practice-assignment-2

May 4, 2025

[1]:
```
import pandas as pd
```

[5]:
```
df=pd.read_csv("Performance.csv")
df
```

[5]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total
0	1	Varad	90.0	76.0	31.0	45.0	242
1	2	NaN	94.0	85.0	37.0	40.0	256
2	3	NaN	55.0	52.0	75.0	61.0	243
3	4	NaN	76.0	66.0	65.0	NaN	207
4	5	NaN	NaN	73.0	22.0	46.0	141
5	6	NaN	34.0	74.0	NaN	71.0	179
6	7	NaN	47.0	48.0	92.0	95.0	282
7	8	NaN	81.0	91.0	11.0	30.0	213
8	9	NaN	40.0	36.0	17.0	88.0	181
9	10	NaN	86.0	77.0	75.0	78.0	316
10	11	NaN	38.0	47.0	52.0	69.0	206
11	12	NaN	82.0	69.0	94.0	69.0	314
12	13	NaN	94.0	82.0	11.0	81.0	268
13	14	NaN	36.0	60.0	28.0	NaN	124
14	15	NaN	30.0	86.0	69.0	30.0	215
15	16	NaN	70.0	44.0	76.0	68.0	258
16	17	NaN	94.0	62.0	NaN	64.0	220
17	18	NaN	NaN	97.0	52.0	85.0	234
18	19	NaN	83.0	71.0	33.0	35.0	222
19	20	NaN	36.0	90.0	35.0	93.0	254
20	21	NaN	98.0	44.0	44.0	NaN	186
21	22	NaN	52.0	40.0	90.0	62.0	244
22	23	NaN	77.0	36.0	11.0	40.0	164
23	24	NaN	90.0	NaN	13.0	49.0	152
24	25	NaN	NaN	38.0	15.0	88.0	141
25	26	NaN	81.0	44.0	65.0	97.0	287
26	27	NaN	94.0	32.0	61.0	54.0	241
27	28	NaN	77.0	30.0	38.0	37.0	182
28	29	NaN	95.0	85.0	52.0	88.0	320
29	30	NaN	46.0	49.0	83.0	84.0	262

```
    Percentage Result
0        60.50   Pass
1        64.00   Pass
2        60.75   Pass
3        51.75   Pass
4        35.25   Pass
5        44.75   Fail
6        70.50   Pass
7        53.25   Pass
8        45.25   Pass
9        79.00   Pass
10       51.50   Pass
11       78.50   Pass
12       67.00   Pass
13       31.00   Pass
14       53.75   Fail
15       64.50   Pass
16       55.00   Pass
17       58.50   Pass
18       55.50   Pass
19       63.50   Pass
20       46.50   Pass
21       61.00   Pass
22       41.00   Pass
23       38.00   Pass
24       35.25   Fail
25       71.75   Fail
26       60.25   Pass
27       45.50   Pass
28       80.00   Pass
29       65.50   Pass
```

[7]:
```
print(df.head())
print(df.tail())
print(df.shape)
print(df.size)
```

```
   Roll_no   Name  DSBDA_marks  CC_marks  AI_marks  WT_marks  Total  \
0        1  Varad         90.0      76.0      31.0      45.0    242
1        2    NaN         94.0      85.0      37.0      40.0    256
2        3    NaN         55.0      52.0      75.0      61.0    243
3        4    NaN         76.0      66.0      65.0       NaN    207
4        5    NaN          NaN      73.0      22.0      46.0    141

   Percentage Result
0       60.50   Pass
1       64.00   Pass
```

```
2       60.75   Pass
3       51.75   Pass
4       35.25   Pass
    Roll_no Name  DSBDA_marks  CC_marks  AI_marks  WT_marks  Total  \
25       26  NaN         81.0      44.0      65.0      97.0    287
26       27  NaN         94.0      32.0      61.0      54.0    241
27       28  NaN         77.0      30.0      38.0      37.0    182
28       29  NaN         95.0      85.0      52.0      88.0    320
29       30  NaN         46.0      49.0      83.0      84.0    262

    Percentage Result
25       71.75   Fail
26       60.25   Pass
27       45.50   Pass
28       80.00   Pass
29       65.50   Pass
(30, 9)
270
```

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

[9]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total
0	False	False	False	False	False	False	False
1	False	True	False	False	False	False	False
2	False	True	False	False	False	False	False
3	False	True	False	False	False	True	False
4	False	True	True	False	False	False	False
5	False	True	False	False	True	False	False
6	False	True	False	False	False	False	False
7	False	True	False	False	False	False	False
8	False	True	False	False	False	False	False
9	False	True	False	False	False	False	False
10	False	True	False	False	False	False	False
11	False	True	False	False	False	False	False
12	False	True	False	False	False	False	False
13	False	True	False	False	False	True	False
14	False	True	False	False	False	False	False
15	False	True	False	False	False	False	False
16	False	True	False	False	True	False	False
17	False	True	True	False	False	False	False
18	False	True	False	False	False	False	False
19	False	True	False	False	False	False	False
20	False	True	False	False	False	True	False
21	False	True	False	False	False	False	False
22	False	True	False	False	False	False	False
23	False	True	False	True	False	False	False
24	False	True	True	False	False	False	False

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

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

[11]:
```
df.dtypes
```

[11]:
```
Roll_no          int64
Name            object
DSBDA_marks    float64
CC_marks       float64
AI_marks       float64
WT_marks       float64
Total            int64
```

```
Percentage     float64
Result          object
dtype: object
```

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

[13]:
```
Index(['Roll_no', 'Name', 'DSBDA_marks', 'CC_marks', 'AI_marks', 'WT_marks',
       'Total', 'Percentage', 'Result'],
      dtype='object')
```

[15]:
```
df[0:5]
```

[15]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	Percentage	Result
0	1	Varad	90.0	76.0	31.0	45.0	242	60.50	Pass
1	2	NaN	94.0	85.0	37.0	40.0	256	64.00	Pass
2	3	NaN	55.0	52.0	75.0	61.0	243	60.75	Pass
3	4	NaN	76.0	66.0	65.0	NaN	207	51.75	Pass
4	5	NaN	NaN	73.0	22.0	46.0	141	35.25	Pass

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

[17]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	Percentage	Result
0	1	Varad	90.0	76.0	31.0	45.0	242	60.50	Pass
1	2	NaN	94.0	85.0	37.0	40.0	256	64.00	Pass
2	3	NaN	55.0	52.0	75.0	61.0	243	60.75	Pass

[19]:
```
df.loc[0:2,'DSBDA_marks':'WT_marks']
```

[19]:

	DSBDA_marks	CC_marks	AI_marks	WT_marks
0	90.0	76.0	31.0	45.0
1	94.0	85.0	37.0	40.0
2	55.0	52.0	75.0	61.0

[21]:
```
df.iloc[1:3]
```

[21]:
```
   Roll_no Name  DSBDA_marks  CC_marks  AI_marks  WT_marks  Total  Percentage  \
1        2  NaN         94.0      85.0      37.0      40.0    256       64.00
2        3  NaN         55.0      52.0      75.0      61.0    243       60.75

  Result
1   Pass
2   Pass
```

[23]:
```
df.iloc[1:5,1:5]
```

[23]:
```
  Name  DSBDA_marks  CC_marks  AI_marks
1  NaN         94.0      85.0      37.0
2  NaN         55.0      52.0      75.0
3  NaN         76.0      66.0      65.0
4  NaN          NaN      73.0      22.0
```

[25]:
```
print(df.isnull().any())
print(df.isnull().sum())
```

```
Roll_no        False
Name            True
DSBDA_marks     True
CC_marks        True
AI_marks        True
WT_marks        True
Total          False
Percentage     False
Result         False
dtype: bool
Roll_no         0
Name           29
DSBDA_marks     3
CC_marks        1
AI_marks        2
WT_marks        3
Total           0
Percentage      0
Result          0
dtype: int64
```

[27]:
```
vmm = []
for col in df.columns:
    if df[col].isna().any():
        vmm.append(col)
vmm
```

[27]:
```
['Name', 'DSBDA_marks', 'CC_marks', 'AI_marks', 'WT_marks']
```

[29]:
```
import numpy as np
df.replace(np.nan,value=0)
```

[29]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	Percentage	Result
0	1	Varad	90.0	76.0	31.0	45.0	242	60.50	Pass
1	2	0	94.0	85.0	37.0	40.0	256	64.00	Pass
2	3	0	55.0	52.0	75.0	61.0	243	60.75	Pass
3	4	0	76.0	66.0	65.0	0.0	207	51.75	Pass
4	5	0	0.0	73.0	22.0	46.0	141	35.25	Pass
5	6	0	34.0	74.0	0.0	71.0	179	44.75	Fail
6	7	0	47.0	48.0	92.0	95.0	282	70.50	Pass
7	8	0	81.0	91.0	11.0	30.0	213	53.25	Pass
8	9	0	40.0	36.0	17.0	88.0	181	45.25	Pass
9	10	0	86.0	77.0	75.0	78.0	316	79.00	Pass
10	11	0	38.0	47.0	52.0	69.0	206	51.50	Pass
11	12	0	82.0	69.0	94.0	69.0	314		
12	13	0	94.0	82.0	11.0	81.0	268		
13	14	0	36.0	60.0	28.0	0.0	124		
14	15	0	30.0	86.0	69.0	30.0	215		
15	16	0	70.0	44.0	76.0	68.0	258		
16	17	0	94.0	62.0	0.0	64.0	220		
17	18	0	0.0	97.0	52.0	85.0	234		
18	19	0	83.0	71.0	33.0	35.0	222		
19	20	0	36.0	90.0	35.0	93.0	254		
20	21	0	98.0	44.0	44.0	0.0	186		
21	22	0	52.0	40.0	90.0	62.0	244		
22	23	0	77.0	36.0	11.0	40.0	164		
23	24	0	90.0	0.0	13.0	49.0	152		
24	25	0	0.0	38.0	15.0	88.0	141		
25	26	0	81.0	44.0	65.0	97.0	287		
26	27	0	94.0	32.0	61.0	54.0	241		
27	28	0	77.0	30.0	38.0	37.0	182		
28	29	0	95.0	85.0	52.0	88.0	320		
29	30	0	46.0	49.0	83.0	84.0	262		

```
11       78.50   Pass
12       67.00   Pass
13       31.00   Pass
14       53.75   Fail
15       64.50   Pass
16       55.00   Pass
17       58.50   Pass
18       55.50   Pass
19       63.50   Pass
20       46.50   Pass
21       61.00   Pass
22       41.00   Pass
23       38.00   Pass
24       35.25   Fail
25       71.75   Fail
26       60.25   Pass
27       45.50   Pass
28       80.00   Pass
29       65.50   Pass
```

[31]:
```
df.fillna(1)
```

[31]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	\
0	1	Varad	90.0	76.0	31.0	45.0	242	
1	2	1	94.0	85.0	37.0	40.0	256	
2	3	1	55.0	52.0	75.0	61.0	243	
3	4	1	76.0	66.0	65.0	1.0	207	
4	5	1	1.0	73.0	22.0	46.0	141	
5	6	1	34.0	74.0	1.0	71.0	179	
6	7	1	47.0	48.0	92.0	95.0	282	
7	8	1	81.0	91.0	11.0	30.0	213	
8	9	1	40.0	36.0	17.0	88.0	181	
9	10	1	86.0	77.0	75.0	78.0	316	
10	11	1	38.0	47.0	52.0	69.0	206	
11	12	1	82.0	69.0	94.0	69.0	314	
12	13	1	94.0	82.0	11.0	81.0	268	
13	14	1	36.0	60.0	28.0	1.0	124	
14	15	1	30.0	86.0	69.0	30.0	215	
15	16	1	70.0	44.0	76.0	68.0	258	
16	17	1	94.0	62.0	1.0	64.0	220	
17	18	1	1.0	97.0	52.0	85.0	234	
18	19	1	83.0	71.0	33.0	35.0	222	
19	20	1	36.0	90.0	35.0	93.0	254	
20	21	1	98.0	44.0	44.0	1.0	186	
21	22	1	52.0	40.0	90.0	62.0	244	
22	23	1	77.0	36.0	11.0	40.0	164	
23	24	1	90.0	1.0	13.0	49.0	152	

```
24       25       1          1.0      38.0      15.0      88.0    141
25       26       1         81.0      44.0      65.0      97.0    287
26       27       1         94.0      32.0      61.0      54.0    241
27       28       1         77.0      30.0      38.0      37.0    182
28       29       1         95.0      85.0      52.0      88.0    320
29       30       1         46.0      49.0      83.0      84.0    262
```

```
    Percentage Result
0        60.50   Pass
1        64.00   Pass
2        60.75   Pass
3        51.75   Pass
4        35.25   Pass
5        44.75   Fail
6        70.50   Pass
7        53.25   Pass
8        45.25   Pass
9        79.00   Pass
10       51.50   Pass
11       78.50   Pass
12       67.00   Pass
13       31.00   Pass
14       53.75   Fail
15       64.50   Pass
16       55.00   Pass
17       58.50   Pass
18       55.50   Pass
19       63.50   Pass
20       46.50   Pass
21       61.00   Pass
22       41.00   Pass
23       38.00   Pass
24       35.25   Fail
25       71.75   Fail
26       60.25   Pass
27       45.50   Pass
28       80.00   Pass
29       65.50   Pass
```

[35]:
```
print(df['DSBDA_marks']==df['DSBDA_marks'].fillna(df['DSBDA_marks'].mean()))
print(df['CC_marks']==df['CC_marks'].fillna(df['CC_marks'].median()))
print(df['WT_marks']==df['WT_marks'].fillna(df['WT_marks'].mean()))
print(df['AI_marks']==df['AI_marks'].fillna(df['AI_marks'].median()))
```

```
0     True
1     True
2     True
```

```
3      True
4     False
5      True
6      True
7      True
8      True
9      True
10     True
11     True
12     True
13     True
14     True
15     True
16     True
17    False
18     True
19     True
20     True
21     True
22     True
23     True
24    False
25     True
26     True
27     True
28     True
29     True
Name: DSBDA_marks, dtype: bool
0      True
1      True
2      True
3      True
4      True
5      True
6      True
7      True
8      True
9      True
10     True
11     True
12     True
13     True
14     True
15     True
16     True
17     True
18     True
19     True
```

```
20     True
21     True
22     True
23    False
24     True
25     True
26     True
27     True
28     True
29     True
Name: CC_marks, dtype: bool
0      True
1      True
2      True
3     False
4      True
5      True
6      True
7      True
8      True
9      True
10     True
11     True
12     True
13    False
14     True
15     True
16     True
17     True
18     True
19     True
20    False
21     True
22     True
23     True
24     True
25     True
26     True
27     True
28     True
29     True
Name: WT_marks, dtype: bool
0      True
1      True
2      True
3      True
4      True
5     False
```

```
6      True
7      True
8      True
9      True
10     True
11     True
12     True
13     True
14     True
15     True
16    False
17     True
18     True
19     True
20     True
21     True
22     True
23     True
24     True
25     True
26     True
27     True
28     True
29     True
Name: AI_marks, dtype: bool
```

[37]:
```
df['Name']=df['Name'].fillna(df['Name'].mode())
```

[39]:
```
df
```

[39]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	\
0	1	Varad	90.0	76.0	31.0	45.0	242	
1	2	NaN	94.0	85.0	37.0	40.0	256	
2	3	NaN	55.0	52.0	75.0	61.0	243	
3	4	NaN	76.0	66.0	65.0	NaN	207	
4	5	NaN	NaN	73.0	22.0	46.0	141	
5	6	NaN	34.0	74.0	NaN	71.0	179	
6	7	NaN	47.0	48.0	92.0	95.0	282	
7	8	NaN	81.0	91.0	11.0	30.0	213	
8	9	NaN	40.0	36.0	17.0	88.0	181	
9	10	NaN	86.0	77.0	75.0	78.0	316	
10	11	NaN	38.0	47.0	52.0	69.0	206	
11	12	NaN	82.0	69.0	94.0	69.0	314	
12	13	NaN	94.0	82.0	11.0	81.0	268	
13	14	NaN	36.0	60.0	28.0	NaN	124	
14	15	NaN	30.0	86.0	69.0	30.0	215	
15	16	NaN	70.0	44.0	76.0	68.0	258	

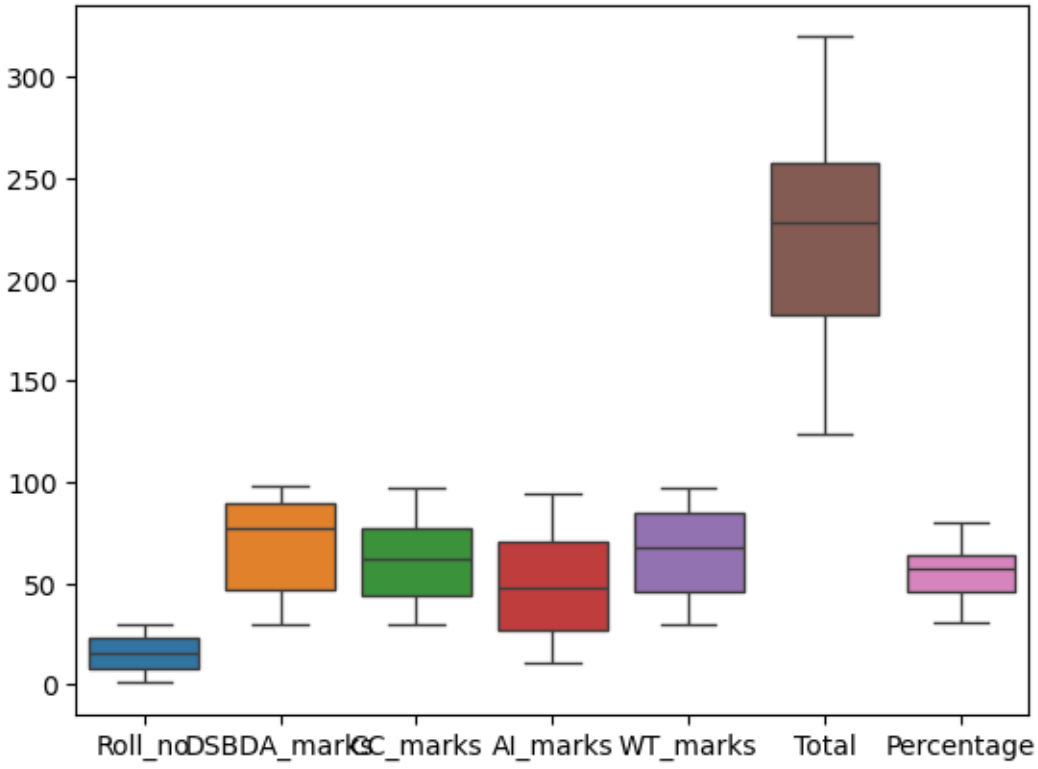
16	17	NaN	94.0	62.0	NaN	64.0	220
17	18	NaN	NaN	97.0	52.0	85.0	234
18	19	NaN	83.0	71.0	33.0	35.0	222
19	20	NaN	36.0	90.0	35.0	93.0	254
20	21	NaN	98.0	44.0	44.0	NaN	186
21	22	NaN	52.0	40.0	90.0	62.0	244
22	23	NaN	77.0	36.0	11.0	40.0	164
23	24	NaN	90.0	NaN	13.0	49.0	152
24	25	NaN	NaN	38.0	15.0	88.0	141
25	26	NaN	81.0	44.0	65.0	97.0	287
26	27	NaN	94.0	32.0	61.0	54.0	241
27	28	NaN	77.0	30.0	38.0	37.0	182
28	29	NaN	95.0	85.0	52.0	88.0	320
29	30	NaN	46.0	49.0	83.0	84.0	262

	Percentage	Result
0	60.50	Pass
1	64.00	Pass
2	60.75	Pass
3	51.75	Pass
4	35.25	Pass
5	44.75	Fail
6	70.50	Pass
7	53.25	Pass
8	45.25	Pass
9	79.00	Pass
10	51.50	Pass
11	78.50	Pass
12	67.00	Pass
13	31.00	Pass
14	53.75	Fail
15	64.50	Pass
16	55.00	Pass
17	58.50	Pass
18	55.50	Pass
19	63.50	Pass
20	46.50	Pass
21	61.00	Pass
22	41.00	Pass
23	38.00	Pass
24	35.25	Fail
25	71.75	Fail
26	60.25	Pass
27	45.50	Pass
28	80.00	Pass
29	65.50	Pass

[41]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

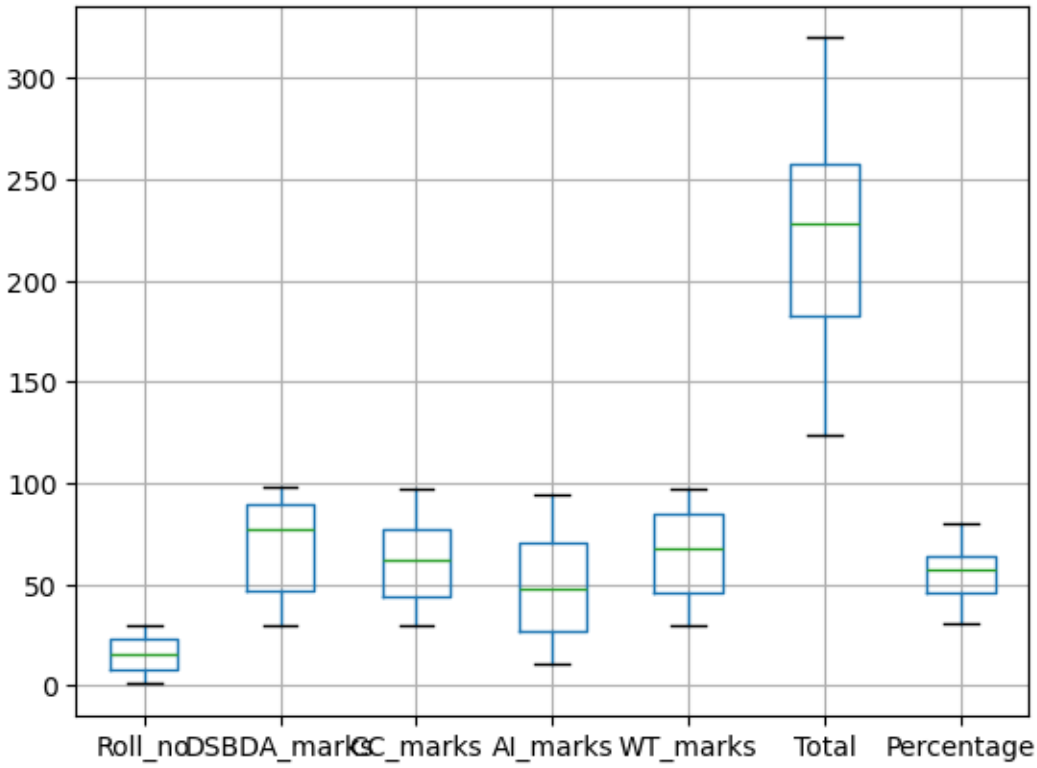
[43]:
```
sns.boxplot(df)
```

[43]: <Axes: >



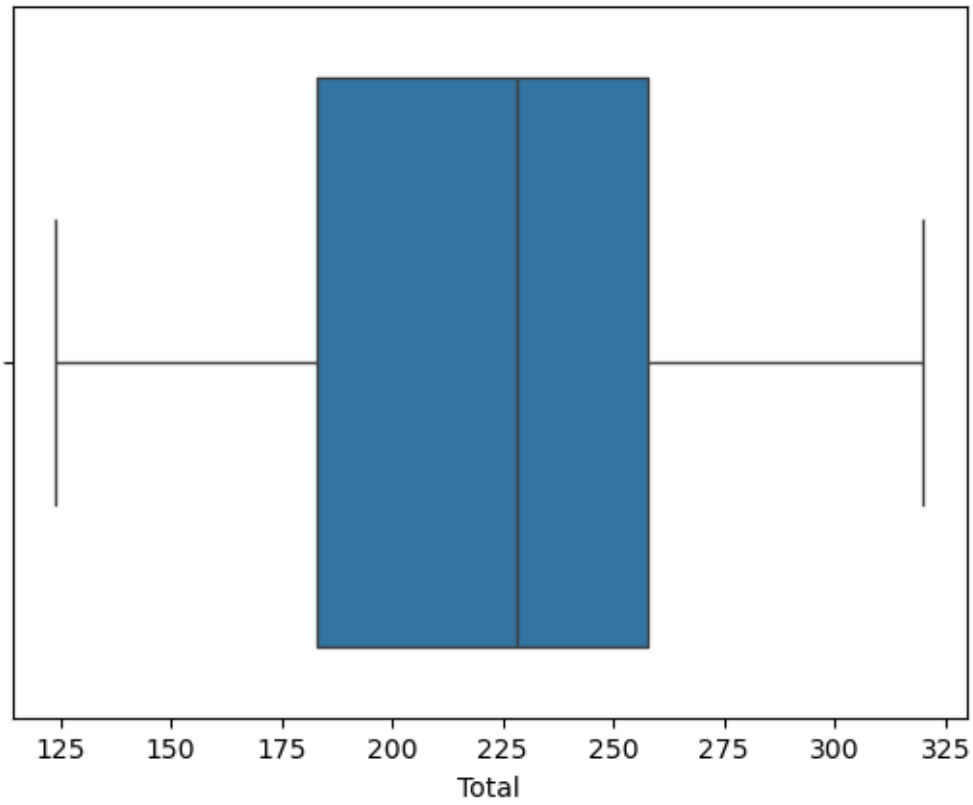
[45]:
```
df.boxplot()
```

[45]: <Axes: >



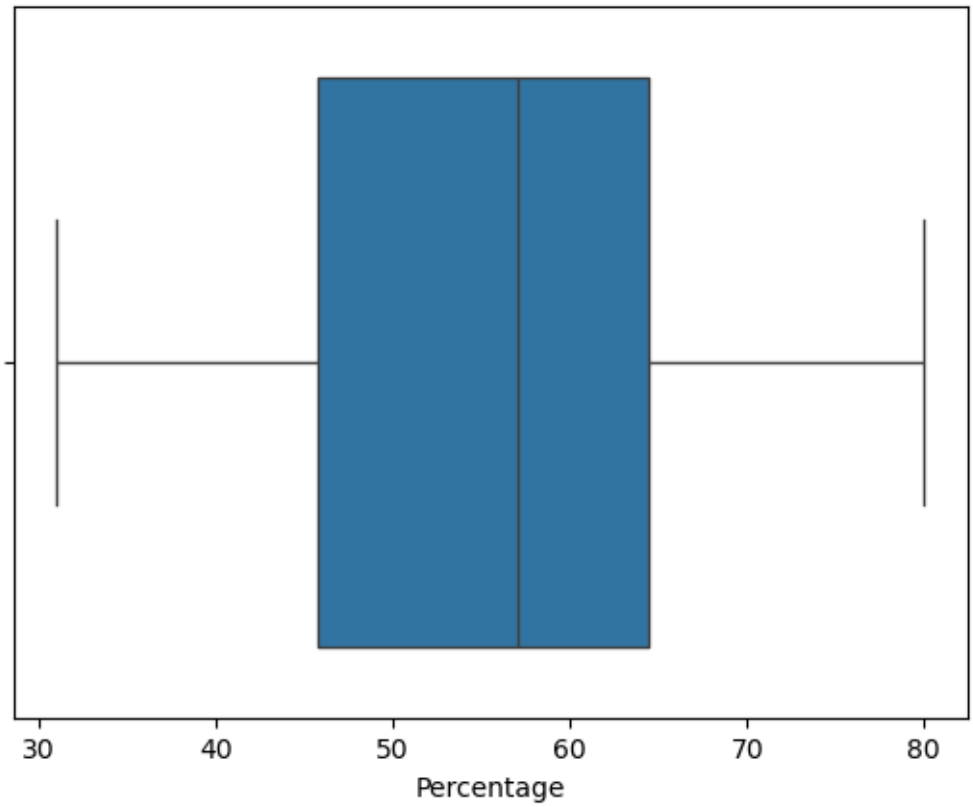
[47]:
```
sns.boxplot(x=df.Total)
```

[47]:
```
<Axes: xlabel='Total'>
```



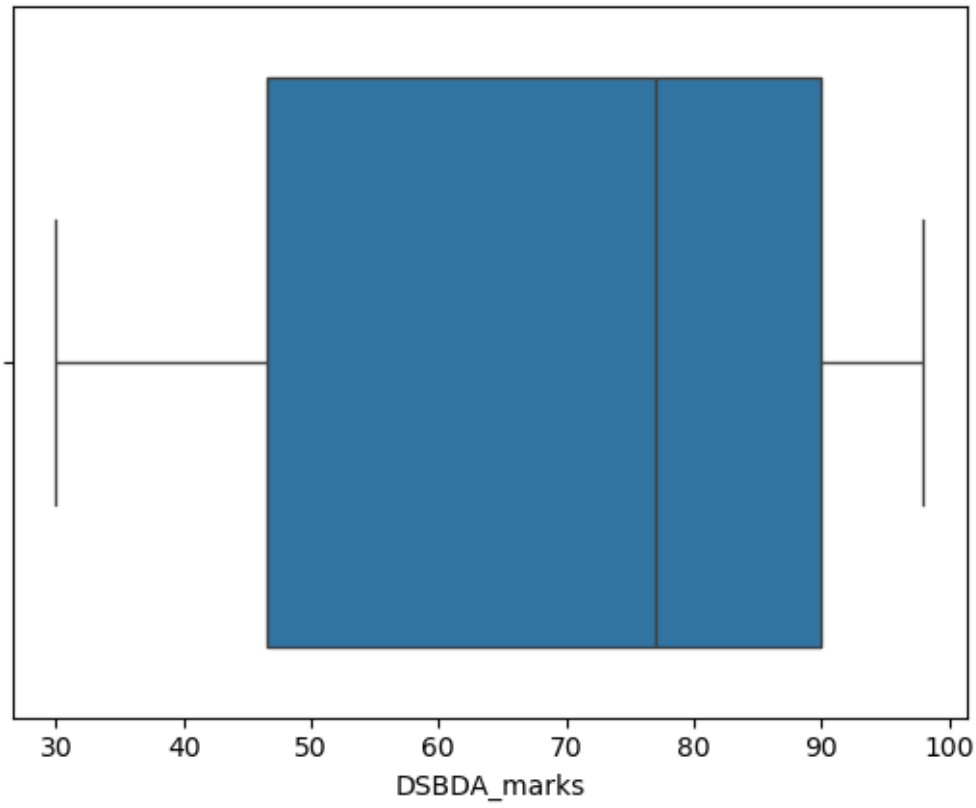
[49]:
```
sns.boxplot(x=df.Percentage)
```

[49]:
```
<Axes: xlabel='Percentage'>
```

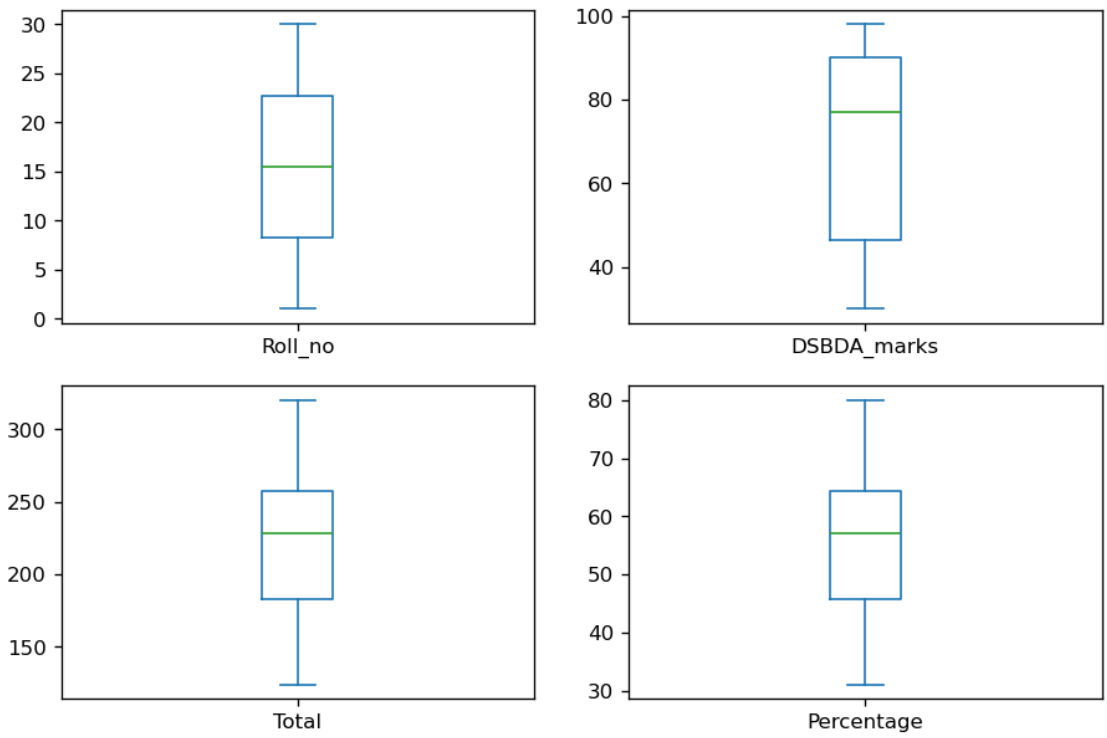


[51]:
```
sns.boxplot(x=df.DSBDA_marks)
```

[51]:
```
<Axes: xlabel='DSBDA_marks'>
```



[53]:
```
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (9, 6)
df_list = ['Roll_no','DSBDA_marks','Total','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
```



[55]:
```
#Interquartile range
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 = 45.75, Q3 = 64.375, IQR = 18.625, Lower_limit = 17.8125, Upper_limit =
92.3125
```

[57]:
```
df[(df['Percentage'] < Lower_limit) | (df['Percentage'] > Upper_limit)] #␣
 ↪outlier data
```

[57]:
```
Empty DataFrame
Columns: [Roll_no, Name, DSBDA_marks, CC_marks, AI_marks, WT_marks, Total,
Percentage, Result]
Index: []
```

[59]:
```
import numpy as np
mean = np.mean(df.Total)
std = np.std(df.Total)
```

```
print('mean of the dataset is', mean)
print('std. deviation is', std)
```

```
mean of the dataset is 225.13333333333333
std. deviation is 52.117644698210306
```

[61]:
```
threshold = 3
outlier = []
for i in df.Total:
    z = (i-mean)/std
    if z > threshold:
        outlier.append(i)
print('outlier in dataset is', outlier)
```

```
outlier in dataset is []
```

[63]:
```
#Remove outlier
outliers=[]
for i in df.Percentage:
    if i<Lower_limit or i>Upper_limit:
        outliers.append(i)
print("outliers are",outliers)
```

```
outliers are []
```

[65]:
```
Upper_limit
```

[65]: 92.3125

[67]:
```
Lower_limit
```

[67]: 17.8125

[69]:
```
df[df.Percentage<Lower_limit].index
```

[69]: Index([], dtype='int64')

[71]:
```
df1=df.drop(df[df.Percentage<Lower_limit].index) #normal data without outlier
```

[73]:
```
df1.shape
```

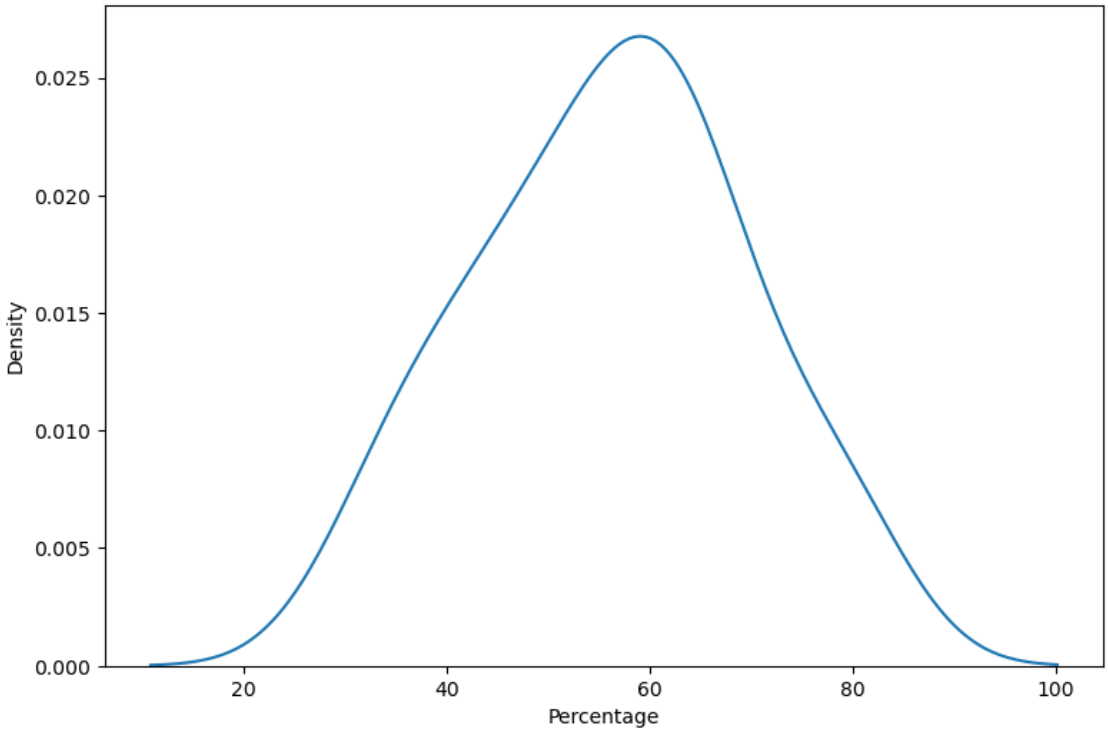
[73]: (30, 9)

[75]:
```
#outlier data
df2=df[df.Percentage<Lower_limit]
df2
```

[75]:
```
Empty DataFrame
Columns: [Roll_no, Name, DSBDA_marks, CC_marks, AI_marks, WT_marks, Total,
Percentage, Result]
Index: []
```

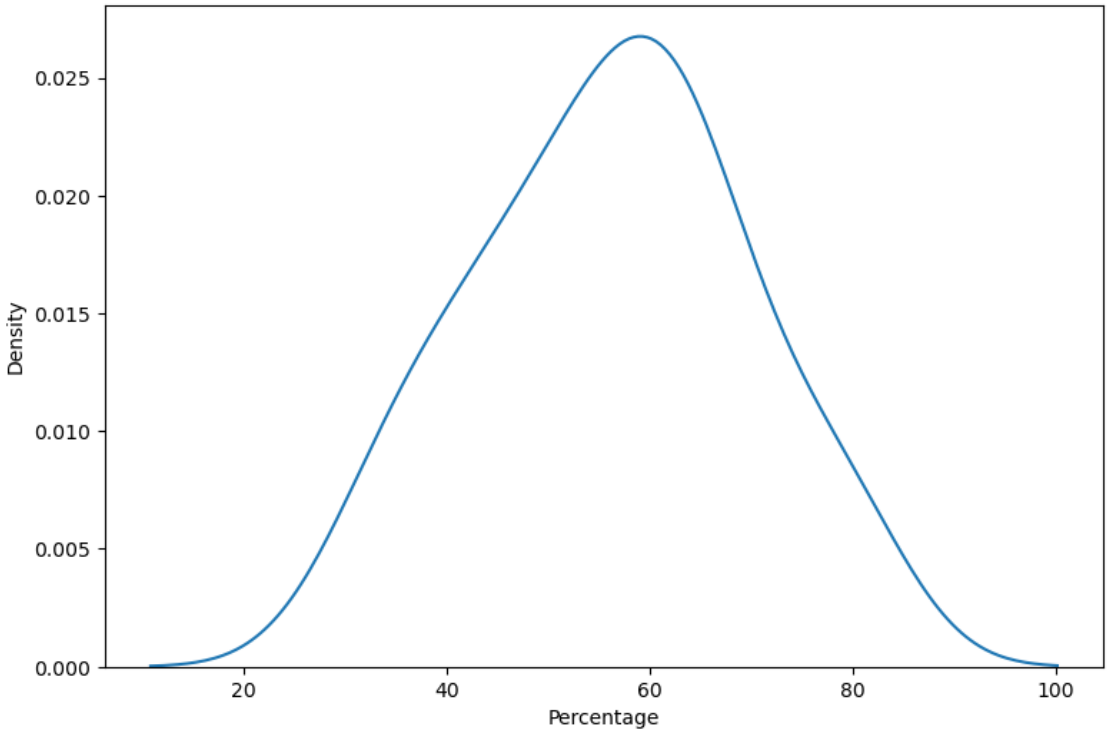
[77]:
```
sns.kdeplot(df.Percentage)
```

[77]:
```
<Axes: xlabel='Percentage', ylabel='Density'>
```



[79]:
```
sns.kdeplot(df1.Percentage)
```

[79]:
```
<Axes: xlabel='Percentage', ylabel='Density'>
```



[81]:
```
df.Percentage
```

[81]:
```
0     60.50
1     64.00
2     60.75
3     51.75
4     35.25
5     44.75
6     70.50
7     53.25
8     45.25
9     79.00
10    51.50
11    78.50
12    67.00
13    31.00
14    53.75
15    64.50
16    55.00
17    58.50
18    55.50
19    63.50
20    46.50
```

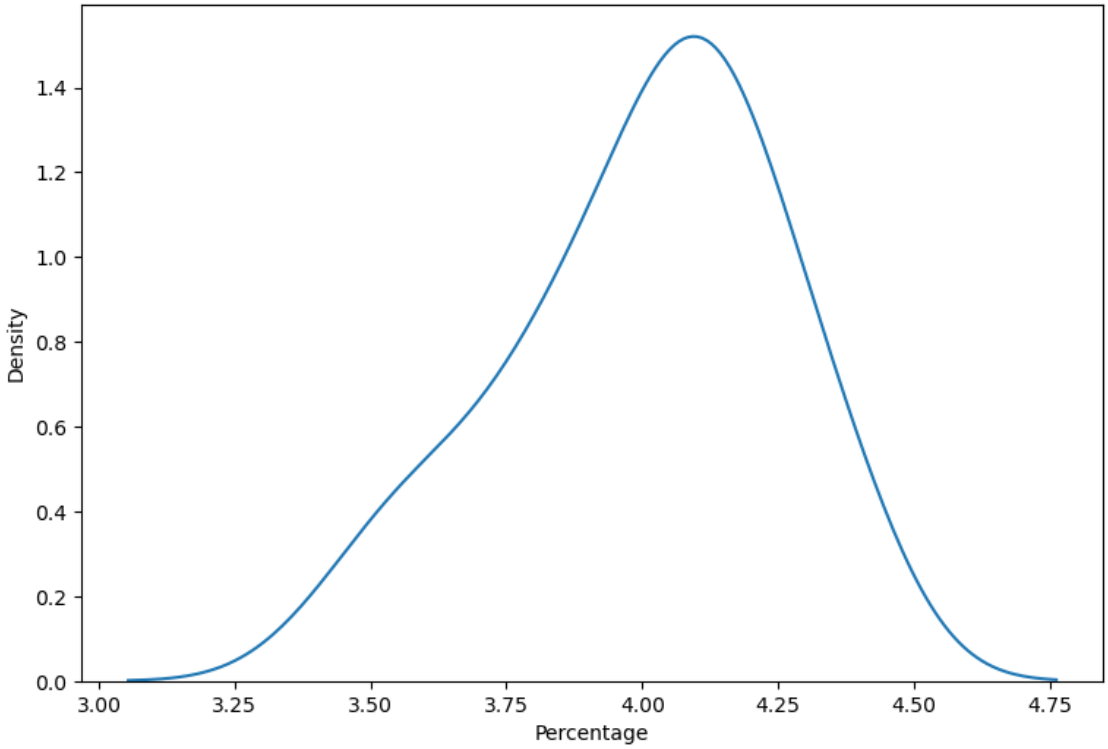
```
21    61.00
22    41.00
23    38.00
24    35.25
25    71.75
26    60.25
27    45.50
28    80.00
29    65.50
Name: Percentage, dtype: float64
```

[83]:
```
log_percentage=np.log(df.Percentage)
log_percentage
```

[83]:
```
0     4.102643
1     4.158883
2     4.106767
3     3.946424
4     3.562466
5     3.801091
6     4.255613
7     3.974998
8     3.812203
9     4.369448
10    3.941582
11    4.363099
12    4.204693
13    3.433987
14    3.984344
15    4.166665
16    4.007333
17    4.069027
18    4.016383
19    4.151040
20    3.839452
21    4.110874
22    3.713572
23    3.637586
24    3.562466
25    4.273188
26    4.098503
27    3.817712
28    4.382027
29    4.182050
Name: Percentage, dtype: float64
```

[85]:
```
sns.kdeplot(log_percentage)
```

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



[87]:
```
#Data Normalization
from sklearn import preprocessing # step1 :Import pandas and sklearn library
 ↪for preprocessing
df.head()
```

[87]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total	Percentage	Result
0	1	Varad	90.0	76.0	31.0	45.0	242	60.50	Pass
1	2	NaN	94.0	85.0	37.0	40.0	256	64.00	Pass
2	3	NaN	55.0	52.0	75.0	61.0	243	60.75	Pass
3	4	NaN	76.0	66.0	65.0	NaN	207	51.75	Pass
4	5	NaN	NaN	73.0	22.0	46.0	141	35.25	Pass

[89]:
```
min_max_scaler = preprocessing.MinMaxScaler() #min-max scalar
```

[91]:
```
x=df.iloc[1:,:7] #separte input from dataset
x
```

[91]:

	Roll_no	Name	DSBDA_marks	CC_marks	AI_marks	WT_marks	Total
1	2	NaN	94.0	85.0	37.0	40.0	256
2	3	NaN	55.0	52.0	75.0	61.0	243
3	4	NaN	76.0	66.0	65.0	NaN	207
4	5	NaN	NaN	73.0	22.0	46.0	141
5	6	NaN	34.0	74.0	NaN	71.0	179
6	7	NaN	47.0	48.0	92.0	95.0	282
7	8	NaN	81.0	91.0	11.0	30.0	213
8	9	NaN	40.0	36.0	17.0	88.0	181
9	10	NaN	86.0	77.0	75.0	78.0	316
10	11	NaN	38.0	47.0	52.0	69.0	206
11	12	NaN	82.0	69.0	94.0	69.0	314
12	13	NaN	94.0	82.0	11.0	81.0	268
13	14	NaN	36.0	60.0	28.0	NaN	124
14	15	NaN	30.0	86.0	69.0	30.0	215
15	16	NaN	70.0	44.0	76.0	68.0	258
16	17	NaN	94.0	62.0	NaN	64.0	220
17	18	NaN	NaN	97.0	52.0	85.0	234
18	19	NaN	83.0	71.0	33.0	35.0	222
19	20	NaN	36.0	90.0	35.0	93.0	254
20	21	NaN	98.0	44.0	44.0	NaN	186
21	22	NaN	52.0	40.0	90.0	62.0	244
22	23	NaN	77.0	36.0	11.0	40.0	164
23	24	NaN	90.0	NaN	13.0	49.0	152
24	25	NaN	NaN	38.0	15.0	88.0	141
25	26	NaN	81.0	44.0	65.0	97.0	287
26	27	NaN	94.0	32.0	61.0	54.0	241
27	28	NaN	77.0	30.0	38.0	37.0	182
28	29	NaN	95.0	85.0	52.0	88.0	320
29	30	NaN	46.0	49.0	83.0	84.0	262

[93]:
```
x_scaled = min_max_scaler.fit_transform(x) # Create an object to transform the␣
 ↪data to fit minmax processor
df_normalized = pd.DataFrame(x_scaled) #normalized data
df_normalized
```

```
C:\Users\Varad\anaconda3\Lib\site-packages\sklearn\utils\_array_api.py:701:
RuntimeWarning: All-NaN slice encountered
  return xp.asarray(numpy.nanmin(X, axis=axis))
C:\Users\Varad\anaconda3\Lib\site-packages\sklearn\utils\_array_api.py:718:
RuntimeWarning: All-NaN slice encountered
  return xp.asarray(numpy.nanmax(X, axis=axis))
```

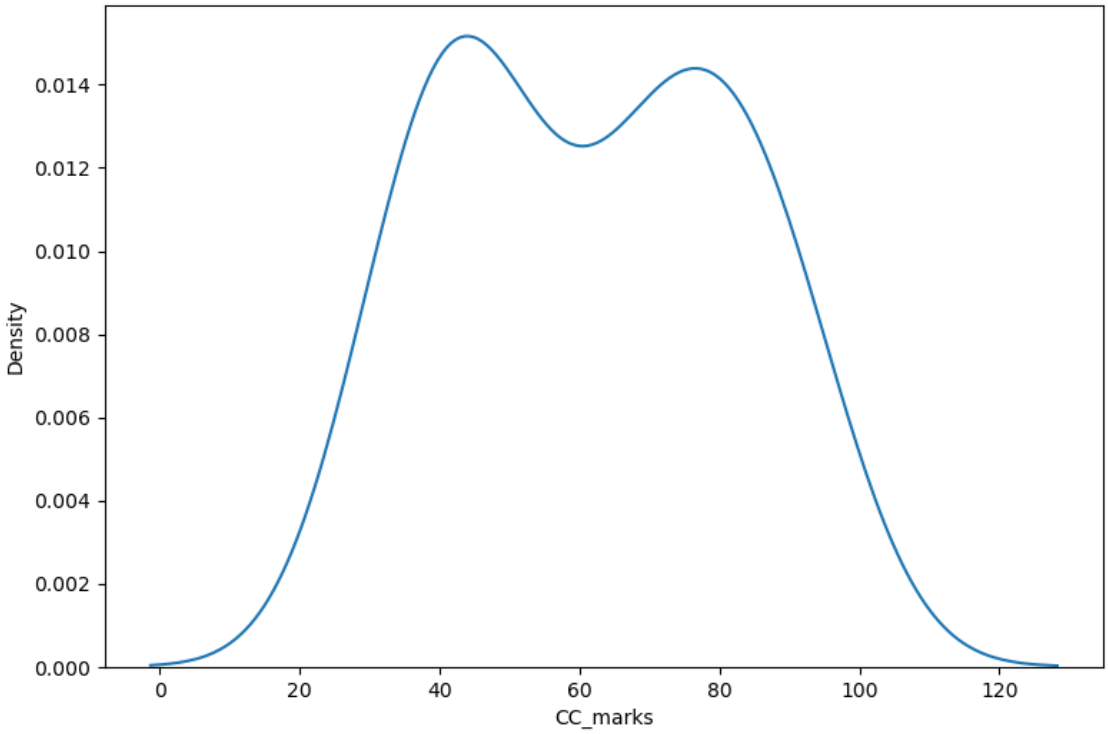
[93]:

	0	1	2	3	4	5	6
0	0.000000	NaN	0.941176	0.820896	0.313253	0.149254	0.673469
1	0.035714	NaN	0.367647	0.328358	0.771084	0.462687	0.607143
2	0.071429	NaN	0.676471	0.537313	0.650602	NaN	0.423469
3	0.107143	NaN	NaN	0.641791	0.132530	0.238806	0.086735
4	0.142857	NaN	0.058824	0.656716	NaN	0.611940	0.280612
5	0.178571	NaN	0.250000	0.268657	0.975904	0.970149	0.806122
6	0.214286	NaN	0.750000	0.910448	0.000000	0.000000	0.454082
7	0.250000	NaN	0.147059	0.089552	0.072289	0.865672	0.290816
8	0.285714	NaN	0.823529	0.701493	0.771084	0.716418	0.979592
9	0.321429	NaN	0.117647	0.253731	0.493976	0.582090	0.418367
10	0.357143	NaN	0.764706	0.582090	1.000000	0.582090	0.969388
11	0.392857	NaN	0.941176	0.776119	0.000000	0.761194	0.734694
12	0.428571	NaN	0.088235	0.447761	0.204819	NaN	0.000000
13	0.464286	NaN	0.000000	0.835821	0.698795	0.000000	0.464286
14	0.500000	NaN	0.588235	0.208955	0.783133	0.567164	0.683673
15	0.535714	NaN	0.941176	0.477612	NaN	0.507463	0.489796
16	0.571429	NaN	NaN	1.000000	0.493976	0.820896	0.561224
17	0.607143	NaN	0.779412	0.611940	0.265060	0.074627	0.500000
18	0.642857	NaN	0.088235	0.895522	0.289157	0.940299	0.663265
19	0.678571	NaN	1.000000	0.208955	0.397590	NaN	0.316327
20	0.714286	NaN	0.323529	0.149254	0.951807	0.477612	0.612245
21	0.750000	NaN	0.691176	0.089552	0.000000	0.149254	0.204082
22	0.785714	NaN	0.882353	NaN	0.024096	0.283582	0.142857
23	0.821429	NaN	NaN	0.119403	0.048193	0.865672	0.086735
24	0.857143	NaN	0.750000	0.208955	0.650602	1.000000	0.831633
25	0.892857	NaN	0.941176	0.029851	0.602410	0.358209	0.596939
26	0.928571	NaN	0.691176	0.000000	0.325301	0.104478	0.295918
27	0.964286	NaN	0.955882	0.820896	0.493976	0.865672	1.000000
28	1.000000	NaN	0.235294	0.283582	0.867470	0.805970	0.704082

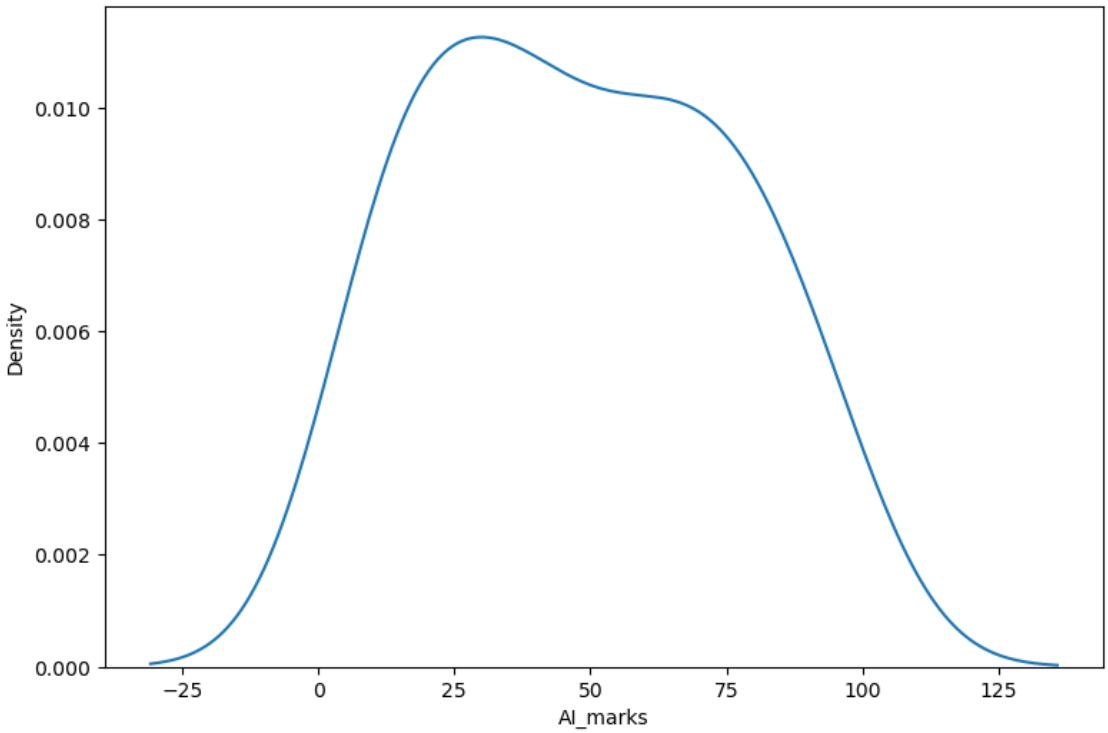
[95]:
```
import seaborn as sns
df_normalized.skew()
```

[95]:
```
0    0.000000
1         NaN
2   -0.392879
3    0.141873
4    0.089596
5   -0.234489
6   -0.020194
dtype: float64
```

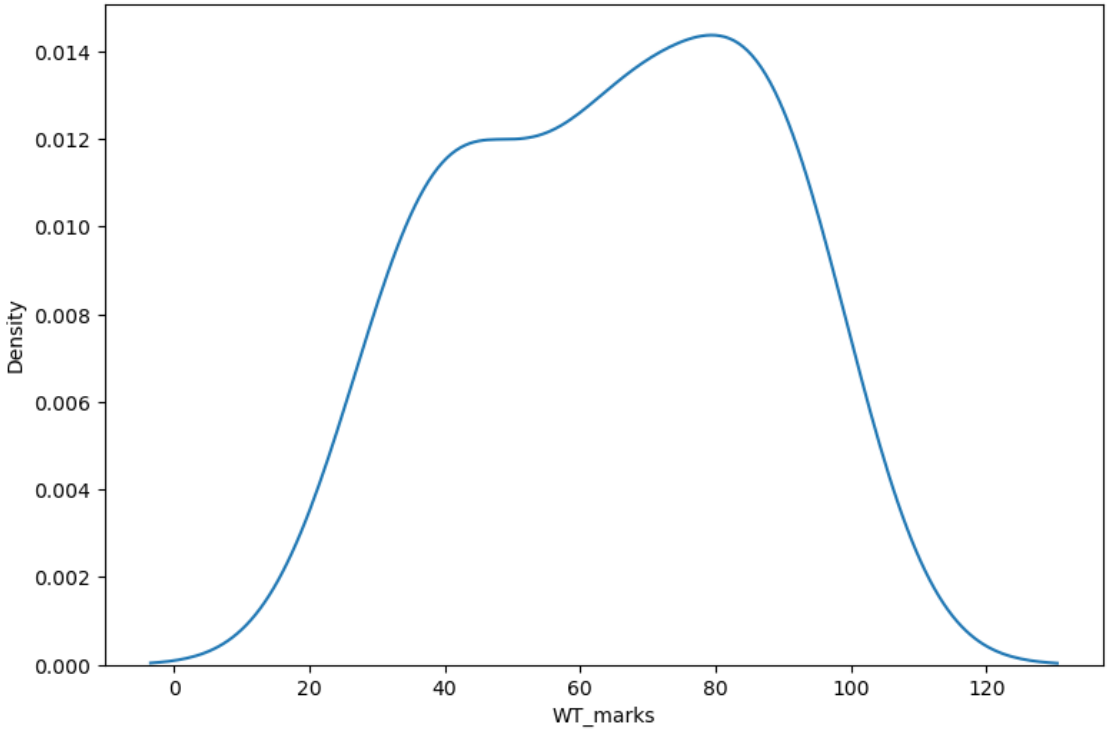
[97]:
```
sns.kdeplot(df.CC_marks);
```



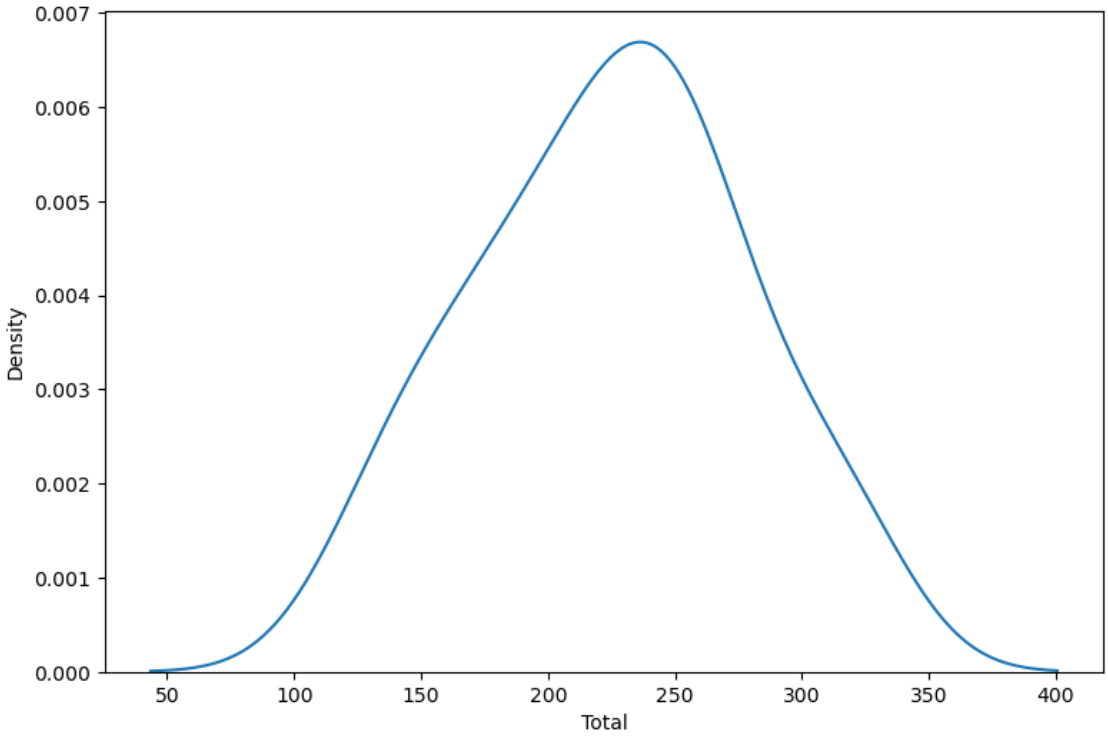
[99]:
```
sns.kdeplot(df.AI_marks);
```



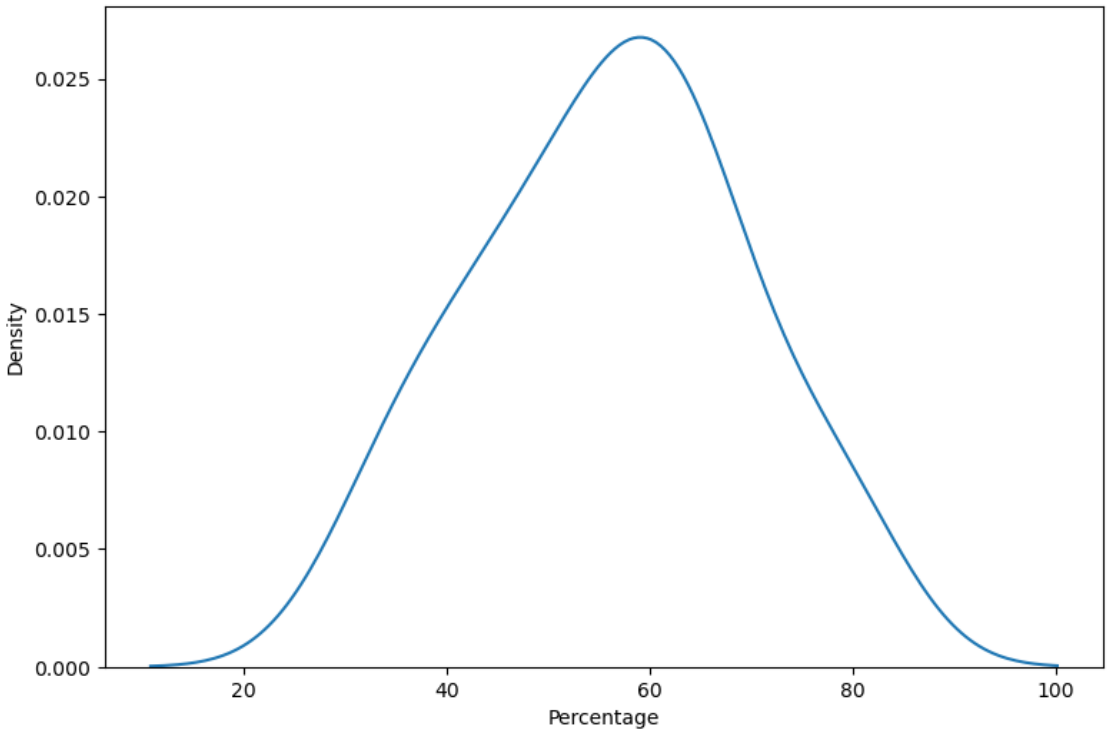
[101]:
```
sns.kdeplot(df.WT_marks);
```



[103]:
```
sns.kdeplot(df.Total);
```



[105]:
```
sns.kdeplot(df.Percentage);
```



[107]:
```
vm_ap_data={'Roll_num':[1,2,3,4,5],
            'Name':["Varad","Pratham","Sachin","Rohit","Rahul"],
            'DSBDA_Marks':[90,86,np.NaN,84,89],
            'AI_Marks':[85,80,90,np.NaN,69],
            'CC_Marks':[95,84,np.NaN,68,80],
            'WT_Marks':[89,80,64,np.NaN,69]}
vm_ap_data
```

[107]:
```
{'Roll_num': [1, 2, 3, 4, 5],
 'Name': ['Varad', 'Pratham', 'Sachin', 'Rohit', 'Rahul'],
 'DSBDA_Marks': [90, 86, nan, 84, 89],
 'AI_Marks': [85, 80, 90, nan, 69],
 'CC_Marks': [95, 84, nan, 68, 80],
 'WT_Marks': [89, 80, 64, nan, 69]}
```

[109]:
```
df=pd.DataFrame(vm_ap_data)
df
```

[109]:

	Roll_num	Name	DSBDA_Marks	AI_Marks	CC_Marks	WT_Marks
0	1	Varad	90.0	85.0	95.0	89.0
1	2	Pratham	86.0	80.0	84.0	80.0
2	3	Sachin	NaN	90.0	NaN	64.0
3	4	Rohit	84.0	NaN	68.0	NaN
4	5	Rahul	89.0	69.0	80.0	69.0

[111]:
```
df.columns
```

[111]:
```
Index(['Roll_num', 'Name', 'DSBDA_Marks', 'AI_Marks', 'CC_Marks', 'WT_Marks'],
dtype='object')
```

[ ]: