Name: Bishakha
SJSU ID: 019178354

Project EDA
Student Success Factor Analysis

Fa25: ISE-201 Math Foundation for Decision and Data Sciences

Understand problem & Data

With the advent of technology , everyone around seems more interested in gadgets than people . Societies have changed and new norms have been instated .Parents now spend much less time with their kids than in the previous generations for several reasons , which seems valid too in the current world. Kids are sent to playschools where they are left unattended , which creates an emotional void and in turn affects their entire life from mental health to education to social life. My idea behind choosing this project was to really understand and analyse the factors which most affect the academic performance of the students in K-12. The source for the sample dataset used is kaggle. The dataset contains numerical variables such as Hours studied, tutoring sessions, Attendance, sleep hour, physical activities and categorical variables such as extracurricular activities , parental educational level, Motivation Level, peer Influence , Family Income , gender and other factors.

Import & inspect data

The dataset was analyzed using Python. For importing and inspecting the dataset , I used libraries like Pandas , seaborn , matplotlib and sklearn. The dataset contains 6607 rows and 20 columns. Details about the column are shown in the picture below.

```
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Hours_Studied               6607 non-null   int64
 1   Attendance                  6607 non-null   int64
 2   Parental_Involvement        6607 non-null   object
 3   Access_to_Resources         6607 non-null   object
 4   Extracurricular_Activities  6607 non-null   object
 5   Sleep_Hours                 6607 non-null   int64
 6   Previous_Scores             6607 non-null   int64
 7   Motivation_Level            6607 non-null   object
 8   Internet_Access             6607 non-null   object
 9   Tutoring_Sessions           6607 non-null   int64
 10  Family_Income               6607 non-null   object
 11  Teacher_Quality             6529 non-null   object
 12  School_Type                 6607 non-null   object
 13  Peer_Influence              6607 non-null   object
 14  Physical_Activity           6607 non-null   int64
 15  Learning_Disabilities       6607 non-null   object
 16  Parental_Education_Level    6517 non-null   object
 17  Distance_from_Home          6540 non-null   object
 18  Gender                      6607 non-null   object
 19  Exam_Score                  6607 non-null   int64
dtypes: int64(7), object(13)
```

The following table describes Mean , Std , quartiles , min and max for numerical columns.

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores	Tutoring_Sessions	Physical_Activity	Exam_Score
count	6607.000000	6607.000000	6607.00000	6607.000000	6607.000000	6607.000000	6607.000000
mean	19.975329	79.977448	7.02906	75.070531	1.493719	2.967610	67.235659
std	5.990594	11.547475	1.46812	14.399784	1.230570	1.031231	3.890456
min	1.000000	60.000000	4.00000	50.000000	0.000000	0.000000	55.000000
25%	16.000000	70.000000	6.00000	63.000000	1.000000	2.000000	65.000000
50%	20.000000	80.000000	7.00000	75.000000	1.000000	3.000000	67.000000
75%	24.000000	90.000000	8.00000	88.000000	2.000000	4.000000	69.000000
max	44.000000	100.000000	10.00000	100.000000	8.000000	6.000000	101.000000

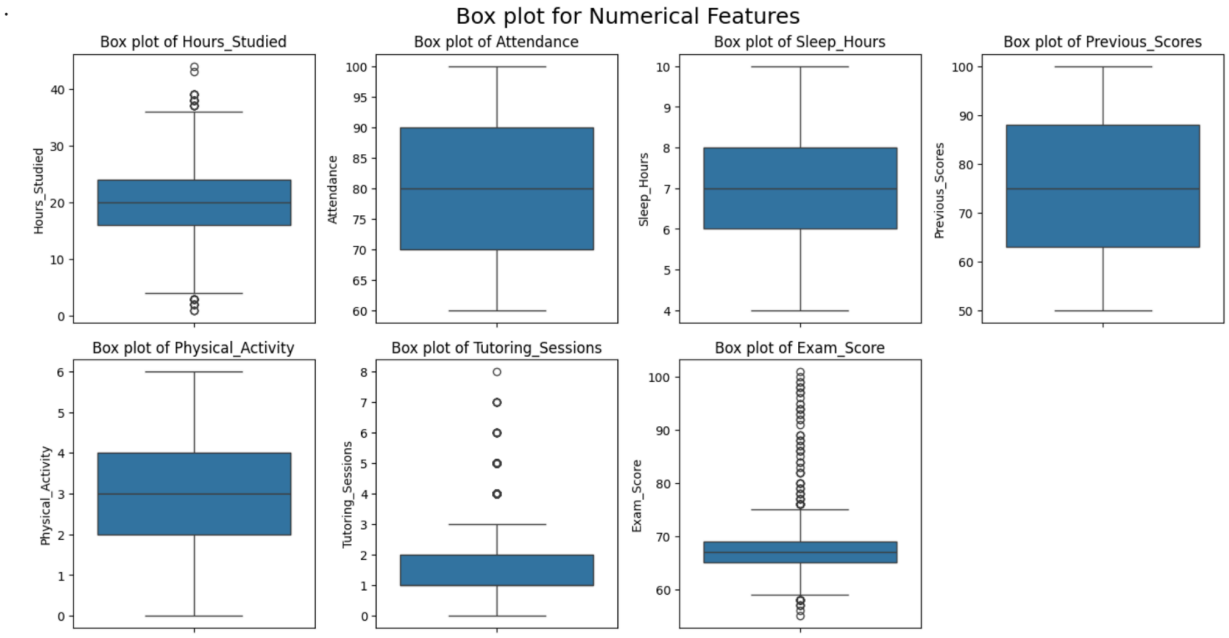
Handle Missing data

While inspecting the data , I found that a few rows (almost 1%) for Teacher quality , Distance from home , and Parental Education Level were empty. Since these are categorical columns , it makes sense to fill all empty rows with the mode value for these columns. Also , there were no duplicate rows found in the dataset.

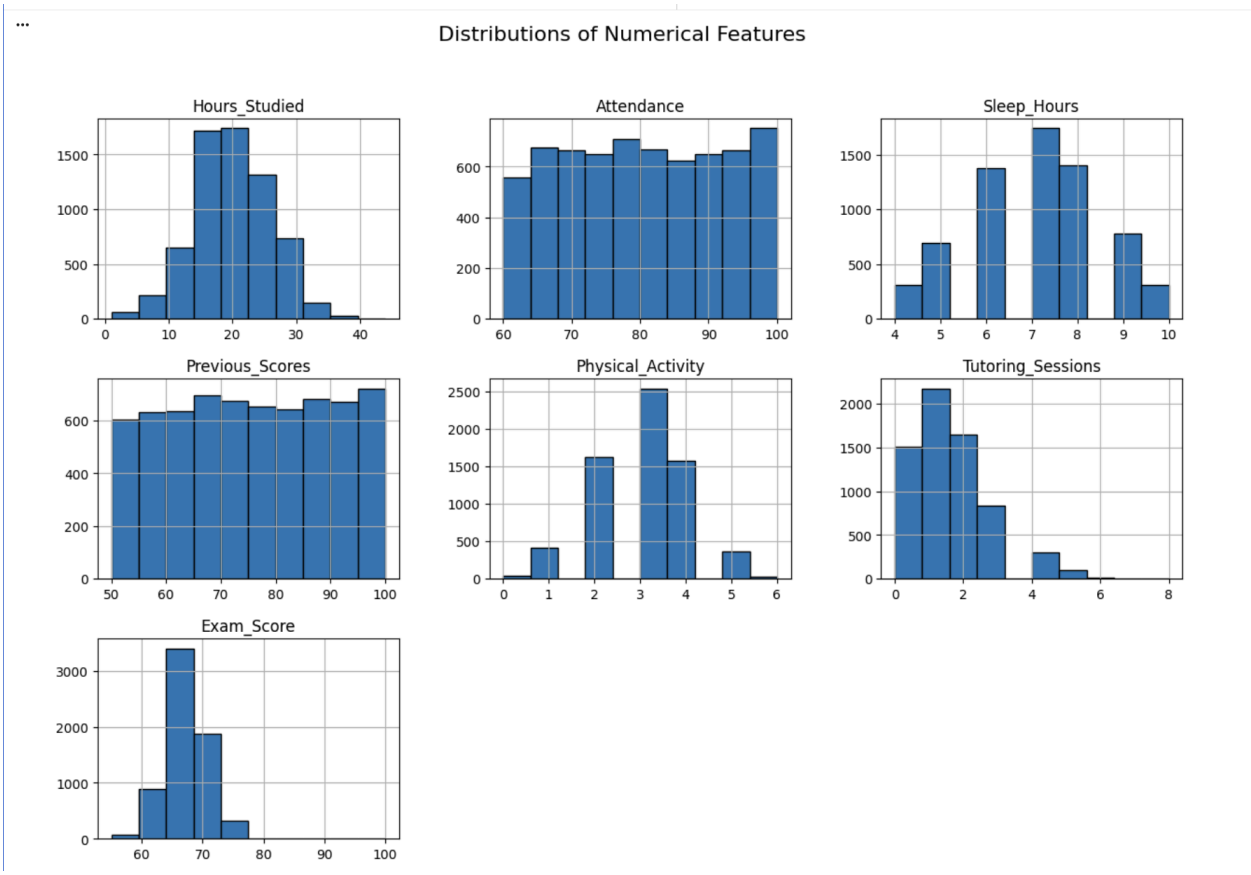
Explore data patterns

For exploring the data patterns , I used boxPlots , histograms for numerical features, and countplot for categorical features, which gives frequency distribution for different features across different categories.

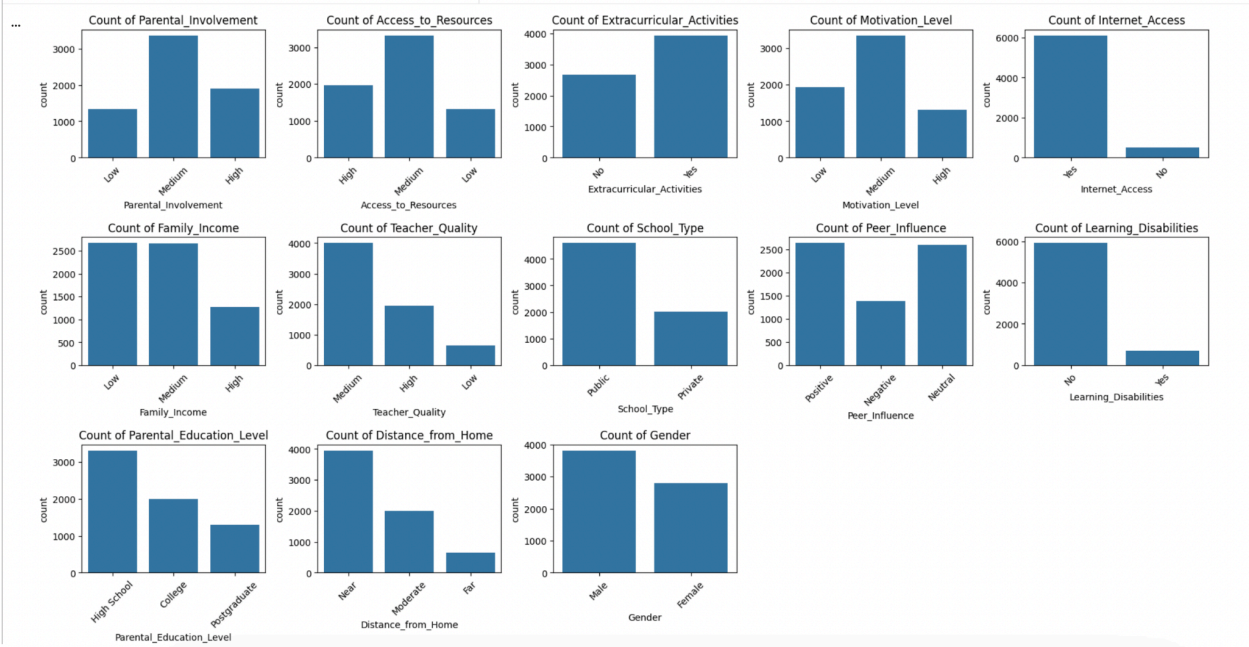
BOXPLOTS FOR ALL NUMERICAL COLUMNS



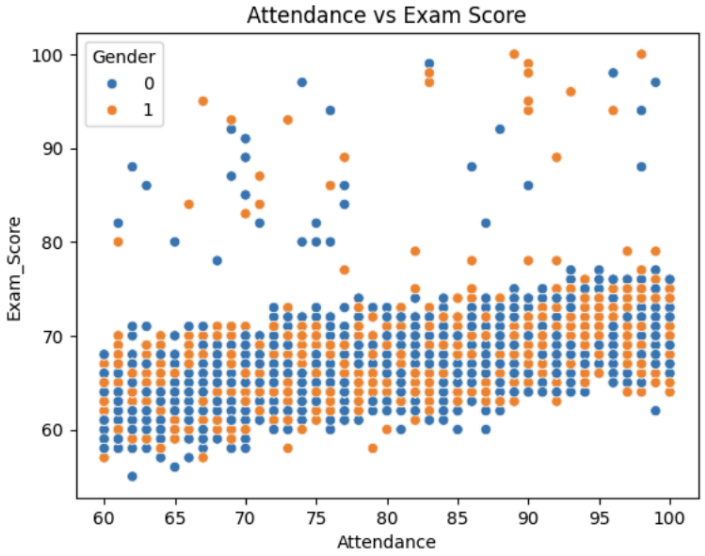
HISTOGRAMS FOR ALL NUMERICAL COLUMNS



COUNTPLOT FOR CATEGORICAL COLUMNS

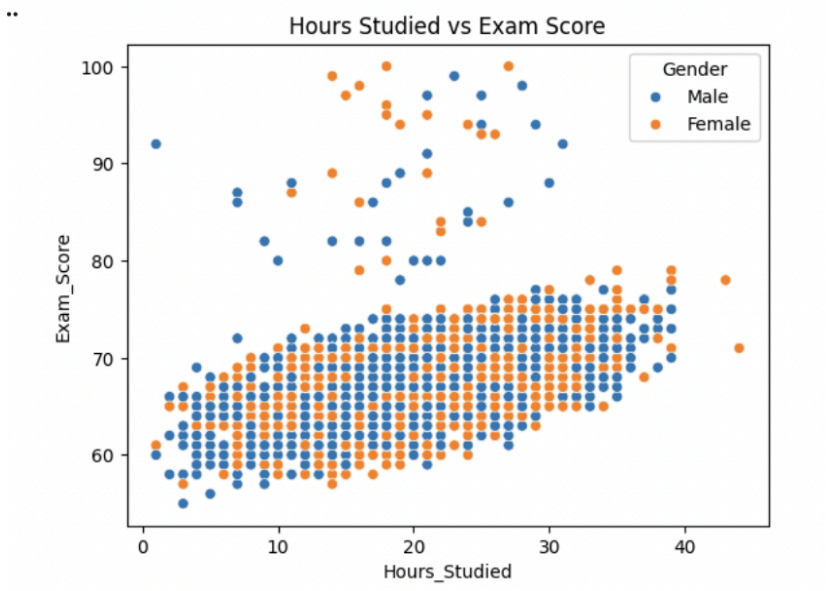


For understanding the distribution of data between two numerical features , I used Scatterplot. For Example this diagram shows the distribution of exam score with respect to attendance.



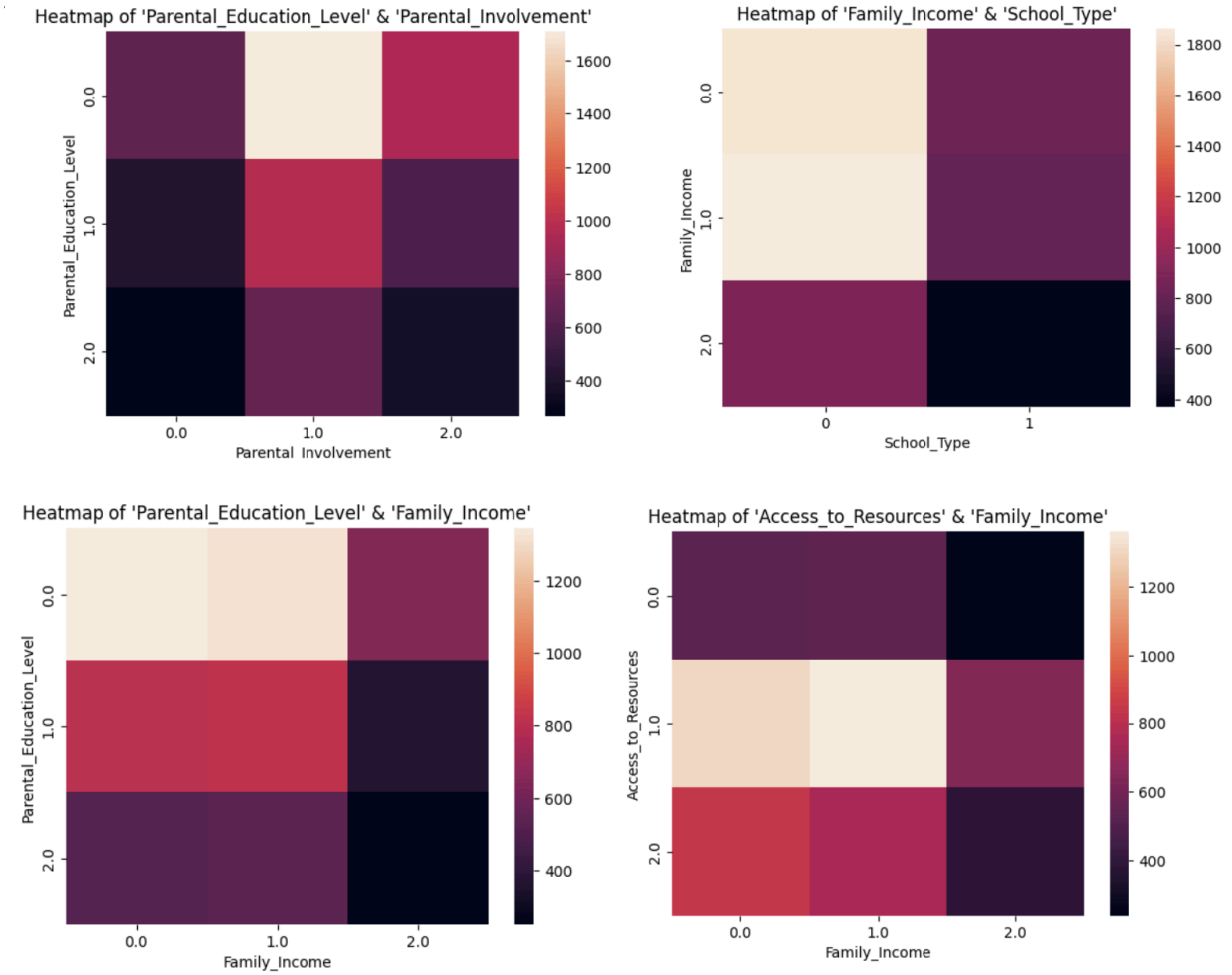
SCATTER PLOT FOR ATTENDANCE VS EXAM SCORE

Here is another diagram which shows the distribution of Exam Score with respect to hours studied.



SCATTER PLOT FOR HOURS STUDIED VS EXAM SCORE

Also Finally to understand the distribution between two categorical features , I used crosstabs. Following heatmap diagrams show data distribution between two categorical features , including Parental Education Level, Parental Involvement , Family Income , School Type and Access to Resources



Handle Outliers

In the box plot diagram , we see outliers for 3 features , Exam Score , Tutoring Sessions and Hours studied.

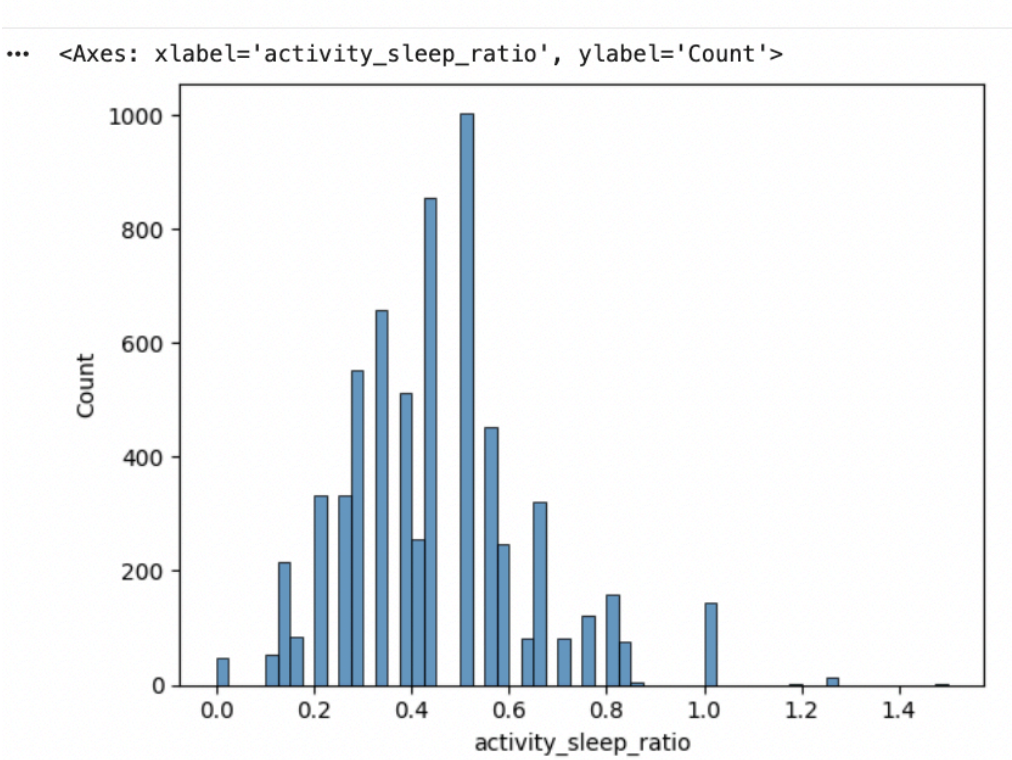
For Exam_score , 1 student has a score greater than 100. Since it is 101, it looks like it is a typing mistake and has been corrected to 100. Rest of the outliers are fine because student scores can vary till 100.

For the Tutoring session 1266 students are shown as outliers in the box plot. These many students can't be outliers. Some students may require more hours of tutoring sessions , so it does not need to be handled.

For Hours Studied , Some students may study for longer hours , which is perfectly fine. It does not look like the dataset is wrong or needs correction.

Data Transformation

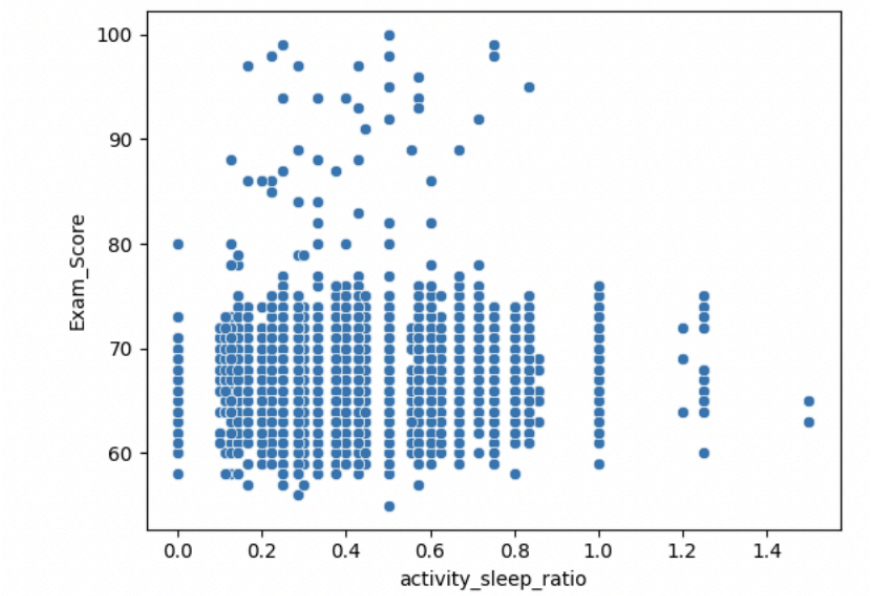
1. None of the numerical columns hold big values , so there is no need for scaling the dataset.
2. Many categorical columns have ordinal values , for eg Parental Involvement, Parental Education Level , Access to Resources etc and also nominal values like Extracurricular Activities, internet Access etc. So, For encoding categories with ordinal value , I used ordinal encoding and for those with nominal values , I used label encoding.
3. Next I wanted to analyze the activity to sleep ratio to see how active vs those with more sedentary lifestyle perform in the exams. I calculated the ratio for physical activity by sleep hours. Looking at the histogram , the ratio is almost distributed normally with little skewed to the right.



Low ratio indicates that students have a sedentary lifestyle, perform less physical activity and sleep more, whereas higher ratio indicates that the child is very active and performs more physical activity and sleep less.

Mode is around .5, which means most students have balanced physical activity vs sleep hours.

If we look at the correlation and distribution between activity_sleep_ratio and exam_score , we can see that correlation is not significant. But in the distribution chart we see that most of the top sorcerers have either a balanced or sedentary lifestyle , which makes sense .




```
dataset1[["activity_sleep_ratio","Exam_Score"]].corr()
```

	activity_sleep_ratio	Exam_Score
activity_sleep_ratio	1.000000	0.035205
Exam_Score	0.035205	1.000000

I also tried the correlation between activity_sleep_ratio and Hours_Studied . Though the value is not significant , it's a negative value , which suggests that as the activity_sleep_ratio increases , which means physical activity increases, the hours studied decreases . This also holds true in a practical sense.

```
dataset1[["activity_sleep_ratio","Hours_Studied"]].corr()
```

	activity_sleep_ratio	Hours_Studied
activity_sleep_ratio	1.000000	-0.000292
Hours_Studied	-0.000292	1.000000

4. Finally , I wanted to check if Parental_Education_Level and Parental_Involvement have any correlation with each other and with Exam_Score. Since these are categorical features , I encoded them first with ordinal encoding. Then , I tried to look at the ratio , but that did not make any sense and was hard to conclude. Then I looked at the correlation between Parental_Education_Level and Parental_Involvement. The correlation value is insignificant , though it's a negative , which indicates that the more educated parents spend less time with their children.

This makes sense in practical life. We see that people with higher degrees are so occupied in their work that they barely spend time with their children.

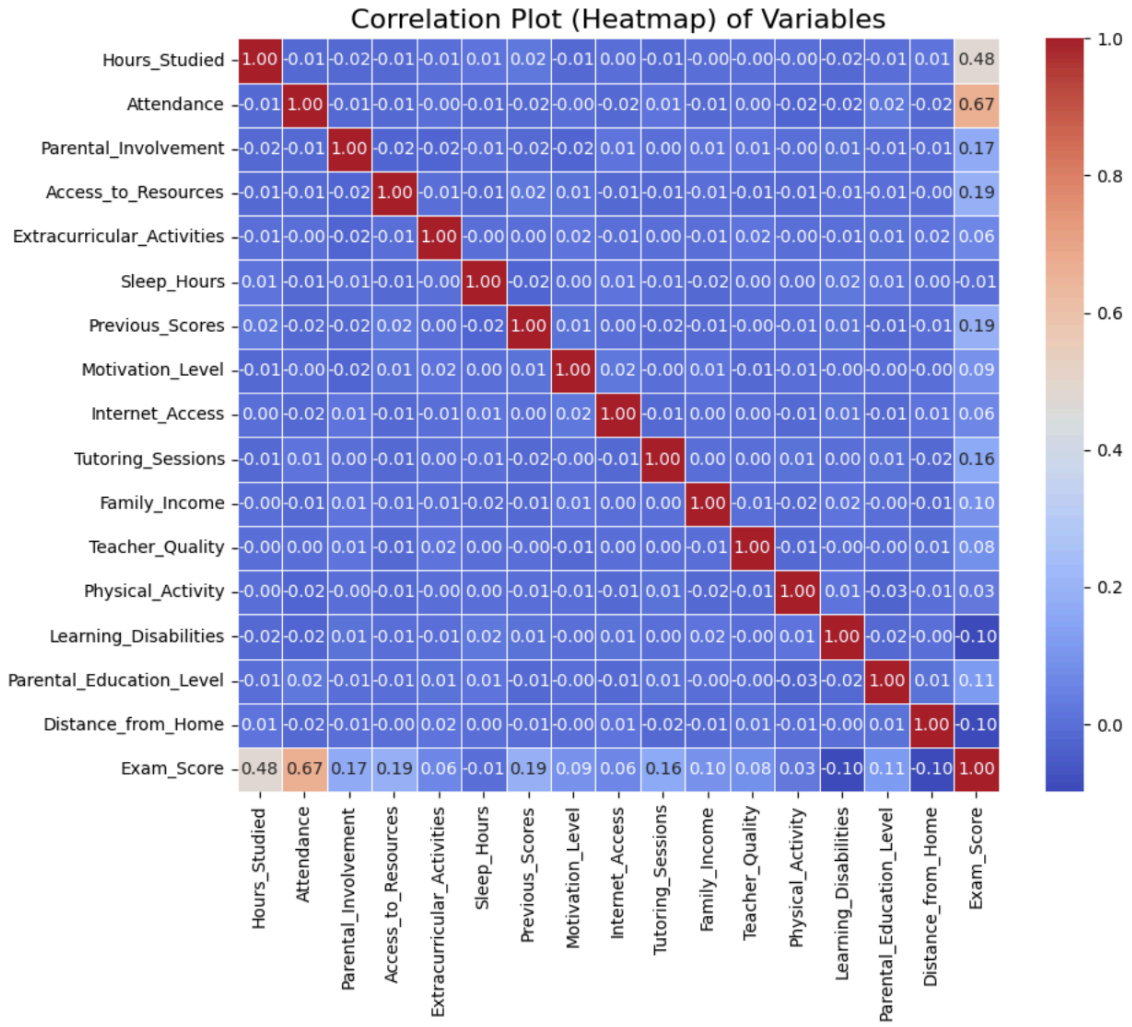
```
dataset1[["Parental_Education_Level","Parental_Involvement"]].corr(method="spearman")
```

	Parental_Education_Level	Parental_Involvement
Parental_Education_Level	1.000000	-0.005899
Parental_Involvement	-0.005899	1.000000

We also see this in correlation heatmap , that parental involvement has a significant impact on students' exam scores.

Correlations

For finding correlation between variables , I used the corr() function with method “Spearman”.



I dropped a few columns for correlation , for eg Peer_Influence , school_Type , because encoding just represents numbers that don't have any meaning attached to it.

Insights

- The data for most of the features is ***normally distributed*** , except a few. For Eg: Almost 6000 students out of 6700 have access to the internet and almost 5900 students out of total don't have any learning disability.
- Few Features are also ***Right-Skewed*** , For eg: Parental education level , distance from home , teacher quality.
- Most of the top scorers have a balanced or sedentary lifestyle.
- ***Attendance*** and ***Hour studied*** have the highest correlation with ***exam score***.
- ***Distance from home*** and ***learning disability*** has ***negative correlation*** with exam score, which indicates that as the learning disability increases , the exam score decreases , which holds true for real life. Also as the distance from home increases , the exam score decreases , because students may be travelling for long time which affects their performance.
- ***Parental Involvement , Access to Resources , Previous Score, Tutoring Session , Family Income , Parental Educational Level*** also have significant correlation with exam score, which clearly indicates that all these factors contribute to the success of a child in academics .
- ***Parental Involvement*** and ***Parental Educational Level*** have -ve correlation , which means that parents with higher degrees spend less time with kids.

Limitations

After going through the dataset and performing the EDA , it looks like the dataset is realistic , but not enough . More samples of the dataset will be required to get bigger and accurate numbers.

Also I felt that it's hard to relate , process and combine categorical features especially with nominal values. It's hard to interpret these columns and reach a conclusion.

Though the direction of the numbers and the summary of the EDA seems real and authentic , more dataset is needed to reach precise and clear conclusions.

Appendix

November 25, 2025

[1]:
```
from google.colab import drive
```

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import OrdinalEncoder,OneHotEncoder
```

[3]:
```
dataset1 = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
↪Fall25_ISE_Mathematics/Project_Datasets/StudentPerformanceFactors.csv')
```

[4]:
```
dataset1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6607 entries, 0 to 6606
Data columns (total 20 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Hours_Studied               6607 non-null   int64
 1   Attendance                  6607 non-null   int64
 2   Parental_Involvement        6607 non-null   object
 3   Access_to_Resources         6607 non-null   object
 4   Extracurricular_Activities  6607 non-null   object
 5   Sleep_Hours                 6607 non-null   int64
 6   Previous_Scores             6607 non-null   int64
 7   Motivation_Level            6607 non-null   object
 8   Internet_Access             6607 non-null   object
 9   Tutoring_Sessions           6607 non-null   int64
 10  Family_Income               6607 non-null   object
 11  Teacher_Quality             6529 non-null   object
 12  School_Type                 6607 non-null   object
 13  Peer_Influence              6607 non-null   object
 14  Physical_Activity           6607 non-null   int64
 15  Learning_Disabilities       6607 non-null   object
 16  Parental_Education_Level    6517 non-null   object
 17  Distance_from_Home          6540 non-null   object
 18  Gender                      6607 non-null   object
 19  Exam_Score                  6607 non-null   int64
```

```
dtypes: int64(7), object(13)
memory usage: 1.0+ MB
```

[5]:
```
dataset1.shape
```

[5]:
```
(6607, 20)
```

[6]:
```
dataset1.head(20)
```

[6]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level
0	23	84	Low	High	No	7	73	Low
1	19	64	Low	Medium	No	8	59	Low
2	24	98	Medium	Medium	Yes	7	91	Medium
3	29	89	Low	Medium	Yes	8	98	Medium
4	19	92	Medium	Medium	Yes	6	65	Medium
5	19	88	Medium	Medium	Yes	8	89	Medium
6	29	84	Medium	Low	Yes	7	68	Low
7	25	78	Low	High	Yes	6	50	Medium
8	17	94	Medium	High	No	6	80	High
9	23	98	Medium	Medium	Yes	8	71	Medium
10	17	80	Low	High	No	8	88	Medium
11	17	97	Medium	High	Yes	6	87	Low
12	21	83	Medium	Medium	Yes	8	97	Low
13	9	82	Medium	Medium	Yes	8	72	Medium
14	10	78	Medium	High	Yes	8	74	Medium
15	17	68	Medium	Medium				
16	14	60	Medium	Low				
17	22	70	Low	Medium				
18	15	80	Medium	Medium				
19	12	75	Medium	High				

15	No	8	70	Medium
16	Yes	10	65	Low
17	Yes	6	82	Medium
18	Yes	9	91	Low
19	Yes	7	58	Medium

	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	\
0	Yes	0	Low	Medium	
1	Yes	2	Medium	Medium	
2	Yes	2	Medium	Medium	
3	Yes	1	Medium	Medium	
4	Yes	3	Medium	High	
5	Yes	3	Medium	Medium	
6	Yes	1	Low	Medium	
7	Yes	1	High	High	
8	Yes	0	Medium	Low	
9	Yes	0	High	High	
10	No	4	Medium	High	
11	Yes	2	Low	High	
12	Yes	2	Medium	Medium	
13	Yes	2	Medium	Medium	
14	Yes	1	Low	Medium	
15	Yes	2	Medium	Medium	
16	Yes	0	High	Medium	
17	Yes	1	Low	High	
18	Yes	3	Low	Medium	
19	Yes	3	Medium	Medium	

	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities	\
0	Public	Positive	3	No	
1	Public	Negative	4	No	
2	Public	Neutral	4	No	
3	Public	Negative	4	No	
4	Public	Neutral	4	No	
5	Public	Positive	3	No	
6	Private	Neutral	2	No	
7	Public	Negative	2	No	
8	Private	Neutral	1	No	
9	Public	Positive	5	No	
10	Private	Neutral	4	No	
11	Private	Neutral	2	No	
12	Public	Positive	4	No	
13	Private	Positive	3	No	
14	Private	Neutral	4	No	
15	Private	Positive	4	No	
16	Private	Positive	3	No	
17	Public	Neutral	3	No	

```
18      Public       Positive                 2                   No
19     Private       Positive                 4                   No

   Parental_Education_Level Distance_from_Home  Gender  Exam_Score
0               High School              Near    Male          67
1                   College          Moderate  Female          61
2              Postgraduate              Near    Male          74
3               High School          Moderate    Male          71
4                   College              Near  Female          70
5              Postgraduate              Near    Male          71
6               High School          Moderate    Male          67
7               High School               Far    Male          66
8                   College              Near    Male          69
9               High School          Moderate    Male          72
10                  College          Moderate    Male          68
11              High School              Near    Male          71
12              High School              Near    Male          70
13             Postgraduate              Near    Male          66
14             Postgraduate              Near    Male          65
15              High School              Near  Female          64
16                  College              Near    Male          60
17              High School              Near  Female          65
18                  College          Moderate  Female          67
19                  College              Near    Male          66
```

[7]:
```
dataset1.describe()
```

[7]:
```
       Hours_Studied   Attendance  Sleep_Hours  Previous_Scores  \
count    6607.000000  6607.000000   6607.00000      6607.000000
mean       19.975329    79.977448      7.02906        75.070531
std         5.990594    11.547475      1.46812        14.399784
min         1.000000    60.000000      4.00000        50.000000
25%        16.000000    70.000000      6.00000        63.000000
50%        20.000000    80.000000      7.00000        75.000000
75%        24.000000    90.000000      8.00000        88.000000
max        44.000000   100.000000     10.00000       100.000000

       Tutoring_Sessions  Physical_Activity   Exam_Score
count        6607.000000        6607.000000  6607.000000
mean            1.493719           2.967610    67.235659
std             1.230570           1.031231     3.890456
min             0.000000           0.000000    55.000000
25%             1.000000           2.000000    65.000000
50%             1.000000           3.000000    67.000000
75%             2.000000           4.000000    69.000000
max             8.000000           6.000000   101.000000
```

[8]:
```
dataset1[dataset1.isnull().any(axis=1) > 0]
```

[8]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	\
33	14	60	High	Medium	
127	17	97	Medium	Medium	
240	15	87	Low	Medium	
275	23	82	Low	Medium	
316	24	90	Low	Low	
…	…	…	…	…	
6502	23	64	Medium	Medium	
6579	9	84	Medium	Medium	
6589	22	90	Low	High	
6594	9	90	High	High	
6596	17	92	Medium	Medium	

	Extracurricular_Activities	Sleep_Hours	Previous_Scores	\
33	No	5	50	
127	No	8	89	
240	No	4	54	
275	Yes	8	94	
316	No	7	83	
…	…	…	…	
6502	No	7	75	
6579	No	6	74	
6589	No	5	99	
6594	Yes	7	79	
6596	No	7	66	

	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	\
33	Medium	Yes	2	Medium	
127	Medium	Yes	1	Low	
240	Medium	Yes	1	Medium	
275	Medium	Yes	1	Medium	
316	Medium	Yes	0	Low	
…	…	…	…	…	
6502	Medium	Yes	2	Medium	
6579	Medium	Yes	5	High	
6589	Medium	Yes	1	Low	
6594	Low	Yes	4	High	
6596	Low	Yes	2	Low	

	Teacher_Quality	School_Type	Peer_Influence	Physical_Activity	\
33	Medium	Public	Neutral	3	
127	NaN	Public	Neutral	4	
240	Medium	Public	Neutral	6	
275	Medium	Public	Negative	2	
316	Medium	Private	Positive	3	

```
…              …         …             …                  …
6502           High      Public       Positive                2
6579            NaN      Public        Neutral                2
6589            Low     Private       Positive                2
6594           High      Public       Positive                4
6596            NaN      Public       Negative                3

     Learning_Disabilities Parental_Education_Level Distance_from_Home  \
33                      No                 College                NaN
127                     No             High School                Far
240                     No                    NaN           Moderate
275                    Yes             High School                NaN
316                     No                 College                NaN
…                      …                       …                  …
6502                    No                    NaN               Near
6579                    No             High School               Near
6589                    No                 College                NaN
6594                    No             High School                NaN
6596                    No             High School               Near

      Gender  Exam_Score
33    Female          61
127     Male          69
240     Male          65
275     Male          66
316     Male          68
…       …            …
6502  Female          66
6579    Male          67
6589  Female          70
6594    Male          70
6596    Male          66

[229 rows x 20 columns]
```

1 *Missing Values*

[9]:
```
missing_data_count = dataset1.isnull().sum()
print(missing_data_count)
```

```
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
```

[10]:
```
missing_data_percent=dataset1.isnull().sum()*100/len(dataset1)
print(missing_data_percent)
```

```
Hours_Studied                 0.000000
Attendance                    0.000000
Parental_Involvement          0.000000
Access_to_Resources           0.000000
Extracurricular_Activities    0.000000
Sleep_Hours                   0.000000
Previous_Scores               0.000000
Motivation_Level              0.000000
Internet_Access               0.000000
Tutoring_Sessions             0.000000
Family_Income                 0.000000
Teacher_Quality               1.180566
School_Type                   0.000000
Peer_Influence                0.000000
Physical_Activity             0.000000
Learning_Disabilities         0.000000
Parental_Education_Level      1.362192
Distance_from_Home            1.014076
Gender                        0.000000
Exam_Score                    0.000000
dtype: float64
```

[11]:
```
missing_data=pd.DataFrame({'count-missing': missing_data_count, 'percentage':␣
↪missing_data_percent})
print(missing_data)
```

```
                           count-missing  percentage
Hours_Studied                          0    0.000000
Attendance                             0    0.000000
```

```
Parental_Involvement                0    0.000000
Access_to_Resources                 0    0.000000
Extracurricular_Activities          0    0.000000
Sleep_Hours                         0    0.000000
Previous_Scores                     0    0.000000
Motivation_Level                    0    0.000000
Internet_Access                     0    0.000000
Tutoring_Sessions                   0    0.000000
Family_Income                       0    0.000000
Teacher_Quality                    78    1.180566
School_Type                         0    0.000000
Peer_Influence                      0    0.000000
Physical_Activity                   0    0.000000
Learning_Disabilities               0    0.000000
Parental_Education_Level           90    1.362192
Distance_from_Home                 67    1.014076
Gender                              0    0.000000
Exam_Score                          0    0.000000
```

[12]:
```
print(missing_data[missing_data['count-missing']>0])
```

```
                        count-missing  percentage
Teacher_Quality                    78    1.180566
Parental_Education_Level           90    1.362192
Distance_from_Home                 67    1.014076
```

2 *Handling Missing Values*

[13]:
```
#dataset1['Teacher_Quality'].fillna(dataset1['Teacher_Quality'].mode()[0],␣
 ↪inplace=True)
missing_data_col=missing_data[missing_data['count-missing']>0].index
print(missing_data_col)
for each in missing_data_col:
  dataset1.fillna({each: dataset1[each].mode()[0]},inplace=True)


dataset1.isnull().sum()
```

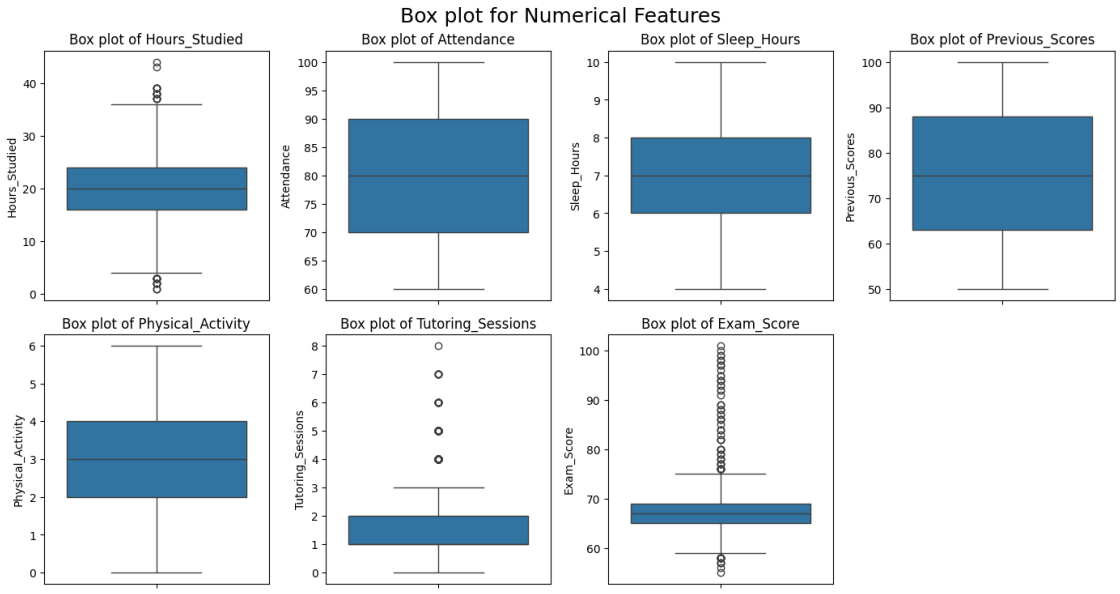
```
Index(['Teacher_Quality', 'Parental_Education_Level', 'Distance_from_Home'],
dtype='object')
```

[13]:
```
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
```

3 *Boxplots for Numerical data*

[14]:
```
numerical_cols = ["Hours_Studied", "Attendance", "Sleep_Hours",
                  "Previous_Scores", "Physical_Activity", "Tutoring_Sessions",␣
 ↪"Exam_Score"]

plt.figure(figsize=(14, 11))
for i, col in enumerate(numerical_cols, 1):
    plt.subplot(3, 4, i)
    sns.boxplot(dataset1[col])
    plt.xticks(rotation=30)
    plt.title(f"Box plot of {col}")
plt.suptitle("Box plot for Numerical Features", fontsize=18)
plt.tight_layout()
plt.show()
```



4 *Handling Outliers*

[15]:
```
print(f'Count of outliers for Tutoring_Sessions:␣
↪{len(dataset1[dataset1['Tutoring_Sessions'] >= 3])}')
```

```
Count of outliers for Tutoring_Sessions: 1266
```

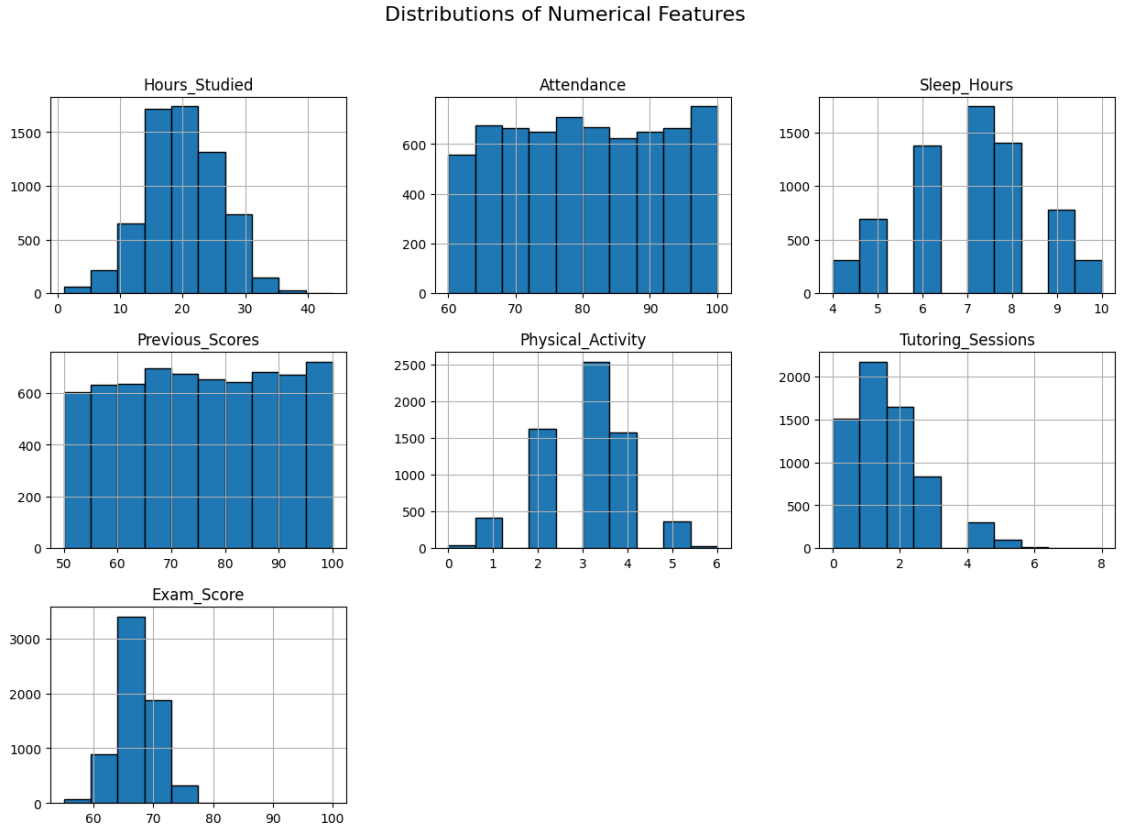
[16]:
```
print(f'Count of outliers for Exam_Score : {len(dataset1[dataset1['Exam_Score']␣
↪> 100])}')
dataset1[dataset1['Exam_Score'] > 100]
dataset1.loc[dataset1["Exam_Score"] > 100, "Exam_Score"] = 100
```

```
Count of outliers for Exam_Score : 1
```

1. For Columns Exam_score , 1 student has score greater than 100. Since , it is 101, it looks like it is a typing mistake and has been corrected to 100.
2. For Tutoring session 1266 students are shown as outliers in the box plot. These many students can't be outliers. Some students may require more hours of tutoring session , so it does not need to be handled.
3. For Hours Studied , Some students may study for longer hours , which is perfectly fine. It does not look like the dataset is wrong or needs correction.

5 *Histogram for numerical Columns*

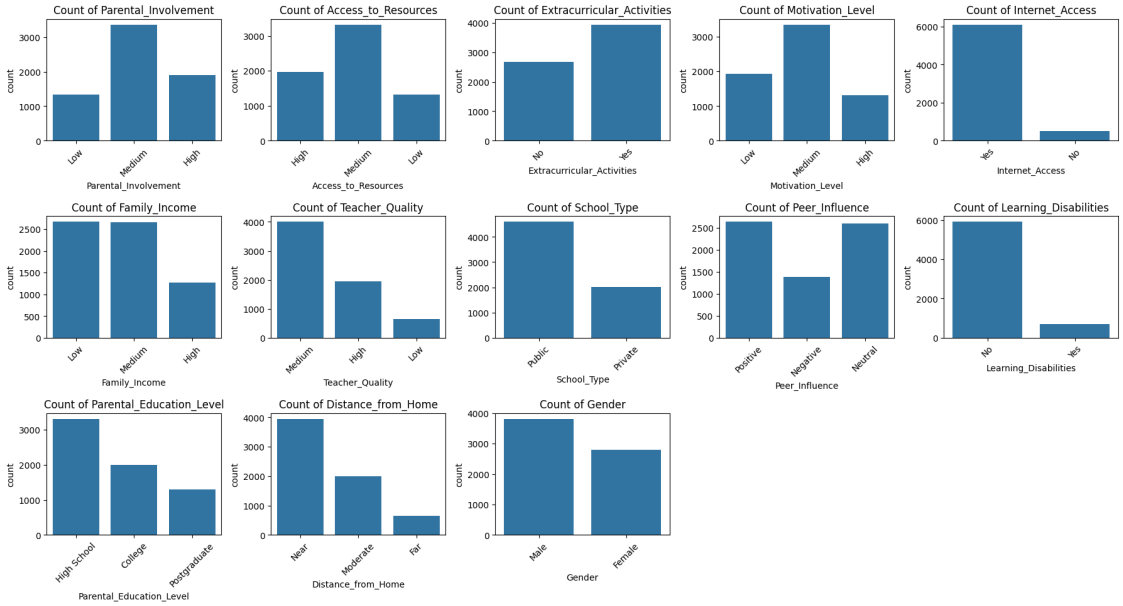
[17]:
```
dataset1[numerical_cols].hist(figsize=(15, 10), bins=10, edgecolor="black")
plt.suptitle("Distributions of Numerical Features", fontsize=16)
plt.show()
```



6 *Count Plot for Categorical Data*

[18]:
```
#### Distribution chart for categorical data
cat_cols =␣
 ↪["Parental_Involvement","Access_to_Resources","Extracurricular_Activities",
          ␣
 ↪"Motivation_Level","Internet_Access","Family_Income","Teacher_Quality",
           "School_Type","Peer_Influence","Learning_Disabilities",
           "Parental_Education_Level","Distance_from_Home","Gender"]

plt.figure(figsize=(18,15))

for i, col in enumerate(cat_cols, 1):
```

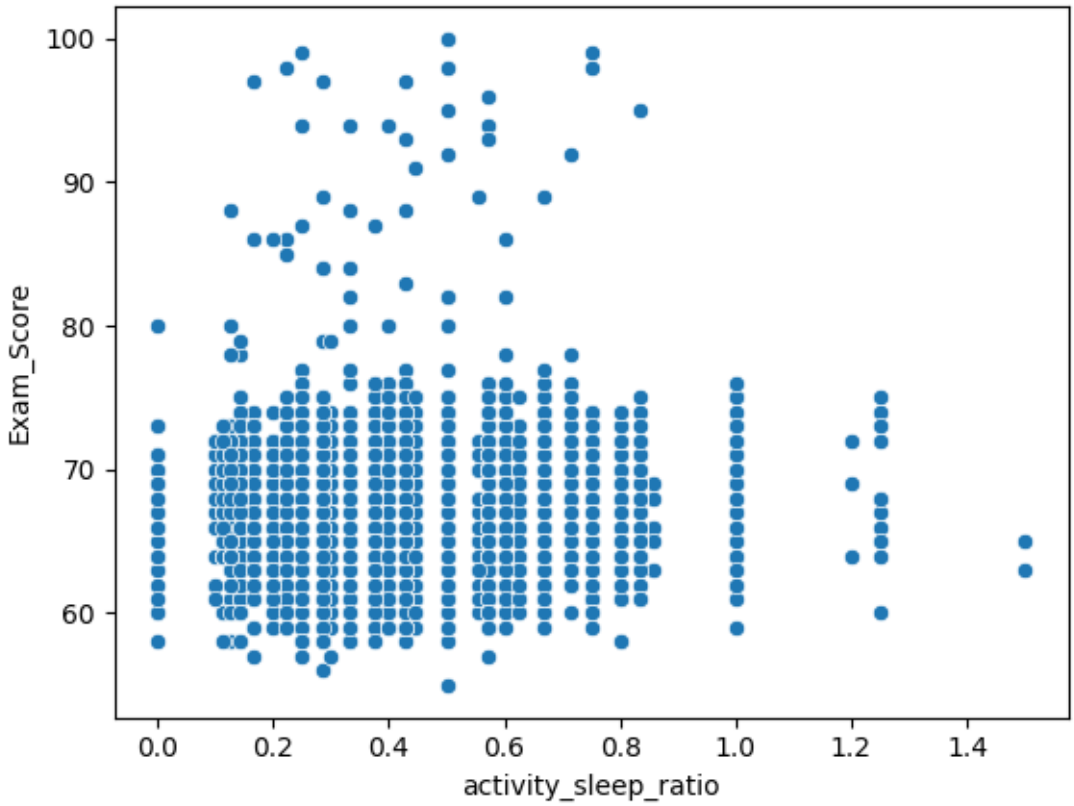
```
    plt.subplot(5, 5, i)    # 5 rows, 3 columns grid (total 15 slots for 13␣
 ↪plots)
    sns.countplot(x=col, data=dataset1)
    plt.title(f"Count of {col}")
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



[20]:
```
dataset1["activity_sleep_ratio"] = dataset1["Physical_Activity"] /␣
 ↪dataset1["Sleep_Hours"]
```

[21]:
```
sns.scatterplot(x="activity_sleep_ratio", y="Exam_Score", data=dataset1)
```

[21]:
```
<Axes: xlabel='activity_sleep_ratio', ylabel='Exam_Score'>
```



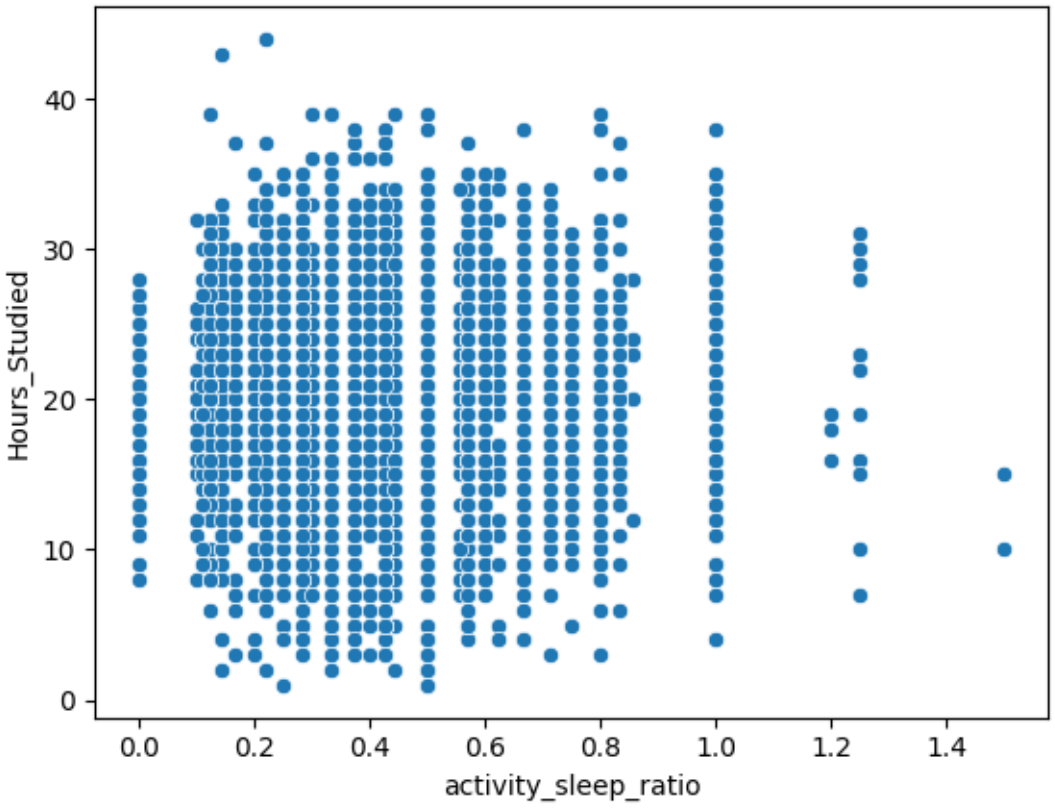
[32]:
```
dataset1[["activity_sleep_ratio","Exam_Score"]].corr()
```

[32]:

	activity_sleep_ratio	Exam_Score
activity_sleep_ratio	1.000000	0.035205
Exam_Score	0.035205	1.000000

[38]:
```
sns.scatterplot(x="activity_sleep_ratio", y="Hours_Studied", data=dataset1)
```

[38]:
```
<Axes: xlabel='activity_sleep_ratio', ylabel='Hours_Studied'>
```



[39]:
```
dataset1[["activity_sleep_ratio","Hours_Studied"]].corr()
```

[39]:

	activity_sleep_ratio	Hours_Studied
activity_sleep_ratio	1.000000	-0.000292
Hours_Studied	-0.000292	1.000000

7 *Ordinal Encoding*

[19]:
```
ordinal_cols =␣
  ↪["Parental_Involvement","Access_to_Resources","Family_Income","Teacher_Quality","Motivation
encoder = OrdinalEncoder(categories=[['Low', 'Medium', 'High']])
#dataset1['Parental_Involvement_encoded'] = encoder.
  ↪fit_transform(dataset1[['Parental_Involvement']])
for each in ordinal_cols:
  dataset1[each] = dataset1[each].astype(str)
  dataset1[each] = encoder.fit_transform(dataset1[[each]])

encoder1 = OrdinalEncoder(categories=[['High School','College','Postgraduate']])
dataset1['Parental_Education_Level'] = encoder1.
  ↪fit_transform(dataset1[['Parental_Education_Level']])
```

```
encoder2 = OrdinalEncoder(categories=[['Near','Moderate','Far']])
dataset1['Distance_from_Home'] = encoder2.
 ↪fit_transform(dataset1[['Distance_from_Home']])
```

8 *Label/Binary Encoding*

[22]:
```
dataset1['Gender'] = dataset1['Gender'].map({'Male': 0, 'Female': 1})
dataset1['School_Type'] = dataset1['School_Type'].map({'Public': 0, 'Private':␣
 ↪1})
dataset1['Extracurricular_Activities'] = dataset1['Extracurricular_Activities'].
 ↪map({'No': 0, 'Yes': 1})
dataset1['Internet_Access'] = dataset1['Internet_Access'].map({'No': 0, 'Yes':␣
 ↪1})
dataset1['Peer_Influence'] = dataset1['Peer_Influence'].map({'Negative': 0,␣
 ↪'Neutral': 1, 'Positive': 2})
dataset1['Learning_Disabilities'] = dataset1['Learning_Disabilities'].map({'No':
 ↪ 0, 'Yes': 1})
```

[132]:
```
dataset1.head()
```

[132]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	\
0	23	84	0.0	2.0	
1	19	64	0.0	1.0	
2	24	98	1.0	1.0	
3	29	89	0.0	1.0	
4	19	92	1.0	1.0	

	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	\
0	0	7	73	0.0	
1	0	8	59	0.0	
2	1	7	91	1.0	
3	1	8	98	1.0	
4	1	6	65	1.0	

	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Quality	\
0	1	0	0.0	1.0	
1	1	2	1.0	1.0	
2	1	2	1.0	1.0	
3	1	1	1.0	1.0	
4	1	3	1.0	2.0	

	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities	\
0	0	2	3	0	

```
1              0                0                4                    0
2              0                1                4                    0
3              0                0                4                    0
4              0                1                4                    0
```

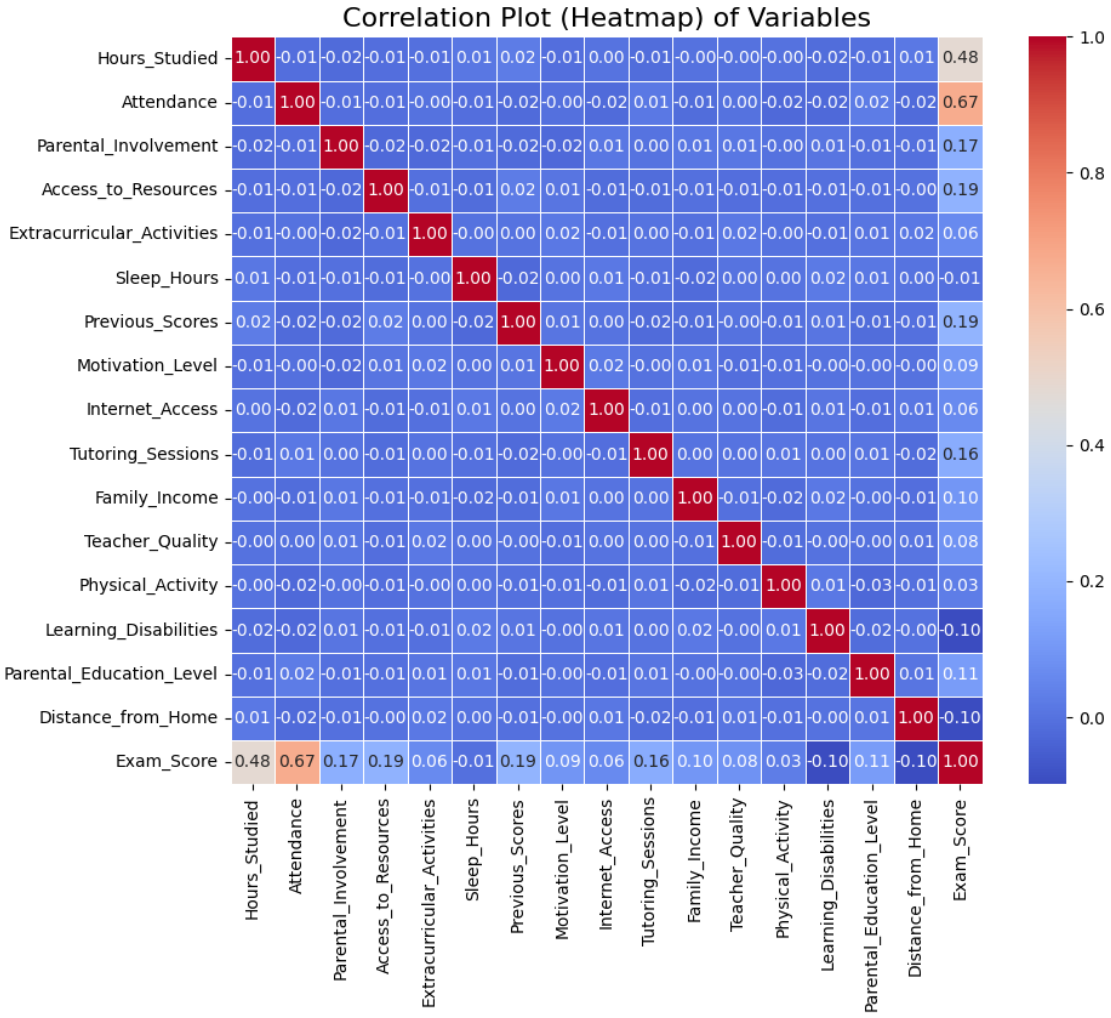
```
   Parental_Education_Level  Distance_from_Home  Gender  Exam_Score
0                       0.0                 0.0       0          67
1                       1.0                 1.0       1          61
2                       2.0                 0.0       0          74
3                       0.0                 1.0       0          71
4                       1.0                 0.0       1          70
```

9 *Correlation HeatMap*

10 *Correlation Matrix For Columns*

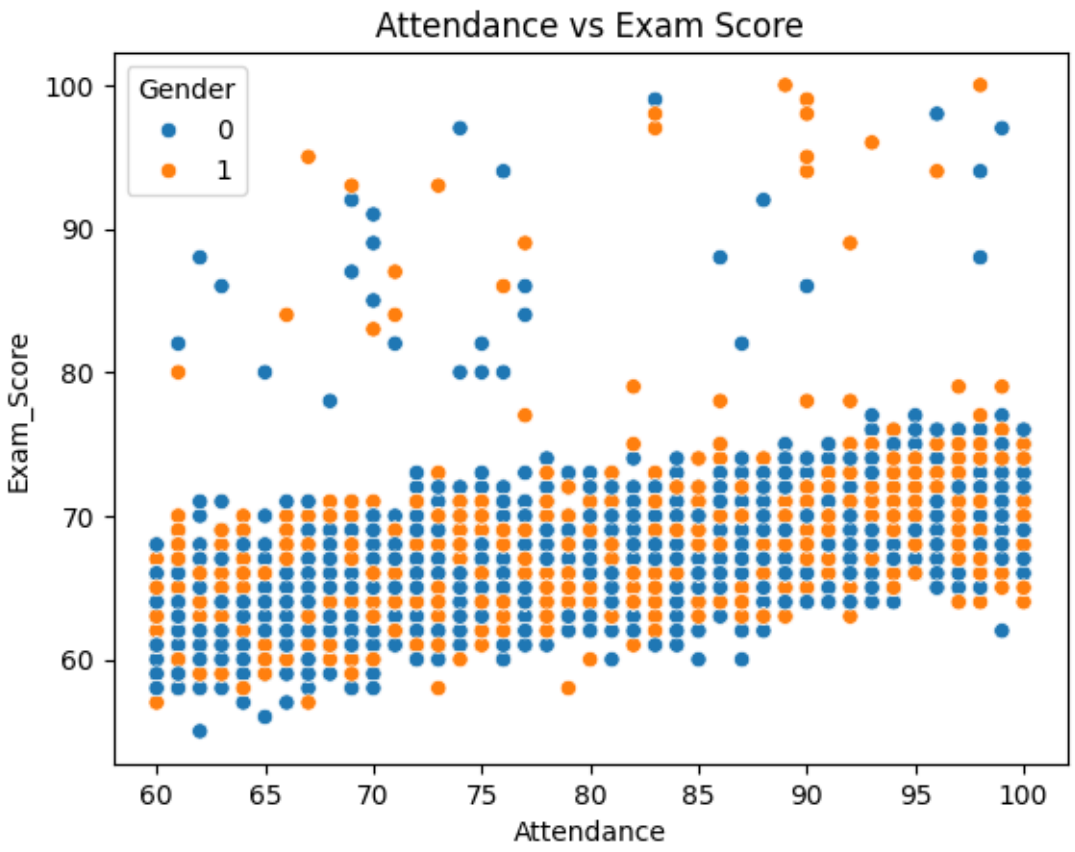
[25]:
```
dataset1_Corr_Columns = dataset1.
↪drop(["Gender","School_Type","Peer_Influence","activity_sleep_ratio"],␣
↪axis=1)
```

[26]:
```
correlation_matrix = dataset1_Corr_Columns.corr(method="spearman")
plt.figure(figsize=(10, 8))
sns.heatmap(
    correlation_matrix,
    annot=True,          # Show the correlation values on the heatmap
    cmap='coolwarm',     # Use a diverging color palette
    fmt=".2f",           # Format the annotations to two decimal places
    linewidths=.5,       # Add lines to separate the cells
)

plt.title('Correlation Plot (Heatmap) of Variables', fontsize=16)
plt.show()
```



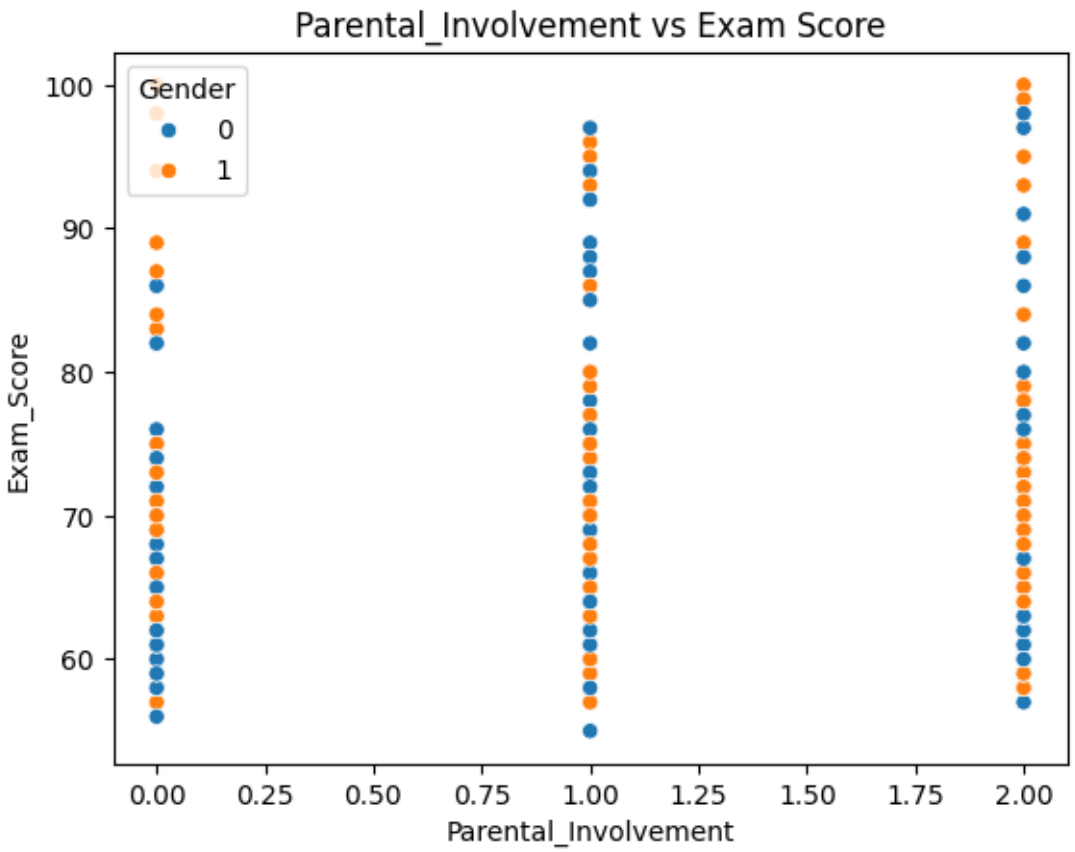
11 *More Data Exploration*

[135]:
```
sns.scatterplot(data=dataset1, x='Attendance', y='Exam_Score', hue='Gender')
plt.title("Attendance vs Exam Score")
plt.show()
```

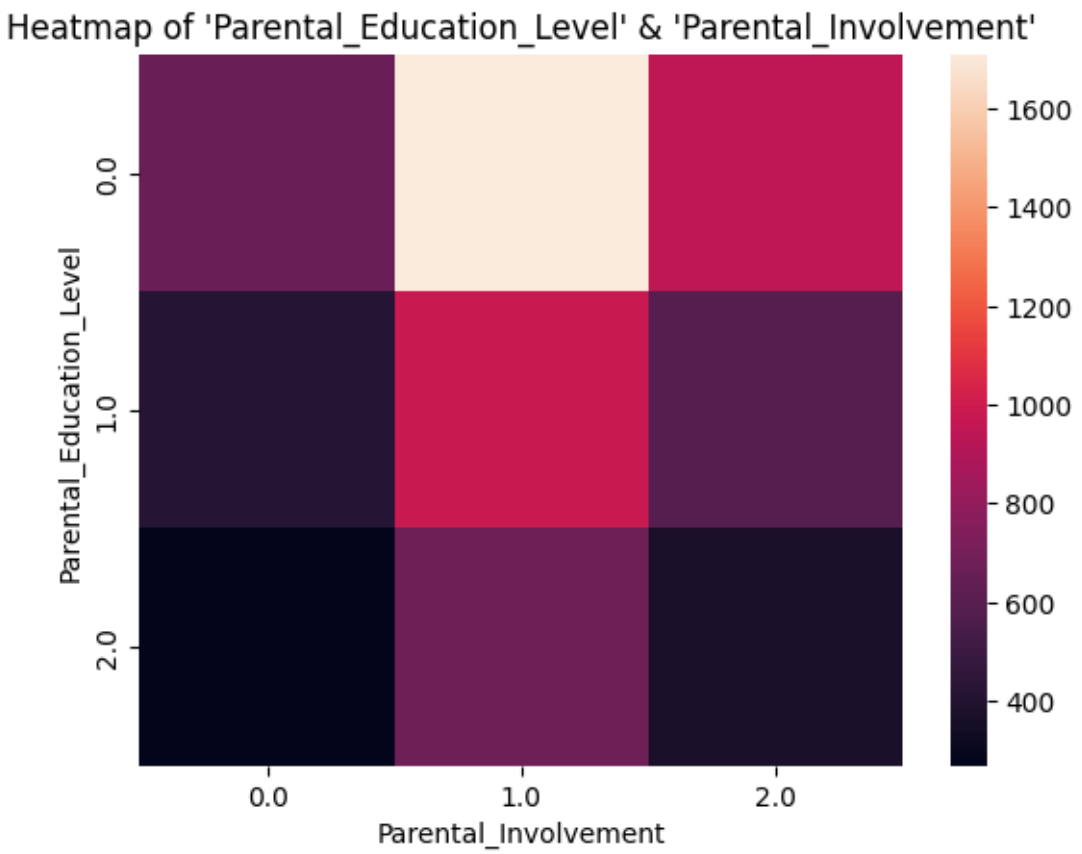


[136]:
```
sns.scatterplot(data=dataset1, x='Hours_Studied', y='Exam_Score', hue='Gender')
plt.title("Hours Studied vs Exam Score")
plt.show()
```

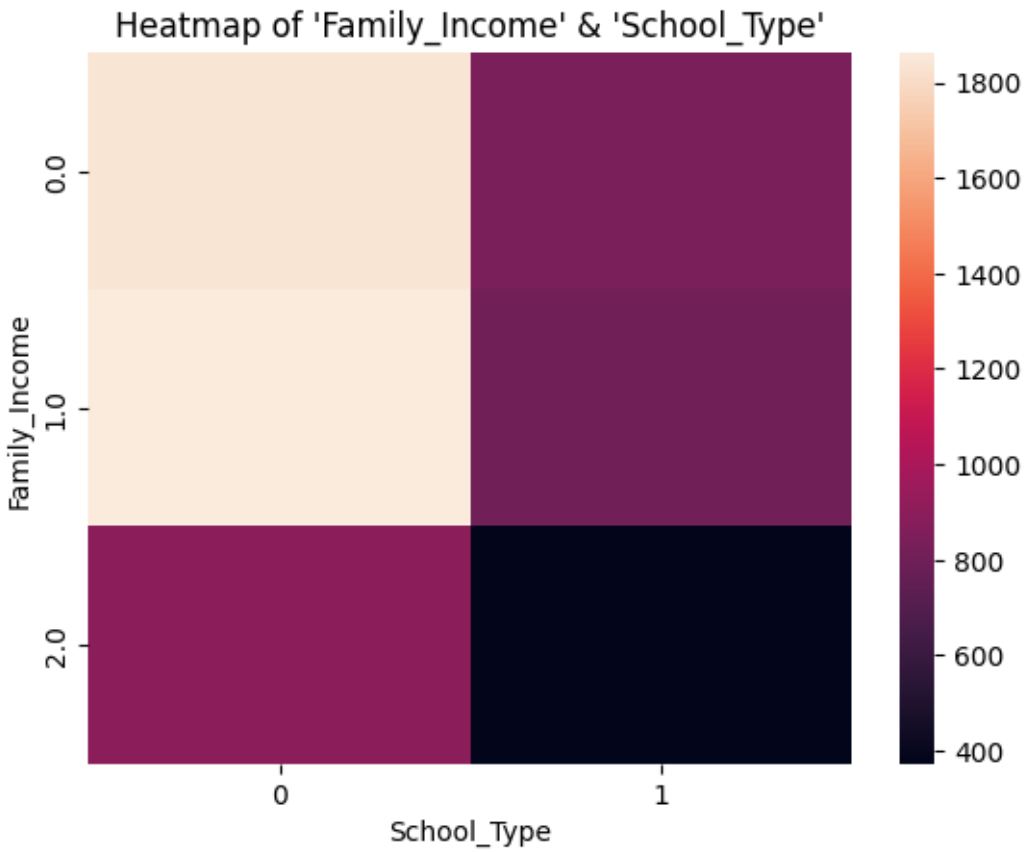
[137]:
```
sns.scatterplot(data=dataset1, x='Parental_Involvement', y='Exam_Score',␣
↪hue='Gender')
plt.title("Parental_Involvement vs Exam Score")
plt.show()
```



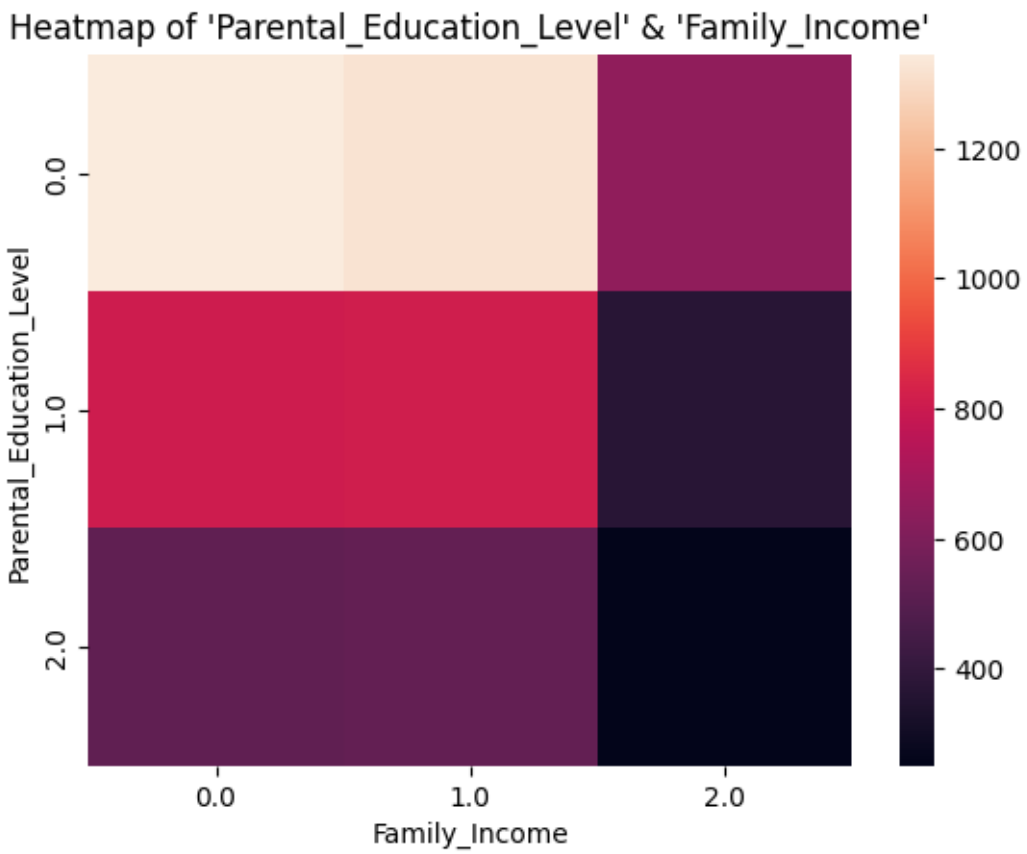
[138]:
```
cross_tab=pd.
  ↪crosstab(dataset1['Parental_Education_Level'],dataset1['Parental_Involvement'])
sns.heatmap(cross_tab)
plt.title("Heatmap of 'Parental_Education_Level' & 'Parental_Involvement'")
plt.show()
```



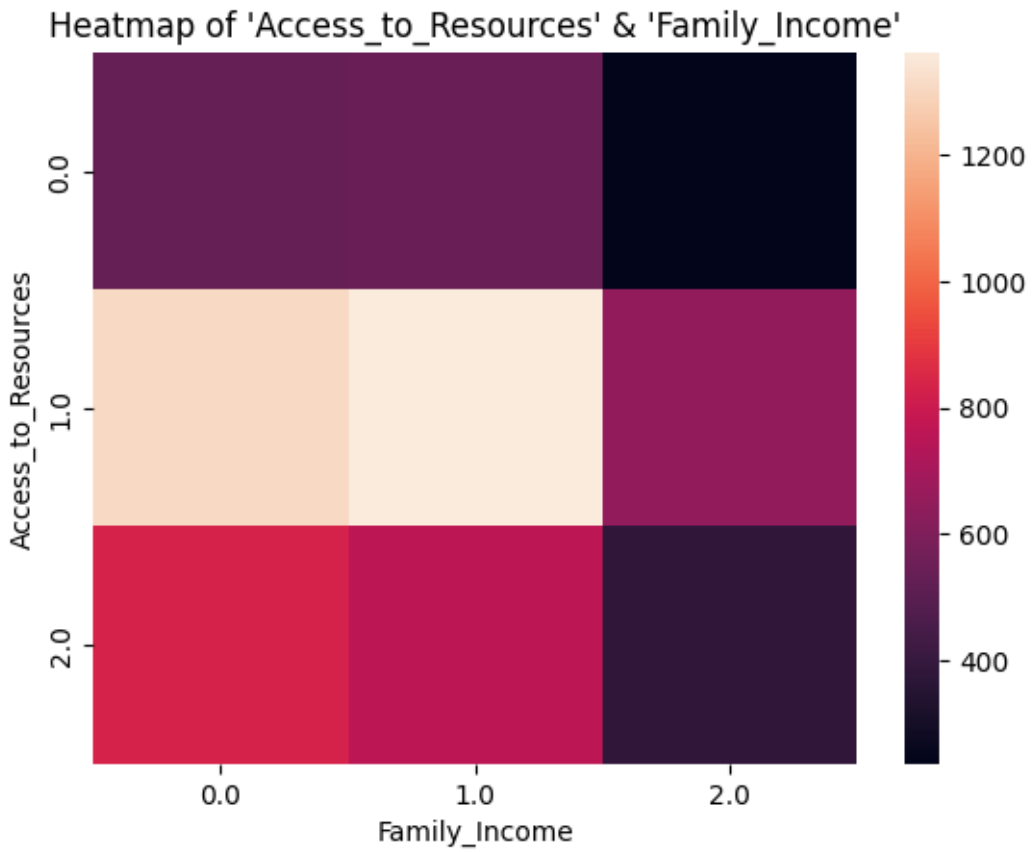
[139]:
```
cross_tab=pd.crosstab(dataset1['Family_Income'],dataset1['School_Type'])
sns.heatmap(cross_tab)
plt.title("Heatmap of 'Family_Income' & 'School_Type'")
plt.show()
```



[140]:
```
cross_tab=pd.
  ↪crosstab(dataset1['Parental_Education_Level'],dataset1['Family_Income'])
sns.heatmap(cross_tab)
plt.title("Heatmap of 'Parental_Education_Level' & 'Family_Income'")
plt.show()
```



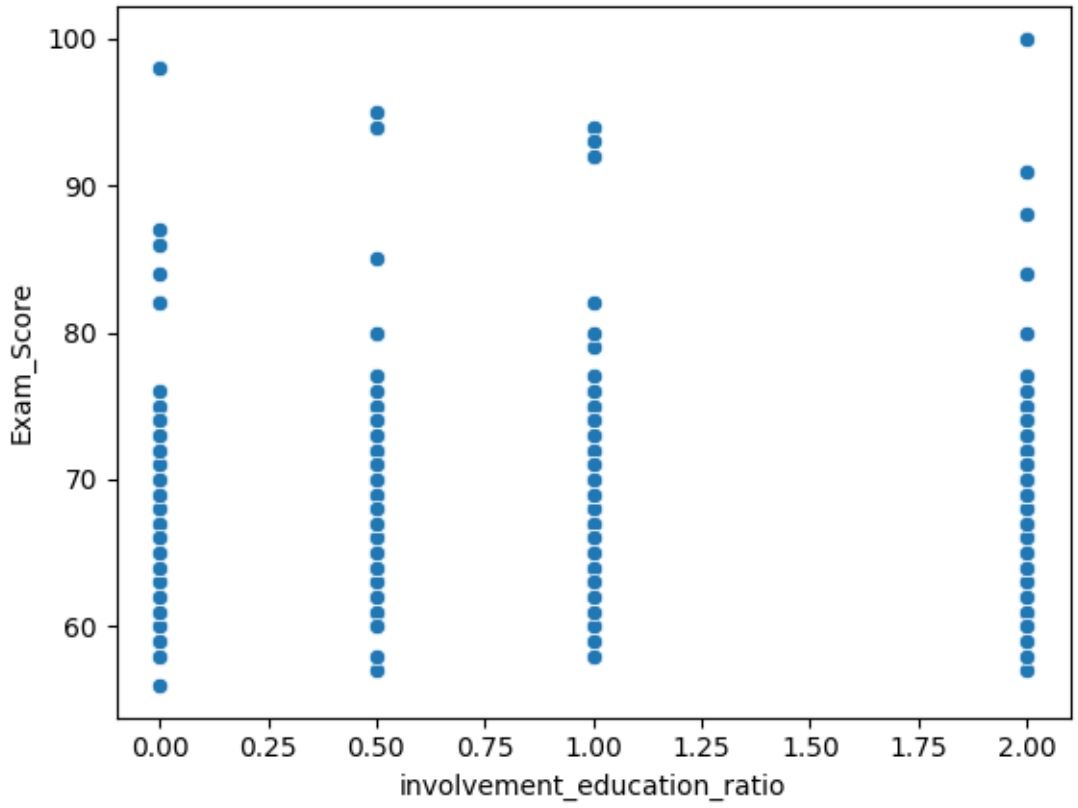
[141]:
```
cross_tab=pd.crosstab(dataset1['Access_to_Resources'],dataset1['Family_Income'])
sns.heatmap(cross_tab)
plt.title("Heatmap of 'Access_to_Resources' & 'Family_Income'")
plt.show()
```



[27]:
```
dataset1["involvement_education_ratio"] = dataset1["Parental_Involvement"] /␣
↪dataset1["Parental_Education_Level"]
```

[28]:
```
sns.scatterplot(x="involvement_education_ratio", y="Exam_Score", data=dataset1)
```

[28]:
```
<Axes: xlabel='involvement_education_ratio', ylabel='Exam_Score'>
```



[33]:
```
dataset1[["involvement_education_ratio","Exam_Score"]].corr()
```

[33]:

	involvement_education_ratio	Exam_Score
involvement_education_ratio	1.000000	0.104539
Exam_Score	0.104539	1.000000

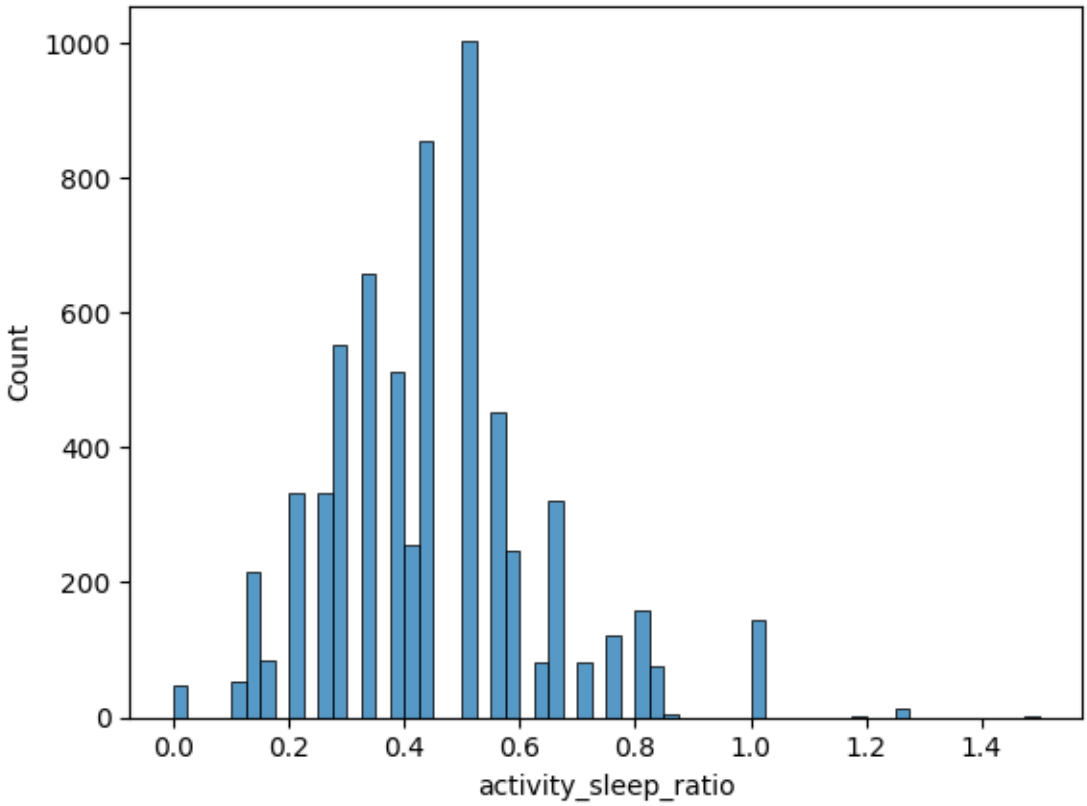
[37]:
```
dataset1[["Parental_Education_Level","Parental_Involvement"]].
↪corr(method="spearman")
```

[37]:

	Parental_Education_Level	Parental_Involvement
Parental_Education_Level	1.000000	-0.005899
Parental_Involvement	-0.005899	1.000000

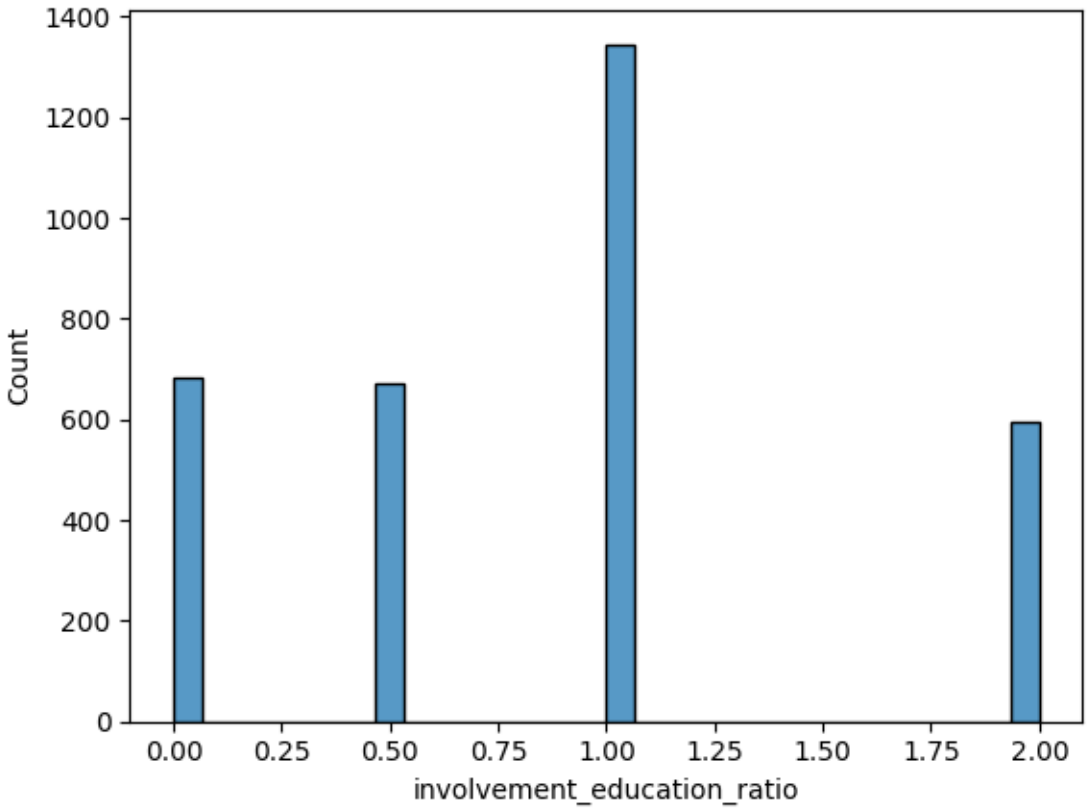
[34]:
```
sns.histplot(dataset1["activity_sleep_ratio"])
```

[34]:
```
<Axes: xlabel='activity_sleep_ratio', ylabel='Count'>
```



[35]:
```
sns.histplot(dataset1["involvement_education_ratio"])
```

[35]:
```
<Axes: xlabel='involvement_education_ratio', ylabel='Count'>
```


1400
1200
1000
800
600
400
200
0
Count
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
involvement_education_ratio