

PANDAS

```
import pandas as pd
```

series

```
mydata1 = ["ananya", "aishwitha", "monisha", "nayana", "gouthmi"]
Ser1 = pd.Series(mydata1)
print(Ser1)
```

```
0      ananya
1    aishwitha
2      monisha
3      nayana
4      gouthmi
dtype: object
```

```
Ser1[1]
```

```
'aishwitha'
```

```
mydata1 = ["anany", "aishwitha", "monisha", "nayana", "monisha"]
roll = [1, 2, 3, 4, 5]
Ser2 = pd.Series(mydata1, index=roll)
print(Ser2)
```

```
1      anany
2    aishwitha
3      monisha
4      nayana
5      monisha
dtype: object
```

```
Ser2[2]
```

```
'aishwitha'
```

```
Ser2.to_csv("C:\my files\mydata1.csv")
```

dataframes

```
mydict = {"Names" : ["ram", "raj", "raju"],
          "Age" : [19, 20, 19],
          "City" : ["ckm", "mnglr", "bnblr"]}
print(mydict)
```

```
{'Names': ['ram', 'raj', 'raju'], 'Age': [19, 20, 19], 'City': ['ckm', 'mnglr', 'bnblr']}
```

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

	Names	Age	City
0	ram	19	ckm
1	raj	20	mnglr
2	raju	19	bnblr

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

load data

```
df1 = pd.read_csv("C:\my files\sampledatal.CSV")
```

```
df1.head()
```

	name	dept	sem1	sem2	sem3
0	aishwitha	ISE	8.3	8.34	9.0
1	monisha	ISE	NaN	8.20	8.9
2	gouthami	ISE	8.0	8.00	8.0
3	pavi	ISE	8.0	NaN	8.2
4	nayana	ISE	9.0	9.00	NaN

loading large file

```
import pandas as pd
```

```
diab_df = pd.read_csv("C:\my files\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] #datafram.ioc[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

skin,insu,mass,pedi,age,pre,pres ==> independent(feature)

class ==> deperdent(target)

```
diab_df.rename(columns={"plas":"glucose"})
```

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

```

```
[102 rows x 9 columns]
```

```
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.rename(columns={"plas":"glucose"},inplace = True)
#datafram.rename(colums={"old":"new"},inplace = True) if inplace is
not given it will just show but not save

```

```
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 col name"] = content
#converting glucose from mg to mmol and creating new col

```

```
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_above100 = diab_df[diab_df["glucose"]>100]
glucose_above100.head(7)

```

	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

```

      glucose_in_mmol
0      8.214008
2     10.156510
4      7.603508
5      6.438006
7      6.382506
8     10.933511
9      6.937507

```

```

glucose_below100 = diab_df[diab_df["glucose"]<100]
glucose_below100.head(7)

```

	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
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

	glucose_in_mmol
1	4.717505
3	4.939505
6	4.329004
21	5.494505
27	5.383505
32	4.884005
33	5.106005

create a filter data set which has only the rows with age btw 20 and 30

```
age_20and30 = diab_df[(diab_df['age']>20)&(diab_df['age']<30)]
age_20and30.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

#group by class and calucate the average

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

class	
tested_negative	31.238095
tested_positive	40.589744

Name: age, dtype: float64

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.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.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.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.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

```
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(excle)

```
dia_ex = pd.read_excel("C:\my files\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_sheet2 = pd.read_excel("C:\my files\
diabetes.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

text file

```
df_txt = pd.read_csv("C:\my files\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:\my files\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 datatype

```
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

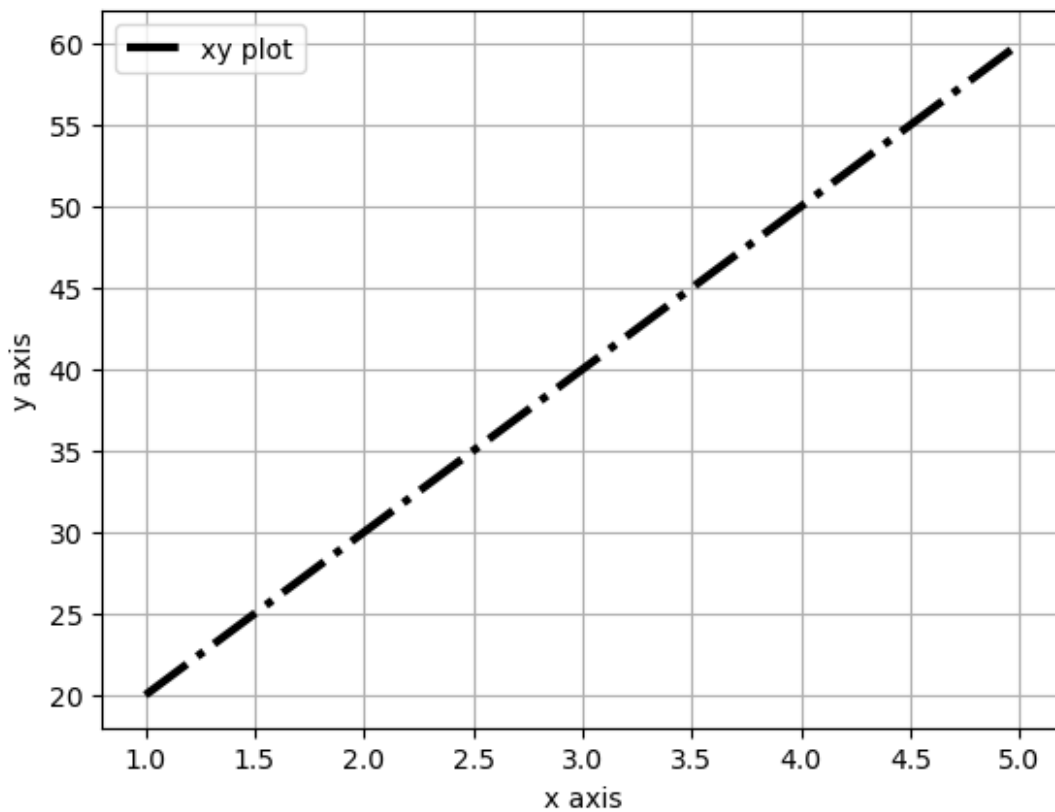
```

x=[1,2,3,4,5]
y=[20,30,40,50,60]

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

<matplotlib.legend.Legend at 0x14b0d5e9f50>

```

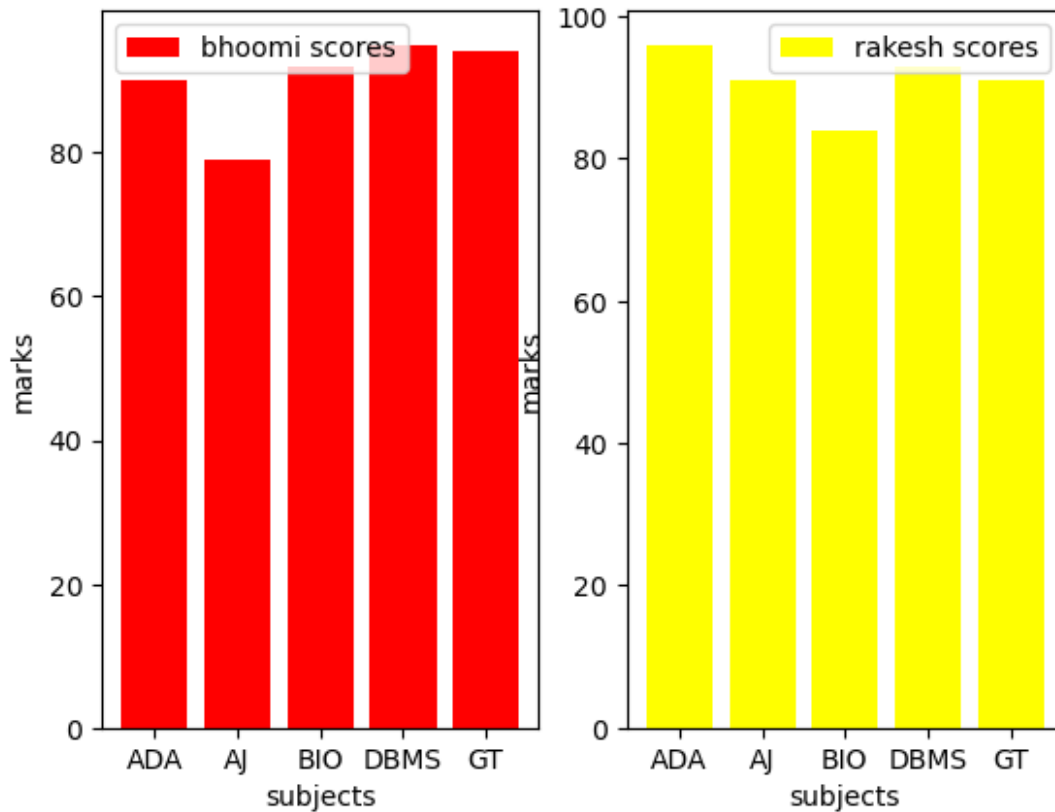


```

import matplotlib.pyplot as plt
sub = ["ADA", "AJ", "BIO", "DBMS", "GT"]
bhoomi = [90, 79, 92, 95, 94]
rakesh = [96, 91, 84, 93, 91]
#firstplot
plt.subplot(1,2,1)
plt.bar(sub, bhoomi, color = 'red', label = "bhoomi scores")
plt.xlabel("subjects")
plt.ylabel("marks")
plt.legend()
#second plot
plt.subplot(1,2,2)

```

```
plt.bar(sub, rakesh , color = 'yellow',label = "rakesh scores" )
plt.xlabel("subjects")
plt.ylabel("marks")
plt.legend()
plt.show()
```



```
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()
```

