In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
Likius = pd.read_csv(r"C:\Users\Thomas Hamutoko\Downloads\data analysis\StudentPerformanceFactors_project.csv")
```

In [4]:
```
Likius.head(10)
```

Out[4]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities	Pa
0	23	84	Low	High	No	7	73	Low	Yes	0	Low	Medium	Public	Positive	3	No	
1	19	64	Low	Medium	No	8	59	Low	Yes	2	Medium	Medium	Public	Negative	4	No	
2	24	98	Medium	Medium	Yes	7	91	Medium	Yes	2	Medium	Medium	Public	Neutral	4	No	
3	29	89	Low	Medium	Yes	8	98	Medium	Yes	1	Medium	Medium	Public	Negative	4	No	
4	19	92	Medium	Medium	Yes	6	65	Medium	Yes	3	Medium	High	Public	Neutral	4	No	
5	19	88	Medium	Medium	Yes	8	89	Medium	Yes	3	Medium	Medium	Public	Positive	3	No	
6	29	84	Medium	Low	Yes	7	68	Low	Yes	1	Low	Medium	Private	Neutral	2	No	
7	25	78	Low	High	Yes	6	50	Medium	Yes	1	High	High	Public	Negative	2	No	
8	17	94	Medium	High	No	6	80	High	Yes	0	Medium	Low	Private	Neutral	1	No	
9	23	98	Medium	Medium	Yes	8	71	Medium	Yes	0	High	High	Public	Positive	5	No	

In [5]:
```
#checking for dublicates
duplicate_rows = Likius[Likius.duplicated()]
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")
```

```
Number of duplicate rows: 0
```

In [6]:
```
Likius.drop_duplicates()
```

Out[6]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities
0	23	84	Low	High	No	7	73	Low	Yes	0	Low	Medium	Public	Positive	3	No
1	19	64	Low	Medium	No	8	59	Low	Yes	2	Medium	Medium	Public	Negative	4	No
2	24	98	Medium	Medium	Yes	7	91	Medium	Yes	2	Medium	Medium	Public	Neutral	4	No
3	29	89	Low	Medium	Yes	8	98	Medium	Yes	1	Medium	Medium	Public	Negative	4	No
4	19	92	Medium	Medium	Yes	6	65	Medium	Yes	3	Medium	High	Public	Neutral	4	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6602	25	69	High	Medium	No	7	76	Medium	Yes	1	High	Medium	Public	Positive	2	No
6603	23	76	High	Medium	No	8	81	Medium	Yes	3	Low	High	Public	Positive	2	No
6604	20	90	Medium	Low	Yes	6	65	Low	Yes	3	Low	Medium	Public	Negative	2	No
6605	10	86	High	High	Yes	6	91	High	Yes	2	Low	Medium	Private	Positive	3	No
6606	15	67	Medium	Low	Yes	9	94	Medium	Yes	0	Medium	Medium	Public	Positive	4	No

6607 rows × 20 columns

In [ ]:
```

```

In [ ]:
```

```

In [7]:
```
Likius=Likius.drop_duplicates()
Likius
```

Out[7]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities
0	23	84	Low	High	No	7	73	Low	Yes	0	Low	Medium	Public	Positive	3	No
1	19	64	Low	Medium	No	8	59	Low	Yes	2	Medium	Medium	Public	Negative	4	No
2	24	98	Medium	Medium	Yes	7	91	Medium	Yes	2	Medium	Medium	Public	Neutral	4	No
3	29	89	Low	Medium	Yes	8	98	Medium	Yes	1	Medium	Medium	Public	Negative	4	No
4	19	92	Medium	Medium	Yes	6	65	Medium	Yes	3	Medium	High	Public	Neutral	4	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6602	25	69	High	Medium	No	7	76	Medium	Yes	1	High	Medium	Public	Positive	2	No
6603	23	76	High	Medium	No	8	81	Medium	Yes	3	Low	High	Public	Positive	2	No
6604	20	90	Medium	Low	Yes	6	65	Low	Yes	3	Low	Medium	Public	Negative	2	No
6605	10	86	High	High	Yes	6	91	High	Yes	2	Low	Medium	Private	Positive	3	No
6606	15	67	Medium	Low	Yes	9	94	Medium	Yes	0	Medium	Medium	Public	Positive	4	No

6607 rows × 20 columns

In [9]:
```
#deleting outliers at the end and front
Likius["Parental_Involvement"].str.strip("/123...")
```

Out[9]:
```
0          Low
1          Low
2       Medium
3          Low
4       Medium
         ...
6602      High
6603      High
6604    Medium
6605      High
6606    Medium
Name: Parental_Involvement, Length: 6607, dtype: object
```

In [10]:
```
Likius["Parental_Involvement"]=Likius["Parental_Involvement"].str.strip("/123...")
```

In [12]:
```
missing_values = Likius.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

```
Missing values in each column:
 Hours_Studied                  0
Attendance                      0
Parental_Involvement            0
Access_to_Resources             0
Extracurricular_Activities      0
Sleep_Hours                     0
Previous_Scores                 0
Motivation_Level                0
Internet_Access                 0
Tutoring_Sessions               0
Family_Income                   0
Teacher_Quality                78
School_Type                     0
Peer_Influence                  0
Physical_Activity               0
Learning_Disabilities           0
Parental_Education_Level       90
Distance_from_Home             67
Gender                          0
Exam_Score                      0
dtype: int64
```

In [13]:
```
#separating integer colums from string columns
numeric_cols = Likius.select_dtypes(include=['float64', 'int64']).columns
strings_cols = Likius.select_dtypes(include=['object']).columns
```

In [14]:
```
#filling the null values
for col in numeric_cols:
    Likius[col] = Likius[col].fillna(Likius[col].mean())

for col in strings_cols:
    Likius[col] = Likius[col].fillna(Likius[col].mode()[0])
```

In [15]:
```
print(Likius.isnull().sum())
```

```
Hours_Studied                  0
Attendance                     0
Parental_Involvement           0
Access_to_Resources            0
Extracurricular_Activities     0
Sleep_Hours                    0
Previous_Scores                0
Motivation_Level               0
Internet_Access                0
Tutoring_Sessions              0
Family_Income                  0
Teacher_Quality                0
School_Type                    0
Peer_Influence                 0
Physical_Activity              0
Learning_Disabilities          0
Parental_Education_Level       0
Distance_from_Home             0
Gender                         0
Exam_Score                     0
dtype: int64
```

In [16]:
```
def remove_outliers_iqr(Likius, columns):
    for col in columns:
        # Calculate Q1 (25th percentile) and Q3 (75th percentile)
        Q1 = Likius[col].quantile(0.25)
        Q3 = Likius[col].quantile(0.75)
        IQR = Q3 - Q1

        # Define bounds for outliers
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        # Remove rows with outliers
        Likius = Likius[(Likius[col] >= lower_bound) & (Likius[col] <= upper_bound)]

    return Likius
```

In [17]:
```
numeric_columns = ['Hours_Studied', 'Attendance', 'Sleep_Hours', 'Previous_Scores',
                   'Tutoring_Sessions', 'Physical_Activity', 'Exam_Score']
```

In [18]:
```
Likius = remove_outliers_iqr(Likius, numeric_columns)
```

In [19]:
```
# Save the cleaned dataset
cleaned_file_path = 'C:/Users/Thomas Hamutoko/Downloads/data analysis/StudentPerformanceFactors_cleaned.csv'
Likius.to_csv(cleaned_file_path, index=False)
```

In [29]:
```
# Check the cleaned data
Likius_cleaned.head()
```

Out[29]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities	Pa
0	23	84	Low	High	No	7	73	Low	Yes	0	Low	Medium	Public	Positive	3	No	
1	19	64	Low	Medium	No	8	59	Low	Yes	2	Medium	Medium	Public	Negative	4	No	
2	24	98	Medium	Medium	Yes	7	91	Medium	Yes	2	Medium	Medium	Public	Neutral	4	No	
3	29	89	Low	Medium	Yes	8	98	Medium	Yes	1	Medium	Medium	Public	Negative	4	No	
4	19	92	Medium	Medium	Yes	6	65	Medium	Yes	3	Medium	High	Public	Neutral	4	No	

In [20]:
```
Likius.columns
```

Out[20]:
```
Index(['Hours_Studied', 'Attendance', 'Parental_Involvement',
       'Access_to_Resources', 'Extracurricular_Activities', 'Sleep_Hours',
       'Previous_Scores', 'Motivation_Level', 'Internet_Access',
       'Tutoring_Sessions', 'Family_Income', 'Teacher_Quality', 'School_Type',
       'Peer_Influence', 'Physical_Activity', 'Learning_Disabilities',
       'Parental_Education_Level', 'Distance_from_Home', 'Gender',
       'Exam_Score'],
      dtype='object')
```

In [21]:
```
Likius.columns.values
```

Out[21]:
```
array(['Hours_Studied', 'Attendance', 'Parental_Involvement',
       'Access_to_Resources', 'Extracurricular_Activities', 'Sleep_Hours',
       'Previous_Scores', 'Motivation_Level', 'Internet_Access',
       'Tutoring_Sessions', 'Family_Income', 'Teacher_Quality',
       'School_Type', 'Peer_Influence', 'Physical_Activity',
```

```
       'Learning_Disabilities', 'Parental_Education_Level',
       'Distance_from_Home', 'Gender', 'Exam_Score'], dtype=object)
```

In [ ]: