**Name: Dharavath Sai Kiran**

**DATA ANALYTICS PROJECT**

**Github Link:** [**https://github.com/betechie/IAC-Data-Analytics-Intern-Project**](https://github.com/betechie/IAC-Data-Analytics-Intern-Project)

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.ticker as ticker

import statsmodels.api as sm

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.metrics import confusion\_matrix

from mpl\_toolkits.mplot3d import Axes3D

import plotly.express as px

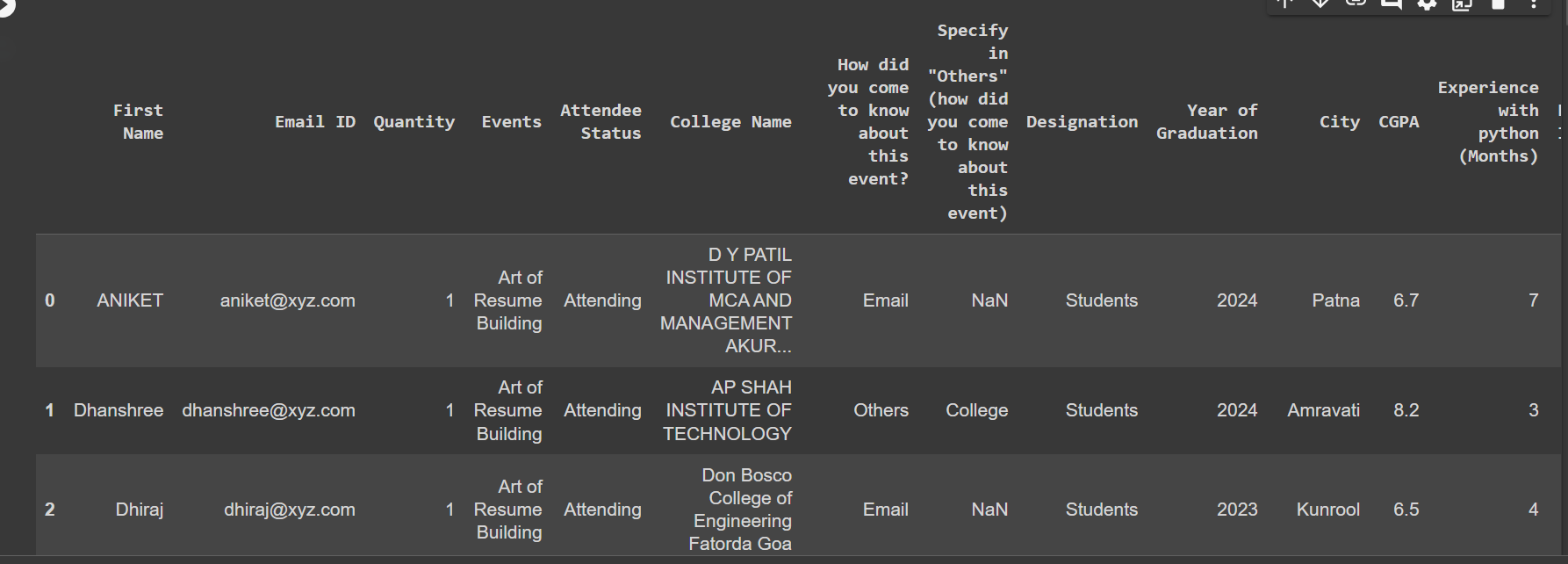
# Display the Dataset

import pandas as pd

import numpy as np

d = pd.read\_excel(r"/content/Data analyst Data.xlsx")

d.head()



column\_names = d.columns.tolist()

column\_names

['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']

1 . How many Unique Students are present in the dataset?

unique\_students = d['First Name'].unique()

for student\_name in unique\_students:

    print(student\_name)

ANIKET

Dhanshree

Dhiraj

Pooja

Aayush

Mrunali

Durga

Ruchit

Mayuresh

ROHIT

Nutan

Yogesh

Pranali

Gaurang

Saba

Apurva

Tanmay

SUMIT

Shubham

Shreyas

Shivani

Swapnil

Manasi

Hemant

Gaurav

Harshal

Jitesh

2. What is the average GPA of the students ?

r = d['CGPA'].mean()

print("Mean of CGPA: ",r)

Mean of CGPA: 8.038475684511647

3. What is the distribution of students across different graduation years ?

graduation = d['Year of Graduation'].unique()

print(graduation)

[2024 2023 2025 2026]

graduation\_year\_counts = d['Year of Graduation'].value\_counts()

plt.figure(figsize=(10, 6))

sns.set(style="whitegrid")

sns.barplot(x=graduation\_year\_counts.index, y=graduation\_year\_counts.values, palette="viridis")

plt.xlabel('Graduation Year')

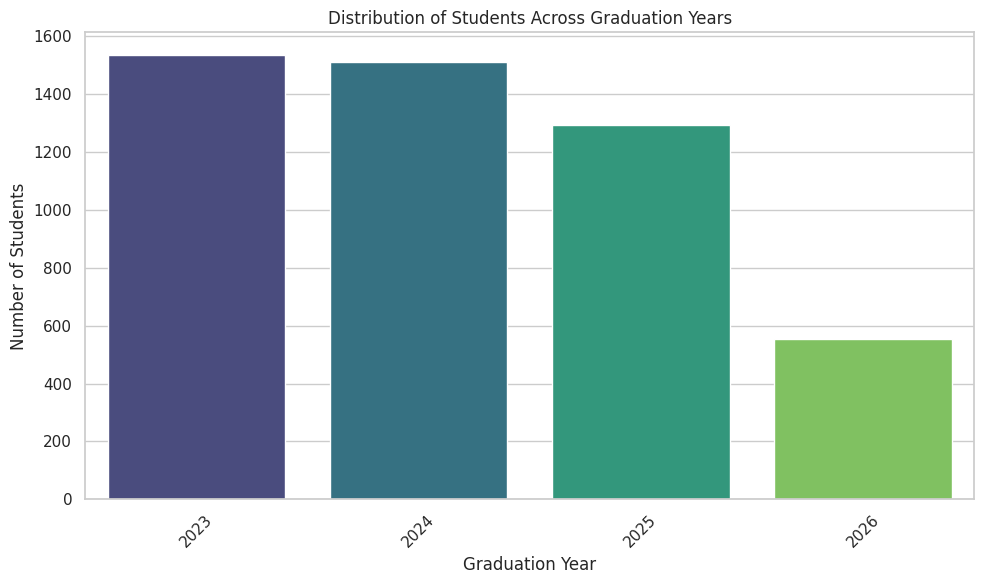
plt.ylabel('Number of Students')

plt.title('Distribution of Students Across Graduation Years')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



4. What is the distribution of student’s experience with Python Programming ?

students\_df = d[d['Designation'] == 'Students']

experience\_counts = students\_df['Experience with python (Months)'].value\_counts()

plt.figure(figsize=(10, 6))

sns.set(style="whitegrid")

ax = sns.barplot(x=experience\_counts.index, y=experience\_counts.values, palette="magma")

plt.xlabel('Experience with Python (Months)')

plt.ylabel('Number of Students')

plt.title('Distribution of Students\' Experience with Python Programming')

plt.xticks(rotation=45)

plt.tight\_layout()

def format\_func(value, tick\_number):

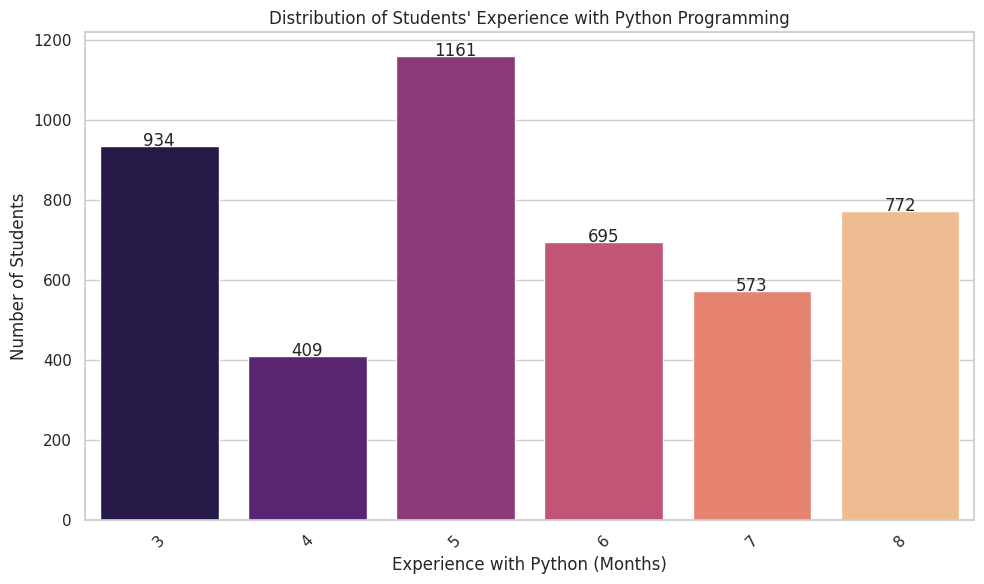
    return f'{int(value)}'

ax.yaxis.set\_major\_formatter(ticker.FuncFormatter(format\_func))

for p in ax.patches:

    ax.text(p.get\_x() + p.get\_width() / 2., p.get\_height(), f'{int(p.get\_height())}', ha="center")

plt.show()



5. What is the average family income of the student ?

# Preprocess the Family Income column

def preprocess\_income\_range(income\_range):

    numeric\_part = ''.join(filter(str.isdigit, income\_range))

    return int(numeric\_part)

d['Family Income'] = d['Family Income'].apply(preprocess\_income\_range)

# Calculate the average family income

average\_family\_income = d[d['Designation'] == 'Students']['Family Income'].mean()

print(f"The average family income of students is: {average\_family\_income:.2f}")

# Create a futuristic-style bar plot

plt.figure(figsize=(10, 6))

sns.set\_style("darkgrid")

sns.barplot(x=["Average Family Income"], y=[average\_family\_income], palette="plasma")

plt.ylabel('Family Income')

plt.title('Average Family Income of Students')

plt.tight\_layout()

# Show the futuristic-style plot

plt.show()

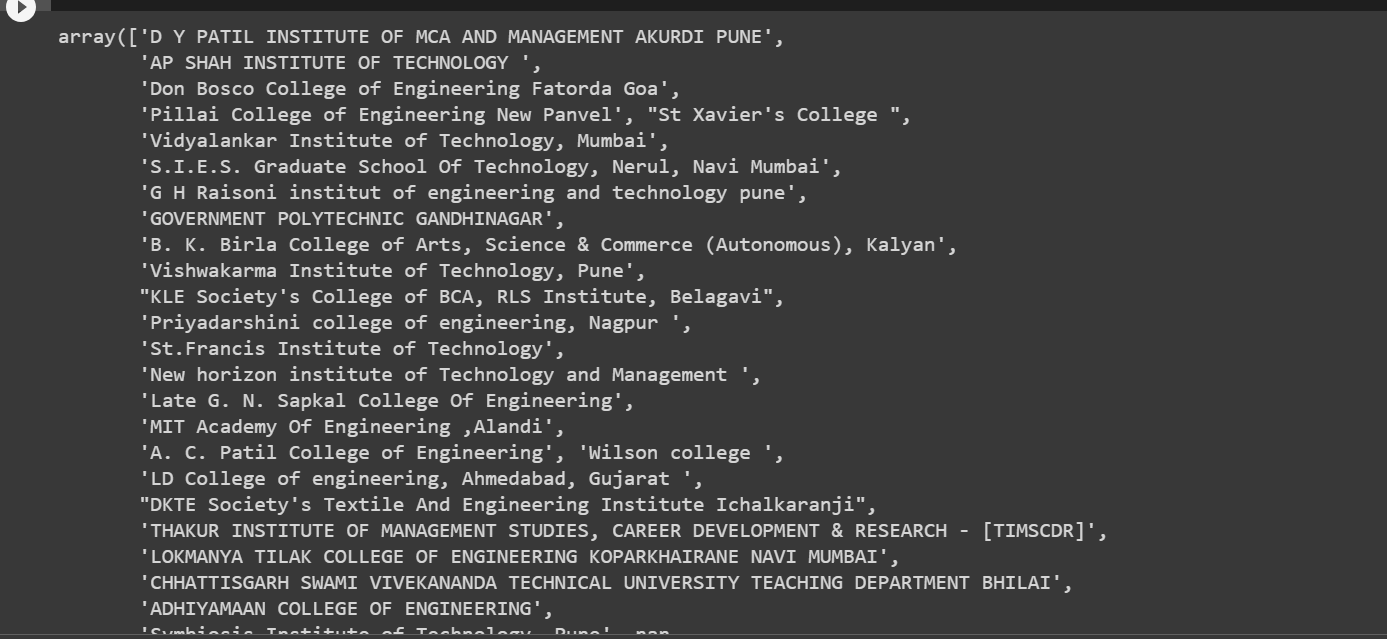
The average family income of students is: 3.79



6. How does the GPA vary among different Colleges ?

collegeName = d["College Name"].unique()

collegeName



top\_colleges = d.groupby('College Name')['CGPA'].mean()

plt.figure(figsize=(10, 6))

sns.set\_style("dark")

sns.barplot(x=top\_colleges.index, y=top\_colleges.values, palette="plasma")

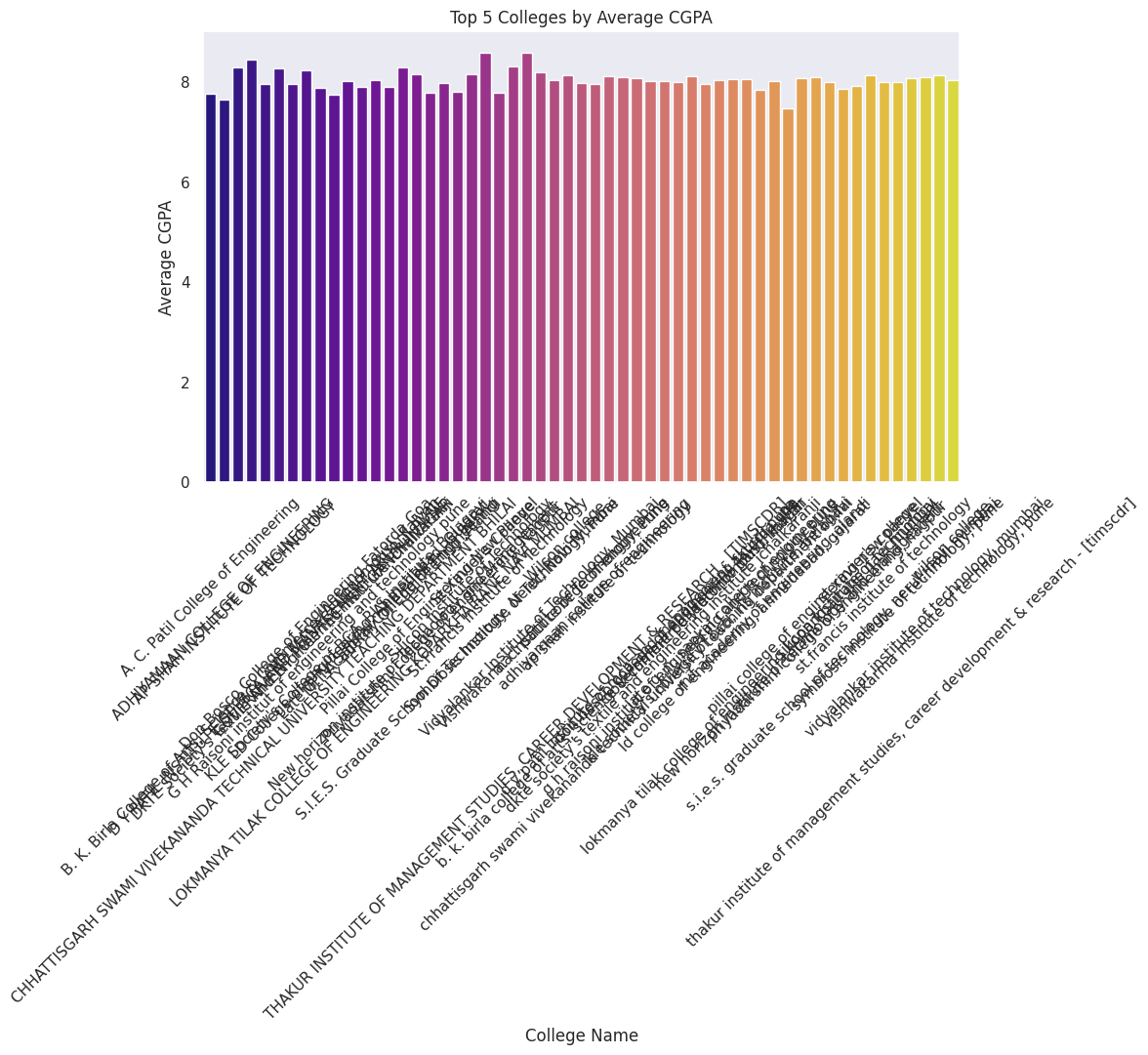
plt.xlabel('College Name')

plt.ylabel('Average CGPA')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



top\_colleges = d.groupby('College Name')['CGPA'].mean().nlargest(5)

plt.figure(figsize=(10, 6))

sns.set\_style("dark")

sns\_barplot = sns.barplot(x=top\_colleges.index, y=top\_colleges.values, palette="plasma")

plt.xlabel('College Name')

plt.ylabel('Average CGPA')

plt.title('Top 5 Colleges by Average CGPA')

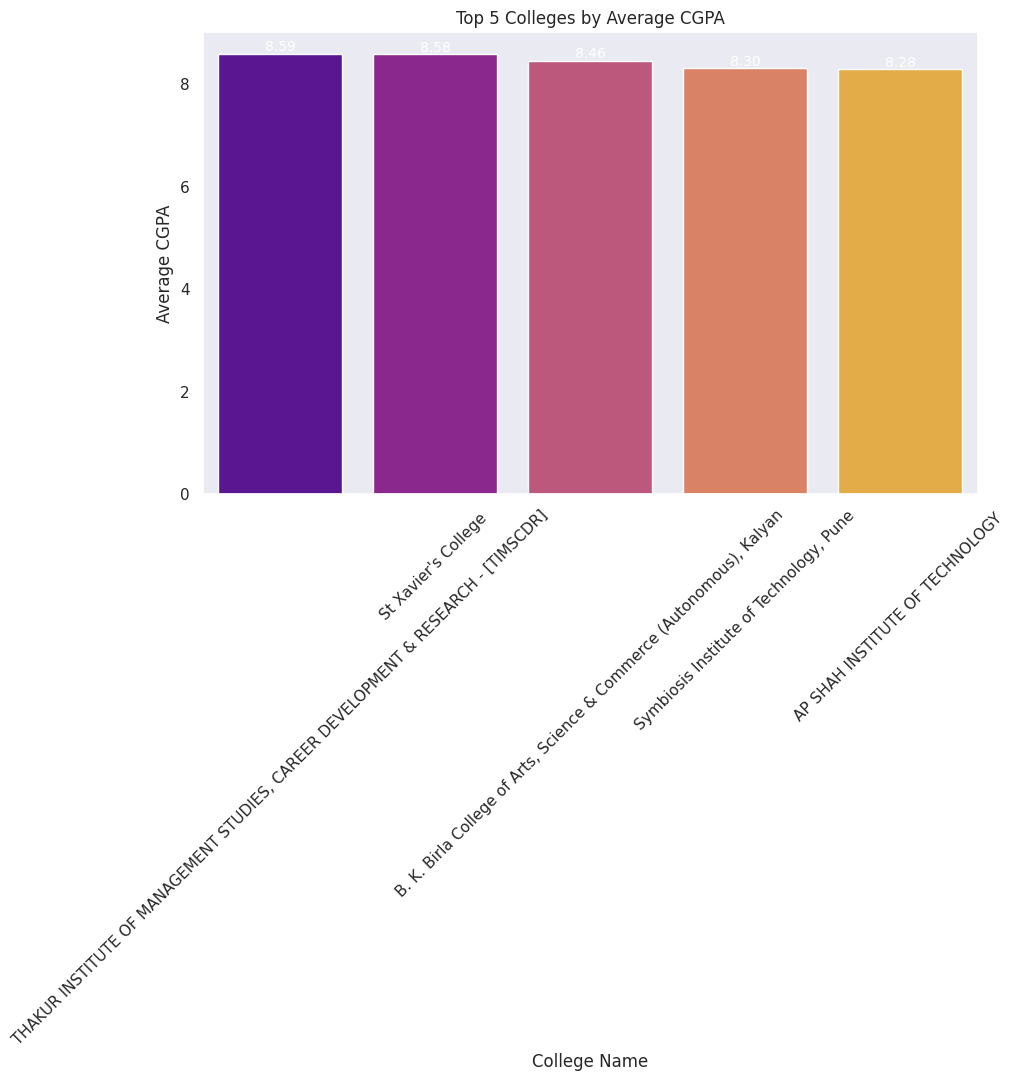
plt.xticks(rotation=45)

for index, value in enumerate(top\_colleges.values):

    sns\_barplot.text(index, value, f'{value:.2f}', ha='center', va='bottom', fontsize=10, color='white')

plt.tight\_layout()

plt.show()



7. Are there any outliers in the quantity attribute?

No, there are no outliers in the quantity attribute.

Q1 = d['Quantity'].quantile(0.25)

Q3 = d['Quantity'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

outliers = d[(d['Quantity'] < lower\_bound) | (d['Quantity'] > upper\_bound)]

print("Potential Outliers:")

print(outliers)

plt.figure(figsize=(8, 6))

sns.boxplot(x=d['Quantity'])

plt.xlabel('Number of Courses Completed')

plt.title('Distribution of Number of Courses Completed (Quantity)')

plt.tight\_layout()

plt.show()

Potential Outliers:

Empty DataFrame

Columns: [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]

Index: []



8. What is the average GPA for students from each city ?

average = d[d['Designation'] == 'Students'].groupby('City')['CGPA'].mean().reset\_index()

print("Average CGPA for Students from Each City:")

print(average)

Average CGPA for Students from Each City:

City CGPA

0 Agartala 7.687500

1 Agra 8.000000

2 Ahemdabad 8.212245

3 Ajmer 8.271429

4 Akola 8.013514

.. ... ...

172 Vidisha 7.735897

173 Vijaywada 7.995000

174 Wardha 8.328571

175 konark 7.984615

176 kullu 7.933333

[177 rows x 2 columns]

plt.figure(figsize=(12, 6))

plt.bar(d['City'], d['CGPA'],  color='b')

plt.xlabel('City')

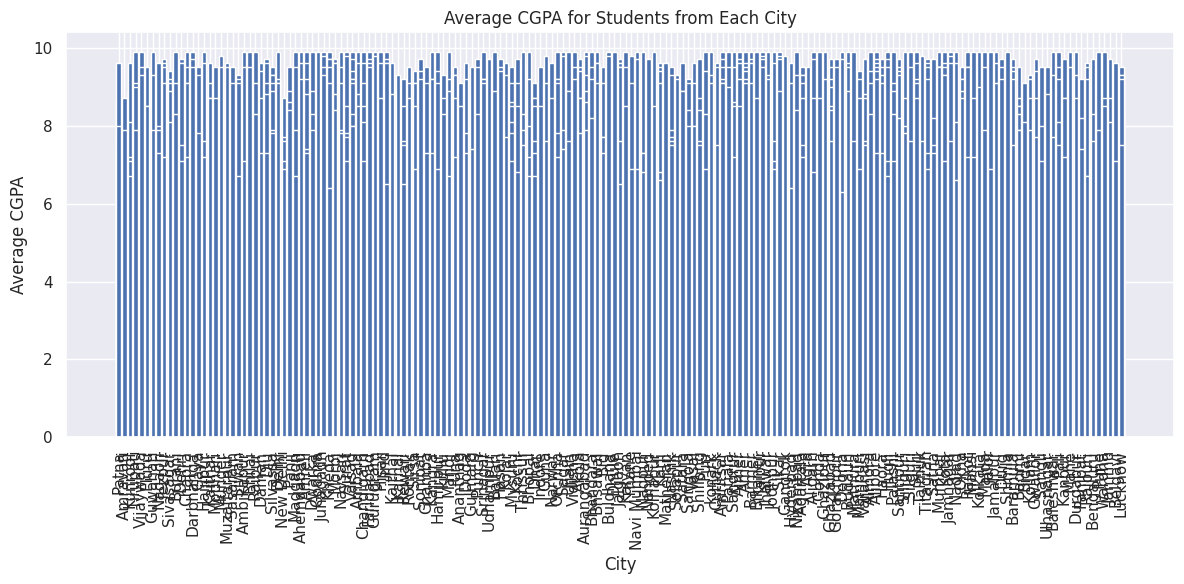
plt.ylabel('Average CGPA')

plt.title('Average CGPA for Students from Each City')

plt.xticks(rotation=90)

plt.grid(True)

plt.tight\_layout()



9. Can we identify any Relationship between Family Income and GPA.

students\_df = d[d['Designation'] == 'Students']

plt.figure(figsize=(10, 6))

sns.scatterplot(data=students\_df, x='Family Income', y='CGPA')

plt.xlabel('Family Income')

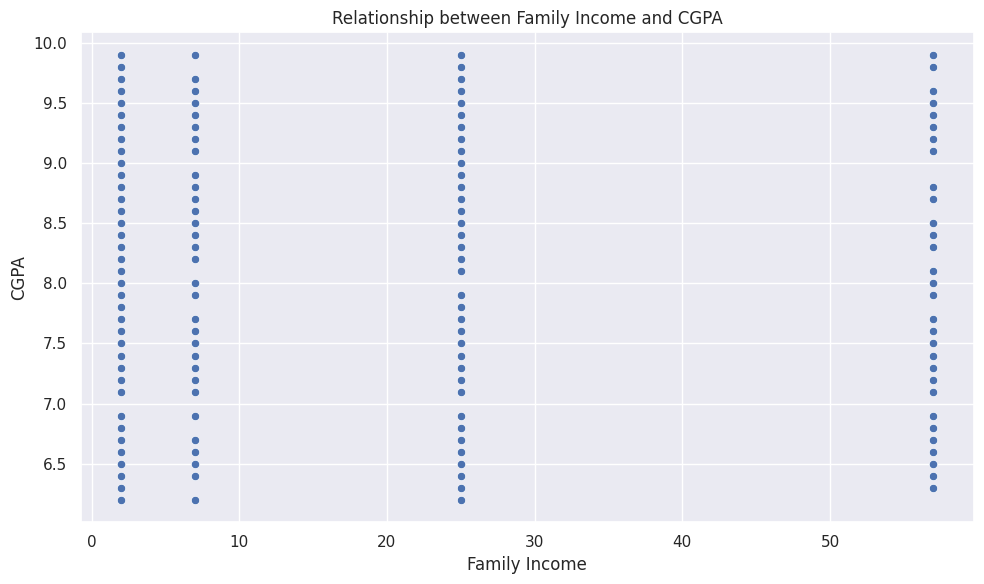
plt.ylabel('CGPA')

plt.title('Relationship between Family Income and CGPA')

plt.grid(True)

plt.tight\_layout()

plt.show()



correlation\_matrix = students\_df[['Family Income', 'CGPA']].corr()

plt.figure(figsize=(8, 6))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', center=0)

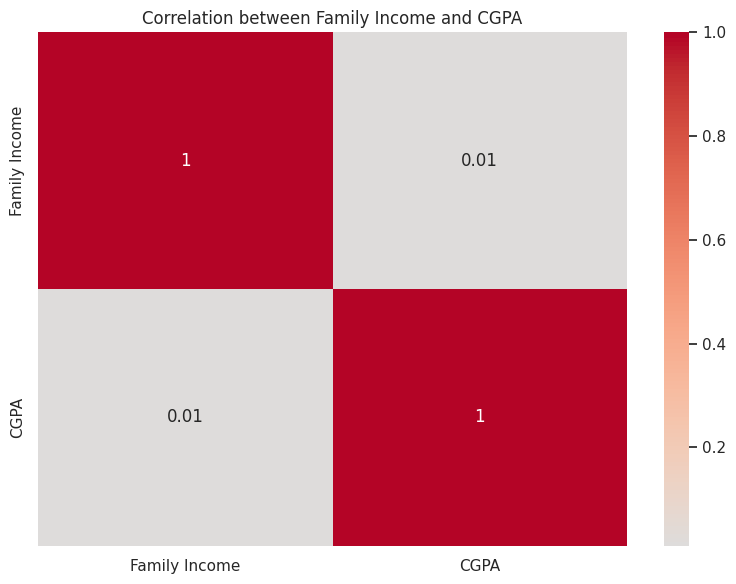
plt.title('Correlation between Family Income and CGPA')

plt.tight\_layout()

plt.show()

correlation = students\_df['Family Income'].corr(students\_df['CGPA'])

print(f"Correlation between Family Income and CGPA: {correlation:.2f}")



correlation\_matrix = students\_df[['Family Income', 'CGPA']].corr()

plt.figure(figsize=(10, 6))

sns.regplot(x='Family Income', y='CGPA', data=students\_df, scatter\_kws={'s': 10}, line\_kws={'color': 'red'})

plt.title('Relationship Between Family Income and CGPA')

plt.xlabel('Family Income')

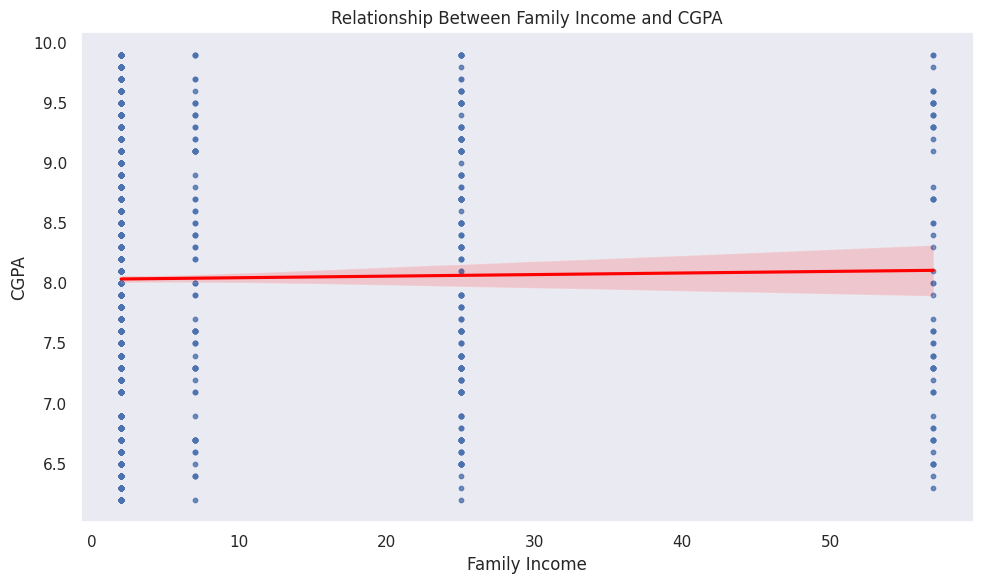
plt.ylabel('CGPA')

plt.tight\_layout()

plt.show()

correlation = students\_df['Family Income'].corr(students\_df['CGPA'])

print(f"Correlation between Family Income and CGPA: {correlation:.2f}")

Correlation between Family Income and CGPA: 0.01

**MODERATE QUESTIONS**

11. How does the expected salary vary based on factors like 'GPA' , 'Family Income' and 'Expected Salary'

# Create scatter plots or regression plots to visualize the relationships

plt.figure(figsize=(16, 6))

# Scatter plot: Expected Salary vs GPA

plt.subplot(1, 3, 1)

sns.scatterplot(data= d, x='CGPA', y='Expected salary (Lac)')

plt.title('Expected Salary vs GPA')

plt.xlabel('GPA')

plt.ylabel('Expected Salary')

# Scatter plot: Expected Salary vs Family Income

plt.subplot(1, 3, 2)

sns.scatterplot(data=d, x='Family Income', y='Expected salary (Lac)')

plt.title('Expected Salary vs Family Income')

plt.xlabel('Family Income')

plt.ylabel('Expected Salary')

# Scatter plot: Expected Salary vs Experience with Python

plt.subplot(1, 3, 3)

sns.scatterplot(data=d, x='Experience with python (Months)',

y='Expected salary (Lac)')

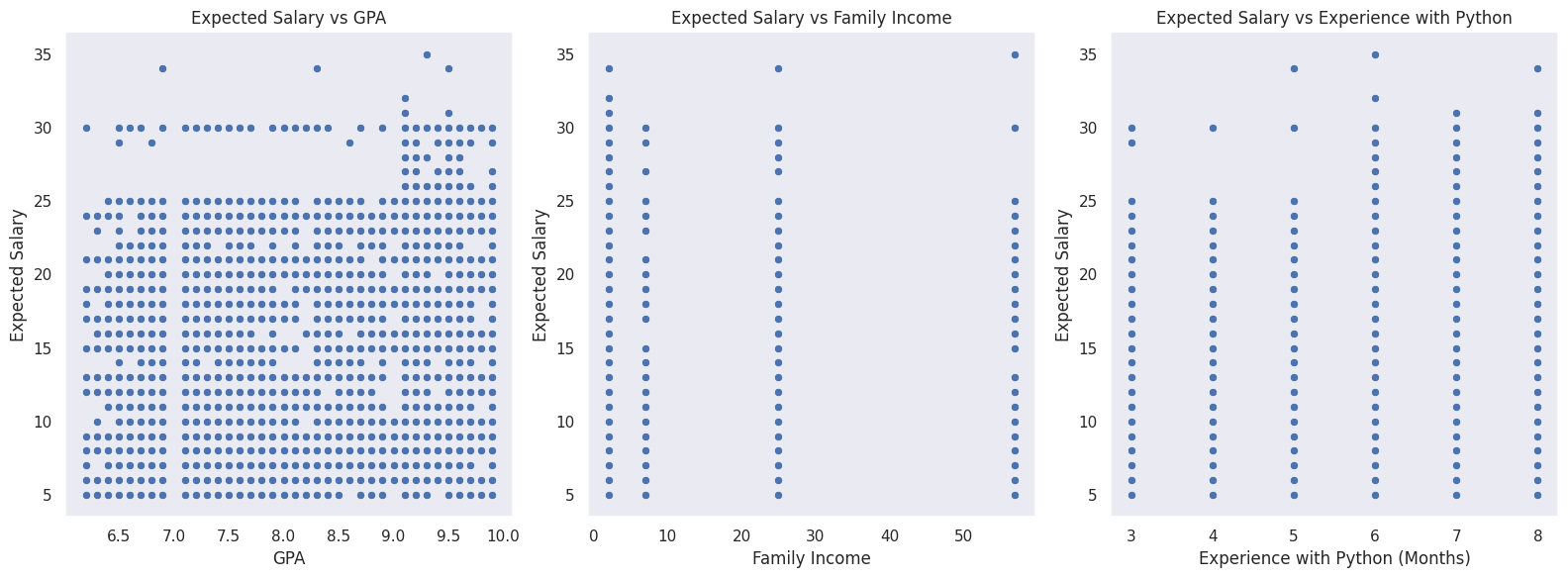
plt.title('Expected Salary vs Experience with Python')

plt.xlabel('Experience with Python (Months)')

plt.ylabel('Expected Salary')

plt.tight\_layout()

plt.show()



12. Which event tend to attract more students from specific fields of study?

# Create a count plot to visualize the distribution of attendees from different fields of study for each event

plt.figure(figsize=(12, 6))

sns.countplot(data=d, x='Events', hue='Designation') # Replace 'Designation' with the correct column name

plt.title('Distribution of Students\' Fields of Study for Each Event')

plt.xlabel('Event')

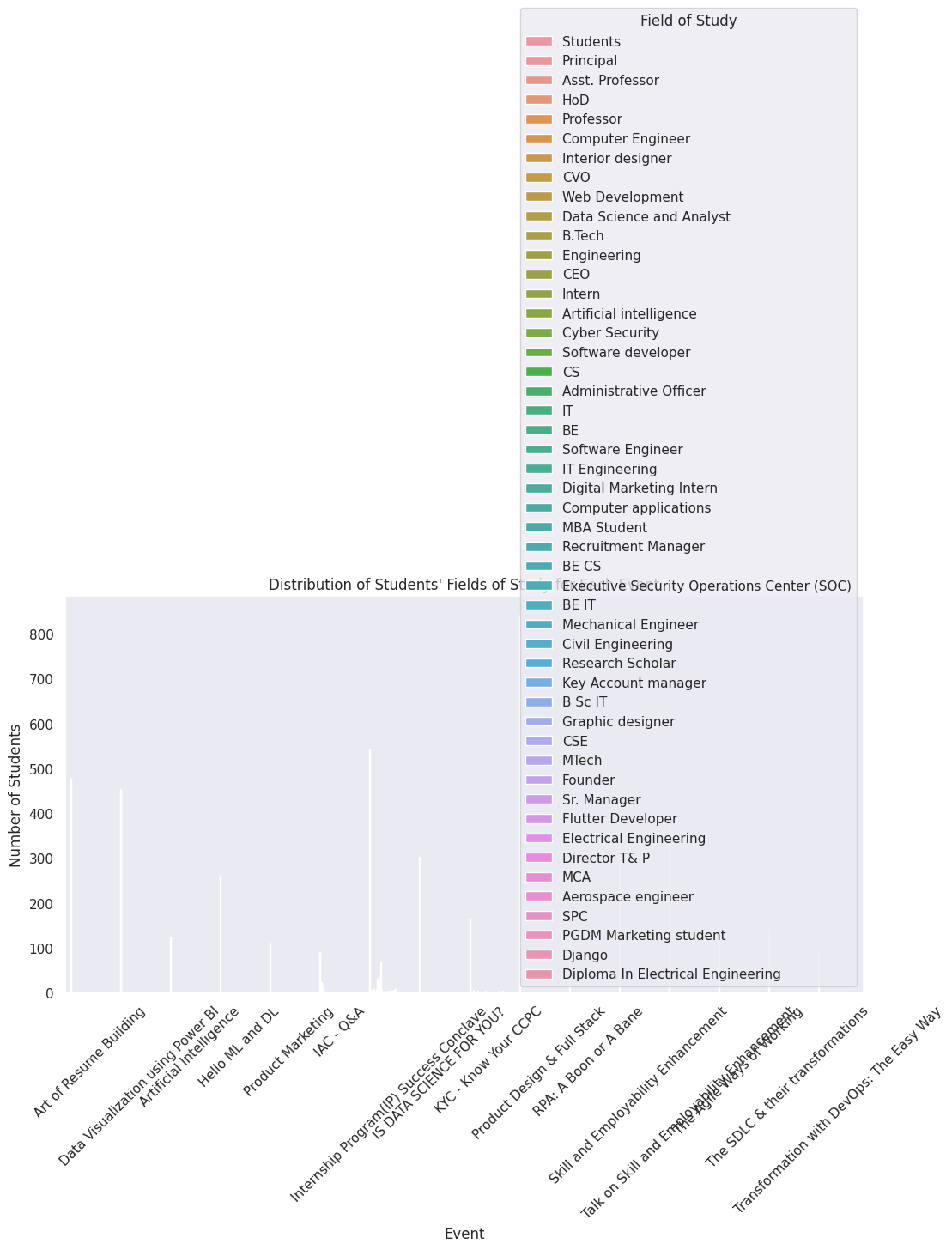
plt.ylabel('Number of Students')

plt.xticks(rotation=45)

plt.legend(title='Field of Study')

plt.tight\_layout()

plt.show()



13. Do students in leadership positions tend to have higher GPAs?

#Do students in leadership positions tend to have higher GPAs?

plt.figure(figsize=(8, 6))

sns.scatterplot(data=d, x='Leadership- skills', y='CGPA')

plt.title('Leadership Skills vs GPA')

plt.xlabel('Leadership Skills')

plt.ylabel('GPA')

plt.show()



14.How many students graduating by the end of 2024 tend to have higher GPAs?

#. How many students graduating by the end of 2024 tend to have higher GPAs?

plt.figure(figsize=(10, 6))

sns.countplot(data=d, x='Year of Graduation', hue='CGPA')

plt.title('Distribution of GPAs for Different Graduation Years')

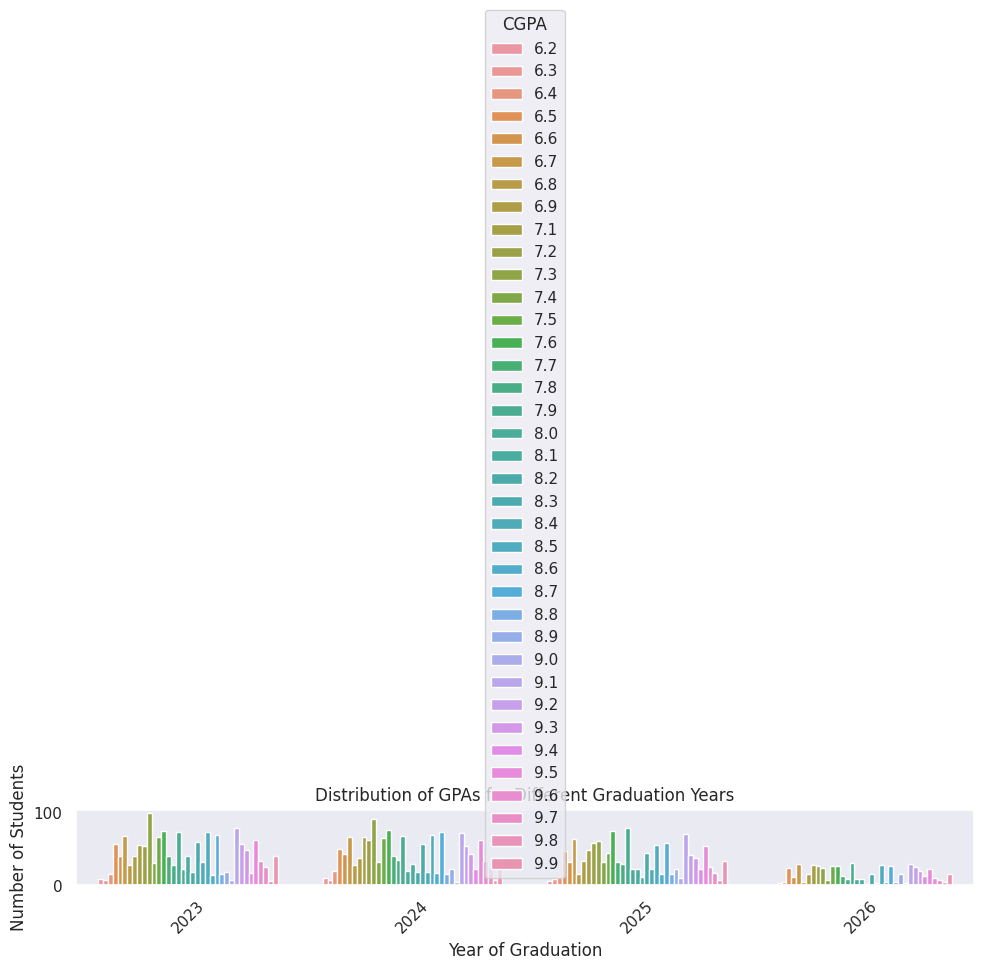
plt.xlabel('Year of Graduation')

plt.ylabel('Number of Students')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



15.Which promotion channel brings in more student participations for the event?

#Which promotion channel brings in more student participations for the event?

plt.figure(figsize=(10, 6))

sns.countplot(data=d, x='How did you come to know about this event?', hue='Attendee Status')

plt.title('Distribution of Students\' Participation by Promotion Channel')

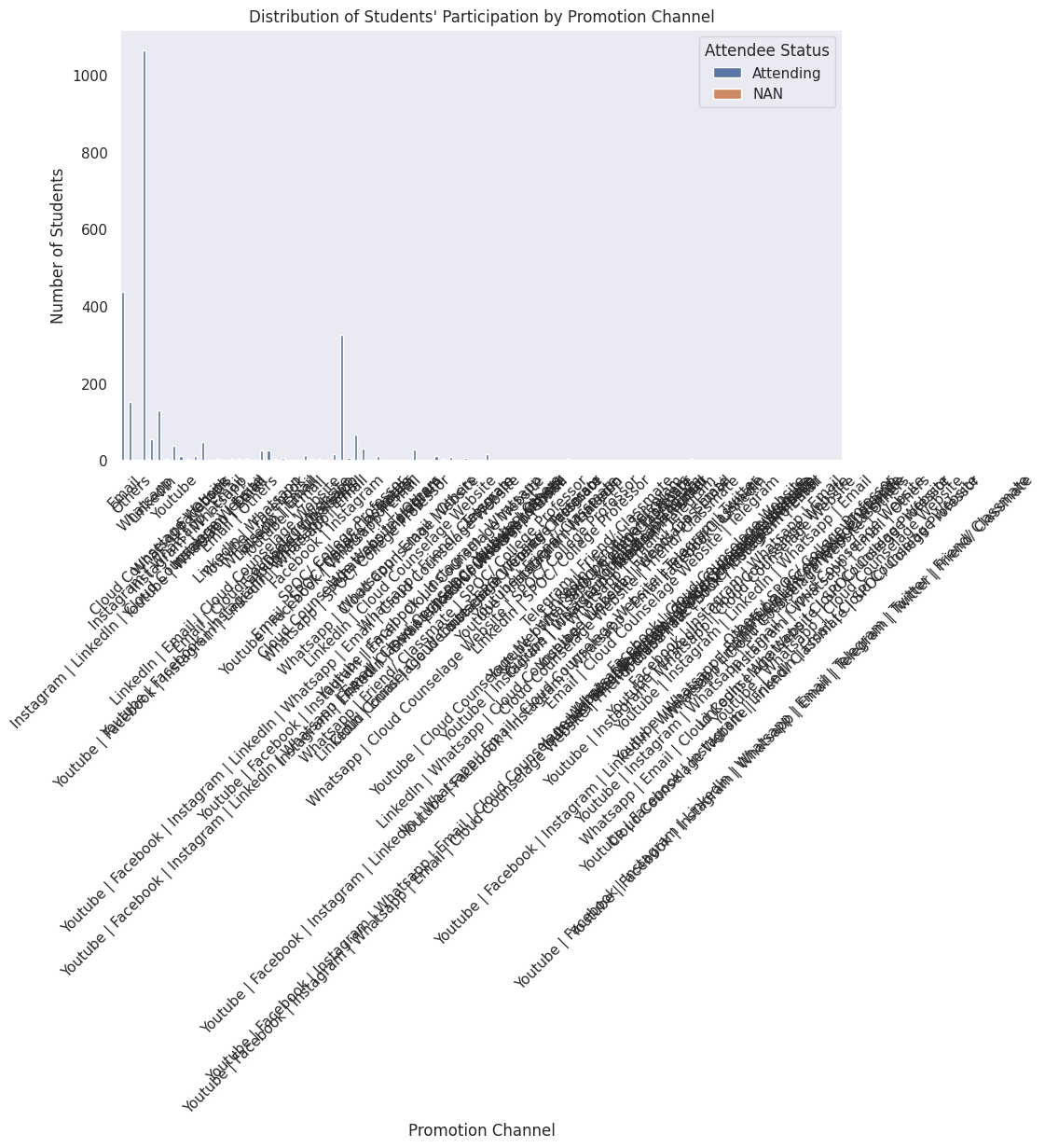
plt.xlabel('Promotion Channel')

plt.ylabel('Number of Students')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



16.Find the total number of students who attended events related to Data Science?

#Find the total number of students who attended events related to Data Science?

data\_science\_events = ['Data Science Event 1', 'Data Science Event 2','Data Science Event 3']

data\_science\_attendees = d[d['Events'].isin(data\_science\_events)].shape[0]

print("Total number of students who attended Data Science events:", data\_science\_attendees)

Total number of students who attended Data Science events: 0

17. Do students with high CGPA and more experience tend to have higher salary expectations?

#Do students with high CGPA and more experience tend to have higher salary expectations?

plt.figure(figsize=(10, 6))

sns.scatterplot(data=d, x='CGPA', y='Expected salary (Lac)', hue='Experience with python (Months)')

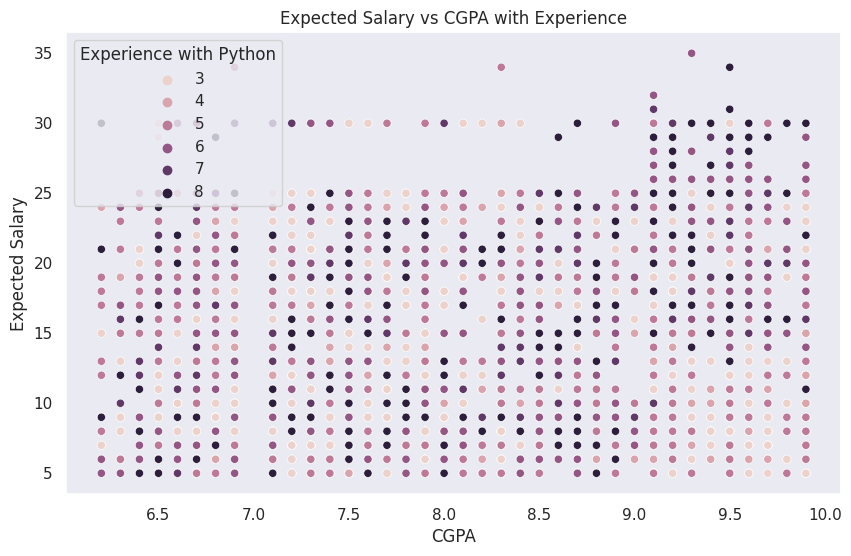
plt.title('Expected Salary vs CGPA with Experience')

plt.xlabel('CGPA')

plt.ylabel('Expected Salary')

plt.legend(title='Experience with Python')

plt.show()



18. How many students know about the event from their colleges? Which of these Top 5 colleges?

# How many students know about the event from their colleges? Which of these Top 5 colleges?

top\_colleges = d['College  Name'].value\_counts().nlargest(5)

plt.figure(figsize=(10, 6))

sns.barplot(x=top\_colleges.index, y=top\_colleges.values)

plt.title('Top 5 Colleges: Students\' Awareness about the Event from Their Colleges')

plt.xlabel('College Name')

plt.ylabel('Number of Students')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

