**Exploring Characteristics of Songs on Spotify**

By Alec Klessig, Travis Clark, Chandler Morris, & Mychael Solis-Wheeler

# Introduction

(*This section is authored by Travis Clark)*

Music is an art which is enjoyed around the world. However, when asked why they enjoy music, people often give explanations based on ambiguous language. Answers such as “The song made me feel good.” and “That song was upbeat, and it had a catchy tune.” are subjective statements related to people’s personal tastes. And yet, some songs are consistently popular for long periods of time. Why are some songs more popular than others? What makes a “good” song?

As is true with other art, a song’s popularity is affected by factors external to the song, such as the culture of the area and time in which the song was published. This report focuses on the content of the song itself. Are there aspects to the audio features of a song which affect the chance that the song will be popular? If so, which features make the biggest difference? Are there some audio features which tend to appear together in popular songs? These are the kinds of questions this report aims to answer.

Exploring the qualities of popular songs is not unique to this report. Others have explored this topic with mixed results. For example, the chapter in *Music Data Mining* titled “Hit Song Science” [1] explores the popularity of songs and some attempts to detect patterns in the audio features of these songs. It concludes that, so far, attempts to predict the popularity of a song have been unsuccessful and that the task is complicated by various social and psychological effects. Meanwhile, Wang’s “Predicting Hit Songs with MIDI Musical Features” [2] demonstrates some success in using features such as instruments, melody, and beats to classify songs as popular or not popular.

On a related point, it seems reasonable to assume that music streaming services, such as Spotify, use other metrics in addition to popularity within their recommender systems. For example, a music streaming service might recommend high energy songs to someone who listens to fast and loud music. Therefore, this report explores relationships between metrics other than popularity as well.

This report explores a dataset on Kaggle.com titled “Spotify Audio Features” [3]. This dataset contains 116,372 rows. Each row describes one track (i.e. song) on Spotify. The dataset has 17 columns, each describing an attribute of a track. Some columns, such as “artist name” and “track id” are identifiers. Some columns, like “duration”, “key” and “time signature” are easily measurable attributes of a track. Other columns, such as “danceability”, “speechiness”, and “valence” contain values generated by algorithms created by Spotify which estimate more complex audio features in a track. Initially, before any analysis was performed, the authors were especially interested in the column “popularity”, which contains values generated by a Spotify algorithm to estimate the current popularity of a track. The algorithm is mostly concerned with the number of plays of a track and how recent those plays are. While the other columns remain constant over time, popularity is updated every several days, so the values for this column are current only up to the time of data collection, which was on December 3, 2018. A full description of each column in the dataset is in Appendix A. This report only considers a subset of the available attributes: danceability, duration\_ms, energy, loudness, tempo, valence, and popularity.

The following sections describe how the data was cleaned; the methods used to explore the data, including dimension reduction analysis, cluster analysis, and confirmatory factor analysis with the results of each of these methods provided. All R code used to perform the tasks in each section is provided in Appendix B.

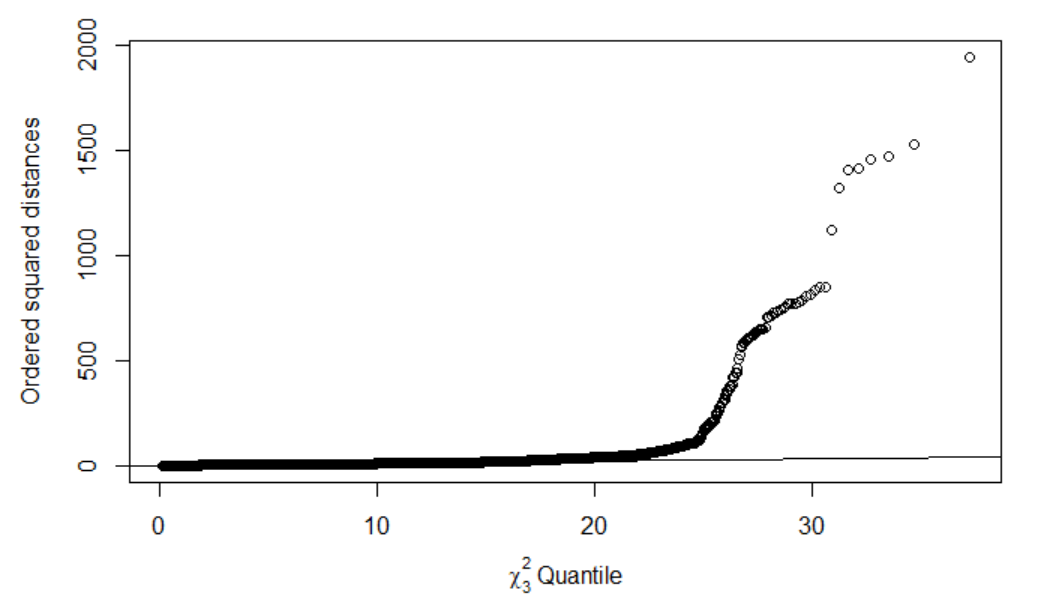
# Data Cleaning

*(This section is authored by Travis Clark)*

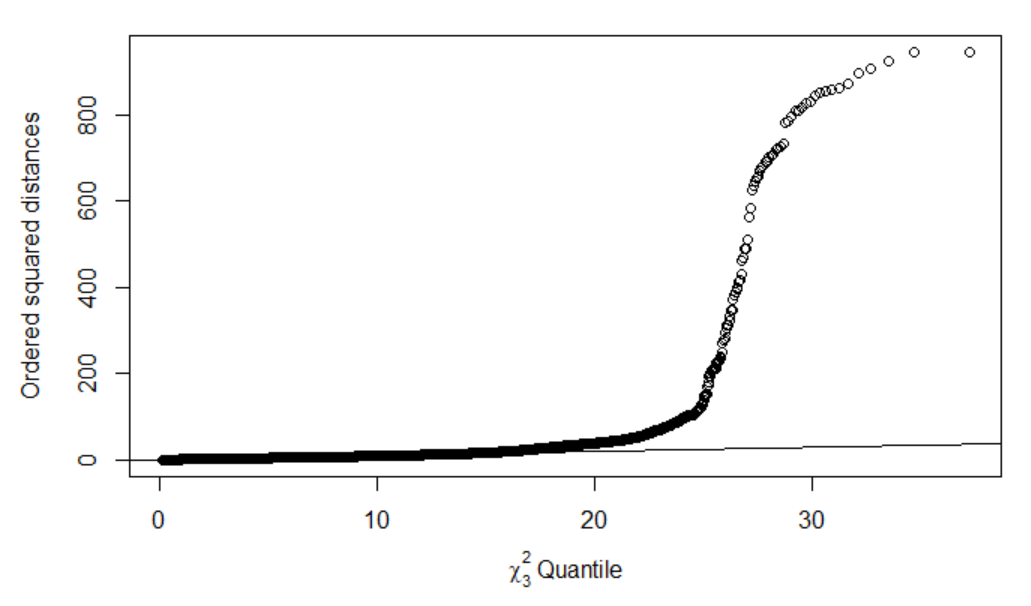
Before any analysis was performed, the data had to be cleaned. This section describes the cleaning process used on the source dataset.

First, any columns we were not interested in were removed. This left 7 variables remaining in the data. After checking that the desired columns were there and were assigned the correct data types, the data was examined to detect any missing values; there were none. This made sense, as the data was generated by Spotify’s API, which was able to calculate all features for any track. Then, another pass through the data checked for duplicate rows; there were none. The data was then scaled to ease analysis and interpretation later on.

Next, the data was visually tested for multivariate normality using a chi-squared plot of the Mahalanobis distances of each point, drawing the line y = x for reference. This plot is shown in Figure 1a. Since a sizeable portion of the points were far away from the line and from the origin, the authors concluded the data was not multivariate normal. In addition, there were eight outliers which stood out in Figure 1. The outliers were removed, and a new chi-squared plot which did not contain the outliers was drawn. This new plot is shown in Figure 1b.



1. With outliers



(b) No outliers

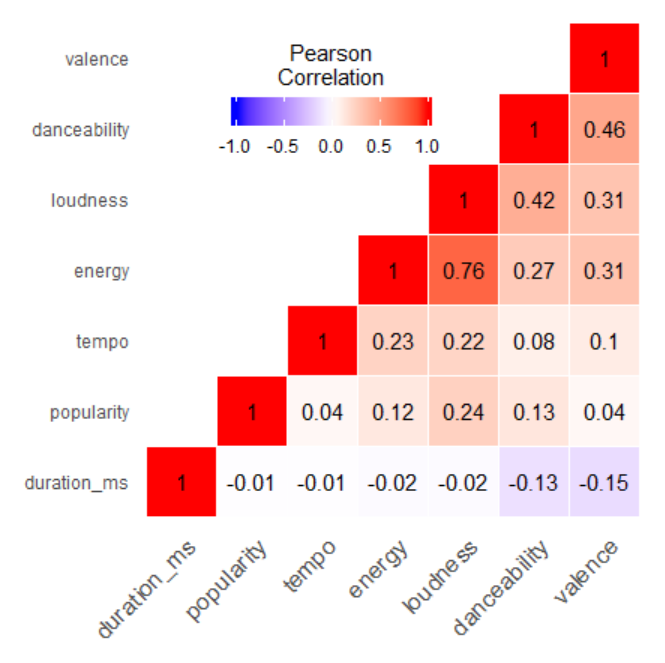
**Figure 1: Chi-squared plot used to test for multivariate normality, with outliers (a) and without outliers (b)**

To determine why the outliers deviated so far from the rest of the data, the outliers were extracted, and their z-scores were examined. All eight outliers had reasonable z-scores for every variable except duration\_ms. For each of the eight outliers, the z-score for duration\_ms was greater than 30! Therefore, the outliers deviate from the rest of the data because they are relatively extremely long songs!

# Initial Exploration of Data

*(This section is authored by Travis Clark.)*

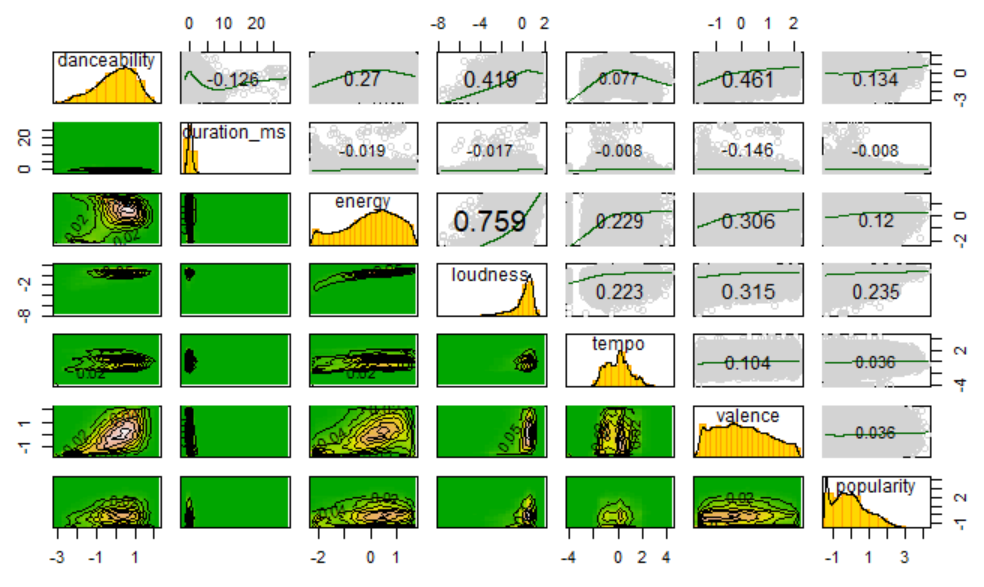
After the data was cleaned, some of the basic multivariate characteristics of the data were explored. By reusing code found online [4], a heatmap of the correlation matrix of the data was created, and it is shown in Figure 2 below:



**Figure 2. A heatmap of the correlation matrix of the cleaned Spotify data**

Most of the correlation coefficients in Figure 2 are low, having a magnitude less than 0.4. There are a few exceptions: energy and loudness, danceability and loudness, and danceability and valence all have correlation coefficients greater than 0.4. But generally, the correlations of the variable pairs are low. These low coefficients signaled poor results for analyses which functioned best with highly correlated data, such as principal component analysis (PCA) and exploratory factor analysis (EFA). The authors kept this in mind when exploring the data.

A second initial visualization was created using the kdepairs function in R. This visualization is shown in Figure 3 below:



**Figure 3. A matrix of visualizations. 2D kernel density estimates and contours in the lower half, histograms of each variable on the diagonal, and scatterplots with lowess smooth curves and Pearson correlation coefficients in the top half**

The authors drew some conclusions from Figure 2. First, the histograms of each variable showed none of the variables were univariate normal, so it made sense that the data was not multivariate normal. Also, the histograms showed visual insight on why some data points were outliers: the duration\_ms distribution appeared highly skewed! Thus, tracks on the upper extremes of duration\_ms would be farther from the other tracks and function as outliers in the data.

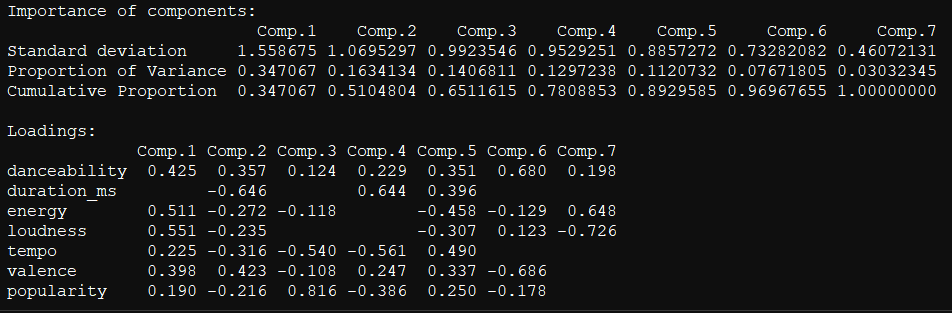
Second, the contours showed there may be some clusters somewhere in the data. Not all variable pairs showed signs of clusters, but it seemed tracks were clustered in variables such as danceability, energy, and valence.

# Dimension Reduction Analysis

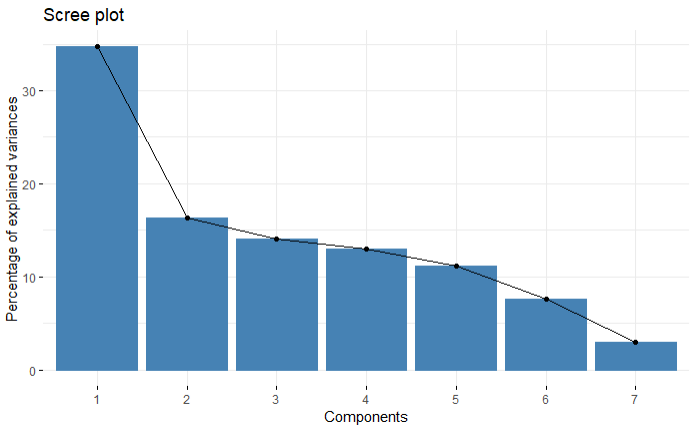
*(This section is authored by Chandler Morris)*

In order to reduce the dimensionality of our original Spotify dataset while also accounting for as much of the original variation as possible, principal component analysis (PCA) was performed. Concerning other dimension reduction techniques, multidimensional scaling was not appropriate due to the number of observations in the data. The data had no categorical variables, so correspondence analysis (CA) was not suitable for this dataset. Finally, canonical correlation analysis (CCA) was not performed before there were not two distinct sets of variables.

Performing PCA on the dataset:

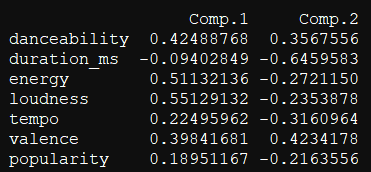


After performing PCA for the Spotify data set, we found the following insights. The first component covers 34.71%, and the second component covers 16.33% of the variance of the data. The total amount of variance covered by the first two principal components is 51.04%. Figure 4 is a scree plot displaying the percentage of explained variables for each principal component.



**Figure 4: Percentage of explained variance between components.**

The loading coefficients for the first 2 components are as follows:



Principal component 1 seems to have contrast between energy, loudness and danceability. Principal component 2 seems to have contrast between duration\_ms and valence.

The linear equations for pc1 is: pc1 = 0.425 \* DANCEABILITY - 0.094 \* DURATION\_MS + 0.511 \* ENERGY + 0.551 \* LOUDNESS + 0.225 \* TEMPO + 0.398 \* VALENCE + 0.189 \* POPULARITY

The linear equations for pc2 is: PC2 = 0.357 \* DANCEABILITY - 0.646 \* DURATION\_MS - 0.272 \* ENERGY - 0.235 \* LOUDNESS - 0.316 \* TEMPO + 0.423 \* VALENCE - 0.216 \* POPULARITY

As you can see from the biplots in Figure 5 below, the correlation between valence and danceability, and energy and loudness are extremely high. The correlation between tempo and popularity is not very high and not well represented in the first two principal components. The points are removed from the biplot in Figure 5b because the number of observations is large.

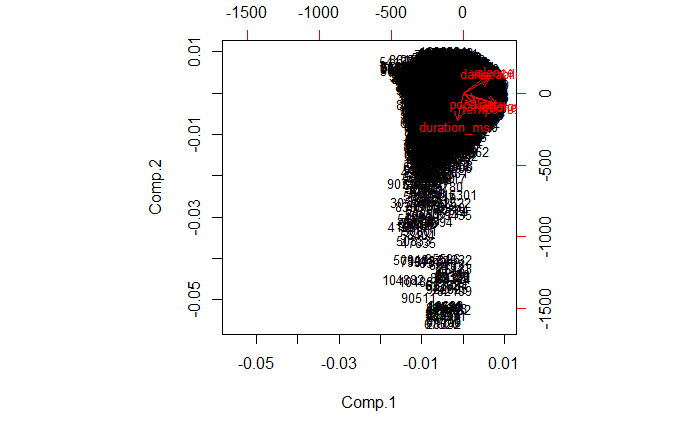


Figure 5 (a)

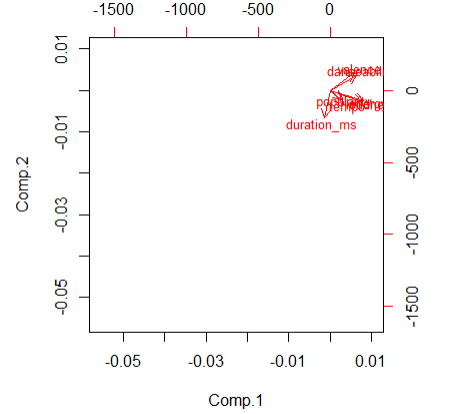


Figure 5 (b)

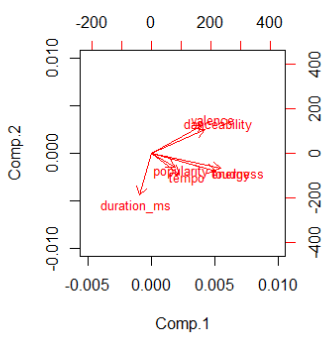


Figure 5 (c)

**Figure 5: Biplots (a) contains points for the data, (b) contains no points for better viewing of the arrows, (c) contains no points, and has a limit for x and y for a close up of the arrows.**

Exploratory and Confirmatory Factor Analysis

*(This section is authored by Alec Klessig)*

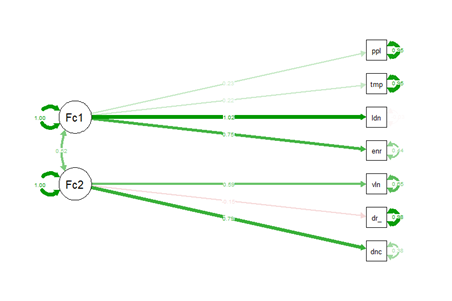
In order to perform confirmatory factor analysis, a theory had to be established in order to specify the model that would be used. This was achieved through performing Exploratory Factor Analysis first in order to suggest potential latent variables and which observed variables pair with them to ultimately describe the model in confirmatory factor analysis.

However, both factor analysis methods require a number of factors to begin with. Beginning with one factor, the exploratory method is run using the correlation of the sliced Spotify dataset. However, when calculating the root-mean-square error for one factor, the RMSE is 0.076, which is greater than the desired 0.05. Thus, exploratory factor analysis is then run again with 2 factors, yielding an RMSE of about 0.028, which is less than 0.05 and therefore acceptable for moving on to confirmatory factor analysis.

When running this method, loadings from the exploratory factor analysis are used in order to suggest variables for the factors. For the first run through, only variables with loadings greater than 0.4 on the factors are considered. This gives *energy* and *loudness* under Factor 1, and *danceability* and *valence* under Factor 2. The model is then specified for use with the confirmatory method, and the following measures of fit are given in Table 1:

|  |  |  |
| --- | --- | --- |
| Goodness-of-fit | Adjusted Goodness-of-fit | SRMR |
| 0.9858048 | 0.8580481 | 0.02900474 |

**Table 1. Measures of fit for first CFA model**

While the goodness-of-fit and standard root means square difference are both good (greater than 0.95 and less than 0.05, respectively), the adjusted goodness-of-fit index is less than the desired 0.95. According to the adjusted fit index, the model is not confirmed. Because of this, the model specified was revisited and redefined. The new model specified uses every variable from exploratory factor analysis, such the given variables for Factor 1: *energy, loudness, tempo, popularity* and Factor 2: *danceability, duration\_ms, valence* depending on which factor they were more loaded on. Using this new model, the confirmatory analysis method was ran again. The path diagram for the model is shown in Figure 6, along with the measures of fit in Table 2. 

**Figure 6. Path diagram for updated CFA model**

|  |  |  |
| --- | --- | --- |
| Goodness-of-fit | Adjusted Goodness-of-fit | SRMR |
| 0.9817887 | 0.9607757 | 0.0315774 |

**Table 2. Measures of fit for updated CFA model**

In this case, all the measures of fit are acceptable (GFI and AGFI > 0.95, SRMR < 0.05) so the data does support this CFA model. The p-value for the model is reported as 0, though this is likely sensitive to the issue that the original dataset consists of over 100,000 observations, thus we rely on the measures of fit more than the p-value for determining whether the model is confirmed. This analysis shows that given these two suggested factors, we fail to reject the null hypothesis that the restricted covariance matrix from confirmatory factor analysis is equal to the non-restricted covariance matrix.

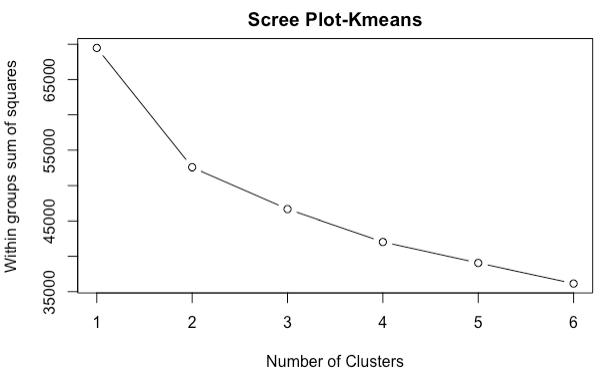
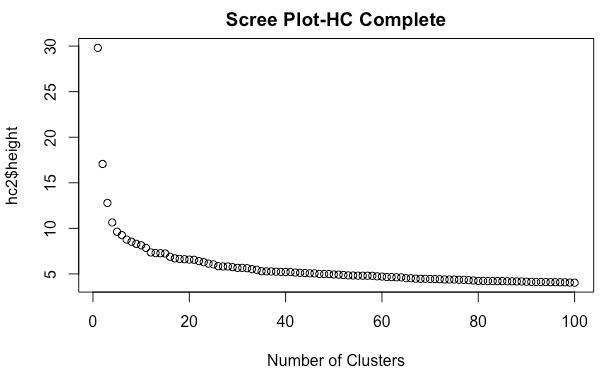
This model is represented by the two factors, interpreted as *Power* and *Vibe*. The overall power of a song is measured by things such as its loudness, energy and tempo, all of which are higher for higher power songs, or vice versa. The vibe of a song indicates its overall tone, represented by mainly danceability and valence as well as duration. Higher values for danceability and valence, and lower values for duration, correspond to a more positive vibe for the song, whereas the opposite is interpreted as more negative overall in tone.

# Cluster Analysis

*(This section is authored by Mychael Solis-Wheeler)*

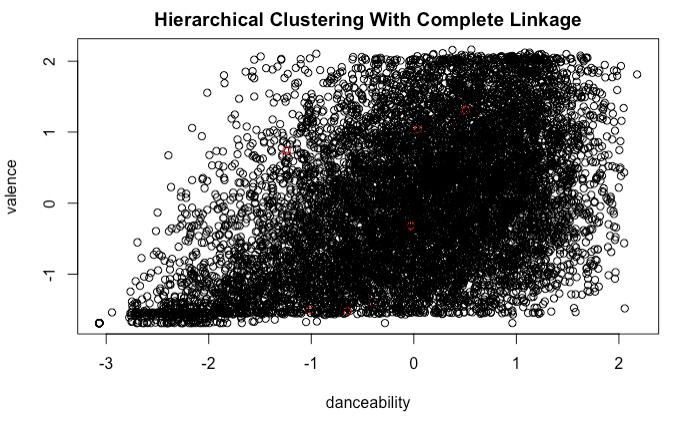
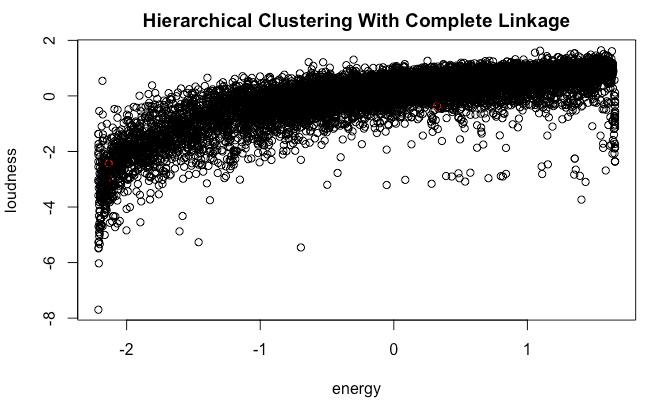
Since the original dataset had over 100,000 observations (n = 100,000), the memory needed to produce clustering results for all the data was too much for the hardware used in analysis. Therefore, a random sample of 10,000 (n = 10,000) was taken from the original dataset to reduce memory requirements. Scree plots, clustering using hierarchical, k-means, and model-based clustering techniques, along with their respective contingency tables, were all performed to best visualize and interpret the randomized sample of the original data.

First, scree plots were created for hierarchical clustering using 3 types of linkage: single, complete, and average, with the “elbow” being identified at 2 clusters across all linkage types. Additionally, when a scree plot was created for k-means clustering with an nstart of 10, the elbow was identified at 2 clusters. The scree plots of hierarchical clustering using complete linkage and k-means clustering are shown below in Figure 7.



**Figure 7: Scree plot for hierarchical clustering with complete linkage (left) and for k-means clustering (right).**

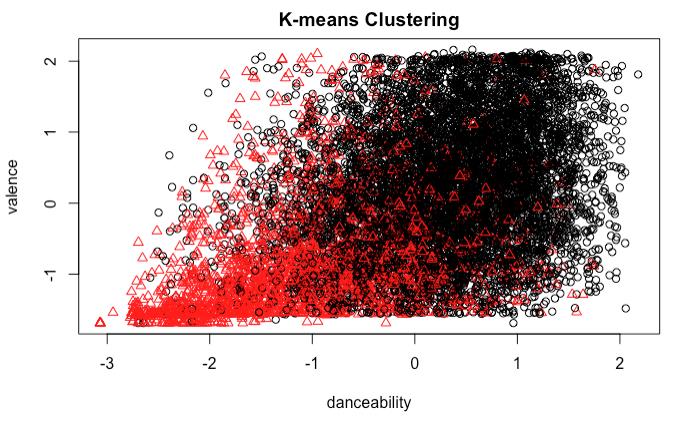
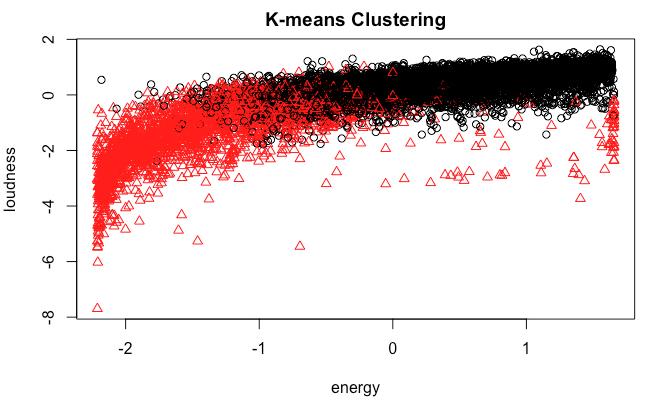
Since the highest correlation was previously found to be between energy and loudness and the second highest correlation was found to be between danceability and valence, exploring these pairs of variables for clusters was attempted first. There was extreme disproportionality between counts in the clusters for each hierarchical linkage type. Therefore, hierarchical clustering was not optimal for best interpreting and visualizing this data. For instance, the contingency table for complete linkage resulted in Cluster 1 having 9992 observations while Cluster 2 had only 8. This extreme clustering disproportionality is further highlighted visually with Cluster 1 (Black) and Cluster 2 (Red) shown (or not well shown in Red’s case) in Figure 8 below for the following pairs of variables: energy vs loudness and danceability vs valence:



**Figure 8: Hierarchical clustering with complete linkage (k = 2) for energy vs loudness (left) and for danceability vs valence (right)**

These extreme disproportionality trends were also similarly found when using single and average linkage. Therefore, hierarchical clustering was not the choice for cluster analysis in interpreting and visualizing this data.

K-means clustering had reduced disproportionality compared to hierarchical clustering. For instance, the contingency table for k-means clustering resulted in Cluster 1 having 7770 observations and Cluster 2 having 2230. Therefore, this clustering provided a better visualization with Cluster 1 (Black) and Cluster 2 (Red) shown below for the following pairs of variables (energy vs loudness and danceability vs valence):



**Figure 9: K-means clustering (k = 2) for energy vs loudness (left) and for danceability vs valence (right).**

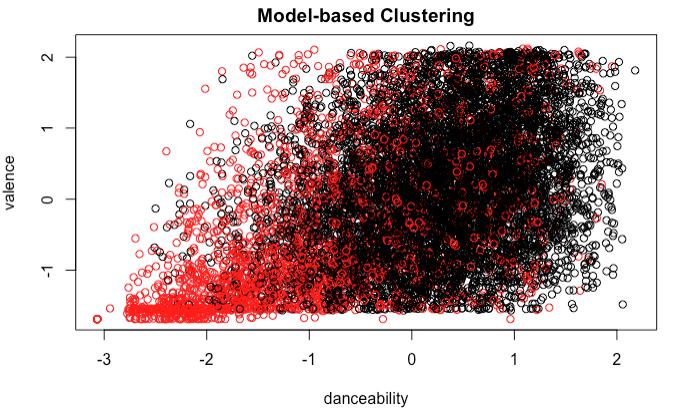
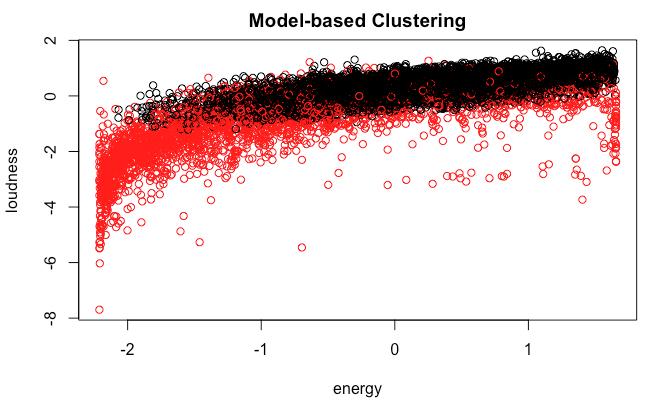
Based on visualizing the clusters in energy vs loudness, one can interpret that there are two types of songs: those that have high loudness and energy and those that have low loudness and energy. Additionally, when visualizing the clusters in danceability vs valence, one can interpret that there are also two types of songs: those that are highly danceable and make you feel good and those that are not danceable and convey a negative mood, such as sadness or anger. Based on the cluster centroids across all 7 variables, the differences between those clusters of songs are further highlighted below in Table 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | danceability  centroid | duration\_ms  centroid | energy  centroid | loudness  centroid | tempo  centroid | valence  centroid | popularity  centroid |
| Cluster 1 | 0.286 | -0.045 | 0.383 | 0.402 | 0.141 | 0.243 | 0.115 |
| Cluster 2 | -1.039 | 0.111 | -1.279 | -1.342 | -0.479 | -0.782 | -0.350 |

**Table 4: Cluster centroids of k-means clustering (k = 2) for all variables.**

There is a trend in the centroid values for Cluster 1. Most are positive (the exception being duration\_ms). Cluster 2 has mostly negative centroid values. This indicates that Cluster 1 songs are slightly above average in all 7 characteristics and Cluster 2 songs are below average, particularly in variables of danceability, energy, and loudness.

The final clustering analysis technique used was model-based clustering. Model-based clustering also had reduced disproportionality compared to hierarchical clustering. For instance, the contingency table for model-based clustering resulted in Cluster 1 having 7651 observations and Cluster 2 having 2349. However, because of how model-based clustering works, Cluster 1 (Black) and Cluster 2 (Red) overlapped as shown below using the following pairs of variables (energy vs loudness and danceability vs valence):



**Figure 10: Model-based clustering (k = 2) for energy vs loudness (left) and for danceability vs valence (right).**

Interpretations similar to those of k-means clustering can be made by viewing the plots in Figure 10. Based from the cluster centroids across all 7 variables, the differences between the model-based clusters are highlighted below in Table 5.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | danceability  centroid | duration\_ms  centroid | energy  centroid | loudness  centroid | tempo  centroid | valence  centroid | popularity  centroid |
| Cluster 1 | 0.27 | -0.081 | 0.293 | 0.403 | 0.093 | 0.166 | 0.149 |
| Cluster 2 | -0.92 | 0.222 | -0.9 | -1.257 | -0.291 | -0.48 | -0.435 |

**Table 5: Cluster centroids of model-based clustering (k = 2) for all variables.**

There is still a trend in the centroid values for Cluster 1: most are positive (the exception being duration\_ms). Cluster 2 has mostly negative centroid values. Just with k-means centroid values, these centroid values also support that are two distinguishable clusters of songs, particularly considering energy, loudness, danceability, and valence.

Unfortunately, there is an issue with this method. The model-based clustering performed using the Mclust function in R assumes that the data used as input is approximately multivariate normal. However, as discussed earlier in the report, this Spotify data is not multivariate normal, so using the function Mclust is not appropriate here.

K-means clustering was the best method for discovering clusters in the data. Hierarchical clustering produced clusters too disproportionate in size. Model-based clustering using Mclust did not produce reliable results since the data was not approximately multivariate normal. K-means clustering produced two clusters with clear interpretations from the centroids: songs which were slightly above average in all characteristics and songs which were below average, especially in danceability, energy, and loudness.

# Conclusion

*(This section is authored by Mychael Solis-Wheeler)*

## Applications of Results

From the analysis techniques conducted on this dataset, the result suggest that songs that tend to be loud also tend to be high-energy. Also, songs that are danceable or high in danceability tend to generally make one feel good or have a high valence. This may indicate that moving to the rhythm of a beat of a song will promote oneself towards feeling more positive than negative.

Finally, songs do not need to be fully described by all 7 variables used from this dataset. Rather, simply having 2 variables that may be a combination of some of the previous 7 variables could best explain or represent the interpretation of the result. Those 2 variables can be *Power* and *Vibe*.

## Lessons Learned

There were many lessons learned from analyzing this dataset of Spotify song characteristics. First, ideally, assumptions are met for certain analysis techniques to efficiently give results from the data interpreted. But the reality is that most real world data available will be unedited, unfiltered, have missing values, or even missing variables. In other words, real world data is messy, which may not give the clean results and outputs desired. This was an initial challenge for our analysis team, especially in the reduction process of this big dataset. For instance, in the clustering portion, the memory needed to run clusters of over 100,000 observations from the original dataset was not available and thus, not possible to run the R programming. An alternative solution was needed to reduce the size of the data while still maintaining the integrity of the dataset. Therefore, a random sample of the data of just 10,000 observations was conducted to make clustering with the available hardware a feasible task.

Also, a balance must be maintained between cleaning and completeness of the data. Too much cleaning may lead to time wasted and delays in finding interpretations and value from analysis. Worse still, it could lead to biases in the results because of the “fixes” used to clean the data. Too little cleaning may lead to difficulties when analyzing the data; the data may just end up being cleaned along the way, which is less efficient than doing it pre-analysis and could even cause propagating errors. Therefore, good judgement is necessary to navigate through that balance.

Furthermore, sometimes judgement calls are needed when interpreting or visualizing data. These judgments become more subjective the messier the data is. Visualizing data allows for more mistakes in interpreting the data if those visualizations do not have a solid quantitative foundation. As the data becomes more complex, such as by increasing the number of observations or variables, then analysis becomes harder to perform and interpret. Therefore, one could consider data reduction techniques to lessen data complexity.

# References

[1] Tao Li, et al., editor. "Hit song science." *Music data mining* (2012): 305-326.

[2] Wang, Kedao. "Predicting Hit Songs with MIDI Musical Features."

[3] Spotify. “Spotify Audio Features.” *Kaggle.com*, 31 Mar. 2019, <https://www.kaggle.com/tomigelo/spotify-audio-features>

[4] Statistical tools for high-throughput data analysis (STHDA) ggplot2 : Quick correlation matrix heatmap - R software and data visualization [Source code]. <http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization>.

# Appendix A: Variable Definition

Table A1 shows the definitions for all variables in the full Spotify data set.

|  |  |
| --- | --- |
| **Name** | **Description** |
| artist\_name | Name of the artist of the track |
| track\_id | Spotify ID of the track |
| track\_name | Name of the track |
| acousticness | A confidence measure from 0.0 to 1.0 of whether the track is acoustic, where 1.0 represents high confidence |
| danceability | Measures how suitable a song 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. |
| duration\_ms | Duration of the track in milliseconds |
| energy | 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. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. |
| instumentalness | Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. |
| key | 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. If no key was detected, the value is -1 |
| liveness | Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. |
| loudness | The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). |
| mode | Mode indicates 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. |
| speechiness | Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. |
| tempo | The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. |
| time\_signature | 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). |
| valence | A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. 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). |
| popularity | The popularity of the track: a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. |