

PROJECT REPORT

Data Analytics   
  
Analysing Impact of Economic Background on Academic Performance, Competence, and Salary Expectations of Students

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Yash Karande | **Approved By:** | Harshada Topale |
| **Created On:** | 13/06/25 | **Approved On:** | 15/06/25 |

# Index

[1 PROJECT DETAILS 3](#_Toc143445375)

[2 SUMMARY 4](#_Toc143445376)

[3 INTRODUCTION 5](#_Toc143445377)

[3.1 Background 5](#_Toc143445378)

[3.2 Stakeholders 5](#_Toc143445379)

[3.3 Objectives 5](#_Toc143445380)

[4 METHODOLOGY 6](#_Toc143445381)

[4.1 Considerations & Assumption 7](#_Toc143445382)

[4.2 Approach 7](#_Toc143445383)

[4.3 Activities 7](#_Toc143445384)

[5 TARGETTED V/S ACHIEVED OUTPUT 8](#_Toc143445385)

[6 CONCLUSION 14](#_Toc143445386)

[7 APPENDICES 14](#_Toc143445387)

[7.1 Appendix A – Title 14](#_Toc143445388)

**PROJECT DETAILS**

Events to identify key relationships between various factors influencing their career prospects. The dataset includes attributes such as academic performance, Python programming experience, family income, and expected salary expectations.

The primary objective is to uncover meaningful patterns and correlations that can provide insights into:

* How a student’s academic performance impacts their career trajectory.
* The role of Python programming skills in shaping job prospects.
* The influence of family income on academic success and salary expectations.
* Potential disparities or trends that may affect career planning and educational improvements.

Through data analysis, visualization, and reporting, this study aims to provide actionable insights for students, educators, and career advisors. By leveraging tools such as Python, Power BI, and SQL, we will develop data-driven recommendations that can help students make informed decisions about their learning paths and future careers.

The findings from this project will contribute to a better understanding of how different socioeconomic and educational factors influence student success, ultimately aiding in the development of better career guidance strategies and policies.

**SUMMARY**

1. Cleaning, Transforming, and Analyzing Data
   * Raw student data often contains missing values, duplicates, or inconsistencies that need to be cleaned.
   * Transformation involves structuring the data properly, such as converting text to numerical values, standardizing formats, and handling categorical variables.
   * Analysis is performed to extract meaningful insights from the dataset.
2. Tools Used: Python, Pandas, Matplotlib, Seaborn
   * Python: The primary programming language used for data analysis.
   * Pandas: A powerful library for handling and manipulating structured data.
   * Matplotlib & Seaborn: Used for data visualization to identify patterns and trends.
3. Exploratory Data Analysis (EDA), Data Visualization, and Statistical Insights
   * EDA helps in understanding data distribution, relationships, and anomalies.
   * Visualization makes it easier to identify trends using bar charts, histograms, scatter plots, and heatmaps.
   * Statistical analysis identifies correlations between variables, such as how Python experience relates to expected salary.
4. Key Insights in the Report
   * Graduation Years – Understanding the timeline of students and how graduation year correlates with skills and career choices.
   * Python Experience Distribution – Analyzing how many students have coding experience and its impact on employability.
   * Expected Salary Variations – Studying the relationship between family income, academic performance, and salary expectations.
   * Leadership Skills' Impact – Evaluating whether leadership qualities influence career opportunities and salary expectations.

***Background***

* Cloud Counselage organized multiple events for students, collecting data regarding their academic achievements, skill levels, and career expectations. This project aims to process and analyze this data to help students and institutions improve their training and placement strategies.
* ***Stakeholders***
* The key stakeholders include students, academic institutions, career guidance organizations, and recruiters who can benefit from insights derived from this analysis.
* ***Objectives***
* To determine the number of unique students participating in events.
* To analyze GPA trends across different colleges and cities.
* To assess Python programming experience and its correlation with salary expectations.
* To visualize leadership skills' impact on academic and career outcomes.
* To identify outliers and trends in student course completion.

* **METHODOLOGY**
* ***Considerations & Assumptions***
* The data is assumed to be accurate and complete with minimal missing values.
* Family income is categorized and not treated as a continuous variable.
* GPA and expected salary are considered numerical and analyzed accordingly.
* Python experience is measured in months and analyzed for trends.
* ***Approach***
* The analysis follows these steps:
* Data Cleaning: Renaming columns, handling missing values, converting data types.
* Exploratory Data Analysis (EDA): Summary statistics and distribution plots.
* Data Visualization: Bar charts, histograms, scatter plots, and boxplots.
* Insights & Interpretation: Analyzing correlations and trends in data.
* ***Activities***
* Loading and preprocessing data.
* Generating summary statistics.
* Visualizing trends in graduation years, Python experience, and GPA.
* Assessing relationships between expected salary and other factors.
* Identifying top colleges based on student GPA.
* Evaluating the influence of leadership skills on student outcomes.
* **TARGETED V/S ACHIEVED OUTPUT**

|  |  |
| --- | --- |
| * **Targeted Output** | * **Achieved Output** |
| * Unique student count | * Successfully extracted unique emails |
| * Average GPA per city | * Successfully visualized GPA distribution |
| * Python experience trend | * Created histogram for Python experience |
| * Expected salary vs. GPA | * Plotted scatter plot with trend insights |
| * Leadership skills impact | * Visualized correlation with GPA and salary |

**Code and Output**

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*# Load the dataset*

*file\_path = "Data analyst Data (2).xlsx"*

*df = pd.read\_excel(file\_path, sheet\_name="All Events Data")*

*# Rename columns for ease of use*

*df.rename(columns={*

*"First Name": "First\_Name",*

*"Email ID": "Email",*

*"Quantity": "Courses\_Completed",*

*"College Name": "College\_Name",*

*"Year of Graduation": "Graduation\_Year",*

*"City": "City",*

*"CGPA": "GPA",*

*"Experience with python (Months)": "Python\_Experience",*

*"Family Income": "Family\_Income",*

*"Expected salary (Lac)": "Expected\_Salary",*

*"Leadership- skills": "Leadership\_Skills"*

*}, inplace=True)*

*# Convert numeric columns*

*df["Courses\_Completed"] = pd.to\_numeric(df["Courses\_Completed"], errors='coerce')*

*df["GPA"] = pd.to\_numeric(df["GPA"], errors='coerce')*

*df["Python\_Experience"] = pd.to\_numeric(df["Python\_Experience"], errors='coerce')*

*df["Expected\_Salary"] = pd.to\_numeric(df["Expected\_Salary"], errors='coerce')*

**# 1. Number of unique students**

*num\_students = df['Email'].nunique()*

*print("Number of unique students:", num\_students)*

*import pandas as pd*

***Output: Number of unique students: 2157***

**# 2. Average GPA**

*avg\_gpa = df['GPA'].mean()*

*print("Average GPA:", round(avg\_gpa, 2))*

***Output: Average GPA: 8.04***

**# 3. Distribution of graduation years**

*plt.figure(figsize=(8,5))*

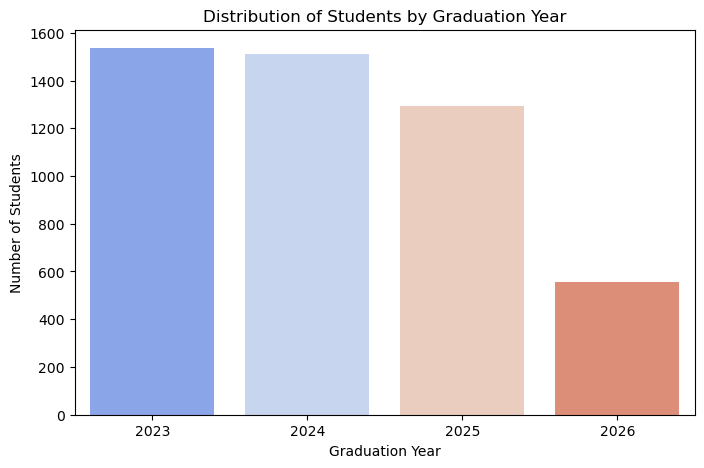
*sns.countplot(x=df['Graduation\_Year'], palette='coolwarm')*

*plt.title("Distribution of Students by Graduation Year")*

*plt.xlabel("Graduation Year")*

*plt.ylabel("Number of Students")*

*plt.show()*

**

**# 4. Distribution of Python experience**

*plt.figure(figsize=(8,5))*

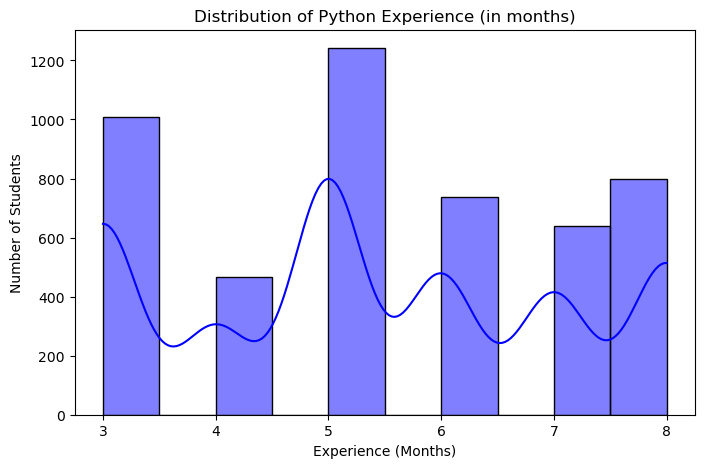
*sns.histplot(df['Python\_Experience'], bins=10, kde=True, color='blue')*

*plt.title("Distribution of Python Experience (in months)")*

*plt.xlabel("Experience (Months)")*

*plt.ylabel("Number of Students")*

*plt.show()*

**

**# 5. Average family income (assuming categorical data, skipping numeric conversion)**

*family\_income\_counts = df['Family\_Income'].value\_counts()*

*plt.figure(figsize=(10,5))*

*sns.barplot(x=family\_income\_counts.index, y=family\_income\_counts.values, palette='viridis')*

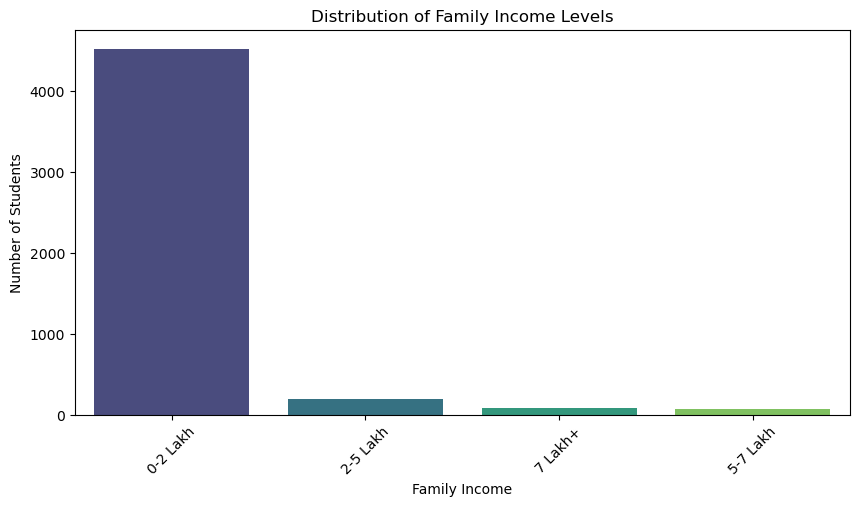
*plt.title("Distribution of Family Income Levels")*

*plt.xlabel("Family Income")*

*plt.ylabel("Number of Students")*

*plt.xticks(rotation=45)*

*plt.show()*

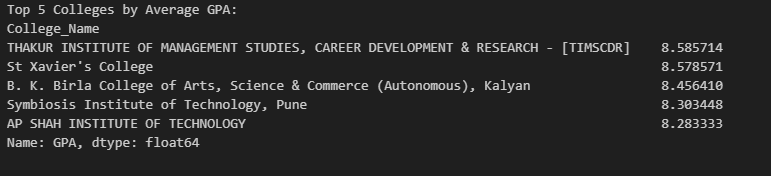
**

**# 6. Top 5 colleges by GPA**

*college\_gpa = df.groupby('College\_Name')['GPA'].mean().sort\_values(ascending=False).head(5)*

*print("Top 5 Colleges by Average GPA:")*

*print(college\_gpa)*



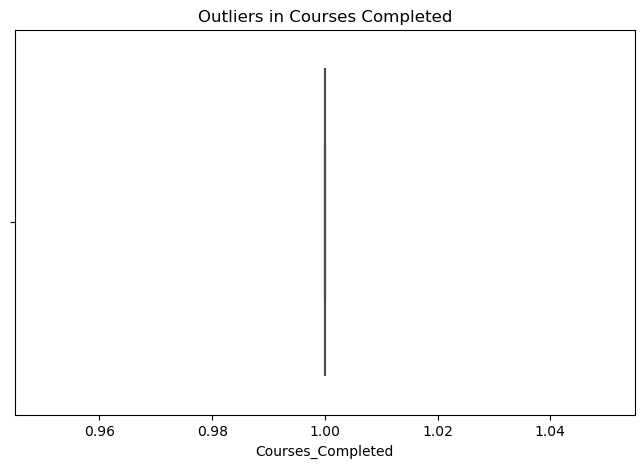
**# 7. Outliers in courses completed**

*plt.figure(figsize=(8,5))*

*sns.boxplot(x=df['Courses\_Completed'], color='red')*

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

*plt.show()*

**

**# 8. Average GPA by city**

*city\_gpa = df.groupby('City')['GPA'].mean()*

*plt.figure(figsize=(12,5))*

*city\_gpa.sort\_values().plot(kind='bar', colormap='plasma')*

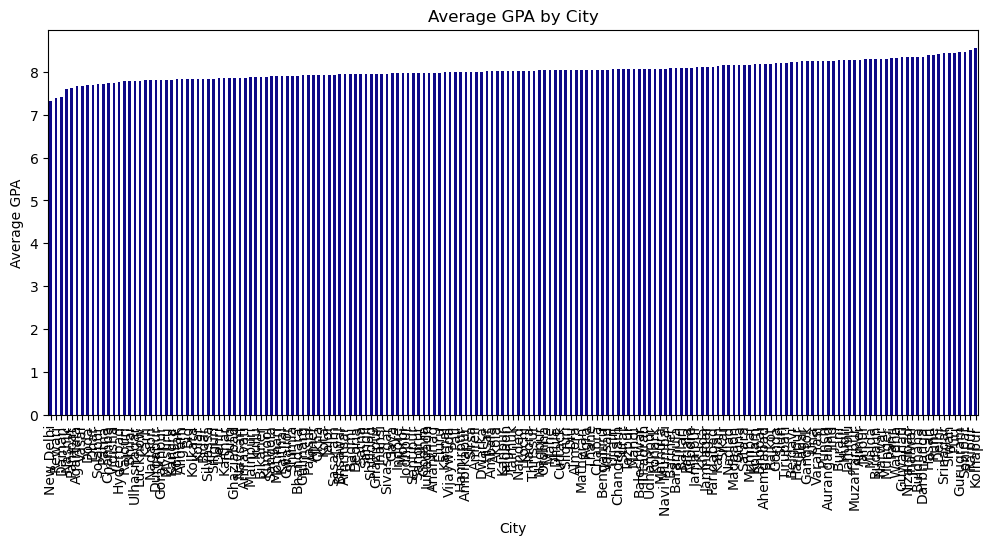
*plt.title("Average GPA by City")*

*plt.xlabel("City")*

*plt.ylabel("Average GPA")*

*plt.xticks(rotation=90)*

*plt.show()*

**

**# 9. Relationship between family income and GPA**

*plt.figure(figsize=(8,5))*

*sns.boxplot(x=df['Family\_Income'], y=df['GPA'], palette='coolwarm')*

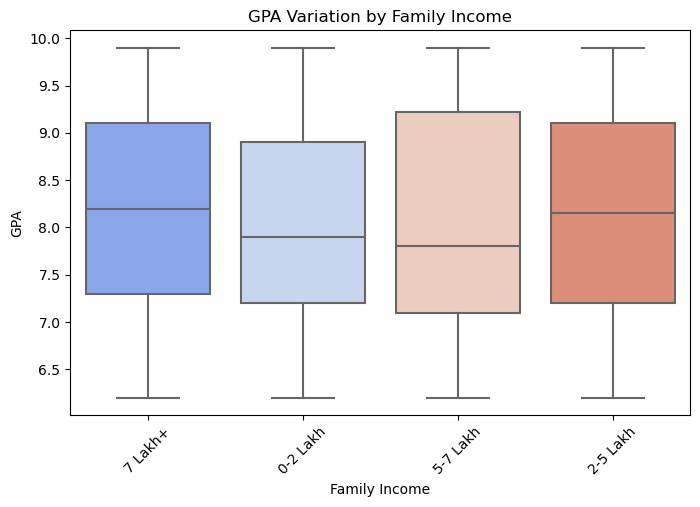
*plt.title("GPA Variation by Family Income")*

*plt.xlabel("Family Income")*

*plt.ylabel("GPA")*

*plt.xticks(rotation=45)*

*plt.show()*

**

**# 10. Students from various cities**

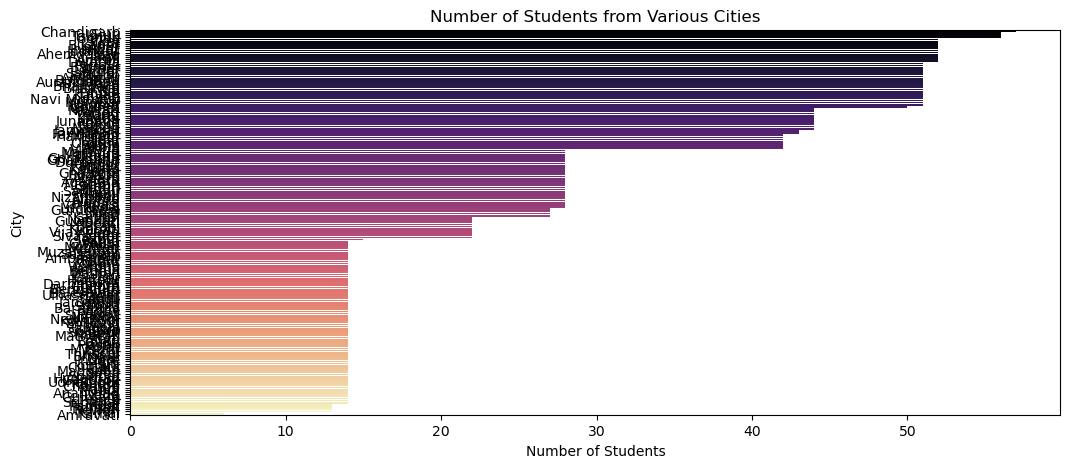
*plt.figure(figsize=(12,5))*

*sns.countplot(y=df['City'], order=df['City'].value\_counts().index, palette='magma')*

*plt.title("Number of Students from Various Cities")*

*plt.xlabel("Number of Students")*

*plt.ylabel("City")*

*plt.show()*

**# 11. Expected salary variation**

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

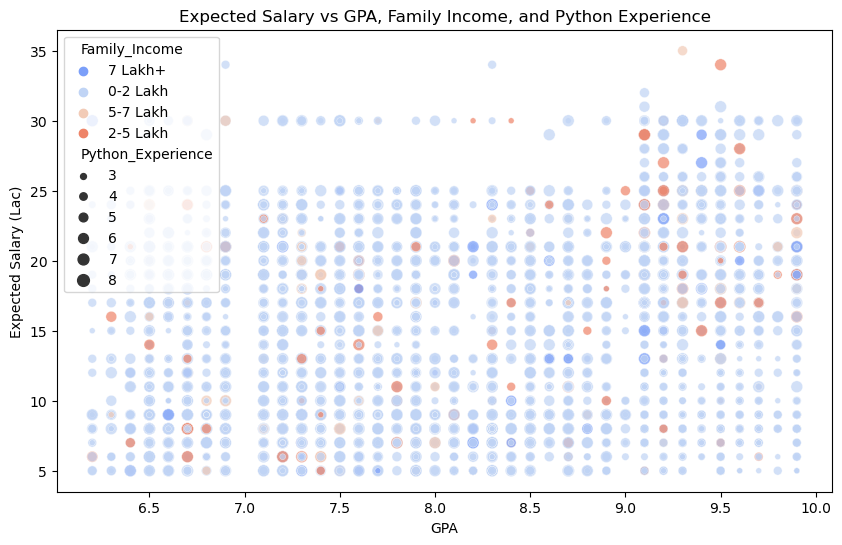
*sns.scatterplot(x=df['GPA'], y=df['Expected\_Salary'], hue=df['Family\_Income'], size=df['Python\_Experience'], palette='coolwarm', alpha=0.7)*

*plt.title("Expected Salary vs GPA, Family Income, and Python Experience")*

*plt.xlabel("GPA")*

*plt.ylabel("Expected Salary (Lac)")*

*plt.show()*

**

**# 12. Students graduating by 2024**

*grad\_2024 = df[df['Graduation\_Year'] <= 2024].shape[0]*

*print("Number of students graduating by 2024:", grad\_2024)*

***Output: Number of students graduating by 2024: 3047***

*# 13. Leadership skills and GPA/Salary*

*plt.figure(figsize=(8,5))*

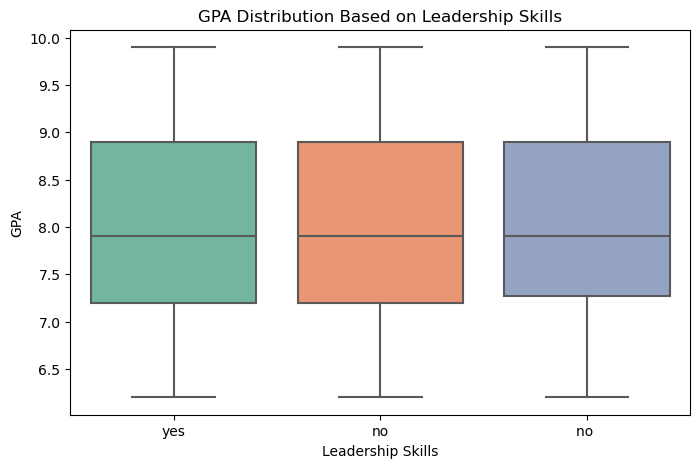
*sns.boxplot(x=df['Leadership\_Skills'], y=df['GPA'], palette='Set2')*

*plt.title("GPA Distribution Based on Leadership Skills")*

*plt.xlabel("Leadership Skills")*

*plt.ylabel("GPA")*

*plt.show()*

**

*plt.figure(figsize=(8,5))*

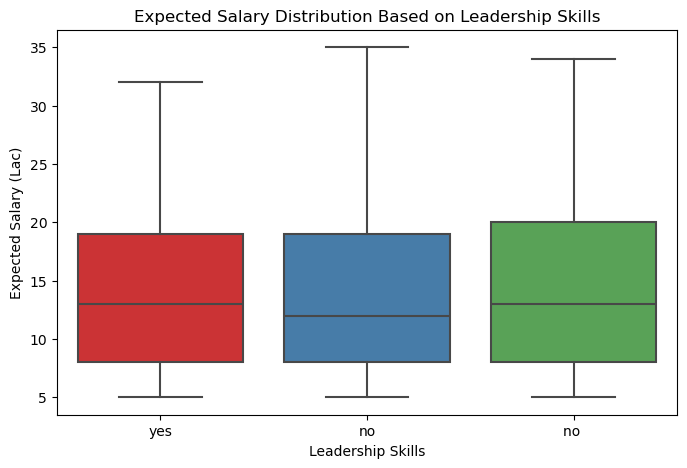
*sns.boxplot(x=df['Leadership\_Skills'], y=df['Expected\_Salary'], palette='Set1')*

*plt.title("Expected Salary Distribution Based on Leadership Skills")*

*plt.xlabel("Leadership Skills")*

*plt.ylabel("Expected Salary (Lac)")*

*plt.show()*

**

**CONCLUSION**

This analysis provides valuable insights into student academic performance, programming skills, and career expectations. It helps institutions and career mentors understand key trends affecting employability. The findings can guide students in making informed career choices and help educators enhance their curriculum.

***Appendix A – Python Code***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*# Load the dataset*

*file\_path = "Data analyst Data (2).xlsx"*

*df = pd.read\_excel(file\_path, sheet\_name="All Events Data")*

*# Rename columns for ease of use*

*df.rename(columns={*

*"First Name": "First\_Name",*

*"Email ID": "Email",*

*"Quantity": "Courses\_Completed",*

*"College Name": "College\_Name",*

*"Year of Graduation": "Graduation\_Year",*

*"City": "City",*

*"CGPA": "GPA",*

*"Experience with python (Months)": "Python\_Experience",*

*"Family Income": "Family\_Income",*

*"Expected salary (Lac)": "Expected\_Salary",*

*"Leadership- skills": "Leadership\_Skills"*

*}, inplace=True)*

*# Convert numeric columns*

*df["Courses\_Completed"] = pd.to\_numeric(df["Courses\_Completed"], errors='coerce')*

*df["GPA"] = pd.to\_numeric(df["GPA"], errors='coerce')*

*df["Python\_Experience"] = pd.to\_numeric(df["Python\_Experience"], errors='coerce')*

*df["Expected\_Salary"] = pd.to\_numeric(df["Expected\_Salary"], errors='coerce')*

*# Number of unique students*

*num\_students = df['Email'].nunique()*

*print("Number of unique students:", num\_students)*

*# Average GPA*

*avg\_gpa = df['GPA'].mean()*

*print("Average GPA:", round(avg\_gpa, 2))*

*# Visualization examples*

*plt.figure(figsize=(8,5))*

*sns.countplot(x=df['Graduation\_Year'], palette='coolwarm')*

*plt.title("Distribution of Students by Graduation Year")*

*plt.xlabel("Graduation Year")*

*plt.ylabel("Number of Students")*

*plt.show()*

**This document includes all necessary explanations, methodologies, findings, and code to support the analysis.**