Intern Name: Avinash B Sutar

Problem Statement: Absence of insights for the relationship between student’s economic background, academic performance, competence and expected salary.

1.How many unique students are included in a dataset?

Ans: unique students in a dataset: 4894

Code:

```
df = pd.read_excel('projectdata.xlsx')

num_rows = df. shape [0]

print ("unique students are included in a dataset:", num_rows)
```

Conclusion: This code uses the panda’s library to load data from an Excel file, calculates the number of rows (unique students) in the Data Frame, and prints the count.

2.What is average GPA of Students?

Ans: GPA: 8.038475684511647

Code:

```
import pandas as pd

column_name = 'GPA'

mean = df[column_name].mean()

print(" Average GPA  of Sudents:", mean)
```

Conclusion: This code uses pandas to read data from an Excel file, specifically from the 'GPA' column. It then calculates the mean GPA and prints the result.

3.What is distribution of students among different graduation year?

Ans:

```
  Graduation     0
0       2023  1536
1       2024  1511
2       2025  1292
3       2026   555
```

Code:

```
graduation_year= df.groupby(['Graduation']).size().reset_index()

print(graduation_year)
```

Conclusion: This code groped the data according to ‘Graduation’ column and it prints the graduation year.

4.What is Distribution of Students are experienced with python Programming?

Ans: Distribution of student's experienced with Python Programming:

```
Python Experience  count
0             5   1242
1             3   1008
2             8    800
3             6    738
4             7    640
5             4    466
```

Code:

```
python_experience_counts = df['PythonExperience'].value_counts().reset_index()
print("Distribution of student's experiemced with Python Programming:")
print(python_experience_counts)
```

Conclusion: This code calculates the distribution of students' Python experience levels and then prints the count of students with each level of experience.

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

Ans: Average Family Income: 2.3134450347364117

Code:

```
import pandas as pd
pd.set_option('future.no_silent_downcasting', True)
income_mapping = {
    '0-2 Lakh': 2,
    '7 Lakh+': 8,
    '5-7 Lakh': 7,
    '2-5 Lakh': 5
}
df['Family Income'] = df['FamilyIncome'].replace(income_mapping).infer_objects(copy=False)
average_income = df['Family Income'].mean()
print("Average Family Income:", average_income)
```

Conclusion: This code gives a average family income of the students

6.How does GPA vary among different colleges? (Top 5)

Ans:

```
                                   college                   GPA
0 THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER...   8.585714
1                     St Xavier's College                 8.578571
2 B. K. Birla College of Arts, Science & Commerce         8.456410
3         Symbiosis Institute of Technology, Pune         8.303448
4             AP SHAH INSTITUTE OF TECHNOLOGY                8.283333
```

Code:

```
college_gpa = df.groupby('college')['GPA'].mean()

top_colleges = college_gpa.sort_values(ascending=False).head(5).reset_index()

print(top_colleges)
```

Conclusion: It then calculates the average GPA for each college, sorts the colleges in descending order of GPA, and prints the top 5 colleges with the highest average GPAs.

7.Are there any outliers in a quantity?

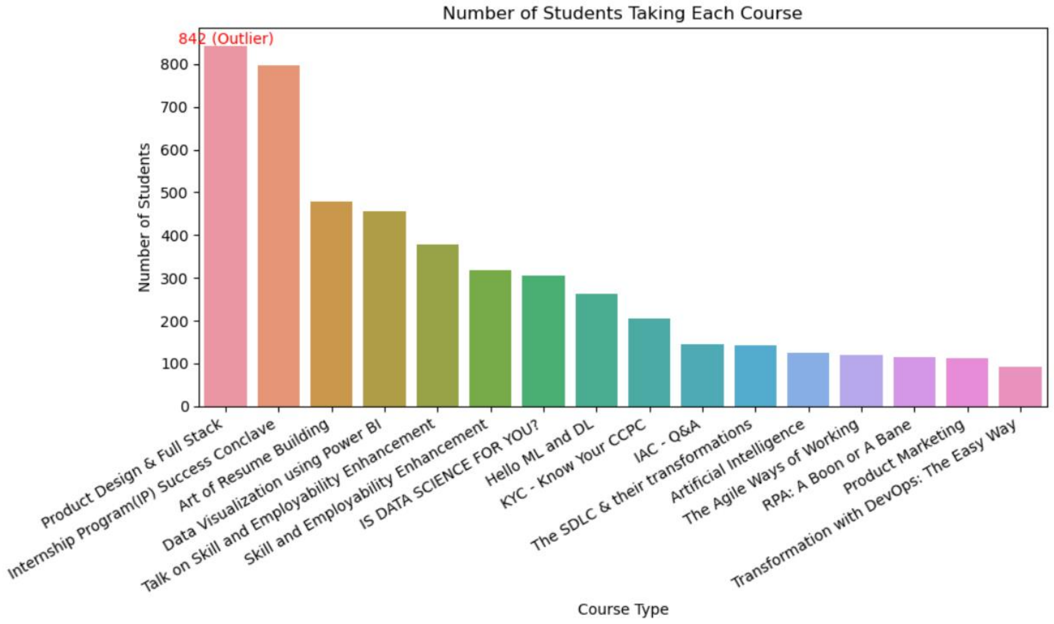
Ans:

```
Outliers in Quantity Obtained (Number of Students Taking Each Course):
Events
Product Design & Full Stack    842
Name: count, dtype: int64
```

Code:

```
course_counts = df['Events'].value_counts()

Q1 = course_counts.quantile(0.25)

Q3 = course_counts.quantile(0.75)

IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR

outliers = course_counts[(course_counts < lower_bound) | (course_counts > upper_bound)]

print("Outliers in Quantity Obtained (Number of Students Taking Each Course):")

print(outliers)


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

sns.barplot(x=course_counts.index, y=course_counts.values)
```

```
plt.xlabel('Course Type')
plt.ylabel('Number of Students')
plt.title('Number of Students Taking Each Course')

for i in outliers.index:
    plt.text(course_counts.index.get_loc(i), course_counts.loc[i], f'{course_counts.loc[i]} (Outlier)', color='red', ha='center', va='bottom')

plt.xticks(rotation=30, ha='right')
plt.tight_layout()
plt.show()
```



Conclusion: This code analyzes the distribution of the number of students taking each course (Events) in the dataset. It calculates the Interquartile Range (IQR) and identifies outliers by finding courses with counts outside the lower and upper bounds defined by 1.5 times the IQR. It then prints the courses that are outliers in terms of student enrolment, and gives the bar plot for visualization.

8.Average GPA of Students from each City?

Ans: Average GPA of students from each city (in decreasing order):

```
City
Kolhapur 8.557143
Raipur 8.507143
Sonipat 8.464286
```

Gurugram 8.459259

Puri 8.450000

Siwan 8.450000

Srinagar 8.435714

Delhi 8.414286

Pune 8.400000

Hasan 8.392857

Darbhanga 8.357143

Buldhana 8.352941

Jhalwar 8.348077

Nizambad 8.342857

Guwahati 8.336364

Wardha 8.328571

Panji 8.321429

Munger 8.307143

Narwar 8.300000

Budaun 8.292857

Malda 8.289286

Jaipur 8.288462

Ajmer 8.284314

Muzaffarpur 8.278571

Jammu 8.278571

Hugli 8.275000

Burani 8.268182

Jind 8.262963

Aurangabad 8.258824

Punch 8.257143

Varanasi 8.253571

Una 8.250000

Gangtok 8.250000

Sagar 8.245238

Haijipur 8.228571

Belgavi 8.221429

Tirupati 8.213636

Siuri 8.212500

Gonda 8.201786

Rajkot 8.193182

Ahemdabad 8.190385

Kollam 8.171429

Bhopal 8.171429

Mumbai 8.164706

Satara 8.164286

Guna 8.157143

Madgaon 8.157143

Nadiad 8.154545

Sikar 8.150000

Kalyan 8.131373

Faridabad 8.118605

Jamnagar 8.118182

Kheda 8.111364

Jamalpur 8.107143

Alipore 8.100000

Patiala 8.094118

Ballari 8.092857

Baramula 8.085714

Barmer 8.078431

Navi Mumbai 8.076471

Mathura 8.071429

konark 8.071429

Titagrah 8.064286

Udhampur 8.064286

Baleshwar 8.064286

Mandi 8.064286

Tezpur 8.063636

Surat 8.061364

Chandigarh 8.059649

Ujjain 8.059524

Valsad 8.052273

Bengaluru 8.050000

Chapra 8.050000

Thane 8.049020

Agra 8.046429

Matheran 8.042857

Almora 8.039286

Silguri 8.035714

Cuttack 8.035714

Diu 8.035714

Dhule 8.035294

Mahe 8.035294

Orchha 8.033333

Nagpur 8.031373

Kochi 8.028571

Thrissur 8.028571

Haora 8.028571

Nanded 8.027451

Talmuk 8.023214

Jalgaon 8.021569

Akola 8.021429

Kanpur 8.021429

Ambala 8.018605

Hisar 8.011111

Dwarka 8.001923

Amreli 8.001923

Ambikapur 8.000000

Sirsa 8.000000

Hamirpur 7.992857

Pali 7.988462

Vijaywada 7.986364

Karnal 7.984615

Eluru 7.981818

Anantnag 7.978571

Junagadh 7.975000

Godhra 7.974000

Sangrur 7.972549

Santipur 7.971429

Jodhpur 7.967308

Morbi 7.965909

Indore 7.964286

Gaya 7.964286

Sivasagar 7.959091

Ghazipur 7.957143

Shillong 7.957143

Jhansi 7.957143

Sangli 7.957143

Satna 7.954762

Badmi 7.950000

Deoria 7.950000

Amritsar 7.945098

Bhsawal 7.943137

Sasaram 7.935714

Jalor 7.934615

Kota 7.925000

Okha 7.925000

Amer 7.923529

Gulmarg 7.921429

Palashi 7.921429

Bhandara 7.909804

Shimla 7.907143

Gwalior 7.907143

Kaithal 7.903704

Motihari 7.900000

Sangola 7.892857

Kunrool 7.886364

Bikaner 7.880769

kullu 7.878571

Mainpuri 7.875000

Amravati 7.866667

Navsari 7.854545

Ghaziabad 7.853571

Kaithar 7.850000

Durg 7.850000

Dehri 7.850000

Bidar 7.835714

Siliguri 7.835714

Silvassa 7.835714

Kolar 7.835714

Kolkata 7.832143

Bid 7.829412

Aligarh 7.828571

Rohtak 7.823077

Kangra 7.814286

Mysuru 7.814286

Gorakhpur 7.810714

Durgapur 7.810714

Nagaon 7.809091

Karli 7.800000

Lucknow 7.792857

Ulhasnagar 7.785714

Rajouri 7.778571

Bhilai 7.778571

Hyderbad 7.764286

Vidisha 7.738095

Chamba 7.728571

Patna 7.726667

Solapur 7.714286

Dhar 7.700000

Doda 7.685714

Vasai 7.671429

Agartala 7.660714

Panipat 7.615385

Nashik 7.592857

Daman 7.421429

Rewari 7.392308

New Delhi 7.307143

Code:

average_gpa_by_city = df.groupby('City')['GPA'].mean().reset_index()

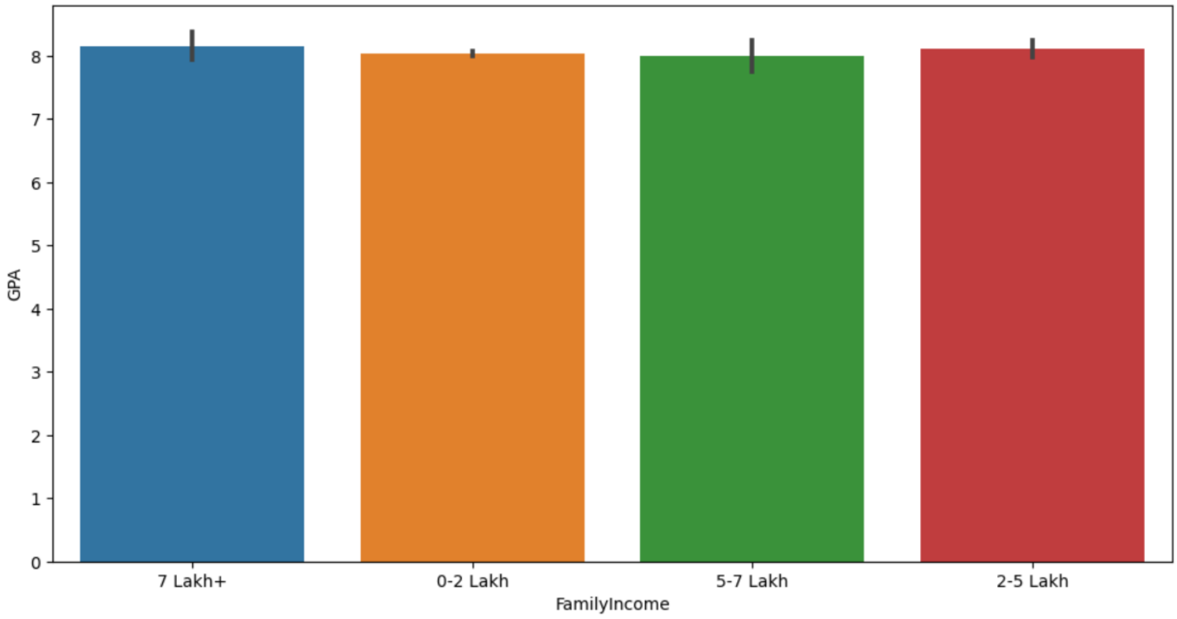
print(average_gpa_by_city)

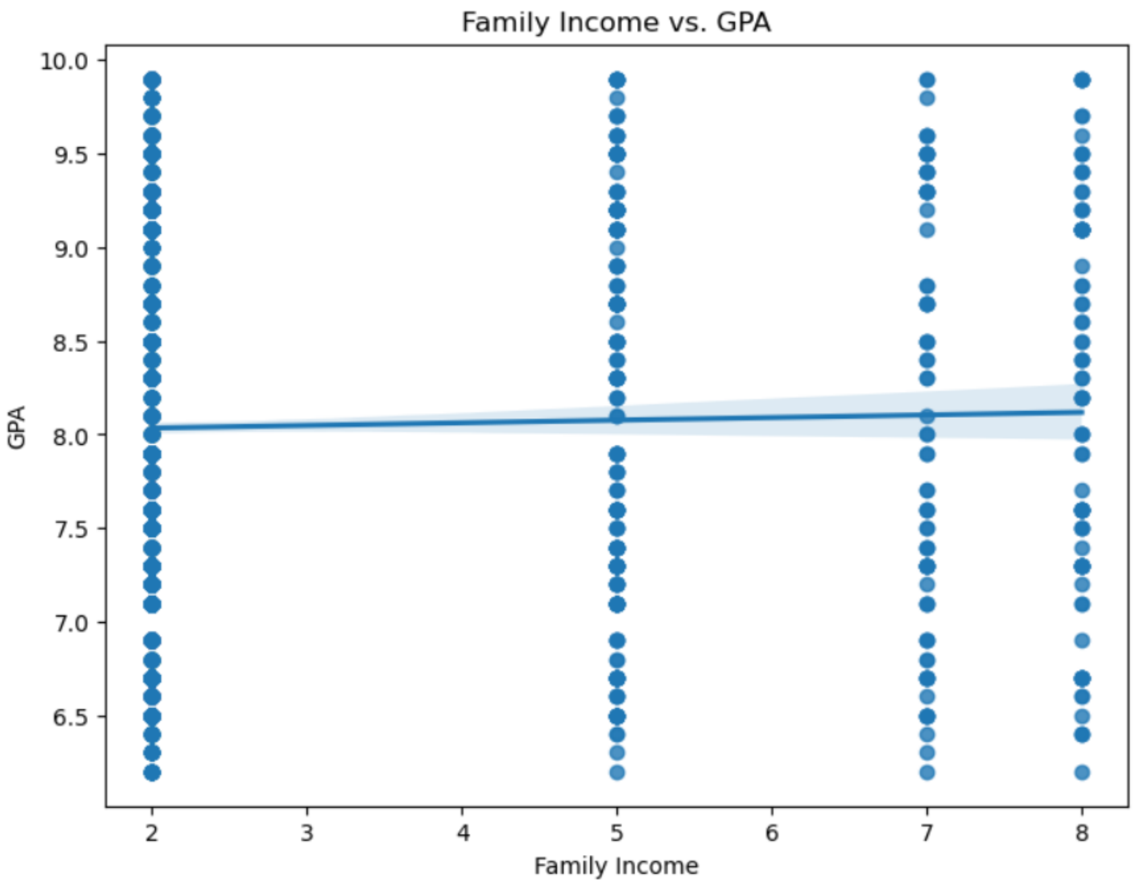
Conclusion: This code provides average GPA of students from each city in decreasing order.

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

Ans:

```
<Axes: xlabel='FamilyIncome', ylabel='GPA'>
```





```
Correlation Coefficient: 0.016073287562741973
```

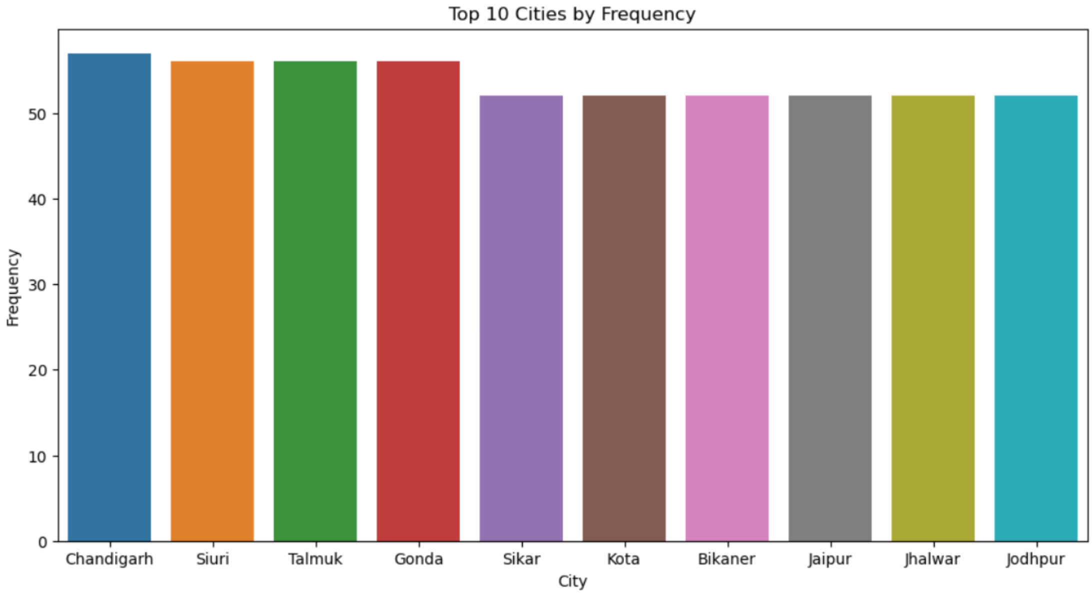
Code:

```
plt.rcParams['figure.figsize']=(12,6)
sns.barplot(x='FamilyIncome',y='GPA',data=df)
df['FamilyIncome'] = df['FamilyIncome'].replace(income_mapping).astype(float)


plt.figure(figsize=(8, 6))
sns.regplot(x='FamilyIncome', y='GPA', data=df)
plt.xlabel('Family Income')
plt.ylabel('GPA')
plt.title('Family Income vs. GPA')
plt.show()


correlation_coefficient = df['FamilyIncome'].corr(df['GPA'])
print("Correlation Coefficient:", correlation_coefficient)
```

Conclusion: This code uses the Seaborn library to create a scatterplot of 'Family Income' on the x-axis and 'GPA' on the y-axis 'GPA'. It labels the axes, sets a title for the plot, and displays the plot with a specified figure size. The plot visualizes the relationship between family income and GPA for the given data.

10.How many Students from various cities? (using Data visualization Tool)

Ans:



Code:

```
city_counts = df['City'].value_counts()

city_counts_df = city_counts.reset_index()

city_counts_df.columns = ['City', 'Frequency']

top_10_various_cities=city_counts_df.head(10)

plt.rcParams['figure.figsize']=(12,6)

sns.barplot(x='City',y='Frequency',data=top_10_various_cities)

plt.title('Top 10 Cities by Frequency')

plt.show()
```

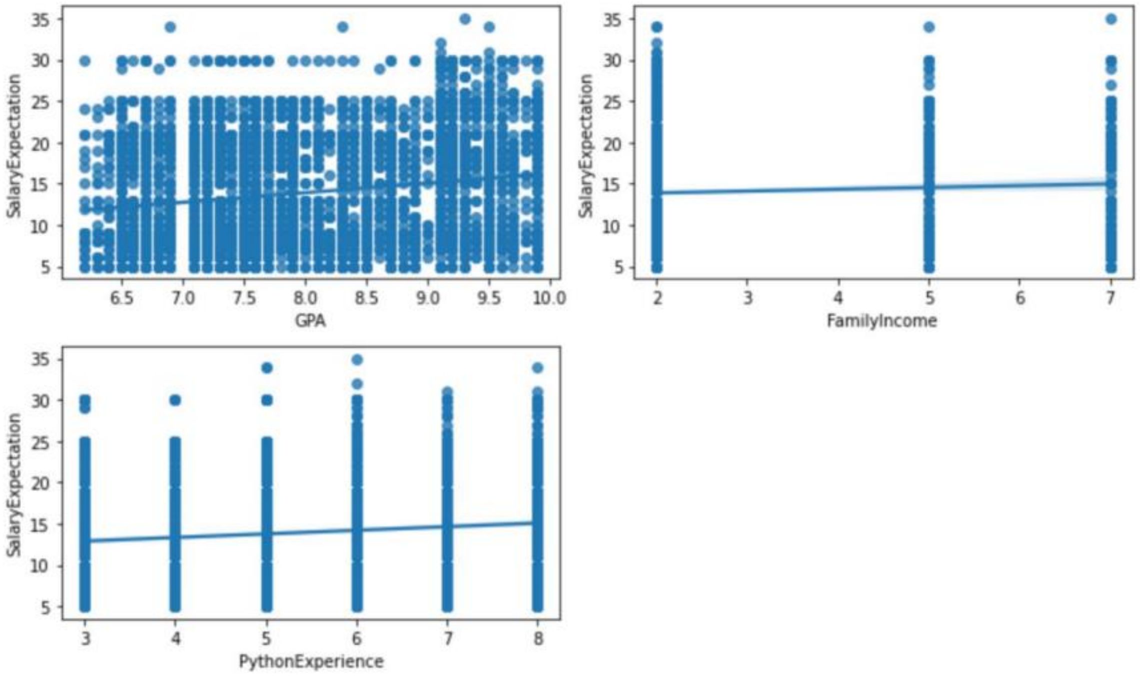
Conclusion: This code provides top 10 cities by frequency by plotting a graph between Frequency vs City

11.How does the expected salary vary based on factors like CGPA,family Income, months of experience in python language?

Ans:

```
Name                                   object
Email ID                               object
Quantity                               int64
Events                                 object
AttendeeStatus                         object
college                                object
How did you come to know about this event?     object
KnowsEvent                             object
Designation                            object
Graduation                             int64
```

```
City                              object
GPA                                float64
PythonExperience                        int64
FamilyIncome                       float64
SalaryExpectation                    int64
LeadershipSkills                  object
Family Income                       int64
```



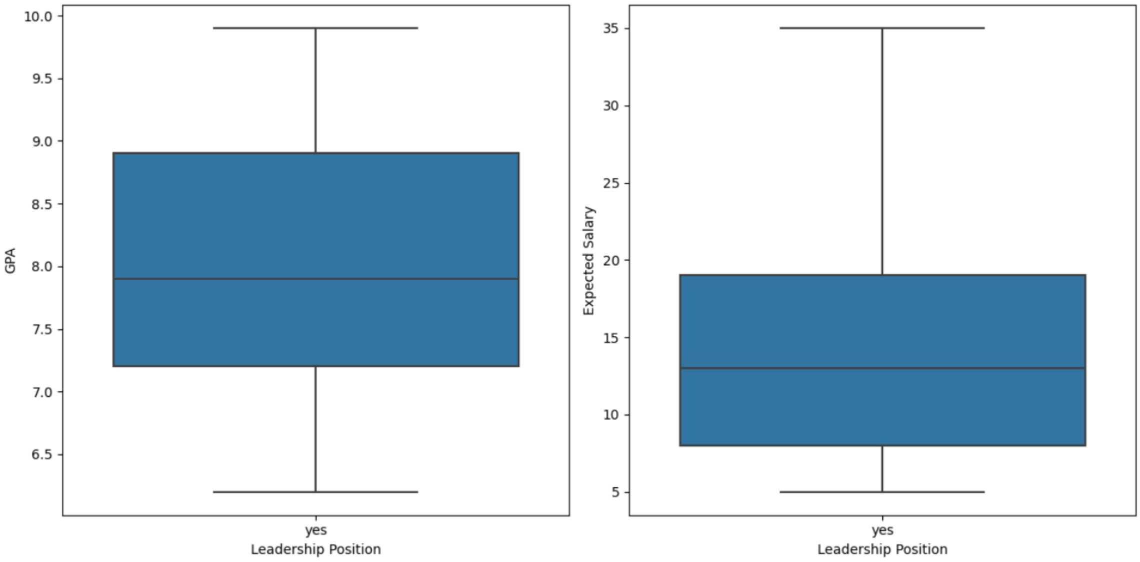
Code:

```
print(df.dtypes)

df['GPA'] = df['GPA'].astype(float)

df['SalaryExpectation'] = df['SalaryExpectation'].astype(float)

df['FamilyIncome'] = df['FamilyIncome'].astype(float)

df['PythonExperience'] = df['PythonExperience'].astype(float)

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

plt.subplot(2, 2, 1)

sns.regplot(x='GPA', y='SalaryExpectation', data=df)

plt.subplot(2, 2, 2)

sns.regplot(x='FamilyIncome', y='SalaryExpectation', data=df)

plt.subplot(2, 2, 3)

sns.regplot(x='PythonExperience', y='SalaryExpectation', data=df)

plt.tight_layout()

plt.show()
```

Conclusion: This code gives the relationship between Salary Expectation vs GPA, Salary Expectation and Family Income and Salary Expectation vs Family Income.

12.Which event tend to atracts more students for specific field of study?

Ans:



Code:

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.boxplot(x='LeadershipSkills', y='GPA', data=df)
plt.xlabel('Leadership Position')
plt.ylabel('GPA')
plt.subplot(1, 2, 2)
sns.boxplot(x='LeadershipSkills', y='SalaryExpectation', data=df)
plt.xlabel('Leadership Position')
plt.ylabel('Expected Salary')
plt.tight_layout()
plt.show()
```

Conclusion: This code creates a side-by-side comparison of two boxplots using Seaborn. The first subplot (1, 2, 1) shows the relationship between 'Leadership Skills' and 'GPA', while the second subplot (1, 2, 2) illustrates the relationship between 'Leadership Skills' and 'Salary Expectation'. Each subplot displays how these variables are distributed within different leadership positions.

15.how many students are graduating by the end of 2024

Ans: Number of students graduating by the end of 2024: 1511

Code:

```
students_graduating_2024 = df[df['Graduation'] == 2024]
number_of_students_graduating_2024 = len(students_graduating_2024)
print(f"Number of students graduating by the end of 2024: {number_of_students_graduating_2024}")
```

Conclusion: From the output of the code we can conclude that 1511 students are graduating by the end oy year 2024.

16.Which marketing effects better in gaining attention from the students?

Ans:

```
 How did you come to know about this event?  count
0                                 Whatsapp   1067
1                                    Email    438
2                    SPOC/ College Professor    326
3                                   Others    153
4                   Cloud Counselage Website    129
5         Whatsapp | SPOC/ College Professor     67
6                                 LinkedIn     55
7                                 Facebook     48
8                                  Youtube     37
9                         Friend/ Classmate     30
```

Code:

```
social_media_counts = df['How did you come to know about this event?'].value_counts()
top_10_social_media = social_media_counts.head(10).reset_index()
print(top_10_social_media)
```

Conclusion: This code analyses the distribution of how students came to know about an event, specifically focusing on marketing channels. It counts the occurrences of each channel, then identifies and prints the top 10 marketing channels with the highest counts in the dataset

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

Ans: Total number of students who attended the Data Science related events is: 306

Code:

```
data_science_attendees = df[df['Events'] == 'IS DATA SCIENCE FOR YOU?']
number_of_attendees = len(data_science_attendees)
print("Total number of students who attended the Data Science related events is:", number_of_attendees)
```

Conslusion: This code filters the DataFrame to select students who attended the 'IS DATA SCIENCE FOR YOU?' event. It then calculates the number of attendees by finding the length of the filtered DataFrame and prints the total number of students who attended the Data Science-related event.

18. How many students know about the event from their colleges? Mention top 5 colleges for it.

Ans: 17 Students know about the events from their colleges.

Code:

```
students_know_events = df[df['KnowsEvent'] == True]

number_of_students_know_events = students_know_events.shape[1]

print(f"Number of students who know about the events from their colleges: {number_of_students_know_events}")

students_know_events = df[df['KnowsEvent']=='college'].value_counts()\
```

Conclusion: : This will correctly count the number of students who know about events from their colleges and we conclude that 17 Students know the events from their Colleges.

19. What is the average expected salary of students having more than 8.5 cgpa or having experience in python greater than 2 months?

Ans:

Case i) Average expected salary of students with CGPA > 8.5 : 15.685150955021564

Case ii) Average expected salary of students with Python experience > 2 months : 13.935635472006538

Code :

```
Case i) filtered_data = df[df['GPA'] > 8.5]

average_salary = filtered_data['SalaryExpectation'].mean()

print("Average expected salary of students with CGPA > 8.5  :", average_salary)

case ii) filtered_data = df[df['PythonExperience'] > 2]

average_salary = filtered_data['SalaryExpectation'].mean()

print("Average expected salary of students with Python experience > 2 months  :", average_salary)
```

Conclusion: we can conclude that the CGPA > 8.5 can expect average salary of 15.68 Lakhs and students having python experience more than 2 years can expect sala