Pandas

```
import pandas as pd
```

series

```
mydata1=["virat","abd","faf","maxi","siraj"]
ser1=pd.Series(mydata1)
print(ser1)
```

```
0    virat
1      abd
2      faf
3     maxi
4    siraj
dtype: object
```

```
ser1[3]
```

```
'maxi'
```

```
mydata1=["virat","abd","faf","maxi","siraj"]
j_no=['18','17','13','32','73']
ser1=pd.Series(mydata1,j_no)
print(ser1)
```

```
18    virat
17      abd
13      faf
32     maxi
73    siraj
dtype: object
```

```
ser1['17']
```

```
'abd'
```

```
ser1.to_csv("C:\my python files\mydata1.csv")
```

DataFrames

```
mydict={"Names":["Raj","Rakesh","Sumit"],
        "Age":[19,19,20],
        "City":["Raichur","Shimoga","Bidar"]
       }
print(mydict)
```

```
{'Names': ['Raj', 'Rakesh', 'Sumit'], 'Age': [19, 19, 20], 'City':
['Raichur', 'Shimoga', 'Bidar']}
```

```
dict_df=pd.DataFrame(mydict)
print(dict_df)
```

```
    Names  Age     City
0     Raj   19  Raichur
1  Rakesh   19  Shimoga
2   Sumit   20    Bidar
```

```
dict_df.to_csv("C:\my python files\mydict.csv")
```

Load data

```
df1=pd.read_csv("C:\mypythonfiles\sampledata.csv")

df1.head()
```

```
      Name Dept  Sem1  Sem2  Sem3
0      Sam  ECE   6.7   8.9   7.2
1   Rakesh  ISE   7.9   8.9   NaN
2    Sumit  ISE   8.2   7.9   8.1
3  prajwal  ISE   8.4   7.8   8.3
4    vijay  ISE   NaN   7.5   8.9
```

loading a large files

```
diab_df=pd.read_csv("C:\mypythonfiles\diabetcsvsmall.csv")
diab_df.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
```

```
diab_df.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
```

Access

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

```
12    57
13    59
```

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

```
diab_df.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
```

```
diab_df.iloc[12:19,3:8] #datframe.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,skin,mass,preg,pedi,age, ==> independent(Feature) class ==> dependent ==>Target (depends on feature)

```
diab_df.rename(columns ={"plas" : "Glucose"},inplace=True)
#datdframe.rename(columns={"old":"new"}, inplace+True}
```

```
diab_df.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
```

```
diab_df['Glucose_in_mmol'] = diab_df['Glucose']/18.018
#dataframe['new column_name'] = content
#converting glucose from mg to mmol and creating the new column
```

```
diab_df.head(12)
```

```
    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

10   4.0      110  92.0   0.0     0  37.6  0.191   30  tested_negative

11  10.0      168  74.0   0.0     0  38.0  0.537   34  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
10         6.105006
11         9.324009
```

Filter and groups

```
fil_age_30less=diab_df[diab_df['age']<30]
#new df=your df [condition]
fil_age_30less.head()
```

```
    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_less_100=diab_df[diab_df['Glucose']<100]
glucose_less_100.head()

    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=diab_df[diab_df['Glucose']>100]
glucose_above_100.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

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 filter dataset which as only the rows with age b/w 20 and 30

```
fil_age_20_to_30=diab_df[(diab_df['age']>20) &(diab_df['age']<30)]
fil_age_20_to_30.head(7)
```

```
    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


    Glucose_in_mmol
3          4.939505
6          4.329004
7          6.382506
20         6.993007
23         6.604507
27         5.383505
31         8.769009
```

Grouping and derving

```
#group by class and calculate the average
group_class_ins=diab_df.groupby('class')['age'].mean()

print(group_class_ins)
```

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

```
group_class_ins=diab_df.groupby('class')['insu'].mean()
group_class_ins
```

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

```
group_class_min =diab_df.groupby('class')['age'].min()
group_class_min
#The least age of diabetic people is 25
#the least age of non-diabetic people is 21
```

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

```
group_class_max =diab_df.groupby('class')['age'].max()
group_class_max
#the maximum age of diabetic people is 60
#the maximum age of non-diabetic people is 60
```

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

cleaning data

Handling the null

```
diab_df.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
```

```
diab_df.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]
```

```
diab_df.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
```

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

```
diab_df.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
```

```
diab_df.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
```

Handling Duplicates

```
diab_df.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
```

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

```
diab_df.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("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("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
```

```
dia_ex=pd.read_excel("C:\mypythonfiles\
diabetes.xlsx",sheet_name="Hello")
dia_ex.head()
```

```
Empty DataFrame
Columns: [hello, guys, how, are ]
Index: []
```

```
#loading the txt file
df_txt=pd.read_csv("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("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 the datatypes

```
df_txt['SEM_int']=df_txt['SEM1'].astype(int)
df_txt.head()
```

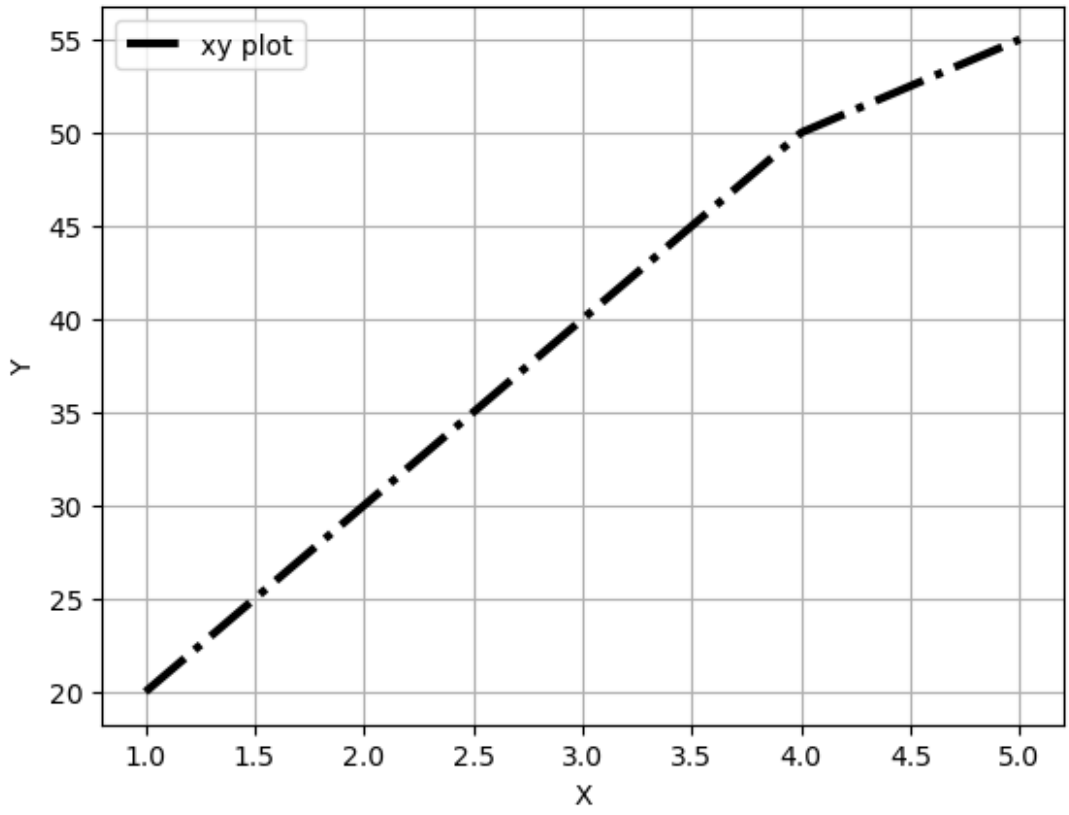
```
    Names Initials  SEM1  SEM2  SEM3 Grade  SEM_int
0     Joe        K   9.8  10.0   9.9    A+        9
1  Rajesh        M   8.9   9.1   9.3     A        8
```

```
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

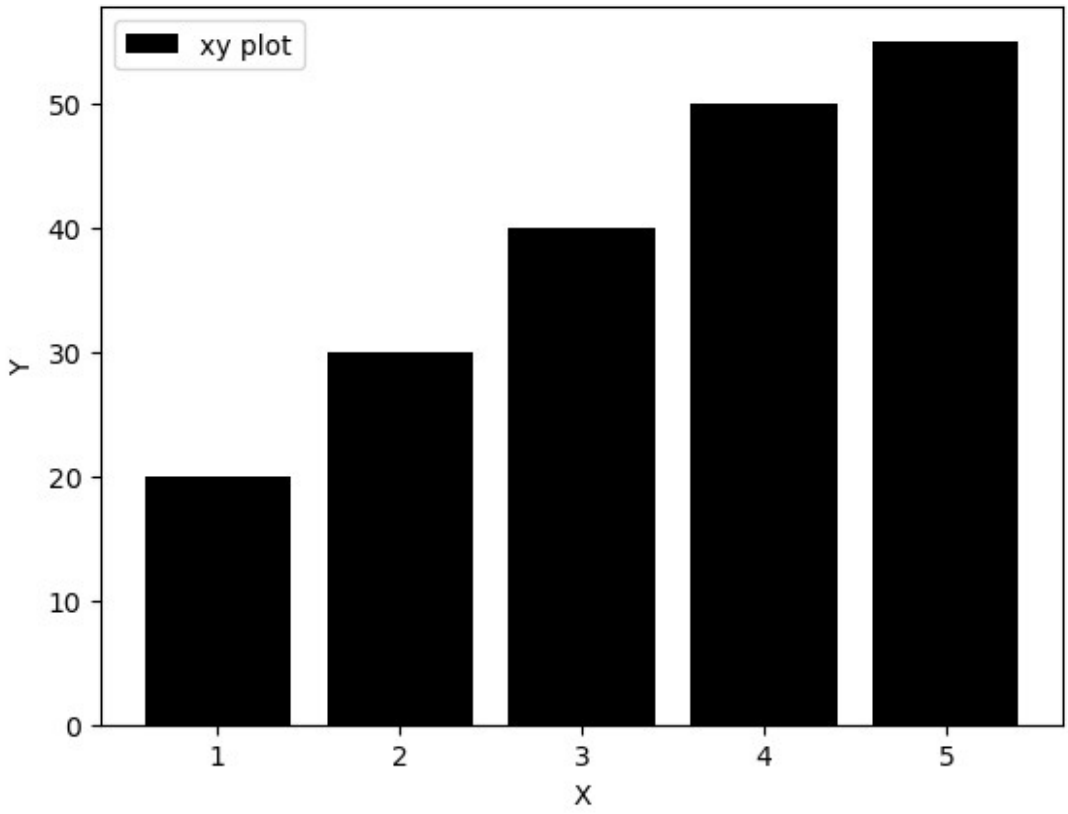
```
x=[1,2,3,4,5]
y=[20,30,40,50,55]

import matplotlib.pyplot as plt
plt.plot(x,y, color='k',label="xy plot",linestyle="-.",linewidth=3)
plt.xlabel("X")
plt.ylabel("Y")
plt.grid()
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1ebbd2e21d0>
```

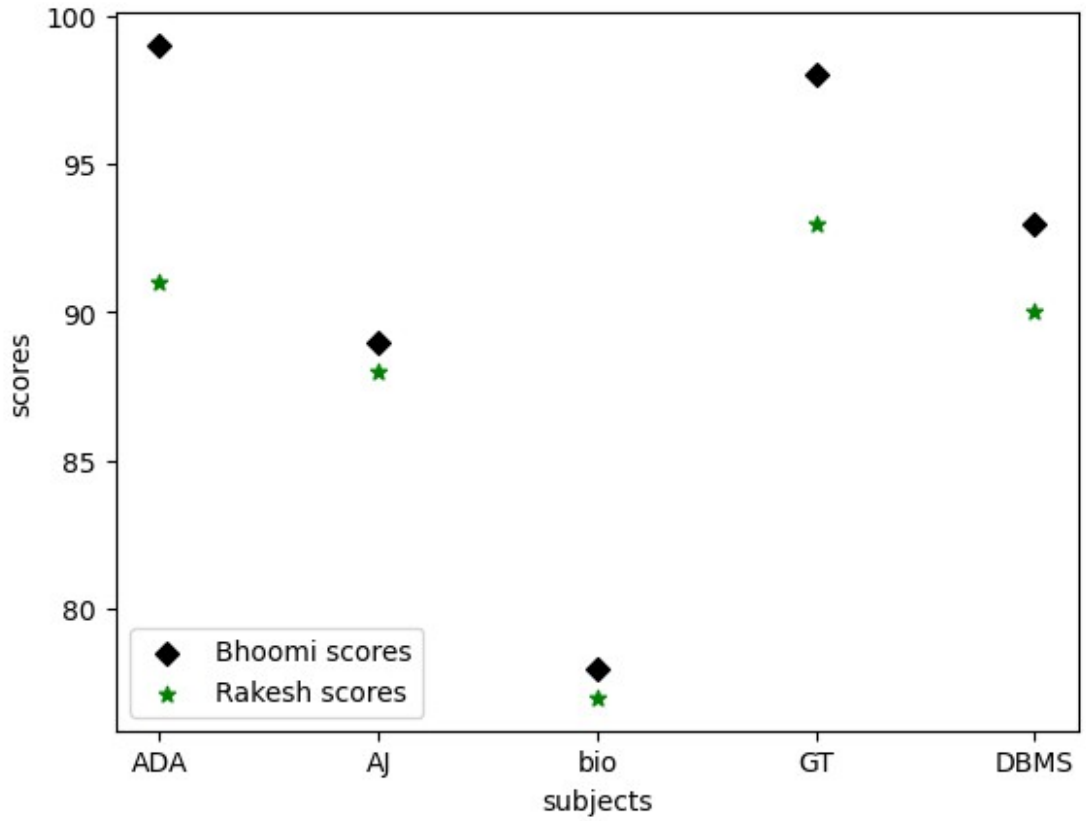



```
import matplotlib.pyplot as plt
plt.bar(x,y, color='k',label="xy plot",linestyle="-.",linewidth=3)
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1ebbd3567d0>
```

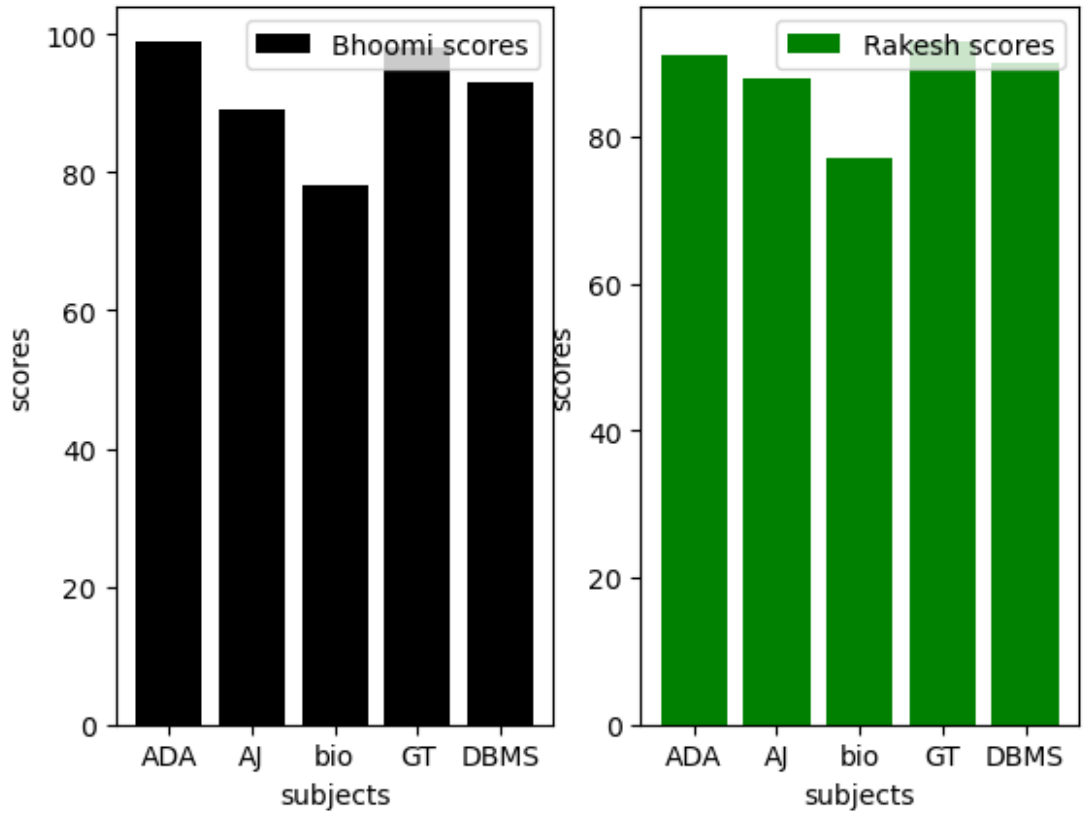



```
sub=["ADA","AJ","bio","GT","DBMS"]
Bhoomi=[99,89,78,98,93]
Rakesh=[91,88,77,93,90]
plt.scatter(sub,Bhoomi,color='k',label="Bhoomi scores",marker="D")
plt.scatter(sub,Rakesh,color='Green',label="Rakesh scores",marker="*")
plt.xlabel("subjects")
plt.ylabel("scores")
plt.legend()

<matplotlib.legend.Legend at 0x1ebbdc66710>
```



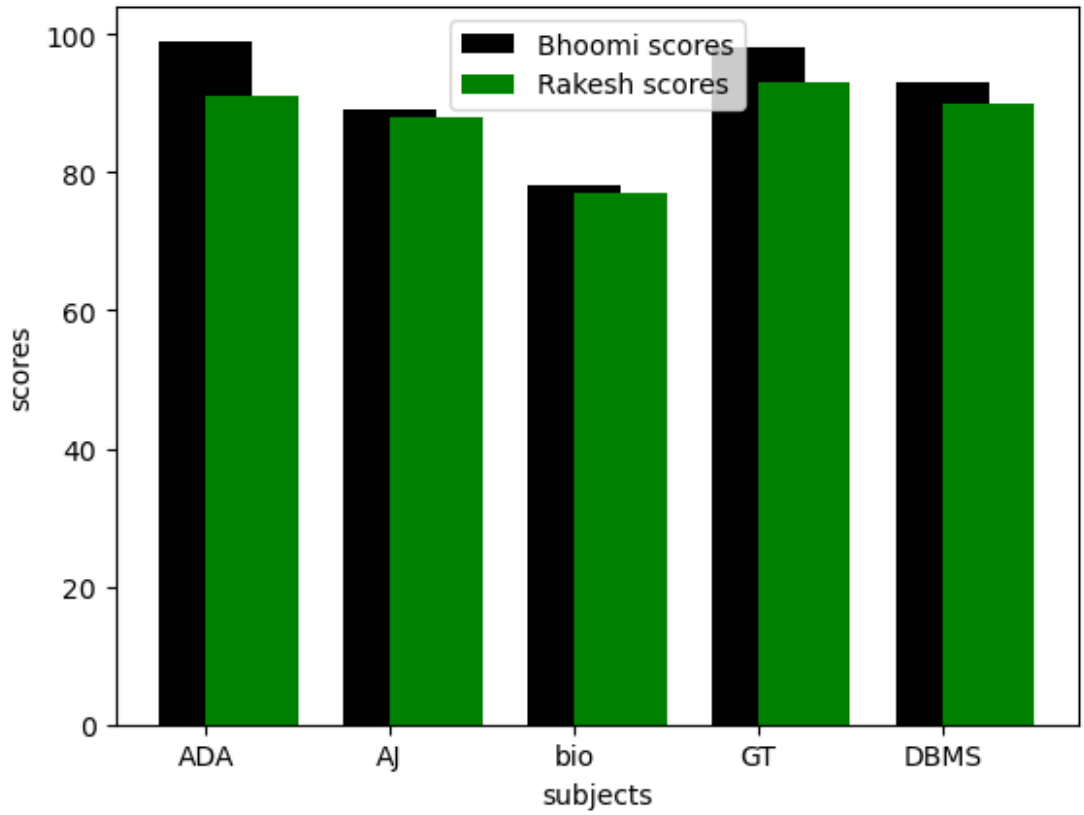

```
sub=["ADA","AJ","bio","GT","DBMS"]
Bhoomi=[99,89,78,98,93]
Rakesh=[91,88,77,93,90]
plt.subplot(1,2,1)
plt.bar(sub,Bhoomi,color='k',label="Bhoomi scores")
plt.xlabel("subjects")
plt.ylabel("scores")
plt.legend()
plt.subplot(1,2,2)
plt.bar(sub,Rakesh,color='Green',label="Rakesh scores")
plt.xlabel("subjects")
plt.ylabel("scores")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1ebc228a290>
```



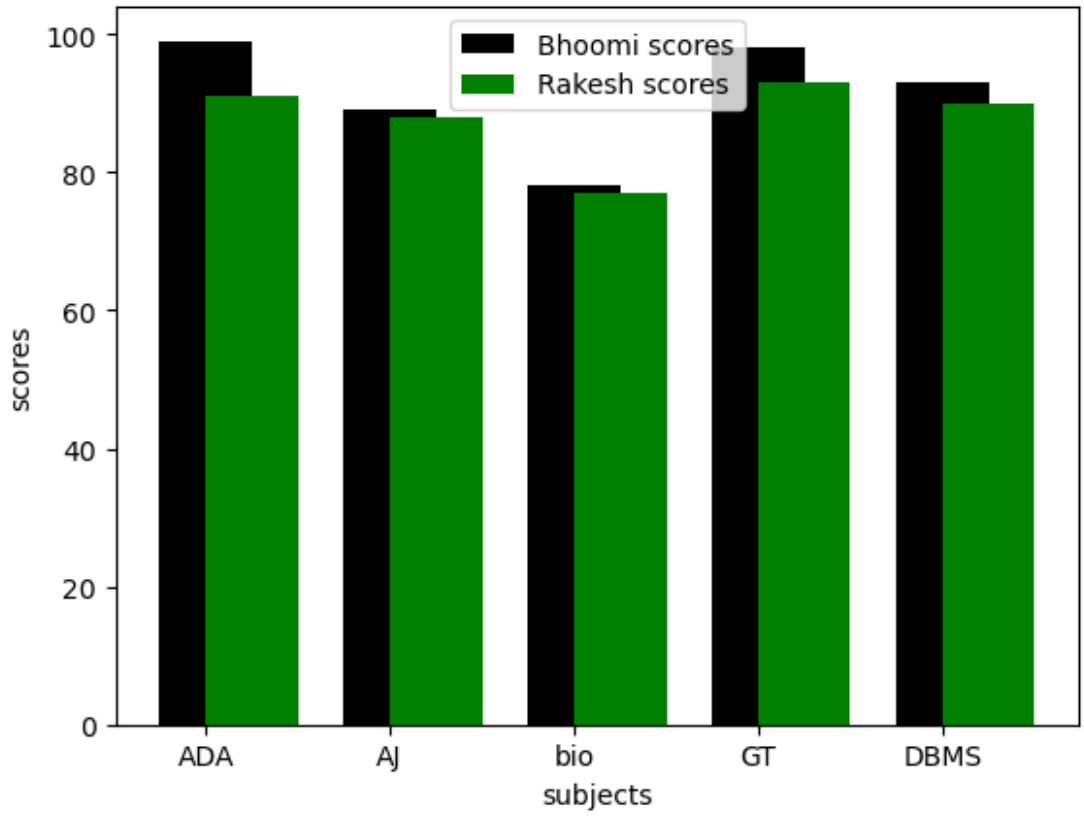

```
sub=["ADA","AJ","bio","GT","DBMS"]
Bhoomi=[99,89,78,98,93]
Rakesh=[91,88,77,93,90]
plt.bar(sub,Bhoomi,color='k',label="Bhoomi
scores",width=0.5,align="center")
plt.bar(sub,Rakesh,color='Green',label="Rakesh
scores",width=0.5,align="edge")
plt.xlabel("subjects")
plt.ylabel("scores")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1ebc24b6710>
```

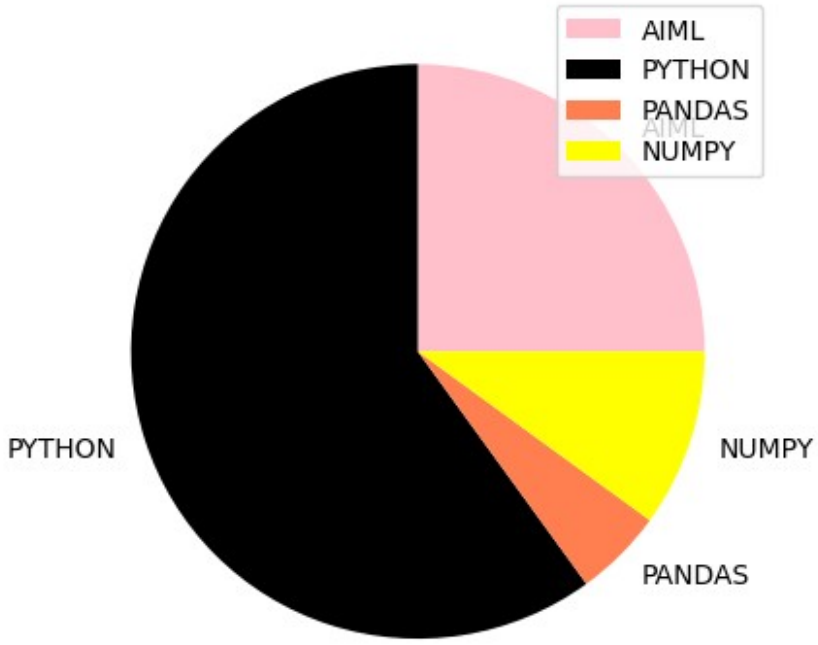



```
sub=["ADA","AJ","bio","GT","DBMS"]
Bhoomi=[99,89,78,98,93]
Rakesh=[91,88,77,93,90]
plt.bar(sub,Bhoomi,color='k',label="Bhoomi
scores",width=0.5,align="center")
plt.bar(sub,Rakesh,color='Green',label="Rakesh
scores",width=0.5,align="edge")
plt.xlabel("subjects")
plt.ylabel("scores")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1ebbd4bded0>
```




```
import numpy as np

a=np.array([25,60,5,10])
labe=["AIML","PYTHON","PANDAS","NUMPY"]
color=['pink','black','coral','yellow']
plt.pie(a,labels= labe,colors=color)
plt.legend()
plt.show()
```


AIML
PYTHON
PANDAS
NUMPY
AIML
PYTHON
NUMPY
PANDAS