Spotify: Top 200 Songs from 2020-2021

By Katie Foster, Maggie Lewis, Jack Roberts, Anna Raditz

Dr. Eynon

CSC-272 Intro to Data Mining

11 December 2021

Project Overview:

In a world of constant media through applications like TikTok, Instagram, and music in television commercials, songs are constantly released and their popularity ever-changing. Our hypothesis is that there might be a way to predict what makes a song popular by analyzing certain attributes like time of year, release date, genre, and danceability of a song. We take demographics and geographic location out of the analysis to see if there's a way to predict a song's success in World's Top 200.

Section 1: Project Objective

In our project, we analyzed the top charted songs of 2020 and 2021 and took note of when and how long a song was on the top 200 list and its release date. Along with that, we created a model using Weka to see if there was a correlation in attributes to see if we could predict why a song was popular and what attributes might make a song popular during a certain season. While the models we created struggled to accurately find correlation between certain attributes of popular songs, we were able to find some relationship between the top streamed songs vs the release season of the song and top streamed songs vs the number of genres a song has. In this report, we use Spotify top charted data to make these predictions, along with discussing how we acquired our data, analysis and what we needed to do to prepare it for analysis, along with discussing our results and making conclusions based off of these results.

Section 2: Data and Data Sources

Our data has been sourced from Kaggle, an online resource containing thousands of template data sets for statistical practice and research. The authors acknowledge spotifycharts.com and Spotify Python Library for allowing this data to be compiled. This data set falls under the group level of analysis, because it includes data on over 1500 unique songs with hundreds of artists. The data itself is of fairly high quality, with all but 11 out of 1556 songs having complete sets of 22 variables, however the same cannot be said for the quality in which the data collection methods have been communicated or the usefulness of this data for our intended purposes. Firstly, the authors do not explain their collection methods in any significant detail. Other than mentioning the two sources listed above and brief descriptions of each

variable, we were left completely guessing as to how the data has been collected and interpreted by the authors. Secondly, the data has not been as useful as we originally hoped. This is due to a variety of factors, including a lack of variables valuable to our purposes. In total, we removed or mostly ignored 8 of the 22 variables and only a handful of the remaining variables appear to have any significant correlation with our dependent variables (highest charting position, weeks charted, streams). In addition to having more variables, it likely would have been beneficial to us to have data that covered more years instead of being limited to just 2020 and 2021. Having this additional data would have allowed us to look more deeply into the relationships between our dependent variables and how they change over time. Our data had to be cleansed and processed before it was ready for analysis. We got rid of several attributes such as Song Title and the actual dates it was on the charts as well as created a Release Year and Release Season column.

Section 3: Fundamental Data Analysis

Part 1 - Univariate Analysis and Theories

Figure 1: Five Number Summary and Mean of Transformed Data

```
> summary(df)
      HCP            NumCharted        Streams            Followers          Popularity      Danceability
 Min.   :  1.00   Min.   :  1.00   Min.   : 4176083   Min.   :    4883   Min.   :  0.00   Min.   :0.150
 1st Qu.: 37.00   1st Qu.:  1.00   1st Qu.: 4915080   1st Qu.: 2123734   1st Qu.: 65.00   1st Qu.:0.599
 Median : 80.00   Median :  4.00   Median : 5269163   Median : 6852509   Median : 73.00   Median :0.707
 Mean   : 87.83   Mean   : 10.68   Mean   : 6337136   Mean   :14716903   Mean   : 70.09   Mean   :0.690
 3rd Qu.:137.00   3rd Qu.: 12.00   3rd Qu.: 6452492   3rd Qu.:22698747   3rd Qu.: 80.00   3rd Qu.:0.796
 Max.   :200.00   Max.   :142.00   Max.   :48633449   Max.   :83337783   Max.   :100.00   Max.   :0.980

     Energy         Loudness        Speechiness      Acousticness         Liveness          Tempo
 Min.   :0.0540   Min.   :-25.166   Min.   :0.0232   Min.   :0.0000255   Min.   :0.0197   Min.   : 46.72
 1st Qu.:0.5320   1st Qu.: -7.491   1st Qu.:0.0456   1st Qu.:0.0485000   1st Qu.:0.0966   1st Qu.: 97.96
 Median :0.6420   Median : -5.990   Median :0.0765   Median :0.1610000   Median :0.1240   Median :122.01
 Mean   :0.6335   Mean   : -6.348   Mean   :0.1237   Mean   :0.2486945   Mean   :0.1812   Mean   :122.81
 3rd Qu.:0.7520   3rd Qu.: -4.711   3rd Qu.:0.1650   3rd Qu.:0.3880000   3rd Qu.:0.2170   3rd Qu.:143.86
 Max.   :0.9700   Max.   :  1.509   Max.   :0.8840   Max.   :0.9940000   Max.   :0.9620   Max.   :205.27

    Duration         Valence        ReleaseYear
 Min.   : 30133   Min.   :0.0320   2020   :783
 1st Qu.:169266   1st Qu.:0.3430   2021   :396
 Median :193591   Median :0.5120   2019   :181
 Mean   :197941   Mean   :0.5147   2018   : 43
 3rd Qu.:218902   3rd Qu.:0.6910   2017   : 16
 Max.   :588139   Max.   :0.9790   1905   : 15
                                   (Other):111
>
```

This image depicts the five number summary and mean of our transformed data which includes relevant variables. Several important things can be noted from this summary. First, the skew that

has plagued our analysis can be easily seen in the dependent variables such as HCP (min: 1, max: 200, median: 80), NumCharted (min: 1, max: 142, median: 4), and Streams (min: ~4 million, max: ~49 million, median: ~5 million). Significant skew also exists in several independent variables, including Speechiness (min: ~0.02, max: 0.88, median: 0.07), Acousticness (min: ~0.00, max: ~0.99, median: ~0.16), and Liveness (min: ~0.02, max: ~0.96, median: ~0.12). Visualized, these skews look like the following figures:

Figure 2: Histogram of HCP

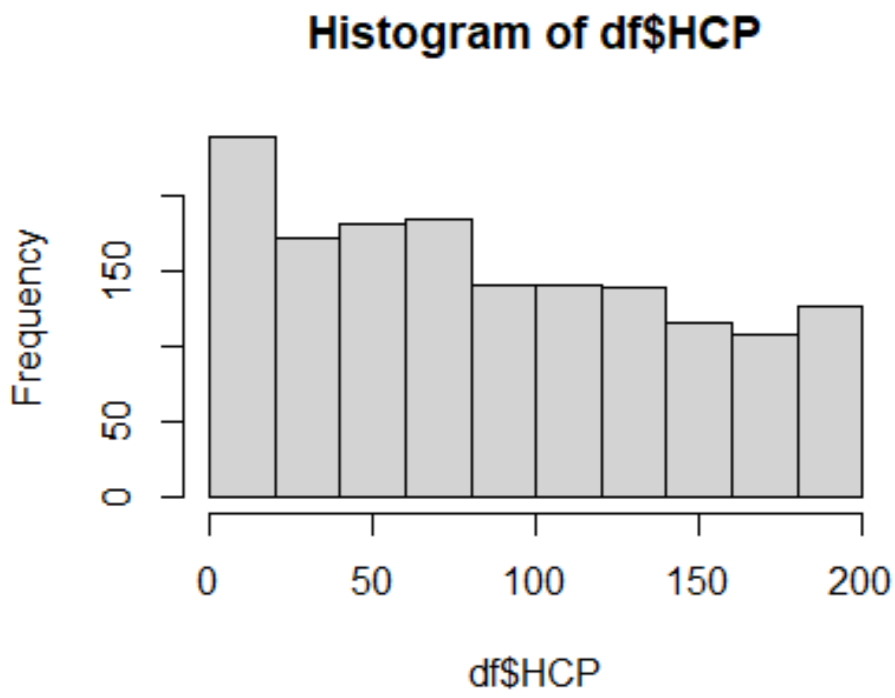


Figure 3: Histogram of NumCharted

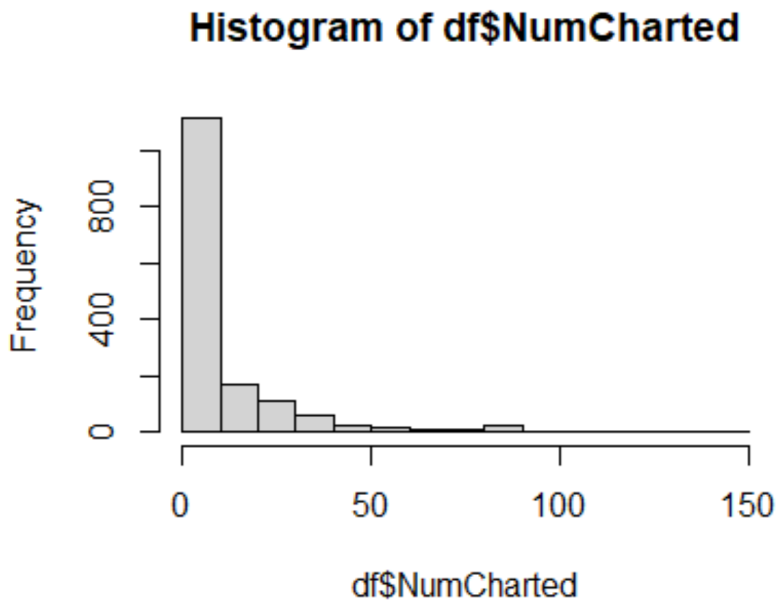


Figure 4: Histogram of Streams

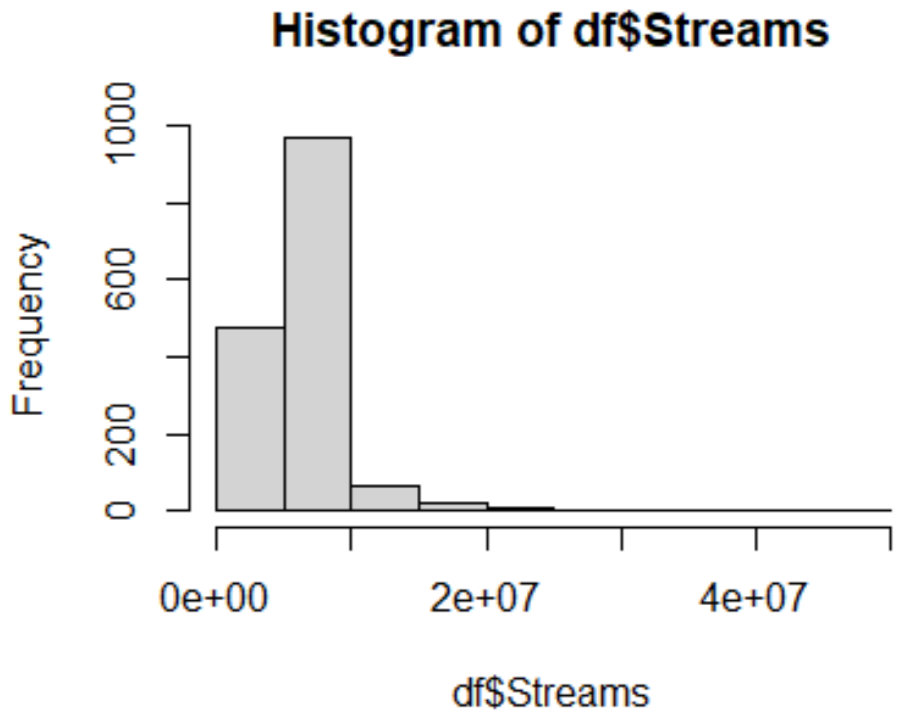


Figure 5: Histogram of Speechiness

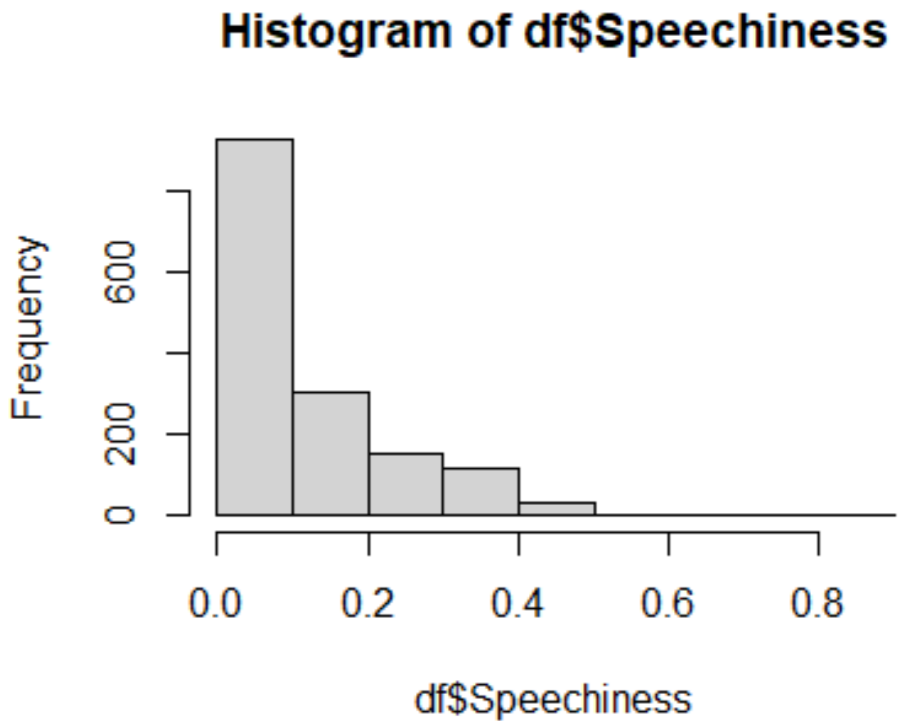


Figure 6: Histogram of Acousticness

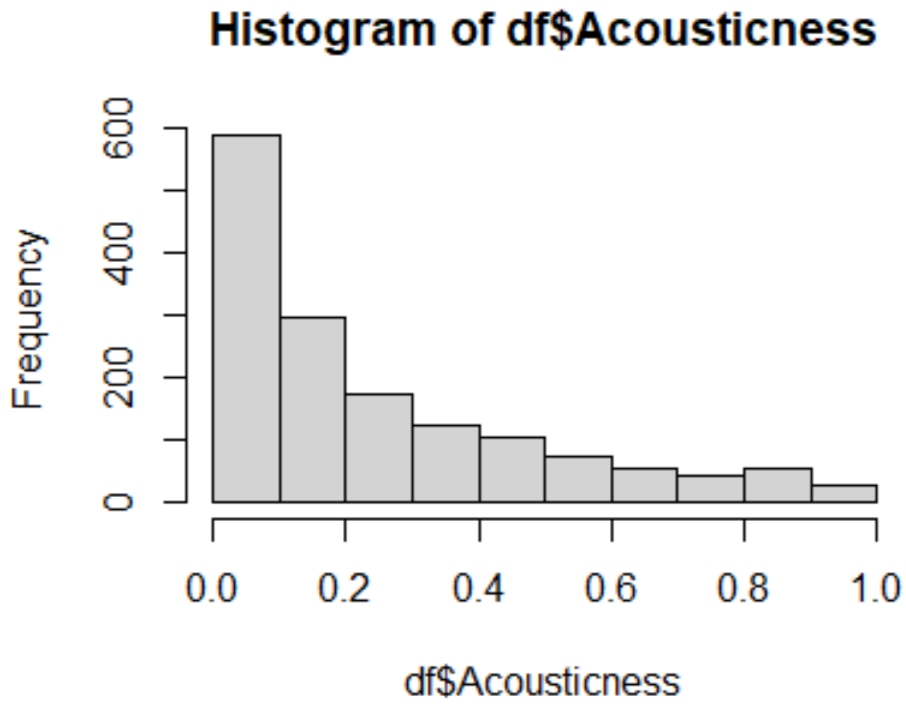
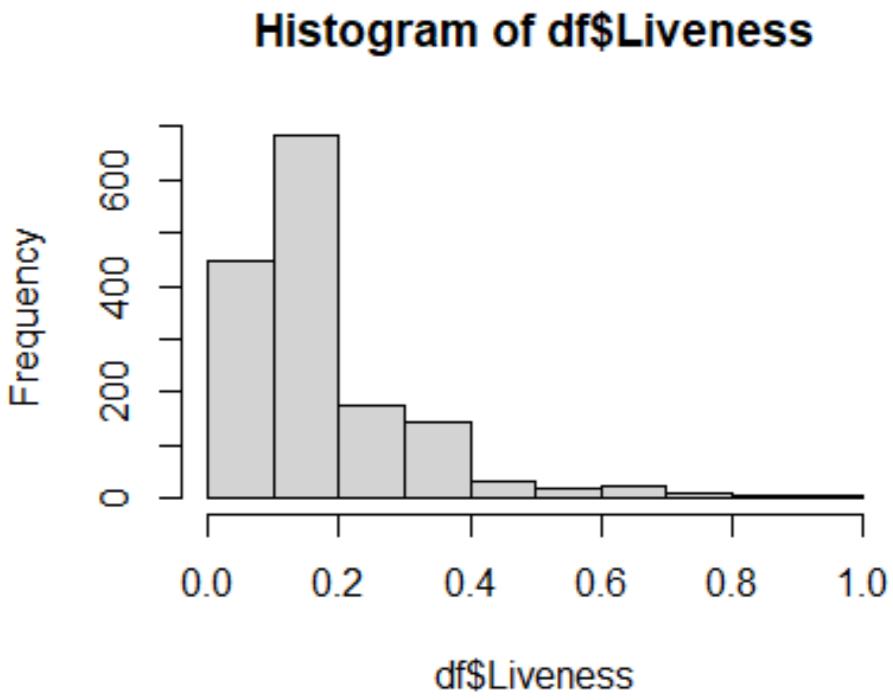


Figure 7: Histogram of Liveness

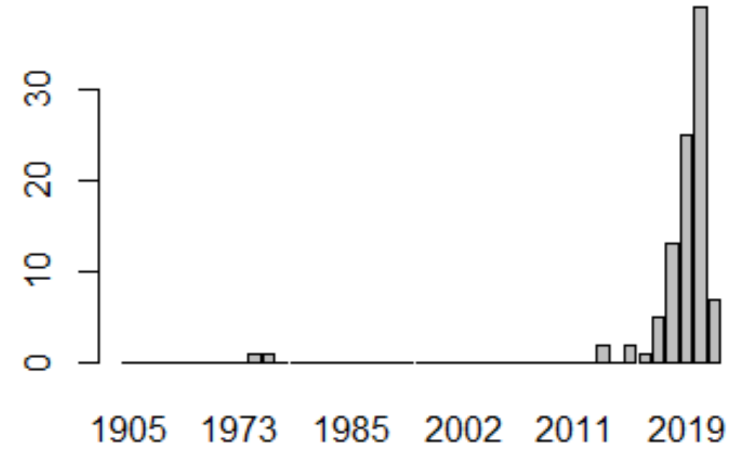


In order to account for the skew of highest charting position (Figure 3), we attempted to filter and split the data based on the quantiles of number of times charted as described in the further sections and displayed in the RScript located in Appendix 3.

We originally believed streams (Figure 4) might be highly skewed because the data set contains songs that were released as early as 1905. Our assumption was that the total streams of an old song that has been released for many years obviously will have a higher value for total streams than a newly released song. However, plotting top streams songs vs release year does not support this correlation. Instead, in recent years as Spotify has increased in popularity, popular songs are receiving more and more streams.

As an extension of this, we assumed older songs would have a tendency to have a higher number of weeks charted. However, as seen in Figure 8 below, this again is not the case. A comparison between top charted songs and release years reveals little to no correlation between a high number of weeks charted and an old release date.

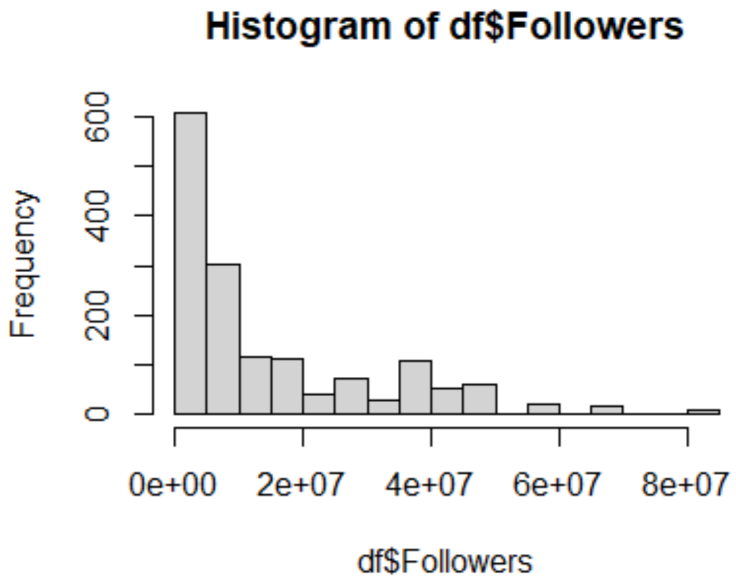
Figure 8: Most Charted Songs' (>= 39 weeks) Release Years



Number of times charted (Figure 3) is also interesting because the skew of this attribute indicates that most of the top streamed songs from 2020-2021 were charted a low number of times with a few songs being charted a lot more.

Lastly, we believe Followers (Figure 9) have some outliers on the high end due to the presence of extremely popular artists such as Taylor Swift and some outliers on the low end due to the resurgence of old songs from social media.

Figure 9: Histogram of Followers

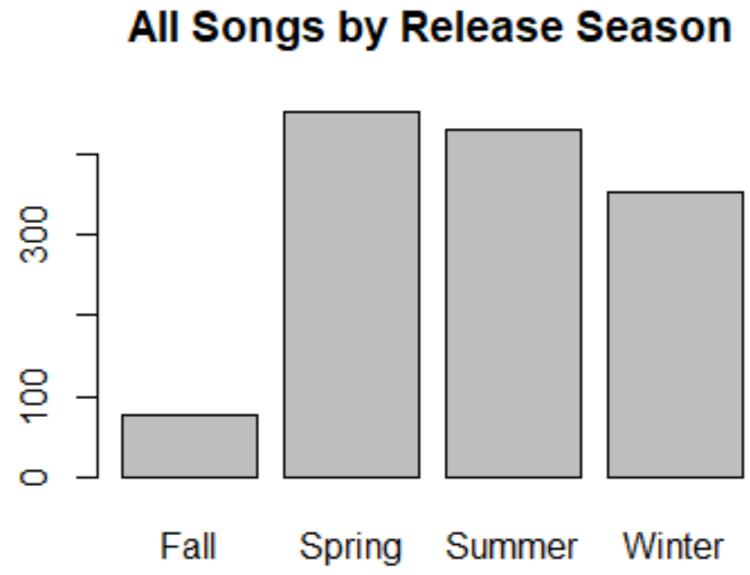


Lastly, Appendix 1 contains the correlation matrix of the numerical data. There are a few interesting correlations worth noting. The strongest correlation is .73 between energy and loudness. It definitely makes sense that as a song gets louder, it has more energy. Danceability and valence and danceability and energy both have moderate positive correlations. There are also moderate negative correlations between acousticness and danceability, acousticness and loudness, and acousticness and energy.

Part 2 - Bivariate Analysis and Theories

Our analysis of bivariate relationships is not the most in-depth, as time was dedicated early on to making the data usable and then later to the multivariate analysis. However, we still came across some interesting results. Although limited, from our exploration of bivariate relationships we are able to make some interesting inferences and conclusions. First, songs by release season.

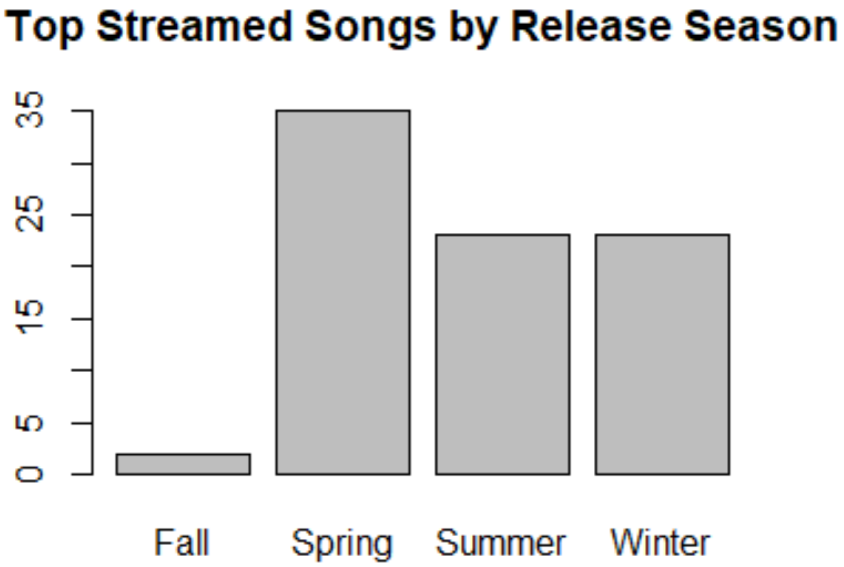
Figure 10: All Songs by Release Season



We found this chart extremely interesting and were surprised by the significant lack of Fall releases. We expected a lot fewer releases in Winter, but were not surprised that Spring and Summer had relatively high releases. We theorize that artists prefer to release songs while a lot of people will be at the beach, at parties, and similar situations. Also, it is possible that artists want

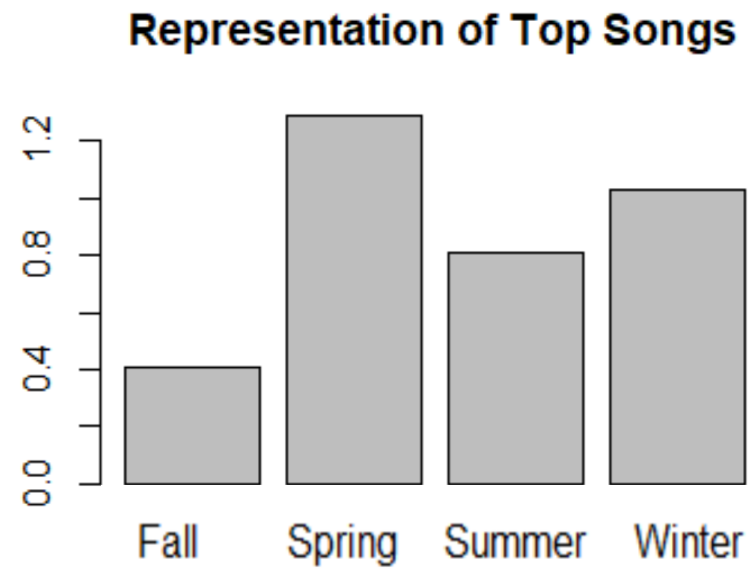
to release their songs with sufficient time to be competitive at the American Music Awards, which occurs annually in November.

Figure 11: Top Streamed Songs (>= 10 million) by Release Season



This chart is arguably even more surprising, as only 2/83 top streamed songs were released in Fall. However, given our assumptions about Figure 10, this disparity can be more easily explained.

Figure 12: Representation of Top Streamed Songs by Release Season



This chart displays the representation of top streamed songs by Release Season. It is calculated by dividing the number of top streamed songs per season by the total number of top streamed songs. Those numbers are then divided by the number of songs by season divided by the total number of songs. This means that Fall releases are underrepresented by 60% among top streamed songs, while Spring is overrepresented by about 130%.

After this, we looked at the relationship between the number of streams of top streamed songs and release season. We began by looking at the average streams of all songs by release season (Figure 13) followed by the average streams of top streamed songs by release season (Figure 14).

Figure 13: Average Streams of All Songs by Release Season

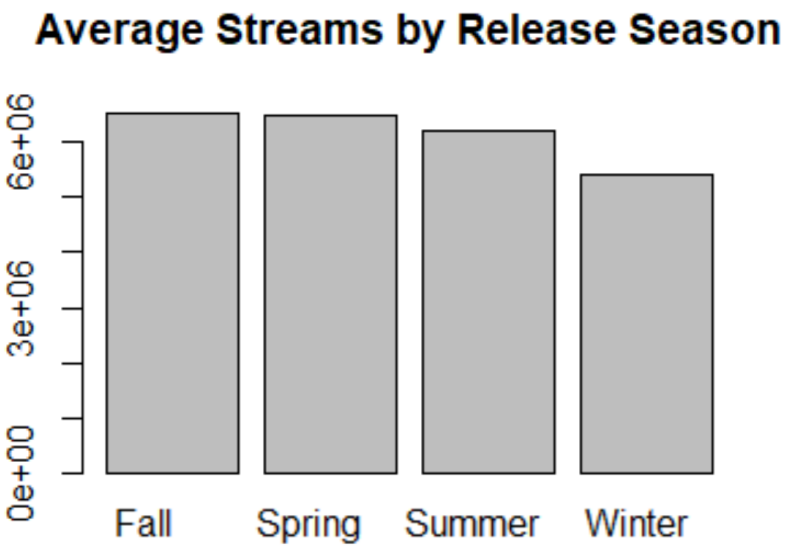
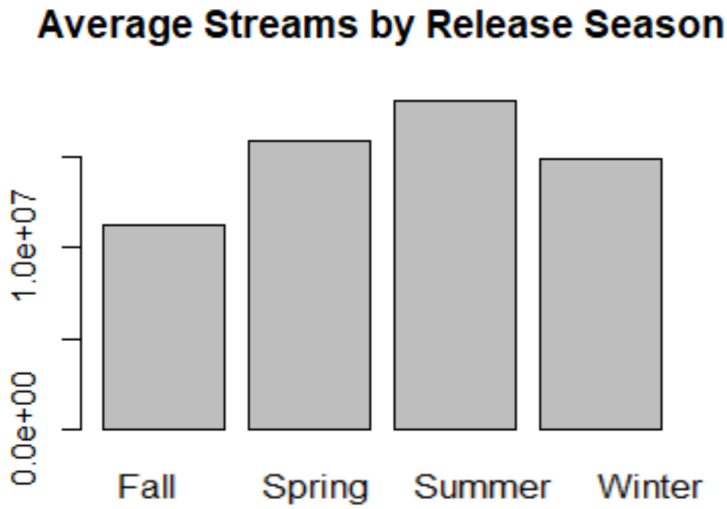
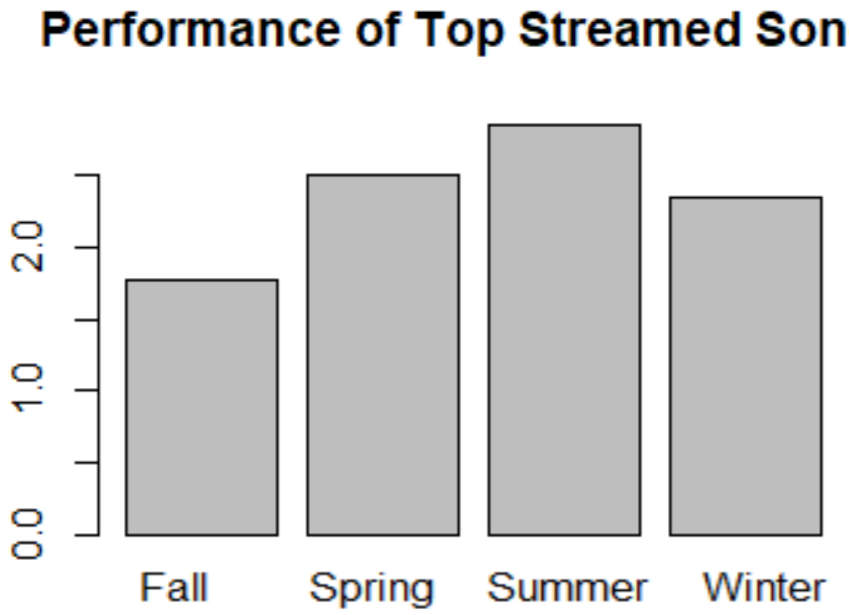


Figure 14: Average Streams of Top Streamed Songs by Release Season



These charts lead us to create the following chart, which compares the performance of top streamed songs vs all songs by release season.

Figure 15: Performance of Top Streamed Songs vs All Songs by Release Season



This means that, on average, top streamed songs released in fall receive about 1.75x more streams than all songs released in fall. Top Summer songs, by comparison, receive over 3x more streams.

Next, we investigated what relationship, if any, exists between streams and the number of genres a song has. Also note that going forward, “genre” will mean the number of genres a song has, not the type of genre it is.

Figure 16: Percentage of Songs by Genre

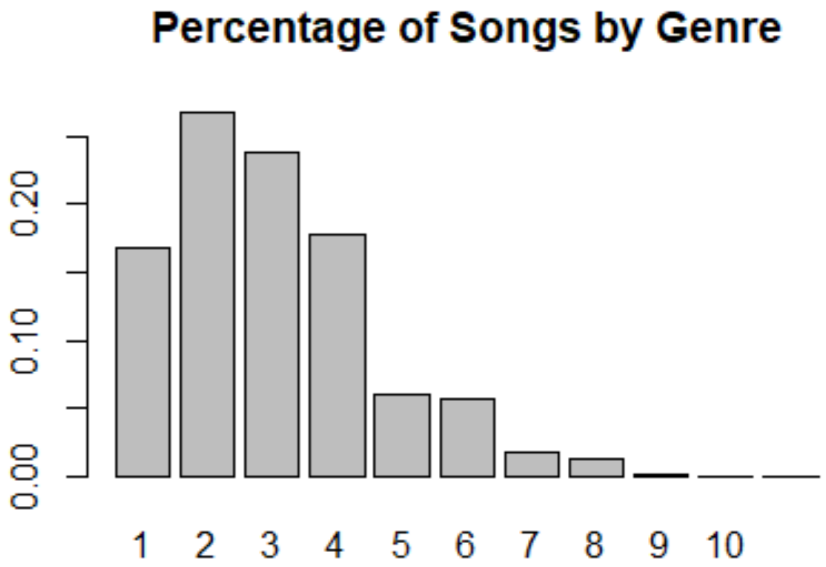


Figure 17: Percentage of Streams by Genre

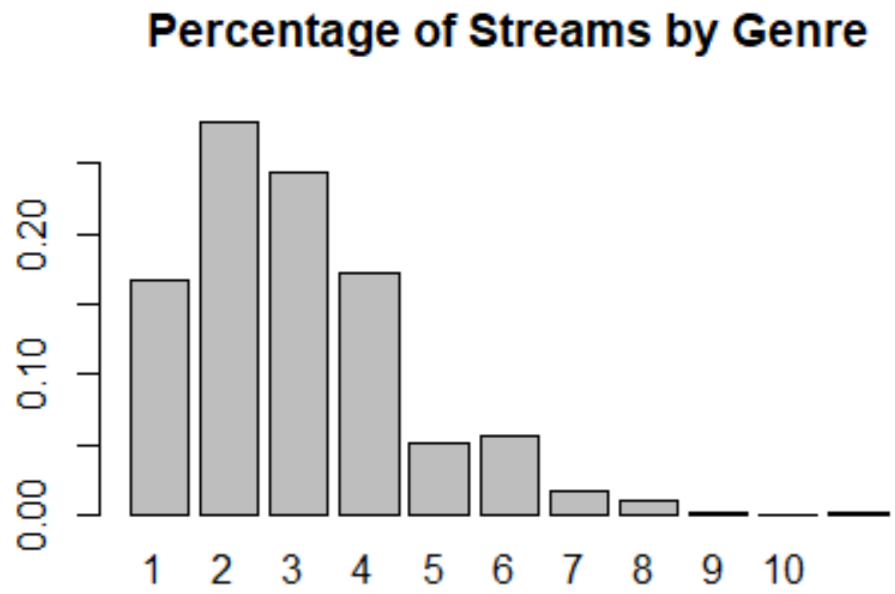


Figure 18: Stream Performance of Songs by Genre

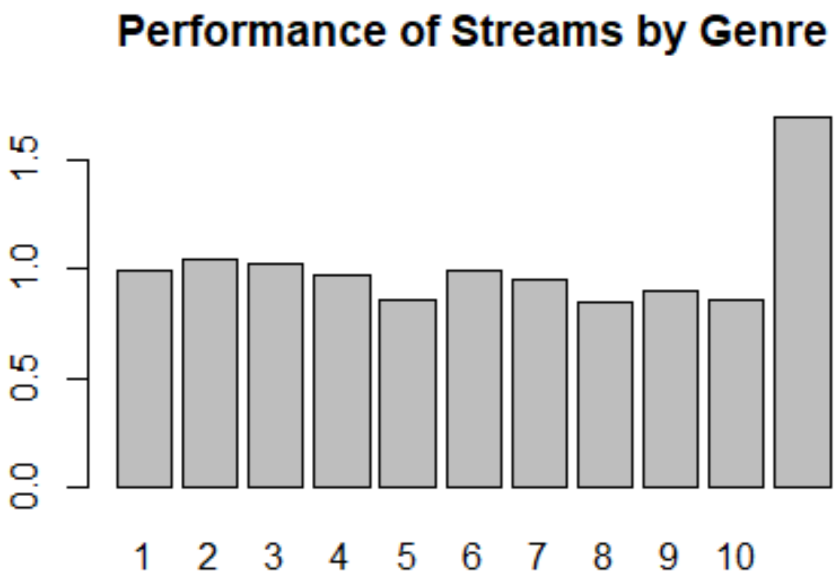


Figure 17 is calculated by finding the percentage of streams for each genre. This can then be used to create Figure 18, which represents the percentage of how many streams each genre has compared to the percentage of songs that have that number of genres. This means that on average, each genre has about as many streams as would be expected given how prevalent each genre is. There also appears to be a slight negative correlation between number of genres and percentage of streams. The exception to this is the one song with 11 genres, which has about 70% more streams than would be expected given that there is only one song.

We continued by looking for more relationships between the number of genres and top streamed songs.

Figure 19: Percentage of Top Streamed Songs by Genre

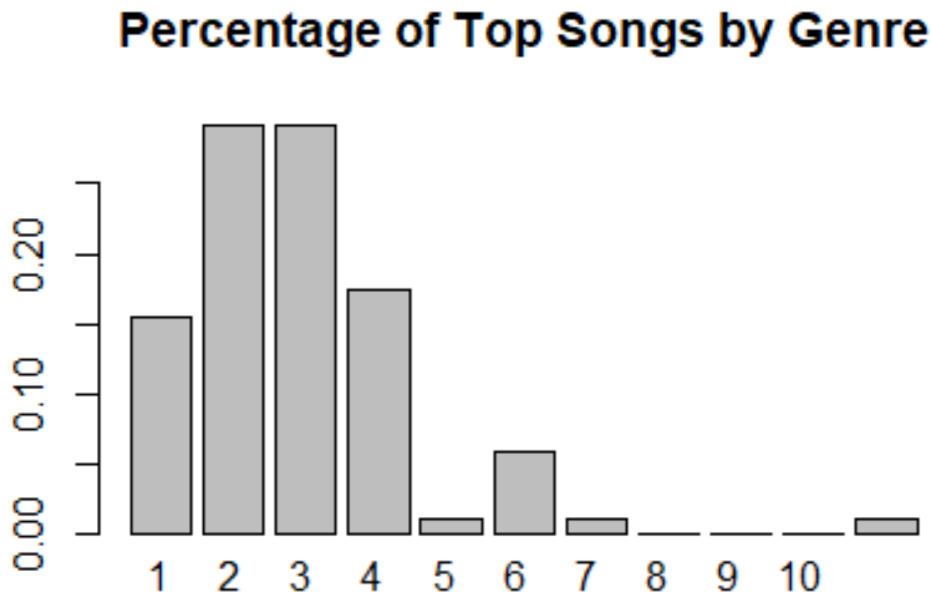
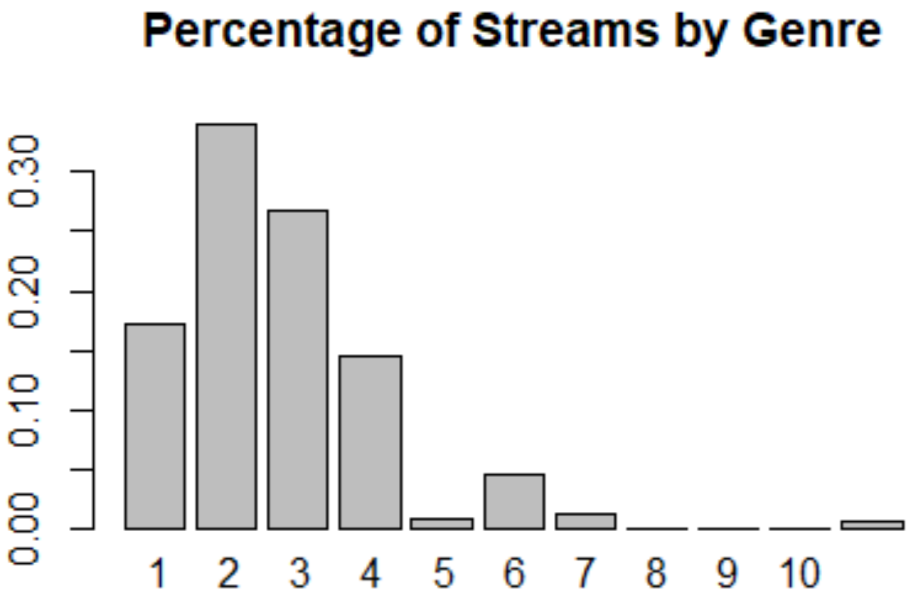
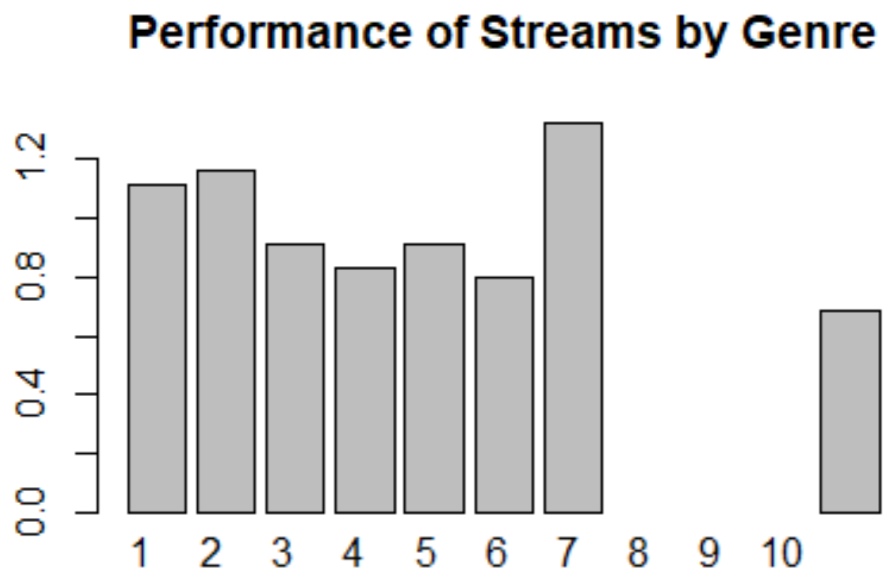


Figure 20: Percentage of Streams of Top Streamed Songs by Genre



Figures 19 and 20 are found the same way at 17 and 18, respectfully. This leads us to the following chart:

Chart 21: Stream Performance of Top Streamed Songs by Genre



Again, we see a downwards trend of streams as the number of genres increases.

Lastly, we looked into the relationship between release season and danceability. We suspected that summer songs would tend to have a higher danceability. Our finds can be seen below:

Figure 22: Histogram of Release Season

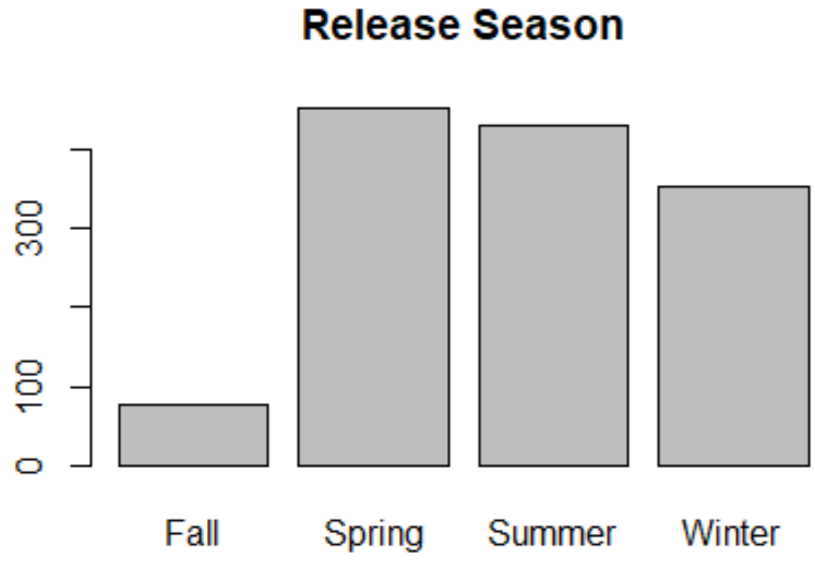


Figure 23: Adjusted Danceability by Release Season

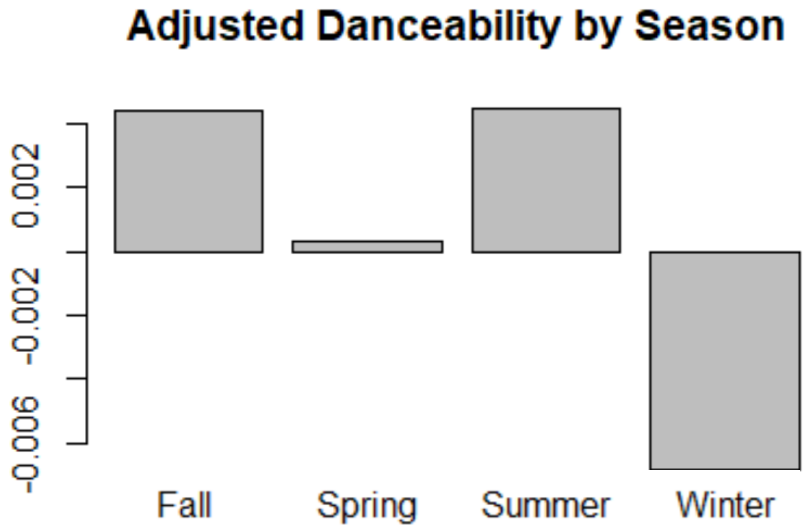


Figure 23 is calculated by subtracting the mean danceability of all songs from each season's song's mean danceability. This tells us that, although slight, songs released in Fall and Summer tend to have a higher danceability while Spring songs remain neutral and Winter songs tend to have lower danceability.

This concludes our fundamental data analysis. Although we would have liked to pursue this research in more depth, we turned our focus to the multivariate analysis, which follows this section. That being said, we still believe our fundamental data analysis has been beneficial in getting a basic understanding of some of the relationships between variables, particularly with streams. Although not the full picture, this basic understanding served as a starting point for our multivariate research.

Section 4: Multivariate Analysis

When we initially began to run models in R, we realized our data had some issues in its current form. Our first linear regression model we ran, with Streams as the dependent variable, gave us a QQ plot that seemed to fit okay until it got to the upper right corner where it shot up. Due to the Streams Model QQ plot irregularities we began to contemplate other options such as using other variables as the dependent variable or splitting our data into several pieces on Streams.

We began to run models in R with HCP and Number of Times Charted (numCharted). The QQ plots resulting from these models looked to be more promising so we proceeded with various models using HCP and NumCharted as dependent variables. Figure 8 in appendix 2 represents our best model run in R based on correlation coefficient and root mean squared error. This model had HCP as the dependent variable with a correlation coefficient of 0.6312 and a root mean squared error of 44.17 and an adjusted r squared of 0.3985 . In this model Streams, Followers, Popularity and NumCharted are the most significant.

Additionally, we ran linear regression models in R with numCharted as well as numCharted split in several different ways. The best NumCharted model in R with a 0.368 r squared gives us a correlation coefficient of 0.6066.

We also looked into splitting the data on the NumCharted variable into several groups, less than 2, 3 to 7, 8 to 30, and 31 to 142. We ran models to determine which splits seemed to preliminarily be best. We then moved to Weka and began to run M5P, Linear Regression, and REPTree on our full data set with numCharted as the class variable. We ran the same test with each of our splits with numCharted as the class variable. Next, we ran the same three types of models in Weka on our full data set with HCP as the class variable.

Figures 1-7 in appendix 2 show tables of experiments run with the correlation coefficient, root mean squared error and some observations we made when running the models. As shown in Figure 1 we ran several extra models with our HCP data as we were trying to decide which to use. We ultimately went with Linear Regression, M5P and the REPTree because they fit well with Linear Regression and Dr. Eynon had mentioned both to our group.

As we ran the linear regression models, we initially explored some changes in default settings, specifically dealing with attribute selection and collinearity, but ultimately chose to stay with the default settings for our comparison. For our initial models with HCP as the dependent variable, we had correlation coefficients between around 0.5 and 0.85 and root mean squared errors between around 30 and 48. All of the results were pretty complex and did not seem to give any clear indication of what factors could be causing the HCP.

We then ran models with NumCharted as the dependent variable. We began with using all the observations in the data set and got correlation coefficients between around 0.56 and 0.79 and mean squared errors around 10.5 and 13. We then ran our three different models with the four splits mentioned above. The first split had similar correlation coefficients but very low root mean squared errors of below 0.3, which was much lower than any other models we ran. We then ran the other 3 split sections. They had significantly less instances included than the 653 in the first split, so we believe that could have caused some of our issues. When we ran the other splits the correlation coefficients were similar to the model with all the observations and root mean squared errors in the high teens.

We believe our models turned out the way they did due to our data itself. It did not give much indication as to the factors that make a song chart highly or chart for many weeks. We were able to get some models with low errors and higher correlation coefficients, but were not able to extract any real answers to our questions from this data.

Section 5: Improving the Models

After running our initial splits on numCharted we determined there had to be a better option than our four splits with much higher errors than the first split. In our preliminary models we split the data into several (5) splits based on numCharted and ran models in Weka with each split's data. This gave us sections of data with very few instances and high errors which can be seen in Figures 4 through 6. We then went back to R to find better alternatives to our original splits. After trying several ranges of values we split the data into two sections: the original split one of two or less and all the rest. These showed more promising results than the 3 split version of that half of the data.

Section 6: Next Steps

Our goal was to determine what factors make a song popular in regards to Highest Charting Position, Number of Streams or Number of Times Charted. We are not able to explain this

specifically with our current data set. Based on our various models with several different dependent variables we were not able to find any real clear indicators. We went into the project with thoughts of a few variables, such as loudness or duration, making a huge difference in the popularity but that is not what we found. Another thing that complicated matters was the Release Year category. There were so many different years it tended to make the regression very complicated, which we know is never ideal. If we had more time and resources we may have tried to find a way to separate those years into bins or even created a variable such as Popular in Release Year.

If we were to continue our analysis, we believe that we would need a larger data set containing more years as two years of data in a time such as the pandemic may not truly be the best indicator of popularity. Also, having more variables such as artist label or social media following could be helpful indicators. Another idea we thought about was the possibility of in the future turning our question on its head and trying to predict a category such as Very Popular, Popular, and Not Popular. We think that could have been another interesting way to approach our question and if we had unlimited time we could explore all avenues such as this one.

With more data and possibly more variables we feel we would have had a higher likelihood of finding some answer to our problem, but as with all projects of this nature it is never certain.

Appendices:

Appendix 1

Table 1: Data Dictionary

Attribute	Description	Type
Highest Charting Position	The highest position that the song has been on in the Spotify Top 200 Weekly	Numeric

	Global Charts in 2020 & 2021.	
Number of Times Charted	The number of times that the song has been on in the Spotify Top 200 Weekly Global Charts in 2020 & 2021.	Numeric
Number of Genre	The song belongs to according to Spotify.	Numeric
Release Season	Time of year (Winter, Spring, Summer, Fall) the song was released. Winter: December, January February ; Spring: March, April, May ; Summer: June, July, August ; Fall: September, October, November	Converted to a Factor in R
Release Year	The year that the song was released.	Converted to a Factor in R
Popularity	The popularity of the track. The value is between 0 and 100, with 100 being the most popular.	Numeric

Danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. As danceability gets closer to one, a song is considered to be more danceable and as it gets closer to zero, it is considered to be less danceable. One might assume that at least in present day music.	Numeric
Energy	Energy is 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.	Numeric
Loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track. Values typical range between -60 and 0 db.	Numeric (-60 to 1)

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.	Numeric (0 to 1)
Acousticness	Measures whether or not a track is purely acoustic. As it approaches one, that represents a higher confidence that a track is acoustic.	Numeric (0 to 1)
Liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.	Numeric (0 to 1)
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.	Numeric

Duration	Length of song	Numeric
Valence	Represents the overall happiness of a song. As it is closer to one that represents a sad song, while it approaches five that represents a happier song.	Numeric (1 to 5)
Streams	Number of streams a song has	Numeric
Artist Followers	Followers that the primary artist has on Spotify.	Numeric

```
> #summarize our data
> summary(df)
      HCP           NumCharted        Streams           Followers          Popularity      Danceability       Energy
 Min.   :  1.00   Min.   :  1.00   Min.   : 4176083   Min.   :    4883   Min.   :  0.00   Min.   :0.150   Min.   :0.0540
 1st Qu.: 37.00   1st Qu.:  1.00   1st Qu.: 4915080   1st Qu.: 2123734   1st Qu.: 65.00   1st Qu.:0.599   1st Qu.:0.5320
 Median : 80.00   Median :  4.00   Median : 5269163   Median : 6852509   Median : 73.00   Median :0.707   Median :0.6420
 Mean   : 87.83   Mean   : 10.68   Mean   : 6337136   Mean   :14716903   Mean   : 70.09   Mean   :0.690   Mean   :0.6335
 3rd Qu.:137.00   3rd Qu.: 12.00   3rd Qu.: 6452492   3rd Qu.:22698747   3rd Qu.: 80.00   3rd Qu.:0.796   3rd Qu.:0.7520
 Max.   :200.00   Max.   :142.00   Max.   :48633449   Max.   :83337783   Max.   :100.00   Max.   :0.980   Max.   :0.9700

    Loudness        Speechiness      Acousticness        Liveness          Tempo          Duration        Valence
 Min.   :-25.166   Min.   :0.0232   Min.   :0.0000255   Min.   :0.0197   Min.   : 46.72   Min.   : 30133   Min.   :0.0320
 1st Qu.: -7.491   1st Qu.:0.0456   1st Qu.:0.0485000   1st Qu.:0.0966   1st Qu.: 97.96   1st Qu.:169266   1st Qu.:0.3430
 Median : -5.990   Median :0.0765   Median :0.1610000   Median :0.1240   Median :122.01   Median :193591   Median :0.5120
 Mean   : -6.348   Mean   :0.1237   Mean   :0.2486945   Mean   :0.1812   Mean   :122.81   Mean   :197941   Mean   :0.5147
 3rd Qu.: -4.711   3rd Qu.:0.1650   3rd Qu.:0.3880000   3rd Qu.:0.2170   3rd Qu.:143.86   3rd Qu.:218902   3rd Qu.:0.6910
 Max.   :  1.509   Max.   :0.8840   Max.   :0.9940000   Max.   :0.9620   Max.   :205.27   Max.   :588139   Max.   :0.9790

 ReleaseSeason   ReleaseYr      NumGenre
       :231     2020   :783   Min.   : 1.000
 Fall  : 78     2021   :396   1st Qu.: 2.000
 Spring:452     2019   :181   Median : 3.000
 Summer:431     2018   : 43   Mean   : 3.012
 Winter:353     2017   : 16   3rd Qu.: 4.000
                1905   : 15   Max.   :11.000
                (Other):111
> |
```

```
> cor(df[ ,1:14])
                    HCP  NumCharted      Streams   Followers    Popularity Danceability       Energy    Loudness Speechiness
HCP           1.00000000 -0.41774776 -0.295442316 -0.23372324 -0.1641673405   0.01714931  0.063025588  0.03216599  0.04124832
NumCharted   -0.41774776  1.00000000 -0.060542175  0.02745826  0.2327955754   0.02702598 -0.061139284  0.03122547 -0.06021586
Streams      -0.29544232 -0.06054218  1.000000000  0.10325112  0.1231838925  -0.08129108  0.004144562 -0.03115498 -0.06261971
Followers    -0.23372324  0.02745826  0.103251122  1.00000000  0.1043577212  -0.09757578 -0.065613356 -0.03326448 -0.07296828
Popularity   -0.16416734  0.23279558  0.123183893  0.10435772  1.0000000000   0.02843469  0.094690518  0.15876747 -0.03209063
Danceability  0.01714931  0.02702598 -0.081291077 -0.09757578  0.0284346918   1.00000000  0.142129561  0.23492774  0.23739441
Energy        0.06302559 -0.06113928  0.004144562 -0.06561336  0.0946905176   0.14212956  1.000000000  0.73261637  0.02398908
Loudness      0.03216599  0.03122547 -0.031154977 -0.03326448  0.1587674680   0.23492774  0.732616374  1.00000000 -0.01882265
Speechiness   0.04124832 -0.06021586 -0.062619706 -0.07296828 -0.0320906336   0.23739441  0.023989080 -0.01882265  1.00000000
Acousticness -0.01292375  0.04665109  0.034054120  0.02383032 -0.0912446107  -0.31679837 -0.542399209 -0.47743099 -0.13143647
Liveness      0.01271784 -0.05843646  0.042105457 -0.01249110 -0.0294596347  -0.11451842  0.124693063  0.04314065  0.07277436
Tempo         0.02623532 -0.04830727  0.053458430 -0.01988115 -0.0249508765  -0.04021855  0.113351613  0.10437123  0.11125547
Duration     -0.03395584  0.03398038  0.015963535  0.14214457  0.0820957254  -0.10138991  0.056623993  0.07526221 -0.08989537
Valence       0.04536177  0.02156962  0.038381046 -0.10880361 -0.0009533924   0.36162713  0.356324546  0.29876225  0.03803174
             Acousticness     Liveness        Tempo     Duration       Valence
HCP          -0.012923750  0.012717838  0.026235315 -0.033955843  0.0453617674
NumCharted    0.046651091 -0.058436461 -0.048307266  0.033980383  0.0215696179
Streams       0.034054120  0.042105457  0.053458430  0.015963535  0.0383810461
Followers     0.023830324 -0.012491097 -0.019881147  0.142144568 -0.1088036066
Popularity   -0.091244611 -0.029459635 -0.024950877  0.082095725 -0.0009533924
Danceability -0.316798370 -0.114518415 -0.040218554 -0.101389913  0.3616271282
Energy       -0.542399209  0.124693063  0.113351613  0.056623993  0.3563245456
Loudness     -0.477430989  0.043140645  0.104371233  0.075262209  0.2987622517
Speechiness  -0.131436470  0.072774355  0.111255467 -0.089895370  0.0380317448
Acousticness  1.000000000 -0.005469275 -0.061632009 -0.046010058 -0.0969974764
Liveness     -0.005469275  1.000000000 -0.018265150  0.019685282  0.0078815248
Tempo        -0.061632009 -0.018265150  1.000000000 -0.004671354  0.0575630210
Duration     -0.046010058  0.019685282 -0.004671354  1.000000000 -0.1199813808
Valence      -0.096997476  0.007881525  0.057563021 -0.119981381  1.0000000000
> |
```

Appendix 2

Figure 1.

Below Test were performed as HCP as Class Attribute in Weka

Model	Cor. Coe.	Root Mean Squared Error	Discussion
M5P Tree	0.8537	30.2737	12 rules in model tree, used 10 fold cross validation, not super helpful per Eynon, HCP Class Variable, num charted first split then streams
Random Forest	0.8573	30.8958	Used 10 fold cross validation, wanted to see what would

			happen,HCP Class Variable
Weka Linear Regression 1	0.6009	46.5682	Ran in Weka, first attempt, left settings as default, HCP Class Variable, took out several variables like Danceability, loudness, energy, tempo, believe Weka took them out do to colinearity or irrelevance
Linear Regression 2	0.6009	46.5682	Second attempt, changes, eliminate collinear attributes to False, same model
Linear Regression 3	0.6015	46.5635	Third attempt, selection method no attribute selection and kept eliminate collinear to false, model included 17 of 17 attributes, was more complicated as it included many more attributes

REPTree	0.7902	35.8886	Size of tree was 372, too much to visualize well, split on num charted first then streams then many splits on release year
Linear Regression	0.5518	48.4877	No release year, only 8 attributes, wanted to see a more simplified equation Artisticness and Valence seemed to have highest impact
Linear Regression	0.5516	48.495	Set to no attribute selection so it utilized all 15 (release year taken out by us)

Figure 2.

With NumCharted as Class Attribute No Split

M5P	0.7903	10.594	Default settings with cross validation then release year
LR 1	0.5707	13.4795	Default settings
LR 2	0.5697	13.4971	No attribute selection and false for remove collinear

REPTree	0.7131	11.5909	192 size of tree, smaller tree for HCP, first split on popularity

Figure 3.

With NumCharted as Class Attribute Split #1

*Number of Times Charted = 0-2, 653 instances

M5P	0.7324	0.2867	24 rules, splits on HC first
LR	0.5786	0.3456	Default settings
REPTree	0.6072	0.3374	Size of tree is 59, HCP top of tree

Figure 4.

With NumCharted as Class Attribute Split #2

*Number of Times Charted = 3-7, 100 instances

M5P	0.5639	21.4397	Model tree not good, only 1 rule, probably due to size
LR	0.5501	21.8	Used default settings, only used HCP, Streams, Loudness and Release Year

REPTree	0.6043	20.8408	Size of tree is 53, first split release, tree looks insane

Figure 5.

With NumCharted as Class Attribute Split #3

*Number of Times Charted = 8-30, 224

M5P	0.6675	16.7603	5 rules, streams at top of tree
LR	0.511	19.4961	Smaller equation, valance seems high in this one
REPTree	0.493	19.863	Split on release year first, size of tree is 53

Figure 6.

With NumCharted as Class Attribute Split #4

*Number of Times Charted = 31-142, 144

M5P	0.4065	18.9207	6 rules, release year first split
LR	0.4574	18.7698	Smaller, 5 independent variables, majority based on release year

REPTree	.3783	19.3819	Size is 47, release year then streams

Figure 7.

With NumCharted as Class Attribute Split #5

*Number of Times Charted = 3-142, 806 instances

M5P	0.7092	13.6084	13 rules, popularity first split
LR	0.4926	16.9156	Utilized most of attributes
REPTree	0.6351	15.0386	Size 148, popularity first split, utilized lots of the attributes

Appendix 3 to N. It would be a good idea to copy and paste Weka output, R scripts and output that you refer to in the body of your written project.

Figure 8.

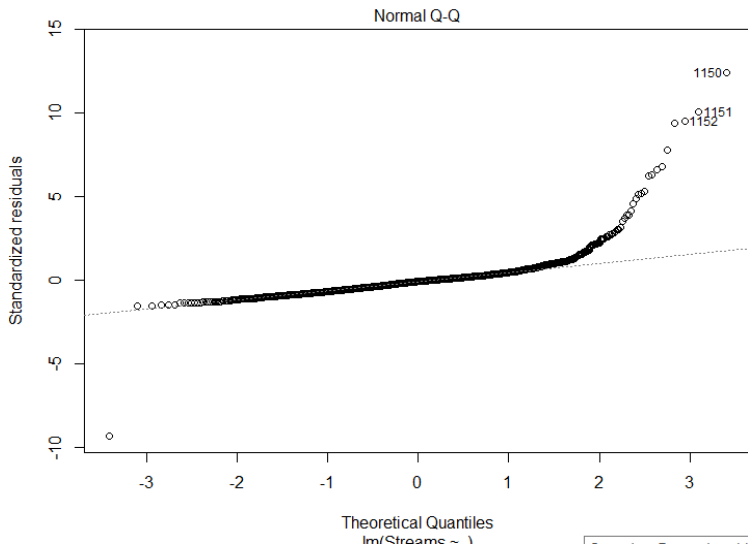
Best Linear Model from R using HCP as dependent:

```
Call:
lm(formula = HCP ~ NumCharted + Streams + Followers + Popularity +
    Danceability + Energy + Loudness + Speechiness + Acousticness +
    Liveness + Tempo + Duration + Valence + ReleaseSeason + NumGenre +
    ReleaseYr, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-116.831  -33.100   -1.763   32.283  132.163

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.646e+02  2.113e+01   7.787 1.28e-14 ***
NumCharted        -1.780e+00  7.705e-02 -23.105  < 2e-16 ***
Streams           -4.791e-06  3.697e-07 -12.958  < 2e-16 ***
Followers         -7.071e-07  7.426e-08  -9.522  < 2e-16 ***
Popularity        -1.622e-01  7.927e-02  -2.047  0.04086 *
Danceability      -1.800e+00  1.018e+01  -0.177  0.85968
Energy             6.247e+00  1.212e+01   0.515  0.60647
Loudness           4.158e-01  7.368e-01   0.564  0.57257
Speechiness        7.348e+00  1.126e+01   0.653  0.51403
Acousticness       7.960e+00  6.014e+00   1.324  0.18580
```

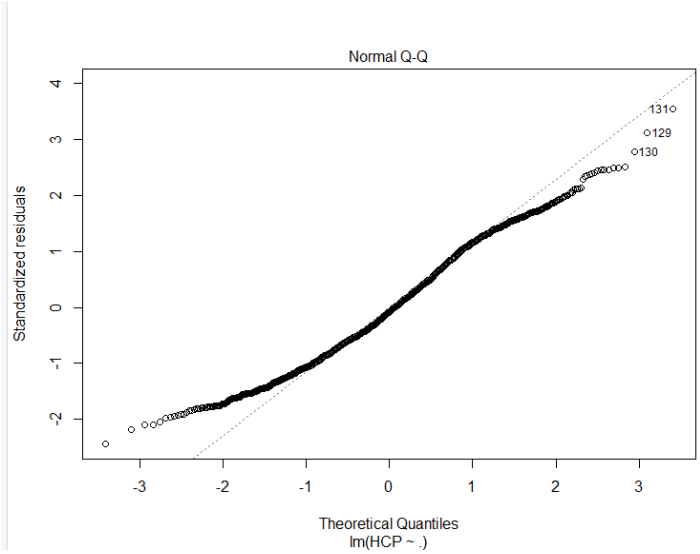
```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 45.09 on 1483 degrees of freedom
Multiple R-squared:  0.4223,    Adjusted R-squared:  0.3985
F-statistic: 17.77 on 61 and 1483 DF,  p-value: < 2.2e-16

> sqrt(mean(best1$residuals^2))
[1] 44.17121
> sqrt(.3985)
[1] 0.6312686
```

QQ Plots:

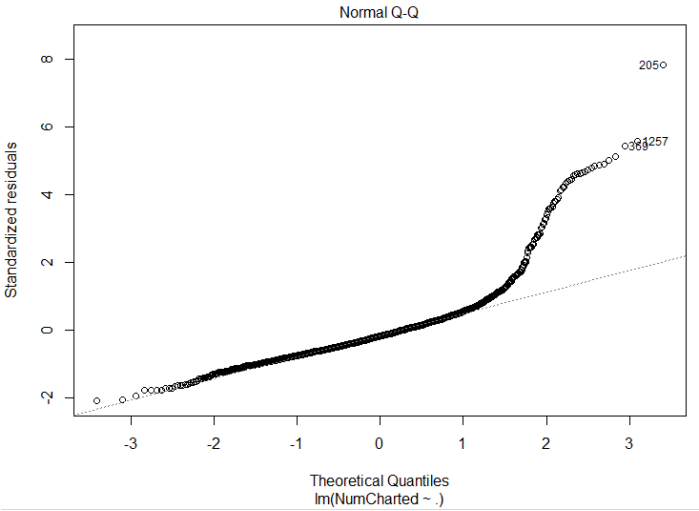
Streams as the dependent variable:



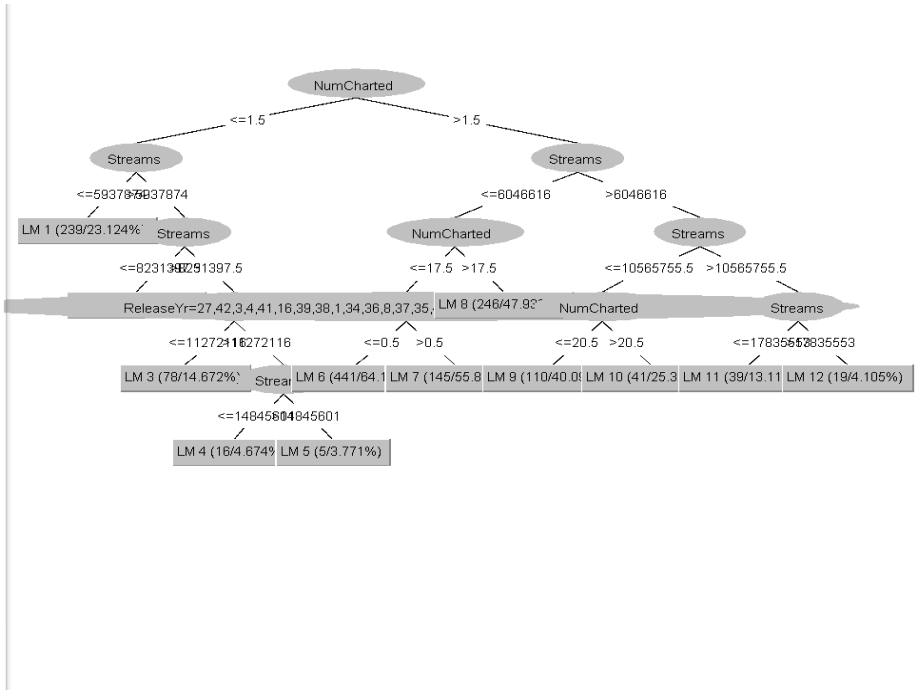
Highest charting position as the dependent variable:



NumCharted as the dependent variable:



Model tree produced from Weka with HCP as the class attribute:


NumCharted
<=1.5
>1.5
Streams
Streams
<=6046616
>6046616
NumCharted
Streams
<=17.5
>17.5
<=10565755.5
>10565755.5
Streams
<=0.5
>0.5
<=20.5
>20.5
LM 1 (239/23.124%
LM 6 (441/64.1
LM 7 (145/55.8
LM 9 (110/40.0
LM 10 (41/25.3
LM 11 (39/13.11
LM 12 (19/4.105%)
LM 4 (16/4.674%
LM 5 (5/3.771%)

RScript:

ssion_Diagnostics (2).R × | Spotify script.R × | Useful_R_Stuff (1).r × | NewScript1.R* × | df ×

Source on Save

```
df <- read.csv("C:/Users/foste/Downloads/Spot.csv")


library(dplyr)
library(foreign)


#make chr factors
str(df)
df$ReleaseSeason <- factor(df$ReleaseSeason)
df$ReleaseYr <- factor(df$ReleaseYr)
#df$Genre <- factor(df$Genre)
str(df)

#write arff file
?write.arff
arff1 <- write.arff(df, file="Spot.arff")

#summarize our data
summary(df)
#corr matrix
?cor
cor(df[ ,1:14])

#quantiles- we can do the quantiles of our dependent variables to determine where/if to split data
quantile(df$Streams,probs = seq(0, 1, 1/10))
quantile(df$HCP,probs = seq(0, 1, 1/10))
quantile(df$NumCharted,probs = seq(0, 1, 1/10))

#histograms
hist(df$Streams)
hist(df$HCP)
hist(df$NumCharted)
hist(df$Popularity)

#first linear model with streams that resulted in poor QQ plot
new1 <- lm(Streams ~ .,data=df)
plot(new1)
summary(new1)

#first linear model with Highest charting position as dependent
#resulted in a nice looking QQ plot that fit well to the linear line
new2 <-lm(HCP ~ .,data=df)
plot(new2)
summary(new2)

#first linear model for NumCharted as dependent
#.368 r squared
new3 <-lm(NumCharted ~.,data=df)
plot(new3)
summary(new3)
```

```r
#splits for numcharted
#write all of the split to a arff file for weka
#split1
y1<- df %>%
  filter(NumCharted<=2)
mod<- lm(NumCharted ~ ., data=y1)
plot(mod)
summary(mod)

arff2 <- write.arff(y1, file="1stSplit.arff")

#split2
y2<- df %>%
  filter(NumCharted>=3)%>%
  filter(HCP<=7)
mod1<- lm(NumCharted ~ ., data=y2)
plot(mod1)
summary(mod1)

arff3 <- write.arff(y2, file="2ndSplit.arff")

#split3
y3<- df %>%
  filter(NumCharted>=8)%>%
  filter(HCP<=30)
mod2<- lm(NumCharted ~ ., data=y3)
plot(mod2)
summary(mod2)

arff4 <- write.arff(y3, file="3rdSplit.arff")


#split4
y4<- df %>%
  filter(NumCharted>=31)%>%
  filter(HCP<=142)
mod3<- lm(NumCharted ~ ., data=y4)
plot(mod3)
summary(mod3)

arff5 <- write.arff(y4, file="4thSplit.arff")

#after running models in weka, we decided to make only 2 splits; the first split the below o
#split5
y6<- df %>%
  filter(NumCharted>=3)%>%
  filter(HCP<=142)
mod6<- lm(NumCharted ~ ., data=y4)
plot(mod6)
summary(mod6)

arff5 <- write.arff(y6, file="5thSplit.arff")
```

```r
#TO DO: splits, stepAIC, trees/ regression models in weka

## Feature selection for OLS Regression
# page 203 of the textbook
# Stepwise Backward, Forward or both
library(MASS)

stepAIC(new2,direction="backward")
stepAIC(new2, direction="forward")
stepAIC(new2, direction="both")

#selected by the backward splitAIC command .397
best <- lm(HCP ~ NumCharted + Streams + Followers + Popularity +
             Valence + ReleaseYr, data = df)
plot(best)
summary(best)

#selected by the forward and both splitAIC commands and it generates .3985 R squared
#this is the same as model as HCP ~. which I ran earlier in a above command
best1 <- lm(formula = HCP ~ NumCharted + Streams + Followers + Popularity +
              Danceability + Energy + Loudness + Speechiness + Acousticness +
              Liveness + Tempo + Duration + Valence + ReleaseSeason + NumGenre + ReleaseYr, data = df)
summary(best1)

#root mean squared error
sqrt(mean(best1$residuals^2))

#correlation
sqrt(.3985)
```

```r
#additional testing:
#filter best1 into 3 using the quantiles for HCP
#ultimately decided not to split on highest charting position because the QQ plot looked nice without filtering
#split1
s1 <- df %>%
  filter(HCP<=64)
ms1<- lm(HCP ~ ., data=s1)
plot(ms1)
summary(ms1)

#split2
s2 <- df %>%
  filter(64>=HCP) %>%
  filter(HCP<=148.0)
ms2<- lm(HCP ~ ., data=s2)
plot(ms2)
summary(ms2)

#split3
s3 <- df %>%
  filter(149.0>=HCP) %>%
  filter(HCP<=200)
ms3<- lm(HCP ~ ., data=s3)
plot(ms3)
summary(ms3)

#attempt to filter by streams
#decided not to use streams as dependent due to finding better ones such as HCP and NumCharted
#filter our data by streams using quantiles for guidance
split1 <- df %>%
  filter(Streams<=6034094)

split2 <- df %>%
  filter(Streams>=8960569)

#new models based on splits
SplitMod <- lm(Streams ~ ., data=split1)
plot(SplitMod)
summary(SplitMod)

SplitMod2 <- lm(Streams ~ ., data=split2)
plot(SplitMod2)


#using data set without num genre and with genre as a factor- .3987 r squared but higher error
test <-lm(formula = HCP ~ NumCharted + Streams + Followers + Popularity +
            Valence + ReleaseSeason + ReleaseYr, data = df)
summary(test)
```