

SI 206 Final Project - Fall 2024

Pari Dar, Monica He, Samantha Pui

GITHUB LINK: <https://github.com/monhemu/SI206-Final-Project.git>

1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)

Original Goals:

- Use Genius API, Billboard API, and Wikipedia web scraping to determine whether pop songs tended to take up more of the weekly top 40 around the beginning of the last 3 major recessions in America
 - Quickly discovered that none of these sites (or other APIs) provided genre data
 - Found that the genius API was hard to work with
 - Billboard was paywalled

Goals we Achieved:

- Observe the relationship among the top 25 songs on iTunes, the nationalities of their respective artists, and news articles about those artists.
 - See if there is a correlation between when the artist trends in the news vs. when their songs first charted. We will do this based off of the past 3 months of news with music artists from the US, based on top 25 songs per week.
 - See which songs stayed on the charts the longest by week from a 3 month time period.
 - View the breakdown of non-American countries that represent the top 25 songs' respective artists.

Our APIs/Sites:

- <https://mediastack.com/>
 - Find the most popular news about the top 5 trending music artists in the US for the past 3 months
- <https://kworkb.net/ww/archive/>
 - Web scrape for top 25 songs each week for last 3 months
- https://musicbrainz.org/doc/MusicBrainz_API
 - Retrieve artist information about their country of origin.

2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)

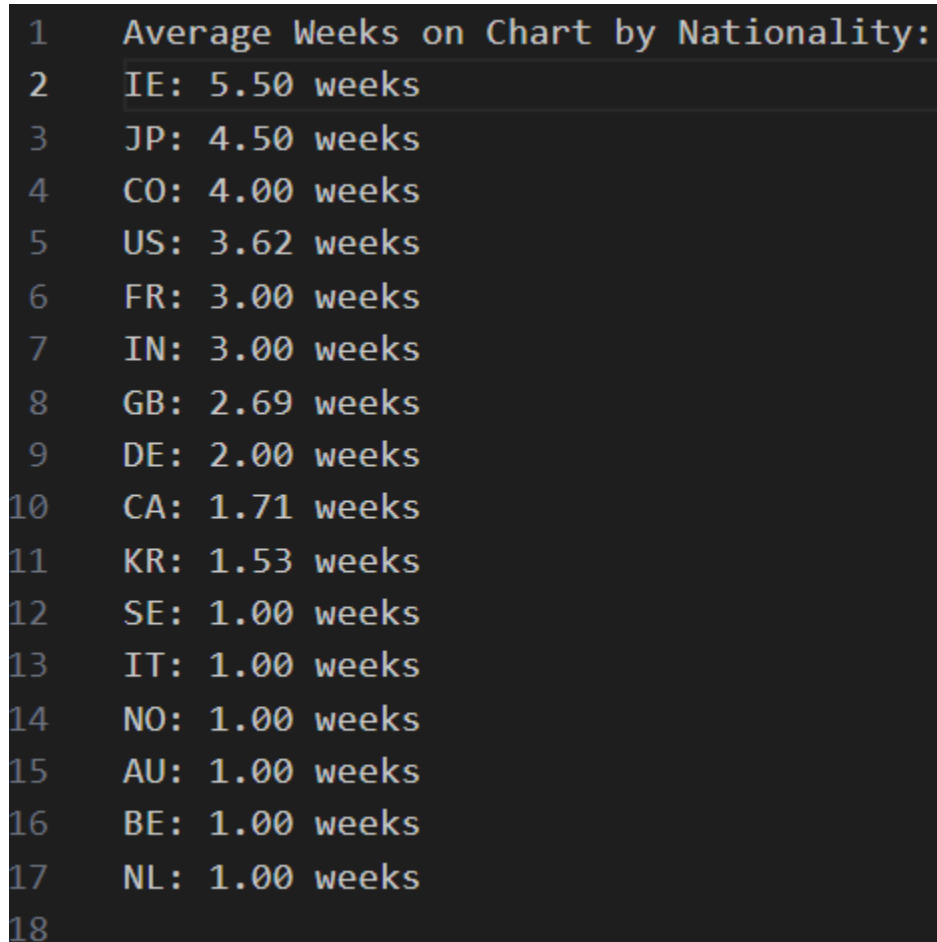
We were able to work with all 3 of our intended APIs/sites. We were able to track and create visualizations for the amount of weeks songs trended based off of charts from the past 3 months.

We were also able to create visualizations for the top 5 charting artists from the US vs. the amount of news written about them over the same time period.

3. The problems that you faced (10 points)

The biggest challenge of our challenge was settling on a project idea and scope, since we frequently ran into the roadblock of finicky or unusable APIs. Our topic had around 4-5 iterations before our final. We had to rewrite some tables in order to get JOIN to work properly. Mediastack has limited API pulls which created an issue.

4. The calculations from the data in the database (i.e. a screenshot) (10 points)



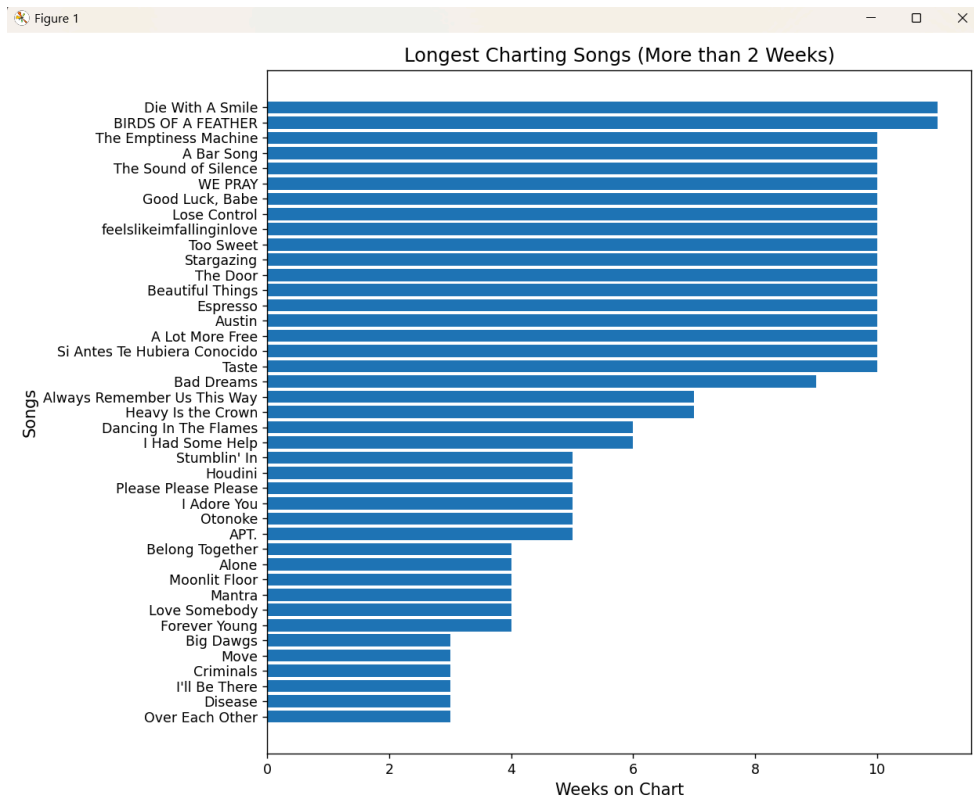
A screenshot of a terminal window with a dark background and light-colored text. The text is a list of 17 items, each consisting of a number followed by a nationality code and a value in weeks. The list is titled 'Average Weeks on Chart by Nationality:'. The items are: 1 IE: 5.50 weeks, 2 JP: 4.50 weeks, 3 CO: 4.00 weeks, 4 US: 3.62 weeks, 5 FR: 3.00 weeks, 6 IN: 3.00 weeks, 7 GB: 2.69 weeks, 8 DE: 2.00 weeks, 9 CA: 1.71 weeks, 10 KR: 1.53 weeks, 11 SE: 1.00 weeks, 12 IT: 1.00 weeks, 13 NO: 1.00 weeks, 14 AU: 1.00 weeks, 15 BE: 1.00 weeks, 16 NL: 1.00 weeks, and 17. The list is numbered 1 through 17 on the left side of the terminal.

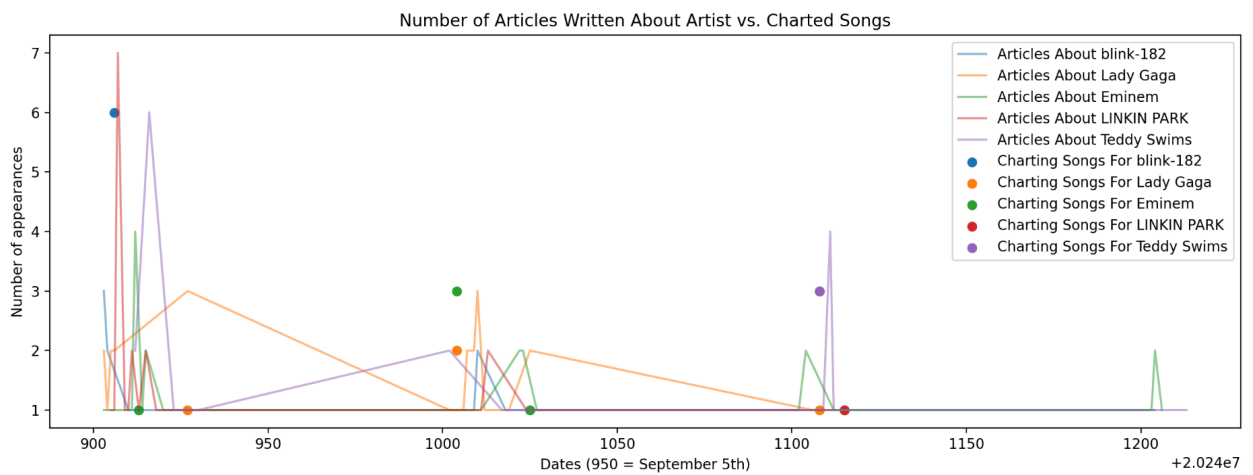
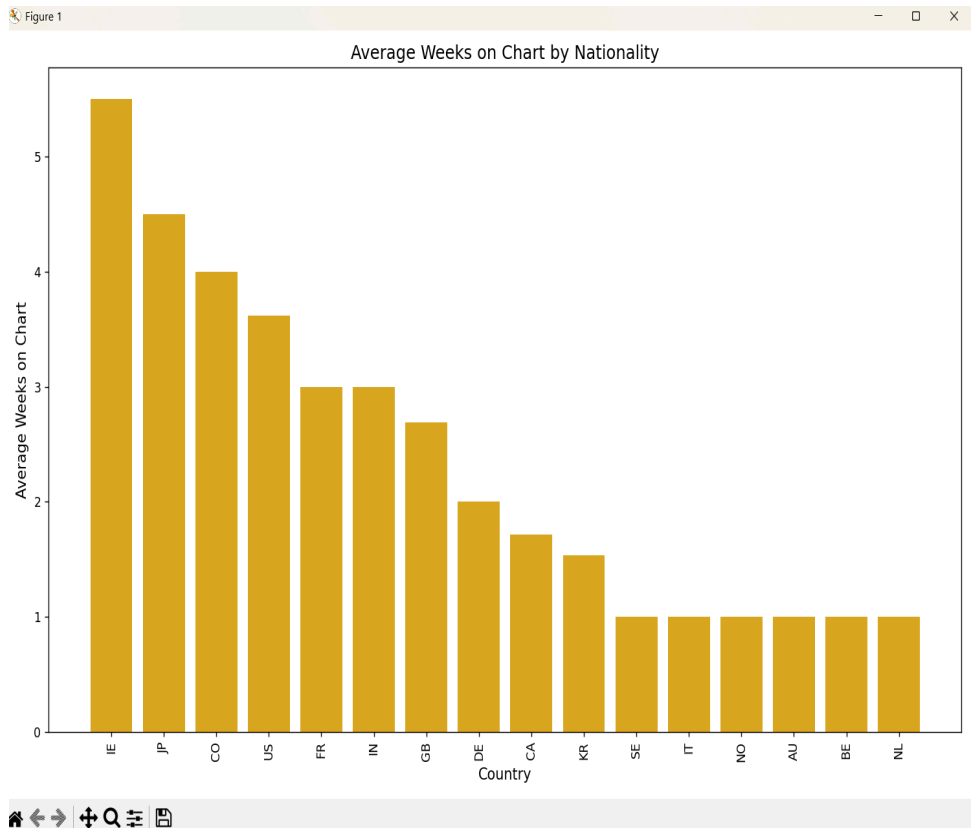
Rank	Nationality	Average Weeks on Chart
1	IE	5.50 weeks
2	JP	4.50 weeks
3	CO	4.00 weeks
4	US	3.62 weeks
5	FR	3.00 weeks
6	IN	3.00 weeks
7	GB	2.69 weeks
8	DE	2.00 weeks
9	CA	1.71 weeks
10	KR	1.53 weeks
11	SE	1.00 weeks
12	IT	1.00 weeks
13	NO	1.00 weeks
14	AU	1.00 weeks
15	BE	1.00 weeks
16	NL	1.00 weeks
17		

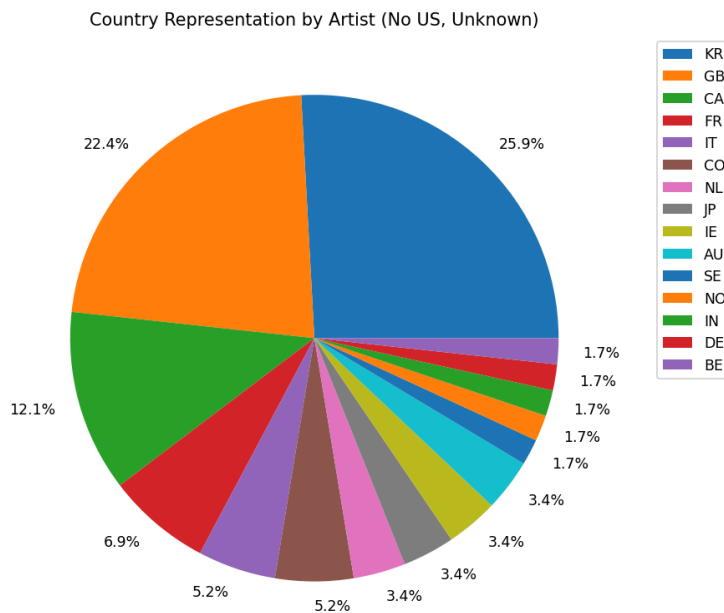
```
1 Average # of Articles Published per Month for Top 5 US Artists
2
3 Artist: blink-182
4
5 | | | Average # of Articles Published in September: 1.75
6 | | | Average # of Articles Published in October: 2.0
7 | | | Average # of Articles Published in November: 1.0
8 | | | Average # of Articles Published in December: 0.25
9
10 Artist: Lady Gaga
11
12 | | | Average # of Articles Published in September: 3.0
13 | | | Average # of Articles Published in October: 4.5
14 | | | Average # of Articles Published in November: 1.0
15 | | | Average # of Articles Published in December: 1.25
16
17 Artist: Eminem
18
19 | | | Average # of Articles Published in September: 5.25
20 | | | Average # of Articles Published in October: 5.0
21 | | | Average # of Articles Published in November: 1.75
22 | | | Average # of Articles Published in December: 1.25
23
24 Artist: LINKIN PARK
25
26 | | | Average # of Articles Published in September: 8.0
27 | | | Average # of Articles Published in October: 5.75
28 | | | Average # of Articles Published in November: 2.75
29 | | | Average # of Articles Published in December: 1.75
30
31 Artist: Teddy Swims
32
33 | | | Average # of Articles Published in September: 8.5
34 | | | Average # of Articles Published in October: 5.75
35 | | | Average # of Articles Published in November: 3.25
36 | | | Average # of Articles Published in December: 2.25
```

```
1 Top Artists by Country
2
3 Top Artists from US
4 1: blink-182, number of songs on charts: 6
5 2: Lady Gaga, number of songs on charts: 5
6 3: Eminem, number of songs on charts: 5
7 4: LINKIN PARK, number of songs on charts: 4
8 5: Teddy Swims, number of songs on charts: 3
9 6: Sabrina Carpenter, number of songs on charts: 3
10 7: Megan Thee Stallion, number of songs on charts: 2
11 8: Max McNow, number of songs on charts: 2
12 9: Chappell Roan, number of songs on charts: 2
13 10: Billie Eilish, number of songs on charts: 2
14
15 Top Artists from KR
16 1: j-hope, number of songs on charts: 1
17 2: YUTA, number of songs on charts: 1
18 3: YEONJUN, number of songs on charts: 1
19 4: XG, number of songs on charts: 1
20 5: TZUYU, number of songs on charts: 1
21 6: Stray Kids, number of songs on charts: 1
22 7: ROSÉ, number of songs on charts: 1
23 8: MEOVV, number of songs on charts: 1
24 9: Jimin, number of songs on charts: 1
25 10: JENNIE, number of songs on charts: 1
26
27 Top Artists from GB
28 1: Coldplay, number of songs on charts: 4
29 2: The Cure, number of songs on charts: 2
30 3: Tears for Fears, number of songs on charts: 2
31 4: Perrie, number of songs on charts: 2
32 5: Duran Duran, number of songs on charts: 2
33 6: Sophie Ellis-Bextor, number of songs on charts: 1
34
35 Top Artists from IE
36 1: U2, number of songs on charts: 1
```

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







6. Instructions for running your code (10 points)

Populating the main database

- Run the KWORB file 7 times to create and populate the songs table with enough data for the artist and news scraping tables, there will be 175 rows in songs
- run the MusicBrainz file 4 times
- Run the News_Scraping file 5 times

Visualizations:

- Run Vis1, Vis2, Vis3, and Vis4

Calculations:

- Run calculation_1, calculation_2, and calculation_3

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

DATA GATHERING

Kwordb scraping:

- Setup: imports, ensuring formatting works with utf-8, creating a list of dates (Fridays from last 3 months) to collect data from, connecting to main database

- Creating artist and song tables, making sure they both create unique IDs and then describing their columns
- Iterating through each date (by setting up URL with dates from that list we made earlier) and using BSoup to grab first 40 columns on the Kword page, finding the first artist name + the song name
- If artist & song combo already in the table, add 1 to counter of how many weeks that song has been on charts
 - Else, add song info to table, if artist already exists in artist table, use that artist id, otherwise give artist new row in artist table and assign that artist id to the song in song table
- Once counter hits 25 new songs, add them all to the database at once
- Close connection to the database

MusicBrainz_Data.py:

- def db_setup(db_name)
 - Sets up a cursor and connection to the main database
- def create_countries_table(cur, conn)
 - Creates a countries and country_to_artist tables for necessary data and future id assignments
 - Makes 25 requests at a time to MusicBrainz API for artist information using the artist names in the artists table
 - Sleeps for 1.1 seconds after every request to abide by the rate limit
- def assign_country_ids(cur, conn)
 - Adds the country_id column to the artists table and uses the country
- def main()
 - Calls the previous two functions

News_Scraping.py:

- def set_up_database(db_name)
 - Sets up a cursor and connection to the main database
- def create_news_dict(artist_name)
 - Pulls the most popular 25 articles about a given music artist from Medistack, then puts it into a dictionary
- def create_news_database(dict, curr, conn)
 - Given a dictionary of news article about a certain artist, a cursor, and connection, this function will add the name, description, and date of an article to the database

- def main()
 - Gets the top 5 most popular artists from the US based off of the amount of times their songs appear in the charts
 - Creates a News table to store article information in
 - Keeps track of the number of rows in the News table so only 25 articles are added at a time / changes which artist news is being added

VISUALIZATIONS

Vis1.py:

- This file grabs all the songs + how many weeks they charted for
- It then filters the songs by whether or not they were on the chart for more than a week (to ensure that they were truly popular songs)
- The visualization shows how long each song was charting for over the course of the last 3 months

Vis2.py:

- This bar graph is essentially a visual representation of calculation 1
- We group all artists for each country and take an average of how long their songs chart for
- We then plot the countries in descending order of how long the average song made by someone in that country has stayed in the top 40 songs for (over the last 3 months)

Vis3.py:

- Gathers the name, artist id, and # of songs for the top 5 most popular artists in the US by joining the artists and songs tables
- Find the dates of an artists' charting songs from the songs table
- Adds it to a dictionary with the artist name as the primary key, date as inner key, and # of times a song appeared on that date as value
- Next, creates a similar dict but instead of # of times a song appears on a certain date, it's the # of times a news article about a certain artist is published on a certain date (gets article dates from the News table)
- Creates a scatter plot for the # of times an artists' songs charted
- Creates a line plot superimposed on top of the scatter plot for the # of times a news article was published about an artist on a certain date

Vis4.py:

- Select country names and the aggregate counts of the songs released by artists from each one.

- Splits the fetched information into two lists for labels and counts to calculate percentage sizes, we dropped the US and Unknown to get a better idea of the total breakdown amongst the countries with lower count numbers, and displays it in the form of a pie chart.
8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	Item Description	Location of Resource	Result (did it solve the issue?)
11/23	AudioDB	https://www.theaudiodb.com/free_music_api	AudioDB did not contain complete data of all songs, making it unusable for our idea
11/23	Historic Top40 Weekly Charts Web Scrape	https://top40weekly.com/	No, found out that any non-current data was paywalled
11/23	Genius Lyrics	https://docs.genius.com/	Did not have genre data as we had hoped
11/24	Wikipedia	https://en.wikipedia.org/wiki/List_of_recessions_in_the_United_States	Ended up not going with recession pop idea, so we did not use Wikipedia web scraping
12/3	SerpAPI	https://serpapi.com/google-trends-api	Considered using, but News API fit better into our needs for our project
12/3	Mediastack	https://mediastack.com/documentation	Yes! But it was difficult to use at times due to being limited to 100 pulls a month
12/12	Open Meteo	https://open-meteo.com/en/docs/historical-weather-api	Weather API, but idea ended up not using due to not having adequate data
12/14	Kwordb	https://kwordb.net/ww/archive/	Yes! Free, easily scrapeable version of

			billboard
12/14	MusicBrainz	https://musicbrainz.org/doc/MusicBrainz_API	Yes! Used to get artists and their home countries