Exploring Music Trends and Popularity Factors

2023-05-25

Names: Samuel Warren, Anshul Sadh-Gauri

1. Project Overview.

Introduction

Songs on music platforms often emerge out of nowhere, when it comes to popularity. Is there a reason why songs like Old Town Road became as popular as it did? While some artists will command a few million plays regardless of their release, other newer artists look for just one of their songs to get popular. We will be investigating data of different songs throughout history in search of commonalities between popular songs, and if any of these common characteristics have changed throughout the years. The data has certain listed factors that will help answer our guiding questions.

Our guiding questions are:

- How do factors such as Genre, BPM, Length, Decibel level, and Speechiness effect a songs popularity?
- Are certain levels of these characteristics more common at specific times in the past?

Hypothesis: Starting from 1956, as we get closer and closer to 2019, we will see a decrease in a songs length, a increase in BPM and a constant decibel level.

This project covers at most pretty basic music ideas. There is no in depth musical concepts and as such all readers should be able to understand the project in its entirely.

The only thing that could be of some confusion is the actual variables in the data sets such as Decibel level. These variables will be explained in the following section.

2. Data and Resources Used.

Our project contain 1 data set that contains data and common characteristics of songs. It contains roughly 2000 of pretty popular songs spanning the years 1956-2019. This data set will be named SP2000.

SP2000 was taken from a user on kaggle. This user compiled 2000 songs and ran them through a Spotify API. This API returns characteristics for each song inputted. These characteristics can be seen below. This was done last as of 2019, as such it only includes songs from 2019 or earlier.

In SP2000, there were originally 15 columns. We decided to remove 8 of these columns and Add one column

Top.Genre was changed so that it only included basic Genres. For example alternative rock and classic rock are both included in 'rock'. Adding on, certain genres that didn't fit into a general genre were filled into the other column, excluding 'dutch cabaret' since it had 51 songs with it as its genre.

The added column is a numeric value starting at 1 and ending at 63 that corresponds to a year starting at 1958 and ending at 2019. This is used so that a Linear Model can pick a more general year value, instead of having to select an exact year.

Lastly we correctly assigned each variable to a numerical or categorical data

- **Index**: This variable is a unique value for each row in the data set.
  - Data Type: factor
  - Range/Levels: 1-1994
- **Title**: This variable is the title of the song.
  - Data Type: categorical
  - Range/Levels: unique title
- **Artist**: This variable is the artist of the song.
  - Data Type: categorical
  - Range/Levels: unique artist
- **Top.Genre**: This variable tells the genre of music the song is. The genre has been generalized.
  - Data Type: numeric
  - Range/Levels: adult standards, country, Dutch cabaret, electric/dance, folk, funk, g funk, hip hop, indie, jazz/blues, metal, other
- **Year**: This variable records the year when the song was released.
  - Data Type: categorical
  - Range/Levels: 1956-2019
- **Beats.Per.Minute..BPM**: This variable measures the pace of a song/how fast a song is. Specifically, it records the amount of beats a song has in one minute.
  - Data Type: numeric
  - Range/Levels: 37-206
- **Loudness..db.**: This variable measures the loudness of a song. This is in a ratio comparison to other songs on Spotify. In this instance, a dB level of -2 represents the loudest in our data set, and -27 represents the lowest dB level.
  - Data Type: numeric
  - Range/Levels: -27-(-2)

- **Speechiness**: This variable records the amount of words in a song. This amount is then compared as a ratio to the other songs on the playlist.
  - Data Type: numeric
  - Range/Levels: 2-55
- **Popularity**: This variable records the popularity of a song. This is calculated by combining a few different factors such as the number of plays, user engagement (shares/likes), and the percentage of the song listened to on average. The final value is a ratio from 1-100.
  - Data Type: numeric
  - Range/Levels: 1-100
- **Length..Duration**: This variable represents the length of a song in seconds.
  - Data Type: numeric
  - Range/Levels: 93-966
- **YearNumber**: This variable corresponds to the year category. This is simply for use with models and plots.
  - Data Type: numeric
  - Range/Levels: 1-63

3. Analysis.

SP2000

In SP2000, there were originally 15 columns. We decided to remove 8 of these columns and Add one column

Top.Genre was changed so that it only included basic Genres. For example alternative rock and classic rock are both included in ‘rock’. Adding on, certain genres that didn’t fit into a general genre were filled into the other column, excluding ‘dutch cabaret’ since it had 51 songs with it as its genre.

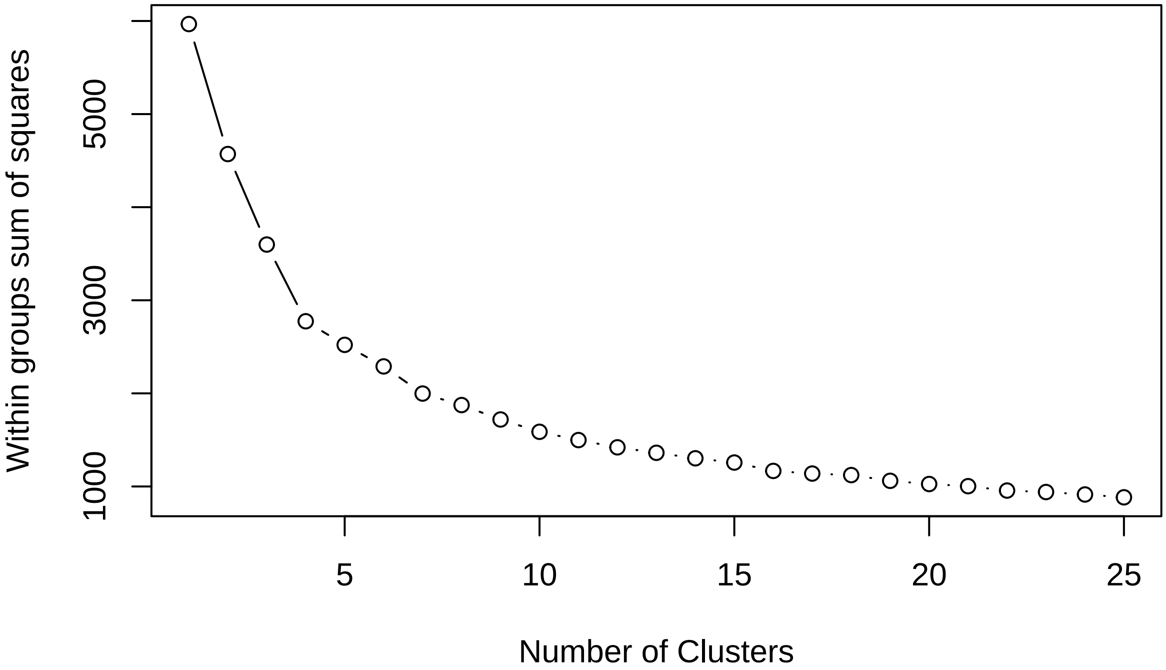
The added column is a numeric value starting at 1 and ending at 63 that corresponds to a year starting at 1958 and ending at 2019. This is used so that a Linear Model can pick a more general year value, instead of having to select an exact categorical year. As well its used in spliting up years for plots.

Lastly we correctly assigned each variable to a numerical or categorical data. Those variables can be seen above.

The first model we are going run is a cluster sampling with kmeans. The reason we want to do this is to figure out if the variables are grouped together by year. In other words if certain years have certain values for certain variables it means that there is some correlation between the two.

We can first create a dataframe with only the variables we want. These variables will be loudness dB, Length, and Beats per minute, represented by Loudness..db., Length..Duration, and Beats.Per.Minute..BPM. After creating it, we can scale the data to work with kmeans.

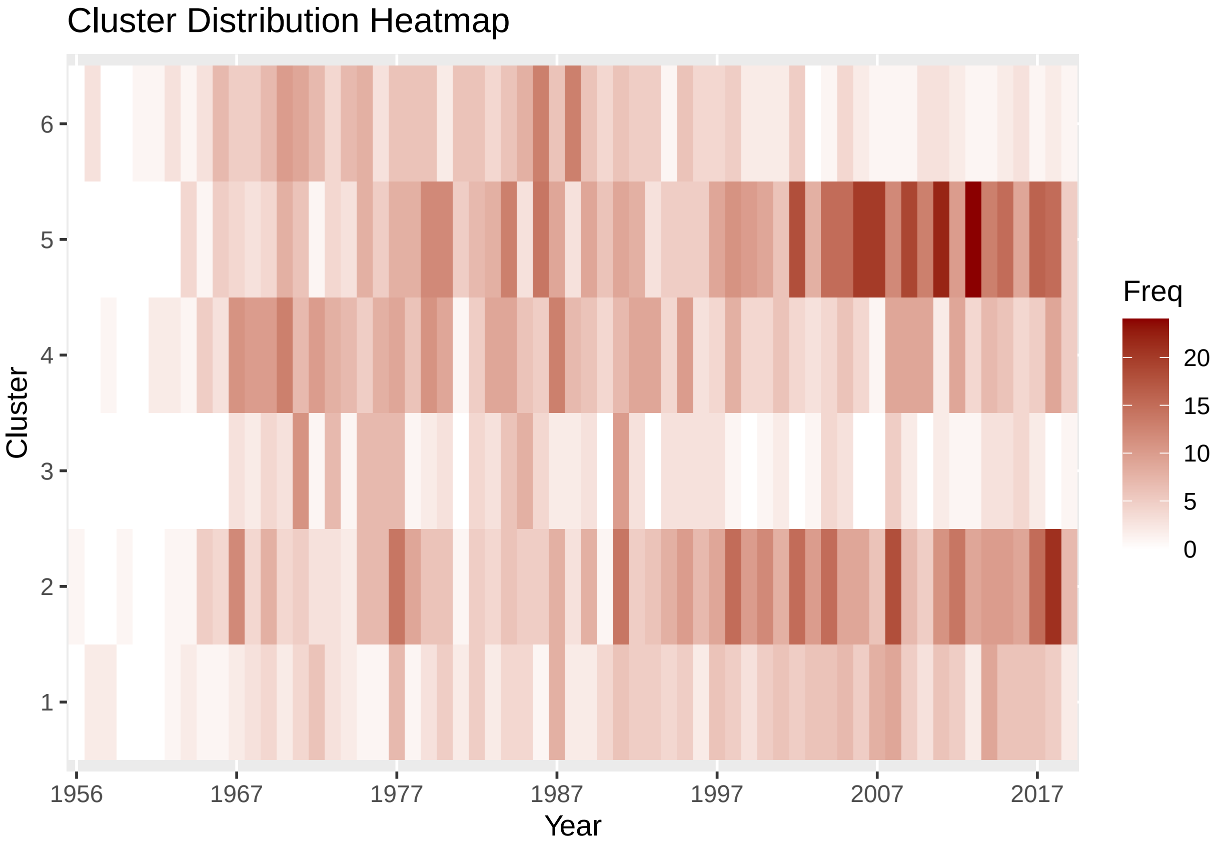
Next we can use a function to display a wss (Within sum of squares) plot to find the correct amount of clusters.



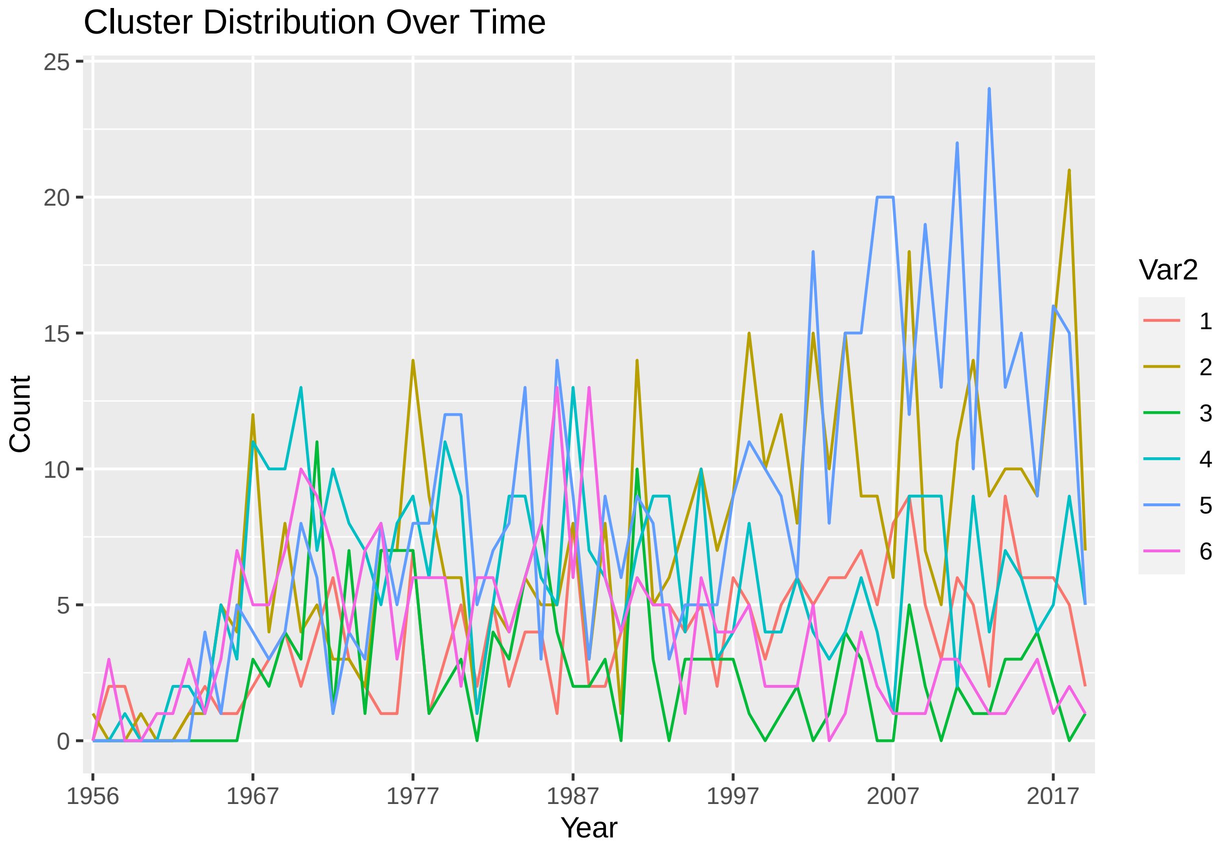
Based on the wss plot, its clear the correct amount of clusters is 6. Using this information we can use kmeans to cluster the data.

```
## [1] 240 453 152 371 522 252
```

After creating the clusters, each cluster has a few hundred songs in each, which is a good balance. Next, we are going to create a few plots to visualize the clusters and hopefully see some trends throughout them.



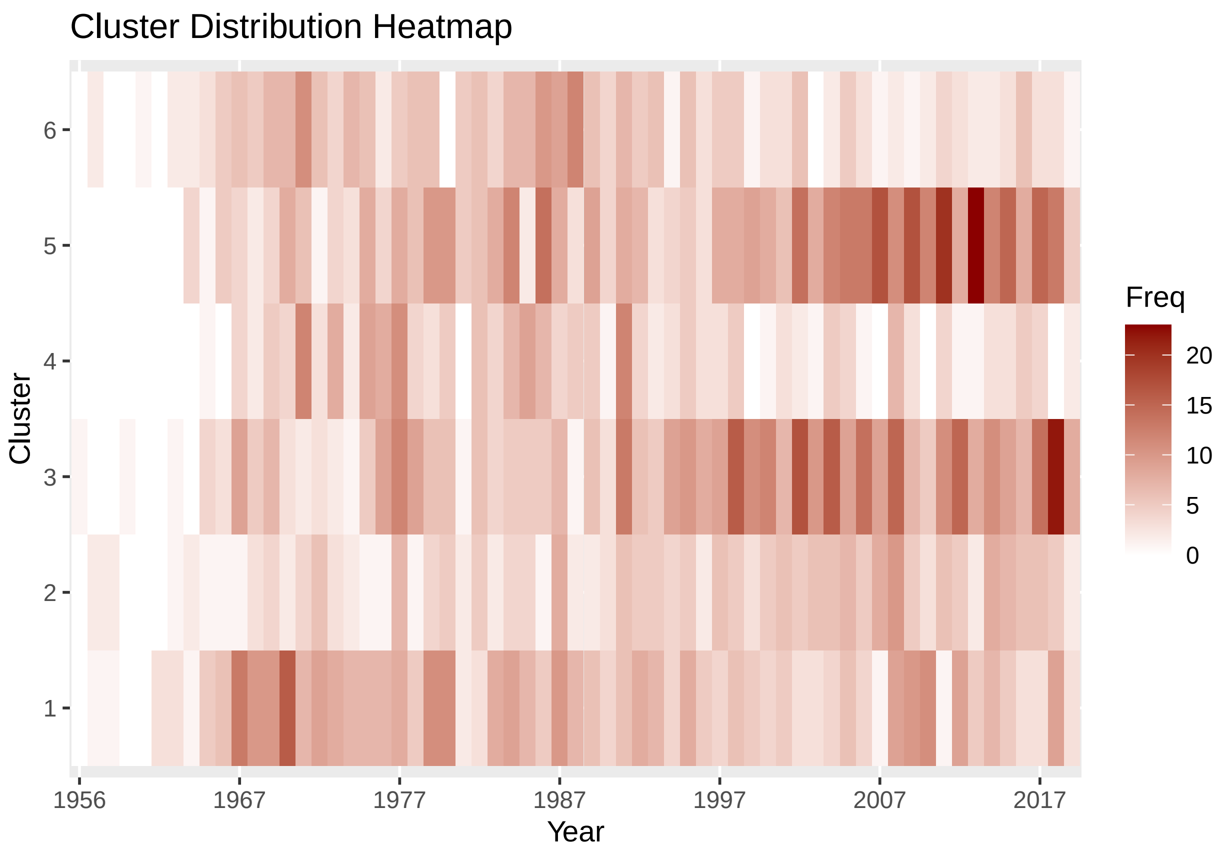
The heat map does a good job in showing the trends of each cluster. Each cluster has similar songs. The songs that make up each cluster will have similar values for BPM, Decibel Level and Length. It seems that overtime, 3 of the six clusters, Clusters 1, 2, and 5 increase in frequency as the years get more current. On the other hand clusters 6 , 3, decrease with time. The last one, cluster 4, stays relatively the same in frequency.



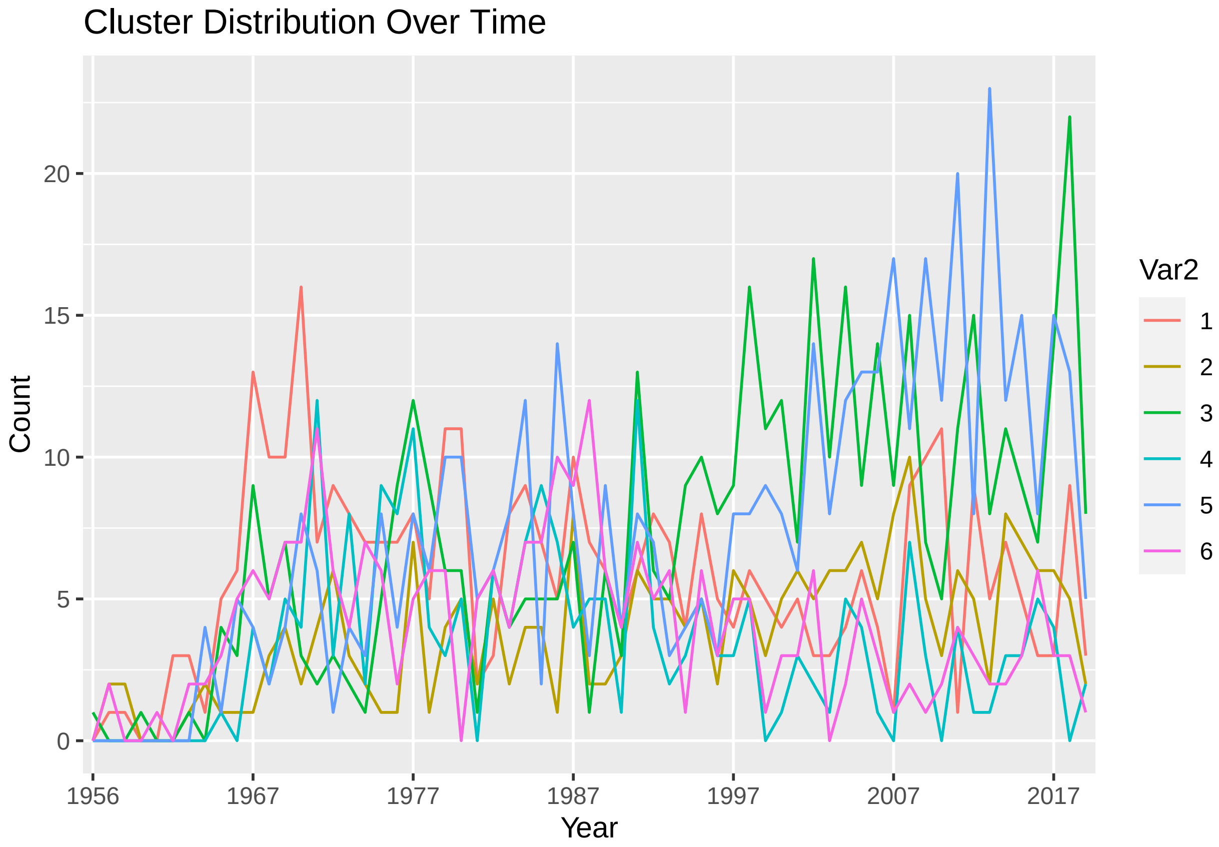
This line plot does a good job in visualizing the clusters as well. This plot provides a similar representation of the data as the previous plot. 3 of the clusters 1 , 2 and 5 are increasing with time, 2 clusters, clusters 3 and 6 are decreasing and the last one, cluster 4 remains constant.

Overall based on these two plots it is pretty clear that song variables or in other words the characteristics of songs change overtime. Its very unlikely that a cluster, not to mention 5, would center around a certain year without a connection to time. Since the clusters are concentrated over a set of time,

While kmeans works great in determining clusters of data, its important to use more then one model to account for error. Next we will be running a PAM clustering method.



This heat map shows a similar representation of the data compared to kmeans. There is again an increase in frequency as time gets closer to 2019, in 3 of the clusters. Of the other 3 clusters, 2 of them appear to decrease in frequency over time and the last stays relatively constant throughout.



Again, the line plot represents the data in a similar fashion to the plots above.

Overall the PAM and Kmeans clustering models both show the connection between time, and the certain values of the variables associated with Length, Decibel level, and BPM. All but one of the clusters is concentrated over a specific period of time, instead of being spread evenly throughout the years. As such songs that occur at the periods of time the clusters are concentrated at, are recognizable based on there BPM, length and decibel level.

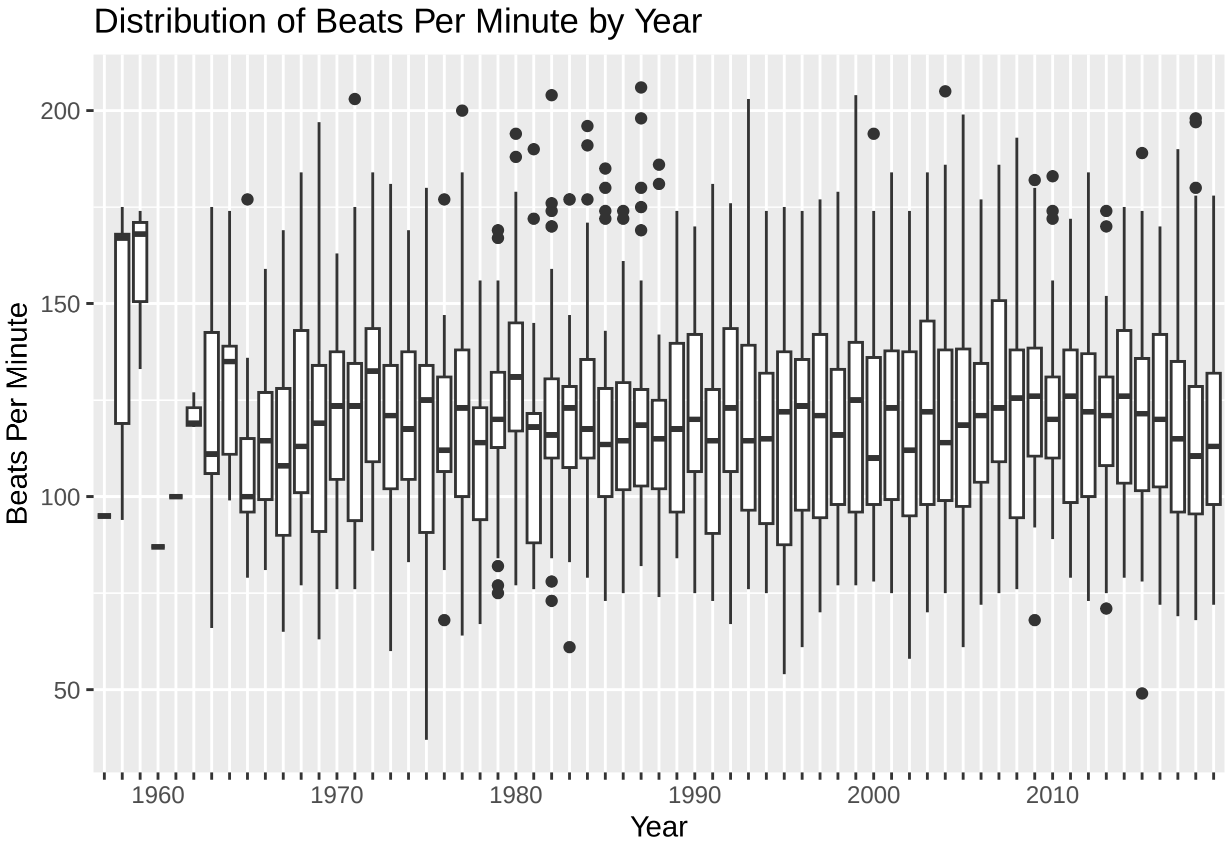
Now that we have established that certain levels of BPM, decibel level and length have specific values at certain times, we can run a linear model, predicting YearNumber, a numerical data variable that ranges from 1-63, with BPM, decibel level, and length as predictors individually.

First lets run the linear model with beats per minute as the predictor.

```
##                            Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)             37.24884224 1.59089296 23.4137954 2.824910e-107
## Beats.Per.Minute..BPM.   0.00654783 0.01288529  0.5081631  6.113954e-01
```

Since the p-value is 0.611, it is not significant. As such, over time, the beats per minute of a song stays constant.

Here is another plot that shows the average beats per minute over time.



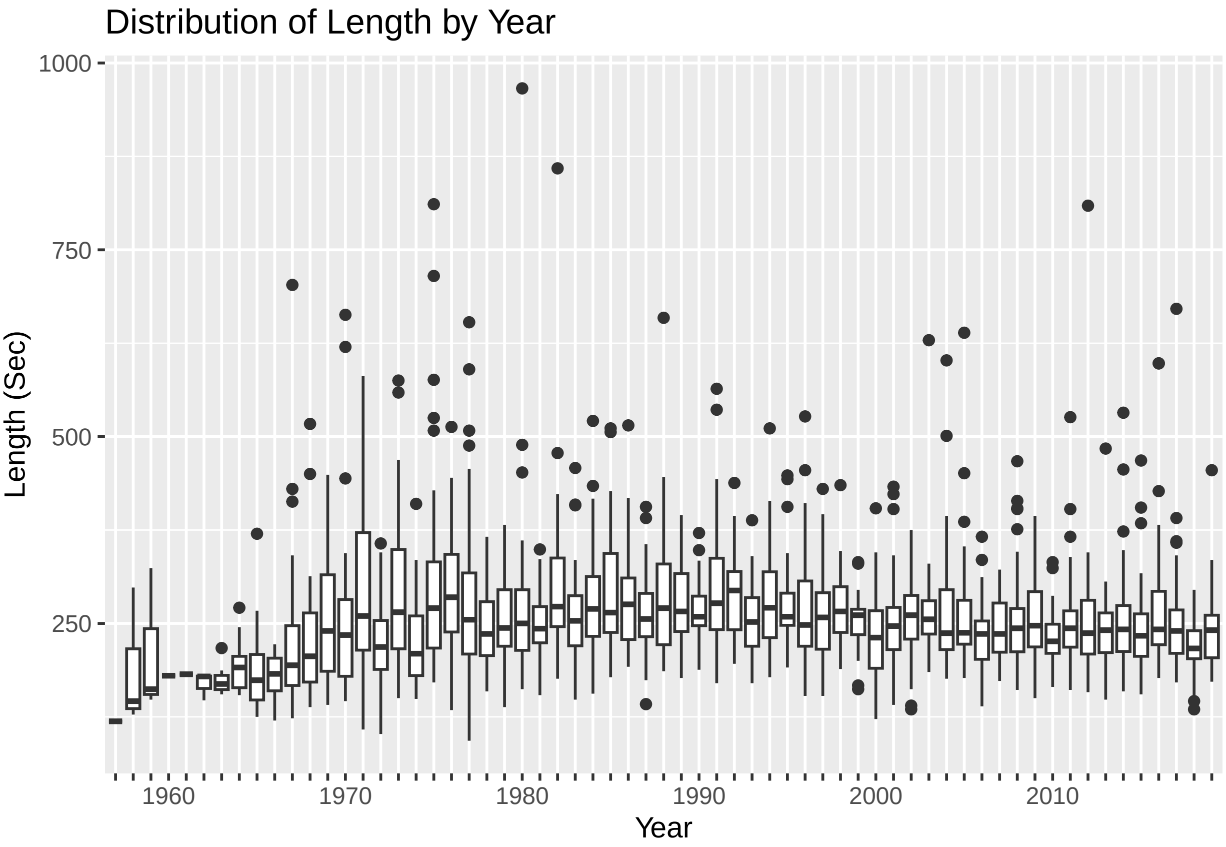
This plot confirms the statement above. Over time, the beats per minute of a song on average stays roughly constant.

Next lets run the model with Length as the predictor.

```
##                       Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)      39.270972174 1.21265369 32.384326 3.937336e-185
## Length..Duration -0.004742598 0.00444644 -1.066605  2.862795e-01
```

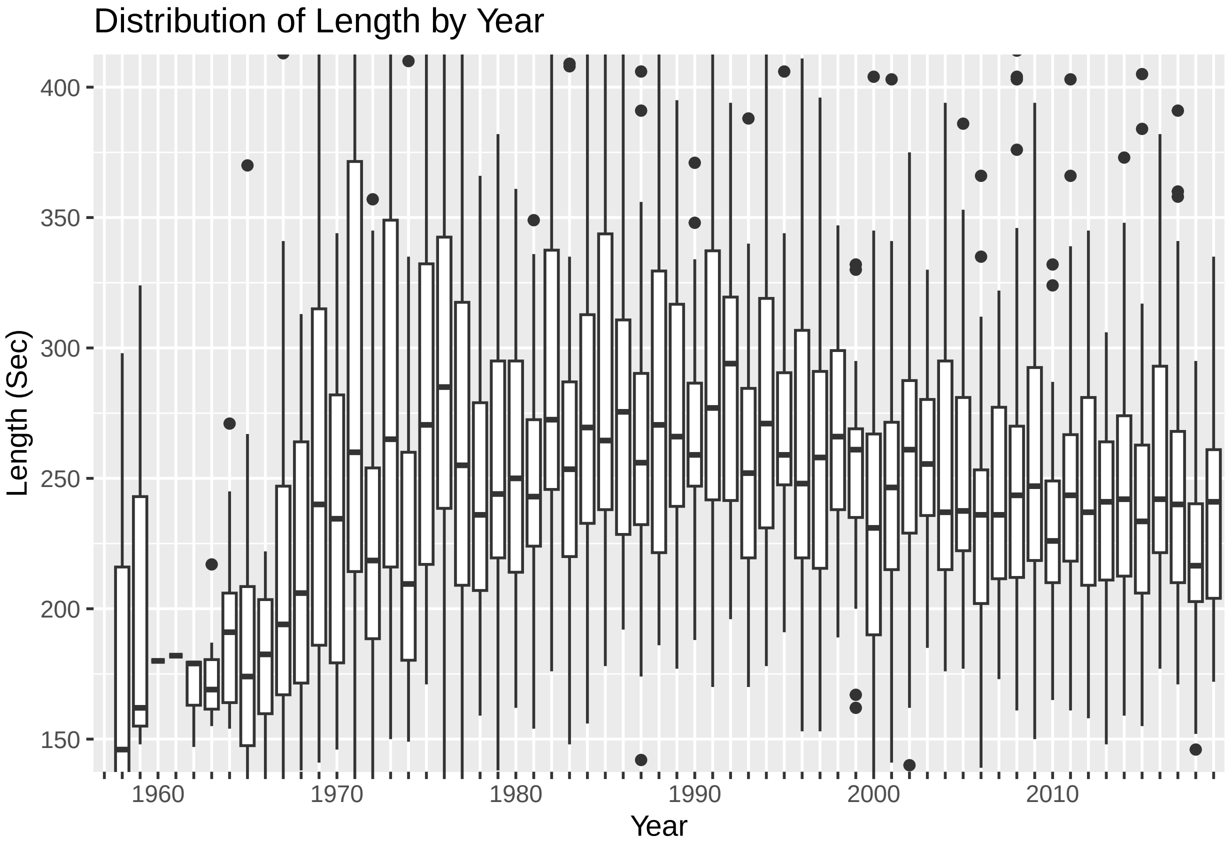
The p-value is greater than 0.05 at 0.286, and as such is not significant. Over time, the Length of a song has no correlation, and stays relatively constant.

Here is a plot to that shows the average length over time.



This plot seems to disagree with the initial conclusion from above. While the average length doesn't increase or decrease drastically after 1970. The average does peak during the 1980s and 1990s. The following years in the 2000s and 2010s show a slightly lower average length.

The following plot is closer zoomed it to be clearing in viewing the trends:



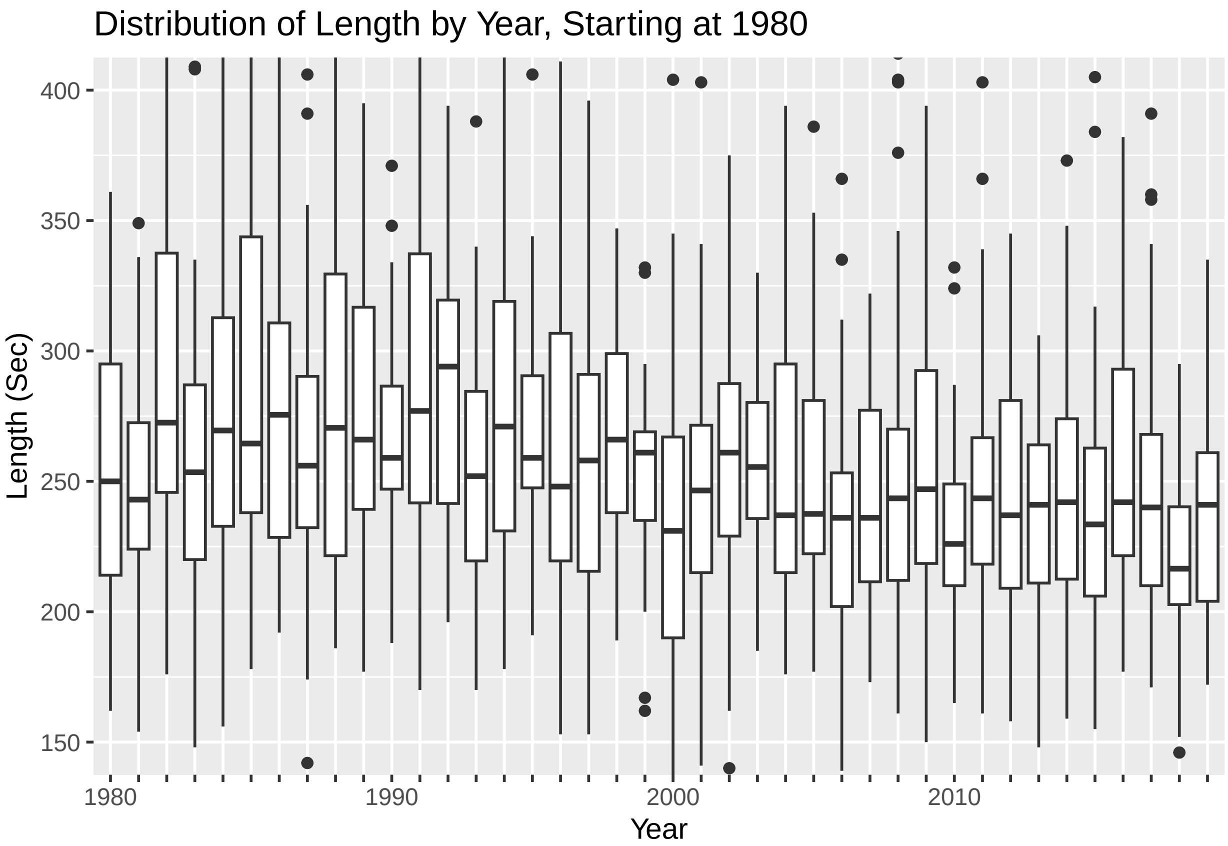
Its clear that the length of a song does change overtime. The song has an higher average length in the 1980s and 1990s, followed by a average decrease in length during the 2000s and 2010s. I believe the difference happened because the distribution of length by year is not linear. The length increases and decreases over the years 1956-2019.

Now if we take a linear model starting at 1980 and ending at 2019 we should see an overall decrease in length.

```
##                     Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept)       53.0920048 1.093605101 48.547693 7.287135e-308
## Length..Duration -0.0291609 0.004007705 -7.276208  5.562155e-13
```

Based on the summary the p-value is less than 0.05 at 5.562155e-13. Based on this, and since the coefficient is negative its clear that the length of a song has decreased since 1980.

Here's another plot to visualize it.

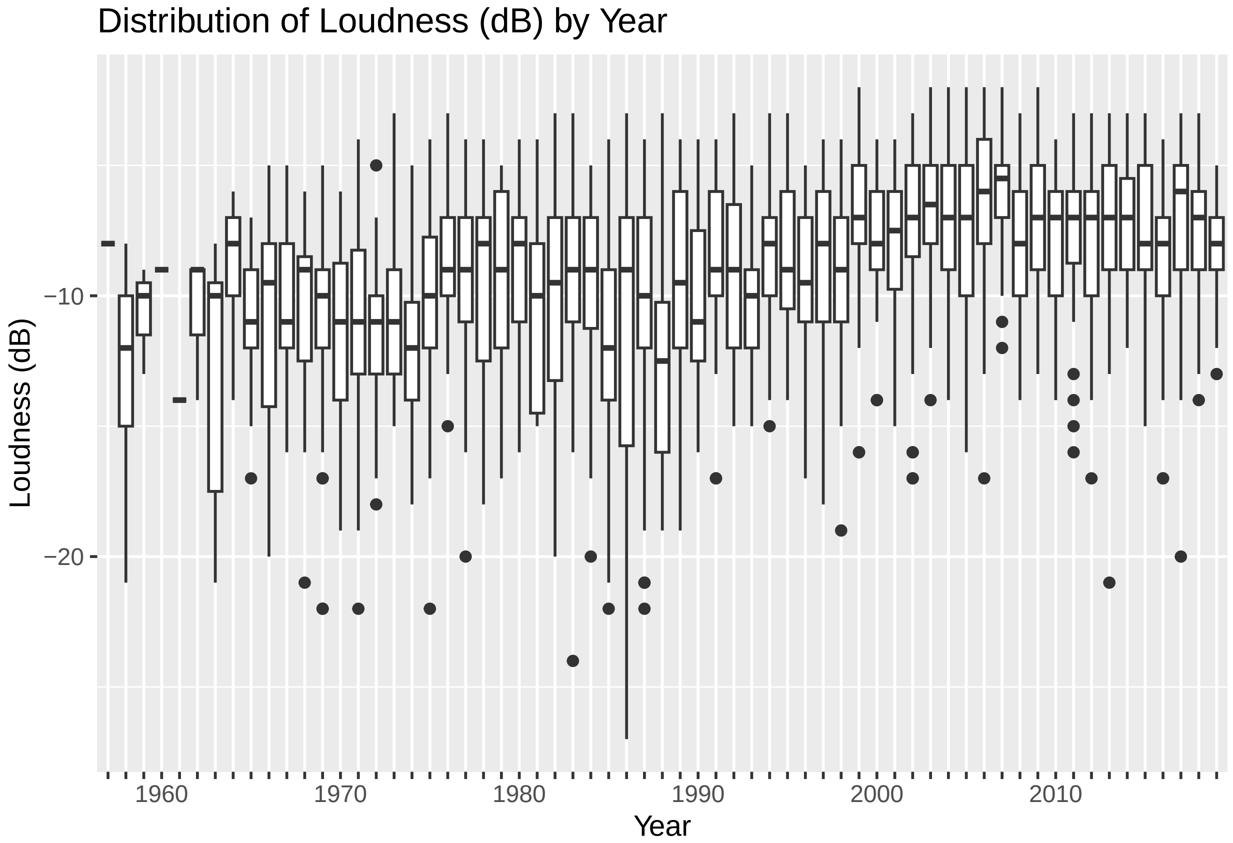


Finally lets take a look at decibel level over time.

```
##                  Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)   51.677111 0.90437058 57.14152 0.000000e+00
## Loudness..dB.  1.515405 0.09313468 16.27111 5.563034e-56
```

Decibel level appears to be significant. Its p-value is less than 0.05 at 5.563034e-56. As such, since the coefficient is positive, over time the decibel level of a song increases.

Here is plot that confirms our conclusions.



Besides a few outlier years around the late 1980s, the decibel level of a song increases over time, as we get closer to 2019. Ignoring outliers, the lowest point happens around the early 1970s, and the peak around from 2006 until 2019

The plots above helps us answer our guiding question: Are certain levels of these characteristics more common at specific times in the past?

Overall based on all of the plots above its clear that some of the variables change overtime. Beats Per Minute stays roughly constant over time. The Average value per year hovers around 120 BPM.

The length of a song changes overtime. It originally starts at roughly 180 seconds or 3 minutes. This is followed by an increase until the 80s and 90s where it peaks at around 270 seconds 3 minutes 30 seconds. After, there is a decrease in average length to around 250 seconds today or 4 minutes and 10 seconds.

Finally, the decibel level increases constantly over time. It average in 1956 is around -10 dB on the Spotify API scale. This is followed by a steady increase in dB to when it peaks out at around -7 in 2006. This level at 2006 has stayed constant to 2019.

Next we are going to run a model that predicts a songs popularity based on the variables Length, BPM, Decibel level, Genre, and Speechiness

```
##                            Estimate  Std. Error      t value      Pr(>|t|)
## (Intercept)             73.05778183 2.131528712  34.27482886 2.358493e-202
## Beats.Per.Minute..BPM.  -0.01404656 0.010421775  -1.34780867  1.778748e-01
## Loudness..dB.            0.54058674 0.084093776   6.42837989  1.613340e-10
## Length..Duration        -0.01688194 0.003668022  -4.60246511  4.441195e-06
## Top.Genrecountry        -9.49558846 3.897984542  -2.43602517  1.493770e-02
## Top.Genredutch cabaret  -16.91741423 2.150154211  -7.86800042  5.890892e-15
## Top.Genreelectric/dance  -0.56236338 1.911062132  -0.29426745  7.685845e-01
## Top.Genrefolk             1.76826613 3.113659395   0.56790609  5.701634e-01
## Top.Genrefunk            -4.55443393 3.795984021  -1.19980324  2.303600e-01
## Top.Genreg funk          -0.51770577 5.928035039  -0.08733177  9.304167e-01
## Top.Genrehip hop         -2.69074905 2.654318929  -1.01372485  3.108385e-01
## Top.Genreindie          -26.10293926 1.890463472 -13.80769301  1.827147e-41
## Top.Genrejazz/blues      -2.87659287 4.238194362  -0.67873076  4.973882e-01
## Top.Genremetal            1.71888107 1.845187600   0.93154814  3.516842e-01
## Top.Genreother          -15.53353776 1.945530152  -7.98421846  2.379591e-15
## Top.Genrepop             -4.89932517 1.349928396  -3.62932225  2.914072e-04
## Top.Genrerock            -1.88553898 1.266771676  -1.48846001  1.367896e-01
## Top.Genresoul             2.90145051 2.288703338   1.26772678  2.050453e-01
## Speechiness               0.27354448 0.069608971   3.92973032  8.797449e-05
```

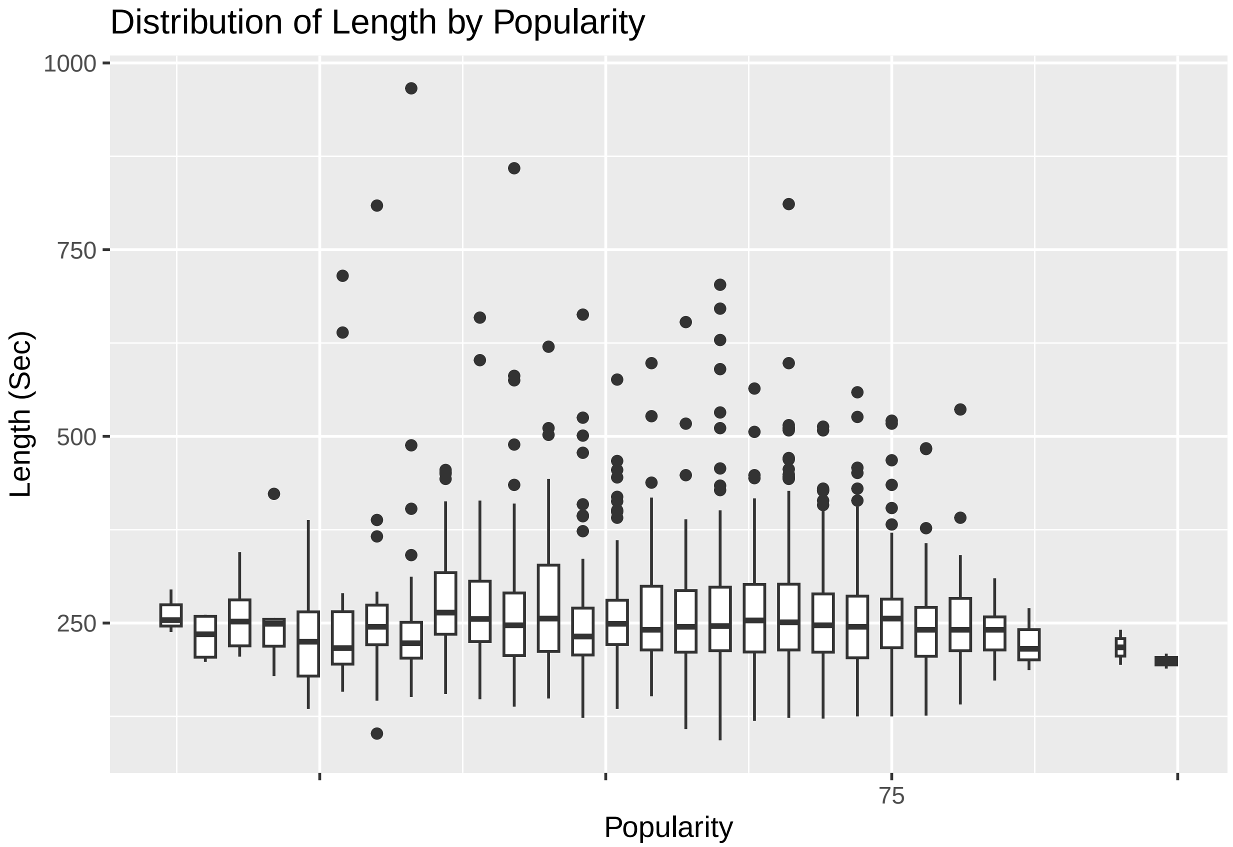
After running the model, not all of the variables are significant and as such, do not contribute to popularity. BPM is not significant. When looking at Genres, its clear that only certain ones seem to impact the overall popularity. It seems that Genre is not significant since the vast majority of genres are above the p-value and the ones that are below are a low sample size. The only outlier to this is the pop category. Its safe to say that it is the most popular without even needing to run a model. Pop makes up 435 of the 1990 songs. Its popularity stands out more than the other categories.

Next we run the model over just those 3 variables. Also we are going to plot 3 graphs comparing the variables to popularity individually.

```
##                      Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)      65.740542674 1.382291147 47.559114 0.000000e+00
## Loudness..dB.     0.599013824 0.087472882  6.847995 9.950048e-12
## Length..Duration -0.008782697 0.003894982 -2.254875 2.424968e-02
## Speechiness       0.297218431 0.072302465  4.110765 4.104014e-05
```

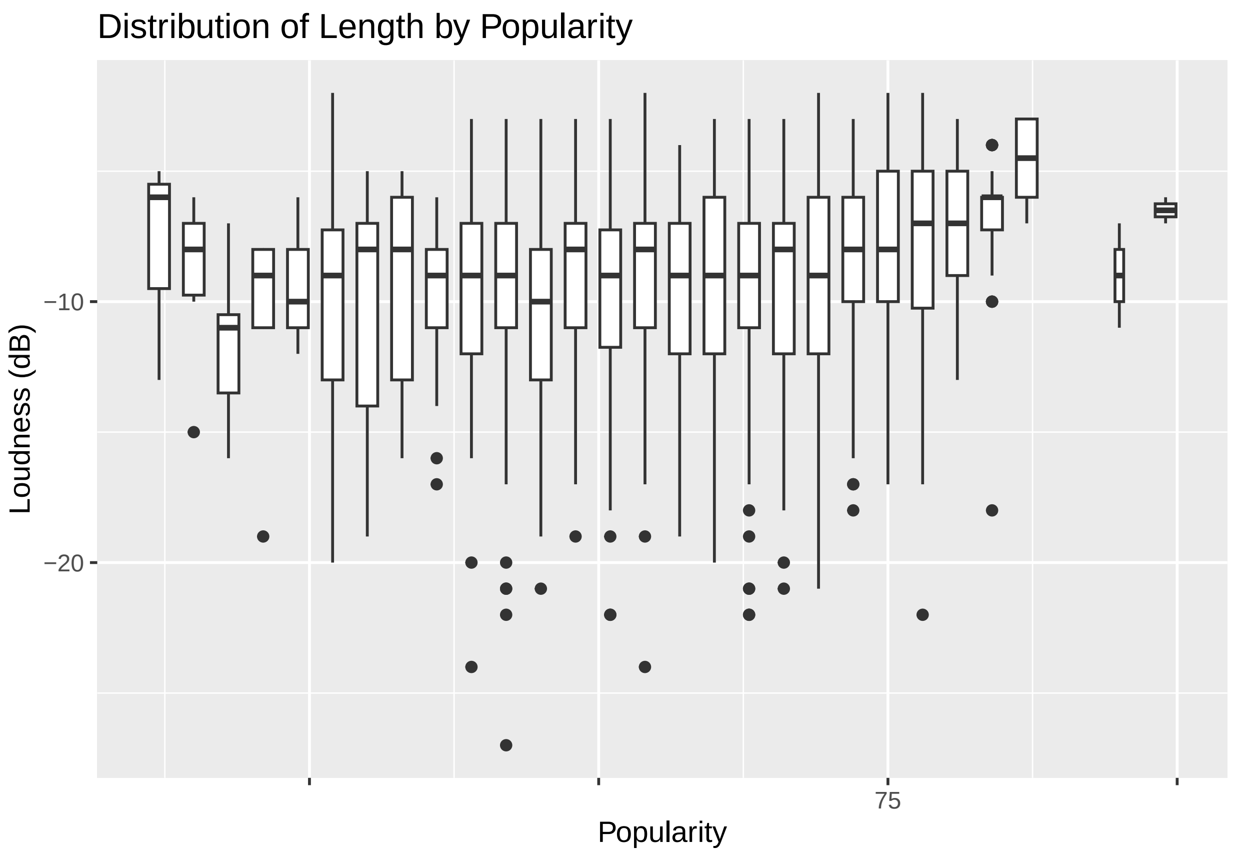
Based on the new model, all of the variables are significant with a p-value below 0.05.

Here a plot of Length versus Popularity

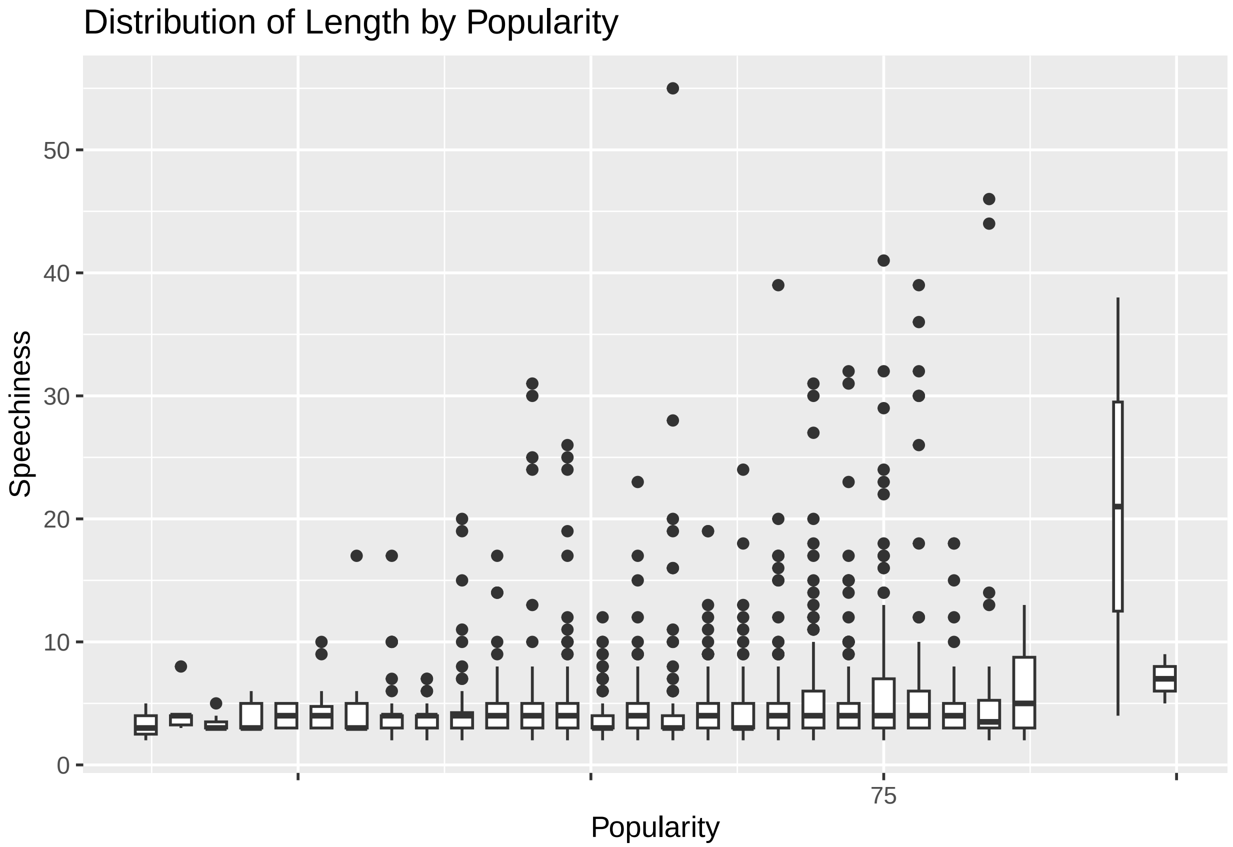


Here a plot of Loudness versus Popularity

Distribution of Length by Popularity



Here a plot of Speechiness versus Popularity



Based on the model and the two plots its clear that Length has little to no effect on popularity. On average a song that is 8 minutes long is less popular, than one that is two minutes. That being said its hard to confirm this, since the vast majority of songs sit below 4 minutes (240 secs), and the number of songs that are that long are low. The p-value for length is less than 0.05, which is significant. That being said, the coefficient associated is only -0.008782697. This means for every extra second the popularity goes down by less than 0.01 on a scale of 0-100. Overall a length of a song, does not seem to effect its popularity based on this data set.

On the other hand decibel level seems to have a slight impact on popularity. While there a large number of songs near -5 decibels. I still think that the positive slope has some significance. The p-value is extremely significant at 9.950048e-12. The coefficient is also pretty big at 0.599013824. Since all of the decibel values are negative it means that the louder a song gets, the more popular it is on average. Songs the average of songs that have a popularity ratibg above 75 are greater than those that are less by a small margin. Overall based on this data set it can be said that the louder a song is the more popular it its.

Lastly the Speechiness value seems to effect Popularity somewhat significantly. The Speechiness value hovers around 4-5 on average for most of the popularity levels. When popularity is very high, the Speechiness value seem to have lots of outliers. Lots of the data points around or above 75 popularity are above the average Speechiness significantly at 20 or 30. Overall I believe that Speechiness has a small impact on Popularity, that is as Speechiness increases the popularity of a song increases. Its important to be said that the vast major of songs have a low Speechiness value. In total it contribute to popularity somewhat significantly.

Summary and Conclusions.

Throughout the data exploration many things were learned. Our guiding questions: How do factors such as Genre, BPM, Length, Decibel level, and Speechiness effect a songs popularity? and Are certain levels of these characteristics more common at specific times in the past? were able to get answered. First off its clear that only some of those variables effect popularity. Based on the models, Decibel Level, a songs genre, and Speechiness effect a songs popoularity. While Decibel level and Speechiness can give a popularity a minor boost changing a genre to a pop or indie style can really help your songs gain attention. If I were to create a song it would be a wordy, average paced at 120 BPM, louder pop song. At the same time these are really the building blocks for a "perfect song" there is still much to explore like popular lyrical topics, but this is a good start. The second guiding question was answered also. Beats Per Minute stays roughly constant over time. The Average value per year hovers around 120 BPM. The length of a song changes overtime. It originally starts at roughly 180 seconds or 3 minutes. This is followed by an increase until the 80s and 90s where it peaks at around 270 seconds 3 minutes 30 seconds. After, there is a decrease in average length to around 250 seconds today or 4 minutes and 10 seconds.Finally, the decibel level increases constantly over time. Its average in 1956 is around -10 dB on the Spotify API scale. This is followed by a steady increase in dB to when it peaks out at around -7 in 2006. This level at 2006 has stayed constant to 2019. There is tons of data that can uncovered on with the Spotify API and even more history of songs to use.

If I were continuing the project even more I would look for an even bigger data set. By adding more music you can make the conclusions even more concrete. Also as said above I would look more into lyrics of music and actual musical notes to see commonalities.

References

These are links to where we found our data set. The user on Kaggle was doing a similar project, but they focused more on evaluating the Spotify API, and created music recommendation software. Our project was focused more of the trends of the actually data sets.

Spotify 2000 songs: https://www.kaggle.com/datasets/iamsumat/spotify-top-2000s-mega-dataset

Most 100 Most Streamed as of 2019: https://www.kaggle.com/datasets/pavan9065/top-100-most-streamed-songs-on-spotify

Link to the actual csv data in a google drive folder: https://drive.google.com/drive/folders/1F-k5S_4IjiKfnF-4zMvNsApFTYQHq0Ij?usp=sharing

All of out methods can be found within the r documentation, or within the labs/ class activities. Some outside documention was used. Mainly it was to help display the plots with ggplot2.

Here are links to the websites

https://ggplot2.tidyverse.org/

https://www.analyticsvidhya.com/blog/2022/03/a-comprehensive-guide-on-ggplot2-in-r/#:~:text=We%20use%20the%20fu

https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples/

Chat GPT was also used when questions arose about using ggplot, or how to interpret models/plots.