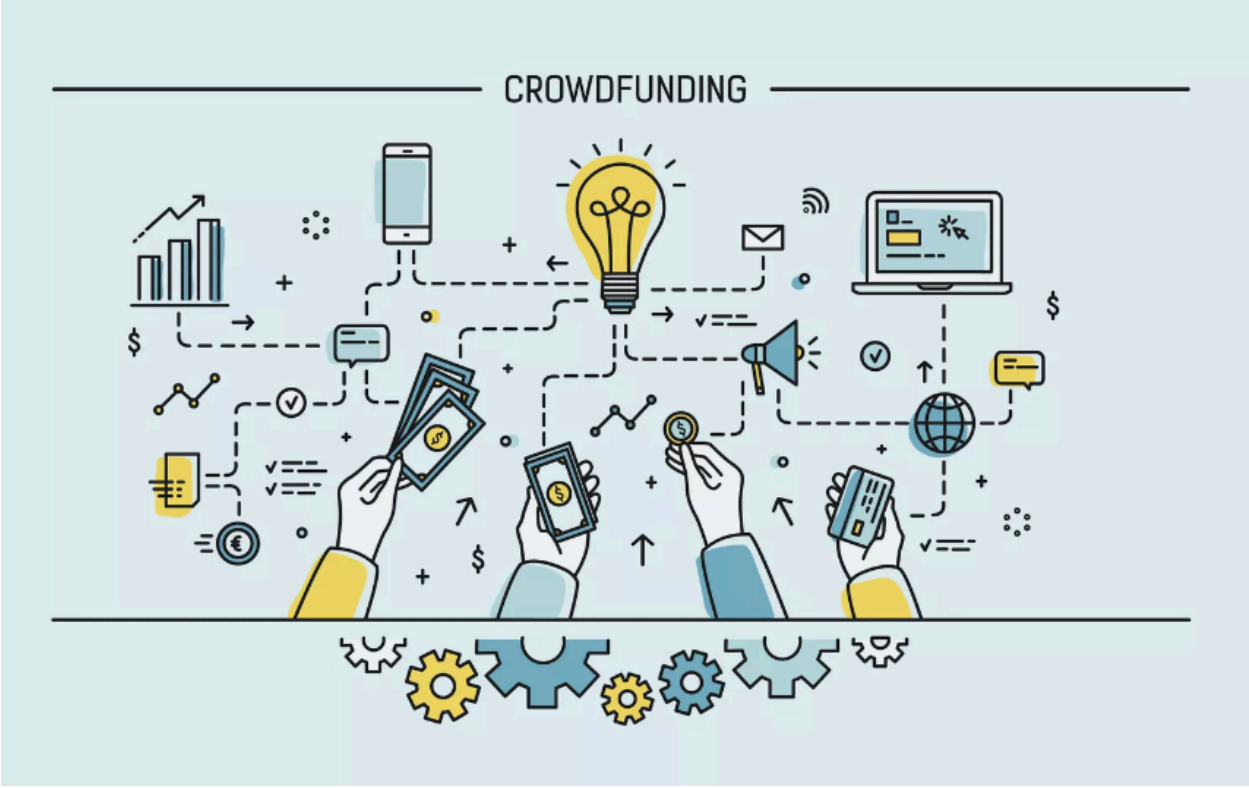
Sarah Ruth
Andrew Vick
Elizabeth Viramontes

Project 2 - Group 13
Crowdfunding Analysis

Bootcamp: DATA PT EAST APRIL 041524



07/24/2024

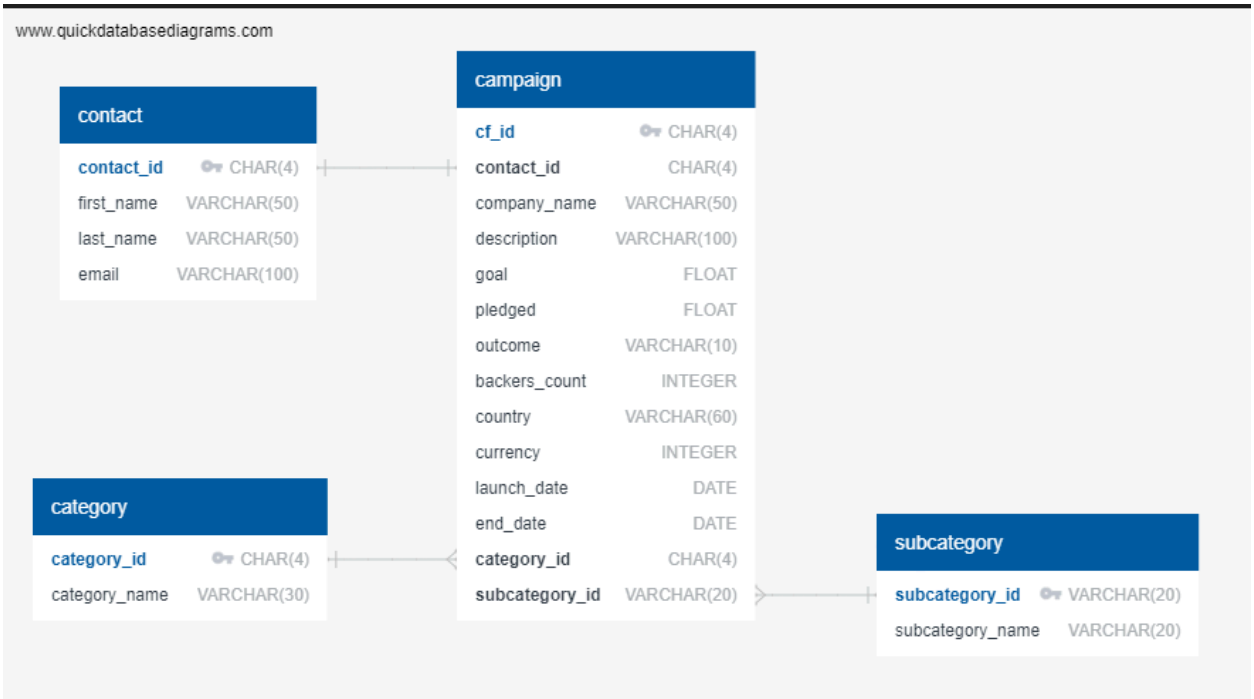
I. Introduction

For this project, our team: Andrew Vick, Elizabeth Viramontes, and Sarah Ruth collaborated to complete an Extract, Transform, and Load pipeline. We were given two Excel files, and a starter code notebook. The instructions were to design a relational database, extract the data from the Excel files and transform it to match our database schema. Then we loaded everything into the database to analyze the data by writing queries and generating visualizations.

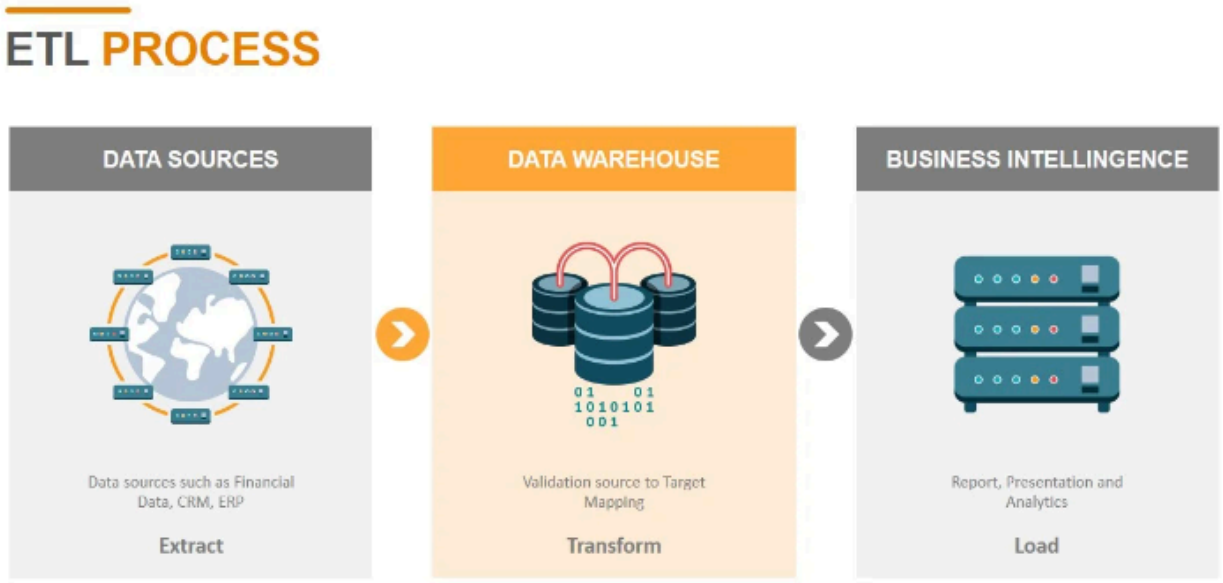
The basis of the project is formed by information obtained about crowdfunding, with different campaigns in some countries carried out on different dates.

II. Database Design Considerations

The Crowdfunding database was initially designed using the Quick DBD tool. After reviewing all of the materials provided, we were able to visualize and simplify the relationships between the data as well as facilitate an understanding of how to structure the database. From here, the ERD diagram and schema were generated illustrating how the tables were related, defining data types, primary keys, foreign keys, and other necessary constraints.



III. Extract/Transform/Load Overview



- **Extract**

To extract the data from the Excel files, we utilized our Jupyter Notebook and Pandas to read all of the data from crowdfunding.xlxs and the contacts.xlsx into 2 DataFrames. Then we examined the data and looked at the data types, columns, rows, and for any null values.

- **Transform**

In the next step of the process, we utilized our Jupyter notebook with pandas, numpy, datetime and Json libraries to transform the crowdfunding database and contacts database into 4 relational databases (category, subcategory, campaign, and contacts). Each new database was designed to have columns that corresponded with the tables that were generated in the ERD diagram.

The category, subcategory, and campaign DataFrames were created from data that we pulled from the crowdfunding dataframe. We had to split columns, create numpy arrays, use list comprehension, rename columns, drop columns, convert data types, and merge DataFrames before exporting our final results as CSV files.

The contacts dataframe was created by iterating through the dataframe and converting each row into a dictionary. Then we filtered out rows, created a formatted list to append, and then created a new dataframe to add each list of values. Columns were split and dropped before exporting the finished product as a CSV file.

- **Load**

Using the schema obtained from QuIckDBD, tables were created in PostgreSQL to accommodate data from our four CSV files, ensuring seamless integration and relational data management.

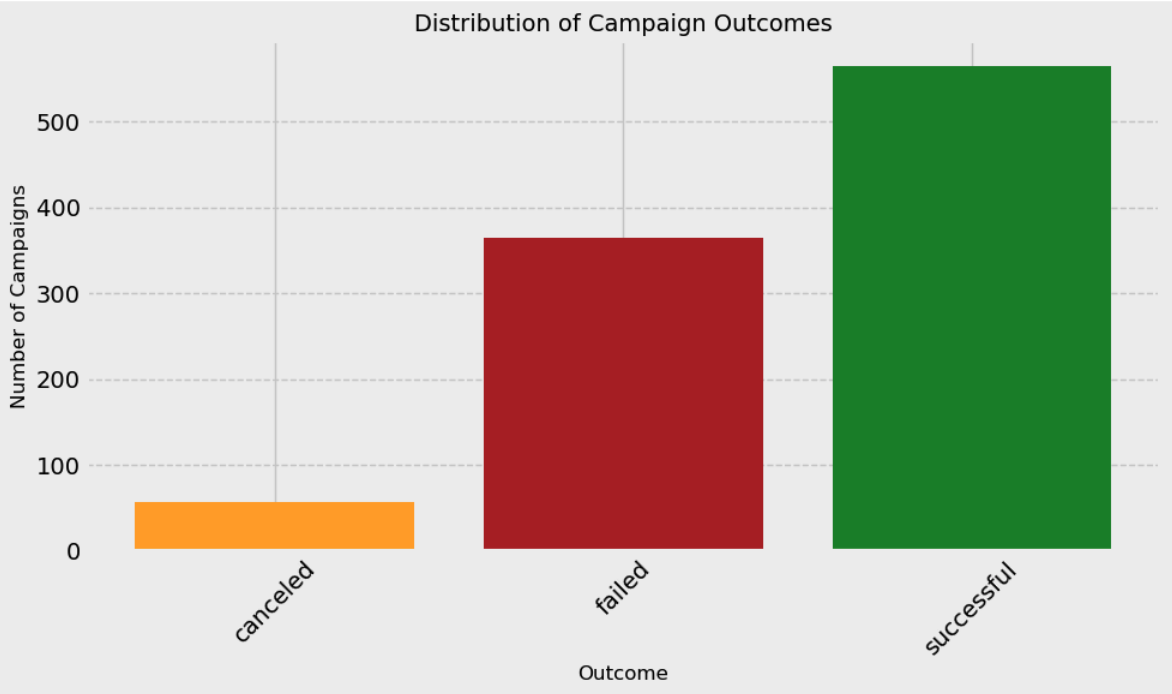
A new Jupyter notebook was developed to load the data from the four CSV files into the PostgreSQL database. Through this notebook, each CSV file was connected to each respective table within the crowdfunding_db database, establishing a direct link between raw data and structured database storage.

In the load notebook, we inserted code to connect to PostgresQL by creating the engine and inspector with SQLalchemy. From here, we created a dictionary with the tables and CSV files, defined the tables and used a “for loop” to load the CSV files with pandas.

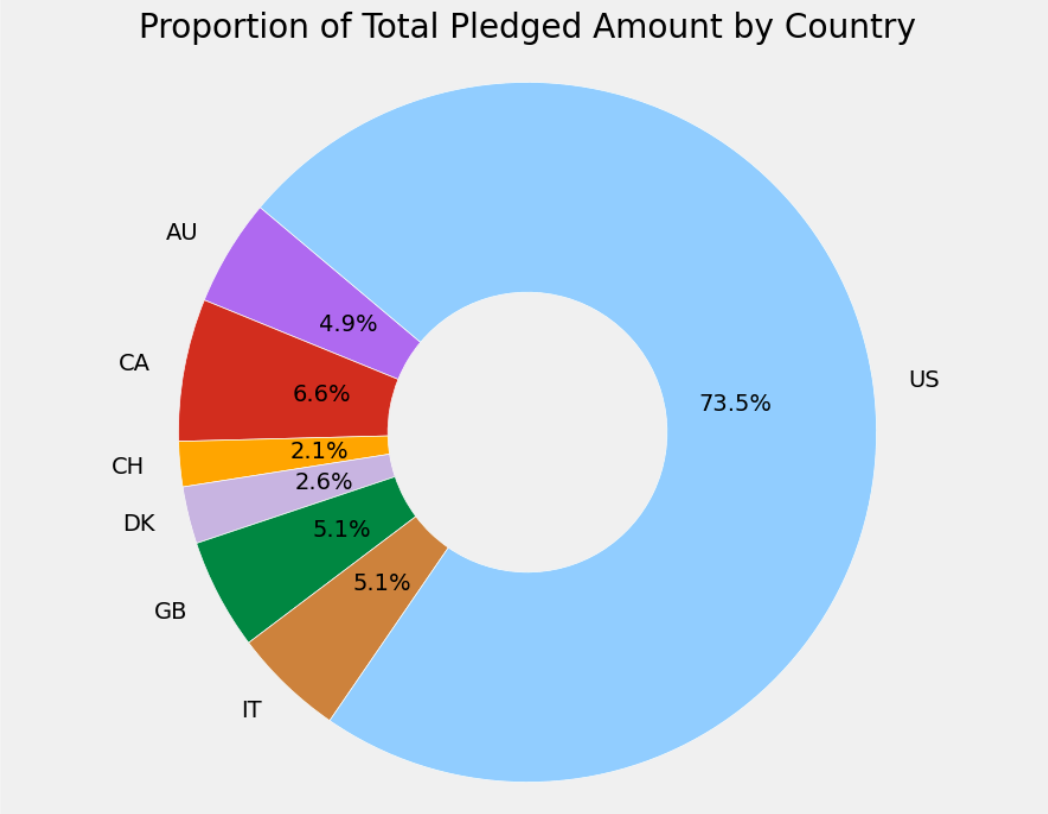
The tables were successfully loaded in the PostgreSQL database and a verification was done to ensure all tables were correctly loaded.

IV. Analysis

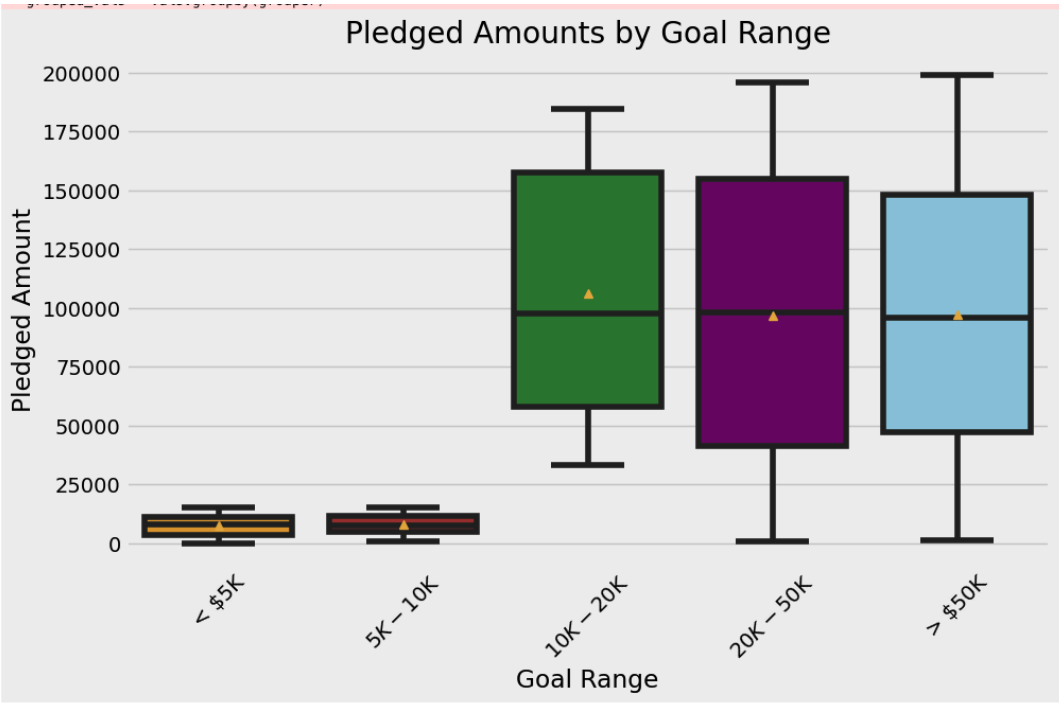
For the analysis in this project, we used the SQL created CSV file on the campaign data for our visualizations. We converted the CSV file into a sqlite file, so that we could use sqlalchemy in a Jupyter Notebook. We defined all the columns used, then we created a session to interact with the campaign database.



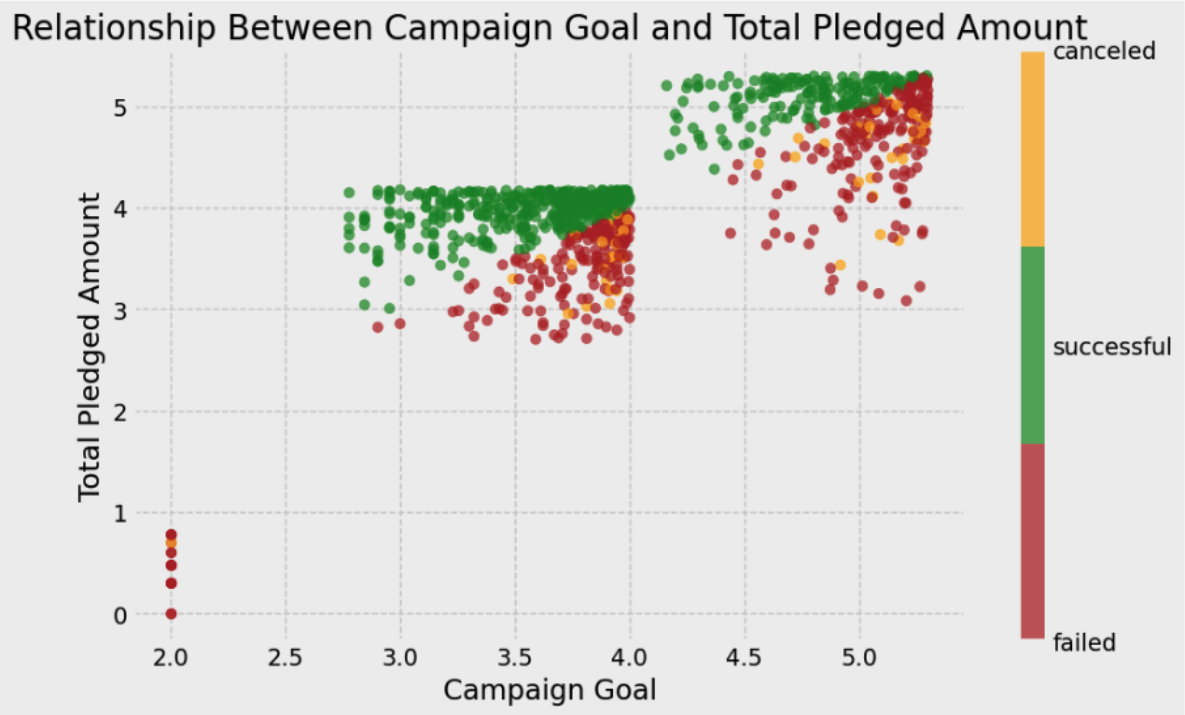
- Distribution of Campaign Outcomes: We queried the database for each outcome excluding any live campaigns. We thought the best way to illustrate the campaign outcomes was with a bar chart. According to the data, campaigns end up being more successful than not. Obviously, we do not know the outcomes of live campaigns, thus the exclusion.



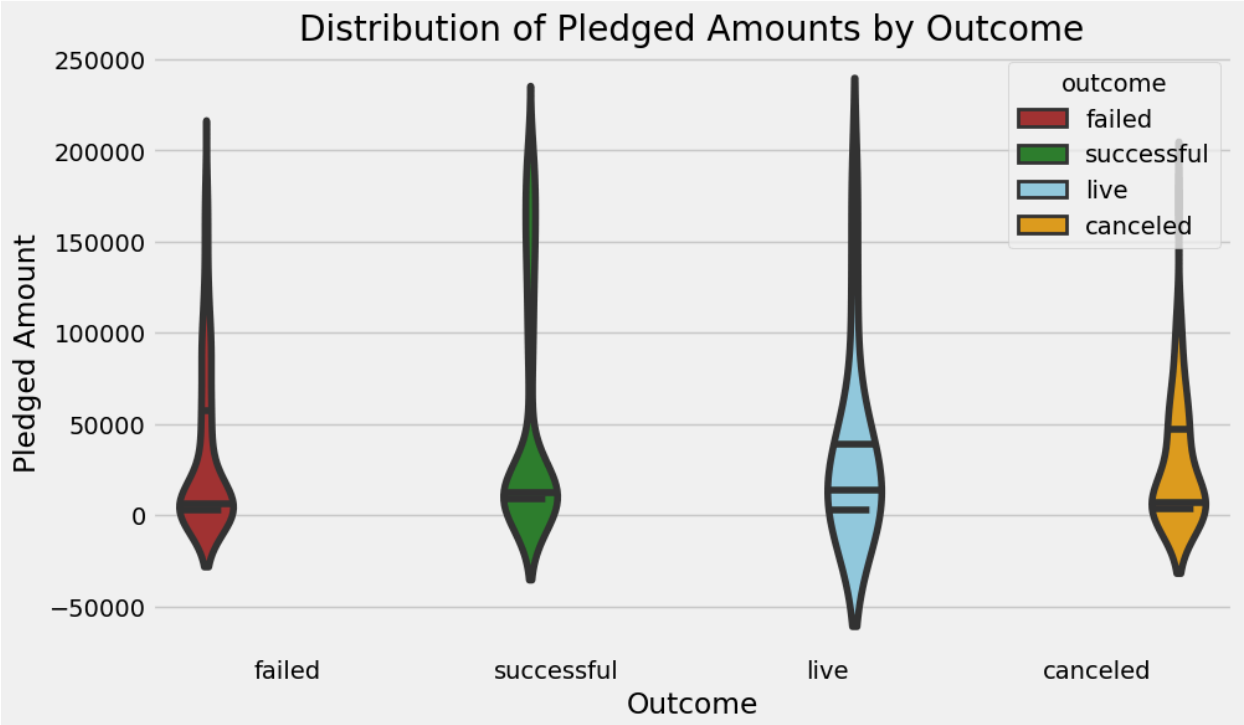
- Proportion of Total Pledged Amount by Country: To illustrate the pledged amount by each country, we thought a pie chart was the best way to represent the data. When analyzing the total amount pledged, the United States dominates the chart by an overwhelming majority with 73.5%. The country with the second largest contribution is Canada which comes in at measly 6.6%. This pie chart effectively showcases the United States as the primary contributor.



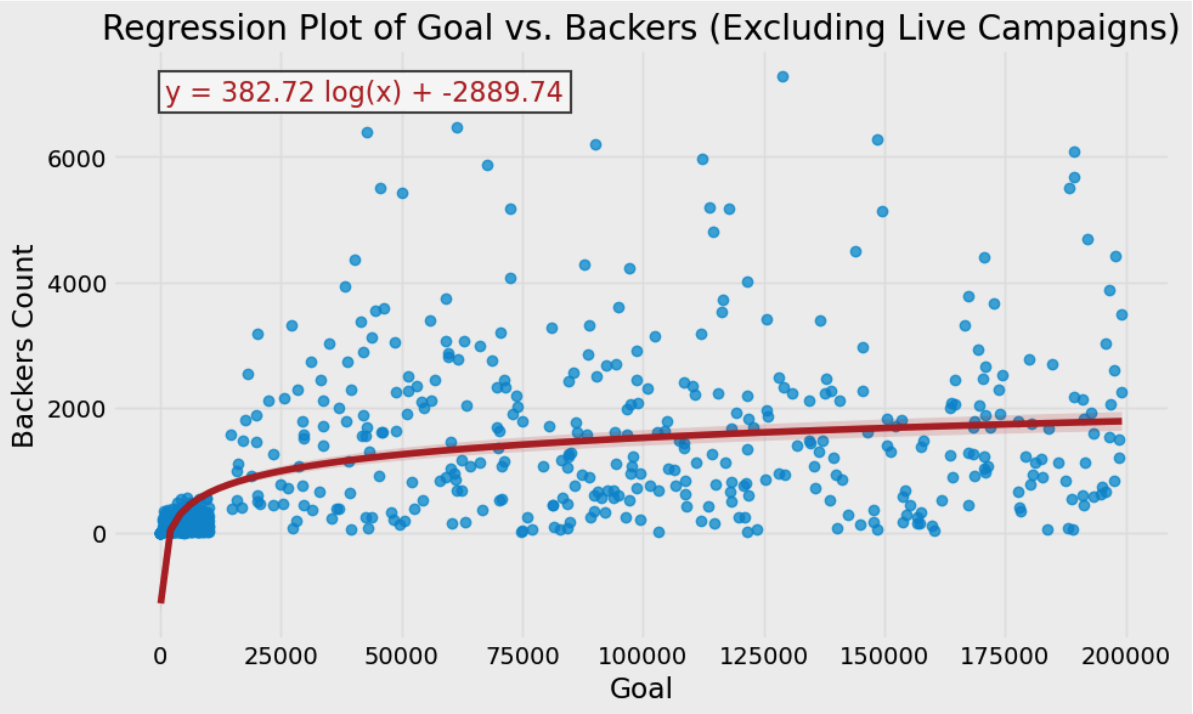
- Pledged Amounts by Goal Range: For our box plot on pledged amounts, we split up the goal ranges from less than $5,000 all the way to over $50,000. The box plot visual reveals that campaigns with higher goal ranges tend to attract more pledges. We can also see that the median increases as the goals rise. There is also greater variety in pledged amounts as the higher the goals become. According to the box plot, there is a positive correlation between goal ranges and pledged amounts.



- Relationship Between Campaign Goal and Total Pledged: The scatter plot we created illustrates that successful campaigns usually have higher pledged amounts compared to campaigns that failed or were canceled. Should be the common sense conclusion, but the scatter plot also shows the overlap of successful and failed campaigns. While campaigns with higher pledged amounts were successful more often than not, higher pledges are not a guarantee that a campaign will succeed.



- Distribution of Pledged Amounts by Outcome: Our violin plot shows that all campaigns have a wide range of pledged amounts. Successful, failed, live, and even canceled campaigns have pledges that range from $0 to over $200,000. Failed campaigns cluster around lower pledged amounts nearing $0, while canceled and live campaigns have fairly similar pledged amounts. Overall, the violin plot shows that a wide range of pledged amounts will not guarantee the success of a campaign.



- Regression Plot of Goal vs. Backers: For our final visualization, we created a regression plot for all the campaigns excluding the live ones. The regression line illustrates a positive correlation between a campaign's goal and the amount of backers. While there is a positive correlation, the amount of backers does begin to plateau. So, a higher goal will not always translate to a higher number of backers.

V. Bias and Limitations

We have identified the following Biases and Limitations of the original dataset that was extracted from the Excel files and also of the project as whole. They include the following:

- The data does not include what type of ages the crowdfunding is aimed at and that influences why some categories are more successful than others.
- The lack of information at what socioeconomic level it is aimed at can influence failure or success.
- If the type of food and the social level of the area where the crowdfunding took place is included, it would help to know why it failed or succeeded.
- The number of people who attended each event, there is only one donation average, but this could be that for example 10 people donated $1 and 20 people donated $100.
- The age at which the event is aimed, for example: adults, youth, family members, etc., would help to have an idea of why there was failure or success, there for monetary donations.
- The event schedule (time) could help you know why some campaigns failed or succeeded.
- The reason why a campaign was canceled would help to make better decisions in the following campaigns, for example if it was due to the weather, the place could not be reserved on time, cancellations by musicians, cooks, etc.

VI. Conclusion/Reflections

With the correct use of QuickDBD for schema design, PostgreSQL for database implementation and a dedicated Jupyter notebook for CSV integration, we were able to facilitate the data analysis for the Crowdfunding database.