Music Rank

Introduction and Executive Summary

Since we are all music lovers, we thought it would be interesting to see what features of a song make it popular. Music is used in various forms of entertainment, as a marketing tactic, for healing purposes, as a form of communication and for personal consumption. It has become an integral part of our society due to its endless power. We are curious to identify which qualities of a song results in its success in the market.

Our main objective for this project is that we want to use models to analyze the elements in what creates a popular song for different platforms. The purpose of doing this is to give individuals an understanding on what elements are taken more heavily into consideration for each respective platform in determining the popularity of the song.

Our project targets four platforms: Spotify, Apple, Deezer, and Shazam. People from around the world listen to music on all of these applications and each year they become increasingly more popular. Over the past few years, these platforms have become progressively well-known since more individuals stream music on them everyday. Each of these platforms create their own charts where the top streamed songs are ranked in 2023 and we want to determine how they came to that conclusion.

Background of the dataset

This data set contains the most streamed Spotify songs of 2023 collected as of July 2023. There are many metrics that decide what songs would make it on the charts including streams, listener engagement, social media presence, and more. For this project, we don't take these variables into consideration. Our main focus is analyzing the intrinsic features of these "popular songs" determined by the organizer of this data. A note to take into account is that each chart has a different amount of rankings. For example, Spotify's highest rank is 147, Apple's is 275, Deezer's is 58, and Shazam's is 1451. We were unable to find the total song count of each chart, so we will take the chart value at face value. 1 means the best and anything after that is not as good but still good enough to be on the charts.

Data Description

This dataset lists the most popular songs of 2023 as listed on Spotify:
https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023

The dataset originally contained 24 variables and 954 observations. To tidy the data, we removed all the blanks, zeroes, and input errors. We also removed the variables track names, artist count, artist names and the variables dependent on time such as released year, released month, released day, Spotify playlist, Apple playlist, Deezer playlist, and streams. Resulting in 14 variables and 953 observations.

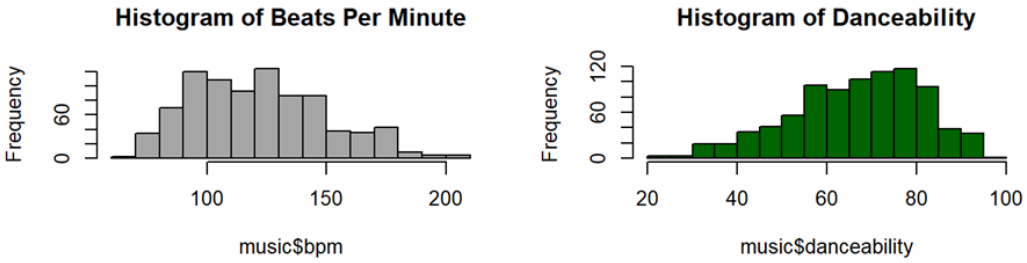
In order to interpret the preferred features of each chart we will have four response variables. Spotify Charts with 493 observations, Apple Charts with 764 observations, Shazam Charts with 508 observations, and Deezer Charts with 353 observations.

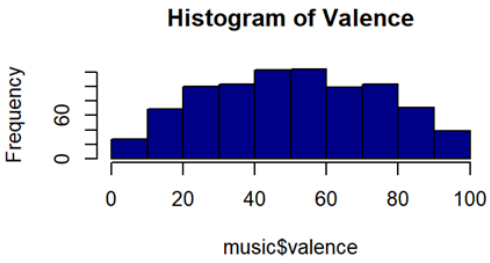
- spotify_charts: Presence and rank of the song on Spotify charts
- apple_charts: Presence and rank of the song on Apple Music charts
- deezer_charts: Presence and rank of the song on Deezer charts
- shazam_charts: Presence and rank of the song on Shazam charts
- bpm: Beats per minute, a measure of song tempo
- key: Key of the song

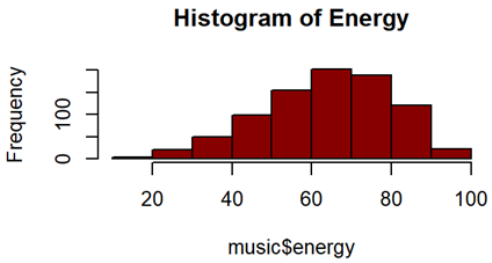
- mode: Mode of the song (major or minor)
- danceability_%: Percentage indicating how suitable the song is for dancing
- valence_%: Positivity of the song's musical content
- energy_%: Perceived energy level of the song
- acousticness_%: Amount of acoustic sound in the song
- instrumentalness_%: Amount of instrumental content in the song
- liveness_%: Presence of live performance elements
- speechiness_%: Amount of spoken words in the song

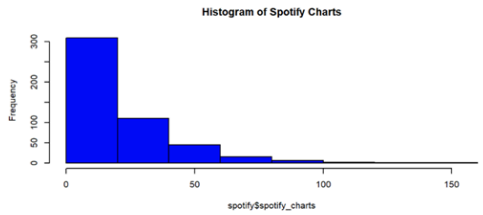
Data Exploration

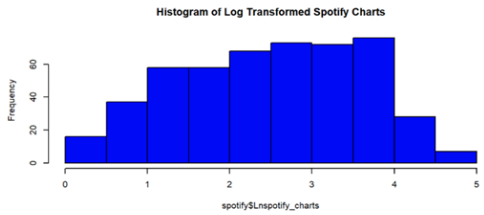
In order to understand the distribution of each of our variables we generated the following histograms.

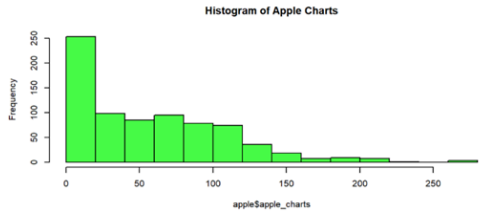


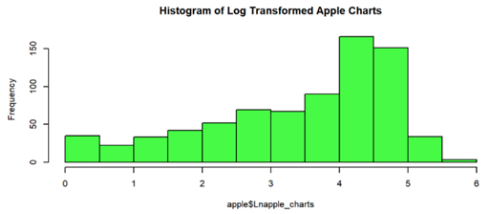


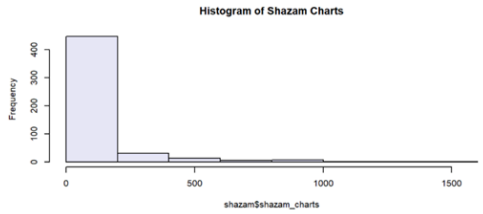


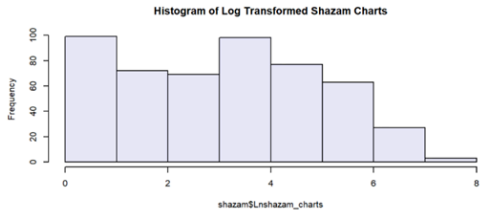


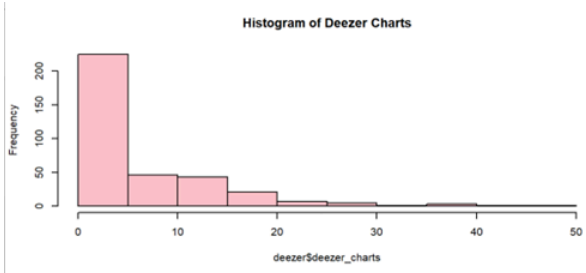


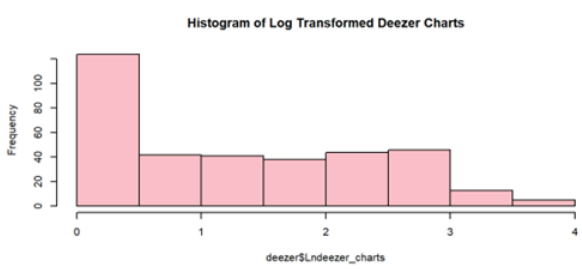


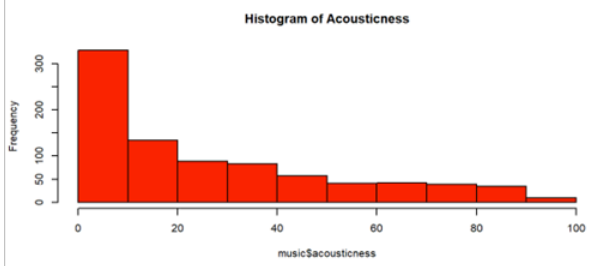


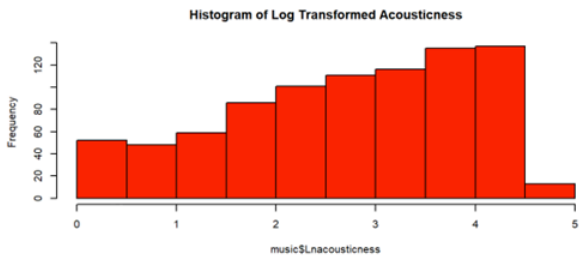


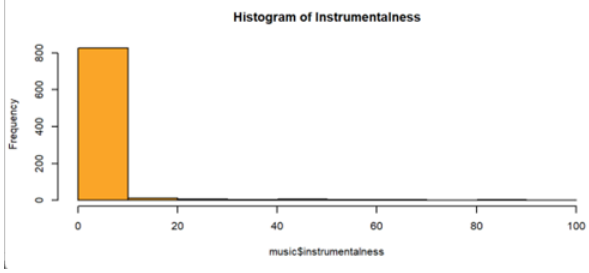


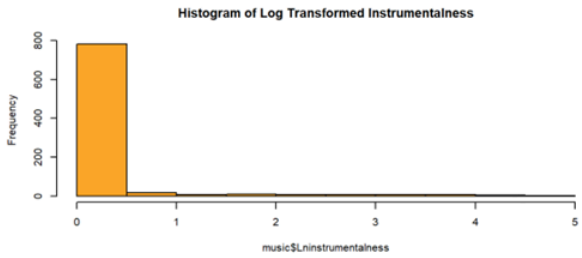

Histogram of Deezer Charts
Frequency
0
50
100
150
200
0
10
20
30
40
50
deezer$deezer_charts

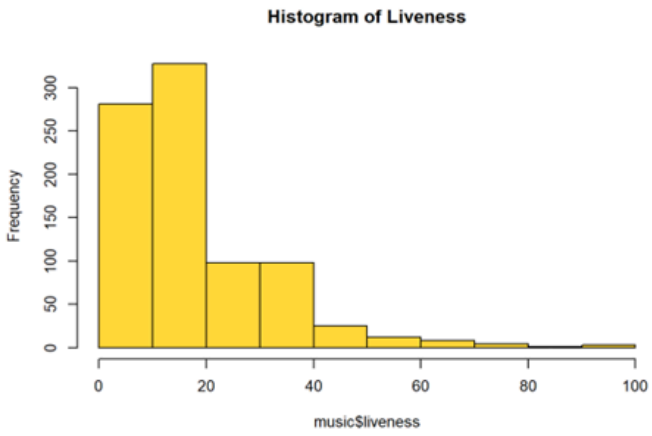

Histogram of Log Transformed Deezer Charts
Frequency
0
20
40
60
80
100
0
1
2
3
4
deezer$Lndeezer_charts

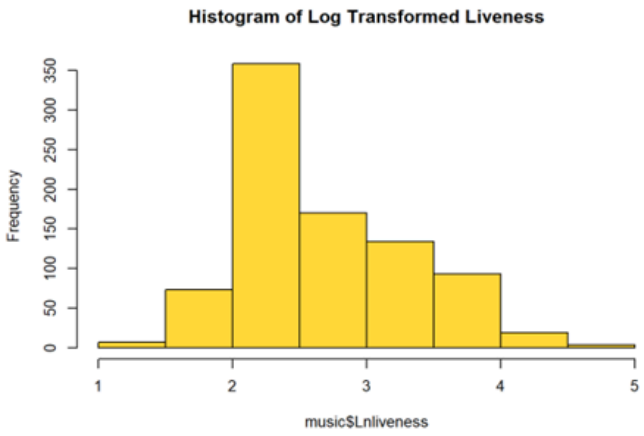

Histogram of Acousticness
Frequency
0
50
100
200
300
0
20
40
60
80
100
music$acousticness

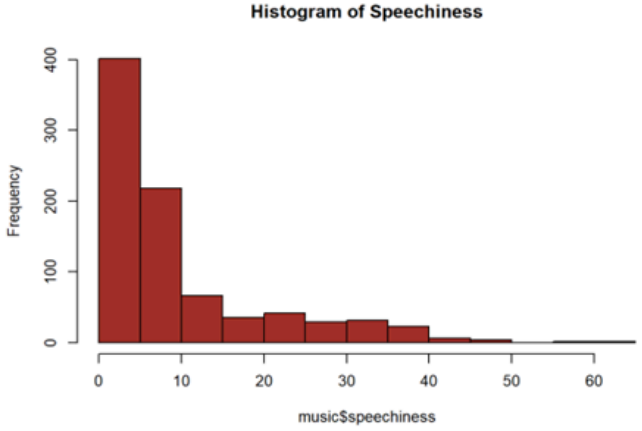

Histogram of Log Transformed Acousticness
Frequency
0
20
40
60
80
120
0
1
2
3
4
5
music$Lnacousticness

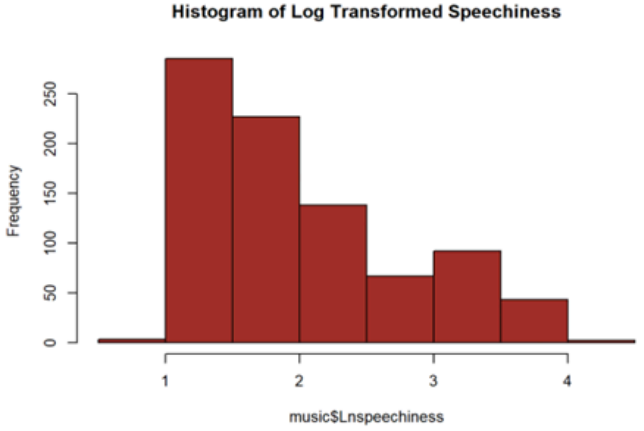

Histogram of Instrumentalness
Frequency
0
200
400
600
800
0
20
40
60
80
100
music$instrumentalness


Histogram of Log Transformed Instrumentalness
Frequency
0
200
400
600
800
0
1
2
3
4
5
music$Lninstrumentalness


Histogram of Liveness
Frequency
0
50
100
150
200
250
300
0
20
40
60
80
100
music$liveness


Histogram of Log Transformed Liveness
Frequency
0
50
100
150
200
250
300
350
1
2
3
4
5
music$Lnliveness


Histogram of Speechiness
Frequency
0
100
200
300
400
0
10
20
30
40
50
60
music$speechiness


Histogram of Log Transformed Speechiness
Frequency
0
50
100
150
200
250
1
2
3
4
music$Lnspeechiness

Since our outcome is continuous, we would like to run a linear regression and thus log transform the highly skewed variables to follow the normality criteria. Log transforming skewed variables normalizes their distribution, stabilizes the variance, eliminates trends making it more linear, and reduces outliers. However, Lnshazam charts, Lndeezer charts, Lnacousticness, Lninstrumentalness, and Lnspeechiness still do not look normal so we will need to inspect their residuals.

Listed below are the summary statistics of each our outcomes and their variables:

```
> summary(spotify)
 spotify_charts        bpm              key          mode       danceability
 Min.   :  1.00   Min.   : 67.0   C#     : 69   Major:270   Min.   :29.00
 1st Qu.:  5.00   1st Qu.:100.0   F      : 59   Minor:223   1st Qu.:58.00
 Median : 13.00   Median :122.0   B      : 52               Median :70.00
 Mean   : 20.79   Mean   :124.6   G      : 51               Mean   :67.56
 3rd Qu.: 31.00   3rd Qu.:142.0   F#     : 48               3rd Qu.:78.00
 Max.   :147.00   Max.   :206.0   A      : 43               Max.   :95.00
                                  (Other):171
    valence          energy       acousticness    instrumentalness    liveness
 Min.   : 4.00   Min.   :25.0   Min.   : 0.00   Min.   : 0.000   Min.   : 4.00
 1st Qu.:36.00   1st Qu.:56.0   1st Qu.: 5.00   1st Qu.: 0.000   1st Qu.:10.00
 Median :53.00   Median :67.0   Median :18.00   Median : 0.000   Median :12.00
 Mean   :52.37   Mean   :65.8   Mean   :25.55   Mean   : 1.101   Mean   :17.84
 3rd Qu.:70.00   3rd Qu.:77.0   3rd Qu.:40.00   3rd Qu.: 0.000   3rd Qu.:22.00
 Max.   :97.00   Max.   :97.0   Max.   :97.00   Max.   :90.000   Max.   :92.00

  speechiness     Lnspotify_charts   Lnliveness     Lnspeechiness
 Min.   : 2.000   Min.   :0.000    Min.   :1.386   Min.   :0.6931
 1st Qu.: 4.000   1st Qu.:1.609    1st Qu.:2.303   1st Qu.:1.3863
 Median : 5.000   Median :2.565    Median :2.485   Median :1.6094
 Mean   : 9.369   Mean   :2.494    Mean   :2.675   Mean   :1.9147
 3rd Qu.:10.000   3rd Qu.:3.434    3rd Qu.:3.091   3rd Qu.:2.3026
 Max.   :64.000   Max.   :4.990    Max.   :4.522   Max.   :4.1589

 Lnacousticness  Lninstrumentalness
 Min.   :0.000   Min.   :0.0000
 1st Qu.:1.792   1st Qu.:0.0000
 Median :2.944   Median :0.0000
 Mean   :2.682   Mean   :0.1495
 3rd Qu.:3.714   3rd Qu.:0.0000
 Max.   :4.585   Max.   :4.5109
```

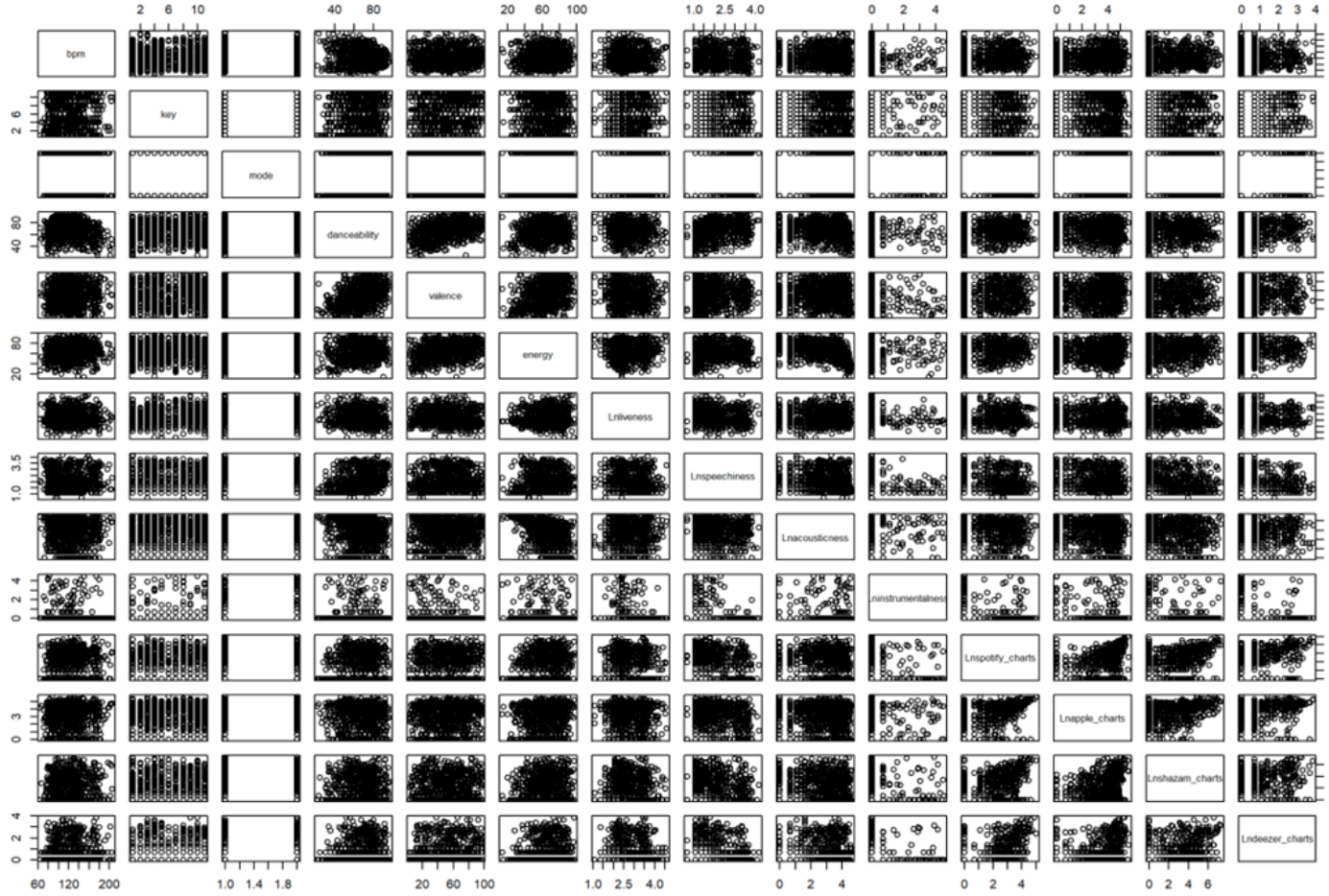
```
> summary(apple)
  apple_charts        bpm              key          mode       danceability
 Min.   :  1.00   Min.   : 65.0   C#     :105   Major:428   Min.   :23.00
 1st Qu.: 13.00   1st Qu.:100.0   G      : 85   Minor:336   1st Qu.:57.00
 Median : 47.50   Median :120.0   G#     : 84               Median :70.00
 Mean   : 57.26   Mean   :122.7   F      : 79               Mean   :67.31
 3rd Qu.: 90.00   3rd Qu.:140.0   D      : 75               3rd Qu.:79.00
 Max.   :275.00   Max.   :206.0   B      : 73               Max.   :96.00
                                  (Other):263
    valence          energy       acousticness    instrumentalness
 Min.   : 4.00   Min.   :14.00   Min.   : 0.00   Min.   : 0.000
 1st Qu.:32.00   1st Qu.:54.00   1st Qu.: 5.00   1st Qu.: 0.000
 Median :51.00   Median :66.00   Median :17.00   Median : 0.000
 Mean   :51.15   Mean   :64.78   Mean   :26.05   Mean   : 1.454
 3rd Qu.:69.25   3rd Qu.:77.00   3rd Qu.:41.00   3rd Qu.: 0.000
 Max.   :97.00   Max.   :97.00   Max.   :97.00   Max.   :90.000

    liveness      speechiness    Lnapple_charts   Lnliveness
 Min.   : 3.00   Min.   : 2.00   Min.   :0.000   Min.   :1.099
 1st Qu.:10.00   1st Qu.: 4.00   1st Qu.:2.565   1st Qu.:2.303
 Median :12.00   Median : 6.00   Median :3.861   Median :2.485
 Mean   :18.39   Mean   :10.19   Mean   :3.418   Mean   :2.701
 3rd Qu.:25.00   3rd Qu.:11.00   3rd Qu.:4.500   3rd Qu.:3.219
 Max.   :97.00   Max.   :64.00   Max.   :5.617   Max.   :4.575

 Lnspeechiness   Lnacousticness  Lninstrumentalness
 Min.   :0.6931   Min.   :0.000   Min.   :0.0000
 1st Qu.:1.3863   1st Qu.:1.792   1st Qu.:0.0000
 Median :1.7918   Median :2.890   Median :0.0000
 Mean   :1.9880   Mean   :2.698   Mean   :0.1841
 3rd Qu.:2.3979   3rd Qu.:3.738   3rd Qu.:0.0000
 Max.   :4.1589   Max.   :4.585   Max.   :4.5109
```

```
> summary(shazam)
 shazam_charts        bpm              key          mode       danceability
 Min.   :   1.00   Min.   : 67.0   C#     : 66   Major:281   Min.   :24.00
 1st Qu.:   4.00   1st Qu.:100.0   G      : 57   Minor:227   1st Qu.:58.00
 Median :  24.00   Median :122.0   F      : 55               Median :70.00
 Mean   :  92.56   Mean   :123.3   D      : 51               Mean   :67.69
 3rd Qu.:  84.25   3rd Qu.:140.0   B      : 49               3rd Qu.:78.00
 Max.   :1451.00   Max.   :206.0   G#     : 48               Max.   :96.00
                                   (Other):182
    valence          energy       acousticness    instrumentalness
 Min.   : 4.00   Min.   :14.00   Min.   : 0.00   Min.   : 0.000
 1st Qu.:32.00   1st Qu.:55.00   1st Qu.: 5.00   1st Qu.: 0.000
 Median :51.00   Median :66.00   Median :17.00   Median : 0.000
 Mean   :51.09   Mean   :65.21   Mean   :25.34   Mean   : 1.659
 3rd Qu.:70.00   3rd Qu.:77.00   3rd Qu.:39.25   3rd Qu.: 0.000
 Max.   :97.00   Max.   :97.00   Max.   :97.00   Max.   :91.000

    liveness      speechiness    Lnshazam_charts   Lnliveness
 Min.   : 3.00   Min.   : 2.00   Min.   :0.000   Min.   :1.099
 1st Qu.:10.00   1st Qu.: 4.00   1st Qu.:1.386   1st Qu.:2.303
 Median :12.00   Median : 6.00   Median :3.178   Median :2.485
 Mean   :17.83   Mean   :10.02   Mean   :2.992   Mean   :2.684
 3rd Qu.:23.00   3rd Qu.:11.00   3rd Qu.:4.434   3rd Qu.:3.135
 Max.   :92.00   Max.   :64.00   Max.   :7.280   Max.   :4.522

 Lnspeechiness   Lnacousticness  Lninstrumentalness
 Min.   :0.6931   Min.   :0.000   Min.   :0.0000
 1st Qu.:1.3863   1st Qu.:1.792   1st Qu.:0.0000
 Median :1.7918   Median :2.890   Median :0.0000
 Mean   :1.9696   Mean   :2.679   Mean   :0.1939
 3rd Qu.:2.3979   3rd Qu.:3.695   3rd Qu.:0.0000
 Max.   :4.1589   Max.   :4.585   Max.   :4.5218
```

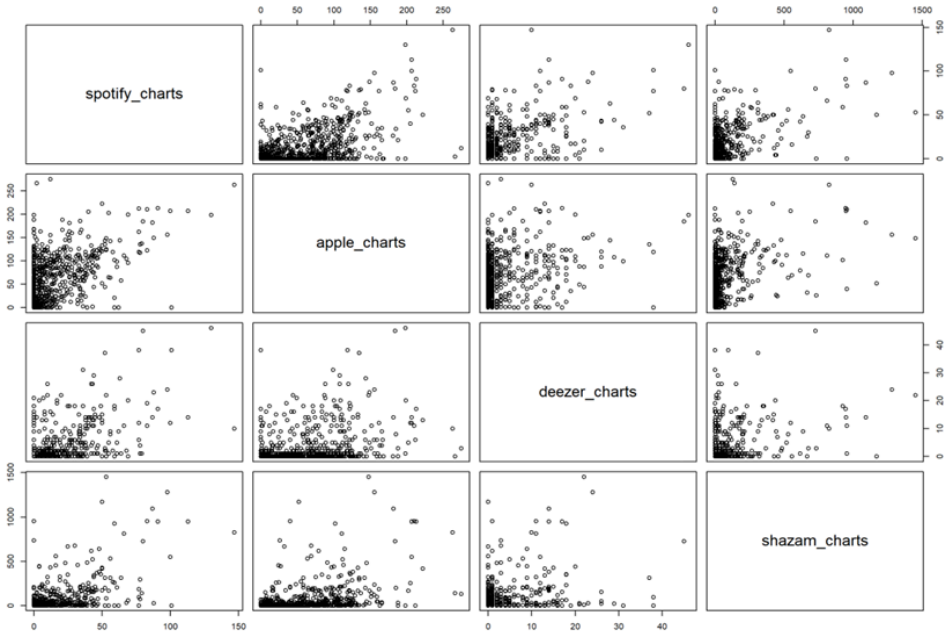
```
> summary(deezer)
 deezer_charts        bpm              key          mode       danceability
 Min.   : 1.00   Min.   : 67.0   C#     : 47   Major:187   Min.   :25.00
 1st Qu.: 1.00   1st Qu.:100.0   F      : 43   Minor:166   1st Qu.:59.00
 Median : 3.00   Median :123.0   G      : 42               Median :70.00
 Mean   : 6.36   Mean   :124.9   B      : 38               Mean   :68.93
 3rd Qu.: 9.00   3rd Qu.:142.0   F#     : 32               3rd Qu.:79.00
 Max.   :46.00   Max.   :204.0   G#     : 32               Max.   :95.00
                                 (Other):119
    valence          energy       acousticness    instrumentalness
 Min.   : 4.00   Min.   :14.00   Min.   : 0.00   Min.   : 0.000
 1st Qu.:38.00   1st Qu.:57.00   1st Qu.: 5.00   1st Qu.: 0.000
 Median :55.00   Median :68.00   Median :18.00   Median : 0.000
 Mean   :55.01   Mean   :66.99   Mean   :24.93   Mean   : 1.054
 3rd Qu.:75.00   3rd Qu.:78.00   3rd Qu.:37.00   3rd Qu.: 0.000
 Max.   :97.00   Max.   :97.00   Max.   :97.00   Max.   :63.000

    liveness      speechiness    Lndeezer_charts   Lnliveness
 Min.   : 4.00   Min.   : 2.000   Min.   :0.000   Min.   :1.386
 1st Qu.:10.00   1st Qu.: 4.000   1st Qu.:0.000   1st Qu.:2.303
 Median :12.00   Median : 6.000   Median :1.099   Median :2.485
 Mean   :18.47   Mean   : 9.708   Mean   :1.221   Mean   :2.711
 3rd Qu.:26.00   3rd Qu.:10.000   3rd Qu.:2.197   3rd Qu.:3.258
 Max.   :92.00   Max.   :64.000   Max.   :3.829   Max.   :4.522

 Lnspeechiness   Lnacousticness  Lninstrumentalness
 Min.   :0.6931   Min.   :0.000   Min.   :0.0000
 1st Qu.:1.3863   1st Qu.:1.792   1st Qu.:0.0000
 Median :1.7918   Median :2.944   Median :0.0000
 Mean   :1.9522   Mean   :2.701   Mean   :0.1433
 3rd Qu.:2.3026   3rd Qu.:3.638   3rd Qu.:0.0000
 Max.   :4.1589   Max.   :4.585   Max.   :4.1589
```

These statistics reiterate the skewness of some log transformed variables as the mean should be very close to the median for a normal distribution.

To further explore the data, we will visualize the relationships between the variables. First, we will use the following scatterplots.
This is a scatterplot of the entire dataset.

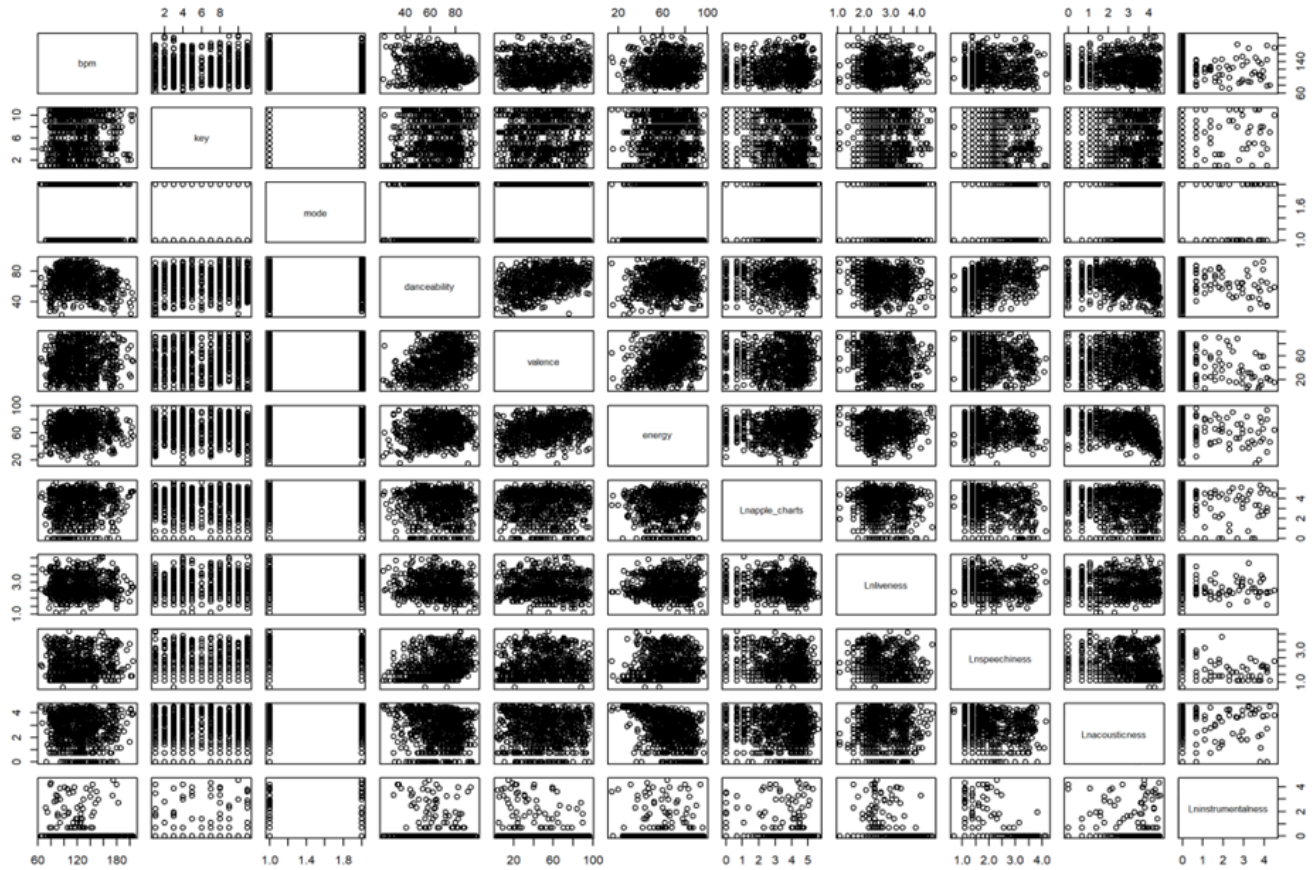


This is the scatterplot of response variables.

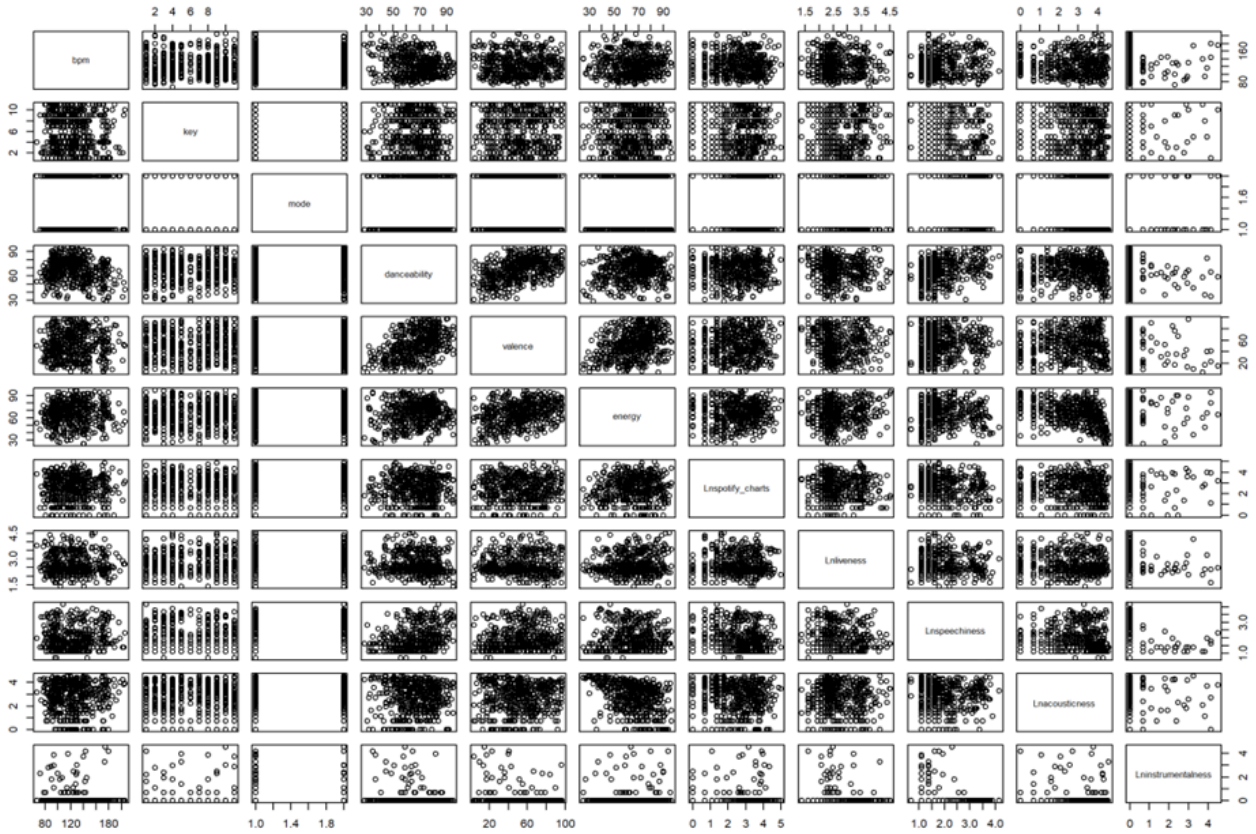


This shows that there is not a strong correlation between charts. A song ranked first on one chart could possibly not be included in another chart.

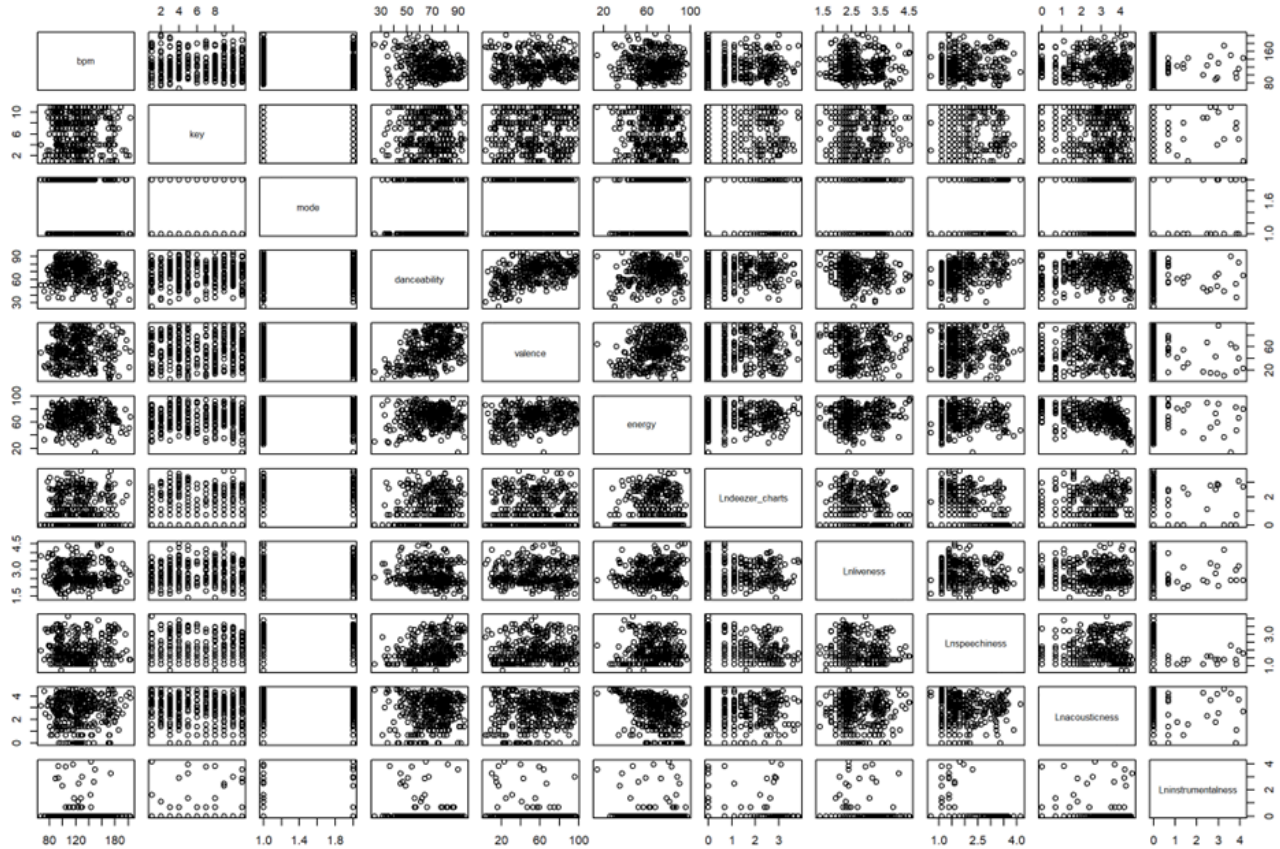
This is the scatterplot of LnApple Charts.



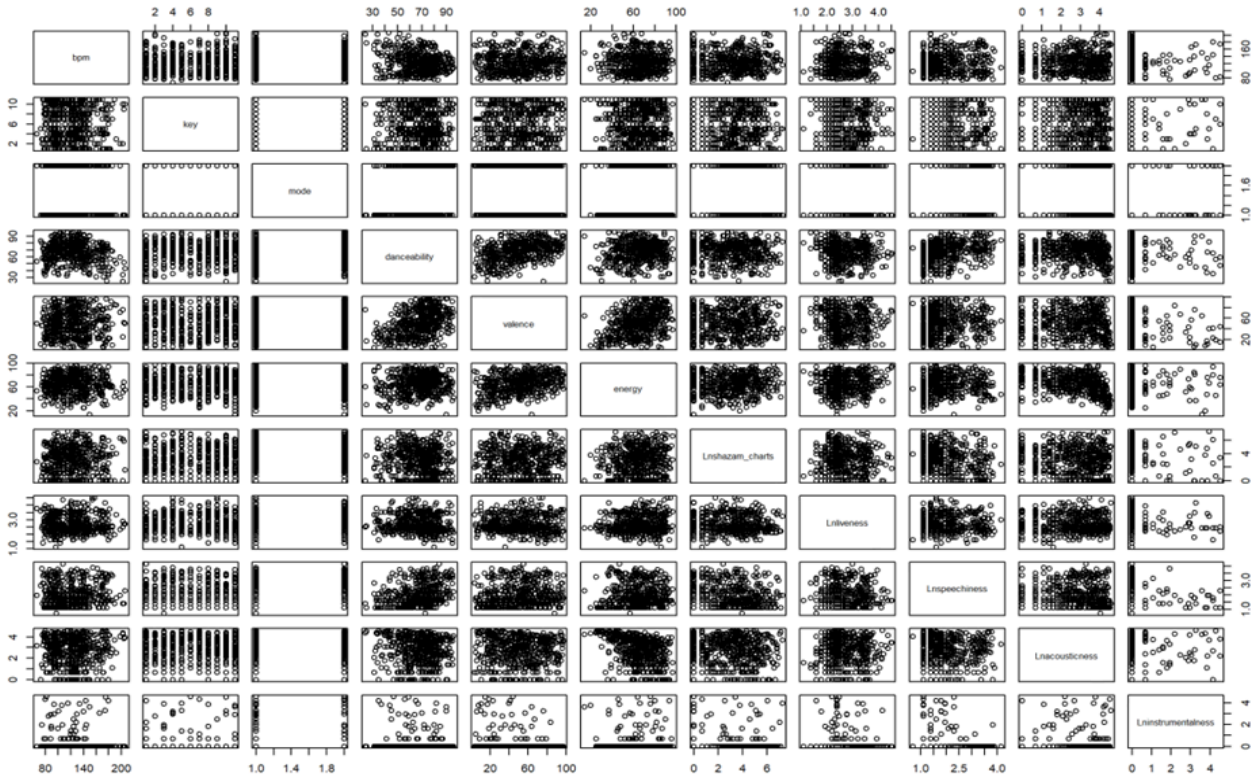
This is the scatterplot of Ln Spotify Charts.



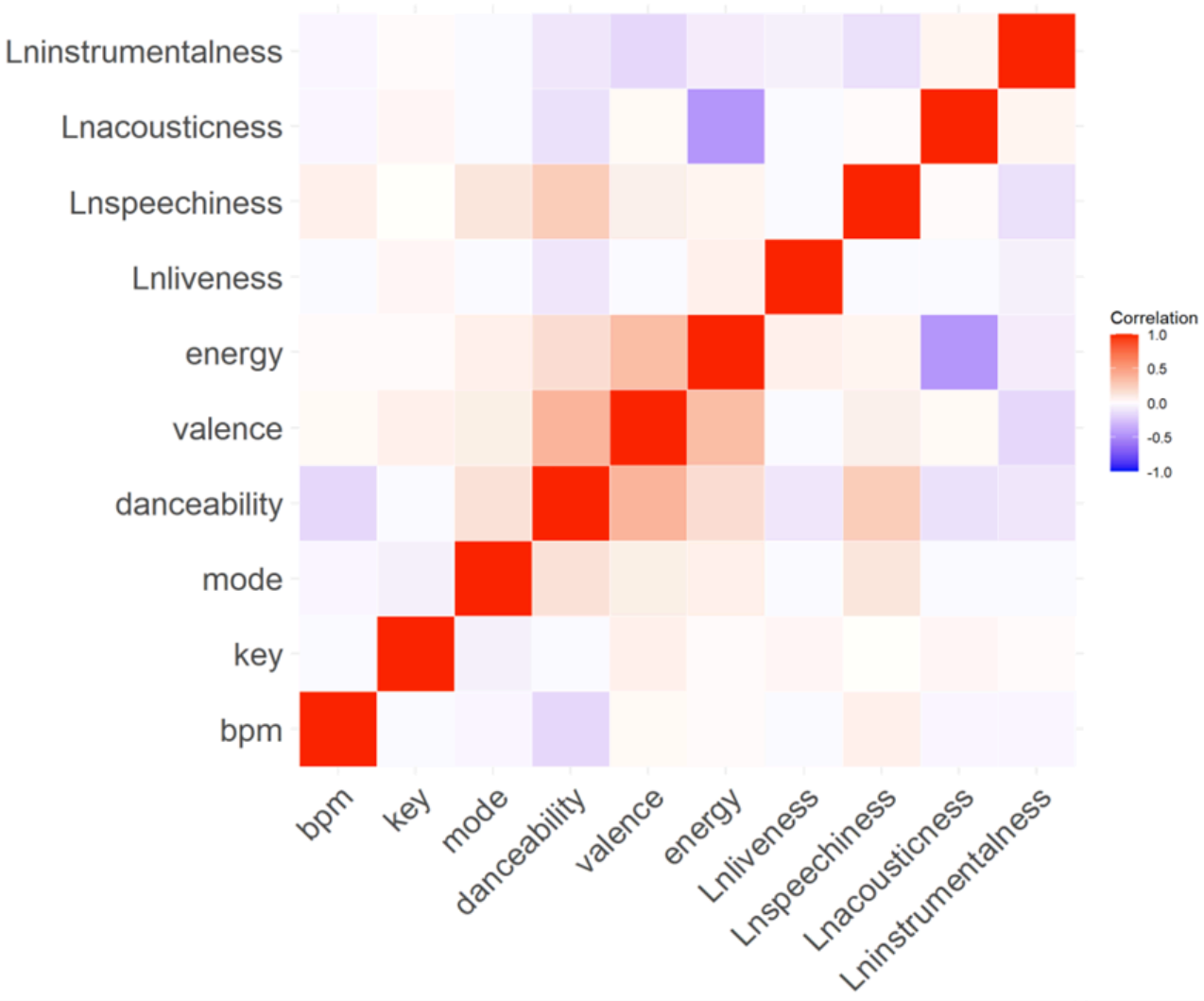
This is the scatterplot of Ln Deezer Charts.



This is the scatterplot of Ln Shazam Charts.

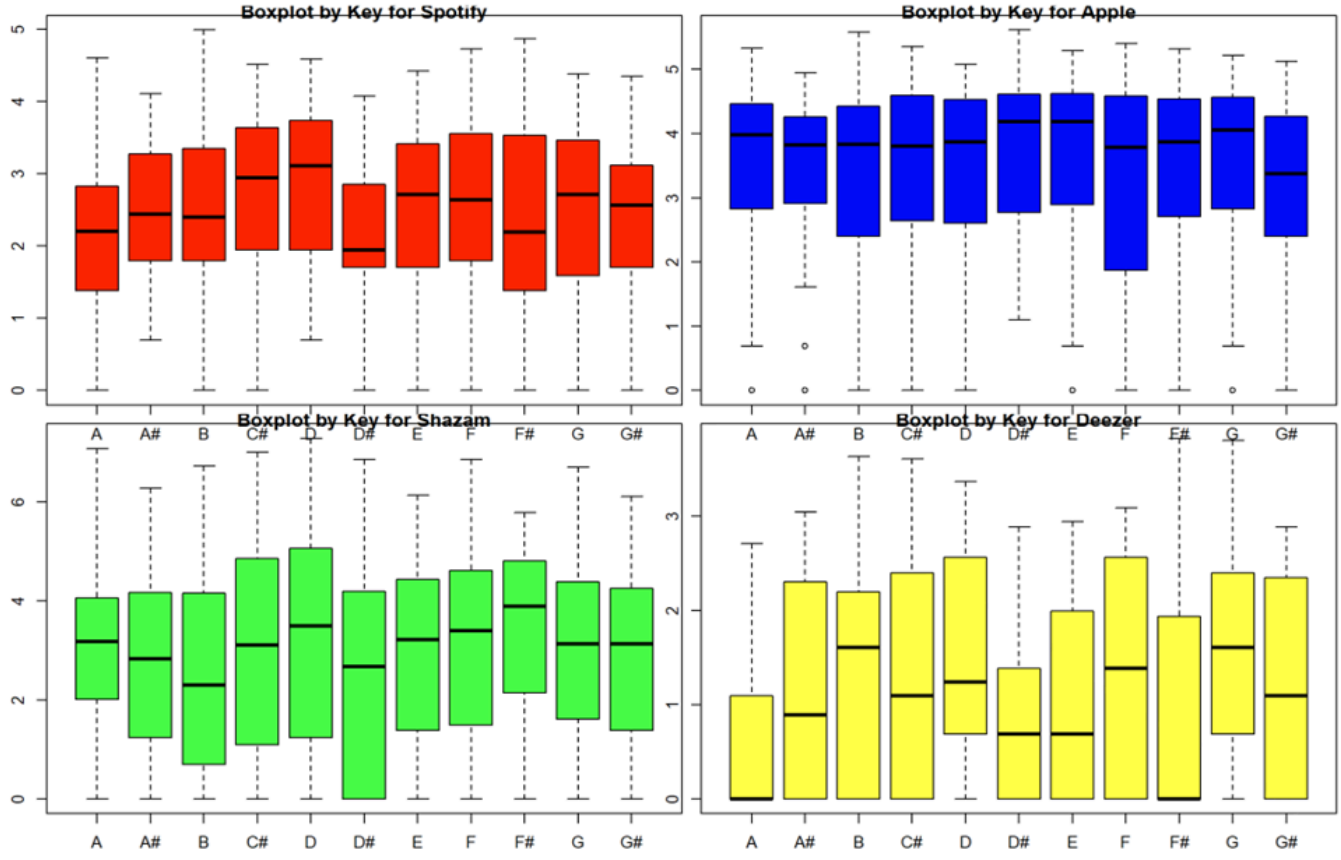


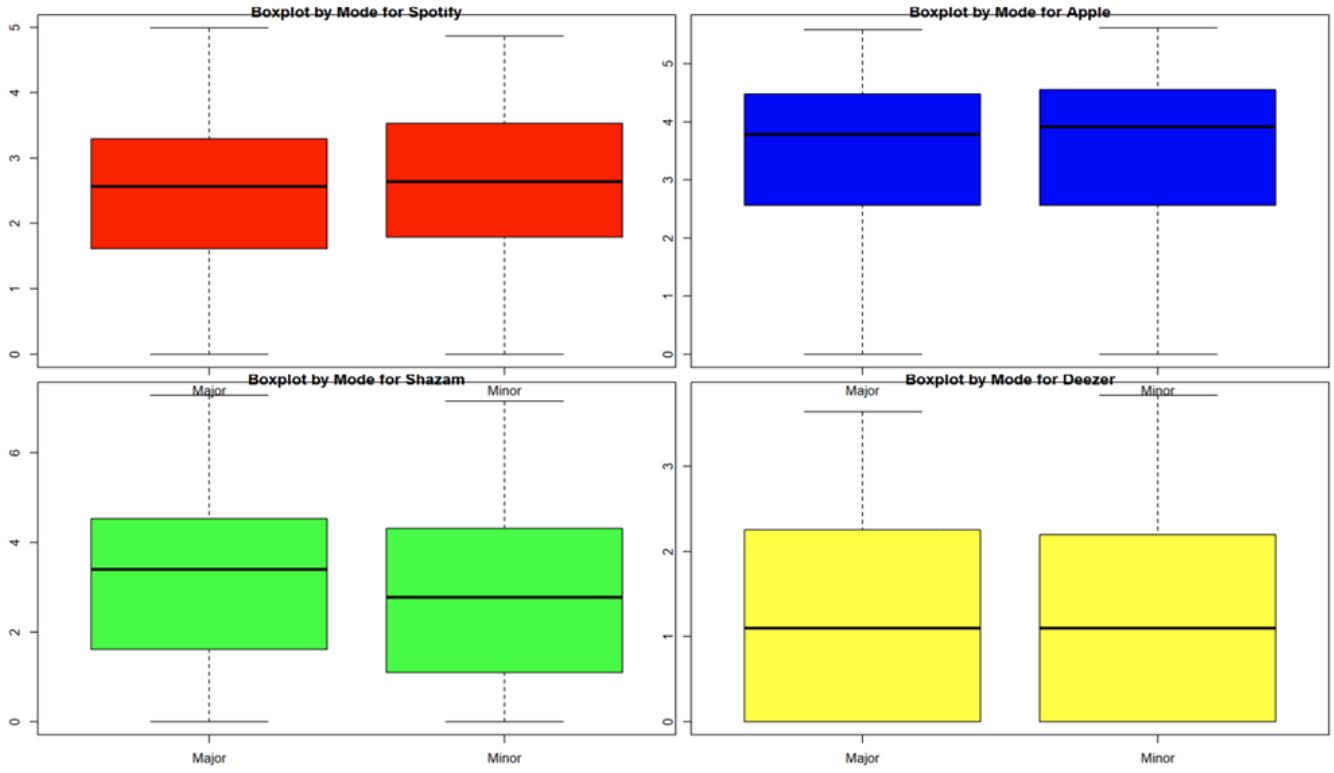
The following is a heatmap that also visualizes the correlation between variables. Red signifies perfectly positively correlated with a correlation coefficient of 1. Correlation decreases the lighter the red gets until it gets white which means not correlated at all with a correlation coefficient of 0. As the color gets darker purple the correlation coefficient decreases to -1.



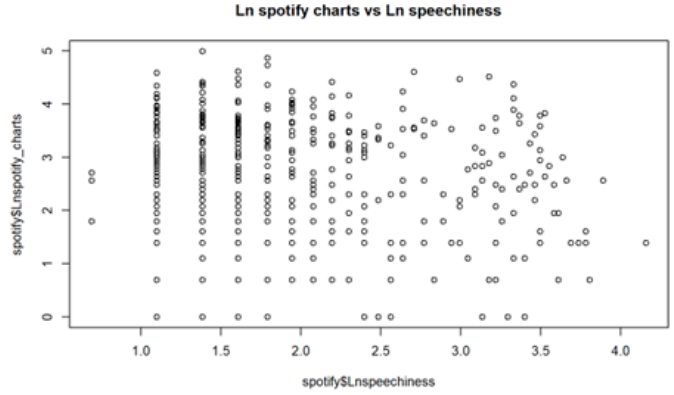
We can see that danceability and valence are strongly positively correlated as well as energy and valence. However, energy and danceability are not strongly positively correlated. Thus, we can note that popular songs who have a positive message are more likely to be danceable songs and as energy increases in a song the more positive the message of the song will be. Energy and acousticness are strongly negatively correlated. In other words, songs that are more earthy toned and created with natural instruments such as guitars and pianos tend to have less energy.

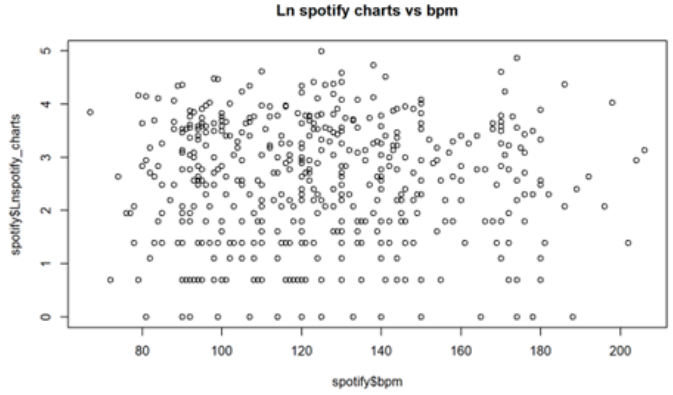
The following box plots are the relationships between the four outcome variables and our two qualitative variables: mode (2 levels) and key (11 levels).

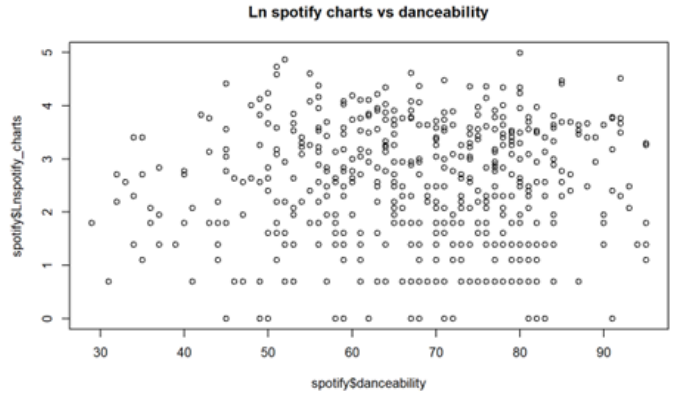


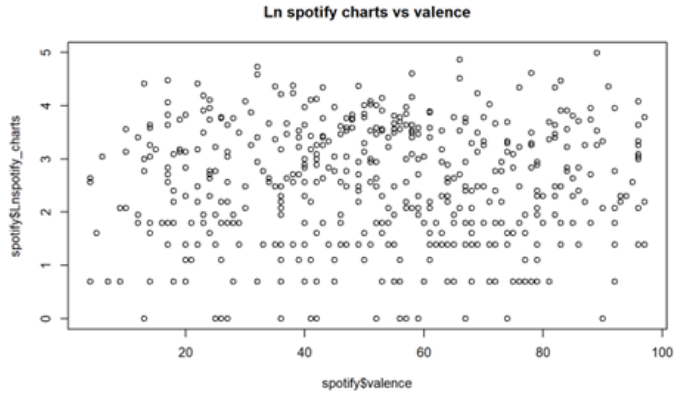


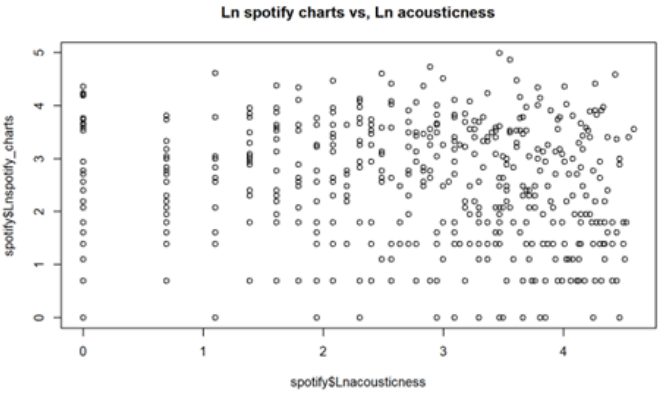
The following plots illustrate how the linear best fit line would capture the data. As we can see the best fit line would not be able to capture most of the data and would result in a high RSE but there is still some value we gain from the regressions.

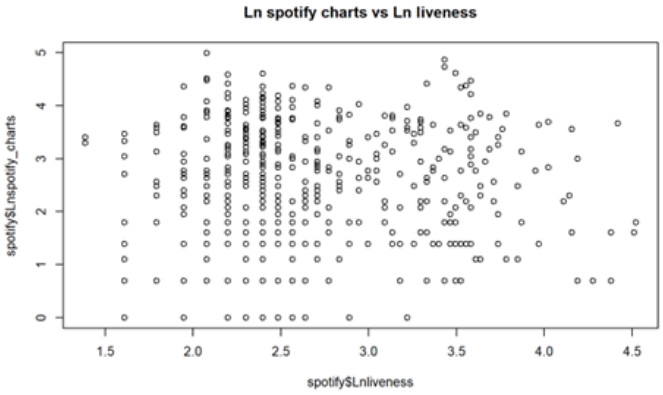


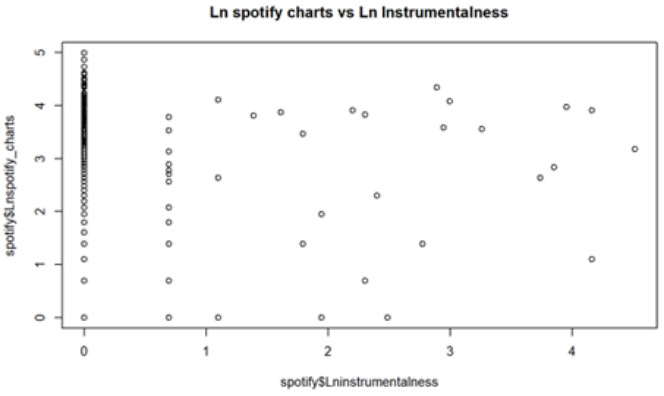


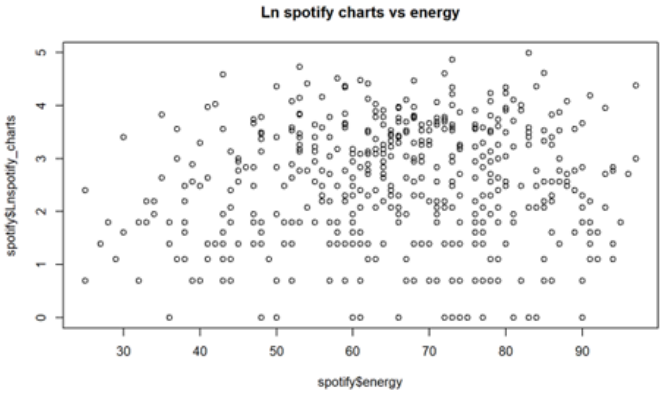


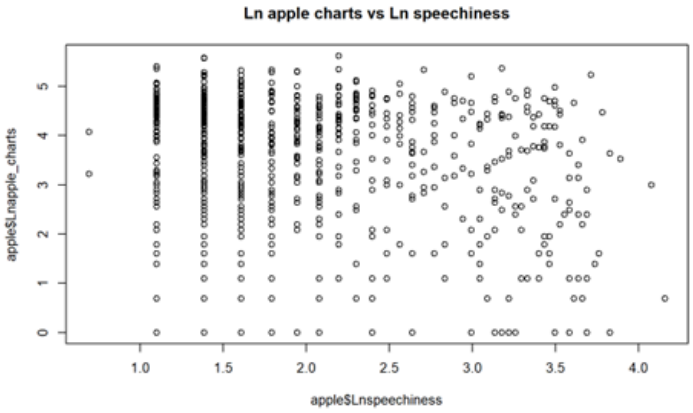


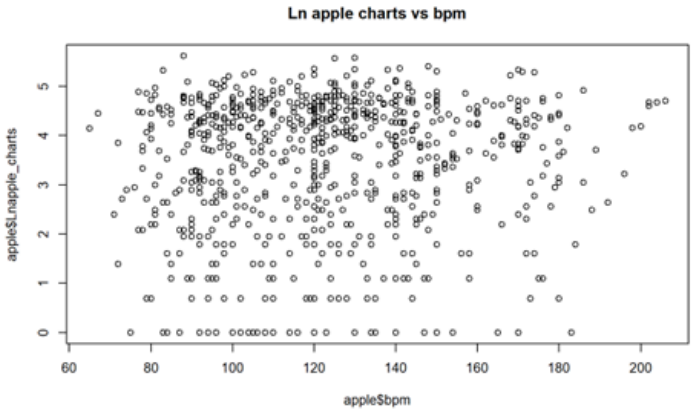


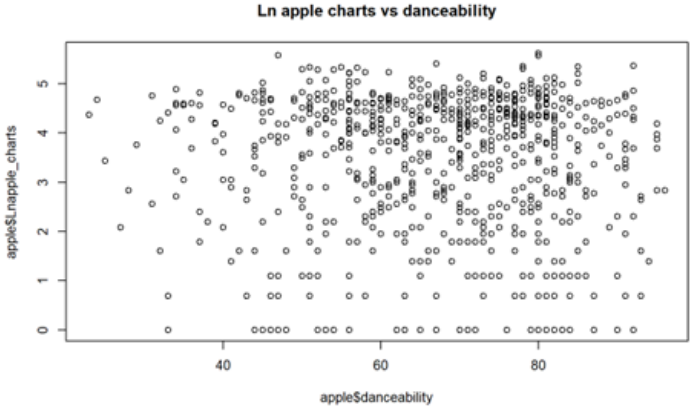


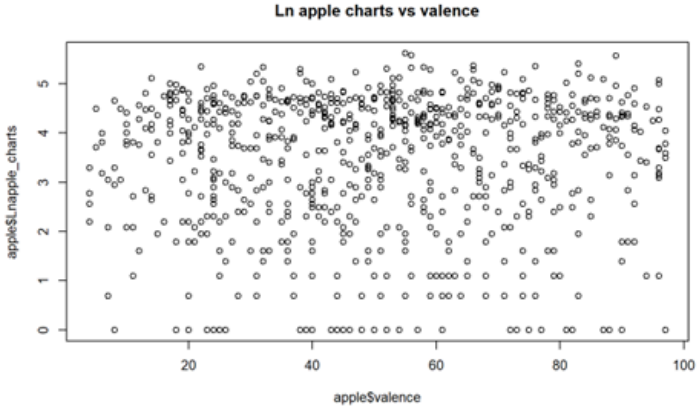


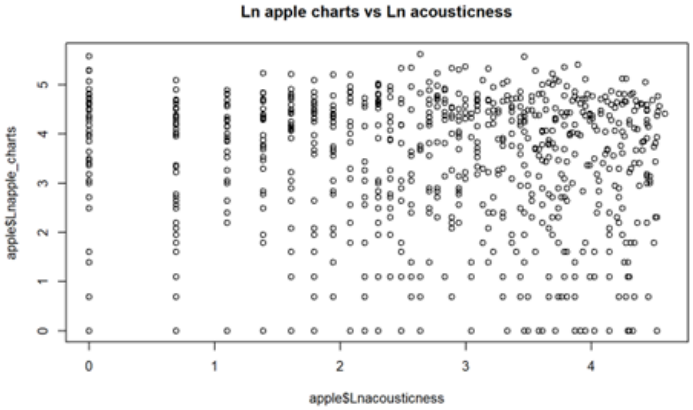


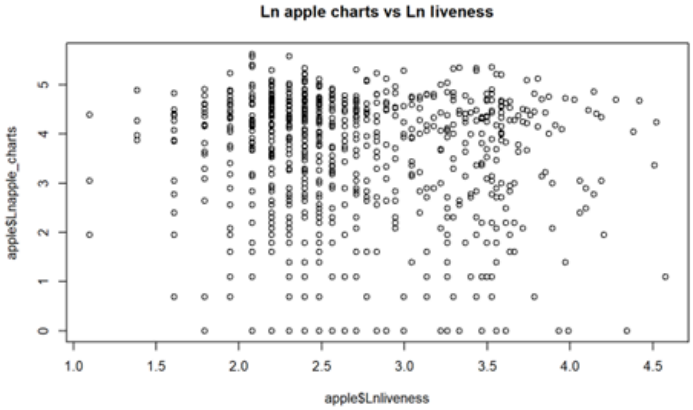

Ln apple charts vs Ln speechiness
apple$Lnapple_charts
apple$Lnspeechiness

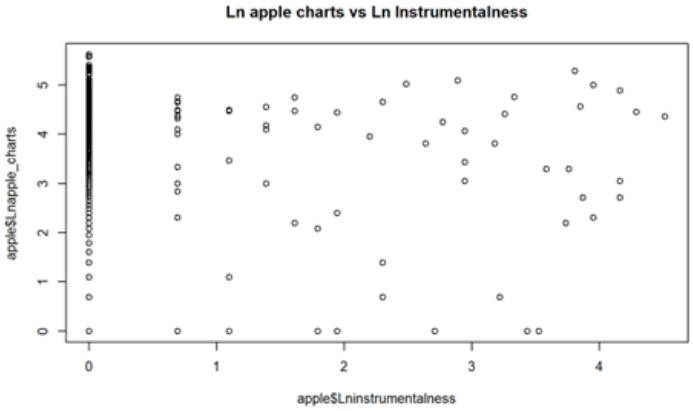

Ln apple charts vs bpm
apple$Lnapple_charts
apple$bpm

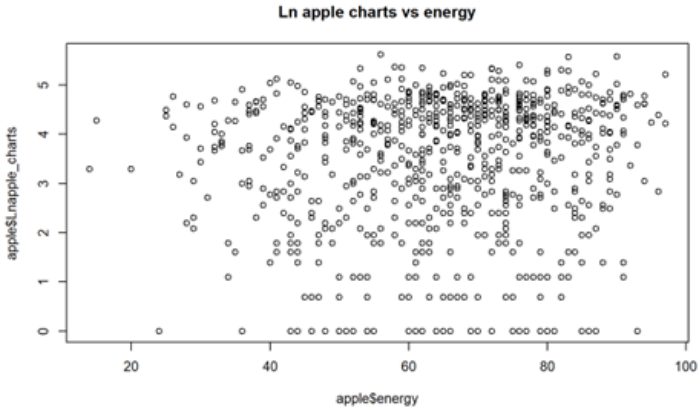

Ln apple charts vs danceability
apple$Lnapple_charts
apple$danceability

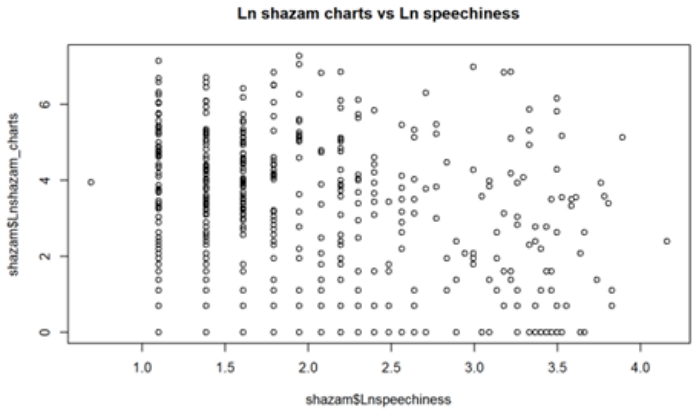

Ln apple charts vs valence
apple$Lnapple_charts
apple$valence

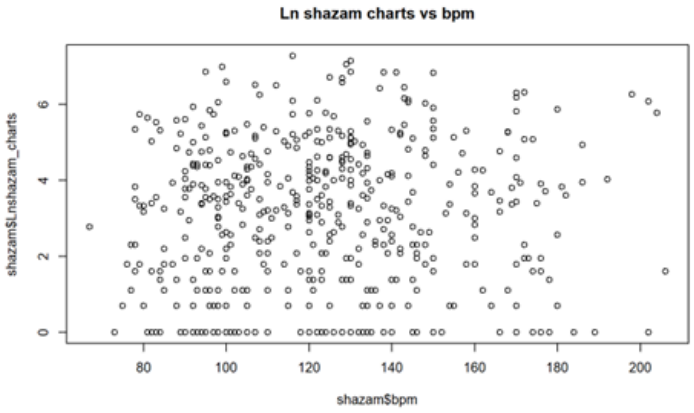

Ln apple charts vs Ln acousticness
apple$Lnapple_charts
apple$Lnacousticness

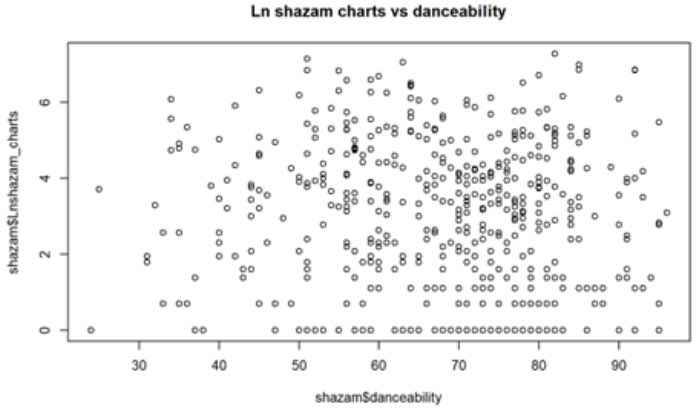

Ln apple charts vs Ln liveness
apple$Lnapple_charts
apple$Lnliveness

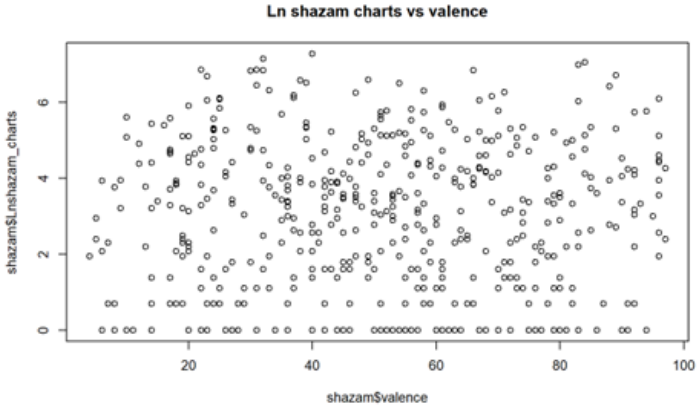

Ln apple charts vs Ln Instrumentalness
apple$Lnapple_charts
apple$Lninstrumentalness

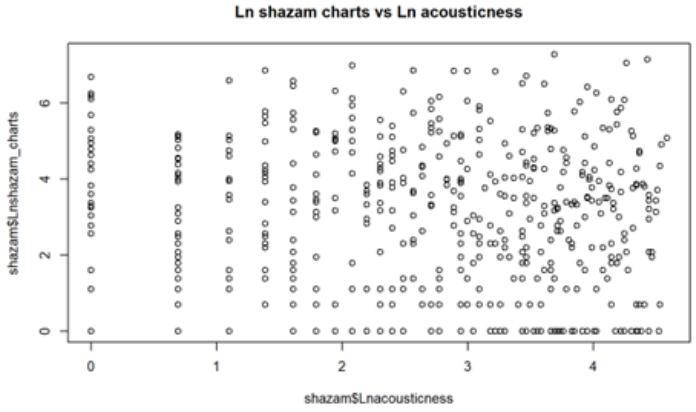

Ln apple charts vs energy
apple$Lnapple_charts
apple$energy

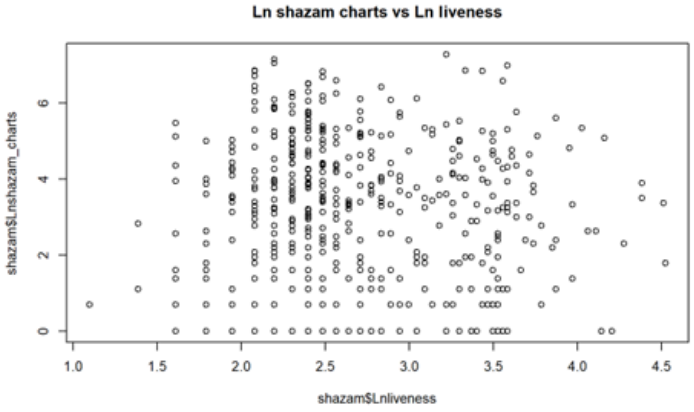

Ln shazam charts vs Ln speechiness
shazam$Lnshazam_charts
shazam$Lnspeechiness


Ln shazam charts vs bpm
shazam$Lnshazam_charts
shazam$bpm

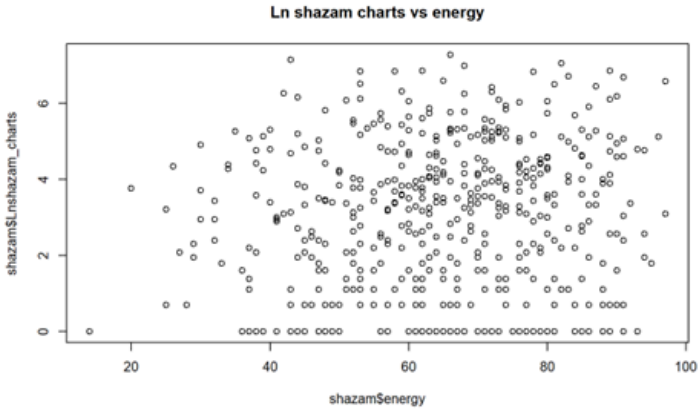

Ln shazam charts vs danceability
shazam$Lnshazam_charts
shazam$danceability

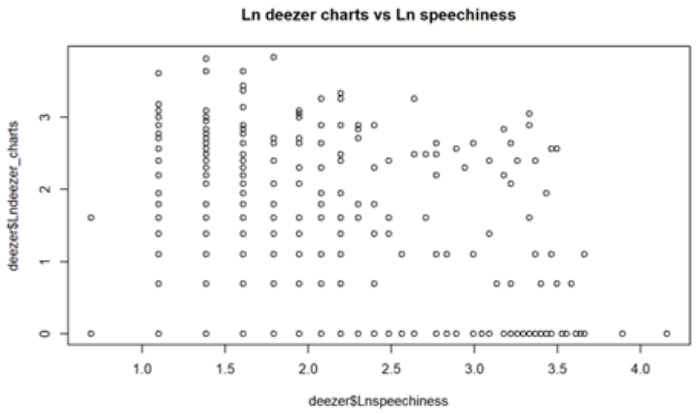

Ln shazam charts vs valence
shazam$Lnshazam_charts
shazam$valence

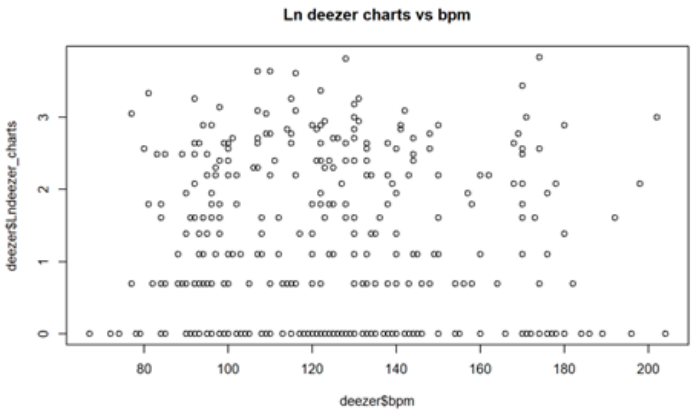

Ln shazam charts vs Ln acousticness
shazam$Lnshazam_charts
shazam$Lnacousticness

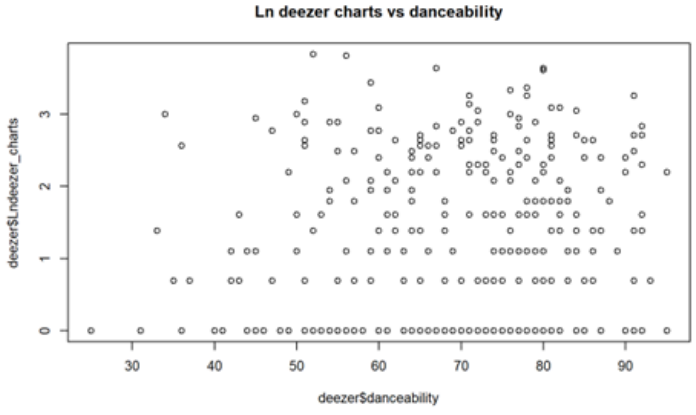

Ln shazam charts vs Ln liveness
shazam$Lnshazam_charts
shazam$Lnliveness

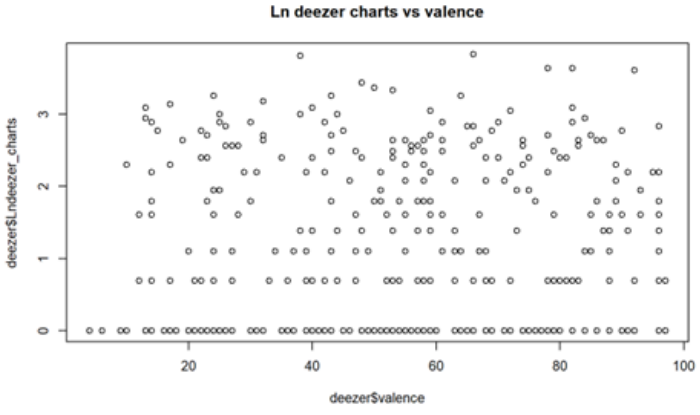

Ln shazam charts vs Ln Instrumentalness
shazam$Lnshazam_charts
shazam$Lninstrumentalness

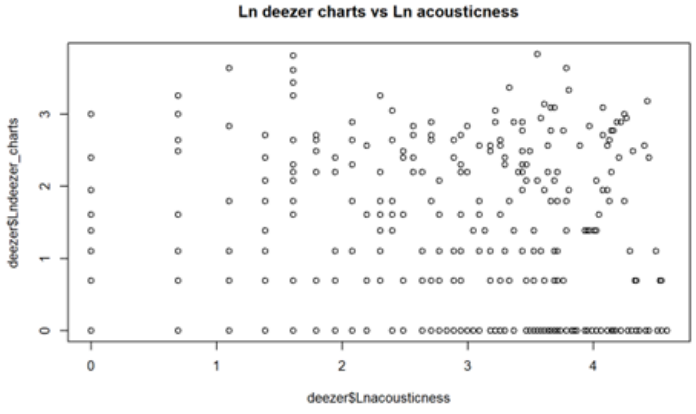

Ln shazam charts vs energy
shazam$Lnshazam_charts
shazam$energy

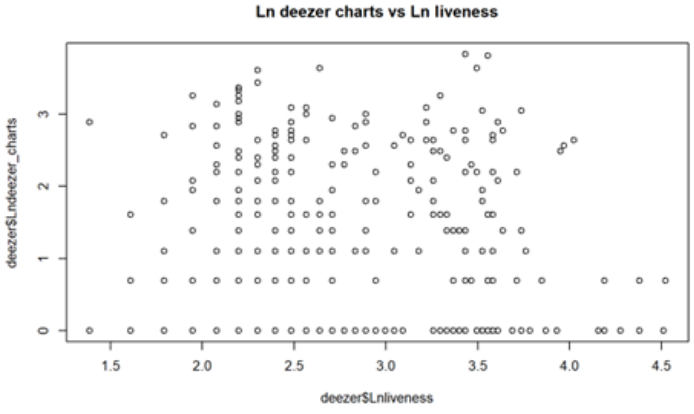

Ln deezer charts vs Ln speechiness
deezer$Lndeezer_charts
deezer$Lnspeechiness

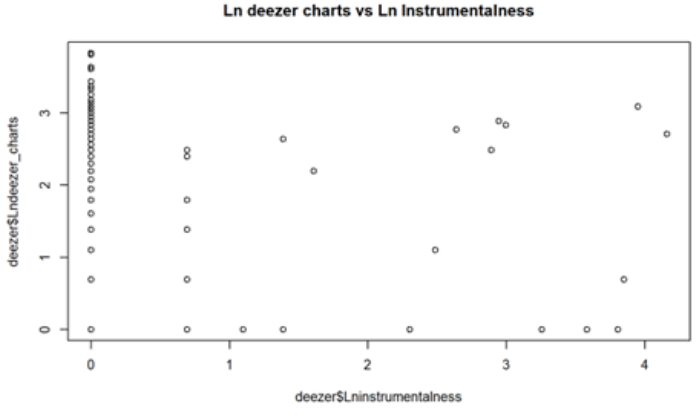

Ln deezer charts vs bpm
deezer$Lndeezer_charts
deezer$bpm

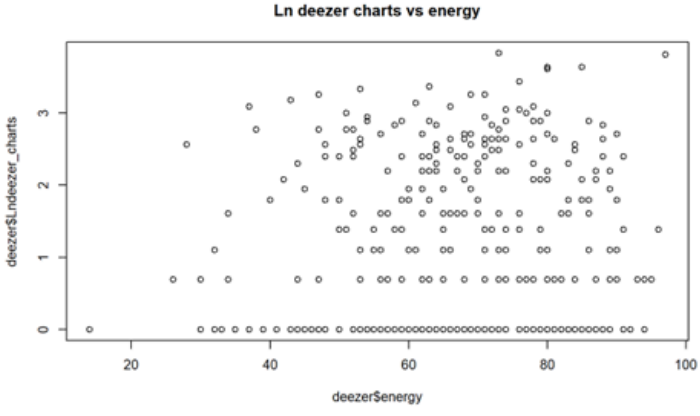

Ln deezer charts vs danceability
deezer$Lndeezer_charts
deezer$danceability


Ln deezer charts vs valence
deezer$Lndeezer_charts
deezer$valence

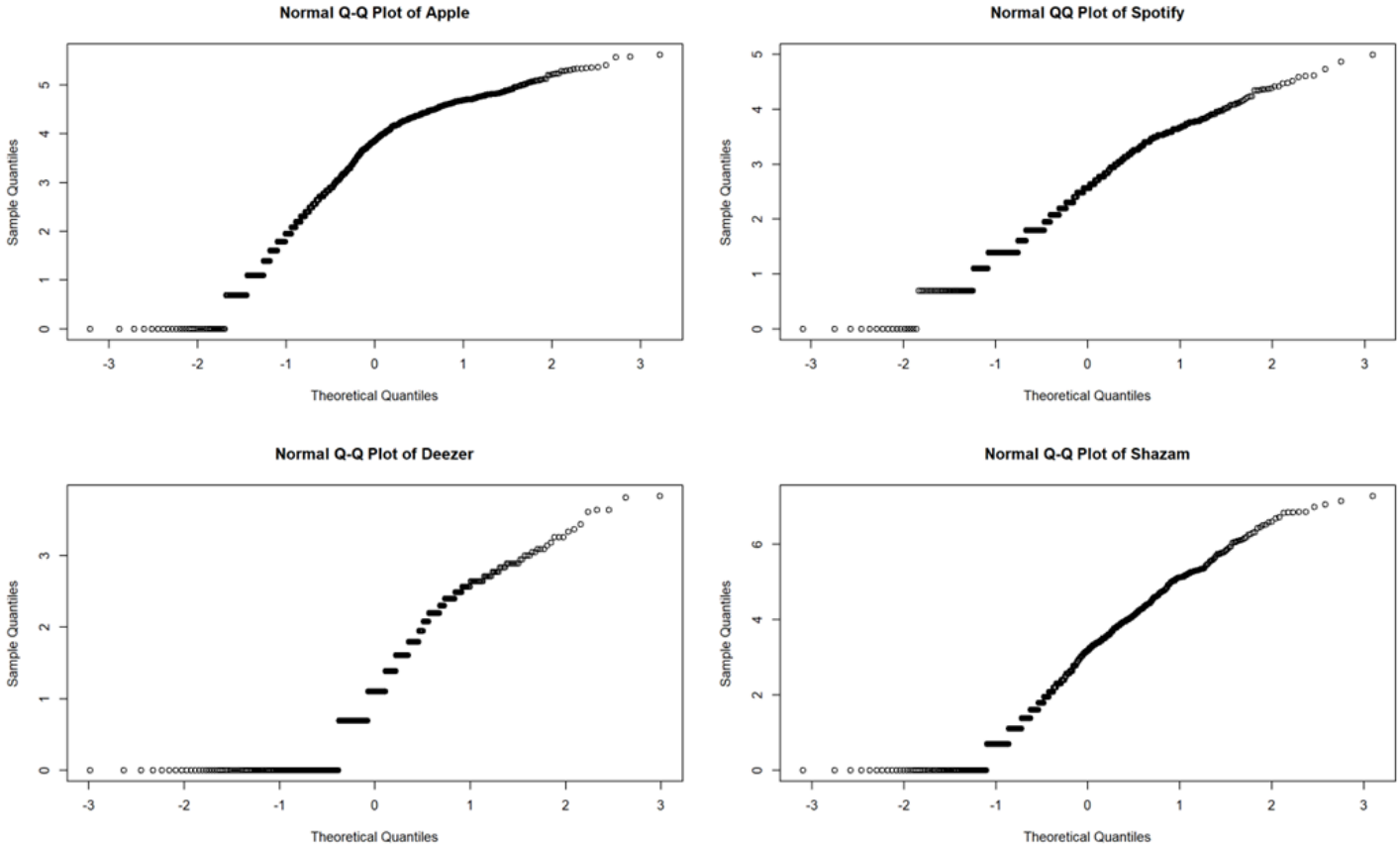

Ln deezer charts vs Ln acousticness
deezer$Lndeezer_charts
deezer$Lnacousticness


Ln deezer charts vs Ln liveness
deezer$Lndeezer_charts
deezer$Lnliveness


Ln deezer charts vs Ln Instrumentalness
deezer$Lndeezer_charts
deezer$Lninstrumentalness


Ln deezer charts vs energy
deezer$Lndeezer_charts
deezer$energy

The normality assumption was run on each dependent variable as follows.



For the best outcome linear regressions are run on dependent variables that are normal, as can be argued for each case.

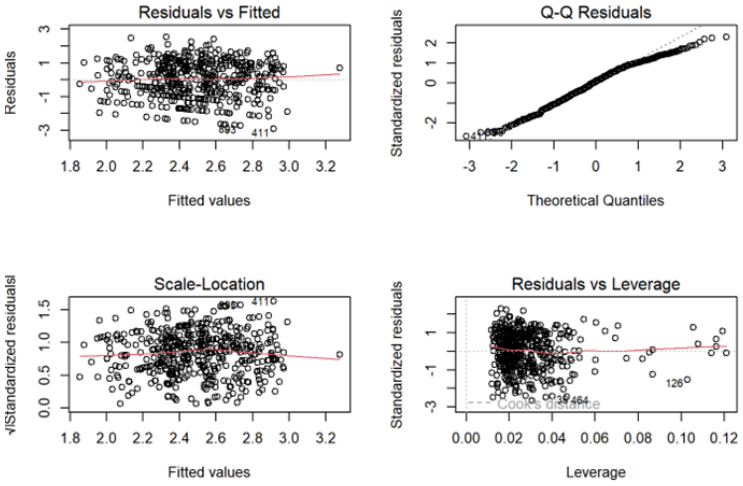
Simple linear regression

For Spotify, key was the only predictor with statistically significant at 95% or higher.

```
Call:
lm(formula = spotify$Lnspotify_charts ~ spotify$key)

Residuals:
     Min       1Q   Median       3Q      Max
-2.76352 -0.78678  0.09107  0.90005  2.54792

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.13607    0.17029  12.544  < 2e-16 ***
spotify$keyA#  0.25776    0.25226   1.022  0.30738
spotify$keyB   0.32326    0.23017   1.404  0.16083
spotify$keyC#  0.62745    0.21696   2.892  0.00400 **
spotify$keyD   0.67325    0.24225   2.779  0.00566 **
spotify$keyD# -0.03774    0.33485  -0.113  0.91030
spotify$keyE   0.31543    0.25421   1.241  0.21528
spotify$keyF   0.45284    0.22390   2.022  0.04368 *
spotify$keyF#  0.18355    0.23447   0.783  0.43411
spotify$keyG   0.42557    0.23119   1.841  0.06626 .
spotify$keyG#  0.26015    0.24082   1.080  0.28057
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.117 on 482 degrees of freedom
Multiple R-squared:  0.03247,   Adjusted R-squared:  0.01239
F-statistic: 1.617 on 10 and 482 DF,  p-value: 0.0985
```

We then wanted to see the effects of other predictors and after running several models, this model explained a higher fraction of variation compared to other models (highest R squared). The interaction term is saying that the danceability of a song is dependent on the key in which it is played and even though energy is not statistically significant it has a major impact on the rank of a song on Spotify's charts.

```
Call:
lm(formula = spotify$Lnspotify_charts ~ spotify$key:spotify$danceability +
    spotify$energy + spotify$Lninstrumentalness)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9153 -0.7788  0.1056  0.8969  2.5396

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                           1.9551555  0.3165971   6.176 1.41e-09 ***
spotify$energy                        0.0049223  0.0033323   1.477   0.1403
spotify$Lninstrumentalness            0.0976444  0.0824118   1.185   0.2367
spotify$key:spotify$danceability     -0.0032055  0.0044060  -0.728   0.4673
spotify$keyA#:spotify$danceability    0.0013054  0.0044693   0.292   0.7704
spotify$keyB:spotify$danceability     0.0028062  0.0041376   0.678   0.4980
spotify$keyC#:spotify$danceability    0.0073053  0.0041240   1.771   0.0771 .
spotify$keyD:spotify$danceability     0.0074527  0.0044154   1.688   0.0921 .
spotify$keyD#:spotify$danceability   -0.0020248  0.0059303  -0.341   0.7329
spotify$keyE:spotify$danceability     0.0025526  0.0046991   0.543   0.5872
spotify$keyF:spotify$danceability     0.0048688  0.0041765   1.166   0.2443
spotify$keyF#:spotify$danceability    0.0002595  0.0042836   0.061   0.9517
spotify$keyG:spotify$danceability     0.0032358  0.0041302   0.783   0.4337
spotify$keyG#:spotify$danceability    0.0013205  0.0044333   0.298   0.7659
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.113 on 479 degrees of freedom
Multiple R-squared:  0.04547,   Adjusted R-squared:  0.01957
F-statistic: 1.755 on 13 and 479 DF,  p-value: 0.04746
```



Model Selection and Further Statistical Analysis

The following models confirm our previous assumptions made during the data exploration.

Spotify Full Model

```
Residual standard error: 21.37 on 482 degrees of freedom
Multiple R-squared:  0.02106,   Adjusted R-squared:  0.0007519
F-statistic: 1.037 on 10 and 482 DF,  p-value: 0.4108
```

Apple Full Model and Significant Variables

```
Residual standard error: 49.1 on 753 degrees of freedom
Multiple R-squared:  0.04608,   Adjusted R-squared:  0.03341
F-statistic: 3.638 on 10 and 753 DF,  p-value: 9.431e-05

apple$speechiness      -0.75871    0.18768  -4.043 5.83e-05 ***
```

Deezer Full Model and Significant Variables

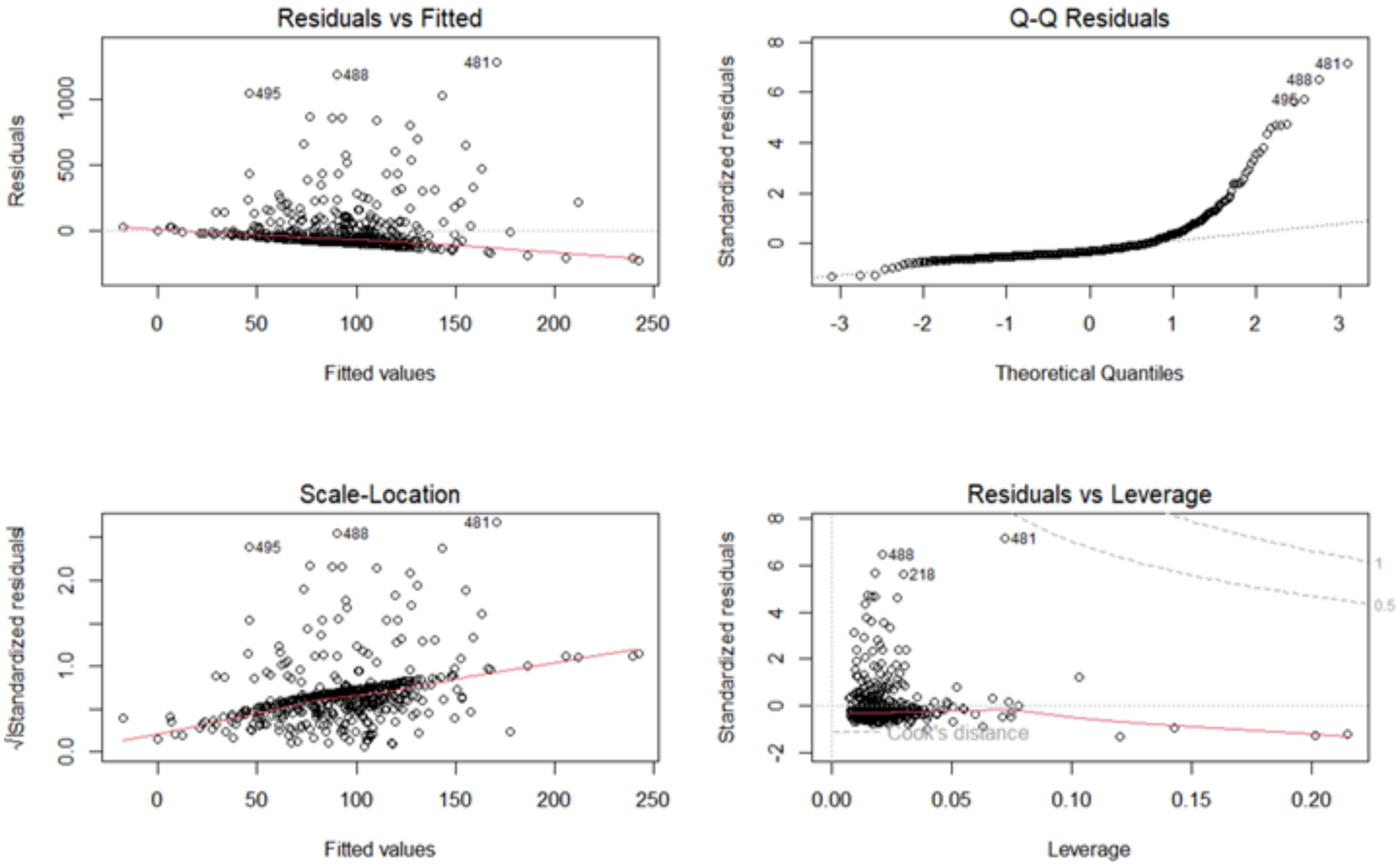
```
Residual standard error: 7.518 on 342 degrees of freedom
Multiple R-squared:  0.03875,   Adjusted R-squared:  0.01065
F-statistic: 1.379 on 10 and 342 DF,  p-value: 0.1884

deezer$speechiness      -0.119552   0.046067  -2.595  0.00986 **
deezer$energy            0.066191   0.033739   1.962  0.05059 .
deezer$danceability      0.071375   0.036982   1.930  0.05443 .
```

Shazam Full Model and Significant Variables

```
Residual standard error: 185.4 on 497 degrees of freedom
Multiple R-squared:  0.02865,   Adjusted R-squared:  0.009105
F-statistic: 1.466 on 10 and 497 DF,  p-value: 0.1488

shazam$energy             1.5378     0.6584   2.336   0.0199 *
```

Violation of Linear Model Assumptions

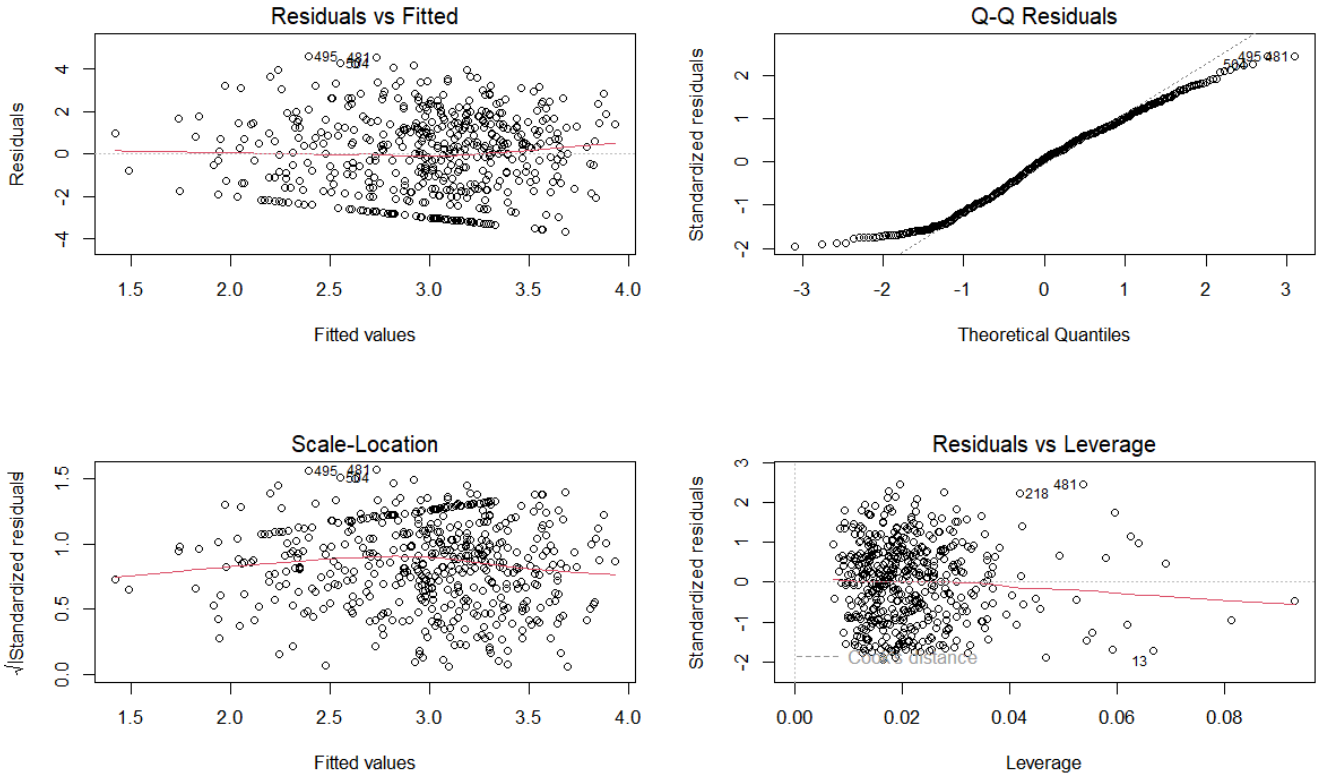
Shazam Full Model Plots



For example, in this Shazam model multiple residual assumptions are violated.

The Q-Q plot shows a textbook violation of the normality assumption while the funneling of the residuals in the scale-location plot violates homoskedasticity.

Shazam Log-Transformed Full Model



Before:

Shazam Full Model and Significant Variables

```
Residual standard error: 185.4 on 497 degrees of freedom
Multiple R-squared:  0.02865,   Adjusted R-squared:  0.009105
F-statistic: 1.466 on 10 and 497 DF,  p-value: 0.1488

shazam$energy              1.5378     0.6584   2.336   0.0199 *
```

After:

Shazam Log-Transformed Full Model and Significant Variables

```
Residual standard error: 1.904 on 497 degrees of freedom
Multiple R-squared:  0.05278,   Adjusted R-squared:  0.03373
F-statistic:  2.77 on 10 and 497 DF,  p-value: 0.002451
shazam$lnspeechiness      -0.4776806  0.1190673  -4.012 6.95e-05 ***
shazam$energy              0.0125689  0.0063460   1.981 0.048188 *
```

As you can see, after log-transforming the skewed variables, normality and homoskedasticity are now satisfied. Even linearity has improved.

This reflects in our summary statistics where the model has improved in all areas and is now statistically significant and thus useful.

This is the summary statistics for the Spotify Model after log-transforming all of the skewed variables.

```
Residual standard error: 1.128 on 482 degrees of freedom
Multiple R-squared:  0.01229,   Adjusted R-squared:  -0.008207
F-statistic: 0.5995 on 10 and 482 DF,  p-value: 0.8146
```

Applying the same approach to all of our models did not produce the same result.

Apple Model Log-Transforming the response variable and instrumentalness

```
Residual standard error: 1.364 on 753 degrees of freedom
Multiple R-squared:  0.04611,   Adjusted R-squared:  0.03345
F-statistic:  3.64 on 10 and 753 DF,  p-value: 9.335e-05
```

Spotify Model Square Root Transforming the response variable and instrumentalness

```
Residual standard error: 2.123 on 482 degrees of freedom
Multiple R-squared:  0.02604,   Adjusted R-squared:  0.005836
F-statistic: 1.289 on 10 and 482 DF,  p-value: 0.2337
```

Deezer Model Log-Transforming the response variable and acousticness

```
Residual standard error: 1.11 on 342 degrees of freedom
Multiple R-squared:  0.05897,   Adjusted R-squared:  0.03146
F-statistic: 2.143 on 10 and 342 DF,  p-value: 0.0209
```

Shazam Model Log-Transforming the response, acousticness, instrumentalness, and speechiness

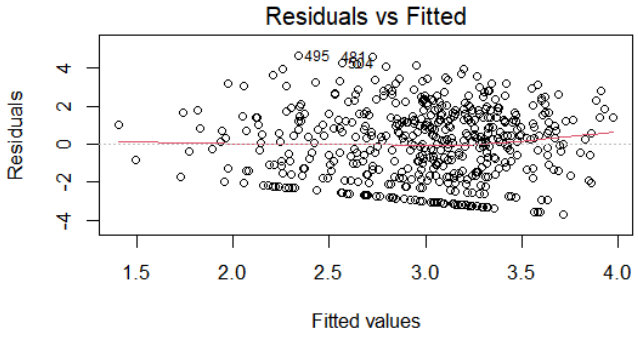
```
Residual standard error: 1.902 on 497 degrees of freedom
Multiple R-squared:  0.0543,    Adjusted R-squared:  0.03527
F-statistic: 2.854 on 10 and 497 DF,  p-value: 0.001826
```

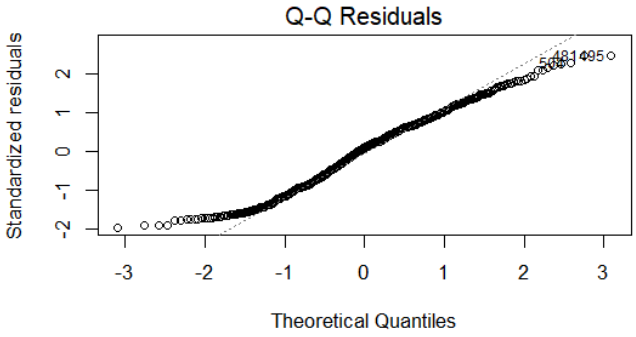
The above transformed models have each improved over the original models.

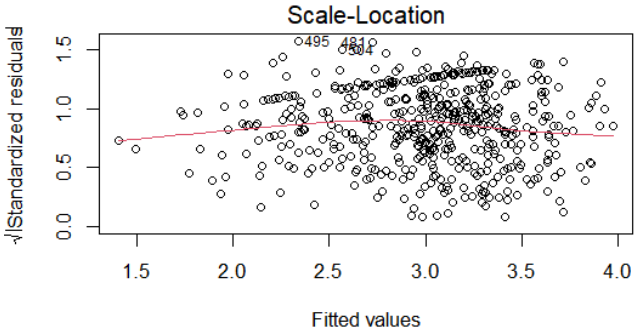
Residual Plots

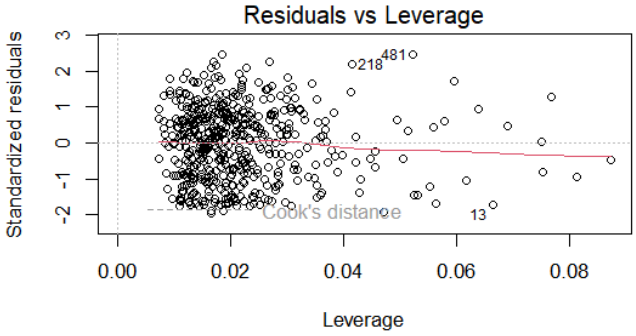
Here are the residual plots for our four transformed models. We see that both the residual assumptions and the summary statistics for each model have improved.

Shazam residual plots with one skewed variable

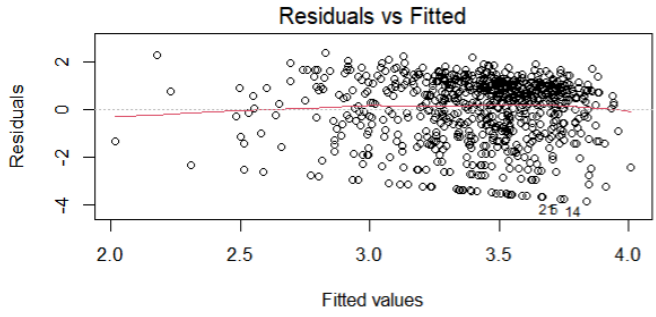


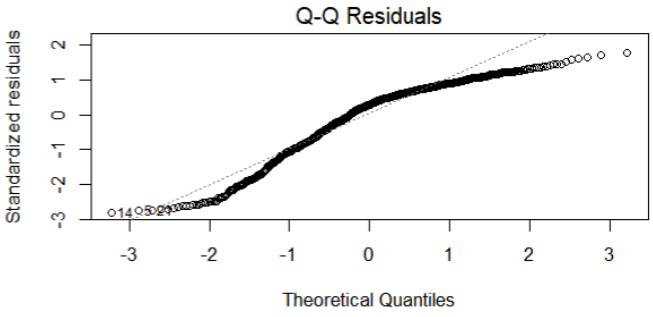


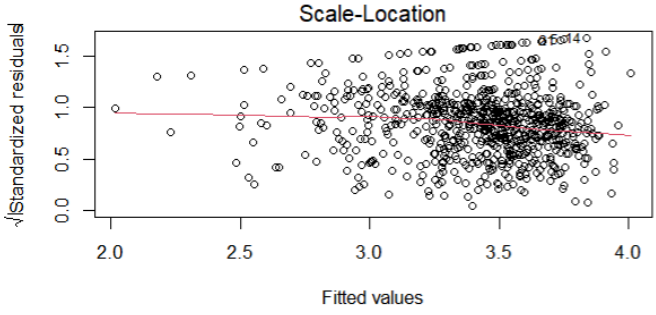


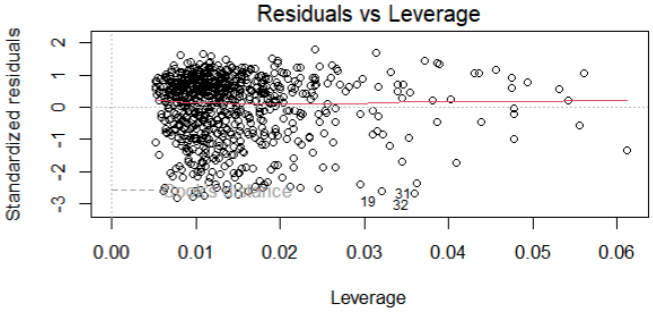


Apple residual plots with three skewed variables

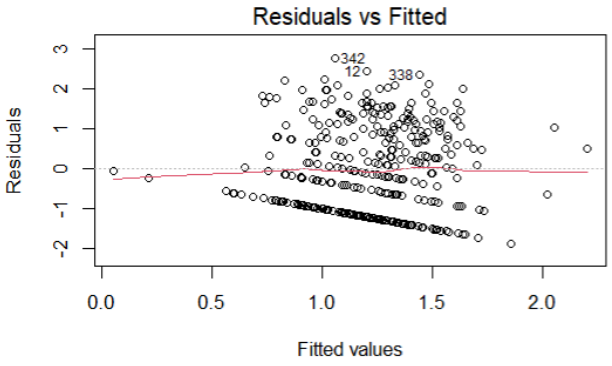


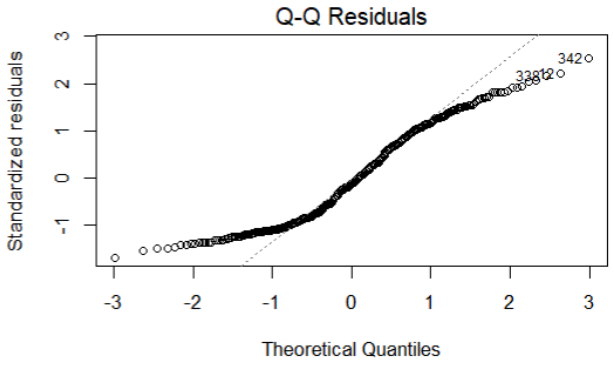


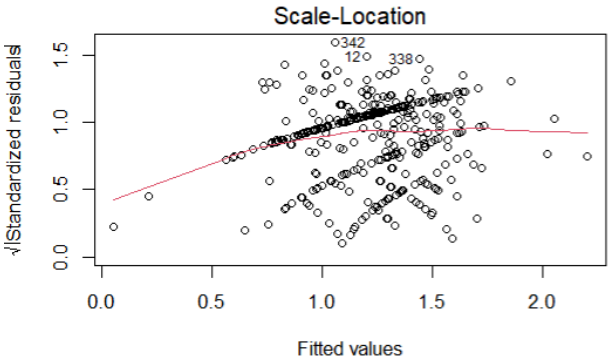


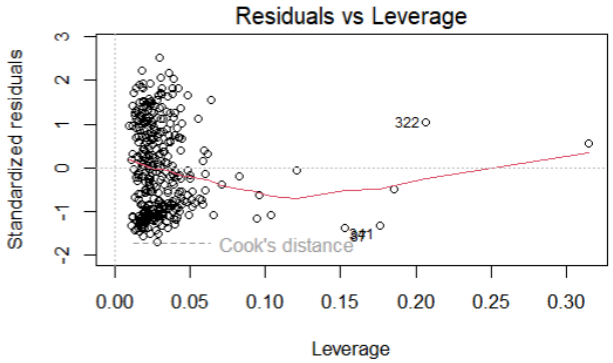


Deezer residual plots with three skewed variables

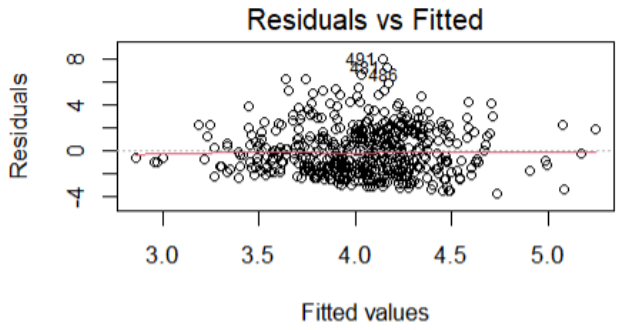


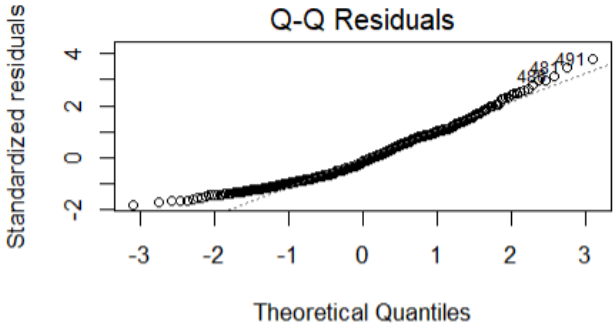


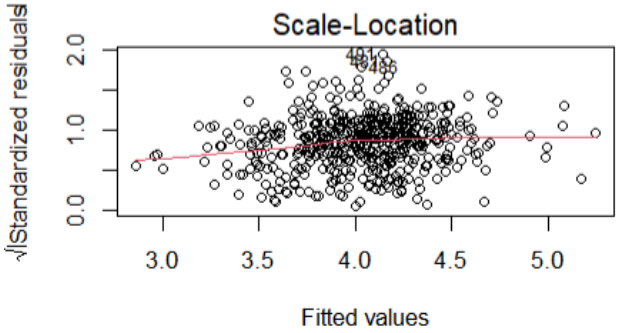


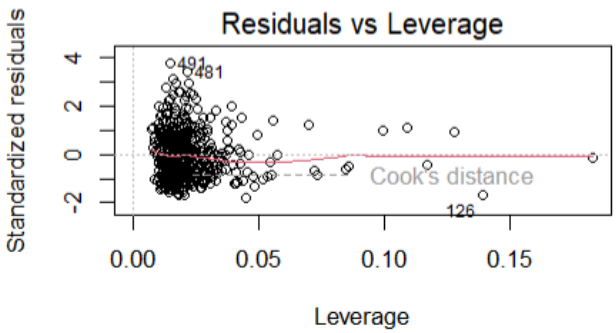


Spotify residual plots with three skewed variables







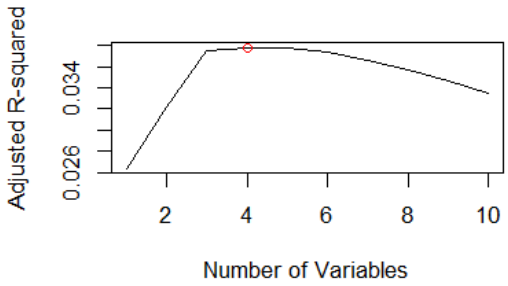


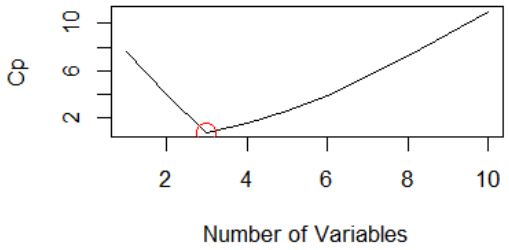
Best Subset Selection

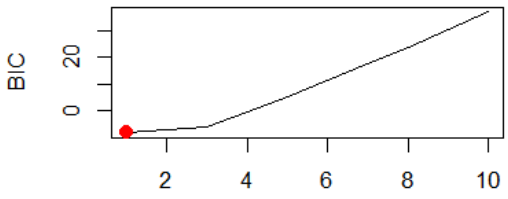
This is the order Best Subset Selection selected our predictors for our log-transformed Apple model.

	bpm	key	mode	danceability	valence	energy	acousticness	liveness	speechiness	lninstrumentalness
1 (1)	" "	" "	" "	" "	" "	" "	" "	" "	"*"	" "
2 (1)	" "	" "	" "	" "	" "	" "	"*"	" "	"*"	" "
3 (1)	"*"	" "	" "	" "	" "	" "	"*"	" "	"*"	" "
4 (1)	"*"	" "	" "	" "	" "	" "	"*"	" "	"*"	"*"
5 (1)	"*"	" "	" "	" "	" "	" "	"*"	"*"	"*"	"*"
6 (1)	"*"	" "	" "	" "	" "	"*"	"*"	"*"	"*"	"*"
7 (1)	"*"	" "	" "	" "	" "	"*"	"*"	"*"	"*"	"*"
8 (1)	"*"	" "	" "	"*"	"*"	"*"	"*"	"*"	"*"	"*"
9 (1)	"*"	" "	" "	"*"	"*"	"*"	"*"	"*"	"*"	"*"
10 (1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

These graphs tell us that according to the Best Subset Selection, we should use the four predictor model the first four variables selected to maximize adjusted R-squared, the the three predictor model first 3 variables to minimize cp and the one predictor model only the first variable to minimize BIC.







Cross Validation

Here we used a validation set approach to cross validate each of the three models selected by Best Subset Selection and chose the one with the lowest test MSE.

Apple Model chosen by Best Subset Selection using Cross Validation - Test MSE = 1.84466

```
Residual standard error: 1.361 on 760 degrees of freedom
Multiple R-squared:  0.04134,   Adjusted R-squared:  0.03756
F-statistic: 10.93 on 3 and 760 DF,  p-value: 4.964e-07
```

Shazam Model chosen by Best Subset Selection using Cross Validation - Test MSE = 3.374389

```
Residual standard error: 1.897 on 503 degrees of freedom
Multiple R-squared:  0.0482,    Adjusted R-squared:  0.04063
F-statistic: 6.368 on 4 and 503 DF,  p-value: 5.274e-05
```

Spotify Model chosen by Best Subset Selection using Cross Validation - Test MSE = 3.68372

```
Residual standard error: 2.117 on 489 degrees of freedom
Multiple R-squared:  0.01757,   Adjusted R-squared:  0.01154
F-statistic: 2.915 on 3 and 489 DF,  p-value: 0.0339
```

Deezer Model chosen by Best Subset Selection using Cross Validation - Test MSE = 1.147455

```
Residual standard error: 1.106 on 347 degrees of freedom
Multiple R-squared:  0.05227,   Adjusted R-squared:  0.03861
F-statistic: 3.828 on 5 and 347 DF,  p-value: 0.002177
```

As you can see, these models have improved a significant amount from their full model and according to their F statistics, they are now useful in explaining the variability in the data.

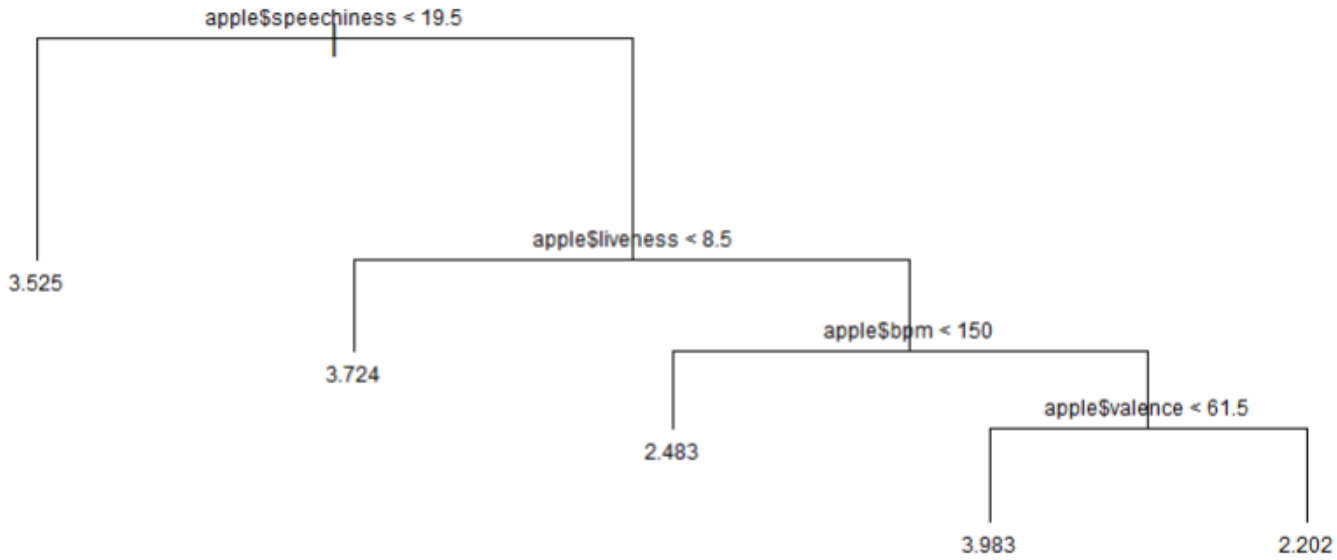
Decision Trees

Test MSE

	Apple	Shazam	Spotify	Deezer
Regression-Trees	2.383296	4.588403	6.658721	1.850556
Bagging	2.039657	3.707626	4.387446	1.443602
Random Forest	2.010689	3.641602	4.19717	1.430631
Boosting	2.547079	4.184465	5.601217	1.717676
Best Subset	1.84466	3.374389	3.68372	1.147455

Although best subset still has the lowest test MSE, we can still use these methods to gain a better understanding of our data.

Apple Charts Regression Tree



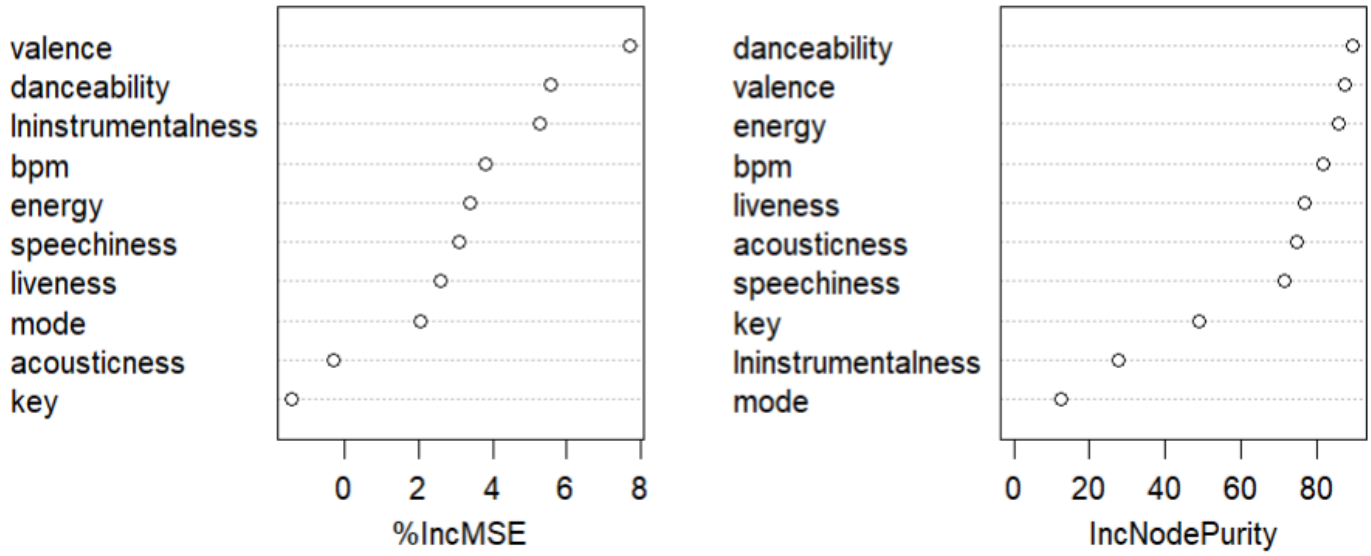
Speechiness is the most important factor in determining the ranking for Apple Charts. Given that a song has less than 19.5% speechiness, the percent of live elements in the song play little role in its ranking.

Among those songs with more than 19.5% spoken words, if less than 8.5% of the song has live elements, bpm is not important in its ranking.

Among those songs with more than 19.5% speechiness and 8.5% liveness, if the song has less than 150 bpm than the valence in the song is not important in how it gets ranked in Apple charts.

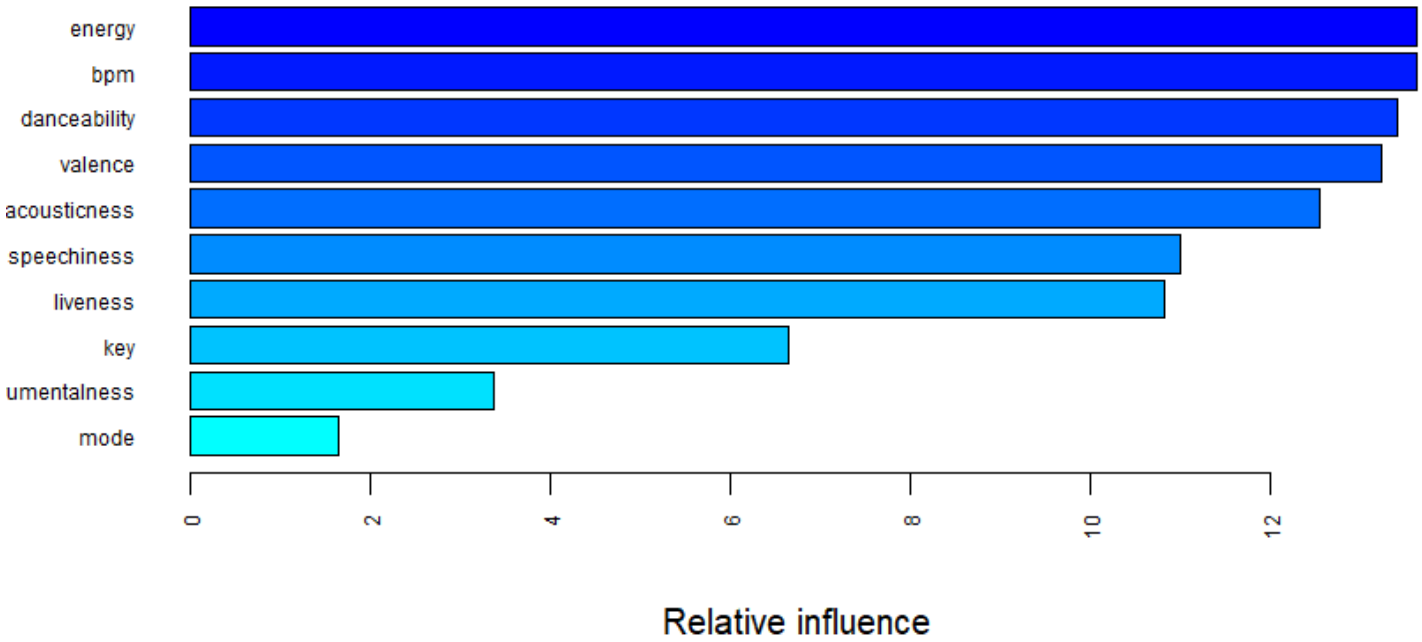
However, if the song has less than 19.5% speechiness, less than 8.5% liveness, and less than 150 bpm, then the percentage of valence in the song plays a significant role in its ranking.

The Percentage Increase in Mean Squared Error and Increase in Node Purity Graph for Apple Charts using Random Forest



According to these graphs, the amount of positivity in the song and how suitable the song is for dancing are significant factors in determining ranking while mode and key are not.

Relative Influence of the predictors for Apple Charts using Boosting



This shows us that in addition to danceability and valence, energy and bpm are important predictors as well and affirms our findings that key and mode are not.

Ridge and Lasso

Ridge and Lasso are two shrinkage methods that we can use to gain further insight into our data.

Test MSE

	Apple	Shazam	Spotify	Deezer
Ridge	1.952162	3.621752	3.771967	1.237161
Lasso	1.982805	3.625017	3.78869	1.239982
Regression-Trees	2.383296	4.588403	6.658721	1.850556
Bagging	2.039657	3.707626	4.387446	1.443602
Random Forest	2.010689	3.641602	4.19717	1.430631
Boosting	2.547079	4.184465	5.601217	1.717676
Best Subset	1.84466	3.374389	3.68372	1.147455

Although Best Subset still has the lowest test MSE, Ridge is a close second. This indicates that the ranking for Apple Charts is truly a function of many predictors.

Apple Charts Ridge Coefficient Estimates

(Intercept)	bpm	key	mode
3.418498e+00	3.715360e-39	-6.143447e-39	6.346143e-39
danceability	valence	energy	acousticness
-3.984137e-39	1.727838e-39	6.602097e-39	-4.216157e-39
liveness	speechiness	lninstrumentalness	
-2.843925e-39	-2.375212e-38	-3.447276e-38	

This shows that key, danceability, acousticness, liveness, speechiness and lninstrumentalness are positively correlated with ranking while bpm, mode, valence and energy are negatively correlated with ranking.

Apple Charts Lasso Coefficient Estimates

(Intercept)	bpm	key	mode
3.430688067	0.000000000	0.000000000	0.000000000
danceability	valence	energy	acousticness
0.000000000	0.000000000	0.000000000	0.000000000
liveness	speechiness	lninstrumentalness	
0.000000000	-0.001196342	0.000000000	

A 1% increase in speechiness results in a 0.1197058% increase in Apple Charts ranking.

Shazam Charts Lasso Coefficient Estimates

(Intercept)	bpm	key	mode
3.02560258	0.00000000	0.00000000	0.00000000
danceability	valence	energy	liveness
0.00000000	0.00000000	0.00000000	0.00000000
lnacousticness	lninstrumentalness	lnspeechiness	
0.00000000	0.00000000	-0.01719973	

A 1% increase in speechiness results in a 0.017348497% increase in Shazam Charts ranking.

Spotify Charts Lasso Coefficient Estimates

(Intercept)	bpm	key	mode
4.0467792753	0.0000000000	0.0000000000	0.0000000000
danceability	valence	energy	acousticness
0.0000000000	0.0000000000	0.0000000000	-0.0005400072
liveness	speechiness	srinstrumentalness	
0.0000000000	0.0000000000	0.0000000000	

0.0005400072^2 = 0.000000292
A 1% increase in acousticness results in a 0.000000292 increase in Spotify Charts ranking.

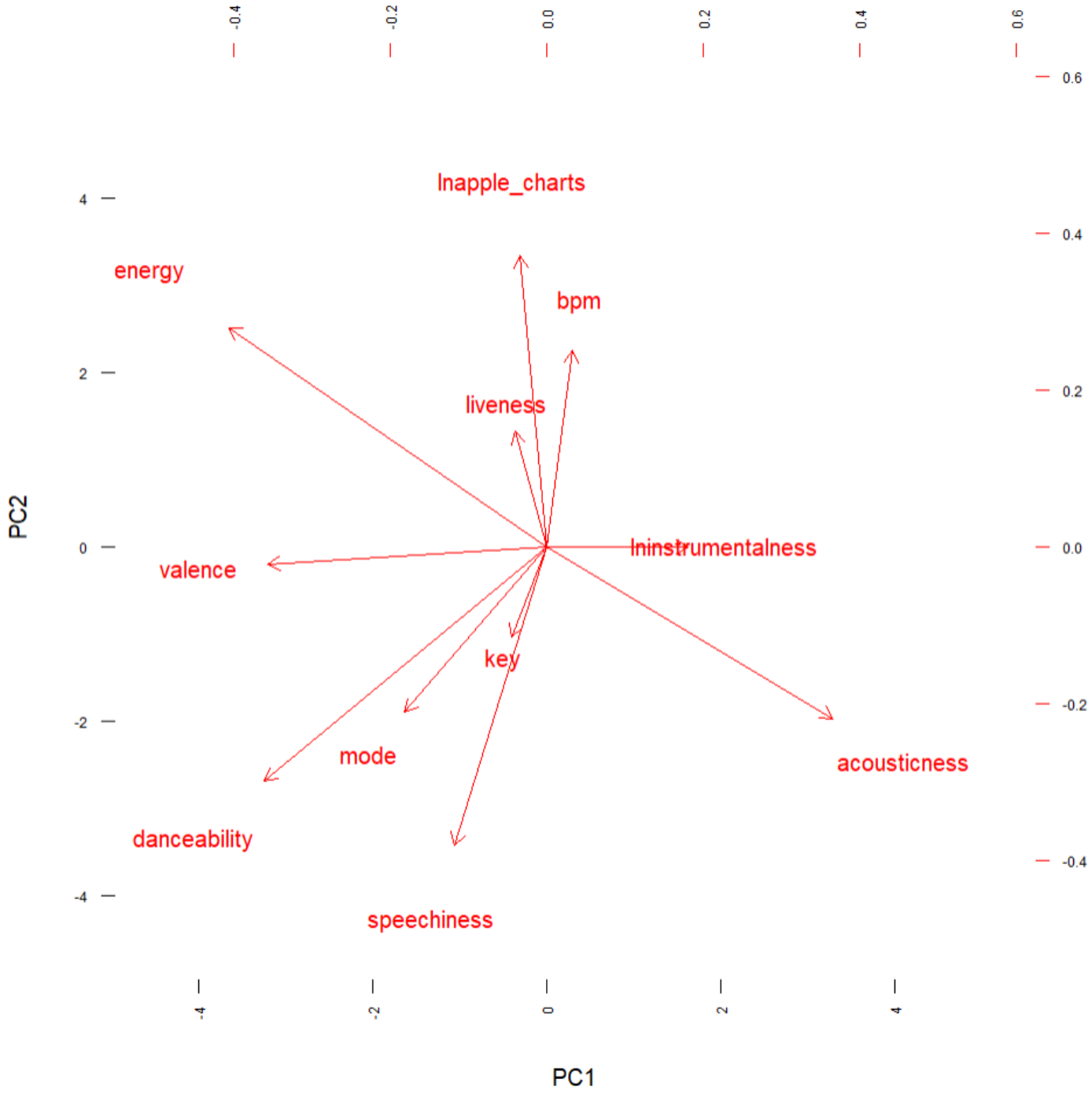
Deezer Charts Lasso Coefficient Estimates

(Intercept)	bpm	key	mode	danceability
1.2297875528	0.0000000000	0.0000000000	0.0000000000	0.0000000000
valence	energy	instrumentalness	liveness	speechiness
0.0000000000	0.0000000000	0.0000000000	0.0000000000	-0.0008756656
lnacousticness				
0.0000000000				

A 1% increase in speechiness results in a 0.0876049% increase in Deezer Charts ranking.

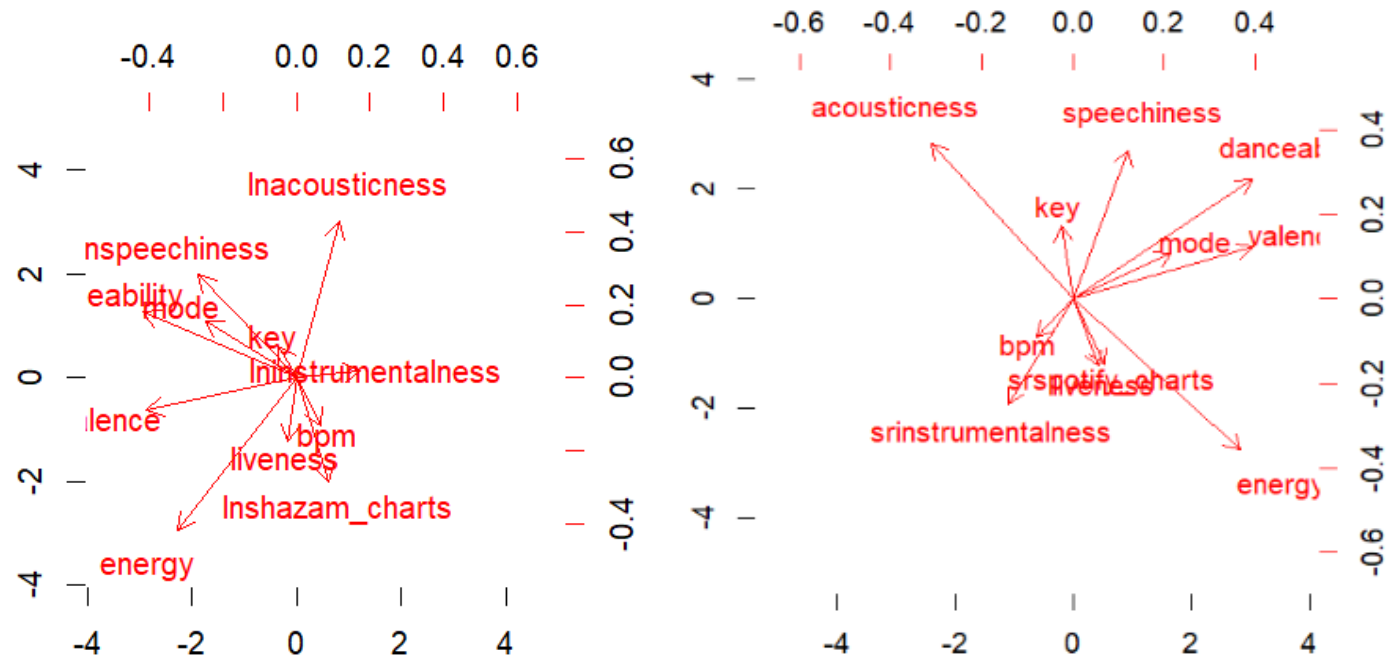
These Lasso coefficient estimates show that speechiness is the most significant predictor for the ranking in Apple Charts as well as acousticness with both having a positive correlation.

Apple Charts PCA Biplot



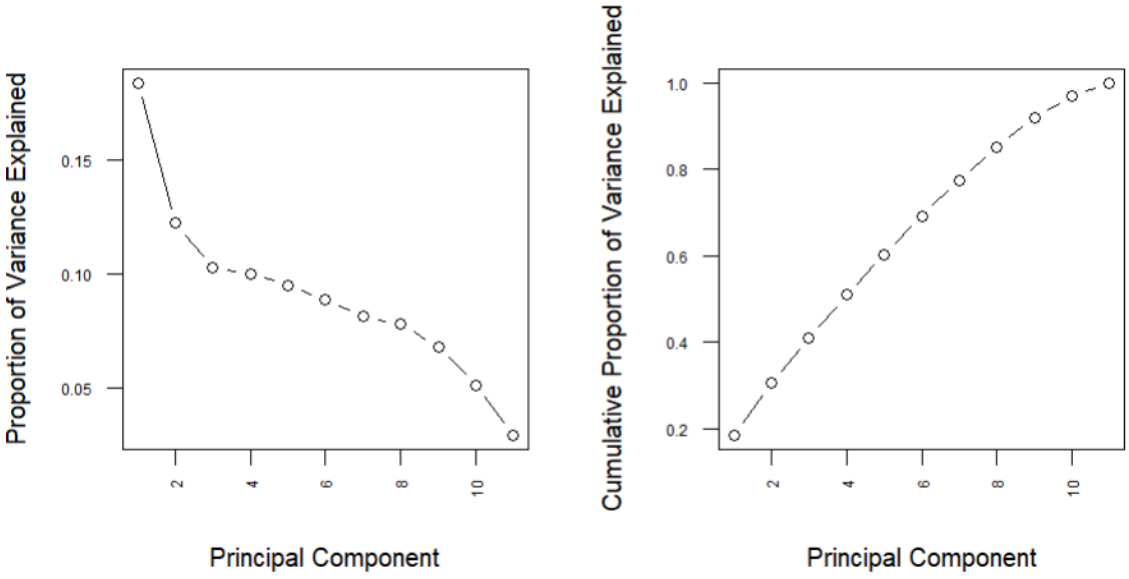
Since Bpm energy and liveness are pointing towards PC1 (loading vectors) we can say they are correlated with each other and PC1 captures a lot of their data. Valence, danceability, mode, and speechiness which are correlated with each other are orthogonal to PC1's loading vectors so a lot of their data can be captured in PC2 and acousticness and instrumentalness are orthogonal to PC1 and PC2 so a lot of their data can be captured by PC3.

Shazam Charts and Spotify Charts PCA Biplot



These charts further show how these variables relate to another. For example, we see the contradictory relationship between acousticness and energy. As the percentage of acousticness in a song goes up, the percentage of energy in a song goes down and vice versa. The size of the arrows also indicate the significance of the variable in the data with acousticness and energy being the most significant and key and bpm being among the least significant for these specific rankings.

Apple Charts PVE and cumulative PVE



These charts show that even when the data is reduced to a low-dimensional representation of the data, like a sausage, one principal component still cannot accurately explain a good fraction of the variance.

This explains why even our most significant models explain a small fraction of the variance.

Final Models

Apple Charts

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.295696   0.231959  14.208  < 2e-16 ***
apple$bpm          0.004040   0.001765   2.289   0.0224 *
apple$acousticness -0.004686   0.001963  -2.387   0.0172 *
apple$speechiness  -0.024602   0.005036  -4.885 1.26e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.361 on 760 degrees of freedom
Multiple R-squared:  0.04134,   Adjusted R-squared:  0.03756
F-statistic: 10.93 on 3 and 760 DF,  p-value: 4.964e-07
```

Holding fixed all other variables, on average,

a 1% increase in speechiness results in a 2.4907126% increase in ranking,

a 1% increase in acousticness results in a 0.4696996% increase in ranking, and

a 1 beat increase in beats per minute results in a 0.4048172% decrease in ranking.

Overall, this model is statistically significant in explaining 3.756% of the variability in the data.

Shazam Charts

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.080644   0.580134   5.310 1.65e-07 ***
shazam$bpm          0.004060   0.003036   1.337   0.1818
shazam$mode        -0.236785   0.172688  -1.371   0.1709
shazam$energy       0.010329   0.005314   1.944   0.0525 .
shazam$lnspeechiness -0.467300   0.112303  -4.161 3.73e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.897 on 503 degrees of freedom
Multiple R-squared:  0.0482,    Adjusted R-squared:  0.04063
F-statistic: 6.368 on 4 and 503 DF,  p-value: 5.274e-05
```

Holding fixed all other variables, on average, a 1% increase in speechiness results in a 0.4673% increase in ranking, and a 1% increase in energy results in a 1.0382528% decrease in ranking.

Overall, this model is statistically significant in explaining 4.063% of the variability in the data.

Spotify Charts

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                3.93882    0.21642  18.200   <2e-16 ***
spotify$acousticness      -0.00902    0.00398  -2.266   0.0239 *
spotify$key                0.04878    0.03029   1.611   0.1079
spotify$srinstrumentalness 0.14763    0.09332   1.582   0.1143
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.117 on 489 degrees of freedom
Multiple R-squared:  0.01757,   Adjusted R-squared:  0.01154
F-statistic: 2.915 on 3 and 489 DF,  p-value: 0.0339
```

Holding fixed all other variables, on average, a 1% increase in acousticness results in a 0.00008136 increase in ranking. Overall, this model is statistically significant in explaining 1.154% of the variability in the data.

Deezer Charts

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.037971   0.420067  -0.090 0.928028
deezer$speechiness   -0.022231   0.006645  -3.345 0.000911 ***
deezer$danceability   0.013632   0.005084   2.682 0.007679 **
deezer$energy         0.009309   0.004539   2.051 0.041029 *
deezer$lnacousticness 0.055288   0.039343   1.405 0.160828
deezer$valence       -0.004072   0.002968  -1.372 0.170893
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.106 on 347 degrees of freedom
Multiple R-squared:  0.05227,   Adjusted R-squared:  0.03861
F-statistic: 3.828 on 5 and 347 DF,  p-value: 0.002177
```

Holding fixed all other variables, on average, a 1% increase in speechiness results in a 2.247995% increase in ranking, a 1% increase in danceability results in a 1.3725339% decrease in ranking, and a 1% increase in energy results in a 0.9352464% decrease in ranking.

Overall, this model is statistically significant in explaining 3.861% of the variability in the data.

Conclusion

In conclusion, the question we were interested in investigating was what makes a song go viral. The question remains largely a mystery as we were only able to explain the tip of the iceberg, about 4% to be exact. We uncovered relationships between predictors, such as the opposing natures of energy and acousticness and how they relate to predicting a song's ranking. We were able to distinguish the key variables in predicting the response as well as those that were insignificant, such as mode. We saw how each individual method had their own unique view of the data from the Boosting method identifying energy as the most influential predictor to Lasso selecting speechiness as a significant factor. Through PCA, we saw in depth how these variables related to each other and through the PVE and cumulative PVE plots, we saw how not even a principal component could explain a good fraction of the variance showing just how unpredictable trends in music can be. We were able to insinuate that this was a high dimensional dataset from the model regularization models thus revealing that there were many variables that we were not privy to and important variables that we removed which all contributed to creating models with low confidence. Although we conclude this analysis with our question largely unanswered, we can say with confidence that we know the answer to at least 1.154% of it.

Appendix

```
#Set Working Directory
setwd()

#Loading Data
music=read.csv("spotify-2023.csv",header = TRUE, stringsAsFactors=T )
View(music)
str(music)

################
#TIDY THE DATA #
################

#delete the following columns: track name, artist(s) name, artist count, released year, released month, released day
music <- music[, -1:-6]
# delete the following columns: in_spotify_playlists, streams, in_apple_playlists, in_deezer_playlists
music <- music[,c(-1,-3,-4,-6)]
View(music)
str(music)

#Remove the blanks in keys
music <- music[music$key != "", ]
#Manually remove the level created for blanks because R did not
music$key <- droplevels(music$key)

#Remove the commas in shazam charts so R can change it from qualitative to quantitative
music$shazam_charts <- as.integer(gsub(",", "", music$shazam_charts))
View(music)
str(music)

##########################################
#SEPARATE OUR OUTCOMES AND FURTHER TIDY UP #
##########################################

spotify <- music[,c(-2,-3,-4)]
apple <- music[,c(-1,-3,-4)]
deezer <- music[,c(-1,-2,-4)]
shazam <- music[,c(-1,-2,-3)]

#Remove the zeros from the outcome variables
spotify <- spotify[spotify$spotify_charts != 0, ]
apple <- apple[apple$apple_charts != 0, ]
deezer <- deezer[deezer$deezer charts != 0. ]
shazam <- shazam[shazam$shazam_charts != 0, ]

#Remove the blanks in shazam charts, R automatically replaced the zeros with NA's
shazam <- na.omit(shazam)

###################
#Data Exploration#
###################

#Look at the distribution of all the variables
hist(spotify$spotify_charts)
hist(apple$apple_charts)
hist(deezer$deezer_charts)
hist(shazam$shazam_charts)
hist(music$bpm)
hist(music$danceability)
hist(music$valence)
hist(music$energy)
hist(music$acousticness)
hist(music$instrumentalness)
hist(music$liveness)
hist(music$speechiness)

par(mfrow=c(2,2))
hist(music$bpm, col="darkgray", main=paste("Histogram of Beats Per Minute"))
hist(music$danceability, col="darkgreen", main=paste("Histogram of Danceability"))
hist(music$valence, col="darkblue", main=paste("Histogram of Valence"))
hist(music$energy, col="darkred", main=paste("Histogram of Energy"))
```

```r
#log transform highly right skewed variables
spotify$Lnspotify_charts <- log(spotify$spotify_charts)
apple$Lnapple_charts <- log(apple$apple_charts)
shazam$Lnshazam_charts <- log(shazam$shazam_charts)
deezer$Lndeezer_charts <- log(deezer$deezer_charts)
music$Lnliveness <- log(music$liveness)
music$Lnspeechiness <- log(music$speechiness)
spotify$Lnliveness <- log(spotify$liveness)
spotify$Lnspeechiness <- log(spotify$speechiness)
apple$Lnliveness <- log(apple$liveness)
apple$Lnspeechiness <- log(apple$speechiness)
shazam$Lnliveness <- log(shazam$liveness)
shazam$Lnspeechiness <- log(shazam$speechiness)
deezer$Lnliveness <- log(deezer$liveness)
deezer$Lnspeechiness <- log(deezer$speechiness)

#log transform highly right skewed variables with zero inputs
music$Lnacousticness <- log(music$acousticness+1)
music$Lninstrumentalness <- log(music$instrumentalness+1)
spotify$Lnacousticness <- log(spotify$acousticness+1)
spotify$Lninstrumentalness <- log(spotify$instrumentalness+1)
apple$Lnacousticness <-log(apple$acousticness+1)
apple$Lninstrumentalness <- log(apple$instrumentalness+1)
shazam$Lnacousticness <- log(shazam$acousticness+1)
shazam$Lninstrumentalness <- log(shazam$instrumentalness+1)
deezer$Lnacousticness <- log(deezer$acousticness+1)
deezer$Lninstrumentalness <- log(deezer$instrumentalness+1)
music$Lnspotify_charts <- log(music$spotify_charts+1)
music$Lnapple_charts <- log(music$apple_charts+1)
music$Lnshazam_charts <- log(music$shazam_charts+1)
music$Lndeezer_charts <- log(music$deezer_charts+1)

# variables before and after log transformation
par(mfrow=c(3,2))
hist(spotify$spotify_charts, col ="blue", main = paste("Histogram of Spotify Charts"))
hist(spotify$Lnspotify_charts, col = "blue", main = paste("Histogram of Log Transformed Spotify Charts") )
hist(apple$apple_charts, col ="green", main = paste("Histogram of Apple Charts"))
hist(apple$Lnapple_charts, col ="green", main =paste("Histogram of Log Transformed Apple Charts"))
hist(shazam$shazam_charts, col = "lavender", main = paste("Histogram of Shazam Charts"))
hist(shazam$Lnshazam_charts, col = "lavender", main = paste("Histogram of Log Transformed Shazam Charts"))

par(mfrow=c(3,2))
hist(deezer$deezer_charts, col = "pink", main = paste("Histogram of Deezer Charts"))
hist(deezer$Lndeezer_charts, col = "pink", main = paste("Histogram of Log Transformed Deezer Charts"))
hist(music$acousticness, col ="red", main = paste("Histogram of Acousticness"))
hist(music$Lnacousticness, col ="red", main = paste("Histogram of Log Transformed Acousticness"))
hist(music$instrumentalness, col = "orange", main = paste("Histogram of Instrumentalness"))
hist(music$Lninstrumentalness, col ="orange", main = paste("Histogram of Log Transformed Instrumentalness"))

par(mfrow= c(2,2))
hist(music$liveness, col="gold", main = paste("Histogram of Liveness"))
hist(music$Lnliveness, col ="gold", main = paste("Histogram of Log Transformed Liveness"))
hist(music$speechiness, col="brown", main = paste("Histogram of Speechiness"))
hist(music$Lnspeechiness, col="brown", main = paste("Histogram of Log Transformed Speechiness"))

#Check correlation of variables

pairs(music[,c(5:10,15:22)])
pairs(music[,1:4])
pairs(spotify[,-c(1,8:11)])
pairs(apple[,-c(1,8:11)])
pairs(shazam[,-c(1,8:11)])
pairs(deezer[,-c(1,8:11)])
```

```r
#Convert qualitative to quantitative so that R can reproduce the correlation coefficients
music$mode <- as.numeric(music$mode)
music$key <- as.numeric(music$key)

cor(music)
correlation_matrix <- cor(music)


#install.packages("ggplot2")
#install.packages("reshape2")

# Plot correlation matrix as a heatmap
library(ggplot2)
library(reshape2) # Needed to convert correlation matrix to long format

# Convert correlation matrix to long format
correlation_data <- melt(correlation_matrix)

# Plot heatmap
ggplot(correlation_data, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 12)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1)) +
  coord_fixed() +
  labs(x = NULL, y = NULL)

# Lets do a heatmap with only the predictors to get a closer look

music2 <- music[, c(5:10,15:18)]

correlation_matrix2 <- cor(music2)
correlation_data2 <- melt(correlation_matrix2)
ggplot(correlation_data2, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 20)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 20, hjust = 1)) +
  coord_fixed() +
  labs(x = NULL, y = NULL)

#Convert back to qualitative variables
music$mode <- as.factor(music$mode)
music$key <- as.factor(music$key)

# Take a look at the distribution of the response variable by the level of a qualitative predictor
par(mfrow=c(2,2))
plot(spotify$key, spotify$Lnspotify_charts, ylab="Log of Spotify Charts", xlab="A=1, A#=2, B=3, C#=4, D=5, D#=6, E=7, F=8, F#=9, G=10, G#=11", main="Boxplot by Key fo
plot(apple$key, apple$Lnapple_charts, ylab="Log of Apple Charts", xlab="A=1, A#=2, B=3, C#=4, D=5, D#=6, E=7, F=8, F#=9, G=10, G#=11", main="Boxplot by Key for Apple"
plot(shazam$key, shazam$Lnshazam_charts, ylab="Log of Shazam Charts", xlab="A=1, A#=2, B=3, C#=4, D=5, D#=6, E=7, F=8, F#=9, G=10, G#=11", main="Boxplot by Key for Sh
plot(deezer$key, deezer$Lndeezer_charts, ylab="Log of Deezer Charts", xlab="A=1, A#=2, B=3, C#=4, D=5, D#=6, E=7, F=8, F#=9, G=10, G#=11", main="Boxplot by Key for De

par(mfrow=c(2,2))
plot(spotify$mode, spotify$Lnspotify_charts, ylab="Log of Spotify Charts", xlab="Major=1, Minor=2", main="Boxplot by Mode for Spotify", col="red")
plot(apple$mode, apple$Lnapple_charts, ylab="Log of Apple Charts", xlab="Major=1, Minor=2", main="Boxplot by Mode for Apple", col="blue")
plot(shazam$mode, shazam$Lnshazam_charts, ylab="Log of Shazam Charts", xlab="Major=1, Minor=2", main="Boxplot by Mode for Shazam",col="green")
plot(deezer$mode, deezer$Lndeezer_charts, ylab="Log of Deezer Charts", xlab="Major=1, Minor=2", main="Boxplot by Mode for Deezer", col="yellow")

summary(spotify)
summary(apple)
summary(shazam)
summary(deezer)
```

```r
#Checking to see how the line of best fit would look for each variable
#SPOTIFY
plot(spotify$Lnspeechiness, spotify$Lnspotify_charts, main = "Ln spotify charts vs Ln speechiness")
plot(spotify$bpm, spotify$Lnspotify_charts, main = "Ln spotify charts vs bpm")
plot(spotify$danceability,spotify$Lnspotify_charts, main = "Ln spotify charts vs danceability")
plot(spotify$valence, spotify$Lnspotify_charts, main = "Ln spotify charts vs valence")
plot(spotify$Lnacousticness, spotify$Lnspotify_charts, main = "Ln spotify charts vs, Ln acousticness")
plot(spotify$Lnliveness, spotify$Lnspotify_charts, main = "Ln spotify charts vs Ln liveness")
plot(spotify$Lninstrumentalness, spotify$Lnspotify_charts, main = "Ln spotify charts vs Ln Instrumentalness")
plot(spotify$energy, spotify$Lnspotify_charts, main = "Ln spotify charts vs energy")

#APPLE
plot(apple$Lnspeechiness, apple$Lnapple_charts, main = "Ln apple charts vs Ln speechiness")
plot(apple$bpm, apple$Lnapple_charts, main = "Ln apple charts vs bpm")
plot(apple$danceability,apple$Lnapple_charts, main = "Ln apple charts vs danceability")
plot(apple$valence, apple$Lnapple_charts, main = "Ln apple charts vs valence")
plot(apple$Lnacousticness, apple$Lnapple_charts, main = "Ln apple charts vs Ln acousticness")
plot(apple$Lnliveness, apple$Lnapple_charts, main = "Ln apple charts vs Ln liveness")
plot(apple$Lninstrumentalness, apple$Lnapple_charts, main = "Ln apple charts vs Ln Instrumentalness")
plot(apple$energy, apple$Lnapple_charts, main = "Ln apple charts vs energy")

#SHAZAM
plot(shazam$Lnspeechiness, shazam$Lnshazam_charts, main = "Ln shazam charts vs Ln speechiness")
plot(shazam$bpm, shazam$Lnshazam_charts, main = "Ln shazam charts vs bpm")
plot(shazam$danceability,shazam$Lnshazam_charts, main = "Ln shazam charts vs danceability")
plot(shazam$valence, shazam$Lnshazam_charts, main = "Ln shazam charts vs valence")
plot(shazam$Lnacousticness, shazam$Lnshazam_charts, main = "Ln shazam charts vs Ln acousticness")
plot(shazam$Lnliveness, shazam$Lnshazam_charts, main = "Ln shazam charts vs Ln liveness")
plot(shazam$Lninstrumentalness, shazam$Lnshazam_charts, main = "Ln shazam charts vs Ln Instrumentalness")
plot(shazam$energy, shazam$Lnshazam_charts, main = "Ln shazam charts vs energy")

#DEEZER
plot(deezer$Lnspeechiness, deezer$Lndeezer_charts, main = "Ln deezer charts vs Ln speechiness")
plot(deezer$bpm, deezer$Lndeezer_charts, main = "Ln deezer charts vs bpm")
plot(deezer$danceability,deezer$Lndeezer_charts, main = "Ln deezer charts vs danceability")
plot(deezer$valence, deezer$Lndeezer_charts, main = "Ln deezer charts vs valence")
plot(deezer$Lnacousticness, deezer$Lndeezer_charts, main = "Ln deezer charts vs Ln acousticness")
plot(deezer$Lnliveness, deezer$Lndeezer_charts, main = "Ln deezer charts vs Ln liveness")
plot(deezer$Lninstrumentalness, deezer$Lndeezer_charts, main = "Ln deezer charts vs Ln Instrumentalness")
plot(deezer$energy, deezer$Lndeezer_charts, main = "Ln deezer charts vs energy")
```

```r
#Checking the normality of the dependent variable
qqnorm(apple$Lnapple_charts, main = "Normal Q-Q Plot of Apple")
qqnorm(spotify$Lnspotify_charts, main = "Normal QQ Plot of Spotify")
qqnorm(deezer$Lndeezer_charts, main = "Normal Q-Q Plot of Deezer")
qqnorm(shazam$Lnshazam_charts, main = "Normal Q-Q Plot of Shazam")

qqnorm(spotify$bpm, main = "Normal Q-Q Plot of BPM")
qqnorm(spotify$Lnacousticness, main = "Normal Q-Q Plot of Ln Acousticness")
qqnorm(spotify$danceability, main = "Normal Q-Q Plot of Danceability")
qqnorm(spotify$Lninstrumentalness, main = "Normal Q-Q Plot of Ln Instrumentalness")
qqnorm(spotify$Lnspeechiness, main = "Normal Q-Q Plot of Ln Speechiness")
qqnorm(spotify$Lnliveness, main = "Normal Q-Q Plot of Ln Liveness")
qqnorm(spotify$valence, main = "Normal Q-Q Plot of Valence")
qqnorm(spotify$energy, main = "Normal Q-Q Plot of Energy")

##########################
#End of Data Exploration#
##########################
```

```r
#SLR for each predictor for Spotify
SpotifyA <-lm(spotify$Lnspotify_charts~spotify$Lninstrumentalness)
summary(SpotifyA)

SpotifyB <- lm(spotify$Lnspotify_charts~spotify$Lnacousticness)
summary(SpotifyB)
SpotifyC <- lm(spotify$Lnspotify_charts~spotify$Lnspeechiness)
summary(SpotifyC)
SpotifyD <- lm(spotify$Lnspotify_charts~spotify$Lnliveness)
summary(SpotifyD)
SpotifyE <- lm(spotify$Lnspotify_charts~spotify$energy)
summary(SpotifyE)
SpotifyF <- lm(spotify$Lnspotify_charts~spotify$valence)
summary(SpotifyF)
SpotifyG <- lm(spotify$Lnspotify_charts~spotify$danceability)
summary(SpotifyG)
SpotifyH <- lm(spotify$Lnspotify_charts~spotify$mode)
summary(SpotifyH)
SpotifyI <- lm(spotify$Lnspotify_charts~spotify$key)
summary(SpotifyI)
par(mfrow=c(2,2))
plot(SpotifyI)
SpotifyJ <- lm(spotify$Lnspotify_charts~spotify$bpm)
summary(SpotifyJ)
```

```r
#Check the effect of other predictors
SpotifyK <- lm(spotify$Lnspotify_charts ~ spotify$bpm + spotify$key + spotify$mode + spotify$danceability + spotify$valence  + spotify$energy + spotify$Lnacousticnes
summary(SpotifyK)

SpotifyM <- lm(spotify$Lnspotify_charts ~  spotify$key:spotify$danceability + spotify$energy + spotify$Lninstrumentalness)
summary(SpotifyM)
par(mfrow=c(2,2))
plot(SpotifyM)

SpotifyN <- lm(spotify$Lnspotify_charts ~ spotify$bpm + spotify$key + spotify$danceability  + spotify$energy + spotify$Lnacousticness + spotify$Lninstrumentalness  +
summary(SpotifyN)

SpotifyO <- lm(spotify$Lnspotify_charts ~ spotify$key + spotify$danceability  + spotify$energy + spotify$Lnacousticness + spotify$Lninstrumentalness + spotify$Lnspee
summary(SpotifyO)

SpotifyP <- lm(spotify$Lnspotify_charts ~  spotify$key + spotify$danceability + spotify$energy + spotify$Lninstrumentalness)
summary(SpotifyP)

SpotifyQ <- lm(spotify$Lnspotify_charts ~ spotify$key:spotify$danceability + spotify$energy + spotify$Lninstrumentalness+spotify$mode)
summary(SpotifyQ)
```

Apple.R

2024-04-21

```
#####
# Dataframe Creation
#####

apple=read.csv("Spotify-2023.csv")
apple <- apple[, -c(1,2,3,4,5,6,7,8,9,10,12,13,14)]
apple <- apple[apple$key != "", ]
apple$mode <- ifelse(apple$mode == "Major", 1, apple$mode)
apple$mode <- ifelse(apple$mode == "Minor", 2, apple$mode)
apple$key <- ifelse(apple$key == "A", 1, apple$key)
apple$key <- ifelse(apple$key == "A#", 2, apple$key)
apple$key <- ifelse(apple$key == "B", 3, apple$key)
apple$key <- ifelse(apple$key == "C#", 4, apple$key)
apple$key <- ifelse(apple$key == "D", 5, apple$key)
apple$key <- ifelse(apple$key == "D#", 6, apple$key)
apple$key <- ifelse(apple$key == "E", 7, apple$key)
apple$key <- ifelse(apple$key == "F", 8, apple$key)
apple$key <- ifelse(apple$key == "F#", 9, apple$key)
apple$key <- ifelse(apple$key == "G", 10, apple$key)
apple$key <- ifelse(apple$key == "G#", 11, apple$key)
apple$mode <- as.numeric(as.factor(apple$mode))
apple$key <- as.numeric(as.factor(apple$key))
apple <- apple[apple$apple_charts != 0, ]
```

```r
#####
# Full Model
#####

model <- lm(apple$apple_charts ~
              apple$bpm
            + apple$key
            + apple$mode
            + apple$danceability
            + apple$valence
            + apple$energy
            + apple$acousticness
            + apple$instrumentalness
            + apple$liveness
            + apple$speechiness)

summary(model)

par(mfrow=c(2,2))
plot(model)

# Checking for Skewed Variables

hist(apple$apple_charts)

par(mfrow=c(3,3))

hist(apple$bpm)
hist(apple$key)
hist(apple$danceability)
hist(apple$valence)
hist(apple$energy)
hist(apple$acousticness)
hist(apple$instrumentalness)
hist(apple$liveness)
hist(apple$speechiness)

dev.off()
```

```r
#####
# New Full Model
#####

apple$lnapple_charts = log(apple$apple_charts)
apple$lninstrumentalness = log(apple$instrumentalness+.1)

apple <- apple[, -which(names(apple) == "apple_charts")]
apple <- apple[, -which(names(apple) == "instrumentalness")]


model <- lm(apple$lnapple_charts ~
              apple$bpm
            + apple$key
            + apple$mode
            + apple$danceability
            + apple$valence
            + apple$energy
            + apple$acousticness
            + apple$lninstrumentalness
            + apple$liveness
            + apple$speechiness)

summary(model)

par(mfrow=c(2,2))
plot(model)
dev.off()
```

```r
#####
# Best Subset Selection
#####

library(leaps)

bestsubset.full = regsubsets(lnapple_charts~., apple, nvmax=10)
bestsubset.summary=summary(bestsubset.full)
bestsubset.summary

which.max(bestsubset.summary$adjr2)

which.min(bestsubset.summary$cp)

which.min(bestsubset.summary$bic)

par(mfrow=c(2,2))

plot(bestsubset.summary$adjr2, xlab = "Number of Variables", ylab="Adjusted R-squared",
type="l")
points(4,bestsubset.summary$adjr2[4], col="red")
plot(bestsubset.summary$cp, xlab = "Number of Variables", ylab="Cp", type="l")
points(3, bestsubset.summary$cp[3], col="red", cex=2)
plot(bestsubset.summary$bic, xlab = "Number of Variables", ylab="BIC", type="l")
points(1, bestsubset.summary$bic[1], col="red", cex=2, pch=20)

dev.off()
```

```r
#####
# Cross Validating the Three Different Models Using the Validation Set Approach
#####

set.seed(1)
train = sample(1:nrow(apple), nrow(apple)/2)

lnadjr2model <- lm(apple$lnapple_charts ~
                     apple$bpm
                   + apple$acousticness
                   + apple$lninstrumentalness
                   + apple$speechiness, subset=train)

lncpmodel <- lm(apple$lnapple_charts ~
                  apple$bpm
                + apple$acousticness
                + apple$speechiness, subset=train)

lnbicmodel <- lm(apple$lnapple_charts ~
                   apple$speechiness, subset=train)

mean((apple$lnapple_charts-predict(lnadjr2model, apple))[-train]^2)

mean((apple$lnapple_charts-predict(lncpmodel, apple))[-train]^2)

mean((apple$lnapple_charts-predict(lnbicmodel, apple))[-train]^2)

lncpmodel <- lm(apple$lnapple_charts ~
                  apple$bpm
                + apple$acousticness
                + apple$speechiness)

summary(lncpmodel)
```

```r
#####
# Regression Tree
#####

library(tree)

lntree = tree(apple$lnapple_charts~
                  apple$bpm
                + apple$key
                + apple$mode
                + apple$danceability
                + apple$valence
                + apple$energy
                + apple$acousticness
                + apple$lninstrumentalness
                + apple$liveness
                + apple$speechiness)

summary(lntree)

plot(lntree)
text(lntree, cex=0.6)

lncvtree=cv.tree(lntree, K=5)
plot(lncvtree$size, lncvtree$dev, type="b")

lnprunetree=prune.tree(lntree, best=2)
plot(lnprunetree)
text(lnprunetree, cex=0.6)

# Run regression tree and get an estimated test error

set.seed(1)

train = sample(1:nrow(apple), nrow(apple)/2)

tree.apple = tree(lnapple_charts ~ ., apple, subset = train)

summary(tree.apple)

plot(tree.apple)
text(tree.apple, pretty = 0)

yhat = predict(tree.apple, newdata = apple[-train, ])
apple.test = apple[-train, "lnapple_charts"]
plot(yhat, apple.test)
abline(0,1)
mean((yhat - apple.test)^2)
```

```r
# Run bagging

library(randomForest)

set.seed(1)

bag.apple = randomForest(lnapple_charts ~ .,
                         data = apple, subset = train, mtry = 10,
                         importance = TRUE)
bag.apple

# Estimate a test error

yhat.bag = predict(bag.apple, newdata = apple[-train, ])
plot(yhat.bag, apple.test)
abline(0,1)
mean((yhat.bag - apple.test)^2)

# Run Random forest and get an estimated test error

set.seed(1)

rf.apple = randomForest(lnapple_charts ~ ., data = apple, subset = train,
                        mtry = 4, importance = TRUE)
yhat.rf = predict(rf.apple, newdata = apple[-train,])
mean((yhat.rf - apple.test)^2)

## [1] 2.010689

importance(rf.apple)

varImpPlot(rf.apple)

# Run Boosting and estimate a test error

library(gbm)

boost.apple = gbm(lnapple_charts ~ ., data = apple[train,],
                  distribution = "gaussian",
                  n.trees = 5000, interaction.depth = 4, shrinkage=0.01)
par(las = 2)
par(cex.axis = 0.62)
summary(boost.apple)

plot(boost.apple, i = "energy")
plot(boost.apple, i = "bpm")

yhat.boost = predict(boost.apple, newdata = apple[-train,], n.trees = 5000)
mean((yhat.boost - apple.test)^2)

## [1] 2.547079
```

```r
#####
# Ridge Regression
#####

# Running the model on the full dataset

library(glmnet)

grid=10^seq(10,-2, length=100)

x = model.matrix(lnapple_charts~., apple)[,-1]
y = apple$lnapple_charts

ridge=glmnet(x,y, alpha=0, lambda=grid)
plot(ridge)

#Use cross validation to select the best lambda
set.seed(1)
cv.out = cv.glmnet(x, y, alpha=0)
plot(cv.out)

bestlambda = cv.out$lambda.1se
bestlambda

ridge.out=glmnet(x,y, alpha=0)
predict(ridge.out, type="coefficients", s=bestlambda)[1:11,]

# Create training and test datasets

set.seed(1)
train = sample(1:nrow(apple), nrow(apple)/2)
test = setdiff(1:nrow(apple), train)

appletrain = apple[train, ]
appletest = apple[test, ]


# Fit model using the training set

xtrain = model.matrix(lnapple_charts~., appletrain)[,-1]
ytrain = appletrain$lnapple_charts

xtest = model.matrix(lnapple_charts~., appletest)[,-1]
ytest = appletest$lnapple_charts

# Use cross validation to select the best lambda
set.seed(1)
cv.out = cv.glmnet(x, y, alpha=0)

bestlambda = cv.out$lambda.1se
bestlambda

ridge.out=glmnet(x,y, alpha=0, lambda = bestlambda)
```

```r
# Make predictions on the test set
predictions = predict(ridge.out, newx = xtest)

# Calculate Mean Squared Error (MSE) on the test set
testmse = mean((predictions - ytest)^2)

# Print the test MSE
print(testmse)
```

```
## [1] 1.952162
```

```r
#####
# The Lasso
#####

# Running the model on the full dataset
x = model.matrix(lnapple_charts~., apple)[,-1]
y = apple$lnapple_charts

lasso=glmnet(x, y, alpha=1, lambda=grid)
plot(lasso)

set.seed(1)
cv.out = cv.glmnet(x, y, alpha=1)
plot(cv.out)

bestlambda = cv.out$lambda.1se
bestlambda

lasso.out=glmnet(x,y, alpha=1, lambda=grid)
predict(lasso.out, type="coefficients", s=bestlambda)[1:11,]

# Fit model using the training set

# Use cross validation to select the best lambda
set.seed(1)
cv.out = cv.glmnet(xtrain, ytrain, alpha=1)

bestlambda = cv.out$lambda.1se
bestlambda

lasso.out=glmnet(xtrain,ytrain, alpha=1, lambda = bestlambda)

# Make predictions on the test set
predictions = predict(lasso.out, newx = xtest)

# Calculate Mean Squared Error (MSE) on the test set
testmse = mean((predictions - ytest)^2)

# Print the test MSE
print(testmse)
```

```
#####
# PCA
#####

apply(apple, 2, mean)

apply(apple, 2, sd)

# prcomp perform principal components analysis

pr.out = prcomp(apple, scale = TRUE)

pr.out$center # This gives the mean of the original variables

pr.out$scale # This gives the sd of the original variables

pr.out$rotation # This gives loading vectors of the principal components

dim(pr.out$x) # x is the matrix with principal component score vectors

biplot(pr.out, scale = 0)

biplot(pr.out, scale = 0, col = c("white", "red"))

pr.out$sdev # This gives the standard deviation of each principal component

pr.var = pr.out$sdev^2
pve = pr.var/sum(pr.var)
pve # This is the proportion variance explained

par(mfrow = c(1,2))
plot(pve, xlab = "Principal Component", ylab = "Proportion of Variance Explained",
     lim = c(0,1), type = "b")

plot(cumsum(pve), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained", lim = c(0,1), type = "b")
```

Each member's contribution to the project

Ann-Marie: Introduction, Background of Data, Data Exploration, Spotify's findings

Thomas: Introduction and Executive Summary, Introduction & Overview and Background of Dataset Portion of Presentation, Editing Presentation and Report

Samuel: Model Selection and Further Statistical Analysis, Heat Maps, Conclusion