

Project Title-To improve the student performance in examination.

- Domain - Education Sector
- Description:-
 - This project uses Python to analyze student data and predict academic performance. It applies data preprocessing, exploratory data analysis to identify key factors affecting results such as study hours, attendance, and past failures. The system predicts whether a student is likely to pass or fail are used for insights, by teachers and students.

In []:

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib as mp
import scipy
import math
```

Step:-1

Business Problem Understanding


- To improve the student performance in examinations.

Step:-2

```
In [2]: df=pd.read_csv('StudentPerformance.csv')
df.head()
```

Out[2]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sl
0	23	84	Low	High	No	
1	19	64	Low	Medium	No	
2	24	98	Medium	Medium	Yes	
3	29	89	Low	Medium	Yes	
4	19	92	Medium	Medium	Yes	



Data Exploration:-

```
In [3]: 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
```

```
In [4]: df.keys().to_list()
```

```
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',  
        'Parental_Education_Level',  
        'Distance_from_Home',  
        'Gender',  
        'Exam_Score']
```

```
In [5]: df.isnull().sum()
```

```
Out[5]: 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 [ ]:
```

```
In [6]: df.dtypes
```

```
Out[6]: Hours_Studied          int64
Attendance                    int64
Parental_Involvement          object
Access_to_Resources            object
Extracurricular_Activities     object
Sleep_Hours                    int64
Previous_Scores                int64
Motivation_Level               object
Internet_Access                object
Tutoring_Sessions              int64
Family_Income                  object
Teacher_Quality                object
School_Type                    object
Peer_Influence                 object
Physical_Activity              int64
Learning_Disabilities          object
Parental_Education_Level       object
Distance_from_Home             object
Gender                         object
Exam_Score                     int64
dtype: object
```

```
In [ ]:
```

```
In [7]: df.shape
```

```
Out[7]: (6607, 20)
```

Data exploration columns wise:-

```
In [ ]:
```

'Studied_Hours'

- Data types-Continuous variable
- No need to clean
- No need to change data type

- Data type-->Count variable(Discrete)

```
In [8]: df.rename(columns={'Hours_Studied':'Studied_Hours'},inplace=True)
df.head(2)
```

```
Out[8]:
```

	Studied_Hours	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sl
0	23	84	Low	High		
1	19	64	Low	Medium		

```
In [9]: df['Studied_Hours'].dtypes # No need to cahnge data type
```

```
Out[9]: dtype('int64')
```

```
In [10]: df['Studied_Hours'].isnull().sum() # No need to clean
```

```
Out[10]: 0
```

```
In [ ]:
```

Attendance

- data type-Continuous variable(Discrete)
- No need to fill it.
- No need to cahnge the data types

```
In [11]: df['Attendance'].isnull().sum() # no need to fill
```

```
Out[11]: 0
```

```
In [12]: df['Attendance'].dtypes
```

```
Out[12]: dtype('int64')
```

```
In [ ]:
```

'Parental_Involvement'

- Data Type-Discrete variable
- No null value
- data types is object -no need to change
- Three is three unique value- 'Low', 'Medium', 'High'
- Medium > High > Low

```
In [13]: df['Parental_Involvement'].isnull().sum()
```

```
Out[13]: 0
```

```
In [14]: df['Parental_Involvement'].dtypes
```

```
Out[14]: dtype('O')
```

```
In [15]: df['Parental_Involvement'].unique()
```

```
Out[15]: array(['Low', 'Medium', 'High'], dtype=object)
```

```
In [16]: df['Parental_Involvement'].value_counts()
```

```
Out[16]: Parental_Involvement
Medium    3362
High      1908
Low       1337
Name: count, dtype: int64
```

```
In [ ]:
```

'Access_to_Resources'

- Data type-Discrete variable
- No null Value
- Three Unique value -low , medium & high
- medium > low > high

```
In [ ]:
```

```
In [17]: df['Access_to_Resources'].isnull().sum()
```

```
Out[17]: 0
```

```
In [18]: df['Access_to_Resources'].unique()
```

```
Out[18]: array(['High', 'Medium', 'Low'], dtype=object)
```

```
In [19]: df['Access_to_Resources'].value_counts()
```

```
Out[19]: Access_to_Resources
Medium    3319
High      1975
Low       1313
Name: count, dtype: int64
```

```
In [20]: df['Access_to_Resources'].dtypes
```

```
Out[20]: dtype('O')
```

```
In [ ]:
```

'Extracurricular_Activities'

- Data type-Discrete variable
- No null Value
- No need to change data types
- Two unique values-Yes & No
- Yes > No

```
In [ ]:
```

```
In [21]: df['Extracurricular_Activities'].isnull().sum()
```

```
Out[21]: 0
```

```
In [22]: df['Extracurricular_Activities'].dtypes
```

```
Out[22]: dtype('O')
```

```
In [23]: df['Extracurricular_Activities'].unique()
```

```
Out[23]: array(['No', 'Yes'], dtype=object)
```

```
In [24]: df['Extracurricular_Activities'].value_counts()
```

```
Out[24]: Extracurricular_Activities
Yes      3938
No       2669
Name: count, dtype: int64
```

```
In [ ]:
```

'Sleep_Hours'

- Data type-continuous variable()
- No null value
- No need to cahnge the data type
- order of sleeping hours (7>8>6>9>5>10>4)

```
In [ ]:
```

```
In [25]: df['Sleep_Hours'].isnull().sum()
```

```
Out[25]: 0
```

```
In [26]: df['Sleep_Hours'].unique()
```

```
Out[26]: array([ 7,  8,  6, 10,  9,  5,  4], dtype=int64)
```

```
In [27]: df['Sleep_Hours'].value_counts()
```

```
Out[27]: Sleep_Hours
7      1741
8      1399
6      1376
9       775
5       695
10      312
4       309
Name: count, dtype: int64
```

```
In [28]: df['Sleep_Hours'].dtypes
```

```
Out[28]: dtype('int64')
```

```
In [ ]:
```

'Previous_Scores'

- Data Types-continuous variable
- No need to cahnge data type
- No null values

```
In [ ]:
```

```
In [29]: df['Previous_Scores'].isnull().sum()
```

```
Out[29]: 0
```

```
In [30]: df['Previous_Scores'].dtypes
```

```
Out[30]: dtype('int64')
```

```
In [ ]:
```

'Motivation_Level'

- Data Type-Discrete variable
- No need to change data type
- No null value
- Three unique value
- Order(Medium > Low > High)

```
In [ ]:
```

```
In [31]: df['Motivation_Level'].isnull().sum()
```

```
Out[31]: 0
```

```
In [32]: df['Motivation_Level'].dtypes
```

```
Out[32]: dtype('O')
```

```
In [33]: df['Motivation_Level'].unique()
```

```
Out[33]: array(['Low', 'Medium', 'High'], dtype=object)
```

```
In [34]: df['Motivation_Level'].value_counts()
```

```
Out[34]: Motivation_Level
Medium    3351
Low       1937
High      1319
Name: count, dtype: int64
```

```
In [ ]:
```

'Internet_Access'

- Data Type-Discrete variable
- No need to change the data type
- No null values
- Two unique(Yes & No)
- Yes > NO

```
In [ ]:
```

```
In [35]: df['Internet_Access'].isnull().sum()
```

```
Out[35]: 0
```

```
In [36]: df['Internet_Access'].dtypes
```

```
Out[36]: dtype('O')
```

```
In [37]: df['Internet_Access'].unique()
```

```
Out[37]: array(['Yes', 'No'], dtype=object)
```

```
In [38]: df['Internet_Access'].value_counts()
```

```
Out[38]: Internet_Access
Yes      6108
No        499
Name: count, dtype: int64
```

```
In [ ]:
```

'Tutoring_Sessions'

- Dsta Type-Continuous Variable
- No need to change the data type
- No null value
- 'Tutoring_Sessions' - from 0 hr to 8 hrs

```
In [ ]:
```

```
In [39]: df['Tutoring_Sessions'].isnull().sum()
```

```
Out[39]: 0
```

```
In [40]: df['Tutoring_Sessions'].dtypes
```

```
Out[40]: dtype('int64')
```

```
In [41]: df['Tutoring_Sessions'].unique()
```

```
Out[41]: array([0, 2, 1, 3, 4, 5, 6, 7, 8], dtype=int64)
```

```
In [42]: df['Tutoring_Sessions'].value_counts()
```

```
Out[42]: Tutoring_Sessions
1      2179
2      1649
0      1513
3       836
4       301
5       103
6        18
7         7
8         1
Name: count, dtype: int64
```

```
In [ ]:
```

'Family_Income'

- Data Type-Discrete variable
- No need to change data type
- No null value
- Three unique value
- order of income=low > medium > high

```
In [ ]:
```

```
In [43]: df['Family_Income'].isnull().sum()
```


Out[43]: 0

```
In [44]: df['Family_Income'].dtypes
```

Out[44]: dtype('O')

```
In [45]: df['Family_Income'].unique()
```

Out[45]: array(['Low', 'Medium', 'High'], dtype=object)

```
In [46]: df['Family_Income'].value_counts()
```

Out[46]: Family_Income
Low 2672
Medium 2666
High 1269
Name: count, dtype: int64

```
In [ ]:
```

'Teacher_Quality'

- Data Type-Discrete variable
- Some missing value
- No need to change data types
- Three unique value-High, Low & Medium
- medium > high > low

```
In [ ]:
```

```
In [47]: df['Teacher_Quality'].isnull().sum()
```

Out[47]: 78

```
In [48]: df['Teacher_Quality'].dtypes
```

Out[48]: dtype('O')

```
In [49]: df['Teacher_Quality'].unique()
```

Out[49]: array(['Medium', 'High', 'Low', nan], dtype=object)

```
In [50]: df['Teacher_Quality'].value_counts()
```

Out[50]: Teacher_Quality
Medium 3925
High 1947
Low 657
Name: count, dtype: int64

'School_Type'

- Data type-Discrete variable
- No need to change data type
- No null value
- Two unique value(Public ,Private)
- Public > Private

```
In [ ]:
```

```
In [51]: df['School_Type'].isnull().sum()
```

```
Out[51]: 0
```

```
In [52]: df['School_Type'].dtypes
```

```
Out[52]: dtype('O')
```

```
In [53]: df['School_Type'].unique()
```

```
Out[53]: array(['Public', 'Private'], dtype=object)
```

```
In [54]: df['School_Type'].value_counts()
```

```
Out[54]: School_Type
Public      4598
Private     2009
Name: count, dtype: int64
```

```
In [ ]:
```

'Peer_Influence'

- Data types-Discrete variable
- No need to change to change data types
- No null values
- Three unique values('Positive', 'Negative', 'Neutral')
- Order('Positive' > 'Neutral' > 'Negative')

```
In [ ]:
```

```
In [55]: df['Peer_Influence'].isnull().sum()
```

```
Out[55]: 0
```

```
In [56]: df['Peer_Influence'].dtypes
```

```
Out[56]: dtype('O')
```

```
In [57]: df['Peer_Influence'].unique()
```

```
Out[57]: array(['Positive', 'Negative', 'Neutral'], dtype=object)
```

```
In [58]: df['Peer_Influence'].value_counts()
```

```
Out[58]: Peer_Influence
Positive    2638
Neutral     2592
Negative    1377
Name: count, dtype: int64
```

```
In [ ]:
```

'Physical_Activity'

- Data types-Continuous variable
- NO need to cahnge data type
- No missing Value
- 'Physical_Activity' order=From 0 to 6 hrs

```
In [ ]:
```

```
In [59]: df['Physical_Activity'].isnull().sum()
```

```
Out[59]: 0
```

```
In [60]: df['Physical_Activity'].dtypes
```

```
Out[60]: dtype('int64')
```

```
In [61]: df['Physical_Activity'].unique()
```

```
Out[61]: array([3, 4, 2, 1, 5, 0, 6], dtype=int64)
```

```
In [62]: df['Physical_Activity'].value_counts()
```

```
Out[62]: Physical_Activity
3      2545
2      1627
4      1575
1       421
5       361
0        46
6         32
Name: count, dtype: int64
```

```
In [ ]:
```

'Learning_Disabilities'

- Data Types=Discrete variable
- No missing value
- No need to change data type
- Two unique value

```
In [ ]:
```

```
In [63]: df['Learning_Disabilities'].isnull().sum()
```

```
Out[63]: 0
```

```
In [64]: df['Learning_Disabilities'].dtypes
```

```
Out[64]: dtype('O')
```

```
In [65]: df['Learning_Disabilities'].unique()
```

```
Out[65]: array(['No', 'Yes'], dtype=object)
```

```
In [66]: df['Learning_Disabilities'].value_counts()
```

```
Out[66]: Learning_Disabilities
No      5912
Yes     695
Name: count, dtype: int64
```

```
In [ ]:
```

'Parental_Education_Level'

- Data type-Discrete Variable

- Some missing Values
- No need to cahnge data type
- Three unique value('High School', 'College', 'Postgraduate')
- Order='High School' > 'College' > 'Postgraduate'

In []:

In [67]: `df['Parental_Education_Level'].isnull().sum()`

Out[67]: 90

In [68]: `df['Parental_Education_Level'].dtypes`

Out[68]: dtype('O')

In [69]: `df['Parental_Education_Level'].unique()`

Out[69]: array(['High School', 'College', 'Postgraduate', nan], dtype=object)

In [70]: `df['Parental_Education_Level'].value_counts()`

Out[70]: Parental_Education_Level
High School 3223
College 1989
Postgraduate 1305
Name: count, dtype: int64

In []:

'Distance_from_Home'

- Data type=Discrete Variable
- Some Missing Value
- NO need to change data type
- Three missing value('Near', 'Moderate', 'Far')
- Order='Near' > 'Moderate' > 'Far'

In []:

In [71]: `df['Distance_from_Home'].isnull().sum()`

Out[71]: 67

In [72]: `df['Distance_from_Home'].dtypes`

Out[72]: dtype('O')

In [73]: `df['Distance_from_Home'].unique()`

Out[73]: array(['Near', 'Moderate', 'Far', nan], dtype=object)

In [74]: `df['Distance_from_Home'].value_counts()`

Out[74]: Distance_from_Home
Near 3884
Moderate 1998
Far 658
Name: count, dtype: int64

In []:

'Gender'

- Data Type=Discrete Variable
- No need to cahnge data type
- No missing values
- Two unique value('Male', 'Female')
- Order=('Male' > 'Female')

In []:

In [75]: `df['Gender'].isnull().sum()`

Out[75]: 0

In [76]: `df['Gender'].dtypes`

Out[76]: dtype('O')

In [77]: `df['Gender'].unique()`

Out[77]: array(['Male', 'Female'], dtype=object)

In [78]: `df['Gender'].value_counts()`

Out[78]: Gender
Male 3814
Female 2793
Name: count, dtype: int64

In []:

'Exam_Score'

- Data type=Continuous Varoable
- No need to cahnge data type
- No missing values

In []:

In [79]: `df['Exam_Score'].isnull().sum()`

Out[79]: 0

In [80]: `df['Exam_Score'].dtypes`

Out[80]: dtype('int64')

In [81]: `df.head(2)`

Out[81]:

	Studied_Hours	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sl
0	23	84	Low	High	No	
1	19	64	Low	Medium	No	

```
In [82]: df.columns.to_list()
```

```
Out[82]: ['Studied_Hours',  
          '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']
```

```
In [83]: Discrete=['Studied_Hours','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']
```

Data Preprocessing:-

- Data Cleaning
- Dimension Reduction
- Data Transformation

```
In [84]: df.duplicated().sum()
```

```
Out[84]: 0
```

```
In [85]: df.isnull().sum()
```

```
Out[85]: Studied_Hours          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 []:

In [86]: `missing_values=['Teacher_Quality','Parental_Education_Level','Distance_from_Home']
df[missing_values].describe()`

Out[86]:

	Teacher_Quality	Parental_Education_Level	Distance_from_Home
count	6529	6517	6540
unique	3	3	3
top	Medium	High School	Near
freq	3925	3223	3884

In []:

In [87]: `df['Teacher_Quality'] = df['Teacher_Quality'].fillna(df['Teacher_Quality'].mode()[0])
df['Parental_Education_Level']=df['Parental_Education_Level'].fillna(df['Parental_Education_Level'].mode()[0])
df['Distance_from_Home']=df['Distance_from_Home'].fillna(df['Distance_from_Home'].mode()[0])`

In [88]: `df.head(10)`

Out[88]:

	Studied_Hours	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sl
0	23	84	Low	High	No	
1	19	64	Low	Medium	No	
2	24	98	Medium	Medium	Yes	
3	29	89	Low	Medium	Yes	
4	19	92	Medium	Medium	Yes	
5	19	88	Medium	Medium	Yes	
6	29	84	Medium	Low	Yes	
7	25	78	Low	High	Yes	
8	17	94	Medium	High	No	
9	23	98	Medium	Medium	Yes	

In [89]: `df.isnull().sum()`

```
Out[89]: Studied_Hours      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 [90]: df.to_csv("CleanStudentData.csv",index=False)
```

Step-4:- Data Analysing

```
In [ ]:
```

```
In [91]: df.describe(include = 'all')
```

Out[91]:

	Studied_Hours	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities
count	6607.000000	6607.000000	6607	6607	6607
unique	NaN	NaN	3	3	3
top	NaN	NaN	Medium	Medium	Yes
freq	NaN	NaN	3362	3319	3950
mean	19.975329	79.977448	NaN	NaN	NaN
std	5.990594	11.547475	NaN	NaN	NaN
min	1.000000	60.000000	NaN	NaN	NaN
25%	16.000000	70.000000	NaN	NaN	NaN
50%	20.000000	80.000000	NaN	NaN	NaN
75%	24.000000	90.000000	NaN	NaN	NaN
max	44.000000	100.000000	NaN	NaN	NaN

```
In [ ]:
```

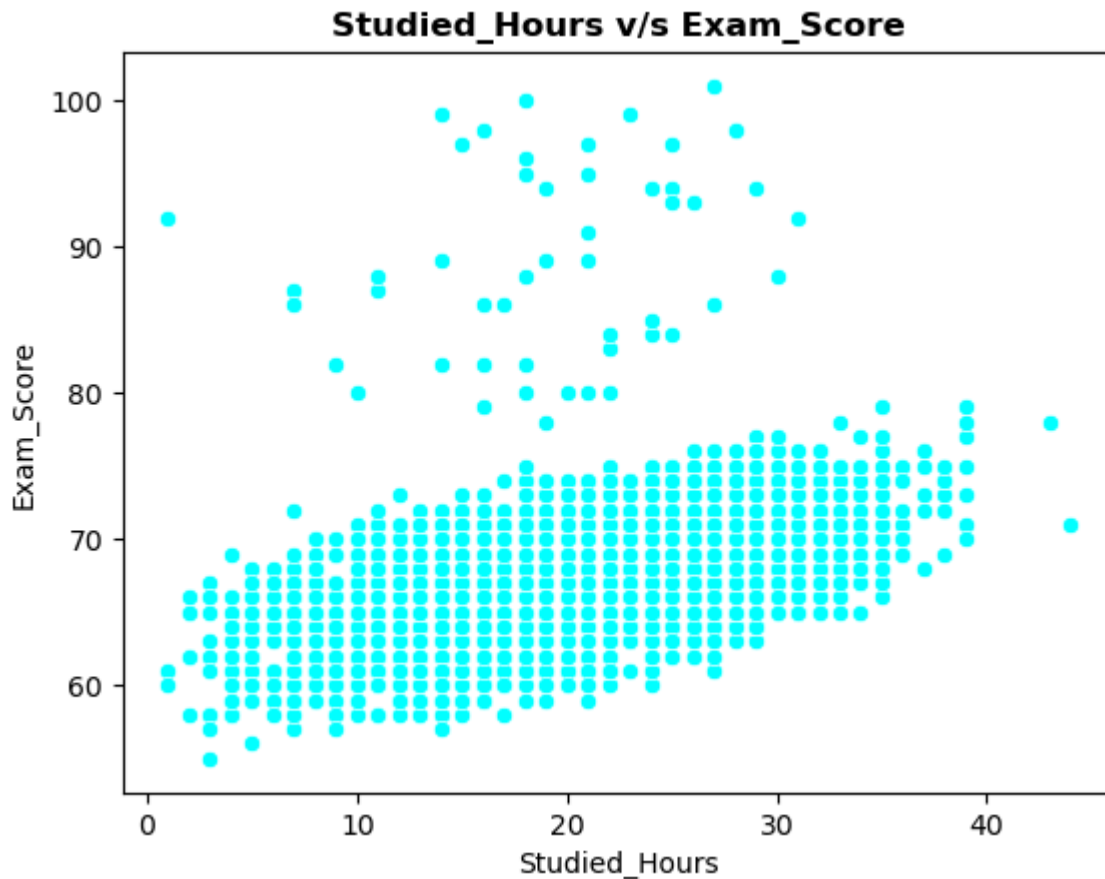
```
In [92]: df.keys()
```



```
Out[92]: Index(['Studied_Hours', '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 [93]: sns.scatterplot(x=df['Studied_Hours'],y=df['Exam_Score'],color='cyan')  
plt.title('Studied_Hours v/s Exam_Score',fontweight="bold")  
plt.show()
```

#Q1=who get good score?

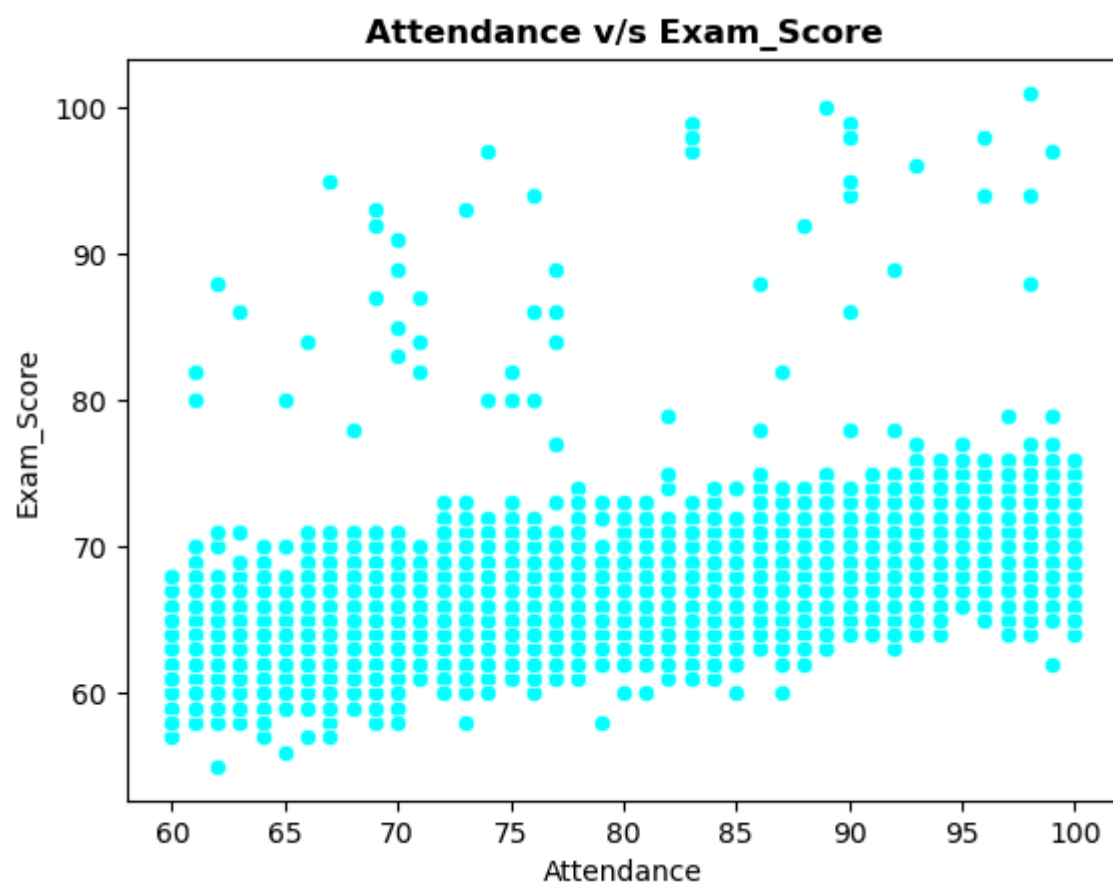


```
In [94]: #Ans-One who study b/w 20 hrs to 30 hrs whose score is better
```

```
In [ ]:
```

```
In [95]: sns.scatterplot(x=df['Attendance'],y=df['Exam_Score'],color='cyan')  
plt.title('Attendance v/s Exam_Score',fontweight="bold")  
# plt.savefig("Attendance v/s Exam_Score.jpg")  
plt.show()
```

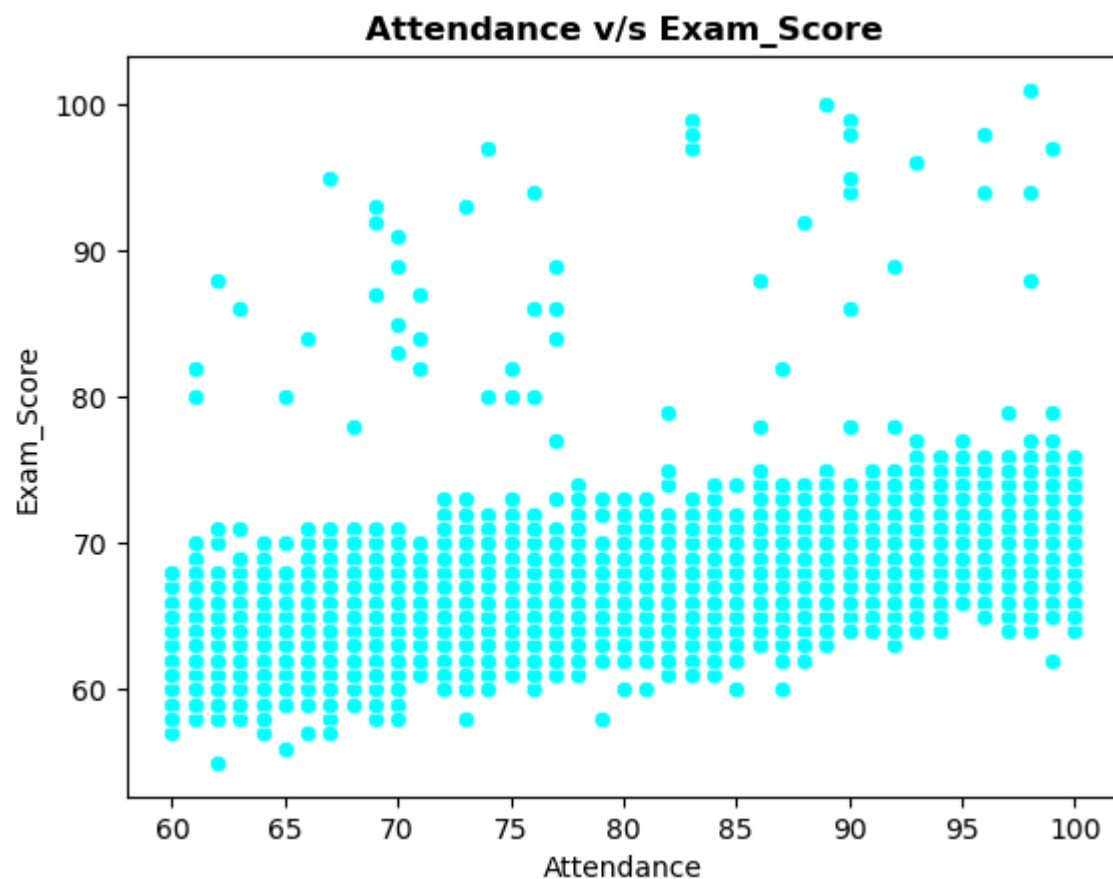
#Q2-the student whose attendance is good .then what is their scores ?



In [96]: *#Ans-the students whose attendance is above 90+ their score is the best*

In []:

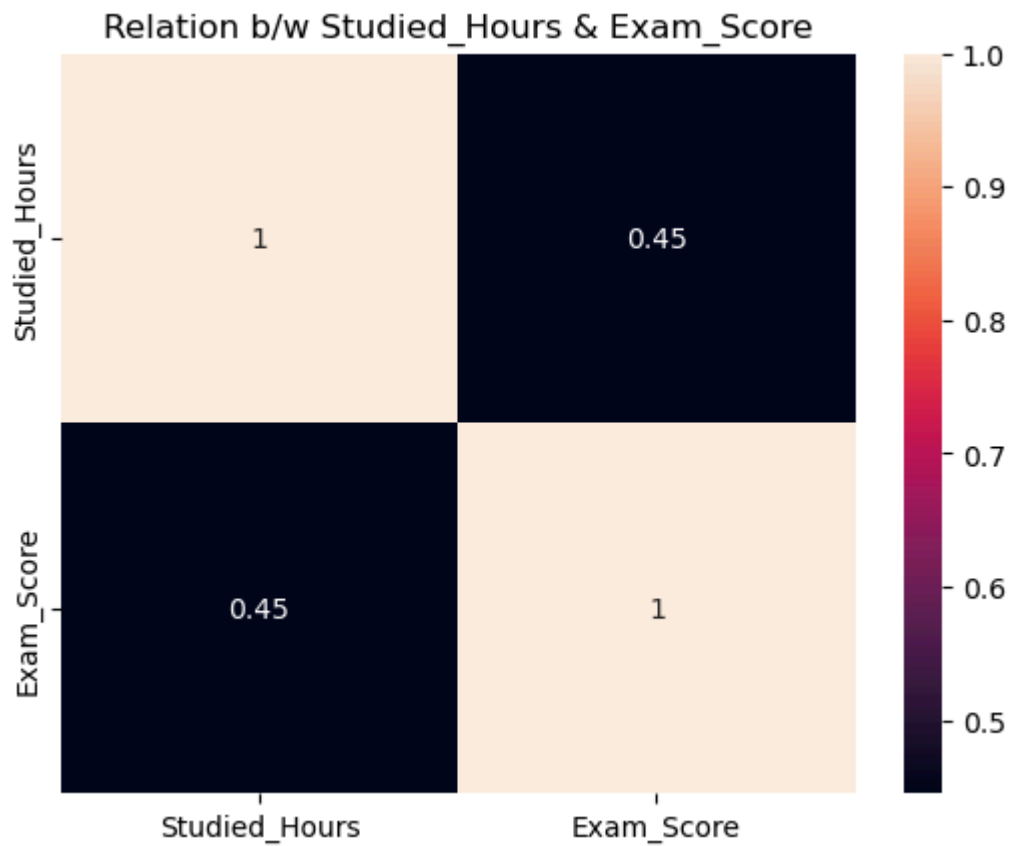
```
In [97]: sns.scatterplot(x=df['Attendance'],y=df['Exam_Score'],color='cyan')
plt.title('Attendance v/s Exam_Score',fontweight="bold")
plt.show()
```



```
In [98]: a=df[['Studied_Hours','Exam_Score']].corr()
# print(a)
plt.title('Relation b/w Studied_Hours & Exam_Score')

sns.heatmap(a,annot=True)
plt.savefig("studiedh.jpg")
plt.figure(figsize=(4,3))

plt.show()
```



<Figure size 400x300 with 0 Axes>

Corelation is weak b/w Studied_Hours & Exam_Score

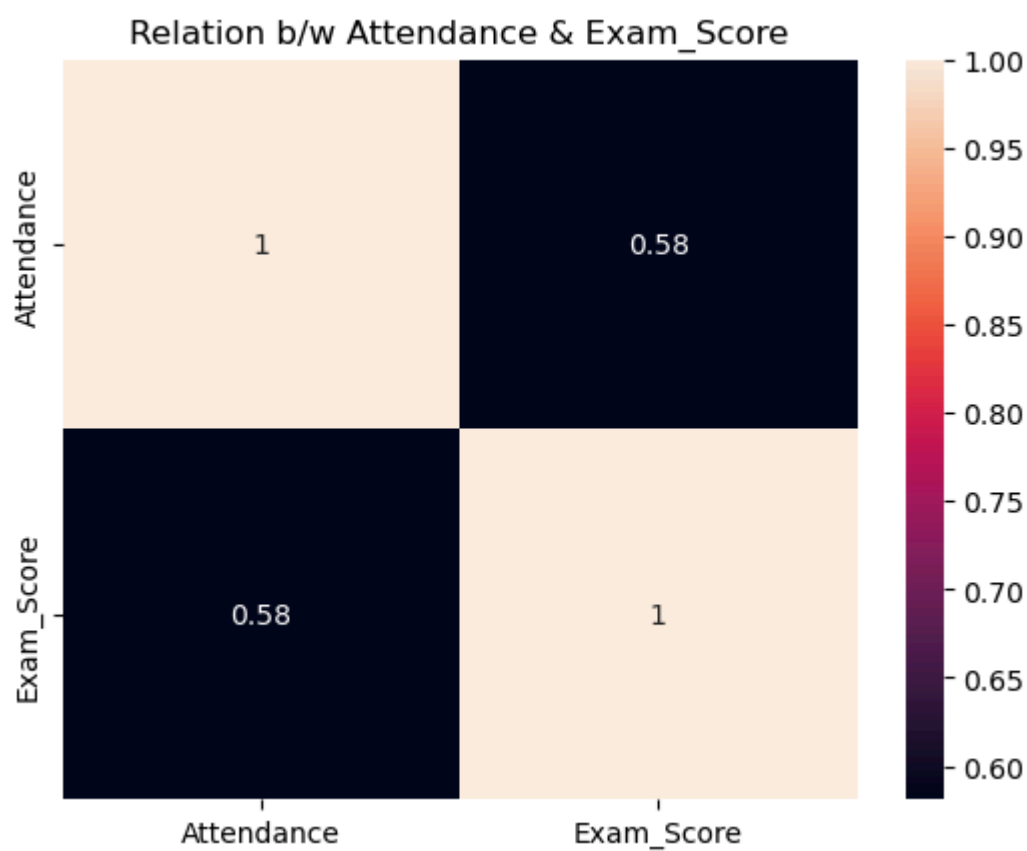
- So, A few effect on Exam_Score

In []:

```
In [99]: b=df[['Attendance','Exam_Score']].corr()
print(b)
plt.title('Relation b/w Attendance & Exam_Score')
sns.heatmap(b,annot=True)
plt.savefig("Attendance.jpg")
plt.figure(figsize=(4,3))

plt.show()
```

	Attendance	Exam_Score
Attendance	1.000000	0.581072
Exam_Score	0.581072	1.000000



In []:

Corelation is moderate b/w Attendance & Exam_Score

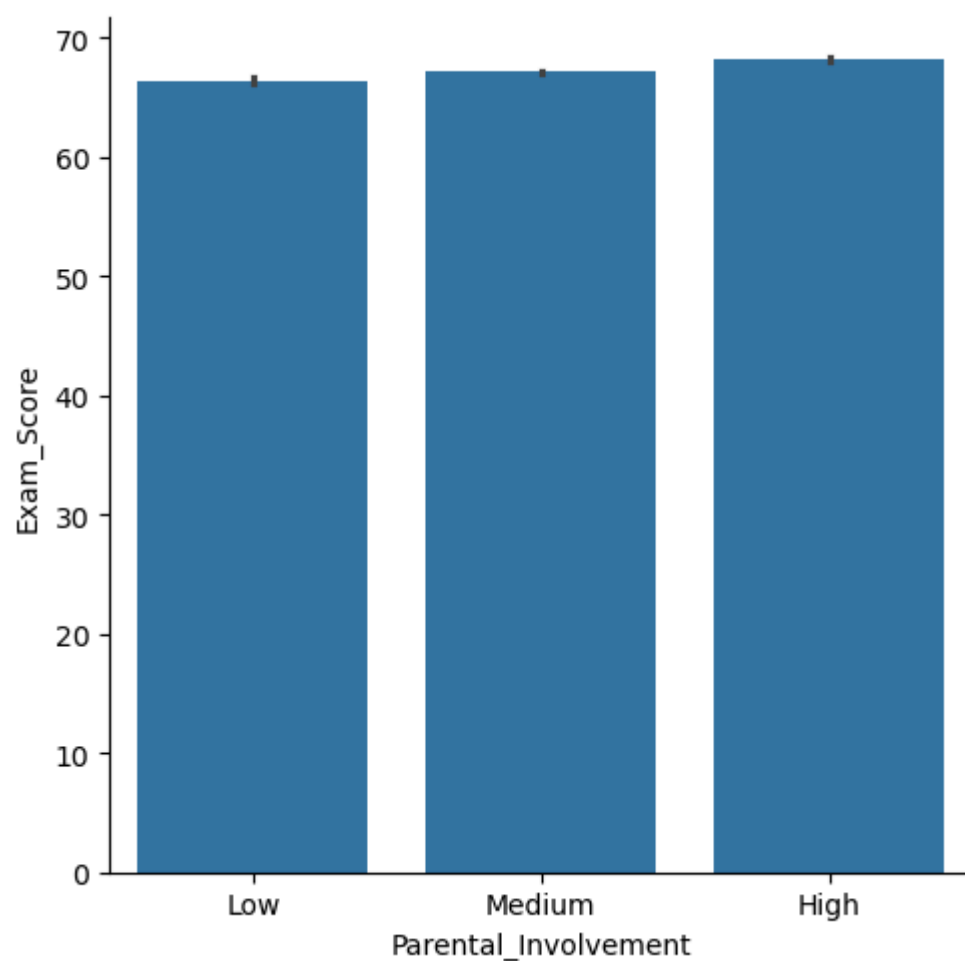
- So, A few effect on Exam_Score

In []:

```
In [100... df.groupby('Parental_Involvement')['Exam_Score'].mean()
```

```
Out[100... Parental_Involvement
High      68.092767
Low       66.358265
Medium    67.098156
Name: Exam_Score, dtype: float64
```

```
In [101... sns.catplot(x='Parental_Involvement',y='Exam_Score',data=df,kind='bar')
plt.show()
```



No effect on Exam Score by Parental_Involvement

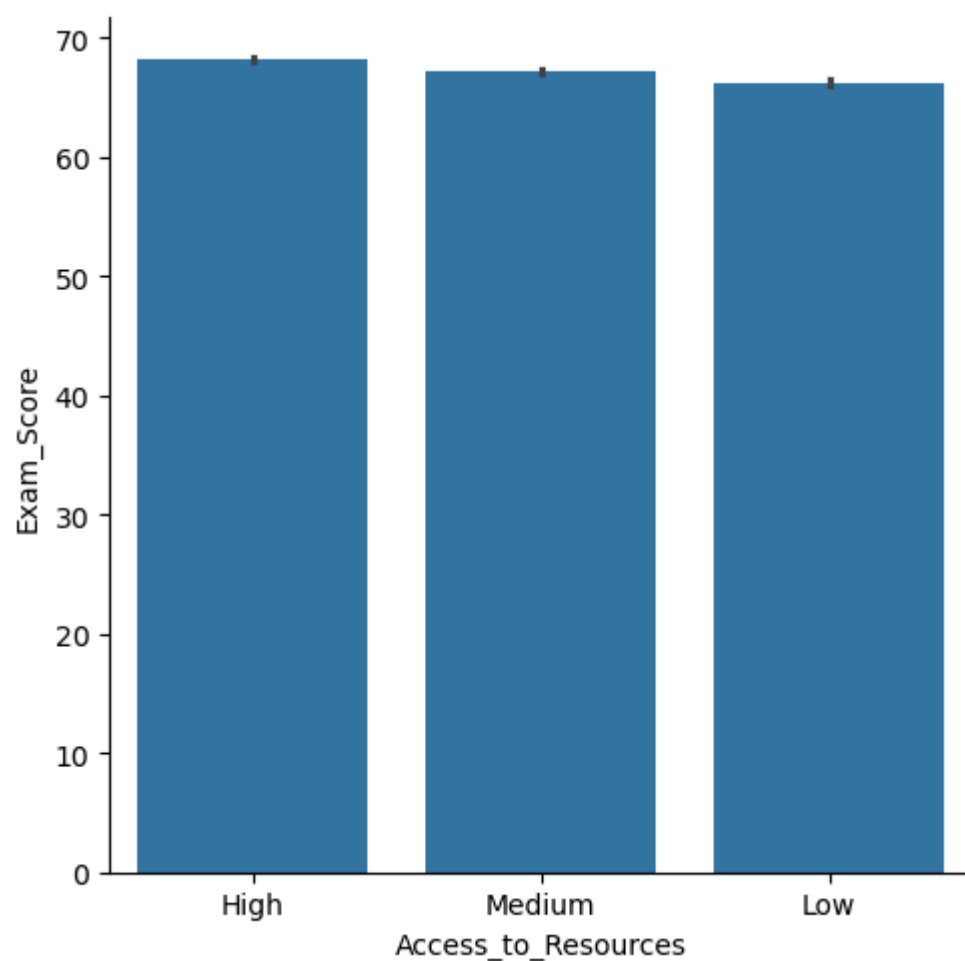
- Here mean score is almost same

In []:

```
In [102...] df.groupby('Access_to_Resources')['Exam_Score'].mean()
```

```
Out[102...] Access_to_Resources
High      68.092152
Low       66.203351
Medium    67.134378
Name: Exam_Score, dtype: float64
```

```
In [103...] sns.catplot(x='Access_to_Resources',y='Exam_Score',data=df,kind='bar')
plt.savefig('reso.jpg')
plt.show()
```



No effect on Exam Score by Access_to_Resources

- Here mean score is almost same

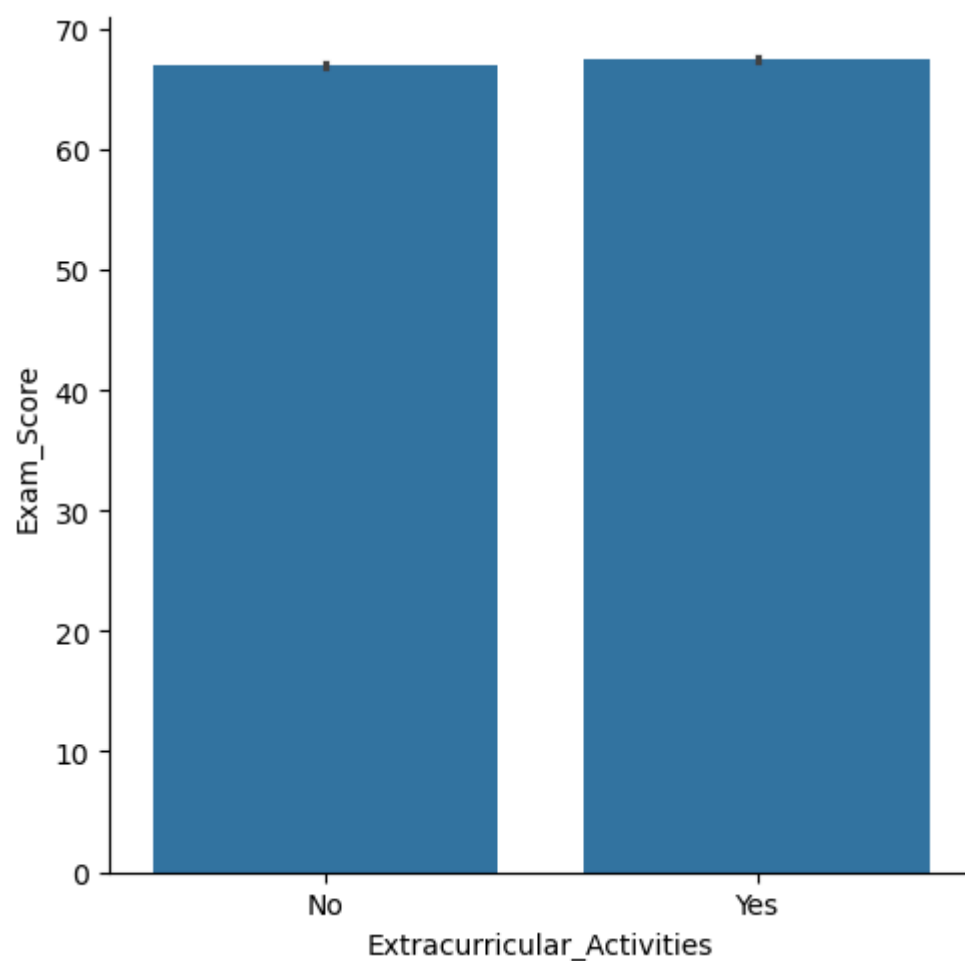
In []:

```
In [104... df.groupby('Extracurricular_Activities')['Exam_Score'].mean()
```

```
Out[104... Extracurricular_Activities  
No      66.931435  
Yes     67.441849  
Name: Exam_Score, dtype: float64
```

In []:

```
In [105... sns.catplot(x='Extracurricular_Activities',y='Exam_Score',data=df,kind='bar')  
plt.show()
```



In []:

No effect on Exam Score by Extracurricular_Activities

- Here mean score is almost same

In []:

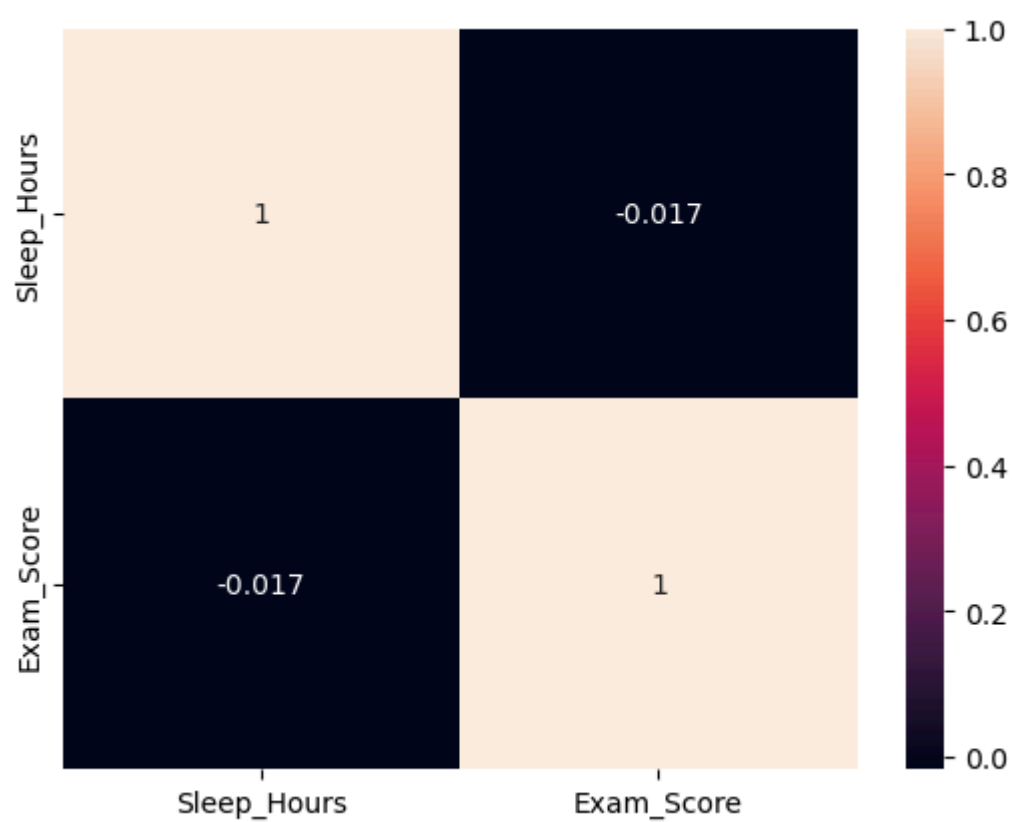
```
In [106... cor=df[['Sleep_Hours','Exam_Score']].corr()  
cor
```

Out[106...

	Sleep_Hours	Exam_Score
Sleep_Hours	1.000000	-0.017022
Exam_Score	-0.017022	1.000000

In []:

```
In [107... sns.heatmap(cor,annot=True)  
plt.savefig('sleephour.jpg')  
plt.show()
```



In []:

Here -ve corr. but near to 0 so a few effect on exam score

In []:

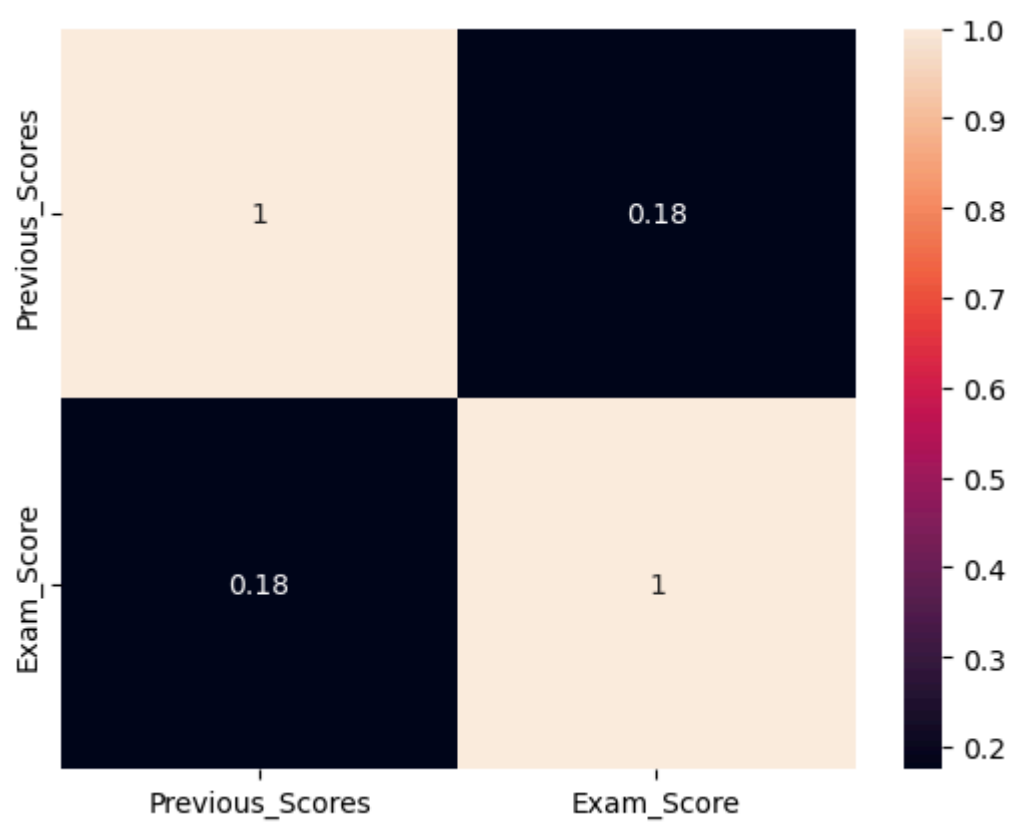
```
In [108... score=df[['Previous_Scores', 'Exam_Score']].corr()
score
```

Out[108...

	Previous_Scores	Exam_Score
Previous_Scores	1.000000	0.175079
Exam_Score	0.175079	1.000000

In []:

```
In [109... sns.heatmap(score,annot=True)
plt.show()
```

In []:

No effect on exam score from previous score

In []:

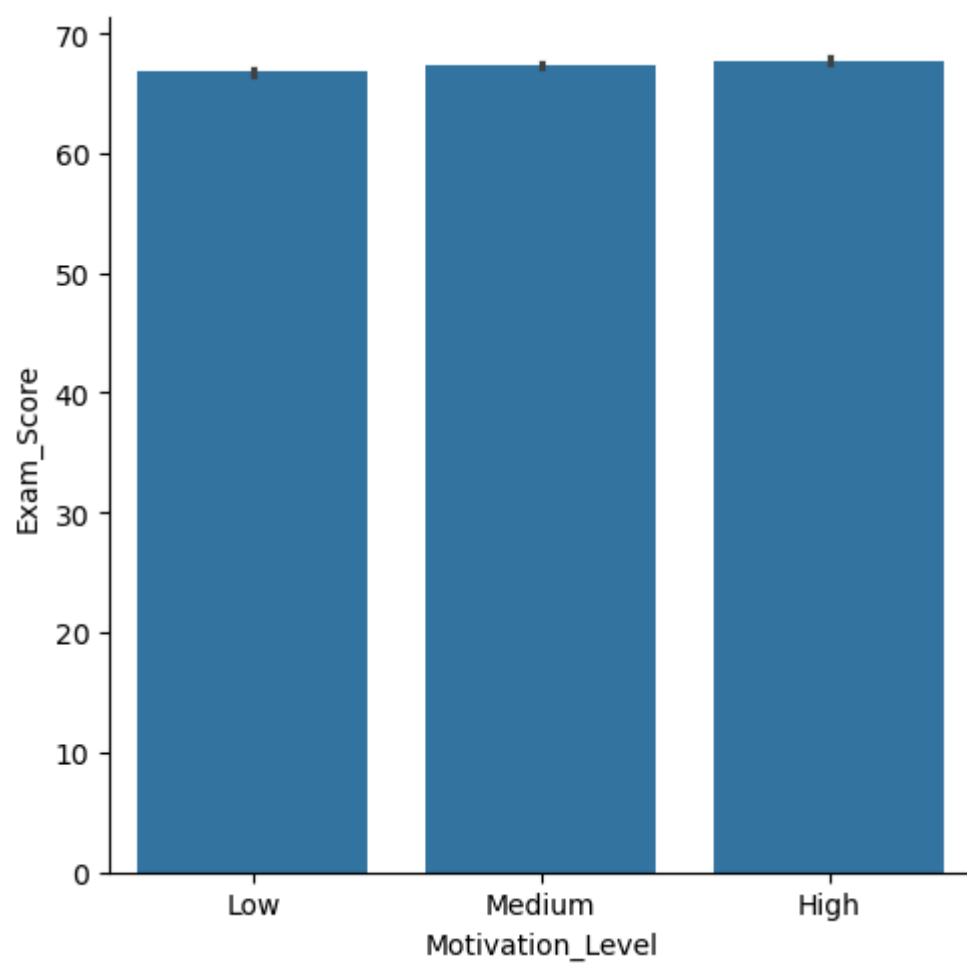
In []:

```
In [110... df.groupby('Motivation_Level')['Exam_Score'].mean()
```

```
Out[110... Motivation_Level
High      67.704321
Low       66.752194
Medium    67.330648
Name: Exam_Score, dtype: float64
```

In []:

```
In [111... sns.catplot(x='Motivation_Level',y='Exam_Score',data=df,kind='bar')
plt.show()
```



In []:

No effect on exam score from 'Motivation_Level'

In []:

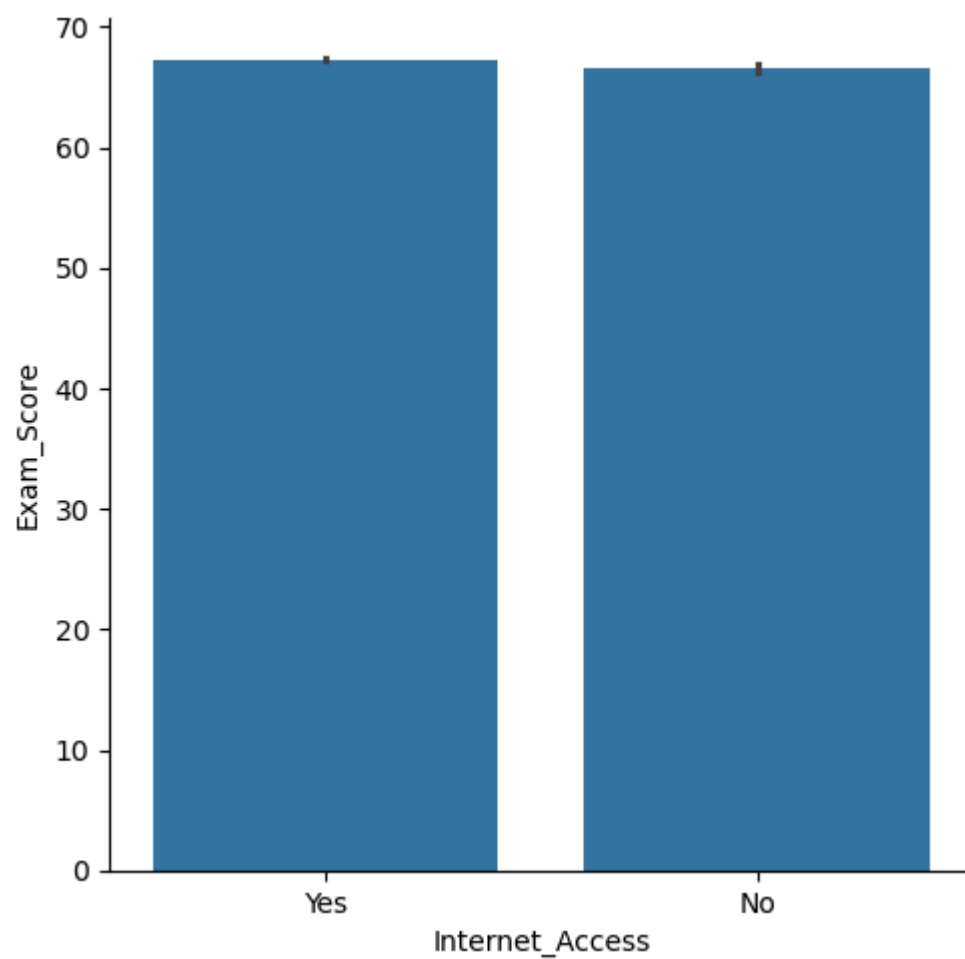
In []:

```
In [112...] df.groupby('Internet_Access')['Exam_Score'].mean()
```

```
Out[112...] Internet_Access
No      66.535070
Yes     67.292895
Name: Exam_Score, dtype: float64
```

In []:

```
In [113...] sns.catplot(x='Internet_Access',y='Exam_Score',kind='bar',data=df)
plt.show()
```



In []:

No effect on exam score from 'Internet_Access'

In []:

In []:

In [114... `ts=df[['Tutoring_Sessions', 'Exam_Score']].corr()
ts`

Out[114...

	Tutoring_Sessions	Exam_Score
Tutoring_Sessions	1.000000	0.156525
Exam_Score	0.156525	1.000000

In []:

In [115... `sns.heatmap(ts,annot=True)
plt.show()`



In []:

A few effect on exam score from 'Tutoring_Sessions'

In []:

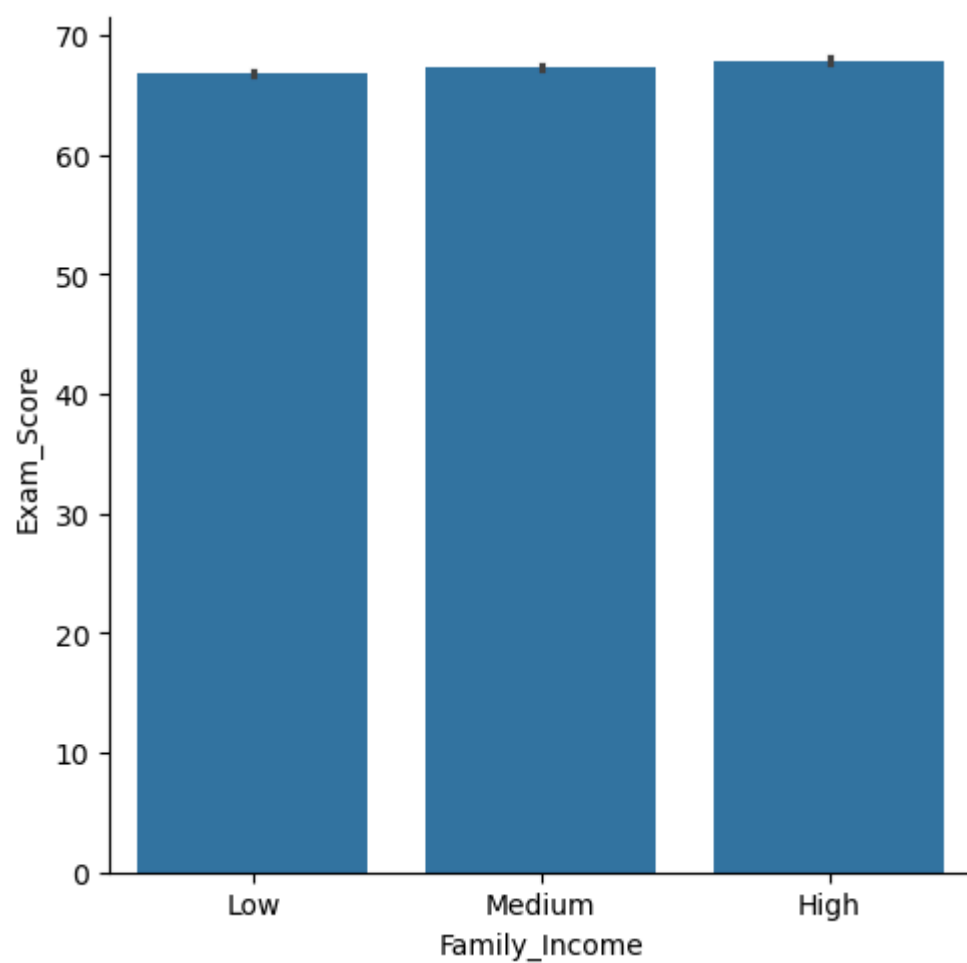
In []:

In [116... `df.groupby('Family_Income')['Exam_Score'].mean()`

Out[116... Family_Income
High 67.842396
Low 66.848428
Medium 67.334959
Name: Exam_Score, dtype: float64

In []:

In [117... `sns.catplot(x='Family_Income',y='Exam_Score',data=df,kind='bar')`
`plt.show()`



In []:

No effect on exam score from 'Family_Income'

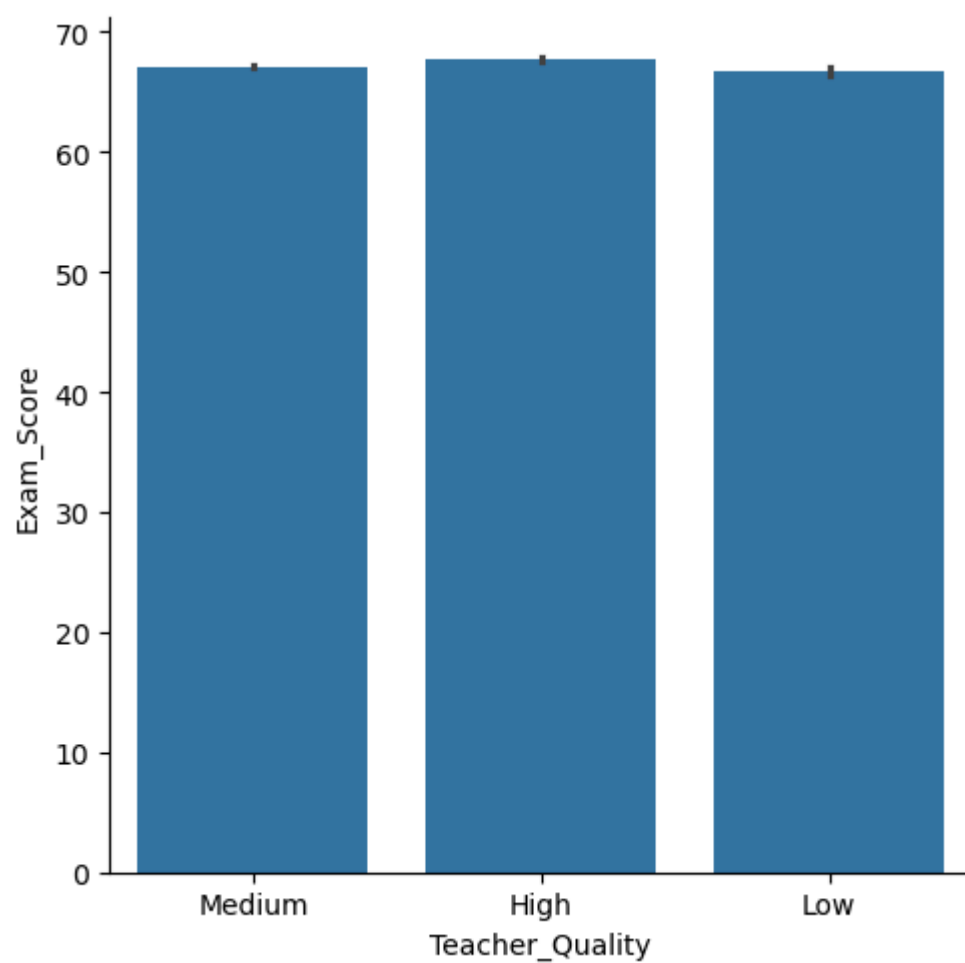
In []:

In [118... `df.groupby('Teacher_Quality')['Exam_Score'].mean()`

Out[118... `Teacher_Quality`
High 67.676939
Low 66.753425
Medium 67.100175
Name: Exam_Score, dtype: float64

In []:

In [119... `sns.catplot(x='Teacher_Quality',y='Exam_Score',data=df,kind='bar')`
`plt.show()`



No effect on exam score from 'Teacher_Quality'

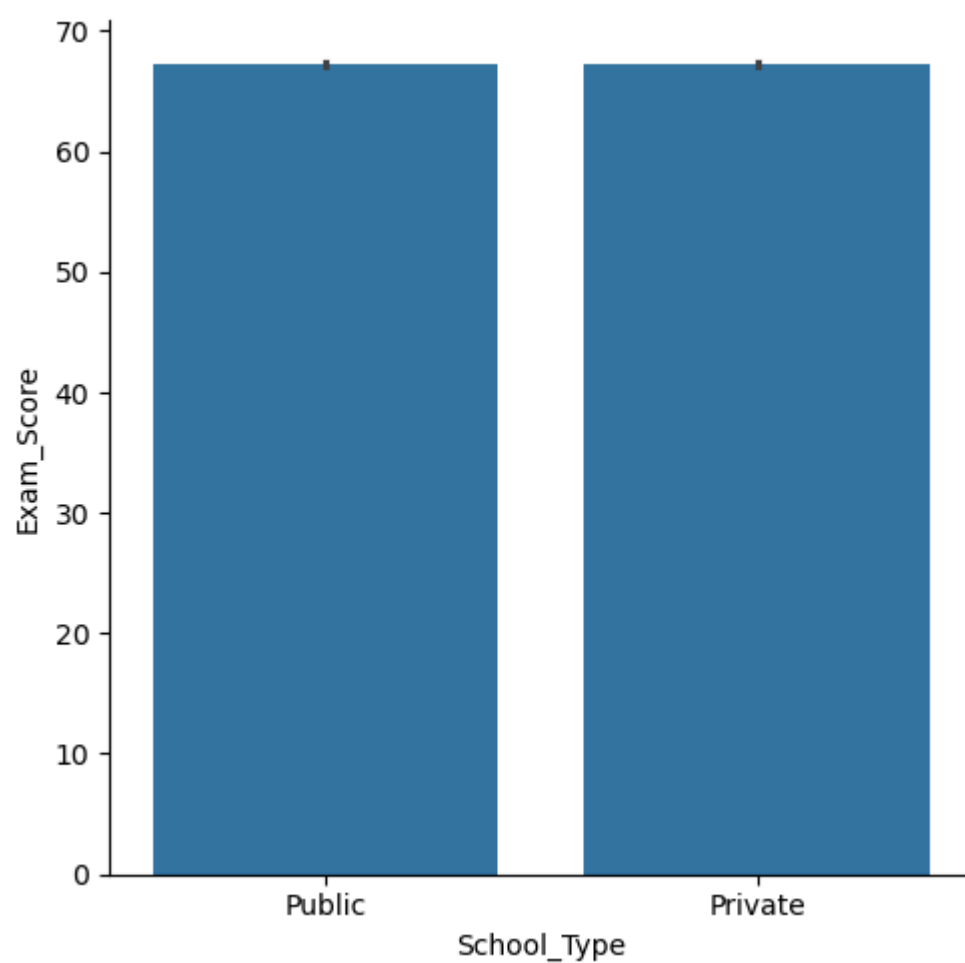
In []:

In [120... `df.groupby('School_Type')['Exam_Score'].mean()`

Out[120...
School_Type
Private 67.287705
Public 67.212919
Name: Exam_Score, dtype: float64

In []:

In [121... `sns.catplot(x='School_Type',y='Exam_Score',data=df,kind='bar')`
`plt.show()`



No effect on exam score from 'School_Type'

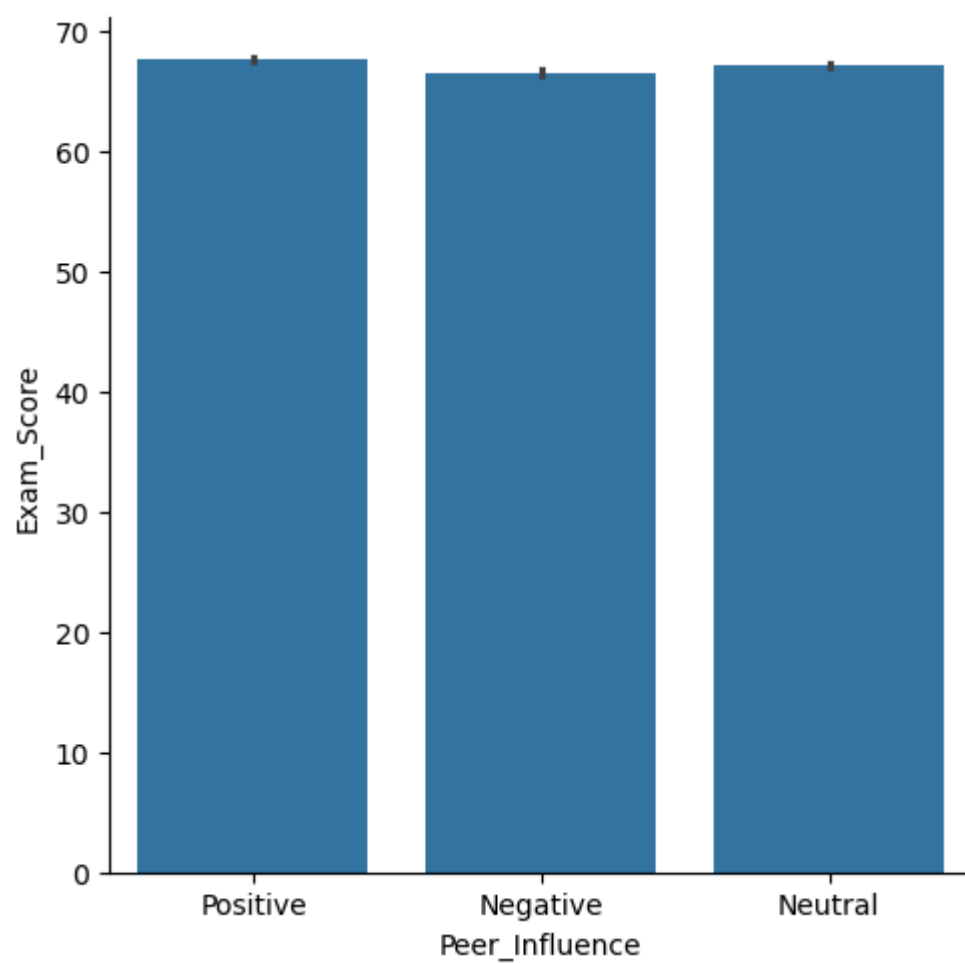
In []:

```
In [122...] df.groupby('Peer_Influence')['Exam_Score'].mean()
```

```
Out[122...] Peer_Influence
Negative    66.564270
Neutral     67.197917
Positive    67.623199
Name: Exam_Score, dtype: float64
```

In []:

```
In [123...] sns.catplot(x='Peer_Influence',y='Exam_Score',data=df,kind='bar')
plt.show()
```



No effect on exam score from 'Peer_Influence'

In []:

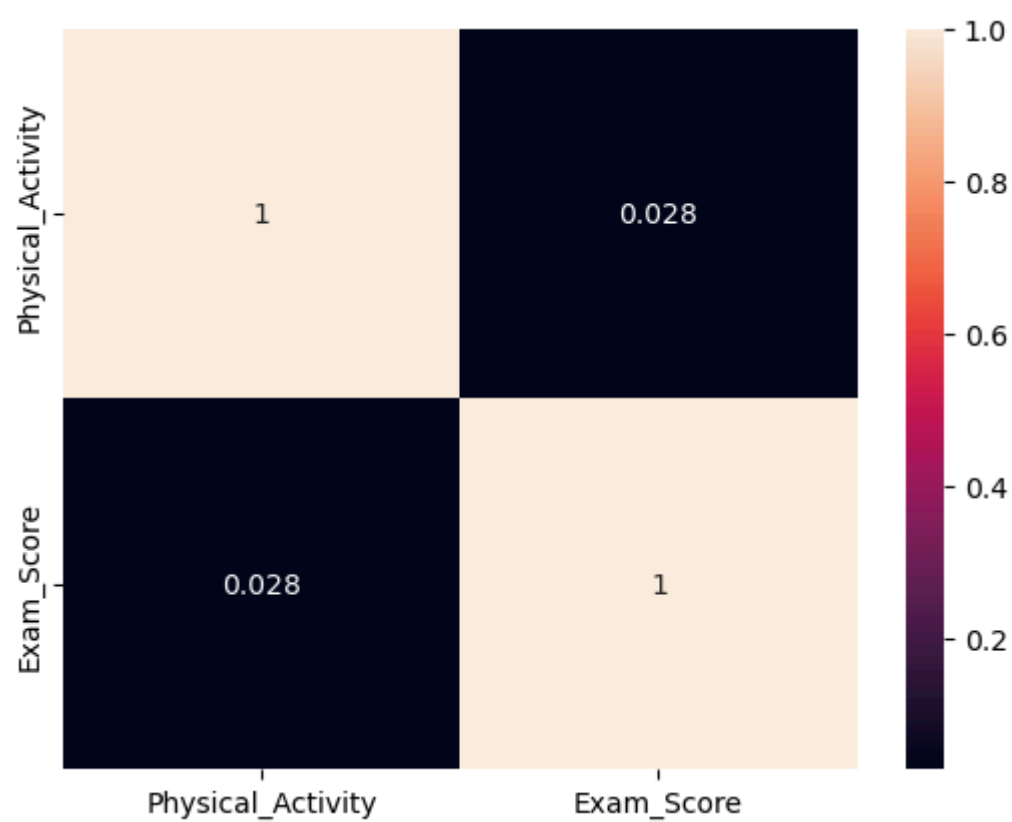
```
In [124... pa=df[['Physical_Activity', 'Exam_Score']].corr()  
pa
```

Out[124...

	Physical_Activity	Exam_Score
Physical_Activity	1.000000	0.027824
Exam_Score	0.027824	1.000000

In []:

```
In [125... sns.heatmap(pa,annot=True)  
plt.show()
```

In []:

NO effect on exam score from 'Physical_Activity'

In []:

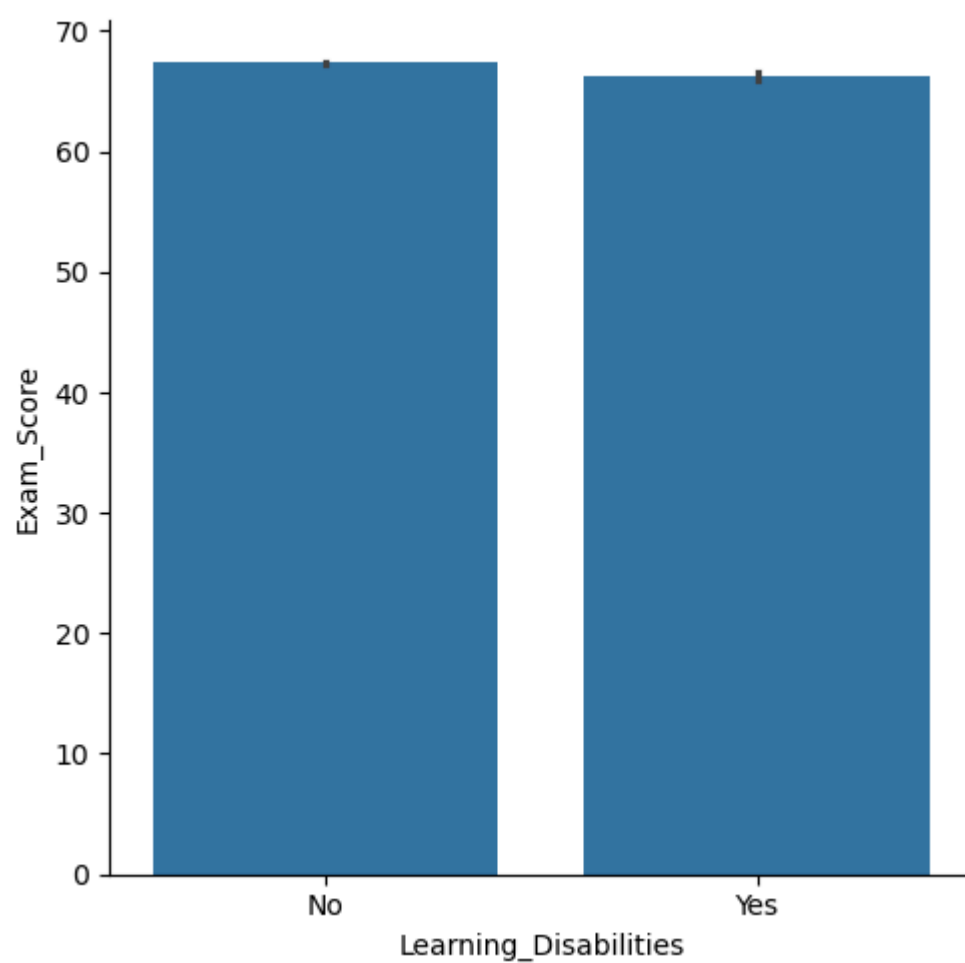
In []:

In [126... `df.groupby('Learning_Disabilities')['Exam_Score'].mean()`

Out[126... Learning_Disabilities
No 67.349120
Yes 66.270504
Name: Exam_Score, dtype: float64

In []:

In [127... `sns.catplot(x='Learning_Disabilities',y='Exam_Score',data=df,kind='bar')`
`plt.show()`



No effect on exam score from 'Learning_Disabilities'

In []:

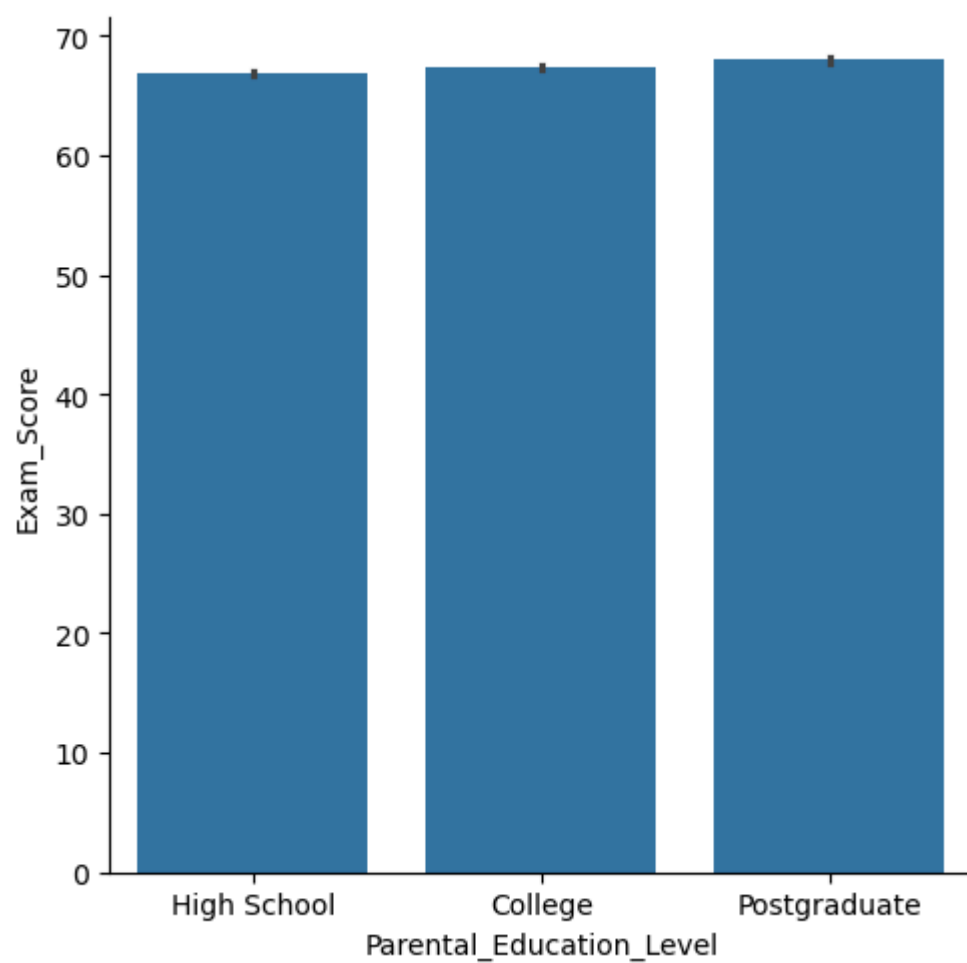
In []:

```
In [128...] df.groupby('Parental_Education_Level')['Exam_Score'].mean()
```

```
Out[128...] Parental_Education_Level
College      67.315737
High School  66.897978
Postgraduate 67.970881
Name: Exam_Score, dtype: float64
```

In []:

```
In [129...] sns.catplot(x='Parental_Education_Level',y='Exam_Score',data=df,kind='bar')
plt.show()
```



NO effect on exam score from 'Parental_Education_Level'

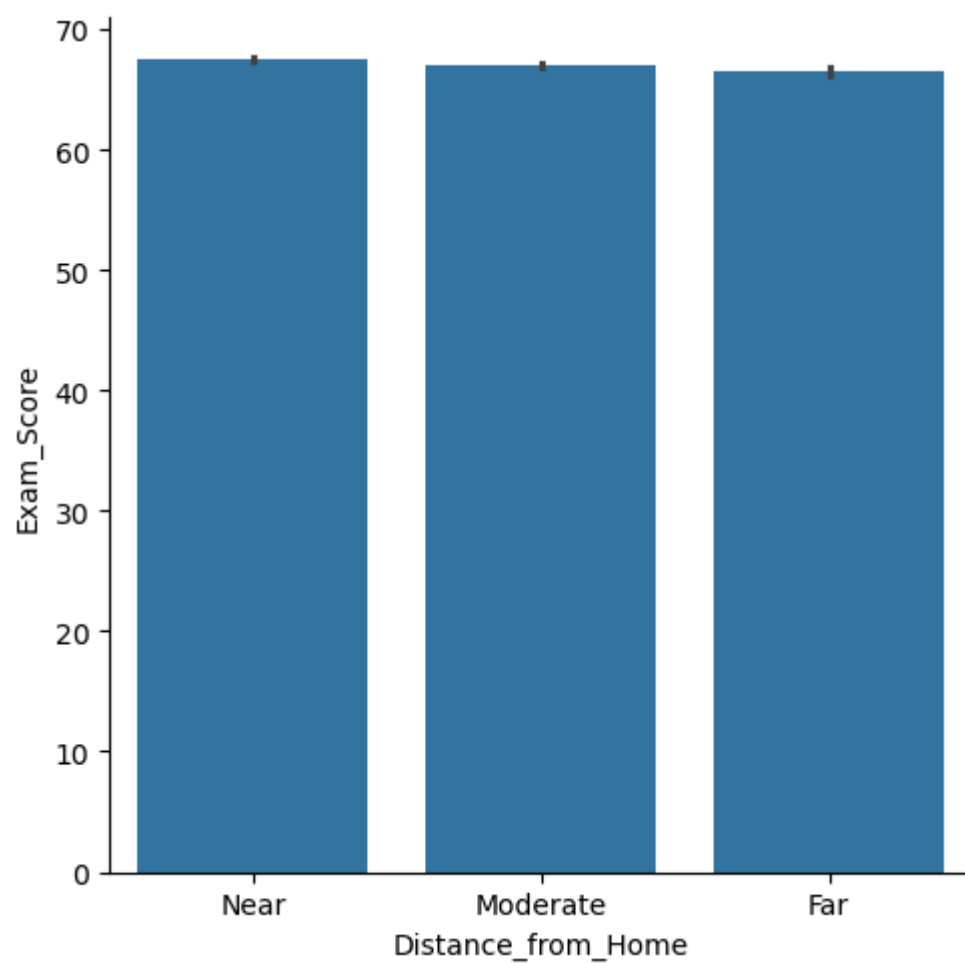
In []:

```
In [130... df.groupby('Distance_from_Home')['Exam_Score'].mean()
```

```
Out[130... Distance_from_Home
Far          66.457447
Moderate     66.981481
Near         67.493799
Name: Exam_Score, dtype: float64
```

In []:

```
In [131... sns.catplot(x='Distance_from_Home',y='Exam_Score',data=df,kind='bar')
plt.show()
```



No effect on exam score from 'Distance_from_Home'

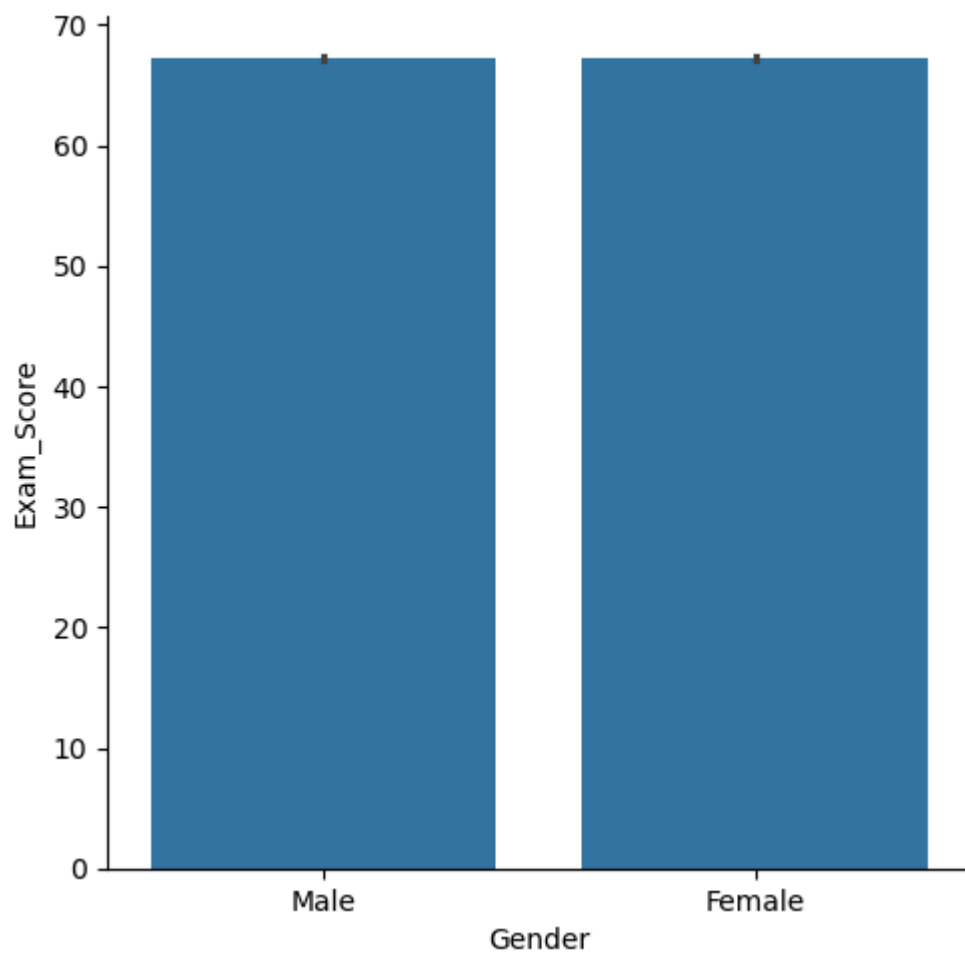
In []:

```
In [132... df.groupby('Gender')['Exam_Score'].mean()
```

```
Out[132... Gender
Female    67.244898
Male      67.228894
Name: Exam_Score, dtype: float64
```

In []:

```
In [133... sns.catplot(x='Gender',y='Exam_Score',data=df,kind='bar')
plt.show()
```



In []:

No effect on exam score from 'Gender'

In []:

In []:

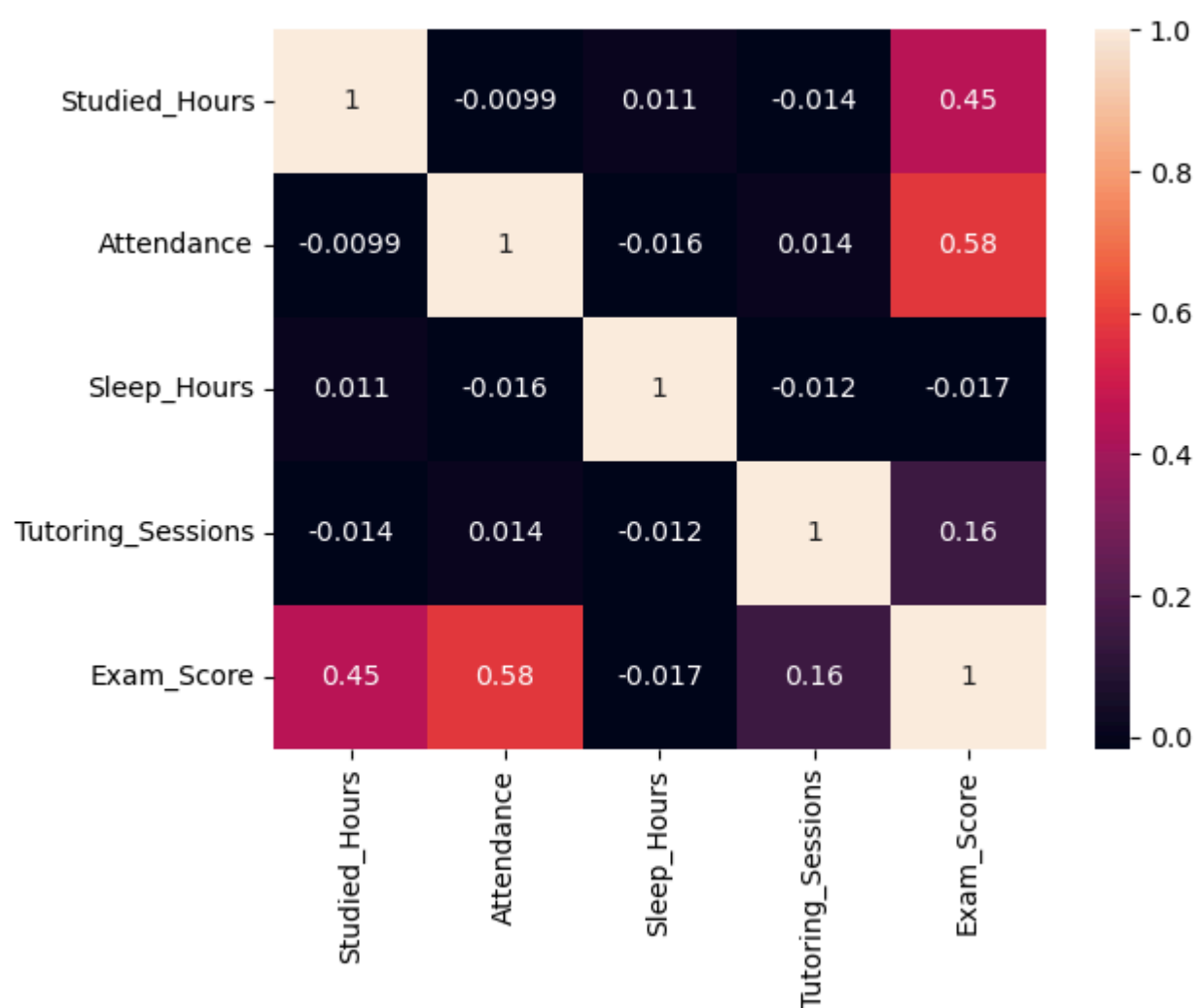
In [134... `ef_col=df[['Studied_Hours','Attendance','Sleep_Hours','Tutoring_Sessions','Exam_Score']].corr`
`# not_ef_col=[]`
`ef_col`

Out[134...

	Studied_Hours	Attendance	Sleep_Hours	Tutoring_Sessions	Exam_Score
Studied_Hours	1.000000	-0.009908	0.010977	-0.014282	0.445455
Attendance	-0.009908	1.000000	-0.015918	0.014324	0.581072
Sleep_Hours	0.010977	-0.015918	1.000000	-0.012216	-0.017022
Tutoring_Sessions	-0.014282	0.014324	-0.012216	1.000000	0.156525
Exam_Score	0.445455	0.581072	-0.017022	0.156525	1.000000

In []:

In [135... `sns.heatmap(ef_col,annot=True)`
`plt.savefig('finalplot.jpg')`
`plt.figure(figsize=(4,3))`
`plt.show()`



<Figure size 400x300 with 0 Axes>

In []:

Conclusion of this project:-

- 1.Attendance and Studied hours are most affected to exam score

Less affected factors are:-

- 1.Tutorial Session

Badly affected factors are:-

- 1.Sleep Hours

```
In [136... sns.heatmap(ef_col,annot=True)
plt.xticks(rotation=15)
plt.yticks(rotation=75)
plt.savefig('finalplot1.jpg')
# plt.figure(figsize=(4,3))
plt.show()
```

