ...    ...    ...    ...

97   False    False  False   True  False  False  False  False  False

98   False    False  False  False  False  False  False  False  False

99    True    False  False  False  False  False  False  False  False

100  False    False  False  False  False  False  False  False  False

101  False    False  False  False  False  False  False  False  False


     Glucose_in_mmol
0              False
1              False
2              False
```

```
3               False
4               False
..                ...
97              False
98              False
99              False
100             False
101             False

[102 rows x 10 columns]
```

```
Ddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             101 non-null    float64
 1   Glucose          102 non-null    int64  
 2   pres             101 non-null    float64
 3   skin             101 non-null    float64
 4   insu             102 non-null    int64  
 5   mass             101 non-null    float64
 6   pedi             101 non-null    float64
 7   age              102 non-null    int64  
 8   class            102 non-null    object 
 9   Glucose_in_mmol  102 non-null    float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.1+ KB
```

```
Ddf.dropna(inplace=True)
```

```
Ddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             98 non-null     float64
 1   Glucose          98 non-null     int64  
 2   pres             98 non-null     float64
 3   skin             98 non-null     float64
 4   insu             98 non-null     int64  
 5   mass             98 non-null     float64
 6   pedi             98 non-null     float64
 7   age              98 non-null     int64  
 8   class            98 non-null     object 
 9   Glucose_in_mmol  98 non-null     float64
```

```
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

```
Ddf.isnull().sum()
```

```
preg                0
Glucose             0
pres                0
skin                0
insu                0
mass                0
pedi                0
age                 0
class               0
Glucose_in_mmol     0
dtype: int64
```

Handling Duplicates

```
Ddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             98 non-null     float64
 1   Glucose          98 non-null     int64  
 2   pres             98 non-null     float64
 3   skin             98 non-null     float64
 4   insu             98 non-null     int64  
 5   mass             98 non-null     float64
 6   pedi             98 non-null     float64
 7   age              98 non-null     int64  
 8   class            98 non-null     object 
 9   Glucose_in_mmol  98 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

```
Ddf.drop_duplicates(inplace=True)
```

```
Ddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 96 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             96 non-null     float64
```

```
 1   Glucose          96 non-null     int64
 2   pres             96 non-null     float64
 3   skin             96 non-null     float64
 4   insu             96 non-null     int64
 5   mass             96 non-null     float64
 6   pedi             96 non-null     float64
 7   age              96 non-null     int64
 8   class            96 non-null     object
 9   Glucose_in_mmol  96 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.2+ KB
```

Reading Other Formats

```
dia_ex=pd.read_excel(r"C:\Mypythonfiles\diabetes.xlsx")

dia_ex.head()
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive

```
dia_ex=pd.read_excel(r"C:\Mypythonfiles\diabetes.xlsx",
sheet_name="dora")
dia_ex.head()
```

	Dead	Alive
0	yes	no
1	yes	no
2	yes	no
3	yes	no
4	yes	no

```
df_txt=pd.read_csv(r"C:\Mypythonfiles\grades.txt")

df_txt.head()
```

	Names	Initials	SEM1	SEM2	SEM3	Grade
0		Joe	K	9.8	10	9.9 A+
1		Rajesh	M	8.9	9.1	9.3 A
2		Kissan	V	9.9	9.3	9.2 A
3		Mary	N	7.7	8	7.1 B
4		Jeen	K	9.8	9.1	9.9 A+

```
df_txt=pd.read_csv(r"C:\Mypythonfiles\grades.txt",sep=" ")
df_txt.head()
```

```
    Names Initials  SEM1  SEM2  SEM3 Grade
0     Joe        K   9.8  10.0   9.9    A+
1  Rajesh        M   8.9   9.1   9.3     A
2  Kissan        V   9.9   9.3   9.2     A
3    Mary        N   7.7   8.0   7.1     B
4    Jeen        K   9.8   9.1   9.9    A+
```

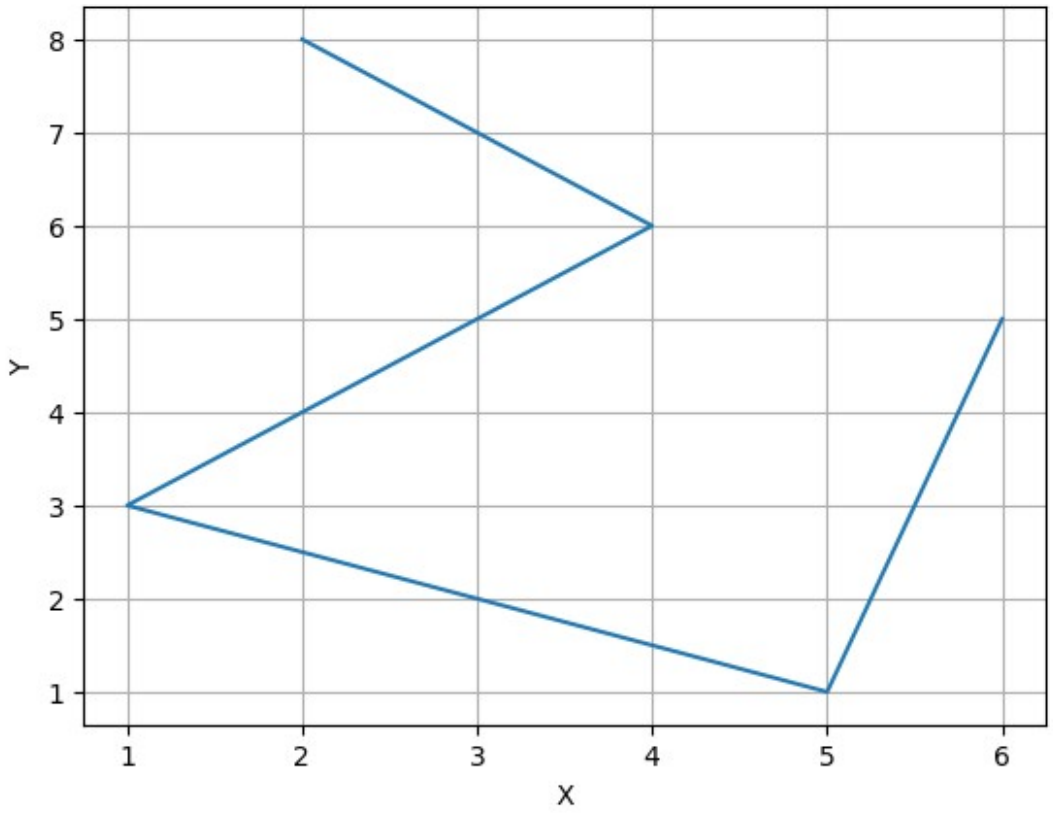
Modifying data types

```
df_txt["SEM1_int"]=df_txt["SEM1"].astype(int)

df_txt.head()
```

```
    Names Initials  SEM1  SEM2  SEM3 Grade  SEM1_int
0     Joe        K   9.8  10.0   9.9    A+         9
1  Rajesh        M   8.9   9.1   9.3     A         8
2  Kissan        V   9.9   9.3   9.2     A         9
3    Mary        N   7.7   8.0   7.1     B         7
4    Jeen        K   9.8   9.1   9.9    A+         9
```

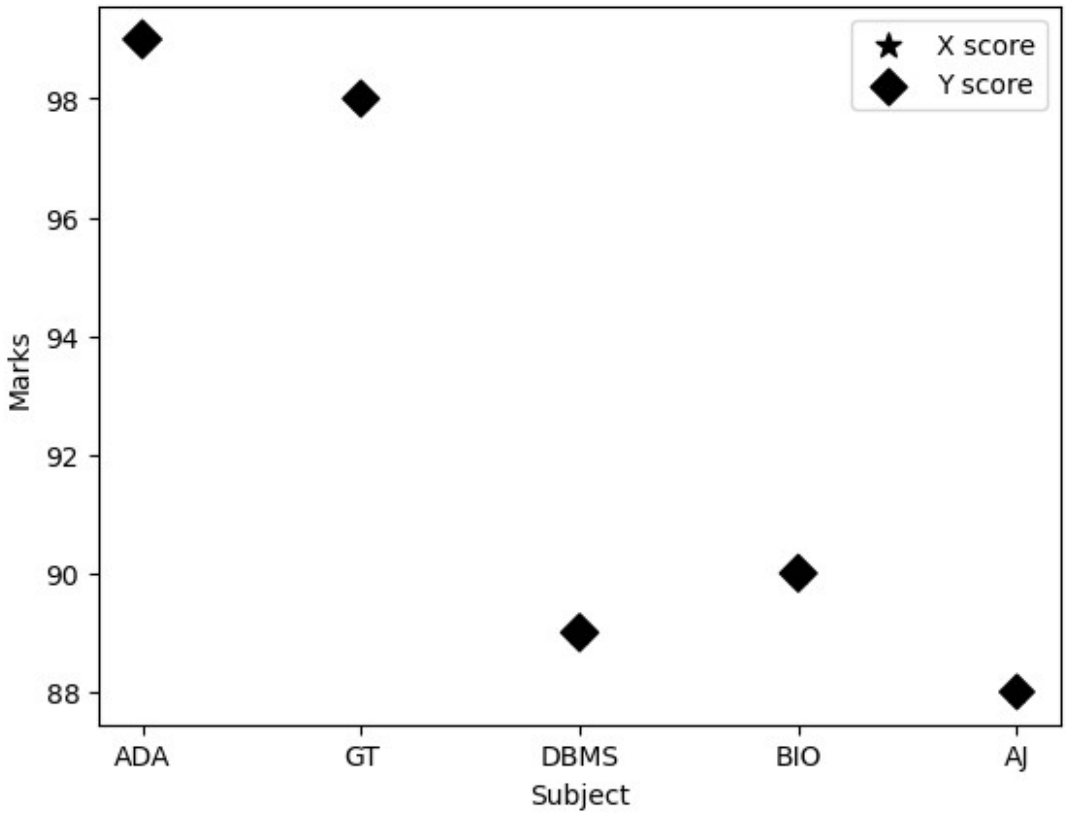
MatplotLib

```
import matplotlib.pyplot as plt
x=[2,4,1,5,6]
y=[8,6,3,1,5]
plt.plot(x,y)
plt.xlabel("X")
plt.ylabel("Y")
plt.grid()
```

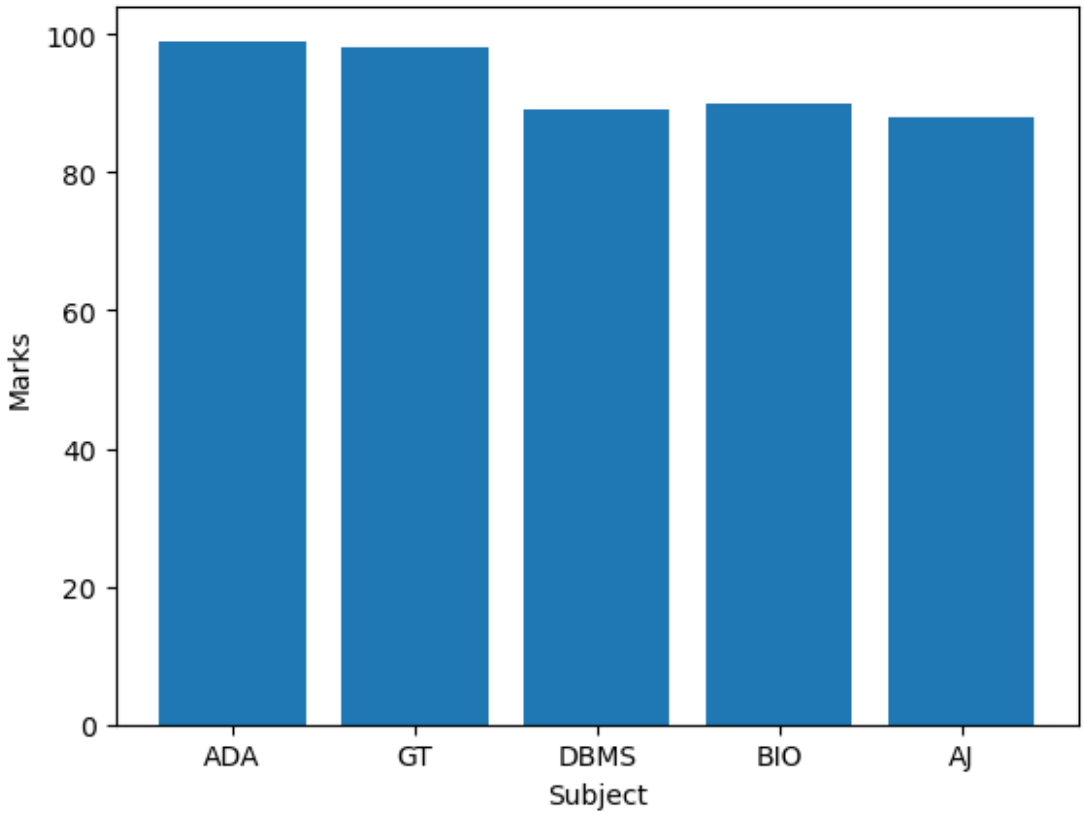



```
sub=["ADA","GT","DBMS","BIO","AJ"]
x=[99,98,89,90,88]
y=[98,89,94,92,80]
plt.scatter(sub,x,y,color='k',label="X score",marker="*")
plt.scatter(sub,x,y,color='k',label="Y score",marker="D")
plt.xlabel("Subject")
plt.ylabel("Marks")
plt.legend()
```

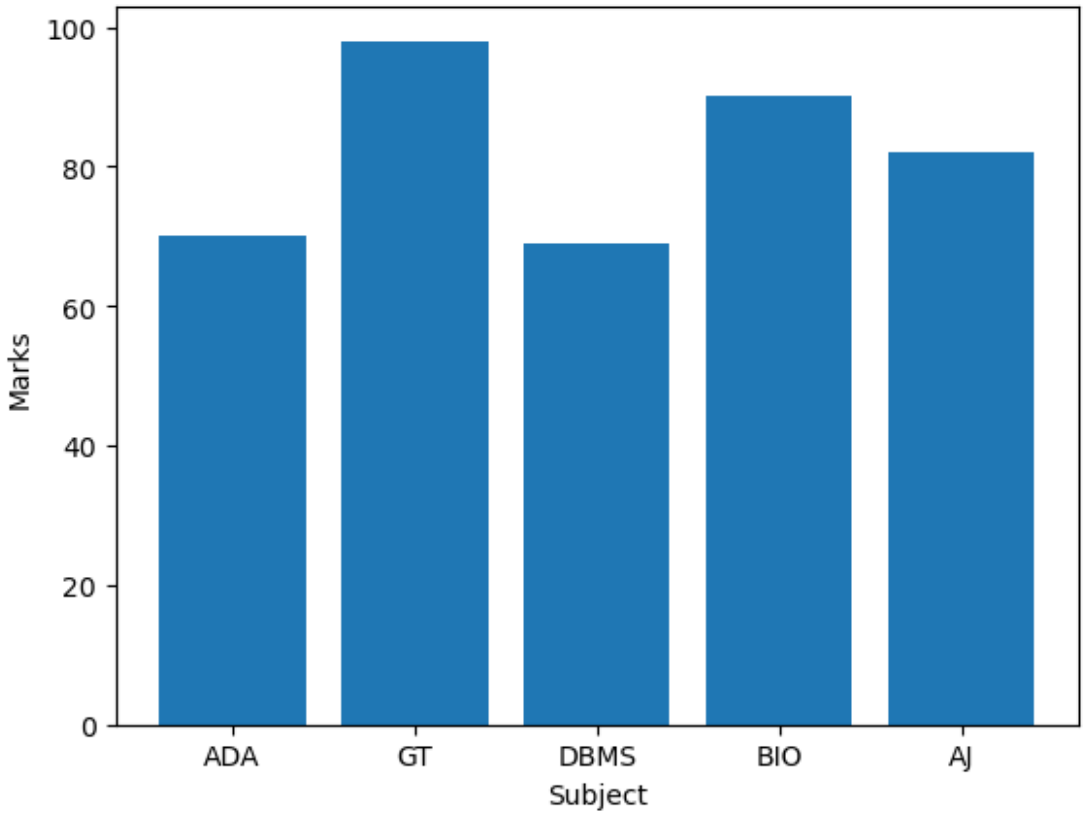
```
<matplotlib.legend.Legend at 0x1faf7c991f0>
```




```
sub=["ADA","GT","DBMS","BIO","AJ"]
x=[99,98,89,90,88]
plt.bar(sub,x)
plt.xlabel("Subject")
plt.ylabel("Marks")
plt.show()
```

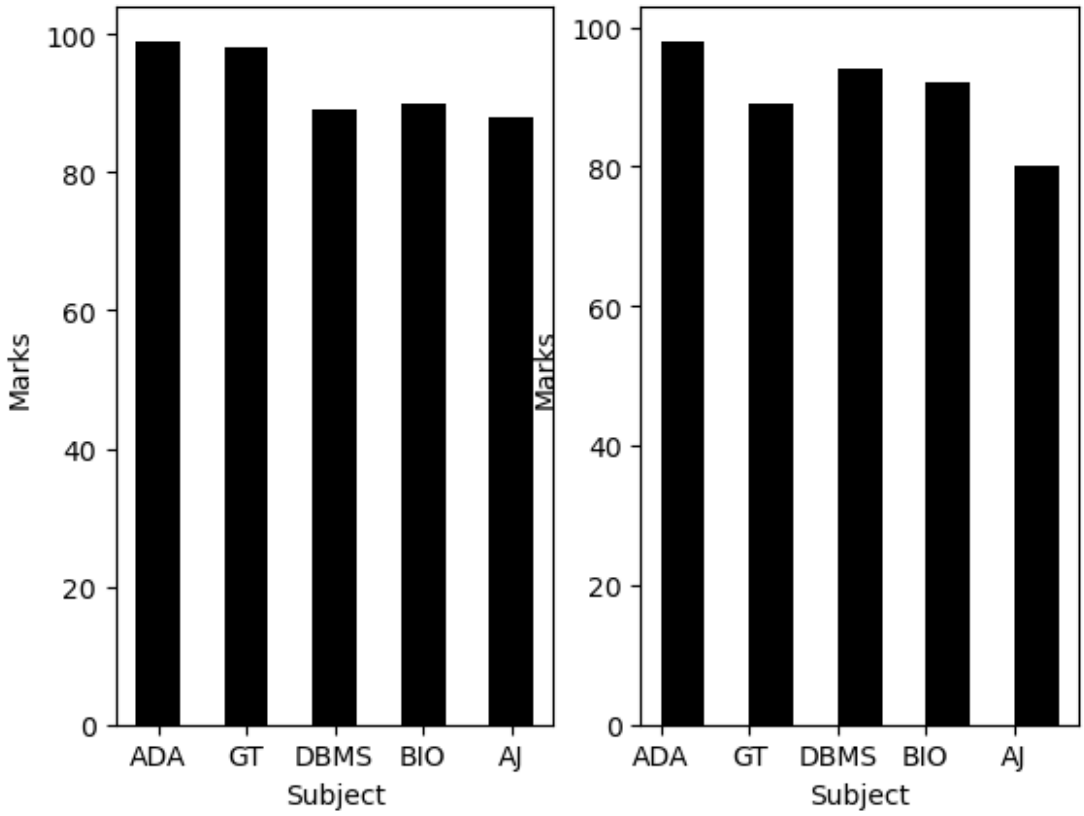



```
sub=["ADA","GT","DBMS","BIO","AJ"]
y=[70,98,69,90,82]
plt.bar(sub,y)
plt.xlabel("Subject")
plt.ylabel("Marks")
plt.show()
```



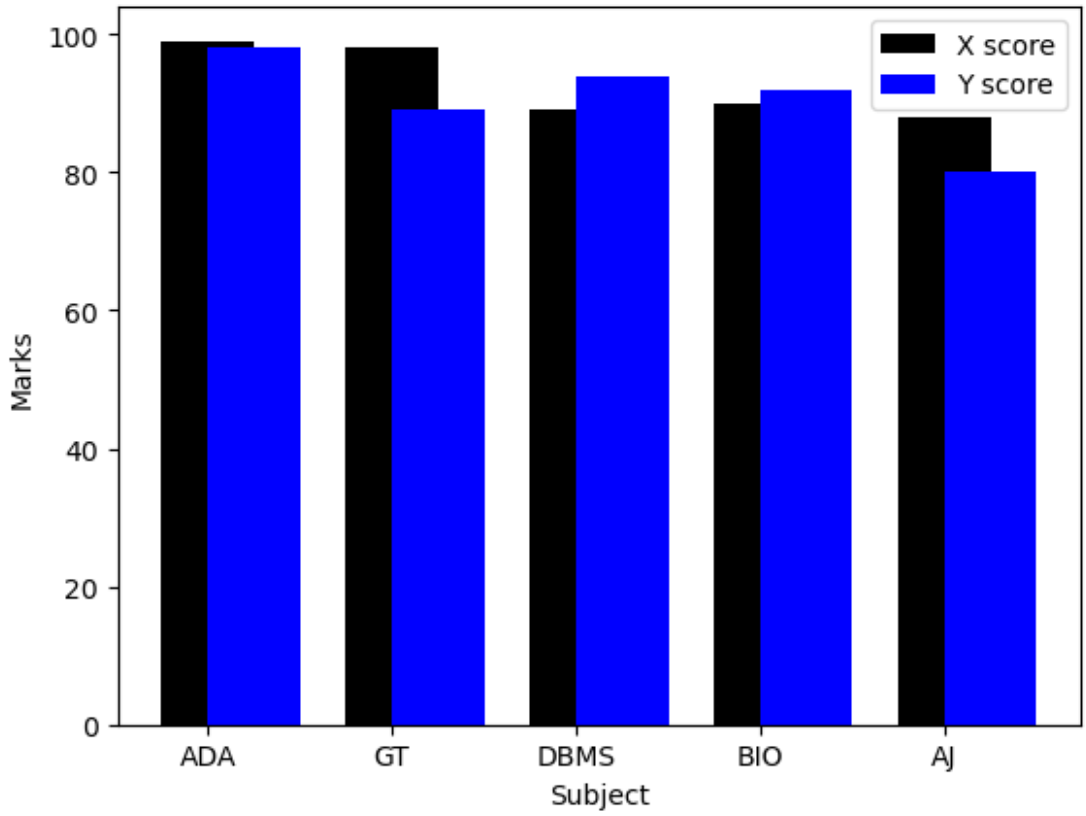

```
sub=["ADA","GT","DBMS","BIO","AJ"]
x=[99,98,89,90,88]
y=[98,89,94,92,80]
plt.subplot(1,2,1)
plt.bar(sub,x,color='k',label="X Score",width=0.5,align="center")
plt.xlabel("Subject")
plt.ylabel("Marks")

#second graph
plt.subplot(1,2,2)
plt.bar(sub,y,color='k',label="Y Score",width=0.5,align="edge")
plt.xlabel("Subject")
plt.ylabel("Marks")
```

```
Text(0, 0.5, 'Marks')
```

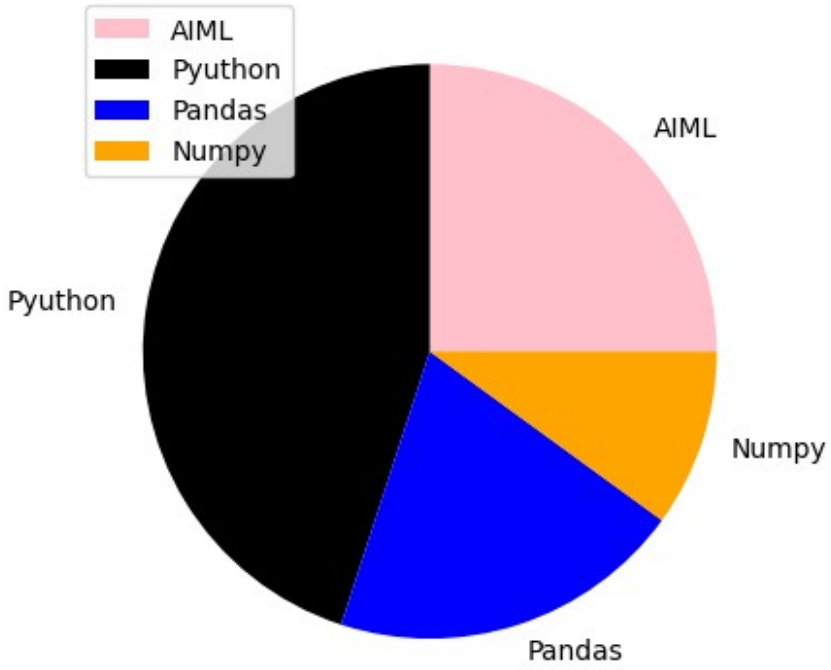



```
sub=["ADA","GT","DBMS","BIO","AJ"]
x=[99,98,89,90,88]
y=[98,89,94,92,80]
plt.bar(sub,x,color='k',label="X score",width=0.5,align="center")
plt.bar(sub,y,color='b',label="Y score",width=0.5,align="edge")
plt.xlabel("Subject")
plt.ylabel("Marks")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1fa8062aed0>
```




```
import numpy as np
a=np.array([25,45,20,10])
label=["AIML","Pyuthon","Pandas","Numpy"]
color=["pink","black","blue","orange"]
plt.pie(a,labels=label,colors=color)
plt.legend()
plt.show()
```


AIML
Pyuthon
Pandas
Numpy
AIML
Pyuthon
Numpy
Pandas