Analyzing Spotify Trends and Predicting Song Popularity

Danail Krzhalovski Sandra Andovska

5/28/2020

1. Data

Spotify is a digital music service that enables users to remotely source millions of different songs on various record labels from a laptop, smartphone or other device. To recommend new music to users, and to be able to internally classify songs, Spotify assigns each song values from 13 different attributes/features. These features are mostly numerical values, but include some categorical data as well (the key the song is in, for instance). Spotify also assigns each song a popularity score, based on total number of clicks/listens. This dataset contains approximately 19000 songs from different Spotify playlists (i.e. '00s Rock Anthems', '50 Latin Classics', 'Alternative Hip Hop', etc.), which includes a popularity score and a set of metrics for each one. The dataset is available on Kaggle [1]. There are two .cvs files, and after merging and removing irrelevant information we are working with the following 16 variables:

- **song_name**: The name of the song observed.
- **arist_name**: The name of the artist.
- **song_duration_ms**: Duration of the song in miliseconds. This will be changed to minutes for convenience.
- **acousticness**: A confidence measure from 0.0 to 1.0 of whether the song is acoustic. 1.0 represents high confidence the song is acoustic.
- **danceability**: Danceability describes how suitable a song is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- **energy**: This 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. For example, death metal has high energy, while a Bach prelude scores low on the scale. Features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. Songs with an energy value closer to 1 are considered highly energetic.
- **instrumentalness**: Predicts whether a song contains no vocals.
- **key**: This represents the overall musical key the song is composed in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C#/Db, 2 = D, and so on. For easier interpretation, we substitute the integers with their Pitch notation: 0 = C, 1 = C#, 2 = D, 3 = D#, 4 = E, 5 = F, 7 = F#, 8 = G, 9 = G#, 10 = A, 11 = A#, 12 = B.
- **liveness**: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the song was performed live. A value above 0.8 provides strong likelihood that the song is live.

- **loudness**: This explains the overall loudness of a song in decibels (dB), which are averaged across the entire song and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
- **audio_mode**: The mode of a composition indicates the modality (major or minor) of a song, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0 in the dataset, but for easier interpretation we modify these variables as we did we the keys.
- **speechiness**: It detects the presence of spoken words in a song. The more exclusively speech-like the recording, the closer to 1.0 the attribute value. We can say that values below 0.33 most likely represent instrumental music and other non-speech-like tracks.
- **tempo**: The overall estimated tempo of a song in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
- **time_signature**: An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
- **audio_valence**: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a song. Songs with high valence sound more positive.
- **song_popularity**: This is our dependent variable which takes values between 0 and 100 based on total streams.

Below is a simple overview of our data:

```
##                    song_name           artist_name song_duration_ms
## 1 Boulevard of Broken Dreams             Green Day           262333
## 2                  In The End           Linkin Park           216933
## 3          Seven Nation Army     The White Stripes           231733
## 4                  By The Way Red Hot Chili Peppers           216933
## 5          How You Remind Me            Nickelback           223826
## 6           Bring Me To Life           Evanescence           235893
##   acousticness danceability energy instrumentalness key liveness loudness
## 1     0.005520        0.496  0.682         2.94e-05   8   0.0589   -4.095
## 2     0.010300        0.542  0.853         0.00e+00   3   0.1080   -6.407
## 3     0.008170        0.737  0.463         4.47e-01   0   0.2550   -7.828
## 4     0.026400        0.451  0.970         3.55e-03   0   0.1020   -4.938
## 5     0.000954        0.447  0.766         0.00e+00  10   0.1130   -5.065
## 6     0.008950        0.316  0.945         1.85e-06   4   0.3960   -3.169
##   audio_mode speechiness   tempo time_signature audio_valence song_popularity
## 1          1      0.0294 167.060              4         0.474              73
## 2          0      0.0498 105.256              4         0.370              66
## 3          1      0.0792 123.881              4         0.324              76
## 4          1      0.1070 122.444              4         0.198              74
## 5          1      0.0313 172.011              4         0.574              56
## 6          0      0.1240 189.931              4         0.320              80
```

2. Data Exploration

We can see that we are working with 18835 observations. Two of these features are represented as factors: names and the artists of the songs. Judging by their count, we see that "song_name" has 13070 levels which is less than the number of observations. This doesn't seem too strange, since there are a lot of songs titled

the same. But even so, the songs are collected from different playlists so we need to search for duplicates and remove them. Of course, prior to any analysis, we check whether our data has any missing values, counting column-wise:

```
## 'data.frame':    18835 obs. of  16 variables:
##  $ song_name       : Factor w/ 13070 levels "'Til I Get It Right",..: 1607 5660 9804 1817 5093 1698 6
##  $ artist_name     : Factor w/ 7564 levels "$ain't","$uicideBoy$",..: 2575 3911 6858 5472 4817 2068 5
##  $ song_duration_ms: int  262333 216933 231733 216933 223826 235893 199893 213800 222586 203346 ...
##  $ acousticness    : num  0.00552 0.0103 0.00817 0.0264 0.000954 0.00895 0.000504 0.00148 0.00108 0.0
##  $ danceability    : num  0.496 0.542 0.737 0.451 0.447 0.316 0.581 0.613 0.33 0.542 ...
##  $ energy          : num  0.682 0.853 0.463 0.97 0.766 0.945 0.887 0.953 0.936 0.905 ...
##  $ instrumentalness: num  2.94e-05 0.00 4.47e-01 3.55e-03 0.00 1.85e-06 1.11e-03 5.82e-04 0.00 1.04e-
##  $ key             : int  8 3 0 0 10 4 4 2 1 9 ...
##  $ liveness        : num  0.0589 0.108 0.255 0.102 0.113 0.396 0.268 0.152 0.0926 0.136 ...
##  $ loudness        : num  -4.09 -6.41 -7.83 -4.94 -5.07 ...
##  $ audio_mode      : int  1 0 1 1 1 0 0 1 1 1 ...
##  $ speechiness     : num  0.0294 0.0498 0.0792 0.107 0.0313 0.124 0.0624 0.0855 0.0917 0.054 ...
##  $ tempo           : num  167 105 124 122 172 ...
##  $ time_signature  : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ audio_valence   : num  0.474 0.37 0.324 0.198 0.574 0.32 0.724 0.537 0.234 0.374 ...
##  $ song_popularity : int  73 66 76 74 56 80 81 76 80 81 ...
```

```
##        song_name      artist_name song_duration_ms     acousticness
##                0                0                0                0
##     danceability           energy instrumentalness              key
##                0                0                0                0
##         liveness         loudness       audio_mode      speechiness
##                0                0                0                0
##            tempo   time_signature    audio_valence  song_popularity
##                0                0                0                0
```

After removing duplicates, we are now working with 14926 observations.

```
## [1] 14926    16
```

As mentioned earlier, the data was already cleaned up and tidied, but some additional modifications had to be made. We created factors for the 'audio_mode' and 'key' variables to make the data easier to interpret. Anyone with at least a bit of musical knowledge would prefer and actually find it easier to understand the analysis if the person could see the keys (C, C#, etc.) and modes (major, minor) instead of going back to the description of features to check their numerical values. We too transform the song duration from miliseconds to minutes, and factorize the time signature since it does not provide us a true numerical property.

```
##                    song_name           artist_name song_duration acousticness
## 1 Boulevard of Broken Dreams             Green Day      4.372217     0.005520
## 2                 In The End           Linkin Park      3.615550     0.010300
## 3          Seven Nation Army     The White Stripes      3.862217     0.008170
## 4                 By The Way Red Hot Chili Peppers      3.615550     0.026400
## 5          How You Remind Me            Nickelback      3.730433     0.000954
## 6           Bring Me To Life           Evanescence      3.931550     0.008950
##   danceability energy instrumentalness key liveness loudness audio_mode
## 1        0.496  0.682         2.94e-05  G#   0.0589   -4.095      major
## 2        0.542  0.853         0.00e+00  D#   0.1080   -6.407      minor
```

```
## 3        0.737  0.463         4.47e-01   C   0.2550   -7.828      major
## 4        0.451  0.970         3.55e-03   C   0.1020   -4.938      major
## 5        0.447  0.766         0.00e+00  A#   0.1130   -5.065      major
## 6        0.316  0.945         1.85e-06   E   0.3960   -3.169      minor
##   speechiness   tempo time_signature audio_valence song_popularity
## 1      0.0294 167.060              4         0.474              73
## 2      0.0498 105.256              4         0.370              66
## 3      0.0792 123.881              4         0.324              76
## 4      0.1070 122.444              4         0.198              74
## 5      0.0313 172.011              4         0.574              56
## 6      0.1240 189.931              4         0.320              80
```

We begin our analysis by checking which artists are a part of Spotify playlists the most. Throughout the years, Lady Gaga has been dominant in the pop genre since 2010s, so the fact that she showed up first in our list does not surprise us. Following is Drake, a major influencer on all generations, bringig his fusion of hip hop and contemporary R&B with trap and dancehall music. Kanye West has too shown to be a remarkable alternative and experimental hip hop artist, with his fair share in the pop genre. Eminem is one of the most famous artists in hip hop making his debut in 1996. By looking at the entire list, we can see that all artists are from different time periods, genres and cultures, so we can say that our data is not biased towards a specific type of music nor a time period.

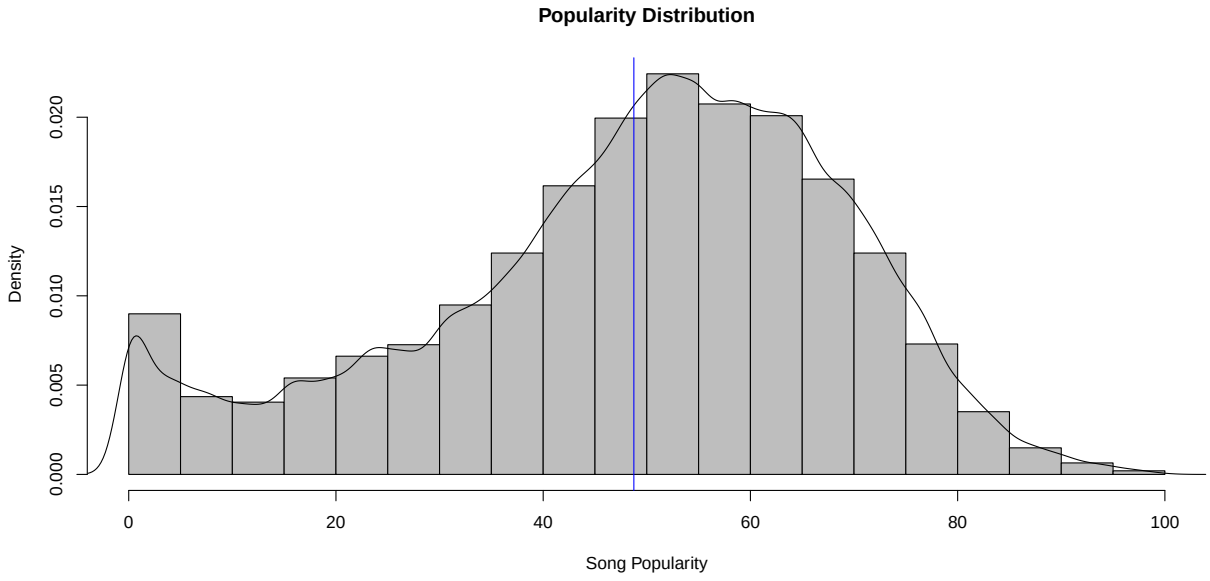
```
##                 Number of Songs
## Lady Gaga                    57
## Drake                        45
## Kanye West                   44
## Eminem                       25
## Khalid                       25
## Ed Sheeran                   23
## David Guetta                 22
## Gucci Mane                   22
## Kendrick Lamar               22
## Celia Cruz                   20
## Maroon 5                     20
## Future                       19
## Major Lazer                  19
## The Beatles                  19
## Imagine Dragons              18
## R3HAB                        18
## Rihanna                      18
## The Weeknd                   18
## Hank Williams                17
## Johnny Cash                  17
```

To review our dataset's content and shape, we provide the summaries and conclude that there is a lot of skewness in most of the predictors, except maybe for the duration of the song and valence. We can observe a negative skew in song popularity (our target variable) as well. The results are represented as follows, and we continue with a more detailed analysis.

```
##    song_name         artist_name    song_duration     acousticness
##  Fire   :    8   Lady Gaga :   57   Min.   : 0.200   Min.   :0.000001
##  Heaven :    8   Drake     :   45   1st Qu.: 3.066   1st Qu.:0.023600
##  Alright:    7   Kanye West:   44   Median : 3.531   Median :0.139000
##  Better :    7   Eminem    :   25   Mean   : 3.649   Mean   :0.270452
```

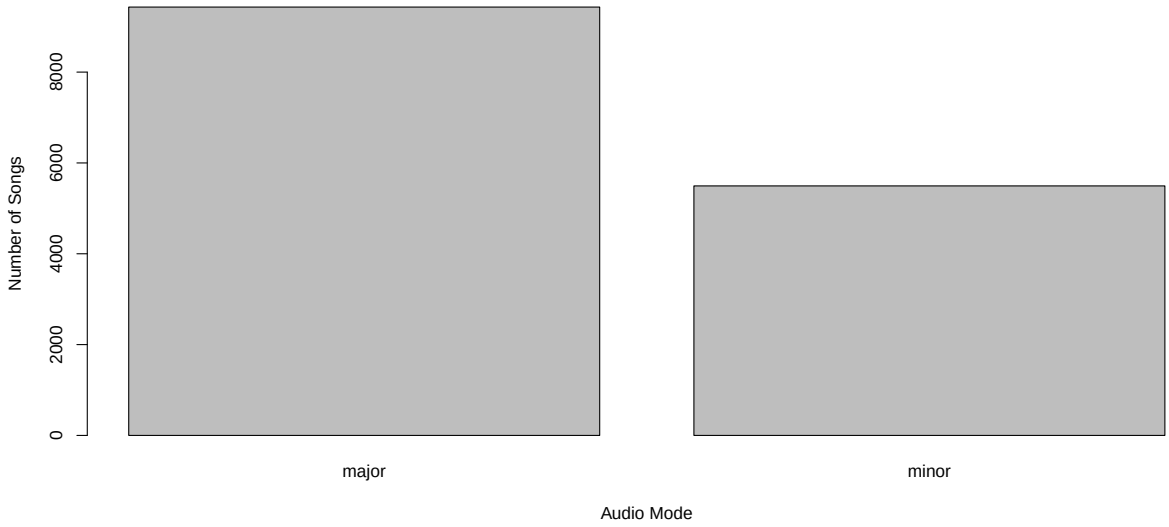
```
##  Breathe:    7   Khalid    :   25   3rd Qu.: 4.079   3rd Qu.:0.458000
##  Fall   :    7   Ed Sheeran:   23   Max.   :29.989   Max.   :0.996000
##  (Other):14882   (Other)   :14707
##   danceability        energy        instrumentalness          key
##  Min.   :0.0000   Min.   :0.00107   Min.   :0.0000000   C      :1734
##  1st Qu.:0.5240   1st Qu.:0.49600   1st Qu.:0.0000000   G      :1654
##  Median :0.6360   Median :0.67200   Median :0.0000208   C#     :1594
##  Mean   :0.6245   Mean   :0.63976   Mean   :0.0920668   A      :1410
##  3rd Qu.:0.7400   3rd Qu.:0.81800   3rd Qu.:0.0051050   D      :1399
##  Max.   :0.9870   Max.   :0.99900   Max.   :0.9970000   F      :1257
##                                                         (Other):5878
##     liveness         loudness       audio_mode    speechiness
##  Min.   :0.0109   Min.   :-38.768   major:9432   Min.   :0.00000
##  1st Qu.:0.0930   1st Qu.: -9.389   minor:5494   1st Qu.:0.03720
##  Median :0.1220   Median : -6.750                Median :0.05410
##  Mean   :0.1804   Mean   : -7.677                Mean   :0.09942
##  3rd Qu.:0.2240   3rd Qu.: -4.991                3rd Qu.:0.11300
##  Max.   :0.9860   Max.   :  1.585                Max.   :0.94100
##
##      tempo        time_signature audio_valence    song_popularity
##  Min.   :  0.00   0:    3        Min.   :0.0000   Min.   :  0.00
##  1st Qu.: 98.12   1:   67        1st Qu.:0.3320   1st Qu.: 37.00
##  Median :120.02   3:  684        Median :0.5270   Median : 52.00
##  Mean   :121.11   4:13977        Mean   :0.5270   Mean   : 48.75
##  3rd Qu.:139.94   5:  195        3rd Qu.:0.7278   3rd Qu.: 63.75
##  Max.   :242.32                  Max.   :0.9840   Max.   :100.00
##
```

By taking a a look at how the popularity scores are distributed, one thing is noticeable instantly. The data appears to be slightly negatively skewed, with majority of the songs having a popularity score more than 40. The mean popularity score at a value of 48.75 and median is 52.

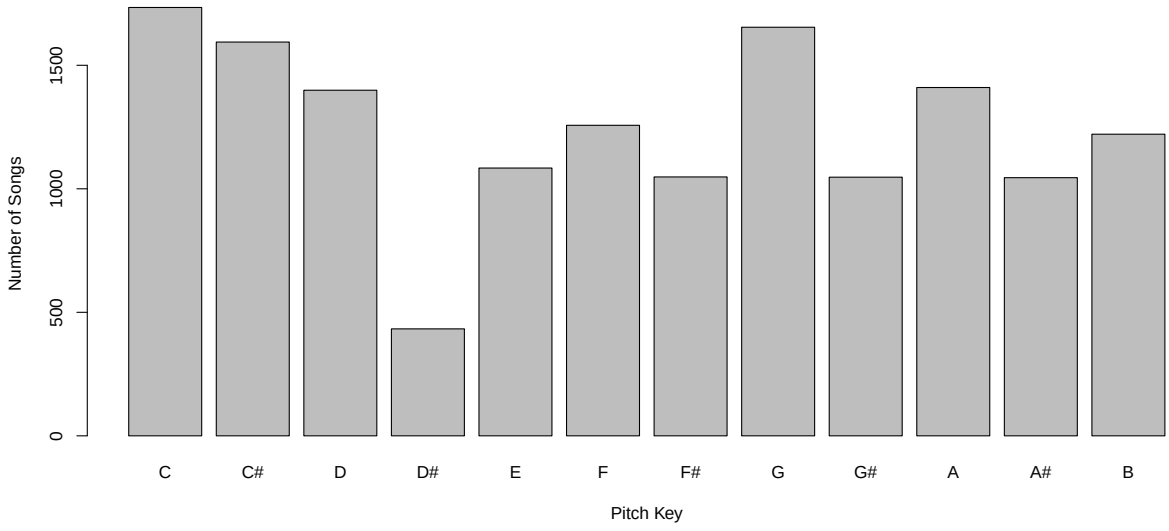


We can assume that the audio mode, key, tempo and time signature of a song are composition elements, which we can make us of to describe and classify music.

In simple terms, the emotional center of music comes from one of two places: the major chord or the minor chord. If you're listening to music and you can sense happiness and you're at ease, you're probably listening to a song that uses mostly major chords to create that feeling. Any chord has a relative major or minor version. If we wanted to get moodier and possibly sad, we could think of contrasting those major chords into minors. It looks like people lean more towards songs with a major mode than those with a minor mode. Does this mean people like happier songs?



In addition to the audio mode, the association of musical keys with specific emotional or qualitative characteristic was fairly common prior to the 20th century. When Mozart or Beethoven or Schubert wrote a piece in a Ab major, for example, they were well aware that this was the 'key of the grave' and knew that many in their audiences were as well. In a study by Spotify's Kenny Ning, it was shown that more than a third of 30 million songs observed are in one of four keys: G major, C major, D major, or A major. This reveals the kind of sounds we tend to see more commonly in music: bold, upbeat majors tend to outvote the moodier minors. In our data, the most common key is C, but very close to it are G and C# (Db).



After doing some research, we were able to find poetic interpretations of these keys, written by Christian Schubart - a German poet, organist, composer, and journalist that layed out his thougts in "Ideas for an Aesthetic in Audio Art" (originally titled "Ideen zu einer Aesthetik der Tonkunst"), published in 1806 [2]:

- **C major**: Completely pure. Its character is: innocence, simplicity, naïvety, children's talk.
- **C minor**: Declaration of love and at the same time the lament of unhappy love. All languishing, longing, sighing of the love-sick soul lies in this key.

- **C# major** (equivalent to D-flat): A leering key, degenerating into grief and rapture. It cannot laugh, but it can smile; it cannot howl, but it can at least grimace its crying. Consequently only unusual characters and feelings can be brought out in this key.

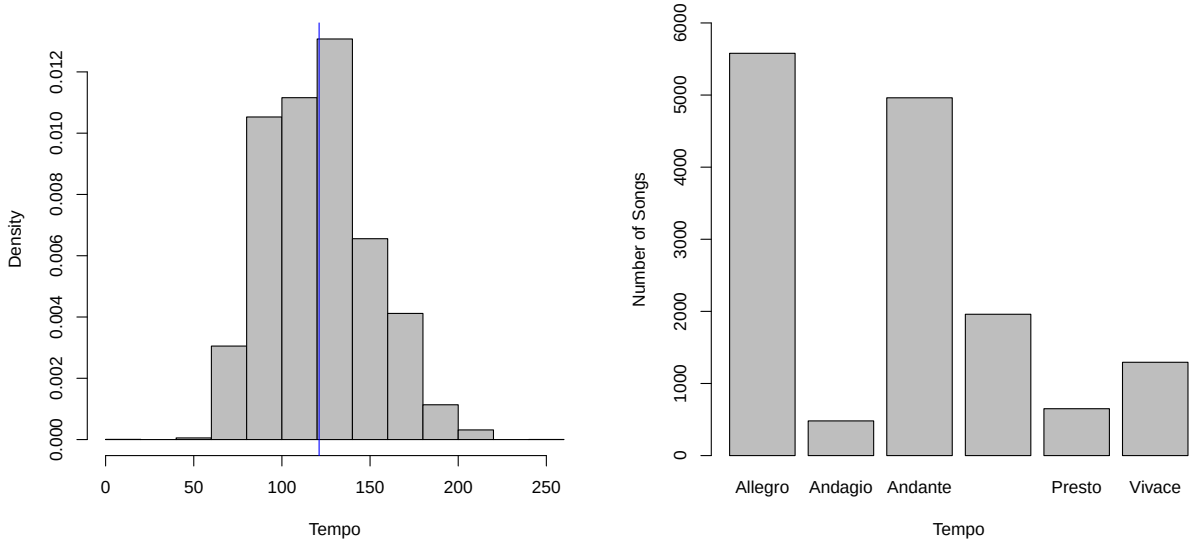
- **C# minor**: Penitential lamentation, intimate conversation with God, the friend and help-meet of life; sighs of disappointed friendship and love lie in its radius.

- **G major**: Everything rustic, idyllic and lyrical, every calm and satisfied passion, every tender gratitude for true friendship and faithful love,–in a word every gentle and peaceful emotion of the heart is correctly expressed by this key.
- **G minor**: Discontent, uneasiness, worry about a failed scheme; bad-tempered gnashing of teeth; in a word: resentment and dislike.

The least common musical key popular songs are composed in is D#:

- **D# major** (equivalent to E-flat): The key of love, of devotion, of intimate conversation with God.
- **D# minor**: Feelings of the anxiety of the soul's deepest distress, of brooding despair, of blackest depression, of the most gloomy condition of the soul. Every fear, every hesitation of the shuddering heart, breathes out of horrible D# minor. If ghosts could speak, their speech would approximate this key.

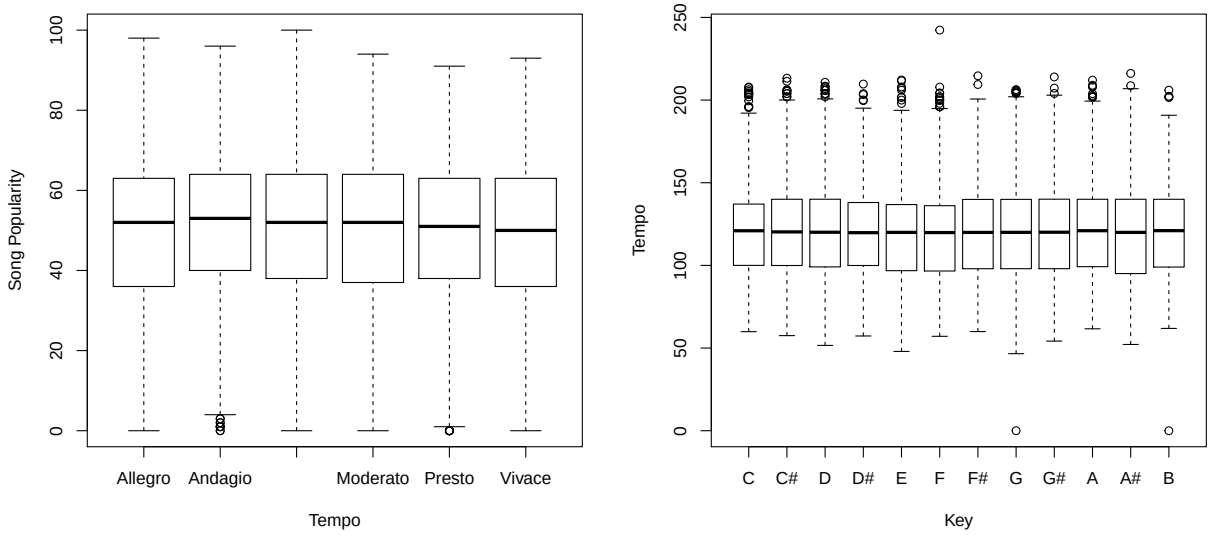
Popular songs tend to have a faster tempo with an average BPM of 121.105. This is commonly known as 'allegro', ranging from 120 - 156 BPM. Modern music tempos are in this range: techno (120-140 BPM), house (115-130 BPM), hip-hop (80-120), and similar.



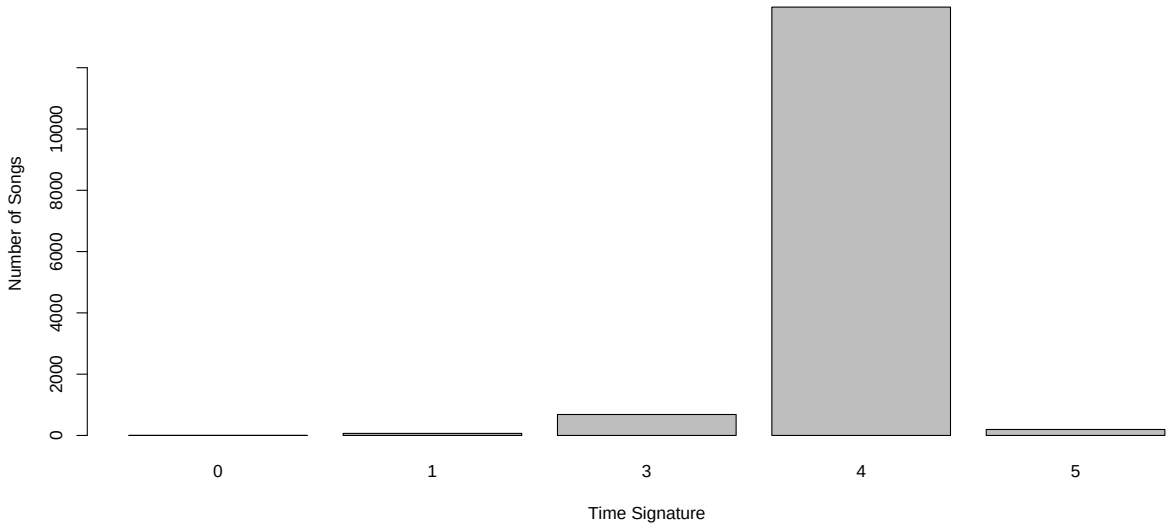
To elaborate, we can show the number of songs falling under each category of tempo [3]:

- **Adagio** - slowly with great expression (66-76 bpm)
- **Andante** - at a walking pace (76-108 bpm)
- **Moderato** - at a moderate speed (108-120 bpm)
- **Allegro** - fast, quickly, and bright (120-156 bpm)
- **Vivace** - lively and fast (156-176 bpm)
- **Presto** - very, very fast (168-200 bpm)

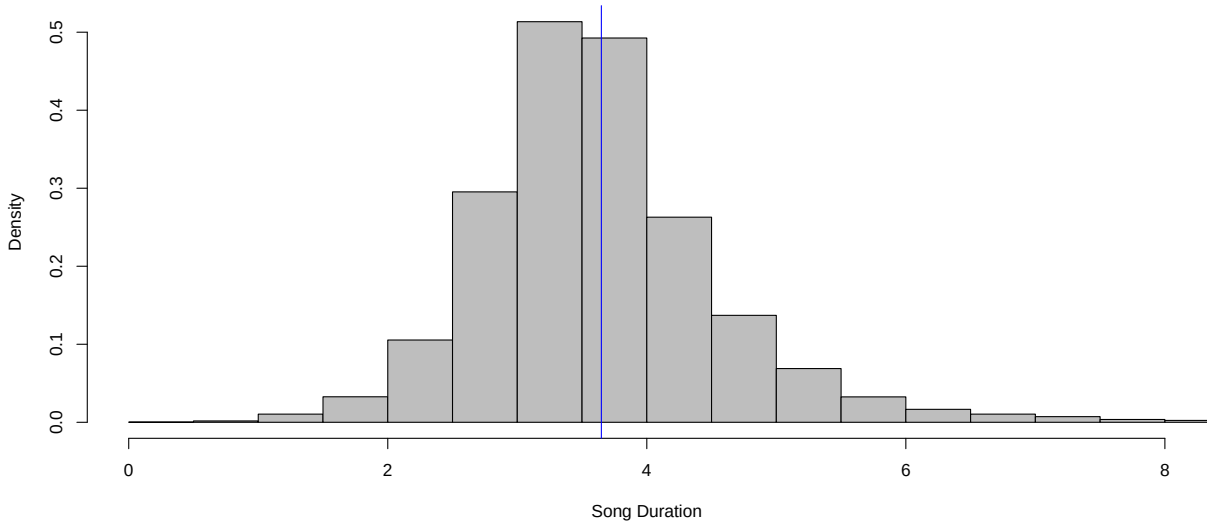
and see that opposed to most songs being in allegro, the least are in andagio. We also have a clear indication of the consistency of mean tempos across all songs and their popularities, with a symmetrical distribution. We could say that the data is approximately normally distributed in terms of the songs' tempo. We too can observe a consistency of mean tempos across all keys.



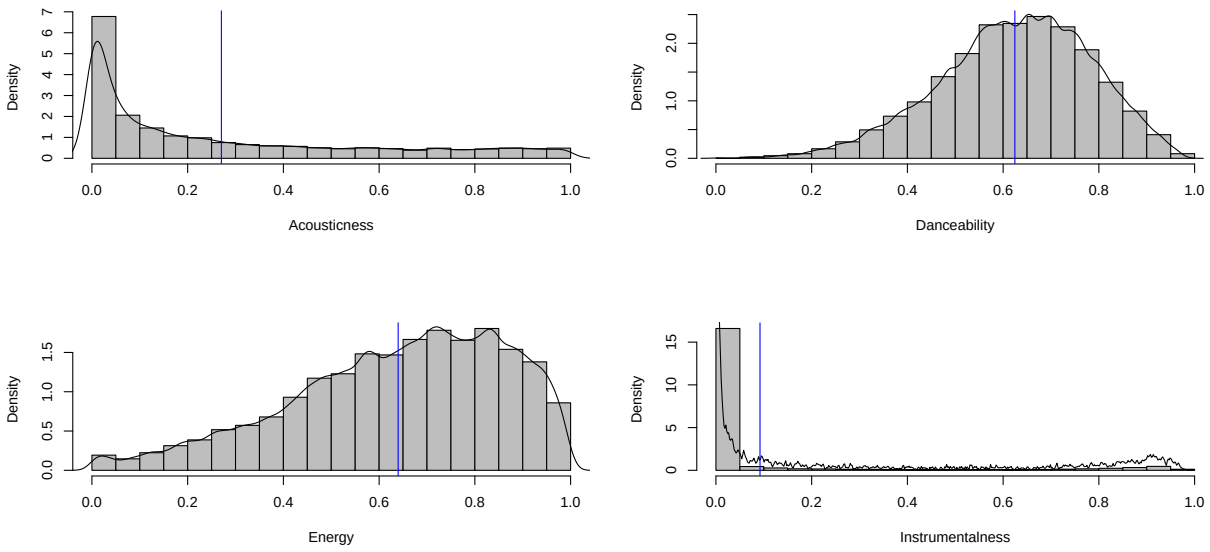
The most common time signature used in music, and not surprisingly in our dataset, is the 4/4 meter (denoted as 4 in the column of time signatures). This is known as "4x4"or "common time".

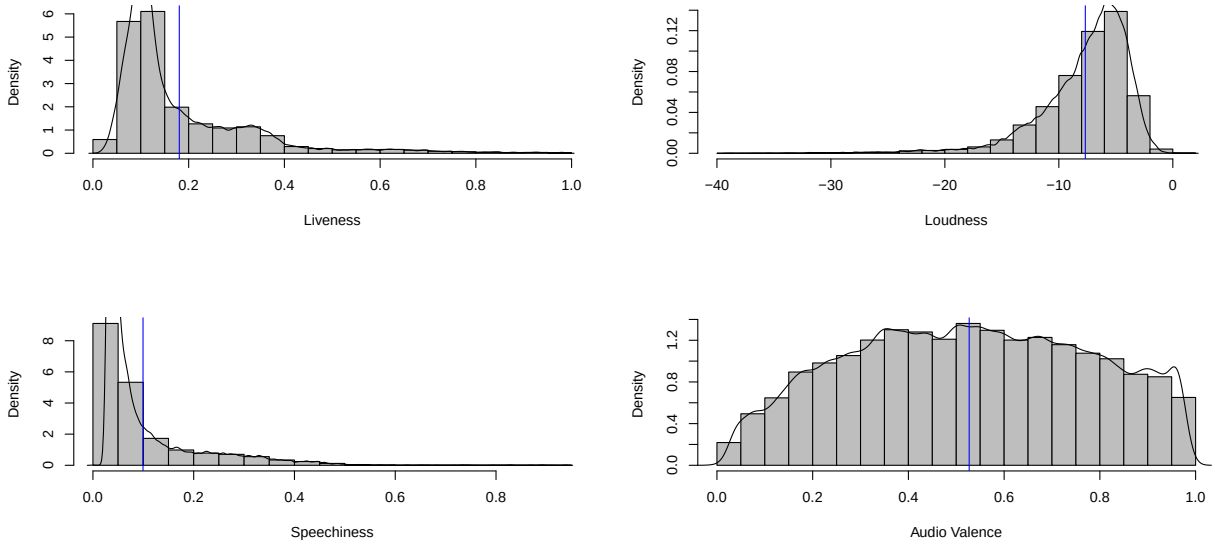


We too can see that the average duration of a song is between 3 and 4 minutes:



Now, we continue by analyzing the univariate distribution of the numerical variables not included in the previous plots:





As observed in the summaries of the variables, the majority are either highly-left or highly-right skewed, while 'danceability' and 'energy' have somewhat a similar distribution. This leads to the assumption that they might be correlated. Note that:

- Danceability is most dense in the interval [0.5, 0.8] telling us that popular songs are quite danceable.
- Energy peaks around 0.7 and is most dense around the same interval.
- Audio valance has a symmetrical distribution and it too might be correlated with danceability and energy, considering happy songs make people energetic and wanting to dance.
- The majority of the songs tend to have very low liveness, i.e. they are not recorded during a live session, and songs are between -10 and 0 dB. Acousticness and speechiness seem to have a similar distribution, saying that most songs are acoustic and have less spoken words (which too might be concluded looking at the instrumentalness over the songs).

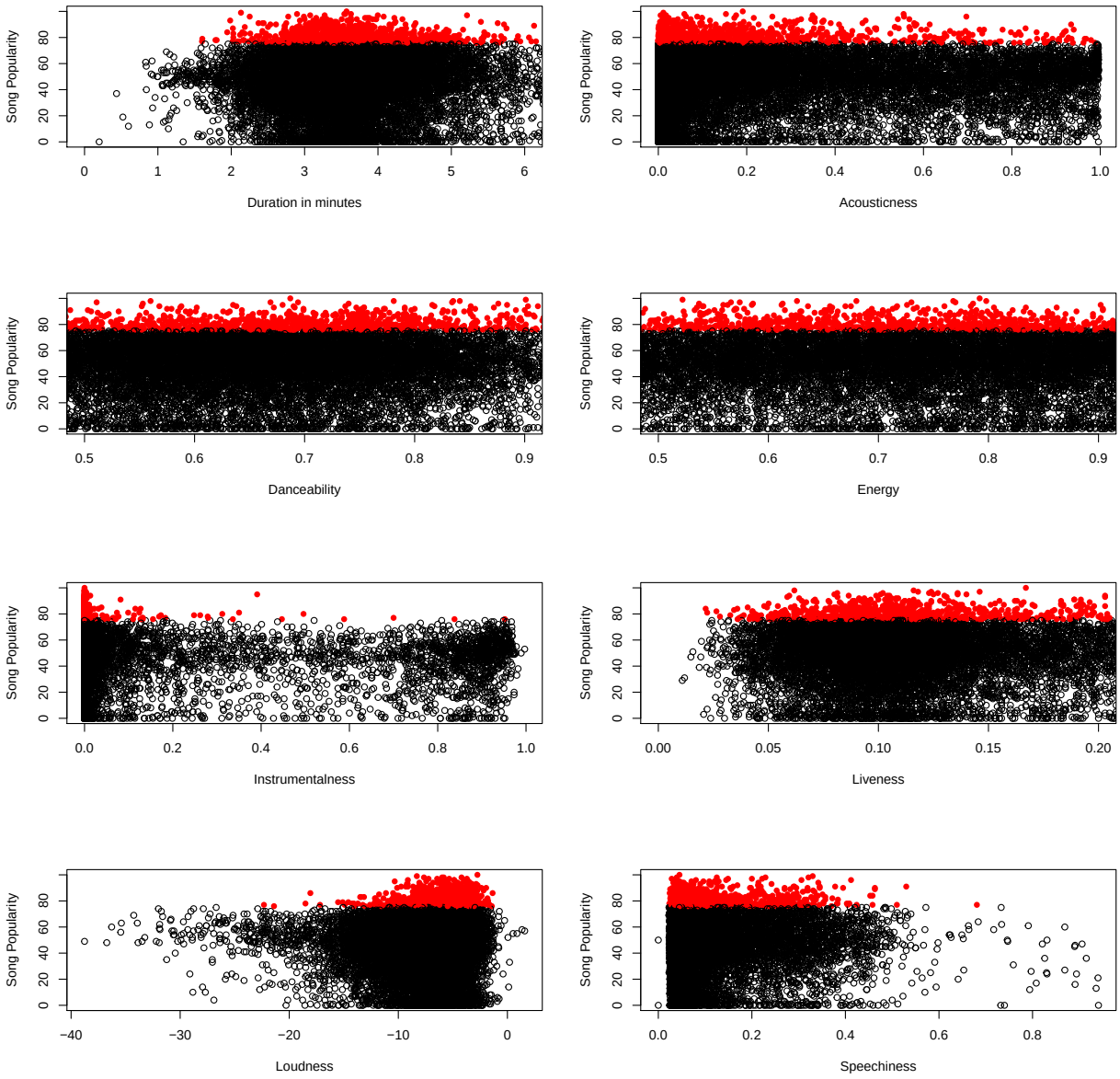
We can divide the song popularity into low and high, based on whether it is below or above 75. We choose 75, since this is the border of the top 25% of songs. Continuing our analysis, we observe that:

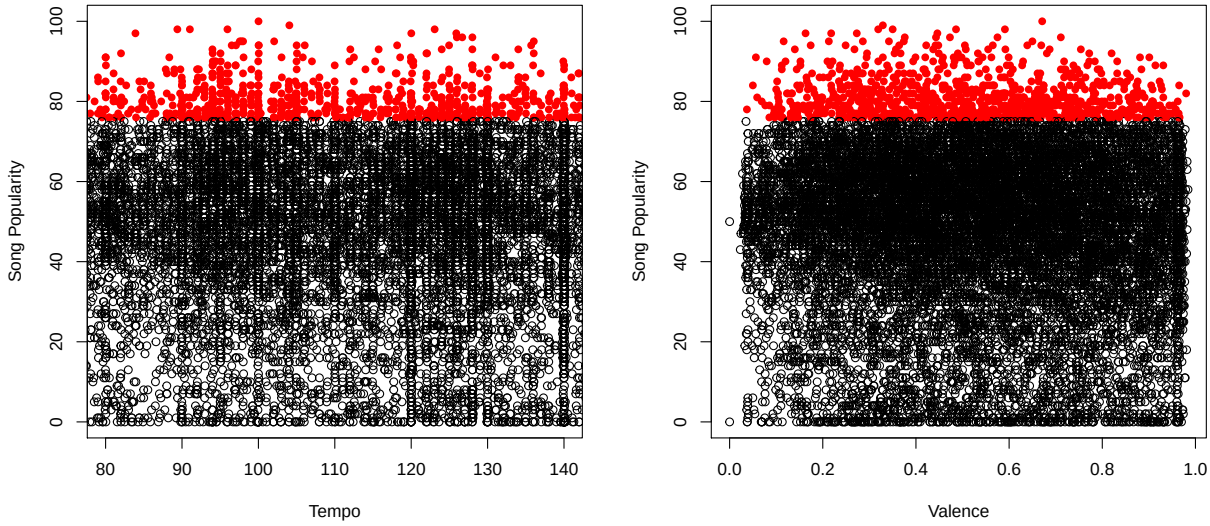
- 981 songs are very popular, with a rating above 75, which comprises approximately 61% of our total data.
- Songs that are more popular tend to last 3-4 minutes, and their acousticness level is below 0.2 meaning that the more a song is popular, the less likely that it's acoustic.
- Danceability and energy have a very similar mass, saying when songs are more popular they have a moderate-to-high value in both of them, i.e. the songs are more danceable and energetic.
- Instrumentalness is most likely to be 0 when a song is very popular, and when liveness is more than 0.8 we can say that there is strong likelihood that the song is studio-recorded. Our values of liveness for songs with popularity > 75 are concentrated around 0.1.

- Popular songs tend to be louder, mostly ranging from -6 to -4 dB, and contain less-to-no lyrics.
- As mentioned, the average tempo is around 120 BPM, with popular songs having a tempo in the range of (90,130). No surprise, people also like to listen to moderately cheerful songs - just the right amount of mellow and happy.

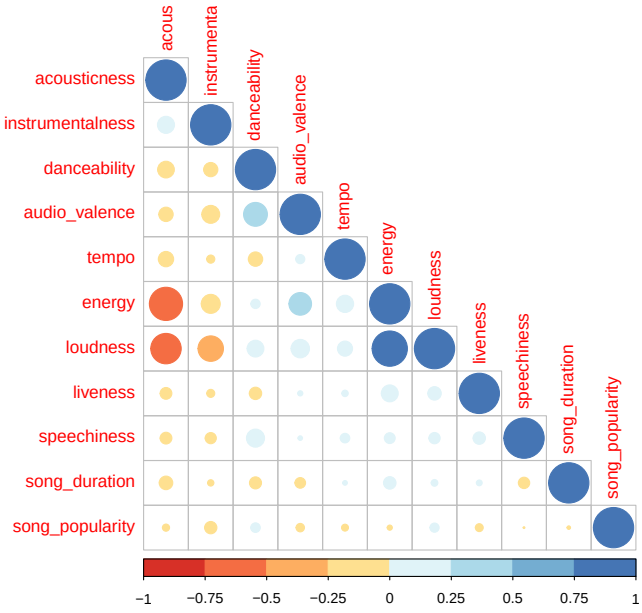
```
## Number of highly popular songs:  981
```

```
## Percentage of highly popular songs:  61.3125
```

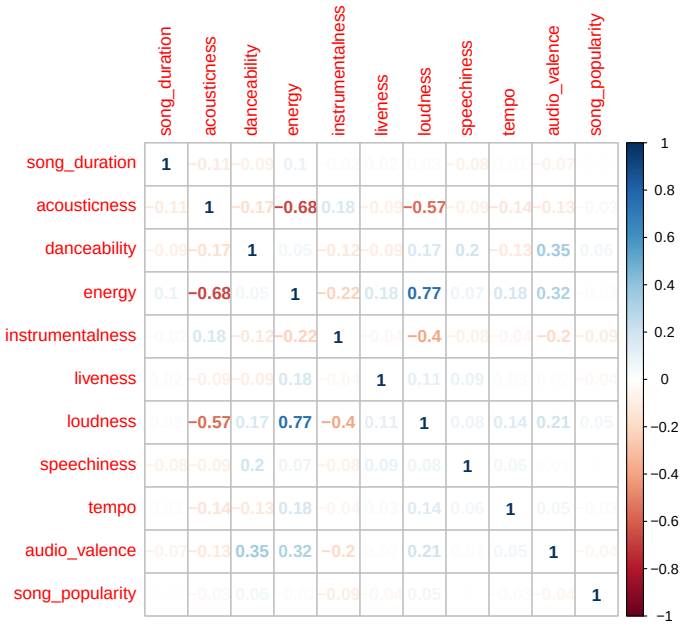




We further need to check the covariance matrix to see the direction of the linear relationship between variables, but since correlation measures both its direction and strength, we use a colour correlation plot just to see how our data behaves.



And the numerical values for the correlation between features are shown in the following figure:



In the plots above, we observe in the strongest tones existing correlations between some of the predictors. The strongest correlation is between loudness and energy, meaning if the loudness of a track increases then chances of it being energetic are quite higher and vice versa. The second LARGEST is a negative correlation between energy and acOusticness. Hence, we face a problem of multicollinearity.

3. Regression Models

Judging by the looks of our data, and the relationship among the predictors and the response variable, we do not expect a linear model to help us predict a song's popularity, but we do try out several in the following section.

3.1. Linear Regression

We start by checking how the null model fits the data. This essentially has the same meaning as "null hypothesis": the model if the null hypothesis is true. This explains the marginal distribution on response variable not taking into account any predictors, i.e. really just the mean of the outcome. We show the R2 statistic providing us a proportion of variance explained, and Root Mean Squared Error. The mean squared error tells us how close a regression line is to a set of points, and we are simply taking its root.

```
##
## Call:
## lm(formula = song_popularity ~ 1, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.751 -11.751   3.249  14.999  51.249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.7509     0.1668   292.3   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.38 on 14925 degrees of freedom
```

R2 is 0, indicating that this model explains none of the variability of the response data around its mean. RMSE is equivalent to Residual Standard Error, and its value for the null model is 20.3794652.

Next, we fit the full (saturated) We start by estimating the full model, containing all regressors available. The aim is to check the signifcance of each predictor. Here, we give rise to the issue of multicolinearity. We use R's function vif() which stands for Variance Inflation Factors and assesses how much the variance of an estimated regression coefficient increases if the predictors are correlated. If no factors are correlated, VIF's will be all 1.

```
##                      GVIF Df GVIF^(1/(2*Df))
## song_duration    1.048139  1        1.023786
## acousticness     2.055786  1        1.433801
## danceability     1.463095  1        1.209584
## energy           3.906748  1        1.976549
## instrumentalness 1.267751  1        1.125945
## key              1.145984 11        1.006213
## liveness         1.062911  1        1.030976
## loudness         3.031050  1        1.740991
## audio_mode       1.113837  1        1.055385
## speechiness      1.122503  1        1.059482
## tempo            1.078182  1        1.038355
## time_signature   1.123786  4        1.014695
## audio_valence    1.419079  1        1.191251
```

We see that all VIF values are around 1, excluding acousticness with a VIF of 2.0557, energy with 3.9067, and loudness with 3.0310. Since energy has the highest VIF, we decide to drop it from our dataset and refit our model and double check whether a predictor would cause an issue in our model.

```
##                      GVIF Df GVIF^(1/(2*Df))
## song_duration    1.040430  1        1.020015
## acousticness     1.582849  1        1.258113
## danceability     1.340921  1        1.157981
## instrumentalness 1.227024  1        1.107711
## key              1.144927 11        1.006171
## liveness         1.048448  1        1.023938
## loudness         1.766375  1        1.329050
## audio_mode       1.113686  1        1.055313
## speechiness      1.118138  1        1.057420
## tempo            1.076754  1        1.037667
## time_signature   1.111210  4        1.013268
## audio_valence    1.217447  1        1.103380
```

Now, by checking the VIF values of the coefficient, we can observe that all of them are around 1. This justifies our choice to drop one of the predictors at the mere beginning.
We continue by checking the summary of the full model:

$$popularity = \beta_0 + \beta_1 duration + \beta_2 acousticness + \beta_3 danceability + \beta_4 instrumentalness + \beta_5 key + \beta_6 liveness + \beta_7 loudness + \beta_8 mode + \beta_9 speechiness + \beta_{10} tempo + \beta_{11} signature + \beta_{12} valence$$

```
##
## Call:
## lm(formula = song_popularity ~ ., data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.835 -11.341   2.973  14.723  49.261
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      43.931789  11.687240   3.759 0.000171 ***
## song_duration    -0.236131   0.162061  -1.457 0.145122
## acousticness     -0.490793   0.696667  -0.704 0.481140
## danceability      8.283547   1.209247   6.850 7.67e-12 ***
## instrumentalness -8.025050   0.760488 -10.552  < 2e-16 ***
## keyC#             1.476751   0.705860   2.092 0.036444 *
## keyD             -0.509240   0.725548  -0.702 0.482771
## keyD#            -0.284885   1.089123  -0.262 0.793655
## keyE             -0.003492   0.789221  -0.004 0.996470
## keyF              0.453760   0.753206   0.602 0.546892
## keyF#             1.781391   0.797577   2.234 0.025531 *
## keyG             -0.531999   0.693702  -0.767 0.443155
## keyG#            -0.144608   0.792039  -0.183 0.855133
## keyA             -0.520417   0.727469  -0.715 0.474385
## keyA#             1.094147   0.800505   1.367 0.171702
## keyB              0.896895   0.766661   1.170 0.242071
## liveness         -4.912967   1.164947  -4.217 2.49e-05 ***
## loudness          0.124839   0.054595   2.287 0.022231 *
## audio_modeminor  -0.803428   0.361207  -2.224 0.026144 *
## speechiness      -4.376519   1.685801  -2.596 0.009438 **
## tempo            -0.016123   0.005898  -2.734 0.006268 **
## time_signature1   7.855605  11.922473   0.659 0.509976
## time_signature3   8.631406  11.699000   0.738 0.460654
## time_signature4   9.594996  11.676414   0.822 0.411237
## time_signature5  10.643056  11.758745   0.905 0.365417
## audio_valence    -7.483818   0.735173 -10.180  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.17 on 14900 degrees of freedom
## Multiple R-squared:  0.02235,    Adjusted R-squared:  0.02071
## F-statistic: 13.62 on 25 and 14900 DF,  p-value: < 2.2e-16
```

We can see that the p-value of the F-statistic is $<$ 2.2e-16, which is highly significant. This means that, at least, one of the predictor variables is significantly related to the target variable, in our case - song popularity. To see which predictor variables are significant, we examine the coefficients table, which shows the estimate of regression beta coefficients and the associated t-statitic p-values:

```
##                     Estimate   Std. Error       t value     Pr(>|t|)
## (Intercept)      43.93178886 11.687239999   3.758953257 1.712772e-04
## song_duration    -0.23613147  0.162060673  -1.457055976 1.451220e-01
## acousticness     -0.49079347  0.696666959  -0.704487936 4.811400e-01
## danceability      8.28354658  1.209246910   6.850169732 7.665196e-12
## instrumentalness -8.02505046  0.760488431 -10.552495126 6.106932e-26
```

```
## keyC#             1.47675140  0.705860080   2.092130492 3.644372e-02
## keyD             -0.50923985  0.725547676  -0.701869592 4.827715e-01
## keyD#            -0.28488451  1.089122800  -0.261572437 7.936547e-01
## keyE             -0.00349208  0.789220966  -0.004424718 9.964697e-01
## keyF              0.45375972  0.753205587   0.602438064 5.468917e-01
## keyF#             1.78139066  0.797577096   2.233502777 2.553059e-02
## keyG             -0.53199877  0.693702437  -0.766897648 4.431545e-01
## keyG#            -0.14460774  0.792038827  -0.182576578 8.551327e-01
## keyA             -0.52041742  0.727468781  -0.715381100 4.743848e-01
## keyA#             1.09414702  0.800505078   1.366820832 1.717021e-01
## keyB              0.89689463  0.766660691   1.169871674 2.420713e-01
## liveness         -4.91296689  1.164946884  -4.217331238 2.486707e-05
## loudness          0.12483869  0.054594820   2.286639755 2.223082e-02
## audio_modeminor  -0.80342812  0.361207179  -2.224286130 2.614412e-02
## speechiness      -4.37651872  1.685801285  -2.596105932 9.437953e-03
## tempo            -0.01612336  0.005897724  -2.733828184 6.267662e-03
## time_signature1   7.85560503 11.922473406   0.658890547 5.099762e-01
## time_signature3   8.63140590 11.698999808   0.737790071 4.606536e-01
## time_signature4   9.59499567 11.676414446   0.821741615 4.112371e-01
## time_signature5  10.64305594 11.758744721   0.905118377 3.654173e-01
## audio_valence    -7.48381802  0.735172709 -10.179673341 2.933943e-24
```

For a given the predictor, the t-statistic evaluates whether or not there is significant association between the predictor and the outcome variable, that is whether the beta coefficient of the predictor is significantly different from zero. It can be seen that, changing the following variables is significantly associated to changes in song popularity:

- danceability
- instrumentalness
- liveness
- speechiness
- tempo
- audio valence

i.e. they all have a p-value <0.001, meaning that there is less than a 0.1% chance that the coefficient might be equal to 0 and thus be insignificant. But the estimates of the regression coefficients are subject to sampling uncertainty. Therefore, we will never exactly estimate the true value of these parameters from sample data in an empirical application. However, we may construct confidence intervals, at a 95% level:

```
##                         2.5 %       97.5 %
## (Intercept)       21.02335848 66.840219245
## song_duration     -0.55379036  0.081527415
## acousticness      -1.85634654  0.874759608
## danceability       5.91327365 10.653819518
## instrumentalness  -9.51570149 -6.534399439
## keyC#              0.09317867  2.860324122
## keyD              -1.93140269  0.912922988
## keyD#             -2.41969938  1.849930373
```

```
## keyE              -1.55046241  1.543478253
## keyF              -1.02261604  1.930135468
## keyF#              0.21804128  3.344740038
## keyG              -1.89174101  0.827743480
## keyG#             -1.69710143  1.407885949
## keyA              -1.94634586  0.905511025
## keyA#             -0.47494157  2.663235599
## keyB              -0.60585479  2.399644041
## liveness          -7.19640631 -2.629527459
## loudness           0.01782611  0.231851260
## audio_modeminor   -1.51143870 -0.095417543
## speechiness       -7.68089694 -1.072140490
## tempo             -0.02768363 -0.004563098
## time_signature1  -15.51391182 31.225121872
## time_signature3  -14.30007516 31.562886962
## time_signature4  -13.29221530 32.482206635
## time_signature5  -12.40553251 33.691644389
## audio_valence     -8.92484711 -6.042788933
```

For some of the estimates, the interval does not contain the value zero. Those that we suspect do not have a great effect on the response variable are:

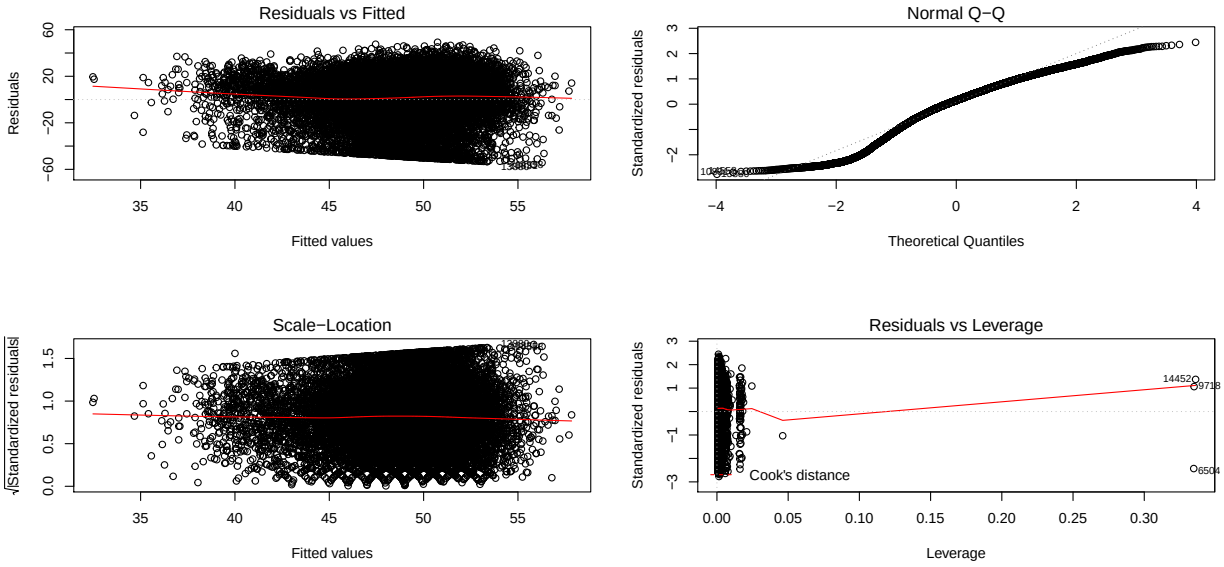
- song duration
- acousticness
- key
- time signature.

The overall quality of the model can be assesed by examining RMSE, equaling to 20.1673798, which is not a huge improvement from the null model. Neither has the variability around the mean been explained, now equaling to 0.0223457. We suspect that some of the predictors influence others. We suspect that the effect of acousticness on popularity also depends on instrumentalness, similarly to how valance has an effect on popularity as it might be dependent on loudness and danceability. We too expect that the effect of loudness on popularity depends on liveness too.

```
##
## Call:
## lm(formula = song_popularity ~ . + acousticness:instrumentalness +
##     audio_valence:loudness:danceability + liveness:loudness,
##     data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.876 -11.069   2.856  14.458  50.757
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       41.13776   11.61645   3.541 0.000399 ***
## song_duration                      0.01530    0.16190   0.094 0.924727
## acousticness                      -3.29082    0.71806  -4.583 4.62e-06 ***
## danceability                      16.06783    1.44720  11.103  < 2e-16 ***
```

```
## instrumentalness                    -16.85062    1.02999 -16.360  < 2e-16 ***
## keyC#                                 1.49865    0.70008   2.141 0.032315 *
## keyD                                 -0.32141    0.71954  -0.447 0.655102
## keyD#                                 0.05513    1.08010   0.051 0.959294
## keyE                                  0.45105    0.78324   0.576 0.564707
## keyF                                  0.62442    0.74696   0.836 0.403200
## keyF#                                 1.92438    0.79090   2.433 0.014979 *
## keyG                                 -0.26021    0.68804  -0.378 0.705299
## keyG#                                -0.10701    0.78537  -0.136 0.891624
## keyA                                 -0.32458    0.72159  -0.450 0.652853
## keyA#                                 1.29214    0.79387   1.628 0.103620
## keyB                                  1.00377    0.76027   1.320 0.186763
## liveness                              1.82550    2.54053   0.719 0.472428
## loudness                             -0.36508    0.10155  -3.595 0.000325 ***
## audio_modeminor                      -1.01887    0.35841  -2.843 0.004479 **
## speechiness                          -5.31229    1.67512  -3.171 0.001521 **
## tempo                                -0.01042    0.00586  -1.779 0.075330 .
## time_signature1                       1.45120   11.82952   0.123 0.902365
## time_signature3                       1.72068   11.60901   0.148 0.882172
## time_signature4                       2.72429   11.58622   0.235 0.814109
## time_signature5                       3.38635   11.66889   0.290 0.771665
## audio_valence                         0.92340    1.29487   0.713 0.475779
## acousticness:instrumentalness        20.49181    1.89877  10.792  < 2e-16 ***
## liveness:loudness                     0.93399    0.32166   2.904 0.003693 **
## danceability:loudness:audio_valence   1.68304    0.21633   7.780 7.73e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20 on 14897 degrees of freedom
## Multiple R-squared:  0.03904,    Adjusted R-squared:  0.03723
## F-statistic: 21.61 on 28 and 14897 DF,  p-value: < 2.2e-16
```

Although explanation of variability has bumped up to 0.0390365, we can see that the RMSE is now 19.9964998 and has not decreased by even 0.5%, which is not at all significant. Hence, we continue by analyzing the residuals on our full model.



Residuals vs Fitted

There isn't a clear pattern in the residuals and they are m distributed. Since, there isn't any evidence of heteroscedasticity (non-constant variance in the errors).

Normal Q-Q

Notice the points fall along a line in the middle of the graph, but curve off in the extremities. This means our data has more extreme values. We could not fix the issue, because applying various transformation to the response variable yielded even worse results.

Scale-Location

The presence of heteroscedasticity can be seen also from how the studentized residuals spread along the ranges of predicted variables. We can see that the residuals are spread equally, even though our horizontal line isn't quite straight.

Residuals vs Leverage

We can see that there are points which are outside the red dashed Cook's distance line. These are points that would be influential in the model and removing them would likely alter the regression results.

```
## [1] 14923    13
```

```
##
## Call:
## lm(formula = song_popularity ~ ., data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.809 -11.343   2.976  14.720  49.252
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      51.870833   2.823129  18.374  < 2e-16 ***
## song_duration    -0.244245   0.162097  -1.507  0.13189
## acousticness     -0.472157   0.696661  -0.678  0.49795
## danceability      8.238986   1.209509   6.812 1.00e-11 ***
## instrumentalness -8.047878   0.760620 -10.581  < 2e-16 ***
```

```
## keyC#             1.491977   0.705865   2.114  0.03456 *
## keyD             -0.496376   0.725516  -0.684  0.49388
## keyD#            -0.274547   1.089017  -0.252  0.80096
## keyE              0.007137   0.789170   0.009  0.99278
## keyF              0.464069   0.753160   0.616  0.53780
## keyF#             1.794300   0.797537   2.250  0.02448 *
## keyG             -0.495391   0.693810  -0.714  0.47523
## keyG#            -0.131719   0.792003  -0.166  0.86791
## keyA             -0.508589   0.727440  -0.699  0.48447
## keyA#             1.106694   0.800471   1.383  0.16682
## keyB              0.895868   0.766859   1.168  0.24273
## liveness         -4.910030   1.164869  -4.215 2.51e-05 ***
## loudness          0.126194   0.054591   2.312  0.02081 *
## audio_modeminor  -0.798848   0.361165  -2.212  0.02699 *
## speechiness      -4.425014   1.685939  -2.625  0.00868 **
## tempo            -0.016404   0.005901  -2.780  0.00544 **
## time_signature3   0.779001   2.588781   0.301  0.76348
## time_signature4   1.749456   2.483989   0.704  0.48126
## time_signature5   2.797145   2.859864   0.978  0.32806
## audio_valence    -7.491754   0.735083 -10.192  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.16 on 14898 degrees of freedom
## Multiple R-squared:  0.02234,    Adjusted R-squared:  0.02076
## F-statistic: 14.18 on 24 and 14898 DF,  p-value: < 2.2e-16
```

We observe no significant decrease in RMSE (20.1647085) after removing the observations. Although there is a change of sign in some predictors, there is no real change in their significance, nor in performance of the model. However, we leave them out of the dataset. As we can see, many of the variables used in our multiple regression model are in fact not associated with the response (i.e. they are non-significant) and including such irrelevant variables leads to unnecessary complexity in the resulting model. This was also suspected when we constructed confidence intervals for the estimated coefficients of the regression. Some of the intervals did not contain 0, which leads to the rejection of the null hypothesis i.e. they should have a say in predicting song popularity.

```
##                        2.5 %       97.5 %
## (Intercept)      46.33715155 57.404514621
## song_duration    -0.56197455  0.073485363
## acousticness     -1.83769874  0.893384435
## danceability      5.86819952 10.609772472
## instrumentalness -9.53878719 -6.556968204
## keyC#             0.10839473  2.875559587
## keyD             -1.91847628  0.925723358
## keyD#            -2.40915388  1.860060574
## keyE             -1.53973316  1.554008156
## keyF             -1.01221797  1.940355565
## keyF#             0.23103051  3.357570290
## keyG             -1.85534363  0.864561413
## keyG#            -1.68414329  1.420705340
## keyA             -1.93446101  0.917282649
## keyA#            -0.46232723  2.675714755
## keyB             -0.60727096  2.399006523
## liveness         -7.19331665 -2.626742874
```

```
## loudness          0.01918904  0.233198029
## audio_modeminor  -1.50677641 -0.090919744
## speechiness      -7.72966207 -1.120366376
## tempo            -0.02797049 -0.004837658
## time_signature3  -4.29532879  5.853329862
## time_signature4  -3.11946875  6.618380078
## time_signature5  -2.80854072  8.402831366
## audio_valence    -8.93260720 -6.050900916
```

This is the reason why we perform a feature selection using backward selection. We start with all variables in the model, and remove the variable with the largest p-value. The new (p-1) - variable model is fit, and the variable with the largest p-value is removed. Here, we consider Akaike Information Criterion (AIC) and remove model that shows lowest AIC.

```
## Start:  AIC=89680.39
## song_popularity ~ song_duration + acousticness + danceability +
##     instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + time_signature + audio_valence
##
##                    Df Sum of Sq     RSS   AIC
## - time_signature    3       998 6058755 89677
## - acousticness      1       187 6057944 89679
## - key              11      8854 6066611 89680
## <none>                          6057757 89680
## - song_duration     1       923 6058680 89681
## - audio_mode        1      1989 6059747 89683
## - loudness          1      2173 6059930 89684
## - speechiness       1      2801 6060558 89685
## - tempo             1      3142 6060900 89686
## - liveness          1      7224 6064982 89696
## - danceability      1     18867 6076625 89725
## - audio_valence     1     42236 6099993 89782
## - instrumentalness  1     45521 6103278 89790
##
## Step:  AIC=89676.85
## song_popularity ~ song_duration + acousticness + danceability +
##     instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + audio_valence
##
##                    Df Sum of Sq     RSS   AIC
## - acousticness      1       292 6059047 89676
## <none>                          6058755 89677
## - key              11      8951 6067707 89677
## - song_duration     1       937 6059693 89677
## - audio_mode        1      1997 6060752 89680
## - loudness          1      2216 6060971 89680
## - speechiness       1      2716 6061471 89682
## - tempo             1      3238 6061993 89683
## - liveness          1      7171 6065926 89693
## - danceability      1     19899 6078655 89724
## - audio_valence     1     41913 6100669 89778
## - instrumentalness  1     45761 6104517 89787
##
## Step:  AIC=89675.57
```

```
## song_popularity ~ song_duration + danceability + instrumentalness +
##     key + liveness + loudness + audio_mode + speechiness + tempo +
##     audio_valence
##
##                     Df Sum of Sq     RSS   AIC
## <none>                           6059047 89676
## - song_duration      1       820 6059867 89676
## - key               11      9053 6068100 89676
## - audio_mode         1      1967 6061014 89678
## - speechiness        1      2669 6061717 89680
## - tempo              1      3084 6062131 89681
## - loudness           1      4297 6063344 89684
## - liveness           1      7084 6066132 89691
## - danceability       1     20650 6079698 89724
## - audio_valence      1     41979 6101026 89777
## - instrumentalness   1     45473 6104520 89785
```

We observe that we have excluded 2 predictors from the fit (time signature and acousticness), such that the selected model is:

$$popularity = \beta_0 \ + \beta_1 duration \ + \beta_2 key \ + \beta_3 mode \ + \beta_4 speechiness \ + \beta_5 tempo \ + \beta_6 loudness$$
$$+ \beta_7 liveness \ + \beta_8 danceability \ + \beta_9 valence \ + \beta_{10} instrumentalness$$

```
##
## Call:
## lm(formula = song_popularity ~ song_duration + key + audio_mode +
##     speechiness + tempo + loudness + liveness + danceability +
##     audio_valence + instrumentalness, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.822 -11.363   2.986  14.705  49.211
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      53.369855   1.481531  36.023  < 2e-16 ***
## song_duration    -0.228112   0.160665  -1.420  0.15569
## keyC#             1.509311   0.705594   2.139  0.03245 *
## keyD             -0.499508   0.725454  -0.689  0.49112
## keyD#            -0.332309   1.087596  -0.306  0.75996
## keyE             -0.005610   0.789087  -0.007  0.99433
## keyF              0.449765   0.752932   0.597  0.55028
## keyF#             1.799797   0.797383   2.257  0.02401 *
## keyG             -0.512981   0.693621  -0.740  0.45957
## keyG#            -0.151004   0.791808  -0.191  0.84876
## keyA             -0.522644   0.727344  -0.719  0.47242
## keyA#             1.078943   0.800113   1.348  0.17752
## keyB              0.907227   0.766706   1.183  0.23672
## audio_modeminor  -0.794157   0.361063  -2.199  0.02786 *
## speechiness      -4.285768   1.672722  -2.562  0.01041 *
## tempo            -0.016147   0.005863  -2.754  0.00590 **
## loudness          0.151452   0.046589   3.251  0.00115 **
## liveness         -4.857645   1.163758  -4.174 3.01e-05 ***
```

```
## danceability      8.481296   1.190084   7.127 1.08e-12 ***
## audio_valence    -7.460624   0.734241 -10.161  < 2e-16 ***
## instrumentalness -8.017524   0.758129 -10.575  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.16 on 14902 degrees of freedom
## Multiple R-squared:  0.02213,    Adjusted R-squared:  0.02082
## F-statistic: 16.86 on 20 and 14902 DF,  p-value: < 2.2e-16
```

The coefficients that have the largest effect on our target varialbe are:

- Danceability with 8.481296, meaning with each unit increase, song popularity increases by roughly 8.48;
- Instrumentalness with -8.017524, meaning with each unit increase, song popularity decreases by roughly 8.02;
- Audio valence with -7.460624, meaning with each unit increase, song popularity decreases by roughly 7.46.

Of course we all suspected that the more danceable a song is the more popularity it would gain, but what struck us off guard is that with every increase of audio valence of a song, its popularity decreases **dramitically**. This might be due to the fact that music features subtle and gradual changes along these dimensions (song attributes) within a broader emotion category remaining the same throughout the musical segment. Sometimes, listeners do not even perceive any discrete emotion in the music. They merely perceive a certain level of arousal. The effect of instrumentalness is clear, less vocals in a song do not seem that appealing the people's taste.

Once we are done with training our model, we can not just assume that it is going to work well on data that it has not seen before. In other words, we can not be sure that the model will have the desired accuracy and variance in production environment. We need some kind of assurance of the accuracy of the predictions that our model is putting out. For this, we need to validate our model. We use 5-Fold Cross Validaion: we split the entire data randomly into 5 folds, because using 5 or 10 as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. We fit the model using the K-1 i.e. 4 folds, and validate the model using the the remaining K-th fold. We repeat this process until every K-fold serves as the test set. Then take the average of the scores, which will be the performance metric for the model.

Computational issues aside, we choose K-fold CV instead of LOOCV, because it often gives more accurate estimates of the test error rate. To perform this, we use R's built-in functions cv.glm() function. We compare the results of the full and reduced obtained by backward selection. We get two errors in the delta component. The first one is the average mean-squared error that we obtain from doing K-fold CV. The second is the average mean-squared error that you obtain from doing K-fold CV, but with a bias correction because we have not used LOOCV.

```
##                Root Mean Squared Error
## Full model                    20.18114
## Reduced Model                 20.18051
```

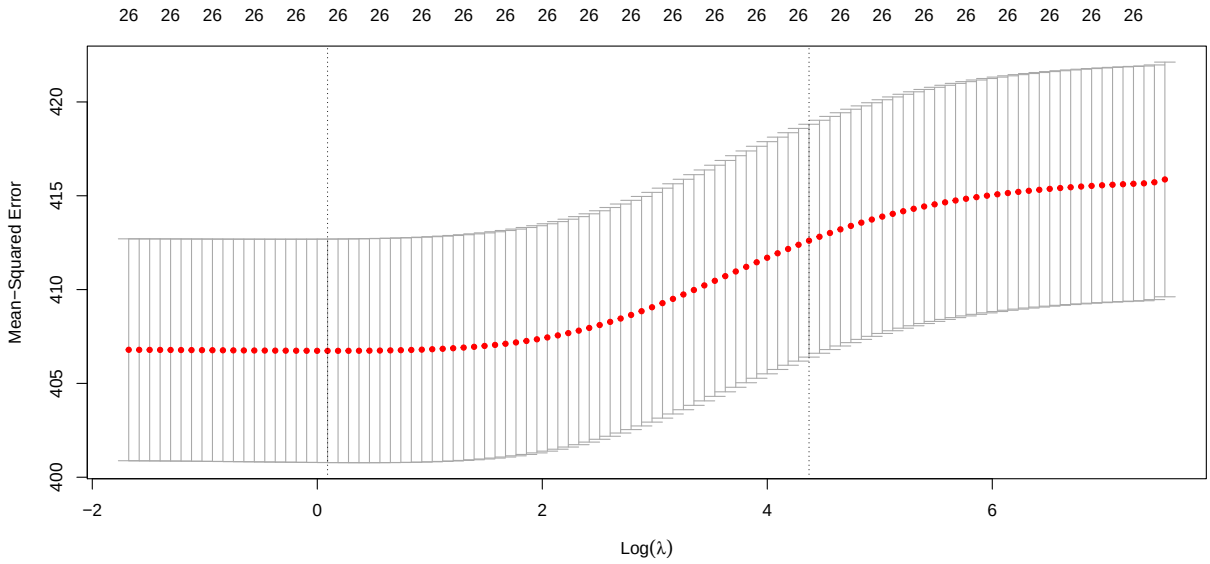
The results vary from time to time, because of the constant update of "random.seed" - a vector containing the random number generator (RNG) state for random number generation in R. 9/20 times the reduced model turned out to be a better fit than the full model, but the error did not reduce significantly. Because we couldn't get a lower error, we suspect that the reason for this is either removing the energy variable (although it seemed justified at the time), or that the variable selection is not good enough, so we further use Ridge regression to impose a penalty term on predictors to impose a bias on the esitmators and LASSO for the latter.

3.2 Ridge Regression

The least square method shown previously finds the coefficients that best fit the data. One more condition to be added is that it also finds the unbiased coefficients. Unbiased means that least squares doesn't consider which independent variable is more important than others, i.e. simply finds the coefficients for a given data set. Therefore, this kind of model becomes more complex as new variables are added. It can be said that an OLS model has the lowest bias and the highest variance. To overcome this issue, we fit a ridge regression to our model. We are trying to find coefficient estimates that would minimize the RSS in a linear model, by adding a shrinkage penalty called L2-norm, which is the sum of the squared coefficients $\lambda \sum_{j=1}^{p} \beta_j^2$ where λ is a constant that can fine-tune amount of the penalty. This penalty is small when the coefficient estimates are close to zero, so it has the effect of shrinking towards 0. The tuning parameter lambda serves to control the relative impact of these two terms on the regression coefficient estimates. As lambda increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias. This too may resolve the issue of multicollinearity, therefore we include the "energy" variable in this model.

To perform Ridge Regression, we divide our data into a training and testing set and fit a ridge regression model using 5-fold Cross Validation to find the optimal value for λ.

We see that with lambda larger than, we have a constant error and there is no change in the plot. As more regressors are pushed to zero, the MSE in the validation set increases. As we decrease the lambda parameter, we can see that MSE decreases. This starts from setting lambda to approxiamtely 2, and stops when lambda is approximately 0 and we have a constant decrease in MSE. The most appropriate values for lambda are plotted with two verical lines: one for the "best" value of lambda and other for 1 standard error above it.



The optimal value of lambda is very close to 0, which means that there is very little-to-no regularization effect and the model we fit will be equivalent to the previous full model. This is because ridge does not shrink any variables to exactly 0, hence all predictors will be used in the fit.

```
## RMSE_ridge:  20.24823
```

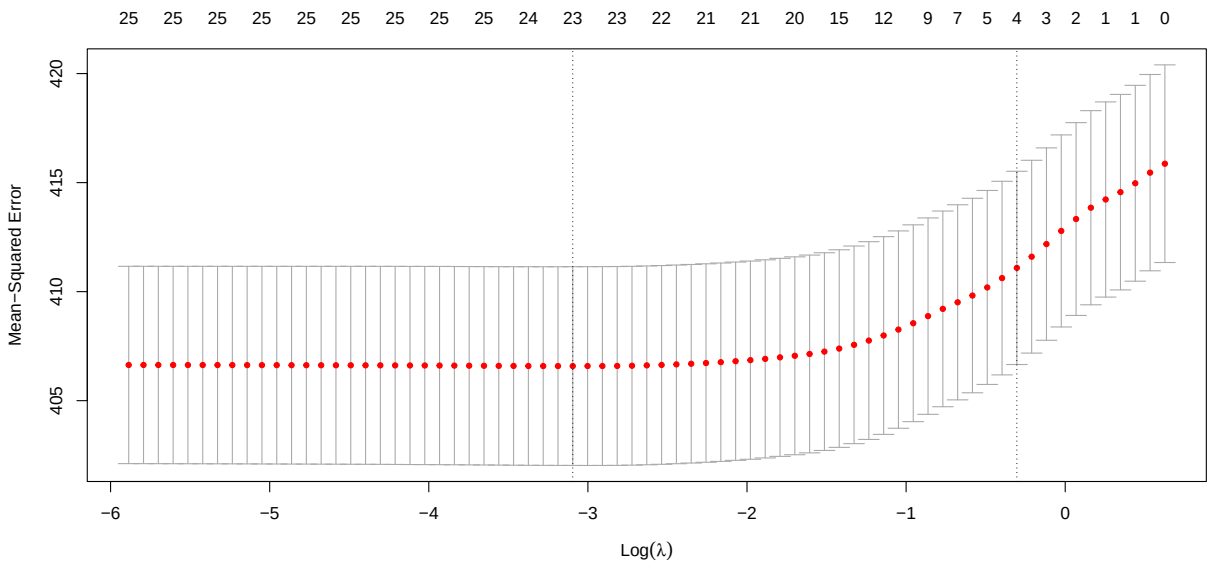
As suspected, performing Ridge Regression did not improve our results, obtaining a RMSE of 20.2482328. Now, we inspect whether the choice of variables for our model poses the problem.

3.3 LASSO

We saw that ridge regression with a wise choice of lambda didn't really outperform least squares. Now, we're interested in whether the LASSO can yield either a more accurate and a more interpretable model. In order to fit a lasso model, we again use the glmnet() function: however, this time we use the argument alpha=1. This type of regression analysis is a shrinkage and variable selection method for linear regression models. The goal of LASSO regression is to obtain the subset of predictors that minimizes prediction error for a quantitative response variable. LASSO does this by imposing a constraint on the model parameters that causes regression coefficients for some variables to shrink toward zero: a penalty term called L1-norm, which is the sum of the absolute coefficients $\lambda \sum_{j=1}^{p} |\beta_j|$ where λ is a constant as in Ridge Regression. Variables with a regression coefficient equal to zero after the shrinkage process are excluded from the model. Variables with non-zero regression coefficients variables are most strongly associated with the response variable.

To perform LASSO, we use the same training and testing set defined previously. We fit the model model using 5-fold Cross Validation to find the optimal value for λ.

Similarly to Ridge Regression, we see that with lambda very large, we have a constant error and there is no change in the plot. As more regressors are pushed to zero, the MSE in the validation set starts to incerase. As we decrease the lambda parameter, i.e. set it to -3 can see a constant MSE decrease.



We see that with an optimal value for lambda of 0.0452786, LASSO has discarded most variables:

```
## 27 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)      49.1612700
## song_duration     .
## acousticness      .
## danceability      2.3675908
## energy            .
## instrumentalness -5.0914759
## keyC#             0.1122503
## keyD              .
## keyD#             .
## keyE              .
## keyF              .
## keyF#             .
```

```
## keyG              .
## keyG#             .
## keyA              .
## keyA#             .
## keyB              .
## liveness          .
## loudness          .
## audio_modeminor   .
## speechiness       .
## tempo             .
## time_signature1   .
## time_signature3   .
## time_signature4   .
## time_signature5   .
## audio_valence    -2.9435239
```

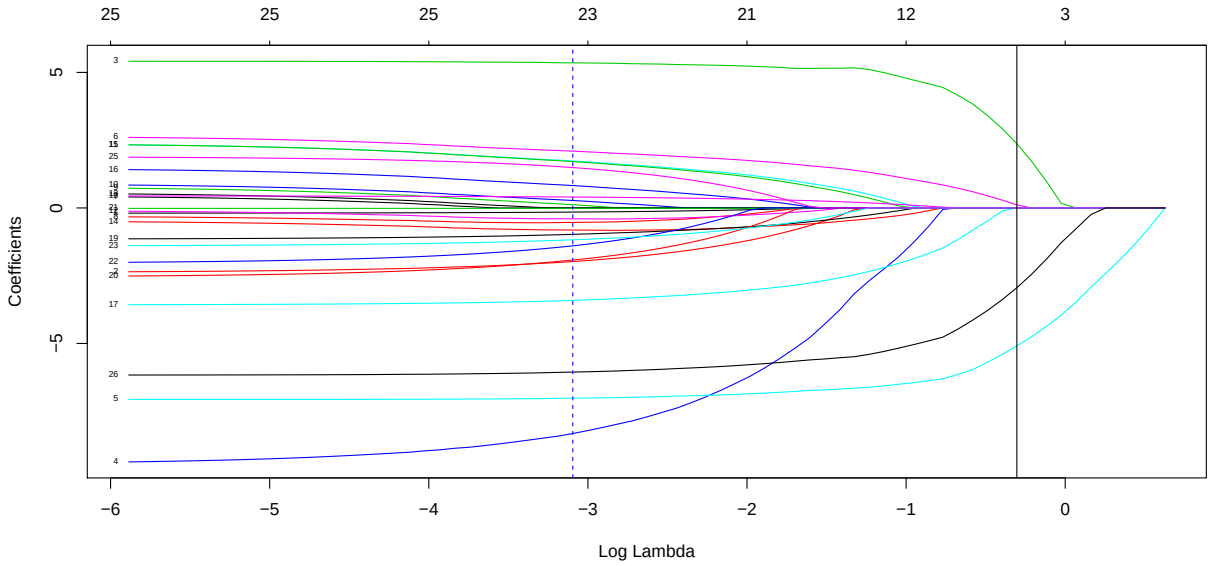
And it has left us the following to work with, which in comparison to our reduced model, does not include most of the predictors:

```
##               name coefficient
## 1      (Intercept)  49.1612700
## 2     danceability   2.3675908
## 4            keyC#   0.1122503
## 5    audio_valence  -2.9435239
## 3 instrumentalness  -5.0914759
```

hence, the minimum value of the RMSE is reached for the following model:

$$popularity = \beta_0 \ + \beta_1 dacenability \ + \beta_2 instrumentalness \ + \beta_3 key_{Csharp} \ + \beta_4 valence$$

We see that what makes a song popular, according to LASSO, is people's perception of danceability and valence. As mentioned previously, songs that are more danceable, such as club songs which we sing to, end up being more popular than the ones containing no vocals. We see LASSO considers whether a song has an overall key of C# as an important factor to our analysis. By plotting the solid black line of ridge regression for lambda within 1 standard error, we seem to get the least squares estimates, and by plotting the optimal vlaue for lambda in a blue dashed line, we get the estimates used to fit our LASSO model.



Producing a RMSE of 20.2054577, we conclude that neither penalty term improves our model.
The final results from our deterministic approaches are the following:

```
##                   Root Mean Squared Error
## Linear Model                     20.18051
## Ridge Regression                 20.24823
## LASSO                            20.20546
```

3.4 Bayesian Approach

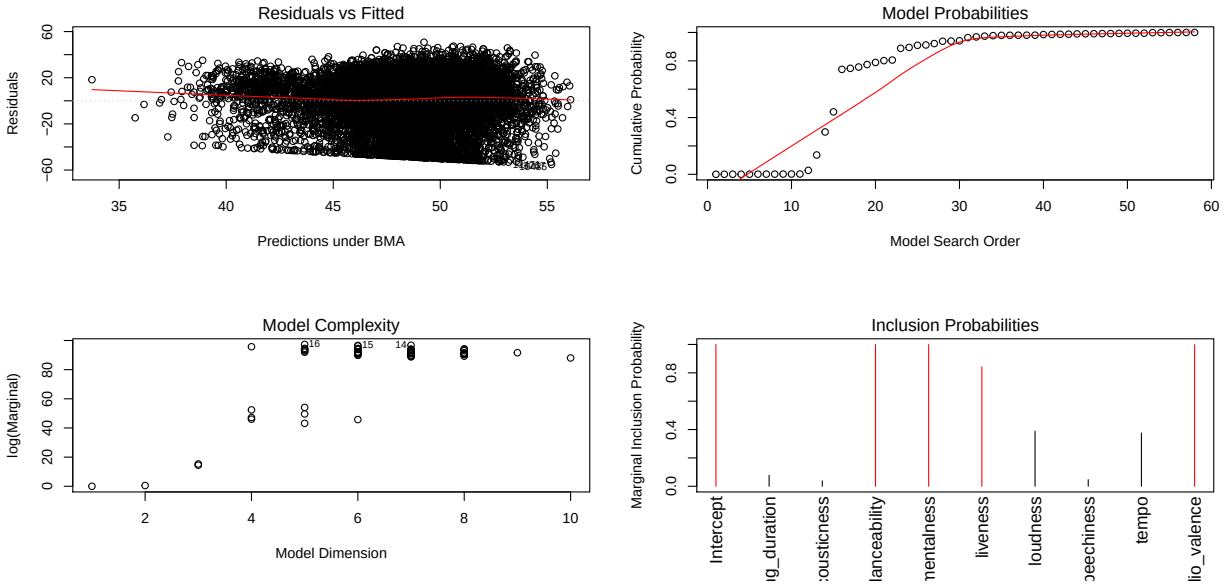
But, what if we don't apply the traditional frequenst statistics to solve a problem? Here we show another approach which is more intuitive and close to how we think about probability in everyday life and yet is a very powerful tool: Bayesian statistics, which has its foundations on conditional probability and Bayes theorem. There is a key element when we want to build a model under Bayesian approach: the Bayes factor - the ratio of the likelihood probability of two competing hypotheses (usually null and alternative hypothesis) and it helps us to quantify the support of a model over another one. In Bayesian modelling, the choice of prior distribution is a key component of the analysis and can modify the results; however, the prior starts to lose weight when we add more data. We use the "BAS" package to conduct a Bayesian regression, using Bayesian Model Averaging (BMA) that provides a mechanism for accounting model uncertainty, and we indicate following parameters:

- **Prior:** Zellner-Siow Cauchy that uses a Cauchy distribution extended for multivariate cases. This is an approximation to the Jeffreys-Zellner-Siow prior that uses the Jeffreys prior on sigma and the Zellner-Siow Cauchy prior on the coefficients.
- **Model prior:** Uniform as a prior distribution on models, to assign equal probabilities to all.
- **Method:** Markov Chain Monte Carlo (MCMC) to improve model search efficiency.

We can see the top 5 models with zero-one indicators for variable inclusion, displayed with a column with the Bayes factor for each model to the highest probability model (BF), the posterior probabilities of the models (PostProbs), the R2 of the models, their dimension (dim) and the log marginal likelihood (logmarg) under the selected prior distribution.

```
##                  P(B != 0 | Y)  model 1    model 2    model 3   model 4
## Intercept           1.00000000  1.00000  1.0000000  1.0000000  1.000000
## song_duration       0.07685547  0.00000  0.0000000  0.0000000  0.000000
## acousticness        0.03847656  0.00000  0.0000000  0.0000000  0.000000
## danceability        0.99931641  1.00000  1.0000000  1.0000000  1.000000
## instrumentalness    0.99960938  1.00000  1.0000000  1.0000000  1.000000
## liveness            0.84130859  1.00000  1.0000000  1.0000000  1.000000
## loudness            0.38935547  0.00000  1.0000000  1.0000000  0.000000
## speechiness         0.04697266  0.00000  0.0000000  0.0000000  0.000000
## tempo               0.37539062  0.00000  1.0000000  0.0000000  1.000000
## audio_valence       0.99833984  1.00000  1.0000000  1.0000000  1.000000
## BF                          NA  1.00000  0.5806098  0.4988002  0.356186
## PostProbs                   NA  0.29960  0.1623000  0.1417000  0.108600
## R2                          NA  0.01910  0.0203000  0.0197000  0.019600
## dim                         NA  5.00000  7.0000000  6.0000000  6.000000
## logmarg                     NA 97.28108 96.7374079 96.5855345 96.248782
##                     model 5
## Intercept          1.000000
## song_duration      0.000000
## acousticness       0.000000
## danceability       1.000000
## instrumentalness   1.000000
## liveness           0.000000
## loudness           0.000000
## speechiness        0.000000
## tempo              0.000000
## audio_valence      1.000000
## BF                 0.204734
## PostProbs          0.082700
## R2                 0.018200
## dim                4.000000
## logmarg           95.695041
```

We further analyze the residuals and see how they differ from the regular multivariate linear regression we observed previously.



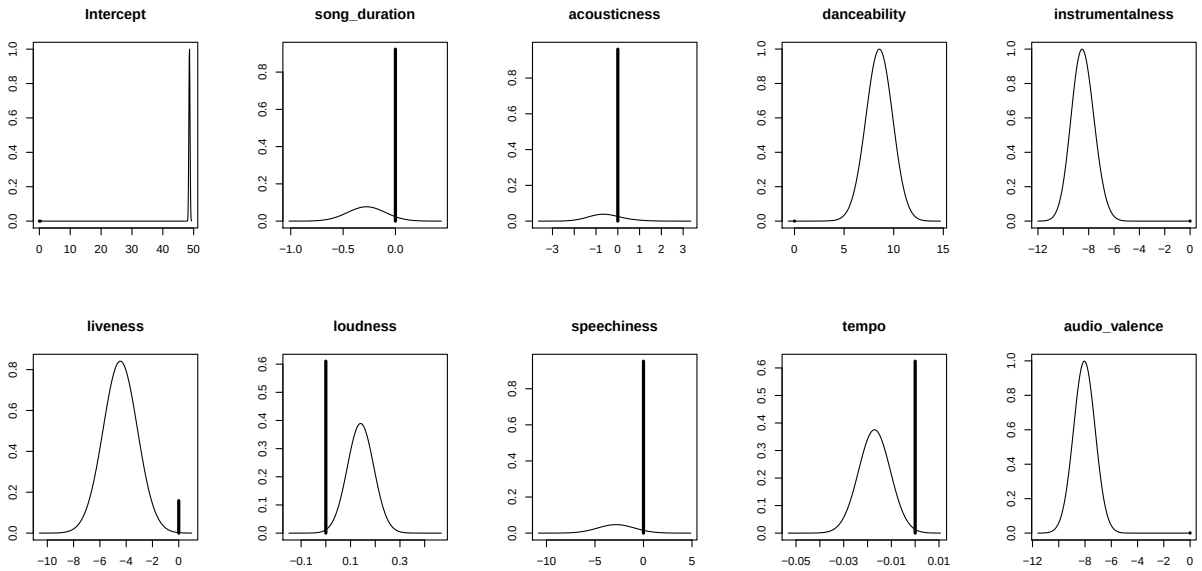
- **Residuals vs Fitted**
  There isn't a clear pattern in the residuals and they are symmetrically distributed. Heteroscedasticity has been confirmed yet again.

- **Model Probabilities**
  This plot displays the cumulative probability of the models in the order they are sampled. This shows that the cumulative probability starts to level off after 20 model trials as each additional models adds only a small increment to the cumulative probability. The model search stops at 50-something, instead of enumerations of 2^9 combinations.

- **Model Complexity**
  This plot shows the dimension of each model, that is the number of regression coefficients including the intercept versus the log of the marginal likelihood of the model. In this case, we can see that highest log marginal can be reached from 5 to 8 dimensions.

- **Inclusion Probabilities**
  We can observe the marginal posterior inclusion probabilities for each of the covariates, with marginal posterior inclusion probabilities that are greater than 0.5 shown in red (important variables for explaining the data and prediction). In the graph, we can see that the most important predictors include those obtained with LASSO as well as liveness.

We obtain the coefficient estimates and standard deviations to be able to examine the marginal distributions for the significant predictors:



The vertical line corresponds to the posterior probability that the coefficient equals to 0. On the other hand, the shaped curve shows the density of possible values where the coefficient is non-zero. It is worthy to mention that the height of the line is scaled to its probability. This implies that the intercept, and danceability, instrumentalness and audio valence show no line denoting non-zero probability. We can also obtain 95% confidence intervals:

```
##                        2.5%      97.5%         beta
## Intercept         48.2636449 49.0031400 48.634612163
## song_duration     -0.2365750  0.0000000 -0.021231921
## acousticness       0.0000000  0.0000000 -0.021097058
## danceability       5.9008880 11.2809382  8.529230901
## instrumentalness -10.1854154 -6.6446821 -8.449723064
## liveness          -6.4968901  0.0000000 -3.743749317
## loudness           0.0000000  0.2048558  0.054801704
## speechiness        0.0000000  0.0000000 -0.132791336
## tempo             -0.0247896  0.0000000 -0.006417062
## audio_valence     -9.6884150 -6.3850803 -8.036542269
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

By fitting the model to the test data, we are able to obtain a RMSE of 20.1400362, which is the lowest but does not differ much from the ones obtained previously.

3.5 Results and Discussion

```
##                   Root Mean Squared Error
## Linear Model                     20.18051
## Ridge Regression                 20.24823
## LASSO                            20.20546
## Bayes Regression                 20.14004
```

What we can conclude is that with an increase in the value of danceability, a song gains in popularity. This has been confirmed by all models. As has the fact that if a song is more instrumental it gains less popularity. What perplexed us was the fact that valence had a negative effect on our target variable. This means that the more happy a song sounds, the less popularity score it will obtain. This might be due to the fact that music features gradual changes along these dimensions (song attributes) within a broader emotion category remaining the same throughout the musical segment. Sometimes, listeners do not even perceive any discrete emotion in the music. They merely perceive a certain level of arousal. What may be of greater impact, according to LASSO, is whether or not the songs has an overall key C#. This key gives us a strange, but pleasant feeling, sometimes uneasy but freeing. We see some of the songs in this musical key:

```
##                                          song_name       artist_name
## 9                                    Mr. Brightside       The Killers
## 18                                      Best of You      Foo Fighters
## 28                                   American Idiot         Green Day
## 48                                      Miss Murder               AFI
## 56                                          The Red          Chevelle
## 62  Taki Taki (with Selena Gomez, Ozuna & Cardi B)          DJ Snake
## 63                                       White Boi    Dillon Francis
## 73                     Banana Clip - Spanish Version            Miguel
## 95                             DÃganle - Tainy Remix      Leslie Grace
## 96                                   Sangria Wine Pharrell Williams
## 100                         Scooby Doo Pa Pa (Remix)           Dj Kass
## 102    1, 2, 3 (feat. Jason Derulo & De La Ghetto)       Sofia Reyes
## 103                              Pa Mi (with Ozuna)        Tory Lanez
## 116                                    Belong To U        Fancy Cars
## 128                                Hey, Soul Sister             Train
## 134                                   We Can't Stop       Miley Cyrus
## 151                                         No One       Alicia Keys
## 155                Single Ladies (Put a Ring on It)          BeyoncÃ©
## 157                                   Lose My Breath   Destiny's Child
## 160                                        Die Young             Kesha
```

What is certain is that even regularizing our linear regression fit did not yield better results, not even when the statistical analysis is undertaken within the context of Bayesian inference. This gives us the motivation to try predicting popularity from another angle. We further conduct an analysis on how correctly a model can classify songs to highly popular and not so popular, i.e. having a popularity score > 75 or not.

4. Classification Models

The following models will be evaluated using several metrics:

- **Accuracy** tells us the proportion of observation the model has classified correctly, i.e. 1-Error Rate.
- **True Positive Rate or Recall** showing percentage of relevant instances that are retrieved i.e. what proportion of songs that are actually with a high popularity score were diagnosed as such.
- **Precision:** a measure that tells us what songs classified as highly popular are actually that popular.

These are obtained from a confusion matrix that depicts all possible outcomes from the model:

- A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class.

- A **false positive** is an outcome where the model incorrectly predicts the positive class. And a **false negative** is an outcome where the model incorrectly predicts the negative class.

4.1 Logistic Regression

We now use logistic regression to build a model that predicts the probability a song is highly popular among users (i.e. has a popularity score above 75). We will begin by transforming the popularity score into a factor of two levels: high and low. After re-ordering its levels for an ease of interpretation, we fit our models to predict the probability of a song being very popular: $P(Y = high\ popularity \mid X)$.

```
##      high
## low     0
## high    1
```

We build our null model - where the intercept is now the log odds of "success", estimated without reference to any predictors. This means that none of the songs' attributes have an effect to its popularity. We continue using all variables for the second, saturated, model but as we did in the linear regression case, we need to check whether some of the predictors have a high VIF value.

```
##                      GVIF Df GVIF^(1/(2*Df))
## song_duration    1.028174  1        1.013989
## acousticness     1.600873  1        1.265256
## danceability     1.427163  1        1.194639
## energy           3.053233  1        1.747350
## instrumentalness 1.017338  1        1.008632
## key              1.176742 11        1.007425
## liveness         1.055728  1        1.027486
## loudness         2.144365  1        1.464365
## audio_mode       1.124579  1        1.060462
## speechiness      1.141510  1        1.068415
## tempo            1.062310  1        1.030684
## time_signature   1.103340  4        1.012369
## audio_valence    1.322228  1        1.149882
```

We observe that energy does not have a high VIF value as it did in the previous case, so we do not exclude it from the model. By looking at the summary of this model:

```
##
## Call:
## glm(formula = song_popularity ~ ., family = "binomial", data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0960  -0.4229  -0.3308  -0.2102   4.3906
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.191e+00  1.699e+02  -0.054   0.9569
## song_duration     6.305e-02  3.633e-02   1.735   0.0827 .
## acousticness     -9.296e-01  1.803e-01  -5.155 2.54e-07 ***
## danceability      1.709e+00  2.694e-01   6.343 2.25e-10 ***
## energy           -2.630e+00  3.191e-01  -8.242  < 2e-16 ***
```

```
## instrumentalness -4.044e+00  5.853e-01  -6.909 4.89e-12 ***
## keyC#             2.359e-01  1.340e-01   1.760   0.0784 .
## keyD             -1.888e-01  1.584e-01  -1.192   0.2333
## keyD#            -6.654e-02  2.455e-01  -0.271   0.7864
## keyE              4.928e-02  1.648e-01   0.299   0.7648
## keyF             -7.750e-02  1.585e-01  -0.489   0.6248
## keyF#            -1.658e-02  1.625e-01  -0.102   0.9187
## keyG             -1.199e-01  1.478e-01  -0.812   0.4170
## keyG#            -2.018e-01  1.678e-01  -1.203   0.2292
## keyA             -1.242e-01  1.562e-01  -0.795   0.4265
## keyA#             1.116e-01  1.618e-01   0.690   0.4901
## keyB              1.429e-01  1.504e-01   0.950   0.3421
## liveness         -2.636e-01  2.548e-01  -1.034   0.3010
## loudness          2.284e-01  1.954e-02  11.689  < 2e-16 ***
## audio_modeminor   4.585e-02  7.287e-02   0.629   0.5292
## speechiness      -3.684e-01  3.519e-01  -1.047   0.2951
## tempo             2.661e-04  1.267e-03   0.210   0.8336
## time_signature1   9.757e+00  1.699e+02   0.057   0.9542
## time_signature3   9.172e+00  1.699e+02   0.054   0.9569
## time_signature4   9.257e+00  1.699e+02   0.054   0.9566
## time_signature5   9.253e+00  1.699e+02   0.054   0.9566
## audio_valence    -7.638e-01  1.641e-01  -4.655 3.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7237.2  on 14925  degrees of freedom
## Residual deviance: 6662.0  on 14899  degrees of freedom
## AIC: 6716
##
## Number of Fisher Scoring iterations: 11
```

we see that using a logistic regression, it gives us the estimate, standard errors, z-score and p-values on each of the coefficients. In comparison with the full linear regression model, we can see that different predictors are significant. Take for instance the predictors where the p-value < 0.001:

- acousticness
- danceability
- energy
- instrumentalness
- loudness
- audio valence

which are different from the regression fit, excluding danceability and instrumentalness.
The estimates now give the change in the log odds of the outcome for a one unit increase in the predictor variable:

- For every one unit change in song duration, the log odds of a song being popular increase by 0.06305;
- For every one unit change in acousticness, the log odds of a song being popular decrease by 0.9296;
- For every one unit change in danceability, the log odds of a song being popular increase by 1.709; etc.

We also see two forms of deviance - the null deviance and the residual deviance. Deviance is a measure of goodness of fit of a generalized linear model. The null deviance shows how well the response variable is predicted by a model that includes only the intercept and the residual deviance reflects the saturated model, and it has reduced by 575.2 points on 14899 degrees of freedom. This is equivalent to comparing a pair of nested models where the null model is given by H_0 : There is no relationship between the X variables and the Y variable, i.e. the predictions are no closer to the actual Y values than you would expect by chance; with the full model under H_1 : All X variables relate with Y. The results from our hypothesis testing tells us that the predictors should not be removed from the model since Pr(>Chi) is very small (<2.2e-16):

```
## Analysis of Deviance Table
##
## Model 1: song_popularity ~ +1
## Model 2: song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + time_signature + audio_valence
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     14925     7237.2
## 2     14899     6662.0 26   575.15 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The logit function is the natural log of the odds that the response variable (in our case - song popularity) equals one of the categories (high popularity vs low popularity). The type="response" option tells R to output probabilities of the form $P(Y = 1 \mid X)$., as opposed to other information such as the logit. These values correspond to the probability of the song being very popular. The first 6 probabilities are shown:

```
##          1          2          3          4          5          6
## 0.10990690 0.05629236 0.02755615 0.05524157 0.08086982 0.06827497
```

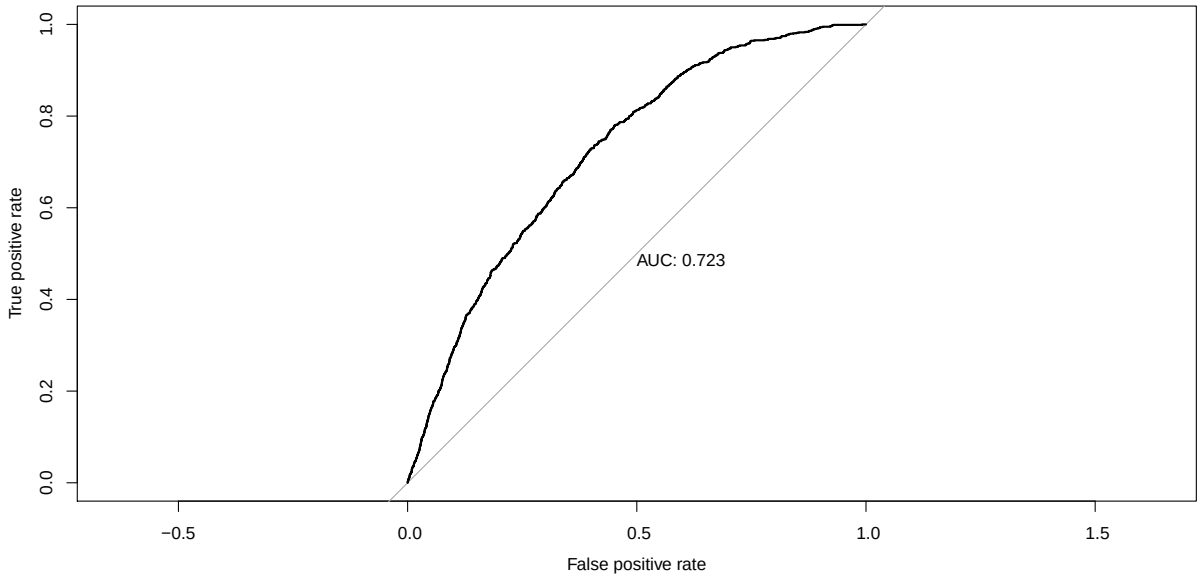
We further note how many observations were correctly or incorrectly classified provided that a highly popular song has a probability above 0.5:

```
##       glm_pred
##          low
##   low  13945
##   high   981
```

- **Accuracy:** 93.43%
- **True Positive Rate:** 0%
- **Specificity:** 100%
- **Precision:** 0%

To improve these results, we can construct a ROC curve that will show us the optimal threshold for our data to help deal with the trade-off between TPR (sensitivity) and specificity (1-FPR). Ideally it would be closer to the top-left corner and indicate good performance. This is effective for imbalanced binary classification, as it focuses on the minority class. We calculate the area under the ROC curve - AUC, which is equivalent to the probability that a randomly chosen positive instance is ranked higher than a randomly

chosen negative instance.




```
##   threshold specificity sensitivity
##  0.06801907  0.60071710  0.72884811
```

So we can see that our model's predictions are 72.3% correct, and we lower the threshold to get the best trade-off and obtain the following results:

```
##       glm_pred_th
##        high  low
##   low  5574 8371
##   high  715  266
```

- **Accuracy:** 60.87%
- **True Positive Rate:** 72.88%
- **Specificity:** 60.03%
- **Precision:** 11.37%

which when we look at for the first time may seem worse (if we focus too much on the error rate), but precision has jumped to 11.37%, similar to how TPR and FPR have increased.

As observed in the linear regression analysis, many of the variables used are not associated with the response (i.e. they are non-significant) and including such irrelevant variables leads to unnecessary complexity in the resulting model. Again, we perform a feature selection using backward selection:

```
## Start:  AIC=6716.04
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + time_signature + audio_valence
##
```

```
##                    Df Deviance    AIC
## - time_signature    4   6663.3 6709.3
## - key              11   6679.2 6711.2
## - tempo             1   6662.1 6714.1
## - audio_mode        1   6662.4 6714.4
## - liveness          1   6663.1 6715.1
## - speechiness       1   6663.2 6715.2
## <none>                  6662.0 6716.0
## - song_duration     1   6664.8 6716.8
## - audio_valence     1   6683.7 6735.7
## - acousticness      1   6689.6 6741.6
## - danceability      1   6703.3 6755.3
## - energy            1   6730.3 6782.3
## - instrumentalness  1   6785.7 6837.7
## - loudness          1   6813.4 6865.4
##
## Step:  AIC=6709.27
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + audio_valence
##
##                    Df Deviance    AIC
## - key              11   6680.5 6704.5
## - tempo             1   6663.3 6707.3
## - audio_mode        1   6663.7 6707.7
## - speechiness       1   6664.4 6708.4
## - liveness          1   6664.4 6708.4
## <none>                  6663.3 6709.3
## - song_duration     1   6666.0 6710.0
## - audio_valence     1   6685.1 6729.1
## - acousticness      1   6691.0 6735.0
## - danceability      1   6706.1 6750.1
## - energy            1   6731.8 6775.8
## - instrumentalness  1   6786.9 6830.9
## - loudness          1   6814.7 6858.7
##
## Step:  AIC=6704.47
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + liveness + loudness + audio_mode +
##     speechiness + tempo + audio_valence
##
##                    Df Deviance    AIC
## - tempo             1   6680.5 6702.5
## - speechiness       1   6681.2 6703.2
## - liveness          1   6681.5 6703.5
## - audio_mode        1   6681.5 6703.5
## <none>                  6680.5 6704.5
## - song_duration     1   6683.2 6705.2
## - audio_valence     1   6703.5 6725.5
## - acousticness      1   6710.1 6732.1
## - danceability      1   6728.2 6750.2
## - energy            1   6750.1 6772.1
## - instrumentalness  1   6804.1 6826.1
## - loudness          1   6834.0 6856.0
```

```
##
## Step:  AIC=6702.5
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + liveness + loudness + audio_mode +
##     speechiness + audio_valence
##
##                    Df Deviance    AIC
## - speechiness       1   6681.2 6701.2
## - liveness          1   6681.5 6701.5
## - audio_mode        1   6681.6 6701.6
## <none>                  6680.5 6702.5
## - song_duration     1   6683.2 6703.2
## - audio_valence     1   6703.5 6723.5
## - acousticness      1   6710.3 6730.3
## - danceability      1   6729.1 6749.1
## - energy            1   6750.1 6770.1
## - instrumentalness  1   6804.1 6824.1
## - loudness          1   6834.1 6854.1
##
## Step:  AIC=6701.22
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + liveness + loudness + audio_mode +
##     audio_valence
##
##                    Df Deviance    AIC
## - audio_mode        1   6682.2 6700.2
## - liveness          1   6682.5 6700.5
## <none>                  6681.2 6701.2
## - song_duration     1   6684.0 6702.0
## - audio_valence     1   6703.7 6721.7
## - acousticness      1   6710.9 6728.9
## - danceability      1   6729.8 6747.8
## - energy            1   6751.9 6769.9
## - instrumentalness  1   6804.1 6822.1
## - loudness          1   6836.4 6854.4
##
## Step:  AIC=6700.15
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + liveness + loudness + audio_valence
##
##                    Df Deviance    AIC
## - liveness          1   6683.4 6699.4
## <none>                  6682.2 6700.2
## - song_duration     1   6685.1 6701.1
## - audio_valence     1   6704.8 6720.8
## - acousticness      1   6711.8 6727.8
## - danceability      1   6732.5 6748.5
## - energy            1   6752.4 6768.4
## - instrumentalness  1   6804.7 6820.7
## - loudness          1   6837.8 6853.8
##
## Step:  AIC=6699.4
## song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + loudness + audio_valence
```

```
##
##                     Df Deviance    AIC
## <none>                   6683.4 6699.4
## - song_duration      1   6686.3 6700.3
## - audio_valence      1   6705.8 6719.8
## - acousticness       1   6713.3 6727.3
## - danceability       1   6735.1 6749.1
## - energy             1   6757.2 6771.2
## - instrumentalness   1   6805.7 6819.7
## - loudness           1   6840.2 6854.2
```

and accordingly the model that would, in theory, maximize predictive capability is:

$$popularity = \beta_0 \ + \beta_1 duration \ + \beta_2 valence \ + \beta_3 acousticness \ + \beta_4 danceability \ + \beta_5 energy + \beta_6 instrumentalness \ + \beta_7 loudness$$

```
##
## Call:
## glm(formula = song_popularity ~ song_duration + audio_valence +
##     acousticness + danceability + energy + instrumentalness +
##     loudness, family = "binomial", data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0493  -0.4251  -0.3342  -0.2138   4.3735
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.02109    0.38900   0.054   0.9568
## song_duration      0.06382    0.03600   1.773   0.0762 .
## audio_valence     -0.77010    0.16291  -4.727 2.28e-06 ***
## acousticness      -0.95744    0.17853  -5.363 8.19e-08 ***
## danceability       1.78075    0.25079   7.101 1.24e-12 ***
## energy            -2.67344    0.31202  -8.568  < 2e-16 ***
## instrumentalness -3.99865    0.58084  -6.884 5.81e-12 ***
## loudness           0.23012    0.01937  11.879  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7237.2  on 14925  degrees of freedom
## Residual deviance: 6683.4  on 14918  degrees of freedom
## AIC: 6699.4
##
## Number of Fisher Scoring iterations: 8
```

Comparing the full and the reduced model obtained by backward selection gives us a result of Pr(>Chi)=0.3171 so we can safely remove some of the predictors.

```
## Analysis of Deviance Table
##
## Model 1: song_popularity ~ song_duration + audio_valence + acousticness +
```

```
##     danceability + energy + instrumentalness + loudness
## Model 2: song_popularity ~ song_duration + acousticness + danceability +
##     energy + instrumentalness + key + liveness + loudness + audio_mode +
##     speechiness + tempo + time_signature + audio_valence
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     14918     6683.4
## 2     14899     6662.0 19   21.362   0.3171
```

Not to run the risk of over-fitting, we consider a train-validation (test) set approach.

```
##
## Call:
## glm(formula = song_popularity ~ song_duration + audio_valence +
##     acousticness + danceability + energy + instrumentalness +
##     loudness, family = "binomial", data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9470  -0.4239  -0.3336  -0.2158   4.3139
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.18136    0.42963  -0.422  0.67292
## song_duration     0.08582    0.03839   2.236  0.02537 *
## audio_valence    -0.90063    0.18232  -4.940 7.81e-07 ***
## acousticness     -0.75050    0.19707  -3.808  0.00014 ***
## danceability      1.87324    0.28022   6.685 2.31e-11 ***
## energy           -2.53561    0.34785  -7.289 3.11e-13 ***
## instrumentalness -3.82263    0.61226  -6.243 4.28e-10 ***
## loudness          0.23127    0.02160  10.709  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5790.8  on 11940  degrees of freedom
## Residual deviance: 5354.7  on 11933  degrees of freedom
## AIC: 5370.7
##
## Number of Fisher Scoring iterations: 8
```

As mentioned previously, the estimates give the change in the log odds of the outcome for a one unit increase in the predictor variable:

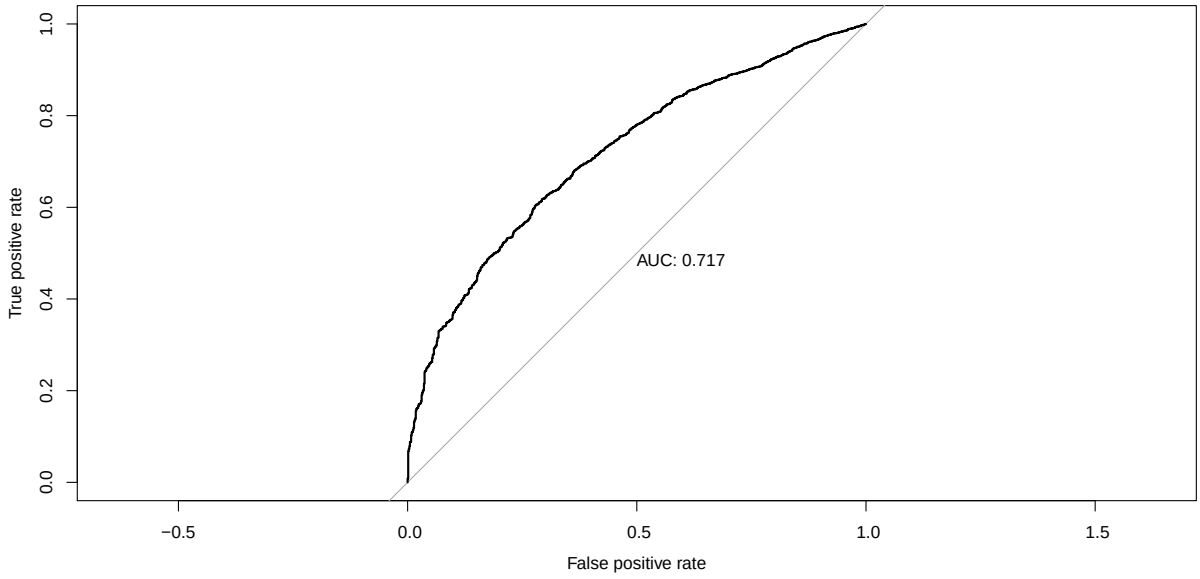
- For every one unit change in song duration, the log odds of a song being popular increases by 0.08;
- For every one unit change in valence, the log odds of a song being popular decreases by 0.9;
- For every one unit change in acousticness, the log odds of a song being popular decreases by 0.75;
- For every one unit change in danceability, the log odds of a song being popular increases by 1.87;
- For every one unit change in energy, the log odds of a song being popular decreases by 2.53;

- For every one unit change in instrumentalness, the log odds of a song being popular decreases by 3.82;
- For every one unit change in loudness, the log odds of a song being popular increases by 0.23.

Since these coefficients can never be known for sure, we construct 95% confidence intervals in terms of their effect on the logg odds of a song being popular:

```
##                       2.5 %      97.5 %
## (Intercept)      0.359482462  1.93826954
## song_duration    1.007119619  1.17305811
## audio_valence    0.284098701  0.58063451
## acousticness     0.319900597  0.69279170
## danceability     3.767588602 11.30269482
## energy           0.040027017  0.15653507
## instrumentalness 0.005713239  0.06429316
## loudness         1.208426705  1.31519304
```

The followig are the **results of the model fitted on the training set**, where judging by AUC the model predicts correctly 71.7% of the data:

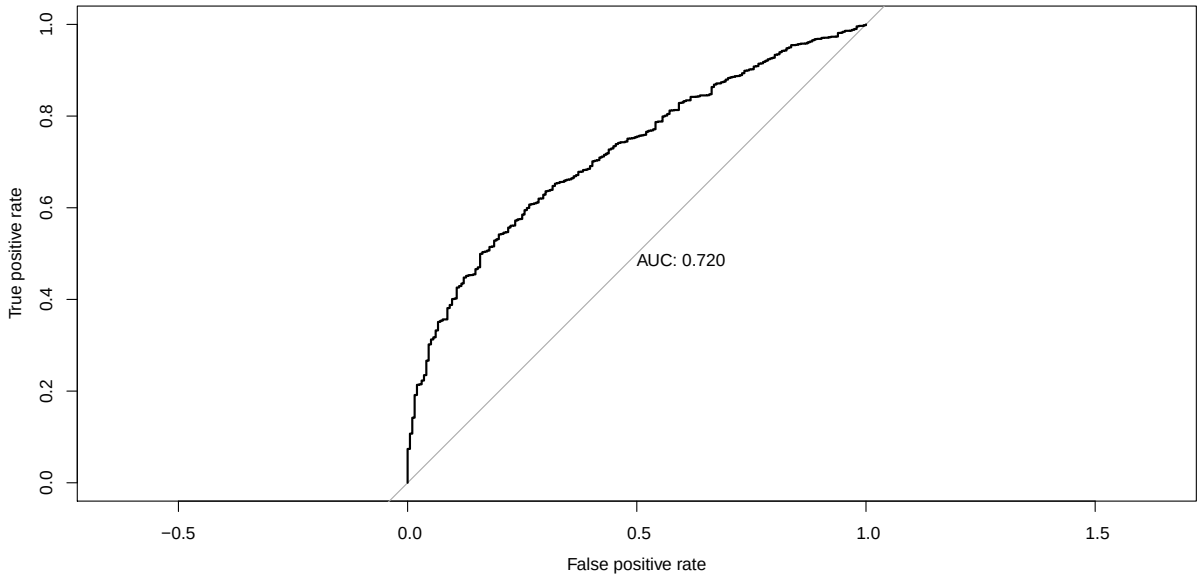



```
##   threshold specificity sensitivity
##  0.06911304  0.71974522  0.60586232
```

```
##       train_pred
##        high  low
##   low  4398 6758
##   high  565  220
```

- **Training accuracy:** 61.33%
- **True Positive Rate:** 71.97%
- **Precision:** 11.38%

And respectively, the **results of the model fitted on the test set**:




```
##       test_pred
##        high  low
##   low  1114 1675
##   high  144   52
```

- **Testing accuracy:** 60.94%
- **True Positive Rate:** 73.47%
- **Precision:** 11.45%

We see that the accuracy of the model is 61%. With respect to the training set, this does not differ. The relevant instances that are retrieved is shown by the TPR equaling to73%, a little higher than the one obtained in training set results. Judging also by the ROC curve, we see that the model's predictions are correct 72% of the time by setting the threshold to 0.0691, obtained from fitting the model on the training set.

4.2 Quadratic Discriminant Analysis

We now try to fit a non-linear boundary between classifiers. To do so, we perform a Quadratic Discriminant Analysis on our dataset. The QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction. It assumes that each class has its own covariance matrix.

```
## Call:
## qda(song_popularity ~ song_duration + audio_valence + acousticness +
##     danceability + energy + instrumentalness + loudness, data = train_set)
##
## Prior probabilities of groups:
##        low       high
```

```
## 0.93426011 0.06573989
##
## Group means:
##      song_duration audio_valence acousticness danceability    energy
## low       3.648959     0.5274530    0.2758200    0.6218511 0.6387022
## high      3.683711     0.5036536    0.2003206    0.6707682 0.6615371
##      instrumentalness  loudness
## low        0.09778559 -7.772139
## high       0.01084694 -6.161431
```

The prior probabilities of the group tell us that 93.42% of training data corresponds to a song having popularity score below 75, while only 6.57% of the observations are very popular among users.
QDA also provides us the group means - estimated average of each predictor within each class. These suggest that when songs have a high popularity score, their song duration, danceability, enery and loudness are greater than in songs with low popularity score. We expected valence to be greater too, but we suspect that this might be due to classes being unbalanced... unless people do feel better with pleasing and mellow music, but not too energetic and danceable. On the contrary, songs with low popularity tend to have greater acousticness and instrumentalness values, which does not come as a surprise.

```
##       qda_pred
##         low high
##   low  2645  144
##   high  168   28
```

- **Testing accuracy:** 89.55%
- **True Positive Rate:** 14.29%
- **Precision:** 16.28%

Comparing this to the logistic model, using QDA has yielded us a boost in accuracy to almost 90%. TPR has decreased and precision of returning relevant instances at a rate of 16.28%.

4.3 Naive Bayes Classification

We additionally decide to fit a generative model - Naive Bayes, which is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

```
##       bayes_pred
##         low high
##   low  2330  459
##   high  125   71
```

- **Testing accuracy:** 80.44%
- **True Positive Rate:** 36.22%
- **Precision:** 13.4%

Although accuracy and precision are higher when compared to the logistic classifier, they are lower when compared to QDA. But, TPR has increased to 36.22 with respect to QDA's results, meaning that now a higher rate of correctly classifying songs with high popularity.

4.4 K-Nearest Neighbours

Since we suspect that songs that share similar features have approximately the same popularity score, we perform a K-Nearest Neighbour classification for the test set from training set. Unlike most algorithms, K-NN is a non-parametric model which means that it does not make any assumptions about the dataset. This makes the algorithm more effective since it can handle realistic data. K-NN is a lazy algorithm, as it memorizes the training data set instead of learning a discriminative function from the training data. For each observation of the validation set, the K nearest (in Euclidean distance) validation set vectors are found, and the classification is decided by majority vote. In our case, we will fit the model using a value of one for K, meaning that it will classify a song based on most similar features with another already classified observation in the training set. To perform this kind of modeling, we have to use the original numerical predictors we had in the dataset, prior to modifications.

```
##       knn_pred
##         low high
##   low  2617  172
##   high  172   24
```

When setting K=1, the model's accuracy is similar to QDA - 88.48%, which is far better than what we got using logistic regression. Precision is now 12.24% which in comparison to QDA now is not much, but higher than logistic regression. As with QDA and Bayes, using KNN we get better results than the logistic regression model.

4.5 Results and Discussion

```
##              Logistic Regression   QDA Bayes  1-NN
## Accuracy                   60.94 89.55 80.44 88.48
## Precision                  11.45 16.28 13.40 12.24
## Recall / TPR               73.47 14.29 36.22 12.24
```

Looking at the results of the classifiers, we can say with certainty that QDA performed the best in term of accuracy with a rate of 89.55%. We get a similar accuracy using 1-NN - 88.48%. Since this is a non-parametric approach, we did expect it to outperform logistic regression, but we suspect that setting K=1 induces high variance and an *overly* flexible decision boundary. Since we have a sufficient number of training examples, and the variance of the classifier is not a major concern, QDA serves as a compromise between K-NN and the logistic regression approach. Though not as flexible as KNN, it still yielded better results, including the ones for precision (16.28%) and recall (TPR) (14.29). But, by fitting the Naive Bayes Classification model, we obtained an accuracy rate of somewhere in between these models - 80.44, but a recall (TPR) of 36.22 which outperformed both QDA and 1-NN. This might be due to the naive assumptions, i.e. independence among features, but speed does come at a cost. So, by using QDA, we see that 14.29% of the songs with a popularity score > 80 were diagnosed as such and that 16.28 of the songs classified as having popularity score >80 are actually that popular. Comparing these results with the ones obtained by the logistic classifier, we see that using logistic regression gives a recall rate almost 5 times larger than QDA - 60.94. This comes as a surprise, since it supports only linear solutions, but we suspect that this is due to the classes being unbalanced.

5. Technical Appendix

5.1 Correlation

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. Pearson's correlation coefficient (r) is a measure of the strength of the association between the two variables.

$$r_{xy} = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

5.2 Variance Inflation Factor (VIF)

The variance inflation factor (VIF) quantifies the extent of correlation between one predictor and the other predictors in a model. It is used for diagnosing collinearity/multicollinearity. Higher values signify that it is difficult to impossible to assess accurately the contribution of predictors to a model. It is obtained by:

$$VIF = \frac{1}{1 - R^2}$$

where R^2 provides us a proportion of variance explained by a model.

5.3 Regression Models

A regression model is a model of relatioship between covariates (predictors) and an outcome (Y).

$$Y = f(x) + \epsilon$$

where the observations are realizations of independent random variables whose averages lie on the straight line $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} ... + \beta_p x_{ip}$ i.e.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i$$

5.3.1 Root Mean Square Error

The mean squared error tells us how close a regression line is to a set of points i.e. the difference between the observed x_i and predicted $\hat{x}_i$ value, and we are simply taking its root.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})^2}$$

5.3.2 Estimate of Test Error using Cross Validation

The idea is to randomly divide the data into K equal-sized parts. By leafing out part k, fit the model to the other K_1 parts (combined), and then obtain predictions for the left-out kth part. This is done in turn for each part $k = 1, 2, ..., K$, and then the results are combined.

- Let the K parts be $C_1, C_2, ..., C_K$, where C_k denotes the indices of the observations in part k. There are n_k observations in part k (when n is a multiple of K, $n_k = \frac{n}{K}$).

- Compute

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k$$

where

$$MSE_k = \frac{1}{n_k} \sum_{i \epsilon C_k} (y_i - \hat{y_i})^2$$

and $\hat{y_i}$ is the fitted value for observation i obtained from the data with part k removed.

5.3.3 Comparing Nested Models

H_0 : There is some or no relationship between the X variables and the Y variable. vs the alternative
H_1 : All X variables relate with Y.

This is observed with the F-statistic:

$$F = \frac{\frac{RSS_{red} - RSS_{full}}{p}}{\frac{RSS_{full}}{n-p-1}}$$

where n is the number of predictors for the full model, and p is the number of predictors for the reduced. Under H_0, this statistic follows a $F_{p,n-p-1}$ distribution.

5.3.4 Confidence Intervals for Regression Coefficients

The interval that contains the true value β_i in 95% of all samples is given by:

$$CI^{\beta}_{0.95} = [\hat{\beta} - 1.96SE(\hat{\beta}),\ \hat{\beta} + 1.96SE(\hat{\beta})]$$

5.3.5 Ridge Reression

The ridge regression coefficient estimates are the values that minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 - \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $\lambda \geq 0$ is a *tuning parameter*, that fine-tunes the amoung of penalty and is determined seperately.

5.3.6 LASSO

The LASSO is a relatively recent alternative to rigde regression that also performs a variable selection. The LASSO coefficients minimize the quantity

$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 - \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

5.3.7 Bayes Regression

5.4 Classification Models

5.4.1 Metrics

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate\ (Recall) = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

5.4.2 Logistic Regression

$$log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}$$

such that

$$\pi_i = \frac{e^{(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip})}}{1 + e^{(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip})}}$$

5.4.3 Confidence Intervals for Coefficients

The interval that contains the odds ratio e^{β_i} in 95% of all samples is given by:

$$CI^{\beta}_{0.95} = [e^{\hat{\beta} - 1.96SE(\hat{\beta})},\ e^{\hat{\beta} + 1.96SE(\hat{\beta})}]$$

5.4.4 Comparing Nested Models

H_0 : There is some or no relationship between the X variables and the Y variable. vs the alternative
H_1 : All X variables relate with Y.

This is observed with a deviance difference test, where deviance is equal to

$$Dev(\hat{\beta}, y) = -2 \sum_{i=1}^{n} [y_i log(\hat{\pi}_i) + (1 - y_i) log(1 - \hat{\pi})]$$

and under H_0, the difference $D = Dev_{red} - Dev_{full}$ follows a $\chi^2_{df_{red} - df_{full}}$
where df_{red} are the degrees of freedom for the reduced model, and df_{full} are the degrees of freedom for the full model.

5.4.5 Quadratic Discriminant Analysis

This classifiers results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction. QDA assumes that an observation from the kth class is of the form X $N(\mu_k, \sum_k)$, where $\sum_k$ is a covariance matrixs for the kth class. Under this assumption, the Bayes classifier assigns an observation $X = x$ to the class for which

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k) - \frac{1}{2}log|\sum_k| + log\pi_k$$

is largest. So the QDA classifier involves plugging estimates for $\sum_k$, μ_k, and π_k into the equation, and then assigning an observation $X = x$ to the class for which this quantity is largest. The quantity x appears as a quadratic function.

5.4.6 K - Nearest Neighbours

Given a positive integer K and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by N_0. It then estimates the conditional probability for class k as the fraction of points in N_0 whose response values equal k:

$$P(Y = k \mid X = x_0) = \frac{1}{K}\sum_{i \epsilon N_0} I(y_i = k)$$

Finally, KNN applies Bayes rule and classifies the test observation x_0 to the class with the largest probability.

5.4.7 Naive Bayes Classification

Naive Bayes Classifier calculates the probability of a class given a set of predictor values (i.e $P(Y_i = k \mid x_1, x_2, ..., x_n)$). Inputting ths into Bayes' theorem:

$$P(Y_i \mid x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n \mid Y_i = k)P(Y_i = k)}{P(x_1, x_2, ..., x_n)}$$

where Naive Bayes algorith assume that all predictors are independent of each other and under this assumption

$$P(x_1, x_2, ..., x_n \mid Y_i = k) = P(x_1 \mid Y_i = k)P(x_2 \mid Y_i = k)...P(x_n \mid Y_i = k$$

5.5 Backward Stepwise Selection

Input: A full model with p predictors, M_p. *Output:* Single model with best performance among the ones tested.

1. Let M_p be the full model with p predictors.

2. for $k = p, p-1, ..., 1$:

- Consider all k models that contain all but one predictor in M_k.

- Choose the one having smallest RSS or highest R^2, it is M_{k-1}.

3. Choose a model among the best selected, $M_0, ...M_p$, through a given criterion (cross-validated predicted error, C_p, AIC, BIC or $adjusted R^2$).

This is a heuristic research strategy that provides locally optimal solutions that approach an optimal solution globally in a reasonable amount of time. We use it as a cheaper strategy to find a satisfactory model.

5.5.1 Method for Evaluating and Comparing Models

Akaike's information criterion (AIC) was used to compare models with different number of predictors. This method makes a mathematical adjustment to the training error rate in order to estimate the test error rate.

$$AIC = -2l(\hat{\theta}) + 2d$$

In the case of linear model with Gaussian errors:

$$AIC = -2l(\hat{\beta}, \hat{\theta}) + 2d = n \log\left(\frac{RSS}{n}\right) + 2d$$

We use it in the backward selection. We used AIC to have a selection method capable of storing models with a large number of variables.