

```
In [ ]: # full code in one cell

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler

df=pd.read_csv('Academic_performace .csv')
df.isnull().sum()

df['Age'].fillna(df['Age'].median(), inplace=True)
df['Grade Level'].fillna(df['Grade Level'].mean(), inplace=True)
df['Attendance'].fillna(df['Attendance'].mean(), inplace=True)
df['Final Exam Score'].fillna(df['Final Exam Score'].mean(), inplace=True)

cols = ['Age', 'Grade Level', 'Attendance', 'Final Exam Score']
df.boxplot(column=cols)
plt.show()

for col in cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    df = df[(df[col] >= lower) & (df[col] <= upper)]

df.boxplot(column=cols)
plt.show()

scaler = MinMaxScaler()
df['Attendance_Normalized'] = scaler.fit_transform(df[['Attendance']])

print(df.head())
```

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

```
In [3]: df=pd.read_csv('Academic_performace .csv')
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: Age                1
Grade Level              2
Attendance               2
Final Exam Score         1
dtype: int64
```

```
In [5]: df['Age'].fillna(df['Age'].median(), inplace=True)
df['Grade Level'].fillna(df['Grade Level'].mean(), inplace=True)
df['Attendance'].fillna(df['Attendance'].mean(), inplace=True)
df['Final Exam Score'].fillna(df['Final Exam Score'].mean(), inplace=True)
```

C:\Users\Chatura Karankal\AppData\Local\Temp\ipykernel_4352\2174622022.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace 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 behaves as a copy.

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

```
df['Age'].fillna(df['Age'].median(), inplace=True)
```

C:\Users\Chatura Karankal\AppData\Local\Temp\ipykernel_4352\2174622022.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace 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 behaves as a copy.

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

```
df['Grade Level'].fillna(df['Grade Level'].mean(), inplace=True)
```

C:\Users\Chatura Karankal\AppData\Local\Temp\ipykernel_4352\2174622022.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace 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 behaves as a copy.

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

```
df['Attendance'].fillna(df['Attendance'].mean(), inplace=True)
```

C:\Users\Chatura Karankal\AppData\Local\Temp\ipykernel_4352\2174622022.py:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace 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 behaves as a copy.

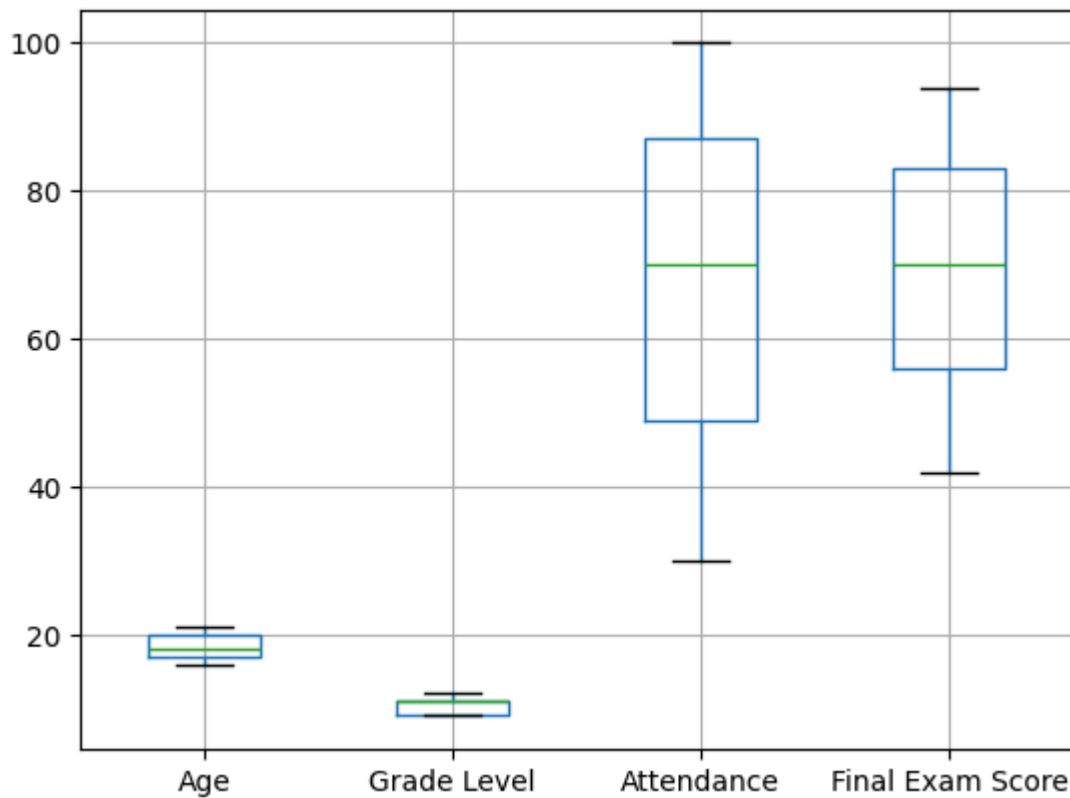
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['Final Exam Score'].fillna(df['Final Exam Score'].mean(), inplace=True)
```

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

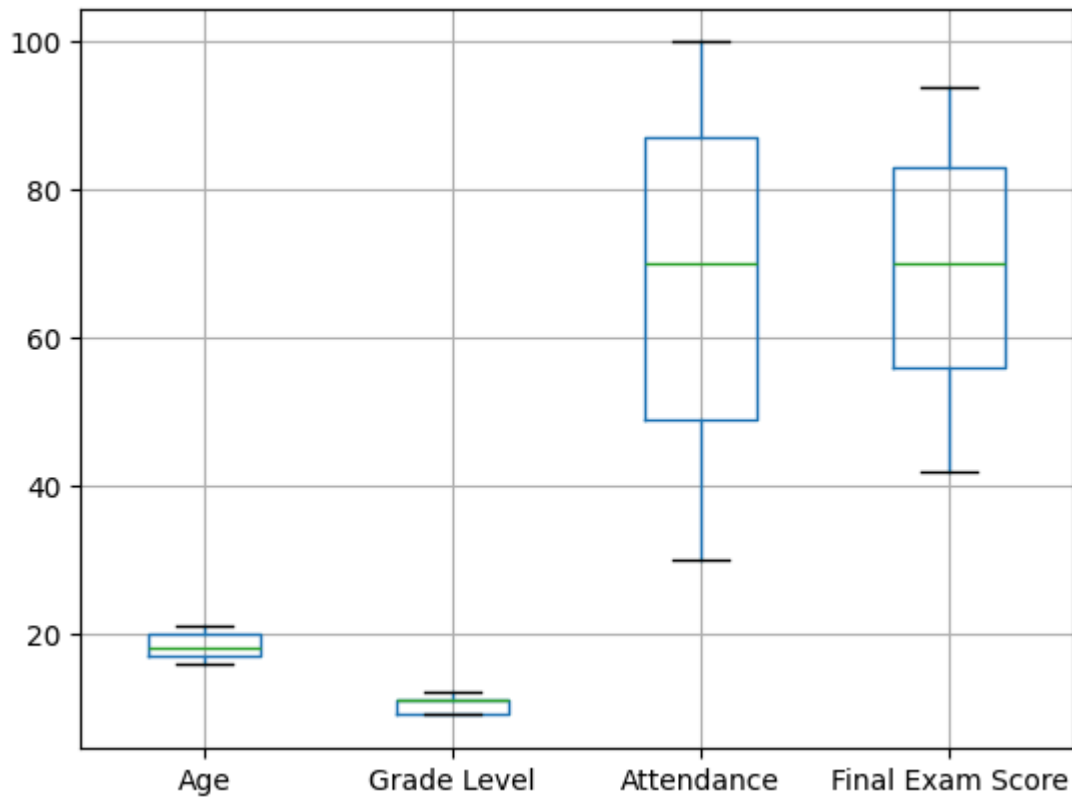
```
Out[6]: Age          0
Grade Level        0
Attendance         0
Final Exam Score   0
dtype: int64
```

```
In [7]: cols = ['Age', 'Grade Level', 'Attendance', 'Final Exam Score']
df.boxplot(cols)
plt.show()
```



```
In [8]: for col in cols:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower = Q1 - 1.5 * IQR
        upper = Q3 + 1.5 * IQR
        df = df[(df[col] >= lower) & (df[col] <= upper)]
```

```
In [9]: df.boxplot(cols)
plt.show()
```



```
In [13]: scaler = MinMaxScaler()
df[cols] = scaler.fit_transform(df[cols])
```

```
In [14]: df.head()
```

```
Out[14]:
```

	Age	Grade Level	Attendance	Final Exam Score
0	0.2	0.333333	0.328571	0.442308
1	0.6	0.000000	0.485714	0.230769
2	0.8	1.000000	0.057143	0.711538
3	1.0	0.000000	0.585714	0.230769
4	1.0	0.000000	0.871429	0.461538

```
In [ ]:
```