Flannery O'Donnell & Katherine Berry
Professor Barbara Ericson
SI 206 Final Project
Group Name: Flashing Rainbow

The goals for our project:

When we first decided what we wanted to do for our final project, our plan was to compare the top artists on Spotify and Apple Music. We then ran into a slight error when we realized that we had to pay to use the Apple Music API and would not be able to use data from that platform. We then tried to access the Soundcloud API as our backup plan, but got the error message "Due to the high amount of requests recently, we will no longer be processing API application requests at this time. We are working to re-evaluate our process to make it more efficient." After trying to access many other music platforms' APIs we successfully ended up deciding to use Billboard.com's API.

The goals that were achieved:

Our new goal was to use Billboard.com's top tracks and Spotify's top tracks and put these songs and their corresponding artists into a database. This goal was achieved, and we then took it a step further. We calculated how many times each artist appeared on the top tracks of both Billboard.com and Spotify, so that we could create visualizations to compare which artists are more successful on each platform.

The files that contain the calculations from the data in the database:

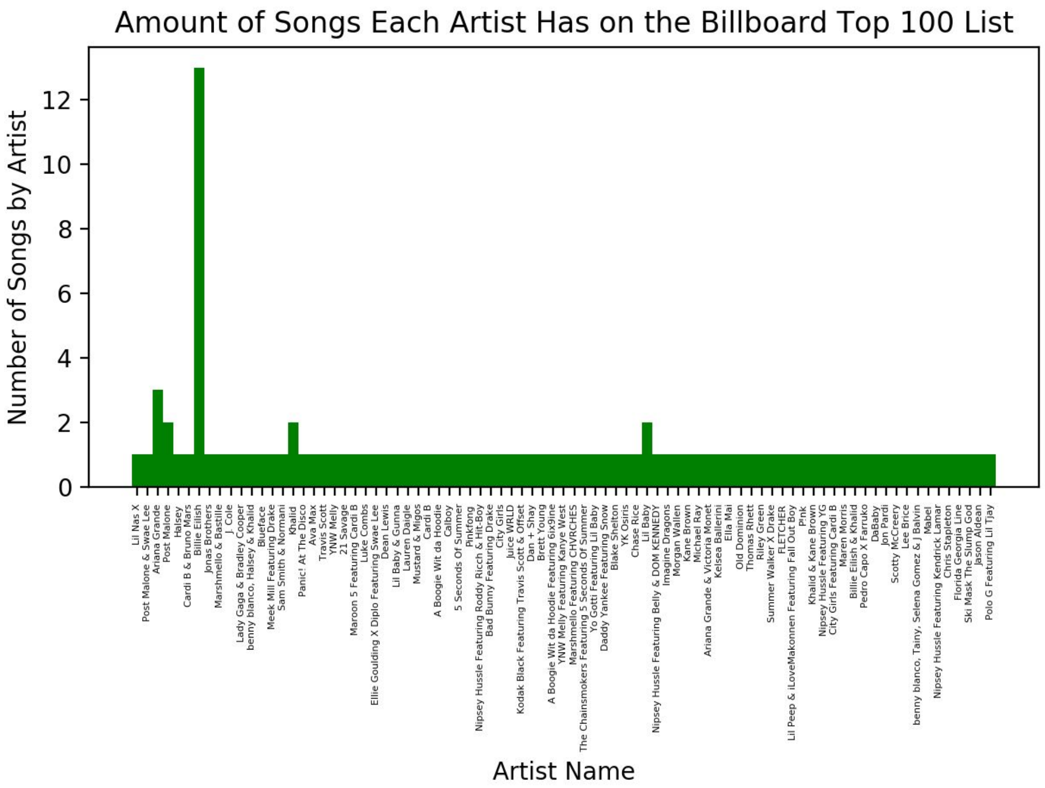
The following are the results of the calculations from the data in the database for Billboard.com (which also appears in a text file contained in our Final Project zip file):

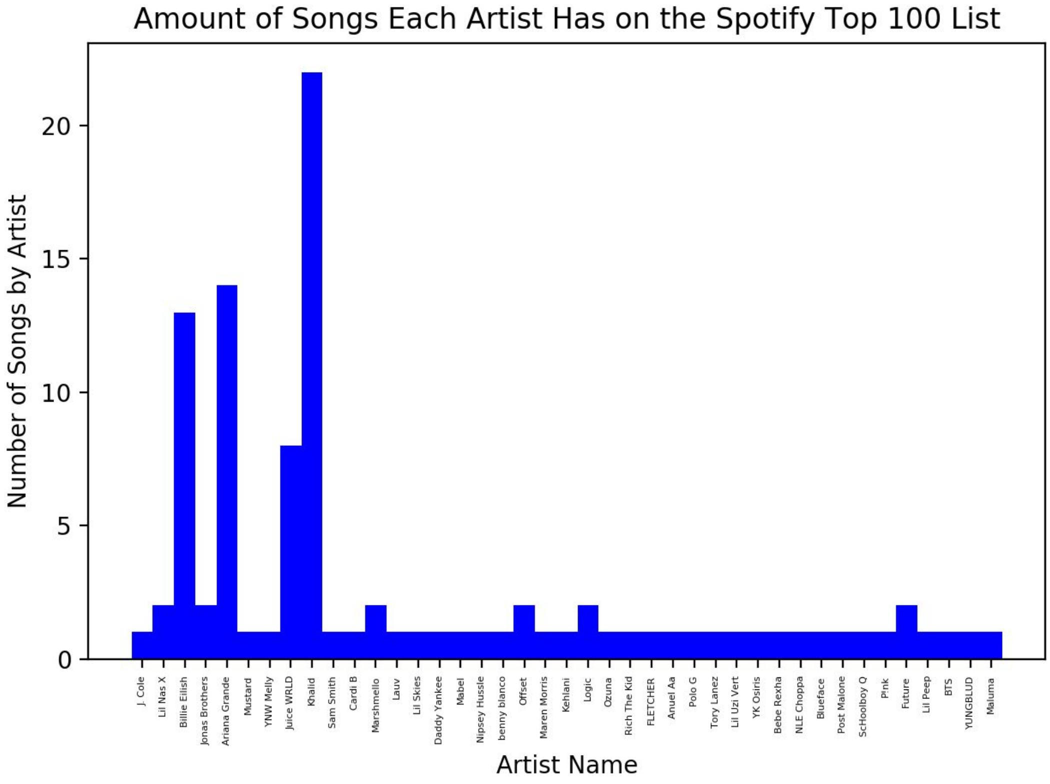
{"Lil Nas X": 1, "Post Malone & Swae Lee": 1, "Ariana Grande": 3, "Post Malone": 2, "Halsey": 1, "Cardi B & Bruno Mars": 1, "Billie Eilish": 13, "Jonas Brothers": 1, "Marshmello & Bastille": 1, "J. Cole": 1, "Lady Gaga & Bradley Cooper": 1, "benny blanco, Halsey & Khalid": 1, "Blueface": 1, "Meek Mill Featuring Drake": 1, "Sam Smith & Normani": 1, "Khalid": 2, "Panic! At The Disco": 1, "Ava Max": 1, "Travis Scott": 1, "YNW Melly": 1, "21 Savage": 1, "Maroon 5 Featuring Cardi B": 1, "Luke Combs": 1, "Ellie Goulding X Diplo Featuring Swae Lee": 1, "Dean Lewis": 1, "Lil Baby & Gunna": 1, "Lauren Daigle": 1, "Mustard & Migos": 1, "Cardi B": 1, "A Boogie Wit da Hoodie": 1, "Calboy": 1, "5 Seconds Of Summer": 1, "Pinkfong": 1, "Nipsey Hussle Featuring Roddy Ricch & Hit-Boy": 1, "Bad Bunny Featuring Drake": 1, "City Girls": 1, "Juice WRLD": 1, "Kodak Black Featuring Travis Scott & Offset": 1, "Dan + Shay": 1, "Brett Young": 1, "A Boogie Wit da Hoodie Featuring 6ix9ine": 1, "YNW Melly Featuring Kanye West": 1, "Marshmello Featuring CHVRCHES": 1, "The Chainsmokers Featuring 5 Seconds Of Summer": 1, "Yo Gotti Featuring Lil Baby": 1, "Daddy Yankee Featuring Snow": 1, "Blake Shelton": 1, "YK Osiris": 1, "Chase Rice": 1, "Lil Baby": 2, "Nipsey Hussle Featuring Belly & DOM KENNEDY": 1, "Imagine Dragons": 1, "Morgan Wallen": 1, "Kane Brown": 1, "Michael Ray": 1, "Ariana Grande & Victoria Monet": 1, "Kelsea Ballerini": 1, "Ella Mai": 1, "Old Dominion": 1, "Thomas Rhett": 1, "Riley Green": 1, "Summer Walker X Drake": 1, "FLETCHER": 1, "Lil Peep & iLoveMakonnen Featuring Fall Out Boy": 1, "P!nk": 1, "Khalid & Kane Brown": 1, "Nipsey Hussle Featuring YG": 1, "City Girls Featuring Cardi B": 1, "Maren Morris": 1, "Billie Eilish & Khalid": 1, "Pedro Capo X Farruko": 1, "DaBaby": 1, "Jon Pardi": 1, "Scotty McCreery": 1, "Lee Brice": 1, "benny blanco, Tainy, Selena Gomez & J Balvin": 1, "Mabel": 1, "Nipsey Hussle Featuring Kendrick Lamar": 1, "Chris Stapleton": 1, "Florida Georgia Line": 1, "Ski Mask The Slump God": 1, "Jason Aldean": 1, "Polo G Featuring Lil Tjay": 1}

The following are the results of the calculations from the data in the database for Spotify (which also appears in a text file contained in our Final Project zip file):

{"J. Cole": 1, "Lil Nas X": 2, "Billie Eilish": 13, "Jonas Brothers": 2, "Ariana Grande": 14, "Mustard": 1, "YNW Melly": 1, "Juice WRLD": 8, "Khalid": 22, "Sam Smith": 1, "Cardi B": 1, "Marshmello": 2, "Lauv": 1, "Lil Skies": 1, "Daddy Yankee": 1, "Mabel": 1, "Nipsey Hussle": 1, "benny blanco": 1, "Offset": 2, "Maren Morris": 1, "Kehlani": 1, "Logic": 2, "Ozuna": 1, "Rich The Kid": 1, "FLETCHER": 1, "Anuel Aa": 1, "Polo G": 1, "Tory Lanez": 1, "Lil Uzi Vert": 1, "YK Osiris": 1, "Bebe Rexha": 1, "NLE Choppa": 1, "Blueface": 1, "Post Malone": 1, "ScHoolboy Q": 1, "P!nk": 1, "Future": 2, "Lil Peep": 1, "BTS": 1, "YUNGBLUD": 1, "Maluma": 1}

<u>The visualizations that we created:</u>


Amount of Songs Each Artist Has on the Billboard Top 100 List


Amount of Songs Each Artist Has on the Spotify Top 100 List

Instructions for running our code:

You will need to have the following items installed in order to run our code:

- billboard.py
- spotipy
- sqlite3
- matplotlib
- json

Run the files in the following order: 1. fin-proj.py 2. billboard_barchart.py 3. spotify_barchart.py

To run the file, fin-proj.py you must enter the following into the terming:

```
python3 fin-proj.py
```

To run the file, billboard_barchart.py you must enter the following into the terming:

```
python3 billboard_barchart.py
```

To run the file, spotify_barchart.py you must enter the following into the terming:

```
python3 spotify_barchart.py
```

Documentation for each function:

- **grab_data_and_populate_database()**
    - First, this function authenticates Spotify with our Spotify Client ID and Secret.
    - Next, it grabs data from the billboard API using the .ChartData attribute. We chose the Chart “Hot 100” in order to get the top 100 songs from Billboard.com.
    - Next, it connects to the sqlite database and created a table called ’Billboard_top’ and for each song in the top 100 list, this function inserts its title and artist’s name into the table on SQL. Last, it commits the changes to the database.
    - Next, it grabs data from the Spotify API using the .search attribute to get the top 100 songs from Spotify.
    - Next, it connects to the sqlite database and created a table called ’Spotify_top’ and for each song in the top 100 list, this function inserts its title and artist’s name into the table on SQL. Last, it commits the changes to the database.

    - **To summarize, this function grabs the top 100 songs from Billboard.com and Spotify and populates the sqlite database with the corresponding titles and artist names of these songs.**

- **create_billboard_barchart()**
    - First, it connects to the ‘Billboard_top.sqlite’ data table and counts the number of times each artist appears on the ‘Hot 100’ Billboard chart. It creates a dictionary where the keys are the artists’ names and the values are the frequencies of each artist’s appearance on the top chart.
    - Next, it writes out this calculated data (artist_dict) to a text file.
    - Next, it assigns the keys of the dictionary (the artists’ names) as the x values and the values of the dictionary (the frequencies) as the y values.
    - Next, it plots a bar chart with these x and y values and it adjusts the centering, the width of the bars, the color of the bars, and the arrangement of the x-axis labels.
    - Next, it assigns a label to the y axis and a label to the x axis.
    - Next, it assigns a title to the bar graph.
    - Next, it adjusts the placement of the x-axis labels, tightens the layout of the graph, saves it as ‘billboard_artist_freq.png’ and, finally, displays the bar graph.
- **create_spotify_barchart()**
    - Similar to the previous function, it first connects to the ‘Spotify_top.sqlite’ data table and counts the number of times each artist appears on Spotify’s top 100. It creates a dictionary where the keys are the artists’ names and the values are the frequencies of each artist’s appearance on the top chart.
    - Next, it writes out this calculated data (spotify_artist_dict) to a text file.
    - Next, it assigns the keys of the dictionary (the artists’ names) as the x values and the values of the dictionary (the frequencies) as the y values.
    - Next, it plots a bar chart with these x and y values and it adjusts the centering, the width of the bars, the color of the bars, and the arrangement of the x-axis labels.
    - Next, it assigns a label to the y axis and a label to the x axis.
    - Next, it assigns a title to the bar graph.
    - Next, it adjusts the placement of the x-axis labels, tightens the layout of the graph, saves it as ‘spotify_artist_freq.png’ and, finally, displays the bar graph.

Resources Used:

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
April 7, 2019	How to git pull	https://stackoverflow.com/questions/1408790/how-do-i-pull-my-project-from-github	No
April 9, 2019	How to Install spotipy	https://www.youtube.com/watch?v=tmt5SdvTqUI	yes
April 9, 2019	Learn more about spotipy	https://spotipy.readthedocs.io/en/latest/	yes
April 10, 2019	Learn how to install billboard API and about its chart entry attributes	https://github.com/guoguo12/billboard-charts	yes
April 13, 2019	How to create databases in SQLite	SI 206 Discussion Section 10: https://umich.instructure.com/courses/270541/files/folder/Discussions/Discussion%2010?preview=10953952	yes
April 14, 2019	Using matplotlib	SI 206 Discussion 11: https://umich.instructure.com/courses/270541/files/folder/Discussions/Discussion%2011?preview=11041926	yes
April 16, 2019	How to get data in JSON file	https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/	yes