

## STA 141B Project

Members: Ryan Smith, Isaiah Valencia, Lauren Chan, Yuhong Fang

### Introduction:

Our Shiny Application will consist of musical data on several Spotify artists. It will allow the user to choose from several different genres in a dropdown, and return four popular artists per selected genre. The next dropdown will prompt the user to select which artist(s) they would like to view data on. The data for each of these artists will be collected on a per-track basis, and will include the following information about each of the selected artists' tracks: artist name, album name, album release year, danceability, energy, key, loudness, liveness, valence, tempo, and several other data pieces per track. The data is pulled from the Spotify API. Once the user selects their genre and artists in the dropdown menus, several visual graphs return and interpret the information for each of the artists selected. When the user selects artists, they will be shown a list of recommended artists based on the input artists, and several graphs that describe the relationships of the data.

### Description of the source of the data:

The data we used was collected from the Spotify API, which provides data about various features of the tracks of an artist. A lot of the information about the tracks is objective, such as the length of a track or the loudness of a track. There is, however, a lot of data about each track that is calculated by Spotify. An example of this is

danceability, which is on a scale of 0.0 to 1.0 for each track, and is calculated using elements such as “tempo, rhythm stability, beat strength, and overall regularity”. We decided to use the layman's interpretation of danceability, which Spotify describes as “how suitable a track is for dancing”. A list of the variables we collected and their corresponding descriptions from the Spotify API documentation:

1. Artist\_name: The name of the artist
2. Genre: The genre of the album
3. Popularity: Popularity of an artist from 0 to 100
4. Followers.total: Information about the followers of an artist
5. Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track
6. Liveness: Detection of the presence of an audience in the recording (0.0 to 1.0)
7. Danceability: The suitability of a track for dancing (from 0.0 to 1.0)
8. Acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic
9. Loudness: The overall loudness of a track in decibels
10. Energy: A measure from 0.0 to 1.0 of intensity and activity
11. Duration\_ms: The duration of the track in milliseconds
12. Album\_release\_year: The year of the release of the album
13. Tempo: The overall estimated tempo of a track in beats per minute.

How you retrieve and process the data:

The source data for our project was collected from the Spotify API. To access the API, we used the package 'spotify' in RStudio, which required creating a spotify developer account to get an access token. (We initially tried to use manual queries but Professor Lai said that because it is a more difficult two-factor authentication system for the spotify API, we were allowed to use the 'spotify' package). We used the function `get_artist_audio_features` with the user-selected artists as arguments to retrieve the majority of the information on those artists. We also used the function `'get_recommendations'` with the input artists as arguments in order to find similar artists to recommend. To get the data on the selected artists, we set up two dropdown menus. The first allows the user to select one of four genres, which then shows four random popular artists from each selected genre using the function `'get_genre_artists'`. Then, the user selects their desired artists. We get the data from all 16 artists when the app initially loads (4 from each of the four genres, 'pop', 'blues', 'wonky', 'rock'), and then filter it based on the user-selected options. Then, within each plot we select which variables and filters we need and filter down the overall data set.

#### User guide of the application:

1. When the app is run, the user must wait for the data to load, which can take up to 30 seconds.
2. When the dataset is loaded, the application will open and the user will be prompted to select their desired genre(s) within the first dropdown. The user can select between 1 and 4 of the genres, 'pop', 'blues', 'wonky', and 'rock'.

3. The next dropdown will include 4 artists per selected genre, and it will allow the user to select anywhere from 1-16 artists of interest .
4. If the user chooses to select one artist, a table of recommended artists appears below the dropdown options. On the right hand side, several visual charts about the selected artist emerge. The summary table of the artist's audio features appear on top, along with a bar chart of this information provided below. This includes information about the selected artist's tracks, mentioned above in the Introduction. These audio features are also utilized in the following visual graphs and charts, including scatterplots of the artist's valence vs liveness and valence vs danceability, the time progression of tempo from each of the artist's albums (based on album release date), and a histogram of average track length of the artist.
5. When the user chooses two or more artists (regardless of picking from the same or different genre), the same summary, graphs, and charts appear with different colors corresponding to the multiple selected artists. Below the two dropdown menus, a list of 10 recommended artists based on the selected artists will also appear.
6. Many of our plots were created using the Plotly package, which allows the user to interact with the graphs. For example, you can put your mouse cursor over a specific point on the time-series chart to view the values for a specific album. You can also double click one artist at a time on the histogram, for example, to view only the data for that one artist.