practical2-dsbda

March 4, 2025

1 *Piyusha Supe 23CO315*

Title of the Assignment: Data Wrangling, II

Create an "Academic performance" dataset of students and perform the following operations using Python. 1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them. 2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them. 3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution. Reason and document your approach properly. 1. Import all the required Python Libraries.

1. Import all the required Python Libraries

[3]:
```
import pandas as pd
import numpy as np
```

2. Creation of Dataset using Microsoft Excel. 3.Load the Dataset into pandas dataframe

[1]:
```
from google.colab import files
files.upload()
```

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

```
Saving academics.csv to academics (1).csv
```

[1]:
```
{'academics (1).csv': b'sr,rollno,term,attendance,s1,s2,s3,s4,s5,totalmarks,perc
entage,result\r\n1,220012,A,20,56,4,80,8,15,163,32.6,FAIL\r\n2,220013,A,62,3,10,
70,72,80,235,47,PASS\r\n3,220014,A,38,0,45,4,29,70,148,29.6,FAIL\r\n4,220015,A,9
3,58,26,52,5,29,170,34,FAIL\r\n5,220016,B,27,3,0,48,100,79,230,46,PASS\r\n6,2200
17,B,80,99,77,38,43,,257,51.4,PASS\r\n7,220018,B,67,24,64,25,31,77,221,44.2,PASS
\r\n8,220019,B,95,54,20,93,48,38,253,50.6,PASS\r\n9,220020,A,28,9,73,78,29,67,25
6,51.2,PASS\r\n10,220021,A,76,54,7,59,82,52,254,50.8,PASS\r\n11,220022,A,8,88,92
,51,41,69,341,4000,PASS\r\n12,220023,A,77,,15,24,24,41,104,20.8,FAIL\r\n13,22002
4,B,35,14,15,36,68,30,163,32.6,FAIL\r\n14,220025,B,72,77,16,,59,94,246,49.2,PASS
\r\n15,220026,B,90,62,65,38,31,47,243,48.6,PASS\r\n16,220027,B,83,57,25,69,79,28
,258,51.6,PASS\r\n17,220028,A,29,65,34,90,73,43,305,300,PASS\r\n18,220029,A,53,8
6,41,77,32,61,297,59.4,PASS\r\n19,220030,A,60,73,0,24,40,61,198,39.6,FAIL\r\n20,
```

220031,A,83,48,52,9,85,30,224,300,PASS\r\n21,220032,B,4,60,77,31,94,14,276,55.2,
PASS\r\n22,220033,B,40,54,51,51,23,10,189,37.8,FAIL\r\n23,220034,B,34,48,60,71,3
8,32,249,49.8,PASS\r\n24,220035,B,33,41,6,86,72,88,293,58.6,PASS\r\n25,220036,A,
79,91,86,22,43,33,275,55,PASS\r\n26,220037,A,48,60,25,87,23,11,206,41.2,PASS\r\n
27,220038,A,66,53,63,100,76,89,381,3022,PASS\r\n28,220039,A,22,49,65,,98,35,247,
49.4,PASS\r\n29,220040,B,35,23,30,52,37,100,242,48.4,PASS\r\n30,220041,B,51,500,
48,50,10,600,192,38.4,FAIL\r\n31,220042,B,51,20,58,73,9,1,161,32.2,FAIL\r\n32,22
0043,B,100,21,85,56,28,98,288,57.6,PASS\r\n33,220044,A,48,34,67,400,84,,226,45.2
,PASS\r\n34,220045,A,21,27,66,81,36,29,239,400000,PASS\r\n35,220046,A,27,,21,97,
64,19,201,40.2,PASS\r\n36,220047,A,81,37,4,76,23,58,198,39.6,FAIL\r\n37,220048,B
,37,70,51,40,33,86,280,56,PASS\r\n38,220049,B,59,98,2,99,67,40,306,61.2,PASS\r\n
39,220050,B,13,10,28,400,59,45,149,29.8,FAIL\r\n40,220051,B,37,500,67,57,36,52,2
58,51.6,PASS\r\n41,220052,A,10,70,22,29,27,,148,29.6,FAIL\r\n42,220053,A,46,83,5
1,31,42,84,291,58.2,PASS\r\n43,220054,A,96,42,92,61,68,34,297,59.4,PASS\r\n44,22
0055,A,18,0,55,13,71,80,219,43.8,PASS\r\n45,220056,B,10,48,8,27,76,57,216,43.2,P
ASS\r\n46,220057,B,99,58,93,52,54,47,304,60.8,PASS\r\n47,220058,B,75,,30,95,68,4
4,237,47.4,PASS\r\n48,220059,B,24,25,9,26,75,3,138,27.6,FAIL\r\n49,220060,A,56,7
9,54,65,,42,240,48,PASS\r\n50,220061,A,84,80,73,36,88,2,279,55.8,PASS\r\n'}

[4]:
```python
df = pd.read_csv("/content/academics.csv")
```

[5]:
```python
df
```

[5]:

	sr	rollno	term	attendance	s1	s2	s3	s4	s5	totalmarks	\
0	1	220012	A	20	56.0	4	80.0	8.0	15.0	163	
1	2	220013	A	62	3.0	10	70.0	72.0	80.0	235	
2	3	220014	A	38	0.0	45	4.0	29.0	70.0	148	
3	4	220015	A	93	58.0	26	52.0	5.0	29.0	170	
4	5	220016	B	27	3.0	0	48.0	100.0	79.0	230	
5	6	220017	B	80	99.0	77	38.0	43.0	NaN	257	
6	7	220018	B	67	24.0	64	25.0	31.0	77.0	221	
7	8	220019	B	95	54.0	20	93.0	48.0	38.0	253	
8	9	220020	A	28	9.0	73	78.0	29.0	67.0	256	
9	10	220021	A	76	54.0	7	59.0	82.0	52.0	254	
10	11	220022	A	8	88.0	92	51.0	41.0	69.0	341	
11	12	220023	A	77	NaN	15	24.0	24.0	41.0	104	
12	13	220024	B	35	14.0	15	36.0	68.0	30.0	163	
13	14	220025	B	72	77.0	16	NaN	59.0	94.0	246	
14	15	220026	B	90	62.0	65	38.0	31.0	47.0	243	
15	16	220027	B	83	57.0	25	69.0	79.0	28.0	258	
16	17	220028	A	29	65.0	34	90.0	73.0	43.0	305	
17	18	220029	A	53	86.0	41	77.0	32.0	61.0	297	
18	19	220030	A	60	73.0	0	24.0	40.0	61.0	198	
19	20	220031	A	83	48.0	52	9.0	85.0	30.0	224	
20	21	220032	B	4	60.0	77	31.0	94.0	14.0	276	
21	22	220033	B	40	54.0	51	51.0	23.0	10.0	189	
22	23	220034	B	34	48.0	60	71.0	38.0	32.0	249	

23	24	220035	B	33	41.0	6	86.0	72.0	88.0	293
24	25	220036	A	79	91.0	86	22.0	43.0	33.0	275
25	26	220037	A	48	60.0	25	87.0	23.0	11.0	206
26	27	220038	A	66	53.0	63	100.0	76.0	89.0	381
27	28	220039	A	22	49.0	65	NaN	98.0	35.0	247
28	29	220040	B	35	23.0	30	52.0	37.0	100.0	242
29	30	220041	B	51	500.0	48	50.0	10.0	600.0	192
30	31	220042	B	51	20.0	58	73.0	9.0	1.0	161
31	32	220043	B	100	21.0	85	56.0	28.0	98.0	288
32	33	220044	A	48	34.0	67	400.0	84.0	NaN	226
33	34	220045	A	21	27.0	66	81.0	36.0	29.0	239
34	35	220046	A	27	NaN	21	97.0	64.0	19.0	201
35	36	220047	A	81	37.0	4	76.0	23.0	58.0	198
36	37	220048	B	37	70.0	51	40.0	33.0	86.0	280
37	38	220049	B	59	98.0	2	99.0	67.0	40.0	306
38	39	220050	B	13	10.0	28	400.0	59.0	45.0	149
39	40	220051	B	37	500.0	67	57.0	36.0	52.0	258
40	41	220052	A	10	70.0	22	29.0	27.0	NaN	148
41	42	220053	A	46	83.0	51	31.0	42.0	84.0	291
42	43	220054	A	96	42.0	92	61.0	68.0	34.0	297
43	44	220055	A	18	0.0	55	13.0	71.0	80.0	219
44	45	220056	B	10	48.0	8	27.0	76.0	57.0	216
45	46	220057	B	99	58.0	93	52.0	54.0	47.0	304
46	47	220058	B	75	NaN	30	95.0	68.0	44.0	237
47	48	220059	B	24	25.0	9	26.0	75.0	3.0	138
48	49	220060	A	56	79.0	54	65.0	NaN	42.0	240
49	50	220061	A	84	80.0	73	36.0	88.0	2.0	279

	percentage	result
0	32.6	FAIL
1	47.0	PASS
2	29.6	FAIL
3	34.0	FAIL
4	46.0	PASS
5	51.4	PASS
6	44.2	PASS
7	50.6	PASS
8	51.2	PASS
9	50.8	PASS
10	4000.0	PASS
11	20.8	FAIL
12	32.6	FAIL
13	49.2	PASS
14	48.6	PASS
15	51.6	PASS
16	300.0	PASS
17	59.4	PASS

```
18         39.6   FAIL
19        300.0   PASS
20         55.2   PASS
21         37.8   FAIL
22         49.8   PASS
23         58.6   PASS
24         55.0   PASS
25         41.2   PASS
26       3022.0   PASS
27         49.4   PASS
28         48.4   PASS
29         38.4   FAIL
30         32.2   FAIL
31         57.6   PASS
32         45.2   PASS
33     400000.0   PASS
34         40.2   PASS
35         39.6   FAIL
36         56.0   PASS
37         61.2   PASS
38         29.8   FAIL
39         51.6   PASS
40         29.6   FAIL
41         58.2   PASS
42         59.4   PASS
43         43.8   PASS
44         43.2   PASS
45         60.8   PASS
46         47.4   PASS
47         27.6   FAIL
48         48.0   PASS
49         55.8   PASS
```

4. Data preprocessing

[6]:
```
df.head()
```

[6]:
```
   sr  rollno term  attendance    s1  s2    s3     s4    s5  totalmarks  \
0   1  220012    A          20  56.0   4  80.0    8.0  15.0         163
1   2  220013    A          62   3.0  10  70.0   72.0  80.0         235
2   3  220014    A          38   0.0  45   4.0   29.0  70.0         148
3   4  220015    A          93  58.0  26  52.0    5.0  29.0         170
4   5  220016    B          27   3.0   0  48.0  100.0  79.0         230

   percentage result
0        32.6   FAIL
1        47.0   PASS
```

```
2          29.6    FAIL
3          34.0    FAIL
4          46.0    PASS
```

[7]:
```
df.tail()
```

[7]:
```
    sr  rollno term  attendance    s1  s2    s3    s4    s5  totalmarks  \
45  46  220057    B          99  58.0  93  52.0  54.0  47.0         304
46  47  220058    B          75   NaN  30  95.0  68.0  44.0         237
47  48  220059    B          24  25.0   9  26.0  75.0   3.0         138
48  49  220060    A          56  79.0  54  65.0   NaN  42.0         240
49  50  220061    A          84  80.0  73  36.0  88.0   2.0         279

    percentage result
45        60.8   PASS
46        47.4   PASS
47        27.6   FAIL
48        48.0   PASS
49        55.8   PASS
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   sr          50 non-null     int64
 1   rollno      50 non-null     int64
 2   term        50 non-null     object
 3   attendance  50 non-null     int64
 4   s1          47 non-null     float64
 5   s2          50 non-null     int64
 6   s3          48 non-null     float64
 7   s4          49 non-null     float64
 8   s5          47 non-null     float64
 9   totalmarks  50 non-null     int64
 10  percentage  50 non-null     float64
 11  result      50 non-null     object
dtypes: float64(5), int64(5), object(2)
memory usage: 4.8+ KB
```

[9]:
```
df.describe()
```

[9]:
```
             sr        rollno  attendance         s1         s2         s3  \
count  50.00000      50.00000    50.00000  47.000000  50.000000  48.000000
mean   25.50000  220036.50000    51.60000  68.319149  42.560000  69.354167
```

std	14.57738	14.57738	27.88021	95.889226	28.385222	74.313164
min	1.00000	220012.00000	4.00000	0.000000	0.000000	4.000000
25%	13.25000	220024.25000	28.25000	26.000000	17.000000	34.750000
50%	25.50000	220036.50000	49.50000	54.000000	46.500000	54.000000
75%	37.75000	220048.75000	76.75000	71.500000	65.000000	78.500000
max	50.00000	220061.00000	100.00000	500.000000	93.000000	400.000000

	s4	s5	totalmarks	percentage
count	49.000000	47.000000	50.000000	50.00000
mean	51.040816	60.510638	235.820000	8193.64400
std	25.877081	84.896580	55.999158	56544.83886
min	5.000000	1.000000	104.000000	20.80000
25%	31.000000	30.000000	198.750000	39.75000
50%	43.000000	45.000000	241.000000	48.90000
75%	72.000000	73.500000	275.750000	55.95000
max	100.000000	600.000000	381.000000	400000.00000

[10]:
```python
df.describe(include="all")
```

[10]:

	sr	rollno	term	attendance	s1	s2 \
count	50.00000	50.00000	50	50.00000	47.000000	50.000000
unique	NaN	NaN	2	NaN	NaN	NaN
top	NaN	NaN	A	NaN	NaN	NaN
freq	NaN	NaN	26	NaN	NaN	NaN
mean	25.50000	220036.50000	NaN	51.60000	68.319149	42.560000
std	14.57738	14.57738	NaN	27.88021	95.889226	28.385222
min	1.00000	220012.00000	NaN	4.00000	0.000000	0.000000
25%	13.25000	220024.25000	NaN	28.25000	26.000000	17.000000
50%	25.50000	220036.50000	NaN	49.50000	54.000000	46.500000
75%	37.75000	220048.75000	NaN	76.75000	71.500000	65.000000
max	50.00000	220061.00000	NaN	100.00000	500.000000	93.000000

	s3	s4	s5	totalmarks	percentage	result
count	48.000000	49.000000	47.000000	50.000000	50.00000	50
unique	NaN	NaN	NaN	NaN	NaN	2
top	NaN	NaN	NaN	NaN	NaN	PASS
freq	NaN	NaN	NaN	NaN	NaN	37
mean	69.354167	51.040816	60.510638	235.820000	8193.64400	NaN
std	74.313164	25.877081	84.896580	55.999158	56544.83886	NaN
min	4.000000	5.000000	1.000000	104.000000	20.80000	NaN
25%	34.750000	31.000000	30.000000	198.750000	39.75000	NaN
50%	54.000000	43.000000	45.000000	241.000000	48.90000	NaN
75%	78.500000	72.000000	73.500000	275.750000	55.95000	NaN
max	400.000000	100.000000	600.000000	381.000000	400000.00000	NaN

[11]:
```python
df.shape
```

[11]: (50, 12)

[12]:
```
df.size
```

[12]: 600

[13]:
```
df.columns
```

[13]:
```
Index(['sr', 'rollno', 'term', 'attendance', 's1', 's2', 's3', 's4', 's5',
       'totalmarks', 'percentage', 'result'],
      dtype='object')
```

[14]:
```
df.ndim
```

[14]: 2

[15]:
```
df.dtypes
```

[15]:
```
sr              int64
rollno          int64
term           object
attendance      int64
s1            float64
s2              int64
s3            float64
s4            float64
s5            float64
totalmarks      int64
percentage    float64
result         object
dtype: object
```

[16]:
```
df[0:5]
```

[16]:

	sr	rollno	term	attendance	s1	s2	s3	s4	s5	totalmarks
0	1	220012	A	20	56.0	4	80.0	8.0	15.0	163
1	2	220013	A	62	3.0	10	70.0	72.0	80.0	235
2	3	220014	A	38	0.0	45	4.0	29.0	70.0	148
3	4	220015	A	93	58.0	26	52.0	5.0	29.0	170
4	5	220016	B	27	3.0	0	48.0	100.0	79.0	230

\

	percentage	result
0	32.6	FAIL
1	47.0	PASS
2	29.6	FAIL
3	34.0	FAIL
4	46.0	PASS

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

```
[17]:    sr  rollno term  attendance    s1  s2    s3    s4    s5  totalmarks  \
      0   1  220012    A          20  56.0   4  80.0   8.0  15.0         163
      1   2  220013    A          62   3.0  10  70.0  72.0  80.0         235
      2   3  220014    A          38   0.0  45   4.0  29.0  70.0         148

         percentage result
      0        32.6   FAIL
      1        47.0   PASS
      2        29.6   FAIL
```

```
[18]: df.loc[0:2,'s1':'s5']
```

```
[18]:      s1  s2    s3    s4    s5
      0  56.0   4  80.0   8.0  15.0
      1   3.0  10  70.0  72.0  80.0
      2   0.0  45   4.0  29.0  70.0
```

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

```
[19]:    sr  rollno term  attendance   s1  s2    s3    s4    s5  totalmarks  \
      1   2  220013    A          62  3.0  10  70.0  72.0  80.0         235
      2   3  220014    A          38  0.0  45   4.0  29.0  70.0         148

         percentage result
      1        47.0   PASS
      2        29.6   FAIL
```

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

```
[20]:    rollno term  attendance    s1
      1  220013    A          62   3.0
      2  220014    A          38   0.0
      3  220015    A          93  58.0
      4  220016    B          27   3.0
```

1.1 A. Identification and Handling of Null Values

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

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

```
[21]:       sr  rollno   term  attendance     s1     s2     s3     s4     s5  \
      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
5	False	False	False	False	False	False	False	False	True
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	True	False	False	False	False
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	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False	False	False
25	False	False	False	False	False	False	False	False	False
26	False	False	False	False	False	False	False	False	False
27	False	False	False	False	False	False	True	False	False
28	False	False	False	False	False	False	False	False	False
29	False	False	False	False	False	False	False	False	False
30	False	False	False	False	False	False	False	False	False
31	False	False	False	False	False	False	False	False	False
32	False	False	False	False	False	False	False	False	True
33	False	False	False	False	False	False	False	False	False
34	False	False	False	False	True	False	False	False	False
35	False	False	False	False	False	False	False	False	False
36	False	False	False	False	False	False	False	False	False
37	False	False	False	False	False	False	False	False	False
38	False	False	False	False	False	False	False	False	False
39	False	False	False	False	False	False	False	False	False
40	False	False	False	False	False	False	False	False	True
41	False	False	False	False	False	False	False	False	False
42	False	False	False	False	False	False	False	False	False
43	False	False	False	False	False	False	False	False	False
44	False	False	False	False	False	False	False	False	False
45	False	False	False	False	False	False	False	False	False
46	False	False	False	False	True	False	False	False	False
47	False	False	False	False	False	False	False	False	False
48	False	False	False	False	False	False	False	True	False
49	False	False	False	False	False	False	False	False	False

```
    totalmarks  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
30       False       False   False
31       False       False   False
32       False       False   False
33       False       False   False
34       False       False   False
35       False       False   False
36       False       False   False
37       False       False   False
38       False       False   False
39       False       False   False
40       False       False   False
41       False       False   False
42       False       False   False
43       False       False   False
44       False       False   False
```

```
45       False        False    False
46       False        False    False
47       False        False    False
48       False        False    False
49       False        False    False
```

[22]:
```
df.isna()
```

[22]:
```
       sr  rollno   term  attendance     s1     s2     s3     s4     s5  \
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
5   False   False  False       False  False  False  False  False   True
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   True  False  False  False  False
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  False  False
18  False   False  False       False  False  False  False  False  False
19  False   False  False       False  False  False  False  False  False
20  False   False  False       False  False  False  False  False  False
21  False   False  False       False  False  False  False  False  False
22  False   False  False       False  False  False  False  False  False
23  False   False  False       False  False  False  False  False  False
24  False   False  False       False  False  False  False  False  False
25  False   False  False       False  False  False  False  False  False
26  False   False  False       False  False  False  False  False  False
27  False   False  False       False  False  False   True  False  False
28  False   False  False       False  False  False  False  False  False
29  False   False  False       False  False  False  False  False  False
30  False   False  False       False  False  False  False  False  False
31  False   False  False       False  False  False  False  False  False
32  False   False  False       False  False  False  False  False   True
33  False   False  False       False  False  False  False  False  False
34  False   False  False       False   True  False  False  False  False
35  False   False  False       False  False  False  False  False  False
36  False   False  False       False  False  False  False  False  False
37  False   False  False       False  False  False  False  False  False
```

```
38  False   False  False       False  False  False  False  False  False
39  False   False  False       False  False  False  False  False  False
40  False   False  False       False  False  False  False  False   True
41  False   False  False       False  False  False  False  False  False
42  False   False  False       False  False  False  False  False  False
43  False   False  False       False  False  False  False  False  False
44  False   False  False       False  False  False  False  False  False
45  False   False  False       False  False  False  False  False  False
46  False   False  False       False   True  False  False  False  False
47  False   False  False       False  False  False  False  False  False
48  False   False  False       False  False  False  False   True  False
49  False   False  False       False  False  False  False  False  False

    totalmarks  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
30       False       False   False
31       False       False   False
32       False       False   False
```

```
33       False        False    False
34       False        False    False
35       False        False    False
36       False        False    False
37       False        False    False
38       False        False    False
39       False        False    False
40       False        False    False
41       False        False    False
42       False        False    False
43       False        False    False
44       False        False    False
45       False        False    False
46       False        False    False
47       False        False    False
48       False        False    False
49       False        False    False
```

[23]:
```python
df.isnull().any()
```

[23]:
```
sr            False
rollno        False
term          False
attendance    False
s1             True
s2            False
s3             True
s4             True
s5             True
totalmarks    False
percentage    False
result        False
dtype: bool
```

[24]:
```python
df.isnull().sum()
```

[24]:
```
sr            0
rollno        0
term          0
attendance    0
s1            3
s2            0
s3            2
s4            1
s5            3
totalmarks    0
percentage    0
```

```
result        0
dtype: int64
```

[25]:
```
df.attendance.isnull().sum()
```

[25]: 0

[26]:
```
cols_with_na = []
for col in df.columns:
  if df[col].isna().any():
    cols_with_na.append(col)

cols_with_na
```

[26]: ['s1', 's3', 's4', 's5']

Filling missing values using dropna(), fillna(), replace() : 1. replacing null values with NaN

[27]:
```
df.replace(np.nan,value=0)
```

[27]:
```
    sr  rollno term  attendance    s1  s2    s3     s4    s5  totalmarks  \
0    1  220012    A          20  56.0   4  80.0    8.0  15.0         163
1    2  220013    A          62   3.0  10  70.0   72.0  80.0         235
2    3  220014    A          38   0.0  45   4.0   29.0  70.0         148
3    4  220015    A          93  58.0  26  52.0    5.0  29.0         170
4    5  220016    B          27   3.0   0  48.0  100.0  79.0         230
5    6  220017    B          80  99.0  77  38.0   43.0   0.0         257
6    7  220018    B          67  24.0  64  25.0   31.0  77.0         221
7    8  220019    B          95  54.0  20  93.0   48.0  38.0         253
8    9  220020    A          28   9.0  73  78.0   29.0  67.0         256
9   10  220021    A          76  54.0   7  59.0   82.0  52.0         254
10  11  220022    A           8  88.0  92  51.0   41.0  69.0         341
11  12  220023    A          77   0.0  15  24.0   24.0  41.0         104
12  13  220024    B          35  14.0  15  36.0   68.0  30.0         163
13  14  220025    B          72  77.0  16   0.0   59.0  94.0         246
14  15  220026    B          90  62.0  65  38.0   31.0  47.0         243
15  16  220027    B          83  57.0  25  69.0   79.0  28.0         258
16  17  220028    A          29  65.0  34  90.0   73.0  43.0         305
17  18  220029    A          53  86.0  41  77.0   32.0  61.0         297
18  19  220030    A          60  73.0   0  24.0   40.0  61.0         198
19  20  220031    A          83  48.0  52   9.0   85.0  30.0         224
20  21  220032    B           4  60.0  77  31.0   94.0  14.0         276
21  22  220033    B          40  54.0  51  51.0   23.0  10.0         189
22  23  220034    B          34  48.0  60  71.0   38.0  32.0         249
23  24  220035    B          33  41.0   6  86.0   72.0  88.0         293
24  25  220036    A          79  91.0  86  22.0   43.0  33.0         275
25  26  220037    A          48  60.0  25  87.0   23.0  11.0         206
```

26	27	220038	A	66	53.0	63	100.0	76.0	89.0	381
27	28	220039	A	22	49.0	65	0.0	98.0	35.0	247
28	29	220040	B	35	23.0	30	52.0	37.0	100.0	242
29	30	220041	B	51	500.0	48	50.0	10.0	600.0	192
30	31	220042	B	51	20.0	58	73.0	9.0	1.0	161
31	32	220043	B	100	21.0	85	56.0	28.0	98.0	288
32	33	220044	A	48	34.0	67	400.0	84.0	0.0	226
33	34	220045	A	21	27.0	66	81.0	36.0	29.0	239
34	35	220046	A	27	0.0	21	97.0	64.0	19.0	201
35	36	220047	A	81	37.0	4	76.0	23.0	58.0	198
36	37	220048	B	37	70.0	51	40.0	33.0	86.0	280
37	38	220049	B	59	98.0	2	99.0	67.0	40.0	306
38	39	220050	B	13	10.0	28	400.0	59.0	45.0	149
39	40	220051	B	37	500.0	67	57.0	36.0	52.0	258
40	41	220052	A	10	70.0	22	29.0	27.0	0.0	148
41	42	220053	A	46	83.0	51	31.0	42.0	84.0	291
42	43	220054	A	96	42.0	92	61.0	68.0	34.0	297
43	44	220055	A	18	0.0	55	13.0	71.0	80.0	219
44	45	220056	B	10	48.0	8	27.0	76.0	57.0	216
45	46	220057	B	99	58.0	93	52.0	54.0	47.0	304
46	47	220058	B	75	0.0	30	95.0	68.0	44.0	237
47	48	220059	B	24	25.0	9	26.0	75.0	3.0	138
48	49	220060	A	56	79.0	54	65.0	0.0	42.0	240
49	50	220061	A	84	80.0	73	36.0	88.0	2.0	279

	percentage	result
0	32.6	FAIL
1	47.0	PASS
2	29.6	FAIL
3	34.0	FAIL
4	46.0	PASS
5	51.4	PASS
6	44.2	PASS
7	50.6	PASS
8	51.2	PASS
9	50.8	PASS
10	4000.0	PASS
11	20.8	FAIL
12	32.6	FAIL
13	49.2	PASS
14	48.6	PASS
15	51.6	PASS
16	300.0	PASS
17	59.4	PASS
18	39.6	FAIL
19	300.0	PASS
20	55.2	PASS

```
21         37.8   FAIL
22         49.8   PASS
23         58.6   PASS
24         55.0   PASS
25         41.2   PASS
26       3022.0   PASS
27         49.4   PASS
28         48.4   PASS
29         38.4   FAIL
30         32.2   FAIL
31         57.6   PASS
32         45.2   PASS
33     400000.0   PASS
34         40.2   PASS
35         39.6   FAIL
36         56.0   PASS
37         61.2   PASS
38         29.8   FAIL
39         51.6   PASS
40         29.6   FAIL
41         58.2   PASS
42         59.4   PASS
43         43.8   PASS
44         43.2   PASS
45         60.8   PASS
46         47.4   PASS
47         27.6   FAIL
48         48.0   PASS
49         55.8   PASS
```

2. Filling null values with fill na

[28]:
```
df.fillna(1)
```

[28]:
```
    sr  rollno term  attendance    s1  s2    s3     s4    s5  totalmarks  \
0    1  220012    A          20  56.0   4  80.0    8.0  15.0         163
1    2  220013    A          62   3.0  10  70.0   72.0  80.0         235
2    3  220014    A          38   0.0  45   4.0   29.0  70.0         148
3    4  220015    A          93  58.0  26  52.0    5.0  29.0         170
4    5  220016    B          27   3.0   0  48.0  100.0  79.0         230
5    6  220017    B          80  99.0  77  38.0   43.0   1.0         257
6    7  220018    B          67  24.0  64  25.0   31.0  77.0         221
7    8  220019    B          95  54.0  20  93.0   48.0  38.0         253
8    9  220020    A          28   9.0  73  78.0   29.0  67.0         256
9   10  220021    A          76  54.0   7  59.0   82.0  52.0         254
10  11  220022    A           8  88.0  92  51.0   41.0  69.0         341
11  12  220023    A          77   1.0  15  24.0   24.0  41.0         104
```

12	13	220024	B	35	14.0	15	36.0	68.0	30.0	163
13	14	220025	B	72	77.0	16	1.0	59.0	94.0	246
14	15	220026	B	90	62.0	65	38.0	31.0	47.0	243
15	16	220027	B	83	57.0	25	69.0	79.0	28.0	258
16	17	220028	A	29	65.0	34	90.0	73.0	43.0	305
17	18	220029	A	53	86.0	41	77.0	32.0	61.0	297
18	19	220030	A	60	73.0	0	24.0	40.0	61.0	198
19	20	220031	A	83	48.0	52	9.0	85.0	30.0	224
20	21	220032	B	4	60.0	77	31.0	94.0	14.0	276
21	22	220033	B	40	54.0	51	51.0	23.0	10.0	189
22	23	220034	B	34	48.0	60	71.0	38.0	32.0	249
23	24	220035	B	33	41.0	6	86.0	72.0	88.0	293
24	25	220036	A	79	91.0	86	22.0	43.0	33.0	275
25	26	220037	A	48	60.0	25	87.0	23.0	11.0	206
26	27	220038	A	66	53.0	63	100.0	76.0	89.0	381
27	28	220039	A	22	49.0	65	1.0	98.0	35.0	247
28	29	220040	B	35	23.0	30	52.0	37.0	100.0	242
29	30	220041	B	51	500.0	48	50.0	10.0	600.0	192
30	31	220042	B	51	20.0	58	73.0	9.0	1.0	161
31	32	220043	B	100	21.0	85	56.0	28.0	98.0	288
32	33	220044	A	48	34.0	67	400.0	84.0	1.0	226
33	34	220045	A	21	27.0	66	81.0	36.0	29.0	239
34	35	220046	A	27	1.0	21	97.0	64.0	19.0	201
35	36	220047	A	81	37.0	4	76.0	23.0	58.0	198
36	37	220048	B	37	70.0	51	40.0	33.0	86.0	280
37	38	220049	B	59	98.0	2	99.0	67.0	40.0	306
38	39	220050	B	13	10.0	28	400.0	59.0	45.0	149
39	40	220051	B	37	500.0	67	57.0	36.0	52.0	258
40	41	220052	A	10	70.0	22	29.0	27.0	1.0	148
41	42	220053	A	46	83.0	51	31.0	42.0	84.0	291
42	43	220054	A	96	42.0	92	61.0	68.0	34.0	297
43	44	220055	A	18	0.0	55	13.0	71.0	80.0	219
44	45	220056	B	10	48.0	8	27.0	76.0	57.0	216
45	46	220057	B	99	58.0	93	52.0	54.0	47.0	304
46	47	220058	B	75	1.0	30	95.0	68.0	44.0	237
47	48	220059	B	24	25.0	9	26.0	75.0	3.0	138
48	49	220060	A	56	79.0	54	65.0	1.0	42.0	240
49	50	220061	A	84	80.0	73	36.0	88.0	2.0	279

	percentage	result
0	32.6	FAIL
1	47.0	PASS
2	29.6	FAIL
3	34.0	FAIL
4	46.0	PASS
5	51.4	PASS
6	44.2	PASS

```
7         50.6   PASS
8         51.2   PASS
9         50.8   PASS
10      4000.0   PASS
11        20.8   FAIL
12        32.6   FAIL
13        49.2   PASS
14        48.6   PASS
15        51.6   PASS
16       300.0   PASS
17        59.4   PASS
18        39.6   FAIL
19       300.0   PASS
20        55.2   PASS
21        37.8   FAIL
22        49.8   PASS
23        58.6   PASS
24        55.0   PASS
25        41.2   PASS
26      3022.0   PASS
27        49.4   PASS
28        48.4   PASS
29        38.4   FAIL
30        32.2   FAIL
31        57.6   PASS
32        45.2   PASS
33    400000.0   PASS
34        40.2   PASS
35        39.6   FAIL
36        56.0   PASS
37        61.2   PASS
38        29.8   FAIL
39        51.6   PASS
40        29.6   FAIL
41        58.2   PASS
42        59.4   PASS
43        43.8   PASS
44        43.2   PASS
45        60.8   PASS
46        47.4   PASS
47        27.6   FAIL
48        48.0   PASS
49        55.8   PASS
```

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

[29]:
```
df["s1"] = df["s1"].fillna(df['s1'].mean())
df["s2"] = df["s2"].fillna(df["s2"].mean())
df["s3"] = df["s3"].fillna(df["s3"].mean())
df["s4"] = df["s4"].fillna(df["s4"].mean())
df["s5"] = df["s5"].fillna(df["s5"].mean())
```

[30]:
```
df.head(10)
```

[30]:
```
   sr  rollno term  attendance    s1  s2    s3     s4         s5  totalmarks  \
0   1  220012    A          20  56.0   4  80.0    8.0  15.000000         163
1   2  220013    A          62   3.0  10  70.0   72.0  80.000000         235
2   3  220014    A          38   0.0  45   4.0   29.0  70.000000         148
3   4  220015    A          93  58.0  26  52.0    5.0  29.000000         170
4   5  220016    B          27   3.0   0  48.0  100.0  79.000000         230
5   6  220017    B          80  99.0  77  38.0   43.0  60.510638         257
6   7  220018    B          67  24.0  64  25.0   31.0  77.000000         221
7   8  220019    B          95  54.0  20  93.0   48.0  38.000000         253
8   9  220020    A          28   9.0  73  78.0   29.0  67.000000         256
9  10  220021    A          76  54.0   7  59.0   82.0  52.000000         254

   percentage result
0        32.6   FAIL
1        47.0   PASS
2        29.6   FAIL
3        34.0   FAIL
4        46.0   PASS
5        51.4   PASS
6        44.2   PASS
7        50.6   PASS
8        51.2   PASS
9        50.8   PASS
```

4.Deleting null values using dropna() method

[31]:
```
df.dropna()
```

[31]:
```
    sr  rollno term  attendance         s1  s2         s3          s4  \
0    1  220012    A          20  56.000000   4  80.000000    8.000000
1    2  220013    A          62   3.000000  10  70.000000   72.000000
2    3  220014    A          38   0.000000  45   4.000000   29.000000
3    4  220015    A          93  58.000000  26  52.000000    5.000000
4    5  220016    B          27   3.000000   0  48.000000  100.000000
5    6  220017    B          80  99.000000  77  38.000000   43.000000
6    7  220018    B          67  24.000000  64  25.000000   31.000000
7    8  220019    B          95  54.000000  20  93.000000   48.000000
8    9  220020    A          28   9.000000  73  78.000000   29.000000
9   10  220021    A          76  54.000000   7  59.000000   82.000000
```

10	11	220022	A	8	88.000000	92	51.000000	41.000000
11	12	220023	A	77	68.319149	15	24.000000	24.000000
12	13	220024	B	35	14.000000	15	36.000000	68.000000
13	14	220025	B	72	77.000000	16	69.354167	59.000000
14	15	220026	B	90	62.000000	65	38.000000	31.000000
15	16	220027	B	83	57.000000	25	69.000000	79.000000
16	17	220028	A	29	65.000000	34	90.000000	73.000000
17	18	220029	A	53	86.000000	41	77.000000	32.000000
18	19	220030	A	60	73.000000	0	24.000000	40.000000
19	20	220031	A	83	48.000000	52	9.000000	85.000000
20	21	220032	B	4	60.000000	77	31.000000	94.000000
21	22	220033	B	40	54.000000	51	51.000000	23.000000
22	23	220034	B	34	48.000000	60	71.000000	38.000000
23	24	220035	B	33	41.000000	6	86.000000	72.000000
24	25	220036	A	79	91.000000	86	22.000000	43.000000
25	26	220037	A	48	60.000000	25	87.000000	23.000000
26	27	220038	A	66	53.000000	63	100.000000	76.000000
27	28	220039	A	22	49.000000	65	69.354167	98.000000
28	29	220040	B	35	23.000000	30	52.000000	37.000000
29	30	220041	B	51	500.000000	48	50.000000	10.000000
30	31	220042	B	51	20.000000	58	73.000000	9.000000
31	32	220043	B	100	21.000000	85	56.000000	28.000000
32	33	220044	A	48	34.000000	67	400.000000	84.000000
33	34	220045	A	21	27.000000	66	81.000000	36.000000
34	35	220046	A	27	68.319149	21	97.000000	64.000000
35	36	220047	A	81	37.000000	4	76.000000	23.000000
36	37	220048	B	37	70.000000	51	40.000000	33.000000
37	38	220049	B	59	98.000000	2	99.000000	67.000000
38	39	220050	B	13	10.000000	28	400.000000	59.000000
39	40	220051	B	37	500.000000	67	57.000000	36.000000
40	41	220052	A	10	70.000000	22	29.000000	27.000000
41	42	220053	A	46	83.000000	51	31.000000	42.000000
42	43	220054	A	96	42.000000	92	61.000000	68.000000
43	44	220055	A	18	0.000000	55	13.000000	71.000000
44	45	220056	B	10	48.000000	8	27.000000	76.000000
45	46	220057	B	99	58.000000	93	52.000000	54.000000
46	47	220058	B	75	68.319149	30	95.000000	68.000000
47	48	220059	B	24	25.000000	9	26.000000	75.000000
48	49	220060	A	56	79.000000	54	65.000000	51.040816
49	50	220061	A	84	80.000000	73	36.000000	88.000000

	s5	totalmarks	percentage	result
0	15.000000	163	32.6	FAIL
1	80.000000	235	47.0	PASS
2	70.000000	148	29.6	FAIL
3	29.000000	170	34.0	FAIL
4	79.000000	230	46.0	PASS

5	60.510638	257	51.4	PASS
6	77.000000	221	44.2	PASS
7	38.000000	253	50.6	PASS
8	67.000000	256	51.2	PASS
9	52.000000	254	50.8	PASS
10	69.000000	341	4000.0	PASS
11	41.000000	104	20.8	FAIL
12	30.000000	163	32.6	FAIL
13	94.000000	246	49.2	PASS
14	47.000000	243	48.6	PASS
15	28.000000	258	51.6	PASS
16	43.000000	305	300.0	PASS
17	61.000000	297	59.4	PASS
18	61.000000	198	39.6	FAIL
19	30.000000	224	300.0	PASS
20	14.000000	276	55.2	PASS
21	10.000000	189	37.8	FAIL
22	32.000000	249	49.8	PASS
23	88.000000	293	58.6	PASS
24	33.000000	275	55.0	PASS
25	11.000000	206	41.2	PASS
26	89.000000	381	3022.0	PASS
27	35.000000	247	49.4	PASS
28	100.000000	242	48.4	PASS
29	600.000000	192	38.4	FAIL
30	1.000000	161	32.2	FAIL
31	98.000000	288	57.6	PASS
32	60.510638	226	45.2	PASS
33	29.000000	239	400000.0	PASS
34	19.000000	201	40.2	PASS
35	58.000000	198	39.6	FAIL
36	86.000000	280	56.0	PASS
37	40.000000	306	61.2	PASS
38	45.000000	149	29.8	FAIL
39	52.000000	258	51.6	PASS
40	60.510638	148	29.6	FAIL
41	84.000000	291	58.2	PASS
42	34.000000	297	59.4	PASS
43	80.000000	219	43.8	PASS
44	57.000000	216	43.2	PASS
45	47.000000	304	60.8	PASS
46	44.000000	237	47.4	PASS
47	3.000000	138	27.6	FAIL
48	42.000000	240	48.0	PASS
49	2.000000	279	55.8	PASS

[32]:
```
df.dropna(how='all')
```

[32]:

	sr	rollno	term	attendance	s1	s2	s3	s4	\
0	1	220012	A	20	56.000000	4	80.000000	8.000000	
1	2	220013	A	62	3.000000	10	70.000000	72.000000	
2	3	220014	A	38	0.000000	45	4.000000	29.000000	
3	4	220015	A	93	58.000000	26	52.000000	5.000000	
4	5	220016	B	27	3.000000	0	48.000000	100.000000	
5	6	220017	B	80	99.000000	77	38.000000	43.000000	
6	7	220018	B	67	24.000000	64	25.000000	31.000000	
7	8	220019	B	95	54.000000	20	93.000000	48.000000	
8	9	220020	A	28	9.000000	73	78.000000	29.000000	
9	10	220021	A	76	54.000000	7	59.000000	82.000000	
10	11	220022	A	8	88.000000	92	51.000000	41.000000	
11	12	220023	A	77	68.319149	15	24.000000	24.000000	
12	13	220024	B	35	14.000000	15	36.000000	68.000000	
13	14	220025	B	72	77.000000	16	69.354167	59.000000	
14	15	220026	B	90	62.000000	65	38.000000	31.000000	
15	16	220027	B	83	57.000000	25	69.000000	79.000000	
16	17	220028	A	29	65.000000	34	90.000000	73.000000	
17	18	220029	A	53	86.000000	41	77.000000	32.000000	
18	19	220030	A	60	73.000000	0	24.000000	40.000000	
19	20	220031	A	83	48.000000	52	9.000000	85.000000	
20	21	220032	B	4	60.000000	77	31.000000	94.000000	
21	22	220033	B	40	54.000000	51	51.000000	23.000000	
22	23	220034	B	34	48.000000	60	71.000000	38.000000	
23	24	220035	B	33	41.000000	6	86.000000	72.000000	
24	25	220036	A	79	91.000000	86	22.000000	43.000000	
25	26	220037	A	48	60.000000	25	87.000000	23.000000	
26	27	220038	A	66	53.000000	63	100.000000	76.000000	
27	28	220039	A	22	49.000000	65	69.354167	98.000000	
28	29	220040	B	35	23.000000	30	52.000000	37.000000	
29	30	220041	B	51	500.000000	48	50.000000	10.000000	
30	31	220042	B	51	20.000000	58	73.000000	9.000000	
31	32	220043	B	100	21.000000	85	56.000000	28.000000	
32	33	220044	A	48	34.000000	67	400.000000	84.000000	
33	34	220045	A	21	27.000000	66	81.000000	36.000000	
34	35	220046	A	27	68.319149	21	97.000000	64.000000	
35	36	220047	A	81	37.000000	4	76.000000	23.000000	
36	37	220048	B	37	70.000000	51	40.000000	33.000000	
37	38	220049	B	59	98.000000	2	99.000000	67.000000	
38	39	220050	B	13	10.000000	28	400.000000	59.000000	
39	40	220051	B	37	500.000000	67	57.000000	36.000000	
40	41	220052	A	10	70.000000	22	29.000000	27.000000	
41	42	220053	A	46	83.000000	51	31.000000	42.000000	
42	43	220054	A	96	42.000000	92	61.000000	68.000000	
43	44	220055	A	18	0.000000	55	13.000000	71.000000	
44	45	220056	B	10	48.000000	8	27.000000	76.000000	
45	46	220057	B	99	58.000000	93	52.000000	54.000000	

46	47	220058	B	75	68.319149	30	95.000000	68.000000
47	48	220059	B	24	25.000000	9	26.000000	75.000000
48	49	220060	A	56	79.000000	54	65.000000	51.040816
49	50	220061	A	84	80.000000	73	36.000000	88.000000

	s5	totalmarks	percentage	result
0	15.000000	163	32.6	FAIL
1	80.000000	235	47.0	PASS
2	70.000000	148	29.6	FAIL
3	29.000000	170	34.0	FAIL
4	79.000000	230	46.0	PASS
5	60.510638	257	51.4	PASS
6	77.000000	221	44.2	PASS
7	38.000000	253	50.6	PASS
8	67.000000	256	51.2	PASS
9	52.000000	254	50.8	PASS
10	69.000000	341	4000.0	PASS
11	41.000000	104	20.8	FAIL
12	30.000000	163	32.6	FAIL
13	94.000000	246	49.2	PASS
14	47.000000	243	48.6	PASS
15	28.000000	258	51.6	PASS
16	43.000000	305	300.0	PASS
17	61.000000	297	59.4	PASS
18	61.000000	198	39.6	FAIL
19	30.000000	224	300.0	PASS
20	14.000000	276	55.2	PASS
21	10.000000	189	37.8	FAIL
22	32.000000	249	49.8	PASS
23	88.000000	293	58.6	PASS
24	33.000000	275	55.0	PASS
25	11.000000	206	41.2	PASS
26	89.000000	381	3022.0	PASS
27	35.000000	247	49.4	PASS
28	100.000000	242	48.4	PASS
29	600.000000	192	38.4	FAIL
30	1.000000	161	32.2	FAIL
31	98.000000	288	57.6	PASS
32	60.510638	226	45.2	PASS
33	29.000000	239	400000.0	PASS
34	19.000000	201	40.2	PASS
35	58.000000	198	39.6	FAIL
36	86.000000	280	56.0	PASS
37	40.000000	306	61.2	PASS
38	45.000000	149	29.8	FAIL
39	52.000000	258	51.6	PASS
40	60.510638	148	29.6	FAIL

```
41   84.000000         291        58.2   PASS
42   34.000000         297        59.4   PASS
43   80.000000         219        43.8   PASS
44   57.000000         216        43.2   PASS
45   47.000000         304        60.8   PASS
46   44.000000         237        47.4   PASS
47    3.000000         138        27.6   FAIL
48   42.000000         240        48.0   PASS
49    2.000000         279        55.8   PASS
```

[33]:
```
df.dropna(axis=1)
```

[33]:
```
    sr  rollno term  attendance          s1  s2          s3          s4  \
0    1  220012    A          20   56.000000   4   80.000000    8.000000
1    2  220013    A          62    3.000000  10   70.000000   72.000000
2    3  220014    A          38    0.000000  45    4.000000   29.000000
3    4  220015    A          93   58.000000  26   52.000000    5.000000
4    5  220016    B          27    3.000000   0   48.000000  100.000000
5    6  220017    B          80   99.000000  77   38.000000   43.000000
6    7  220018    B          67   24.000000  64   25.000000   31.000000
7    8  220019    B          95   54.000000  20   93.000000   48.000000
8    9  220020    A          28    9.000000  73   78.000000   29.000000
9   10  220021    A          76   54.000000   7   59.000000   82.000000
10  11  220022    A           8   88.000000  92   51.000000   41.000000
11  12  220023    A          77   68.319149  15   24.000000   24.000000
12  13  220024    B          35   14.000000  15   36.000000   68.000000
13  14  220025    B          72   77.000000  16   69.354167   59.000000
14  15  220026    B          90   62.000000  65   38.000000   31.000000
15  16  220027    B          83   57.000000  25   69.000000   79.000000
16  17  220028    A          29   65.000000  34   90.000000   73.000000
17  18  220029    A          53   86.000000  41   77.000000   32.000000
18  19  220030    A          60   73.000000   0   24.000000   40.000000
19  20  220031    A          83   48.000000  52    9.000000   85.000000
20  21  220032    B           4   60.000000  77   31.000000   94.000000
21  22  220033    B          40   54.000000  51   51.000000   23.000000
22  23  220034    B          34   48.000000  60   71.000000   38.000000
23  24  220035    B          33   41.000000   6   86.000000   72.000000
24  25  220036    A          79   91.000000  86   22.000000   43.000000
25  26  220037    A          48   60.000000  25   87.000000   23.000000
26  27  220038    A          66   53.000000  63  100.000000   76.000000
27  28  220039    A          22   49.000000  65   69.354167   98.000000
28  29  220040    B          35   23.000000  30   52.000000   37.000000
29  30  220041    B          51  500.000000  48   50.000000   10.000000
30  31  220042    B          51   20.000000  58   73.000000    9.000000
31  32  220043    B         100   21.000000  85   56.000000   28.000000
32  33  220044    A          48   34.000000  67  400.000000   84.000000
33  34  220045    A          21   27.000000  66   81.000000   36.000000
```

34	35	220046	A	27	68.319149	21	97.000000	64.000000
35	36	220047	A	81	37.000000	4	76.000000	23.000000
36	37	220048	B	37	70.000000	51	40.000000	33.000000
37	38	220049	B	59	98.000000	2	99.000000	67.000000
38	39	220050	B	13	10.000000	28	400.000000	59.000000
39	40	220051	B	37	500.000000	67	57.000000	36.000000
40	41	220052	A	10	70.000000	22	29.000000	27.000000
41	42	220053	A	46	83.000000	51	31.000000	42.000000
42	43	220054	A	96	42.000000	92	61.000000	68.000000
43	44	220055	A	18	0.000000	55	13.000000	71.000000
44	45	220056	B	10	48.000000	8	27.000000	76.000000
45	46	220057	B	99	58.000000	93	52.000000	54.000000
46	47	220058	B	75	68.319149	30	95.000000	68.000000
47	48	220059	B	24	25.000000	9	26.000000	75.000000
48	49	220060	A	56	79.000000	54	65.000000	51.040816
49	50	220061	A	84	80.000000	73	36.000000	88.000000

	s5	totalmarks	percentage	result
0	15.000000	163	32.6	FAIL
1	80.000000	235	47.0	PASS
2	70.000000	148	29.6	FAIL
3	29.000000	170	34.0	FAIL
4	79.000000	230	46.0	PASS
5	60.510638	257	51.4	PASS
6	77.000000	221	44.2	PASS
7	38.000000	253	50.6	PASS
8	67.000000	256	51.2	PASS
9	52.000000	254	50.8	PASS
10	69.000000	341	4000.0	PASS
11	41.000000	104	20.8	FAIL
12	30.000000	163	32.6	FAIL
13	94.000000	246	49.2	PASS
14	47.000000	243	48.6	PASS
15	28.000000	258	51.6	PASS
16	43.000000	305	300.0	PASS
17	61.000000	297	59.4	PASS
18	61.000000	198	39.6	FAIL
19	30.000000	224	300.0	PASS
20	14.000000	276	55.2	PASS
21	10.000000	189	37.8	FAIL
22	32.000000	249	49.8	PASS
23	88.000000	293	58.6	PASS
24	33.000000	275	55.0	PASS
25	11.000000	206	41.2	PASS
26	89.000000	381	3022.0	PASS
27	35.000000	247	49.4	PASS
28	100.000000	242	48.4	PASS

```
29  600.000000          192        38.4    FAIL
30    1.000000          161        32.2    FAIL
31   98.000000          288        57.6    PASS
32   60.510638          226        45.2    PASS
33   29.000000          239    400000.0    PASS
34   19.000000          201        40.2    PASS
35   58.000000          198        39.6    FAIL
36   86.000000          280        56.0    PASS
37   40.000000          306        61.2    PASS
38   45.000000          149        29.8    FAIL
39   52.000000          258        51.6    PASS
40   60.510638          148        29.6    FAIL
41   84.000000          291        58.2    PASS
42   34.000000          297        59.4    PASS
43   80.000000          219        43.8    PASS
44   57.000000          216        43.2    PASS
45   47.000000          304        60.8    PASS
46   44.000000          237        47.4    PASS
47    3.000000          138        27.6    FAIL
48   42.000000          240        48.0    PASS
49    2.000000          279        55.8    PASS
```

[89]:
```
df.dropna(axis=0,how='any',inplace=True)
```

[90]:
```
df
```

[90]:
```
    sr  rollno term  attendance         s1  s2         s3          s4  \
0    1  220012    A          20  56.000000   4  80.000000    8.000000
1    2  220013    A          62   3.000000  10  70.000000   72.000000
2    3  220014    A          38   0.000000  45   4.000000   29.000000
3    4  220015    A          93  58.000000  26  52.000000    5.000000
4    5  220016    B          27   3.000000   0  48.000000  100.000000
5    6  220017    B          80  99.000000  77  38.000000   43.000000
6    7  220018    B          67  24.000000  64  25.000000   31.000000
7    8  220019    B          95  54.000000  20  93.000000   48.000000
8    9  220020    A          28   9.000000  73  78.000000   29.000000
9   10  220021    A          76  54.000000   7  59.000000   82.000000
10  11  220022    A           8  88.000000  92  51.000000   41.000000
11  12  220023    A          77  68.319149  15  24.000000   24.000000
12  13  220024    B          35  14.000000  15  36.000000   68.000000
13  14  220025    B          72  77.000000  16  69.354167   59.000000
14  15  220026    B          90  62.000000  65  38.000000   31.000000
15  16  220027    B          83  57.000000  25  69.000000   79.000000
16  17  220028    A          29  65.000000  34  90.000000   73.000000
17  18  220029    A          53  86.000000  41  77.000000   32.000000
18  19  220030    A          60  73.000000   0  24.000000   40.000000
19  20  220031    A          83  48.000000  52   9.000000   85.000000
```

20	21	220032	B	4	60.000000	77	31.000000	94.000000
21	22	220033	B	40	54.000000	51	51.000000	23.000000
22	23	220034	B	34	48.000000	60	71.000000	38.000000
23	24	220035	B	33	41.000000	6	86.000000	72.000000
24	25	220036	A	79	91.000000	86	22.000000	43.000000
25	26	220037	A	48	60.000000	25	87.000000	23.000000
26	27	220038	A	66	53.000000	63	100.000000	76.000000
27	28	220039	A	22	49.000000	65	69.354167	98.000000
28	29	220040	B	35	23.000000	30	52.000000	37.000000
29	30	220041	B	51	500.000000	48	50.000000	10.000000
30	31	220042	B	51	20.000000	58	73.000000	9.000000
31	32	220043	B	100	21.000000	85	56.000000	28.000000
32	33	220044	A	48	34.000000	67	400.000000	84.000000
33	34	220045	A	21	27.000000	66	81.000000	36.000000
34	35	220046	A	27	68.319149	21	97.000000	64.000000
35	36	220047	A	81	37.000000	4	76.000000	23.000000
36	37	220048	B	37	70.000000	51	40.000000	33.000000
37	38	220049	B	59	98.000000	2	99.000000	67.000000
38	39	220050	B	13	10.000000	28	400.000000	59.000000
39	40	220051	B	37	500.000000	67	57.000000	36.000000
40	41	220052	A	10	70.000000	22	29.000000	27.000000
41	42	220053	A	46	83.000000	51	31.000000	42.000000
42	43	220054	A	96	42.000000	92	61.000000	68.000000
43	44	220055	A	18	0.000000	55	13.000000	71.000000
44	45	220056	B	10	48.000000	8	27.000000	76.000000
45	46	220057	B	99	58.000000	93	52.000000	54.000000
46	47	220058	B	75	68.319149	30	95.000000	68.000000
47	48	220059	B	24	25.000000	9	26.000000	75.000000
48	49	220060	A	56	79.000000	54	65.000000	51.040816
49	50	220061	A	84	80.000000	73	36.000000	88.000000

	s5	totalmarks	percentage	result
0	15.000000	163	32.6	FAIL
1	80.000000	235	47.0	PASS
2	70.000000	148	29.6	FAIL
3	29.000000	170	34.0	FAIL
4	79.000000	230	46.0	PASS
5	60.510638	257	51.4	PASS
6	77.000000	221	44.2	PASS
7	38.000000	253	50.6	PASS
8	67.000000	256	51.2	PASS
9	52.000000	254	50.8	PASS
10	69.000000	341	4000.0	PASS
11	41.000000	104	20.8	FAIL
12	30.000000	163	32.6	FAIL
13	94.000000	246	49.2	PASS
14	47.000000	243	48.6	PASS

```
15   28.000000        258      51.6  PASS
16   43.000000        305     300.0  PASS
17   61.000000        297      59.4  PASS
18   61.000000        198      39.6  FAIL
19   30.000000        224     300.0  PASS
20   14.000000        276      55.2  PASS
21   10.000000        189      37.8  FAIL
22   32.000000        249      49.8  PASS
23   88.000000        293      58.6  PASS
24   33.000000        275      55.0  PASS
25   11.000000        206      41.2  PASS
26   89.000000        381    3022.0  PASS
27   35.000000        247      49.4  PASS
28  100.000000        242      48.4  PASS
29  600.000000        192      38.4  FAIL
30    1.000000        161      32.2  FAIL
31   98.000000        288      57.6  PASS
32   60.510638        226      45.2  PASS
33   29.000000        239  400000.0  PASS
34   19.000000        201      40.2  PASS
35   58.000000        198      39.6  FAIL
36   86.000000        280      56.0  PASS
37   40.000000        306      61.2  PASS
38   45.000000        149      29.8  FAIL
39   52.000000        258      51.6  PASS
40   60.510638        148      29.6  FAIL
41   84.000000        291      58.2  PASS
42   34.000000        297      59.4  PASS
43   80.000000        219      43.8  PASS
44   57.000000        216      43.2  PASS
45   47.000000        304      60.8  PASS
46   44.000000        237      47.4  PASS
47    3.000000        138      27.6  FAIL
48   42.000000        240      48.0  PASS
49    2.000000        279      55.8  PASS
```

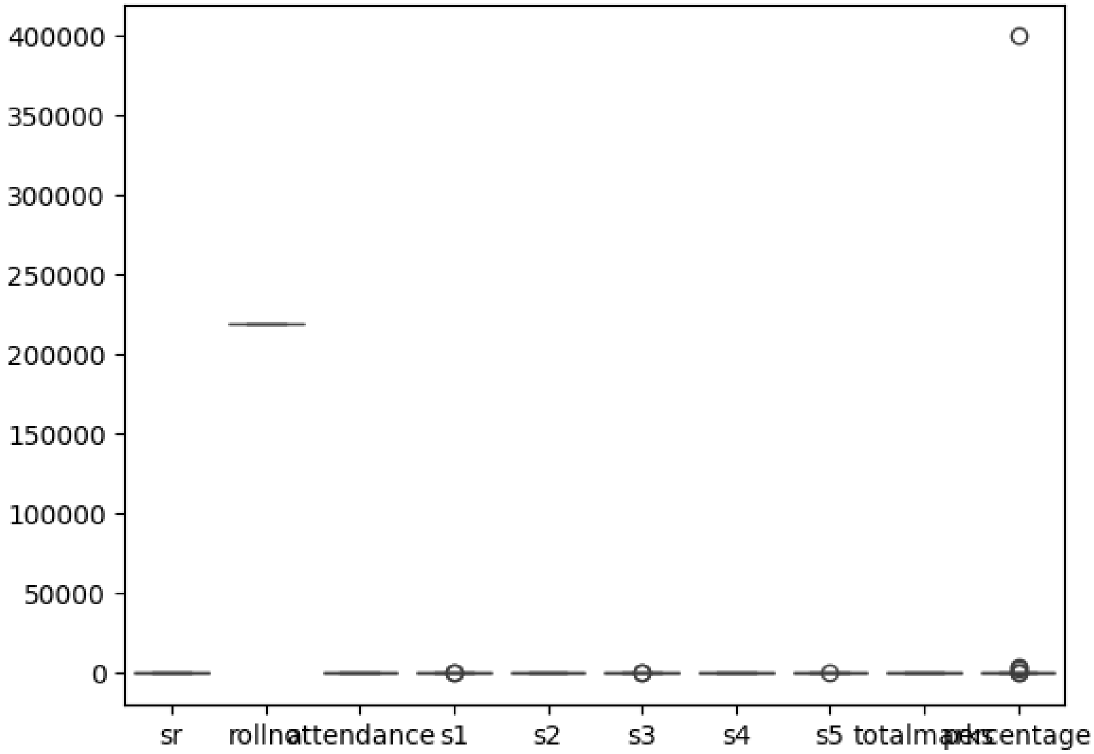
1.2 B. Identification and Handling of Outlier

Detecting Outliers 1. Detecting outliers using Boxplot:

[34]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
```

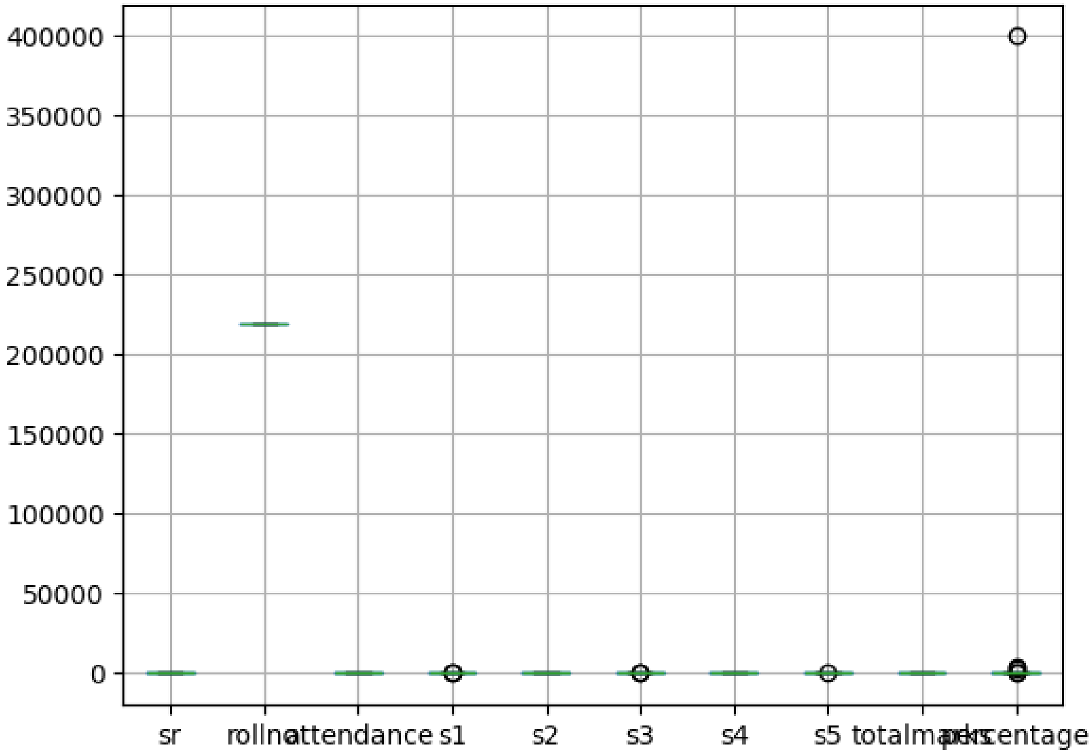
[35]:
```python
sns.boxplot(df)
```

[35]:
```
<Axes: >
```



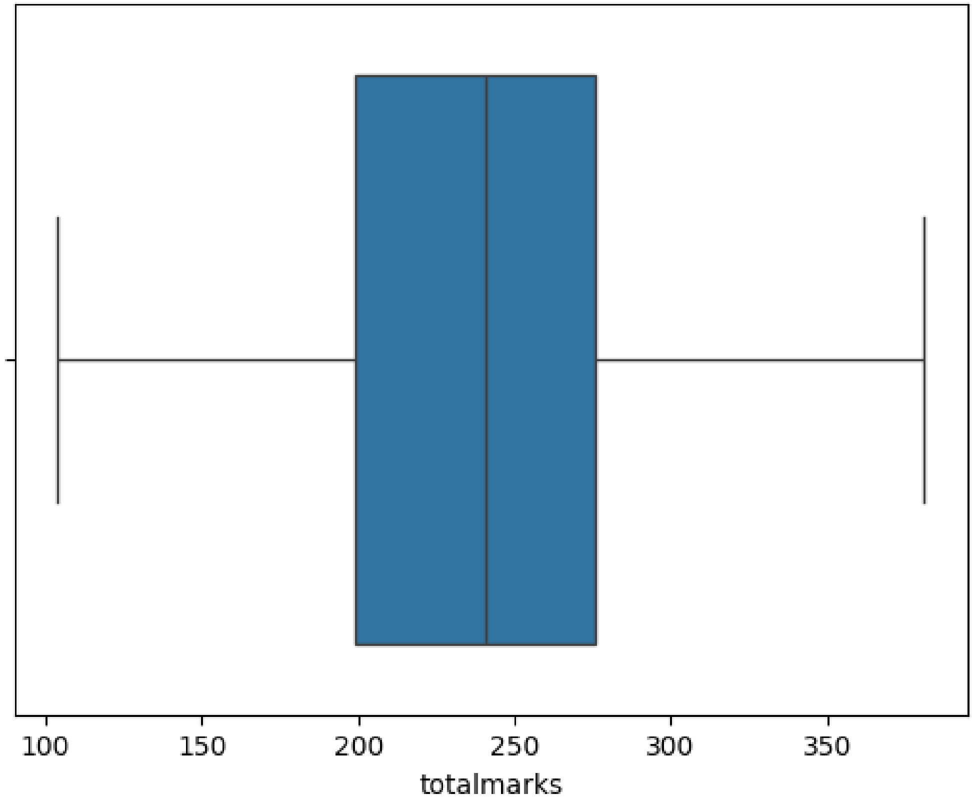
[36]:
```
df.boxplot()
```

[36]: <Axes: >



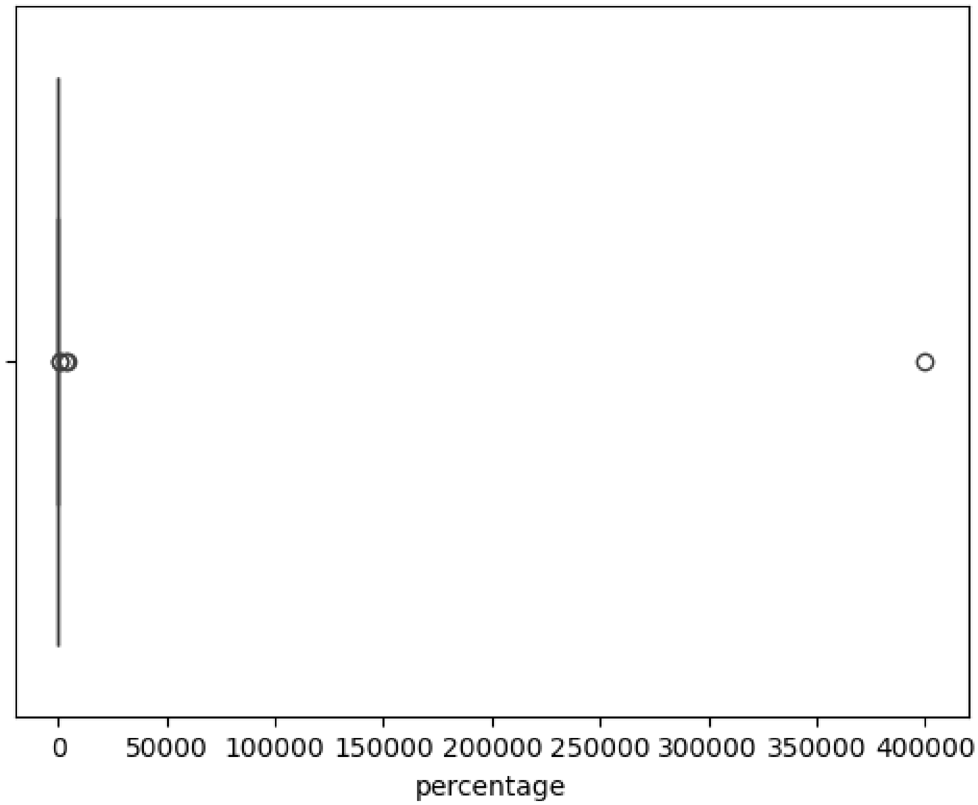
[37]:
```
sns.boxplot(x=df.totalmarks)
```

[37]: <Axes: xlabel='totalmarks'>



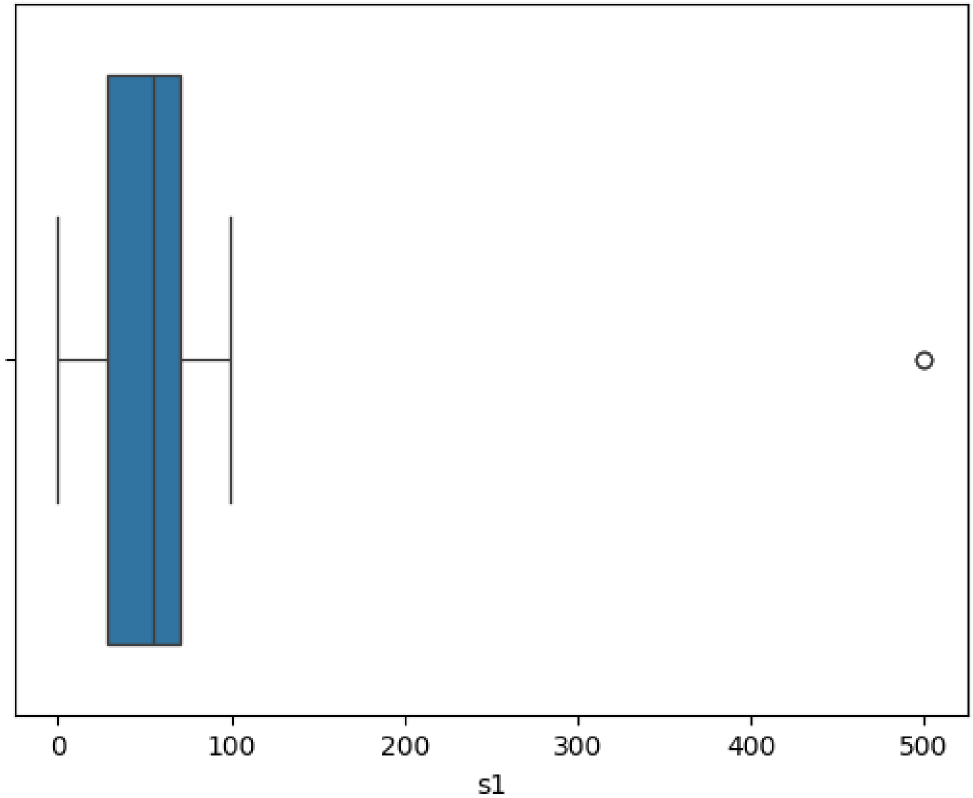
[38]:
```
sns.boxplot(x=df.percentage)
```

[38]: <Axes: xlabel='percentage'>

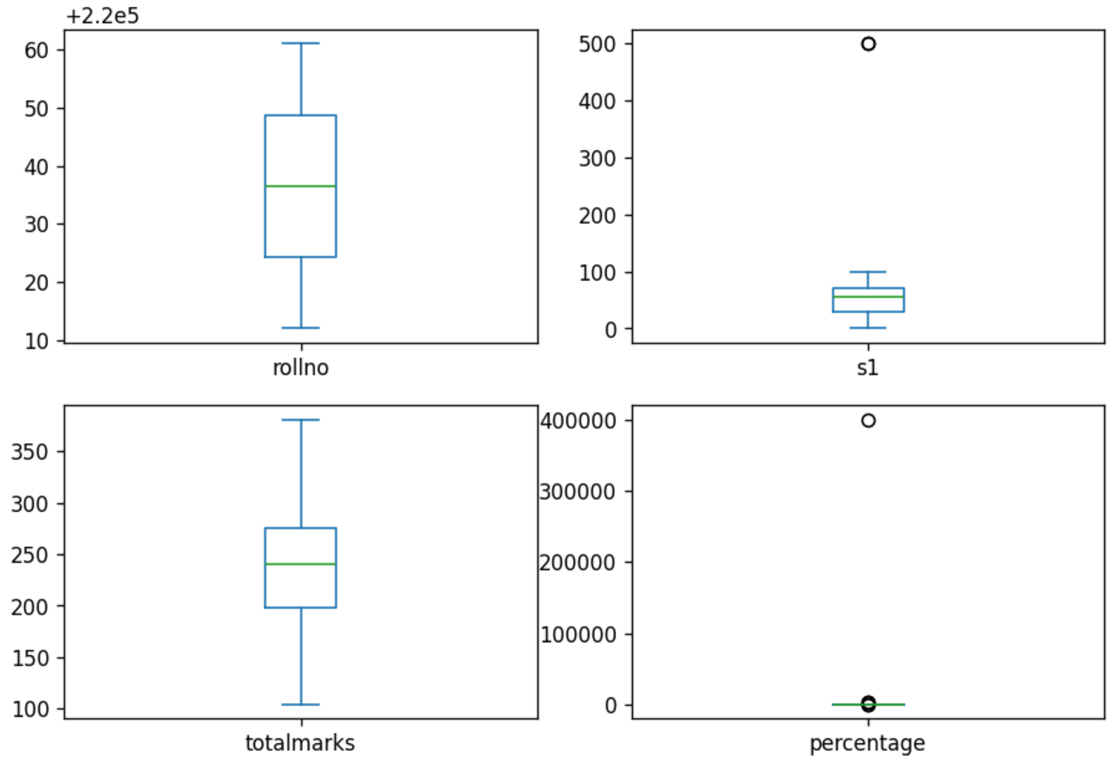


[39]:
```
sns.boxplot(x=df.s1)
```

[39]: <Axes: xlabel='s1'>



[40]:
```
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (9, 6)
df_list = ['rollno','s1','totalmarks','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
```



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

[41]:
```
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 = 39.75, Q3 = 55.95, IQR = 16.200000000000003, Lower_limit =
15.449999999999996, Upper_limit = 80.25
```

[42]:
```
df[(df['percentage'] < Lower_limit) | (df['percentage'] > Upper_limit)]
```

[42]:

	sr	rollno	term	attendance	s1	s2	s3	s4	s5	totalmarks	\
10	11	220022	A	8	88.0	92	51.0	41.0	69.0	341	
16	17	220028	A	29	65.0	34	90.0	73.0	43.0	305	
19	20	220031	A	83	48.0	52	9.0	85.0	30.0	224	
26	27	220038	A	66	53.0	63	100.0	76.0	89.0	381	
33	34	220045	A	21	27.0	66	81.0	36.0	29.0	239	

```
    percentage result
10      4000.0   PASS
16       300.0   PASS
19       300.0   PASS
26      3022.0   PASS
33    400000.0   PASS
```

Handling of Outliers 1.removing the outlier:

[43]:
```
outliers=[]
for i in df.percentage:
  if i<Lower_limit or i>Upper_limit:
       outliers.append(i)
print("outliers are",outliers)
```

```
outliers are [4000.0, 300.0, 300.0, 3022.0, 400000.0]
```

[44]:
```
Upper_limit
```

[44]:
```
80.25
```

[45]:
```
Lower_limit
```

[45]:
```
15.449999999999996
```

[46]:
```
df[df.percentage<Lower_limit].index
```

[46]:
```
Index([], dtype='int64')
```

[47]:
```
df1=df.drop(df[df.percentage<Lower_limit].index)
```

[48]:
```
df1.shape
```

[48]:
```
(50, 12)
```

[51]:
```
df2=df[df.percentage<Lower_limit]
```

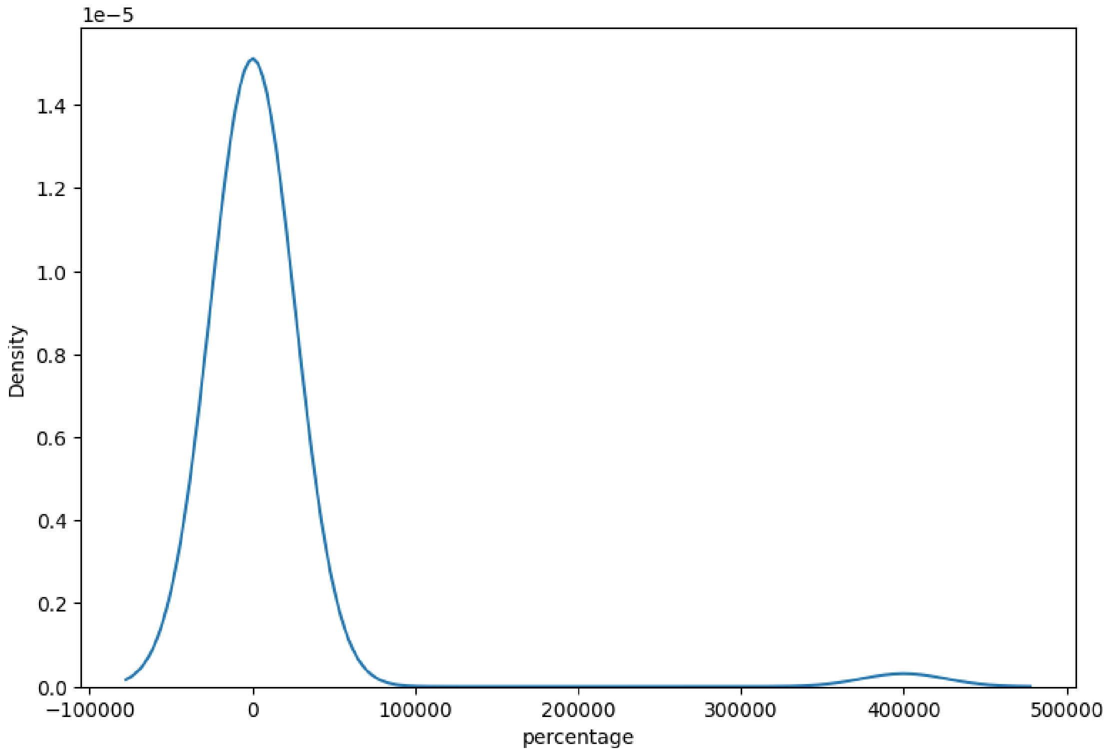
[52]:
```
df2
```

[52]:
```
Empty DataFrame
Columns: [sr, rollno, term, attendance, s1, s2, s3, s4, s5, totalmarks,
percentage, result]
Index: []
```

2.Mean/Median imputation

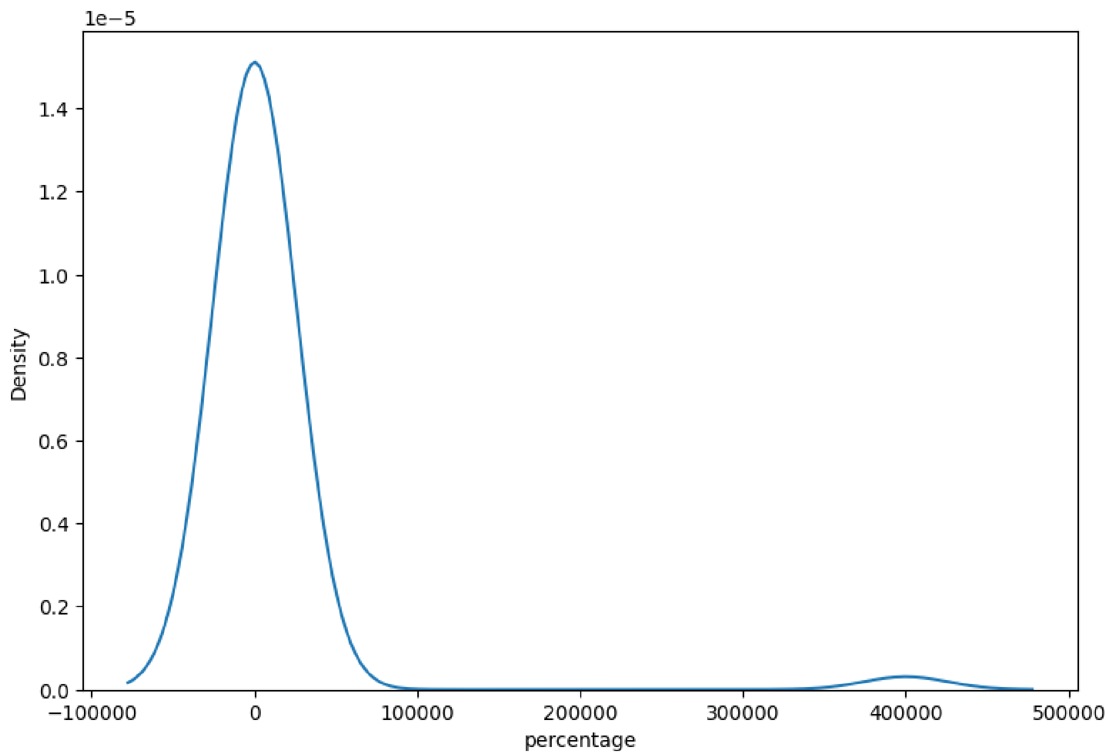
[53]:
```
sns.kdeplot(df.percentage)
```

[53]: <Axes: xlabel='percentage', ylabel='Density'>



[54]:
```
sns.kdeplot(df1.percentage)
```

[54]: <Axes: xlabel='percentage', ylabel='Density'>



[55]:
```
df.percentage
```

[55]:
```
0        32.6
1        47.0
2        29.6
3        34.0
4        46.0
5        51.4
6        44.2
7        50.6
8        51.2
9        50.8
10     4000.0
11       20.8
12       32.6
13       49.2
14       48.6
15       51.6
16      300.0
17       59.4
18       39.6
19      300.0
20       55.2
```

```
21         37.8
22         49.8
23         58.6
24         55.0
25         41.2
26       3022.0
27         49.4
28         48.4
29         38.4
30         32.2
31         57.6
32         45.2
33     400000.0
34         40.2
35         39.6
36         56.0
37         61.2
38         29.8
39         51.6
40         29.6
41         58.2
42         59.4
43         43.8
44         43.2
45         60.8
46         47.4
47         27.6
48         48.0
49         55.8
Name: percentage, dtype: float64
```

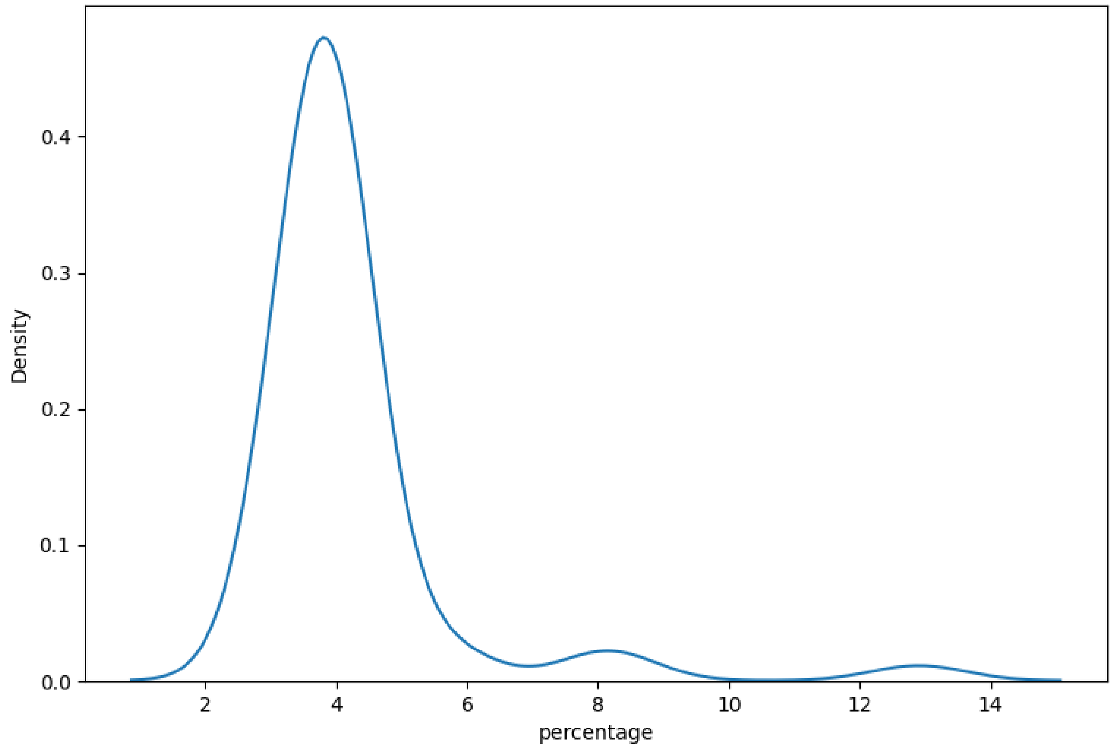
[56]:
```
log_percentage=np.log(df.percentage)
 log_percentage
```

[56]:
```
0      3.484312
1      3.850148
2      3.387774
3      3.526361
4      3.828641
5      3.939638
6      3.788725
7      3.923952
8      3.935740
9      3.927896
10     8.294050
11     3.034953
12     3.484312
```

```
13     3.895894
14     3.883624
15     3.943522
16     5.703782
17     4.084294
18     3.678829
19     5.703782
20     4.010963
21     3.632309
22     3.908015
23     4.070735
24     4.007333
25     3.718438
26     8.013674
27     3.899950
28     3.879500
29     3.648057
30     3.471966
31     4.053523
32     3.811097
33    12.899220
34     3.693867
35     3.678829
36     4.025352
37     4.114147
38     3.394508
39     3.943522
40     3.387774
41     4.063885
42     4.084294
43     3.779634
44     3.765840
45     4.107590
46     3.858622
47     3.317816
48     3.871201
49     4.021774
Name: percentage, dtype: float64
```

[57]:
```
sns.kdeplot(log_percentage)
```

[57]:
```
<Axes: xlabel='percentage', ylabel='Density'>
```



1.3 C. Data Transformation

To change the scale for better understanding of the variable

[58]:
```
import seaborn as sns
```

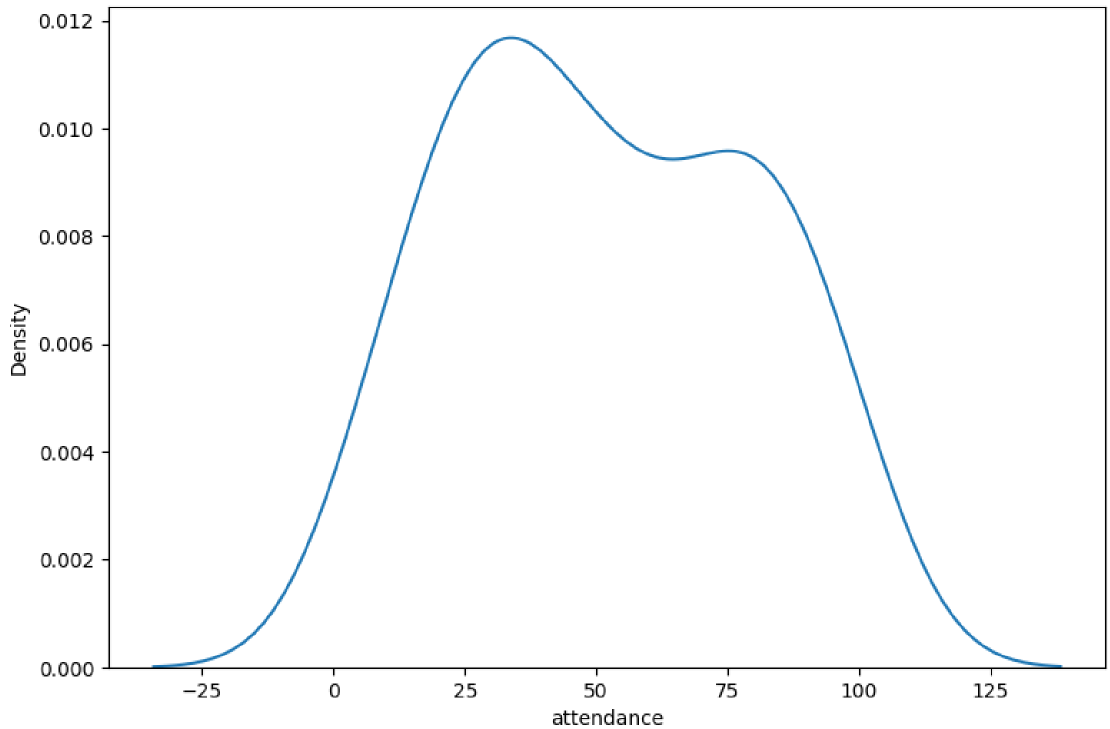
[63]:
```
#skewness in the data
df = df.apply(pd.to_numeric, errors='coerce')
skewness = df.skew()

print(skewness)
```

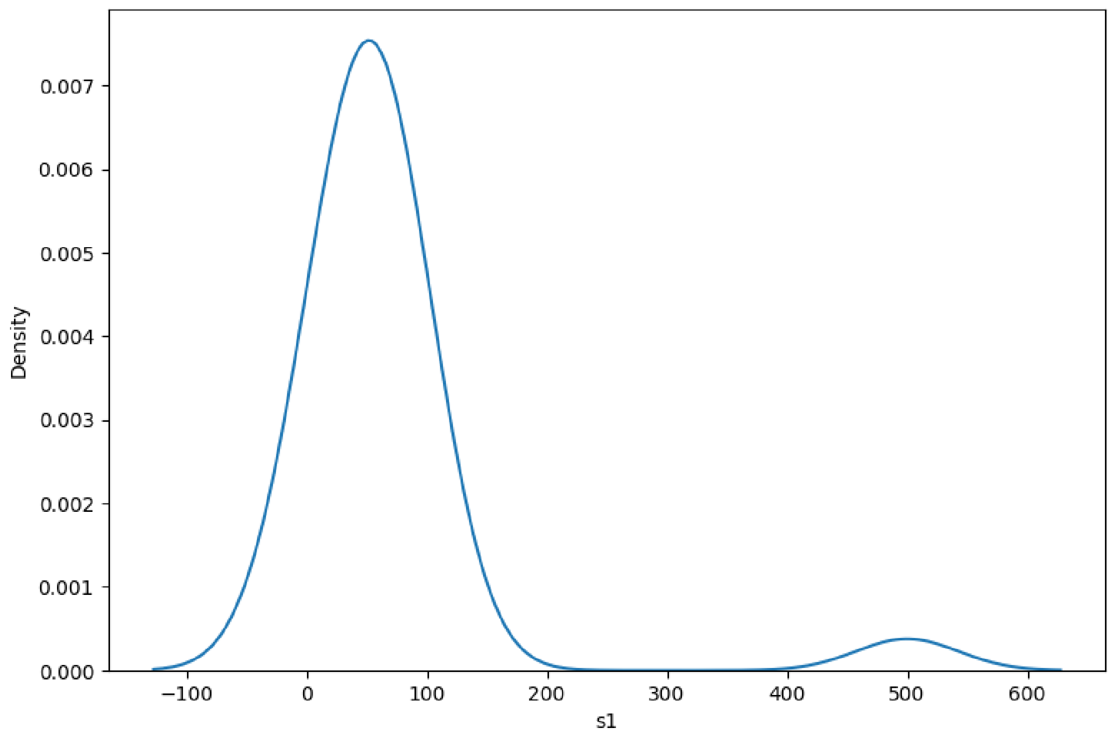
```
sr             0.000000
rollno         0.000000
term                NaN
attendance     0.116953
s1             4.201498
s2             0.100380
s3             3.905146
s4             0.103554
s5             5.953471
totalmarks    -0.069495
percentage     7.069420
```

```
result                NaN
dtype: float64
```

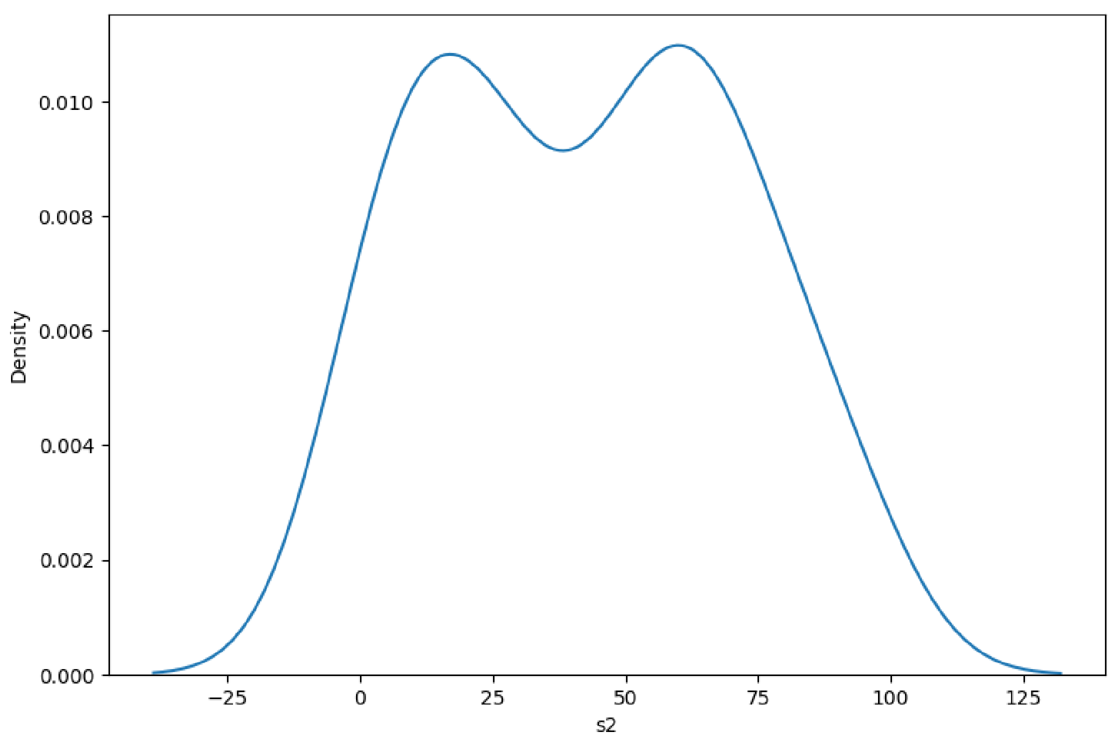
[64]:
```
sns.kdeplot(df.attendance);
```



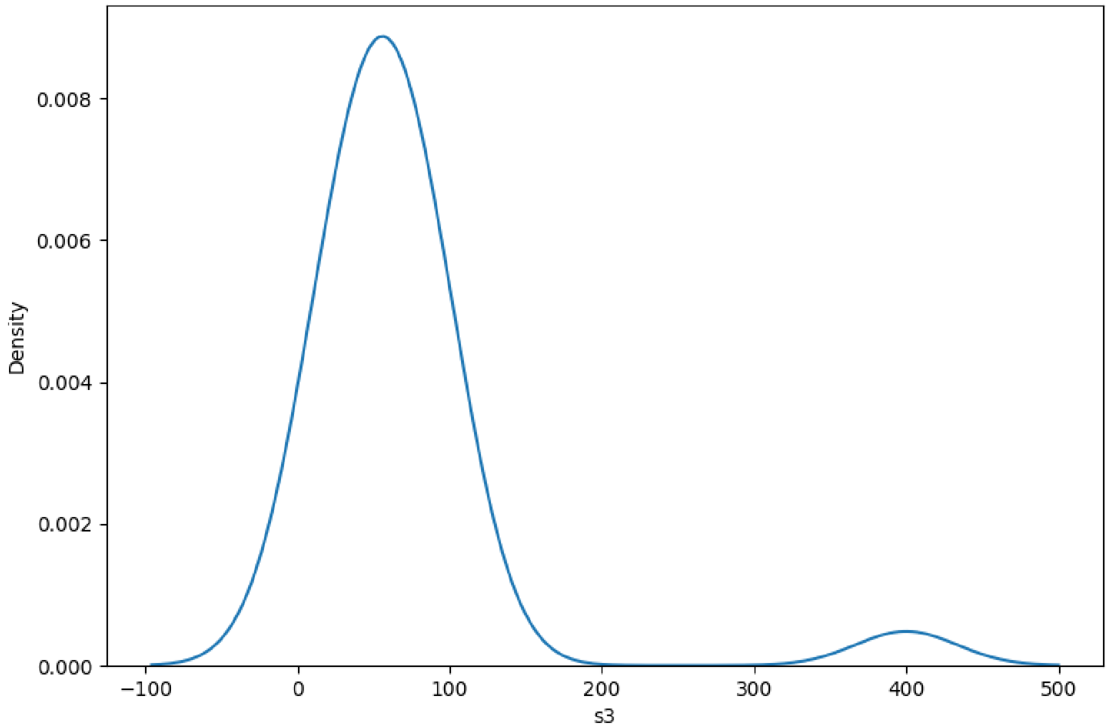
[65]:
```
sns.kdeplot(df.s1);
```



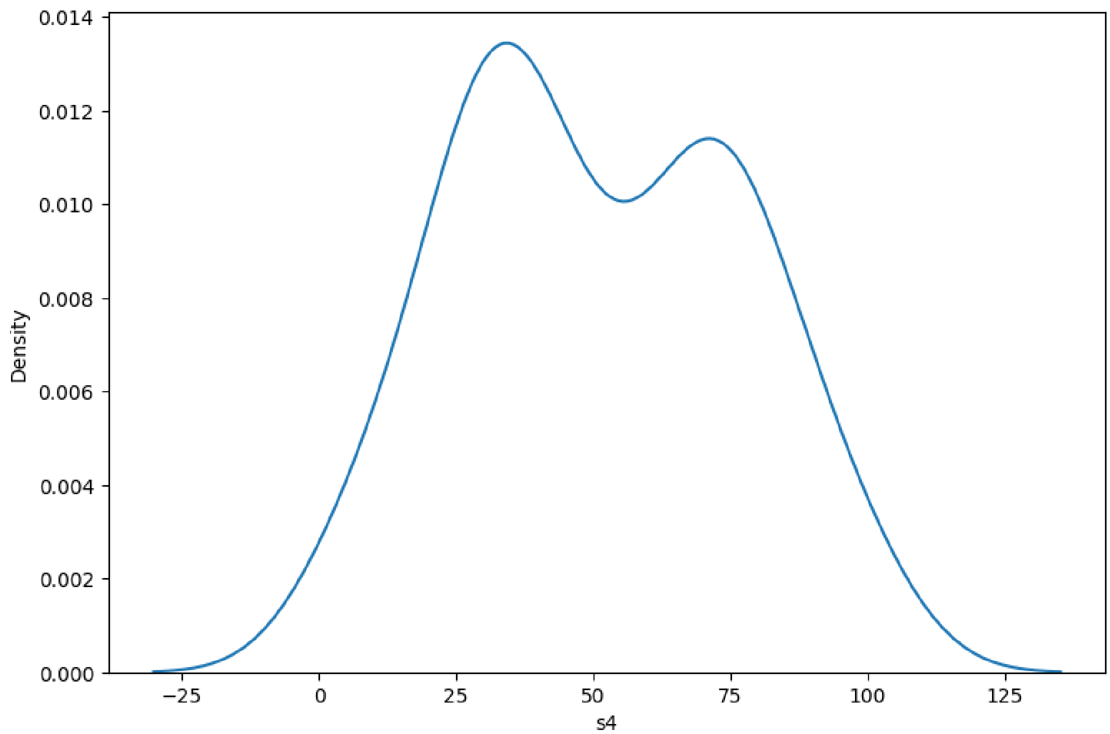
[66]:
```
sns.kdeplot(df.s2);
```



[67]:
```
sns.kdeplot(df.s3);
```

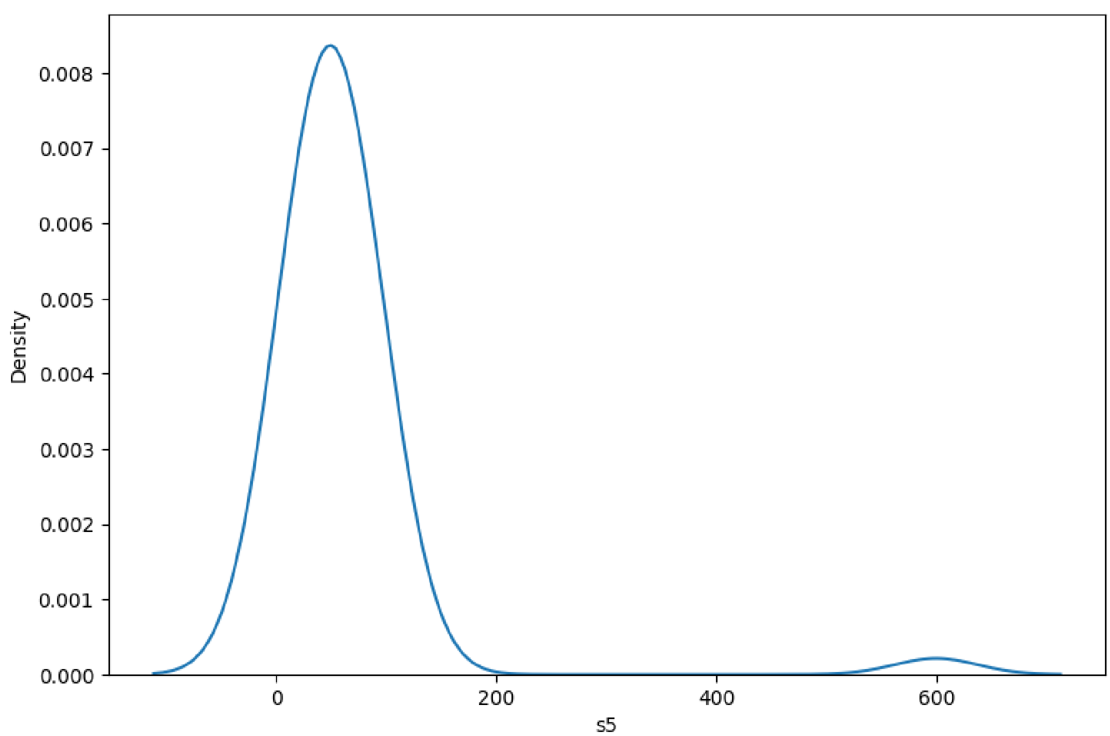


[68]:
```
sns.kdeplot(df.s4);
```

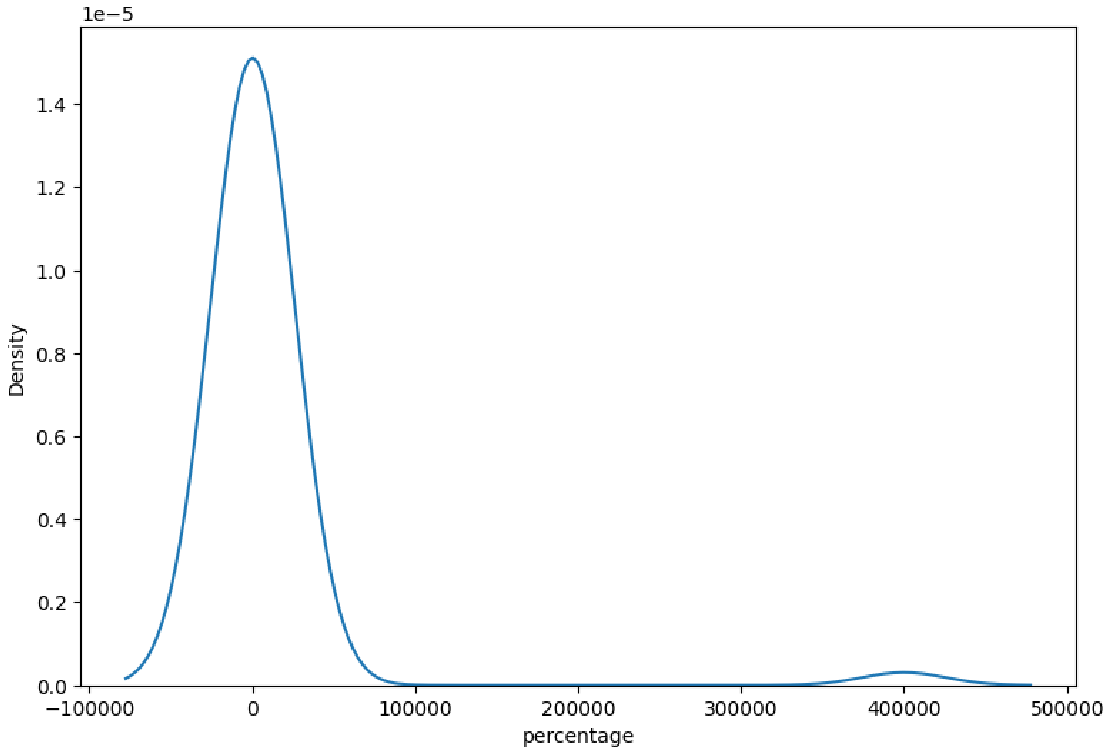


[69]:

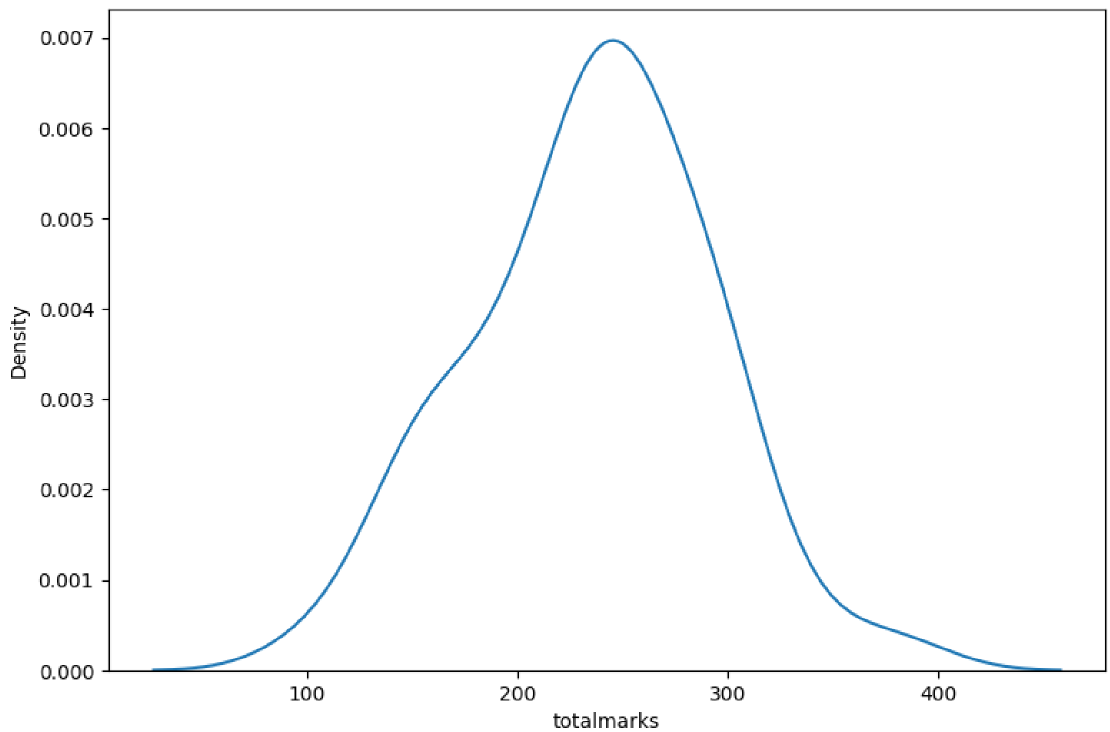
```
sns.kdeplot(df.s5);
```



[70]:
```
sns.kdeplot(df.percentage);
```



[71]:
```
sns.kdeplot(df.totalmarks);
```



1.4 Conclusion

In this way we have explored the functions of the python library for Data Preprocessing, Data Wrangling Techniques and How to Handle missing values and outliers also applied data transformation. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set