Mike Spiegel
2/28/2014

Intro

Popular music is a reflection of the culture and setting in which it was produced. Throughout the past decades, American culture has undergone radical shifts in its tastes, beliefs, norms and values. By combining my existing musical skills with the techniques developed in this class, I believed I could discover some interesting correlations between the musical tastes of past decades. More specifically, I was interested to see what trends (if any) existed between the popular music of past decades.

Sources

This project used two separate API resources, and each API was used for distinctly different sets of the project. The first part of the project involved creating an index of popular music from each year from 1975-1995. To create this index, I made use of the Last.fm API "Get Album Info feature" (documentation here). Each query returned a large amount of metadata in XML form on the album including the album artists, track name and duration of the track. Since I was focused on creating an index of songs, the only information that was required from this data source were the album year, the tracks and the artists. The first step of the processed involving going through approximately 25 years of album information (25 records total).

The second API used was the Echonest API "Song search" method (documentation here) and the "audio attributes" method (documentation here). This API provides a wealth of resources on the acoustic and compositional elements of a song, including tempo, time signature, and key signature, as well as proprietary evaluations of the track such as "danceability" and "energy". Since I was unsure how Echonest produced these scores, I chose to use the more vanilla descriptive variables; specifically, tempo, modality, key signature, and track length. Since these records needed to be created for each track on an album, 205 total records were pulled from the Echonest API.

Mike Spiegel
2/28/2014

Methods

The process of combining these two sources was relatively straightforward. The first portion of the code makes a request from the Last.fm server for album info on *Billboard Top Hits: (insert year here)."* The resulting XML is then processed within this FOR loop using BeautifulSoup. The parser looks for the Artist and Track information for each album, and the resulting information is written into a .txt file. Initially, this information was being placed in a CSV file, but Microsoft Excel does not seem to process Unicode characters. During this step, there were some albums that were missing from Last.fm's record. This years were skipped. However, a separate text file (access_log.txt) was created to show which albums were unable to be pulled from the Last.fm.

The second step of this involved getting the ID values for each song from Echonest, and then passing those values back into Echonest to retrieve the technical metadata for each track. This required opening the index file and passing the values in the artist column through the Echonest artist search parameter to get the Echonest ID. These results were written to a separate CSV file

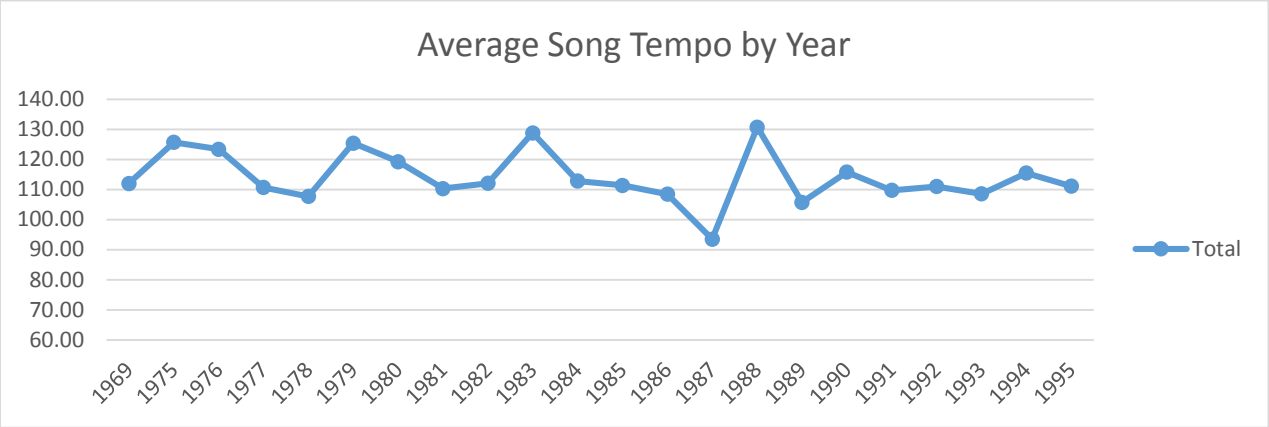
Finally, these individual song IDs were once again passed to the Echonest API, and written to a final csv file that contained all of the necessary information for further analysis.

The two main challenges with this project were various issues generating properly formatted HTML strings for urllib2, and the slow response time of each API. One problem I did not anticipate was passing artists with Unicode characters in their name back to Echonest. For example, the artist "Exposé" would consistently cause the urllib2 module to return a server error, as the HTML string was not equipped to handle the "é" character. Hardcoding a solution to this problem would have been quite difficult, but

fortunately there already existed a python module that would transliterate Unicode strings into ASCII. For example, this module provided a function that would convert “Exposé” to “Expose” By using this module on the strings before they were concatenated onto the stub string, I was able to simply solve this problem.

Analysis and Visualization

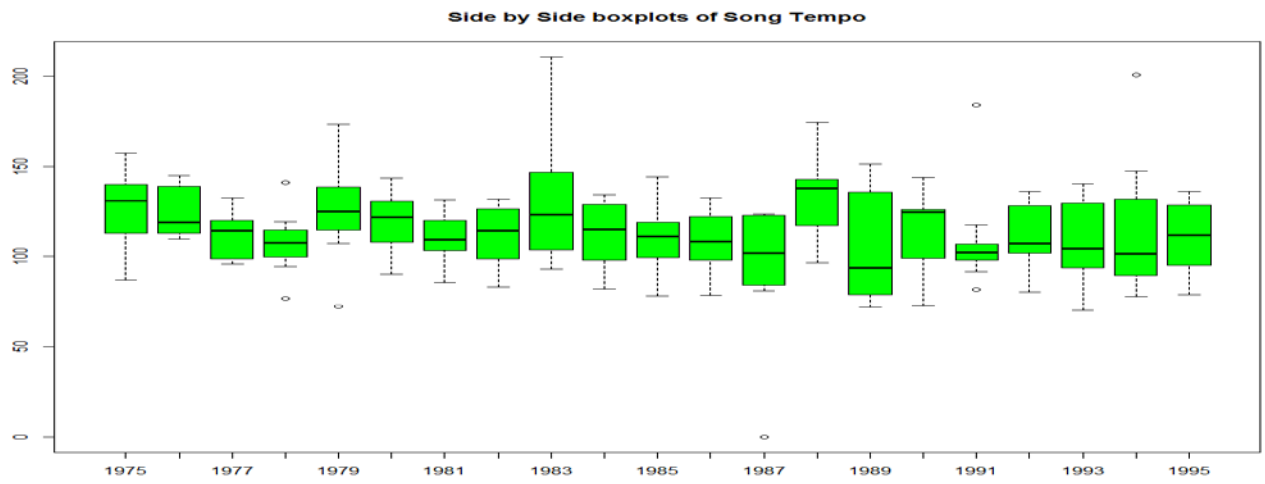
Once I had my final spreadsheet ready, I had several options available for analysis. To get a general sense for how song tempos varied from year to year, I created an Excel Pivotchart that graphed song the average tempos for each year.



One interesting feature of this graph is the averages begin to converge around 110 Beats per minute (BPM) as time goes on. This seemed to imply that popular musical sensibilities were converging around one style of music as time went on. However, this graph did not explain the variance in popular music for each year.

My next step was to load my dataset into R and produce a boxplot graph of each year.

Mike Spiegel
2/28/2014



After examining the plot, it became apparent that not only was the mean tempo of each year becoming more centralized, but that the variance within each year was also becoming smaller. Finally, I examined the overall modality of songs from the entire dataset, and found that ~75% of these songs were in the major modality.

The combination of these findings seemed to point towards some broad trends in popular musical tastes from 1975-1995. Firstly, it seems to indicate the preference in popular music seems to gravitate towards songs that are “danceable”-that is, songs that fall between 110-130 BPM. Secondly, it popular musical tastes seem to favor the major modality, regardless of the decade the song was created in.

This project was incredibly interesting, but there a few things that I would have done differently. I assumed that Last.FM would have a record of each year from 1970-present. Unfortunately, this was not the case, and I had to change the scope of the project to examine the data that was available. If I were to do a similar project, I would scrape a large collection of records (such as Wikipedia) to get a more complete index. Secondly, I would have tried to consider the implications of passing Unicode characters through HTML before spending a couple of hours trying to debug the problem.

Mike Spiegel
2/28/2014

Example URLS:

Last.FM Album Search:

http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=643bb61a651fa9295b7ee2a35945e93d&artist=Various%20Artists&album=Billboard%20Top%20Hits:%201988

Echonest Artist ID search:

http://developer.echonest.com/api/v4/song/search?api_key=9TXIIH4LC2GJOO25A&artist= Paperboy&title=Ditty

Echonest Song Audio Summary:

http://developer.echonest.com/api/v4/song/profile?api_key=9TXIIH4LC2GJOO25A&id=SOGNUYN13EE59352FE&bucket=audio_summary