DSBDA Lab Assignment No. 2

Name: Akash Ganesh Padir
Roll No.: TEB04

Data Wrangling

Missing Data Handling

In [1]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
df1= pd.read_csv("studentsperformance.csv")
```

In [5]:
```
df1
```

Out[5]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [7]:
```
df= pd.read_csv("studentsperformance.csv")
```

In [8]:
```
df
```

Out[8]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

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

Out[9]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_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	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	False
10	False	False	False	False	False	False
11	False	False	False	False	False	False
12	False	False	False	False	False	False
13	False	False	False	False	False	False
14	False	False	False	False	False	False
15	False	False	False	False	False	False
16	False	False	False	False	False	False
17	False	False	False	False	False	False
18	False	False	False	False	False	False
19	False	False	True	False	False	False
20	False	False	False	False	False	False
21	False	False	False	False	False	False
22	False	False	False	False	False	False
23	False	False	False	False	False	False
24	True	False	False	False	False	False
25	False	False	False	False	False	False
26	False	False	False	False	False	False
27	False	False	False	False	False	False
28	False	True	False	False	False	False
29	False	False	False	False	False	False

In [10]:
```
series= pd.isnull(df["Math_score"])
df[series]
```

Out[10]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
24	NaN	81.0	66.0	81	2019	2

In [11]:
```
df.notnull()
```

Out[11]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_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	True	True	True	True	True
4	True	True	True	True	True	True
5	True	True	True	True	True	True
6	True	True	True	True	True	True
7	True	True	True	True	True	True
8	True	True	True	True	True	True
9	True	True	True	True	True	True
10	True	True	True	True	True	True
11	True	True	True	True	True	True
12	True	True	True	True	True	True
13	True	True	True	True	True	True
14	True	True	True	True	True	True
15	True	True	True	True	True	True
16	True	True	True	True	True	True
17	True	True	True	True	True	True
18	True	True	True	True	True	True
19	True	True	False	True	True	True
20	True	True	True	True	True	True
21	True	True	True	True	True	True
22	True	True	True	True	True	True
23	True	True	True	True	True	True
24	False	True	True	True	True	True
25	True	True	True	True	True	True
26	True	True	True	True	True	True
27	True	True	True	True	True	True
28	True	False	True	True	True	True
29	True	True	True	True	True	True

In [12]:

```
series = pd.notnull(df["Math_score"])
df[series]
```

Out[12]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [14]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df["Math_score"]= le.fit_transform(df["Math_score"])
newdf = df
df
```

Out[14]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	14	90.0	70.0	77	2018	2
1	8	77.0	76.0	85	2018	3
2	1	88.0	68.0	92	2019	3
3	15	84.0	71.0	78	2019	2
4	12	81.0	62.0	100	2020	3
5	11	200.0	73.0	82	2020	2
6	11	75.0	65.0	100	2020	3
7	1	80.0	63.0	97	2019	3
8	13	88.0	65.0	82	2018	2
9	2	94.0	73.0	79	2021	1
10	5	79.0	77.0	80	2019	2
11	7	78.0	77.0	90	2020	3
12	14	88.0	77.0	77	2018	2
13	5	90.0	72.0	93	2020	3
14	3	90.0	67.0	79	2020	2
15	12	86.0	64.0	76	2019	2
16	0	95.0	64.0	75	2021	2
17	16	82.0	76.0	95	2019	3
18	14	90.0	74.0	81	2019	2
19	8	82.0	NaN	89	2020	3
20	7	83.0	74.0	77	2021	2
21	9	81.0	63.0	91	2020	3
22	9	91.0	61.0	75	2020	2
23	0	78.0	69.0	75	2020	2
24	17	81.0	66.0	81	2019	2
25	10	83.0	79.0	77	2019	2
26	6	87.0	76.0	95	2020	3
27	0	75.0	63.0	93	2020	3
28	14	NaN	61.0	100	2020	3
29	4	78.0	73.0	98	2018	3

In [15]:

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

Out[15]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [16]:
```
ndf=df
ndf.fillna(0)
```

Out[16]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	0.0	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	0.0	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	0.0	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [19]:
```
m_v= df["Math_score"].mean()
df["Math_score"].fillna(value=m_v, inplace= True)
df
```

Out[19]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.000000	90.0	70.0	77	2018	2
1	70.000000	77.0	76.0	85	2018	3
2	62.000000	88.0	68.0	92	2019	3
3	94.000000	84.0	71.0	78	2019	2
4	78.000000	81.0	62.0	100	2020	3
5	77.000000	200.0	73.0	82	2020	2
6	77.000000	75.0	65.0	100	2020	3
7	62.000000	80.0	63.0	97	2019	3
8	79.000000	88.0	65.0	82	2018	2
9	63.000000	94.0	73.0	79	2021	1
10	66.000000	79.0	77.0	80	2019	2
11	69.000000	78.0	77.0	90	2020	3
12	80.000000	88.0	77.0	77	2018	2
13	66.000000	90.0	72.0	93	2020	3
14	64.000000	90.0	67.0	79	2020	2
15	78.000000	86.0	64.0	76	2019	2
16	61.000000	95.0	64.0	75	2021	2
17	180.000000	82.0	76.0	95	2019	3
18	80.000000	90.0	74.0	81	2019	2
19	70.000000	82.0	NaN	89	2020	3
20	69.000000	83.0	74.0	77	2021	2
21	71.000000	81.0	63.0	91	2020	3
22	71.000000	91.0	61.0	75	2020	2
23	61.000000	78.0	69.0	75	2020	2
24	75.103448	81.0	66.0	81	2019	2
25	76.000000	83.0	79.0	77	2019	2
26	68.000000	87.0	76.0	95	2020	3
27	61.000000	75.0	63.0	93	2020	3
28	80.000000	NaN	61.0	100	2020	3
29	65.000000	78.0	73.0	98	2018	3

In [20]:
```
ndf.replace(to_replace = np.nan, value=-99)
```

Out[20]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.000000	90.0	70.0	77	2018	2
1	70.000000	77.0	76.0	85	2018	3
2	62.000000	88.0	68.0	92	2019	3
3	94.000000	84.0	71.0	78	2019	2
4	78.000000	81.0	62.0	100	2020	3
5	77.000000	200.0	73.0	82	2020	2
6	77.000000	75.0	65.0	100	2020	3
7	62.000000	80.0	63.0	97	2019	3
8	79.000000	88.0	65.0	82	2018	2
9	63.000000	94.0	73.0	79	2021	1
10	66.000000	79.0	77.0	80	2019	2
11	69.000000	78.0	77.0	90	2020	3
12	80.000000	88.0	77.0	77	2018	2
13	66.000000	90.0	72.0	93	2020	3
14	64.000000	90.0	67.0	79	2020	2
15	78.000000	86.0	64.0	76	2019	2
16	61.000000	95.0	64.0	75	2021	2
17	180.000000	82.0	76.0	95	2019	3
18	80.000000	90.0	74.0	81	2019	2
19	70.000000	82.0	-99.0	89	2020	3
20	69.000000	83.0	74.0	77	2021	2
21	71.000000	81.0	63.0	91	2020	3
22	71.000000	91.0	61.0	75	2020	2
23	61.000000	78.0	69.0	75	2020	2
24	75.103448	81.0	66.0	81	2019	2
25	76.000000	83.0	79.0	77	2019	2
26	68.000000	87.0	76.0	95	2020	3
27	61.000000	75.0	63.0	93	2020	3
28	80.000000	-99.0	61.0	100	2020	3
29	65.000000	78.0	73.0	98	2018	3

In [21]:
```
ndf.dropna(how= 'all')
```

Out[21]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.000000	90.0	70.0	77	2018	2
1	70.000000	77.0	76.0	85	2018	3
2	62.000000	88.0	68.0	92	2019	3
3	94.000000	84.0	71.0	78	2019	2
4	78.000000	81.0	62.0	100	2020	3
5	77.000000	200.0	73.0	82	2020	2
6	77.000000	75.0	65.0	100	2020	3
7	62.000000	80.0	63.0	97	2019	3
8	79.000000	88.0	65.0	82	2018	2
9	63.000000	94.0	73.0	79	2021	1
10	66.000000	79.0	77.0	80	2019	2
11	69.000000	78.0	77.0	90	2020	3
12	80.000000	88.0	77.0	77	2018	2
13	66.000000	90.0	72.0	93	2020	3
14	64.000000	90.0	67.0	79	2020	2
15	78.000000	86.0	64.0	76	2019	2
16	61.000000	95.0	64.0	75	2021	2
17	180.000000	82.0	76.0	95	2019	3
18	80.000000	90.0	74.0	81	2019	2
19	70.000000	82.0	NaN	89	2020	3
20	69.000000	83.0	74.0	77	2021	2
21	71.000000	81.0	63.0	91	2020	3
22	71.000000	91.0	61.0	75	2020	2
23	61.000000	78.0	69.0	75	2020	2
24	75.103448	81.0	66.0	81	2019	2
25	76.000000	83.0	79.0	77	2019	2
26	68.000000	87.0	76.0	95	2020	3
27	61.000000	75.0	63.0	93	2020	3
28	80.000000	NaN	61.0	100	2020	3
29	65.000000	78.0	73.0	98	2018	3

In [22]:
```
ndf.dropna(axis=1)
```

Out[22]:

	Math_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.000000	77	2018	2
1	70.000000	85	2018	3
2	62.000000	92	2019	3
3	94.000000	78	2019	2
4	78.000000	100	2020	3
5	77.000000	82	2020	2
6	77.000000	100	2020	3
7	62.000000	97	2019	3
8	79.000000	82	2018	2
9	63.000000	79	2021	1
10	66.000000	80	2019	2
11	69.000000	90	2020	3
12	80.000000	77	2018	2
13	66.000000	93	2020	3
14	64.000000	79	2020	2
15	78.000000	76	2019	2
16	61.000000	75	2021	2
17	180.000000	95	2019	3
18	80.000000	81	2019	2
19	70.000000	89	2020	3
20	69.000000	77	2021	2
21	71.000000	91	2020	3
22	71.000000	75	2020	2
23	61.000000	75	2020	2
24	75.103448	81	2019	2
25	76.000000	77	2019	2
26	68.000000	95	2020	3
27	61.000000	93	2020	3
28	80.000000	100	2020	3
29	65.000000	98	2018	3

In [23]:
```
new_data = ndf.dropna(axis=0, how= 'any')
new_data
```

Out[23]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.000000	90.0	70.0	77	2018	2
1	70.000000	77.0	76.0	85	2018	3
2	62.000000	88.0	68.0	92	2019	3
3	94.000000	84.0	71.0	78	2019	2
4	78.000000	81.0	62.0	100	2020	3
5	77.000000	200.0	73.0	82	2020	2
6	77.000000	75.0	65.0	100	2020	3
7	62.000000	80.0	63.0	97	2019	3
8	79.000000	88.0	65.0	82	2018	2
9	63.000000	94.0	73.0	79	2021	1
10	66.000000	79.0	77.0	80	2019	2
11	69.000000	78.0	77.0	90	2020	3
12	80.000000	88.0	77.0	77	2018	2
13	66.000000	90.0	72.0	93	2020	3
14	64.000000	90.0	67.0	79	2020	2
15	78.000000	86.0	64.0	76	2019	2
16	61.000000	95.0	64.0	75	2021	2
17	180.000000	82.0	76.0	95	2019	3
18	80.000000	90.0	74.0	81	2019	2
20	69.000000	83.0	74.0	77	2021	2
21	71.000000	81.0	63.0	91	2020	3
22	71.000000	91.0	61.0	75	2020	2
23	61.000000	78.0	69.0	75	2020	2
24	75.103448	81.0	66.0	81	2019	2
25	76.000000	83.0	79.0	77	2019	2
26	68.000000	87.0	76.0	95	2020	3
27	61.000000	75.0	63.0	93	2020	3
29	65.000000	78.0	73.0	98	2018	3

In [ ]:

Handling Of Outliers

In [25]:
```
import pandas as pd
import numpy as np
df1= pd.read_csv("studentheight.csv")
```

In [26]:
```
df1
```

Out[26]:

	Name	Height
0	Akash	5.9
1	Ritesh	5.2
2	Shivam	5.1
3	Abhi	5.4
4	Shruti	6.5
5	Janhavi	7.1
6	John	14.2
7	Bob	5.6
8	Imran	1.2

In [27]:
```
df1.shape
```

Out[27]:
```
(9, 2)
```

In [32]:

```
df1['Height']
```

Out[32]:

```
0     5.9
1     5.2
2     5.1
3     5.4
4     6.5
5     7.1
6    14.2
7     5.6
8     1.2
Name: Height, dtype: float64
```

In [34]:

```
df1['Height'].quantile(0.95)
```

Out[34]:

```
11.359999999999996
```

Detect Outliers Using Percentile

In [35]:

```
max_thresold = df1['Height'].quantile(0.95)
max_thresold
```

Out[35]:

```
11.359999999999996
```

In [36]:

```
df1[df1['Height']>max_thresold]
```

Out[36]:

	Name	Height
6	John	14.2

In [37]:

```
min_thresold = df1['Height'].quantile(0.05)
min_thresold
```

Out[37]:

```
2.76
```

remove outliers

In [38]:

```
df1[(df1['Height']<max_thresold) & (df1['Height']> min_thresold)]
```

Out[38]:

	Name	Height
0	Akash	5.9
1	Ritesh	5.2
2	Shivam	5.1
3	Abhi	5.4
4	Shruti	6.5
5	Janhavi	7.1
7	Bob	5.6

In [40]:

```
df2 = df1[(df1['Height']<max_thresold) & (df1['Height']>min_thresold)]
df2.shape
```

Out[40]:

```
(7, 2)
```

In [41]:

```
df2.describe()
```

Out[41]:

	Height
count	7.000000
mean	5.828571
std	0.734199
min	5.100000
25%	5.300000
50%	5.600000
75%	6.200000
max	7.100000

In [42]:
```
df1.shape
```

Out[42]:
```
(9, 2)
```

In [43]:
```
df1.describe()
```

Out[43]:

	Height
count	9.000000
mean	6.244444
std	3.412884
min	1.200000
25%	5.200000
50%	5.600000
75%	6.500000
max	14.200000

In [44]:
```
import pandas as pd
import numpy as np
```

In [45]:
```
df2= pd.read_csv("studentsperformance.csv")
```

In [46]:
```
df2
```

Out[46]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

Outliers Virtualization- BOXPLOT

In [47]:

```
df2.describe()
```

Out[47]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
count	29.000000	29.000000	29.000000	30.000000	30.000000	30.000000
mean	75.103448	88.068966	69.620690	85.633333	2019.466667	2.433333
std	21.699810	22.238851	5.734693	8.833648	0.899553	0.568321
min	61.000000	75.000000	61.000000	75.000000	2018.000000	1.000000
25%	65.000000	80.000000	64.000000	77.250000	2019.000000	2.000000
50%	70.000000	83.000000	70.000000	82.000000	2020.000000	2.000000
75%	78.000000	90.000000	74.000000	93.000000	2020.000000	3.000000
max	180.000000	200.000000	79.000000	100.000000	2021.000000	3.000000

In [48]:

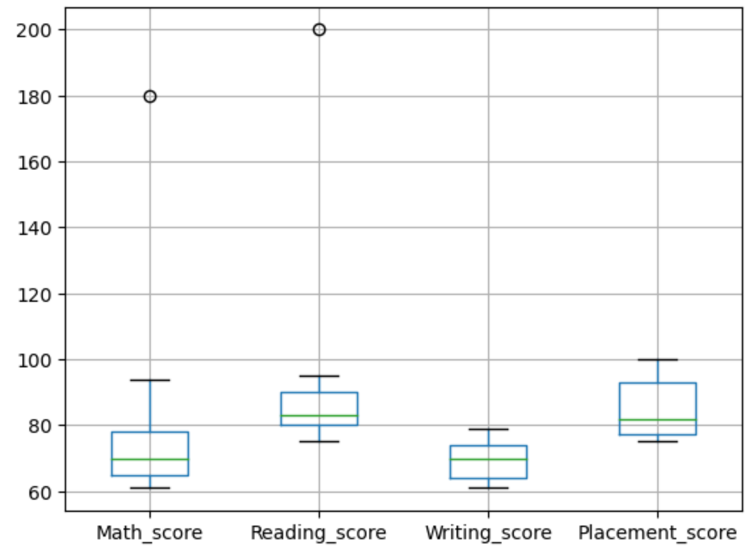
```
col = ['Math_score','Reading_score', 'Writing_score', 'Placement_score' ]
```

In [49]:

```
df2.boxplot(col)
```

Out[49]:

```
<AxesSubplot:>
```



In [50]:

```
print(np.where(df2['Math_score']>90))
print(np.where(df2['Reading_score']<25))
print(np.where(df2['Writing_score']<30))
```

```
(array([ 3, 17], dtype=int64),)
(array([], dtype=int64),)
(array([], dtype=int64),)
```

In [51]:

```
df2.shape
```

Out[51]:

```
(30, 6)
```

Detecting outliers by using IQR (Inter Quantile Range)

In [54]:

```
import pandas as pd
df = pd.read_csv("studentheight.csv")
```

In [55]:
```
df
```

Out[55]:

	Name	Height
0	Akash	5.9
1	Ritesh	5.2
2	Shivam	5.1
3	Abhi	5.4
4	Shruti	6.5
5	Janhavi	7.1
6	John	14.2
7	Bob	5.6
8	Imran	1.2

In [56]:
```
df.describe()
```

Out[56]:

	Height
count	9.000000
mean	6.244444
std	3.412884
min	1.200000
25%	5.200000
50%	5.600000
75%	6.500000
max	14.200000

In [58]:
```
Q1 = df.Height.quantile(0.25)
Q3 = df.Height.quantile(0.75)
Q1,Q3
```

Out[58]:
```
(5.2, 6.5)
```

In [59]:
```
IQR= Q3-Q1
IQR
```

Out[59]:
```
1.2999999999999998
```

In [60]:
```
lower_limit= Q1 - 1.5*IQR
upper_limit= Q3 + 1.5*IQR
lower_limit, upper_limit
```

Out[60]:
```
(3.2500000000000004, 8.45)
```

In [61]:
```
df[(df.Height<lower_limit)|(df.Height>upper_limit)]
```

Out[61]:

	Name	Height
6	John	14.2
8	Imran	1.2

trimming or removiung the outliers

In [63]:
```
df_no_outlier = df[(df.Height>lower_limit)|(df.Height<upper_limit)]
df_no_outlier
```

Out[63]:

	Name	Height
0	Akash	5.9
1	Ritesh	5.2
2	Shivam	5.1
3	Abhi	5.4
4	Shruti	6.5
5	Janhavi	7.1
6	John	14.2
7	Bob	5.6
8	Imran	1.2

```
In [64]: df.shape
```

```
Out[64]: (9, 2)
```

```
In [65]: df_no_outlier.shape
```

```
Out[65]: (9, 2)
```

IQR ON STUDENTS DATA

```
In [66]: import pandas as pd
         df= pd.read_csv("studentsperformance.csv")
```

```
In [112]: df
```

Out[112]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

```
In [113]: Q1 = df.Math_score.quantile(0.25)
          Q3 = df.Math_score.quantile(0.75)
          Q1, Q3
```

```
Out[113]: (65.0, 78.0)
```

In [114]:
```
df.describe()
```

Out[114]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
count	29.000000	29.000000	29.000000	30.000000	30.000000	30.000000
mean	75.103448	88.068966	69.620690	85.633333	2019.466667	2.433333
std	21.699810	22.238851	5.734693	8.833648	0.899553	0.568321
min	61.000000	75.000000	61.000000	75.000000	2018.000000	1.000000
25%	65.000000	80.000000	64.000000	77.250000	2019.000000	2.000000
50%	70.000000	83.000000	70.000000	82.000000	2020.000000	2.000000
75%	78.000000	90.000000	74.000000	93.000000	2020.000000	3.000000
max	180.000000	200.000000	79.000000	100.000000	2021.000000	3.000000

In [115]:
```
IQR= Q3-Q1
IQR
```

Out[115]:
```
13.0
```

In [70]:
```
lower_limit = Q1 - 1.5*IQR
upper_limit = Q3 - 1.5*IQR
lower_limit,upper_limit
```

Out[70]:
```
(45.5, 58.5)
```

In [116]:
```
df[(df.Math_score<lower_limit) | (df.Math_score>upper_limit)]
```

Out[116]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

trimming 0 or removing outlier

In [72]:
```
df_no_outlier = df[(df.Math_score>lower_limit) | (df.Math_score<upper_limit)]
df_no_outlier
```

Out[72]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [73]:
```
df.shape
```

Out[73]:
```
(30, 6)
```

In [74]:
```
df_no_outlier.shape
```

Out[74]:
```
(29, 6)
```

In [120]:
```
import pandas as pd
import numpy as np
```

In [125]:
```
q1 = np.percentile(df['Placement_score'], 25)
q3 = np.percentile(df['Placement_score'], 75)
print(q1,q3)
```

```
77.25 93.0
```

In [126]:
```
IQR = q3-q1
IQR
```

Out[126]:
```
15.75
```

In [127]:
```
lwr_bound = q1-(1.5*IQR)
upr_bound = q3+(1.5*IQR)
print(lwr_bound, upr_bound)
```

```
53.625 116.625
```

In [128]:
```
col = ['Placement_score']
```

In [129]:
```
index_outliers = np.where((df.Placement_score<lwr_bound) | (df.Placement_score>upr_bound))
```

In [130]:
```
df
```

Out[130]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [131]:
```
sample_outliers = df[col][(df[col]<lwr_bound) | (df[col]>upr_bound) ]
sample_outliers
```

Out[131]:

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

Handling of outliers

1. Quantile Based Flooring And Caping

The outlier is capped at certain value above 90th percentile value and floored below 10th percentile value

In [132]:

```
df1= df
df[col]= np.where(df[col]<lwr_bound,lwr_bound,df[col])
df[col]= np.where(df[col]>upr_bound,upr_bound,df[col])
df
```

Out[132]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77.0	2018	2
1	70.0	77.0	76.0	85.0	2018	3
2	62.0	88.0	68.0	92.0	2019	3
3	94.0	84.0	71.0	78.0	2019	2
4	78.0	81.0	62.0	100.0	2020	3
5	77.0	200.0	73.0	82.0	2020	2
6	77.0	75.0	65.0	100.0	2020	3
7	62.0	80.0	63.0	97.0	2019	3
8	79.0	88.0	65.0	82.0	2018	2
9	63.0	94.0	73.0	79.0	2021	1
10	66.0	79.0	77.0	80.0	2019	2
11	69.0	78.0	77.0	90.0	2020	3
12	80.0	88.0	77.0	77.0	2018	2
13	66.0	90.0	72.0	93.0	2020	3
14	64.0	90.0	67.0	79.0	2020	2
15	78.0	86.0	64.0	76.0	2019	2
16	61.0	95.0	64.0	75.0	2021	2
17	180.0	82.0	76.0	95.0	2019	3
18	80.0	90.0	74.0	81.0	2019	2
19	70.0	82.0	NaN	89.0	2020	3
20	69.0	83.0	74.0	77.0	2021	2
21	71.0	81.0	63.0	91.0	2020	3
22	71.0	91.0	61.0	75.0	2020	2
23	61.0	78.0	69.0	75.0	2020	2
24	NaN	81.0	66.0	81.0	2019	2
25	76.0	83.0	79.0	77.0	2019	2
26	68.0	87.0	76.0	95.0	2020	3
27	61.0	75.0	63.0	93.0	2020	3
28	80.0	NaN	61.0	100.0	2020	3
29	65.0	78.0	73.0	98.0	2018	3

In [133]:

```
ninetieth_percentile = np.percentile(df1['Placement_score'],90)
ninetieth_percentile
```

Out[133]:

```
98.2
```

In [137]:

```
df1[col]= np.where(df1[col]>upr_bound, ninetieth_percentile, df1[col])
df1
```

Out[137]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77.0	2018	2
1	70.0	77.0	76.0	85.0	2018	3
2	62.0	88.0	68.0	92.0	2019	3
3	94.0	84.0	71.0	78.0	2019	2
4	78.0	81.0	62.0	100.0	2020	3
5	77.0	200.0	73.0	82.0	2020	2
6	77.0	75.0	65.0	100.0	2020	3
7	62.0	80.0	63.0	97.0	2019	3
8	79.0	88.0	65.0	82.0	2018	2
9	63.0	94.0	73.0	79.0	2021	1
10	66.0	79.0	77.0	80.0	2019	2
11	69.0	78.0	77.0	90.0	2020	3
12	80.0	88.0	77.0	77.0	2018	2
13	66.0	90.0	72.0	93.0	2020	3
14	64.0	90.0	67.0	79.0	2020	2
15	78.0	86.0	64.0	76.0	2019	2
16	61.0	95.0	64.0	75.0	2021	2
17	180.0	82.0	76.0	95.0	2019	3
18	80.0	90.0	74.0	81.0	2019	2
19	70.0	82.0	NaN	89.0	2020	3
20	69.0	83.0	74.0	77.0	2021	2
21	71.0	81.0	63.0	91.0	2020	3
22	71.0	91.0	61.0	75.0	2020	2
23	61.0	78.0	69.0	75.0	2020	2
24	NaN	81.0	66.0	81.0	2019	2
25	76.0	83.0	79.0	77.0	2019	2
26	68.0	87.0	76.0	95.0	2020	3
27	61.0	75.0	63.0	93.0	2020	3
28	80.0	NaN	61.0	100.0	2020	3
29	65.0	78.0	73.0	98.0	2018	3

In [139]:

```
tenth_percentile = np.percentile(df1['Placement_score'],10)
tenth_percentile
```

Out[139]:

```
75.9
```

In [140]:
```
df1[col]= np.where(df1[col]<lwr_bound, tenth_percentile, df1[col])
df1
```

Out[140]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77.0	2018	2
1	70.0	77.0	76.0	85.0	2018	3
2	62.0	88.0	68.0	92.0	2019	3
3	94.0	84.0	71.0	78.0	2019	2
4	78.0	81.0	62.0	100.0	2020	3
5	77.0	200.0	73.0	82.0	2020	2
6	77.0	75.0	65.0	100.0	2020	3
7	62.0	80.0	63.0	97.0	2019	3
8	79.0	88.0	65.0	82.0	2018	2
9	63.0	94.0	73.0	79.0	2021	1
10	66.0	79.0	77.0	80.0	2019	2
11	69.0	78.0	77.0	90.0	2020	3
12	80.0	88.0	77.0	77.0	2018	2
13	66.0	90.0	72.0	93.0	2020	3
14	64.0	90.0	67.0	79.0	2020	2
15	78.0	86.0	64.0	76.0	2019	2
16	61.0	95.0	64.0	75.0	2021	2
17	180.0	82.0	76.0	95.0	2019	3
18	80.0	90.0	74.0	81.0	2019	2
19	70.0	82.0	NaN	89.0	2020	3
20	69.0	83.0	74.0	77.0	2021	2
21	71.0	81.0	63.0	91.0	2020	3
22	71.0	91.0	61.0	75.0	2020	2
23	61.0	78.0	69.0	75.0	2020	2
24	NaN	81.0	66.0	81.0	2019	2
25	76.0	83.0	79.0	77.0	2019	2
26	68.0	87.0	76.0	95.0	2020	3
27	61.0	75.0	63.0	93.0	2020	3
28	80.0	NaN	61.0	100.0	2020	3
29	65.0	78.0	73.0	98.0	2018	3

Handling outlier using Median Value

In [141]:
```
new_df = pd.read_csv("studentsperformance.csv")
```

In [142]:
```
new_df
```

Out[142]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [147]:
```
median= np.median(new_df[col])
median
```

Out[147]:
```
82.0
```

Detecting outlier using z score

In [150]:
```
import numpy as np
from scipy import stats
```

In [151]:
```
df3= pd.read_csv("studentsperformance.csv")
df3
```

Out[151]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

In [152]:
```
df3.shape
```

Out[152]:
```
(30, 6)
```

In [155]:
```
z= np.abs(stats.zscore(df3["Placement_score"]))
print(z)
```

```
0     0.994031
1     0.072921
2     0.733050
3     0.878893
4     1.654160
5     0.418337
6     1.654160
7     1.308744
8     0.418337
9     0.763754
10    0.648615
11    0.502773
12    0.994031
13    0.848189
14    0.763754
15    1.109170
16    1.224309
17    1.078466
18    0.533476
19    0.387634
20    0.994031
21    0.617911
22    1.224309
23    1.224309
24    0.533476
25    0.994031
26    1.078466
27    0.848189
28    1.654160
29    1.423883
Name: Placement_score, dtype: float64
```

In [158]:
```
thresold= 0.60
```

In [159]:
```
sample_outliers = np.where(z<thresold)
sample_outliers
```

Out[159]:
```
(array([ 1,  5,  8, 11, 18, 19, 24], dtype=int64),)
```

In [162]:
```
upperthresold = 1.4
lowerthresold = 0.60

index_outliers = np.where((z<lowerthresold) | (z>upperthresold))
index_outliers
```

Out[162]:
```
(array([ 1,  4,  5,  6,  8, 11, 18, 19, 24, 28, 29], dtype=int64),)
```

Module 2

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

In [2]:
```
import matplotlib.pyplot as plt
```

In [3]:
```
df4= pd.read_csv("normalizationdata.csv")
```

In [4]:
```
df4
```

Out[4]:

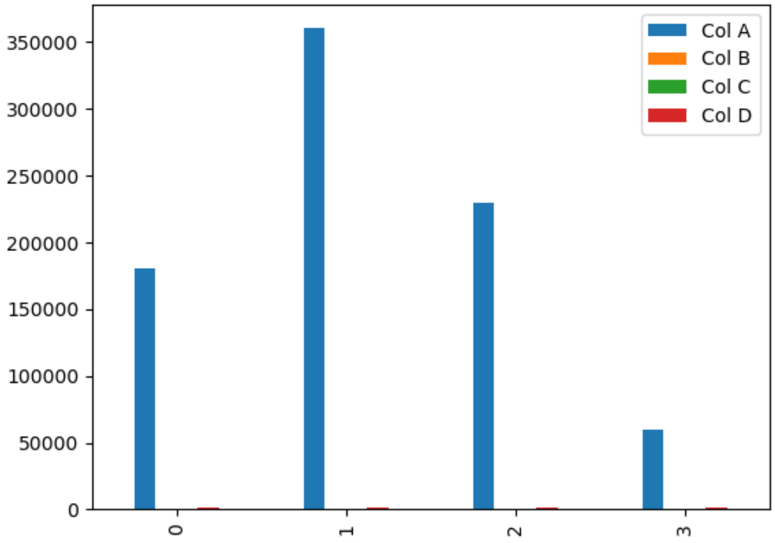
	Col A	Col B	Col C	Col D
0	180000	100	18.9	1400
1	360000	900	23.4	1000
2	230000	230	14.0	1300
3	60000	450	13.5	1500

In [5]:

```
df4.plot(kind = 'bar')
```

Out[5]:

```
<AxesSubplot:>
```



In [31]:

```
df_max_scaled = df4.copy()
for column in df_max_scaled.columns:
    df_max_scaled[column]= df_max_scaled[column]/df_max_scaled[column].abs().max()
```

In [32]:

```
df_max_scaled
```

Out[32]:

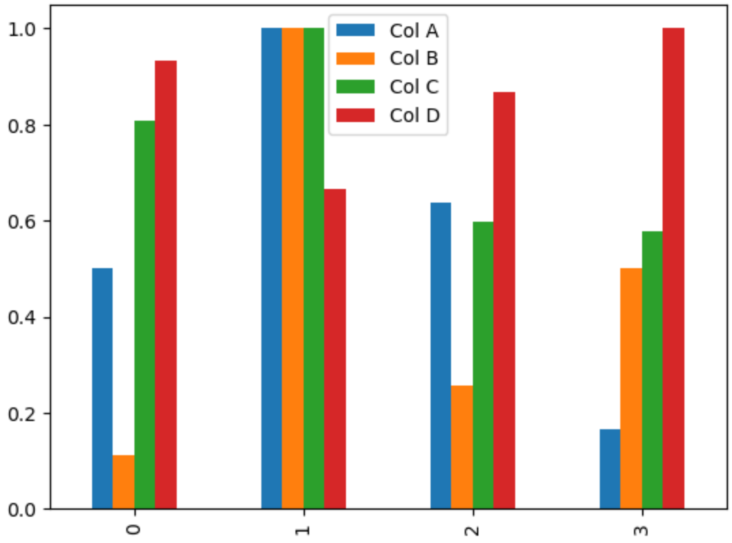
	Col A	Col B	Col C	Col D
0	0.500000	0.111111	0.807692	0.933333
1	1.000000	1.000000	1.000000	0.666667
2	0.638889	0.255556	0.598291	0.866667
3	0.166667	0.500000	0.576923	1.000000

In [9]:

```
df_max_scaled.plot(kind ='bar')
```

Out[9]:

```
<AxesSubplot:>
```



In [53]:

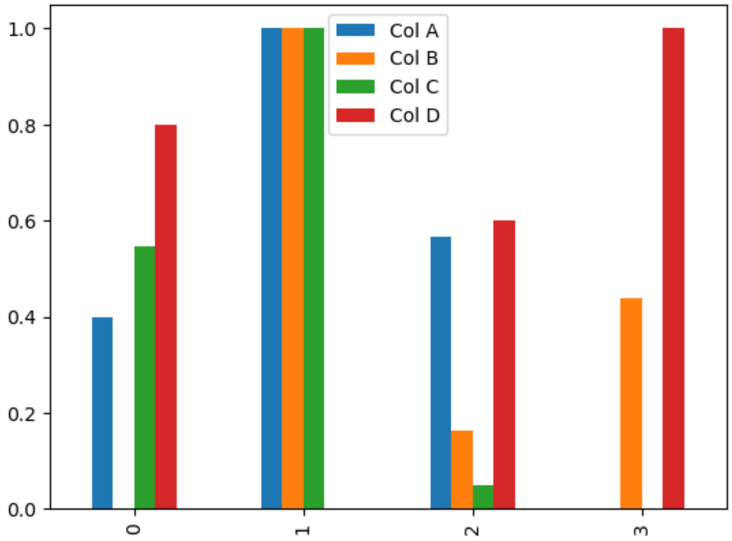
```
df_min_max_scaled = df4.copy()
for column in df_min_max_scaled.columns:
    df_min_max_scaled[column] = (df_min_max_scaled[column] - df_min_max_scaled[column].min()) / (df_min_max_scaled[column].ma
```

In [54]:
```
print(df_min_max_scaled)
```

```
      Col A   Col B     Col C  Col D
0  0.400000  0.0000  0.545455    0.8
1  1.000000  1.0000  1.000000    0.0
2  0.566667  0.1625  0.050505    0.6
3  0.000000  0.4375  0.000000    1.0
```

In [56]:
```
print(df_min_max_scaled.plot(kind= 'bar'))
```

```
AxesSubplot(0.125,0.11;0.775x0.77)
```

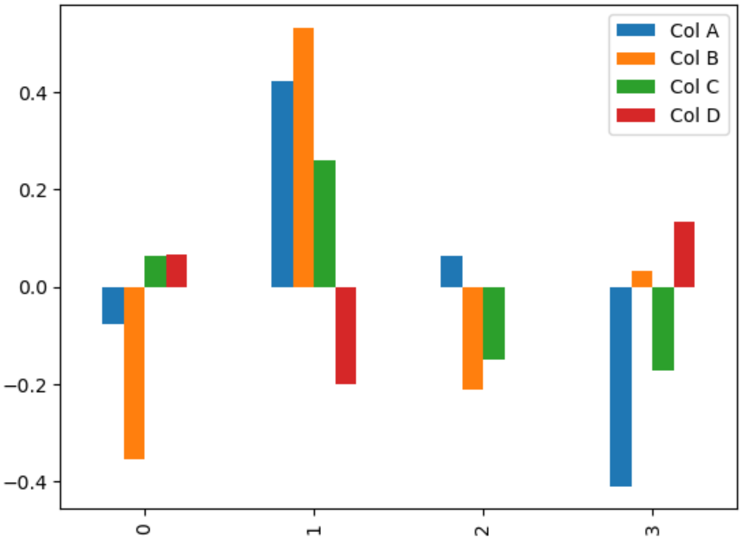


In [57]:
```
df_z_scaled = df4.copy()
for column in df_z_scaled.columns:
    df_z_scaled[column] = (df_z_scaled[column] - df_z_scaled[column].mean()) / (df_z_scaled[column].max() - df_min_max_scaled
display(df_z_scaled)
```

	Col A	Col B	Col C	Col D
0	-0.076389	-0.355729	0.063237	0.066686
1	0.423612	0.533594	0.259488	-0.200058
2	0.062500	-0.211214	-0.150459	0.000000
3	-0.409723	0.033350	-0.172265	0.133372

In [58]:
```
df_z_scaled.plot(kind= 'bar')
```

Out[58]:
```
<AxesSubplot:>
```



Applying normalization technique to student dataset

In [59]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

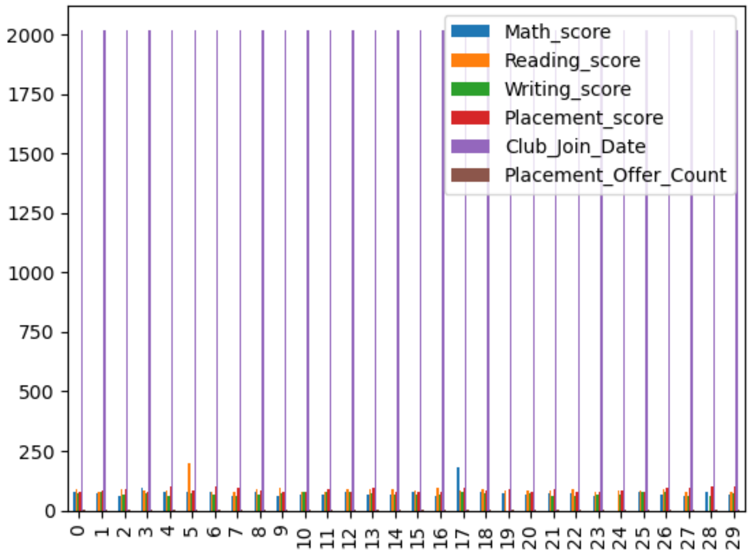
In [61]:
```
df5= pd.read_csv('studentsperformance.csv')
df5
```

Out[61]:

	Math_score	Reading_score	Writing_score	Placement_score	Club_Join_Date	Placement_Offer_Count
0	80.0	90.0	70.0	77	2018	2
1	70.0	77.0	76.0	85	2018	3
2	62.0	88.0	68.0	92	2019	3
3	94.0	84.0	71.0	78	2019	2
4	78.0	81.0	62.0	100	2020	3
5	77.0	200.0	73.0	82	2020	2
6	77.0	75.0	65.0	100	2020	3
7	62.0	80.0	63.0	97	2019	3
8	79.0	88.0	65.0	82	2018	2
9	63.0	94.0	73.0	79	2021	1
10	66.0	79.0	77.0	80	2019	2
11	69.0	78.0	77.0	90	2020	3
12	80.0	88.0	77.0	77	2018	2
13	66.0	90.0	72.0	93	2020	3
14	64.0	90.0	67.0	79	2020	2
15	78.0	86.0	64.0	76	2019	2
16	61.0	95.0	64.0	75	2021	2
17	180.0	82.0	76.0	95	2019	3
18	80.0	90.0	74.0	81	2019	2
19	70.0	82.0	NaN	89	2020	3
20	69.0	83.0	74.0	77	2021	2
21	71.0	81.0	63.0	91	2020	3
22	71.0	91.0	61.0	75	2020	2
23	61.0	78.0	69.0	75	2020	2
24	NaN	81.0	66.0	81	2019	2
25	76.0	83.0	79.0	77	2019	2
26	68.0	87.0	76.0	95	2020	3
27	61.0	75.0	63.0	93	2020	3
28	80.0	NaN	61.0	100	2020	3
29	65.0	78.0	73.0	98	2018	3

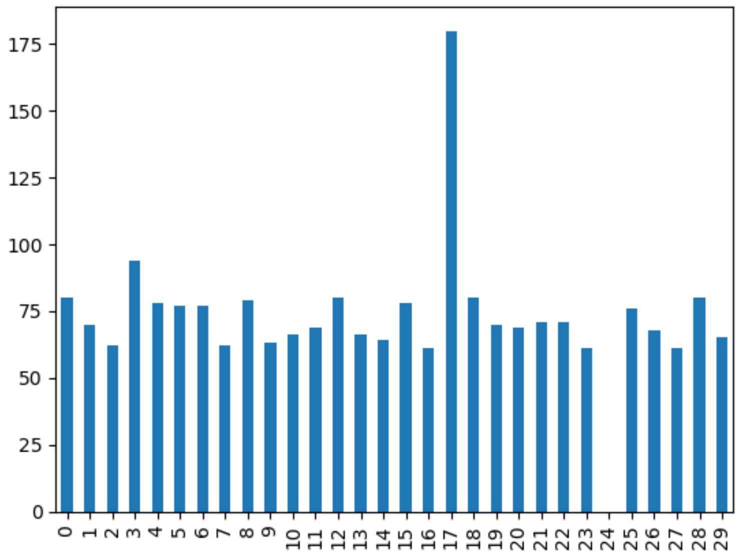
```
In [62]: df5.plot(kind = 'bar')
```

```
Out[62]: <AxesSubplot:>
```



```
In [63]: df5['Math_score'].plot(kind='bar')
```

```
Out[63]: <AxesSubplot:>
```



In [64]:

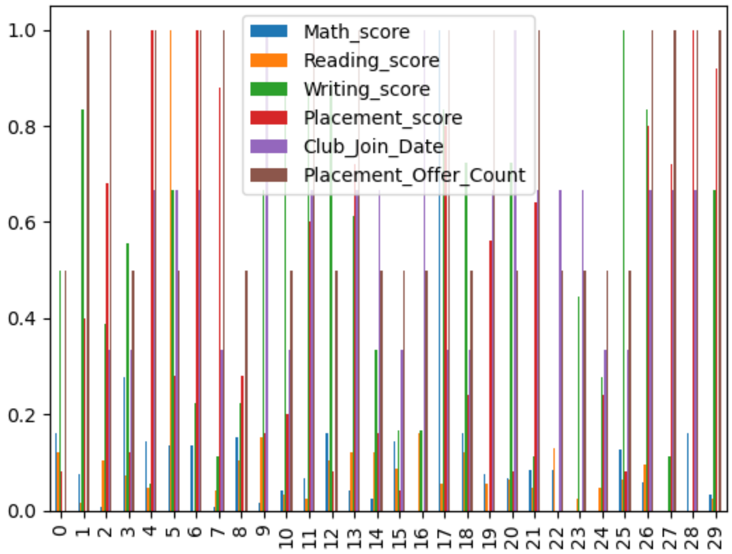
```
df_min_max_scaled = df5.copy()
for column in df_min_max_scaled.columns:
    df_min_max_scaled[column] = (df_min_max_scaled[column] - df_min_max_scaled[column].min()) / (df_min_max_scaled[column].ma
print(df_min_max_scaled)
```

```
    Math_score  Reading_score  Writing_score  Placement_score  Club_Join_Date  \
0     0.159664          0.120       0.500000             0.08        0.000000   
1     0.075630          0.016       0.833333             0.40        0.000000   
2     0.008403          0.104       0.388889             0.68        0.333333   
3     0.277311          0.072       0.555556             0.12        0.333333   
4     0.142857          0.048       0.055556             1.00        0.666667   
5     0.134454          1.000       0.666667             0.28        0.666667   
6     0.134454          0.000       0.222222             1.00        0.666667   
7     0.008403          0.040       0.111111             0.88        0.333333   
8     0.151261          0.104       0.222222             0.28        0.000000   
9     0.016807          0.152       0.666667             0.16        1.000000   
10    0.042017          0.032       0.888889             0.20        0.333333   
11    0.067227          0.024       0.888889             0.60        0.666667   
12    0.159664          0.104       0.888889             0.08        0.000000   
13    0.042017          0.120       0.611111             0.72        0.666667   
14    0.025210          0.120       0.333333             0.16        0.666667   
15    0.142857          0.088       0.166667             0.04        0.333333   
16    0.000000          0.160       0.166667             0.00        1.000000   
17    1.000000          0.056       0.833333             0.80        0.333333   
18    0.159664          0.120       0.722222             0.24        0.333333   
19    0.075630          0.056            NaN             0.56        0.666667   
20    0.067227          0.064       0.722222             0.08        1.000000   
21    0.084034          0.048       0.111111             0.64        0.666667   
22    0.084034          0.128       0.000000             0.00        0.666667   
23    0.000000          0.024       0.444444             0.00        0.666667   
24         NaN          0.048       0.277778             0.24        0.333333   
25    0.126050          0.064       1.000000             0.08        0.333333   
26    0.058824          0.096       0.833333             0.80        0.666667   
27    0.000000          0.000       0.111111             0.72        0.666667   
28    0.159664            NaN       0.000000             1.00        0.666667   
29    0.033613          0.024       0.666667             0.92        0.000000   

    Placement_Offer_Count  
0                     0.5  
1                     1.0  
2                     1.0  
3                     0.5  
4                     1.0  
5                     0.5  
6                     1.0  
7                     1.0  
8                     0.5  
9                     0.0  
10                    0.5  
11                    1.0  
12                    0.5  
13                    1.0  
14                    0.5  
15                    0.5  
16                    0.5  
17                    1.0  
18                    0.5  
19                    1.0  
20                    0.5  
21                    1.0  
22                    0.5  
23                    0.5  
24                    0.5  
25                    0.5  
26                    1.0  
27                    1.0  
28                    1.0  
29                    1.0  
```

```
In [65]: df_min_max_scaled.plot(kind='bar')
```

```
Out[65]: <AxesSubplot:>
```



```
In [66]: df_min_max_scaled.skew()
```

```
Out[66]: Math_score               4.286198
         Reading_score            4.851150
         Writing_score           -0.021881
         Placement_score          0.356732
         Club_Join_Date          -0.198060
         Placement_Offer_Count   -0.325614
         dtype: float64
```

```
In [ ]:
```