# Data Analytics Project Document

**Name:** Jesreel Marbaniang

**Role:** Data Analytics Intern

**Organization:** Cloud Counselage

## Basic Questions:

1. How many unique students are there in the dataset?

   **Code:**

```
In [2]: import pandas as pd
        import matplotlib.pyplot as mpt

In [3]: readxl=pd.read_excel('Data analyst Data.xlsx')

In [4]: df=pd.DataFrame(readxl)

In [7]: df.rename(columns={'Email ID':'email_id'},inplace=True)

In [8]: df.email_id.nunique(dropna=True)
Out[8]: 2157
```

   **Conclusion:** There are 2157 unique students in the dataset. I have used email id to determine this as email id is unique to each student and there can be no duplicates. I had also dropped the students whose rows are having empty values.

2. What is the average GPA of the students?

**Code:**

```
In [1]: import pandas as pd

In [2]: readxl=pd.read_excel('Data analyst Data.xlsx')

In [3]: df=pd.DataFrame(readxl)

In [9]: student=df[df['Designation'].str.strip().str.lower().str.contains('student')].copy()
        student.dropna(subset=['CGPA'],inplace=True)
        student_average_cgpa=student['CGPA'].mean()

In [11]: print("Average student cgpa: ", student_average_cgpa)
         Average student cgpa:  8.0383194016718
```

**Conclusion:** The average CGPA of the students is 8.03

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

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
```

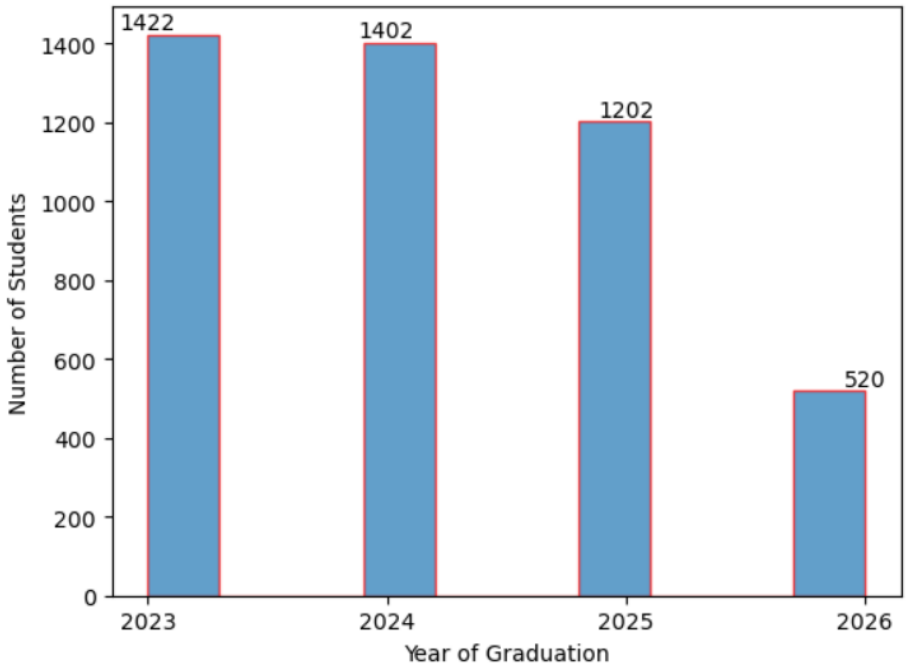
```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
df=pd.DataFrame(readxl)
```

```
student=df[df['Designation'].str.strip().str.lower().str.contains('student')].copy()
graduation= student['Year of Graduation']
```

```
plt.hist(graduation, bins=10, edgecolor='red', alpha=0.7)
plt.xlabel("Year of Graduation")
plt.ylabel("Number of Students")
plt.xticks(range(int(min(graduation)), int(max(graduation)) + 1))
for i in range(int(min(graduation)), int(max(graduation)) + 1):
    count = len(graduation[graduation== i])
    plt.text(i, count + 1, str(count), ha='center', va='bottom')
plt.show()
```

**Conclusion:**



4. What is the distribution of student's experience with Python programming?

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

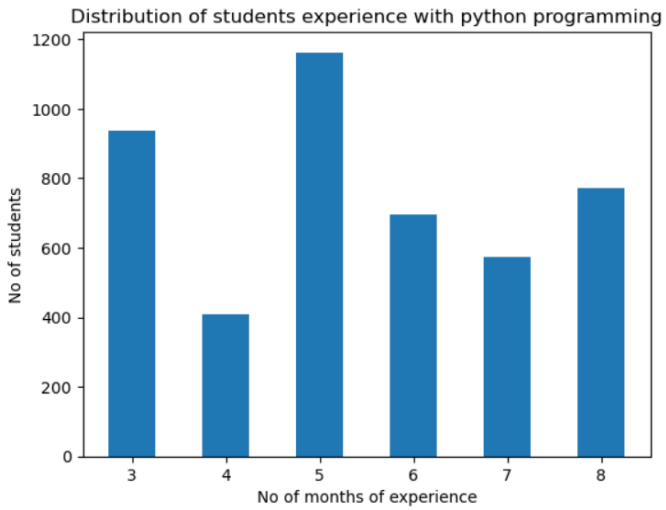
```
df=pd.DataFrame(readxl)
```

```
student=df[df['Designation'].str.strip().str.lower().str.contains('student')].copy()
student.dropna()
print(student['Experience with python (Months)'].unique())
```

```
[7 3 4 5 6 8]
```

```
std_exp1=student[student['Experience with python (Months)']==3]
std_exp2=student[student['Experience with python (Months)']==4]
std_exp3=student[student['Experience with python (Months)']==5]
std_exp4=student[student['Experience with python (Months)']==6]
std_exp5=student[student['Experience with python (Months)']==7]
std_exp6=student[student['Experience with python (Months)']==8]

std1len=len(std_exp1)
std2len=len(std_exp2)
std3len=len(std_exp3)
std4len=len(std_exp4)
std5len=len(std_exp5)
std6len=len(std_exp6)
```

## Conclusion:



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

**Code:**

```
import pandas as pd
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
df=pd.DataFrame(readxl)
```

```
student=df[df['Designation'].str.strip().str.lower().str.contains('student')].copy()
student.dropna()
print(student['Family Income'].unique())
```

```
['7 Lakh+' '0-2 Lakh' '5-7 Lakh' '2-5 Lakh']
```

```
std_inc1=student[student['Family Income']=='0-2 Lakh']
std_inc2=student[student['Family Income']=='2-5 Lakh']
std_inc3=student[student['Family Income']=='5-7 Lakh']
std_inc4=student[student['Family Income']=='7 Lakh+']

std1len=len(std_inc1)
std2len=len(std_inc2)
std3len=len(std_inc3)
std4len=len(std_inc4)
```

```
income_range=["0-200000","200000-500000","500000-700000","700000+"]
frequencies=[std1len,std2len,std3len,std4len]

def calculate_midpoint(range_str):
    if "+" in range_str:
        return int(income_range[income_range.index(range_str) - 1].split('-')[1])
    else:
        start, end = map(int, range_str.split('-'))
        return (start + end) / 2
weighted_sum = sum(f * calculate_midpoint(range_str) for f, range_str in zip(frequencies, income_range))
total_data_points = sum(frequencies)
overall_average_salary = weighted_sum / total_data_points
print("Overall average family income:", overall_average_salary)
```

```
Overall average family income: 127529.69643642764
```

**Conclusion:** The average family income is 127529.6964 Lakhs

6. How does GPA vary among different colleges? (Show top 5 results only)

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
df=pd.DataFrame(readxl)
```

```
average_gpa_by_college = df.groupby('College Name')['CGPA'].mean().sort_values(ascending=False)
```

```
average_gpa_by_college.dropna(inplace=True)
college_show=average_gpa_by_college.head()
college_show
```

```
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
```

```
college=college_show.to_dict()
College_name=list(college.keys())
College_CGPA=list(college.values())
listco=list(zip(College_name, College_CGPA))
df1 = pd.DataFrame(listco,columns=['Name of College', 'Average CGPA'])
df1
```

## Conclusion:

| | Name of College | Average CGPA |
|---|---|---|
| 0 | THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER... | 8.585714 |
| 1 | St Xavier's College | 8.578571 |
| 2 | B. K. Birla College of Arts, Science & Commerc... | 8.456410 |
| 3 | Symbiosis Institute of Technology, Pune | 8.303448 |
| 4 | AP SHAH INSTITUTE OF TECHNOLOGY | 8.283333 |

7. What is the average GPA for students in each city?

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
df=pd.DataFrame(readxl)
```

```
student=df[df['Designation'].str.strip().str.lower().str.contains('student')].copy()
student.dropna(inplace=True)
```

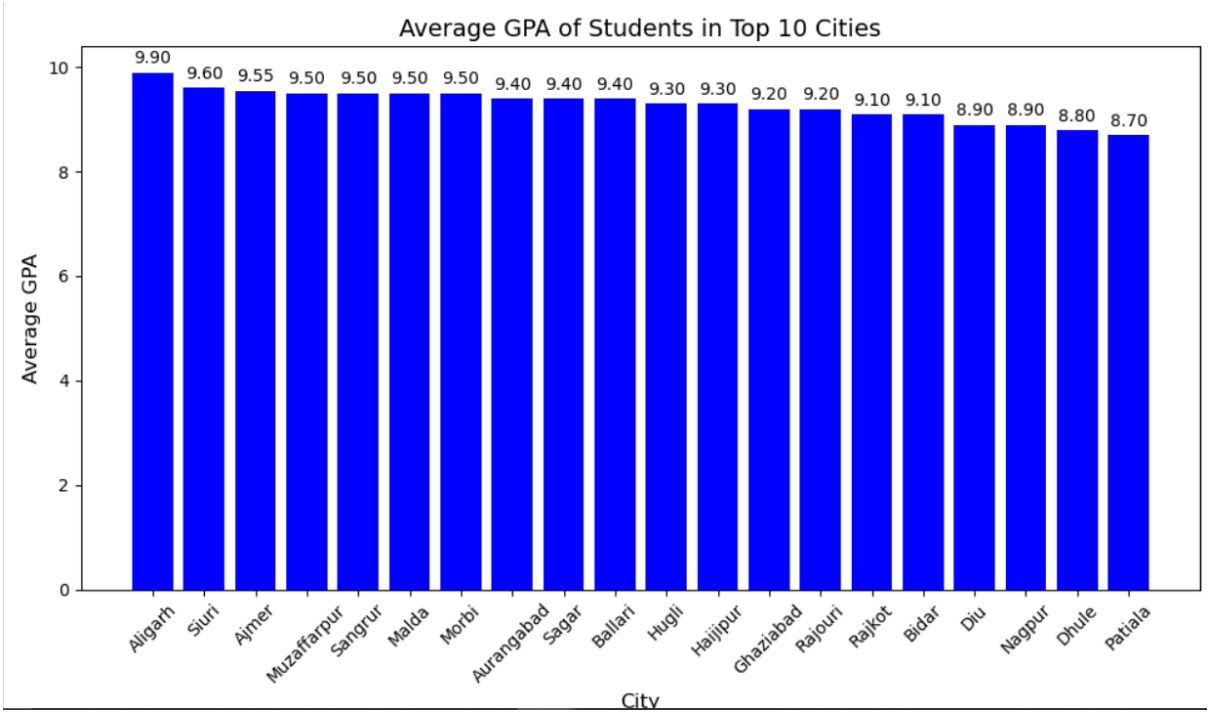
```
print(student['City'].unique())
```

```
['Amravati' 'Tezpur' 'Munger' 'Muzaffarpur' 'Diu' 'Faridabad' 'Rohtak'
 'Sirsa' 'Srinagar' 'Ballari' 'Hasan' 'Vidisha' 'Akola' 'Aurangabad'
 'Bhsawal' 'Amer' 'Jaipur' 'Hyderbad' 'Hugli' 'Silguri' 'Dhule' 'Morbi'
 'Ghaziabad' 'Malda' 'Sagar' 'Thane' 'Navi Mumbai' 'Dwarka' 'Nadiad'
 'Rajkot' 'Nagaon' 'Dehri' 'Haijipur' 'Gurugram' 'Hisar' 'Jind' 'Rewari'
 'Doda' 'Guna' 'Gwalior' 'Kolhapur' 'Ajmer' 'Gonda' 'Siuri' 'Mumbai'
 'Gaya' 'Patiala' 'Aligarh' 'Mathura' 'Sangrur' 'Siliguri' 'Kaithal'
 'Lucknow' 'Haora' 'Pali' 'Raipur' 'Orchha' 'Okha' 'Durgapur' 'Nagpur'
 'Jammu' 'Badmi' 'Bengaluru' 'Bidar' 'Sangli' 'Kota' 'Kanpur' 'Buldhana'
 'Kheda' 'Rajouri' 'Talmuk' 'Gulmarg']
```

```
student['CGPA']=pd.to_numeric(student['CGPA'], errors='coerce')
cgpa_by_city = student.groupby('City')['CGPA'].mean().reset_index()
cgpa_by_city= cgpa_by_city.sort_values(by='CGPA', ascending=False)
cgpa_by_city=cgpa_by_city.head(20)
```

```
plt.figure(figsize=(10, 6))
plt.bar(cgpa_by_city['City'], cgpa_by_city['CGPA'], color='blue')
plt.xlabel('City', fontsize=12)
plt.ylabel('Average GPA', fontsize=12)
plt.title('Average GPA of Students in Top 10 Cities', fontsize=14)
plt.xticks(rotation=45, fontsize=10)
plt.tight_layout()

for i, value in enumerate(cgpa_by_city['CGPA']):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom')

plt.show()
```

## Conclusion:



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

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
df=pd.DataFrame(readxl)
```

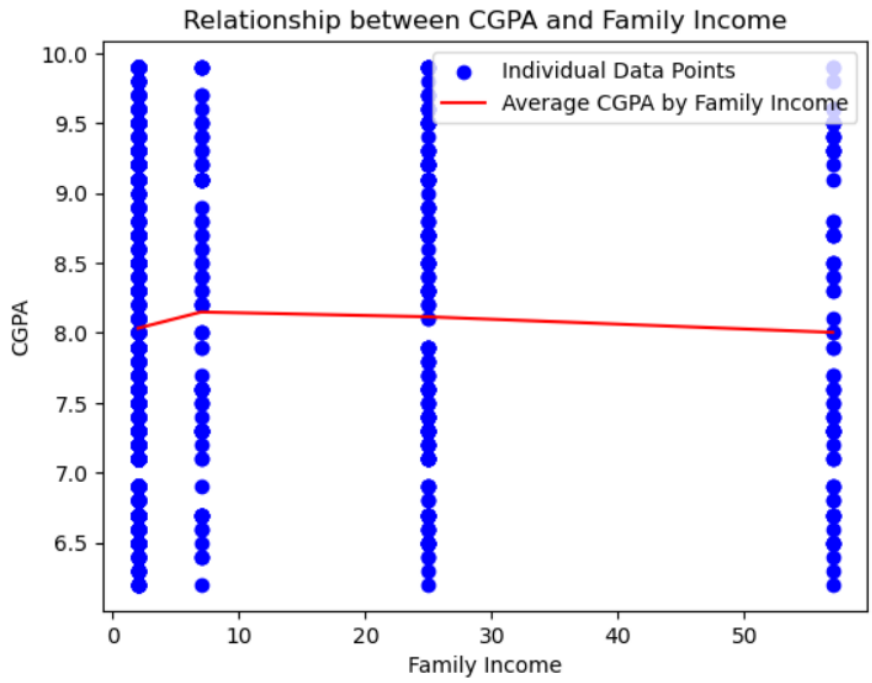
```
df['Family Income']=df['Family Income'].str.replace('[^\d.]','',regex=True)
df['Family Income']=pd.to_numeric(df['Family Income'],errors='coerce')
df['CGPA']=pd.to_numeric(df['CGPA'],errors='coerce')
```

```
data=df.dropna(subset=['Family Income','CGPA'])
```

```
cgpa_count=data['CGPA'].value_counts()
average_income_by_cgpa=data.groupby('Family Income')['CGPA'].mean()
```

```
plt.scatter(df['Family Income'],df['CGPA'], color='b', label='Individual Data Points')
plt.plot(average_income_by_cgpa.index, average_income_by_cgpa.values, color='r', label='Average CGPA by Family Income')
plt.xlabel('Family Income')
plt.ylabel('CGPA')
plt.title('Relationship between CGPA and Family Income')
plt.legend()
plt.show()
```

**Conclusion:**



## Moderate Questions:

1. How many students are from various cities? (Solve it using any data visualization tool)

### Code:

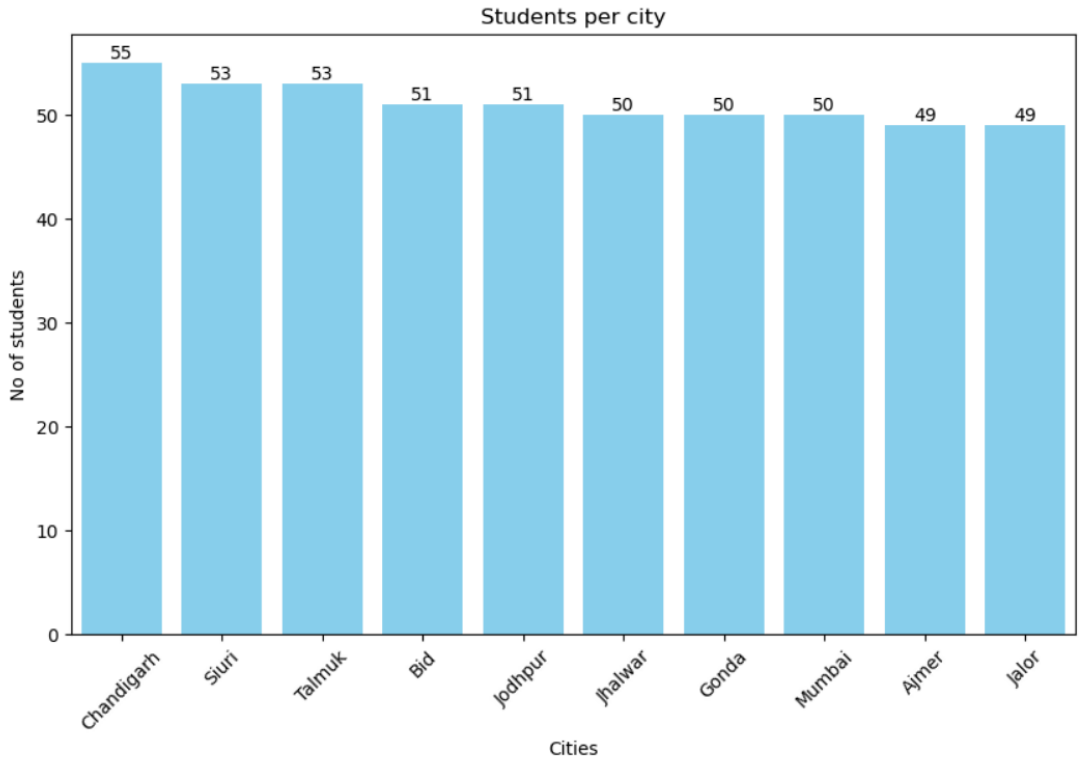
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
student_data=readxl[readxl['Designation']=='Students']
channel_count=student_data['City'].value_counts()
freq=channel_count.head(10)
```

```
plt.figure(figsize=(10, 6))
bars = plt.bar(freq.index, freq.values,color='skyblue')
freq.plot(kind='bar', color='skyblue')
plt.xlabel('Cities')
plt.ylabel('No of students')
plt.title('Students per city')
plt.xticks(rotation=45)

for i, value in enumerate(freq):
    plt.text(freq.index[i], value, f'{value}', ha='center', va='bottom')

plt.show()
```

### Conclusion:



2. How does expected salary vary based on factors like ‘GPA’, ‘Family Income’, ‘Experience with Python (months)’?

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
def categorize_income(value):
    if value < 300000:
        return 'Low'
    elif value < 700000:
        return 'Moderate'
    else:
        return 'High'
```

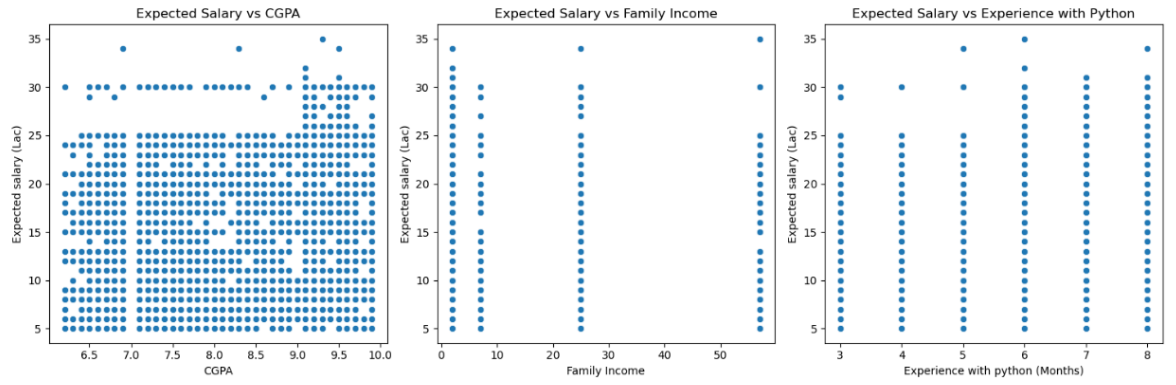
```
excel_sheet = pd.read_excel('Data analyst Data.xlsx')
```

```
excel_sheet['Family Income'] = excel_sheet['Family Income'].str.replace('[^\d.]', '', regex=True).astype(float)
excel_sheet['Family Income Category'] = excel_sheet['Family Income'].apply(categorize_income)
excel_sheet['CGPA'] = pd.to_numeric(excel_sheet['CGPA'], errors='coerce')
excel_sheet['Experience with python (Months)'] = pd.to_numeric(excel_sheet['Experience with python (Months)'], errors='coerce')
```

```
selected_data = excel_sheet.dropna(subset=['Family Income Category', 'CGPA', 'Experience with python (Months)', 'Expected salary
```

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.scatterplot(x='CGPA', y='Expected salary (Lac)', data=selected_data)
plt.title('Expected Salary vs CGPA')

plt.subplot(1, 3, 2)
sns.scatterplot(x='Family Income', y='Expected salary (Lac)', data=selected_data)
plt.title('Expected Salary vs Family Income')

plt.subplot(1, 3, 3)
sns.scatterplot(x='Experience with python (Months)', y='Expected salary (Lac)', data=selected_data)
plt.title('Expected Salary vs Experience with Python')

plt.tight_layout()
plt.show()
```

## Conclusion:



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

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
readxl['Expected salary (Lac)']=pd.to_numeric(readxl['Expected salary (Lac)'],errors='coerce')
readxl['Leadership- skills']=readxl['Leadership- skills'].str.strip()
readxl['CGPA'] = pd.to_numeric(readxl['CGPA'], errors='coerce')
```

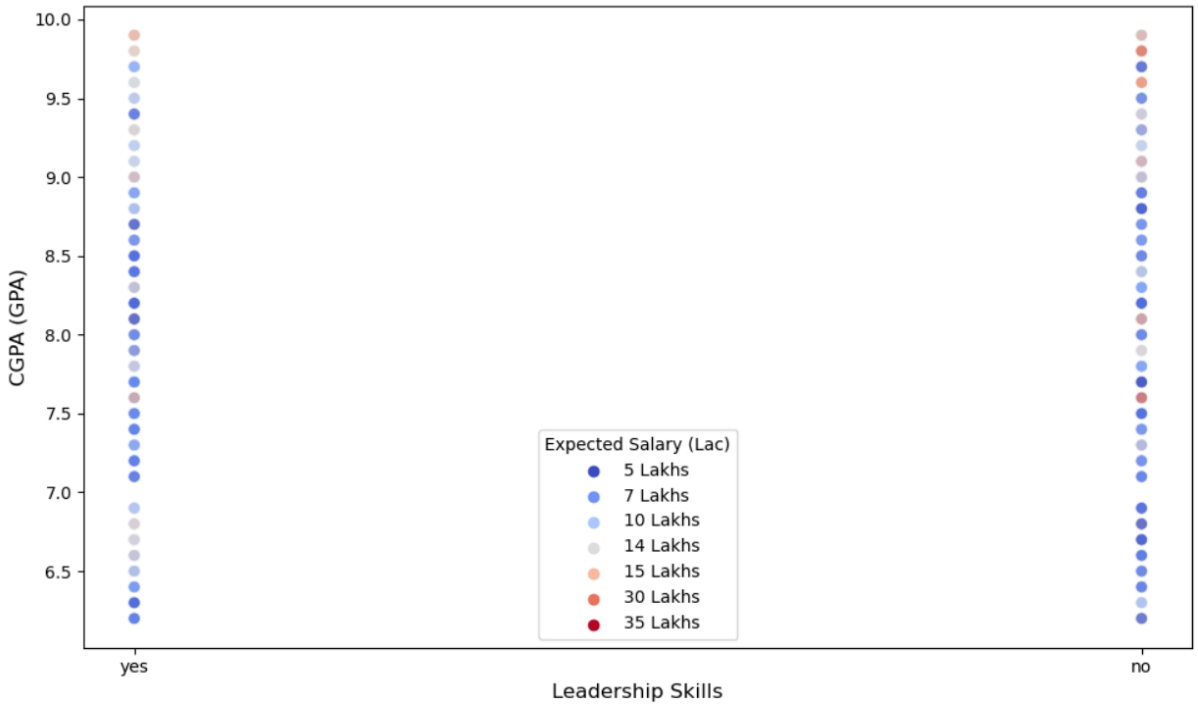
```
print(readxl['Expected salary (Lac)'].unique())
```

```
[ 6  7  8 10 11 20  5  9 21 13 15 14 16 19 17 18 23 25 22 12 24 30 28 26
 29 27 32 34 31 35]
```

```
selected_data = readxl.dropna(subset=['CGPA', 'Expected salary (Lac)','Leadership- skills'])
```

```
plt.figure(figsize=(10, 6))
plot = sns.scatterplot(x='Leadership- skills', y='CGPA', hue='Expected salary (Lac)', data=readxl, palette='coolwarm', s=50, alph

plt.xlabel('Leadership Skills', fontsize=12)
plt.ylabel('CGPA', fontsize=12)

legend = plot.legend(title='Expected Salary (Lac)')
legend_labels = ['5 Lakhs', '7 Lakhs', '10 Lakhs', '14 Lakhs', '15 Lakhs','30 Lakhs','35 Lakhs']
for text, label in zip(legend.texts, legend_labels):
    text.set_text(label)

plt.tight_layout()
plt.show()
```

## Conclusion:



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

## Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
readxl['Year of Graduation'] = pd.to_numeric(readxl['Year of Graduation'], errors='coerce')
students_graduating=readxl.dropna(subset=['Year of Graduation'])
```

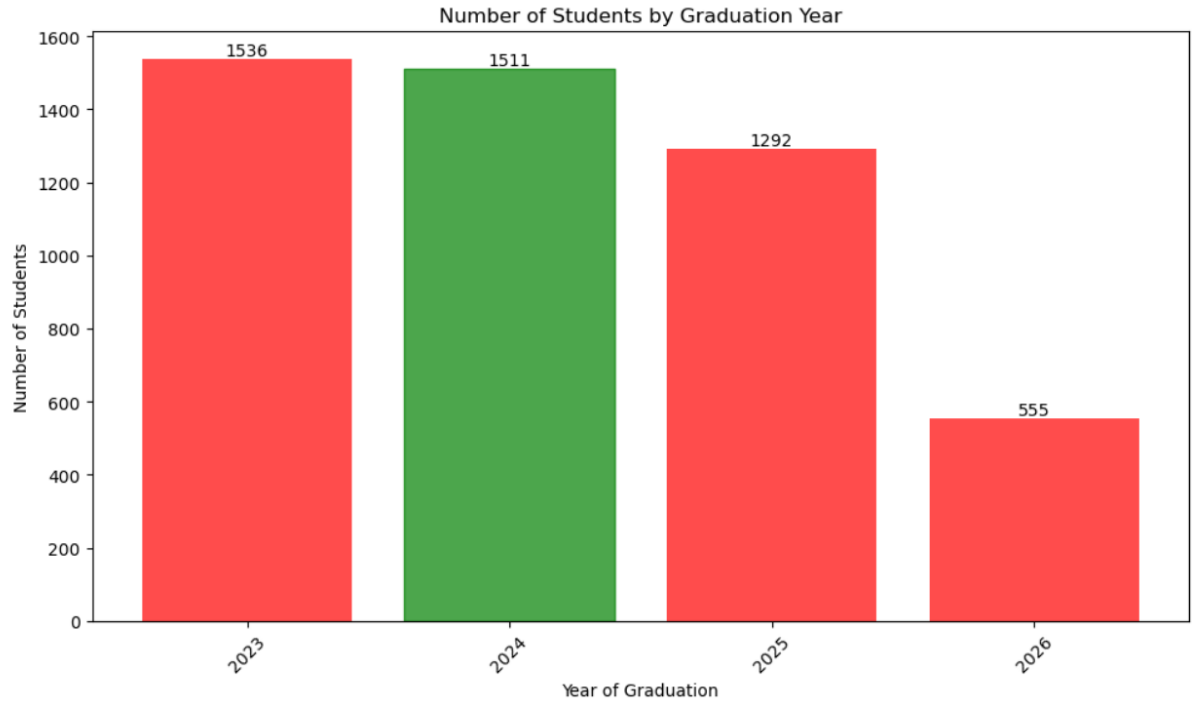
```
print(readxl['Year of Graduation'].unique())
```

```
[2024 2023 2025 2026]
```

```
graduation_year_counts = students_graduating['Year of Graduation'].value_counts().sort_index()
```

```
plt.figure(figsize=(10, 6))
bars = plt.bar(graduation_year_counts.index, graduation_year_counts.values, color='red', alpha=0.7)

for bar in bars:
    if bar.get_height() == graduation_year_counts[2024]:
        bar.set_color('green')

plt.xlabel('Year of Graduation')
plt.ylabel('Number of Students')
plt.title('Number of Students by Graduation Year')
plt.xticks(graduation_year_counts.index, rotation=45)

for i, value in enumerate(graduation_year_counts):
    plt.text(graduation_year_counts.index[i], value, f'{value}', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

## Conclusion:



5. Which promotion channels bring in more student participations in the event?

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
```

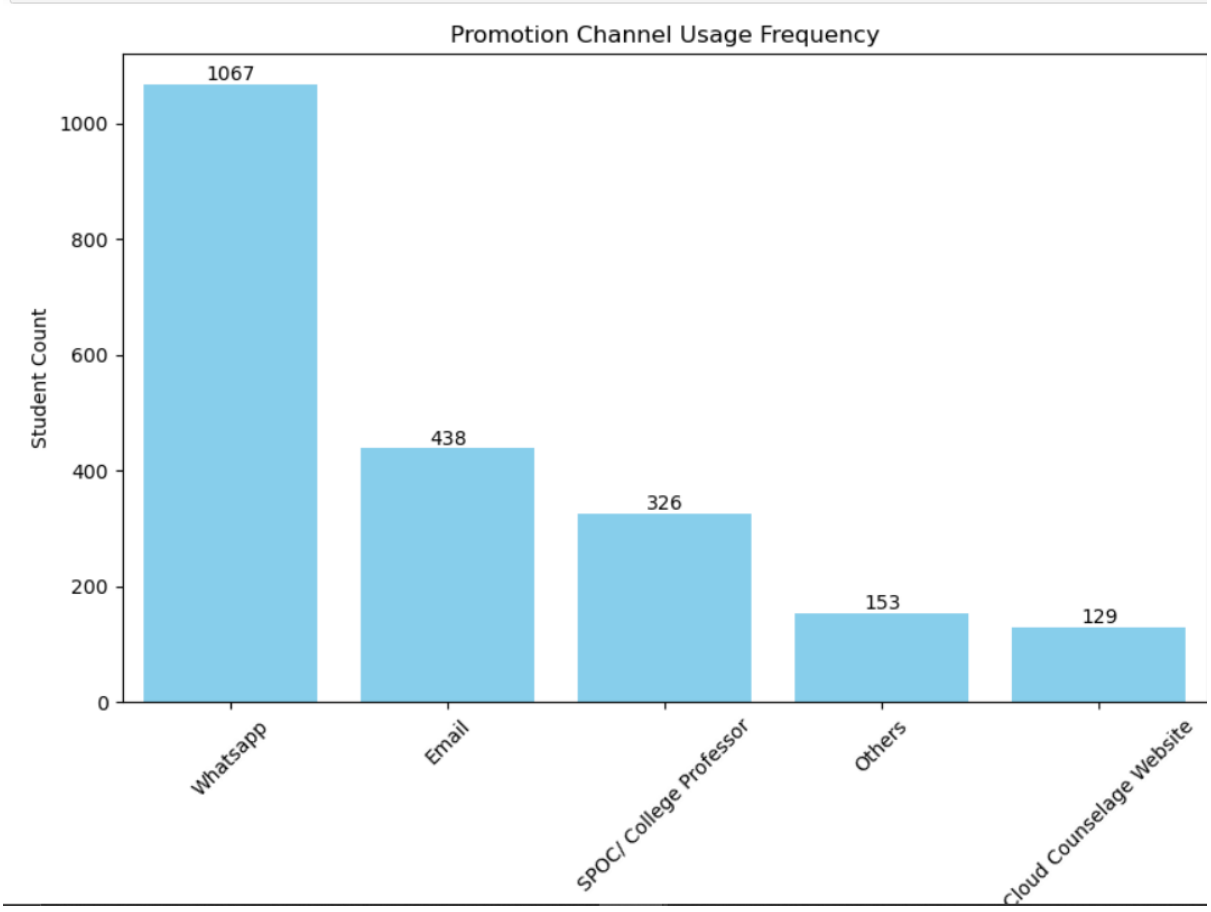
```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
student_data=readxl[readxl['Designation']=='Students']
channel_count=student_data['Promotion Channels'].value_counts()
freq=channel_count.head()
freq
```

```
Whatsapp                  1067
Email                      438
SPOC/ College Professor    326
Others                     153
Cloud Counselage Website   129
Name: Promotion Channels, dtype: int64
```

```
plt.figure(figsize=(10, 6))
freq.plot(kind='bar', color='skyblue')
plt.xlabel('Promotion Channel')
plt.ylabel('Student Count')
plt.title('Promotion Channel Usage Frequency')
plt.xticks(rotation=45)
plt.show()
```

**Conclusion:**



6. Find the total number of students who attended the events related to Data Science? (From all data science related courses)

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
student_data=readxl[readxl['Designation']=='Students']
```

```
student_data['Events'].unique()
```

```
array(['Art of Resume Building', 'Data Visualization using Power BI',
       'Artificial Intelligence', 'Hello ML and DL', 'Product Marketing',
       'IAC - Q&A', 'Internship Program(IP) Success Conclave',
       'IS DATA SCIENCE FOR YOU?', 'KYC - Know Your CCPC',
       'Product Design & Full Stack', 'RPA: A Boon or A Bane',
       'Skill and Employability Enhancement',
       'Talk on Skill and Employability Enhancement',
       'The Agile Ways of Working', 'The SDLC & their transformations',
       'Transformation with DevOps: The Easy Way'], dtype=object)
```

```
_keywords=['Data Science','Artificial Intelligence','ML','Machine Learning','Deep Learning','Data Visualization','Data Analysis']
```

```
data_science_events = student_data[student_data['Events'].str.contains('|'.join(data_science_keywords), case=False)]
```

```
event_count=data_science_events['Events'].value_counts()
freq=event_count
freq
```

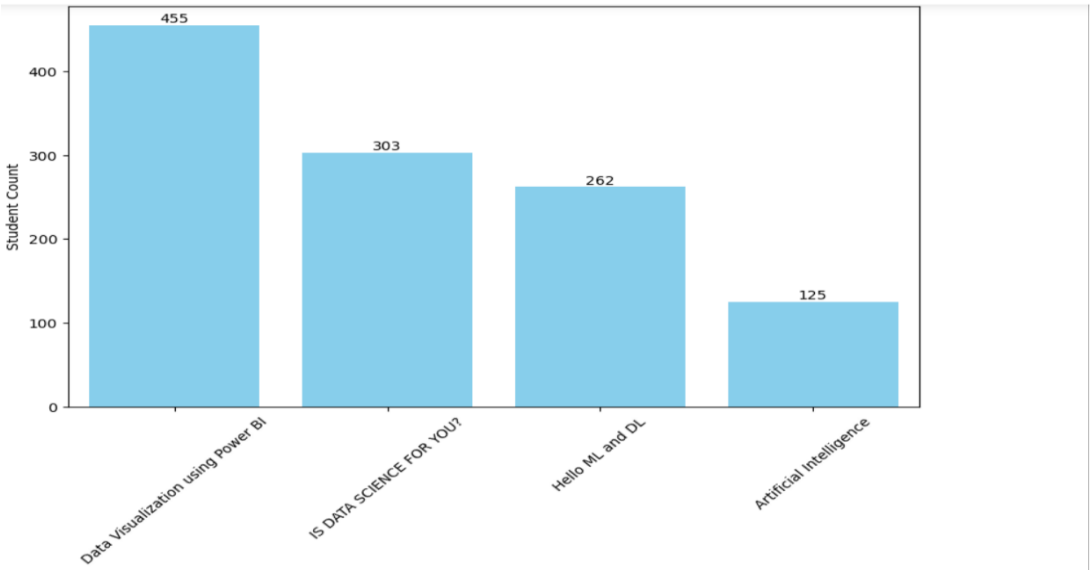
```
Data Visualization using Power BI    455
IS DATA SCIENCE FOR YOU?             303
Hello ML and DL                      262
Artificial Intelligence              125
Name: Events, dtype: int64
```

```
total_attendees=freq.sum()
total_attendees
```

```
1145
```

```
plt.figure(figsize=(10, 6))
bars = plt.bar(freq.index, freq.values,color='skyblue')
freq.plot(kind='bar', color='skyblue')
plt.xlabel('Events')
plt.ylabel('Student Count')
plt.title('No of students attending Data Science and Related Events')
plt.xticks(rotation=45)

for i, value in enumerate(freq):
    plt.text(freq.index[i], value, f'{value}', ha='center', va='bottom')

plt.show()
```

**Conclusion:**



7. Those who have high CGPA and more experience in language those who had high expectations for salary? (Avg)

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

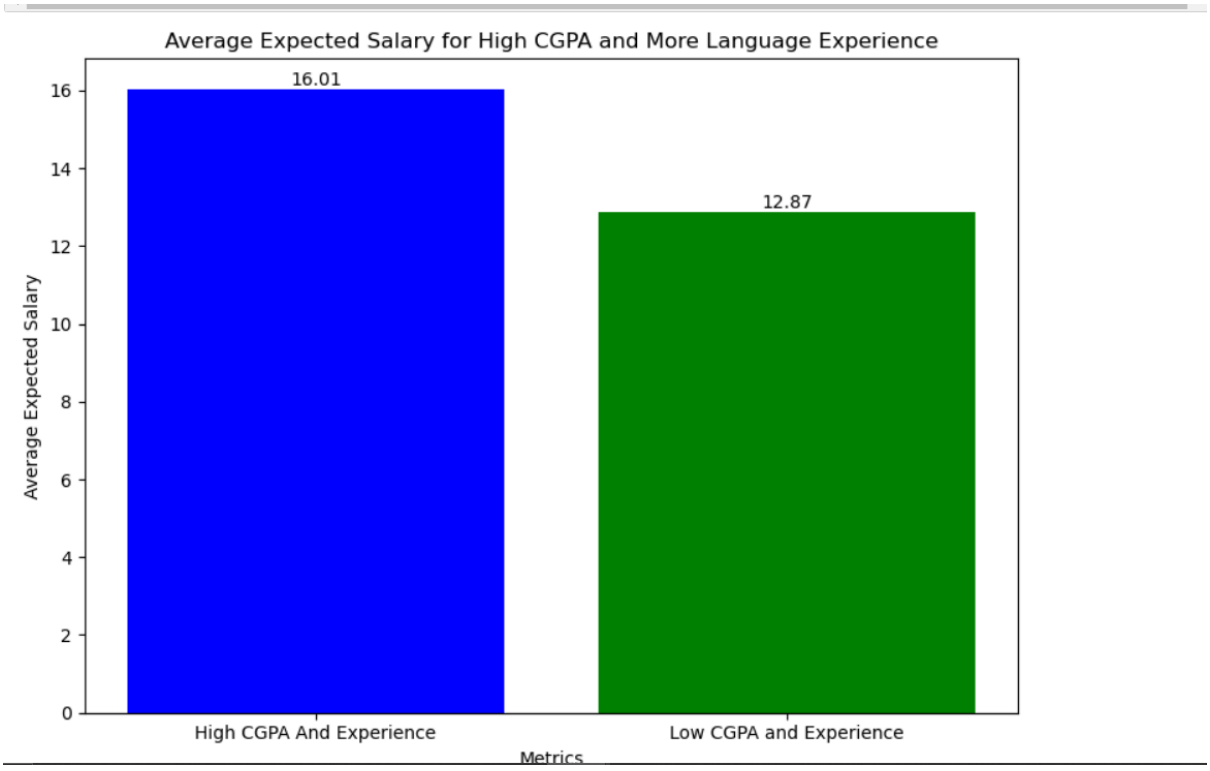
```
high_cgpa_threshold=7.5
high_language_experience_threshold=5
```

```
high_cgpa_exp=readxl[(readxl['CGPA']>high_cgpa_threshold)&(readxl['Experience with python (Months)']>high_language_experience_thr
low_cgpa_exp=readxl[(readxl['CGPA']<=high_cgpa_threshold)&(readxl['Experience with python (Months)']<=high_language_experience_th
```

```
average_salary_high_cgpa=high_cgpa_exp['Expected salary (Lac)'].mean()
average_salary_low_cgpa=low_cgpa_exp['Expected salary (Lac)'].mean()
```

```
plt.figure(figsize=(8,6))
plt.bar(['High CGPA And Experience','Low CGPA and Experience'],[average_salary_high_cgpa,average_salary_low_cgpa],color=['blue','
plt.xlabel('Metrics')
plt.ylabel('Average Expected Salary')
plt.title('Average Expected Salary for High CGPA and More Language Experience')
plt.xticks(rotation=0)

for i, value in enumerate([average_salary_high_cgpa,average_salary_low_cgpa]):
    plt.text(i, value, f'{value:.2f}', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

**Conclusion:**



8. How many students know about the event from their colleges? Which are these top 5 colleges?

**Code:**

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
readxl=pd.read_excel('Data analyst Data.xlsx')
```

```
college_event_counts = readxl[readxl['How did you come to know about this event?'] == 'SPOC/ College Professor']
```

```
college_event_counts = college_event_counts['College Name'].value_counts().reset_index()
college_event_counts.columns = ['College Name', 'Number of Students']
```

```
top_5_colleges = college_event_counts.nlargest(5, 'Number of Students')
```

```
plt.figure(figsize=(10, 6))
sns.set_palette("pastel")
ax = sns.barplot(y='Number of Students', x='College Name', data=top_5_colleges)

plt.title("Students Who Know About the Event from Top 5 Colleges")
plt.ylabel("Number of Students")
plt.xlabel("College Name")

plt.xticks(range(len(top_5_colleges)), top_5_colleges['College Name'], rotation=45, ha='right')

plt.yticks(range(0, top_5_colleges['Number of Students'].max() + 1, 10))

for index, row in top_5_colleges.iterrows():
    plt.annotate(str(row['Number of Students']), xy=(index, row['Number of Students']),ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.show()
```

**Conclusion:**