SI 206 Final Report

Ewelina Papiez

https://github.com/epapiez/FinalSI206-epapiez

Goals

- (My original project goals were to compare the demographics of users on Facebook vs. Instagram, but the APIs were no longer available for public use.)
- To compare the favorites a song achieves on Twitter to its popularity on the Hot100 database.
- Use at least two APIs/websites.
- Create at least two visualizations.

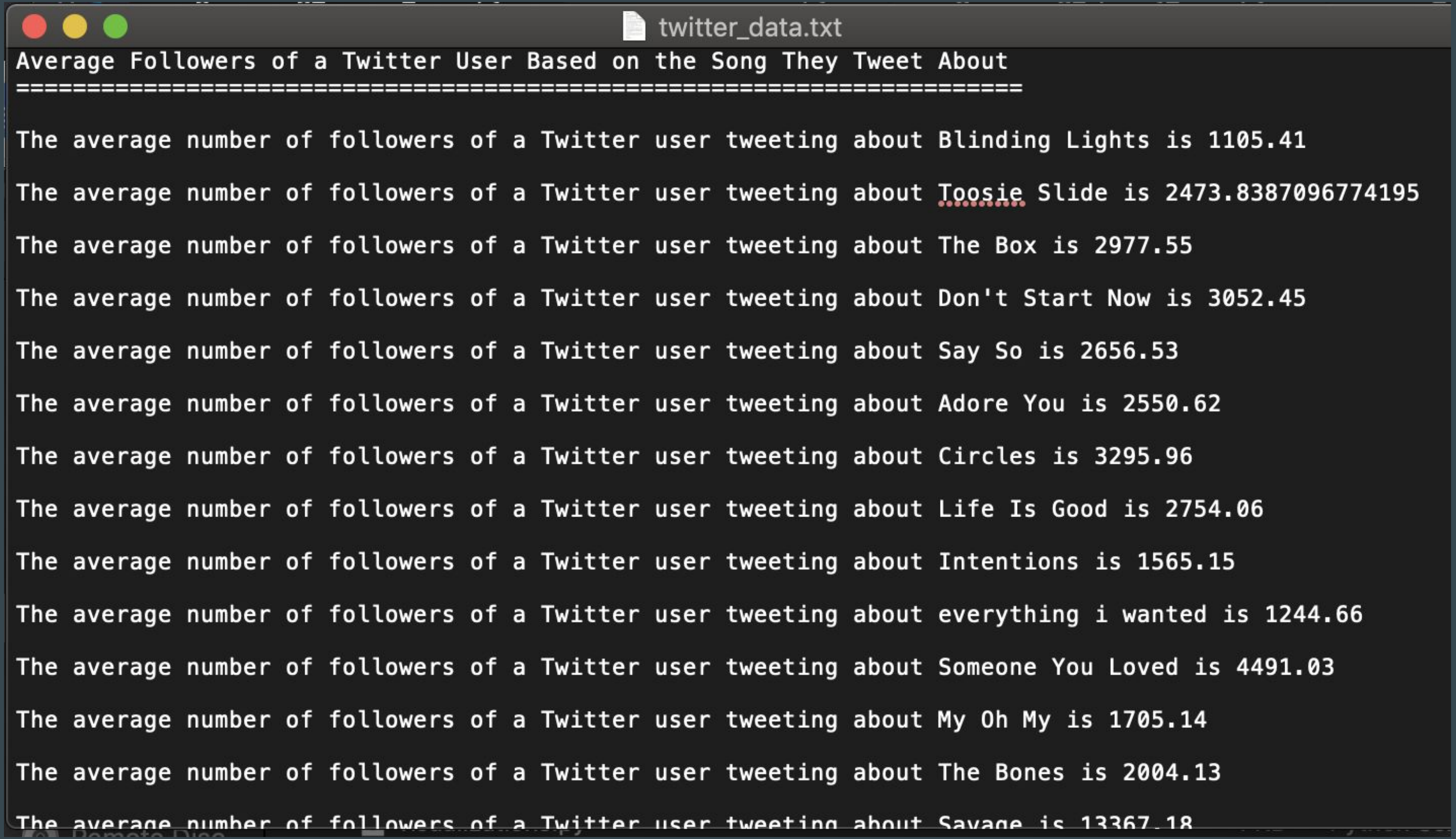
Goals achieved

- I found that it was difficult to find out exactly how many favorites a Tweet that mentions a song receives, so I also decided to find the average amount of followers that a Twitter user tweeting about a certain song has.
- I found the average number of weeks a song is on the Billboard Hot 100.
- I decided to use a third API/website, Spotipy, in order to compare the popularity of a song across three different sources (Spotify, Twitter, Billboard Hot 100). This third API was mostly just a supplemental piece and does not store 20 rows at a time or contain 100 rows of data, but the other two APIs do.
- I created 5 visualizations.

Problems

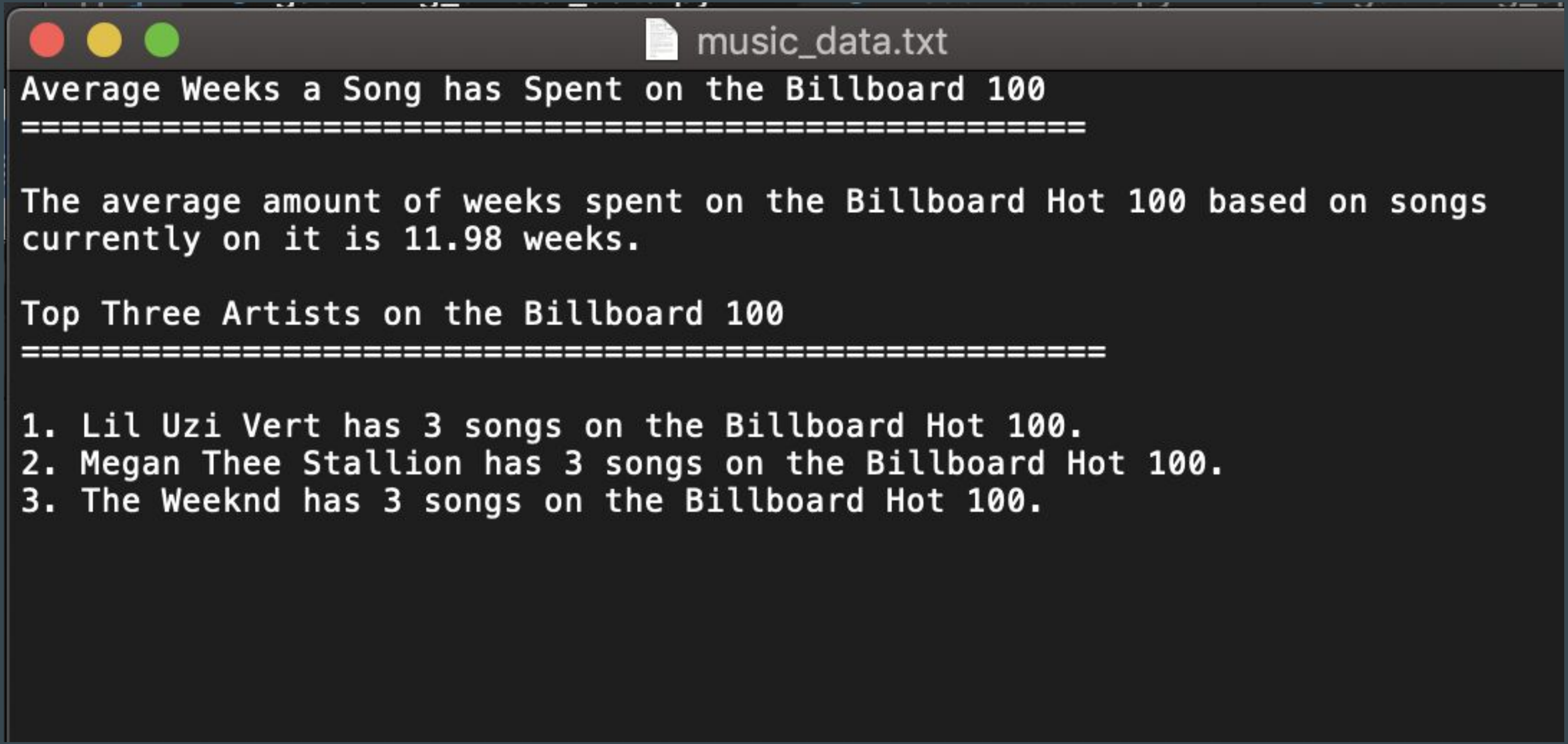
- Unfortunately I had to work on this project by myself due to communication issues with my partner.
- When using Spotipy, it was extremely difficult to find the exact song and artist due to the formatting differences between the Billboard Hot 100 and Spotipy. For example, a song was called “Roses (Imbanek Remix)” on the Billboard Hot 100 but “Roses - Imabnek Remix” on Spotipy, so I had to adjust for these differences to make sure I was finding the popularity of the correct song.
- When using Twitter and adding songs to the Twitter_Data table, there were two songs that were both called “Heartless” but were by two different artists. I had to find a way to differentiate between the two songs. I solved this problem using the artist_id column.

Calculations



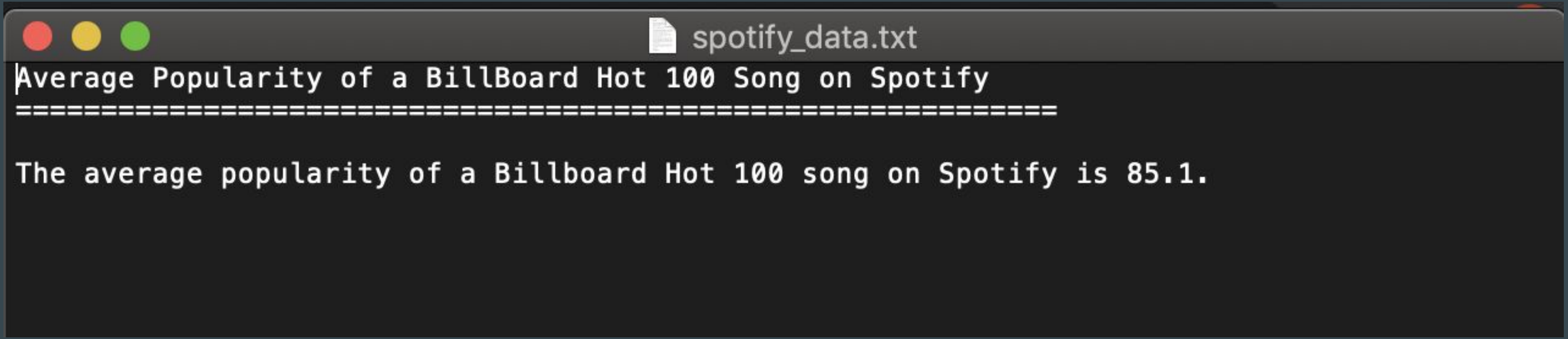
```
Average Followers of a Twitter User Based on the Song They Tweet About
======================================================================

The average number of followers of a Twitter user tweeting about Blinding Lights is 1105.41

The average number of followers of a Twitter user tweeting about Toosie Slide is 2473.8387096774195

The average number of followers of a Twitter user tweeting about The Box is 2977.55

The average number of followers of a Twitter user tweeting about Don't Start Now is 3052.45

The average number of followers of a Twitter user tweeting about Say So is 2656.53

The average number of followers of a Twitter user tweeting about Adore You is 2550.62

The average number of followers of a Twitter user tweeting about Circles is 3295.96

The average number of followers of a Twitter user tweeting about Life Is Good is 2754.06

The average number of followers of a Twitter user tweeting about Intentions is 1565.15

The average number of followers of a Twitter user tweeting about everything i wanted is 1244.66

The average number of followers of a Twitter user tweeting about Someone You Loved is 4491.03

The average number of followers of a Twitter user tweeting about My Oh My is 1705.14

The average number of followers of a Twitter user tweeting about The Bones is 2004.13

The average number of followers of a Twitter user tweeting about Savage is 13367.18
```

Data calculated from gathering_twitter_data.py



```
Average Weeks a Song has Spent on the Billboard 100
=====================================================


The average amount of weeks spent on the Billboard Hot 100 based on songs
currently on it is 11.98 weeks.

Top Three Artists on the Billboard 100
======================================================


1. Lil Uzi Vert has 3 songs on the Billboard Hot 100.
2. Megan Thee Stallion has 3 songs on the Billboard Hot 100.
3. The Weeknd has 3 songs on the Billboard Hot 100.
```

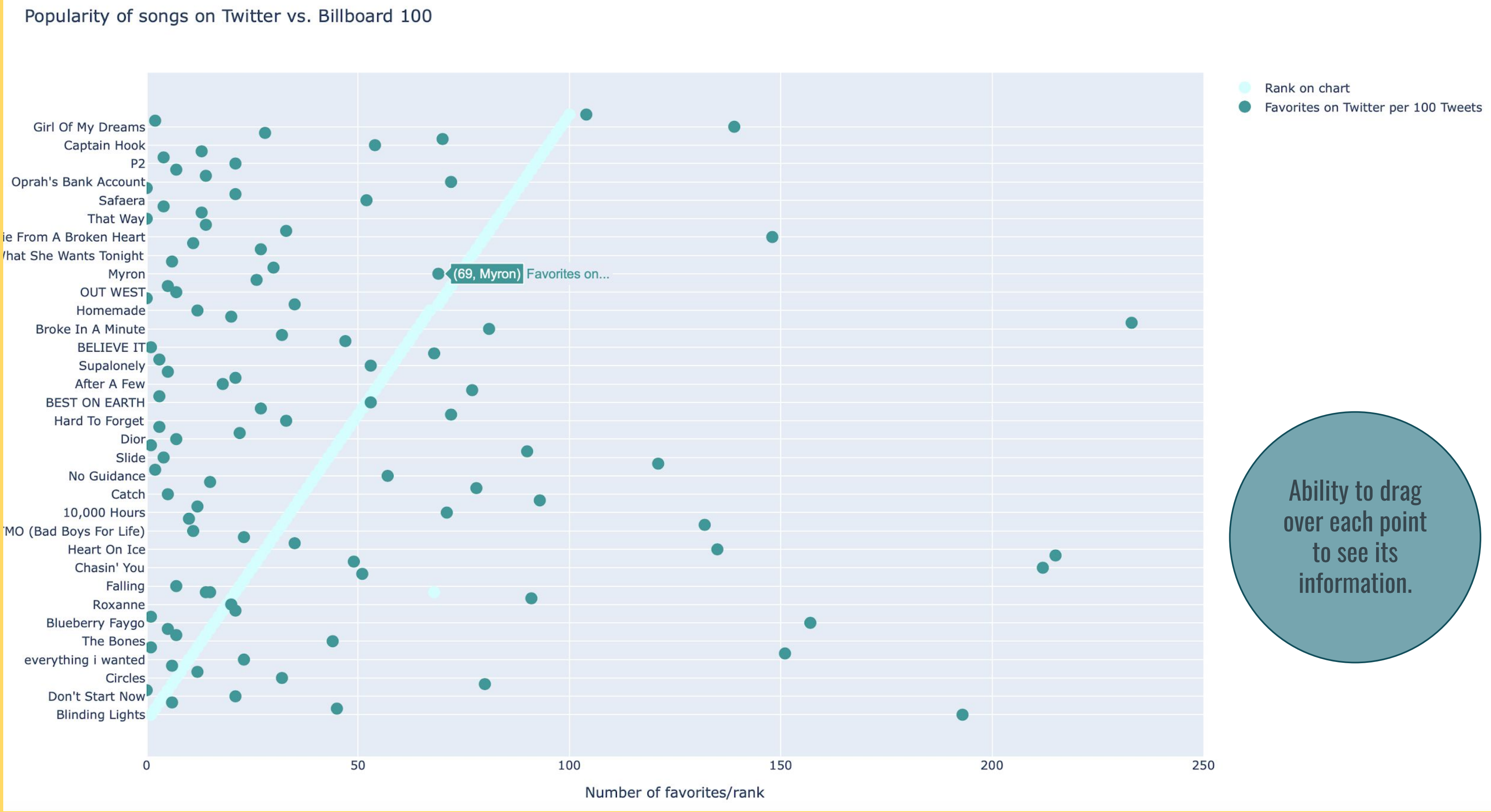
Data calculated from Hot100.py



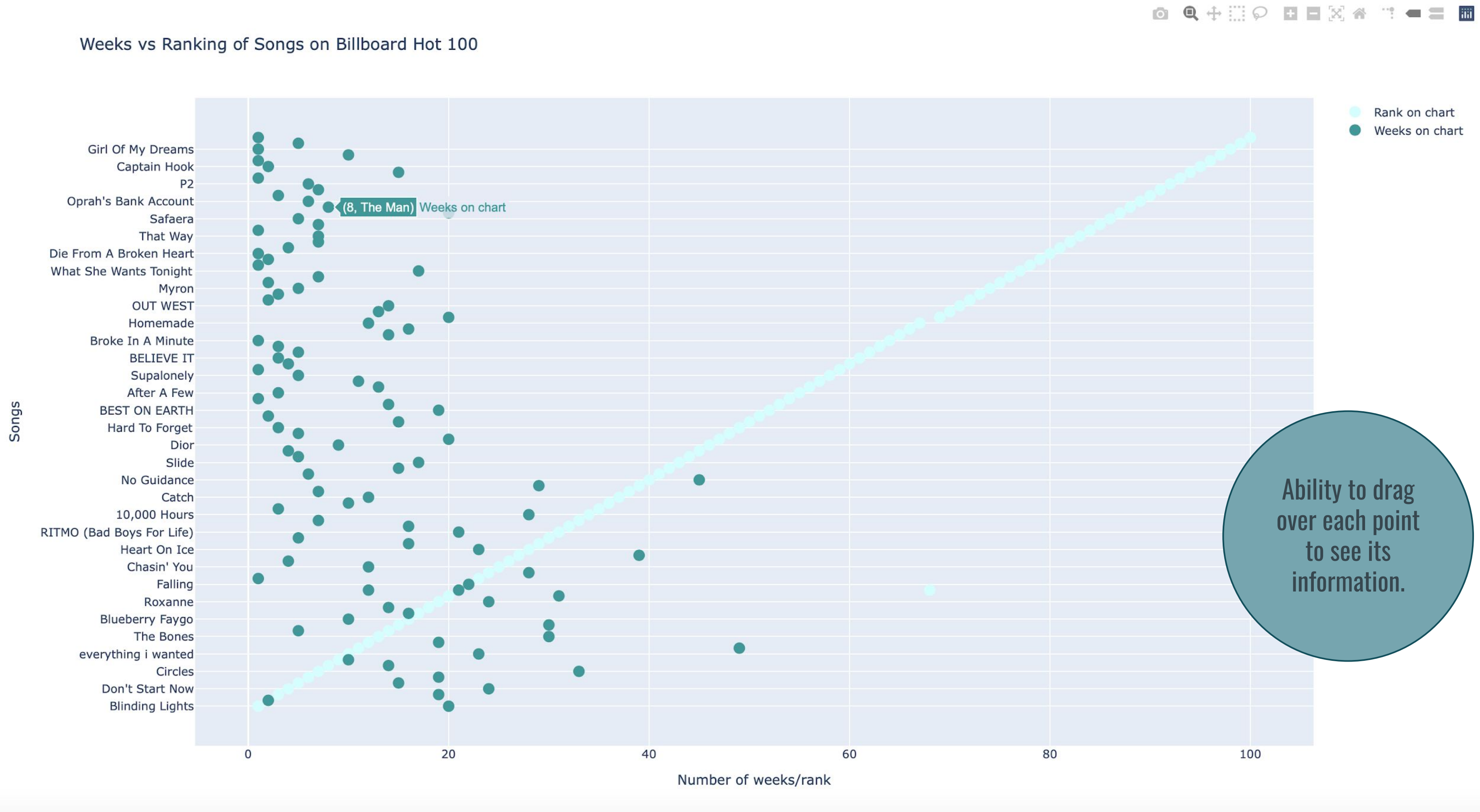
```
Average Popularity of a BillBoard Hot 100 Song on Spotify
=============================================================

The average popularity of a Billboard Hot 100 song on Spotify is 85.1.
```

Data calculated from gathering_spotify_data.py

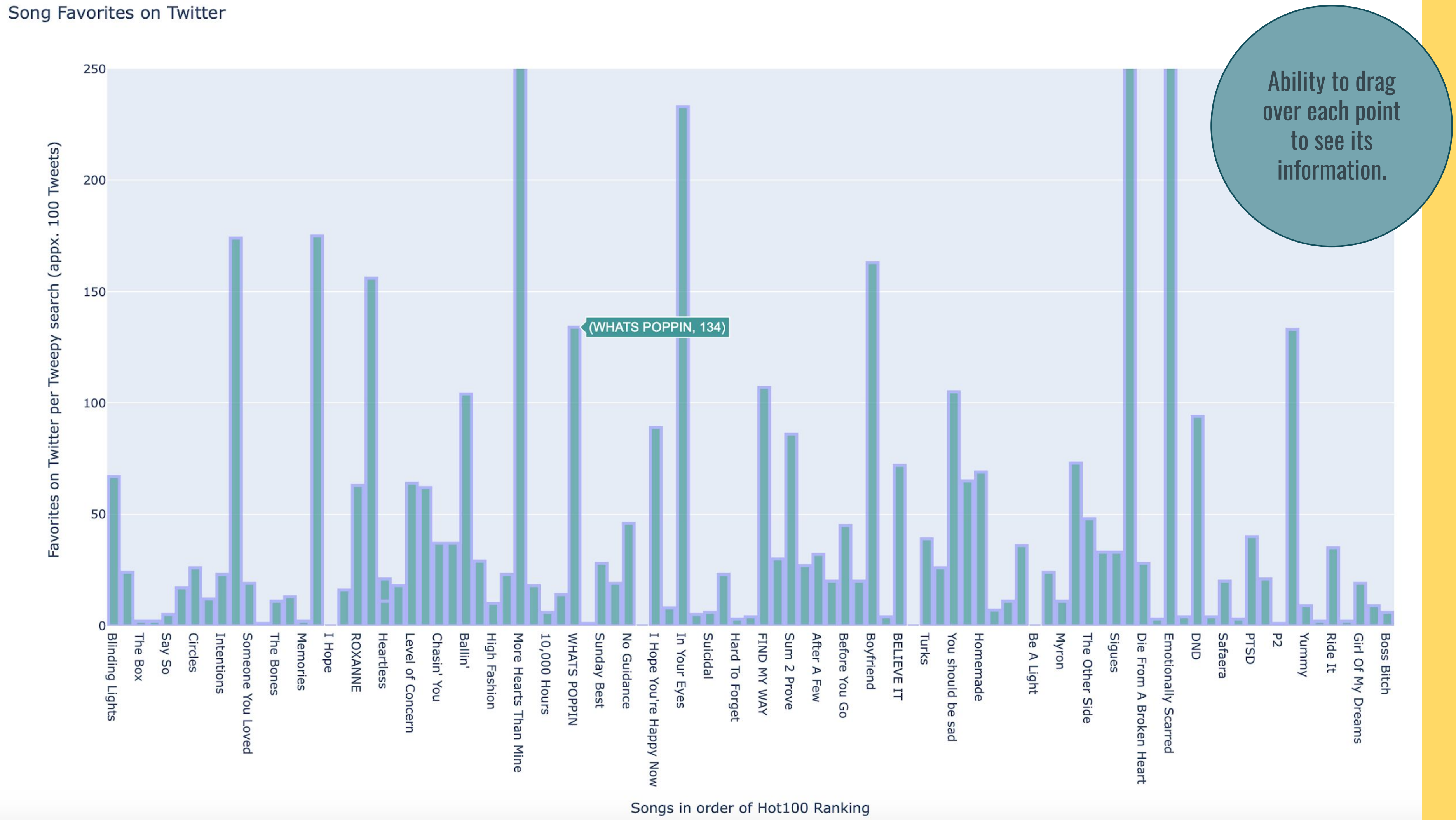
Visualizations



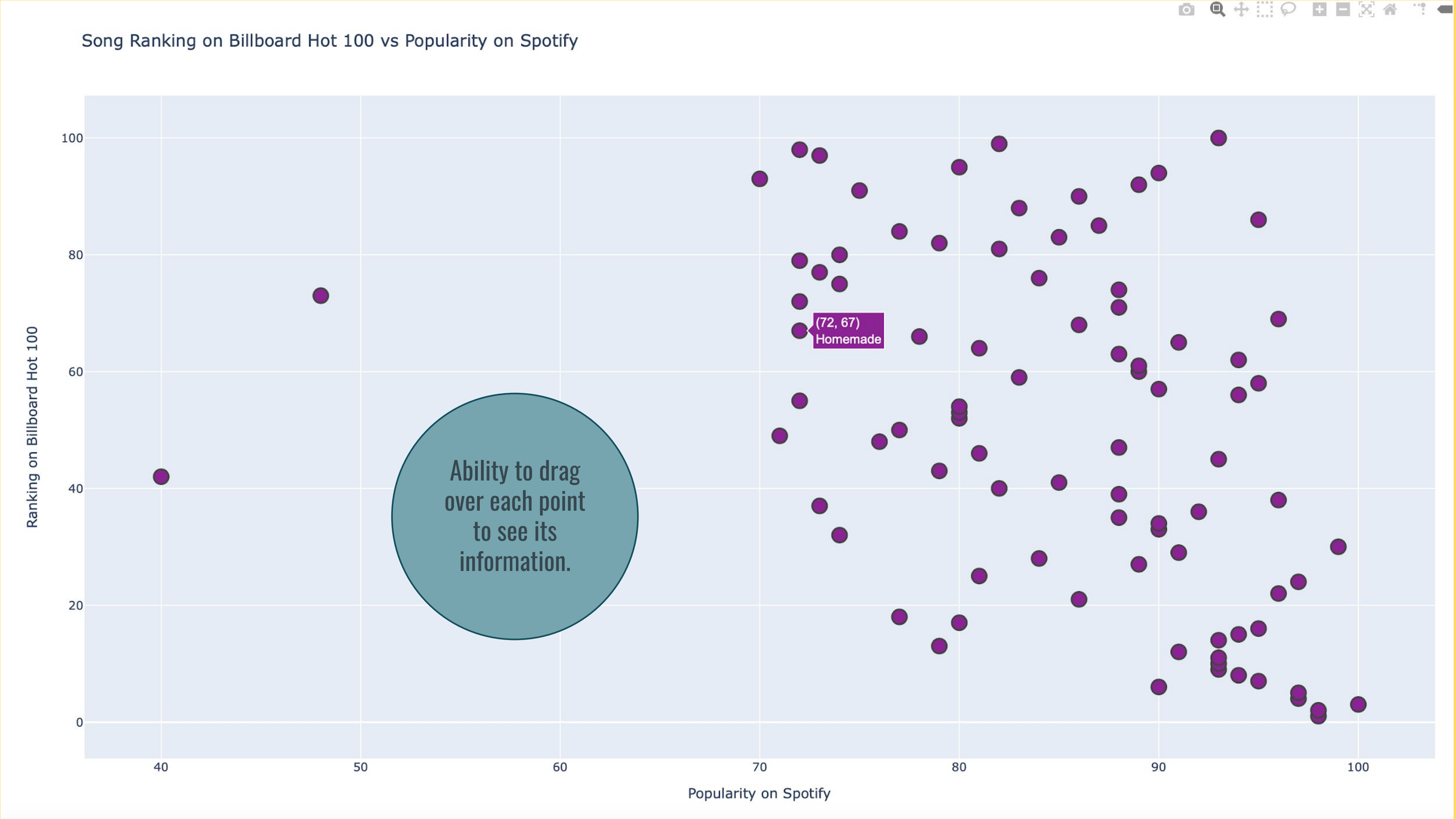
Ability to drag over each point to see its information.



Ability to drag over each point to see its information.

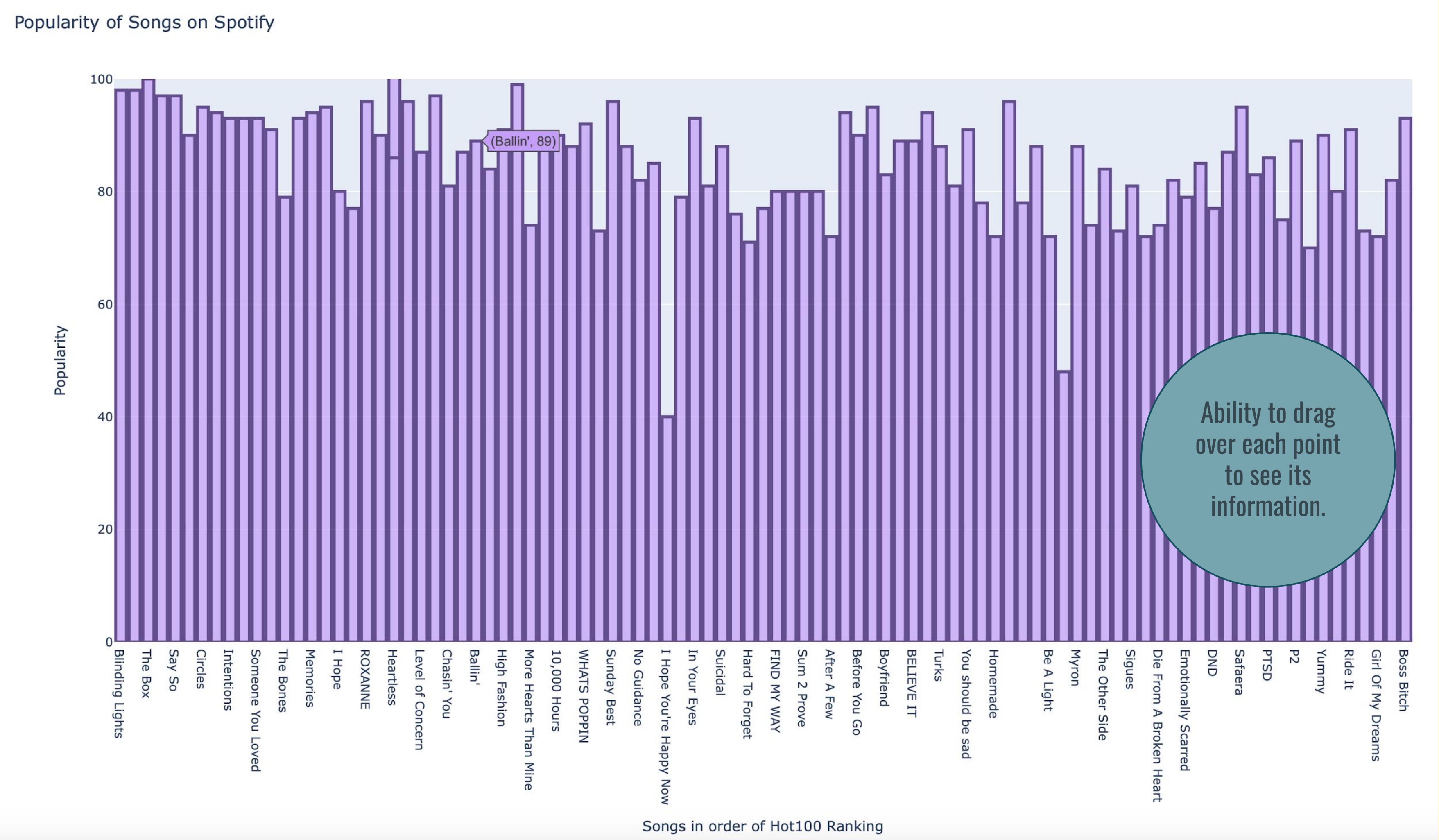


Ability to drag over each point to see its information.



Song Ranking on Billboard Hot 100 vs Popularity on Spotify

Ability to drag over each point to see its information.



Ability to drag over each point to see its information.

Instructions

1. Make sure that the “music.db” database does not already exist in your files. If it does, delete it so that it can be recreated.
2. Make sure that the Spotipy and Tweepy libraries are installed on your machine. Do this by typing “pip install spotipy” and “pip install tweepy” into your terminal.

3. Open up the zipped file. **The first file that should be run is called “hot_100.py”.** This file should be run **five times**. The database will create a table called “Artist_Ids” and “Hot100”. The first time, the “Artist_Ids” table will be filled completely. This is because the “Artist_Ids” must be used in order to complete the “Hot100” table. The “Hot100” table fills in 20 rows of data at a time, resulting in 100 rows.

4. **The second file that should be run is called “gathering_twitter_data.py”.** Run this file **five times.** This one will take a little bit longer to run each time. Two tables will be created, “Follower_Data” and “Twitter_Data”. “Follower_Data” shows the total amount of followers that the Twitter users tweeting about a song have. I used this to calculate the average amount of followers a Twitter user tweeting about a certain song has. “Twitter_Data” shows how many favorites a tweet has per the number of mentions it has. The maximum number of mentions is 100, but some songs do not have 100 mentions. This was used to calculate the average number of favorites a certain song has on Twitter.

5. **The third file that should be run is called "gathering_spotify_data.py".** This file should **only be run once** (it is the third API as a supplemental piece out of curiosity so I didn't have it add twenty rows at a time, the other files have at least one table that add twenty rows at a time).

*This is my supplemental/"for fun" file.

6. Lastly, **the fourth file to run is called “visualizations.py”**. This file will produce the visualizations.

7. To see the calculations made from the data, open “twitter_data.txt” and “movie_data.txt” which should have been created while running the other four files.

8. Open “music.db” to see the completed database, which should contain 5 tables total.

Code Documentation

```
get_songs_and_artists():
    """No inputs. Returns a list of tuples in the format (song, artist, weeks). Uses
BeautifulSoup to read the top 100 songs, along with their artist and the weeks it has
been on the chart"""
setUpDatabase(db_name):
    """Takes the name of a database, a string, as an input. Returns the cursor and
connection to the database."""
fillup_hot_100_table(cur, conn):
    """Takes the database cursor and connection as inputs. Does not return anything.
Fills in the Hot100 table with songs and their artist_ids and how long they've been on
the charts. The creation_id is each song's unique identification number."""
set_up_tables(cur, conn):
    """ Takes the database cursor and connection as inputs. Returns nothing. Creates two
tables, one that will hold artists and their artist_ids, and another that holds the top
100 songs, along with their artist_ids and weeks on chart."""
```

hot_100.py

```
find_top_three_artists():
    """Takes nothing as an input. Returns a list of strings (the three artists with the
most songs on the Billboard Hot 100). Utilizes the function get_songs_and_artists(). """
get_average_weeks(cur, conn):
    """Takes the database cursor and connection as inputs. Returns a string with the
average number of weeks a song is on the chart."""
write_data_to_file(filename, cur, conn):
    """Takes in a filename (string), the database cursor and connection as inputs.
Returns nothing. Creates a file and writes return values of the functions
find_top_three_artists() and get_average_weeks() to the file. """
main():
    """Takes nothing as an input and returns nothing. Calls the functions
setUpDatabase(), set_up_tables(), set_up_artist_id_table(), fill_up_hot_100_table(), and
write_data_to_file(). Closes the database connection."""
```

hot_100.py

```
find_top_three_artists():
    """Takes nothing as an input. Returns a list of strings (the three artists with the
most songs on the Billboard Hot 100). Utilizes the function get_songs_and_artists(). """
get_average_weeks(cur, conn):
    """Takes the database cursor and connection as inputs. Returns a string with the
average number of weeks a song is on the chart."""
write_data_to_file(filename, cur, conn):
    """Takes in a filename (string), the database cursor and connection as inputs.
Returns nothing. Creates a file and writes return values of the functions
find_top_three_artists() and get_average_weeks() to the file. """
main():
    """Takes nothing as an input and returns nothing. Calls the functions
setUpDatabase(), set_up_tables(), set_up_artist_id_table(), fill_up_hot_100_table(), and
write_data_to_file(). Closes the database connection."""
```

gathering_twitter_data.py

```
join_tables(cur, conn):
   """Takes in the database cursor and connection as inputs. Returns a list of tuples in the format
(song, artist) where the artist ids are the same."""
find_artists(artist_name):
   """Takes in an artist name as it is listed in the Hot100 table, then formats it so that it matches
Spotify. Returns a list of strings of each artist that worked on the given song."""
set_up_spotify_table(cur, conn):
   """Takes in the database cursor and connection as inputs. Returns nothing. Creates a table called
Spotify_Data. Searches the Spotify popularity of each song and inserts it into the table."""
average_popularity(cur, conn):
   """Take the database cursor and connection as inputs. Returns an integer, which is the average
popularity of Billboard Hot 100 songs on Spotify."""
write_data_to_file(filename, cur, conn):
   """Takes in a filename (string) as an input and cur,conn. Returns nothing. Creates a file and
writes return value of the function average_popularity() to the file. """
main():
"""Takes no inputs and returns nothing. Calls the function set_up_spotify_table()."""
```

gathering_spotify_data.py

```
main():
    """Takes no inputs and returns nothing. Selects data from the database in order to
create visualizations (two dot plots, a scatterplot, and two bar charts.) """
```

visualizations.py

Resources

Date	Issue description	Location of resource	Result (did it solve the issue?)
4/13/20	Had an error “EOL while scanning string literal”, which I’ve never seen before	https://discover.cs.ucsb.edu/commonerrors/error/1010.xml	Yes, I had a ‘ opening the string and a “ closing the string.
4/13/20	Tweepy documentation used for a variety of reasons to better understand Tweepy	http://docs.tweepy.org/en/latest/	Yes, I was able to better understand the Tweepy library
4/13/20	How to use a database across multiple files in Python	https://stackoverflow.com/questions/51065457/how-to-import-a-sql-file-to-python	Yes I was able to use several files that connect to the same database.
4/14/20	I was getting an error while using Tweepy and didn’t know how to solve it.	https://stackoverflow.com/questions/38775997/getting-this-error-when-using-tweepy	Yes, I added the line that the website suggested to my code.
4/26/20	I was getting a really long decimal for one of my calculations and wanted to round it.	https://realpython.com/python-rounding/	Yes, I was able to round the value.

Date	Issue description	Location of resource	Result (did it solve the issue?)
4/15/20	How to change the count of searches returned by .search (Tweepy)	https://stackoverflow.com/questions/22469713/managing-tweepy-api-search	Yes, I was able to choose how many results to return.
4/21/20	How to get started with Spotipy	https://medium.com/@maxtingle/getting-started-with-spotifys-api-spotipy-197c3dc6353b	Yes, this helped me install Spotipy and showed basic uses.
4/22/20	I wanted to make sure I was correctly checking if a string is inside a string	https://stackoverflow.com/questions/5319922/python-check-if-word-is-in-a-string	Yes, I used this to compare the way a song was written in Spotify vs the Billboard Hot 100.
4/23/20	Looking for different visualizations to create on Plotly and how	https://plotly.com/python/	Yes, I was able to use the documentation to create my own visualizations.

Tables

Tables created from Hot100.py

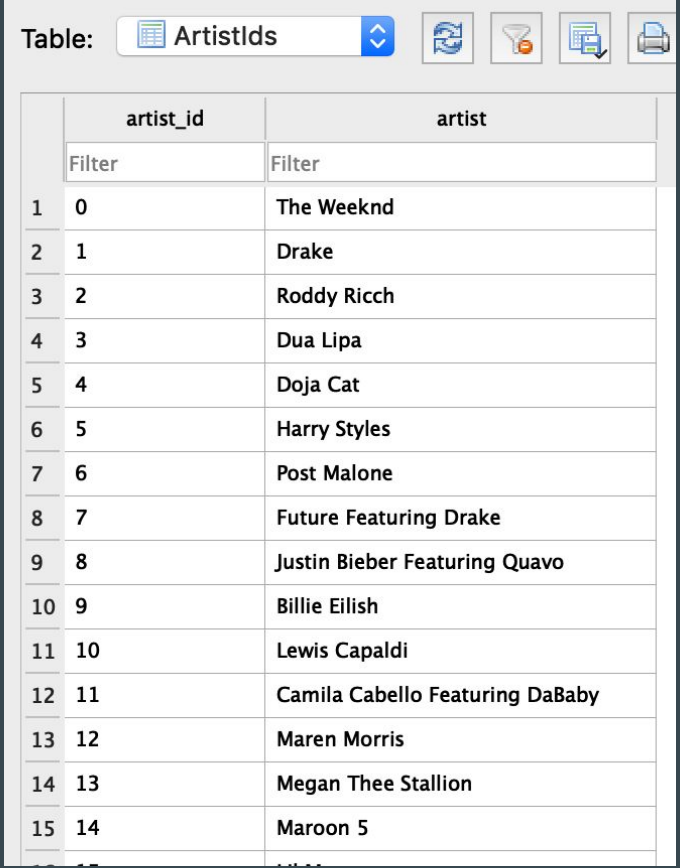


Table: ArtistIds

	artist_id	artist
	Filter	Filter
1	0	The Weeknd
2	1	Drake
3	2	Roddy Ricch
4	3	Dua Lipa
5	4	Doja Cat
6	5	Harry Styles
7	6	Post Malone
8	7	Future Featuring Drake
9	8	Justin Bieber Featuring Quavo
10	9	Billie Eilish
11	10	Lewis Capaldi
12	11	Camila Cabello Featuring DaBaby
13	12	Maren Morris
14	13	Megan Thee Stallion
15	14	Maroon 5

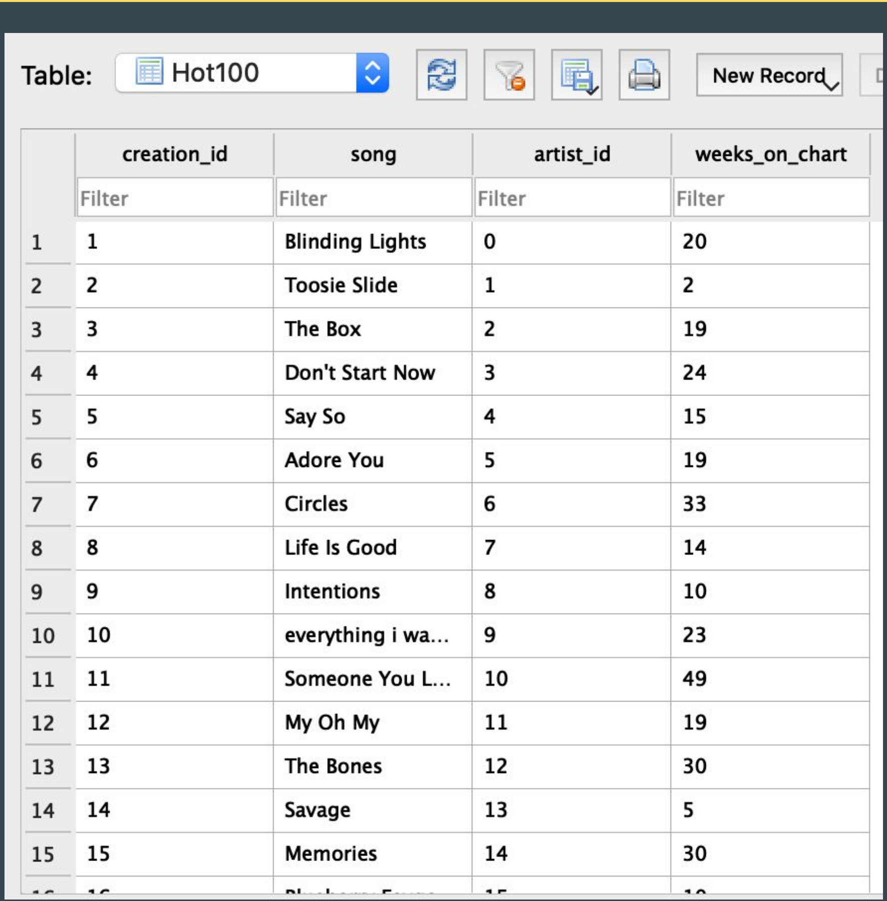
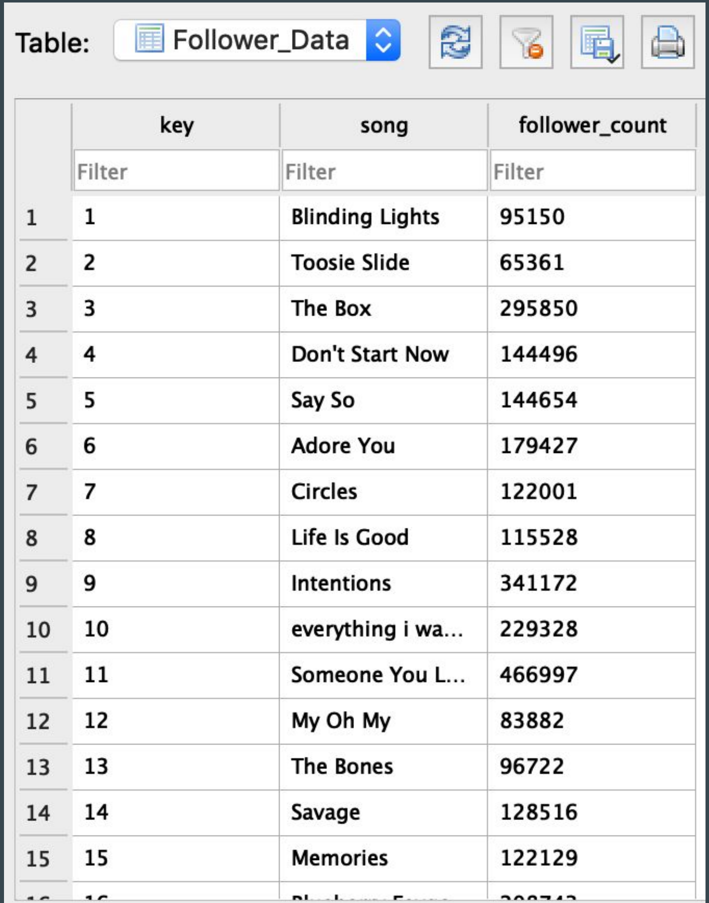


Table: Hot100

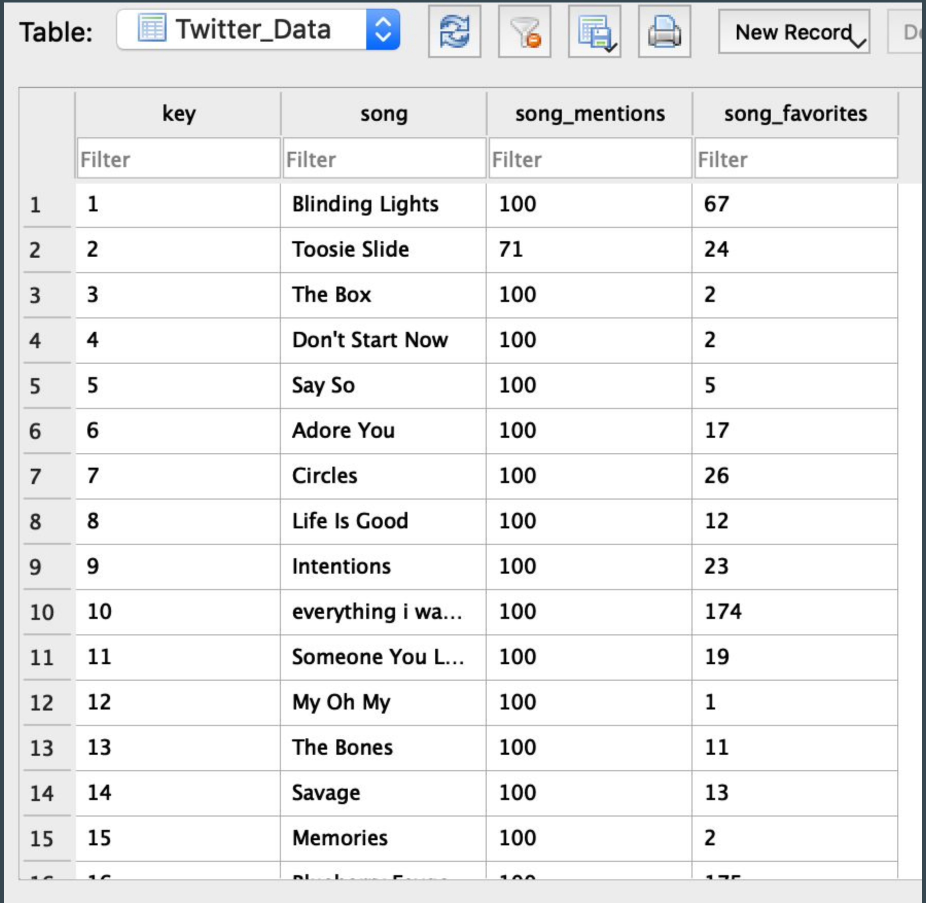
	creation_id	song	artist_id	weeks_on_chart
	Filter	Filter	Filter	Filter
1	1	Blinding Lights	0	20
2	2	Toosie Slide	1	2
3	3	The Box	2	19
4	4	Don't Start Now	3	24
5	5	Say So	4	15
6	6	Adore You	5	19
7	7	Circles	6	33
8	8	Life Is Good	7	14
9	9	Intentions	8	10
10	10	everything i wa...	9	23
11	11	Someone You L...	10	49
12	12	My Oh My	11	19
13	13	The Bones	12	30
14	14	Savage	13	5
15	15	Memories	14	30

*artist_id is the shared key

Tables created from gathering_twitter_data.py



	key	song	follower_count
	Filter	Filter	Filter
1	1	Blinding Lights	95150
2	2	Toosie Slide	65361
3	3	The Box	295850
4	4	Don't Start Now	144496
5	5	Say So	144654
6	6	Adore You	179427
7	7	Circles	122001
8	8	Life Is Good	115528
9	9	Intentions	341172
10	10	everything i wa...	229328
11	11	Someone You L...	466997
12	12	My Oh My	83882
13	13	The Bones	96722
14	14	Savage	128516
15	15	Memories	122129



	key	song	song_mentions	song_favorites
	Filter	Filter	Filter	Filter
1	1	Blinding Lights	100	67
2	2	Toosie Slide	71	24
3	3	The Box	100	2
4	4	Don't Start Now	100	2
5	5	Say So	100	5
6	6	Adore You	100	17
7	7	Circles	100	26
8	8	Life Is Good	100	12
9	9	Intentions	100	23
10	10	everything i wa...	100	174
11	11	Someone You L...	100	19
12	12	My Oh My	100	1
13	13	The Bones	100	11
14	14	Savage	100	13
15	15	Memories	100	2

Table created from gathering_spotify_data.py

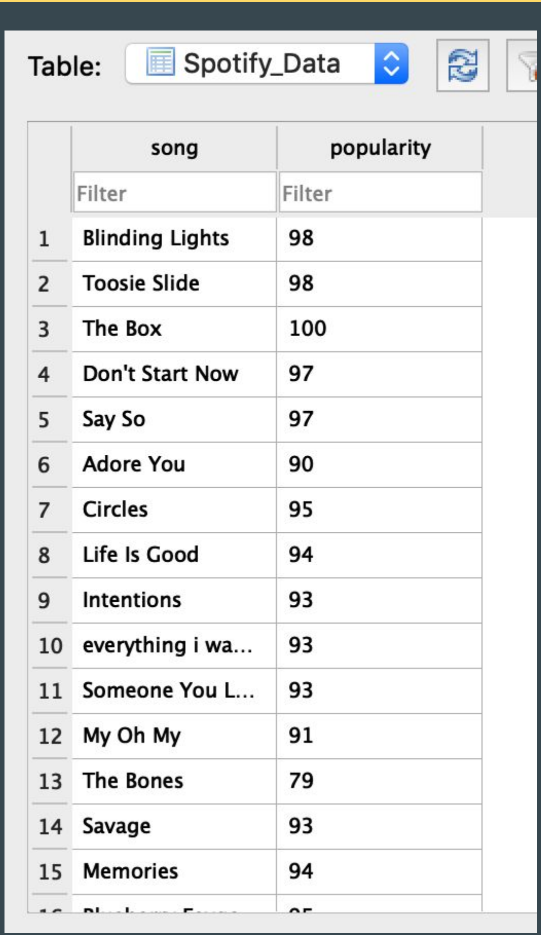


Table: Spotify_Data

	song	popularity
1	Blinding Lights	98
2	Toosie Slide	98
3	The Box	100
4	Don't Start Now	97
5	Say So	97
6	Adore You	90
7	Circles	95
8	Life Is Good	94
9	Intentions	93
10	everything i wa...	93
11	Someone You L...	93
12	My Oh My	91
13	The Bones	79
14	Savage	93
15	Memories	94

*"The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are." -https://community.spotify.com/t5/Content-Questions/Artist-popularity/td-p/4415259