

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

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

```
In [3]: df.head()
```

```
Out[3]:
```

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Percent
0	male	General	NO	SEBA	AHSEC	ENGLISH	Excel
1	male	OBC	WA	SEBA	AHSEC	OTHERS	Excel
2	male	OBC	OA	OTHERS	CBSE	ENGLISH	Excel
3	male	General	WA	SEBA	AHSEC	OTHERS	Excel
4	male	General	OA	SEBA	CBSE	ENGLISH	Excel

```
In [4]: df.isnull().sum()
```

```
Out[4]: Gender          0
Caste          0
coaching        0
Class_ten_education  0
twelve_education  0
medium          0
Class_X_Percentage  0
Class_XII_Percentage  0
Father_occupation  0
Mother_occupation  0
time           0
Performance      0
dtype: int64
```

In [5]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 666 entries, 0 to 665
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Gender                                666 non-null    object
1   Caste                                666 non-null    object
2   coaching                             666 non-null    object
3   Class_ten_education                  666 non-null    object
4   twelve_education                    666 non-null    object
5   medium                               666 non-null    object
6   Class_X_Percentage                  666 non-null    object
7   Class_XII_Percentage                666 non-null    object
8   Father_occupation                   666 non-null    object
9   Mother_occupation                   666 non-null    object
10  time                                 666 non-null    object
11  Performance                          666 non-null    object
dtypes: object(12)
memory usage: 62.6+ KB
```

In [6]: `df.describe()`

Out[6]:

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Pe
count	666	666	666	666	666	666	
unique	2	4	3	3	3	3	
top	male	General	WA	SEBA	AHSEC	ENGLISH	
freq	355	329	449	396	368	536	

In [7]: `df['Performance'].unique()`

Out[7]: `array(['Excellent', 'Vg', 'Good', 'Average'], dtype=object)`

Create a NumPy array where "Excellent" = 3, "Good" = 2, "Average" = 1, and "Poor" = 0 from the "Performance" column.

In [8]: `df['Performance'].replace(['Excellent', 'Vg', 'Good', 'Average'],[3,2,1,0],inplace=True)`

In [9]: `df.head()`

Out[9]:

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Percent
0	male	General	NO	SEBA	AHSEC	ENGLISH	Excel
1	male	OBC	WA	SEBA	AHSEC	OTHERS	Excel
2	male	OBC	OA	OTHERS	CBSE	ENGLISH	Excel
3	male	General	WA	SEBA	AHSEC	OTHERS	Excel
4	male	General	OA	SEBA	CBSE	ENGLISH	Excel

In [10]: `df['Performance'].unique()`

Out[10]: `array([3, 2, 1, 0], dtype=int64)`

Using NumPy, calculate the mean and standard deviation of students' Class X and Class XII performance after converting "Excellent", "Good", etc., into numerical values.

In [11]: `df['Performance'].replace(['Excellent', 'Vg', 'Good', 'Average'],[3,2,1,0],inplace=True)`

In [12]: `df['Class_X_Percentage'].unique()`

Out[12]: `array(['Excellent', 'Vg', 'Good', 'Average'], dtype=object)`

In [13]: `df1=df.copy()`

In [14]: `df['Class_X_Percentage'].replace(['Excellent', 'Vg', 'Good', 'Average'],[3,2,1,0],inplace=True)`

In [15]: `df['Class_XII_Percentage'].replace(['Excellent', 'Vg', 'Good', 'Average'],[3,2,1,0],inplace=True)`

In [16]: `df.head()`

Out[16]:

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Percent
0	male	General	NO	SEBA	AHSEC	ENGLISH	
1	male	OBC	WA	SEBA	AHSEC	OTHERS	
2	male	OBC	OA	OTHERS	CBSE	ENGLISH	
3	male	General	WA	SEBA	AHSEC	OTHERS	
4	male	General	OA	SEBA	CBSE	ENGLISH	

```
In [17]: df1['Class_X_Percentage'].describe()
```

```
Out[17]: count          666
         unique           4
         top      Excellent
         freq          511
         Name: Class_X_Percentage, dtype: object
```

```
In [18]: df['Class_XII_Percentage'].describe()
```

```
Out[18]: count    666.000000
         mean      2.448949
         std       0.762627
         min       0.000000
         25%       2.000000
         50%       3.000000
         75%       3.000000
         max       3.000000
         Name: Class_XII_Percentage, dtype: float64
```

```
In [19]: def exc(df):
         return df[(df['Class_X_Percentage'] == 3) & (df['Class_XII_Percentage'] ==
```

```
In [20]: filtered_df = exc(df)
         filtered_df
```

```
Out[20]:
```

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Percentage
0	male	General	NO	SEBA	AHSEC	ENGLISH	
1	male	OBC	WA	SEBA	AHSEC	OTHERS	
2	male	OBC	OA	OTHERS	CBSE	ENGLISH	
3	male	General	WA	SEBA	AHSEC	OTHERS	
4	male	General	OA	SEBA	CBSE	ENGLISH	
...
627	female	ST	WA	SEBA	AHSEC	ENGLISH	
630	male	ST	NO	SEBA	AHSEC	ENGLISH	
634	male	ST	WA	SEBA	AHSEC	ENGLISH	
635	female	ST	NO	CBSE	CBSE	ENGLISH	
654	male	ST	WA	CBSE	CBSE	ENGLISH	

380 rows × 12 columns



Find the average Class XII percentage for students based on "Father_occupation".

```
In [21]: df['Class_XII_Percentage'].groupby(df['Father_occupation']).mean()
```

```
Out[21]: Father_occupation  
BANK_OFFICIAL      2.434783  
BUSINESS            2.368932  
COLLEGE_TEACHER    2.592593  
CULTIVATOR          2.074074  
DOCTOR              2.854545  
ENGINEER            2.688889  
OTHERS              2.458484  
SCHOOL_TEACHER     2.256881  
Name: Class_XII_Percentage, dtype: float64
```

Find all female students who studied in "ENGLISH" medium and had "Excellent" performance.

```
In [29]: df[(df['Gender']=='female')&(df['medium']=='ENGLISH')&(df['Performance']==3)]
```

Out[29]:

	Gender	Caste	coaching	Class_ten_education	twelve_education	medium	Class_X_Perce
6	female	General	OA	CBSE	CBSE	ENGLISH	
8	female	General	NO	SEBA	AHSEC	ENGLISH	
9	female	General	OA	CBSE	CBSE	ENGLISH	
12	female	General	WA	SEBA	AHSEC	ENGLISH	
17	female	General	OA	CBSE	CBSE	ENGLISH	
25	female	General	WA	SEBA	AHSEC	ENGLISH	
26	female	General	WA	SEBA	AHSEC	ENGLISH	
28	female	General	WA	SEBA	AHSEC	ENGLISH	
30	female	General	NO	CBSE	CBSE	ENGLISH	
32	female	General	NO	SEBA	AHSEC	ENGLISH	
35	female	General	OA	CBSE	AHSEC	ENGLISH	
40	female	General	WA	SEBA	AHSEC	ENGLISH	
43	female	General	OA	OTHERS	CBSE	ENGLISH	
46	female	General	NO	CBSE	CBSE	ENGLISH	
47	female	General	WA	CBSE	CBSE	ENGLISH	
48	female	General	WA	CBSE	CBSE	ENGLISH	
53	female	General	WA	OTHERS	CBSE	ENGLISH	
61	female	General	WA	CBSE	CBSE	ENGLISH	
63	female	General	WA	CBSE	CBSE	ENGLISH	
64	female	General	WA	SEBA	AHSEC	ENGLISH	
68	female	General	WA	CBSE	CBSE	ENGLISH	
69	female	General	OA	CBSE	CBSE	ENGLISH	
78	female	General	WA	CBSE	CBSE	ENGLISH	
80	female	General	WA	CBSE	CBSE	ENGLISH	
84	female	OBC	WA	SEBA	CBSE	ENGLISH	
87	female	OBC	WA	CBSE	CBSE	ENGLISH	
88	female	General	WA	CBSE	CBSE	ENGLISH	
95	female	General	WA	CBSE	CBSE	ENGLISH	
97	female	OBC	WA	SEBA	AHSEC	ENGLISH	
99	female	OBC	WA	SEBA	AHSEC	ENGLISH	
100	female	General	WA	SEBA	AHSEC	ENGLISH	

Does attending coaching (coaching = "YES") correlate with "Excellent" performance? Count how many students with and without coaching scored

```
In [35]: df[(df['coaching']=='YES')&(df['Performance']==3)]
```

Out[35]:

```
In [44]: df[(df['Performance']==3)&(df['coaching']=='NO')]['Performance'].value counts()
```

```
Out[44]: 3      20
          Name: Performance, dtype: int64
```

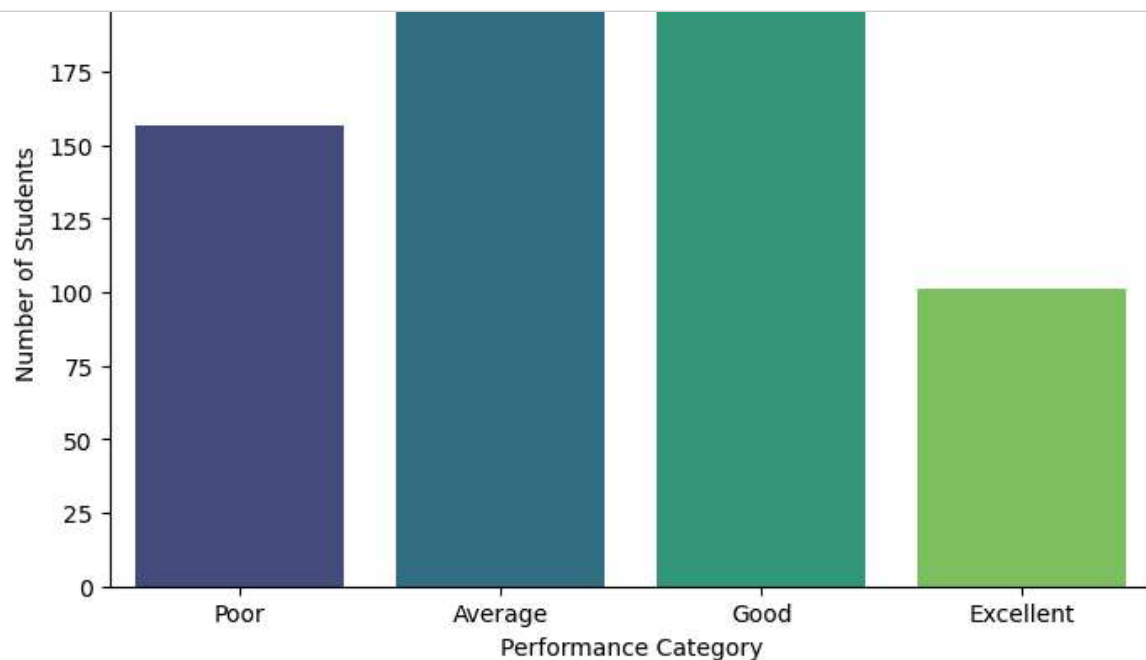
Create a pivot table showing the average "Class_XII_Percentage" by "Caste" and "medium" of instruction.

```
In [52]: pivot_table = df.pivot_table(
    values="Class_XII_Percentage", # Value to average
    index="Caste",                  # Row categories
    columns="medium",               # Column categories
    aggfunc="mean"                 # Aggregation function
)

# Display the pivot table
print(pivot_table)
```


Create a bar chart showing the number of students in each "Performance" category.

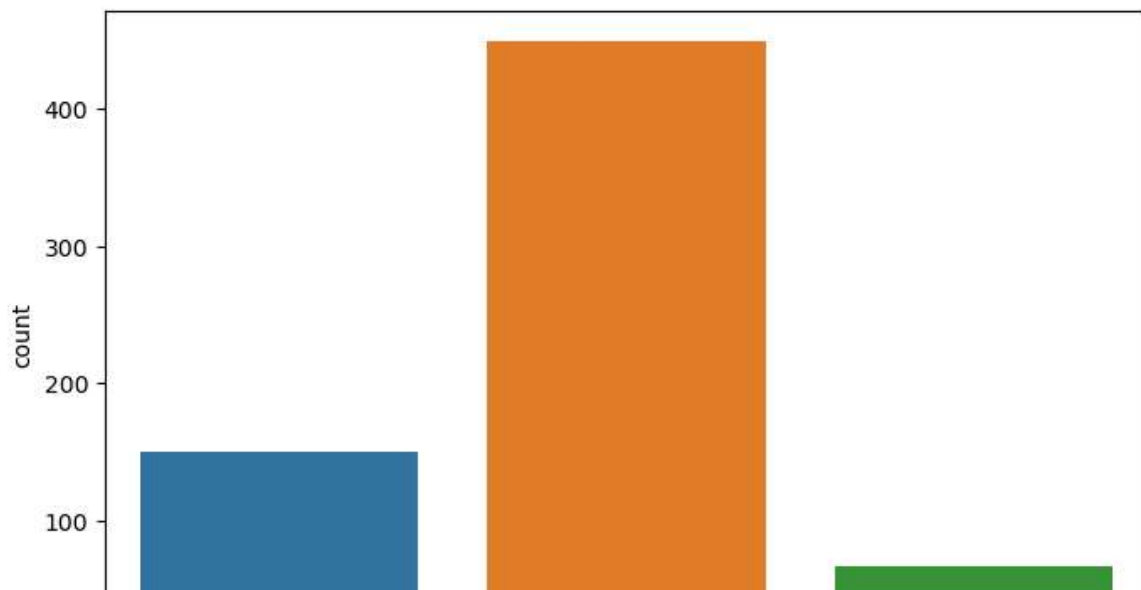
```
In [55]: plt.figure(figsize=(8, 5))
sns.countplot(data=df, x="Performance", palette="viridis")
plt.xlabel("Performance Category")
plt.ylabel("Number of Students")
plt.title("Performance Distribution of Students")
plt.xticks(ticks=[0, 1, 2, 3], labels=["Poor", "Average", "Good", "Excellent"])
plt.show()
```



What does the count plot of the 'coaching' column reveal about the distribution of students

```
In [66]: plt.figure(figsize=(8,5))  
sns.countplot(data=df,x=df['coaching'])
```

```
Out[66]: <Axes: xlabel='coaching', ylabel='count'>
```



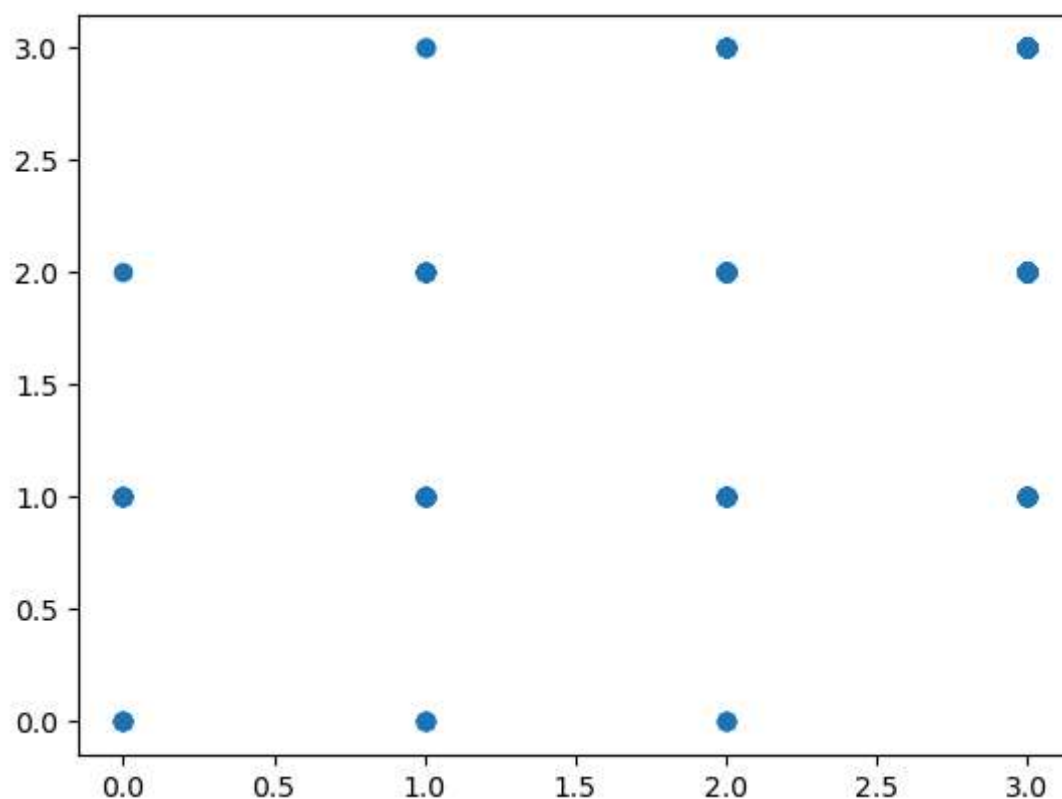
What relationship does the scatter plot between tenth-grade and twelfth-grade scores

```
In [61]: tenth=df['Class_X_Percentage']
```

```
In [63]: twelve=df['Class_XII_Percentage']
```

```
In [64]: plt.scatter(tenth,twelve)
```

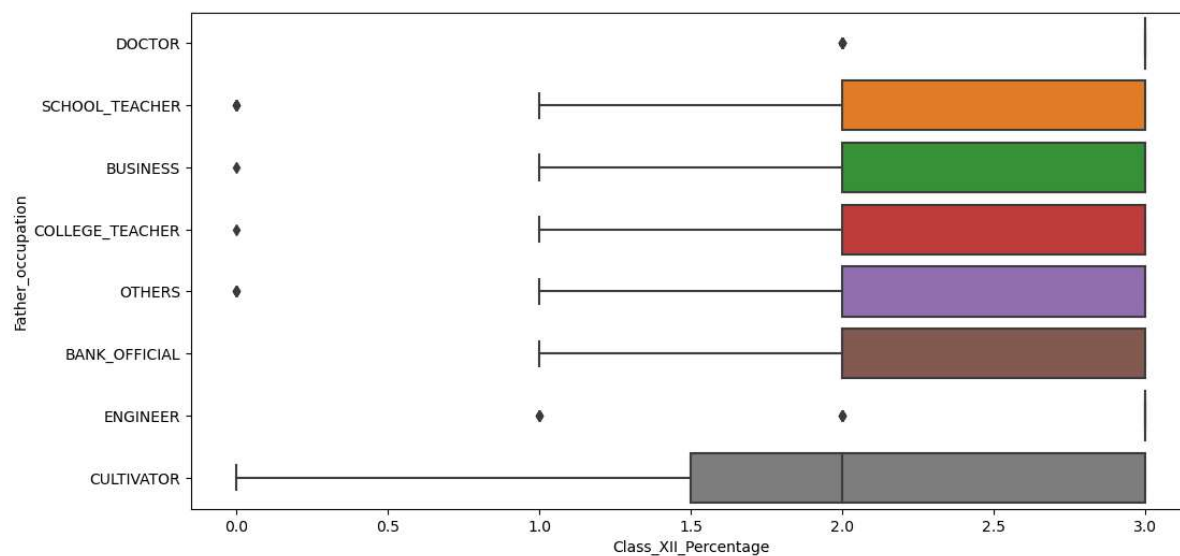
```
Out[64]: <matplotlib.collections.PathCollection at 0x1e0ce999810>
```



How does the box plot illustrate the relationship between students

```
In [69]: plt.figure(figsize=(12, 6))
sns.boxplot(x=df['Class_XII_Percentage'],y=df['Father_occupation'])
```

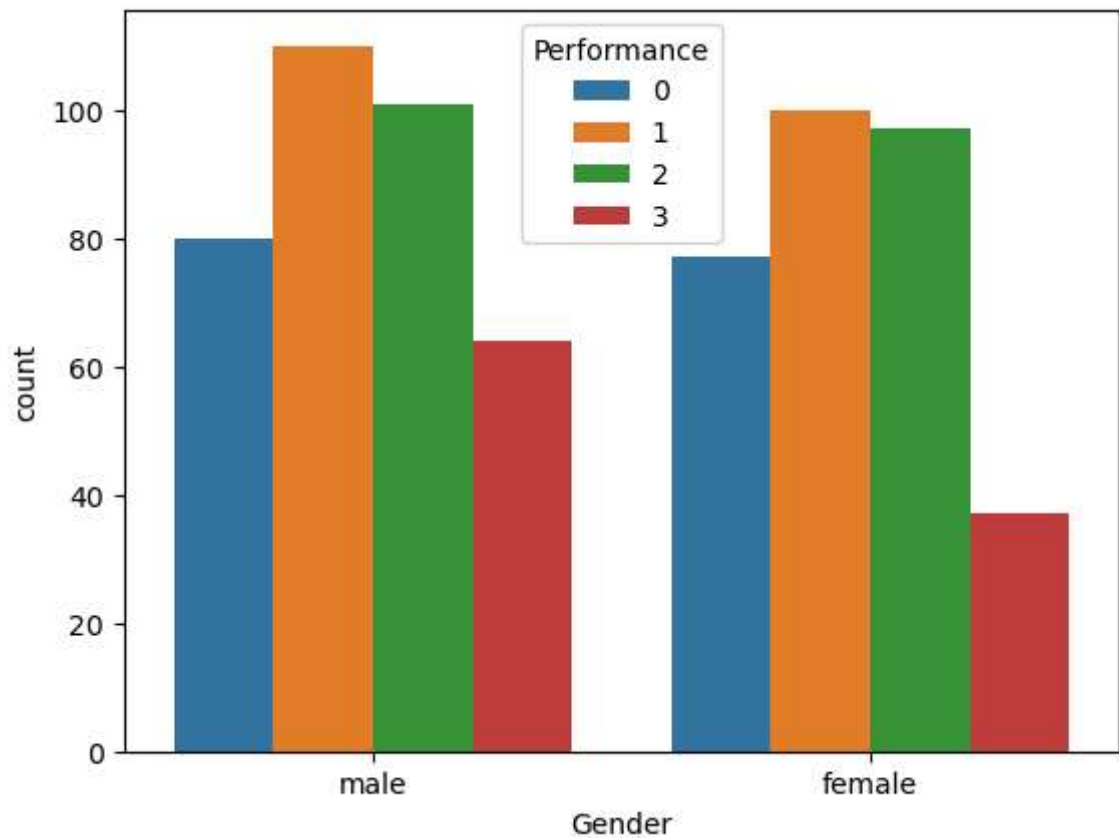
```
Out[69]: <Axes: xlabel='Class_XII_Percentage', ylabel='Father_occupation'>
```



Use a countplot to show the number of male and female students in each "Performance" category.

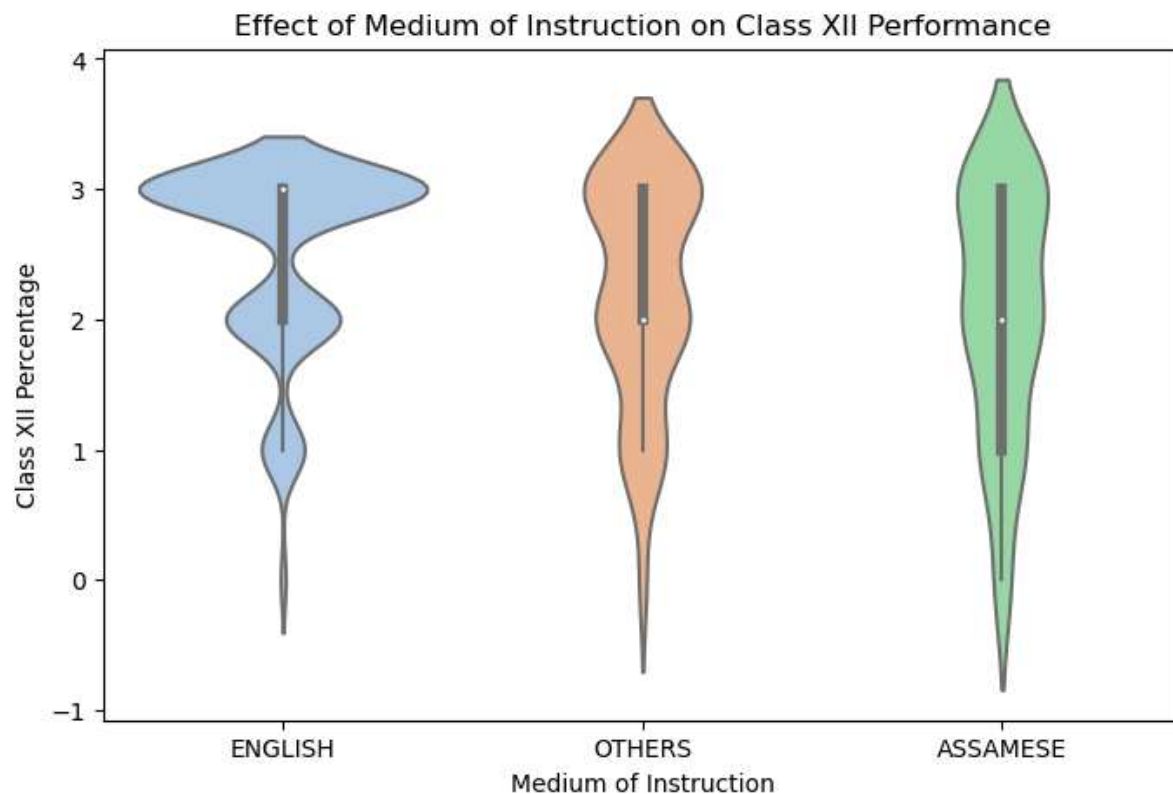
```
In [71]: sns.countplot(x=df['Gender'], hue=df['Performance'])
```

```
Out[71]: <Axes: xlabel='Gender', ylabel='count'>
```



```
In [76]: plt.figure(figsize=(8, 5))
sns.violinplot(data=df, x="medium", y="Class_XII_Percentage", palette="pastel")
plt.xlabel("Medium of Instruction")
plt.ylabel("Class XII Percentage")
plt.title("Effect of Medium of Instruction on Class XII Performance")

# Show the plot
plt.show()
```



In []:

In []: