Assignment-2

1 Assignment No:2

#1. Import all the required Python Libraries.

[ ]:
```
import pandas as pd
```

[ ]:
```
import numpy as np
```

#2. Creation of Dataset using Microsoft Excel.

[ ]:
```
!pip install -q kaggle
```

[ ]:
```
from google.colab import files
```

[ ]:
```
files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving stud_academics_record .csv to stud_academics_record .csv
```

[ ]:
```
{'stud_academics_record .csv': b'Roll no,Name,Term,Attendance,Sub1,Sub2,Sub3,Sub
4,Sub5,Total_Marks,Percentage,Result\r\n1,A,A,75,80,85,77,96,66,479,95.8,Pass\r\
n2,B,A,65,60,NaN,88,41,NaN,361,72.2,Fail\r\n3,C,A,55,77,41,99,52,22,346,69.2,Fai
l\r\n4,D,A,65,88,52,44,63,33,345,69,Pass\r\n5,E,A,45,99,63,55,78,74,414,82.8,Pas
s\r\n6,F,A,96,44,78,66,89,85,458,91.6,Pass\r\n7,G,A,96,55,89,11,45,96,392,78.4,F
ail\r\n8,H,A,85,66,45,22,60,41,319,63.8,Pass\r\n9,I,A,75,11,60,96,77,52,371,74.2
,Fail\r\n10,J,A,65,22,77,41,88,63,356,71.2,Pass\r\n11,K,A,39,33,88,52,99,66,377,
75.4,Pass\r\n12,L,A,45,74,99,63,96,NaN,388,77.6,Fail\r\n13,M,A,78,85,44,78,41,22
,348,69.6,Fail\r\n14,N,A,99,96,55,NaN,52,96,487,97.4,Pass\r\n15,O,A,10,41,66,45,
63,41,266,53.2,Fail\r\n16,P,A,66,52,66,60,78,52,374,74.8,Pass\r\n17,Q,A,44,63,11
,77,89,63,347,69.4,Fail\r\n18,R,A,55,78,22,88,NaN,78,366,73.2,Fail\r\n19,S,A,77,
89,33,99,60,89,447,89.4,Pass\r\n20,T,A,88,45,74,44,77,45,373,74.6,Pass\r\n21,,B,
85,99,NaN,11,99,96,390,78,Pass\r\n22,,B,75,44,22,22,44,41,248,49.6,Pass\r\n23,,B
,65,55,96,96,55,52,419,83.8,Pass\r\n24,,B,39,66,41,41,66,63,316,63.2,Pass\r\n25,
,B,45,66,52,52,66,78,359,71.8,Pass\r\n26,,B,85,11,63,63,11,89,322,64.4,Pass\r\n2
7,,B,75,22,78,78,22,45,320,64,Pass\r\n28,,B,65,33,89,NaN,33,2,222,44.4,Pass\r\n2
9,,B,39,74,45,2,74,44,278,55.6,Pass\r\n30,,B,45,88,10,3,45,55,246,49.2,Pass\r\n'
```



```
}
```

#3.Load the Dataset into pandas dataframe.

[ ]:
```
df=pd.read_csv("stud_academics_record .csv")
```

[ ]:
```
df
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	Total_Marks	\
0	1	A	A	75	80	85.0	77.0	96.0	66.0	479	
1	2	B	A	65	60	NaN	88.0	41.0	NaN	361	
2	3	C	A	55	77	41.0	99.0	52.0	22.0	346	
3	4	D	A	65	88	52.0	44.0	63.0	33.0	345	
4	5	E	A	45	99	63.0	55.0	78.0	74.0	414	
5	6	F	A	96	44	78.0	66.0	89.0	85.0	458	
6	7	G	A	96	55	89.0	11.0	45.0	96.0	392	
7	8	H	A	85	66	45.0	22.0	60.0	41.0	319	
8	9	I	A	75	11	60.0	96.0	77.0	52.0	371	
9	10	J	A	65	22	77.0	41.0	88.0	63.0	356	
10	11	K	A	39	33	88.0	52.0	99.0	66.0	377	
11	12	L	A	45	74	99.0	63.0	96.0	NaN	388	
12	13	M	A	78	85	44.0	78.0	41.0	22.0	348	
13	14	N	A	99	96	55.0	NaN	52.0	96.0	487	
14	15	O	A	10	41	66.0	45.0	63.0	41.0	266	
15	16	P	A	66	52	66.0	60.0	78.0	52.0	374	
16	17	Q	A	44	63	11.0	77.0	89.0	63.0	347	
17	18	R	A	55	78	22.0	88.0	NaN	78.0	366	
18	19	S	A	77	89	33.0	99.0	60.0	89.0	447	
19	20	T	A	88	45	74.0	44.0	77.0	45.0	373	
20	21	NaN	B	85	99	NaN	11.0	99.0	96.0	390	
21	22	NaN	B	75	44	22.0	22.0	44.0	41.0	248	
22	23	NaN	B	65	55	96.0	96.0	55.0	52.0	419	
23	24	NaN	B	39	66	41.0	41.0	66.0	63.0	316	
24	25	NaN	B	45	66	52.0	52.0	66.0	78.0	359	
25	26	NaN	B	85	11	63.0	63.0	11.0	89.0	322	
26	27	NaN	B	75	22	78.0	78.0	22.0	45.0	320	
27	28	NaN	B	65	33	89.0	NaN	33.0	2.0	222	
28	29	NaN	B	39	74	45.0	2.0	74.0	44.0	278	
29	30	NaN	B	45	88	10.0	3.0	45.0	55.0	246	

	Percentage	Result
0	95.8	Pass
1	72.2	Fail
2	69.2	Fail
3	69.0	Pass
4	82.8	Pass
5	91.6	Pass

```
6         78.4   Fail
7         63.8   Pass
8         74.2   Fail
9         71.2   Pass
10        75.4   Pass
11        77.6   Fail
12        69.6   Fail
13        97.4   Pass
14        53.2   Fail
15        74.8   Pass
16        69.4   Fail
17        73.2   Fail
18        89.4   Pass
19        74.6   Pass
20        78.0   Pass
21        49.6   Pass
22        83.8   Pass
23        63.2   Pass
24        71.8   Pass
25        64.4   Pass
26        64.0   Pass
27        44.4   Pass
28        55.6   Pass
29        49.2   Pass
```

#4. Data Preprocessing:

[ ]:
```
df.head()
```

[ ]:
```
   Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
0        1    A    A          75    80  85.0  77.0  96.0  66.0          479
1        2    B    A          65    60   NaN  88.0  41.0   NaN          361
2        3    C    A          55    77  41.0  99.0  52.0  22.0          346
3        4    D    A          65    88  52.0  44.0  63.0  33.0          345
4        5    E    A          45    99  63.0  55.0  78.0  74.0          414

   Percentage Result
0        95.8   Pass
1        72.2   Fail
2        69.2   Fail
3        69.0   Pass
4        82.8   Pass
```

[ ]:
```
df.tail()
```

[ ]:
```
    Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
25       26  NaN    B          85    11  63.0  63.0  11.0  89.0          322
```



```
26       27  NaN    B          75    22  78.0  78.0  22.0  45.0          320
27       28  NaN    B          65    33  89.0   NaN  33.0   2.0          222
28       29  NaN    B          39    74  45.0   2.0  74.0  44.0          278
29       30  NaN    B          45    88  10.0   3.0  45.0  55.0          246

    Percentage Result
25        64.4   Pass
26        64.0   Pass
27        44.4   Pass
28        55.6   Pass
29        49.2   Pass
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Roll no      30 non-null     int64
 1   Name         20 non-null     object
 2   Term         30 non-null     object
 3   Attendance   30 non-null     int64
 4   Sub1         30 non-null     int64
 5   Sub2         28 non-null     float64
 6   Sub3         28 non-null     float64
 7   Sub4         29 non-null     float64
 8   Sub5         28 non-null     float64
 9   Total_Marks  30 non-null     int64
 10  Percentage   30 non-null     float64
 11  Result       30 non-null     object
dtypes: float64(5), int64(4), object(3)
memory usage: 2.9+ KB
```

[ ]:
```
df.describe(include="all")
```

[ ]:
```
          Roll no Name Term  Attendance       Sub1       Sub2       Sub3  \
count   30.000000   20   30   30.000000  30.000000  28.000000  28.000000
unique        NaN   20    2         NaN        NaN        NaN        NaN
top           NaN    A    A         NaN        NaN        NaN        NaN
freq          NaN    1   20         NaN        NaN        NaN        NaN
mean    15.500000  NaN  NaN   64.700000  60.533333  58.714286  56.178571
std      8.803408  NaN  NaN   21.019121  25.761149  25.227641  29.806852
min      1.000000  NaN  NaN   10.000000  11.000000  10.000000   2.000000
25%      8.250000  NaN  NaN   45.000000  44.000000  43.250000  41.000000
50%     15.500000  NaN  NaN   65.000000  64.500000  61.500000  57.500000
75%     22.750000  NaN  NaN   77.750000  79.500000  78.000000  78.000000
```

```
max     30.000000  NaN  NaN   99.000000  99.000000  99.000000  99.000000

             Sub4       Sub5  Total_Marks  Percentage Result
count   29.000000  28.000000    30.000000   30.000000     30
unique        NaN        NaN          NaN         NaN      2
top           NaN        NaN          NaN         NaN   Pass
freq          NaN        NaN          NaN         NaN     21
mean    64.103448  58.892857   357.800000   71.560000    NaN
std     23.359527  24.412109    65.598255   13.119651    NaN
min     11.000000   2.000000   222.000000   44.400000    NaN
25%     45.000000  43.250000   320.500000   64.100000    NaN
50%     63.000000  59.000000   360.000000   72.000000    NaN
75%     78.000000  78.000000   389.500000   77.900000    NaN
max     99.000000  96.000000   487.000000   97.400000    NaN
```

[ ]:
```
df.shape
```

[ ]:
```
(30, 12)
```

[ ]:
```
df.dtypes
```

[ ]:
```
Roll no          int64
Name            object
Term            object
Attendance       int64
Sub1             int64
Sub2           float64
Sub3           float64
Sub4           float64
Sub5           float64
Total_Marks      int64
Percentage     float64
Result          object
dtype: object
```

[ ]:
```
df.columns
```

[ ]:
```
Index(['Roll no', 'Name', 'Term', 'Attendance', 'Sub1', 'Sub2', 'Sub3', 'Sub4',
       'Sub5', 'Total_Marks', 'Percentage', 'Result'],
      dtype='object')
```

[ ]:
```
df[0:4]
```

[ ]:
```
   Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
0        1    A    A          75    80  85.0  77.0  96.0  66.0          479
1        2    B    A          65    60   NaN  88.0  41.0   NaN          361
2        3    C    A          55    77  41.0  99.0  52.0  22.0          346
```



```
3        4    D    A          65    88  52.0  44.0  63.0  33.0          345

   Percentage Result
0        95.8   Pass
1        72.2   Fail
2        69.2   Fail
3        69.0   Pass
```

[ ]:
```
df.loc[0:2]
```

[ ]:
```
   Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
0        1    A    A          75    80  85.0  77.0  96.0  66.0          479
1        2    B    A          65    60   NaN  88.0  41.0   NaN          361
2        3    C    A          55    77  41.0  99.0  52.0  22.0          346

   Percentage Result
0        95.8   Pass
1        72.2   Fail
2        69.2   Fail
```

[ ]:
```
df.loc[0:2,'Sub1':'Sub2']
```

[ ]:
```
   Sub1  Sub2
0    80  85.0
1    60   NaN
2    77  41.0
```

[ ]:
```
df.iloc[1:3]
```

[ ]:
```
   Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
1        2    B    A          65    60   NaN  88.0  41.0   NaN          361
2        3    C    A          55    77  41.0  99.0  52.0  22.0          346

   Percentage Result
1        72.2   Fail
2        69.2   Fail
```

[ ]:
```
df.iloc[1:5,1:5]
```

[ ]:
```
  Name Term  Attendance  Sub1
1    B    A          65    60
2    C    A          55    77
3    D    A          65    88
4    E    A          45    99
```

#A. Identification and Handling of Null Values

Check for missing values in the data using pandas isnull()

[ ]:
```
df.isnull()
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	\
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	True	False	False	True	
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	
5	False	False	False	False	False	False	False	False	False	
6	False	False	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	False	False	
8	False	False	False	False	False	False	False	False	False	
9	False	False	False	False	False	False	False	False	False	
10	False	False	False	False	False	False	False	False	False	
11	False	False	False	False	False	False	False	False	True	
12	False	False	False	False	False	False	False	False	False	
13	False	False	False	False	False	False	True	False	False	
14	False	False	False	False	False	False	False	False	False	
15	False	False	False	False	False	False	False	False	False	
16	False	False	False	False	False	False	False	False	False	
17	False	False	False	False	False	False	False	True	False	
18	False	False	False	False	False	False	False	False	False	
19	False	False	False	False	False	False	False	False	False	
20	False	True	False	False	False	True	False	False	False	
21	False	True	False	False	False	False	False	False	False	
22	False	True	False	False	False	False	False	False	False	
23	False	True	False	False	False	False	False	False	False	
24	False	True	False	False	False	False	False	False	False	
25	False	True	False	False	False	False	False	False	False	
26	False	True	False	False	False	False	False	False	False	
27	False	True	False	False	False	False	True	False	False	
28	False	True	False	False	False	False	False	False	False	
29	False	True	False	False	False	False	False	False	False	

	Total_Marks	Percentage	Result
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False
8	False	False	False
9	False	False	False
10	False	False	False
11	False	False	False

12	False	False	False
13	False	False	False
14	False	False	False
15	False	False	False
16	False	False	False
17	False	False	False
18	False	False	False
19	False	False	False
20	False	False	False
21	False	False	False
22	False	False	False
23	False	False	False
24	False	False	False
25	False	False	False
26	False	False	False
27	False	False	False
28	False	False	False
29	False	False	False

[ ]:
```
df.isna()
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	\
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	True	False	False	True	
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	
5	False	False	False	False	False	False	False	False	False	
6	False	False	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	False	False	
8	False	False	False	False	False	False	False	False	False	
9	False	False	False	False	False	False	False	False	False	
10	False	False	False	False	False	False	False	False	False	
11	False	False	False	False	False	False	False	False	True	
12	False	False	False	False	False	False	False	False	False	
13	False	False	False	False	False	False	False	True	False	
14	False	False	False	False	False	False	False	False	False	
15	False	False	False	False	False	False	False	False	False	
16	False	False	False	False	False	False	False	False	False	
17	False	False	False	False	False	False	False	True	False	
18	False	False	False	False	False	False	False	False	False	
19	False	False	False	False	False	False	False	False	False	
20	False	True	False	False	False	True	False	False	False	
21	False	True	False	False	False	False	False	False	False	
22	False	True	False	False	False	False	False	False	False	
23	False	True	False	False	False	False	False	False	False	
24	False	True	False	False	False	False	False	False	False	

```
25    False   True  False       False  False  False  False  False  False
26    False   True  False       False  False  False  False  False  False
27    False   True  False       False  False  False   True  False  False
28    False   True  False       False  False  False  False  False  False
29    False   True  False       False  False  False  False  False  False

    Total_Marks  Percentage  Result
0         False       False   False
1         False       False   False
2         False       False   False
3         False       False   False
4         False       False   False
5         False       False   False
6         False       False   False
7         False       False   False
8         False       False   False
9         False       False   False
10        False       False   False
11        False       False   False
12        False       False   False
13        False       False   False
14        False       False   False
15        False       False   False
16        False       False   False
17        False       False   False
18        False       False   False
19        False       False   False
20        False       False   False
21        False       False   False
22        False       False   False
23        False       False   False
24        False       False   False
25        False       False   False
26        False       False   False
27        False       False   False
28        False       False   False
29        False       False   False
```

[ ]:
```
df.isnull().any()
```

[ ]:
```
Roll no        False
Name            True
Term           False
Attendance     False
Sub1           False
Sub2            True
Sub3            True
```



```
Sub4            True
Sub5            True
Total_Marks    False
Percentage     False
Result         False
dtype: bool
```

[ ]:
```
df.isnull().sum()
```

[ ]:
```
Roll no         0
Name           10
Term            0
Attendance      0
Sub1            0
Sub2            2
Sub3            2
Sub4            1
Sub5            2
Total_Marks     0
Percentage      0
Result          0
dtype: int64
```

[ ]:
```
df.Attendance.isnull().sum()
```

[ ]:
```
0
```

Make a list of column having missing value

[ ]:
```
cols_with_na = []
for col in df.columns:
  if df[col].isna().any():
    cols_with_na.append(col)
cols_with_na
```

[ ]:
```
['Name', 'Sub2', 'Sub3', 'Sub4', 'Sub5']
```

Filling missing values using dropna(), fillna(), replace() :

1. replacing null values with NaN

[ ]:
```
df.replace(np.nan,value=0)
```

[ ]:
```
    Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
0         1    A    A          75    80  85.0  77.0  96.0  66.0          479
1         2    B    A          65    60   0.0  88.0  41.0   0.0          361
2         3    C    A          55    77  41.0  99.0  52.0  22.0          346
3         4    D    A          65    88  52.0  44.0  63.0  33.0          345
```

4	5	E	A	45	99	63.0	55.0	78.0	74.0	414
5	6	F	A	96	44	78.0	66.0	89.0	85.0	458
6	7	G	A	96	55	89.0	11.0	45.0	96.0	392
7	8	H	A	85	66	45.0	22.0	60.0	41.0	319
8	9	I	A	75	11	60.0	96.0	77.0	52.0	371
9	10	J	A	65	22	77.0	41.0	88.0	63.0	356
10	11	K	A	39	33	88.0	52.0	99.0	66.0	377
11	12	L	A	45	74	99.0	63.0	96.0	0.0	388
12	13	M	A	78	85	44.0	78.0	41.0	22.0	348
13	14	N	A	99	96	55.0	0.0	52.0	96.0	487
14	15	O	A	10	41	66.0	45.0	63.0	41.0	266
15	16	P	A	66	52	66.0	60.0	78.0	52.0	374
16	17	Q	A	44	63	11.0	77.0	89.0	63.0	347
17	18	R	A	55	78	22.0	88.0	0.0	78.0	366
18	19	S	A	77	89	33.0	99.0	60.0	89.0	447
19	20	T	A	88	45	74.0	44.0	77.0	45.0	373
20	21	0	B	85	99	0.0	11.0	99.0	96.0	390
21	22	0	B	75	44	22.0	22.0	44.0	41.0	248
22	23	0	B	65	55	96.0	96.0	55.0	52.0	419
23	24	0	B	39	66	41.0	41.0	66.0	63.0	316
24	25	0	B	45	66	52.0	52.0	66.0	78.0	359
25	26	0	B	85	11	63.0	63.0	11.0	89.0	322
26	27	0	B	75	22	78.0	78.0	22.0	45.0	320
27	28	0	B	65	33	89.0	0.0	33.0	2.0	222
28	29	0	B	39	74	45.0	2.0	74.0	44.0	278
29	30	0	B	45	88	10.0	3.0	45.0	55.0	246

	Percentage	Result
0	95.8	Pass
1	72.2	Fail
2	69.2	Fail
3	69.0	Pass
4	82.8	Pass
5	91.6	Pass
6	78.4	Fail
7	63.8	Pass
8	74.2	Fail
9	71.2	Pass
10	75.4	Pass
11	77.6	Fail
12	69.6	Fail
13	97.4	Pass
14	53.2	Fail
15	74.8	Pass
16	69.4	Fail
17	73.2	Fail
18	89.4	Pass
19	74.6	Pass
20	78.0	Pass
21	49.6	Pass
22	83.8	Pass
23	63.2	Pass
24	71.8	Pass
25	64.4	Pass
26	64.0	Pass
27	44.4	Pass
28	55.6	Pass
29	49.2	Pass

2. Filling null values with fillna()

[ ]:
```
df.fillna(1)
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	Total_Marks	\
0	1	A	A	75	80	85.0	77.0	96.0	66.0	479	
1	2	B	A	65	60	1.0	88.0	41.0	1.0	361	
2	3	C	A	55	77	41.0	99.0	52.0	22.0	346	
3	4	D	A	65	88	52.0	44.0	63.0	33.0	345	
4	5	E	A	45	99	63.0	55.0	78.0	74.0	414	
5	6	F	A	96	44	78.0	66.0	89.0	85.0	458	
6	7	G	A	96	55	89.0	11.0	45.0	96.0	392	
7	8	H	A	85	66	45.0	22.0	60.0	41.0	319	
8	9	I	A	75	11	60.0	96.0	77.0	52.0	371	
9	10	J	A	65	22	77.0	41.0	88.0	63.0	356	
10	11	K	A	39	33	88.0	52.0	99.0	66.0	377	
11	12	L	A	45	74	99.0	63.0	96.0	1.0	388	
12	13	M	A	78	85	44.0	78.0	41.0	22.0	348	
13	14	N	A	99	96	55.0	1.0	52.0	96.0	487	
14	15	O	A	10	41	66.0	45.0	63.0	41.0	266	
15	16	P	A	66	52	66.0	60.0	78.0	52.0	374	
16	17	Q	A	44	63	11.0	77.0	89.0	63.0	347	
17	18	R	A	55	78	22.0	88.0	1.0	78.0	366	
18	19	S	A	77	89	33.0	99.0	60.0	89.0	447	
19	20	T	A	88	45	74.0	44.0	77.0	45.0	373	
20	21	1	B	85	99	1.0	11.0	99.0	96.0	390	
21	22	1	B	75	44	22.0	22.0	44.0	41.0	248	
22	23	1	B	65	55	96.0	96.0	55.0	52.0	419	
23	24	1	B	39	66	41.0	41.0	66.0	63.0	316	
24	25	1	B	45	66	52.0	52.0	66.0	78.0	359	
25	26	1	B	85	11	63.0	63.0	11.0	89.0	322	
26	27	1	B	75	22	78.0	78.0	22.0	45.0	320	
27	28	1	B	65	33	89.0	1.0	33.0	2.0	222	
28	29	1	B	39	74	45.0	2.0	74.0	44.0	278	
29	30	1	B	45	88	10.0	3.0	45.0	55.0	246	

	Percentage	Result
0	95.8	Pass
1	72.2	Fail
2	69.2	Fail
3	69.0	Pass
4	82.8	Pass
5	91.6	Pass
6	78.4	Fail
7	63.8	Pass
8	74.2	Fail
9	71.2	Pass
10	75.4	Pass
11	77.6	Fail
12	69.6	Fail
13	97.4	Pass
14	53.2	Fail
15	74.8	Pass
16	69.4	Fail
17	73.2	Fail
18	89.4	Pass
19	74.6	Pass
20	78.0	Pass
21	49.6	Pass
22	83.8	Pass
23	63.2	Pass
24	71.8	Pass
25	64.4	Pass
26	64.0	Pass
27	44.4	Pass
28	55.6	Pass
29	49.2	Pass

3. filling missing values using mean, median,max, min and standard deviation of that column

[ ]:
```
df['Sub4']=df['Sub4'].fillna(df['Sub4'].mean())
```

[ ]:
```
df
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	\
0	1	A	A	75	80	85.0	77.0	96.000000	66.0	
1	2	B	A	65	60	NaN	88.0	41.000000	NaN	
2	3	C	A	55	77	41.0	99.0	52.000000	22.0	
3	4	D	A	65	88	52.0	44.0	63.000000	33.0	
4	5	E	A	45	99	63.0	55.0	78.000000	74.0	
5	6	F	A	96	44	78.0	66.0	89.000000	85.0	
6	7	G	A	96	55	89.0	11.0	45.000000	96.0	

7	8	H	A	85	66	45.0	22.0	60.000000	41.0
8	9	I	A	75	11	60.0	96.0	77.000000	52.0
9	10	J	A	65	22	77.0	41.0	88.000000	63.0
10	11	K	A	39	33	88.0	52.0	99.000000	66.0
11	12	L	A	45	74	99.0	63.0	96.000000	NaN
12	13	M	A	78	85	44.0	78.0	41.000000	22.0
13	14	N	A	99	96	55.0	NaN	52.000000	96.0
14	15	O	A	10	41	66.0	45.0	63.000000	41.0
15	16	P	A	66	52	66.0	60.0	78.000000	52.0
16	17	Q	A	44	63	11.0	77.0	89.000000	63.0
17	18	R	A	55	78	22.0	88.0	64.103448	78.0
18	19	S	A	77	89	33.0	99.0	60.000000	89.0
19	20	T	A	88	45	74.0	44.0	77.000000	45.0
20	21	NaN	B	85	99	NaN	11.0	99.000000	96.0
21	22	NaN	B	75	44	22.0	22.0	44.000000	41.0
22	23	NaN	B	65	55	96.0	96.0	55.000000	52.0
23	24	NaN	B	39	66	41.0	41.0	66.000000	63.0
24	25	NaN	B	45	66	52.0	52.0	66.000000	78.0
25	26	NaN	B	85	11	63.0	63.0	11.000000	89.0
26	27	NaN	B	75	22	78.0	78.0	22.000000	45.0
27	28	NaN	B	65	33	89.0	NaN	33.000000	2.0
28	29	NaN	B	39	74	45.0	2.0	74.000000	44.0
29	30	NaN	B	45	88	10.0	3.0	45.000000	55.0

	Total_Marks	Percentage	Result
0	479	95.8	Pass
1	361	72.2	Fail
2	346	69.2	Fail
3	345	69.0	Pass
4	414	82.8	Pass
5	458	91.6	Pass
6	392	78.4	Fail
7	319	63.8	Pass
8	371	74.2	Fail
9	356	71.2	Pass
10	377	75.4	Pass
11	388	77.6	Fail
12	348	69.6	Fail
13	487	97.4	Pass
14	266	53.2	Fail
15	374	74.8	Pass
16	347	69.4	Fail
17	366	73.2	Fail
18	447	89.4	Pass
19	373	74.6	Pass
20	390	78.0	Pass
21	248	49.6	Pass

```
22          419        83.8   Pass
23          316        63.2   Pass
24          359        71.8   Pass
25          322        64.4   Pass
26          320        64.0   Pass
27          222        44.4   Pass
28          278        55.6   Pass
29          246        49.2   Pass
```

[ ]:
```
df.head(10)
```

[ ]:
```
   Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
0        1    A    A          75    80  85.0  77.0  96.0  66.0          479
1        2    B    A          65    60   NaN  88.0  41.0   NaN          361
2        3    C    A          55    77  41.0  99.0  52.0  22.0          346
3        4    D    A          65    88  52.0  44.0  63.0  33.0          345
4        5    E    A          45    99  63.0  55.0  78.0  74.0          414
5        6    F    A          96    44  78.0  66.0  89.0  85.0          458
6        7    G    A          96    55  89.0  11.0  45.0  96.0          392
7        8    H    A          85    66  45.0  22.0  60.0  41.0          319
8        9    I    A          75    11  60.0  96.0  77.0  52.0          371
9       10    J    A          65    22  77.0  41.0  88.0  63.0          356

   Percentage Result
0        95.8   Pass
1        72.2   Fail
2        69.2   Fail
3        69.0   Pass
4        82.8   Pass
5        91.6   Pass
6        78.4   Fail
7        63.8   Pass
8        74.2   Fail
9        71.2   Pass
```

4.Deleting null values using dropna() method

In order to drop null values from a dataframe, dropna() function is used. This function drops Rows/Columns of datasets with Null values in different ways. 1. Dropping rows with at least 1 null value 2. Dropping rows if all values in that row are missing

[ ]:
```
df.dropna() #Dropping rows with at least 1 null value
```

[ ]:
```
    Roll no Name Term  Attendance  Sub1  Sub2  Sub3       Sub4  Sub5  \
0        1    A    A          75    80  85.0  77.0  96.000000  66.0
2        3    C    A          55    77  41.0  99.0  52.000000  22.0
3        4    D    A          65    88  52.0  44.0  63.000000  33.0
4        5    E    A          45    99  63.0  55.0  78.000000  74.0
```



```
5        6    F    A          96    44  78.0  66.0  89.000000  85.0
6        7    G    A          96    55  89.0  11.0  45.000000  96.0
7        8    H    A          85    66  45.0  22.0  60.000000  41.0
8        9    I    A          75    11  60.0  96.0  77.000000  52.0
9       10    J    A          65    22  77.0  41.0  88.000000  63.0
10      11    K    A          39    33  88.0  52.0  99.000000  66.0
12      13    M    A          78    85  44.0  78.0  41.000000  22.0
14      15    O    A          10    41  66.0  45.0  63.000000  41.0
15      16    P    A          66    52  66.0  60.0  78.000000  52.0
16      17    Q    A          44    63  11.0  77.0  89.000000  63.0
17      18    R    A          55    78  22.0  88.0  64.103448  78.0
18      19    S    A          77    89  33.0  99.0  60.000000  89.0
19      20    T    A          88    45  74.0  44.0  77.000000  45.0

    Total_Marks  Percentage Result
0           479        95.8   Pass
2           346        69.2   Fail
3           345        69.0   Pass
4           414        82.8   Pass
5           458        91.6   Pass
6           392        78.4   Fail
7           319        63.8   Pass
8           371        74.2   Fail
9           356        71.2   Pass
10          377        75.4   Pass
12          348        69.6   Fail
14          266        53.2   Fail
15          374        74.8   Pass
16          347        69.4   Fail
17          366        73.2   Fail
18          447        89.4   Pass
19          373        74.6   Pass
```

[ ]:
```
df.dropna(how="all") #Dropping rows if all values in that row are missing
```

[ ]:
```
    Roll no Name Term  Attendance  Sub1  Sub2  Sub3       Sub4  Sub5  \
0        1    A    A          75    80  85.0  77.0  96.000000  66.0
1        2    B    A          65    60   NaN  88.0  41.000000   NaN
2        3    C    A          55    77  41.0  99.0  52.000000  22.0
3        4    D    A          65    88  52.0  44.0  63.000000  33.0
4        5    E    A          45    99  63.0  55.0  78.000000  74.0
5        6    F    A          96    44  78.0  66.0  89.000000  85.0
6        7    G    A          96    55  89.0  11.0  45.000000  96.0
7        8    H    A          85    66  45.0  22.0  60.000000  41.0
8        9    I    A          75    11  60.0  96.0  77.000000  52.0
9       10    J    A          65    22  77.0  41.0  88.000000  63.0
10      11    K    A          39    33  88.0  52.0  99.000000  66.0
```

11	12	L	A	45	74	99.0	63.0	96.000000	NaN
12	13	M	A	78	85	44.0	78.0	41.000000	22.0
13	14	N	A	99	96	55.0	NaN	52.000000	96.0
14	15	O	A	10	41	66.0	45.0	63.000000	41.0
15	16	P	A	66	52	66.0	60.0	78.000000	52.0
16	17	Q	A	44	63	11.0	77.0	89.000000	63.0
17	18	R	A	55	78	22.0	88.0	64.103448	78.0
18	19	S	A	77	89	33.0	99.0	60.000000	89.0
19	20	T	A	88	45	74.0	44.0	77.000000	45.0
20	21	NaN	B	85	99	NaN	11.0	99.000000	96.0
21	22	NaN	B	75	44	22.0	22.0	44.000000	41.0
22	23	NaN	B	65	55	96.0	96.0	55.000000	52.0
23	24	NaN	B	39	66	41.0	41.0	66.000000	63.0
24	25	NaN	B	45	66	52.0	52.0	66.000000	78.0
25	26	NaN	B	85	11	63.0	63.0	11.000000	89.0
26	27	NaN	B	75	22	78.0	78.0	22.000000	45.0
27	28	NaN	B	65	33	89.0	NaN	33.000000	2.0
28	29	NaN	B	39	74	45.0	2.0	74.000000	44.0
29	30	NaN	B	45	88	10.0	3.0	45.000000	55.0

	Total_Marks	Percentage	Result
0	479	95.8	Pass
1	361	72.2	Fail
2	346	69.2	Fail
3	345	69.0	Pass
4	414	82.8	Pass
5	458	91.6	Pass
6	392	78.4	Fail
7	319	63.8	Pass
8	371	74.2	Fail
9	356	71.2	Pass
10	377	75.4	Pass
11	388	77.6	Fail
12	348	69.6	Fail
13	487	97.4	Pass
14	266	53.2	Fail
15	374	74.8	Pass
16	347	69.4	Fail
17	366	73.2	Fail
18	447	89.4	Pass
19	373	74.6	Pass
20	390	78.0	Pass
21	248	49.6	Pass
22	419	83.8	Pass
23	316	63.2	Pass
24	359	71.8	Pass
25	322	64.4	Pass

26	320	64.0	Pass
27	222	44.4	Pass
28	278	55.6	Pass
29	246	49.2	Pass

[ ]:
```
df.dropna(axis=1) #Dropping columns with at least 1 null value.
```

[ ]:

	Roll no	Term	Attendance	Sub1	Sub4	Total_Marks	Percentage	Result
0	1	A	75	80	96.000000	479	95.8	Pass
1	2	A	65	60	41.000000	361	72.2	Fail
2	3	A	55	77	52.000000	346	69.2	Fail
3	4	A	65	88	63.000000	345	69.0	Pass
4	5	A	45	99	78.000000	414	82.8	Pass
5	6	A	96	44	89.000000	458	91.6	Pass
6	7	A	96	55	45.000000	392	78.4	Fail
7	8	A	85	66	60.000000	319	63.8	Pass
8	9	A	75	11	77.000000	371	74.2	Fail
9	10	A	65	22	88.000000	356	71.2	Pass
10	11	A	39	33	99.000000	377	75.4	Pass
11	12	A	45	74	96.000000	388	77.6	Fail
12	13	A	78	85	41.000000	348	69.6	Fail
13	14	A	99	96	52.000000	487	97.4	Pass
14	15	A	10	41	63.000000	266	53.2	Fail
15	16	A	66	52	78.000000	374	74.8	Pass
16	17	A	44	63	89.000000	347	69.4	Fail
17	18	A	55	78	64.103448	366	73.2	Fail
18	19	A	77	89	60.000000	447	89.4	Pass
19	20	A	88	45	77.000000	373	74.6	Pass
20	21	B	85	99	99.000000	390	78.0	Pass
21	22	B	75	44	44.000000	248	49.6	Pass
22	23	B	65	55	55.000000	419	83.8	Pass
23	24	B	39	66	66.000000	316	63.2	Pass
24	25	B	45	66	66.000000	359	71.8	Pass
25	26	B	85	11	11.000000	322	64.4	Pass
26	27	B	75	22	22.000000	320	64.0	Pass
27	28	B	65	33	33.000000	222	44.4	Pass
28	29	B	39	74	74.000000	278	55.6	Pass
29	30	B	45	88	45.000000	246	49.2	Pass

[ ]:
```
df.dropna(axis=0,how='any',inplace=True) #Dropping Rows with at least 1 null␣
↪value in CSV file
```

[ ]:
```
df
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	\
0	1	A	A	75	80	85.0	77.0	96.000000	66.0	
2	3	C	A	55	77	41.0	99.0	52.000000	22.0	

3	4	D	A	65	88	52.0	44.0	63.000000	33.0
4	5	E	A	45	99	63.0	55.0	78.000000	74.0
5	6	F	A	96	44	78.0	66.0	89.000000	85.0
6	7	G	A	96	55	89.0	11.0	45.000000	96.0
7	8	H	A	85	66	45.0	22.0	60.000000	41.0
8	9	I	A	75	11	60.0	96.0	77.000000	52.0
9	10	J	A	65	22	77.0	41.0	88.000000	63.0
10	11	K	A	39	33	88.0	52.0	99.000000	66.0
12	13	M	A	78	85	44.0	78.0	41.000000	22.0
14	15	O	A	10	41	66.0	45.0	63.000000	41.0
15	16	P	A	66	52	66.0	60.0	78.000000	52.0
16	17	Q	A	44	63	11.0	77.0	89.000000	63.0
17	18	R	A	55	78	22.0	88.0	64.103448	78.0
18	19	S	A	77	89	33.0	99.0	60.000000	89.0
19	20	T	A	88	45	74.0	44.0	77.000000	45.0

	Total_Marks	Percentage	Result
0	479	95.8	Pass
2	346	69.2	Fail
3	345	69.0	Pass
4	414	82.8	Pass
5	458	91.6	Pass
6	392	78.4	Fail
7	319	63.8	Pass
8	371	74.2	Fail
9	356	71.2	Pass
10	377	75.4	Pass
12	348	69.6	Fail
14	266	53.2	Fail
15	374	74.8	Pass
16	347	69.4	Fail
17	366	73.2	Fail
18	447	89.4	Pass
19	373	74.6	Pass

2 B. Identification and Handling of Outliers

Detecting Outliers

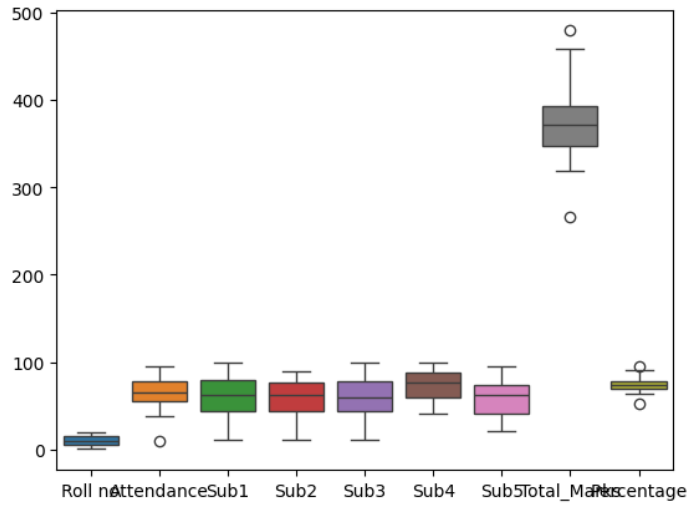
1. Detecting outliers using Boxplot:

[ ]:
```
import seaborn as sns
```

[ ]:
```
import matplotlib.pyplot as plt
```

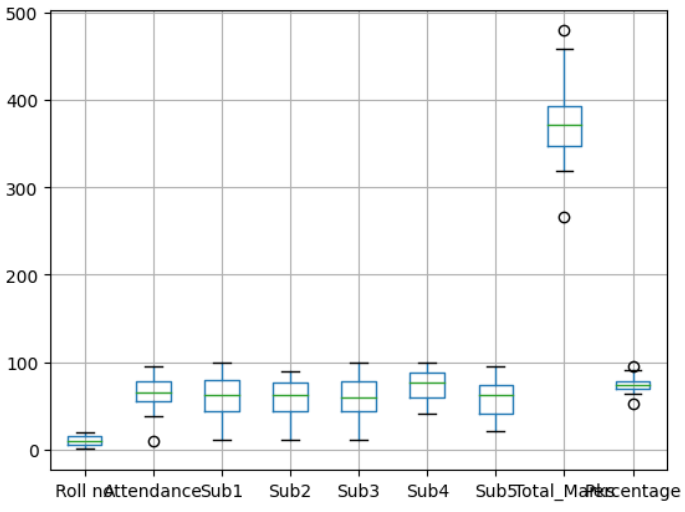
[ ]:
```
sns.boxplot(df)
```


[ ]: <Axes: >



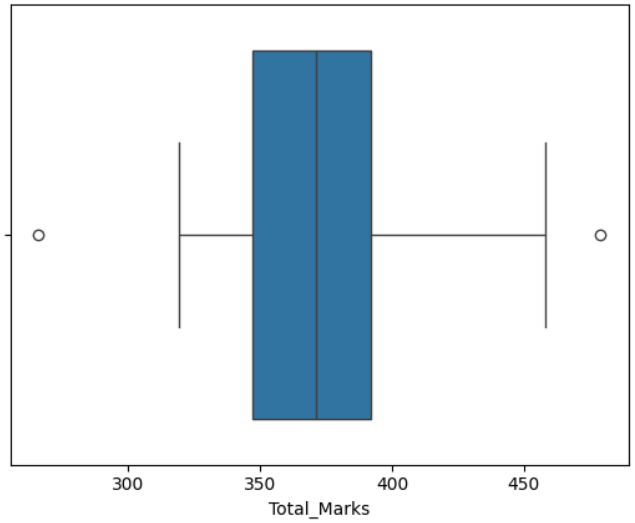
[ ]:
```
df.boxplot()
```

[ ]: <Axes: >



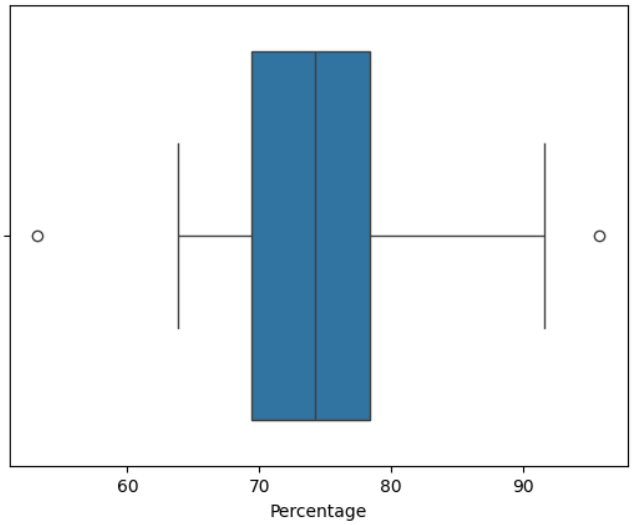
[ ]:
```
sns.boxplot(x=df.Total_Marks)
```

[ ]:
```
<Axes: xlabel='Total_Marks'>
```

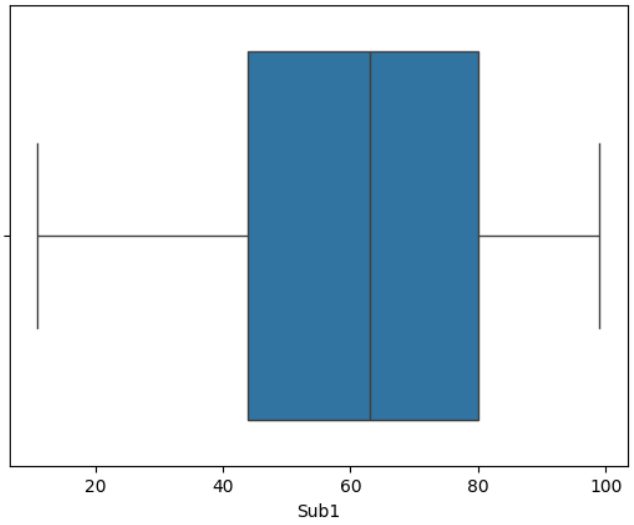
[ ]:
```
sns.boxplot(x=df.Percentage)
```

[ ]:
```
<Axes: xlabel='Percentage'>
```



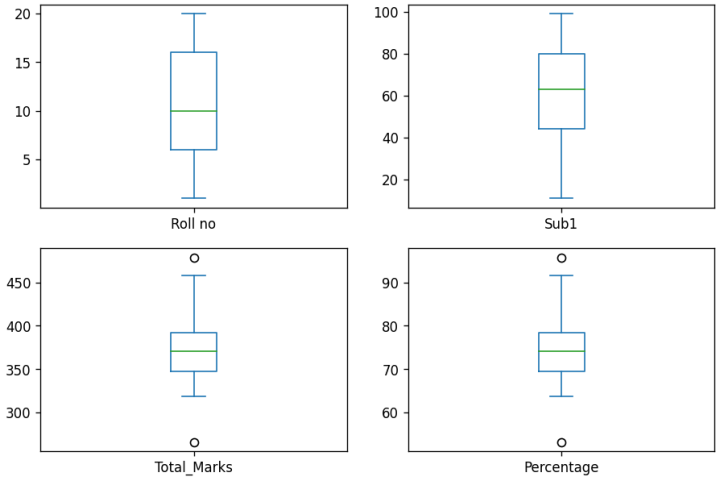
[ ]:
```
sns.boxplot(x=df.Sub1)
```

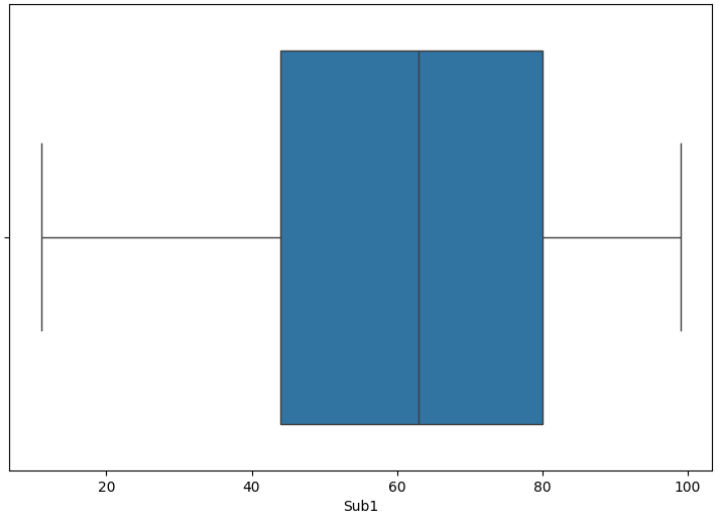
[ ]:
```
<Axes: xlabel='Sub1'>
```

[ ]:
```
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (9, 6)
df_list = ['Roll no','Sub1','Total_Marks','Percentage']
fig, axes = plt.subplots(2, 2)
fig.set_dpi(120)

count=0
for r in range(2):
  for c in range(2):
    _ = df[df_list[count]].plot(kind = 'box', ax=axes[r,c])
    count+=1
```





2.Detect outlier using z-score

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np


# Calculate z-score for Sub1
z_scores_sub1 = np.abs((df['Sub1'] - df['Sub1'].mean()) / df['Sub1'].std())

# Set threshold for outlier detection (e.g., 3 standard deviations)
threshold = 3

# Detect outliers in Sub1
outliers_sub1 = df[z_scores_sub1 > threshold]

# Plot boxplot for Sub1
plt.figure(figsize=(9, 6))
sns.boxplot(x=df['Sub1'])
plt.xlabel('Sub1')
plt.show()
```


3.Detecting outliers using Inter Quantile Range(IQR):

[ ]:
```
Q1 = df['Percentage'].quantile(0.25)
Q3 = df['Percentage'].quantile(0.75)
IQR = Q3 - Q1
Lower_limit = Q1 - 1.5 * IQR
Upper_limit = Q3 + 1.5 * IQR
print(f'Q1 = {Q1}, Q3 = {Q3}, IQR = {IQR}, Lower_limit = {Lower_limit},␣
↪Upper_limit = {Upper_limit}')
```

```
Q1 = 69.4, Q3 = 78.4, IQR = 9.0, Lower_limit = 55.900000000000006, Upper_limit =
91.9
```

[ ]:
```
df[(df['Percentage'] < Lower_limit) | (df['Percentage'] > Upper_limit)]
```

[ ]:

	Roll no	Name	Term	Attendance	Sub1	Sub2	Sub3	Sub4	Sub5	Total_Marks	Percentage	Result
0	1	A	A	75	80	85.0	77.0	96.0	66.0	479	95.8	Pass
14	15	O	A	10	41	66.0	45.0	63.0	41.0	266	53.2	Fail

3 Handling of Outliers

1.Removing the outlier:

[ ]:
```
outliers=[]
for i in df.Percentage:
  if i<Lower_limit or i>Upper_limit:
    outliers.append(i)
print("outliers are",outliers)
```

```
outliers are [95.8, 53.2]
```

[ ]:
```
Upper_limit
```

[ ]:
```
91.9
```

[ ]:
```
Lower_limit
```

[ ]:
```
55.900000000000006
```

[ ]:
```
df[df.Percentage<Lower_limit].index
```

[ ]:
```
Int64Index([14], dtype='int64')
```

[ ]:
```
df1=df.drop(df[df.Percentage<Lower_limit].index)
```

[ ]:
```
df1.shape
```

[ ]:
```
(16, 12)
```

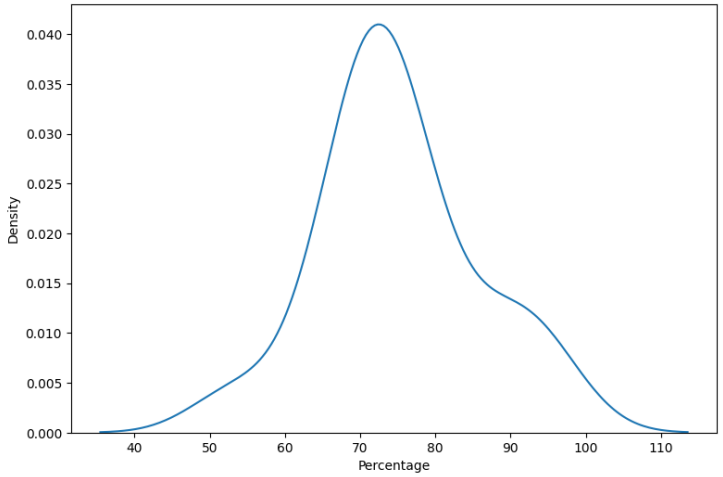
[ ]:
```
df2=df[df.Percentage<Lower_limit]
df2
```

[ ]:
```
    Roll no Name Term  Attendance  Sub1  Sub2  Sub3  Sub4  Sub5  Total_Marks  \
14       15    0    A          10    41  66.0  45.0  63.0  41.0          266

    Percentage Result
14        53.2   Fail
```

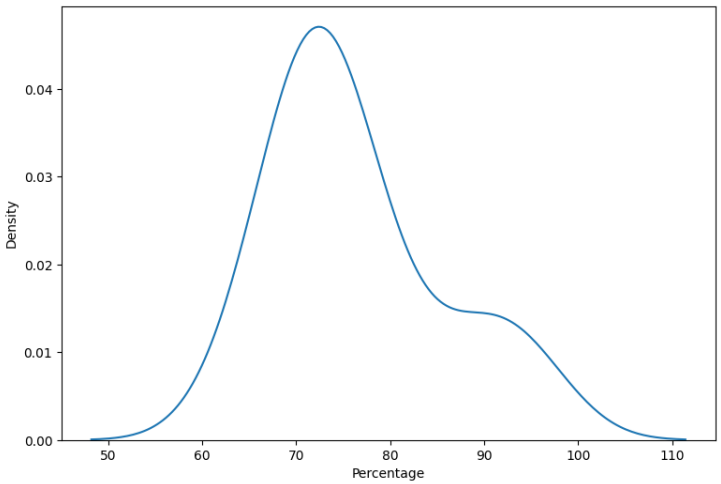
[ ]:
```
sns.kdeplot(df.Percentage)
```

[ ]:
```
<Axes: xlabel='Percentage', ylabel='Density'>
```

[ ]:
```
sns.kdeplot(df1.Percentage)
```

[ ]:
```
<Axes: xlabel='Percentage', ylabel='Density'>
```



[ ]:
```
df.Percentage
```

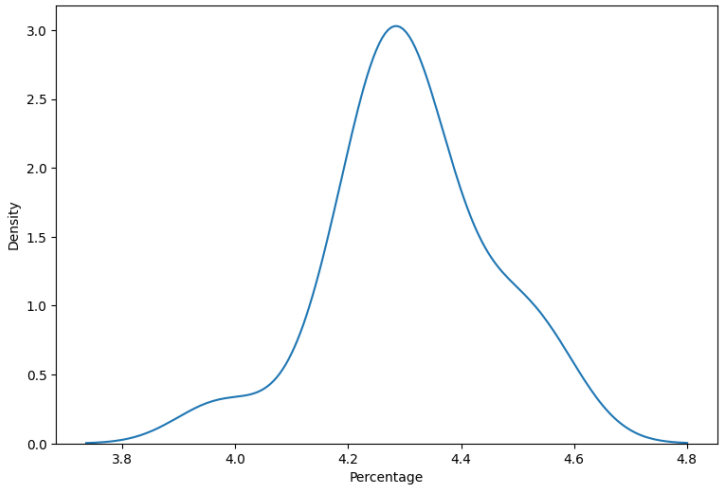
[ ]:
```
0     95.8
2     69.2
3     69.0
4     82.8
5     91.6
6     78.4
7     63.8
8     74.2
9     71.2
10    75.4
12    69.6
14    53.2
15    74.8
16    69.4
17    73.2
18    89.4
19    74.6
Name: Percentage, dtype: float64
```

[ ]:
```
log_percentage=np.log(df.Percentage)
log_percentage
```


[ ]:
```
0     4.562263
2     4.237001
3     4.234107
4     4.416428
5     4.517431
6     4.361824
7     4.155753
8     4.306764
9     4.265493
10    4.322807
12    4.242765
14    3.974058
15    4.314818
16    4.239887
17    4.293195
18    4.493121
19    4.312141
Name: Percentage, dtype: float64
```

[ ]:
```
sns.kdeplot(log_percentage)
```

[ ]:
```
<Axes: xlabel='Percentage', ylabel='Density'>
```



4 C. Data Transformation

Checking the distribution of variables using KDE plot

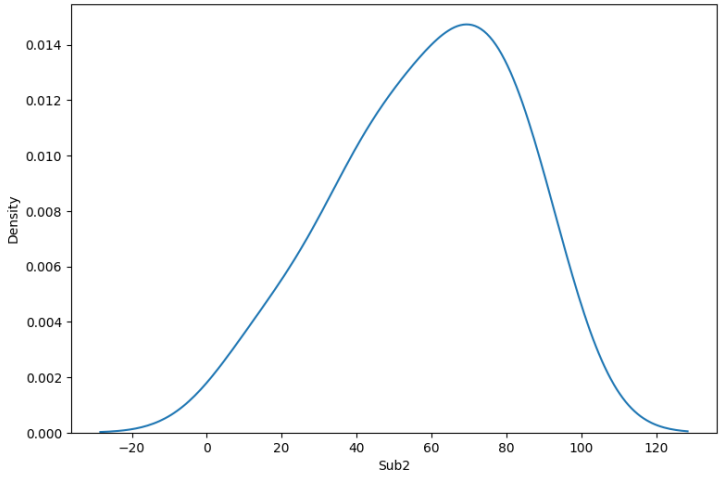
[ ]:
```
import seaborn as sns
```

[ ]:
```
#skewness in the data
df.skew()
```

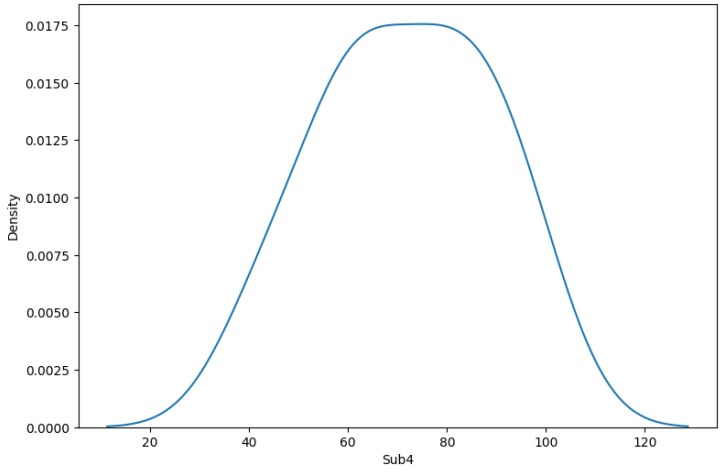
```
<ipython-input-61-c17eff935268>:2: FutureWarning: The default value of
numeric_only in DataFrame.skew is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  df.skew()
```

[ ]:
```
Roll no        0.068925
Attendance    -0.810614
Sub1          -0.351822
Sub2          -0.523061
Sub3          -0.214410
Sub4          -0.141315
Sub5          -0.041157
Total_Marks    0.270321
Percentage     0.270321
dtype: float64
```

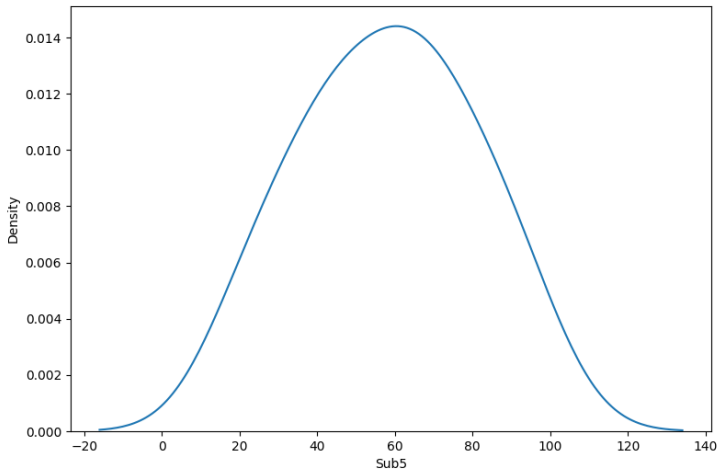
[ ]:
```
sns.kdeplot(df.Sub2);
```

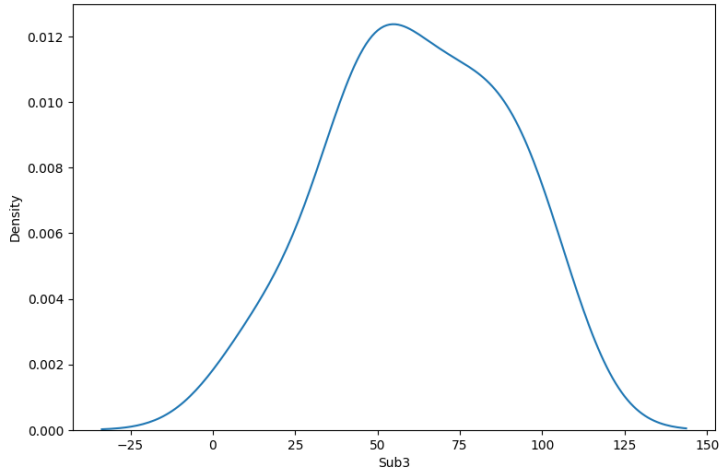
[ ]:
```
sns.kdeplot(df.Sub4);
```



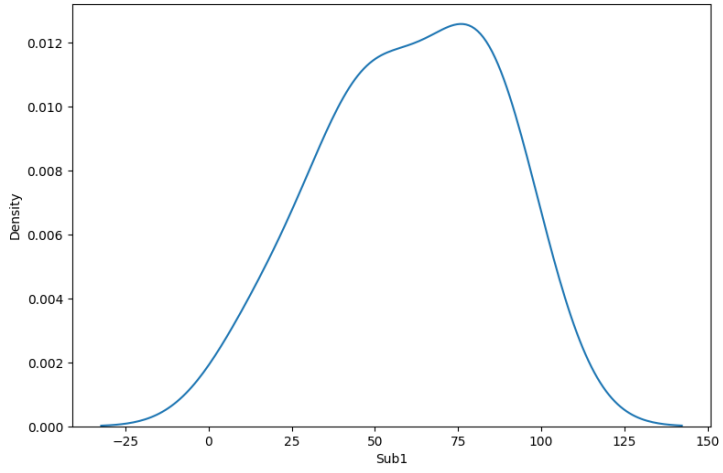
[ ]:
```
sns.kdeplot(df.Sub5);
```



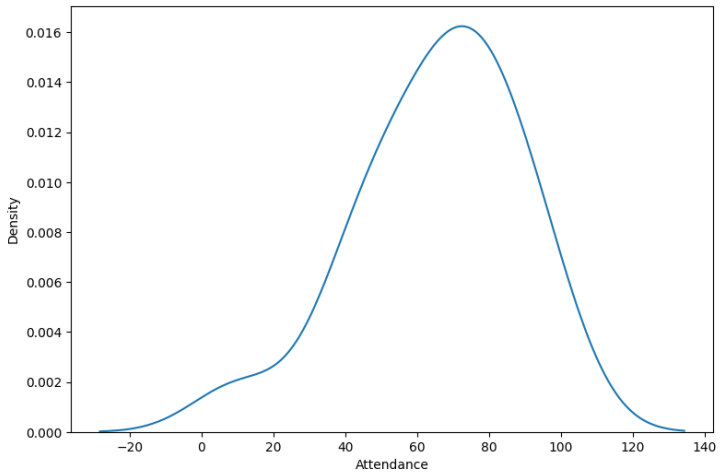
[ ]:
```
sns.kdeplot(df.Sub3);
```

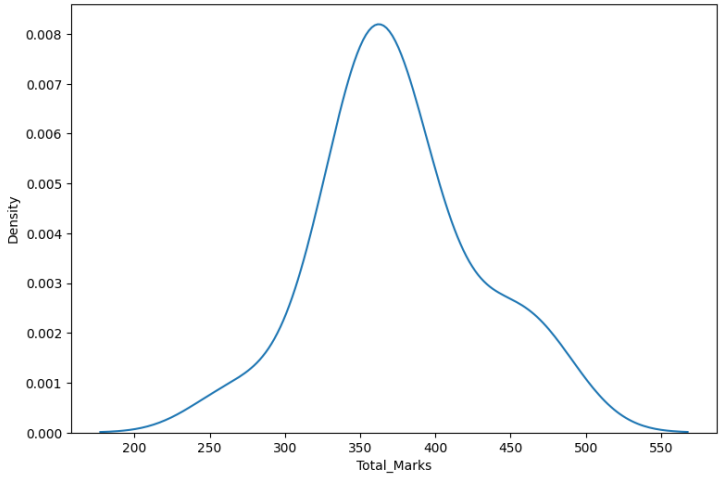
[ ]:
```
sns.kdeplot(df.Sub1);
```



[ ]:
```
sns.kdeplot(df.Attendance);
```



[ ]:
```
sns.kdeplot(df.Total_Marks);
```

[ ]:
```
sns.kdeplot(df.Percentage);
```