In [1]:

```python
import pandas as pd
import numpy as np
df=pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv")
```

In [2]:

```python
df
```

Out[2]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
20	23	NaN	80.0	58.0	82.0	2019	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

In [3]:

```python
df.isnull()
```

Out[3]:

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

In [4]:

```
df.notnull()
```

Out[4]:

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

In [5]:

```
series = pd.isnull(df["Math_Score"])
df[series]
```

Out[5]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
20	23	NaN	80.0	58.0	82.0	2019	2

In [6]:

```
series1 = pd.notnull(df["Math_Score"])
df[series1]
```

Loading [MathJax]/extensions/Safe.js

Out[6]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

In [7]:

```
#fill the  missing values in Math_Score using avg
```

In [8]:

```
average_Math_Score = df['Math_Score'].mean()
df['Math_Score']  = df[Math_Score].replace(np.nan, average_Math_Score)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 2
      1 average_Math_Score = df['Math_Score'].mean()
----> 2 df['Math_Score']  = df[Math_Score].replace(np.nan, average_Math_Score)

NameError: name 'Math_Score' is not defined
```

Loading [MathJax]/extensions/Safe.js

In [9]:

```python
average_Math_Score = df['Math_Score'].mean()
df['Math_Score']  = df['Math_Score'].replace(np.nan, average_Math_Score)
```

In [10]:

```python
average_Math_Score
```

Out[10]:

```
73.96551724137932
```

In [11]:

```python
df
```

Out[11]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	85.0	59.0	80.0	2022	2
1	24	67.000000	75.0	82.0	74.0	2021	2
2	23	71.000000	72.0	78.0	71.0	2022	2
3	23	80.000000	74.0	90.0	83.0	2022	2
4	23	74.000000	76.0	83.0	85.0	2022	2
5	24	67.000000	71.0	77.0	69.0	2021	2
6	23	75.000000	70.0	76.0	78.0	2022	2
7	23	76.000000	73.0	99.0	81.0	2022	2
8	25	73.000000	88.0	78.0	90.0	2019	3
9	25	78.000000	89.0	79.0	91.0	2018	3
10	23	79.000000	75.0	NaN	70.0	2021	2
11	24	76.000000	85.0	42.0	92.0	2019	3
12	25	68.000000	78.0	76.0	85.0	2018	2
13	23	65.000000	NaN	80.0	88.0	2021	3
14	22	76.000000	80.0	59.0	83.0	2020	2
15	23	69.000000	82.0	76.0	89.0	2019	3
16	23	76.000000	79.0	77.0	86.0	2021	3
17	23	74.000000	76.0	81.0	64.0	2018	2
18	23	75.000000	83.0	75.0	87.0	2020	3
19	23	75.000000	75.0	79.0	80.0	2018	2
20	23	73.965517	80.0	58.0	82.0	2019	2
21	24	79.000000	85.0	77.0	91.0	2018	3
22	23	74.000000	88.0	76.0	55.0	2020	3
23	22	67.000000	78.0	76.0	80.0	2018	2
24	25	77.000000	82.0	78.0	85.0	2019	2
25	23	71.000000	89.0	79.0	87.0	2018	3
26	22	80.000000	90.0	80.0	94.0	2020	3
27	23	73.000000	77.0	75.0	82.0	2018	2
28	22	80.000000	84.0	77.0	89.0	2021	3
29	23	77.000000	79.0	76.0	NaN	2018	3

In [12]:

```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

Loading [MathJax]/extensions/Safe.js

```
df['Gender'] = le.fit_transform(df['Gender'])
newdf = df
```

In [13]:

```
df
```

Out[13]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	85.0	59.0	80.0	2022	2
1	24	67.000000	75.0	82.0	74.0	2021	2
2	23	71.000000	72.0	78.0	71.0	2022	2
3	23	80.000000	74.0	90.0	83.0	2022	2
4	23	74.000000	76.0	83.0	85.0	2022	2
5	24	67.000000	71.0	77.0	69.0	2021	2
6	23	75.000000	70.0	76.0	78.0	2022	2
7	23	76.000000	73.0	99.0	81.0	2022	2
8	25	73.000000	88.0	78.0	90.0	2019	3
9	25	78.000000	89.0	79.0	91.0	2018	3
10	23	79.000000	75.0	NaN	70.0	2021	2
11	24	76.000000	85.0	42.0	92.0	2019	3
12	25	68.000000	78.0	76.0	85.0	2018	2
13	23	65.000000	NaN	80.0	88.0	2021	3
14	22	76.000000	80.0	59.0	83.0	2020	2
15	23	69.000000	82.0	76.0	89.0	2019	3
16	23	76.000000	79.0	77.0	86.0	2021	3
17	23	74.000000	76.0	81.0	64.0	2018	2
18	23	75.000000	83.0	75.0	87.0	2020	3
19	23	75.000000	75.0	79.0	80.0	2018	2
20	23	73.965517	80.0	58.0	82.0	2019	2
21	24	79.000000	85.0	77.0	91.0	2018	3
22	23	74.000000	88.0	76.0	55.0	2020	3
23	22	67.000000	78.0	76.0	80.0	2018	2
24	25	77.000000	82.0	78.0	85.0	2019	2
25	23	71.000000	89.0	79.0	87.0	2018	3
26	22	80.000000	90.0	80.0	94.0	2020	3
27	23	73.000000	77.0	75.0	82.0	2018	2
28	22	80.000000	84.0	77.0	89.0	2021	3
29	23	77.000000	79.0	76.0	NaN	2018	3

In [ ]:

```
#null values with NaN
```

In [14]:

```
missing_values = ["Na", "na"]
df = pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv", na_values = missing_values)
df
```

Out[14]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
20	23	NaN	80.0	58.0	82.0	2019	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

In [ ]:

```
#Filling null values with a single value
```

In [15]:

```
ndf=df
ndf.fillna(0)
```

Loading [MathJax]/extensions/Safe.js

Out[15]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	0.0	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	0.0	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
20	23	0.0	80.0	58.0	82.0	2019	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	0.0	2018	3

In [16]:

```
data['Math_Score'] = data['Math_Score'].fillna(data['Math_Score'].mean())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[16], line 1
----> 1 data['Math_Score'] = data['Math_Score'].fillna(data['Math_Score'].mean())

NameError: name 'data' is not defined
```

In [17]:

```
df['Math_Score'] = df['Math_Score'].fillna(df['Math_Score'].mean())
```

In [18]:

```
df
```

```
Out[18]:
    Age  Math_Score  Reading_Score  Writing_Score  Placement_Score  Club_Join_Year  Placement_offer_count
 0   23   73.000000           85.0           59.0             80.0            2022                      2
 1   24   67.000000           75.0           82.0             74.0            2021                      2
 2   23   71.000000           72.0           78.0             71.0            2022                      2
 3   23   80.000000           74.0           90.0             83.0            2022                      2
 4   23   74.000000           76.0           83.0             85.0            2022                      2
 5   24   67.000000           71.0           77.0             69.0            2021                      2
 6   23   75.000000           70.0           76.0             78.0            2022                      2
 7   23   76.000000           73.0           99.0             81.0            2022                      2
 8   25   73.000000           88.0           78.0             90.0            2019                      3
 9   25   78.000000           89.0           79.0             91.0            2018                      3
10   23   79.000000           75.0            NaN             70.0            2021                      2
11   24   76.000000           85.0           42.0             92.0            2019                      3
12   25   68.000000           78.0           76.0             85.0            2018                      2
13   23   65.000000            NaN           80.0             88.0            2021                      3
14   22   76.000000           80.0           59.0             83.0            2020                      2
15   23   69.000000           82.0           76.0             89.0            2019                      3
16   23   76.000000           79.0           77.0             86.0            2021                      3
17   23   74.000000           76.0           81.0             64.0            2018                      2
18   23   75.000000           83.0           75.0             87.0            2020                      3
19   23   75.000000           75.0           79.0             80.0            2018                      2
20   23   73.965517           80.0           58.0             82.0            2019                      2
21   24   79.000000           85.0           77.0             91.0            2018                      3
22   23   74.000000           88.0           76.0             55.0            2020                      3
23   22   67.000000           78.0           76.0             80.0            2018                      2
24   25   77.000000           82.0           78.0             85.0            2019                      2
25   23   71.000000           89.0           79.0             87.0            2018                      3
26   22   80.000000           90.0           80.0             94.0            2020                      3
27   23   73.000000           77.0           75.0             82.0            2018                      2
28   22   80.000000           84.0           77.0             89.0            2021                      3
29   23   77.000000           79.0           76.0              NaN            2018                      3
```

```
In [ ]: #Filling a null values using replace() method
```

```
In [19]: ndf.replace(to_replace = np.nan, value = -99)
```

Out[19]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	85.0	59.0	80.0	2022	2
1	24	67.000000	75.0	82.0	74.0	2021	2
2	23	71.000000	72.0	78.0	71.0	2022	2
3	23	80.000000	74.0	90.0	83.0	2022	2
4	23	74.000000	76.0	83.0	85.0	2022	2
5	24	67.000000	71.0	77.0	69.0	2021	2
6	23	75.000000	70.0	76.0	78.0	2022	2
7	23	76.000000	73.0	99.0	81.0	2022	2
8	25	73.000000	88.0	78.0	90.0	2019	3
9	25	78.000000	89.0	79.0	91.0	2018	3
10	23	79.000000	75.0	-99.0	70.0	2021	2
11	24	76.000000	85.0	42.0	92.0	2019	3
12	25	68.000000	78.0	76.0	85.0	2018	2
13	23	65.000000	-99.0	80.0	88.0	2021	3
14	22	76.000000	80.0	59.0	83.0	2020	2
15	23	69.000000	82.0	76.0	89.0	2019	3
16	23	76.000000	79.0	77.0	86.0	2021	3
17	23	74.000000	76.0	81.0	64.0	2018	2
18	23	75.000000	83.0	75.0	87.0	2020	3
19	23	75.000000	75.0	79.0	80.0	2018	2
20	23	73.965517	80.0	58.0	82.0	2019	2
21	24	79.000000	85.0	77.0	91.0	2018	3
22	23	74.000000	88.0	76.0	55.0	2020	3
23	22	67.000000	78.0	76.0	80.0	2018	2
24	25	77.000000	82.0	78.0	85.0	2019	2
25	23	71.000000	89.0	79.0	87.0	2018	3
26	22	80.000000	90.0	80.0	94.0	2020	3
27	23	73.000000	77.0	75.0	82.0	2018	2
28	22	80.000000	84.0	77.0	89.0	2021	3
29	23	77.000000	79.0	76.0	-99.0	2018	3

In [ ]:

```
#Deleting null values using dropna() method
```

In [20]:

```
ndf.dropna()
```

Loading [MathJax]/extensions/Safe.js

Out[20]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	85.0	59.0	80.0	2022	2
1	24	67.000000	75.0	82.0	74.0	2021	2
2	23	71.000000	72.0	78.0	71.0	2022	2
3	23	80.000000	74.0	90.0	83.0	2022	2
4	23	74.000000	76.0	83.0	85.0	2022	2
5	24	67.000000	71.0	77.0	69.0	2021	2
6	23	75.000000	70.0	76.0	78.0	2022	2
7	23	76.000000	73.0	99.0	81.0	2022	2
8	25	73.000000	88.0	78.0	90.0	2019	3
9	25	78.000000	89.0	79.0	91.0	2018	3
11	24	76.000000	85.0	42.0	92.0	2019	3
12	25	68.000000	78.0	76.0	85.0	2018	2
14	22	76.000000	80.0	59.0	83.0	2020	2
15	23	69.000000	82.0	76.0	89.0	2019	3
16	23	76.000000	79.0	77.0	86.0	2021	3
17	23	74.000000	76.0	81.0	64.0	2018	2
18	23	75.000000	83.0	75.0	87.0	2020	3
19	23	75.000000	75.0	79.0	80.0	2018	2
20	23	73.965517	80.0	58.0	82.0	2019	2
21	24	79.000000	85.0	77.0	91.0	2018	3
23	22	67.000000	78.0	76.0	80.0	2018	2
24	25	77.000000	82.0	78.0	85.0	2019	2
25	23	71.000000	89.0	79.0	87.0	2018	3
26	22	80.000000	90.0	80.0	94.0	2020	3
27	23	73.000000	77.0	75.0	82.0	2018	2
28	22	80.000000	84.0	77.0	89.0	2021	3

In [21]:

```
ndf.dropna(axis = 1)
```

Out[21]:

	Age	Math_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	2022	2
1	24	67.000000	2021	2
2	23	71.000000	2022	2
3	23	80.000000	2022	2
4	23	74.000000	2022	2
5	24	67.000000	2021	2
6	23	75.000000	2022	2
7	23	76.000000	2022	2
8	25	73.000000	2019	3
9	25	78.000000	2018	3
10	23	79.000000	2021	2
11	24	76.000000	2019	3
12	25	68.000000	2018	2
13	23	65.000000	2021	3
14	22	76.000000	2020	2
15	23	69.000000	2019	3
16	23	76.000000	2021	3
17	23	74.000000	2018	2
18	23	75.000000	2020	3
19	23	75.000000	2018	2
20	23	73.965517	2019	2
21	24	79.000000	2018	3
22	23	74.000000	2020	3
23	22	67.000000	2018	2
24	25	77.000000	2019	2
25	23	71.000000	2018	3
26	22	80.000000	2020	3
27	23	73.000000	2018	2
28	22	80.000000	2021	3
29	23	77.000000	2018	3

In [22]:

```
new_data = ndf.dropna(axis = 0, how ='any')
new_data
```

Out[22]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.000000	85.0	59.0	80.0	2022	2
1	24	67.000000	75.0	82.0	74.0	2021	2
2	23	71.000000	72.0	78.0	71.0	2022	2
3	23	80.000000	74.0	90.0	83.0	2022	2
4	23	74.000000	76.0	83.0	85.0	2022	2
5	24	67.000000	71.0	77.0	69.0	2021	2
6	23	75.000000	70.0	76.0	78.0	2022	2
7	23	76.000000	73.0	99.0	81.0	2022	2
8	25	73.000000	88.0	78.0	90.0	2019	3
9	25	78.000000	89.0	79.0	91.0	2018	3
11	24	76.000000	85.0	42.0	92.0	2019	3
12	25	68.000000	78.0	76.0	85.0	2018	2
14	22	76.000000	80.0	59.0	83.0	2020	2
15	23	69.000000	82.0	76.0	89.0	2019	3
16	23	76.000000	79.0	77.0	86.0	2021	3
17	23	74.000000	76.0	81.0	64.0	2018	2
18	23	75.000000	83.0	75.0	87.0	2020	3
19	23	75.000000	75.0	79.0	80.0	2018	2
20	23	73.965517	80.0	58.0	82.0	2019	2
21	24	79.000000	85.0	77.0	91.0	2018	3
23	22	67.000000	78.0	76.0	80.0	2018	2
24	25	77.000000	82.0	78.0	85.0	2019	2
25	23	71.000000	89.0	79.0	87.0	2018	3
26	22	80.000000	90.0	80.0	94.0	2020	3
27	23	73.000000	77.0	75.0	82.0	2018	2
28	22	80.000000	84.0	77.0	89.0	2021	3

In [23]:

```
#Detecting Outlier using Box Plot
```

In [24]:

```
col = ['Math_Score', 'Reading_Score' , 'Writing_Score','Placement_Score']
df[cols].boxplot(|)
```

```
  Cell In[24], line 2
    df[cols].boxplot(|)
                     ^
SyntaxError: invalid syntax
```

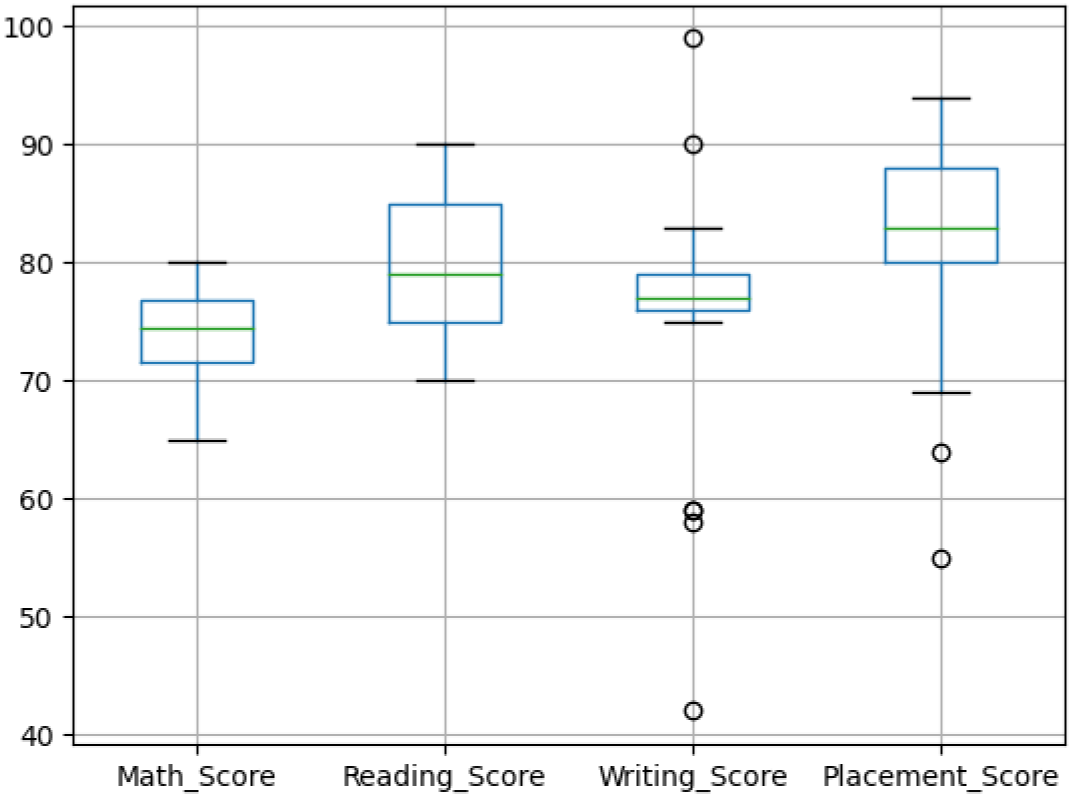
In [25]:

```
cols = ['Math_Score', 'Reading_Score' , 'Writing_Score','Placement_Score']
df[cols].boxplot()
```

Out[25]:

```
<Axes: >
```

Loading [MathJax]/extensions/Safe.js



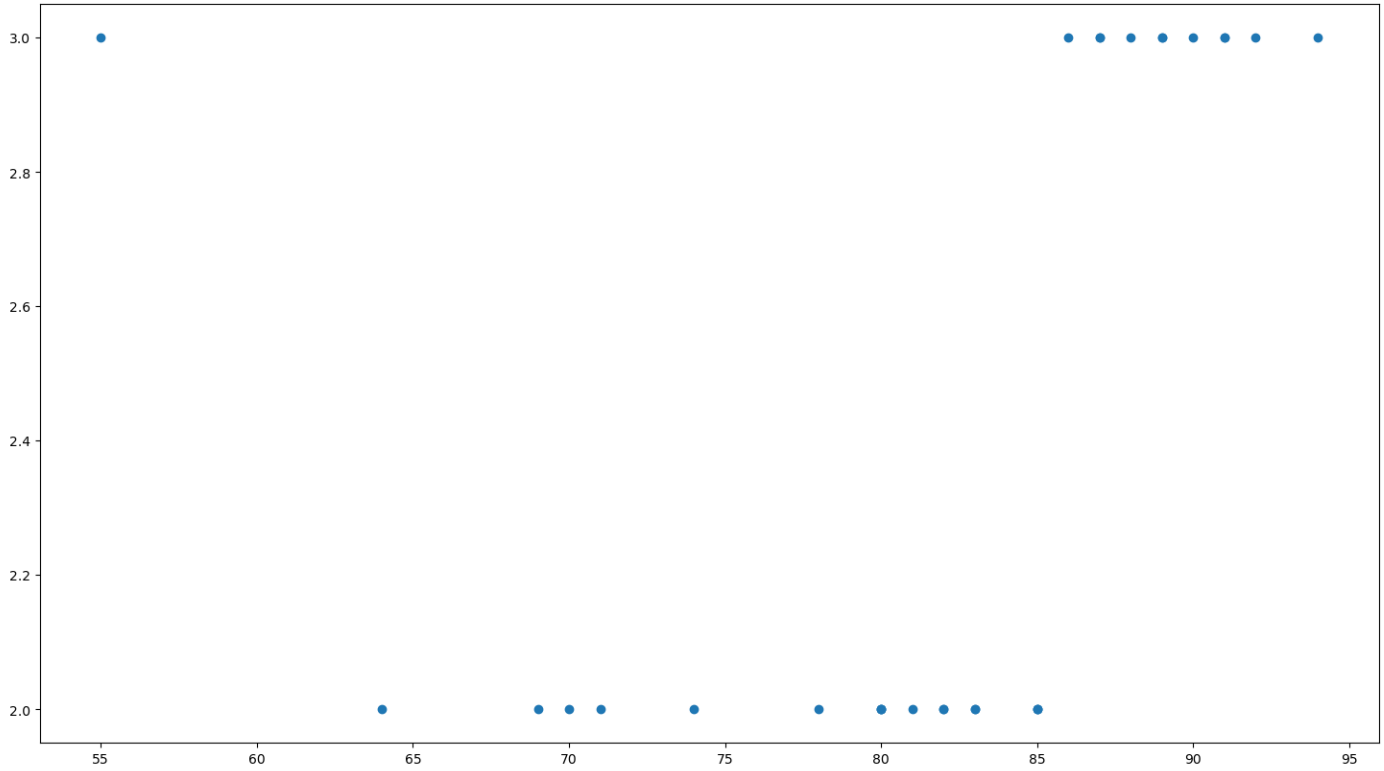
```
In [26]: np.where(df['Math_Score'] > 100)

Out[26]: (array([], dtype=int64),)
```

```
In [27]: np.where(df['Math_Score'] < 80)

Out[27]: (array([ 0,  1,  2,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19, 20, 21, 22, 23, 24, 25, 27, 29], dtype=int64),)
```

```
In [28]: #Detecting Outliers using Scatter Plot
```

```
In [29]: import matplotlib.pyplot as plt
```

```
In [30]: fig, ax = plt.subplots(figsize = (18,10))
         ax.scatter(df['Placement_Score'], df['Placement_offer_count'])
         plt.show()
```



```
In [31]: print(np.where((df['placement score']<50) & (df['Placement_offer_count']>1)))
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3802, in Index.get_loc(self, key, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:138, in pandas._libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:165, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'placement score'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[31], line 1
----> 1 print(np.where((df['placement score']<50) & (df['Placement_offer_count']>1)))

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3807, in DataFrame.__getitem__(self, key)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3804, in Index.get_loc(self, key, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: 'placement score'
```

In [33]:
```
print(np.where((df['Placement_Score'] < 50) & (df['Placement_offer_count'] > 1)))
```

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

In [32]:
```
print(np.where((df['Placement_Score'] > 80) & (df['Placement_offer_count'] < 3)))
```

```
(array([ 3,  4,  7, 12, 14, 20, 24, 27], dtype=int64),)
```

In [ ]:
```
#Detecting outliers using Z-Score:
```

In [34]:
```
import numpy as np
from scipy import stats
```

In [35]:
```
z = np.abs(stats.zscore(df['Math_Score']))
```

Loading [MathJax]/extensions/Safe.js

```
0     0.231695
1     1.671512
2     0.711634
3     1.448092
4     0.008275
5     1.671512
6     0.248244
7     0.488214
8     0.231695
9     0.968153
10    1.208123
11    0.488214
12    1.431542
13    2.151451
14    0.488214
15    1.191573
16    0.488214
17    0.008275
18    0.248244
19    0.248244
20    0.000000
21    1.208123
22    0.008275
23    1.671512
24    0.728183
25    0.711634
26    1.448092
27    0.231695
28    1.448092
29    0.728183
Name: Math_Score, dtype: float64
```

```
In [36]: #define an outlier threshold value is chosen.
         threshold = 0.19
         sample_outliers = np.where(z <threshold)
         sample_outliers
```

```
Out[36]: (array([ 4, 17, 20, 22], dtype=int64),)
```

```
In [37]: #Detecting outliers using Inter Quantile Range(IQR):
```

```
In [38]: import numpy as np
         sorted_rscore= sorted(df['Reading_Score'])
         print(sorted_rscore)
```

```
[70.0, 71.0, 72.0, 73.0, 74.0, 75.0, 75.0, 75.0, 76.0, 76.0, 77.0, 78.0, 78.0, 79.0, 79.
0, 80.0, 80.0, 82.0, 82.0, 83.0, 84.0, 85.0, 85.0, 85.0, 88.0, 88.0, 89.0, nan, 89.0, 9
0.0]
```

```
In [39]: #Calculate and print Quartile 1 and Quartile 3
```

```
In [40]: q1 = np.percentile(sorted_rscore, 25)
         q3 = np.percentile(sorted_rscore, 75)
         print(q1,q3)
```

```
nan nan
```

```
In [41]: #Calculate value of IQR (Inter Quartile Range)
```

```
In [42]: IQR = q3-q1
         lwr_bound = q1-(1.5*IQR)
         upr_bound = q3+(1.5*IQR)
```

```
In [43]: print(lwr_bound, upr_bound)
```

```
nan nan
```

```
In [44]: r_outliers = [ i for i in sorted_rscore if (i< lwr_bound or i>upr_bound)]
         print(r_outliers)
```

```
[]
```

```
In [45]: #Handling of Outliers:
         #Trimming/removing the outlier:
         new_df=df
         for i in sample_outliers:
         new_df.drop(i,inplace=True)
         new_df
```

```
  Cell In[45], line 5
    new_df.drop(i,inplace=True)
    ^
IndentationError: expected an indented block after 'for' statement on line 4
```

```
In [46]: new_df=df
         for i in sample_outliers:
             new_df.drop(i,inplace=True)
         new_df
```

Loading [MathJax]/extensions/Safe.js

Out[46]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
21	24	79.0	85.0	77.0	91.0	2018	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

In [47]:

```
import pandas as pd
import numpy as np

#read the csv file
df = pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv")
df_stud=df

#calculate the 90th percentilevalue
ninetieth_percentile = np.percentile(df_stud['Math_Score'], 90)

#cap values above the 90th percentile and floor values below the 10th percentile
b = np.where(df_stud['Math_Score'] > ninetieth_percentile,ninetieth_percentile, df_stud[

print("New array:",b)
```

```
New array: [73. 67. 71. 80. 74. 67. 75. 76. 73. 78. 79. 76. 68. 65. 76. 69. 76. 74.
 75. 75. nan 79. 74. 67. 77. 71. 80. 73. 80. 77.]
```

In [48]:

```
df_stud.insert(1,"m score",b,True)
```

Out[48]:

	Age	m score	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_
0	23	73.0	73.0	85.0	59.0	80.0	2022	
1	24	67.0	67.0	75.0	82.0	74.0	2021	
2	23	71.0	71.0	72.0	78.0	71.0	2022	
3	23	80.0	80.0	74.0	90.0	83.0	2022	
4	23	74.0	74.0	76.0	83.0	85.0	2022	
5	24	67.0	67.0	71.0	77.0	69.0	2021	
6	23	75.0	75.0	70.0	76.0	78.0	2022	
7	23	76.0	76.0	73.0	99.0	81.0	2022	
8	25	73.0	73.0	88.0	78.0	90.0	2019	
9	25	78.0	78.0	89.0	79.0	91.0	2018	
10	23	79.0	79.0	75.0	NaN	70.0	2021	
11	24	76.0	76.0	85.0	42.0	92.0	2019	
12	25	68.0	68.0	78.0	76.0	85.0	2018	
13	23	65.0	65.0	NaN	80.0	88.0	2021	
14	22	76.0	76.0	80.0	59.0	83.0	2020	
15	23	69.0	69.0	82.0	76.0	89.0	2019	
16	23	76.0	76.0	79.0	77.0	86.0	2021	
17	23	74.0	74.0	76.0	81.0	64.0	2018	
18	23	75.0	75.0	83.0	75.0	87.0	2020	
19	23	75.0	75.0	75.0	79.0	80.0	2018	
20	23	NaN	NaN	80.0	58.0	82.0	2019	
21	24	79.0	79.0	85.0	77.0	91.0	2018	
22	23	74.0	74.0	88.0	76.0	55.0	2020	
23	22	67.0	67.0	78.0	76.0	80.0	2018	
24	25	77.0	77.0	82.0	78.0	85.0	2019	
25	23	71.0	71.0	89.0	79.0	87.0	2018	
26	22	80.0	80.0	90.0	80.0	94.0	2020	
27	23	73.0	73.0	77.0	75.0	82.0	2018	
28	22	80.0	80.0	84.0	77.0	89.0	2021	
29	23	77.0	77.0	79.0	76.0	NaN	2018	

In [49]:

```
#Mean/Median imputation:
```

In [50]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv")

#Plot the box plot for reading score
col = ['reading score']
```

Loading [MathJax]/extensions/Safe.js

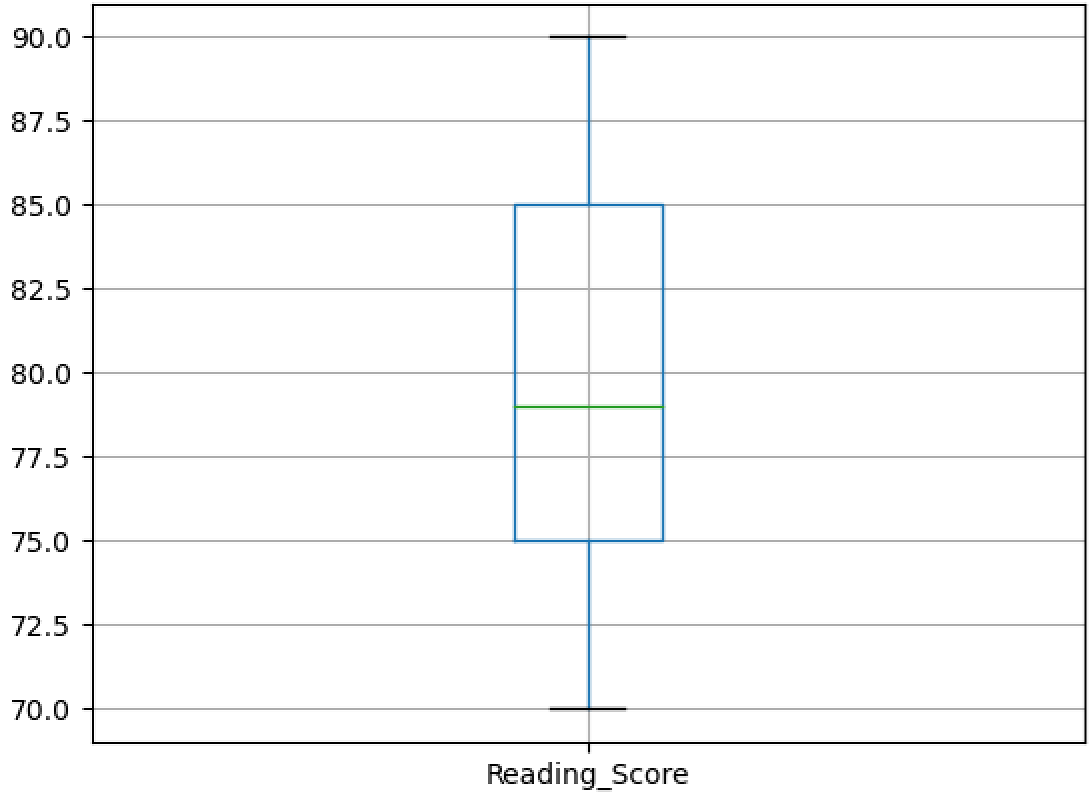
```
df.boxplot(col)
plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[50], line 9
      7 #Plot the box plot for reading score
      8 col = ['reading score']
----> 9 df.boxplot(col)
     10 plt.show()

File ~\anaconda3\lib\site-packages\pandas\plotting\_core.py:516, in boxplot_frame(self, column, by, ax, fontsize, rot, grid, figsize, layout, return_type, backend, **kwargs)
    499 @Substitution(backend=_backend_doc)
    500 @Appender(_boxplot_doc)
    501 def boxplot_frame(
   (...)
    513     **kwargs,
    514 ):
    515     plot_backend = _get_plot_backend(backend)
--> 516     return plot_backend.boxplot_frame(
    517         self,
    518         column=column,
    519         by=by,
    520         ax=ax,
    521         fontsize=fontsize,
    522         rot=rot,
    523         grid=grid,
    524         figsize=figsize,
    525         layout=layout,
    526         return_type=return_type,
    527         **kwargs,
    528     )

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\boxplot.py:458, in boxplot_frame(self, column, by, ax, fontsize, rot, grid, figsize, layout, return_type, **kwds)
    443 def boxplot_frame(
    444     self,
    445     column=None,
   (...)
    454     **kwds,
    455 ):
    456     import matplotlib.pyplot as plt
--> 458     ax = boxplot(
    459         self,
    460         column=column,
    461         by=by,
    462         ax=ax,
    463         fontsize=fontsize,
    464         grid=grid,
    465         rot=rot,
    466         figsize=figsize,
    467         layout=layout,
    468         return_type=return_type,
    469         **kwds,
    470     )
    471     plt.draw_if_interactive()
    472     return ax

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\boxplot.py:435, in boxplot(data, column, by, ax, fontsize, rot, grid, figsize, layout, return_type, **kwds)
    433     columns = data.columns
    434 else:
--> 435     data = data[columns]
    437 result = plot_group(columns, data.values.T, ax, **kwds)
    438 ax.grid(grid)
```

```
File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3813, in DataFrame.__getitem__(self, key)
   3811     if is_iterator(key):
   3812         key = list(key)
-> 3813     indexer = self.columns._get_indexer_strict(key, "columns")[1]
   3815 # take() does not accept boolean indexers
   3816 if getattr(indexer, "dtype", None) == bool:

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:6070, in Index._get_indexer_strict(self, key, axis_name)
   6067 else:
   6068     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 6070 self._raise_if_missing(keyarr, indexer, axis_name)
   6072 keyarr = self.take(indexer)
   6073 if isinstance(key, Index):
   6074     # GH 42790 - Preserve name from an Index

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:6130, in Index._raise_if_missing(self, key, indexer, axis_name)
   6128     if use_interval_msg:
   6129         key = list(key)
-> 6130     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   6132 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
   6133 raise KeyError(f"{not_found} not in index")

KeyError: "None of [Index(['reading score'], dtype='object')] are in the [columns]"
```

In [51]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv")

#Plot the box plot for reading score
col = ['Reading_Score']
df.boxplot(col)
plt.show()
```



Loading [MathJax]/extensions/Safe.js

In [53]:
```
median = np.median(sorted_rscore)
median
```

Out[53]:
```
nan
```

In [54]:
```
refined_df = df
refined_df['Reading_Score'] = np.where(refined_df['Reading_Score'] >upr_bound, median,re
```

In [55]:
```
refined_df
```

Out[55]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
20	23	NaN	80.0	58.0	82.0	2019	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

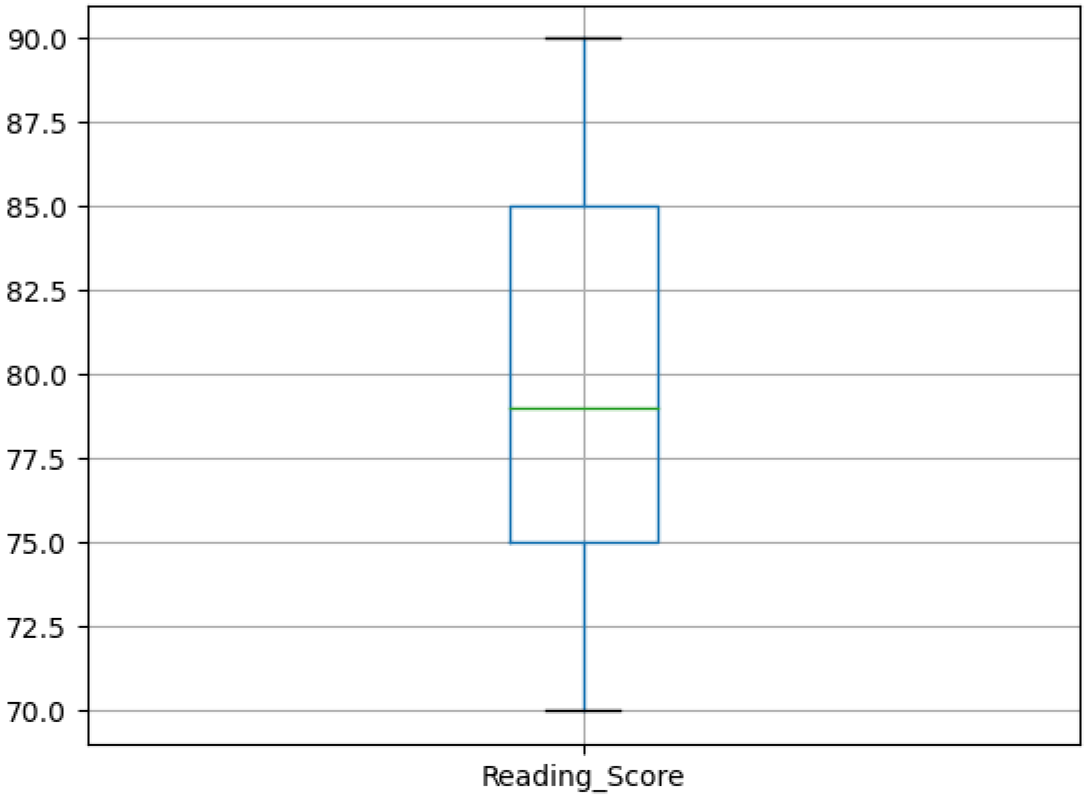
In [56]:
```
#Replace the lower bound outliers using median value
eading_Score'] = np.where(refined_df['Reading_Score'] < lwr_bound, median,r
refined_df
```

Out[56]:

	Age	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Year	Placement_offer_count
0	23	73.0	85.0	59.0	80.0	2022	2
1	24	67.0	75.0	82.0	74.0	2021	2
2	23	71.0	72.0	78.0	71.0	2022	2
3	23	80.0	74.0	90.0	83.0	2022	2
4	23	74.0	76.0	83.0	85.0	2022	2
5	24	67.0	71.0	77.0	69.0	2021	2
6	23	75.0	70.0	76.0	78.0	2022	2
7	23	76.0	73.0	99.0	81.0	2022	2
8	25	73.0	88.0	78.0	90.0	2019	3
9	25	78.0	89.0	79.0	91.0	2018	3
10	23	79.0	75.0	NaN	70.0	2021	2
11	24	76.0	85.0	42.0	92.0	2019	3
12	25	68.0	78.0	76.0	85.0	2018	2
13	23	65.0	NaN	80.0	88.0	2021	3
14	22	76.0	80.0	59.0	83.0	2020	2
15	23	69.0	82.0	76.0	89.0	2019	3
16	23	76.0	79.0	77.0	86.0	2021	3
17	23	74.0	76.0	81.0	64.0	2018	2
18	23	75.0	83.0	75.0	87.0	2020	3
19	23	75.0	75.0	79.0	80.0	2018	2
20	23	NaN	80.0	58.0	82.0	2019	2
21	24	79.0	85.0	77.0	91.0	2018	3
22	23	74.0	88.0	76.0	55.0	2020	3
23	22	67.0	78.0	76.0	80.0	2018	2
24	25	77.0	82.0	78.0	85.0	2019	2
25	23	71.0	89.0	79.0	87.0	2018	3
26	22	80.0	90.0	80.0	94.0	2020	3
27	23	73.0	77.0	75.0	82.0	2018	2
28	22	80.0	84.0	77.0	89.0	2021	3
29	23	77.0	79.0	76.0	NaN	2018	3

In [57]:

```
#Draw the box plot for redefined_df
col = ['Reading_Score']
refined_df.boxplot(col)
plt.show()
```

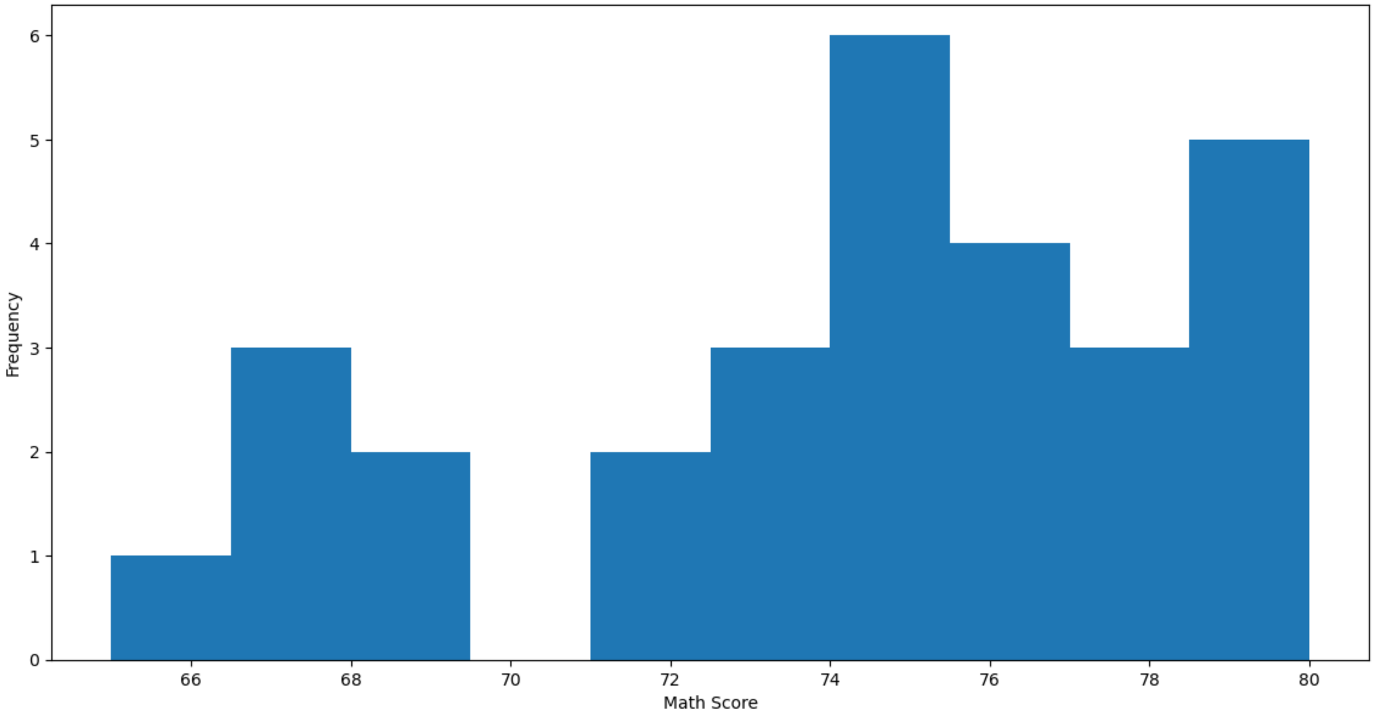


In [58]:

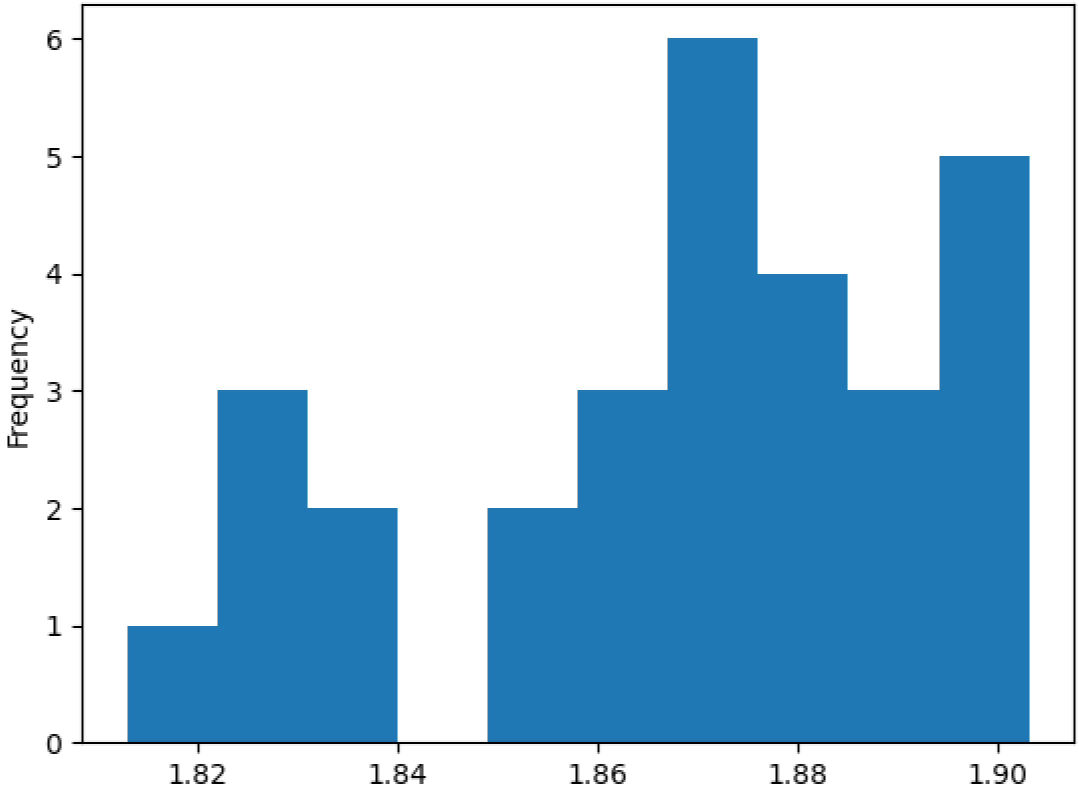
```
#To decrease the skewness and convert distribution into normal distribution

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv("C:/Users/omkar/Downloads/DatasetP2.csv")

z_score = np.abs(stats.zscore(df['Math_Score']))
threshold = 2

outliers = np.where(z_score > threshold)
df_no_outliers = df.drop(outliers[0])
```

In [60]:

```
plt.figure(figsize=(14,7))
df['Math_Score'].plot(kind='hist')

plt.xlabel('Math Score')
plt.ylabel('Frequency')
plt.show()
```



```
In [61]: #Convert the variables to logarithm at the scale 10.
         df['log_math'] = np.log10(df['Math_Score'])
         df['log_math'].plot(kind = 'hist')
         plt.show()
```



```
In [ ]:
```