Lab Assignment no 2

Aim:Create an “Academic performance” dataset of students and perform the following operations using Python.

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

In [15]:
```
import pandas as pd
file_path=r"C:\Users\CNLAB13\Desktop\StudentPerformance.csv"
df=pd.read_csv(file_path)
df.head()

```

Out[15]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2

In [29]:
```
1 df.isnull()
```

Out[29]:

	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	False	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	False	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	False	False	False	False	False

In [37]:
```
1 seseries = pd.notnull(df["Math_Score"])
2 df[series]
```

Out[37]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count

In [32]:
```
series = pd.isnull(df["Reading_Score"])
df[series]
```

Out[32]:

Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count

In [28]:
```
df.notnull()
```

Out[28]:

	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	True	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	True	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	True	True	True	True	True

In [35]:
```
seseries = pd.notnull(df["Reading_Score"])
df[series]
```

Out[35]:

Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count

In [40]:
```
df
```

Out[40]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [44]:
```
ndf=df
ndf.fillna(0)
```

Out[44]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [45]:
```
m_v=df['Reading_Score'].mean()
df['Reading_Score'].fillna(value=m_v, inplace=True)
df
```

Out[45]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [46]:
```
1 ndf.replace(to_replace = np.nan, value = -99)
```

Out[46]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [47]:
```
1 ndf.dropna()
```

Out[47]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [48]:
```
1 ndf.dropna(how = 'all')
```

Out[48]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [49]:
```
1 ndf.dropna(axis = 1)
```

Out[49]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

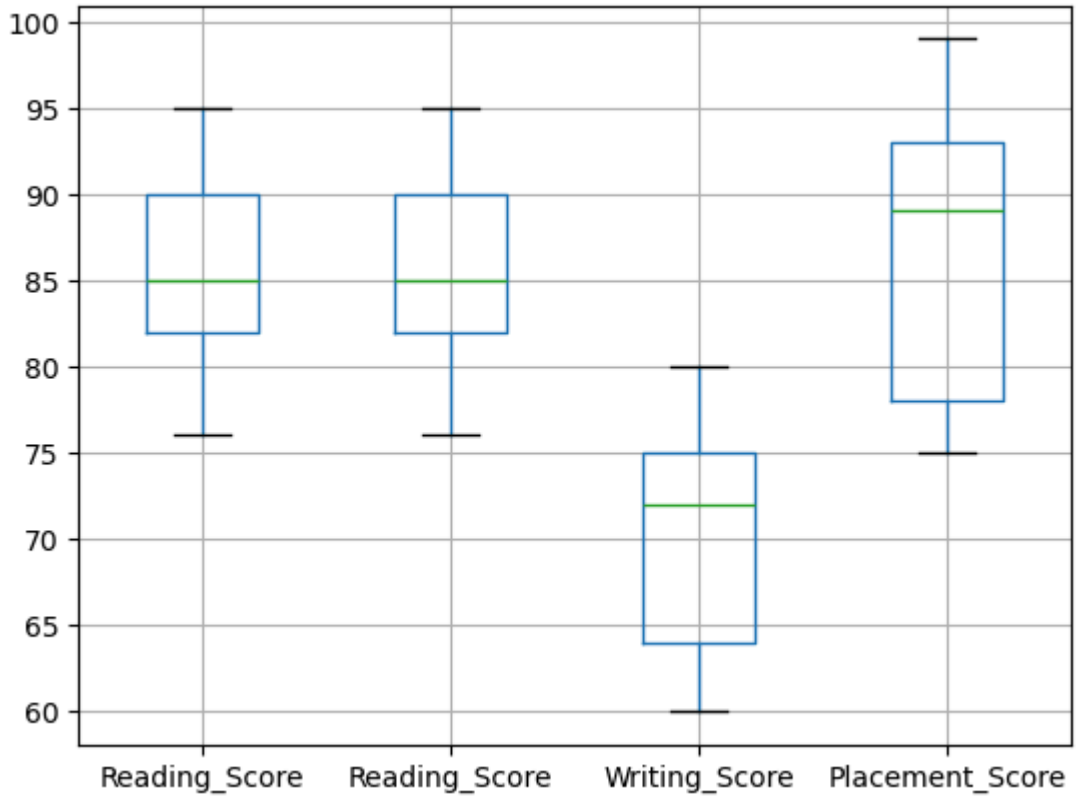
In [50]:
```
new_data =ndf.dropna (axis = 0, how ='any')
new_data
```

Out[50]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
2	64	86	70	93	2018	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
5	77	85	60	97	2021	3
6	76	85	61	99	2018	3
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
9	69	86	60	93	2018	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [51]:
```
col =['Reading_Score', 'Reading_Score', 'Writing_Score', 'Placement_Sc
df.boxplot(col)
```

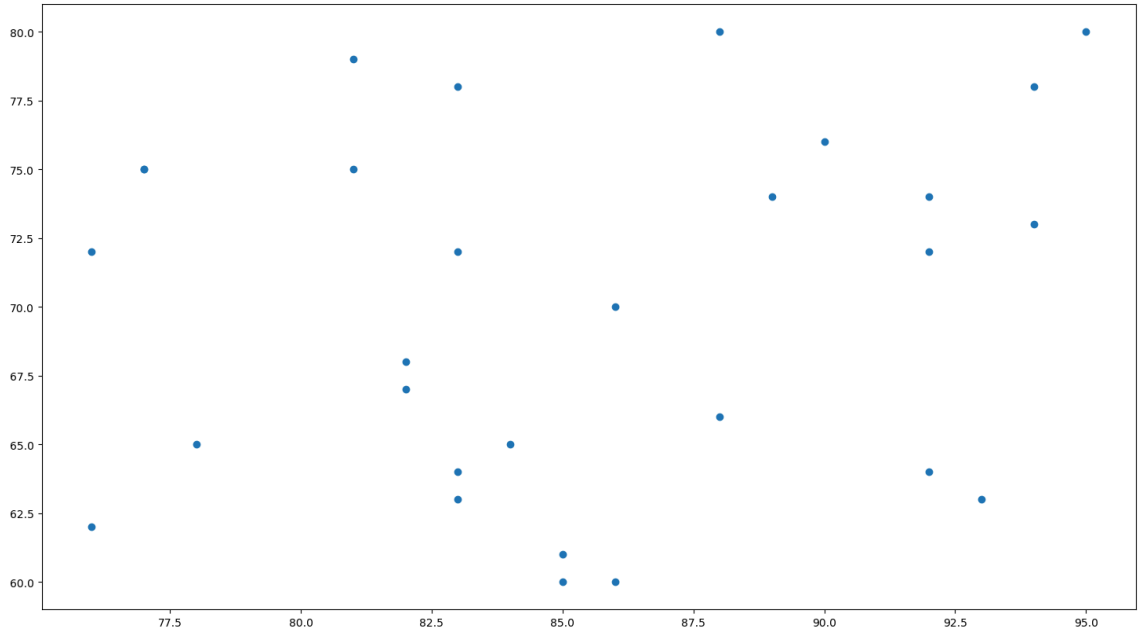
Out[51]: <Axes: >



In [53]:
```
print(np.where(df['Reading_Score']>90))
print(np.where(df['Writing_Score']>90))
```

```
(array([13, 15, 20, 21, 23, 24, 25], dtype=int64),)
(array([], dtype=int64),)
```

In [54]:
```
fig, ax =plt.subplots(figsize = (18,10))
ax.scatter(df['Reading_Score'], df['Writing_Score'])
plt.show()
ax.set_xlabel('(Proportion non-retail business acres)/(town)')
ax.set_ylabel('(Full-value property-tax rate)/($10,000)')
```



Out[54]: Text(4.444444444444452, 0.5, '(Full-value property-tax rate)/($10,000)')

In [55]:
```
print(np.where((df['Reading_Score']<50) & (df['Writing_Score']>1)))
print(np.where((df['Reading_Score']>85) & (df['Writing_Score']<3)))
```

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

In [56]:
```
z = np.abs(stats.zscore(df['Reading_Score']))
```

In [57]:
```
1 print(z)
```

```
0     1.468421
1     0.467225
2     0.115289
3     0.412614
4     1.292453
5     0.060679
6     0.060679
7     1.644388
8     1.468421
9     0.115289
10    0.819160
11    0.588582
12    0.467225
13    1.171096
14    0.412614
15    1.523031
16    0.412614
17    0.412614
18    0.764550
19    0.643193
20    1.171096
21    1.171096
22    0.764550
23    1.523031
24    1.347064
25    1.698999
26    1.644388
27    0.588582
28    0.236646
Name: Reading_Score, dtype: float64
```

In [58]:
```
1 threshold = 0.18
```

In [59]:
```
1 sample_outliers = np.where(z <threshold)
```

In [60]:
```
1 sample_outliers
```

Out[60]:
```
(array([2, 5, 6, 9], dtype=int64),)
```

In [61]:
```
1 sorted_rscore= sorted(df['Reading_Score'])
```

In [62]:
```
1 sorted_rscore
```

Out[62]:
```
[76,
 76,
 77,
 77,
 78,
 81,
 81,
 82,
 82,
 83,
 83,
 83,
 83,
 84,
 85,
 85,
 86,
 86,
 88,
 88,
 89,
 90,
 92,
 92,
 92,
 93,
 94,
 94,
 95]
```

In [63]:
```
1 q1 = np.percentile(sorted_rscore, 25)
2 q3 = np.percentile(sorted_rscore, 75)
3 print(q1,q3)
```

```
82.0 90.0
```

In [64]:
```
1 IQR = q3-q1
```

In [65]:
```
1 lwr_bound = q1-(1.5*IQR)
2 upr_bound = q3+(1.5*IQR)
3 print(lwr_bound, upr_bound)
```

```
70.0 102.0
```

In [66]:
```
new_df=df
for i in sample_outliers:new_df.drop(i,inplace=True)
new_df
```

Out[66]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	64	77	75	93	2020	3
1	78	88	80	93	2020	3
3	94	83	63	98	2021	3
4	61	78	65	84	2020	2
7	64	76	62	77	2019	2
8	75	77	75	91	2020	3
10	61	90	76	92	2019	3
11	68	82	68	89	2019	3
12	72	88	66	77	2018	2
13	79	92	64	78	2020	2
14	73	83	64	76	2020	2
15	64	94	73	83	2021	2
16	74	83	72	99	2020	3
17	60	83	78	75	2020	2
18	65	81	75	92	2020	3
19	63	89	74	83	2018	2
20	80	92	74	75	2019	2
21	71	92	72	93	2018	3
22	72	81	79	89	2020	3
23	62	94	78	79	2018	2
24	74	93	63	89	2021	3
25	63	95	80	76	2018	2
26	65	76	72	77	2021	2
27	65	82	67	81	2019	2
28	79	84	65	91	2018	3

In [67]:
```
file_path=r"C:\Users\CNLAB13\Desktop\StudentPerformance.csv"
df=pd.read_csv(file_path)
```

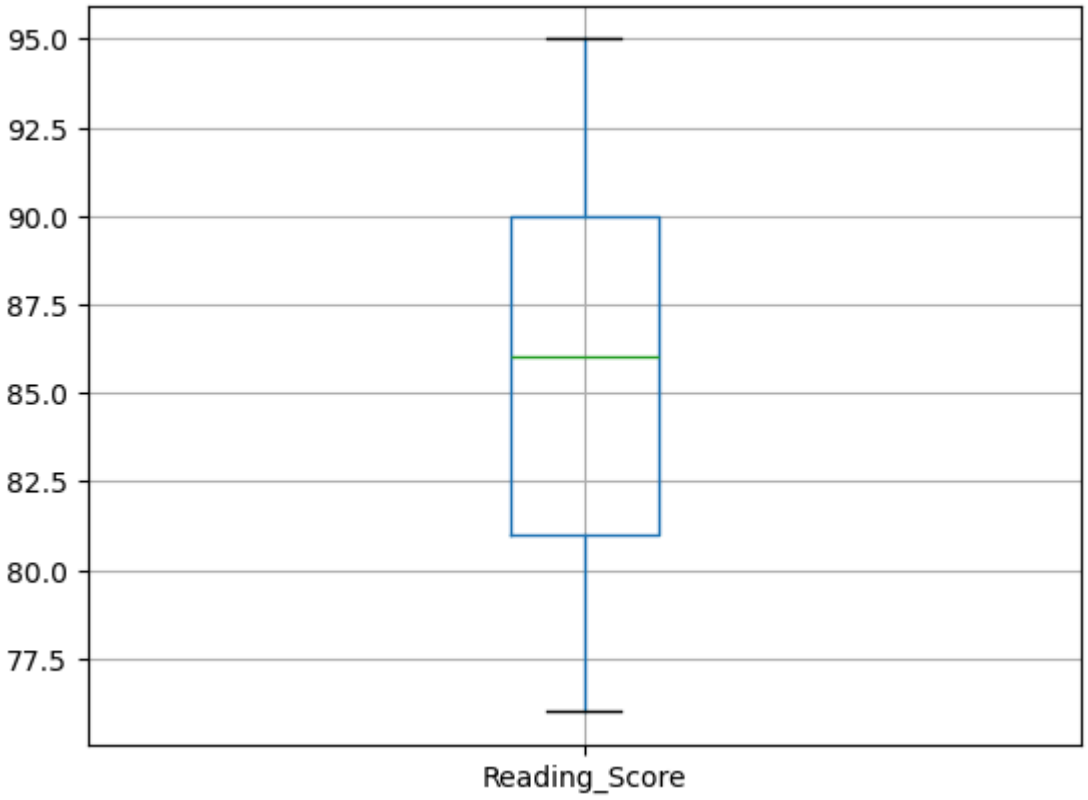
In [68]:
```
df_stud=df
ninetieth_percentile = np.percentile(df_stud['Reading_Score'], 90)
b = np.where(df_stud['Reading_Score']>ninetieth_percentile,
ninetieth_percentile, df_stud['Reading_Score'])
print("New array:",b)
```

```
New array: [84.  80.  91.  86.  90.  87.  76.  79.  82.  89.  76.  81.  9
2.4 79.
 89.  92.4 82.  90.  88.  89.  85.  88.  82.  92.  81.  92.4 77.  82.
 92. ]
```

In [69]:
```
col = ['Reading_Score']

df.boxplot(col)
```

Out[69]: <Axes: >



In [70]:
```
median=np.median(sorted_rscore)
median
```

Out[70]: 85.0

In [71]:
```
refined_df=df
refined_df['Reading_Score'] = np.where(refined_df['Reading_Score'] >upr
```

In [72]:
```
1 df
```

Out[72]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	78	84.0	62	96	2021	3
1	77	80.0	72	97	2019	3
2	64	91.0	67	94	2021	3
3	94	86.0	67	77	2019	2
4	62	90.0	69	80	2018	2
5	70	87.0	62	77	2021	2
6	67	76.0	64	88	2021	3
7	64	79.0	71	76	2018	2
8	76	82.0	80	77	2019	2
9	70	89.0	80	83	2018	2
10	80	76.0	71	96	2020	3
11	75	81.0	71	95	2018	3
12	NaN	94.0	61	99	2021	3
13	76	79.0	65	91	2018	3
14	66	89.0	61	90	2019	3
15	74	95.0	77	95	2019	3
16	74	82.0	67	75	2019	2
17	70	90.0	68	89	2021	3
18	79	88.0	61	91	2019	3
19	80	89.0	76	85	2021	3
20	79	85.0	67	95	2020	3
21	62	88.0	67	98	2021	3
22	61	82.0	77	96	2018	3
23	63	92.0	79	88	2021	3
24	79	81.0	68	82	2019	2
25	68	94.0	63	76	2020	2
26	76	77.0	77	100	2019	3
27	79	82.0	67	89	2020	3
28	68	92.0	72	83	2021	2

In [73]:
```
1 refined_df['Reading_Score'] = np.where(refined_df['Reading_Score'] <lwr
```

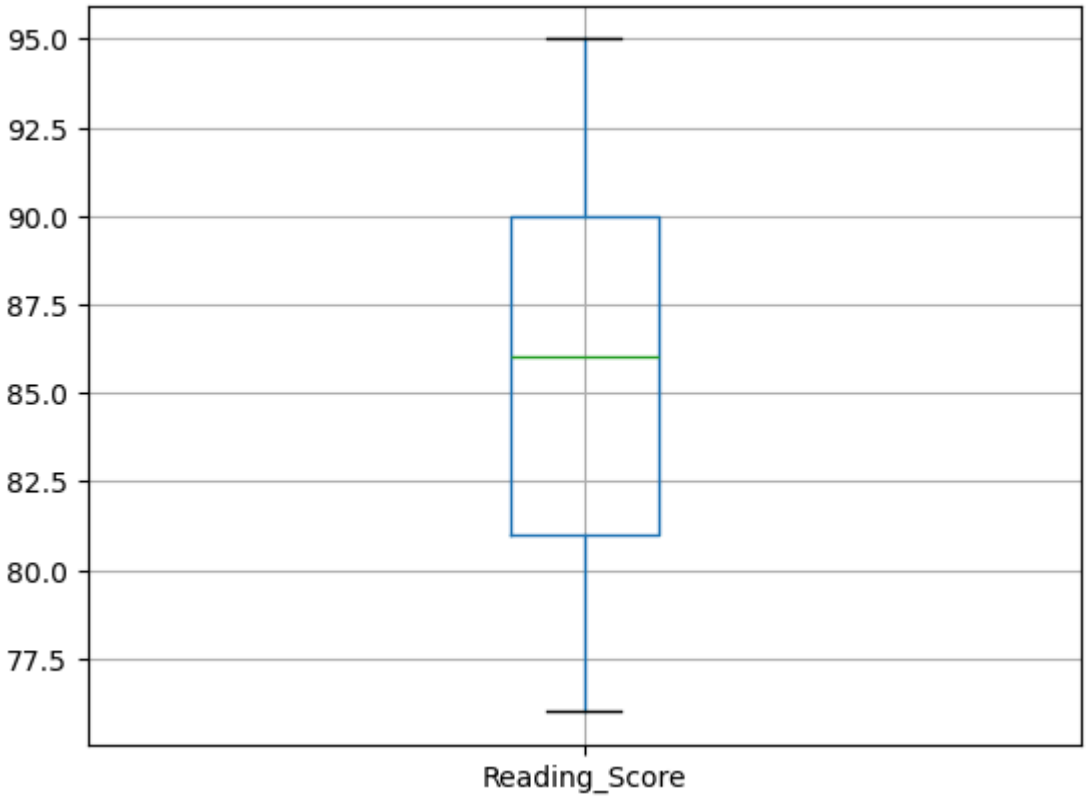
In [74]:
```
1 df
```

Out[74]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	78	84.0	62	96	2021	3
1	77	80.0	72	97	2019	3
2	64	91.0	67	94	2021	3
3	94	86.0	67	77	2019	2
4	62	90.0	69	80	2018	2
5	70	87.0	62	77	2021	2
6	67	76.0	64	88	2021	3
7	64	79.0	71	76	2018	2
8	76	82.0	80	77	2019	2
9	70	89.0	80	83	2018	2
10	80	76.0	71	96	2020	3
11	75	81.0	71	95	2018	3
12	NaN	94.0	61	99	2021	3
13	76	79.0	65	91	2018	3
14	66	89.0	61	90	2019	3
15	74	95.0	77	95	2019	3
16	74	82.0	67	75	2019	2
17	70	90.0	68	89	2021	3
18	79	88.0	61	91	2019	3
19	80	89.0	76	85	2021	3
20	79	85.0	67	95	2020	3
21	62	88.0	67	98	2021	3
22	61	82.0	77	96	2018	3
23	63	92.0	79	88	2021	3
24	79	81.0	68	82	2019	2
25	68	94.0	63	76	2020	2
26	76	77.0	77	100	2019	3
27	79	82.0	67	89	2020	3
28	68	92.0	72	83	2021	2

In [75]:
```
col = ['Reading_Score']
refined_df.boxplot(col)
```

Out[75]: <Axes: >

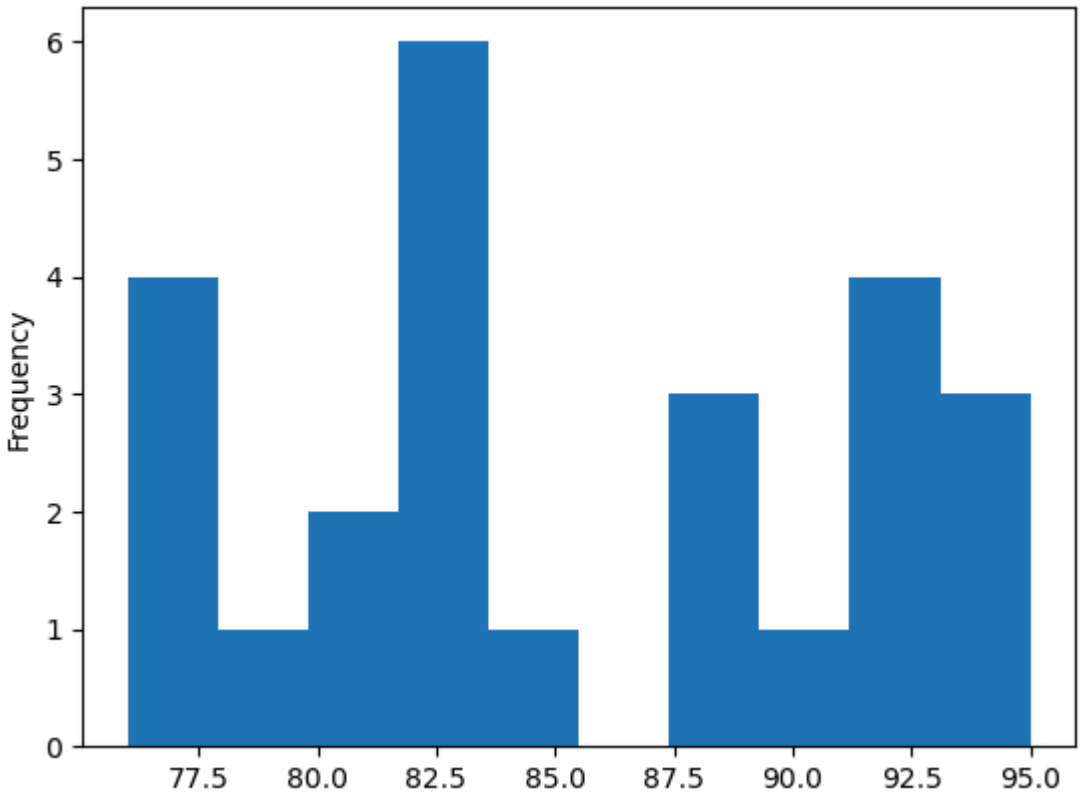


In [76]:
```
1 df
```

Out[76]:

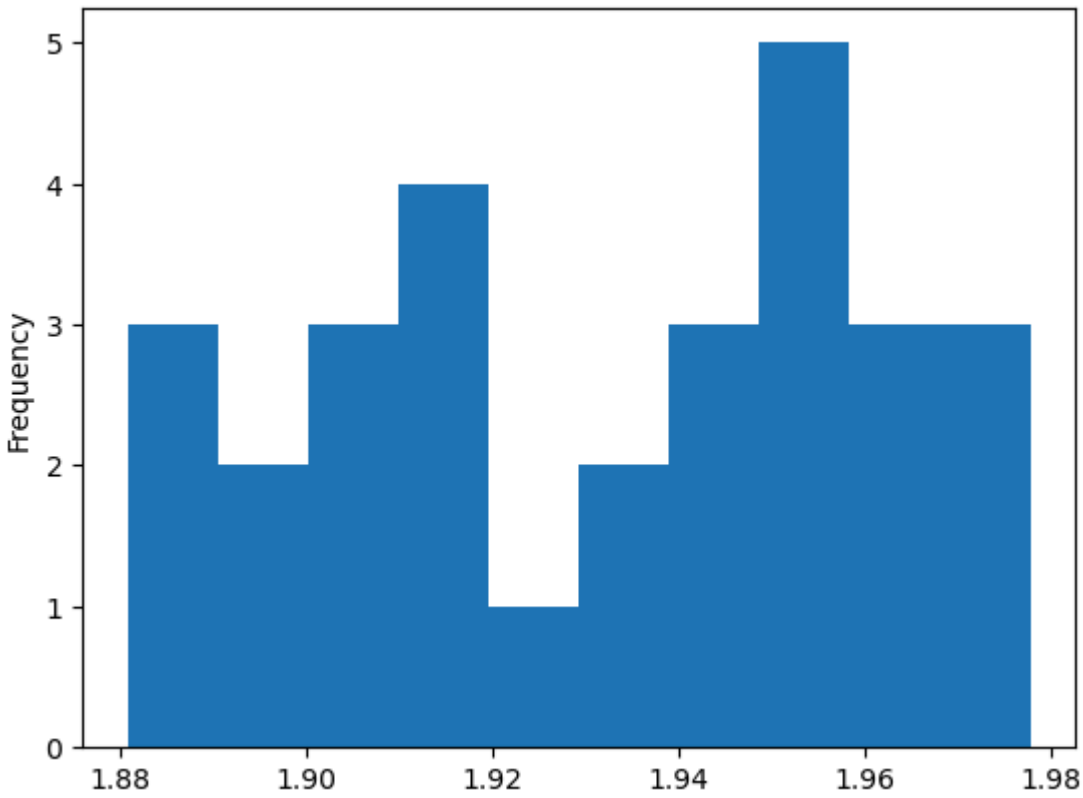
	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement offer count
0	78	84.0	62	96	2021	3
1	77	80.0	72	97	2019	3
2	64	91.0	67	94	2021	3
3	94	86.0	67	77	2019	2
4	62	90.0	69	80	2018	2
5	70	87.0	62	77	2021	2
6	67	76.0	64	88	2021	3
7	64	79.0	71	76	2018	2
8	76	82.0	80	77	2019	2
9	70	89.0	80	83	2018	2
10	80	76.0	71	96	2020	3
11	75	81.0	71	95	2018	3
12	NaN	94.0	61	99	2021	3
13	76	79.0	65	91	2018	3
14	66	89.0	61	90	2019	3
15	74	95.0	77	95	2019	3
16	74	82.0	67	75	2019	2
17	70	90.0	68	89	2021	3
18	79	88.0	61	91	2019	3
19	80	89.0	76	85	2021	3
20	79	85.0	67	95	2020	3
21	62	88.0	67	98	2021	3
22	61	82.0	77	96	2018	3
23	63	92.0	79	88	2021	3
24	79	81.0	68	82	2019	2
25	68	94.0	63	76	2020	2
26	76	77.0	77	100	2019	3
27	79	82.0	67	89	2020	3
28	68	92.0	72	83	2021	2

In [77]:
```
import matplotlib.pyplot as plt
new_df['Reading_Score'].plot(kind ='hist')
df['log_math'] = np.log10(df['Reading_Score'])
```



In [78]:
```
df['log_math'].plot(kind = 'hist')
```

Out[78]: <Axes: ylabel='Frequency'>



Name:Devesh Kashikar
Roll.no:13217

In [ ]:
```
1
```