

PDA_2_1

PANDAS

In [10]:

```
import pandas as pd
```

In [12]:

```
mydata=["amith","deepu","darshan","ranjith"]
ser1=pd.Series(mydata)
print(ser1)
```

```
0      amith
1      deepu
2    darshan
3    ranjith
dtype: object
```

In [14]:

```
mydata=["amith","deepu","darshan","ranjith"]
rollno=["1","2","3","4"]
ser2=pd.Series(mydata)
print(ser2)
```

```
0      amith
1      deepu
2    darshan
3    ranjith
dtype: object
```

In [16]:

```
ser2[3]
```

Out[16]:

```
'ranjith'
```

In [18]:

```
mydata=["amith","deepu","darshan","ranjith"]
rollno=["1","2","3","4"]
ser3=pd.Series(mydata,index=rollno)
print(ser3)
```

```
1      amith
2      deepu
3      darshan
4      ranjith
dtype: object
```

In [20]:

```
ser3.to_csv(r"C:\Users\DELL\Downloads\mydata.csv")
```

DATA FRAMES

In [23]:

```
mydict={"NAME":["DEEPAK","DARSHAN","RANJITH"],
        "AGE":["19","20","19"],
        "CITY":["CHITRADURGA","CHITRADURGA","CHITRADURGA"]}
print(mydict)
```

```
{'NAME': ['DEEPAK', 'DARSHAN', 'RANJITH'], 'AGE': ['19', '20', '19'],
 'CITY': ['CHITRADURGA', 'CHITRADURGA', 'CHITRADURGA']}
```

In [25]:

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

```
      NAME  AGE      CITY
0  DEEPAK   19  CHITRADURGA
1  DARSHAN   20  CHITRADURGA
2  RANJITH   19  CHITRADURGA
```

In [27]:

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

In [29]:

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

In [31]:

```
df1.head()
```

Out[31]:

	Name	Dept	Sem1	Sem2	sem3
0	Deepu	ISE	7.40	7.8	7.9
1	Amith	ISE	5.00	5.0	6.5
2	Darshan	ISE	8.30	8.4	8.4

	Name	Dept	Sem1	Sem2	sem3
3	prajwal	ISE	7.50	7.5	7.5
4	Suresh	ISE	6.45	7.0	NaN

In [33]:

```
diab_df=pd.read_csv(r"C:\Users\DELL\Downloads\diabetcsvsmall.csv")
```

In [35]:

```
diab_df.head()
```

Out[35]:

	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

In [37]:

```
diab_df.tail()
```

Out[37]:

	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

	preg	plas	pres	skin	insu	mass	pedi	age	class
99	NaN	122	90.0	51.0	220	49.7	0.325	31	gativ e teste d_po sitive
100	1.0	163	72.0	0.0	0	39.0	1.222	33	teste d_po sitive
101	1.0	151	60.0	0.0	0	26.1	0.179	22	teste d_ne gativ e

ACCESS

In [40]:

diab_df.loc[10:18]

Out[40]:

	preg	plas	pres	skin	insu	mass	pedi	age	class
10	4.0	110	92.0	0.0	0	37.6	0.191	30	teste d_ne gativ e
11	10.0	168	74.0	0.0	0	38.0	0.537	34	teste d_po sitive
12	10.0	139	80.0	0.0	0	27.1	1.441	57	teste d_ne gativ e
13	1.0	189	60.0	23.0	846	30.1	0.398	59	teste d_po sitive
14	5.0	166	72.0	19.0	175	25.8	0.587	51	teste d_po sitive
15	7.0	100	0.0	0.0	0	30.0	0.484	32	teste d_po sitive
16	0.0	118	84.0	47.0	230	45.8	0.55	31	teste

	preg	plas	pres	skin	insu	mass	pedi	age	class
							1		d_po sitive
17	7.0	107	74.0	0.0	0	29.6	0.25 4	31	teste d_po sitive
18	1.0	103	30.0	38.0	83	43.3	0.18 3	33	teste d_ne gativ e

In [42]:

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

Out[42]:

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

Name: age, dtype: int64

In [44]:

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

Out[44]:

	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

```
insu,mass,preg,plas,age,pres,skin==>Independent(feature)
class==>Dependent==>tangents(depends on features)
```

In [48]:

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

In [50]:

```
diab_df.head()
```

Out[50]:

	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

In [52]:

```
diab_df["Glucose_in_mol"]=diab_df["Glucose"]/18.018
```

In [54]:

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

Out[54]:

	preg	Glucose	pres	skin	insu	mass	pedi	age	class	Glucose_in_mol
0	6.0	148	72.0	35.0	0	33.6	0.62	50	teste	8.21

	preg	Glucose	pres	skin	insu	mass	pedi	age	class	Glucose_index
							7		d_positive	4008
1	1.0	85	66.0	29.0	0	26.6	0.351	31	test_negative	4.717505
2	8.0	183	64.0	0.0	0	23.3	0.672	32	test_positive	10.156510
3	1.0	89	66.0	23.0	94	28.1	0.167	21	test_negative	4.939505
4	0.0	137	40.0	35.0	168	43.1	2.288	33	test_positive	7.603508
5	5.0	116	74.0	0.0	0	25.6	0.201	30	test_negative	6.438006
6	3.0	78	50.0	32.0	88	31.0	0.248	26	test_positive	4.329004
7	10.0	115	0.0	0.0	0	35.3	0.134	29	test_negative	6.382506
8	2.0	197	70.0	45.0	543	30.5	0.158	53	test_positive	10.933511
9	8.0	125	96.0	0.0	0	0.0	0.232	54	test_positive	6.937507

		Gluc				mas				Gluc
	preg	ose	pres	skin	insu	s	pedi	age	class	ose_i
										n_m
										ol
10	4.0	110	92.0	0.0	0	37.6	0.19 1	30	teste d_ne gati ve	6.10 500 6
11	10.0	168	74.0	0.0	0	38.0	0.53 7	34	teste d_po sitiv e	9.32 400 9

filter and groups

In [57]:

```
fil_age_30less=diab_df[diab_df["age"]<30]
fil_age_30less.head(10)
```

Out[57]:

		Gluc				mas				Gluc
	preg	ose	pres	skin	insu	s	pedi	age	class	ose_i
										n_m
										ol
3	1.0	89	66.0	23.0	94	28.1	0.16 7	21	teste d_ne gati ve	4.93 950 5
6	3.0	78	50.0	32.0	88	31.0	0.24 8	26	teste d_po sitiv e	4.32 900 4
7	10.0	115	0.0	0.0	0	35.3	0.13 4	29	teste d_ne gati ve	6.38 250 6
20	3.0	126	88.0	41.0	235	39.3	0.70 4	27	teste d_ne gati ve	6.99 300 7
23	9.0	119	80.0	35.0	0	29.0	0.26 3	29	teste d_po	6.60 450

		Gluc				mas				Gluc
	preg	ose	pres	skin	insu	s	pedi	age	class	ose_i
										n_m
										ol
									sitiv	7
									e	
27	1.0	97	66.0	15.0	140	23.2	0.48	22	teste	5.38
							7		d_ne	350
									gati	5
									ve	
31	3.0	158	76.0	36.0	245	31.6	0.85	28	teste	8.76
							1		d_po	900
									sitiv	9
									e	
32	3.0	88	58.0	11.0	54	24.8	0.26	22	teste	4.88
							7		d_ne	400
									gati	5
									ve	
33	6.0	92	92.0	0.0	0	19.9	0.18	28	teste	5.10
							8		d_ne	600
									gati	5
									ve	
40	3.0	180	64.0	25.0	70	34.0	0.27	26	teste	9.99
							1		d_ne	001
									gati	0
									ve	

In [59]:

```
fil_age_100less=diab_df[diab_df["age"]<100]
fil_age_100less.head(10)
```

Out[59]:

		Gluc				mas				Gluc
	preg	ose	pres	skin	insu	s	pedi	age	class	ose_i
										n_m
										ol
0	6.0	148	72.0	35.0	0	33.6	0.62	50	teste	8.21
							7		d_po	400
									sitiv	8
									e	
1	1.0	85	66.0	29.0	0	26.6	0.35	31	teste	4.71
							1		d_ne	750

	preg	Glucose	pres	skin	insu	mas	pedi	age	class	Glucose_index
									gative	5
2	8.0	183	64.0	0.0	0	23.3	0.672	32	tested_positive	10.156510
3	1.0	89	66.0	23.0	94	28.1	0.167	21	tested_negative	4.939505
4	0.0	137	40.0	35.0	168	43.1	2.288	33	tested_positive	7.603508
5	5.0	116	74.0	0.0	0	25.6	0.201	30	tested_negative	6.438006
6	3.0	78	50.0	32.0	88	31.0	0.248	26	tested_positive	4.329004
7	10.0	115	0.0	0.0	0	35.3	0.134	29	tested_negative	6.382506
8	2.0	197	70.0	45.0	543	30.5	0.158	53	tested_positive	10.933511
9	8.0	125	96.0	0.0	0	0.0	0.232	54	tested_positive	6.937507

In [61]:

```
glu_100=diab_df[diab_df["age"]>100]
glu_100.head(10)
```

preg	Glucose_i	pres	skin	insu	mas	pedi	age	class	Glucose_i n_mol
------	-----------	------	------	------	-----	------	-----	-------	--------------------

In []:

In [69]:

Out[69]:

In [71]:

Out[71]:

In [73]:

Out[73]:

In [77]:

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

Out[77]:

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

CLEANING DATA

HANDLING DATA

In [83]:

```
diab_df.isnull().sum()
```

Out[83]:

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

In [86]:

```
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_mol	102 non-null	float64

```
dtypes: float64(6), int64(3), object(1)
memory usage: 8.1+ KB
```

In [90]:

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

In [92]:

```
diab_df.isnull().sum()
```

Out[92]:

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

HANDLING DUPLICATES

In [95]:

```
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_mol       98 non-null    float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

In [97]:

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

In [99]:

```
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_mol	96 non-null	float64

```
dtypes: float64(6), int64(3), object(1)
```

```
memory usage: 8.2+ KB
```

READING OTHER FORMATS

In [108]:

```
dia_ex=pd.read_excel(r"C:\Users\DELL\Downloads\diabetes.xlsx")
```

In [110]:

```
dia_ex.head()
```

Out[110]:

	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

	preg	plas	pres	skin	insu	mass	pedi	age	class
4	0	137	40	35	168	43.1	2.28 8	33	teste d_po sitive

In [112]:

```
dia_ex_sheet2=pd.read_excel(r"C:\Users\DELL\Downloads\
diabetes.xlsx",sheet_name="dora")
```

In [114]:

```
dia_ex_sheet2.head()
```

Out[114]:

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

loading text file

In [121]:

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

Out[121]:

	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

In [124]:

```
df_text['SEM1_INT']=df_text['SEM1'].astype(int)
```

In [126]:

```
df_text.head()
```

Out[126]:

	Names	Initials	SEM1	SEM2	SEM3	Grade	SEM1_I NT
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

In [11]:

```
X=[1,2,3,4,5]
```

```
Y=[20,30,40,50,55]
```

In [15]:

```
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()
```

Out[15]:

<matplotlib.legend.Legend at 0x1bbc4a87610>

No description has been provided for this image

In [27]:

```
sub=['ADA','AJ','BIO','GT','DBMS']
```

```
deepu=[85,92,90,91,95]
```

```
amith=[90,95,95,92,92]
```

```
plt.scatter(sub,deepu,color='green',label="DEEPU SCORE",marker='*')
```

```
plt.scatter(sub,amith,color='black',label="AMITH SCORE",marker='^')
```

```
plt.xlabel("subject")
```

```
plt.ylabel("score")
```

```
plt.legend()
```


Out[27]:

<matplotlib.legend.Legend at 0x1bbc4b198d0>

No description has been provided for this image

In [39]:

```
sub=['ADA','AJ','BIO','GT','DBMS']
deepu=[87,98,95,87,88]
amith=[87,99,97,98,99]
plt.subplot(1,2,1)
plt.bar(sub,deepu,color='red',label="DEEPU SCORE")
plt.xlabel("subject")
plt.ylabel("score")
plt.legend()

plt.subplot(1,2,1)
plt.bar(sub,amith,color='black',label="AMITH SCORE")
plt.xlabel("subject")
plt.ylabel("score")
plt.legend()
```

Out[39]:

<matplotlib.legend.Legend at 0x1bbc64acad0>

No description has been provided for this image

In [43]:

```
sub=['ADA','AJ','BIO','GT','DBMS']
deepu=[87,98,95,87,88]
amith=[87,99,97,98,99]
plt.bar(sub,deepu,color='green',label="DEEPU SCORE",width=0.5,align="center")
plt.bar(sub,amith,color='black',label="AMITH SCORE",width=0.5,align="edge")
plt.xlabel("subject")
plt.ylabel("score")
plt.legend()
```

Out[43]:

<matplotlib.legend.Legend at 0x1bbc6786350>

No description has been provided for this image

In [45]:

```
import numpy as np
```

In [49]:

```
a=np.array([25,60,9,10])
labe=["ATML","PYTHON","PANDAS","NUMPY"]
color=['black','green','red','yellow']
plt.pie(a,labels=labe,colors=color)
plt.legend()
plt.show()
```

No description has been provided for this image

In []: