The Praguians - Caroline Scanlan & Cori Kollin

1. **The goals for the project:**

The goals for our project were to collect the streaming statistics on spotify and compare them to the number of followers instagram. More specifically, we wanted to calculate the correlation between streams on Spotify with followers on Instagram, as well as discover the most popular genres for people with higher/lower followers on instagram.

2. **The goals that were achieved:**

While we had some issues along the way, we were able to achieve both of our goals and create a chart showing the correlation between the number of followers a user has on instagram compared to the number of streams (monthly) that they have on spotify. We were also about to create a bar graph comparing the most popular genres for the average number of followers.

3. **The problems that you faced:**

Our first problem was that we knew that we wanted to make a bar graph, but did not know how to do that using Seaborn. One of our largest issues was that we continued to get duplicate data in the form of uppercase vs. lowercase letters, or spacing differences, and trying to ensure it was only unique information. You can see the sources that we used in order to fix these issues in the chart below. We frequently checked piazza to learn from our classmates and that definitely helped us avoid other problems we could have had.

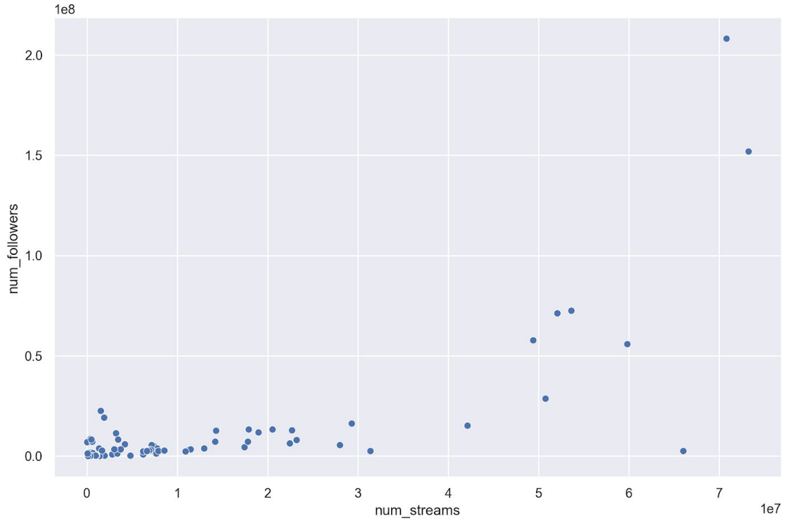
4. **Your file that contains the calculations from the database:**

https://github.com/corikollin/SI206Final

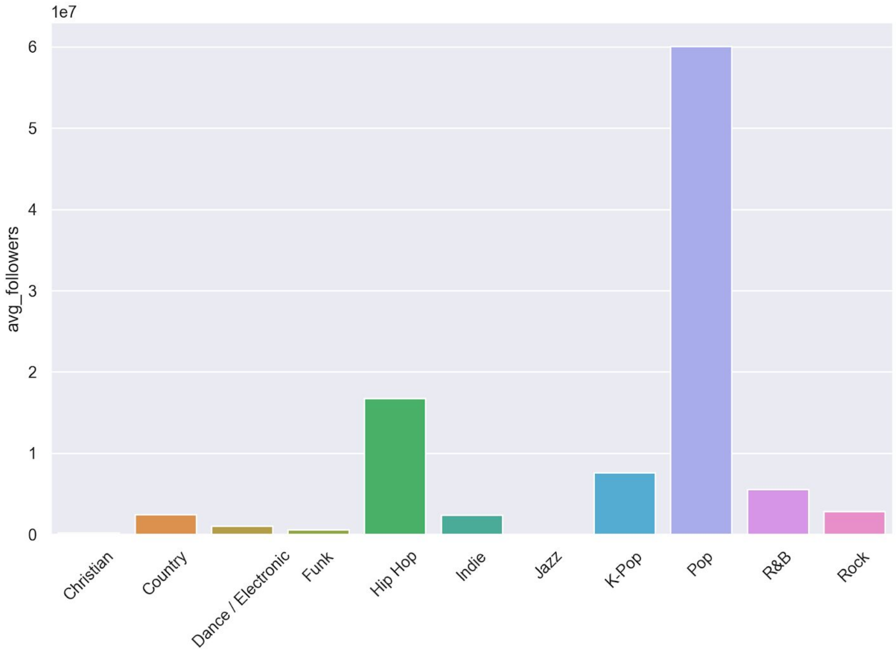
In this repository, genre_average_follwers.txt has our calculations.

5. **The visualization that you created:**

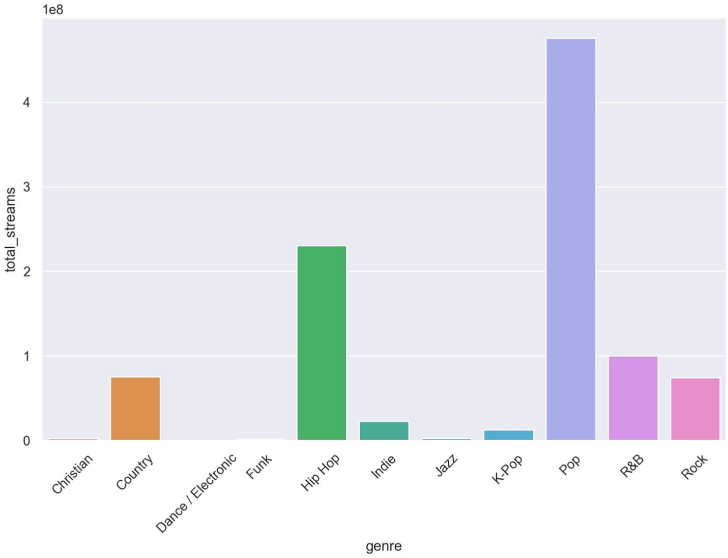
Instagram Followers x Spotify Streams



Most popular genres based on # Instagram followers of average listeners



Total streams per genre



6. **Instructions for running your code:**

   To run our code, type “python3 Final.py” into the command line. The program will begin and prompt the user for a number corresponding to the genre they would like to retrieve data for. Each time a genre is selected, ten artists’ information is pulled so it must be run multiple times to get 100 pieces of data in the database. Once the program has been run for each genre you wish to obtain data for, you may uncomment create_visuals.create_visuals() to create the two visuals shown above with the data pulled from the database and you may uncomment print_genre_averages_to_file(“genre_average_followers.txt”) to print the average number of Instagram followers per genre to the specified text file. Both of these lines are at the bottom of finalproject.py and have comments describing the information above.

7. **Documentation for each function that you wrote. This includes the input and output for each function:**

Finalproject.py (file)

- If_name = ‘_main_’: The main function starts by ensuring the the txt file is blank. Then it includes the list of genres, and prompts the user to select on of the genres based on the number inputted. It takes this input and figures out which genre was chosen. The extension is used to get data from Spotify. The get_artists_in_genre function then connects the artists from the genre. The following lines set up the tables in the database.
    - setGenreTable puts the artists with their genre
    - setIdTable puts the artist with the ID number associated with them

  - setStreamsTable sets up each artist with their number of monthly streams
  - setInstagramTable sets up the artists with their number of instagram followers
  - setGenreAverages sets up each genre with the average number of followers for the genre.

  The function create_visuals.create_visuals can be uncommented when we want to call the functions from create_visuals.py and create our graphs

  The function print_genre_averages_to_file can be uncommented to print the genre averages into our txt file.

- def setIdTable(data): Connects to database and creates a table of artists based on their Id numbers. It ensures that each entry is unique.
- def setGenreTable(genre,data): Connects to database and creates a table of artists with their genre. This function finds the 10 artists we use for each genre.
- Def setInstagramTable(data): Connects to the database and creates a table of each artist and their number of instagram followers.
- Def setStreamsTable(data): Connects to the database and creates a table of each artist and their number of monthly streams.
- Def setGenreAverages(): Connects to the database and creates a table for each genre with the average number of followers.
- Def print_genre_averages_to_file(filename): This function takes the average number of followers per genre and prints them into our txt file.

create_visuals.py(file)

- def create_genre_graph() : Creates the bar graph with the average number of followers for each genre. It takes the information from the database and then selects the specific information that it needs from the table. The data we pulled from is genre_followers. X axis is genre, and Y axis is followers. We used seaborn and created a bar graph.
- def create_top_artists_graph(): Connects to the database and grabs instagram number of followers and number of streams from the instagram table joined with the streams table. We then connected them based on the name, making an ordered pair of streams x followers. We used Seaborn and created a scatterplot graph.
- def create_visuals: function that can be called from the main file in order to run the two functions above.

getdata.py(file)

- def get_artists_in_genre(genre):
  - Opens Spotify and goes to the page of the specified genre. It creates a Beautifulsoup object and we parse it to find the first thing that is a playlist. We go into that to find the string that has information about the artist. We find the artist ID and and name. Create a dictionary with artist and their ID. Return a list of 10 of the top artists in the specified genre.

- def get_insta_username(artist):
    - Not everyone’s Instagram names are formatted the same way so this function searches google to find an artist’s username on Instagram and then returns this information in a string.
- def get_num_followers(user):
    - Goes into Instagram and uses a JSON formatted file to figure out the number of followers there are. Returns the number of Instagram followers based on the username.
- def get_monthly listeners(artist, cur):
    - Gets artist name and ID from the artist table and returns the number of Monthly Spotify Streams of a given artist.

8. You must clearly document all resources you used:

Date	**Issue Description**	**Location of Resource**	**Result (did it solve the issue?)**
12/2	Didn’t know how to create bar graph using seaborn	https://seaborn.pydata.org/tutorial/categorical.html	Yes
12/4	Wanted to get Instagram information in JSON format	https://stackoverflow.com/questions/34906301/how-to-get-other-pages-followers-count-number-in-instagram	Yes
12/4	Needed to load data from sqlite database	https://stackoverflow.com/questions/40010317/python-plotting-pandas-sql-dataframe-with-seaborn	Yes
12/6	Trouble preventing duplicate data entries in database	https://www.w3schools.com/sql/sql_ref_unique.asp	Yes
12/6	Genre labels on x-axis of bar graph were overlapping	https://stackoverflow.com/questions/42528921/how-to-prevent-overlapping-x-axis-labels-in-sns-countplot	Yes