Spotify Songs Analysis Assignment

All analysis codes can be found here:
https://github.com/timehacker/spotify-songs-analysis

1.1 How many songs are there in total in the dataset?

```
> length(unique(spotify_songs$track_id))
[1] 28356
```

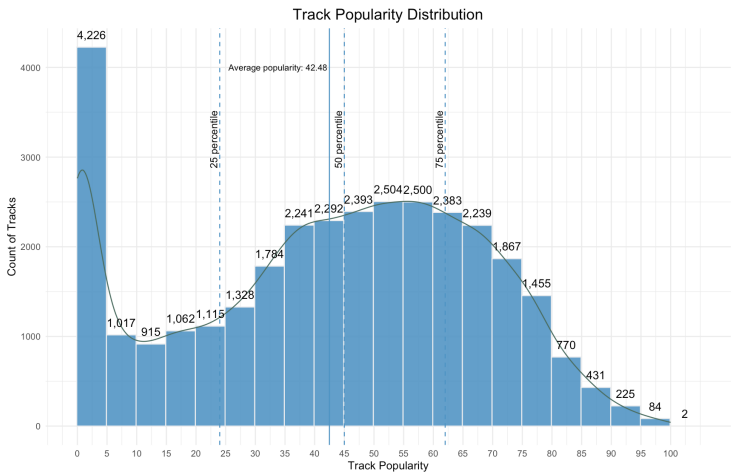
1.2 How many distinct playlists are there in the dataset?

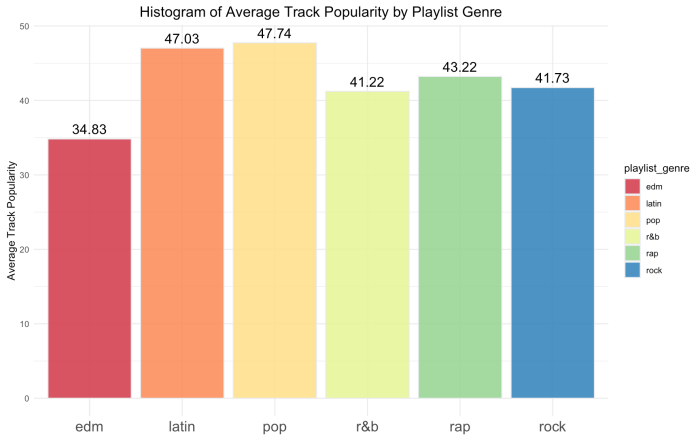
```
> length(unique(spotify_songs$playlist_id))
[1] 471
```

1.3 How many distinct artists are there in the dataset?

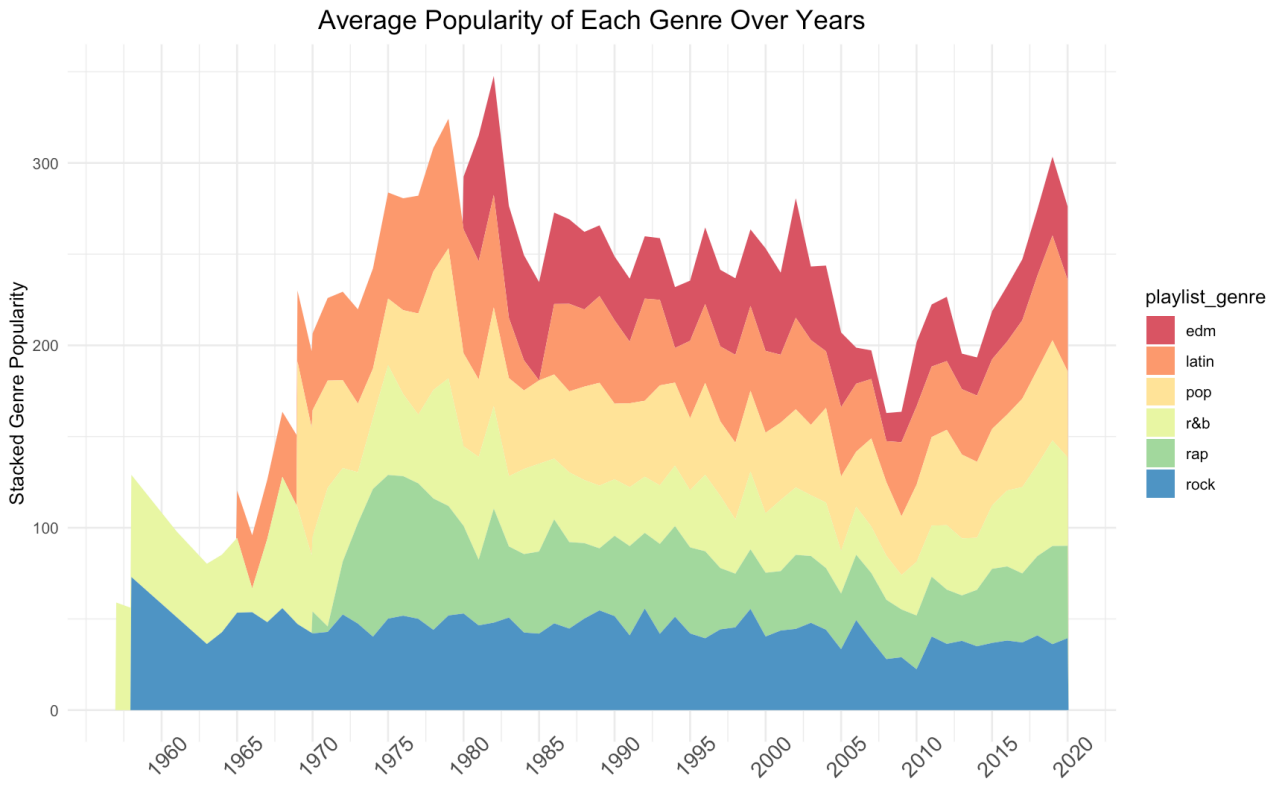
```
> length(unique(spotify_songs$track_artist))
[1] 10693
```

2. Plot a histogram of the overall track popularity and plot a histogram of the average track popularity by playlist genre



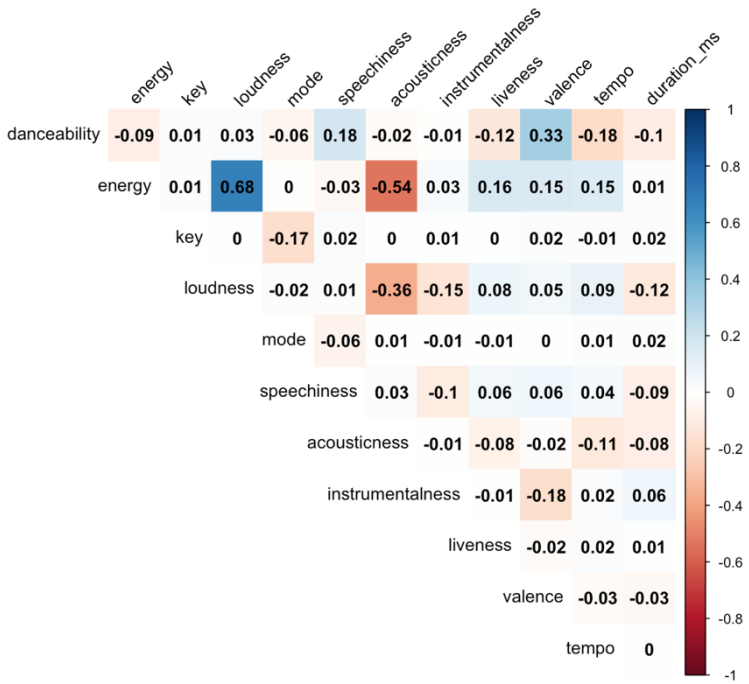


3. Based on track_album_release_date, plot the average popularity of each genre over the years.



4. Identify and discuss the features that make a song more "danceable."

Music common sense tells us that some genres are more danceable than others, so our analysis focuses on the dataset's features with quantitative indicators. Through correlation analysis, we can conclude that:



- **Valence** has a moderate positive correlation(0.3305) with danceability, suggesting that more positive-sounding songs tend to be more danceable
- Followed by **Speechiness**, there's a slight positive correlation(0.1817) with danceability, suggesting that songs with more spoken words might be slightly more danceable.
- It's also notable that for **Tempo**, there's a slight negative correlation (-0.184) with danceability, suggesting that songs with a faster tempo might be slightly less danceable.

The correlation coefficients of the remaining features are relatively low, and at least from the current data set, it is difficult to reveal their impact on danceability. I also tried to build 11 different simple linear regression models based on these features. Although many models were significant(with P-values < 2e-16), the Multiple R-squared of most models was super low. Only the Multiple R-squared of lm(danceability ~ valence, data = spotify_songs) barely reached 0.1092. This once again indicates that Valence can make a song more danceable.

5.1 As an agent of a label company looking for young talents and their songs, identify the top three important features in a song that make it more likely to be popular.

```
> popularity_predict <-
lm(track_popularity~danceability+energy+key+loudness+mode+speechiness+acousticness+instrumentalness+liveness+val
ence+tempo+duration_ms+factor(playlist_genre), data = spotify_songs)
```

I built multiple linear regression models based on the above code. I filter out such features [danceability, energy, loudness, acousticness, instrumentalness, liveness, tempo, duration_ms and different playlist_genre factors] whose P-values indicate statistically significant (< 2e-16).

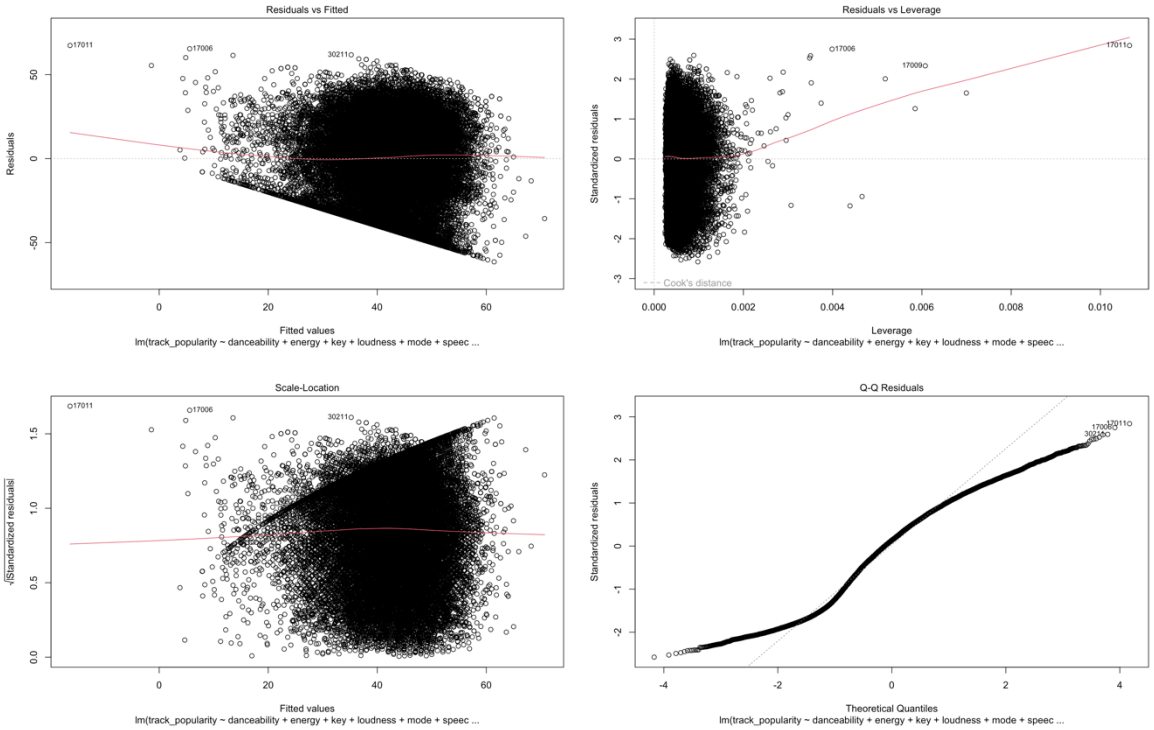
```
> coef_df <- as.data.frame(summary(popularity_predict)$coefficients)
> sorted_coef <- coef_df[order(abs(coef_df$Estimate), decreasing = TRUE), ]
> print(head(sorted_coef, 4))
                          Estimate Std. Error    t value      Pr(>|t|)
(Intercept)               70.22777  1.7264335  40.677949  0.000000e+00
energy                   -29.38964  1.2239487 -24.012150 2.560355e-126
danceability              10.52262  1.1372877   9.252379  2.324350e-20
factor(playlist_genre)pop  9.67635  0.4725093  20.478644  1.268436e-92
```

Based on the above few lines of code, I further extracted the Estimate coefficient and sorted it, and finally got the top three features that affect popularity:

- **Energy** (-29.39): Songs with higher energy tend to be less popular. This might be because high-energy songs can sometimes be overwhelming or less relaxing. In the music industry, it's often about balance - a too energetic song might not appeal to everyone.
- **Danceability** (10.52): Songs that are easier to dance to are more popular. This makes sense, as music is often used for entertainment and enjoyment. Songs that make people want to get up and dance will likely be hits.
- **Pop Genre** (9.68): Songs that belong to the pop genre tend to be more popular. Pop music is designed to appeal to a broad audience, which could explain why it's associated with higher popularity.

5.2 Assess the goodness offit of your model. If the model fits the data well, explain why you think it's possible to "quantify art." If not, discuss what component(s) might be missing.

In my current model, the Multiple R-squared value is 0.09057. This means that my model can explain only approximately 9.06% of the variability in track_popularity . Apparently, it's not good enough. I tried to plot this model on the data set and found that the fit was not ideal.



Although quantifying art is complicated, I still think there is room for improvement:

- First, I did not take the time to clean up the dataset itself and remove some outliers, nor did I check whether there are interaction effects between these features
- Second, we may need more predictor features than we had. A digitized song can contain a large number of features.
- Then, linear regression might not be a good fit, and we may need to seek a prediction model with non-linear relationships or other machine learning algorithms.
- Finally, art is very subjective. We can use models such as K-nearest neighbor algorithms to build a recommendation engine that suits personal tastes. However, aggregating into a single popularity may not be an excellent quantitative goal.

5.3 Evaluate whether your regression suffers from multicollinearity.

```
> vif(popularity_predict)
                          GVIF Df GVIF^(1/(2*Df))
danceability          1.573851  1        1.254532
energy                2.834182  1        1.683503
key                   1.032936  1        1.016335
loudness              2.167543  1        1.472258
mode                  1.048854  1        1.024136
speechiness           1.295070  1        1.138012
acousticness          1.487413  1        1.219595
instrumentalness      1.214197  1        1.101906
liveness              1.052433  1        1.025882
valence               1.393215  1        1.180345
tempo                 1.077496  1        1.038025
duration_ms           1.080212  1        1.039332
factor(playlist_genre) 2.382767  5        1.090707
```

The Variance Inflation Factor (VIF) measures multicollinearity in my model. A rule of thumb is that if the VIF is above 1, then multicollinearity is present, and if it exceeds 5 or 10, it is high.

My results show that none of the VIF values exceed 5, suggesting that multicollinearity is not a major concern in my model.