In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
import pandas as pd
df = pd.read_csv("E:/Student_Perdormance/StudentPerformanceFactors.csv")
print(df.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
None
```

In [11]:
```
import numpy as np
df[['Teacher_Quality', 'Parental_Education_Level', 'Distance_from_Home']] = \
df[['Teacher_Quality','Parental_Education_Level','Distance_from_Home']].fillna
```

Missing values in categorical features were handled by replacing them with "Unknown" to maintain data consistency.

In [12]:
```
df.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             6607 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    6607 non-null   object
 17  Distance_from_Home          6607 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
```

In [13]:
```
df.head(10)
```

Out[13]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extr
0	23	84	Low	High	
1	19	64	Low	Medium	
2	24	98	Medium	Medium	
3	29	89	Low	Medium	
4	19	92	Medium	Medium	
5	19	88	Medium	Medium	
6	29	84	Medium	Low	
7	25	78	Low	High	
8	17	94	Medium	High	
9	23	98	Medium	Medium	

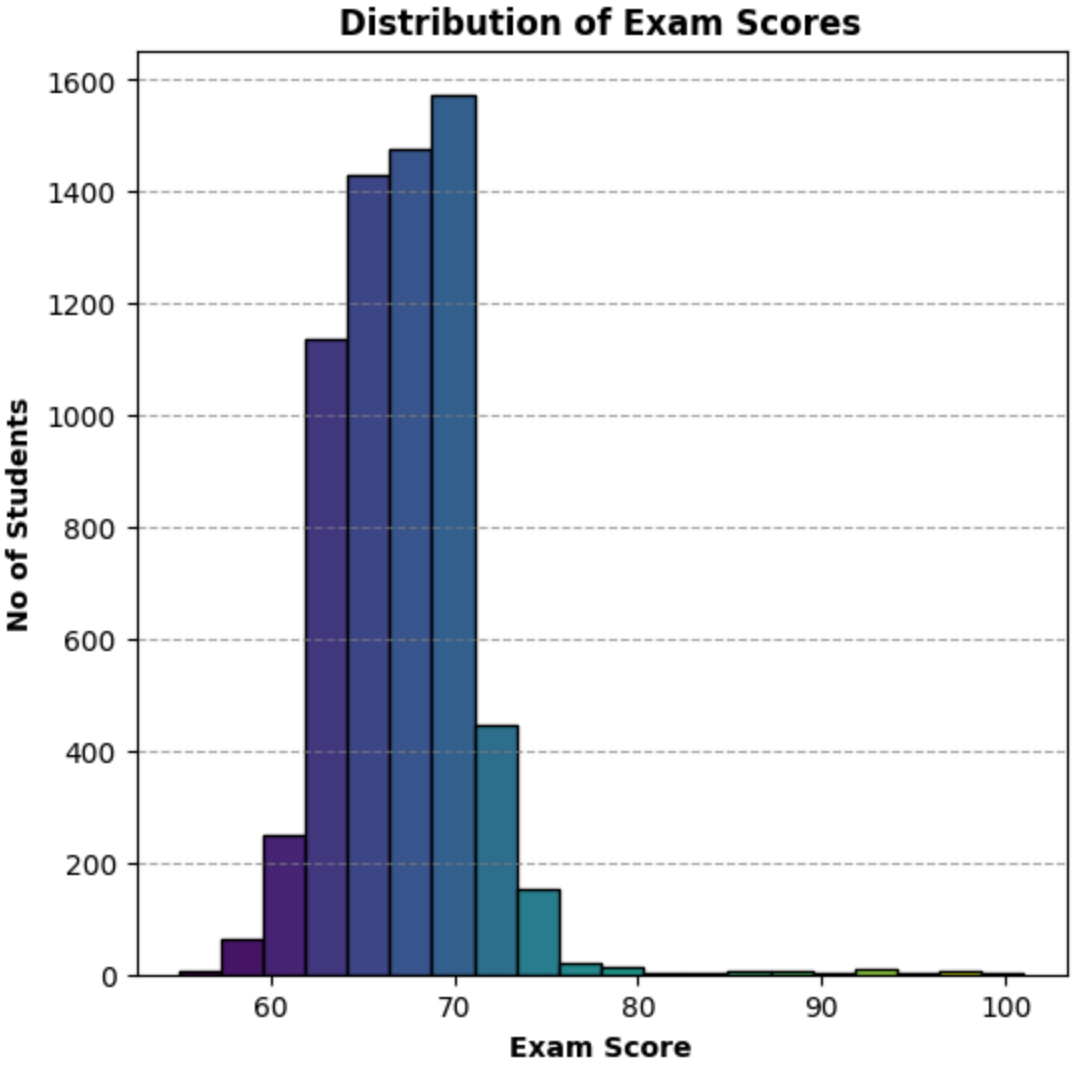
In [14]:
```
df.describe()
```

Out[14]:

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores	Tutoring_Sess
count	6607.000000	6607.000000	6607.00000	6607.000000	6607.00
mean	19.975329	79.977448	7.02906	75.070531	1.49
std	5.990594	11.547475	1.46812	14.399784	1.23
min	1.000000	60.000000	4.00000	50.000000	0.00
25%	16.000000	70.000000	6.00000	63.000000	1.00
50%	20.000000	80.000000	7.00000	75.000000	1.00
75%	24.000000	90.000000	8.00000	88.000000	2.00
max	44.000000	100.000000	10.00000	100.000000	8.00

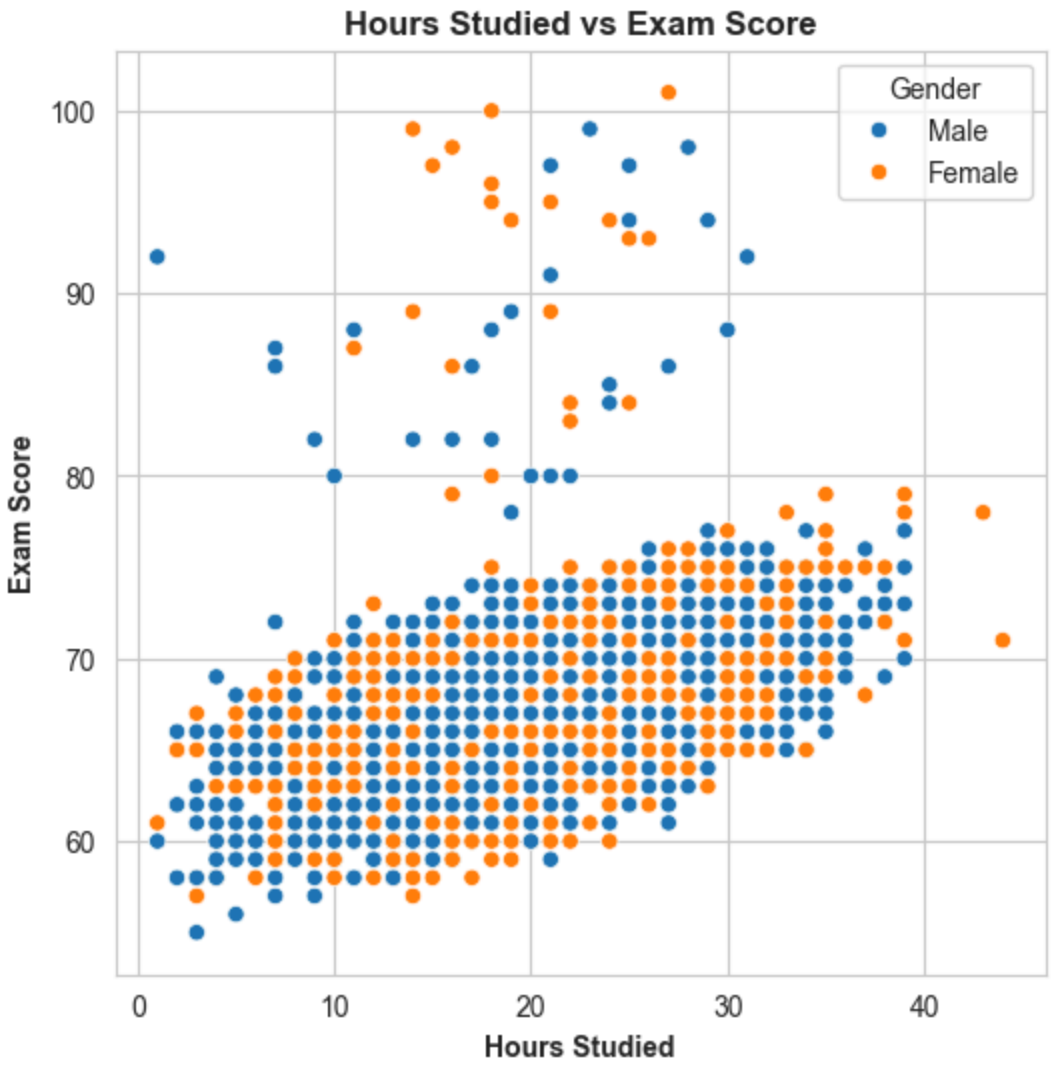
In [31]:

```
import matplotlib.cm as cm
plt.figure(figsize = (6,6))
n, bins, patches = plt.hist(df["Exam_Score"], bins = 20, edgecolor = "black")
colors = cm.viridis(np.linspace(0,1,len(patches)))
for patch, color in zip(patches, colors):
    patch.set_facecolor(color)
plt.xlabel("Exam Score", fontweight = "bold")
plt.ylabel("No of Students", fontweight = "bold")
plt.title("Distribution of Exam Scores", fontweight = "bold")
plt.grid(color = 'grey', axis = 'y', linestyle ="--", alpha = 0.7)
plt.show()
```



Most students’ exam scores are concentrated in the 60–70 range, indicating a common performance level.

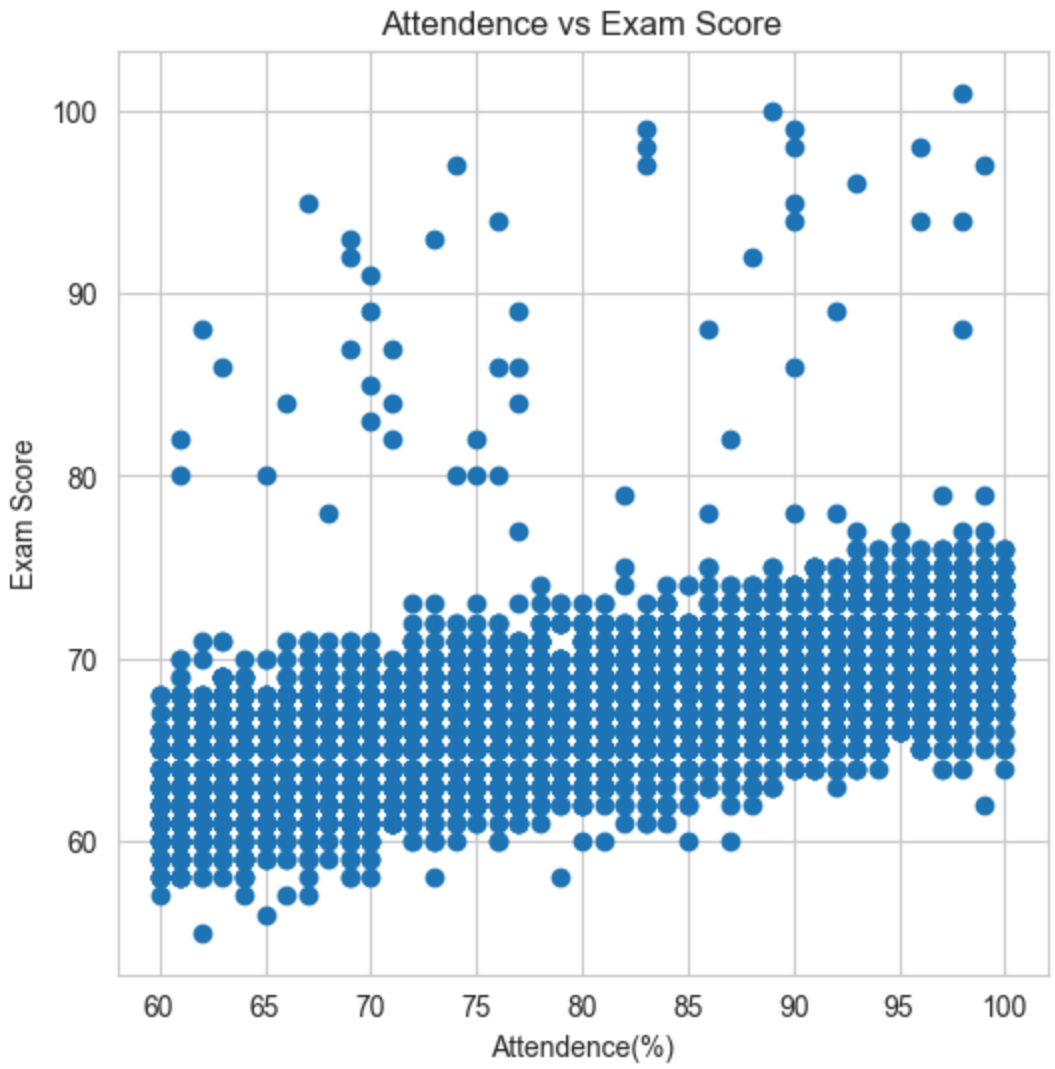
In [47]:
```
plt.figure(figsize = (6,6))
sns.scatterplot(data = df, x = "Hours_Studied", y = "Exam_Score", hue = 'Gende
plt.xlabel("Hours Studied", fontweight = "bold")
plt.ylabel("Exam Score", fontweight = "bold")
plt.title("Hours Studied vs Exam Score", fontweight = "bold")
plt.show()
```



The visualization reveals a positive correlation between study hours and exam scores, showing consistent performance trends across both male and female student demographics.

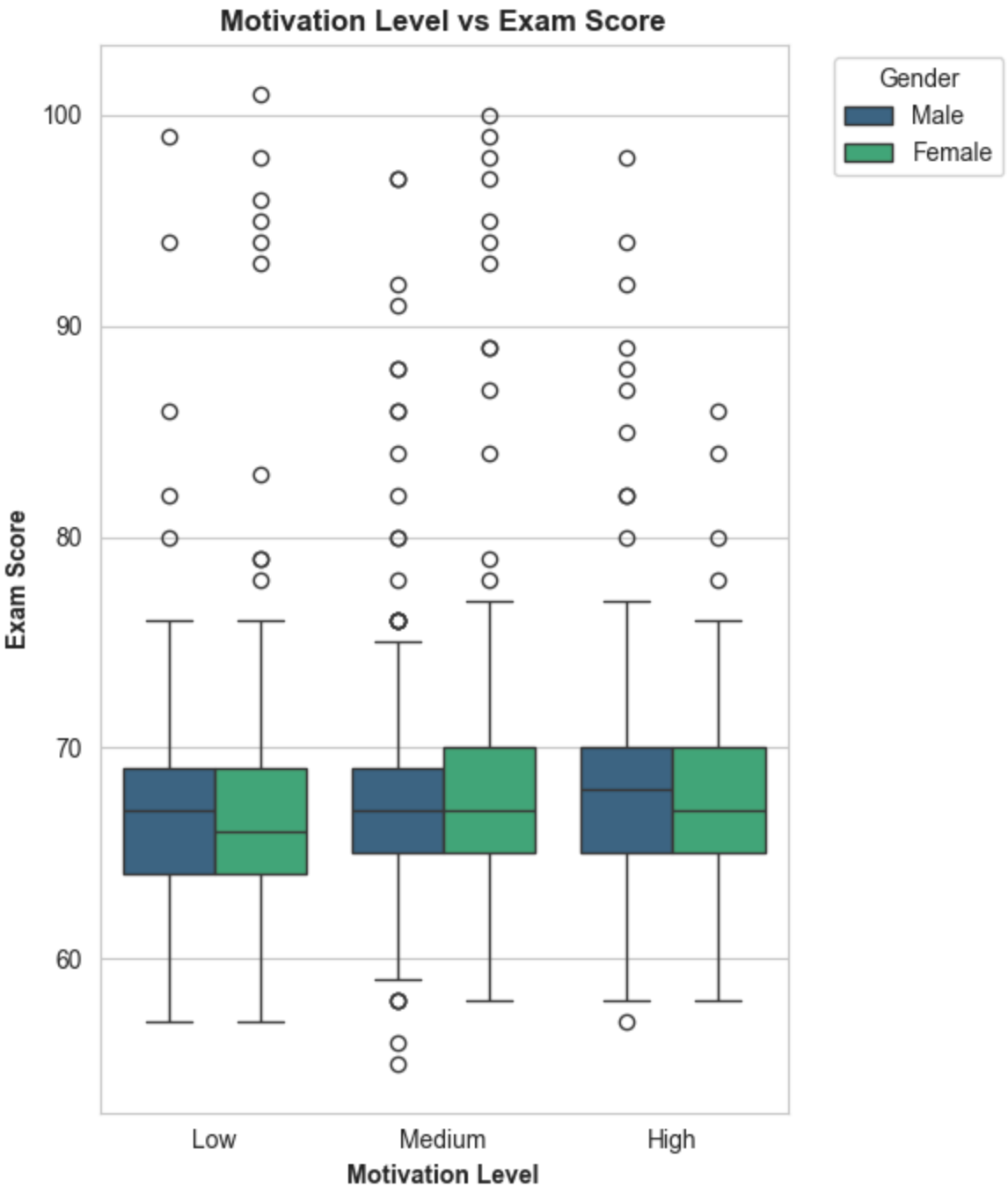
In [103…

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize = (6,6))
plt.scatter(df["Attendance"], df["Exam_Score"])
plt.xlabel("Attendence(%)")
plt.ylabel("Exam Score")
plt.title("Attendence vs Exam Score")
plt.show()
```

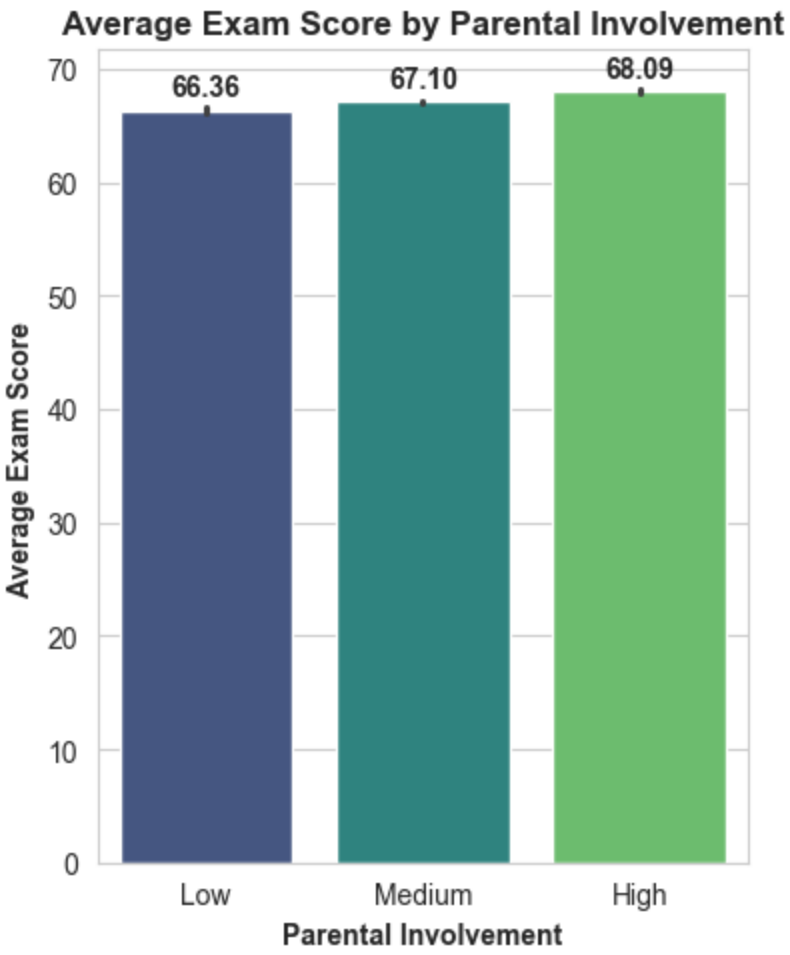


The scatter plot reveals a strong positive correlation between student attendance and exam results, indicating that consistent classroom presence is a key indicator of academic success.

In [73]:
```
plt.figure(figsize = (6,7))
sns.boxplot(data = df, x = "Motivation_Level", y = "Exam_Score", palette = "vi
plt.title("Motivation Level vs Exam Score", fontweight = "bold")
plt.xlabel("Motivation Level", fontweight = "bold")
plt.ylabel("Exam Score", fontweight = "bold")
plt.legend(title="Gender", bbox_to_anchor = (1.05,1), loc = "upper left")
plt.tight_layout()
plt.show()
```



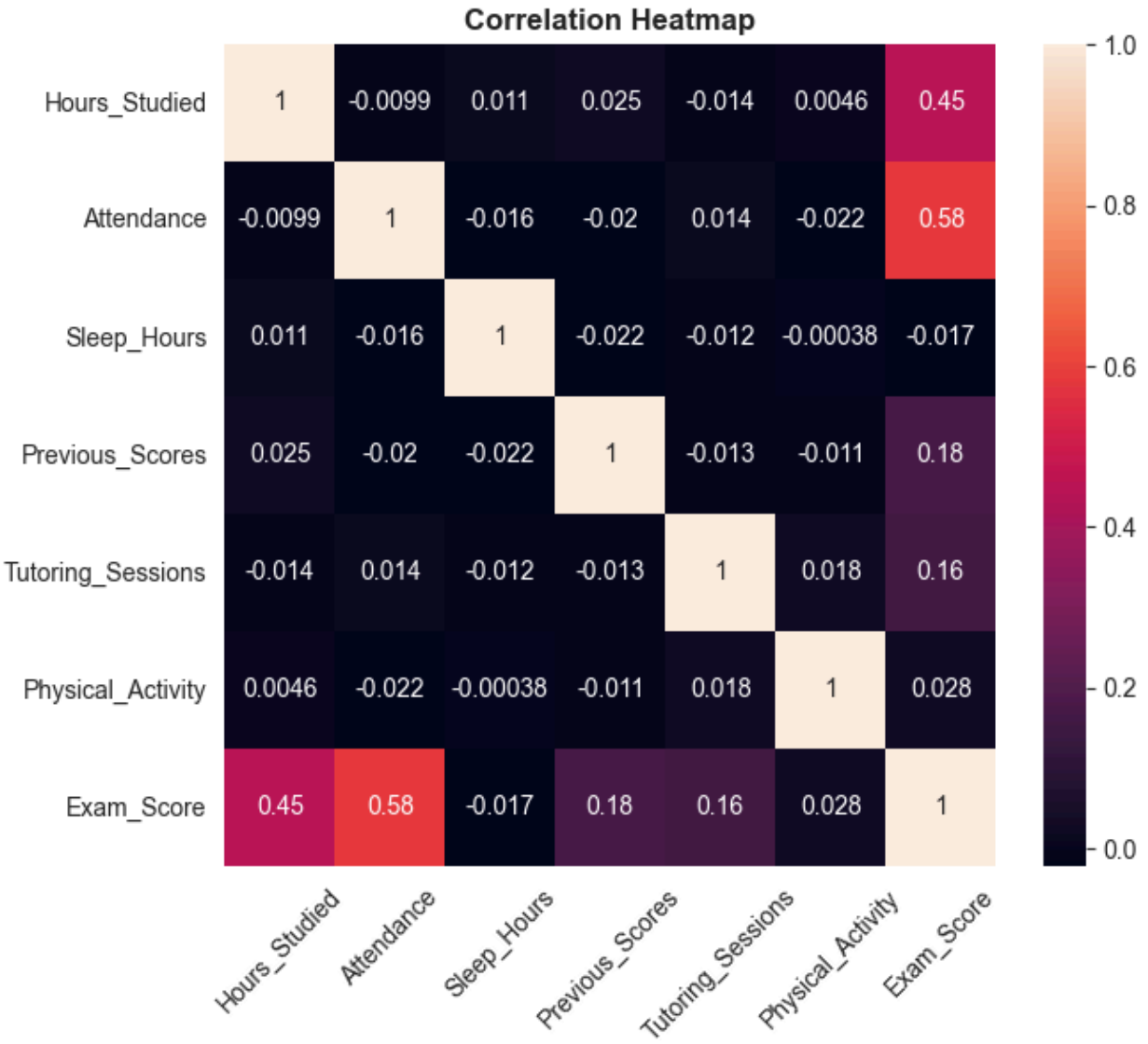
In [95]:
```
avg_scores = df.groupby('Parental_Involvement')['Exam_Score'].mean()
plt.figure(figsize = (4,5))
bx = sns.barplot(data = df, x = 'Parental_Involvement', y = 'Exam_Score', hue
bx.bar_label(bx.containers[0], fmt = '%.2f', padding = 3, fontweight = 'bold')
bx.bar_label(bx.containers[1], fmt = '%.2f', padding = 3, fontweight = 'bold')
bx.bar_label(bx.containers[2], fmt = '%.2f', padding = 3, fontweight = 'bold')
plt.xlabel("Parental Involvement", fontweight = "bold")
plt.ylabel("Average Exam Score",fontweight = "bold")
plt.title("Average Exam Score by Parental Involvement", fontweight = "bold")
plt.legend([],[], frameon=False)
plt.tight_layout()
plt.show()
```



Students with higher parental involvement tend to achieve higher average exam scores.

In [104…

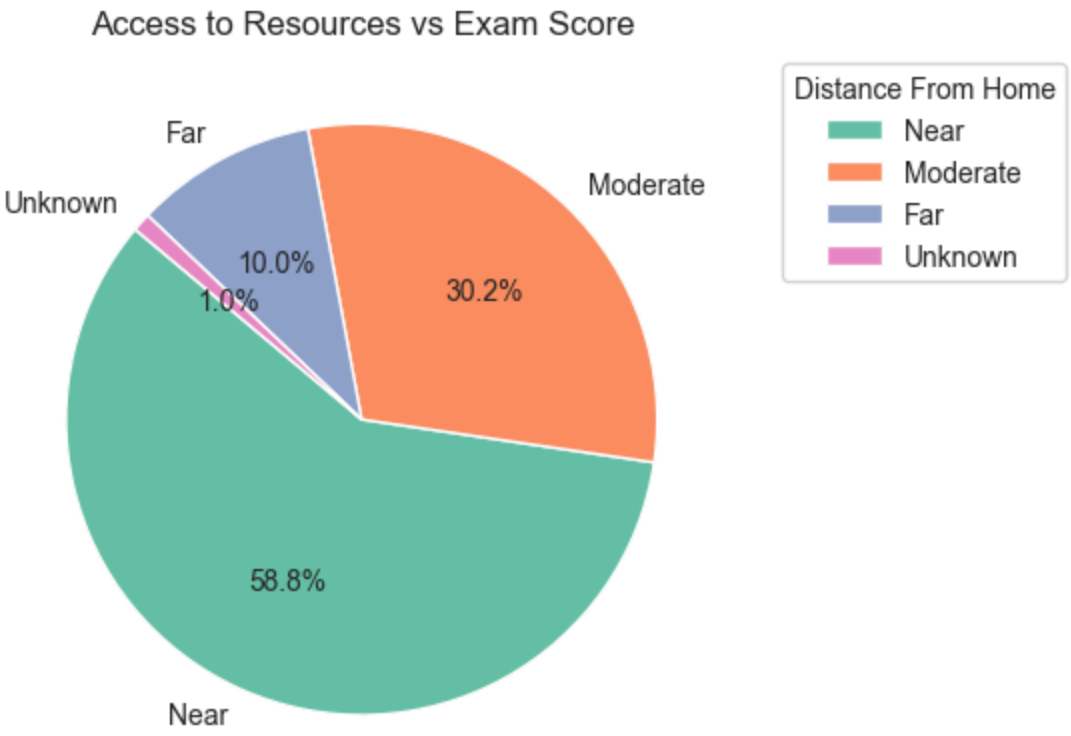
```
plt.figure(figsize = (7,6))
sns.heatmap(df.corr(numeric_only = True), annot = True)
plt.title("Correlation Heatmap", fontweight = "bold")
plt.xticks(rotation = 45)
plt.show()
```



Exam Score shows a positive correlation with Hours Studied and Previous Scores, indicating these factors strongly influence performance.

In [111…

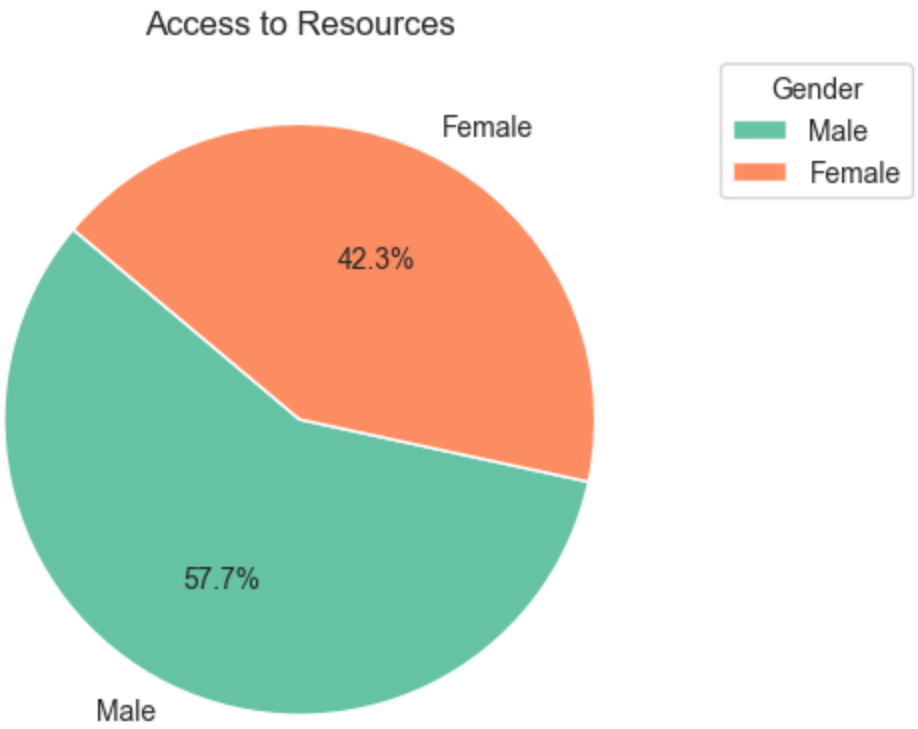
```
plt.figure()
df['Distance_from_Home'].value_counts().plot.pie(autopct='%1.1f%%', colors=sns
plt.ylabel(''); plt.legend(title="Distance From Home", bbox_to_anchor=(1.05, 1
plt.title("Access to Resources vs Exam Score")
plt.legend(title = "Distance From Home", bbox_to_anchor = (1.05,1), loc = "upp
plt.show()
```



Even with high resources, students living further away may show more score variance or lower medians

In [113…

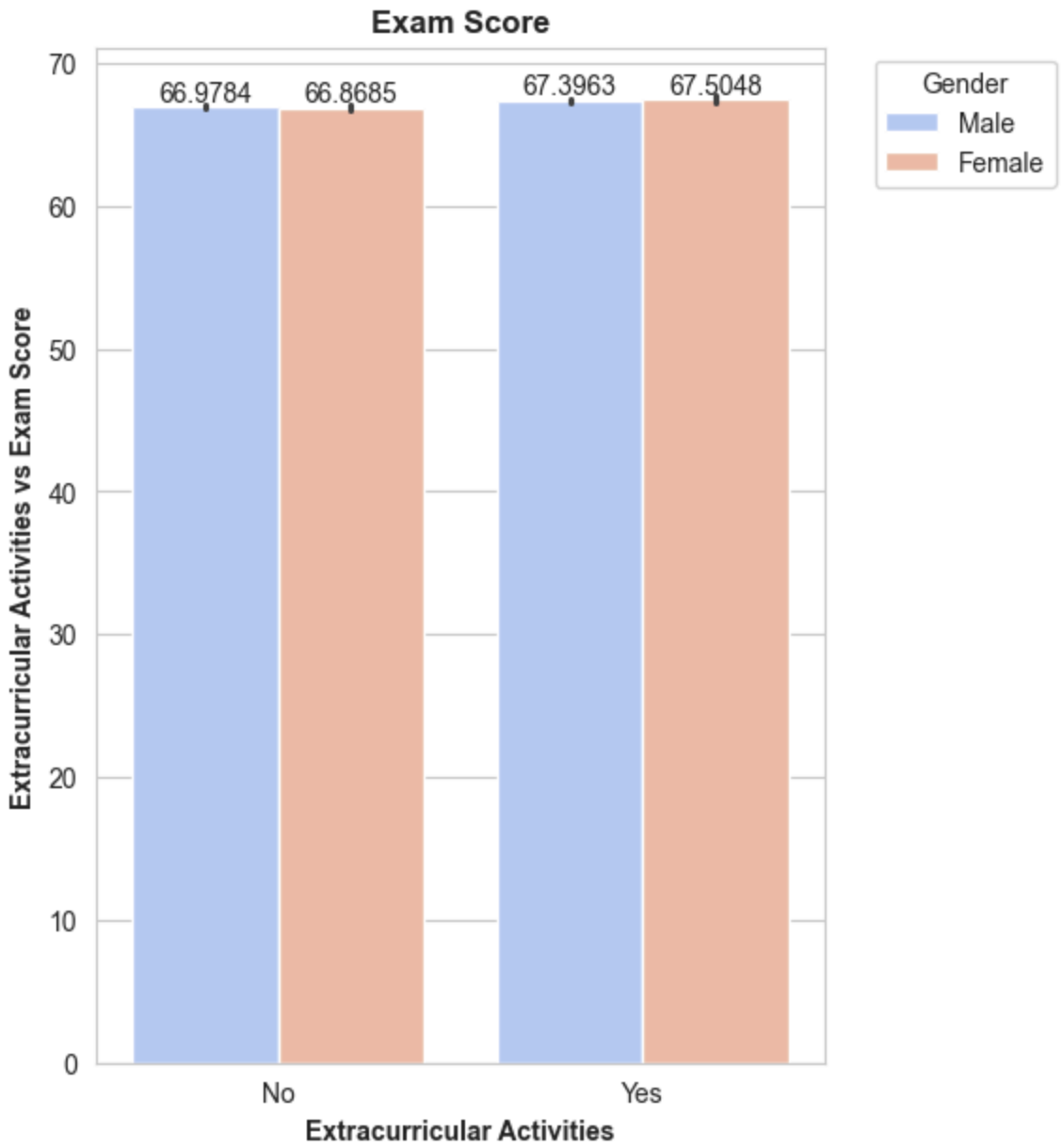
```
plt.figure()
df['Gender'].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_palet
plt.ylabel(''); plt.legend(title="Distance From Home", bbox_to_anchor=(1.05, 1
plt.title("Access to Resources")
plt.legend(title = "Gender", bbox_to_anchor = (1.05,1), loc = "upper left")
plt.show()
```



The gender distribution within the dataset is nearly balanced, ensuring that subsequent analyses of resource access and exam performance are representative across both male and female student populations

In [119…

```
plt.figure(figsize = (5,7))
cx = sns.barplot(x="Extracurricular_Activities", y="Exam_Score", data=df, pale
cx.bar_label(cx.containers[0])
cx.bar_label(cx.containers[1])
plt.xlabel("Extracurricular Activities", fontweight = "bold")
plt.ylabel("Extracurricular Activities vs Exam Score",fontweight = "bold")
plt.title("Exam Score", fontweight = "bold")
plt.legend(title = "Gender", bbox_to_anchor = (1.05,1), loc = "upper left")
plt.show()
```

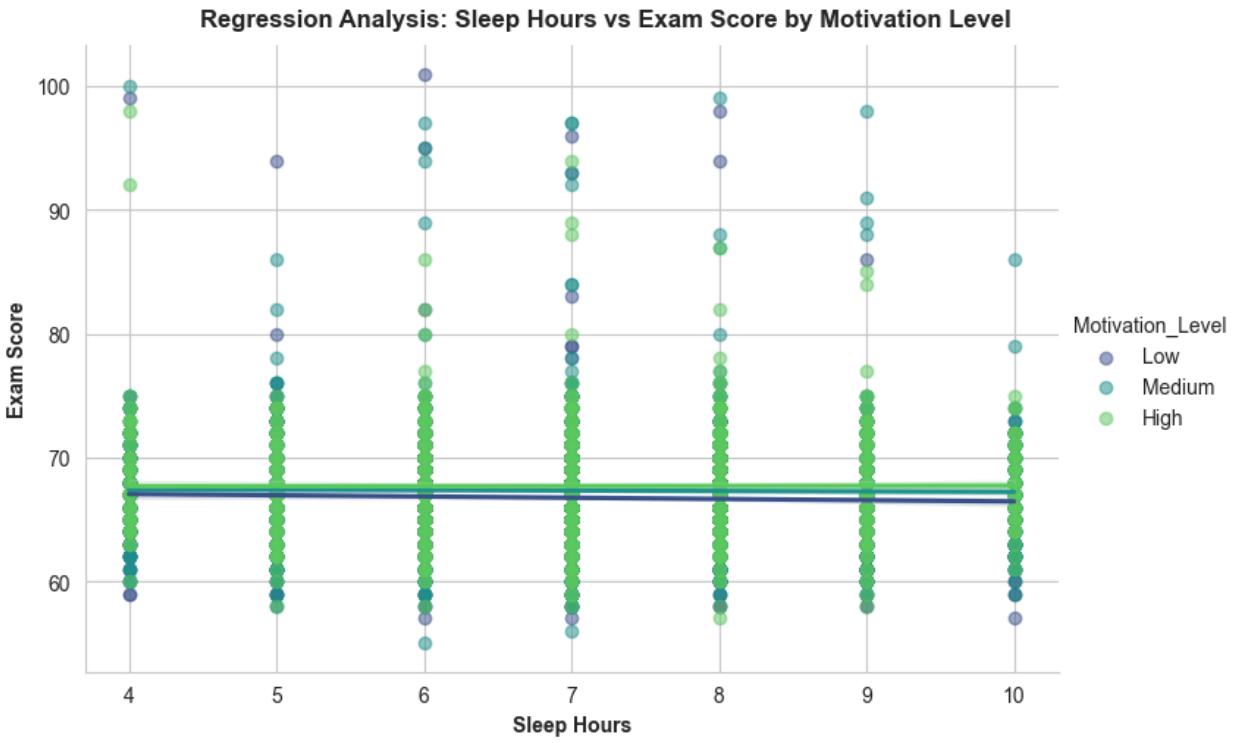


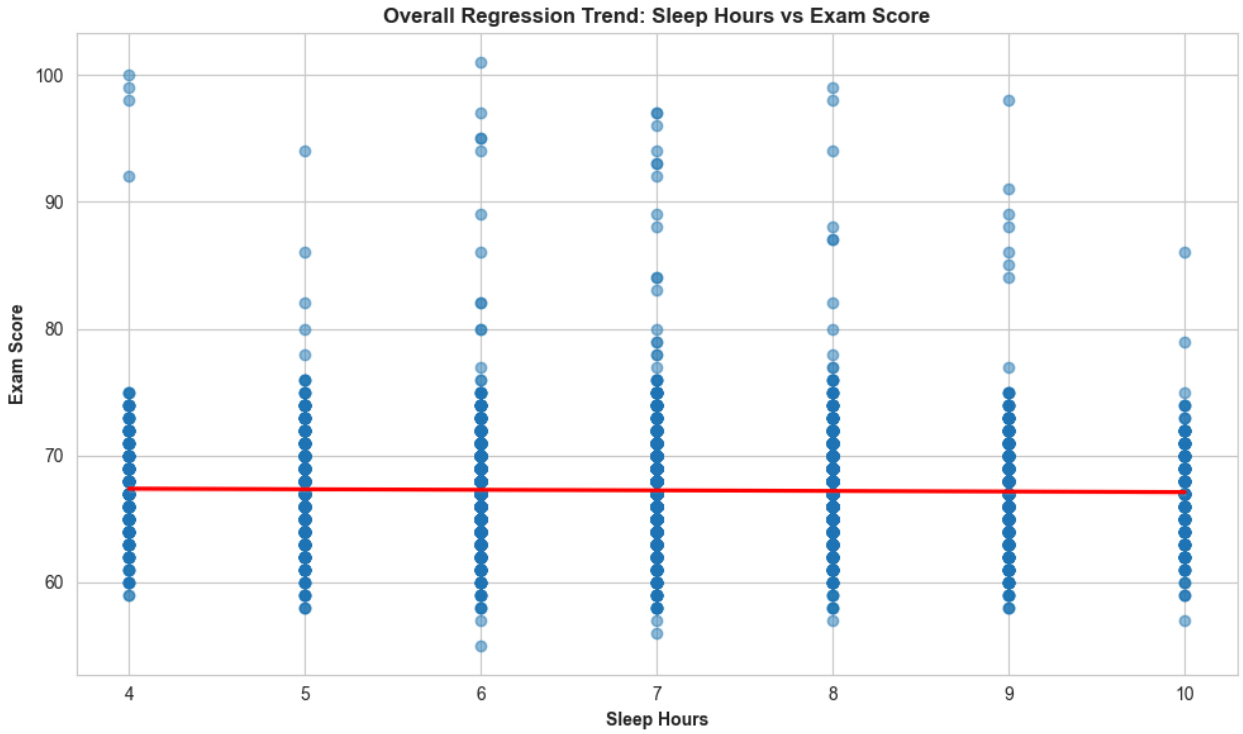
This visualization highlights that students engaged in extracurricular activities generally maintain competitive exam scores, with direct data labels revealing a marginal performance difference between genders across both participation levels

In [136…

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
g = sns.lmplot(
    x="Sleep_Hours",
    y="Exam_Score",
    hue="Motivation_Level",
    data=df,
    palette="viridis",
    aspect=1.5,
    scatter_kws={'alpha':0.5}
)

g.set_axis_labels("Sleep Hours", "Exam Score", fontweight="bold")
g.fig.suptitle("Regression Analysis: Sleep Hours vs Exam Score by Motivation L

plt.show()
plt.figure(figsize=(10, 6))
sns.regplot(
    x="Sleep_Hours",
    y="Exam_Score",
    data=df,
    scatter_kws={'alpha':0.5},
    line_kws={'color':'red'}
)

plt.xlabel("Sleep Hours", fontweight="bold")
plt.ylabel("Exam Score", fontweight="bold")
plt.title("Overall Regression Trend: Sleep Hours vs Exam Score", fontweight="b
plt.tight_layout()
plt.show()
```





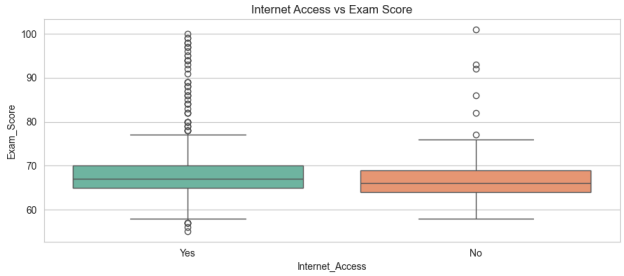
The data supports the hypothesis that sleep hygiene is positively correlated with cognitive performance. To maximize exam outcomes, ensuring adequate sleep is just as vital as study hours.

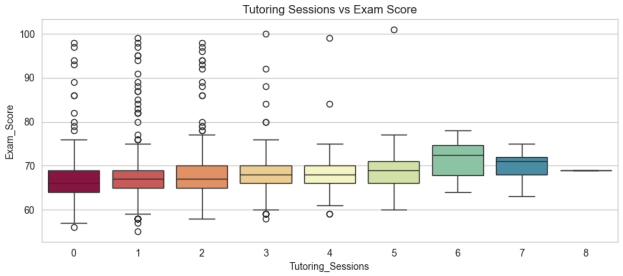
In [138…

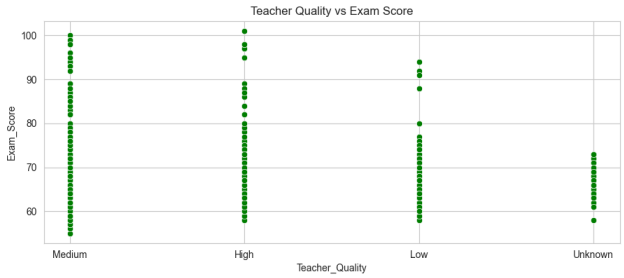
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(18, 16))

# c) Internet Access vs Exam Score
plt.subplot(4, 2, 1)
sns.boxplot(
    x="Internet_Access",
    y="Exam_Score",
    hue="Internet_Access",
    data=df,
    palette="Set2",
    legend=False
)
plt.title("Internet Access vs Exam Score")

# d) Tutoring Sessions vs Exam Score
```

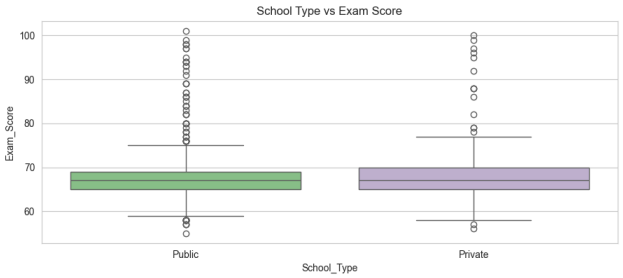
```
plt.subplot(4, 2, 2)
sns.boxplot(
    x="Tutoring_Sessions",
    y="Exam_Score",
    hue="Tutoring_Sessions",
    data=df,
    palette="Spectral",
    legend=False
)
plt.title("Tutoring Sessions vs Exam Score")

# e) Teacher Quality vs Exam Score
plt.subplot(4, 2, 3)
sns.scatterplot(
    x="Teacher_Quality",
    y="Exam_Score",
    data=df,
    color="green"
)
plt.title("Teacher Quality vs Exam Score")

# f) School Type Comparison
plt.subplot(4, 2, 4)
sns.boxplot(
    x="School_Type",
    y="Exam_Score",
    hue="School_Type",
    data=df,
    palette="Accent",
    legend=False
)
plt.title("School Type vs Exam Score")

# g) Peer Influence vs Exam Score
plt.subplot(4, 2, 5)
sns.barplot(
    x="Peer_Influence",
    y="Exam_Score",
    hue="Peer_Influence",
    data=df,
    palette="rocket",
    legend=False
)
plt.title("Peer Influence vs Exam Score")

# h) Physical Activity vs Exam Score
plt.subplot(4, 2, 6)
sns.lineplot(
    x="Physical_Activity",
    y="Exam_Score",
    data=df,
    marker="o",
    color="purple"
```

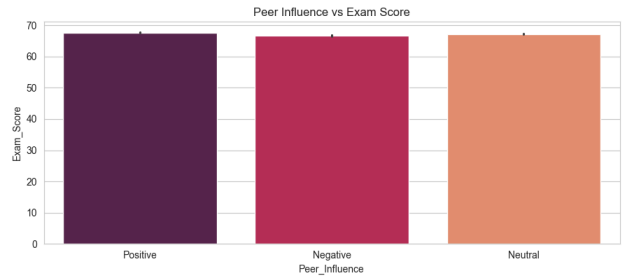
```
)
plt.title("Physical Activity vs Exam Score")

# i) Parental Education Level Impact
plt.subplot(4, 2, 7)
sns.boxplot(
    x="Parental_Education_Level",
    y="Exam_Score",
    hue="Parental_Education_Level",
    data=df,
    palette="cubehelix",
    legend=False
)
plt.title("Parental Education Level vs Exam Score")

# j) Distance from Home vs Exam Score
plt.subplot(4, 2, 8)
sns.scatterplot(
    x="Distance_from_Home",
    y="Exam_Score",
    data=df,
    color="brown"
)
plt.title("Distance from Home vs Exam Score")

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

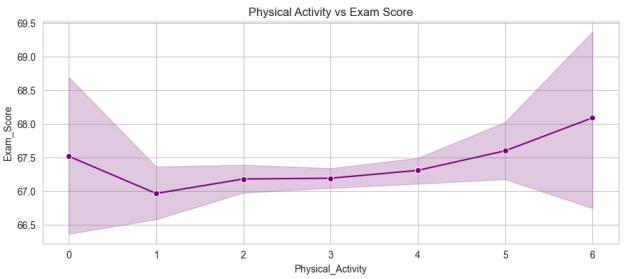

Internet Access vs Exam Score
Exam_Score
Yes
No
Internet_Access

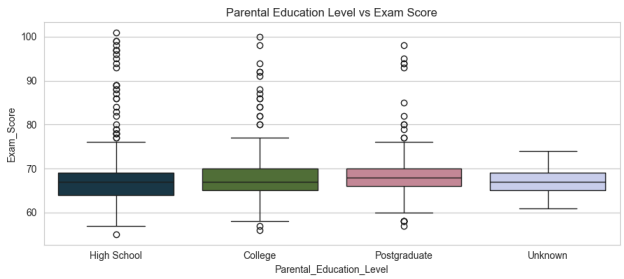

Tutoring Sessions vs Exam Score
Exam_Score
Tutoring_Sessions

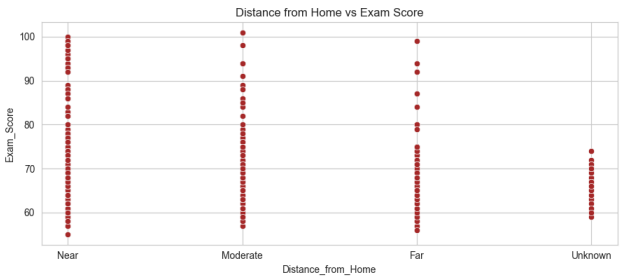

Teacher Quality vs Exam Score
Exam_Score
Medium
High
Low
Unknown
Teacher_Quality


School Type vs Exam Score
Exam_Score
Public
Private
School_Type


Peer Influence vs Exam Score
Exam_Score
Positive
Negative
Neutral
Peer_Influence


Physical Activity vs Exam Score
Exam_Score
Physical_Activity


Parental Education Level vs Exam Score
Exam_Score
High School
College
Postgraduate
Unknown
Parental_Education_Level


Distance from Home vs Exam Score
Exam_Score
Near
Moderate
Far
Unknown
Distance_from_Home

Internet Access vs Exam Score: Students with internet access show a higher median exam score and less score variability compared to those without access.

Tutoring Sessions vs Exam Score: Exam scores generally improve as the number of tutoring sessions increases, indicating a positive academic impact.

Teacher Quality vs Exam Score: A positive trend suggests that higher teacher quality is associated with better student exam performance.

School Type vs Exam Score: Private school students tend to have a slightly higher median exam score than public school students.

Peer Influence vs Exam Score: Positive peer influence corresponds to higher average exam scores compared to neutral or negative influence.

Physical Activity vs Exam Score: Moderate physical activity levels are associated with stable or slightly improved exam scores.

Parental Education Level vs Exam Score: Higher parental education levels are linked to improved median exam scores among students.

Distance from Home vs Exam Score: Exam scores show no strong linear relationship with distance from home, indicating minimal impact.

In [ ]: