In [1]:
```
import pandas as pd
import numpy as  np
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_csv("std_per1.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	Maths_score	Reading_score	Writing_score	Placement_score	Club_join_date	Placement_offe
0	80.0	78.0	60.0	78.0	2023.0	2.
1	92.0	87.0	62.0	84.0	2020.0	2.
2	NaN	91.0	71.0	95.0	2021.0	3.
3	NaN	86.0	65.0	76.0	2022.0	2.
4	NaN	86.0	63.0	87.0	2020.0	3.

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Maths_score      85 non-null     float64
 1   Reading_score    84 non-null     float64
 2   Writing_score    89 non-null     float64
 3   Placement_score  81 non-null     float64
 4   Club_join_date   92 non-null     float64
 5   Placement_offer  91 non-null     float64
dtypes: float64(6)
memory usage: 4.7 KB
```

In [6]:
```
df.isnull().sum().sum()
```

Out[6]:
```
np.int64(72)
```

In [7]:
```
df.describe()
```

Out[7]:

	Maths_score	Reading_score	Writing_score	Placement_score	Club_join_date	Placement_
count	85.000000	84.000000	89.000000	81.000000	92.000000	91.0
mean	71.929412	84.607143	68.539326	86.839506	2022.315217	2.5
std	7.986280	8.933513	7.026017	9.526879	3.016181	0.5
min	60.000000	22.000000	40.000000	34.000000	2020.000000	2.0
25%	65.000000	86.000000	63.000000	81.000000	2021.000000	2.0
50%	71.000000	86.000000	69.000000	87.000000	2022.000000	3.0
75%	78.000000	86.000000	74.000000	95.000000	2023.000000	3.0
max	98.000000	91.000000	80.000000	103.000000	2043.000000	5.0

In [8]:
```
df.notnull().sum()
```

Out[8]:
```
Maths_score        85
Reading_score      84
Writing_score      89
Placement_score    81
Club_join_date     92
Placement_offer    91
dtype: int64
```

In [9]:
```
df.columns
```

Out[9]:
```
Index(['Maths_score', 'Reading_score', 'Writing_score', 'Placement_score',
       'Club_join_date', 'Placement_offer'],
      dtype='object')
```

In [10]:
```
df.drop([40,41,42,45,46],inplace=True)
```

In [11]:
```
df
```

Out[11]:

	Maths_score	Reading_score	Writing_score	Placement_score	Club_join_date	Placement_off
0	80.0	78.0	60.0	78.0	2023.0	2
1	92.0	87.0	62.0	84.0	2020.0	2
2	NaN	91.0	71.0	95.0	2021.0	3
3	NaN	86.0	65.0	76.0	2022.0	2
4	NaN	86.0	63.0	87.0	2020.0	3
...	...	...	...	...	...	
94	NaN	86.0	74.0	80.0	2024.0	2
95	98.0	86.0	60.0	99.0	2020.0	3
96	NaN	86.0	78.0	82.0	2024.0	2
97	NaN	86.0	75.0	83.0	2020.0	2
98	NaN	86.0	78.0	76.0	2020.0	2

94 rows × 6 columns

In [12]:
```
df.isna().sum()
```

Out[12]:
```
Maths_score        12
Reading_score      13
Writing_score       8
Placement_score    17
Club_join_date      5
Placement_offer     6
dtype: int64
```

In [14]:
```
df["Maths_score"].fillna(value=df["Maths_score"].mean(),inplace=True)
```

```
<ipython-input-14-e89a70f66703>:1: FutureWarning: A value is trying to be se
t on a copy of a DataFrame or Series through chained assignment using an inp
lace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behave
s as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) ins
tead, to perform the operation inplace on the original object.


  df["Maths_score"].fillna(value=df["Maths_score"].mean(),inplace=True)
```

In [15]:
```
df
```

Out[15]:

	Maths_score	Reading_score	Writing_score	Placement_score	Club_join_date	Placement_off
0	80.000000	78.0	60.0	78.0	2023.0	2
1	92.000000	87.0	62.0	84.0	2020.0	2
2	71.890244	91.0	71.0	95.0	2021.0	3
3	71.890244	86.0	65.0	76.0	2022.0	2
4	71.890244	86.0	63.0	87.0	2020.0	3
...	...	...	...	...	...	
94	71.890244	86.0	74.0	80.0	2024.0	2
95	98.000000	86.0	60.0	99.0	2020.0	3
96	71.890244	86.0	78.0	82.0	2024.0	2
97	71.890244	86.0	75.0	83.0	2020.0	2
98	71.890244	86.0	78.0	76.0	2020.0	2

94 rows × 6 columns

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
Maths_score         0
Reading_score      13
Writing_score       8
Placement_score    17
Club_join_date      5
Placement_offer     6
dtype: int64
```

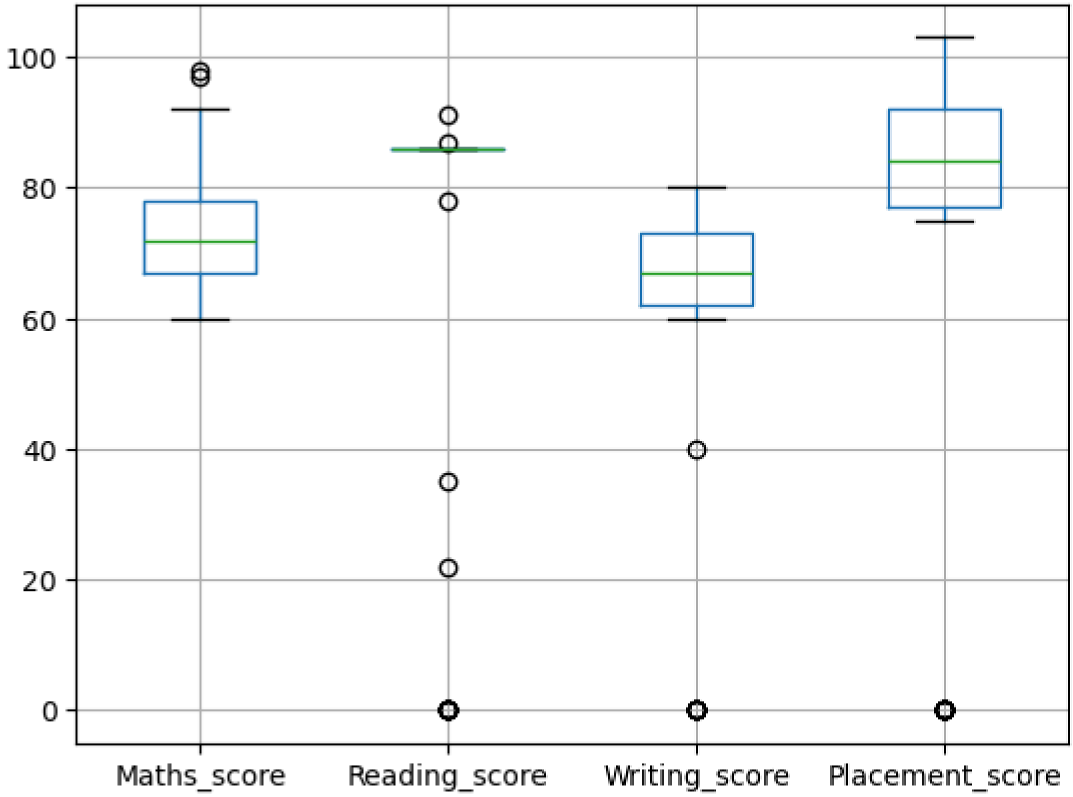
In [17]:
```
df.fillna(0,inplace=True)
```

```
In [18]: df.isnull().sum()
```

```
Out[18]: Maths_score        0
         Reading_score      0
         Writing_score      0
         Placement_score    0
         Club_join_date     0
         Placement_offer    0
         dtype: int64
```

```
In [19]: col=['Maths_score', 'Reading_score', 'Writing_score', 'Placement_score']
         df.boxplot(col)
```

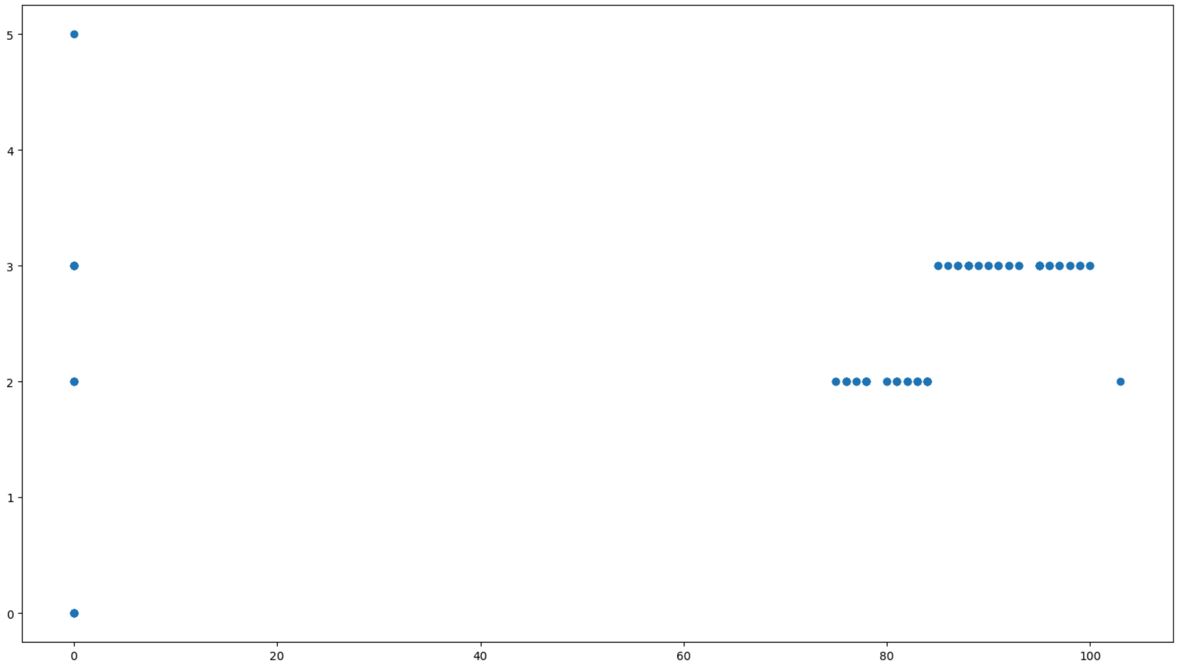
```
Out[19]: <Axes: >
```



```
In [21]: print(np.where(df["Maths_score"]>90))
```

```
(array([ 1, 42, 90]),)
```

In [24]:
```
fig, ax = plt.subplots(figsize = (18,10))
ax.scatter(df['Placement_score'], df['Placement_offer'])
plt.show()
```



In [26]:
```
print(np.where((df['Placement_score']<50) & (df['Placement_offer']>1)))
```

```
(array([12, 13, 14, 16, 17, 36, 37, 57, 58, 59, 71]),)
```

In [27]:
```
from scipy import stats
```

In [28]:
```
z=np.abs(stats.zscore(df['Maths_score']))
```

In [29]:
```
print(z)
```

```
0     1.078803e+00
1     2.675107e+00
2     3.780808e-15
3     3.780808e-15
4     3.780808e-15
          ...
94    3.780808e-15
95    3.473259e+00
96    3.780808e-15
97    3.780808e-15
98    3.780808e-15
Name: Maths_score, Length: 94, dtype: float64
```

In [30]:
```
threshold = 0.18
```

In [31]:
```
outliers=np.where(z<threshold)
outliers
```

Out[31]:
```
(array([ 2,  3,  4, 31, 33, 36, 39, 40, 43, 44, 59, 68, 71, 87, 88, 89, 91,
        92, 93]),)
```

In [32]:

```
sorted_rscore= sorted(df['Reading_score'])
```

In [33]:

```
sorted_rscore
```

```
Out[33]: [0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          22.0,
          35.0,
          78.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
          86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 86.0,
 87.0,
 91.0]
```

In [34]:
```
q1=np.percentile(sorted_rscore,25)
q3=np.percentile(sorted_rscore,75)
```

In [35]:
```
print(q1,q3)
```

```
86.0 86.0
```

In [36]:
```
IQR = q3-q1
```

In [37]:
```
lwr_bound=q1-(1.5*IQR)
upr_bound=q3+(1.5*IQR)
print(lwr_bound, upr_bound)
```

```
86.0 86.0
```

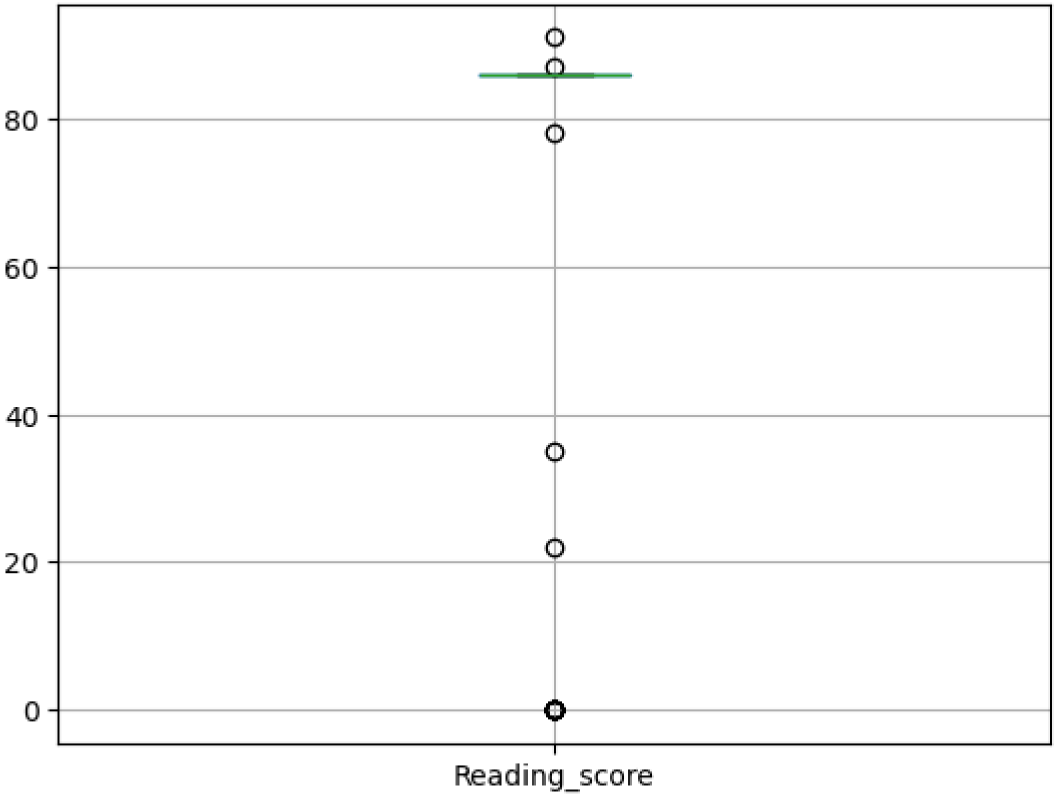
In [38]:
```
r_outliers = []
for i in sorted_rscore:
    if (i<lwr_bound or i>upr_bound):
        r_outliers.append(i)
print(r_outliers)
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 22.0, 35.
0, 78.0, 87.0, 91.0]
```

In [39]:
```
col1=['Reading_score']
df.boxplot(col1)
```

Out[39]:
```
<Axes: >
```

In [40]:
```
plt.show()
```

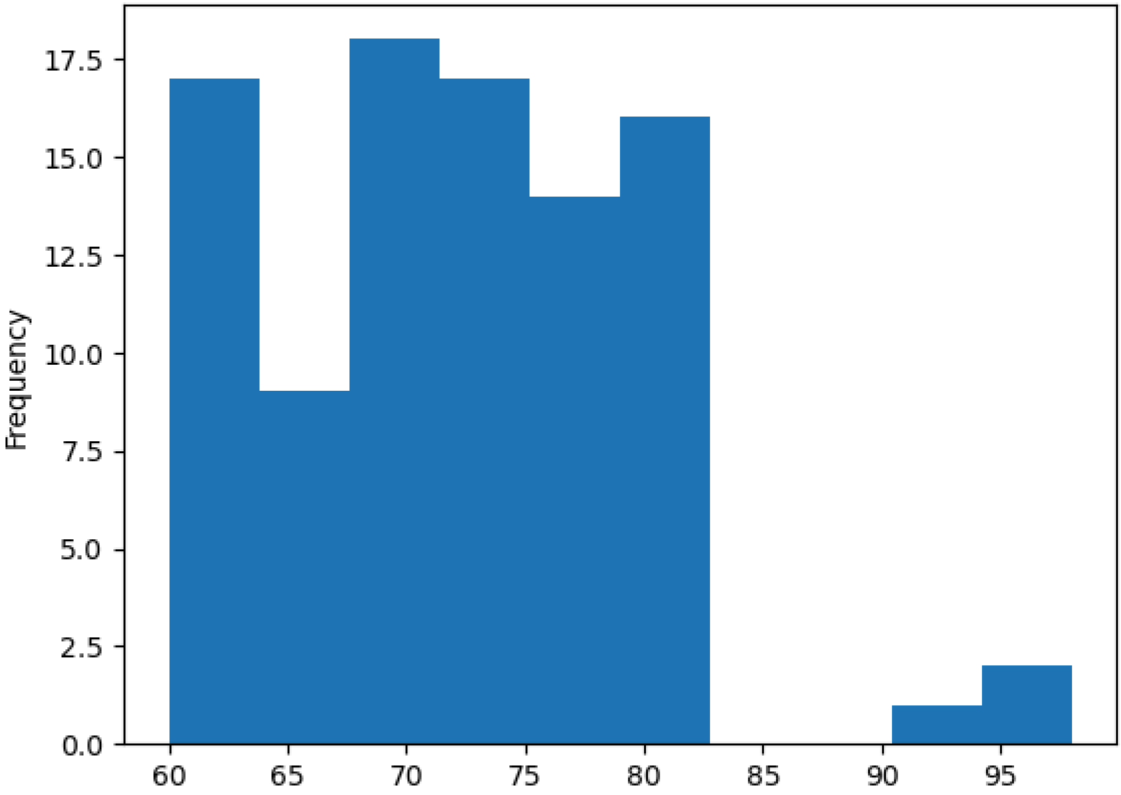


In [41]:
```
import matplotlib.pyplot as plt
```

In [43]:
```
df['Maths_score'].plot(kind='hist')
```

Out[43]:
```
<Axes: ylabel='Frequency'>
```

In [44]:
```
plt.show()
```

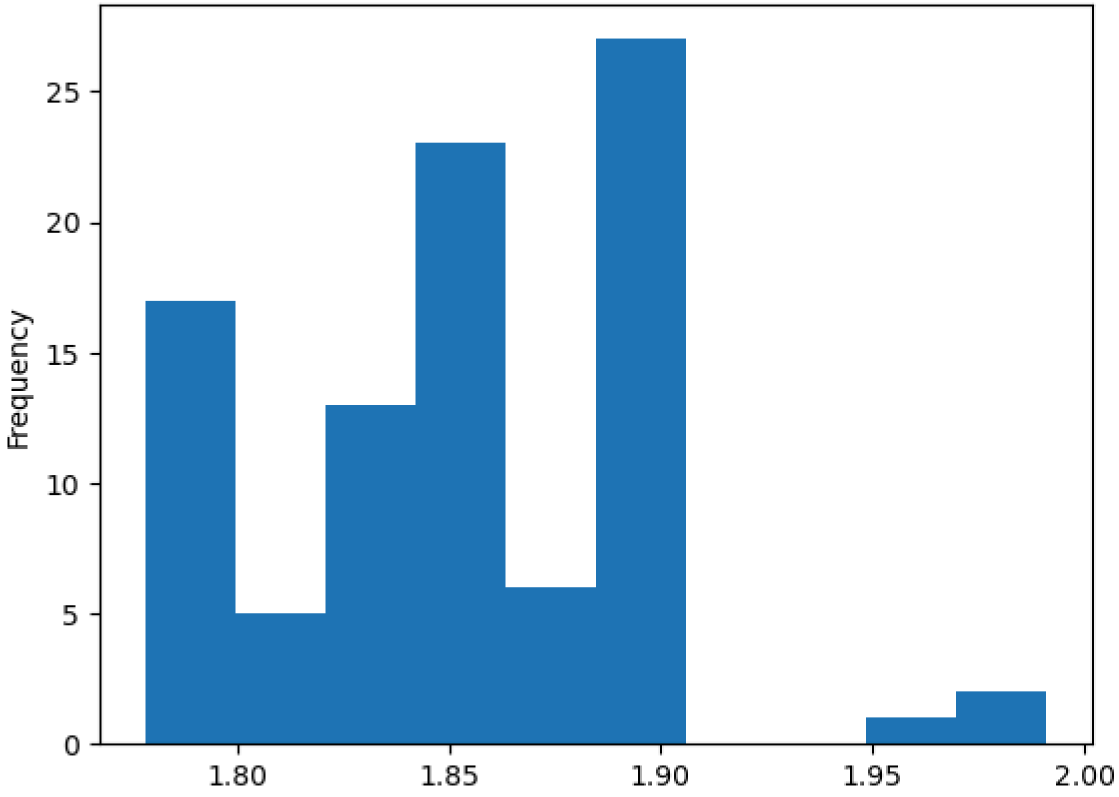


In [46]:
```
df['log_maths'] = np.log10(df['Maths_score'])
```

In [48]:
```
df['log_maths'].plot(kind ='hist')
```

Out[48]:
```
<Axes: ylabel='Frequency'>
```

```
In [49]: plt.show()
```



```
In [ ]:
```