

CS 3300 Project 2

Visualizing Album Ratings vs. Album Sales

Joshua Hong (jh976) • Kaushik Ravikumar (kr437)

The Story

Our visualization focuses on displaying whether there is a relationship between an album's ratings and its number of streams/sales. One of the things that inspired us to create this visualization was the Youtube channel, TheNeedleDrop, which has a reputation for incorrectly reviewing many albums, often giving popular and well-liked albums very low ratings. We were curious to see if there was any correlation between an album's ratings and its popularity on popular streaming platforms.

We utilized Pitchfork (a popular music review website/magazine) ratings and Spotify popularity measurements to display the correlation between an album's reviews and its popularity. Our visualization also helps one determine if genre has any effect/is a factor in determining an album's rating or popularity. To our surprise we found that there was not much of a relationship between an album's popularity and its rating. It seems that the popularity of an album and Pitchfork ratings are often independent from each other, showing that even though many people are exposed to album reviews, this does not prevent people from listening to certain albums.

When viewing the effect of genre, we found that pop averages the highest popularity no matter the rating an album received. However, even the slight relationship we found between the pop genre and Spotify popularity was not significant enough to conclude a definite relationship. Through analysis and observation of the gathered data, we concluded that the Pitchfork ratings are an invalid metric of measurement for popularity. The message we want to convey from our story is that ratings have no bearing and should have no bearing on people's preference and enjoyment of music. In the end the ratings given, no matter the credibility of the source, are a subjective score produced by a third party.

Data

We used a dataset on Kaggle containing over 18,000 albums reviewed by Pitchfork up until 2017. The dataset was found in the link below.

<https://www.kaggle.com/nolanbconaway/pitchfork-data>

This dataset was in the form of a SQLite file. We then used SQL to query the data set and filter it out to only include album reviews from 2016-2017. We then converted the SQLite file to a CSV, so it can easily be imported by D3 into our project. We then wrote a python script, in the file `script.py`. This python script used the album name and artist of each of the albums in the filtered pitchfork data set to make a request to Spotify's API and obtain the popularity for each respective album and other information about the album, such as its release date, the label name, and the number of tracks. We also had to make separate requests to a separate Spotify API endpoint to find the list of genres for each album in the dataset. We wrote the responses of the Spotify API requests into a JSON file called `popularity_data.json`. This JSON file has the following format. The Spotify Popularity Rating is calculated by Spotify ranging from 0 to 100. It takes into consideration the total number of streams for an album and the current popularity trend.

```
{
  "genres": [
    (List of Strings representing each genre it falls under)
  ],
  "artist": "Artist Name",
  "release_date": "YYYY-MM-DD",
  "title": "Title",
  "numTracks": Num Tracks,
  "pop": Popularity Rating from (0 to 100),
  "label": "Record Label Name"
}
```

We then had to further filter out both datasets, because when making the Spotify API requests, there were a few Pitchfork Reviewed Albums that gave us erroneous responses when searched for in Spotify. This involved some manual filtering and some helper python scripts. The final Pitchfork Review Data Set was in `more_filtered_pitchfork_data.csv`. This file has the following columns for each row.

```
reviewid,title,artist,url,score,best_new_music,author,author_type,pub_date,pub_weekday,pub_day,pub_month,pub_year
```

Design Overview

For the design of the visualization, we constructed an interactive graph for the albums found in `more_filtered_pitchfork_data.csv`, where each album's Spotify popularity rankings were plotted against their Pitchfork ratings. We used linear scales for each axis in order to facilitate analysis on the relationship between the two variables, and gave colors according to the album's genre. We used the d3 built-in color scale `d3.schemeSet3` to assign colors for each genre.

One of the challenges we encountered when developing the genre filtering feature of our visualization was the following. Spotify provided us with a list of genres for each album and in order to do genre filtering, we needed there to be a finite list of genres. Thus, we had to use a script and some manual observation to determine the finite list of genres. We came up with the following.

- Pop
- Rap
- Hip Hop
- RnB
- EDM
- Alt/Indie
- Rock
- Ambient
- Metal
- country
- jazz
- Other

In order to determine the single genre for each album, we used the first string in each album's genre array. In this string, we simply checked if it had certain keywords that we determined were appropriate from observation and use of scripts.

In order to display more information about an album, we included a side panel. An album that is selected will have its complete information displayed for the user. We used a d3 arc to visually depict an album's spotify and pitchfork rating as a percentage of the maximum possible ratings.

Interactivity Overview

Everything discussed above are the static components to our visualization. The way we made our visualization interactive and dynamic was by implementing the following features:

1. Hover Over and Click on Circles
2. Search Albums with Search Bar
3. Filter out Albums by genre

Hover Over and Click on Circles

In order to implement the hover over and on click functionality, we added listeners for each circle. A user is able to view an album's title in a little label by hovering over the album's circle on the chart. If the user would like to view the album's complete information, he or she can click on the circle so that it is selected. Once an album is selected, the right side of the visualization displays a SVG with the album's complete information. This complete information includes the album's Spotify rating, Pitchfork album rating, number of tracks, release date, artist, title, label name, etc.

Search Albums with Search Bar

In order to implement the search bar, we had to include a search bar which uses JQuery autocomplete functionality. It essentially searches an array we created with the album name and the artist name separated by a comma. Once a user has selected one of the albums from the filtered suggestions, it selects the album and similarly displays the album's complete information.

Filter out Albums by genre

In order to implement this filtering functionality, we used a clickList to maintain a list of the genres selected. This allows a user to filter the albums to include multiple genres at a time. We changed the opacity to display whether an album was filtered out or included.

Our rationale for any affordance or design relating to interactivity was to assist users digest the information presented by the visualization. The search bar is above the graph to show user's, if need be, they could search for a specific album, and the interactive information is presented next to the graph in order for clear and easy access to album details. The bar for highlighting genres is on the bottom of the page, where the bar is

filled in grey. This design choice allows for each genre to pop and draw users to the application of this feature.

We made each datapoint enlarge and present the album's name as the user passed their mouse over/on them. This allowed users to know both which datapoint and album they were selecting. However, when clicking we designed the interaction to highlight the circle and show its album details on the side. We used this design because in conjunction with the mouseover feature, it would be very intuitive for any user. The same highlighting attribute can be found for the search bar and the genre bar. The search bar, which has very intuitive usage, was designed to be very easy to use and understand, since its sole function was to give the user results based on inputted keywords. The genre bar has 12 choices for genres, where when one is selected, it highlights all of the albums with that specific genre, giving another story to consider.

Individual Contributions

Kaushik Ravikumar

- Data Organization
 - Built Python script to generate second data set with Spotify data for each album
 - Completed organization of data sets so they are both in the same order and are of the same length and where each int value index is corresponding to the same album in both datasets.
 - Filtered out erroneous data that emerged from Spotify API requests
- Built side panel with Album Information
- Built search album functionality

Joshua Hong

- Built Genre filtering functionality
- Debugged numerous interactivity errors, such as the error where multiple albums were selected and had emphasis shown on them at the same time
- Built text wrapping feature to prevent album information text from being cut off