Practical No. 2

Aim : 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.

Code :

In [56]:
```
import pandas as pd

df1 = pd.read_csv("StudentsPerformance.csv")
df1
```

Out[56]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	NaN	68.0	85.0	2019.0	NaN
4	94.0	89.0	75.0	97.0	2020.0	2.0
5	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	76.0	45.0	96.0	2018.0	2.0
11	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	83.0	NaN	75.0	2067.0	1.0
15	NaN	75.0	75.0	78.0	2021.0	2.0
16	69.0	68.0	83.0	99.0	2020.0	2.0
17	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	85.0	75.0	99.0	2018.0	1.0
19	68.0	86.0	61.0	84.0	2021.0	2.0
20	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	82.0	65.0	77.0	2034.0	2.0
22	NaN	93.0	74.0	NaN	2018.0	1.0
23	79.0	88.0	30.0	76.0	NaN	2.0
24	71.0	86.0	69.0	96.0	2018.0	2.0
25	68.0	81.0	79.0	86.0	2018.0	2.0
26	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	80.0	NaN	83.0	2000.0	2.0
28	69.0	81.0	66.0	78.0	2019.0	2.0

In [57]:
```
df1.isnull()
```

Out[57]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	True	False	False	False	True
4	False	False	False	False	False	False
5	False	False	True	False	True	False
6	False	False	False	False	False	False
7	False	False	False	True	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	True
10	False	False	False	False	False	False
11	False	False	False	False	False	False
12	False	False	False	False	True	False
13	False	False	False	False	False	False
14	False	False	True	False	False	False
15	True	False	False	False	False	False
16	False	False	False	False	False	False
17	False	False	False	True	False	False
18	False	False	False	False	False	False
19	False	False	False	False	False	False
20	False	False	False	False	False	False
21	False	False	False	False	False	False
22	True	False	False	True	False	False
23	False	False	False	False	True	False
24	False	False	False	False	False	False
25	False	False	False	False	False	False
26	False	False	False	False	False	True
27	False	False	True	False	False	False
28	False	False	False	False	False	False

In [58]:
```
series = pd.isnull(df1["Math_Score"])
df1[series]
```

Out[58]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
15	NaN	75.0	75.0	78.0	2021.0	2.0
22	NaN	93.0	74.0	NaN	2018.0	1.0

In [59]:
```
1 df1.notnull()
```

Out[59]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	True	True	True	True	True	True
1	True	True	True	True	True	True
2	True	True	True	True	True	True
3	True	False	True	True	True	False
4	True	True	True	True	True	True
5	True	True	False	True	False	True
6	True	True	True	True	True	True
7	True	True	True	False	True	True
8	True	True	True	True	True	True
9	True	True	True	True	True	False
10	True	True	True	True	True	True
11	True	True	True	True	True	True
12	True	True	True	True	False	True
13	True	True	True	True	True	True
14	True	True	False	True	True	True
15	False	True	True	True	True	True
16	True	True	True	True	True	True
17	True	True	True	False	True	True
18	True	True	True	True	True	True
19	True	True	True	True	True	True
20	True	True	True	True	True	True
21	True	True	True	True	True	True
22	False	True	True	False	True	True
23	True	True	True	True	False	True
24	True	True	True	True	True	True
25	True	True	True	True	True	True
26	True	True	True	True	True	False
27	True	True	False	True	True	True
28	True	True	True	True	True	True

In [60]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df1['Writing_Score'] = le.fit_transform(df1['Writing_Score'])
newdf=df1
df1
```

Out[60]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	75.0	87.0	5	80.0	2018.0	2.0
1	63.0	88.0	16	76.0	2021.0	2.0
2	72.0	91.0	3	75.0	2020.0	2.0
3	85.0	NaN	8	85.0	2019.0	NaN
4	94.0	89.0	13	97.0	2020.0	2.0
5	74.0	82.0	17	94.0	NaN	2.0
6	61.0	87.0	7	86.0	2019.0	2.0
7	63.0	89.0	8	NaN	2019.0	2.0
8	78.0	78.0	4	83.0	2021.0	2.0
9	79.0	76.0	3	85.0	2025.0	NaN
10	80.0	76.0	1	96.0	2018.0	2.0
11	76.0	83.0	11	98.0	2021.0	2.0
12	62.0	91.0	10	100.0	NaN	2.0
13	67.0	81.0	6	91.0	2020.0	2.0
14	73.0	83.0	17	75.0	2067.0	1.0
15	NaN	75.0	13	78.0	2021.0	2.0
16	69.0	68.0	15	99.0	2020.0	2.0
17	66.0	33.0	9	NaN	2020.0	2.0
18	50.0	85.0	13	99.0	2018.0	1.0
19	68.0	86.0	2	84.0	2021.0	2.0
20	76.0	75.0	4	100.0	2019.0	2.0
21	61.0	82.0	5	77.0	2034.0	2.0
22	NaN	93.0	12	NaN	2018.0	1.0
23	79.0	88.0	0	76.0	NaN	2.0
24	71.0	86.0	9	96.0	2018.0	2.0
25	68.0	81.0	14	86.0	2018.0	2.0
26	40.0	92.0	12	76.0	2021.0	NaN
27	61.0	80.0	17	83.0	2000.0	2.0
28	69.0	81.0	6	78.0	2019.0	2.0

In [61]:

```
missing_values = ["Na", "na"]
df1 = pd.read_csv("StudentsPerformance.csv", na_values = missing_values)
df1
```

Out[61]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	NaN	68.0	85.0	2019.0	NaN
4	94.0	89.0	75.0	97.0	2020.0	2.0
5	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	76.0	45.0	96.0	2018.0	2.0
11	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	83.0	NaN	75.0	2067.0	1.0
15	NaN	75.0	75.0	78.0	2021.0	2.0
16	69.0	68.0	83.0	99.0	2020.0	2.0
17	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	85.0	75.0	99.0	2018.0	1.0
19	68.0	86.0	61.0	84.0	2021.0	2.0
20	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	82.0	65.0	77.0	2034.0	2.0
22	NaN	93.0	74.0	NaN	2018.0	1.0
23	79.0	88.0	30.0	76.0	NaN	2.0
24	71.0	86.0	69.0	96.0	2018.0	2.0
25	68.0	81.0	79.0	86.0	2018.0	2.0
26	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	80.0	NaN	83.0	2000.0	2.0
28	69.0	81.0	66.0	78.0	2019.0	2.0

In [62]:
```
ndf=df1
ndf.fillna(0)
```

Out[62]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	0.0	68.0	85.0	2019.0	0.0
4	94.0	89.0	75.0	97.0	2020.0	2.0
5	74.0	82.0	0.0	94.0	0.0	2.0
6	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	89.0	68.0	0.0	2019.0	2.0
8	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	76.0	62.0	85.0	2025.0	0.0
10	80.0	76.0	45.0	96.0	2018.0	2.0
11	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	91.0	71.0	100.0	0.0	2.0
13	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	83.0	0.0	75.0	2067.0	1.0
15	0.0	75.0	75.0	78.0	2021.0	2.0
16	69.0	68.0	83.0	99.0	2020.0	2.0
17	66.0	33.0	69.0	0.0	2020.0	2.0
18	50.0	85.0	75.0	99.0	2018.0	1.0
19	68.0	86.0	61.0	84.0	2021.0	2.0
20	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	82.0	65.0	77.0	2034.0	2.0
22	0.0	93.0	74.0	0.0	2018.0	1.0
23	79.0	88.0	30.0	76.0	0.0	2.0
24	71.0	86.0	69.0	96.0	2018.0	2.0
25	68.0	81.0	79.0	86.0	2018.0	2.0
26	40.0	92.0	74.0	76.0	2021.0	0.0
27	61.0	80.0	0.0	83.0	2000.0	2.0
28	69.0	81.0	66.0	78.0	2019.0	2.0

In [63]:
```
m_v=df1['Math_Score'].mean()
df1['Math_Score'].fillna(value=m_v, inplace=True)
df1
```

Out[63]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.00000	87.0	65.0	80.0	2018.0	2.0
1	63.00000	88.0	99.0	76.0	2021.0	2.0
2	72.00000	91.0	62.0	75.0	2020.0	2.0
3	85.00000	NaN	68.0	85.0	2019.0	NaN
4	94.00000	89.0	75.0	97.0	2020.0	2.0
5	74.00000	82.0	NaN	94.0	NaN	2.0
6	61.00000	87.0	67.0	86.0	2019.0	2.0
7	63.00000	89.0	68.0	NaN	2019.0	2.0
8	78.00000	78.0	63.0	83.0	2021.0	2.0
9	79.00000	76.0	62.0	85.0	2025.0	NaN
10	80.00000	76.0	45.0	96.0	2018.0	2.0
11	76.00000	83.0	72.0	98.0	2021.0	2.0
12	62.00000	91.0	71.0	100.0	NaN	2.0
13	67.00000	81.0	66.0	91.0	2020.0	2.0
14	73.00000	83.0	NaN	75.0	2067.0	1.0
15	69.62963	75.0	75.0	78.0	2021.0	2.0
16	69.00000	68.0	83.0	99.0	2020.0	2.0
17	66.00000	33.0	69.0	NaN	2020.0	2.0
18	50.00000	85.0	75.0	99.0	2018.0	1.0
19	68.00000	86.0	61.0	84.0	2021.0	2.0
20	76.00000	75.0	63.0	100.0	2019.0	2.0
21	61.00000	82.0	65.0	77.0	2034.0	2.0
22	69.62963	93.0	74.0	NaN	2018.0	1.0
23	79.00000	88.0	30.0	76.0	NaN	2.0
24	71.00000	86.0	69.0	96.0	2018.0	2.0
25	68.00000	81.0	79.0	86.0	2018.0	2.0
26	40.00000	92.0	74.0	76.0	2021.0	NaN
27	61.00000	80.0	NaN	83.0	2000.0	2.0
28	69.00000	81.0	66.0	78.0	2019.0	2.0

In [64]:
```
1 df1.dropna()
```

Out[64]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	75.00000	87.0	65.0	80.0	2018.0	2.0
1	63.00000	88.0	99.0	76.0	2021.0	2.0
2	72.00000	91.0	62.0	75.0	2020.0	2.0
4	94.00000	89.0	75.0	97.0	2020.0	2.0
6	61.00000	87.0	67.0	86.0	2019.0	2.0
8	78.00000	78.0	63.0	83.0	2021.0	2.0
10	80.00000	76.0	45.0	96.0	2018.0	2.0
11	76.00000	83.0	72.0	98.0	2021.0	2.0
13	67.00000	81.0	66.0	91.0	2020.0	2.0
15	69.62963	75.0	75.0	78.0	2021.0	2.0
16	69.00000	68.0	83.0	99.0	2020.0	2.0
18	50.00000	85.0	75.0	99.0	2018.0	1.0
19	68.00000	86.0	61.0	84.0	2021.0	2.0
20	76.00000	75.0	63.0	100.0	2019.0	2.0
21	61.00000	82.0	65.0	77.0	2034.0	2.0
24	71.00000	86.0	69.0	96.0	2018.0	2.0
25	68.00000	81.0	79.0	86.0	2018.0	2.0
28	69.00000	81.0	66.0	78.0	2019.0	2.0

In [65]:
```
1 df1.dropna(axis = 1)
```

Out[65]:

	Math_Score
0	75.00000
1	63.00000
2	72.00000
3	85.00000
4	94.00000
5	74.00000
6	61.00000
7	63.00000
8	78.00000
9	79.00000
10	80.00000
11	76.00000
12	62.00000
13	67.00000
14	73.00000
15	69.62963
16	69.00000
17	66.00000
18	50.00000
19	68.00000
20	76.00000
21	61.00000
22	69.62963
23	79.00000
24	71.00000
25	68.00000
26	40.00000
27	61.00000
28	69.00000

In [66]:
```
new_data = df1.dropna(axis = 0, how ='any')
new_data

```

Out[66]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.00000	87.0	65.0	80.0	2018.0	2.0
1	63.00000	88.0	99.0	76.0	2021.0	2.0
2	72.00000	91.0	62.0	75.0	2020.0	2.0
4	94.00000	89.0	75.0	97.0	2020.0	2.0
6	61.00000	87.0	67.0	86.0	2019.0	2.0
8	78.00000	78.0	63.0	83.0	2021.0	2.0
10	80.00000	76.0	45.0	96.0	2018.0	2.0
11	76.00000	83.0	72.0	98.0	2021.0	2.0
13	67.00000	81.0	66.0	91.0	2020.0	2.0
15	69.62963	75.0	75.0	78.0	2021.0	2.0
16	69.00000	68.0	83.0	99.0	2020.0	2.0
18	50.00000	85.0	75.0	99.0	2018.0	1.0
19	68.00000	86.0	61.0	84.0	2021.0	2.0
20	76.00000	75.0	63.0	100.0	2019.0	2.0
21	61.00000	82.0	65.0	77.0	2034.0	2.0
24	71.00000	86.0	69.0	96.0	2018.0	2.0
25	68.00000	81.0	79.0	86.0	2018.0	2.0
28	69.00000	81.0	66.0	78.0	2019.0	2.0

In [68]:
```
import numpy as np
import matplotlib.pyplot as plt
print(np.where(df1['Math_Score']>90))
print(np.where(df1['Reading_Score']<25))
print(np.where(df1['Writing_Score']<30))
```

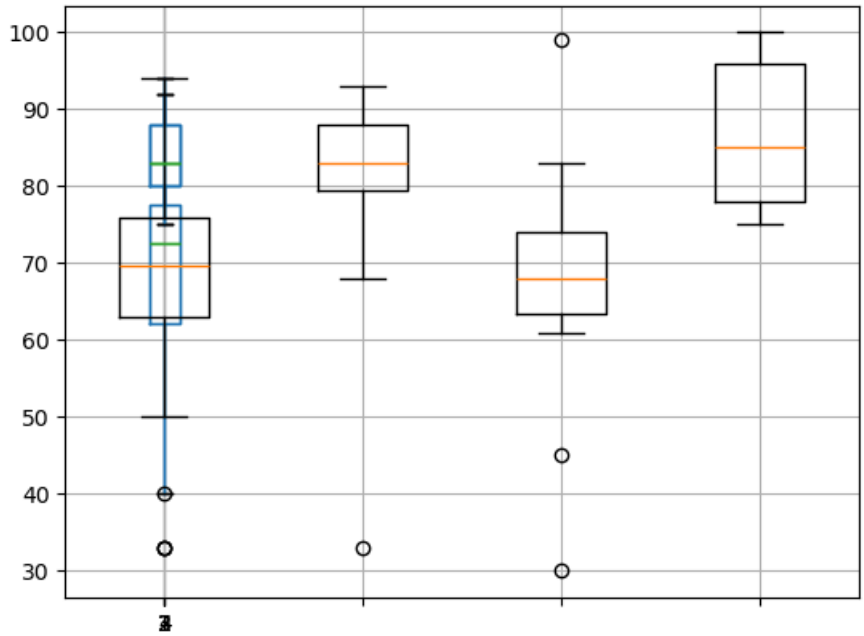
```
(array([4], dtype=int64),)
(array([], dtype=int64),)
(array([], dtype=int64),)
```

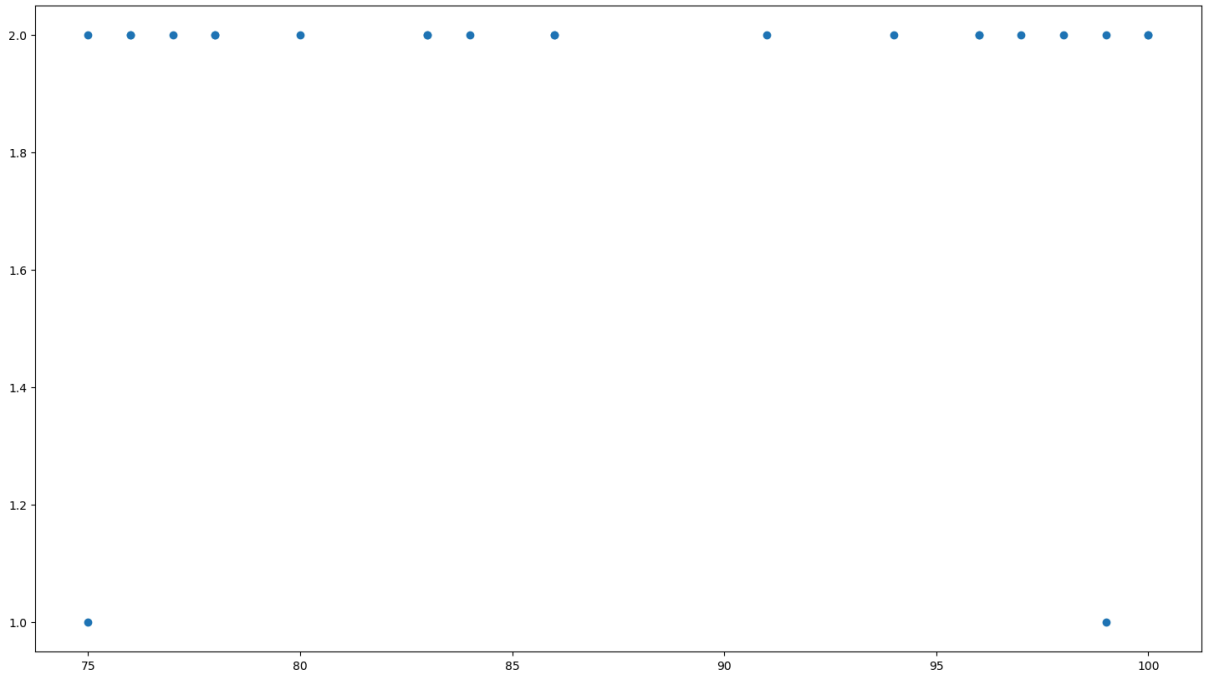
In [69]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

```

In [70]:
```
fig, ax = plt.subplots(figsize = (18,10))
ax.scatter(df1['Placement_Score'], df1['Plcement_Offer_Count'])
plt.show()

ax.set_xlabel('(Proportion non-retail business acres)/(town)')
ax.set_ylabel('(Full-value property-tax rate)/($10,000)'
```





Out[70]: Text(4.444444444444452, 0.5, '(Full-value property-tax rate)/($10,000)')

In [71]:
```
print(np.where((df1['Placement_Score']<50) & (df1['Plcement_Offer_Count']>1)))
print(np.where((df1['Placement_Score']>85) & (df1['Plcement_Offer_Count']<3)))
```

```
(array([], dtype=int64),)
(array([ 4,  5,  6, 10, 11, 12, 13, 16, 18, 20, 24, 25], dtype=int64),)
```

In [72]:
```
import numpy as np
from scipy import stats
```

In [73]:
```
z = np.abs(stats.zscore(df1['Math_Score']))
print(z)
```

```
0     5.288666e-01
1     6.528767e-01
2     2.334308e-01
3     1.513653e+00
4     2.399960e+00
5     4.303880e-01
6     8.498339e-01
7     6.528767e-01
8     8.243024e-01
9     9.227810e-01
10    1.021260e+00
11    6.273452e-01
12    7.513553e-01
13    2.589623e-01
14    3.319094e-01
15    1.399465e-15
16    6.200505e-02
17    3.574409e-01
18    1.933099e+00
19    1.604837e-01
20    6.273452e-01
21    8.498339e-01
22    1.399465e-15
23    9.227810e-01
24    1.349522e-01
25    1.604837e-01
26    2.917885e+00
27    8.498339e-01
28    6.200505e-02
Name: Math_Score, dtype: float64
```

In [74]:
```
threshold = 0.18
sample_outliers = np.where(z <threshold)
sample_outliers
```

Out[74]:
```
(array([15, 16, 19, 22, 24, 25, 28], dtype=int64),)
```

In [75]:
```
sorted_rscore= sorted(df1['Reading_Score'])
```

In [76]:
```
sorted_rscore
```

Out[76]:
```
[33.0,
 68.0,
 75.0,
 75.0,
 76.0,
 76.0,
 78.0,
 80.0,
 81.0,
 81.0,
 81.0,
 81.0,
 82.0,
 82.0,
 83.0,
 83.0,
 86.0,
 87.0,
 87.0,
 88.0,
 91.0,
 nan,
 85.0,
 86.0,
 88.0,
 89.0,
 89.0,
 91.0,
 92.0,
 93.0]
```

In [77]:
```
q1 = np.percentile(sorted_rscore, 33.0)
q3 = np.percentile(sorted_rscore, 91.0)
print(q1,q3)

```

```
nan nan
```

In [78]:
```
IQR = q3-q1
lwr_bound = q1-(1.5*IQR)
upr_bound = q3+(1.5*IQR)
print(lwr_bound, upr_bound)
```

```
nan nan
```

In [79]:
```
r_outliers = []
for i in sorted_rscore:
    if (i<lwr_bound or i>upr_bound):
        r_outliers.append(i)
print(r_outliers)

```

```
[]
```

In [80]:
```
new_df=df1
for i in sample_outliers:
    new_df.drop(i,inplace=True)
new_df
```

Out[80]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	NaN	68.0	85.0	2019.0	NaN
4	94.0	89.0	75.0	97.0	2020.0	2.0
5	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	76.0	45.0	96.0	2018.0	2.0
11	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	83.0	NaN	75.0	2067.0	1.0
17	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	85.0	75.0	99.0	2018.0	1.0
20	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	82.0	65.0	77.0	2034.0	2.0
23	79.0	88.0	30.0	76.0	NaN	2.0
26	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	80.0	NaN	83.0	2000.0	2.0

In [81]:
```
df_stud=df1
ninetieth_percentile = np.percentile(df_stud['Math_Score'], 90)
b = np.where(df_stud['Math_Score']>ninetieth_percentile,
ninetieth_percentile, df_stud['Math_Score'])
print("New array:",b)

```

```
New array: [75.  63.  72.  79.9 79.9 74.  61.  63.  78.  79.  79.9 76.  62.  67.
 73.  66.  50.  76.  61.  79.  40.  61. ]
```

In [82]:
```
df_stud.insert(1,"m score",b,True)
df_stud
```

Out[82]:

	Math_Score	m score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	79.9	NaN	68.0	85.0	2019.0	NaN
4	94.0	79.9	89.0	75.0	97.0	2020.0	2.0
5	74.0	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	79.9	76.0	45.0	96.0	2018.0	2.0
11	76.0	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	73.0	83.0	NaN	75.0	2067.0	1.0
17	66.0	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	50.0	85.0	75.0	99.0	2018.0	1.0
20	76.0	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	61.0	82.0	65.0	77.0	2034.0	2.0
23	79.0	79.0	88.0	30.0	76.0	NaN	2.0
26	40.0	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	61.0	80.0	NaN	83.0	2000.0	2.0

In [93]:
```
col = ['Reading_Score']
df1.boxplot(col)
median=np.median(sorted_rscore)
median
refined_df1=df1
```

In [95]:
```
refined_df1['Reading_Score'] = np.where(refined_df1['Reading_Score'] >upr_bound, median,refined
refined_df1
```

Out[95]:

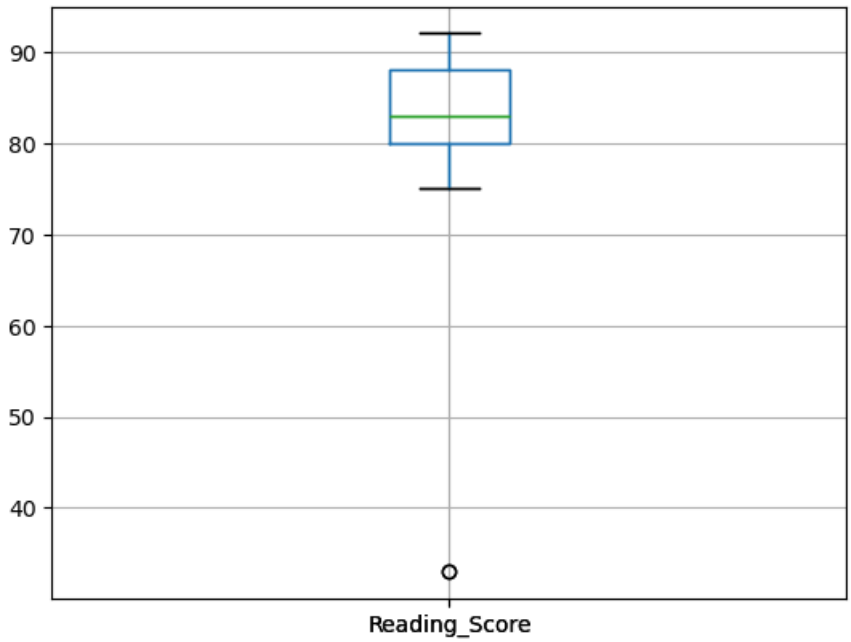
	Math_Score	m score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	PIcement_Offer_Count
0	75.0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	79.9	NaN	68.0	85.0	2019.0	NaN
4	94.0	79.9	89.0	75.0	97.0	2020.0	2.0
5	74.0	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	79.9	76.0	45.0	96.0	2018.0	2.0
11	76.0	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	73.0	83.0	NaN	75.0	2067.0	1.0
17	66.0	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	50.0	85.0	75.0	99.0	2018.0	1.0
20	76.0	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	61.0	82.0	65.0	77.0	2034.0	2.0
23	79.0	79.0	88.0	30.0	76.0	NaN	2.0
26	40.0	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	61.0	80.0	NaN	83.0	2000.0	2.0

In [96]:
```
refined_df1['Reading_Score'] = np.where(refined_df1['Reading_Score'] <lwr_bound, median,refined
refined_df1
```

Out[96]:

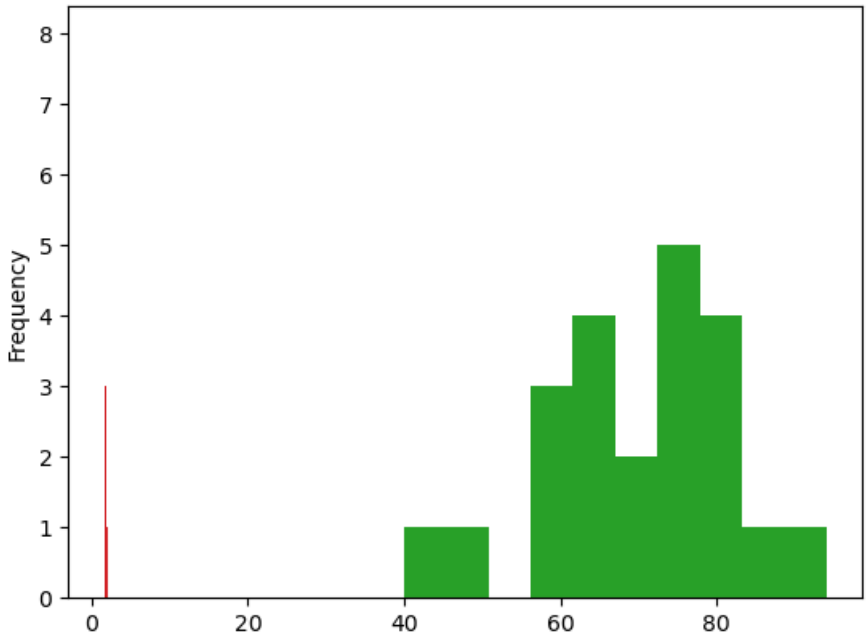
	Math_Score	m score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Plcement_Offer_Count
0	75.0	75.0	87.0	65.0	80.0	2018.0	2.0
1	63.0	63.0	88.0	99.0	76.0	2021.0	2.0
2	72.0	72.0	91.0	62.0	75.0	2020.0	2.0
3	85.0	79.9	NaN	68.0	85.0	2019.0	NaN
4	94.0	79.9	89.0	75.0	97.0	2020.0	2.0
5	74.0	74.0	82.0	NaN	94.0	NaN	2.0
6	61.0	61.0	87.0	67.0	86.0	2019.0	2.0
7	63.0	63.0	89.0	68.0	NaN	2019.0	2.0
8	78.0	78.0	78.0	63.0	83.0	2021.0	2.0
9	79.0	79.0	76.0	62.0	85.0	2025.0	NaN
10	80.0	79.9	76.0	45.0	96.0	2018.0	2.0
11	76.0	76.0	83.0	72.0	98.0	2021.0	2.0
12	62.0	62.0	91.0	71.0	100.0	NaN	2.0
13	67.0	67.0	81.0	66.0	91.0	2020.0	2.0
14	73.0	73.0	83.0	NaN	75.0	2067.0	1.0
17	66.0	66.0	33.0	69.0	NaN	2020.0	2.0
18	50.0	50.0	85.0	75.0	99.0	2018.0	1.0
20	76.0	76.0	75.0	63.0	100.0	2019.0	2.0
21	61.0	61.0	82.0	65.0	77.0	2034.0	2.0
23	79.0	79.0	88.0	30.0	76.0	NaN	2.0
26	40.0	40.0	92.0	74.0	76.0	2021.0	NaN
27	61.0	61.0	80.0	NaN	83.0	2000.0	2.0

In [119]:
```
col = ['Reading_Score']
refined_df.boxplot(col)
plt.show()
```



In [117]:

```
import matplotlib.pyplot as plt
new_df['Math_Score'].plot(kind = 'hist')
df1['log_math'] = np.log10(df1['Math_Score'])
df1['log_math'].plot(kind = 'hist')
plt.show()
```



Name : Mohan Kadambande

Roll no. : 13212 (TECO-b1)