

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [14]:

```
df = pd.read_excel("C:\Data Analytics\Data analyst Data.xlsx")
df.head()
```

Out[14]:

	First Name	Email ID	Quantity	Events	Attendee Status	College Name	How did you come to know about this event?	Specify in "Others" (how did you come to know about this event)	Designation	Year of Graduation
0	ANIKET	aniket@xyz.com	1	Art of Resume Building	Attending	D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKUR...	Email	NaN	Students	2020
1	Dhanshree	dhanshree@xyz.com	1	Art of Resume Building	Attending	AP SHAH INSTITUTE OF TECHNOLOGY	Others	College	Students	2020
2	Dhiraj	dhiraj@xyz.com	1	Art of Resume Building	Attending	Don Bosco College of Engineering Fatorda Goa	Email	NaN	Students	2020
3	Pooja	pooja@xyz.com	1	Art of Resume Building	Attending	Pillai College of Engineering New Panvel	Email	NaN	Students	2020
4	Aayush	aayush@xyz.com	1	Art of Resume Building	Attending	St Xavier's College	Instagram LinkedIn Cloud Counselage Website	NaN	Students	2020

In [15]:

```
df.columns
```

Out[15]:

```
Index(['First Name', 'Email ID', 'Quantity', 'Events', 'Attendee Status',
      'College Name', 'How did you come to know about this event?',
      'Specify in "Others" (how did you come to know about this event)',
      'Designation', 'Year of Graduation', 'City', 'CGPA',
      'Experience with python (Months)', 'Family Income',
      'Expected salary (Lac)', 'Leadership- skills'],
      dtype='object')
```

In [20]:

```
# Distribution of students across different graduation years

graduation_year_distribution = df['Year of Graduation'].value_counts()
print(graduation_year_distribution)
```

```
Year of Graduation
2023    1536
2024    1511
2025    1292
2026     555
Name: count, dtype: int64
```

In [21]:

```
#Average family income of the student

income_mapping = {
    '0-2 Lakh': 1,
    '2-5 Lakh': 3.5,
    '5-7 Lakh': 6,
    '7 Lakh+': 7
}

df['Family Income (Numeric)'] = df['Family Income'].map(income_mapping)
average_family_income = df['Family Income (Numeric)'].mean()
print(f"Average Family Income: {average_family_income} Lakh")
```

Average Family Income: 1.2930118512464241 Lakh

In [28]:

```
#How many unique students are in the dataset

no_of_unique_students = df['Email ID'].nunique()
print(f"Number of unique students: {no_of_unique_students}")

unique_students = df['Email ID'].unique()
print("Unique students:", unique_students)
```

Number of unique students: 2157

Unique students: ['aniket@xyz.com' 'dhanshree@xyz.com' 'dhiraj@xyz.com' ...
'suryaprakash@xyz.com' 'saifali@xyz.com' 'lankesh@xyz.com']

In [29]:

```
#Average GPA of the students

average_gpa = df['CGPA'].mean()
print(f"Average GPA: {average_gpa}")
```

Average GPA: 8.038475684511647

In [31]:

```
#Are there any outliers in the quantity attribute

quantity_outliers = df['Quantity'].describe()
print(quantity_outliers)
```

```
count    4894.0
mean       1.0
std        0.0
min        1.0
25%        1.0
50%        1.0
75%        1.0
max        1.0
Name: Quantity, dtype: float64
```

In [32]:

```
#Distribution of students' experience with Python programming

experience_distribution = df['Experience with python (Months)'].value_counts()
print(experience_distribution)
```

Experience with python (Months)

```
5      1242
3      1008
8       800
6       738
7       640
4       466
Name: count, dtype: int64
```

In [34]:

```
#Average GPA of the students from each city

average_gpa_city = df.groupby('City')['CGPA'].mean()
print(average_gpa_city)
```

```
City
Agartala      7.660714
Agra          8.046429
Ahemdabad     8.190385
Ajmer         8.284314
Akola         8.021429
...
Vidisha       7.738095
Vijaywada     7.986364
Wardha        8.328571
konark        8.071429
kullu         7.878571
Name: CGPA, Length: 177, dtype: float64
```

In [33]:

```
#Can we identify any relationship between family income and GPA

relationship_income_gpa = df[['Family Income (Numeric)', 'CGPA']].corr()
print(relationship_income_gpa)
```

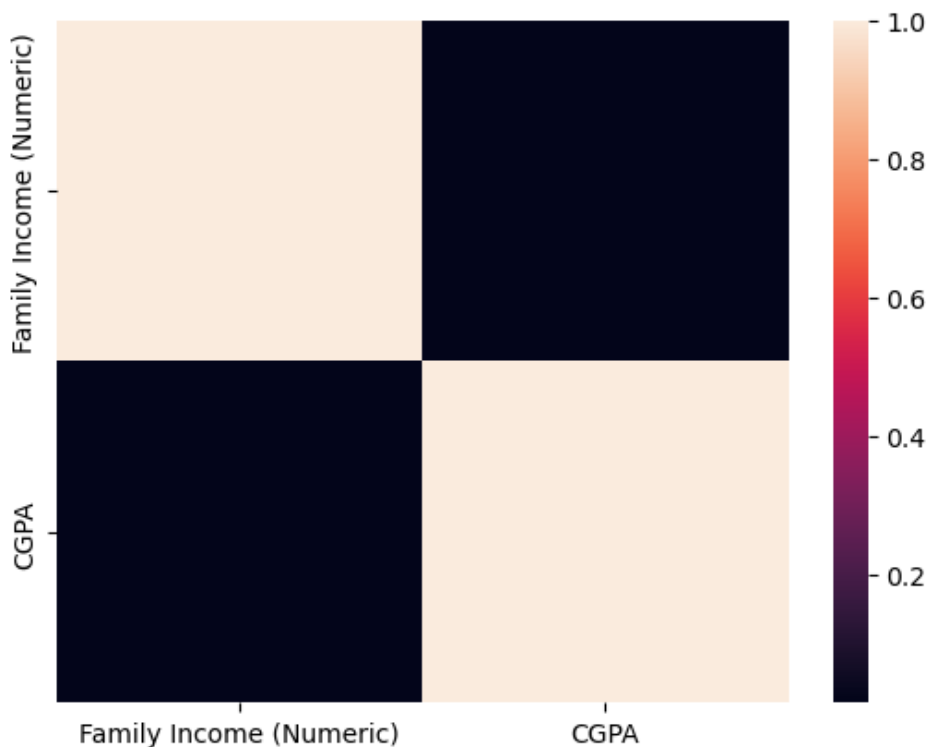
```
                Family Income (Numeric)    CGPA
Family Income (Numeric)    1.000000  0.015311
CGPA                      0.015311  1.000000
```

In [35]:

```
sns.heatmap(relationship_income_gpa)
```

Out[35]:

<Axes: >

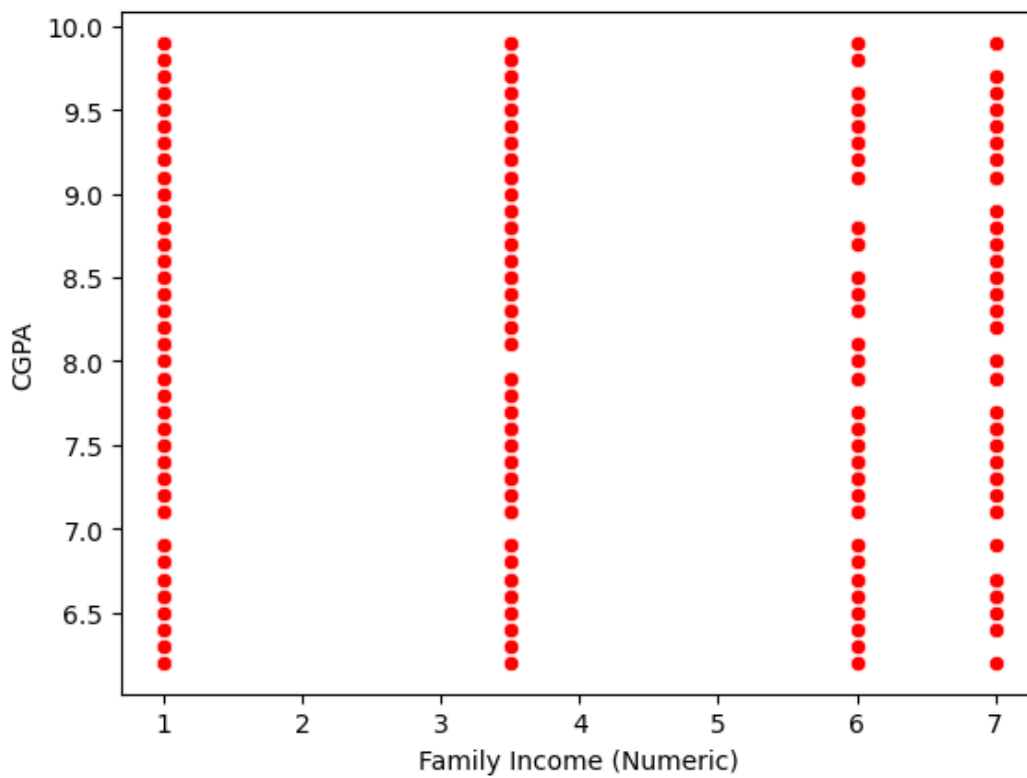


In [38]:

```
sns.scatterplot(x='Family Income (Numeric)', y='CGPA', data=df, color='red')  
plt.plot()
```

Out[38]:

[]



In []: