In [17]:

```
import pandas as pd
import numpy as np
df = pd.read_csv("/home/ubuntu/dataset.csv")
df
```

Out[17]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	NaN
1	80.0	82.0	65.0	NaN	04-11-2021	male
2	67.0	93.0	70.0	96.0	18-12-2019	female
3	79.0	20.0	63.0	81.0	27-07-2019	female
4	62.0	81.0	75.0	86.0	03-07-2021	NaN
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	female
96	60.0	83.0	66.0	90.0	07-07-2019	male
97	74.0	95.0	78.0	81.0	13-04-2018	male
98	69.0	94.0	74.0	92.0	08-10-2018	male
99	74.0	87.0	63.0	79.0	09-07-2021	NaN

100 rows × 8 columns

In [19]:

```
df.isnull()
```

Out[19]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	False	False	False	False	False	True
1	False	False	False	True	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	True
...	...	...	...	...	...	...
95	False	False	False	False	False	False
96	False	False	False	False	False	False
97	False	False	False	False	False	False
98	False	False	False	False	False	False
99	False	False	False	False	False	True

100 rows × 8 columns

In [20]:
```
series = pd.isnull(df["Math_Score"])
df[series]
```

Out[20]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
9	NaN	87.0	75.0	NaN	29-04-2018	male
27	NaN	80.0	78.0	96.0	NaN	female
29	NaN	95.0	76.0	75.0	03-08-2018	male
40	NaN	NaN	78.0	92.0	08-01-2020	male
66	NaN	85.0	73.0	90.0	28-06-2019	male
89	NaN	92.0	80.0	85.0	25-12-2018	female

In [21]:
```
df.notnull()
```

Out[21]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	True	True	True	True	True	False
1	True	True	True	False	True	True
2	True	True	True	True	True	True
3	True	True	True	True	True	True
4	True	True	True	True	True	False
...	...	...	...	...	...	...
95	True	True	True	True	True	True
96	True	True	True	True	True	True
97	True	True	True	True	True	True
98	True	True	True	True	True	True
99	True	True	True	True	True	False

100 rows × 8 columns

In [22]:

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

Out[22]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	NaN
1	80.0	82.0	65.0	NaN	04-11-2021	male
2	67.0	93.0	70.0	96.0	18-12-2019	female
3	79.0	20.0	63.0	81.0	27-07-2019	female
4	62.0	81.0	75.0	86.0	03-07-2021	NaN
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	female
96	60.0	83.0	66.0	90.0	07-07-2019	male
97	74.0	95.0	78.0	81.0	13-04-2018	male
98	69.0	94.0	74.0	92.0	08-10-2018	male
99	74.0	87.0	63.0	79.0	09-07-2021	NaN

94 rows × 8 columns

In [23]:

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

df
```

Out[23]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	2
1	80.0	82.0	65.0	NaN	04-11-2021	1
2	67.0	93.0	70.0	96.0	18-12-2019	0
3	79.0	20.0	63.0	81.0	27-07-2019	0
4	62.0	81.0	75.0	86.0	03-07-2021	2
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	0
96	60.0	83.0	66.0	90.0	07-07-2019	1
97	74.0	95.0	78.0	81.0	13-04-2018	1
98	69.0	94.0	74.0	92.0	08-10-2018	1
99	74.0	87.0	63.0	79.0	09-07-2021	2

100 rows × 8 columns

In [24]:

```
m_v=df['Math_Score'].mean()
df.fillna({"Math_Score":m_v}, inplace=True)
m_v1=df['Reading_Score'].mean()
df.fillna({"Reading_core":m_v1}, inplace=True)
m_v2=df['Writing_Score'].mean()
df.fillna({'Writing_Score':m_v2}, inplace=True)
df
```

Out[24]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	2
1	80.0	82.0	65.0	NaN	04-11-2021	1
2	67.0	93.0	70.0	96.0	18-12-2019	0
3	79.0	20.0	63.0	81.0	27-07-2019	0
4	62.0	81.0	75.0	86.0	03-07-2021	2
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	0
96	60.0	83.0	66.0	90.0	07-07-2019	1
97	74.0	95.0	78.0	81.0	13-04-2018	1
98	69.0	94.0	74.0	92.0	08-10-2018	1
99	74.0	87.0	63.0	79.0	09-07-2021	2

100 rows × 8 columns

In [9]:

```
df.head(10)
```

Out[9]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	67.00000	84.0	77.0	79.0	11-08-2021	2
1	80.00000	82.0	65.0	NaN	04-11-2021	1
2	67.00000	93.0	70.0	96.0	18-12-2019	0
3	79.00000	93.0	63.0	81.0	27-07-2019	0
4	62.00000	81.0	75.0	86.0	03-07-2021	2
5	71.00000	92.0	68.0	76.0	02-09-2019	1
6	78.00000	75.0	60.0	80.0	26-03-2018	1
7	71.00000	87.0	71.0	86.0	14-07-2020	2
8	70.00000	85.0	68.0	99.0	28-02-2018	0
9	70.87234	87.0	75.0	NaN	29-04-2018	1

In [25]:

```
df.dropna()
```

Out[25]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	2
2	67.0	93.0	70.0	96.0	18-12-2019	0
3	79.0	20.0	63.0	81.0	27-07-2019	0
4	62.0	81.0	75.0	86.0	03-07-2021	2
5	71.0	92.0	68.0	76.0	02-09-2019	1
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	0
96	60.0	83.0	66.0	90.0	07-07-2019	1
97	74.0	95.0	78.0	81.0	13-04-2018	1
98	69.0	94.0	74.0	92.0	08-10-2018	1
99	74.0	87.0	63.0	79.0	09-07-2021	2

61 rows × 8 columns

In [26]:

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

Out[26]:

	Math_Score	Writing_Score	Gender	Placement_Count
0	89.0	77.0	2	2
1	80.0	65.0	1	1
2	67.0	70.0	0	1
3	79.0	63.0	0	2
4	62.0	75.0	2	2
...	...	...	...	...
95	62.0	80.0	0	3
96	60.0	66.0	1	2
97	74.0	78.0	1	1
98	69.0	74.0	1	3
99	74.0	63.0	2	3

100 rows × 4 columns

In [27]:
```
new_data  = df.dropna(axis = 0)
new_data
```

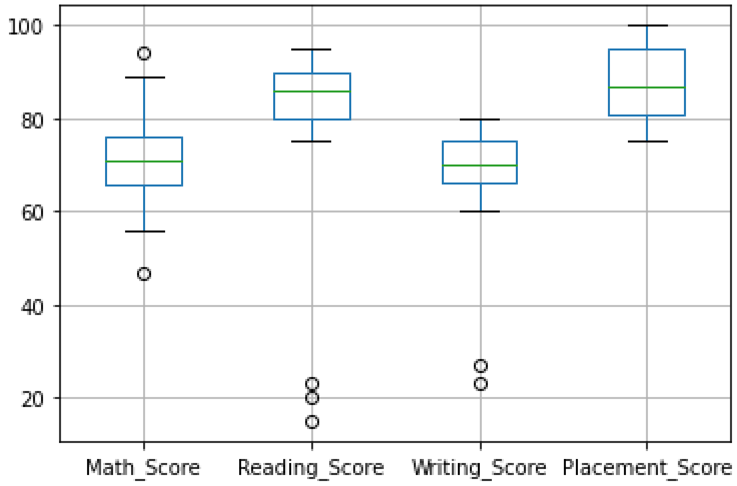
Out[27]:

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Gender
0	89.0	84.0	77.0	79.0	11-08-2021	2
2	67.0	93.0	70.0	96.0	18-12-2019	0
3	79.0	20.0	63.0	81.0	27-07-2019	0
4	62.0	81.0	75.0	86.0	03-07-2021	2
5	71.0	92.0	68.0	76.0	02-09-2019	1
...	...	...	...	...	...	...
95	62.0	79.0	80.0	78.0	09-10-2019	0
96	60.0	83.0	66.0	90.0	07-07-2019	1
97	74.0	95.0	78.0	81.0	13-04-2018	1
98	69.0	94.0	74.0	92.0	08-10-2018	1
99	74.0	87.0	63.0	79.0	09-07-2021	2

61 rows × 8 columns

In [40]:
```
col = ["Math_Score","Reading_Score","Writing_Score","Placement_Scor
df.boxplot(col)
```

Out[40]: <AxesSubplot:>

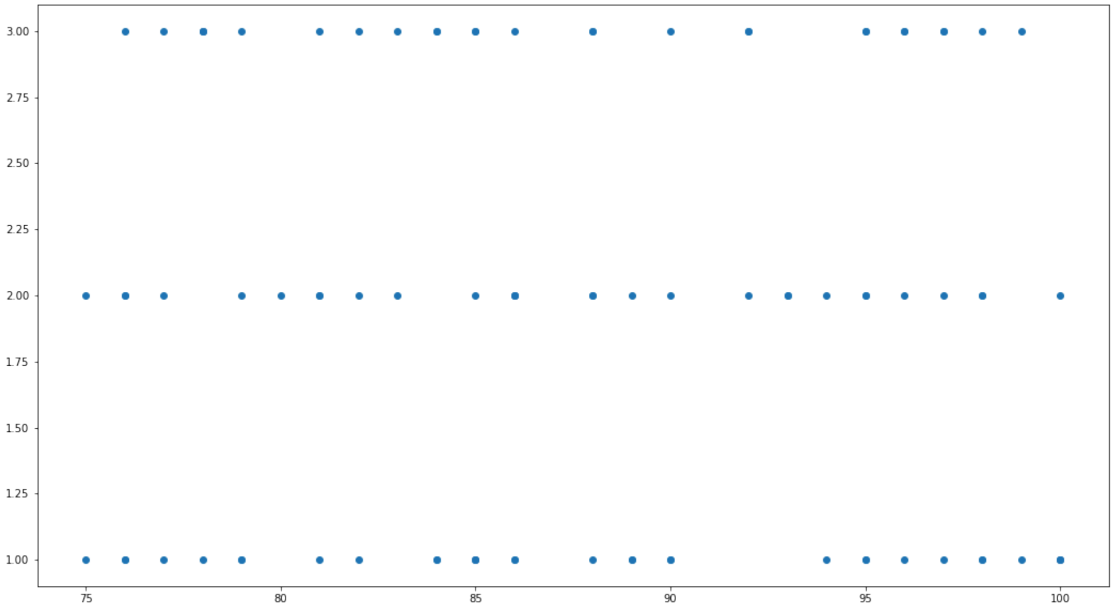


In [30]:
```
print(np.where(df['Math_Score']>90))
print(np.where(df['Reading_Score']<25))
print(np.where(df['Writing_Score']<30))
```

```
(array([14]),)
(array([ 3, 25, 87]),)
(array([ 7, 51]),)
```

```
In [31]: import matplotlib.pyplot as plt
         import pandas as pd
         fig, ax = plt.subplots(figsize=(18, 10))
         ax.scatter(df['Placement_Score'], df['Placement_Count'])

         plt.show()
```



```
In [33]: print(np.where((df['Placement_Score']<50)&(df['Placement_Score']>85
         print(np.where((df['Placement_Count']<2)))
```

```
(array([], dtype=int64),)
(array([ 1,  2, 11, 13, 15, 16, 18, 19, 21, 35, 37, 38, 39, 41, 4
2, 43, 46,
       47, 48, 50, 52, 60, 64, 65, 66, 68, 73, 74, 78, 81, 83, 85,
92, 94,
       97]),)
```

```
In [34]: import numpy as np
         from scipy import stats
         z = np.abs(stats.zscore(df['Math_Score']))
         print(z)
```

```
0     2.462841
1     1.238656
2     0.529612
3     1.102635
4     1.209715
        ...
95    1.209715
96    1.481756
97    0.422532
98    0.257571
99    0.422532
Name: Math_Score, Length: 100, dtype: float64
```

In [35]:
```
threshold = 0.18
sample_outliers = np.where(z <threshold)
sample_outliers
```

Out[35]:
```
(array([ 5,  7,  8,  9, 13, 26, 27, 29, 40, 47, 66, 70, 75, 80, 8
3, 86, 89,
        91, 93]),)
```

In [42]:
```
sorted_score= sorted(new_data['Reading_Score'])
print(sorted_score)
```

```
[20.0, 23.0, 75.0, 75.0, 76.0, 78.0, 78.0, 79.0, 79.0, 80.0, 80.0,
80.0, 80.0, 81.0, 81.0, 81.0, 82.0, 82.0, 82.0, 83.0, 83.0, 83.0,
83.0, 84.0, 85.0, 85.0, 85.0, 86.0, 86.0, 86.0, 86.0, 86.0, 87.0,
87.0, 87.0, 88.0, 88.0, 88.0, 89.0, 89.0, 89.0, 89.0, 89.0, 89.0,
89.0, 90.0, 91.0, 91.0, 91.0, 92.0, 92.0, 92.0, 93.0, 93.0, 94.0,
94.0, 95.0, 95.0, 95.0, 95.0, 95.0]
```

In [43]:
```
q1 = np.percentile(sorted_score, 25)
q3 = np.percentile(sorted_score, 75)
print(q1,q3)
```

```
81.0 90.0
```

In [44]:
```
iqr = q3-q1
lbound = q1-(1.5*iqr)
ubound = q3+(1.5*iqr)
print(lbound, ubound)
```

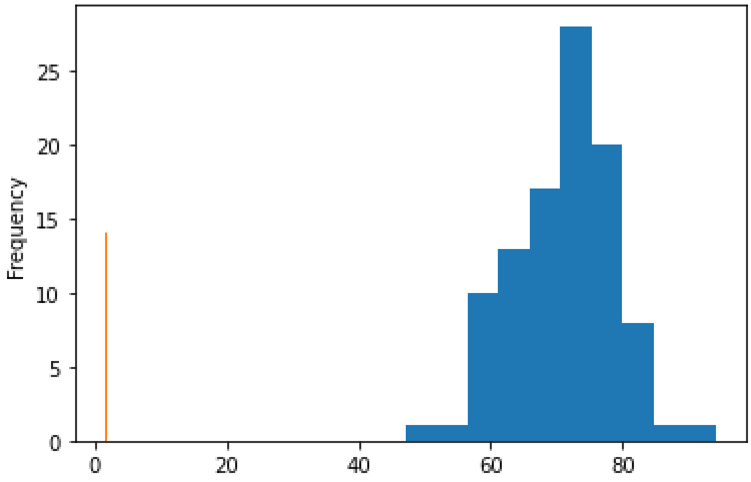
```
67.5 103.5
```

In [46]:
```
r_outliers = []
for i in sorted_score:
    if (i<lbound or i>ubound):
        r_outliers.append(i)
print(r_outliers)
```

```
[20.0, 23.0]
```

```
In [47]: import matplotlib.pyplot as plt
         df['Math_Score'].plot(kind = 'hist')
         df['log_math'] = np.log10(df['Math_Score'])
         df['log_math'].plot(kind = 'hist')
```

```
Out[47]: <AxesSubplot:ylabel='Frequency'>
```


Frequency