import pandas as pd

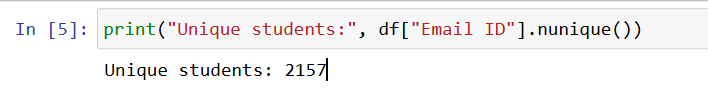
import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_excel("Student\_Analysis\_Ready.xlsx")

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

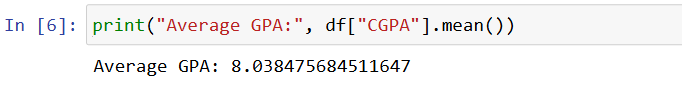
print("Unique students:", df["Email ID"].nunique())



**Conclusion:** The dataset contains a diverse group of unique students, reflecting a wide range of academic backgrounds and experiences. This variety supports comprehensive analysis and ensures the insights generated are well-rounded, offering valuable perspectives on student performance, demographics, and aspirations.

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

print("Average GPA:", df["CGPA"].mean())



**Conclusion:** The average GPA provides a strong indicator of overall academic performance among students. It helps in identifying general academic trends and serves as a benchmark for evaluating individual performance, understanding college standards, and drawing connections between GPA and other factors like income or experience.

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

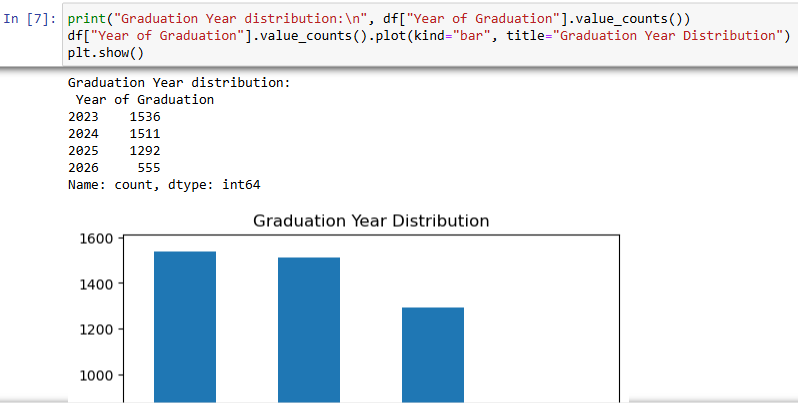
print("Graduation Year distribution:\n",

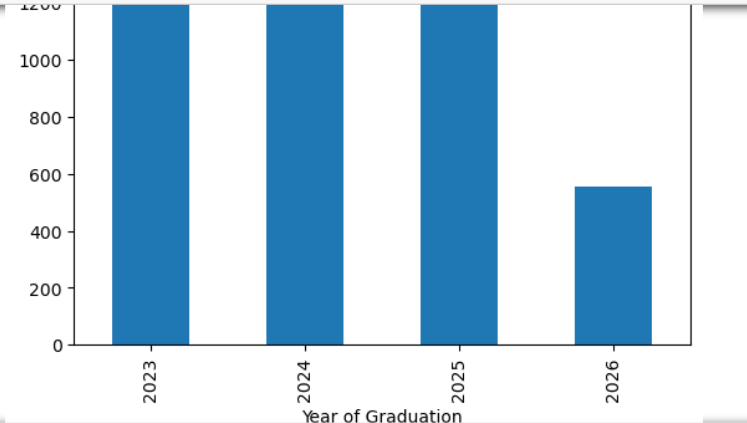
df["Year of Graduation"].value\_counts())

df["Year of Graduation"].value\_counts().plot(kind="bar",

title="Graduation Year Distribution")

plt.show()





**Conclusion:** The distribution of students across various graduation years reveals the representation of different academic batches. It assists in identifying trends over time, planning event targeting, and understanding how different graduation cohorts compare in academic performance, participation, and career expectations.

1. **What is the distribution of student’s experience with Python programming?**

print("Python experience distribution:\n",

df["Experience with python (Months)"].value\_counts())

df["Experience with python (Months)"].plot(kind="hist",

bins=10, title="Python Experience (Months)")

plt.xlabel("Months")

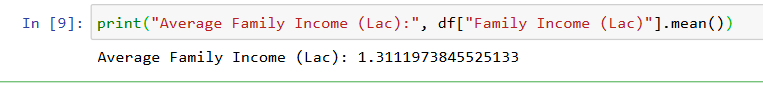
plt.show()



**Conclusion:** Analysing the distribution of Python experience provides insight into students’ technical skill levels. It is essential for assessing readiness for tech roles and helps in tailoring training programs or internships to support less-experienced students while recognizing highly experienced ones for advanced opportunities.

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

print("Average Family Income (Lac):", df["Family Income (Lac)"].mean())



**Conclusion:** The average family income reflects the economic background of the student population. Understanding income levels aids in analysing financial influences on education, identifying students needing support, and correlating economic status with factors such as academic performance, career aspirations, and participation in events.

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

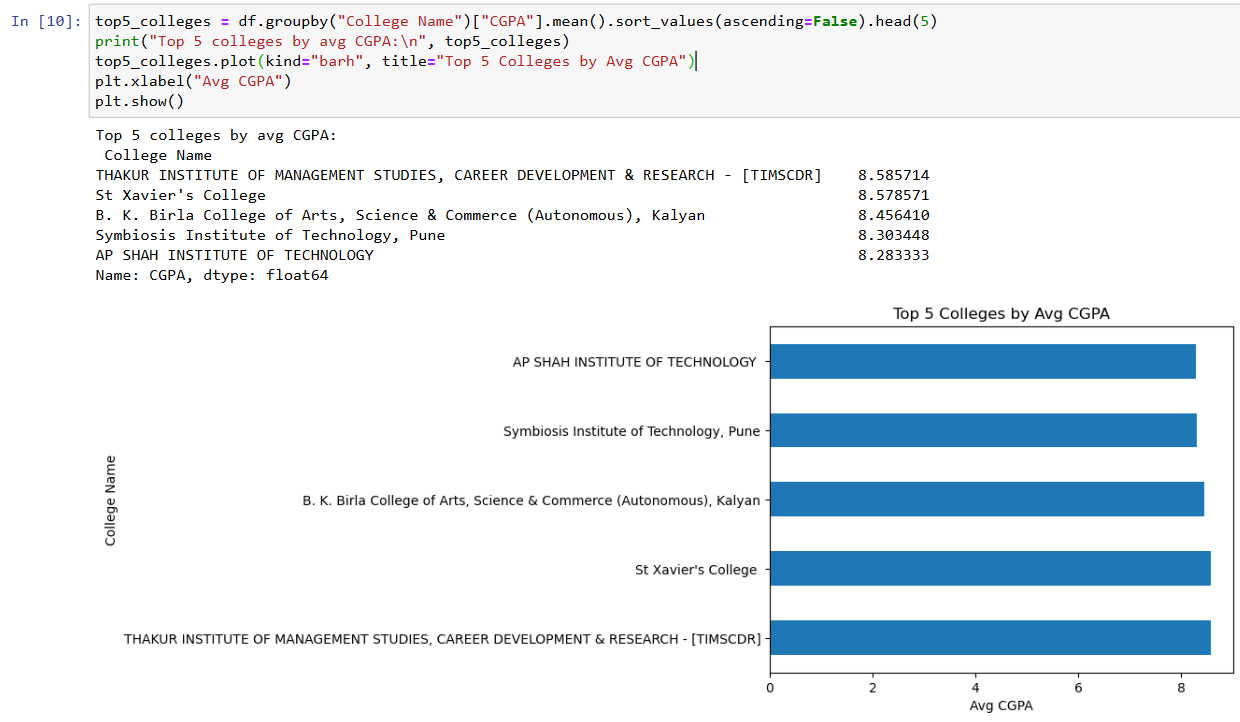
top5\_colleges = df.groupby("College Name")["CGPA"].mean().sort\_values(ascending=False).head(5)

print("Top 5 colleges by avg CGPA:\n", top5\_colleges)

top5\_colleges.plot(kind="barh", title="Top 5 Colleges by Avg CGPA")

plt.xlabel("Avg CGPA")

plt.show()



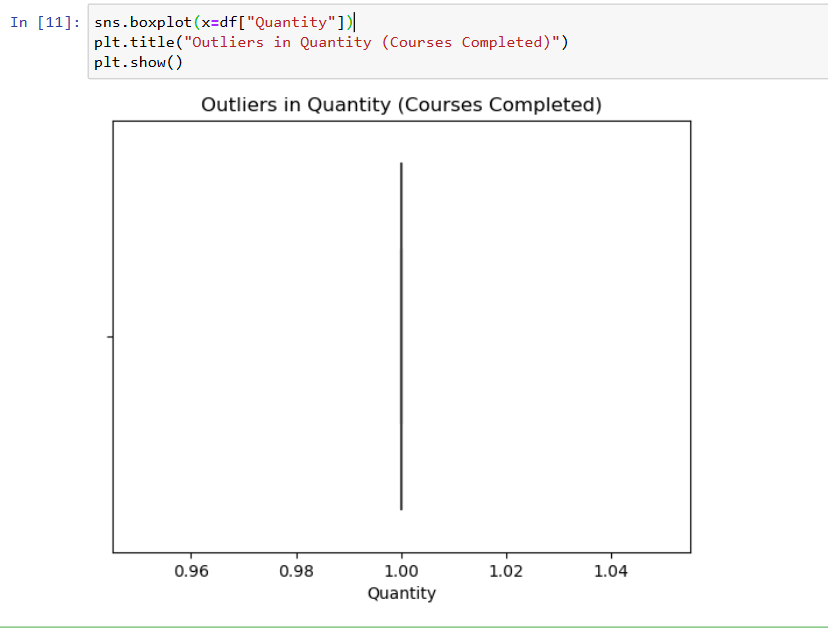
**Conclusion:** GPA variation across colleges highlights academic performance trends and institutional differences. Identifying top-performing colleges helps benchmark standards, recognize high achievers, and understand the educational quality across institutions. This can guide recruitment and resource allocation for colleges and employers alike.

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

sns.boxplot(x=df["Quantity"])

plt.title("Outliers in Quantity (Courses Completed)")

plt.show()



**Conclusion:** Detecting outliers in course quantity helps uncover students with exceptionally high or low engagement. These anomalies may indicate motivated learners or those requiring academic intervention. Recognizing such patterns is useful for improving curriculum structure, mentoring efforts, and personalized academic planning.

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

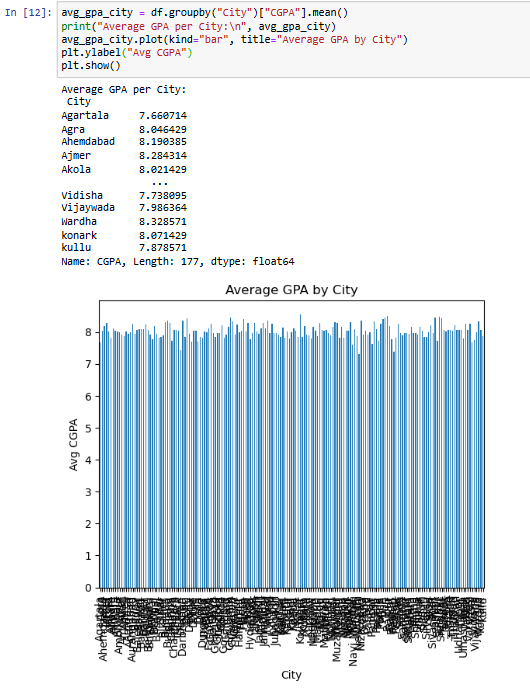
avg\_gpa\_city = df.groupby("City")["CGPA"].mean()

print("Average GPA per City:\n", avg\_gpa\_city)

avg\_gpa\_city.plot(kind="bar", title="Average GPA by City")

plt.ylabel("Avg CGPA")

plt.show()

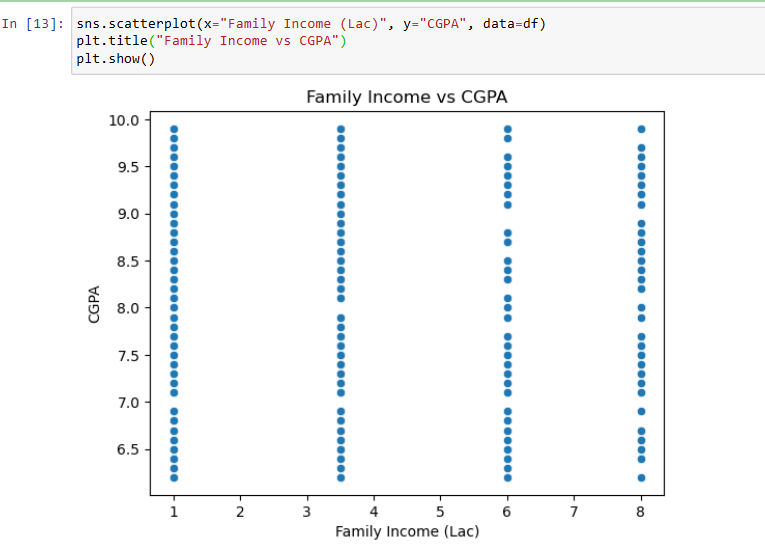


1. **Can we identify any relationship between family income and GPA?**

sns.scatterplot(x="Family Income (Lac)", y="CGPA", data=df)

plt.title("Family Income vs CGPA")

plt.show()

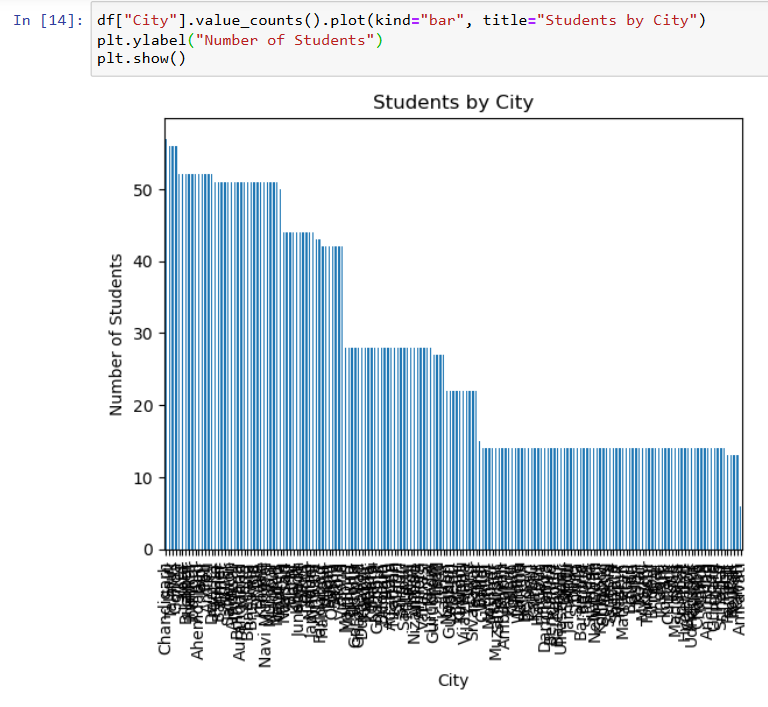


1. **How many students are from various cities? (Visualization)**

df["City"].value\_counts().plot(kind="bar", title="Students by City")

plt.ylabel("Number of Students")

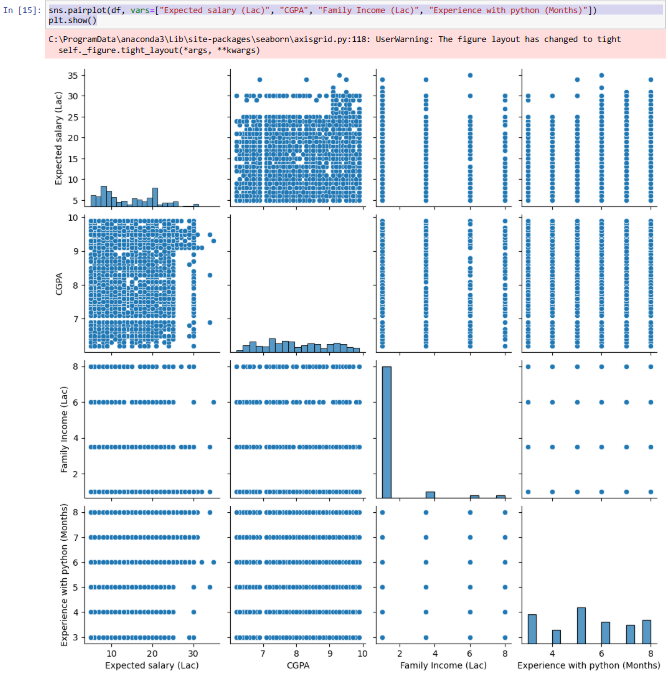
plt.show()



1. **How does the expected salary vary with GPA, family income, and Python experience?**

sns.pairplot(df, vars=["Expected salary (Lac)", "CGPA", "Family Income (Lac)", "Experience with python (Months)"])

plt.show()

****

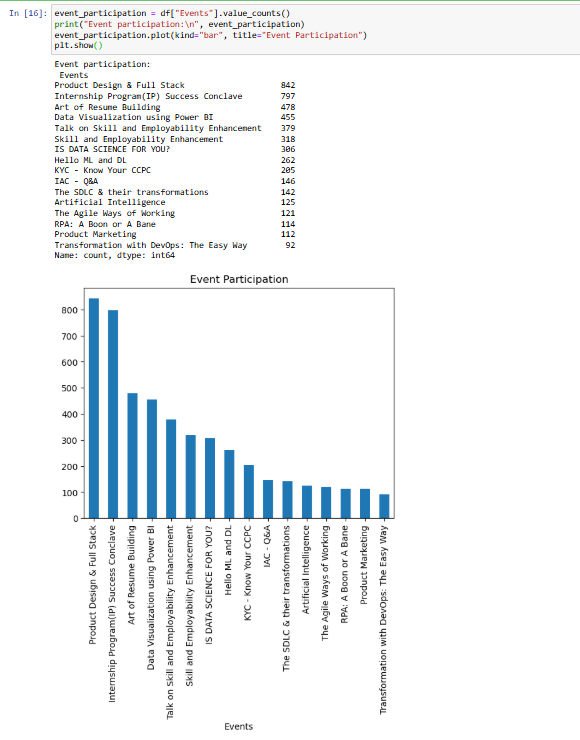
1. **Which event tends to attract more students from specific fields of study?**

event\_participation = df["Events"].value\_counts()

print("Event participation:\n", event\_participation)

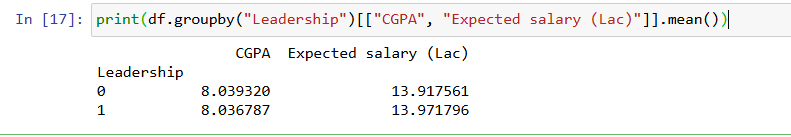
event\_participation.plot(kind="bar", title="Event Participation")

plt.show()



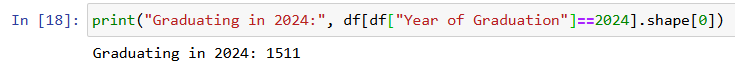
1. **Do students in leadership positions during their college years tend to have higher GPAs or better expected salary**

print(df.groupby("Leadership")[["CGPA", "Expected salary (Lac)"]].mean())



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

print("Graduating in 2024:", df[df["Year of Graduation"]==2024].shape[0])



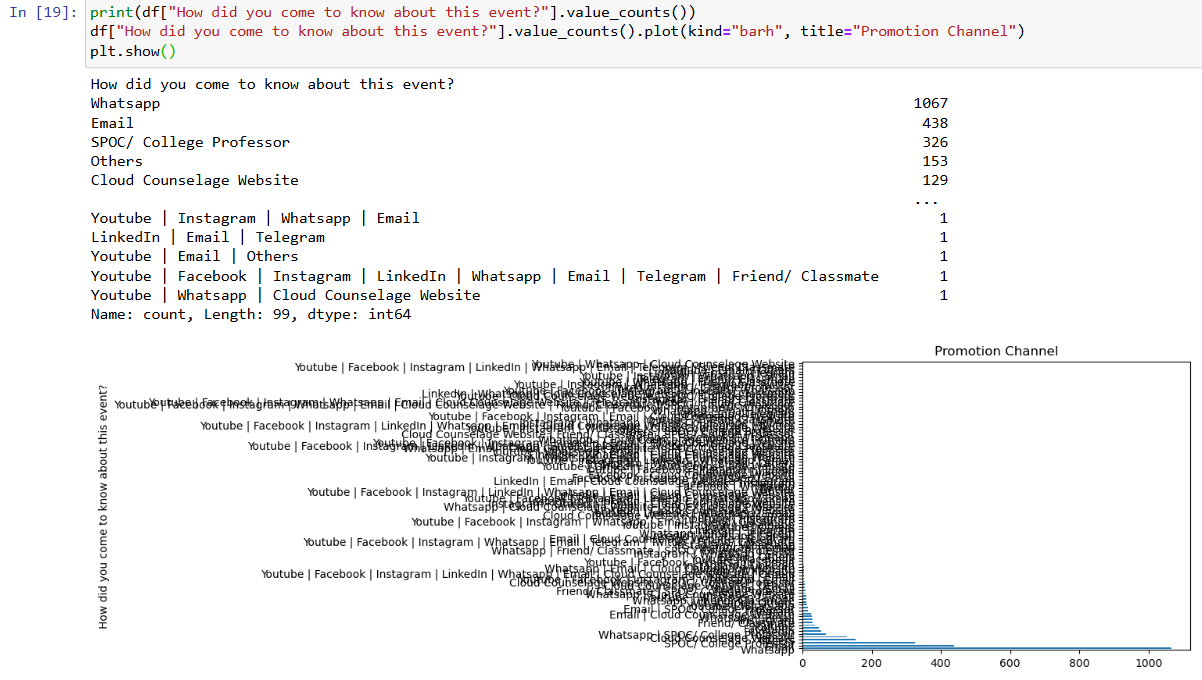
**Conclusion:** Tracking students graduating by 2024 helps with forecasting job market entrants, tailoring career support services, and planning graduation events. It also aids in understanding upcoming workforce trends, employer demand, and academic cycle planning for institutions and recruiters.

1. **Which promotion channel brings in more student participation for the event?**

print(df["How did you come to know about this event?"].value\_counts())

df["How did you come to know about this event?"].value\_counts().plot(kind="barh", title="Promotion Channel")

plt.show()

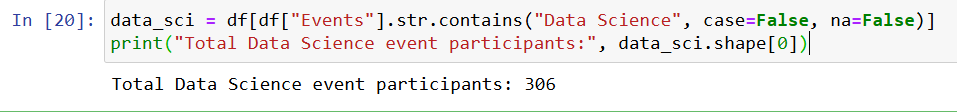


**Conclusion:** Analysing promotion channels reveals which methods most effectively engage students. This information enables better marketing strategies, maximizes event reach, and improves engagement. Leveraging high-performing channels can significantly increase attendance and participation in future academic or career-oriented events.

1. **Find the total number of students who attended the event related to Data Science? (From all Data Science related courses.**

data\_sci = df[df["Events"].str.contains("Data Science", case=False, na=False)]

print("Total Data Science event participants:", data\_sci.shape[0])



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

high = df[(df["CGPA"] > df["CGPA"].mean()) & (df["Experience with python (Months)"] > df["Experience with python (Months)"].mean())]

print("Average expected salary (high CGPA + python exp):", high["Expected salary (Lac)"].mean())

