Mariam Jaoshvili, Casey Sestanovich, Ansley Lewis
Github link: https://github.com/mjaosh/206FinalProject

1. The goals for your project (10 points)

Our goal for this project is to collect Yo Mama jokes, as well as holidays and Harry Potter terminology. We hope to see how many of these Yo Mama jokes reference Harry Potter terminology and holidays data. We then hope to compare the proportion of Yo Mama jokes that reference Harry Potter terminology compared to the proportion of Yo Mama jokes that reference holidays.

2. The goals that were achieved (10 points)

We were able to capture some of the Yo Mama jokes that referenced Harry Potter terminology, however, not all of them due to the nature of the API. We were able to capture Harry Potter data that referenced a select choosing of characters, species, and book titles. Therefore, we were not able to capture all the Harry Potter specific terminology that was referenced in the Yo Mama jokes. Additionally, we were only able to capture US national holidays. This is because our API for the holidays only allowed us to capture holidays that were celebrated in the United States.

3. The problems that you faced (10 points)

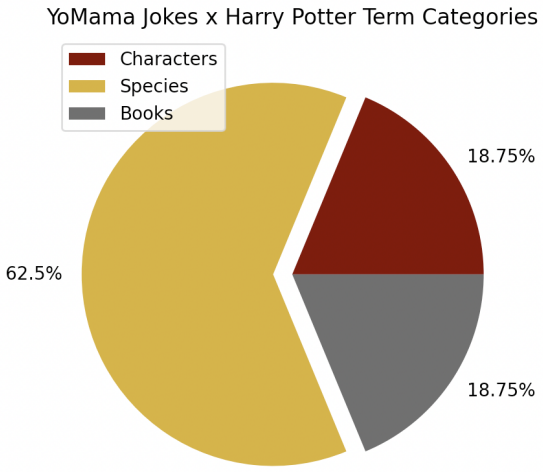
The first problem we faced was that the Yo Mama and Harry Potter APIs were Heroku based and would randomly shut down throughout the project. We did not know when these APIs would be up and running again, therefore, we had to choose new APIs for the Yo Mama and Harry Potter APIs. After we found a new Harry Potter API, this data was separated by characters, species, and book titles. Additionally, within these categories, only 20 pieces of data would be available at one time. Because of this, we had to call the queries for each individual category multiple times to get sufficient data. Finally, cleaning the holiday data was difficult as the data contained many filler words. There was not a uniform way to clean that data to find a representative word that might be in the Yo Mama joke. Therefore, we cleaned the data by removing common words like “day” so that we could find the most representative word of the holiday.

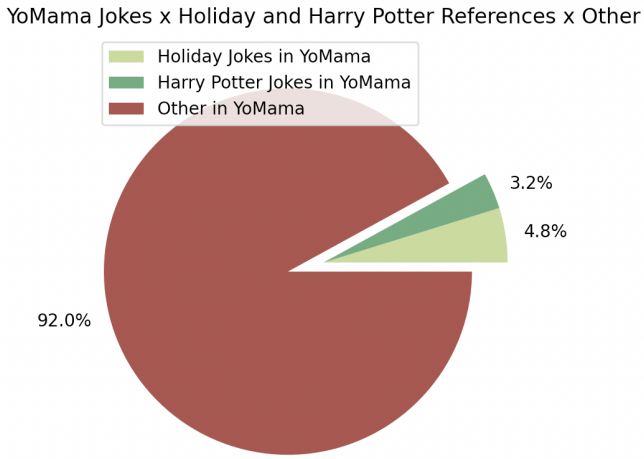
4. Your file that contains the calculations from the data in the database (10 points)

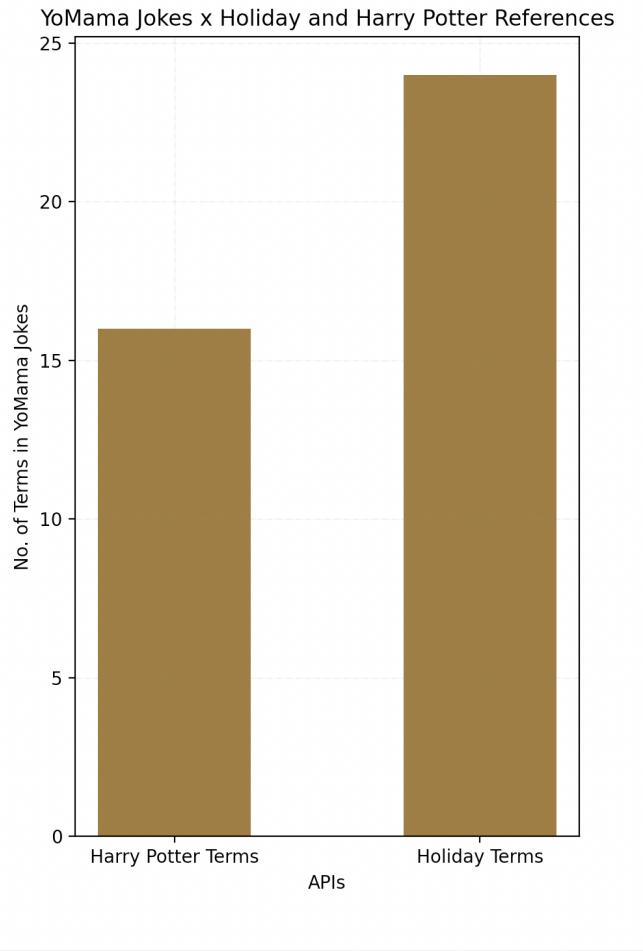
calculations.txt

```
total_harry_jokes: 16
percent_harry_jokes: 3.2
percent_character_jokes: 18.75
percent_species_jokes: 62.5
percent_book_jokes: 18.75
total_holiday_jokes: 24
percent_holiday_jokes: 4.8
```

5. The visualization that you created (i.e. screen shot or image file) (10 points)







6. Instructions for running your code (10 points)

The first step is to delete data.db and the calculations.txt. You run main.py 20 times to get all appropriate data into the database. Finally, when you run main.py the 21st time, it will complete the calculations and visualizations.

7. Documentation for each function that you wrote. This includes the input and output for each function (20 points)

In finalproject.py:

def setUpDatabase(db_name)

Creates the database where all the various tables and data will be stored.
Input: Takes in the desired name of the database
Output: Return the cursor and connection of the database

def create_holiday_table(cur, conn)
This function creates a holiday table within the database.
Input: The cursor and the connection of the database
Output: None

def get_holiday_data(cur, conn, url)
This function retrieves the holiday name and holiday date from the holiday API query and adds the data to the database 24 points at a time. All the data is made all lowercase before putting it in the database.
Input: The cursor and the connection of the database and the URL of the holiday database.
Output: Returns nothing. Prints "Holiday API done" when 144 items have been inserted into the database.

def create_mama_table(cur, conn)
This function creates a table for the Yo Mama jokes within the database.
Input: The cursor and the connection of the database
Output: None

def get_mama_data(cur, conn, url)
This function retrieves the Yo Mama jokes from the Yo Mama API query and adds the data to the database 25 points at a time. All the data is made all lowercase before putting it in the database.
Input: The cursor and the connection of the database and the URL of the holiday database.
Output: Returns nothing.

def create_harry_table(cur, conn)
This function creates a Harry Potter table within the database.
Input: The cursor and the connection of the database

Output: None

def get_harry_data(cur, conn)

This function retrieves the Harry Potter character names, species, and book titles from the Harry Potter API query and adds the data to the database 24 points at a time for the first two times it runs. After that, when the function runs, the data will be added to the database 20 points at a time. All the data is made all lowercase before putting it in the database.

Input: The cursor and the connection of the database and the URL of the Harry Potter database.

Output: Returns nothing.

def create_harry_id_table(cur, conn):

We create a table (HarryPotterTermsCategories) in the database and insert a numbering system for the category of Harry Potter. In this table, we identify that any datapoint in the HarryPotter table that has a 1 next to it is a character, any datapoint that has a 2 next to it is a species, and any datapoint that has a 3 next to it is a book title.

Input:The cursor and the connection of the database

Output: Returns nothing.

In calculations.py:

def gather_mama(cur, conn):

Collects the Yo Mama data from the databases.

Input: The cursor and the connection of the database.

Output: Returns a list of tuples of data from the database.

def gather_harry(cur, conn):

Collects from the Harry Potter data from the databases.

Input: The cursor and the connection of the database.

Output: Returns a list of tuples of data from the database.

def gather_holidays(cur, conn):

Collects from the holidays data from the databases.

Input: The cursor and the connection of the database.
Output: Returns a list of tuples of data from the database.

def get_harry_jokes(cur, conn, mama, harry):
First, it creates the table HPxYoMama. Then, it goes through all the Yo Mama jokes and Harry Potter terms to check if the Harry Potter term is in the Yo Mama joke. We took out generic species terms, like dog and cat, so that the Harry Potter Term was more accurate. If there is a match, it inserts the joke id and the Harry Potter term id into the HPxYoMama table.
Input: The cursor of the database, the connection of the database, list of the Yo Mama jokes, and the list of the Harry Potter terms from the database.
Output: Returns a list of the Harry Potter term ids that were found in the Yo Mama jokes.

def join_harry_id(termId_list, cur, conn):
This function's goal is to use JOIN to get the term_type of every single Harry Potter joke in the Yomama jokes based on the term_id. It inserts the found term_types into the HPxYomama table, as we will use this data for our calculations.
Input: The list of the Harry Potter term ids that were found in the Yo Mama jokes, the cursor of the database, and the connection of the database.
Output: Returns nothing.

def calculate_harry_percentages(cur, conn):
This function calculates the percentage of Yo Mama jokes that reference Harry Potter terminology. Then, it calculates the percentage of the Yo Mama/Harry Potter jokes that reference characters, species, and book titles. We then put the data into the calculations_dict dictionary.
Input: The cursor and the connection of the database.
Output: Returns nothing.

def clean_holiday_data(holidays):
This function cleans the holiday data from the dataset to find the most relevant word to see if that word would be in the Yo Mama joke. We determined the most relevant word by choosing the

longest word in each Holiday entry. We also removed some of the Holiday words that were irrelevant.
Input: The list of the holidays from the database.
Output: Returns nothing.

def get_holiday_jokes(cur, conn, mama, holiday):
First, it creates the table HolidaysxYoMama table. Then, it goes through all the Yo Mama jokes and holiday names to check if the holidays are in the Yo Mama joke. If there is a match, it inserts the joke id and the holiday id into the HPxYoMama table.
Input: The cursor of the database, the connection of the database, list of the Yo Mama jokes, and the list of the holidays terms from the database.
Output: Returns nothing.

def calculate_holiday_percentages(cur, conn):
This function calculates the percentage of Yo Mama jokes that reference holidays and puts it into the calculations_dict dictionary.
Input: The cursor and the connection of the database.
Output: Returns calculations_dict.

def write_calculations():
This function opens a .txt file and writes all the data from calculations_dict.
Input: Nothing.
Output: Returns nothing.

In visualizations.py:
def hp_type_pie(calcs)
This function creates a pie chart documenting the percentage of Harry Potter related Yomama jokes that refer to characters, species, and books. We also created a legend and made the pie chart color scheme as the Gryffindor colors.
Input: Takes in a dictionary full of the calculations.
Output: Returns nothing. Prints out the pie chart.

def in_yomama_pie(calcs)

This function creates a pie chart documenting the percentages of Holiday, Harry Potter, and "other" categorized Yomama Jokes. We also created a legend and made the pie chart color scheme as Christmas colors.

Input: Takes in a dictionary full of the calculations.

Output: Returns nothing. Prints out the pie chart.

def hp_vs_holiday_bar(calcs)

This function creates a bar graph of the total number of Harry Potter related jokes and the total number of holiday related jokes.

Input: Takes in a dictionary full of the calculations.

Output: Returns nothing. Prints out the bar graph.

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	**Issue Description**	**Location of Resource**	**Result**
12/6/2022	When I would try to insert into rows that already had data into them, it would add the data to the bottom of the database rather than in the desired row.	https://www.sqlitetutorial.net/sqlite-update/	It worked - allowed us to insert data into the correct rows.
11/27/2022	Did not know what data types were used in sqlite.	https://www.w3schools.com/sql/sql_datatypes.asp	Allowed me to use the correct data types for all the columns in our database.
12/8/2022	Forgot how to write txt files in python.	https://www.w3schools.com/python/python_file_write.asp	Used this code as an outline to write our own txt file for the calculations data.

12/8/2022	Did not know the format of how to use the lambda function in the python sorted function.	https://www.w3schools.com/python/python_lambda.asp	Allowed me to use the lambda function correctly to sort a list of tuples.
12/8/2022	Need help figuring out how to make a pie chart in python.	https://www.geeksforgeeks.org/plot-a-pie-chart-in-python-using-matplotlib/	We used this website as an outline for our pie chart code and was able to make our individualized pie charts
12/9/2022	Need help figuring out how to make a bar plot python.	https://www.geeksforgeeks.org/bar-plot-in-matplotlib/	We used this website as an outline for our bar graph code and were able to make our individualized bar graph.
12/3/2022	Did not remember the syntax of using join.	https://www.sqlitetutorial.net/sqlite-join/	Was able to use join to successfully combine information from two different tables.