

Pandas

```
import pandas as pd

mydata=["Deepak","Darshan","Amith","Ullas","Raj"]
ser1=pd.Series(mydata)
print(ser1)

0    Deepak
1    Darshan
2     Amith
3     Ullas
4      Raj
dtype: object

ser1[3]

'Ullas'

mydata=["Deepak","Darshan","Amith","Ullas","Raj"]
roll=['A','B','C','D','E']
ser1=pd.Series(mydata,index=roll)
print(ser1)

A    Deepak
B    Darshan
C     Amith
D     Ullas
E      Raj
dtype: object

mydata=["Deepak","Darshan","Amith","Ullas","Raj"]
rollno=['10','09','01','405','27']
ser1=pd.Series(mydata,index=rollno)
print(ser1)

10    Deepak
09    Darshan
01     Amith
405    Ullas
27     Raj
dtype: object

ser1.to_csv(r"C:\Users\DELL\Downloads\mydata.csv")
```

DataFrames

```
mydict = {"Name" : ["Deepak","Darshan","Amith"],
          "Age"  : [18,19,20],
```

```

    "City" : ["Chitradurga","Chitradurga","Tumkur"]}]
print(mydict)

{'Name': ['Deepak', 'Darshan', 'Amith'], 'Age': [18, 19, 20], 'City':
['Chitradurga', 'Chitradurga', 'Tumkur']}

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

```

	Name	Age	City
0	Deepak	18	Chitradurga
1	Darshan	19	Chitradurga
2	Amith	20	Tumkur

```

dict_df.to_csv(r"C:\Users\DELL\Downloads\mydict.csv")

df1=pd.read_csv(r"C:\Users\DELL\Downloads\Samplefile.csv")

df1.head()

```

	Name	Dept	Sem1	Sem2	Sem3
0	Deepu	ISE	7.4	7.8	7.9
1	Manju	ECE	7.0	7.4	7.6
2	Darshu	ISE	8.3	8.4	8.4
3	Prajju	ISE	7.0	NaN	7.3
4	Raj	ISE	7.0	7.4	7.8

```

diab_df=pd.read_csv(r"C:\Users\DELL\Downloads\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[10:18]

```

	preg	plas	pres	skin	insu	mass	pedi	age	class
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
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

```
diab_df.loc[10:18, "age"]
```

10	30
11	34
12	57
13	59
14	51
15	32
16	31
17	31
18	33

Name: age, dtype: int64

```
diab_df.iloc[10:18, 3:8]
```

	skin	insu	mass	pedi	age
10	0.0	0	37.6	0.191	30
11	0.0	0	38.0	0.537	34
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

Feature Engineering

```
diab_df.rename(columns = {"plas" : "Glucose"}, 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
```

```
diab_df.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=diab_df[diab_df["age"]<30]
```

```
fil_age_30less.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

```
glu_100=diab_df[diab_df["Glucose"]>100]
glu_100.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
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
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
12	10.0	139	80.0	0.0	0	27.1	1.441	57	tested_negative

	Glucose_in_mmol
0	8.214008
2	10.156510
4	7.603508
5	6.438006
7	6.382506
8	10.933511
9	6.937507
10	6.105006
11	9.324009
12	7.714508

Create a filter dataset which as only the rows with age between 20 and 30

```
age_20_30=diab_df[(diab_df["age"]>20) & (diab_df["age"]<30)]
age_20_30.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

Grouping and Deriving Results

```
g=diab_df.groupby("class")["age"].mean()
g
```

class	
tested_negative	31.238095

```

tested_positive    40.589744
Name: age, dtype: float64

g=diab_df.groupby("class")['age'].max()
g

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

g=diab_df.groupby("class")['age'].min()
g

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

g=diab_df.groupby("class")['insu'].mean()
g

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

```

Cleaning Data

Handling Null

```

diab_df.isnull().sum()

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

diab_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 9 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
dtypes: float64(5), int64(3), object(1)
memory usage: 7.3+ 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
dtype: int64

```

Handling Duplicate

```
diab_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 9 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

```



```
dtypes: float64(5), int64(3), object(1)
memory usage: 7.7+ KB
```

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

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 96 entries, 0 to 101
```

```
Data columns (total 9 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

```
dtypes: float64(5), int64(3), object(1)
```

```
memory usage: 7.5+ KB
```

Reading Other Formats

```
dia_ex = pd.read_excel(r"C:\Users\DELL\Downloads\diabets1.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_sheet2=pd.read_excel(r"C:\Users\DELL\Downloads\diabets1.xlsx",sheet_name='dora')
```

```
dia_ex_sheet2.head()
```

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

Loading text file

```
df_text=pd.read_csv(r"C:\Users\DELL\Downloads\grades.txt",sep = ' ')
df_text.head(10)
```

	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+
5	Raj	M	8.9	9.1	9.3	A
6	Hassan	V	9.9	9.0	9.2	A
7	Mari	N	7.7	8.0	7.1	B
8	Jess	K	9.8	9.1	9.9	A+
9	Rajini	M	7.0	9.1	9.3	A

Modifying Data type

```
df_text['SEM1_int'] = df_text['SEM1'].astype(int)
df_text.head()
```

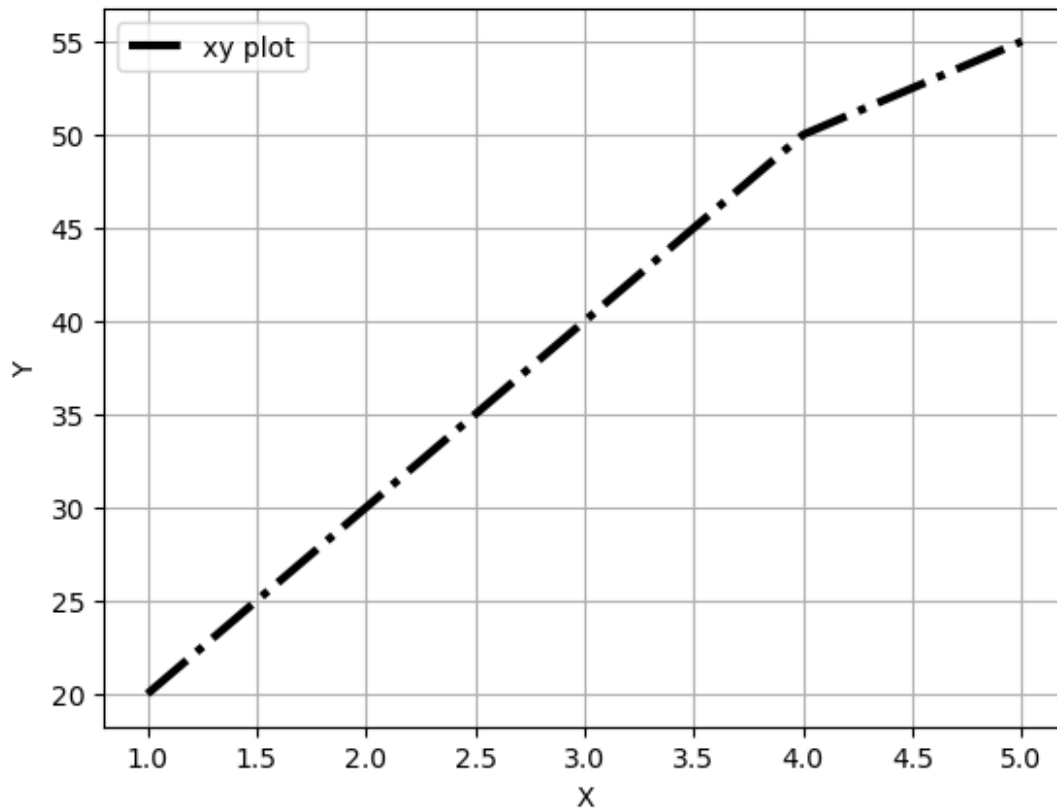
	Names	Initials	SEM1	SEM2	SEM3	Grade	SEM1_int
0	Joe	K	9.8	10.0	9.9	A+	9
1	Rajesh	M	8.9	9.1	9.3	A	8
2	Kissan	V	9.9	9.3	9.2	A	9
3	Mary	N	7.7	8.0	7.1	B	7
4	Jeen	K	9.8	9.1	9.9	A+	9

Matplotlib

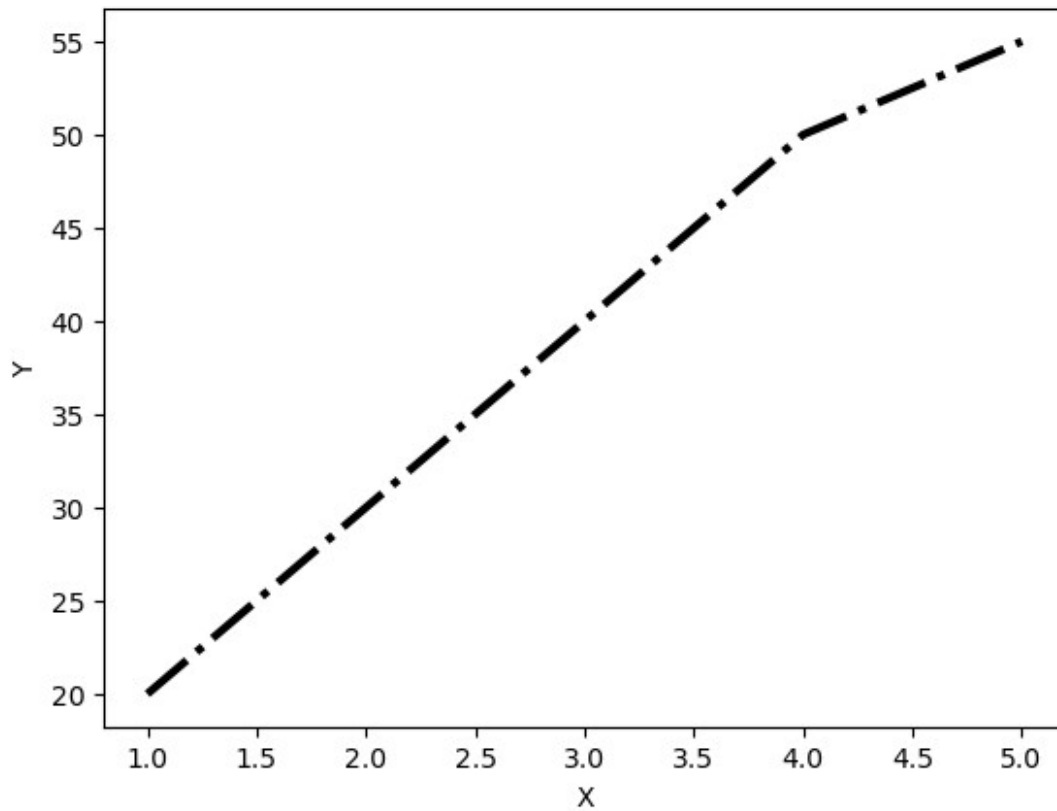
```
x=[1,2,3,4,5]
y=[20,30,40,50,55]

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()
```

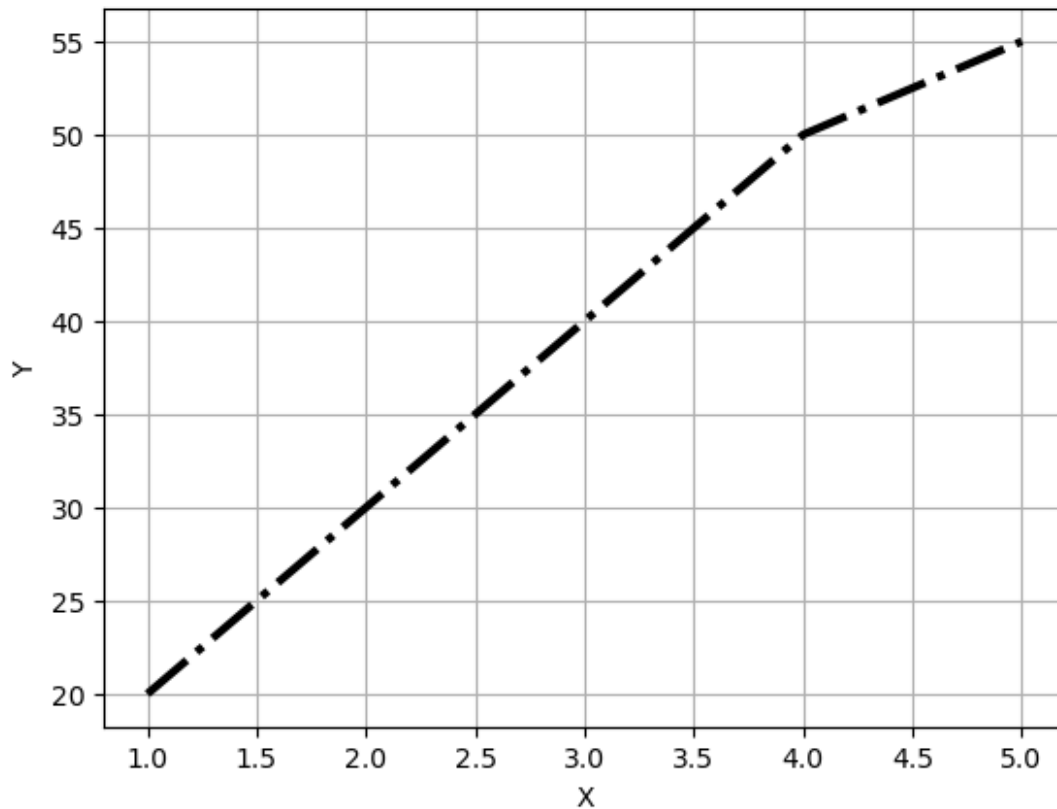
```
<matplotlib.legend.Legend at 0x13503010510>
```



```
import matplotlib.pyplot as plt
plt.plot(x,y,color='k',label='xy plot',linestyle='-.',linewidth=3)
plt.xlabel("X")
plt.ylabel("Y")
Text(0, 0.5, 'Y')
```

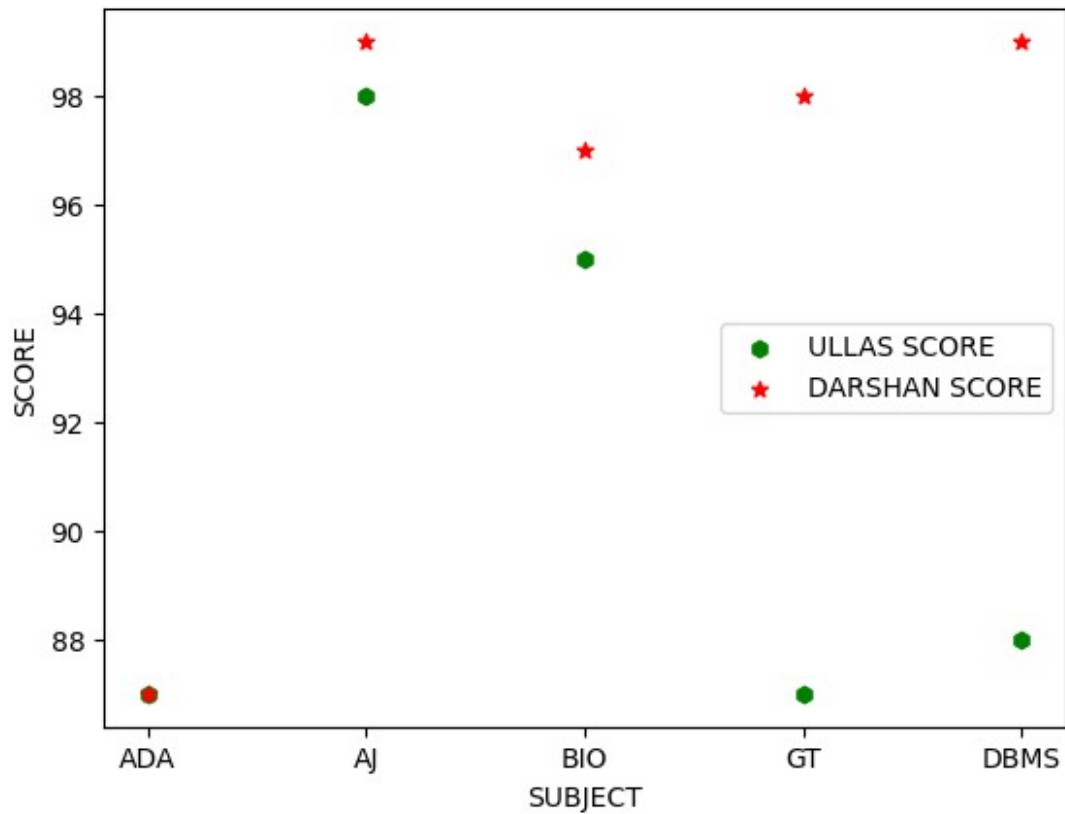


```
import matplotlib.pyplot as plt
plt.plot(x,y,color='k',label='xy plot',linestyle='-.',linewidth=3)
plt.xlabel("X")
plt.ylabel("Y")
plt.grid()
```



```
sub = ['ADA', 'AJ', 'BIO', 'GT', 'DBMS']
Ullas = [87, 98, 95, 87, 88]
Darshan = [87, 99, 97, 98, 99]
plt.scatter(sub, Ullas, color='green', label='ULLAS SCORE', marker='h')
plt.scatter(sub, Darshan, color='red', label='DARSHAN SCORE', marker='*')
plt.xlabel("SUBJECT")
plt.ylabel("SCORE")
plt.legend()
```

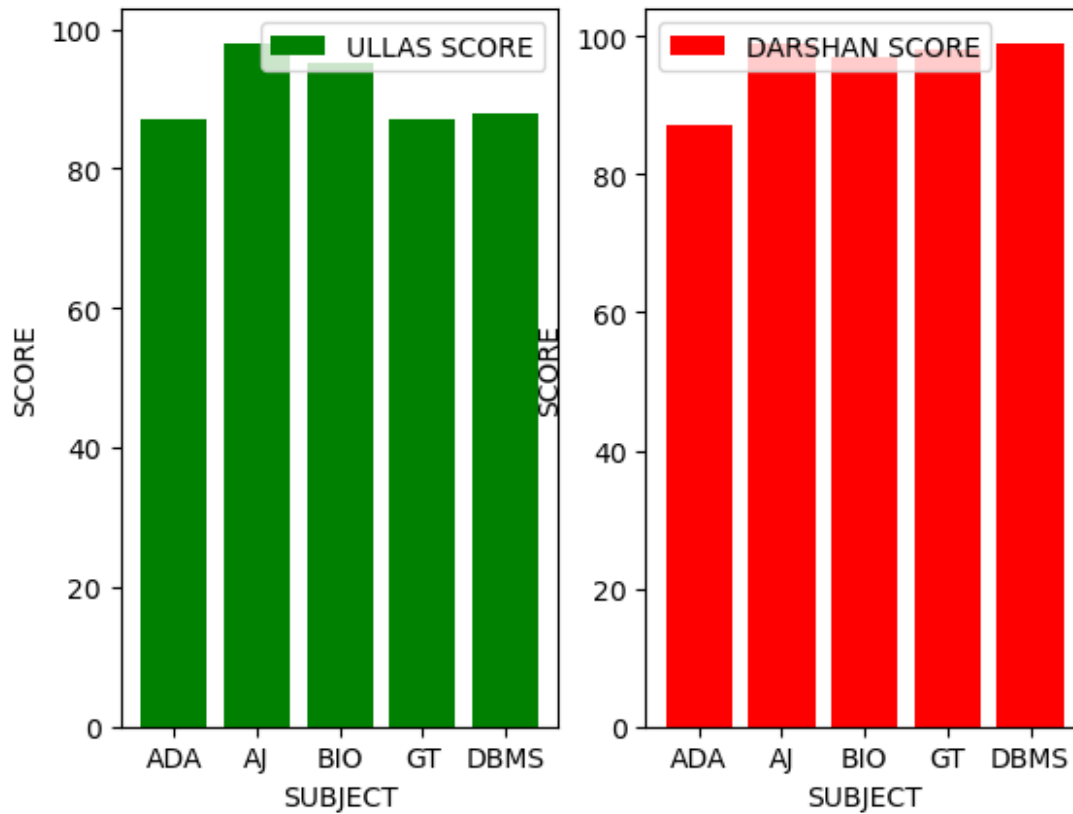
<matplotlib.legend.Legend at 0x135081cde90>



```
sub = ['ADA', 'AJ', 'BIO', 'GT', 'DBMS']
Ullas = [87, 98, 95, 87, 88]
Darshan = [87, 99, 97, 98, 99]
plt.subplot(1, 2, 1)
plt.bar(sub, Ullas, color='green', label='ULLAS SCORE')
plt.xlabel("SUBJECT")
plt.ylabel("SCORE")
plt.legend()

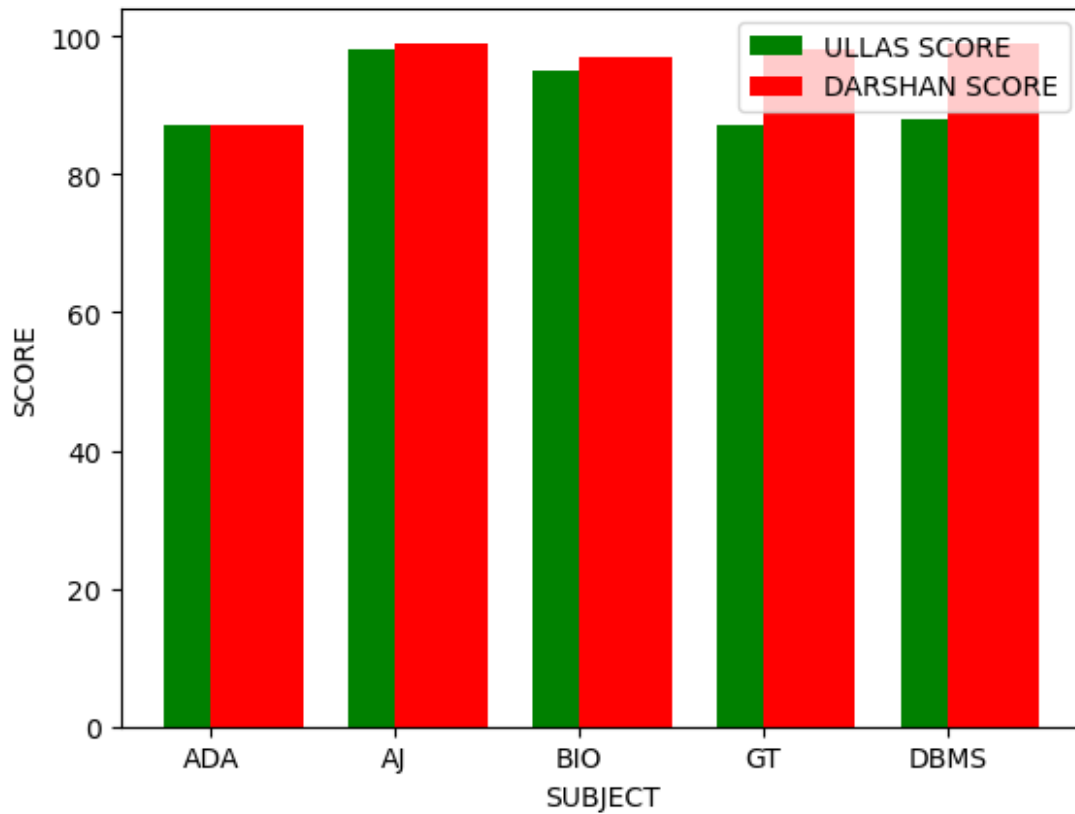
plt.subplot(1, 2, 2)
plt.bar(sub, Darshan, color='red', label='DARSHAN SCORE')
plt.xlabel("SUBJECT")
plt.ylabel("SCORE")
plt.legend()

<matplotlib.legend.Legend at 0x135086e2250>
```



```
sub = ['ADA', 'AJ', 'BIO', 'GT', 'DBMS']
Ullas = [87, 98, 95, 87, 88]
Darshan = [87, 99, 97, 98, 99]
plt.bar(sub, Ullas, color='green', label='ULLAS SCORE', width=0.5, align="center")
plt.bar(sub, Darshan, color='red', label='DARSHAN SCORE', width=0.5, align="edge")
plt.xlabel("SUBJECT")
plt.ylabel("SCORE")
plt.legend()
```

<matplotlib.legend.Legend at 0x135088bc250>



```
import numpy as np  
  
a=np.array([25,60,5,10])  
labe = ["ATML","PYTHON","PANDAS","NUMPY"]  
color = ['coral','green','red','yellow']  
plt.pie(a,labels = labe,colors=color)  
plt.legend()  
plt.show()
```