

```
1 !gdown --id 11hnbngANUVuUaV385XPnyu1q02LgSlg9
```

```
🔄 /usr/local/lib/python3.10/dist-packages/gdown/__main__.py:132: FutureWarning: Option
  warnings.warn(
  Downloading...
  From: https://drive.google.com/uc?id=11hnbngANUVuUaV385XPnyu1q02LgSlg9
  To: /content/StudentPerformanceFactors.csv
  100% 642k/642k [00:00<00:00, 7.70MB/s]
```

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import numpy as np
5 from mpl_toolkits.mplot3d import Axes3D
6 import plotly.express as px
7 from dash import Dash, dcc, html, Input, Output
8 import dash_bootstrap_components as dbc
9
10 # Assuming the data is in a CSV file named 'student_performance.csv'
11 data = pd.read_csv('/content/StudentPerformanceFactors.csv')
12
13 # Convert categorical data to category type for better handling in visualizations
14 for column in data.select_dtypes(include=['object']).columns:
15     data[column] = data[column].astype('category')
16
17 # Descriptive Statistics
18 print(data.describe())
19
20 def plot_univariate():
21     plt.figure(figsize=(20, 20))
22
23     # Numerical Variables
24     for i, col in enumerate([col for col in data.columns if data[col].dtype in ['int64',
25         plt.subplot(5, 3, i)
26         sns.histplot(data[col], kde=True)
27         plt.title(f'Distribution of {col}')]
28
29     # Categorical Variables
30     # Reset the counter i for the second loop
31     for j, col in enumerate([col for col in data.columns if data[col].dtype == 'category'
32         plt.subplot(5, 3, j)
33         sns.countplot(x=col, data=data)
34         plt.title(f'Count of {col}')]
35         plt.xticks(rotation=45)
36
37     plt.tight_layout()
38     plt.show()
39
40 # Bivariate Analysis
```

```
41 def plot_bivariate():
42     # Scatter plot for Hours Studied vs Exam Score
43     plt.figure(figsize=(10, 6))
44     sns.scatterplot(x='Hours_Studied', y='Exam_Score', data=data)
45     plt.title('Hours Studied vs Exam Score')
46     plt.show()
47
48     # Box plot for Parental Involvement vs Exam Score
49     plt.figure(figsize=(10, 6))
50     sns.boxplot(x='Parental_Involvement', y='Exam_Score', data=data)
51     plt.title('Parental Involvement Impact on Exam Score')
52     plt.show()
53
54 # Multivariate Analysis
55 def plot_multivariate():
56     # Pair plot
57     sns.pairplot(data[['Exam_Score', 'Peer_Influence', 'Previous_Scores', 'Hours_Studied',
58     plt.show()
59
60     # 3D Scatter plot
61     fig = plt.figure(figsize=(10, 8))
62     ax = fig.add_subplot(111, projection='3d')
63     ax.scatter(data['Hours_Studied'], data['Sleep_Hours'], data['Exam_Score'])
64     ax.set_xlabel('Hours Studied')
65     ax.set_ylabel('Sleep Hours')
66     ax.set_zlabel('Exam Score')
67     plt.title('3D View of Study, Sleep, and Exam Score')
68     plt.show()
69
70 # Heatmap for correlation
71 def plot_heatmap():
72     plt.figure(figsize=(12, 10))
73
74     # Select only numerical features for correlation calculation
75     numerical_data = data.select_dtypes(include=['int64', 'float64'])
76
77     sns.heatmap(numerical_data.corr(), annot=True, cmap='coolwarm')
78     plt.title('Correlation Heatmap of Numeric Features')
79     plt.show()
80
81 # Interactive Dashboard
82 app = Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
83
84 app.layout = html.Div([
85     html.H1('Student Performance Dashboard'),
86     dcc.Dropdown(
87         id='xaxis-column',
88         options=[{'label': i, 'value': i} for i in ['Teacher_Quality', 'Parental_Educational_Involvement', 'Peer_Influence', 'Previous_Scores', 'Sleep_Hours', 'Hours_Studied']],
89         value='Teacher_Quality'
90     ),
91     dcc.Graph(id='indicator-graphic')
```

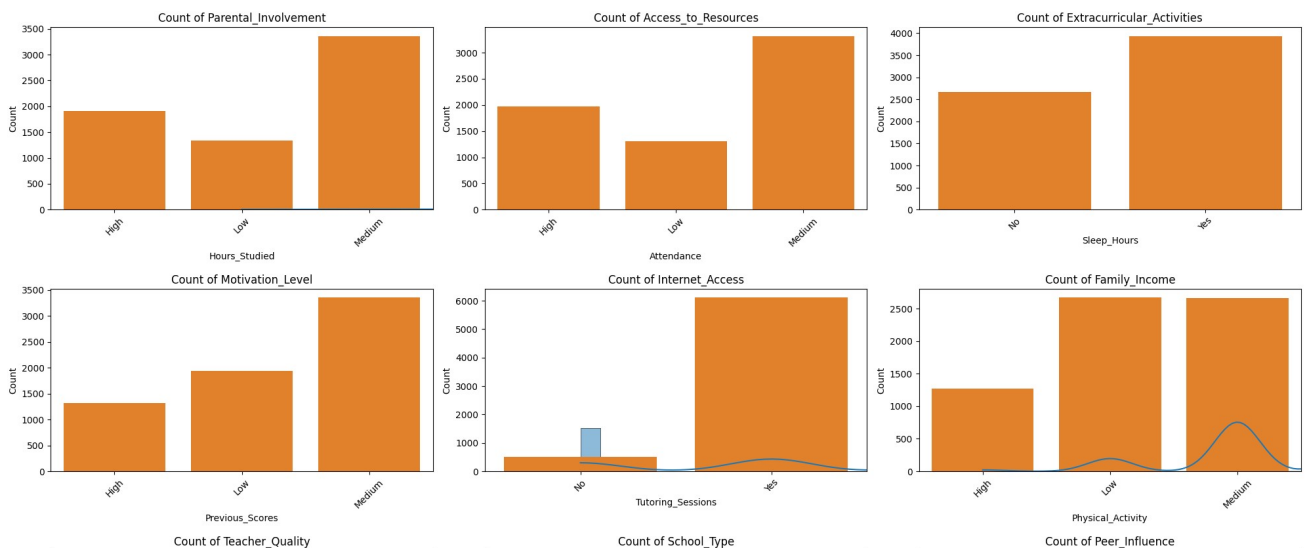
```

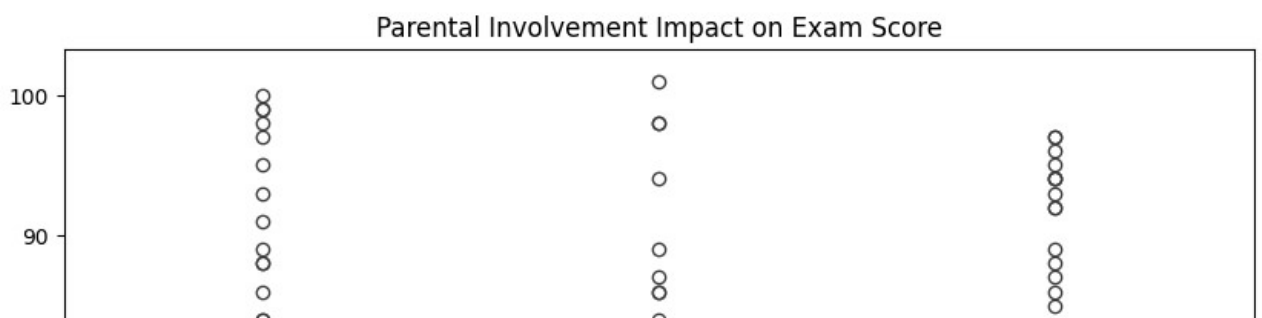
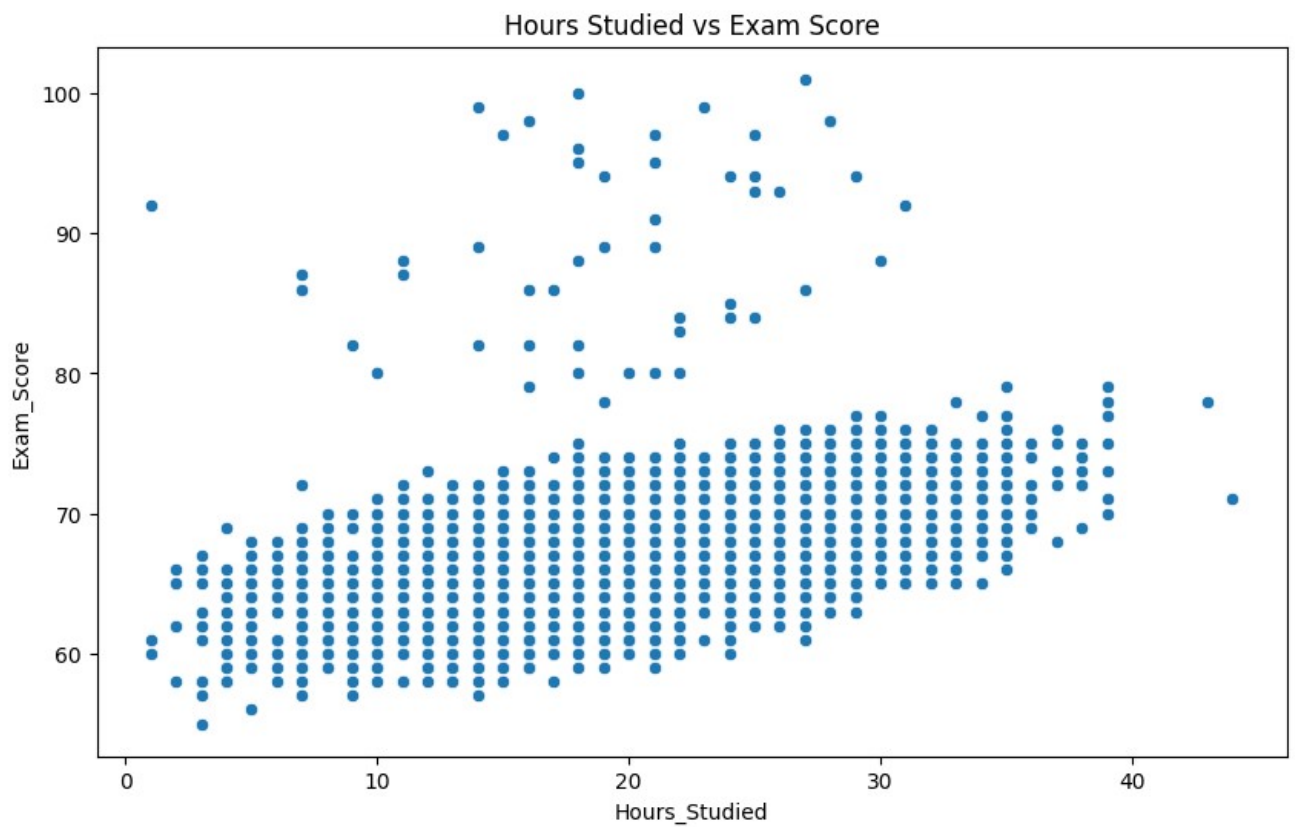
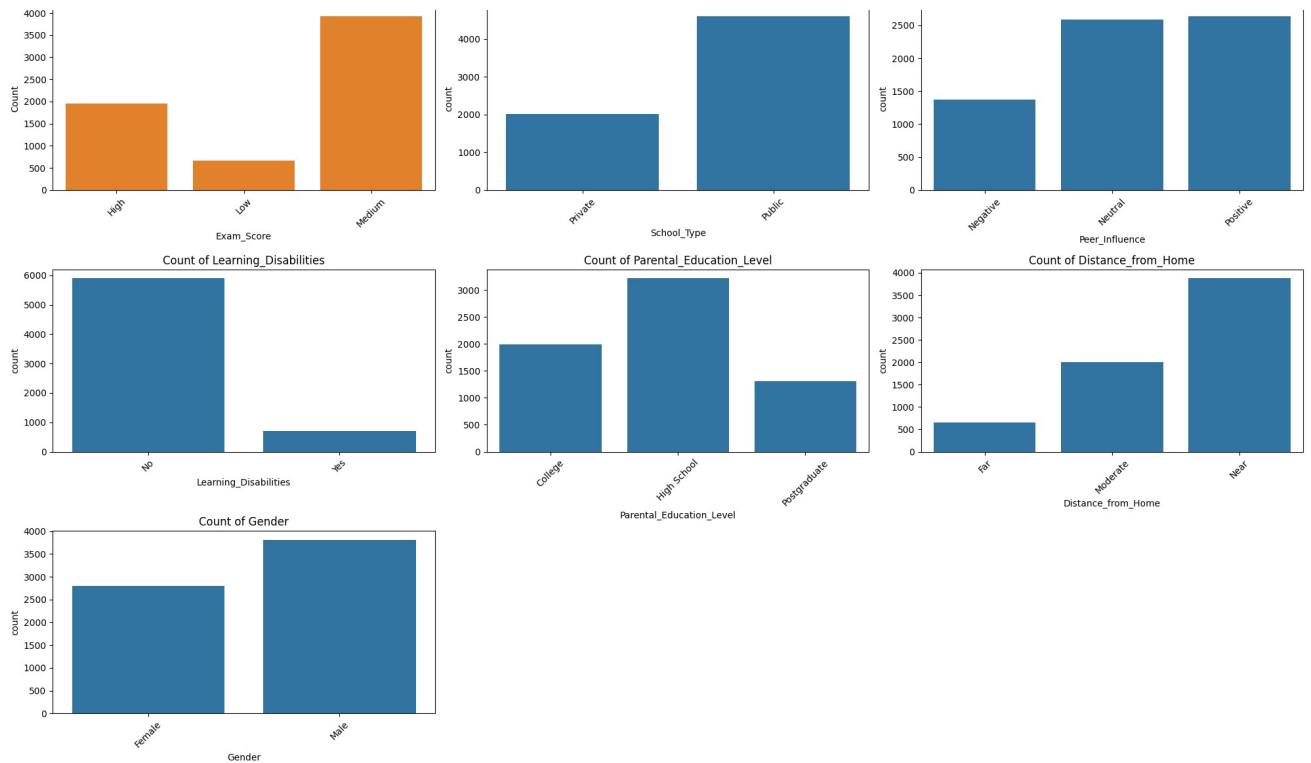
91     dcc.Graph(id= 'indicator-graphic' )
92 ])
93
94 @app.callback(
95     Output('indicator-graphic', 'figure'),
96     Input('xaxis-column', 'value'))
97 def update_graph(xaxis_column_name):
98     fig = px.box(data, x=xaxis_column_name, y="Exam_Score", color=xaxis_column_name,
99                 title=f"Exam Score Distribution by {xaxis_column_name}")
100     return fig
101
102 if __name__ == '__main__':
103     plot_univariate()
104     plot_bivariate()
105     plot_multivariate()
106     plot_heatmap()
107     # Comment out the next line if you don't want to run the dashboard
108     app.run_server(debug=True)

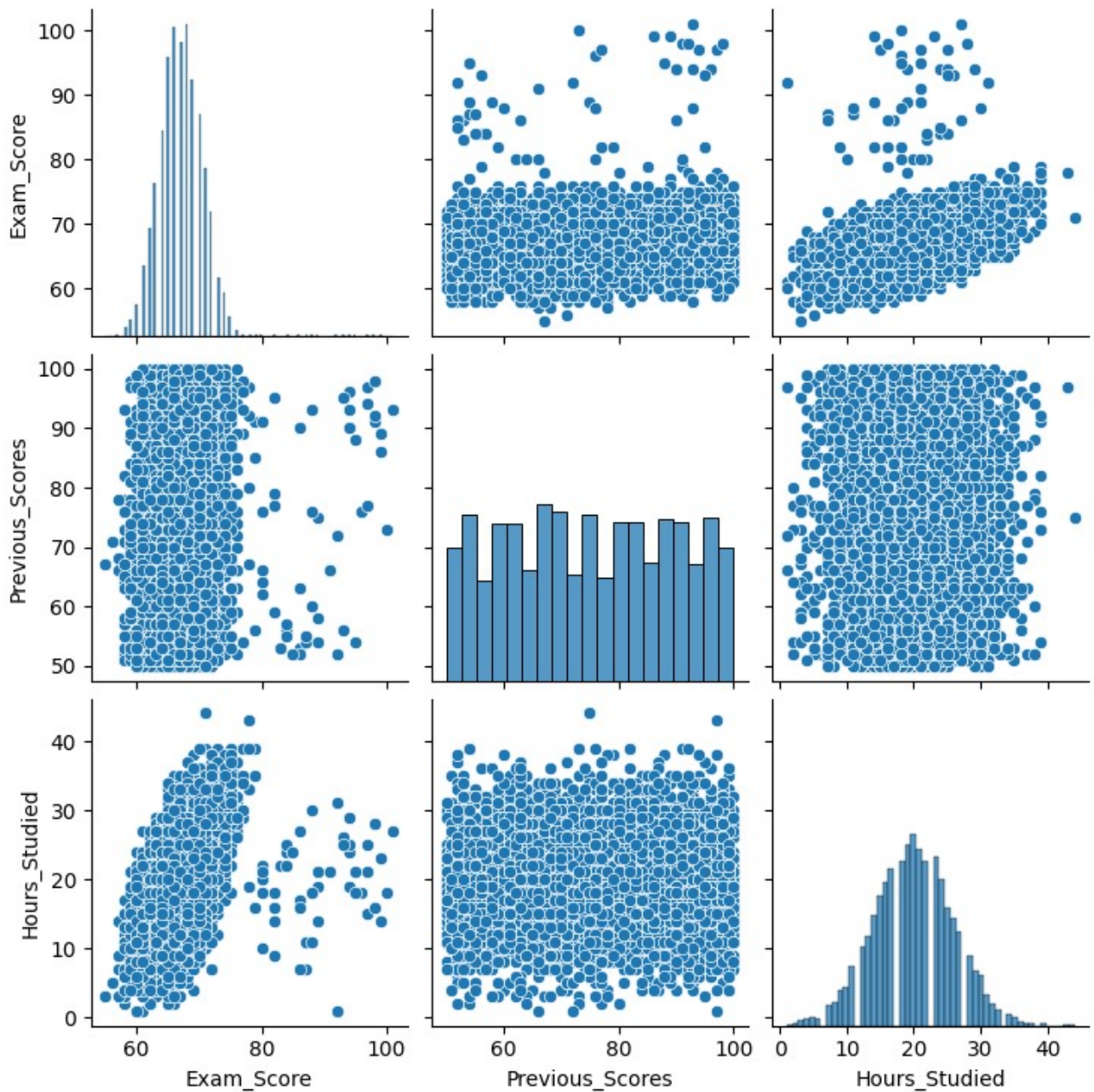
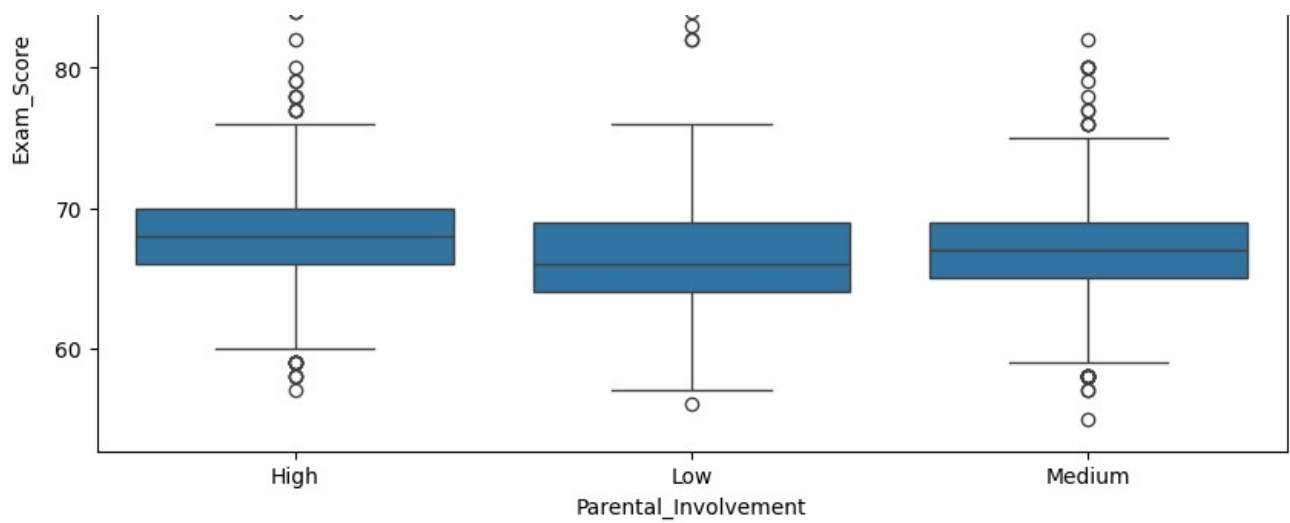
```

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores \
count	6607.000000	6607.000000	6607.000000	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

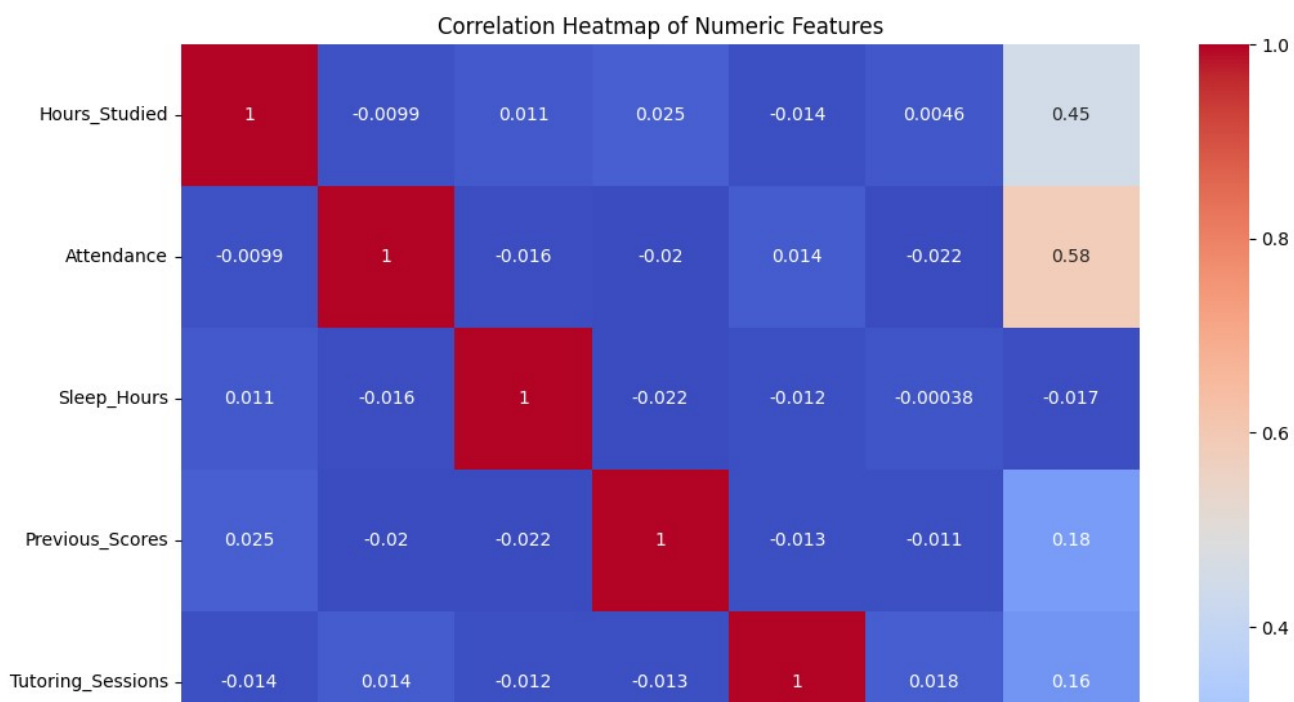
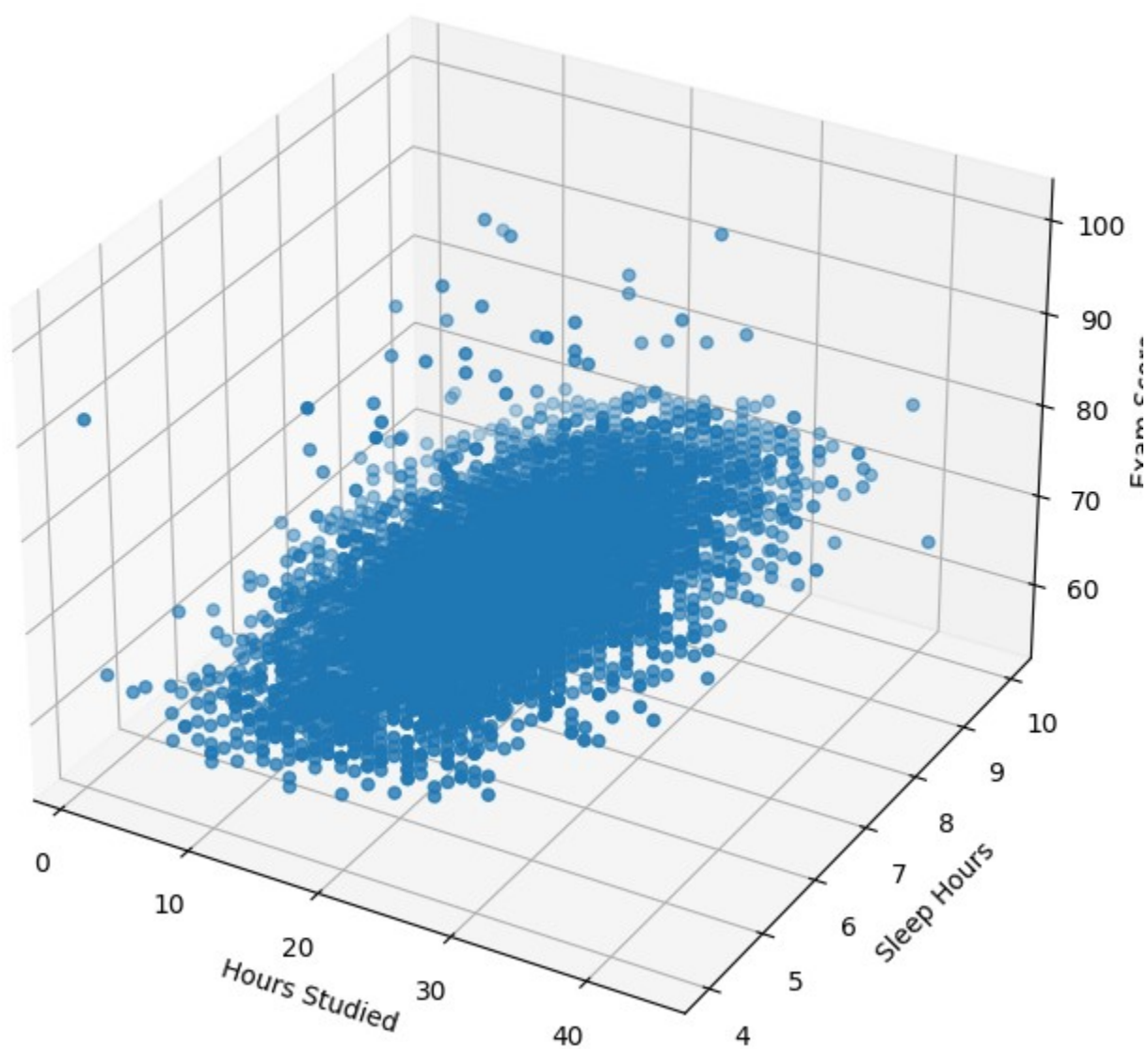


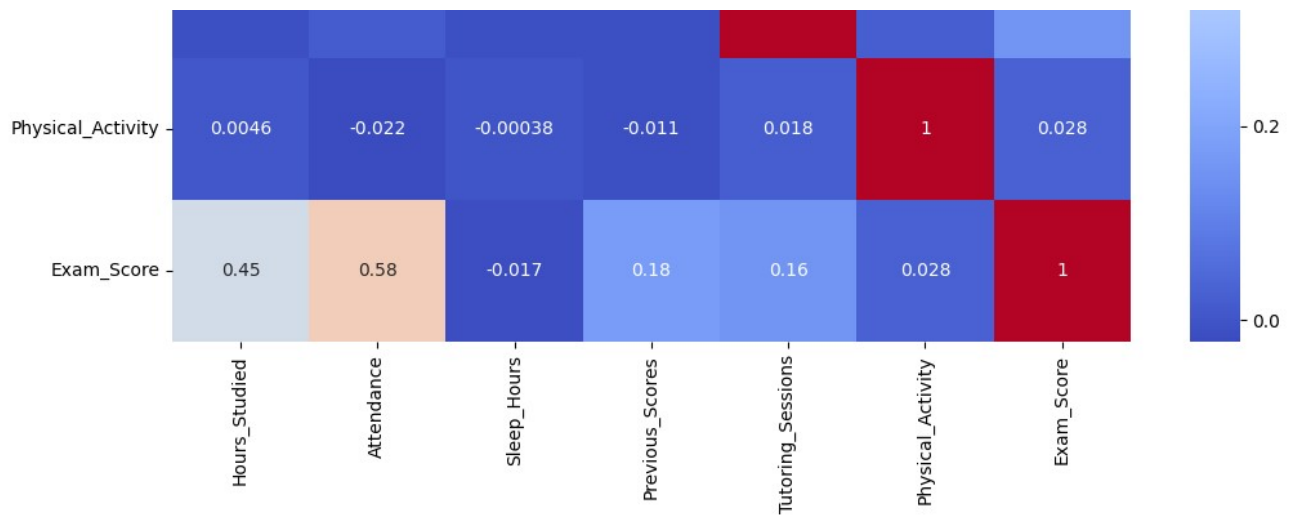




3D View of Study, Sleep, and Exam Score





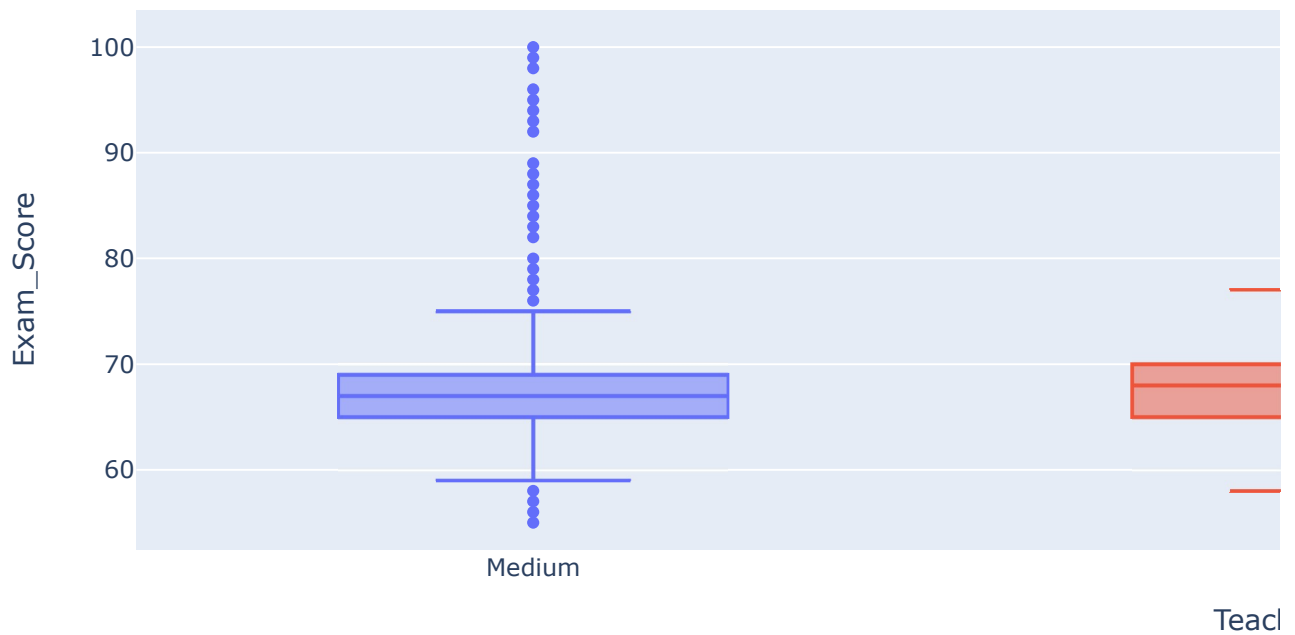


## Student Performance Dashboard

Teacher\_Quality

x ▼

Exam Score Distribution by Teacher\_Quality







1 Start coding or generate with AI.