Overview

This project involved building an ETL (Extract, Transform, Load) pipeline to process crowdfunding data using Python, Pandas, Postgres, and Excel. The main objectives were to extract data from Excel files, transform it into a suitable format, and load it into a PostgreSQL database. Due to the limited time available for this project work had to be divided into parts. Part 1 was to create the usable csv files. Part 2 was to create the working database, and part 3 was to query our data and create visualizations. This project came with no guideline on what we were attempting to find which meant that as a group we had to explore the data and find our own conclusions.

Data and Model

We began by extracting a crowdfunding CSV file, focusing on the 'category' and 'subcategory' headers. To extract and transform the data from the crowdfunding.xlsx Excel file, we first load the Excel file into a DataFrame. We then extract the unique categories from the data. Using these unique categories, we create a new DataFrame with two columns: a "category_id" column, which has entries that sequentially go from "cat1" to "catn" (where n is the number of unique categories), and a "category" column, which contains only the category titles. We then extracted additional information by generating a list of unique values for each header. For each DataFrame, we added headers using these lists of unique values.

For example:

- Categories: 'food', 'music', 'technology', 'theater', 'film & video', 'publishing', 'games', 'photography', 'journalism'.
- Subcategories: 'food trucks', 'rock', 'web', 'plays', 'documentary', 'electric music', 'drama', 'indie rock', 'wearables', 'nonfiction', 'animation', 'video games', 'shorts', 'fiction', 'photography books', 'radio & podcasts', 'metal', 'jazz', 'translations', 'television', 'mobile games', 'world music', 'science fiction', 'audio'.

Next, we created a 'Campaign' DataFrame with the following columns:

- The "cf_id" column.
- The "contact_id" column.
- The "company_name" column.
- The "blurb" column, which is renamed as "description."
- The "goal" column, which is converted to a float datatype.
- The "pledged" column, which is converted to a float datatype.
- The "backers_count" column.
- The "country" column.
- The "currency" column.
- The "launched_at" column, which is renamed as "launch_date" and converted to a datetime format.

- The "deadline" column, which is renamed as "end_date" and converted to a datetime format.
- The "category_id" column, with unique numbers matching the “category_id” from the category DataFrame.
- The "subcategory_id" column, with unique numbers matching the “subcategory_id” from the subcategory DataFrame.

We ensured that columns with numbers, specifically 'goal' and 'pledged,' were converted to float datatypes. We also made sure that columns with dates were converted to a datetime format. Finally, we exported the Campaign DataFrame to a CSV file.

Next, we extracted contact data by creating a new DataFrame. This DataFrame included the following columns:

- contact_id: A unique identifier for each contact person.
- first_name: The first name of the contact person.
- last_name: The last name of the contact person.
- email: The email address of the contact person.

Lastly, we used Python's Pandas library to create a 'Contacts' DataFrame. We imported the JSON module and initialized an empty list to store the extracted values. We then iterated through each row of contact_info_df, converting the JSON string in the first column of each row into a dictionary. This process allowed us to effectively structure the contact data for further processing and analysis. Below is a snapshot of our code, which demonstrates how we converted the JSON strings to dictionaries and then extracted the values from these dictionaries:

```
import json
# Iterate through the contact_info_df and convert each row to a dictionary.
dict_values = []
for i, row in contact_info_df.iterrows(): # itterate over rows
    data = row[0] # starting row
    converted = json.loads(data)
    rowvalues = [v for k, v in converted.items()] # v for value, k for key
    dict_values.append(rowvalues)

# Print out the list of values for each row.
print(dict_values) # prints LARGE block
```

This code helps us organize and extract relevant contact information from the DataFrame for further use.

Results and Challenges

The results for each query were achieved with their own set of challenges. Ranging from incorrect data types to date time not being accepted as valid syntax. When met with these challenges we had to find ways to overcome them as a team in order to meet our deadline.

Our first major challenge came from the date time function:

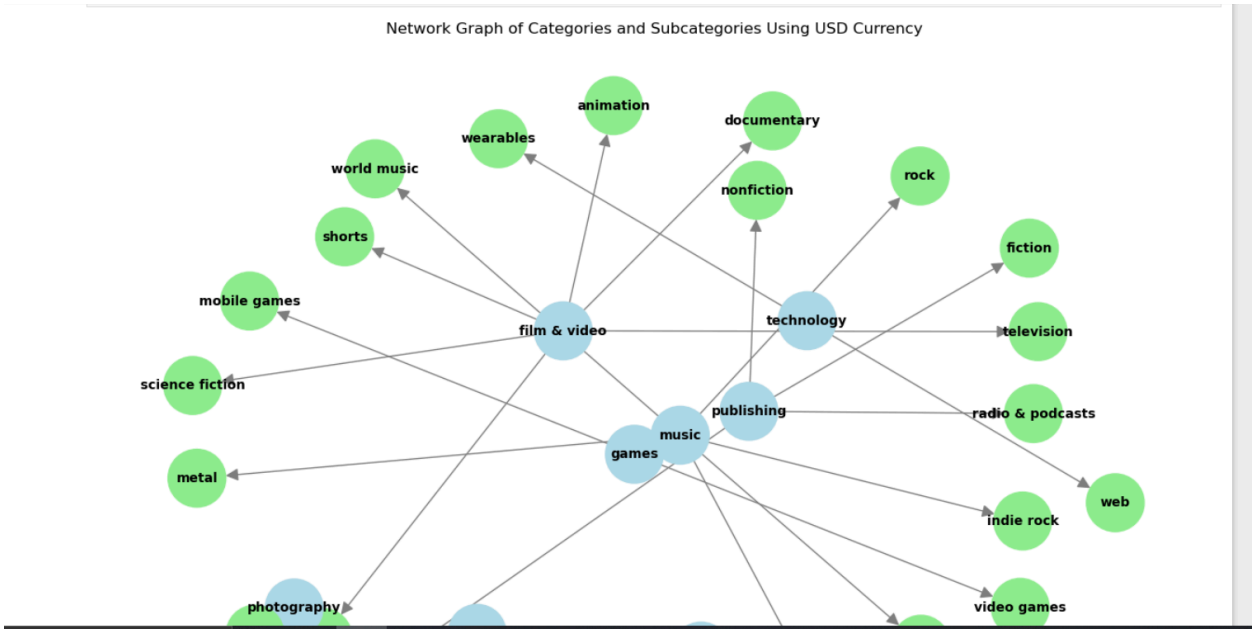
[20]:
```
# Another import needed
import datetime as dt
# How many backers, what did they pledge and what was the outcome
query = "SELECT campaign.end_date, backers_count FROM campaign where campaign.end_date between '2021-08-01' and '2022-08-31'"


# Execute the query and load the results into a DataFrame
campaign_data = pd.read_sql(query, connection)
campaign_data['end_date'] = pd.to_datetime(campaign_data['end_date']).dt.month

# Display the data in the DataFrame
campaign_data.head()
```

[20]:

	end_date	backers_count
0	12	1425
1	1	24
2	8	53
3	8	174

We went through several iterations of this code in order to achieve the desired outcome. Our goal for this code was to be able to look at the months for the campaigns. The primary issue was that date time was not working with our syntax. We soon found out that instead of using date time functions we would have to use .dt as pandas was having issues with our original method.

After some consideration we decided to take a deeper look into the campaigns that were using United States currency as that was most relevant to our point of view. This ended up with us having a data frame called usd_df to be able to look into these campaigns. While no visual was made for this data frame, there was still valuable insight to be had. Primarily, the campaigns from the US were very diverse in categories. From this, we decided to get a deeper look into the connection between category and subcategory which resulted in the following network graph.



This graph helped us understand the categories and subcategories in a different way by providing a visual that was not previously available.

Conclusion

At the start of this project, there were no specific questions we were setting out to answer and no conclusions that we were trying to draw. Instead our focus was to sharpen our ETL skills and overcome problems as we came to them. Even with that being said we were able to draw a few conclusions from our exploration of the data:

- The US has by far the largest distribution of backers for crowdfunding campaigns
- While many crowdfunding campaigns were successful, a campaign is almost as likely to fail as it is to succeed
- Plays and “the arts” make up a vast majority of crowdfunding campaigns
- Campaigns tend to end before the holiday season

There are likely many reasons for our conclusions that would require further looking into the data. Since that is not what this project was dedicated to, we do not have the answers as to why these conclusions are evident. In future work we will look deeper into the data to find answers to any further questions raised by the outcome of our exploration.