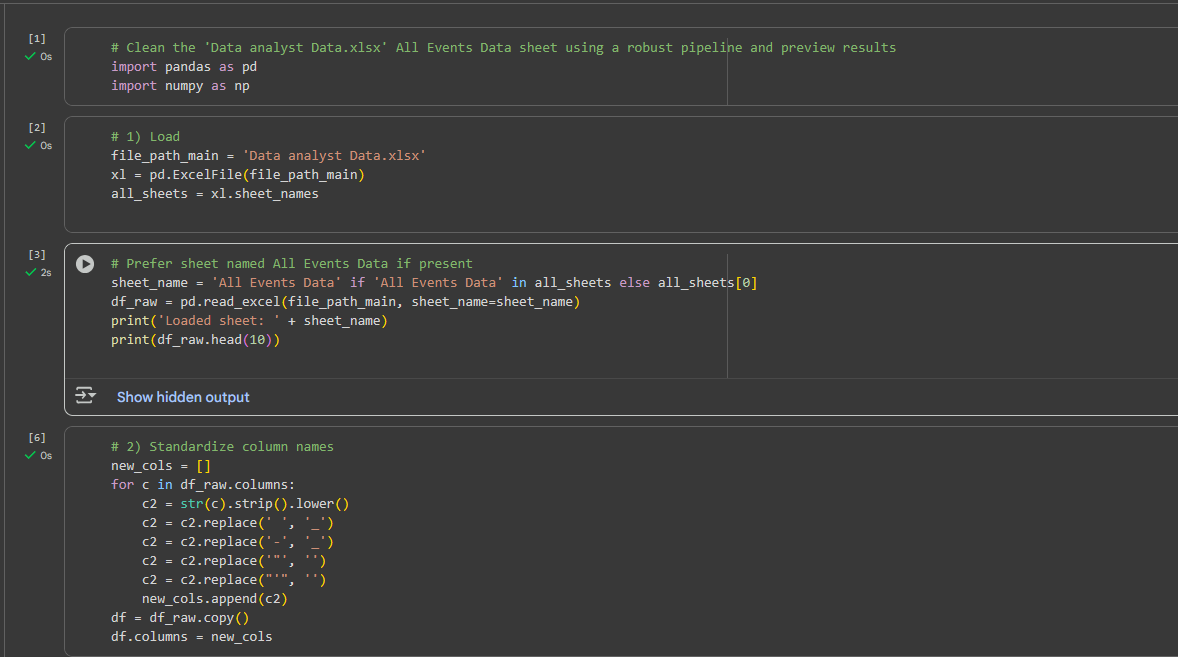
**Cleaning the Data**As I was analysing the data, I found some inconsistencies and discrepancies — the same students had registered for different events, but with varying details such as **Year of Graduation, CGPA, Experience with Python (months), City,** and even **College Name f**or e.g   
let’s look at the case of the student Patel, duplicated entries,multiple entries with varying details except the email  
  
A screenshot of a computer

AI-generated content may be incorrect.

To address these issues, I cleaned the data using Python with the Pandas library. I first loaded the "All Events Data" sheet from the Excel file and standardized the column names by converting them to lowercase and replacing spaces with underscores. I also mapped verbose column names to more concise alternatives, such as converting "Email ID" to "email" and "How did you come to know about this event?" to "how\_did\_you\_know".For string data cleaning, I trimmed whitespace from all string columns, converted email addresses and events to lowercase for consistency, and applied title case to city names. I also replaced 'nan', 'None', and empty strings with proper null values. To consolidate knowledge sources, I merged the "how\_did\_you\_know" and "others\_specify" columns into a single "knowledge\_source" field, using the specific details from "others\_specify" when "Others" was selected.For data type conversions, I converted numeric fields with appropriate bounds—clipping CGPA to the range [0, 10], setting minimum values for Python experience months and expected salary, and converting leadership skills to boolean values. I also  removed negative values from Python experience months.To handle deduplication and conflict resolution, I removed exact duplicate rows and grouped records by email + events to resolve conflicts for students with multiple entries. I used the median for continuous variables like CGPA and the mode for categorical variables, with a median fallback for discrete numeric fields when the mode was unavailable.For additional transformations, I normalized family income values to a standardized scale, processed salary data by converting expected\_salary\_lac to numeric format and handling text entries, and created an is\_student flag(in Power BI) using a multi-criteria approach that considers designation keywords, knowledge source indicators and college affiliation while excluding professional roles.

I also created several calculated columns to enhance analysis capabilities. The Email\_Norm column was created using LOWER(TRIM()) for consistent student identification across multiple event entries. I created a Channel\_First column to extract the primary promotion channel from the knowledge\_source field, handling cases where multiple channels were listed with separators like "|" or "/". For Python experience analysis, I added a Python\_Months\_Num column to ensure proper numeric conversion of experience data.The final cleaned dataset contains 3,902 rows with standardized, validated data ready for analysis, ensuring each student is properly identified and classified for accurate per-student metrics and comprehensive cross-dimensional analysis.



A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Note- Is\_student is applied to every question that wants to get data about students**Q1.How many unique students are included in the dataset?

Q2. What is the average GPA of the students?

Q5. What is the average family income of the student?

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.  
  
  
  
A computer screen shot of text

AI-generated content may be incorrect.

To classify students accurately, I created an is\_student flag using a multi-criteria approach with priority hierarchy: first checking for student-related designation keywords, then knowledge source indicators, followed by college name presence without professional terms. I also developed a designation classification system to categorize the 49 unique designations into meaningful field groups like Data Science, Computer Science, Mechanical, Electrical, and Civil engineering based on keyword matching. These solutions ensured accurate per-student analysis and proper classification for meaningful insights across all demographic and academic performance metrics.

**Conclusion:** I applied the is\_student flag on the whole page and used count distinct function on email , which gives us the number of unique students that is**, 2109**, if I remove the is\_student filter it is 2157. And for cgpa , I had to convert datatype from text to decimal number and then just use average function on it in data pane and average function for family income too which got **8.04** cgpa and **1.29 lakh** family income

**Conclusion:** I used data that is flagged as trusted and designated as Student in filter and simply converted the field of Cgpa to a measure and used aggregation function average on the field to find the average of Gpa which is ,**8.041**

Q3. What is the distribution of students across different graduation years?  
A white background with black text

AI-generated content may be incorrect.  A white background with black text

AI-generated content may be incorrect.

A graph with blue squares

AI-generated content may be incorrect.

For the graduation year distribution analysis, I created a shared "Per Student Grad" calculated table to ensure consistency across all graduation year-related measures. This table groups the data by normalized email addresses and calculates each student's maximum graduation year(In case of multiple entries), filtering for students only and excluding blank values. By creating this centralized table, I eliminated the need to repeat the per-student logic in multiple measures and ensured that all graduation year calculations use the same underlying data.The "Students by Grad Year" measure then references this shared table, counting students for the currently selected graduation year by filtering the Per Student Grad table. Similarly, the "Students by Selected Grad Year" measure uses the same table but includes logic to show the total count when no year is selected and the filtered count when a specific year is chosen. This approach not only improved performance by avoiding redundant calculations but also ensured that all graduation year metrics are perfectly aligned, preventing discrepancies that could arise from different per-student aggregation methods across multiple measures.The shared table approach also made it easier to maintain consistency when adding new graduation year-related measures, as they all reference the same underlying per-student data structure.

**Conclusion:** Out of 2109 students , I wanted to show the distribution across graduation years ,so I had to use a calculated table.I used Year of Graduation and Student by Grad Year to get the distribution and a interactive card alongside for which I used Students by Selected Grad Year. The majority of students are graduating in **2025 (714 students)**, followed by **2024 (609 students)**. The lowest counts are in **2026 (350 students)** and **2023 (436 students)**. Overall, most students are clustered around 2024–2025, showing a peak in 2025, while participation decreases before and after those years.

Q4. What is the distribution of student’s experience with Python programming?  
  
A graph of a graph with numbers and a bar

AI-generated content may be incorrect.

I created a separate **Months table** using GENERATESERIES(0,60,1) to represent the range of possible Python experience in months. Then, in the **Students by Exp Month** measure, I builded a summarized table (PerStudent) that filters only valid students (where is\_student = 1 and email is not blank) and calculates each student’s average Python experience. After that, I took the current month in context from the Months table and count how many students fall into that exact experience bucket. This helps me generate the distribution of students across different months of Python programming experience.

**Conclusion:** The distribution shows that most students have around **5 months of Python programming experience (426 students)**, making it the peak experience level. A moderate number of students reported **3 months (266 students), 4 months (262 students), and 7 months (262 students)**. Fewer students fall in the **6-month group (222 students)**, and the least are in the **8-month group (114 students)**.

Q6 How does the GPA vary among different colleges?  
  
A colorful pie chart with text

AI-generated content may be incorrect.

**Conclusion:** The **average CGPA across colleges** is quite consistent, with only slight variations. **St. Xavier’s College (8.23)** and **Thakur Institute of Management Studies (8.21)** have the highest averages, closely followed by **New Horizon Institute of Technology and Management (8.17)**, **B.K. Birla College of Arts, Science & Commerce (8.15)**, and **KLE Society’s College of BCA, Belagavi (8.13)**.

Overall, all colleges maintain an average CGPA around **8.1-8.2**, showing relatively balanced academic performance with no major outliers

**Q7. Are there any outliers in the quantity (number of courses completed) attribute?**

A white background with blue lines

AI-generated content may be incorrect.  
  
**Conclusion:** The distribution of the **quantity (number of courses completed)** shows that most students fall within the range of **1 to 5 courses and others in 6-7 courses**, with a small number reaching **8 courses**. Since all values are clustered closely together and there are no extreme deviations, **there are no significant outliers** in this attribute.Though before I cleaned the data , there was one email(patel@xyz.com) with 60 courses completed.

Conclusion: The data is consistent, with students completing a fairly similar number of courses (5–7), and no unusual outliers are present.

**Q8: What is the average GPA for students from each city?**

**A map of the world with blue circles

AI-generated content may be incorrect.**

A computer code with white text

AI-generated content may be incorrect.

I calculated the **average CGPA per student by city**. First, I filtered the dataset to include only valid students (where is\_student = 1, email was not blank, CGPA was numeric, and city was not blank). Then, I used SUMMARIZE to group the data by each student and their city, while calculating their individual average CGPA. Finally, I applied AVERAGEX over this summarized table to compute the overall **average CGPA across students within each city**. This ensured the calculation was done at the student level first and then aggregated at the city level.

**Conclusion:** The data suggests that cities like Hyderabad, Bangalore, and Chennai have significant student populations with varying CGPA ranges, while smaller cities also contribute to the overall distribution. The list on the left highlights specific cities with their average CGPA, showing a range from 3.25 to 8.51, with Kolkata leading at 8.51. This indicates a diverse academic performance across the region, with urban centers potentially offering better educational resources, though further analysis would be needed to confirm this trend.

**Q10:How many students from various cities?**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Conclusion:** The map and data indicate the number of students by city, with Ahmedabad having the highest count at 47, followed by Bhilwara and Bhilwara with 45 each. The distribution shows a concentration of students in India, particularly in northern and central regions, with counts ranging from 38 to 47. This suggests that urban and possibly educational hubs like Ahmedabad may have larger student populations, though further analysis would be needed to confirm the underlying factors.

**Q11. How does the expected salary vary based on factors like ‘GPA’, ‘Family income’, ‘Experience with Python (Months)’?**

**A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.**

**Conclusion –** Effect of GPA, Family Income, and Python Experience on Expected Salary

1. **GPA (CGPA):**
   * There is a positive correlation between GPA and expected salary.
   * Students with higher CGPA (above 9) expect significantly higher salaries (peaks at ~17.7 LPA).
   * Lower GPA students (below ~7) generally expect around 6–11 LPA.
2. **Experience with Python (Months):**
   * Expected salary increases steadily with more Python experience.
   * Students with 6–7 months of experience have the highest salary expectations (~14.8 LPA).
   * Those with lower experience (<4 months) expect around 12.5–13.5 LPA.
3. **Family Income:**
   * Salary expectations increase with family income up to around 6 Lakh annual income.
   * After 6 Lakh, salary expectations show a slight drop, suggesting that very high family income doesn’t always correspond to higher expectations.

**Q13.Do students in leadership positions during their college years tend to have higher GPAs or better expected salary?**

A graph of a graph of a person

AI-generated content may be incorrect.

**Conclusion – Impact of Leadership Skills on Expected Salary & CGPA**

1. **Expected Salary:**
   * Students with **leadership skills** have a **higher average expected salary** (≈13.72 LPA) compared to those without leadership skills (≈13.22 LPA).
   * This indicates that leadership experience **positively influences salary expectations**, possibly due to greater confidence or perceived employability.
2. **CGPA:**
   * Interestingly, students with leadership skills have a **slightly lower average CGPA** (≈8.03) compared to those without leadership skills (≈8.04).
   * The difference is very small, suggesting that leadership roles **don’t significantly impact academic performance**, but may slightly divert focus from grades.

**Q14. How many students are graduating by the end of 2024?**

**A screenshot of a graph

AI-generated content may be incorrect.**

Conclusion: The analysis calculates the total number of students whose graduation year is 2024 or earlier. Here, **'Per Student Grad'** is a calculated table that contains one row per student, ensuring duplicates are removed and each student is counted only once. Using this table, we filter the students based on their graduation year (≤ 2024) and then count the resulting rows.

This provides a reliable count of students who are about to complete their studies and are likely to be seeking internships or job opportunities soon. Identifying this segment helps in planning targeted placement drives, final-year training programs, and career counseling sessions to support their transition into the workforce

**Q15. Which promotion channel brings in more student participations for the event?**

**A colorful pie chart with numbers

AI-generated content may be incorrect.**

**A computer code with text

AI-generated content may be incorrect.**

The *Channel First* calculated column is designed to extract the primary promotion channel from each student’s record. It first uses the COALESCE function to pick the first non-blank value between the knowledge\_source and how\_did\_you\_know columns, ensuring that a value is captured even if one column is empty. Next, SUBSTITUTE replaces the | symbols with / and removes any extra spaces to clean up the text. The SEARCH function then locates the first occurrence of /, allowing the formula to return only the text before the slash — effectively giving us the first mentioned channel. If no / is found, the entire cleaned string is returned, and if the value is blank, it returns BLANK(). This ensures each student is tagged with a single, consistent promotion channel, making the participation analysis more accurate.

**Conclusion:**

From the chart, it's clear that **WhatsApp** emerged as the most impactful promotional channel, bringing in **341 student participants**. **Email** followed as the second most effective method with **208 participants**. Other channels like **SPOC/College Professors** (60) and **Others** (65) had a moderate reach, while platforms such as **Instagram**, **Telegram**, and **Twitter** showed minimal effectiveness, with fewer than 10 participants each.

These results reinforce the importance of **direct and instant communication platforms**, especially WhatsApp, when targeting student audiences. It also suggests that traditional outreach methods like institutional SPOCs still hold value, whereas popular social platforms may not be as effective for event promotions in this context.

**Q16.Find the total number of students who attended the events related to Data Science.**

***(From all Data Science related courses.)Top of Form***

***Bottom of Form***

**Q17. Do those who have high CGPA & more experience in language have higher expectations for salary? *(Use Average)***

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

To analyze participation specifically in Data Science–related events, a filter was applied on the event data to include only those records where the event category or name is related to Data Science. This ensures that only relevant events are counted, excluding non-technical or unrelated ones.Also didn’t consider RPA to be a Data Science Event  
  
I analyzed the relationship between CGPA (X-axis) and expected salary in LPA (Y-axis), using The *ExpectedSalary\_Q3* measure calculates the **average expected salary** for students who are in the top 25% (upper quartile) of both CGPA and Python programming experience. First, it computes the **third quartile (Q3)** for cgpa and python\_exp\_months using PERCENTILEX.INC, which helps identify the threshold values above which students fall into the top-performing group. Next, it builds a *PerStudent* table by summarizing data per student (based on their normalized email) and calculating their average CGPA, Python experience months, and expected salary. This ensures that multiple records for the same student are aggregated correctly. Then, it filters this summarized table to include only those students who have CGPA and Python experience above their respective Q3 thresholds and whose salary data is not blank. Finally, it calculates the **average expected salary** for this filtered group, returning BLANK() if no qualifying students exist. This measure is particularly useful for understanding salary expectations among the top-performing and most experienced students.

**Conclusion:** Total of **824 students** attended Data Science–related events across all courses, indicating a strong interest and engagement in this domain. This is a significant portion of the student population and highlights Data Science as one of the most attractive fields among learners.

Furthermore, students with **high CGPA and greater programming experience** demonstrated **higher salary expectations**, with the top 25% expecting an average salary of **16.32 LPA**. This insight aligns with industry trends, where top-performing students with strong technical skills anticipate competitive compensation. These findings can guide event organizers and recruiters to focus more on high-performing students and provide them with targeted career opportunities, internships, and industry connections.

**Q18. How many students know about the event from their colleges? Which are the Top 5 colleges?**

A graph on a pink paper

AI-generated content may be incorrect.

A computer code with text

AI-generated content may be incorrect.

A white background with black and white clouds

AI-generated content may be incorrect.

Students by College measure calculates the number of students from each college who learned about the event through college-related sources. First, it creates a summarized table (*PerStudent*) by grouping data per student (using their normalized email) and retrieving their college name and the primary knowledge source. This ensures that duplicate records per student are removed and only the most relevant information is retained.

Then, the measure filters this table (*FromCollege*) to keep only those students whose knowledge source contains keywords such as "college", "campus", "spoc", or "ambassador", indicating that the information was shared through college-related channels. Finally, it counts the number of students belonging to the currently selected college in the report context, using SELECTEDVALUE for the college\_name.

This approach allows for a clear, college-wise breakdown of student participation driven by campus-based promotion channels, which can be used to evaluate the effectiveness of each college’s outreach efforts.

**Students Who Know Their Events from College** measure calculates the total number of students who came to know about events specifically through college-related channels. It first creates a summarized table (*PerStudent*) at the student level by grouping based on their normalized email and capturing the maximum college\_name and knowledge\_source. This step ensures duplicates are removed and only one record per student is considered.

Next, it filters the summarized table to include only those students whose knowledge\_source contains keywords like **"college"**, **"campus"**, **"spoc"**, or **"ambassador"**, meaning the event information reached them through formal or informal college-based promotions. Finally, COUNTROWS is used to return the total number of such students.

In the report, this measure is displayed as a **card visualization** to show the overall count at a glance. Additionally, it interacts dynamically with the **"Students by College"** visual, allowing users to click on a particular college and instantly see how many students from that college learned about the event via college channels. This provides actionable insights into which colleges are most effective in disseminating event information and encouraging participation.

**Conclusion**  
A total of **242 students** reported that they learned about the events through college-related channels such as campus announcements, SPOCs, or student ambassadors. This shows that colleges play a significant role in spreading event awareness and encouraging participation.

The **Top 5 colleges** that contributed the most to student participation are **MIT Academy of Engineering (21 students)**, **KLE Society's College of BCA (20 students)**, **Vidyalankar Institute of Technology (20 students)**, **Government Polytechnic Gandhinagar (19 students)**, and **Pillai College of Engineering (19 students)**. Together, these colleges account for a major share of students who were informed through college-based channels, reflecting the effectiveness of their outreach efforts.

These insights can help event organizers strengthen partnerships with these top-performing colleges and encourage underrepresented colleges to improve their communication channels for future events.