DATA VISUALISATION: AUDIO FEATURES IN SONGS

Contents:

PROJECT PROPOSAL

Do the Audio Features of Top Songs by Over the Past Decade Exhibit Consistent Patterns?

Abstract

This project aims to analyse the audio features of the most popular songs globally from the last decade, to investigate whether there is a correlation between audio features and the songs' popularity. While there may be some commonalities in the audio features of these songs, it is unlikely that they will all sound the same. Different artists have unique styles and preferences, which are likely reflected in the audio features of their songs. However, it is possible to identify whether certain trends or characteristics contribute to the popularity of these songs.

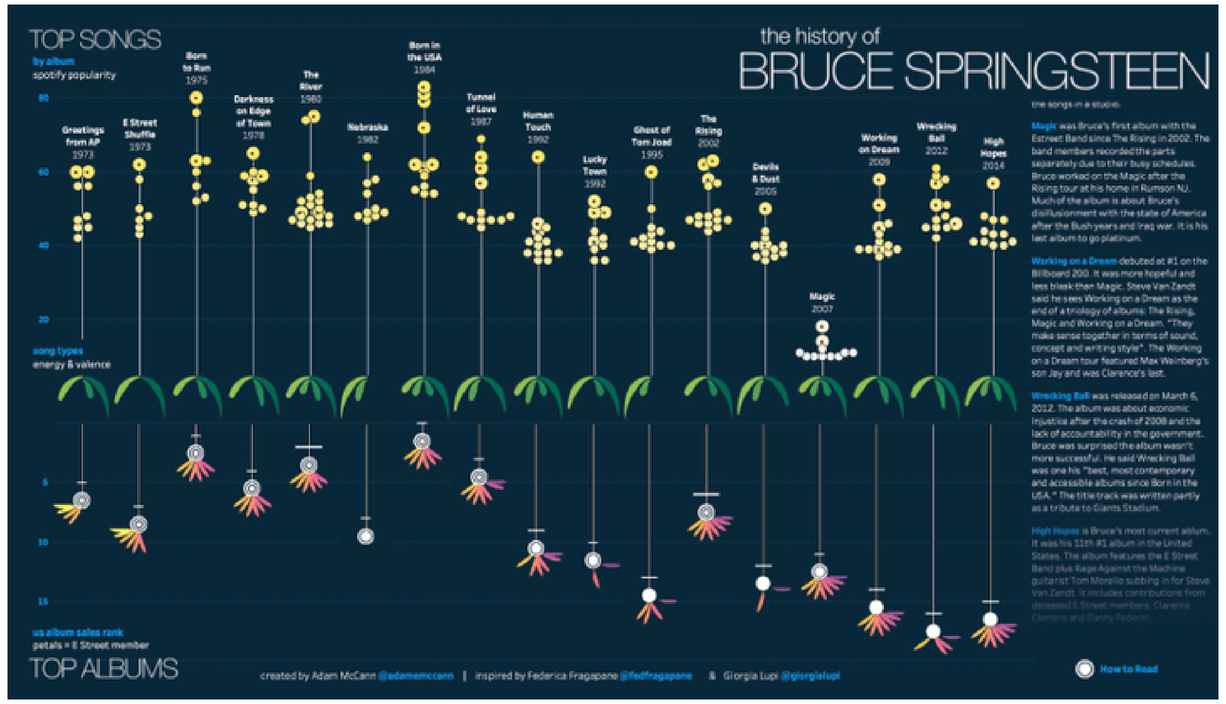
Methodology

- Data Collection: I will utilise the Spotify API to gather information on the top 10 streamed songs globally and extract the audio features for each song, such as danceability, energy, tempo, etc. These features are presented as numerical values.
- Data Analysis: Once the data is collected, I will analyse the audio features to identify any patterns or trends that may exist among the songs. I may use statistical techniques to calculate the average values for each feature, compare them across different artists, and look for correlations between these features and the songs' popularity.
- Visualisation: Create visualisations using p5.js to resemble music-themed equipment. Design different graphs and interactive buttons o allow users to explore and compare the audio features of the songs. These visualisations will allow for easy interpretation of the data.

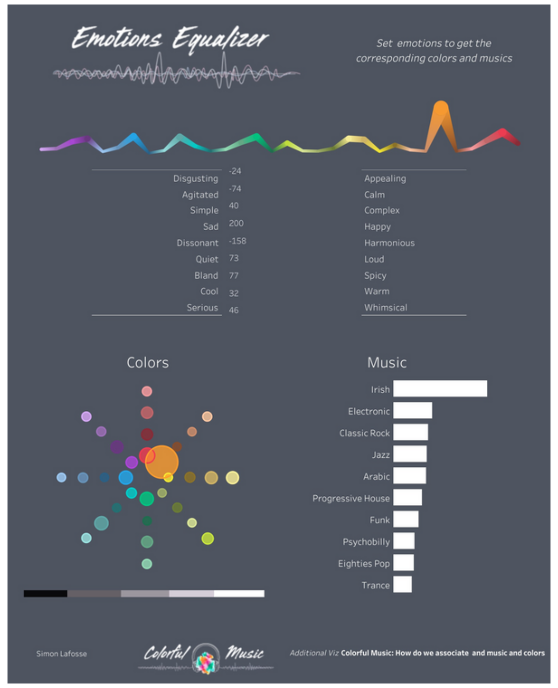
Expected Results

- Based on the analysis and visualisation of the audio features, I expect to see variations among the songs while still identifying some commonalities. For example, I have hypothesised that popular songs will have higher danceability and energy scores.

INSPIRATION

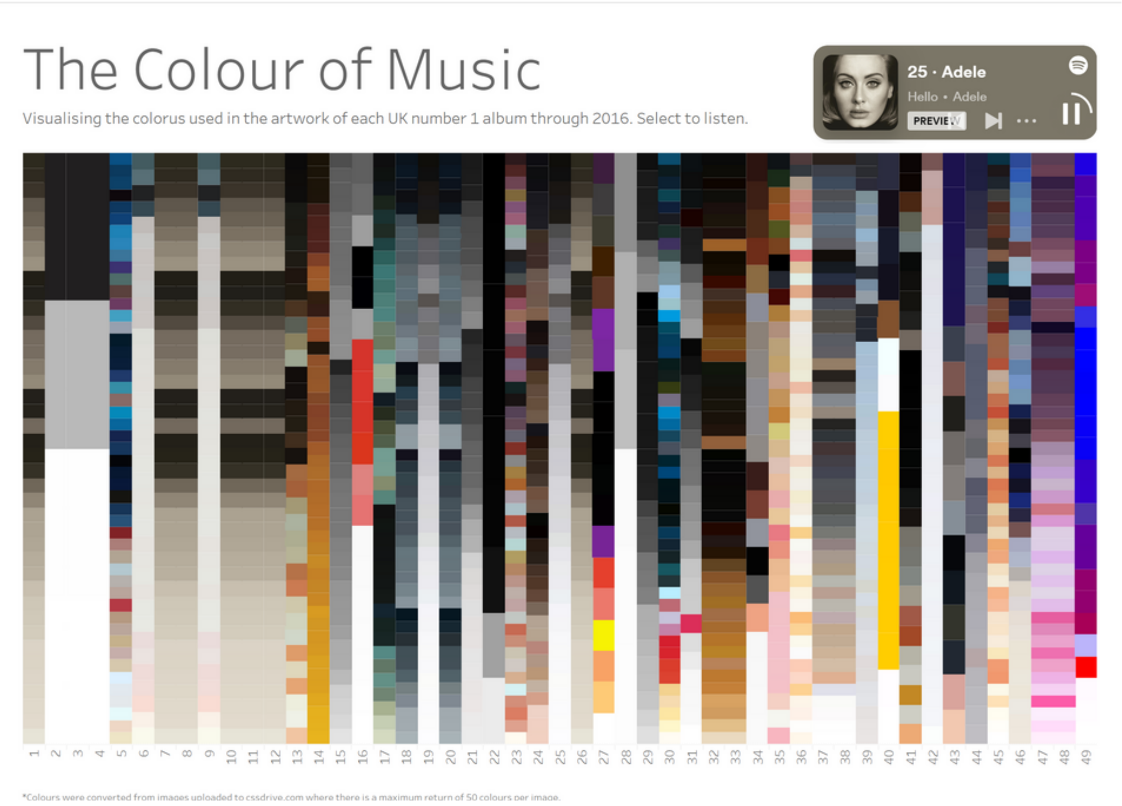


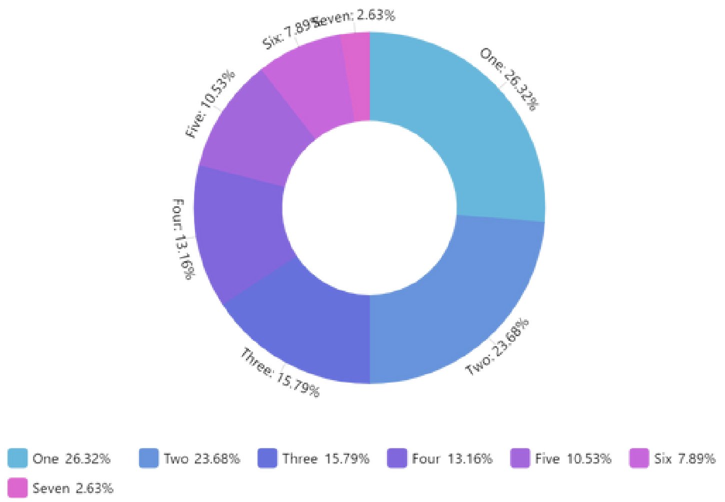
I began looking at music inspired data visualisations and came across this graph which illustrated Bruce Springsteen's top songs. I liked how clearly it was laid out and that the chart allowed users to go through each date.



I started to think about how I could present my data in a music-themed way and started taking inspiration from musical equipment such as equalisers and synthesisers. The visualisations shown above allow the users to play around with different audio features to create different outcomes. I liked the idea of an interactive visualisation as it keeps the user engaged. I thought an equaliser would be an efficient and creative way to display data about music as it could represent many bar charts and holds many buttons which allow for changes in the data being displayed. Additionally, I thought a doughnut chart resembles a vinyl or CD which could be another way to display data creatively.

I also liked this colour wall showing the top albums each week of 2016. When you click on a bar, the track appears in the right hand corner, making the data easy to interpret.





RESEARCH CONTEXTS

Spotify API

The audio features component of the Spotify API service allows users to retrieve various characteristics for each song. These characteristics provide insights into attributes like the tempo or 'loudness' of a track. Spotify API provides a list of audio features, including their corresponding data type and definition, as shown in the table. By utilising this feature, users can gain valuable information about the songs in their library or explore new music based on specific audio attributes (Al-Beitawi, Z., Salehan, M., & Zhang, S. (2020)).

A study in 2018 was conducted to analyse the trends of top songs on Spotify using the audio feature tool on the Spotify API. They used the playlist "Top 100 Trending Spotify Songs" from the years 2017 and 2018. These datasets consisted of the top 100 most-streamed tracks on Spotify for each year, including track ID, song name, artist name, and Spotify audio features. Although the dataset was limited to 200 records, it provided good variability in terms of artists and genres, encompassing over five genres.

The results from the study indicated that the musical characteristics of the most popular songs remained consistent across the two years. The analysis revealed that trending songs in both years exhibited low 'speechiness' (below 0.33), low 'liveness' (below 0.80), and low 'instrumentalness' (average of 0.00). This implies that songs that top the charts typically do not contain spoken words or vocal content, and they are not commonly recorded live in concert settings. Moreover, the analysis showed a consistent trend of high 'danceability' and high energy in the trending songs throughout the two years.

The analysis of the top 100 trending Spotify songs from 2017 and 2018 revealed interesting patterns and trends in their musical attributes. The clustering analysis identified four distinct clusters based on various musical characteristics:

- Cluster #1, the largest cluster, consisted of upbeat and joyful songs with high danceability, high loudness, low instrumentalness, high valence, and low tempo. These tracks were energetic and easy to listen to, driven by relaxing beats and rhythms.
-
- Cluster #2 shared high danceability with Cluster #1 but had lower energy and loudness. It emphasised acoustic instrumentals and had a laidback, rhythmic structure. This cluster represented mellow tracks that still retained a danceable quality.
-
- Cluster #4 encompassed a mix of rap, pop, and dance songs. These tracks had high tempo, high speechiness, and low instrumentalness. They aimed for immediate impact, relying on simple structures and memorable lyrics/hooks.

The analysis revealed that the Pop and Dance genres dominated the top 100 trending songs, with 71 out of 100 songs falling into these categories. Pop and Dance tracks exhibited high loudness and low speechiness, indicating a successful musical structure for achieving chart-topping status.

These findings provide valuable insights into the musical attributes that contribute to the popularity and success of trending songs. Understanding these patterns can guide musicians, producers, and marketers in creating music that resonates with listeners and has the potential to top the chart (Al-Beitawi, Z., Salehan, M., & Zhang, S. (2020)).

SPOTIFY AUDIO FEATURES

Attribute	Data Type	Definition
Key	integer	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.
Mode	integer	The modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Time_signature	integer	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
Danceability	float	Describes how suitable a track is for dancing based on a combination of musical elements, including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable, and 1.0 is most danceable."
Energy	float	A measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Loudness	float	An attribute of auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud.
Speechiness	float	Detects the presence of spoken words in a track." If the speechiness of a song is above 0.66, it is probably made of spoken words, a score between 0.33 and 0.66 is a song that may contain both music and words (e.g. rap music), and a score below 0.33 means the song does not have any speech.
Acousticness	float	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Instrumentalness	float	Represents the number of vocals in the song. The closer it is to 1.0, the greater likelihood the song contains no vocal content.
Liveness	float	Describes the probability that the song was recorded with a live audience. A value above 0.8 provides a strong likelihood that the track is live.
Valence	float	Describes the musical positiveness conveyed by a track, with a measure from 0.0 to 1.0. Tracks with high valence sound more positive (e.g., happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g., sad, depressed, angry).
Tempo	float	Describes the timing of the music or the speed at which a piece of music is played.
Duration_ms	integer	The duration of the track in milliseconds.

RESEARCH CONTEXTS

Factors To Make A Song Popular

"The Production of Success: An Anti-Musicology of the Pop Song", an article written by Antoine Hennion was published in Popular Music in 1983. The article focuses on the production process of pop songs and the challenges of traditional musicological approaches to understanding their success. Hennion presents an alternative perspective on analysing the success of pop songs, departing from the conventional musicological methods that focus on the innate qualities of the music itself. Instead, he examines the social and cultural factors involved in the production and reception of pop songs.

The author argues that success in the pop music industry is not solely determined by the qualities of the music, but rather by a complex set of factors related to production, marketing, and audience reception. He emphasises the role of producers in shaping the final product and shaping its appeal to the intended audience. Hennion also explores the collaborative nature of pop music production, highlighting the interactions between producers, musicians, songwriters, and performers. He emphasises the importance of understanding the social dynamics and power relations within these collaborations in order to understand the creative process and the resulting success or failure of a pop song.

The article also dives into the role of media and marketing in the success of pop songs. Hennion argues that the promotional strategies used by record labels, such as radio airplay, music videos, and advertising, play a crucial role in shaping the perception and reception of a song by the audience. Furthermore, Hennion highlights the active role of the audience in determining the success of a pop song. He suggests that the audience's preferences, tastes, and social context greatly influence the reception and popularity of a song.

Overall, Hennion challenges the traditional musicological approaches that focus solely on the musical qualities of pop songs. He argues for an interdisciplinary perspective that takes into account the social, cultural, and commercial aspects of pop music production and reception. By doing so, he aims to provide a more comprehensive understanding of the factors that contribute to the success of pop songs (Hennion, A. (1983)).

In a study in 2018, researchers analyzed over 500,000 songs released in the UK between 1985 and 2015 to understand the dynamics of music success (Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., & Komarova, N. L. (2018)). They explored the relationship between musical attributes and chart performance, uncovering trends in successful song. To predict song success, the researchers employed random forest models and considered both acoustic features and the "superstar" status of the artist. The analysis contributed to understanding the contribution of musical characteristics to success and the timescale of fashion dynamics in popular music. Data collection involved gathering information from sources such as trade magazine charts, online databases, and crowd-sourced data. The Official Singles Chart and databases like MusicBrainz and AcousticBrainz provided key data and metadata for the study.

The most notable conclusions of this study were :

- Trends in Musical Characteristics: The study found that certain musical features exhibited temporal trends over several decades. The analysis revealed a decline in happiness and brightness, as well as a slight increase in sadness, consistent with previous research on song lyrics. The study also identified a rise in danceability and relaxedness, possibly indicating the growing popularity of dance-based pop music compared to rock-type songs. Additionally, the study observed a decrease in the frequency of male singers in popular music over the last 30 years.

- Comparison of Average and Successful Songs: Successful songs, defined as those appearing in the charts, were found to be significantly different from the majority of songs. Successful songs tended to be happier, brighter, more party-like, more danceable, and less sad than most songs. However, despite these general trends, predicting the success of a specific song based solely on its acoustic features proved challenging.

- Prediction Accuracy: Using only acoustic features, the study achieved a prediction accuracy of 0.74, indicating the presence of missing information. The analysis showed that incorporating non-acoustic features, such as the "superstar" status of an artist, improved the prediction accuracy to 0.86.

- Importance of Musical Characteristics: The study highlighted the significance of musical characteristics in predicting song success. Although the inclusion of non-acoustic factors improved prediction accuracy, the ability to predict success based solely on acoustic features demonstrates the importance of musical characteristics. The analysis quantified the extent of this importance in recent decades of popular music.

- Short-lived Musical Fashion: The study suggested that musical fashion tends to be relatively short-lived. The dynamics of fashion and trend changes in music are complex and challenging to explain. Factors such as the relationship between fashion and social identity, manifestation of the status hierarchy, competition among music suppliers, and the desire to stand out in a crowded market contribute to the fast-paced nature of fashion changes in music.

RESEARCH CONTEXTS

The paper titled "Predicting Song Popularity" by James Pham, Edric Kyauk, and Edwin Park explores the task of predicting song popularity using machine learning algorithms. The authors focus on the "Million Song Dataset", which contains audio features and metadata for approximately one million songs. They evaluate different classification and regression algorithms to predict song popularity and determine which features have the most predictive power.

The dataset used in the study is The Million Song Dataset, which provides audio features and metadata for one million songs. The authors extracted a subset of 10,000 tracks and removed any tracks missing the features they considered. The final dataset consisted of 2,717 tracks, with 90% used for training and 10% for testing. Regarding feature extraction, the authors describe two types of features used: baseline features and additional features. Baseline features include various audio and metadata attributes, while additional features capture non-linear relationships and interactions between existing features. They also employ a bag-of-words approach to incorporate string features such as song names, artist IDs, and genre terms.

The authors define song popularity based on the "song hotness" metric provided by The Echo Nest, a music musical intelligence and data platform. They classify songs as popular if their "song hotness" value is above a certain threshold, specifically the top 25% of the dataset. The methods section outlines the feature selection techniques employed. The authors used forward stepwise selection, backward stepwise selection, and l1 regularization to narrow down the features and find the most relevant ones.

For classification, the authors evaluated several algorithms: logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support vector machines (SVM) with both linear and radial basis function (RBF) kernels, and multilayer perceptron (MLP). They compared the models using metrics such as area under the ROC curve (AUC) and precision, recall, and F1 scores.

In the regression analysis, the authors noticed that classification approaches lose valuable information about the actual song popularity values. They applied multiple linear regression to predict the popularity values.

The experimental results section discusses the outcomes of the feature selection and classification models. The authors present figures showing the selected features and provide AUC values for each classification model. They also mention the precision, recall, and F1 scores for the SVM with a linear kernel.

In summary, the article by Pham, Kyauk, and Park (2016) demonstrated the importance of feature selection in improving model performance for predicting song popularity. The study recognized the existence of diverse models with different strengths, suggesting that song popularity can be measured and predicted through various methods rather than a singular, definitive approach.

The article titled "Evolution of Global Music Trends: An Exploratory and Predictive Approach Based on Spotify Data" explores the analysis of music listening behaviour over time using Spotify data. The authors address several limitations in existing studies and propose a framework to improve the understanding of music preferences, the impact of contextual factors, and the development of a predictive model.

The paper begins by highlighting the opportunities brought by the digital era, allowing people to listen to music anytime and anywhere through streaming services. However, existing studies have limitations in capturing the long-term evolution of music preferences, considering contextual factors, and developing predictive frameworks for global listening trends.

The authors aim to address these limitations and propose an approach to analyse changes in music preferences using time-series clustering. The study focuses on three features of the songs: danceability, positivity, and intensity, and considers contextual variables such as climatology and the incidence of the COVID-19 pandemic in each country. Furthermore, they develop a prediction model that combines song features and contextual factors to forecast the popularity of songs.

The research is significant for the digital entertainment industry, as it provides insights into changes in music consumption over time and helps understand the impact of policies and contextual scenarios. The prediction model can assist in decision-making for music releases and promotional campaigns. The related work section discusses previous studies in music streaming data analysis, including the identification of popular songs, analysis of listener behaviour, global music consumption patterns, and the impact of the COVID-19 pandemic. The authors highlight the novelty of their approach in terms of focusing on the evolution of music listening patterns, considering contextual factors, and developing a predictive model at a larger domain scale.

The analysis of music trends is based on data collected from Spotify's Top 200 daily rankings in 52 different countries over a four-year period. The authors extract mood-related features of songs, including danceability, valence, and energy, using the Spotify Developer Platform.

The paper reveals that countries can be grouped into three large clusters based on their music preferences, with some notable effects due to the pandemic. Northern European and North American countries, along with Australia, prefer songs with a slight level of positivity and high intensity, while danceability has been negatively affected by COVID-19. Latin American countries and countries with Latin origins in Europe prefer highly danceable and positive music. Asian countries, on the other hand, prefer songs with low danceability, slight positivity, and low intensity, with the impact of COVID-19 accentuating these preferences.

Additionally, the paper proposes a multivariate time series prediction model to forecast the most popular type of music in different clusters of countries based on previous consumption patterns. The results show that it is possible to predict the preferred type of music within a certain interval with a relatively low error rate. This prediction could be useful for record labels optimising release schedules and advertising companies selecting soundtracks for campaigns (Terroso-Saenz, F., Soto, J. and Muñoz, A., 2023).

SCRAPING DATA: ORIGINAL

I originally thought about looking at patterns of audio features of top global artists via Spotify API, to see if their features were similar.

```
In [20]:  import requests

In [21]:  # Step 1: Obtain your client ID and client secret from the Spotify Developer Dashboard
          client_id = "61fef63f6aa74663830e1a59207e4426"
          client_secret = "db43e1b52e294fc2a06b3807a35b3c34"

In [22]:  # Step 2: Obtain an access token using the Client Credentials Flow
          def get_access_token(client_id, client_secret):
              url = "https://accounts.spotify.com/api/token"
              headers = {"Content-Type": "application/x-www-form-urlencoded"}
              data = {"grant_type": "client_credentials"}
              auth = (client_id, client_secret)
              response = requests.post(url, headers=headers, data=data, auth=auth)
              response_data = response.json()
              access_token = response_data["access_token"]
              return access_token

In [23]:  access_token = get_access_token(client_id, client_secret)

In [24]:  # Step 3: Search for Beyoncé as an artist
          def search_artist(access_token, artist_name):
              url = f"https://api.spotify.com/v1/search?q={artist_name}&type=artist&limit=1"
              headers = {"Authorization": f"Bearer {access_token}"}
              response = requests.get(url, headers=headers)
              response_data = response.json()
              return response_data

In [25]:  artist_name = "Beyoncé"
          response_data = search_artist(access_token, artist_name)
```

I began scraping data using Spotify API to get data on Beyonce and her top tracks using the client keys and creating an access token.

```
In [8]:  def get_songs_by_artist(token, artist_id):
             url = f"https://api.spotify.com/v1/artists/{artist_id}/top-tracks?country=GB"
             headers = get_auth_header(token)
             result = get(url, headers=headers)
             json_result = json.loads(result.content)["tracks"]
             return json_result

In [9]:  token = get_token()
         result = search_for_artist(token, "Beyoncé")
         artist_id = result["id"]
         songs = get_songs_by_artist(token, artist_id)
         print(songs)
```


[{'album': {'album_group': 'album', 'album_type': 'album', 'artists': [{'external_urls': {'spotify': 'https://open.spotify.com/artist/6vWDO969PvNqNYHIOW5v0m'}, 'href': 'https://api.spotify.com/v1/artists/6vWDO969PvNqNYHIOW5v0m', 'id': '6vWDO969PvNqNYHIOW5v0m', 'name': 'Beyoncé', 'type': 'artist', 'uri': 'spotify:artist:6vWDO969PvNqNYHIOW5v0m'}], 'external_urls': {'spotify': 'https://open.spotify.com/album/6FJxoadUE4JNVwWHghBwnb'}, 'href': 'https://api.spotify.com/v1/albums/6FJxoadUE4JNVwWHghBwnb', 'id': '6FJxoadUE4JNVwWHghBwnb', 'images': [{'height': 640, 'url': 'https://i.scdn.co/image/ab67616d0000b2730e58a0f8308c1ad403d105e7', 'width': 640}, {'height': 300, 'url': 'https://i.scdn.co/image/ab67616d00001e020e58a0f8308c1ad403d105e7', 'width': 300}, {'height': 64, 'url': 'https://i.scdn.co/image/ab67616d000048510e58a0f8308c1ad403d105e7', 'width': 64}], 'is_playable': True, 'name': 'RENAISSANCE', 'release_date': '2022-07-29', 'release_date_precision': 'day', 'total_tracks': 16, 'type': 'album', 'uri': 'spotify:album:6FJxoadUE4JNVwWHghBwnb'}, 'artists': [{'external_urls': {'spotify': 'https://open.spotify.com/artist/6vWDO969PvNqNYHIOW5v0m'}, 'href': 'https://api.spotify.com/v1/artists/6vWDO969PvNqNYHIOW5v0m', 'id': '6vWDO969PvNqNYHIOW5v0m', 'name': 'Beyoncé', 'type': 'artist', 'uri': 'spotify:artist:6vWDO969PvNqNYHIOW5v0m'}], 'disc_number': 1, 'duration_ms': 225388, 'explicit': True, 'external_ids': {'isrc': 'USSM12206232'}, 'external_urls': {'spotify': 'https://open.spotify.com/track/1xzi1Jcr7mEi9K2RfzLOqS'}, 'href': 'https://api.spotify.com/v1/tracks/1xzi1Jcr7mEi9K2RfzLOqS', 'id': '1xzi1Jcr7mEi9K2RfzLOqS', 'is_local': False, 'is_playable': True, 'name': 'CUFF IT', 'popularity': 89, 'preview_url': 'https://p.scdn.co/mp3-preview/1799de9ff42644af9773b6353358e54615696f19?cid=61fef63f6aa74663830e1a59207e4426', 'track_number': 4, 'type': 'track', 'uri': 'spotify:track:1xzi1Jcr7mEi9K2RfzLOqS'}, {'album': {'album_group': 'album', 'album_type': 'album', 'artists': [{'external_urls': {'spotify': 'https://open.spotify.com/artist/6vWDO969PvNqNYHIOW5v0m'}, 'href': 'https://api.spotify.com/v1/artists/6vWDO969PvNqNYHIOW5v0m', 'id': '6vWDO969PvNqNYHIOW5v0m', 'name': 'Beyoncé', 'type': 'artist', 'uri': 'spotify:artist:6vWDO969PvNqNYHIOW5v0m'}], 'external_urls': {'spotify':

Once I had the data I began processing the data to print out my desired data.

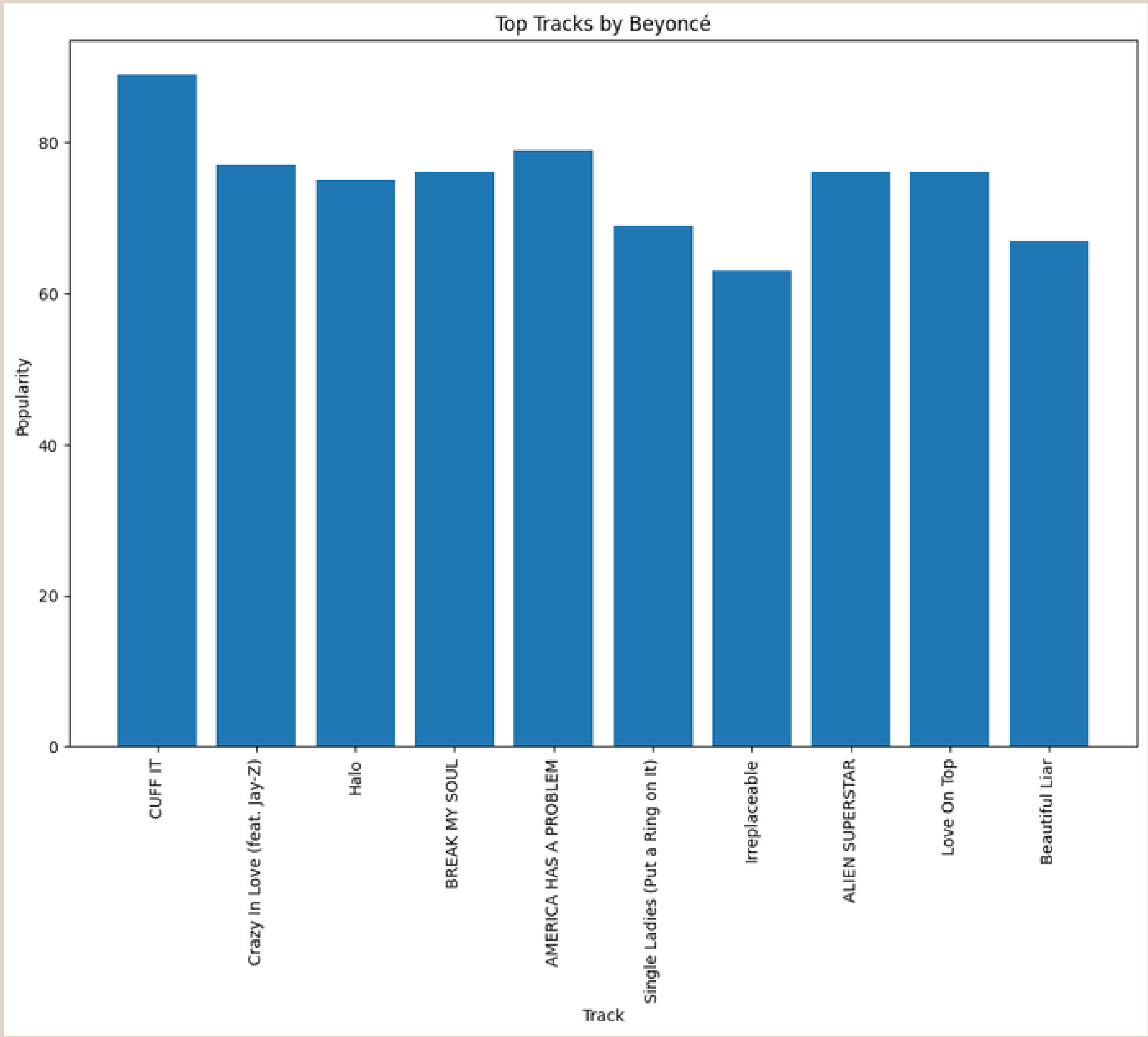
```
In [26]:  # Step 4: Process the response
          if "artists" in response_data and "items" in response_data["artists"]:
              artists = response_data["artists"]["items"]
              if len(artists) > 0:
                  beyonce = artists[0]
                  print("Beyoncé Information:")
                  print("Name:", beyonce["name"])
                  print("Spotify ID:", beyonce["id"])
                  print("Genres:", beyonce["genres"])
                  print("Images:")
                  for image in beyonce["images"]:
                      print(image["url"])
              else:
                  print("No artist found with the name 'Beyoncé'")
          else:
              print("Error occurred during the search")

          Beyoncé Information:
          Name: Beyoncé
          Spotify ID: 6vWDO969PvNqNYHIOW5v0m
          Genres: ['pop', 'r&b']
          Images:
          https://i.scdn.co/image/ab6761610000e5eb12e3f20d05a8d6cfde988715
          https://i.scdn.co/image/ab6761610000517412e3f20d05a8d6cfde988715
          https://i.scdn.co/image/ab6761610000f17812e3f20d05a8d6cfde988715
```

I cleaned the data and asked for it to print out the top 10 songs in order for me to start plotting the data.

```
In [10]:  for idx, song in enumerate(songs):
              print(f"{idx + 1}. {song['name']}")

          1. CUFF IT
          2. Crazy In Love (feat. Jay-Z)
          3. Halo
          4. BREAK MY SOUL
          5. AMERICA HAS A PROBLEM
          6. Single Ladies (Put a Ring on It)
          7. Irreplaceable
          8. ALIEN SUPERSTAR
          9. Love On Top
          10. Beautiful Liar
```

```
In [11]:  def plot_top_tracks(songs):
              track_names = []
              track_popularity = []

              for track in songs:
                  track_names.append(track["name"])
                  track_popularity.append(track["popularity"])

              plt.clf()  # Clear current figure
              plt.figure(figsize=(12, 8))  # Specify a larger figure size
              plt.bar(track_names, track_popularity)
              plt.xlabel("Track")
              plt.ylabel("Popularity")
              plt.title("Top Tracks by Beyoncé")
              plt.xticks(rotation=90)
              plt.show()

          #     # Save data as CSV
          #     data = {"Track": track_names, "Popularity": track_popularity}
          #     df = pd.DataFrame(data)
          #     df.to_csv("top_tracks.csv", index=False)


              plt.close()  # Close the plot
               # Save graph as PNG
              plt.savefig("C:/Users/shaan/Desktop/DataVis/top_tracks.png", dpi=300, bbox_inches="tight")
              plt.show()
```

Once the data was collected I used mat.plot.lib to create a bar graph that measures the popularity of Beyonce's top songs.

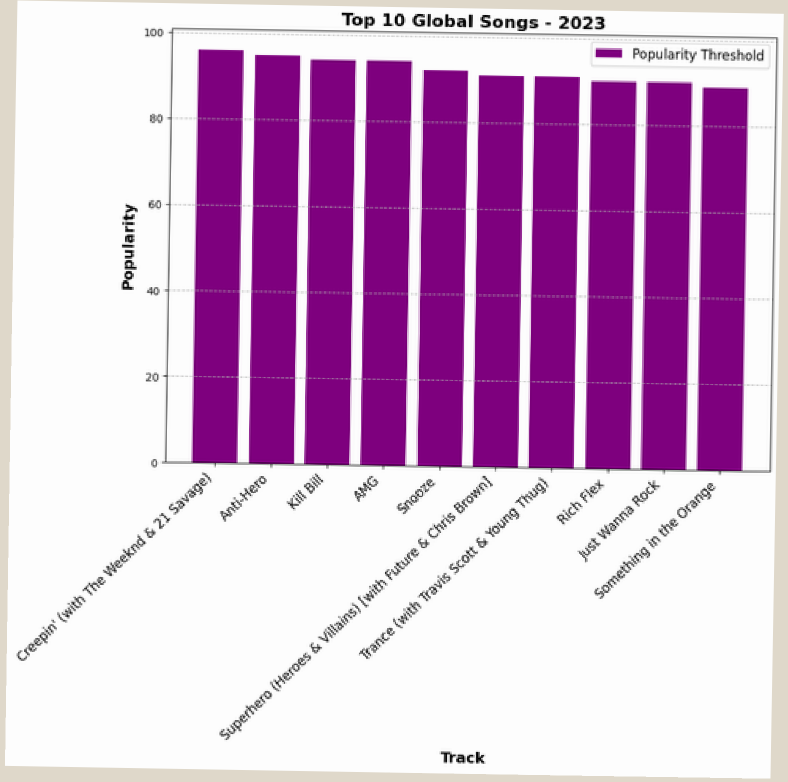


SCRAPING DATA: NEW

To ensure my original question was being answered, I realised I needed to change the data I was try to find and look at tracks over a certain number of years rather than different artists.

```python
playlist_id = '37i9dQZEVXbLnolsZ8PSNw'
playlist = sp.playlist(playlist_id)

# Extract track information
tracks = playlist['tracks']['items']
top_tracks_uk = []

for track in tracks:
    track_info = track['track']
    popularity = track_info['popularity']
    if popularity > 50:
        top_tracks_uk.append(track_info)

# Print the top tracks from the UK
for track in top_tracks_uk:
    name = track['name']
    artists = [artist['name'] for artist in track['artists']]
    print(f"Track: {name} - Artists: {', '.join(artists)}")
```

```
Track: Sprinter - Artists: Dave, Central Cee
Track: Who Told You (feat. Drake) - Artists: J Hus, Drake
Track: Giving Me - Artists: Jazzy
Track: As It Was - Artists: Harry Styles
Track: Miracle (with Ellie Goulding) - Artists: Calvin Harris, Ellie Goulding
Track: UK Rap - Artists: Dave, Central Cee
Track: Daylight - Artists: David Kushner
Track: Good Love - Artists: Hannah Laing, RoRo
Track: Flowers - Artists: Miley Cyrus
Track: Dancing is Healing - Artists: Rudimental, Charlotte Plank, Vibe Chemistry
Track: Trojan Horse - Artists: Dave, Central Cee
Track: REACT - Artists: Switch Disco, Ella Henderson, Robert Miles
Track: (It Goes Like) Nanana - Edit - Artists: Peggy Gou
Track: Late Night Talking - Artists: Harry Styles
Track: Satellite - Artists: Harry Styles
```



I started to gather data on the top 50 songs in the UK. This worked well, however, this data is static, I needed to see data over a period of time to be able to examine if there is a shift in different audio features.

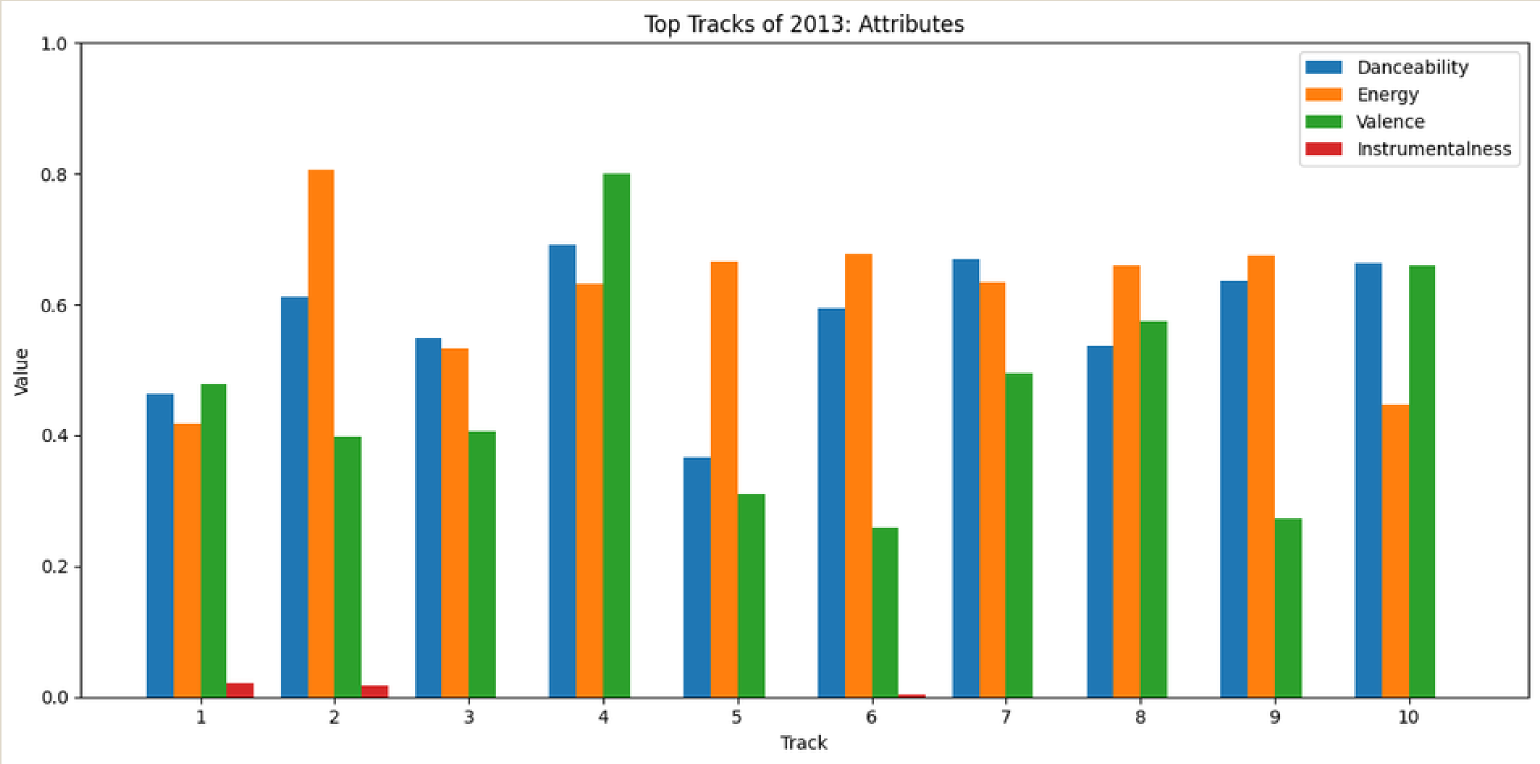
```python
# Define search parameters
search_query = 'year:2023'
limit = 10  # Number of tracks to retrieve

# Search for tracks based on the release date
results = sp.search(q=search_query, type='track', limit=limit)

# Filter and sort the tracks based on popularity
filtered_tracks = [track for track in results['tracks']['items'] if track['popularity'] > 70]
sorted_tracks = sorted(filtered_tracks, key=lambda x: x['popularity'], reverse=True)

# Print the top 10 global songs from the 2023
for i, track in enumerate(sorted_tracks[:10]):
    name = track['name']
    artists = [artist['name'] for artist in track['artists']]
    print(f"{i+1}. Track: {name} - Artists: {', '.join(artists)}")
```

```
1. Track: un x100to - Artists: Grupo Frontera, Bad Bunny
2. Track: Ella Baila Sola - Artists: Eslabon Armado, Peso Pluma
3. Track: La Bebe - Remix - Artists: Yng Lvcas, Peso Pluma
4. Track: Cupid - Twin Ver. - Artists: FIFTY FIFTY
5. Track: WHERE SHE GOES - Artists: Bad Bunny
6. Track: Boy's a Liar Pt. 2 - Artists: PinkPantheress, Ice Spice
7. Track: Search & Rescue - Artists: Drake
8. Track: Fast Car - Artists: Luke Combs
9. Track: Last Night - Artists: Morgan Wallen
10. Track: Last Night - Artists: Morgan Wallen
```

Instead of focusing on the current top 50 songs in the UK, I decided to gather data on a smaller set of globally popular songs from the past 10 years. I started by collecting information on the top 10 songs of 2023 so far. The graph above represents the variation of different audio features for these selected songs. By analysing attributes like danceability, energy and tempo mode, time signature, and duration, we can gain insights into the musical characteristics and trends of these popular tracks.

```python
# Define search parameters
start_year = 2013  # Adjust the start year as needed
end_year = 2022  # Adjust the end year as needed
limit = 10  # Number of tracks to retrieve for each year

# Iterate over each year and fetch the top tracks
for year in range(start_year, end_year + 1):
    search_query = f'year:{year}'
    results = sp.search(q=search_query, type='track', limit=limit)

    # Filter and sort the tracks based on popularity
    filtered_tracks = [track for track in results['tracks']['items'] if track['popularity'] > 70]
    sorted_tracks = sorted(filtered_tracks, key=lambda x: x['popularity'], reverse=True)

    # Print the top tracks for the year
    print(f"Top tracks of {year}:")
    for i, track in enumerate(sorted_tracks[:limit]):
        name = track['name']
        artists = [artist['name'] for artist in track['artists']]
        print(f"{i+1}. Track: {name} - Artists: {', '.join(artists)}")
    print()
```

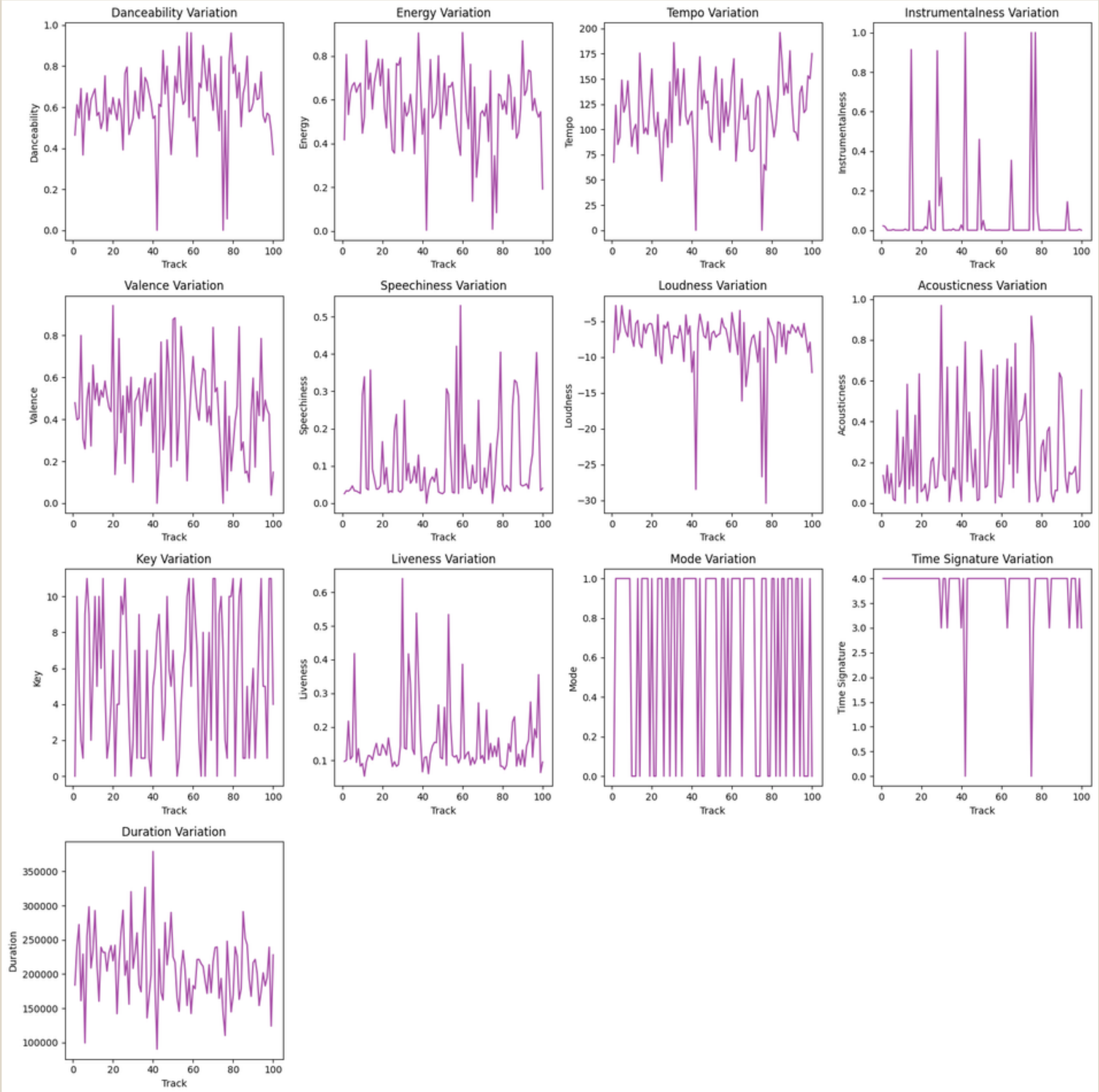
Once the data was collected I used mat.plot.lib to create a bar graph that measures a few attributes of the top songs of 2023. I continued to do this up to 2013 and save the data into a csv file.

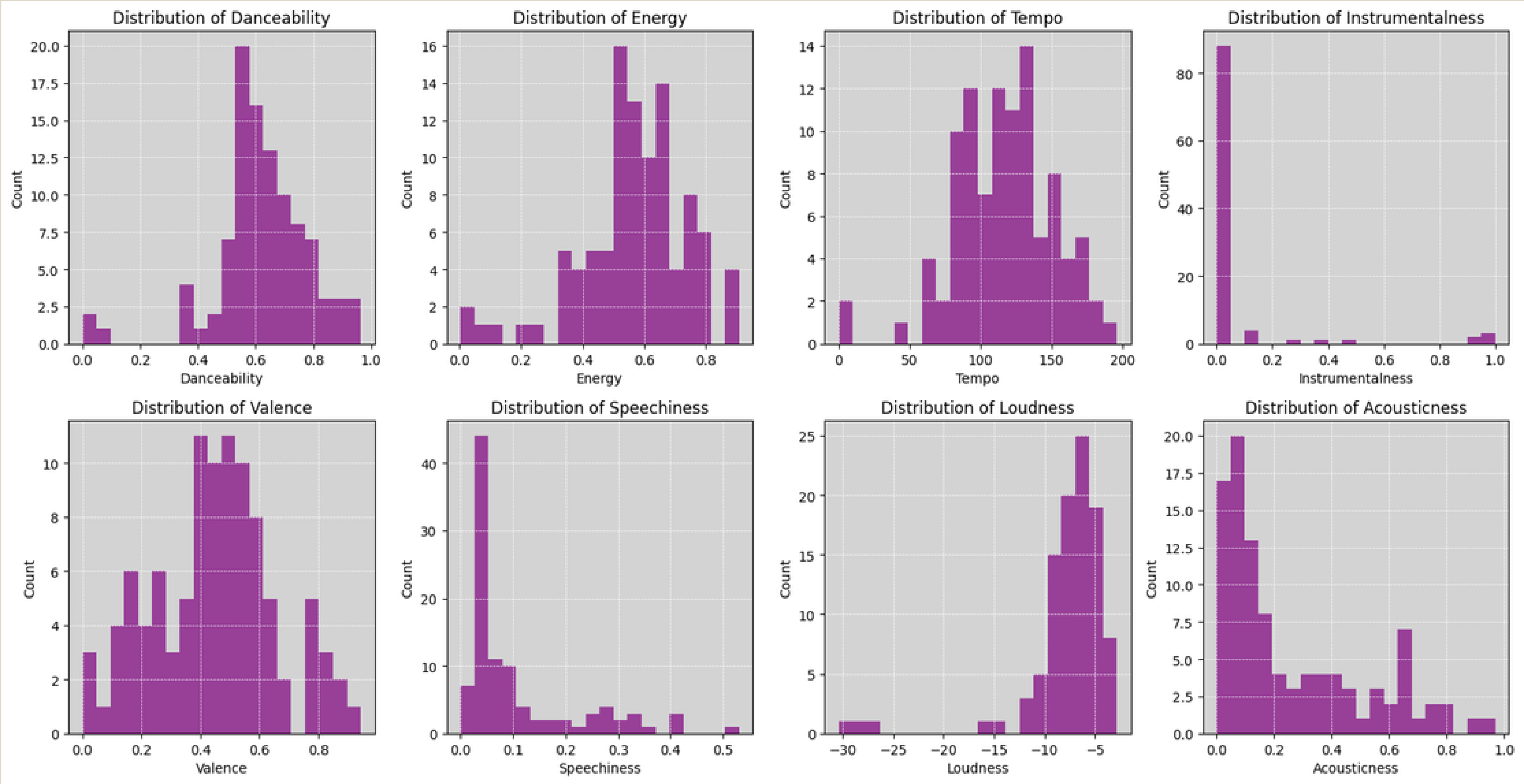


AUDIO FEATURES

```python
            print(f"Instrumentalness: {instrumentalness}")
            print(f"Valence: {valence}")
            print(f"Speechiness: {speechiness}")
            print(f"Loudness: {loudness}")
            print(f"Acousticness: {acousticness}")
            print(f"Key: {key}")
            print(f"Liveness: {liveness}")
            print(f"Mode: {mode}")
            print(f"Time Signature: {time_signature}")
            print(f"Duration: {duration}")
            print()

            # Write the track information and attributes to the CSV file
            writer.writerow({
                'Year': year,
                'Track': name,
                'Artists': ', '.join(artists),
                'Danceability': danceability,
                'Energy': energy,
                'Tempo': tempo,
                'Instrumentalness': instrumentalness,
                'Valence': valence,
                'Speechiness': speechiness,
                'Loudness': loudness,
                'Acousticness': acousticness,
                'Key': key,
                'Liveness': liveness,
                'Mode': mode,
                'Time Signature': time_signature,
                'Duration': duration
            })

    print("CSV file saved successfully.")
```

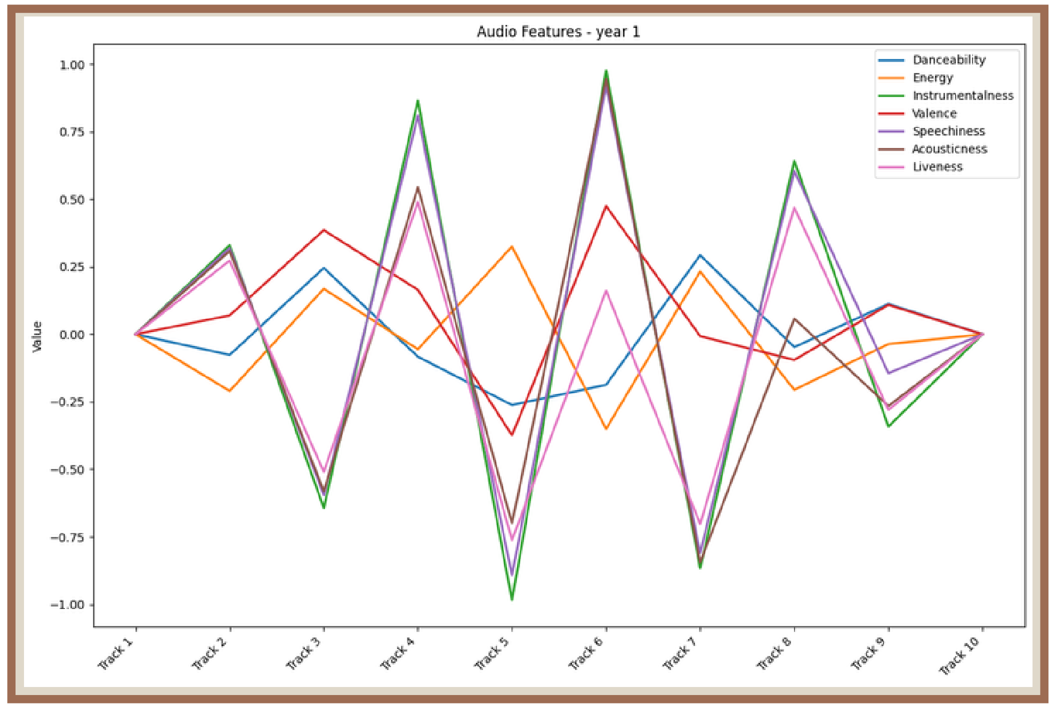
After obtaining the top 100 tracks from the past decade, I started to gather comprehensive information about various audio features available on Spotify. Initially, I wasn't certain which features would be the most relevant to examine. I hypothesised that higher danceability and energy levels were likely to be significant, given the prevalence of these characteristics in many popular songs. However, I also felt it would be valuable to compare all available features initially to identify any noticeable correlations and gain a broader understanding of the data.





In these graphs, I have included all the audio features that I was able to gather from the Spotify API. By exploring various graph designs, I aimed to draw inspiration for my final project. I particularly liked how these graphs resembled synthesiser waves, which added a musical touch to the visual representation. At this stage, my focus shifted towards determining the most crucial features to collect data from. I sought to identify which features would provide valuable insights for my analysis.

DESIGNS & EXPERIMENTS

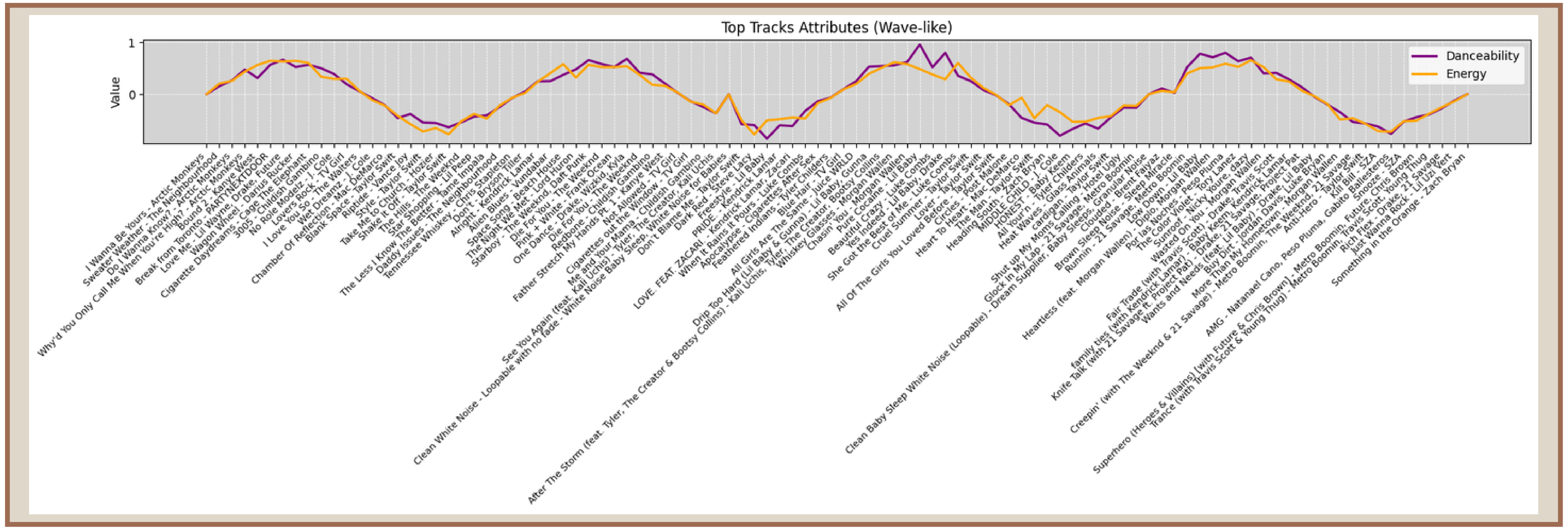


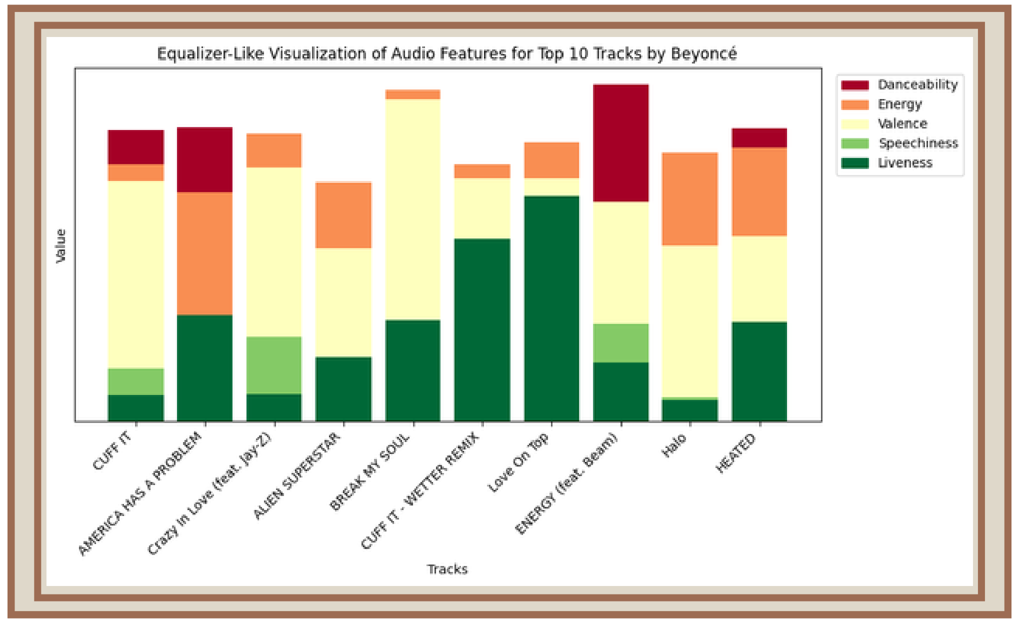

```
audio_features = {
    'Danceability': [],
    'Energy': [],
    'Instrumentalness': [],
    'Valence': [],
    'Speechiness': [],
    'Acousticness': [],
    'Liveness': [],
}

# Read the CSV file
with open('10_years_top_tracks_all.csv', 'r') as csvfile:
    reader = csv.DictReader(csvfile)

    # Iterate over each row and extract attribute values
    for row in reader:
        for feature in audio_features:
            audio_features[feature].append(float(row[feature]))

# Generate x-axis values for each group of tracks
tracks_per_graph = 10  # Number of tracks per graph
num_tracks = len(audio_features['Danceability'])
num_graphs = num_tracks // tracks_per_graph

# Plotting the graphs
for i in range(num_graphs):
    start_idx = i * tracks_per_graph
    end_idx = (i + 1) * tracks_per_graph

    # Create a new figure and axes for each graph
    fig, ax = plt.subplots(figsize=(12, 8))

    # Generate x-axis values (time)
    x = np.linspace(start_idx, end_idx, end_idx - start_idx)

    # Plot each audio feature as a synthesizer-style graph
    for feature in audio_features:
        # Scale the audio feature values to match the synthesizer range
        scaled_values = [val * 2 - 1 for val in audio_features[feature][start_idx:end_idx]]

        # Apply a wave-like function to the scaled values
        y = np.sin(np.pi * x) * scaled_values

        # Plot the feature as a line with varying color
        ax.plot(x, y, label=feature.capitalize(), linewidth=2)

    # Set x-axis labels
    ax.set_xticks(x)
    ax.set_xticklabels([f'Track {j+1}' for j in range(start_idx, end_idx)], rotation=45, ha="right")

    # Set y-axis label and title
```

Now that I had gathered enough data to create visualisations I wanted to experiment with the design of the graphs before moving on to p5.js to simplify the code when moving on to a different software.

I started plotting the graphs in different forms of waves as I initially wanted to create a graph that plotted data similar to sound waves. Here are some examples:

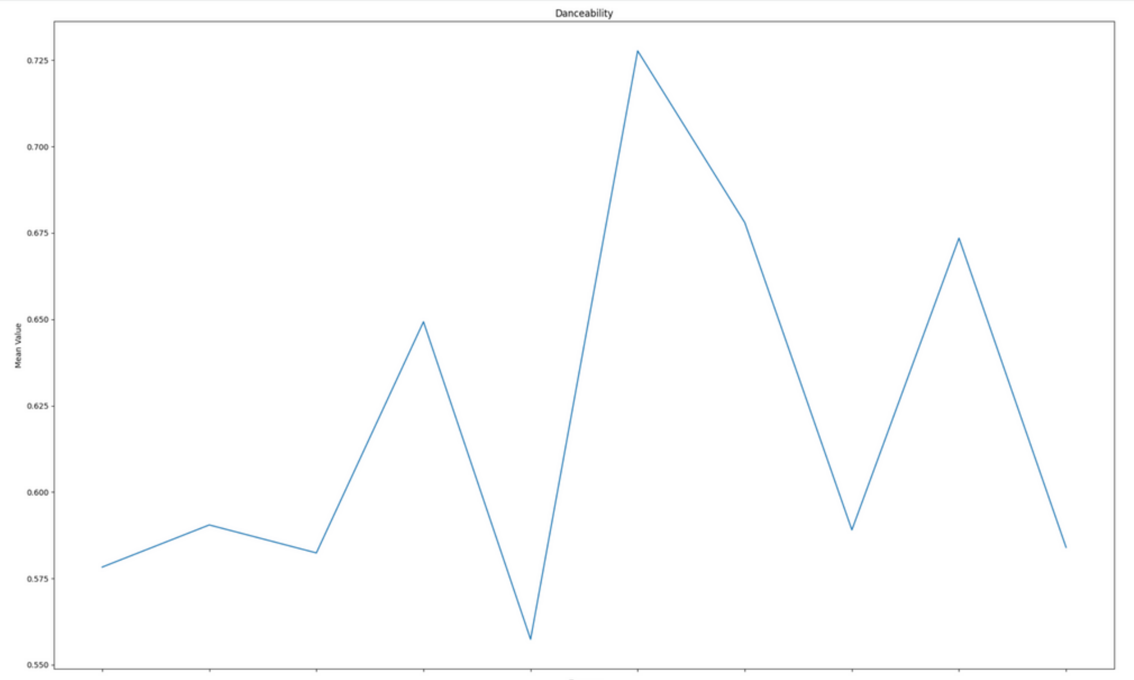
```
        # Append track attributes to the lists
        track_names.append(f"{name} - {', '.join(artists)}")
        danceability_values.append(danceability)
        energy_values.append(energy)

        print(f"Danceability: {danceability}")
        print(f"Energy: {energy}")
        print()

# Create an array of indices for the tracks
indices = np.arange(len(track_names))

# Generate the wave-like effect
x = np.linspace(0, 2 * np.pi, len(track_names))
wave = np.sin(4 * x)

# Create the figure and axes
fig, ax = plt.subplots(figsize=(18, 6))

# Plot danceability as a wave-like line
ax.plot(indices, danceability_values * wave, label='Danceability', color='purple', linewidth=2)

# Plot energy as a wave-like line
ax.plot(indices, energy_values * wave, label='Energy', color='orange', linewidth=2)

# Set the x-axis ticks and labels
ax.set_xticks(indices)
ax.set_xticklabels(track_names, rotation=45, ha='right', fontsize=8)

# Set the y-axis label
ax.set_ylabel('Value')

# Set the title and legend
ax.set_title('Top Tracks Attributes (Wave-like)')
ax.legend()

# Add musical-themed elements
ax.set_facecolor('lightgray')
ax.grid(color='white', linestyle='--', linewidth=0.5)

# Display the graph
plt.tight_layout()
plt.show()
```

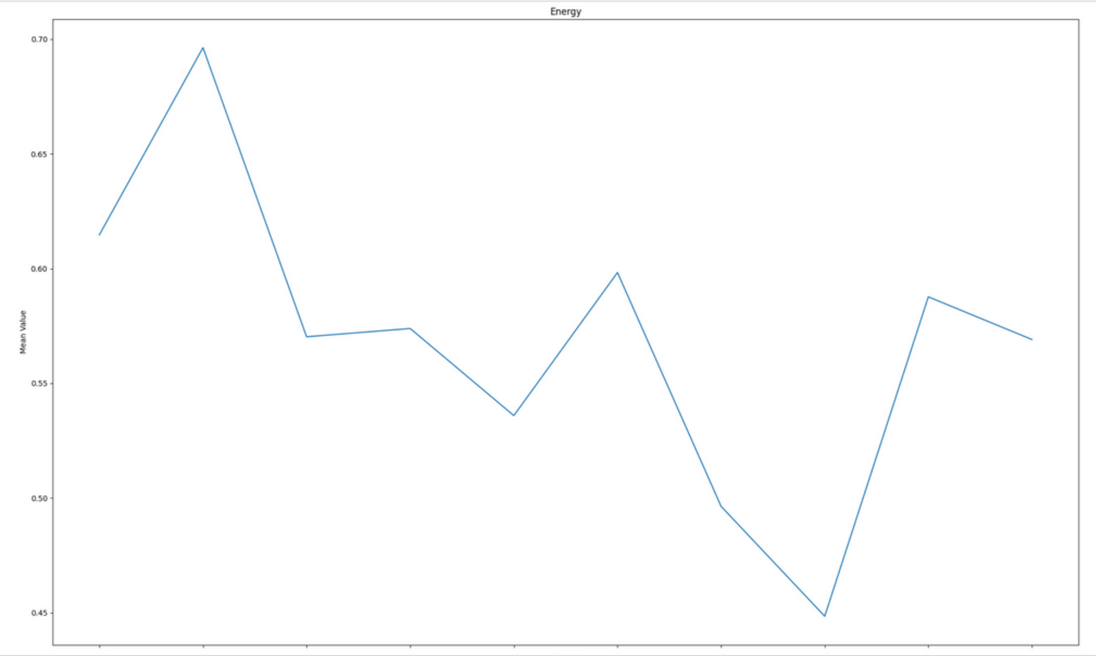


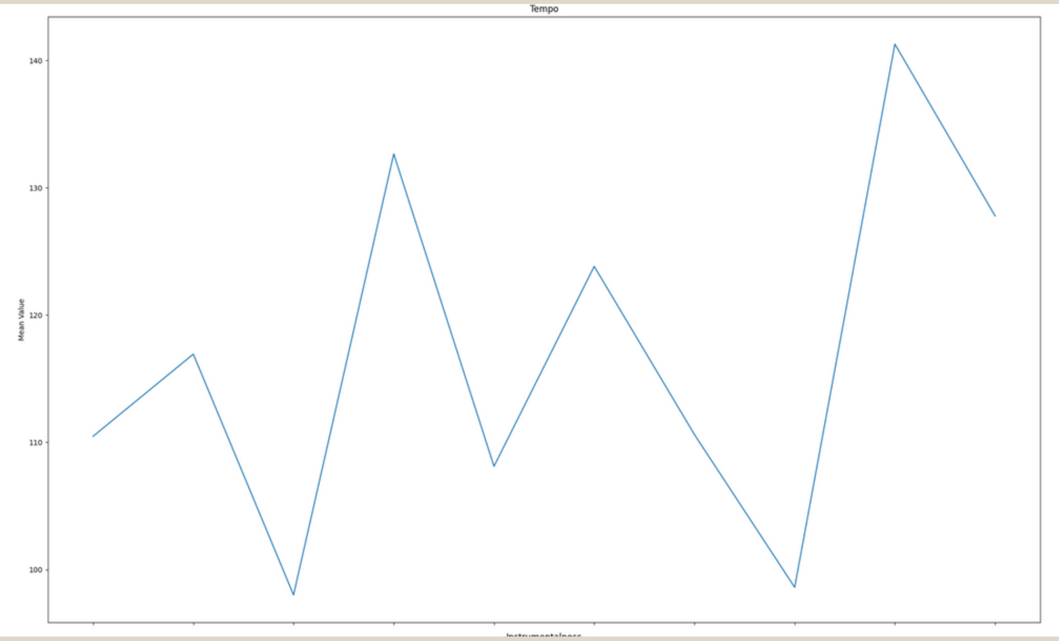


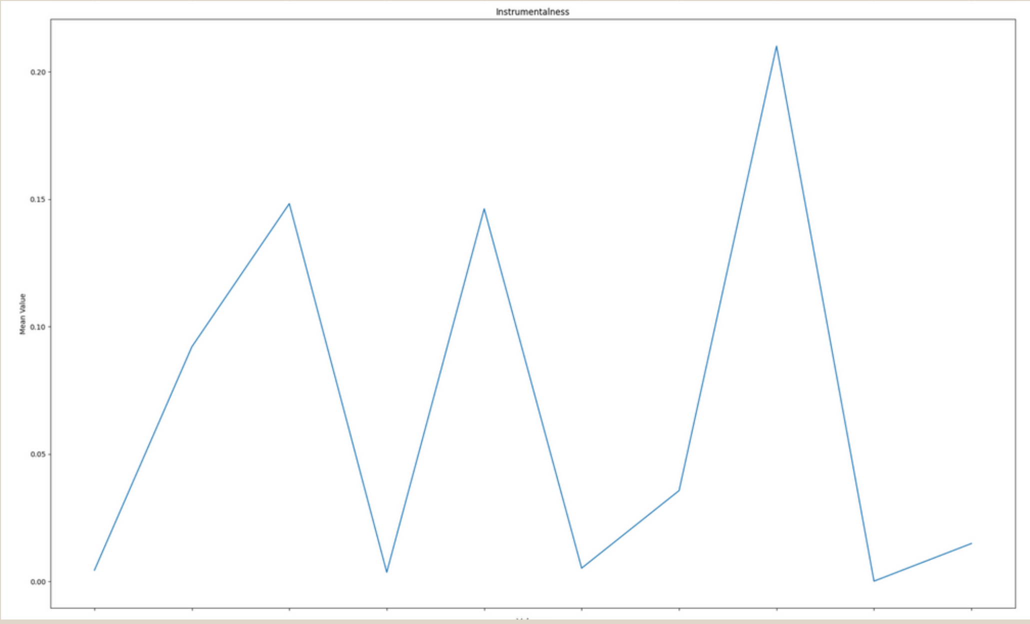
Additionally, I also worked with the previous data I gathered to do with Beyonce's top tracks and stacked the audio features in a bar to replicate equaliser bars. Although the graph resembled my idea, I didn't like the way the data was presented as I didn't find it very accessible or easy to read. I decided to start experimenting on p5.js first instead.

CALCULATIONS









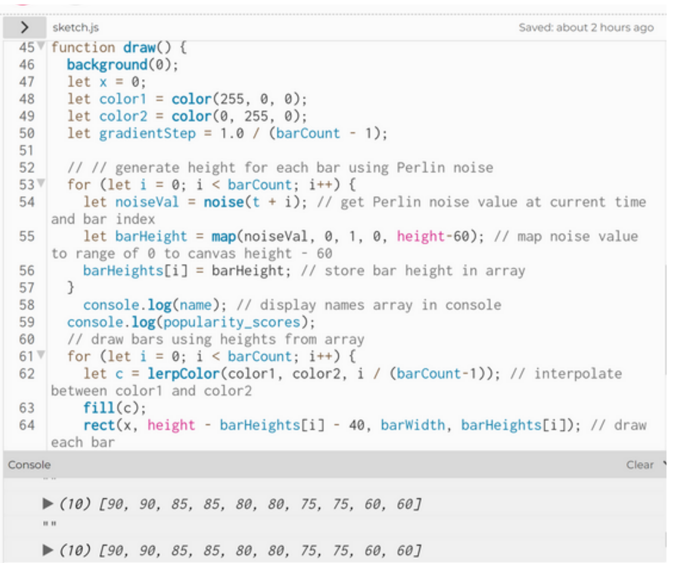

```
# Extracting the audio features from the data
danceability = [float(row['Danceability']) for row in data]
energy = [float(row['Energy']) for row in data]
tempo = [float(row['Tempo']) for row in data]
instrumentalness = [float(row['Instrumentalness']) for row in data]
valence = [float(row['Valence']) for row in data]
speechiness = [float(row['Speechiness']) for row in data]
loudness = [float(row['Loudness']) for row in data]
acousticness = [float(row['Acousticness']) for row in data]

# Calculating statistical measures
feature_means = {
    'Danceability': statistics.mean(danceability),
    'Energy': statistics.mean(energy),
    'Tempo': statistics.mean(tempo),
    'Instrumentalness': statistics.mean(instrumentalness),
    'Valence': statistics.mean(valence),
    'Speechiness': statistics.mean(speechiness),
    'Loudness': statistics.mean(loudness),
    'Acousticness': statistics.mean(acousticness)
}

feature_stds = {
    'Danceability': statistics.stdev(danceability),
    'Energy': statistics.stdev(energy),
    'Tempo': statistics.stdev(tempo),
    'Instrumentalness': statistics.stdev(instrumentalness),
    'Valence': statistics.stdev(valence),
    'Speechiness': statistics.stdev(speechiness),
    'Loudness': statistics.stdev(loudness),
    'Acousticness': statistics.stdev(acousticness)
}

for feature, mean in feature_means.items():
    std = feature_stds[feature]
    print(f"{feature}: Mean={mean:.3f}, Standard Deviation={std:.3f}")
```

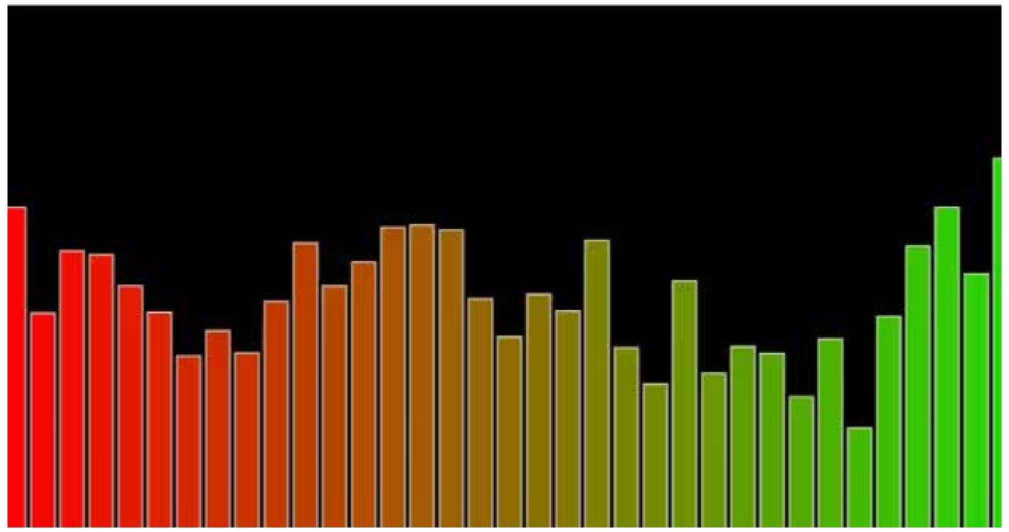
```
Danceability: Mean=0.621, Standard Deviation=0.169
Energy: Mean=0.569, Standard Deviation=0.175
Tempo: Mean=116.827, Standard Deviation=34.607
Instrumentalness: Mean=0.066, Standard Deviation=0.218
Valence: Mean=0.449, Standard Deviation=0.214
Speechiness: Mean=0.108, Standard Deviation=0.113
Loudness: Mean=-7.738, Standard Deviation=4.335
Acousticness: Mean=0.256, Standard Deviation=0.250
```

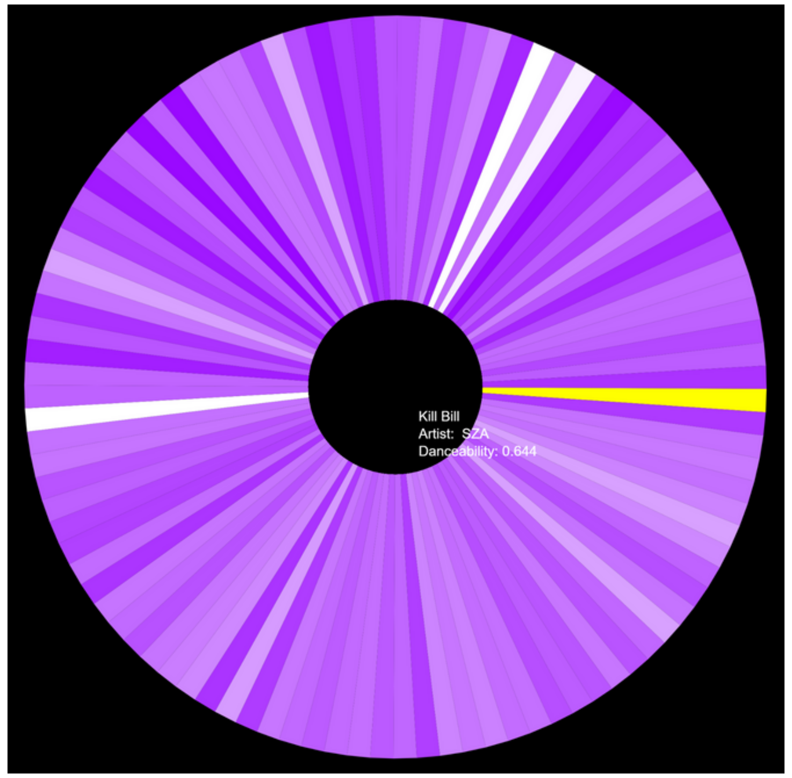
To summarise the data I had collected, I attempted to calculate the average numerical values of each audio features for each year and comparing them to their other years. I wanted to see if there were significant differences between years. In the graphs to the left, it is clear that the audio features are often very different every year. It is hard to determine trends from these graphs alone so I wasn't sure if I wanted to include them in my final visualisations.

EXPERIMENTS

I began experimenting with how the code would look and starting to implement the data into the design creating equaliser-inspired graphs.

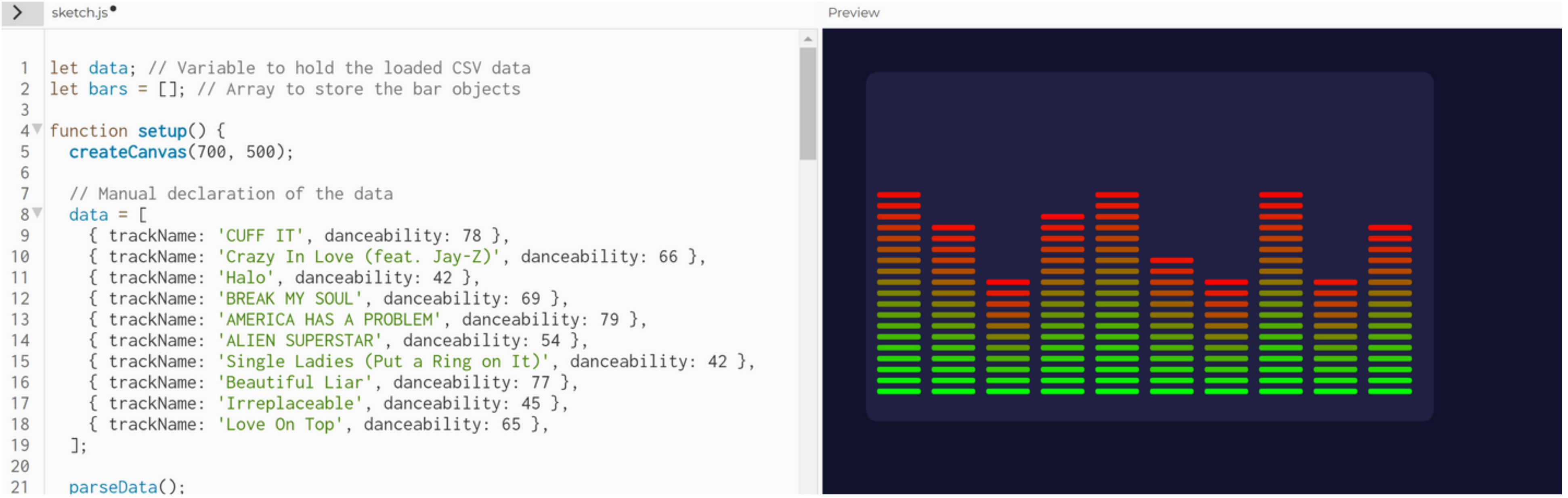






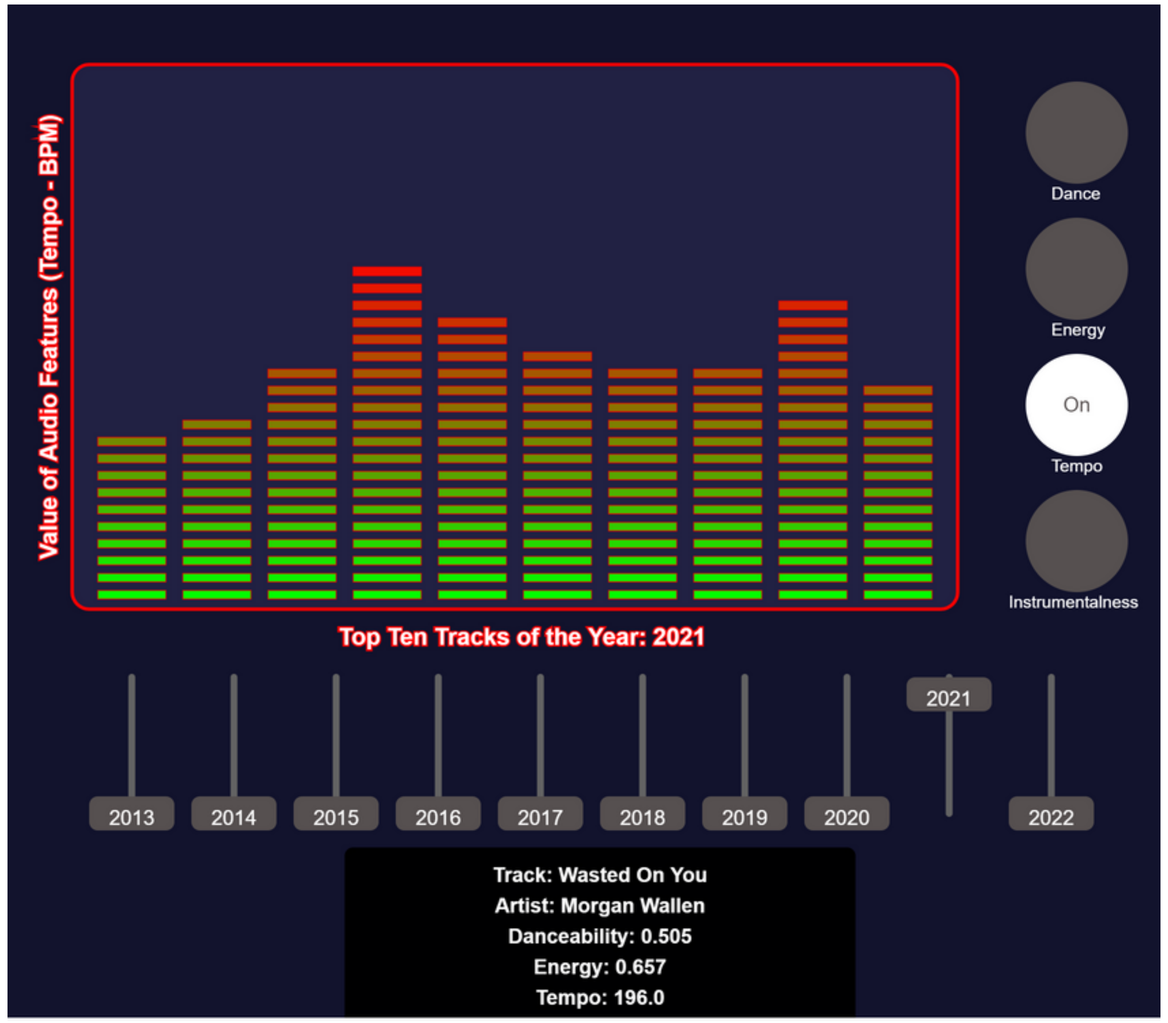
I imported data from a csv file that presented my data in a bar chart as you can see in the image below. Once I was comfortable working with the values in this way I began incorporating the equaliser design into my sketch. I then started working on me second visualisation with the CD.

At first I started to implement data as variables to work with a few pieces of a data whilst I was designing the final product. I used the previous top 10 tracks from Beyonce I had scraped before. I liked the way the equaliser bars looks in the graphs below so this was the design I carried on with when creating the final sketch.



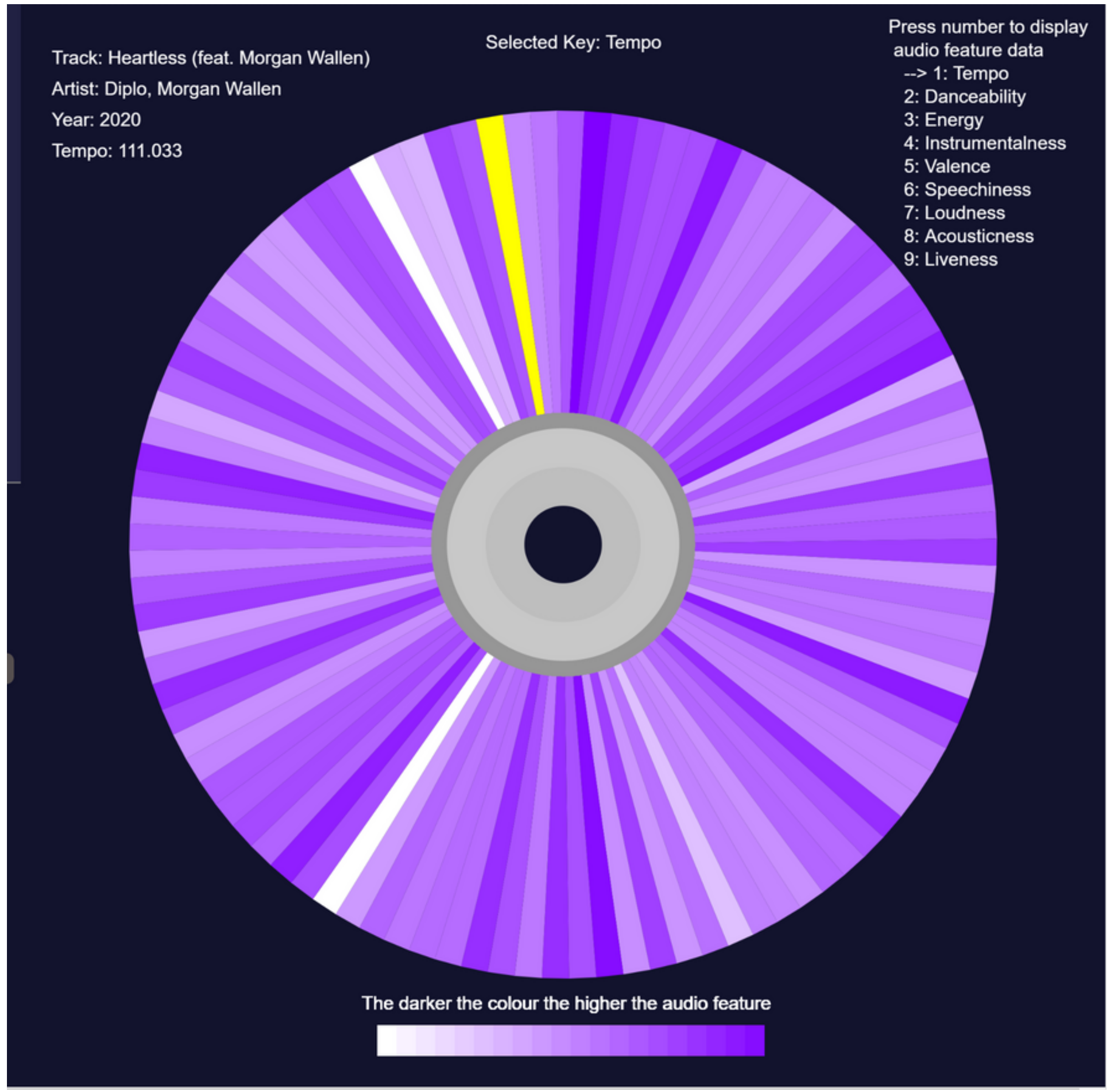
FINAL VISUALISATIONS

Here are the two visualisations I managed to complete, both displaying data of the top 100 tracks over the last decade.



The equaliser bar graph displays 10 tracks at a time per year and the higher the audio features the higher the bar. For example, if the tempo has a higher Beats Per Minute (BPM), the bar will be higher. The X and Y axis change depending on which audio feature or year is selected to make the data more accessible. Additionally, when a user hovers over a bar the track, artist, year and the four audio features available will appear and change depending on which bar the mouse is hovering over. Three audio features in this graph; Danceability, Tempo and Energy were selected because they were the most prevalent in my research about the most popular songs. Therefore I thought one graph including these features would be the most useful. I decided to add a different audio feature, Instrumentalness would highlight any new trends between the other audio features as they are not often compared together.

The CD graph also displays the tracks as bars but instead, all 100 tracks are displayed on the disk in chronological order of the years. Users can click on any bar and track, year and the chosen audio feature will be displayed. As the audio features have numerical values, the darker the colour on the CD, the higher the value of the audio feature. The different shades on the CD resembles a mirror effect on a CD disk. This visualisation has nine audio features as users may be interested lots of different features. The CD graph can be interacted with using a keyboard, in addition to a mouse click, as the the information displays different audio feature data depending on whether a user has pressed 1-9.



CONCLUSIONS - BAR GRAPH

Equaliser - correlations

Based on the equaliser bar graphs, I have conducted the following analysis:

1. Danceability and Energy have a weak positive correlation (0.322), indicating that tracks with higher danceability tend to have slightly higher energy.
2. Danceability and Tempo have a weak negative correlation (-0.052), suggesting that there is no strong relationship between these variables.
3. Danceability and Instrumentalness have a moderate negative correlation (-0.519), indicating that tracks with higher danceability are less likely to be instrumental.
4. Energy and Tempo have a weak negative correlation (-0.198), suggesting that tracks with higher energy tend to have slightly lower tempo.
5. Energy and Instrumentalness have a moderate negative correlation (-0.422), implying that tracks with higher energy are less likely to be instrumental.
6. Tempo and Instrumentalness have a moderate positive correlation (0.441), indicating that tracks with higher tempo are more likely to have instrumental characteristics

.

Based on the average values of these features the trends are:

1. Danceability: The danceability of songs seems to have increased from 2013 to 2018, with a peak in 2018, and has remained relatively consistent since then.
2. Energy: The energy level of songs varies throughout the years without a clear trend. However, there is a slight increase from 2013 to 2014, followed by a slight decrease until 2017, and a slight increase again from 2017 to 2018. After 2018, the energy level remains relatively stable.
3. Tempo: The tempo of songs shows some variation from year to year, but no clear overall trend can be observed.
4. Instrumentalness: The instrumentalness of songs fluctuates over the years, but there is no clear trend. It remains relatively consistent with slight variations.

CONCLUSIONS

After reviewing my project, decided that a third visualisation was required as I felt as though the other two visualisations did not tell enough of a story. If I wanted to answer the question, 'What Makes A Song Popular?' I needed to summarise the data I already had to draw more robust conclusions. I had previously calculated the mean of each audio feature for each year and decided to make synthesiser wave-inspired graphs to showcase the changes in each audio feature over the last decade.

From these final graphs, here are the conclusions that can be drawn:

1. Danceability: The danceability score fluctuates over the years but generally remains within a moderate range. There is a slight increase from 2013 to 2016, followed by a slight decrease in 2017.
2. Energy: The energy level of the tracks shows variations across the years. There is a peak in energy around 2013 and 2014, followed by a decrease in 2015. The energy level then fluctuates without a clear trend until it increases again in 2021.
3. Tempo: The tempo of the tracks also varies over the years. There is a gradual increase in tempo from 2013 to 2021, indicating a trend toward faster-paced music.
4. Instrumentalness: The data shows a decrease in instrumentalness over the years, suggesting that more vocal-oriented tracks have been released.
5. Valence: The data shows a slight decrease in valence from 2013 to 2015, followed by an increase in 2016 and 2017. It then fluctuates without a clear trend until a decrease in 2021.
6. Speechiness: The speechiness score measures the presence of spoken words in the tracks. The data provided shows fluctuations in speechiness without a clear trend over the years.
7. Loudness: The loudness of the tracks remains relatively stable with minor fluctuations throughout the years.
8. Acousticness: The acousticness values indicate the presence of acoustic instruments in the tracks. The data shows a gradual decrease in acousticness from 2013 to 2022, suggesting a shift towards more electronically produced music.
9. Key: The key of the tracks fluctuates over the years without a clear trend.
10. Liveness: The liveness score represents the presence of a live audience in the recording. The data provided shows variations in liveness without a clear trend.

Overall, there are definitely are clear shifts and patterns in what audio features makes songs popular. It is clear that vocal-based songs with a higher energy and a fast tempo are ubiquitous in popular songs. However, songs trends are constantly changing which makes it difficult to draw conclusions from audio features alone. To gather accurate information to answer this question, these visualisations should be compared alongside other factors such as world events, artists popularity and other current trends happening during the time.

LIMITATIONS/CHALLENGES & IMPROVEMENTS

LIMITATIONS/CHALLENGES

- Availability of Data: Spotify API only provides certain data, such as the audio features of songs, data I wanted to collect e.g. the most streamed songs and most followed artists was not available.
- Statistical Significance: Whilst there may be correlations between certain audio features and popularity, it is essential to interpret these findings to avoid making definitive claims without further analysis. Having researched similar studies, it is clear that other determining factors help song popularity.
- Visual Representation: It was important the data was accurately presented, easy to understand and interpret. As a result of this, I felt I was limited in ways I could visualise the data. Whilst I believe my p5.js sketches were clear, as I was working with up to 900 values, there was only so much I could add to my sketches in order for all the code to run smoothly.

IMPROVEMENTS

Overall, I am very happy with how this project has turned out. I have been able to make multiple visualisations, I am pleased with the outcomes and believe they present the data in interesting ways. However, if I had more time, I would make the following changes:

- Summary Visualisation: I had gathered a lot of data in this project and had a lot to work with, but I would have liked a stronger visualisation that brings all the data together.
- Animations: I would have liked to have added some more animations to my equaliser graphs to make it more engaging and visually appealing.
- CD alterations: I would like to add a stop buttons to the spinning feature to make the CD more accessible to users. Additionally I thought about adding more design elements such as a turn table dial to make the visualisations more creative.

REFERENCES

- Al-Beitawi, Z., Salehan, M. and Zhang, S., 2020. What makes a song trend? Cluster analysis of musical attributes for Spotify top trending songs. Journal of Marketing Development and Competitiveness, 14(3), pp.79-91.
- Bertin-Mahieux, T., Ellis, D.P., Whitman, B. and Lamere, P., 2011. The million song dataset.
- Borg, N. I. C. H. O. L. A. S., and G. E. O. R. G. E. Hokkanen. "What makes for a hit pop song? What makes for a pop song." Unpublished Thesis, Stanford University, California, USA (2011).
- Hennion, A., 1983. The production of success: an anti-musicology of the pop song. Popular Music, 3, pp.159-193.
- Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z. and Komarova, N.L., 2018. Musical trends and predictability of success in contemporary songs in and out of the top charts. Royal Society open science, 5(5), p.171274.
- Pham, J., Kyauk, E. and Park, E., 2016. Predicting song popularity. Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep, 26.
- Terroso-Saenz, F., Soto, J. and Muñoz, A., 2023. Evolution of global music trends: An exploratory and predictive approach based on Spotify data. Entertainment Computing, 44, p.100536.
- Zangerle, E., Vötter, M., Huber, R. and Yang, Y.H., 2019, November. Hit Song Prediction: Leveraging Low- and High-Level Audio Features. In ISMIR (pp. 319-326)