**Sanika Gurav**

**Data Analytics Internship**

**INDIVIDUAL PROJECT**

**⏱ 120 HOUR**

**TITLE : ABSENCE OF INSIGHTS FOR THE RELATIONSHIP BETWEEN STUDENT’S ECONOMIC BACKGROUND, ACADEMIC PERFORMANCE, COMPETENCE & EXPECTED SALARY.**

1. **How many unique students are included in the dataset?**

**Code:**

num\_unique\_students = df['First Name'].nunique()

print(f"Number of unique students: {num\_unique\_students}")

**Conclusion:** Number of unique students: 2324

1. **What is the average GPA of the students?**

Code:

average\_cgpa = df['CGPA'].mean()

print(f"Average CGPA: {average\_cgpa}")

**Conclusion:** Average CGPA: 8.038475684511647

1. **What is the distribution of students across different graduation years**

Code: # Count the occurrences of each graduation year

graduation\_year\_counts = df['Year of Graduation'].value\_counts().sort\_index()

# Create a bar plot

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

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

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

plt.xlabel('Year of Graduation')

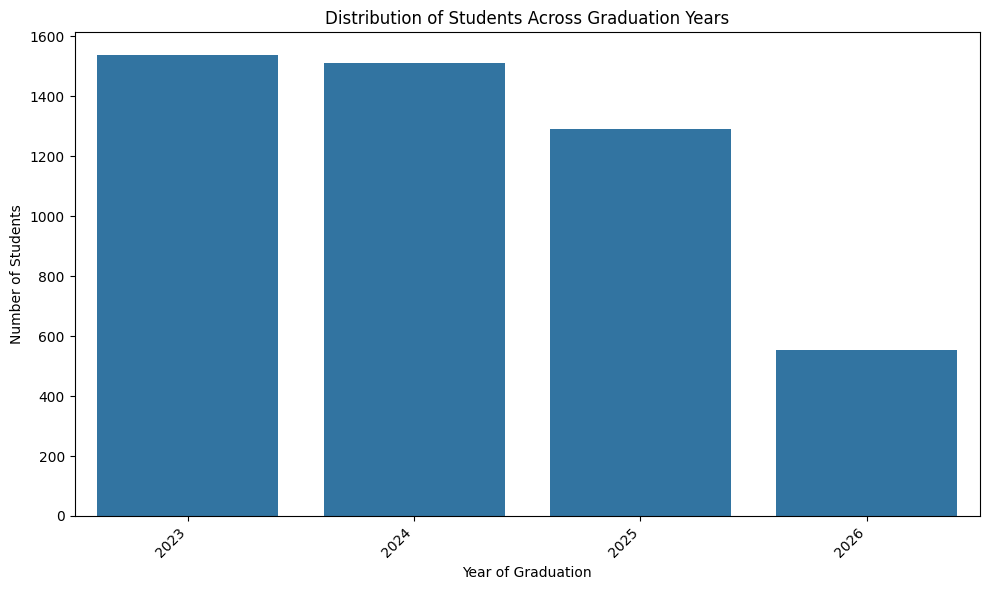
plt.ylabel('Number of Students')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

**Conclusion:**



1. **What is the distribution of students’s experience with python programming?**

Code: # Distribution of student's experience with Python programming

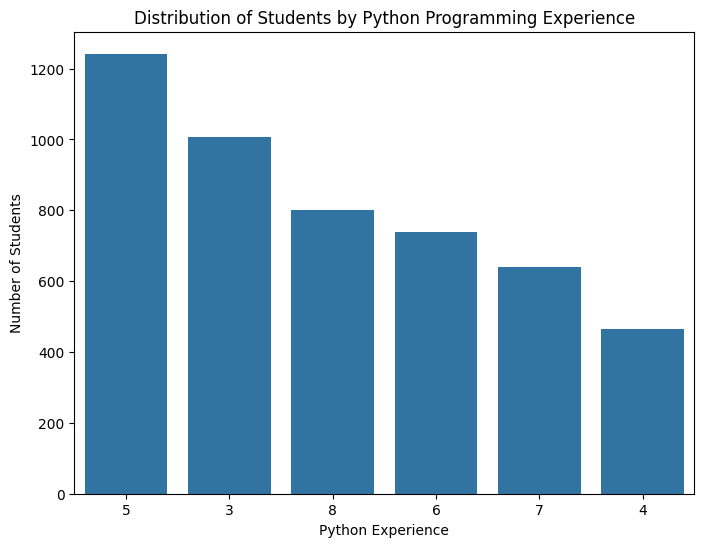
plt.figure(figsize=(8, 6))

sns.countplot(data=df, x='Experience with python (Months)', order=df['Experience with python (Months)'].value\_counts().index)

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

plt.xlabel('Python Experience')

plt.ylabel('Number of Students')

**Conclusion:**

1. **What is the average family income of the students?**

**Code: def income\_to\_numeric(income\_range):**

    if income\_range == '0-2 Lakh':

        return 1  # Midpoint of 0-2 Lakh

    elif income\_range == '2-5 Lakh':

        return 3.5  # Midpoint of 2-5 Lakh

    elif income\_range == '5-7 Lakh':

        return 6  # Midpoint of 5-7 Lakh

    elif income\_range == '7 Lakh+':

        return 8  # Assuming 8 Lakh as a representative value for 7 Lakh+

    else:

        return None # Handle any other cases

# Apply the function to the 'Family Income' column

df['Family Income Numeric'] = df['Family Income'].apply(income\_to\_numeric)

# Calculate the average family income using the new numeric column

average\_family\_income = df['Family Income Numeric'].mean()

print(f"Average Family Income (in Lakhs): {average\_family\_income}")

**Conclusion:** Average Family Income (in Lakhs): 1.3111973845525133

1. **How does the GPA vary among different colleges?(show top 5 results only)**

Code: # Group by 'College Name' and calculate the mean 'CGPA' for each college

college\_cgpa = df.groupby('College Name')['CGPA'].mean().sort\_values(ascending=False)

# Select the top 5 colleges

top\_5\_colleges\_cgpa = college\_cgpa.head(5)

# Print the top 5 results

print("Top 5 Colleges by Average CGPA:")

print(top\_5\_colleges\_cgpa)

# Create a bar plot of the top 5 colleges

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

sns.barplot(x=top\_5\_colleges\_cgpa.index, y=top\_5\_colleges\_cgpa.values)

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

plt.xlabel('College Name')

plt.ylabel('Average CGPA')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

**plt.show()**

**Conclusion:** Top 5 Colleges by Average CGPA:

College Name

THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER DEVELOPMENT & RESEARCH - [TIMSCDR] = 8.585714

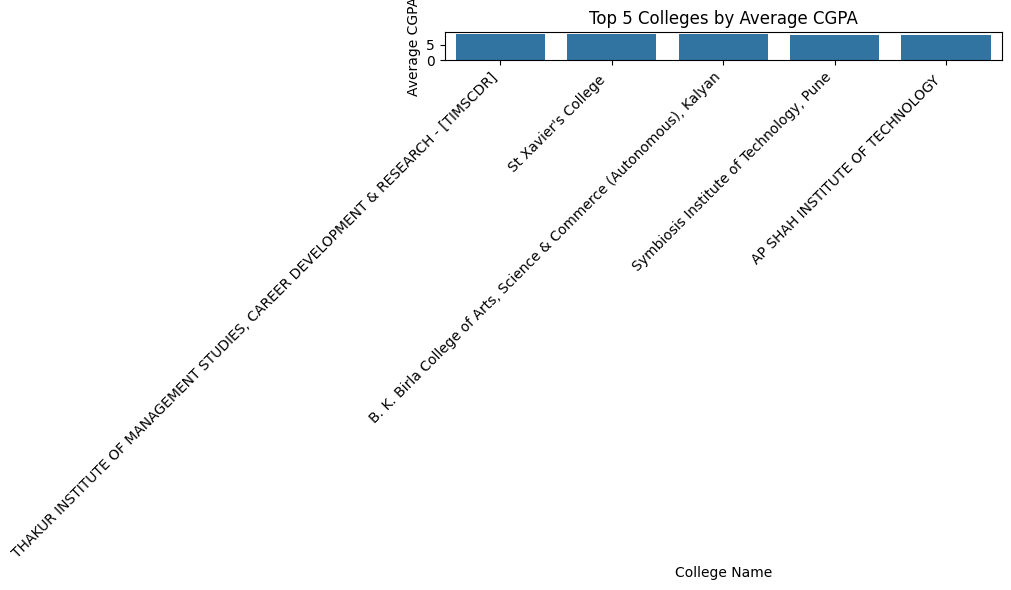
St Xavier's College = 8.578571

B. K. Birla College of Arts, Science & Commerce (Autonomous), Kalyan= 8.456410

Symbiosis Institute of Technology, Pune = 8.303448

AP SHAH INSTITUTE OF TECHNOLOGY = 8.283333

Name: CGPA, dtype: float64



1. **Are there any outliers in the quantity (number of courses completed) attribute?**

**Code: if 'Quantity' in df.columns:**

  Q1 = df['Quantity'].quantile(0.25)

  Q3 = df['Quantity'].quantile(0.75)

  IQR = Q3 - Q1

  # Define the lower and upper bounds for outlier detection

  lower\_bound = Q1 - 1.5 \* IQR

  upper\_bound = Q3 + 1.5 \* IQR

  # Identify outliers

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

  print("\nOutliers in 'Quantity' attribute:")

  display(outliers)

  # Optional: Visualize the distribution and outliers using a box plot

  plt.figure(figsize=(8, 6))

  sns.boxplot(x=df['Quantity'])

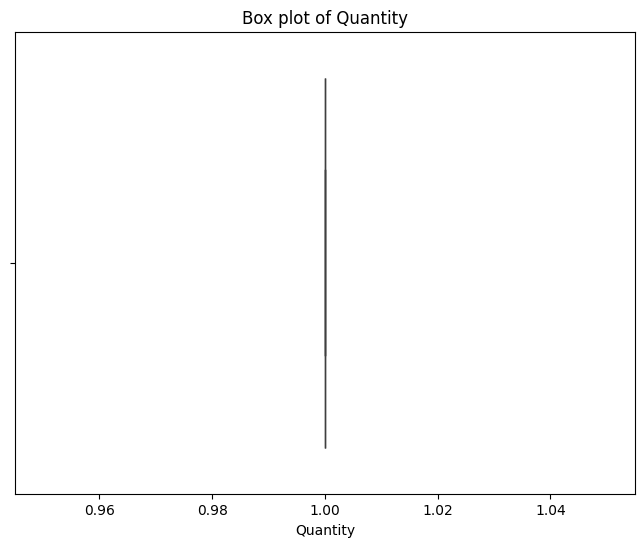
  plt.title('Box plot of Quantity')

  plt.xlabel('Quantity')

  plt.show()

else:

  print("The 'Quantity' column was not found in the DataFrame.")

**Conclusion:**

1. **What is the average GPA for students form each city?**

Code: # Group by 'City' and calculate the mean 'CGPA' for each city

city\_cgpa=df.groupby('City')['CGPA'].mean().sort\_values(ascending=False)

# Print the average CGPA for each city

print("Average CGPA of students from each city:")

print(city\_cgpa)

# For demonstration, plotting all cities

plt.figure(figsize=(12, 8))

sns.barplot(x=city\_cgpa.index, y=city\_cgpa.values, palette='viridis')

plt.title('Average CGPA by City')

plt.xlabel('City')

plt.ylabel('Average CGPA')

plt.xticks(rotation=90, ha='right') # Rotate labels for readability if many cities

plt.tight\_layout()

plt.show()

**Conclusion:** Average CGPA of students from each city:

City

Kolhapur 8.557143

Raipur 8.507143

Sonipat 8.464286

Gurugram 8.459259

Puri 8.450000

...

Panipat 7.615385

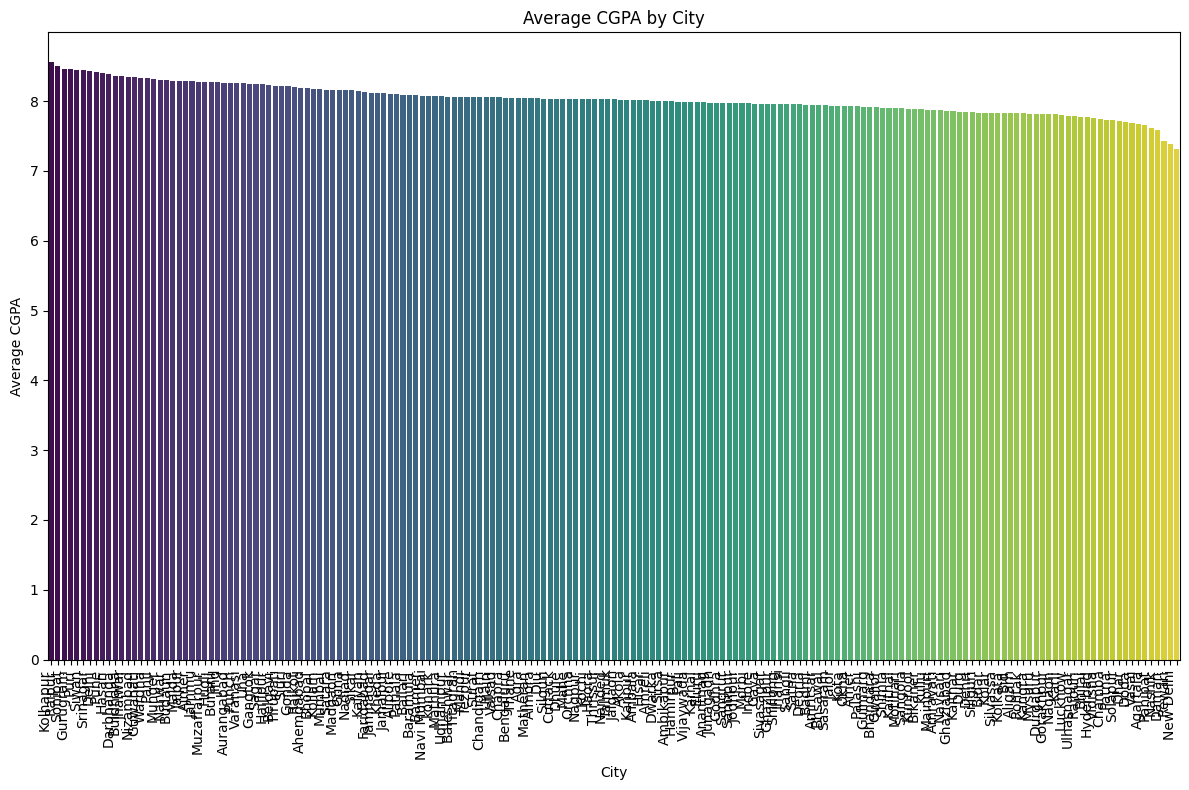
Nashik 7.592857

Daman 7.421429

Rewari 7.392308

New Delhi 7.307143

Name: CGPA, Length: 177, dtype: float64



1. **Can we identify any relationship between family income and GPA?**

Code: # Calculate the correlation between 'Family Income Numeric' and 'CGPA'

correlation = df['Family Income Numeric'].corr(df['CGPA'])

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

# Visualize the relationship using a scatter plot

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

sns.scatterplot(data=df, x='Family Income Numeric', y='CGPA', alpha=0.6)

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

plt.xlabel('Family Income (Numeric - Lakhs)')

plt.ylabel('CGPA')

plt.grid(True)

plt.show()

# You can also use a regression plot to see the trend

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

sns.regplot(data=df, x='Family Income Numeric', y='CGPA', scatter\_kws={'alpha':0.6})

plt.title('Regression Plot: Family Income vs. CGPA')

plt.xlabel('Family Income (Numeric - Lakhs)')

plt.ylabel('CGPA')

plt.grid(True)

plt.show()

# Interpretation of the correlation coefficient:

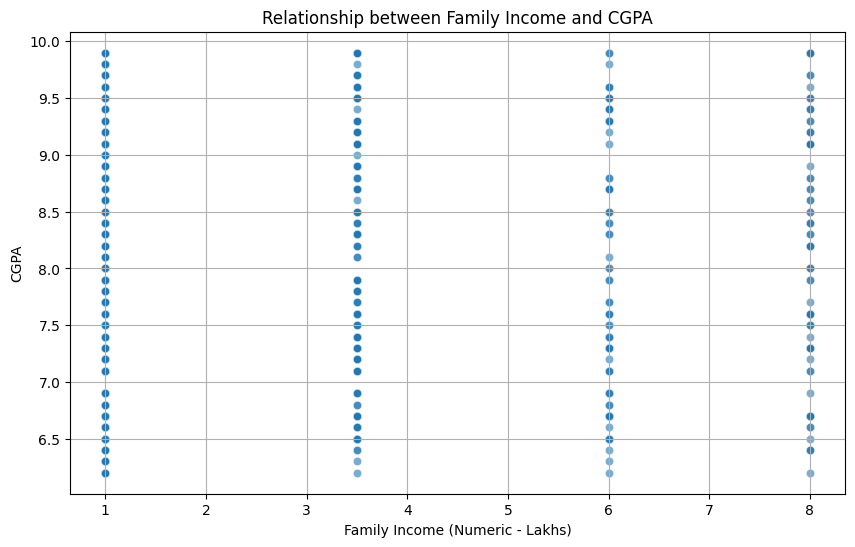
# - A value close to 1 indicates a strong positive linear relationship.

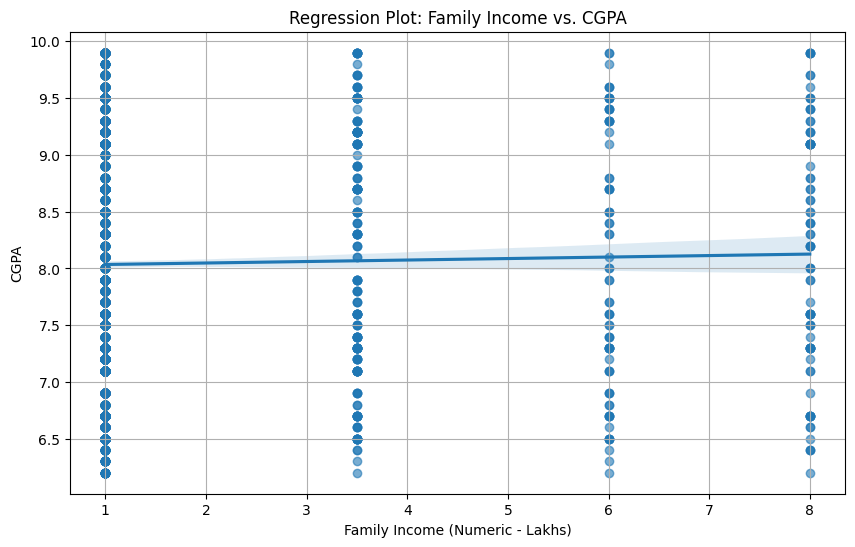
# - A value close to -1 indicates a strong negative linear relationship.

# - A value close to 0 indicates a weak or no linear relationship.

**# Note: Correlation does not imply causation.**

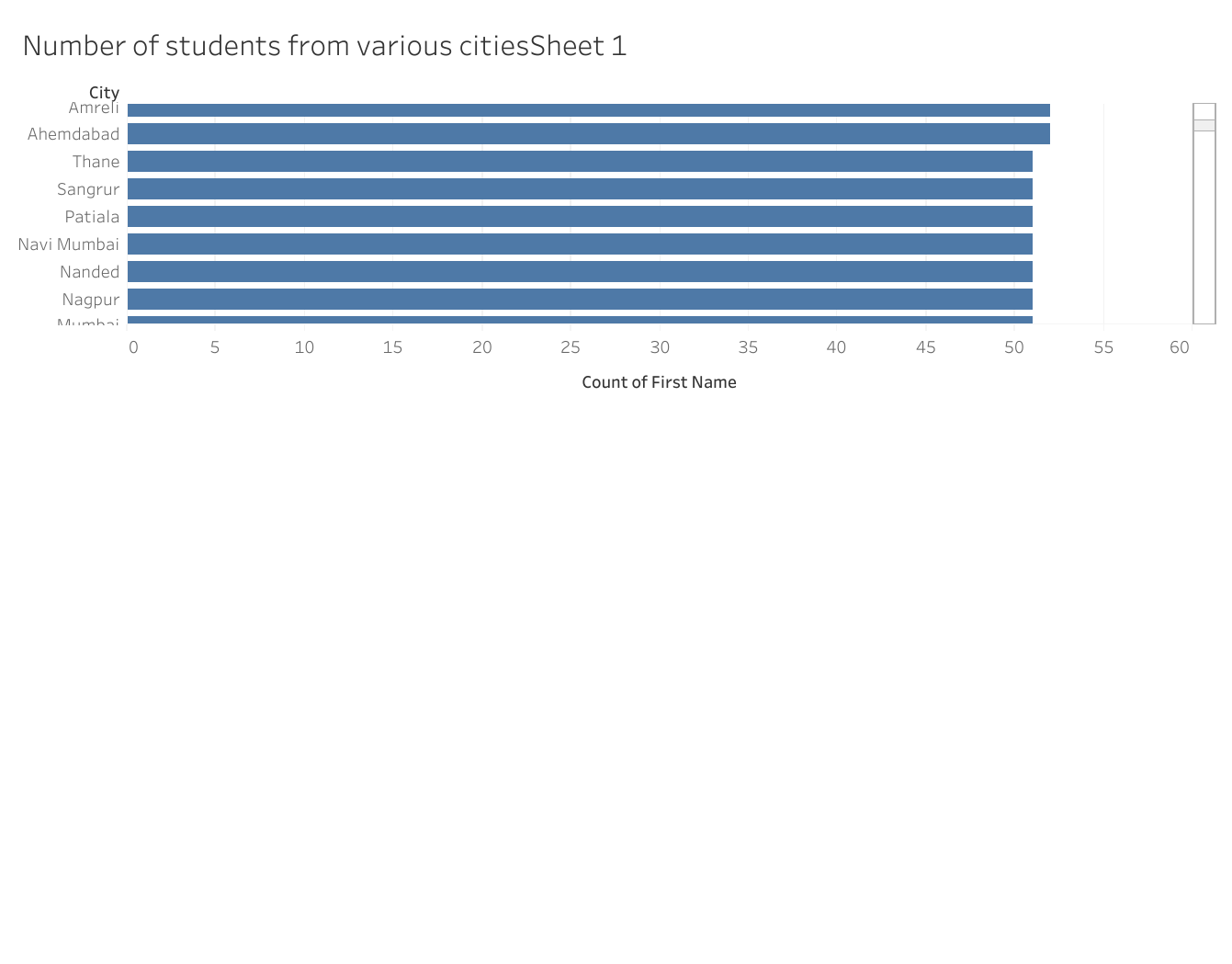
**Conclusion:** Correlation between Family Income and CGPA: 0.02





**10.How many students from various cities?(Solve using data visualisation tool)**

**I have done using Tableau software.**

****

**11.How does the expected salary vary based on factors like ‘GPA’, ‘Family Income’, ‘Experience with python(Months)’?**

**Code:** salary\_analysis = df[['Expected salary (Lac)', 'CGPA', 'Family Income', 'Experience with python (Months)']].dropna()

# Convert Family Income into ordered category

income\_order = ['0-2 Lakh', '2-5 Lakh', '5-7 Lakh', '7 Lakh+']

salary\_analysis['Family Income'] = pd.Categorical(salary\_analysis['Family Income'], categories=income\_order, ordered=True)

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

sns.scatterplot(data=salary\_analysis, x='CGPA', y='Expected salary (Lac)', hue='Family Income', palette='viridis')

plt.title("Expected Salary vs GPA colored by Family Income")

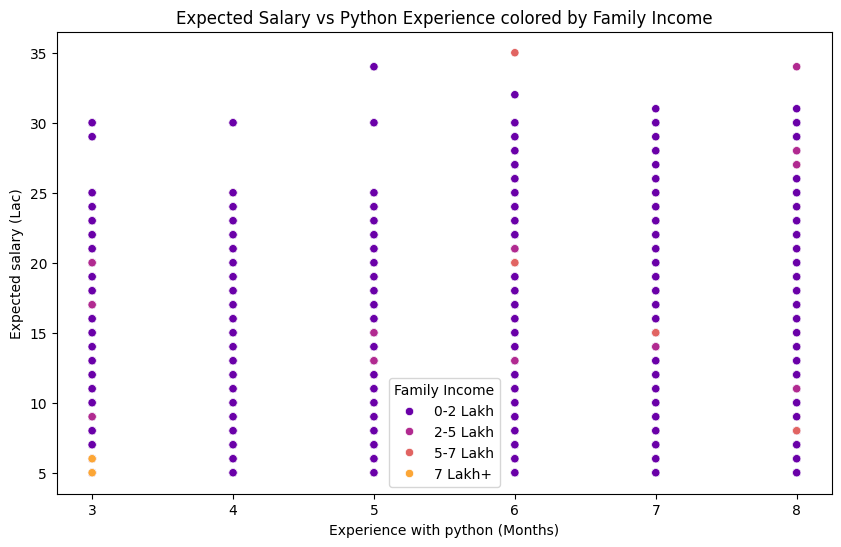
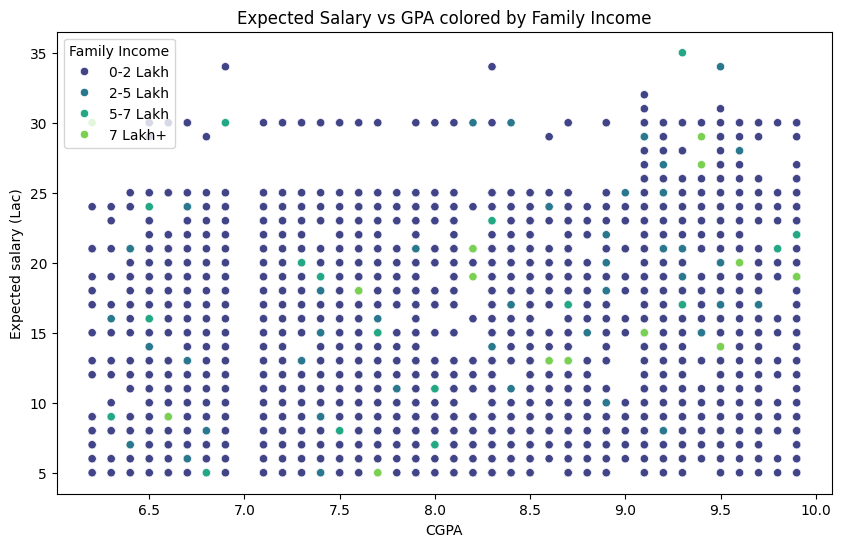
plt.show()

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

sns.scatterplot(data=salary\_analysis, x='Experience with python (Months)', y='Expected salary (Lac)', hue='Family Income', palette='plasma')

plt.title("Expected Salary vs Python Experience colored by Family Income")

plt.show()

**Conclusion:**

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

**Code:** event\_counts = df['Events'].value\_counts()

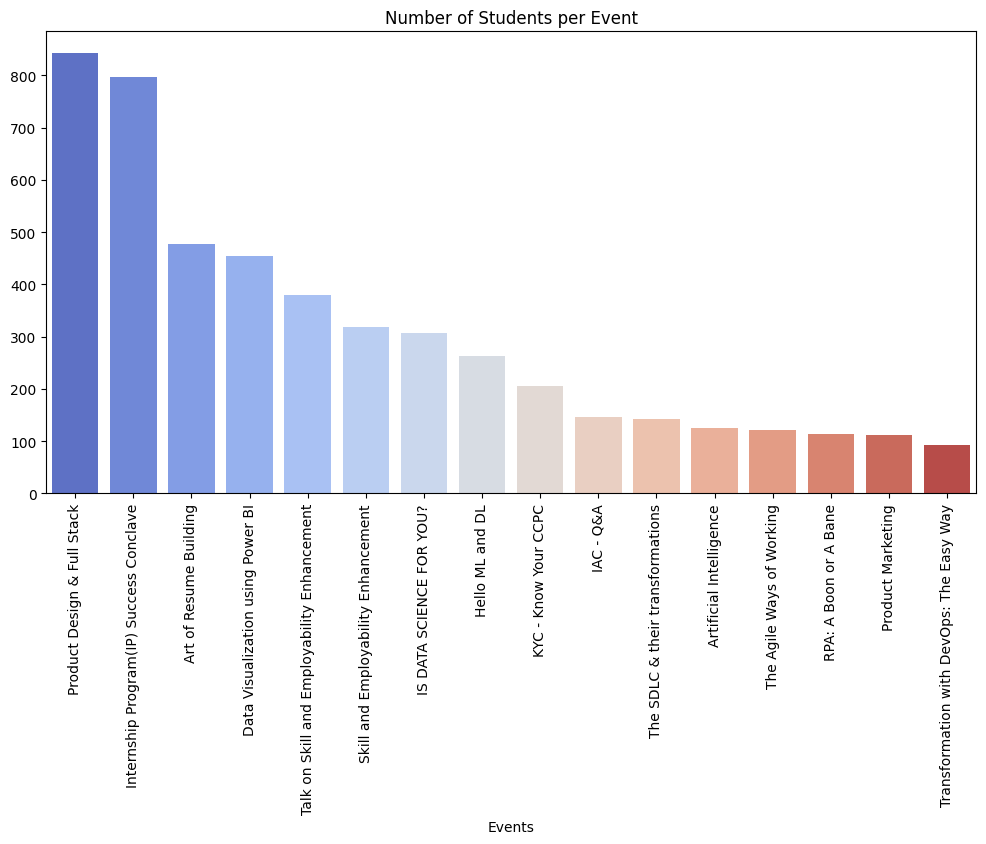
plt.figure(figsize=(12, 6))

sns.barplot(x=event\_counts.index, y=event\_counts.values, palette='coolwarm')

plt.xticks(rotation=90)

plt.title("Number of Students per Event")

plt.show()

**Conclusion:**

**13.Do students in leadership positions during their college years tend to have higher GPAs or better expected salary?**

leadership = df[['Leadership- skills', 'CGPA', 'Expected salary (Lac)']].dropna()

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

sns.boxplot(x='Leadership- skills', y='CGPA', data=leadership)

plt.title("Leadership vs CGPA")

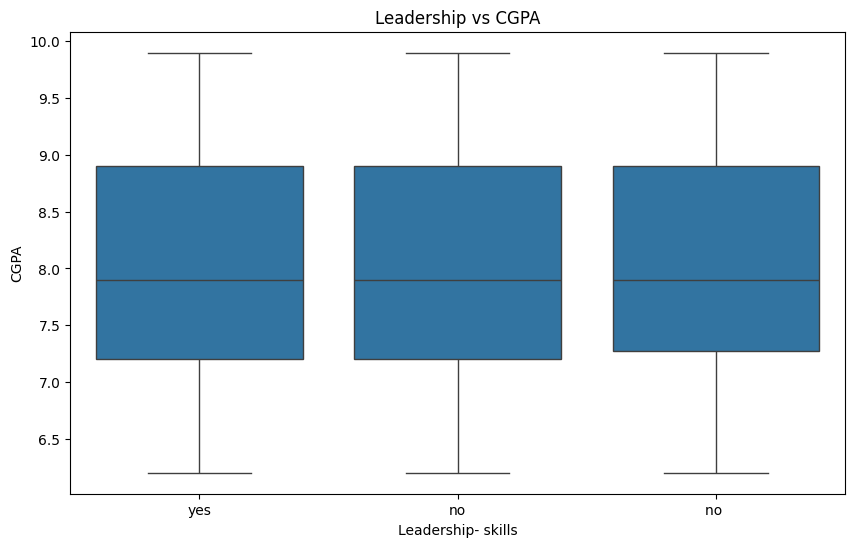
plt.show()

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

sns.boxplot(x='Leadership- skills', y='Expected salary (Lac)', data=leadership)

plt.title("Leadership vs Expected Salary")

plt.show()

**Conclusion:**

**14.How many students are graduating by the end of 2024?**

**Code:** graduates\_2024 = df[df['Year of Graduation'] <= 2024]

print(f"Total students graduating by end of 2024: {graduates\_2024.shape[0]}")

**Conclusion:** Total students graduating by end of 2024: 3047

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

**Code:** promotion\_counts = df['How did you come to know about this event?'].value\_counts()

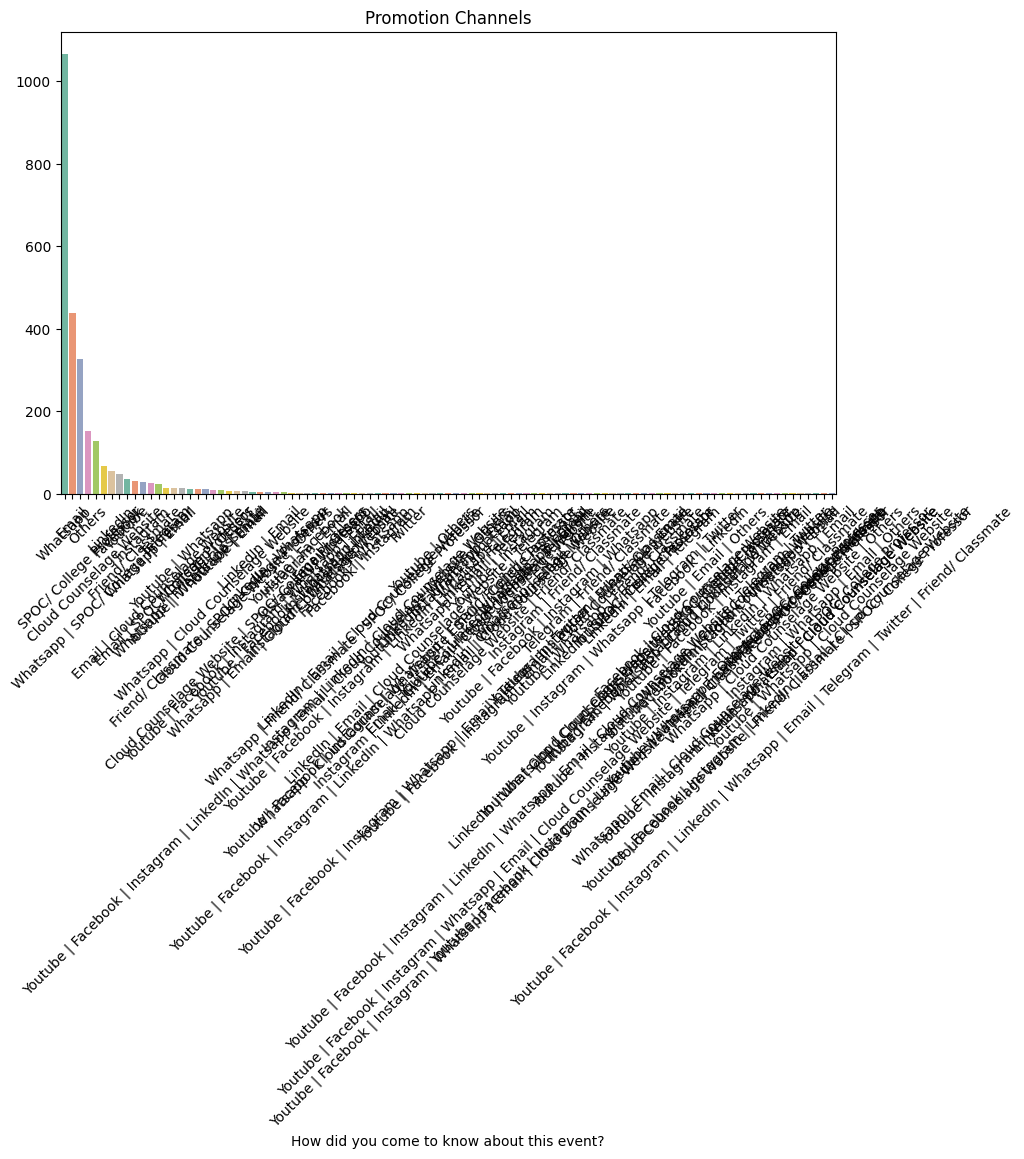
plt.figure(figsize=(10, 6))

sns.barplot(x=promotion\_counts.index, y=promotion\_counts.values, palette='Set2')

plt.xticks(rotation=45)

plt.title("Promotion Channels")

plt.show()

**Conclusion:**

**16.Find total number of students who attended the evnts related to Data science?(From all data science related courses)**

**Code:** ds\_keywords = ['Data Science', 'Python', 'AI', 'Machine Learning']

ds\_events = df[df['Events'].str.contains('|'.join(ds\_keywords), case=False, na=False)]

print(f"Total students attended Data Science related events: {ds\_events.shape[0]}")

**Conclusion:**Total students attended Data Science related events: 306

**17.Those who have high CGPA & more experience in language those who had high expectations for salary(Avg)?**

**Code:** high\_cgpa\_exp = df[(df['CGPA'] >= 8) & (df['Experience with python (Months)'] >= 6)]

avg\_salary = high\_cgpa\_exp['Expected salary (Lac)'].mean()

print(f"Average salary for high CGPA & Python experience students: {avg\_salary:.2f} LPA")

**Conclusion:** Average salary for high CGPA & Python experience students: 16.72 LPA

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

**Code:** college\_counts = df['College Name'].value\_counts().head(5)

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

sns.barplot(x=college\_counts.index, y=college\_counts.values, palette='magma')

plt.xticks(rotation=45)

plt.title("Top 5 Colleges by Participation")

plt.show()

print("\nTop 5 Colleges:\n", college\_counts)

**Conclusion:**