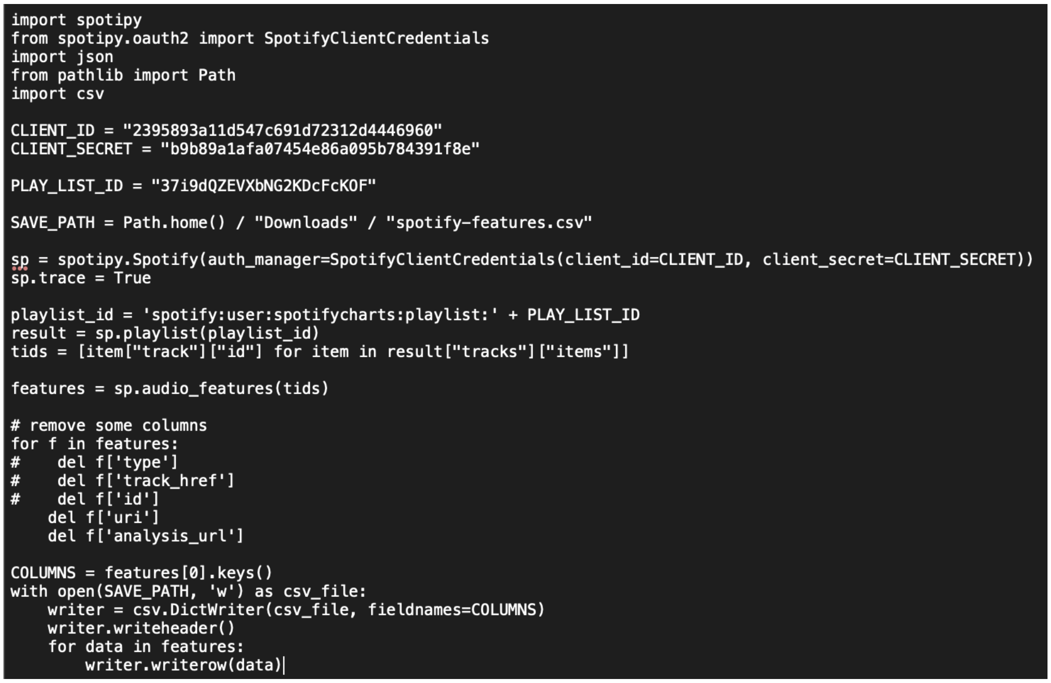
Final Project: Predicting Music Trends with Spotify
INFO 301
Mandy Zhou
2021/12/14

Introduction

Spotify is one of the largest streaming and media service providers in the world that uses data to quantify music, thanks to the emergence of data science applications in all fields. The better they are able to apply data analytics into their service, the better they are able to develop systems and utilize algorithms to generate more revenue for not only themselves, but also their stakeholders. As a music lover, I am passionate about investigating Spotify's unique strategy of making music analysis quantitative. That said, I begin to look into the relationship between data and music.

To begin, I determine that there are several aspects that can make a song popular - the features of the music itself, the album's availability on the market due to the record label recordings, and the artists' popularity. Therefore, I looked into these three directions for my data collection. After conducting research on currently available public data on Spotify tracks, I noticed one issue - the data is not up-to-date. New music comes out every day and the ranking of the most popular songs also change. It won't be relevant if I use data generated from a year ago to predict current trends. For that reason, I looked into Spotify for Developers. According to Spotify, "based on simple REST principles, the Spotify Web API endpoints return JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue". From there, I am able to use Spotify's publicly available sources, apply the methods in python, and generate three datasets: spotify-features, spotify-album, and spotify-artists from the playlist of *Top Songs - Global.* The playlist has 50 records, ranked from the most popular one to the least popular one up-to-date.



```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import json
from pathlib import Path
import csv

CLIENT_ID = "2395893a11d547c691d72312d4446960"
CLIENT_SECRET = "b9b89a1afa07454e86a095b784391f8e"

PLAY_LIST_ID = "37i9dQZEVXbNG2KDcFcKOF"

SAVE_PATH = Path.home() / "Downloads" / "spotify-features.csv"

sp = spotipy.Spotify(auth_manager=SpotifyClientCredentials(client_id=CLIENT_ID, client_secret=CLIENT_SECRET))
sp.trace = True

playlist_id = 'spotify:user:spotifycharts:playlist:' + PLAY_LIST_ID
result = sp.playlist(playlist_id)
tids = [item["track"]["id"] for item in result["tracks"]["items"]]

features = sp.audio_features(tids)

# remove some columns
for f in features:
#    del f['type']
#    del f['track_href']
#    del f['id']
    del f['uri']
    del f['analysis_url']

COLUMNS = features[0].keys()
with open(SAVE_PATH, 'w') as csv_file:
    writer = csv.DictWriter(csv_file, fieldnames=COLUMNS)
    writer.writeheader()
    for data in features:
        writer.writerow(data)
```

Dataset <Spotify-features>

The first dataset is Spotify-features, where all the features of tracks are quantified and provided by Spotify's database. The fields are:

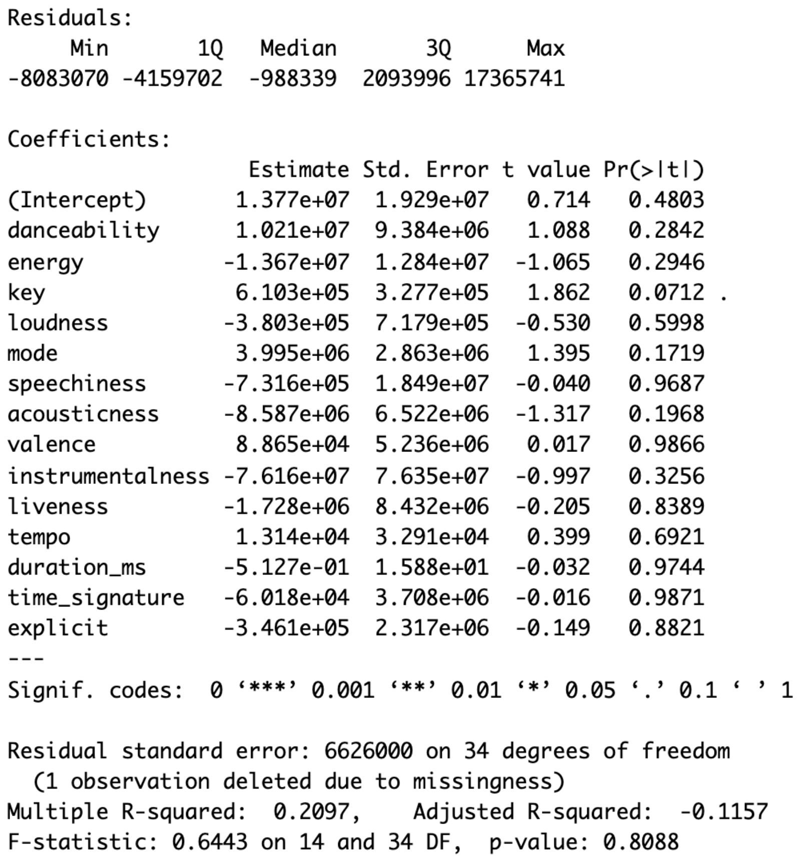
- Rank: the ranking of the song in the Top 50 gongs.
- Id: the track's id in Spotify database
- Title: track's title
- Artist: artist who performed the track
- Playtimes: the total number of play times up-to-date
- Danceability: describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- Energy: energy of Song — 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.
- Key: The key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.
- Loudness: Loudness of the track
- Mode: Indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
- Speechiness: the presence of spoken words in a track.
- Acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic
- Instrumentalness: whether a track contains no vocals
- Liveness: the presence of an audience in the recording
- Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
- Tempo: the speed or pace of a given piece and derives directly from the average beat duration
- Duration_ms: The duration of the track in milliseconds.
- Time_signature: The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
- Explicit: Whether a track contains explicit contents.

Analysis

We first start the analysis by running quantitative analysis on the numerical data variables, focusing on regression and correlation analysis.

- Start with backward selection: Starts from the fullest model with all the variables and systematically drops terms. Here, we are using the fullest regression model with all variables to predict the popularity of song by looking at playtimes

```
featuresfit0 <-
lm(playtimes~danceability+energy+key+loudness+mode+speechiness+acousticness+valence+instrumentalness+liveness+tempo+duration_ms+time_signature+explicit,data=spotify_features)
summary(featuresfit0)
```



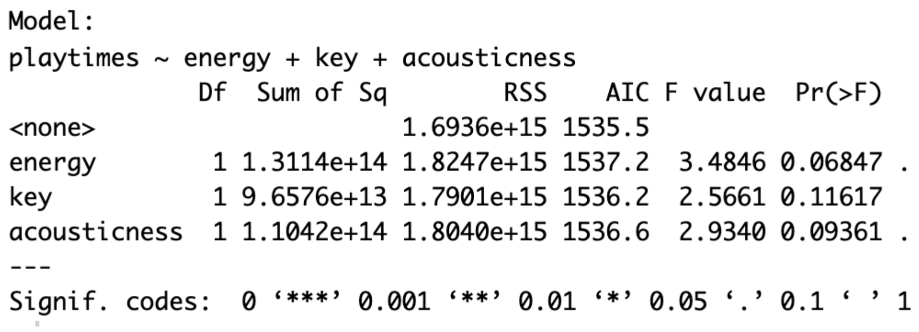
```
Residuals:
     Min       1Q   Median       3Q      Max 
-8083070 -4159702  -988339  2093996 17365741 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)  
(Intercept)       1.377e+07  1.929e+07   0.714   0.4803  
danceability      1.021e+07  9.384e+06   1.088   0.2842  
energy           -1.367e+07  1.284e+07  -1.065   0.2946  
key               6.103e+05  3.277e+05   1.862   0.0712 .
loudness         -3.803e+05  7.179e+05  -0.530   0.5998  
mode              3.995e+06  2.863e+06   1.395   0.1719  
speechiness      -7.316e+05  1.849e+07  -0.040   0.9687  
acousticness     -8.587e+06  6.522e+06  -1.317   0.1968  
valence           8.865e+04  5.236e+06   0.017   0.9866  
instrumentalness -7.616e+07  7.635e+07  -0.997   0.3256  
liveness         -1.728e+06  8.432e+06  -0.205   0.8389  
tempo             1.314e+04  3.291e+04   0.399   0.6921  
duration_ms      -5.127e-01  1.588e+01  -0.032   0.9744  
time_signature   -6.018e+04  3.708e+06  -0.016   0.9871  
explicit         -3.461e+05  2.317e+06  -0.149   0.8821  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6626000 on 34 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2097,	Adjusted R-squared:  -0.1157 
F-statistic: 0.6443 on 14 and 34 DF,  p-value: 0.8088
```

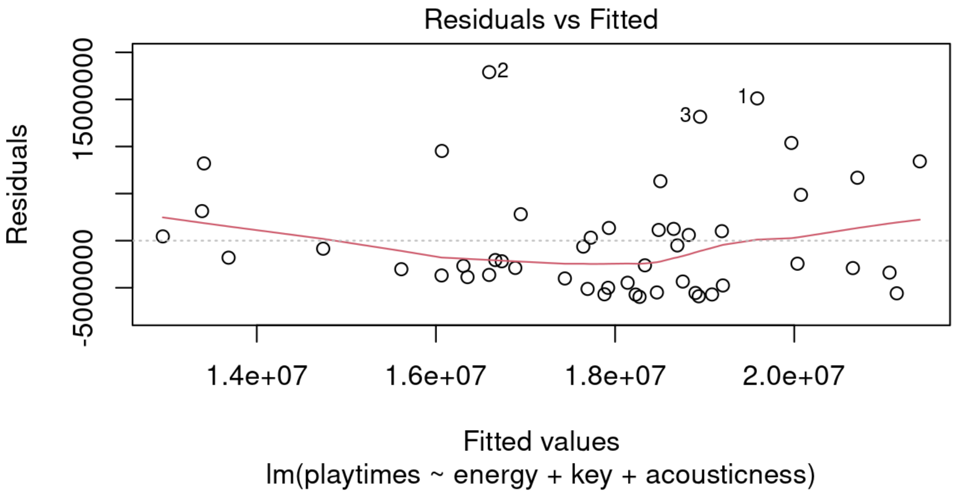
Here, we are able to see that the p-value is too big for any predictions. Among all the 14 features of a track, the majority of them do not have a significant contribution to the song’s popularity.

Therefore, we use the drop1() function to drop the terms, followed by the update function to modify the model. As we remove the values that do not result in a statistically significant detriment to accuracy of fit, we remove the term with the largest nonsignificant p-value in the partial F-test. Going through all the terms, we remove the terms in the sequence of time_signature, speechiness, duration_ms, explicit, liveness, valence, tempo, loudness, danceability, and instrumentalness, leaving the most significant three terms - energy, key, and acousticness.



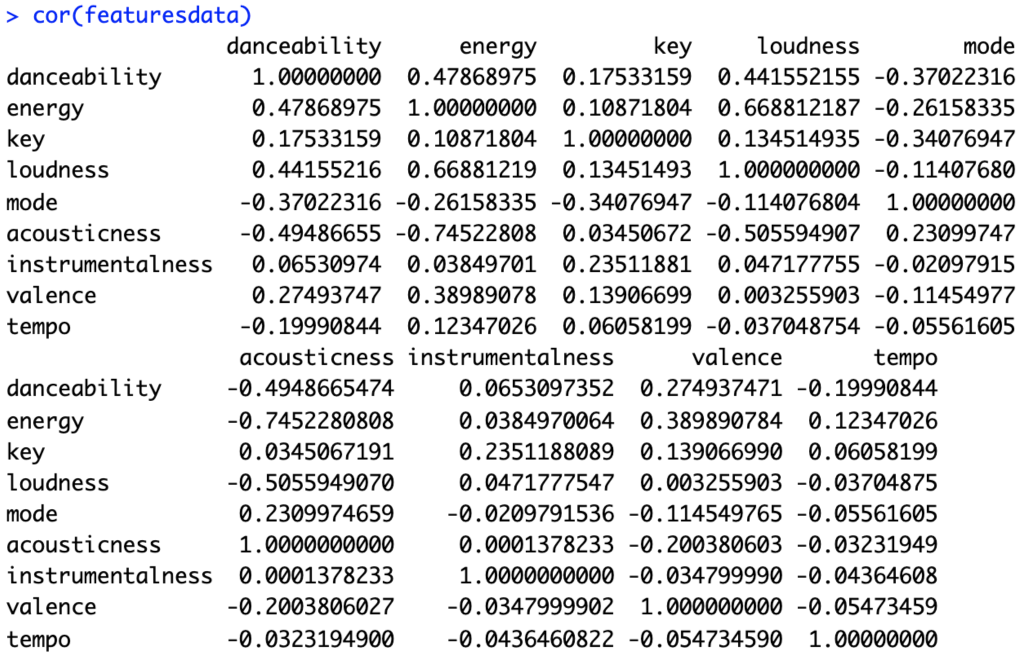
```
Model:
playtimes ~ energy + key + acousticness
             Df  Sum of Sq        RSS    AIC F value  Pr(>F)
<none>                     1.6936e+15 1535.5
energy        1 1.3114e+14 1.8247e+15 1537.2  3.4846 0.06847 .
key           1 9.6576e+13 1.7901e+15 1536.2  2.5661 0.11617
acousticness  1 1.1042e+14 1.8040e+15 1536.6  2.9340 0.09361 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Here, the p-value for energy, key, and acousticness are significantly less than the rest of the terms. Although all three of them still possess relatively large p-values for it to be significant for the prediction, these top three values do contribute the most to the popularity of the tracks for these Top 50 songs.



The model has a slightly significant that energy, key, and acousticness properties contribute to the majority of playtimes for each track. However, the trend is not linear and there are a lot of outliers, as we are able to see that although there is a cluster of data around the fitted line, a number of data are wide-spreaded on the rest of the graph.

We want to further examine the correlation between the top factors that are more significant than the rest, together with their relationship with the tracks’ popularity by making a data frame of these fields and run the correlation analysis.

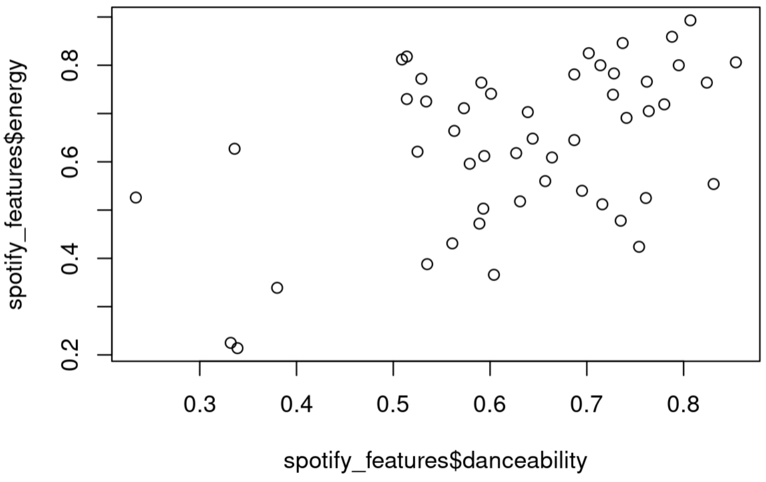


```
> cor(featuresdata)
                 danceability      energy         key     loudness        mode
danceability       1.00000000  0.47868975  0.17533159  0.441552155 -0.37022316
energy             0.47868975  1.00000000  0.10871804  0.668812187 -0.26158335
key                0.17533159  0.10871804  1.00000000  0.134514935 -0.34076947
loudness           0.44155216  0.66881219  0.13451493  1.000000000 -0.11407680
mode              -0.37022316 -0.26158335 -0.34076947 -0.114076804  1.00000000
acousticness      -0.49486655 -0.74522808  0.03450672 -0.505594907  0.23099747
instrumentalness   0.06530974  0.03849701  0.23511881  0.047177755 -0.02097915
valence            0.27493747  0.38989078  0.13906699  0.003255903 -0.11454977
tempo             -0.19990844  0.12347026  0.06058199 -0.037048754 -0.05561605
                  acousticness instrumentalness      valence       tempo
danceability     -0.4948665474     0.0653097352  0.274937471 -0.19990844
energy           -0.7452280808     0.0384970064  0.389890784  0.12347026
key               0.0345067191     0.2351188089  0.139066990  0.06058199
loudness         -0.5055949070     0.0471777547  0.003255903 -0.03704875
mode              0.2309974659    -0.0209791536 -0.114549765 -0.05561605
acousticness      1.0000000000     0.0001378233 -0.200380603 -0.03231949
instrumentalness  0.0001378233     1.0000000000 -0.034799990 -0.04364608
valence          -0.2003806027    -0.0347999902  1.000000000 -0.05473459
tempo            -0.0323194900    -0.0436460822 -0.054734590  1.00000000
```

From the correlation matrix, we notice that there are several fields that possess strong correlation with another variable.

- Energy and danceability

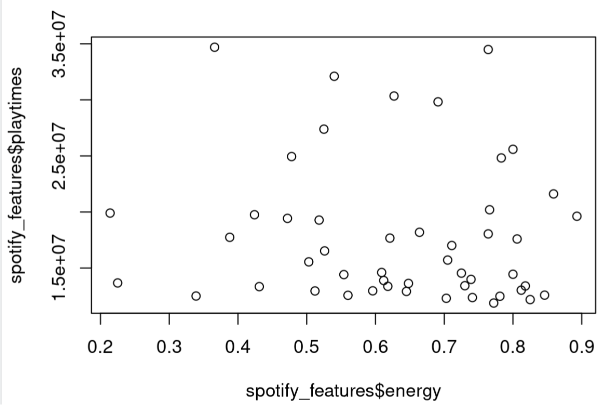
```
energyfit <- glm(danceability~energy, data=spotify_features, family = binomial)
plot(spotify_features$energy~spotify_features$danceability)
curve(predict(energyfit, data.frame(energy=x),type="response"),add=TRUE)
plot(energyfit, which=2)
```

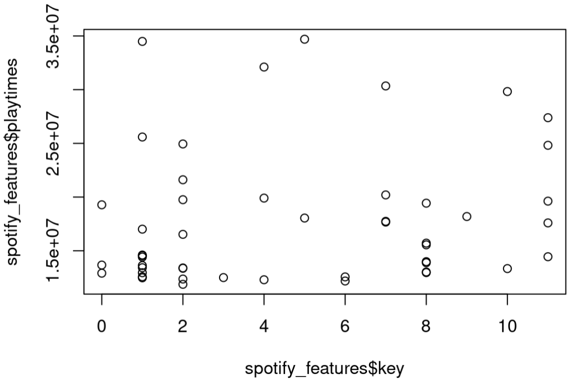


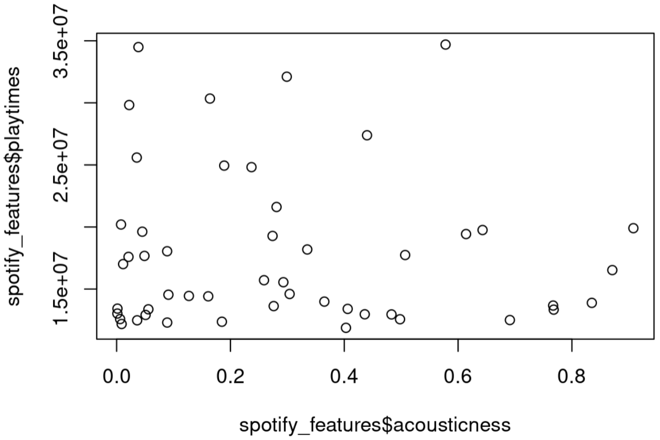
As displayed in the graph, the higher the danceability, the energy of the song seems to be greater as well, proving their relatively strong correlation of 0.49.

Then, we plot the three variables that have the strongest contribution to popularity - energy, key, and acousticness.

```
plot(spotify_features$energy, spotify_features$playtimes)
plot(spotify_features$key, spotify_features$playtimes)
plot(spotify_features$acousticness, spotify_features$playtimes)
```

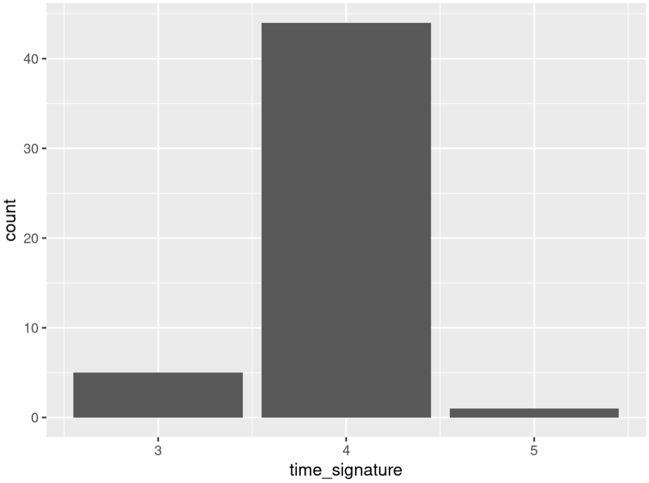


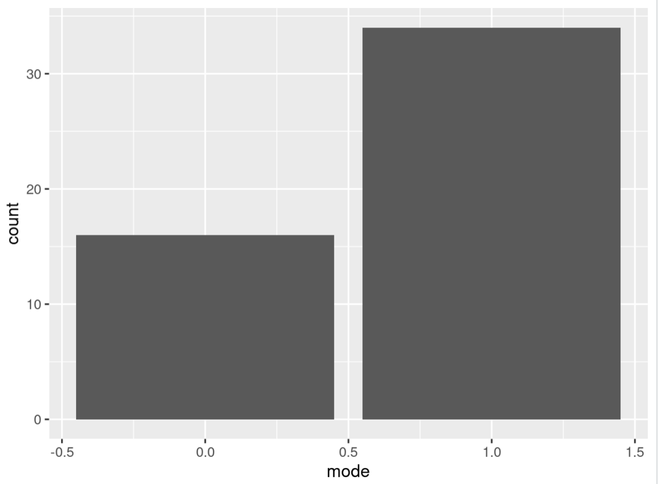


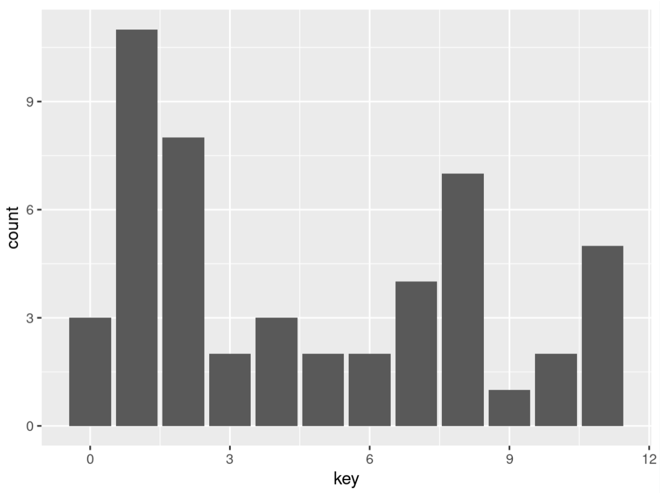


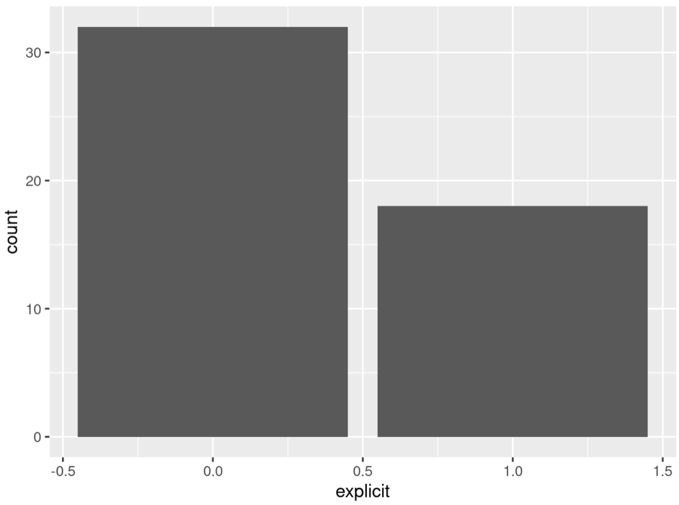
However, although all three of them have the strongest significance in determining the playtimes of tracks, when we look at each variable separately, there isn't a strong correlation.

After looking at the numeric variables, we examine the categorical and dummy variables - time_signature, mode, key, and explicit. Here, we use ggplot to graph the counts as y, and the fields as x to examine the data distribution.









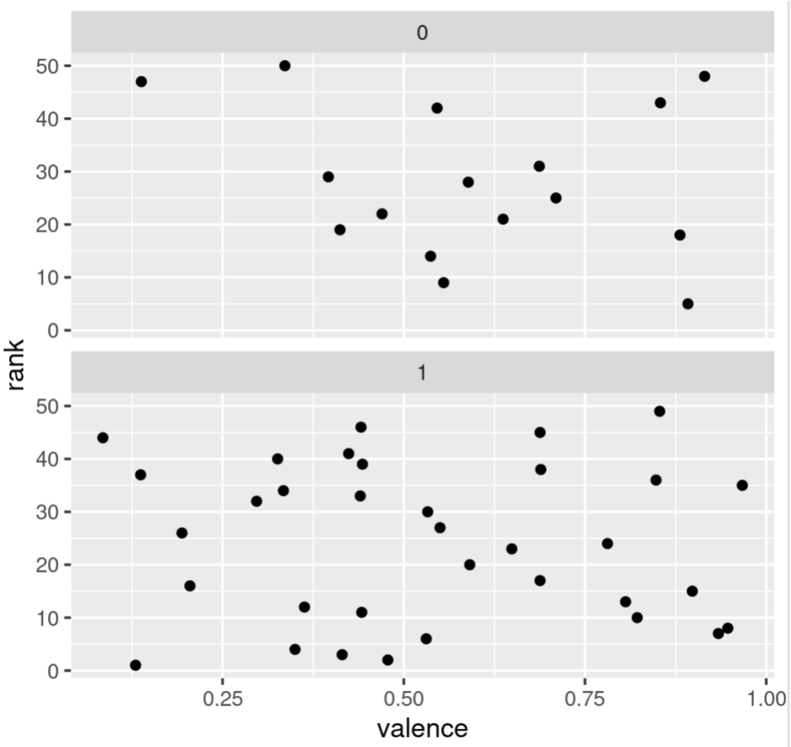
From the graphs, we are able to see that out of the 50 tracks, the majority of the tracks are in four for time_signature; major key for mode; and with few explicit contents. In addition, the majority of the tracks are in C major, D major, and G major, which are the most popular keys for popular music; followed by A minor and E minor, as their relative minor.

From here, we would think that because of the features mentioned above, the more popular the track is, the more positive it would be. I run further analysis on this hypothesis by using scatterplot facet wrap and found something different.

```
#scatterplot facet wrap of valence and key
ggplot(spotify_features, aes(x=valence, y=rank))+geom_point()+facet_wrap(~key,nrow=4)

#scatterplot facet wrap of key and made
#More major mode tracks than minor
ggplot(spotify_features, aes(x=key, y=rank))+geom_point()+facet_wrap(~mode,nrow=1)
```

```
#scatterplot facet wrap of key and mode
ggplot(spotify_features, aes(x=valence, y=rank))+geom_point()+facet_wrap(~mode,nrow=2)
```



As we look closely at the graph, we can find that although there are more tracks in major keys than in minor keys, there are almost equal numbers of tracks that have depressing and sad features compared to their bright and positive features. Therefore, we can conclude that even though the majority of the population enjoy the most common and positive keys in their preferences of music, the content of the song tells another story. For example, while a lot of songs are in major keys and bring out a positive mood to the listeners, the lyrics can be about “break-ups” or sad experiences, which can even make the song more popular.

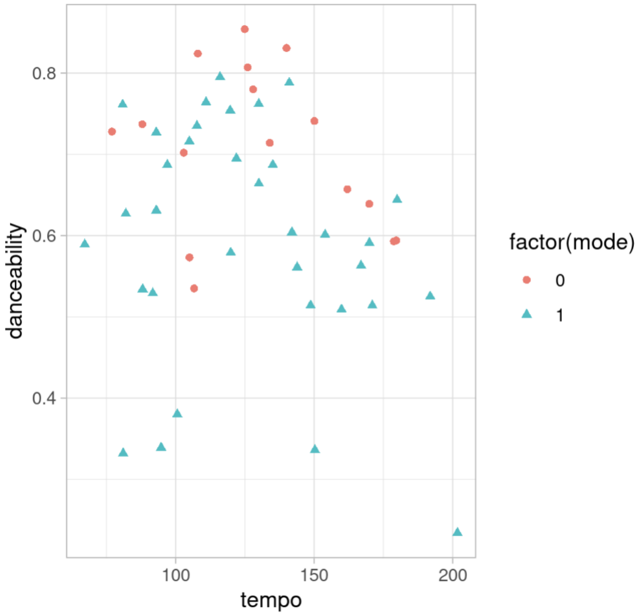
Then, we look at combination analysis on numerical and categorical variables based on their correlation with each other and how they both contribute to popularity.

Tempo, danceability, and energy seems to be strongly correlated based on the analysis above. We used ggplot with aes function to examine how strong they are actually related to mode.

```
ggplot(spotify_features, aes(x=energy, y=danceability))+
   geom_point(aes(col=factor(mode), shape=factor(mode)))+theme_light()
```

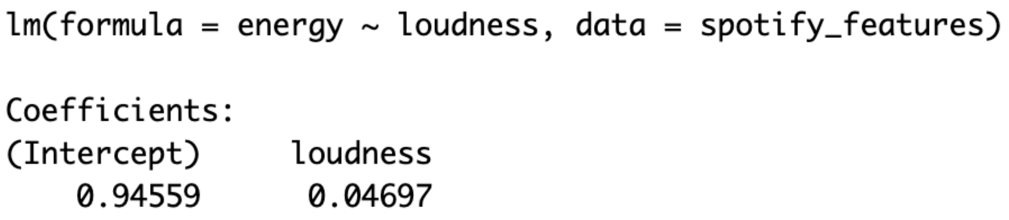
```
ggplot(spotify_features, aes(x=tempo, y=energy))+
  geom_point(aes(col=factor(mode), shape=factor(mode)))+theme_light()

ggplot(spotify_features, aes(x=tempo, y=danceability))+
  geom_point(aes(col=factor(mode), shape=factor(mode)))+theme_light()
```



The strongest graph shows that tracks with tempo between 100 and 150 have relatively high danceability, together with a major key. This makes sense because usually people dance to music with a bright mood and faster speed.

Another interesting factor to look into is the energy's correlation with loudness. Typically, we might think that the more energetic music should be louder in volume. However, the regression model indicates that these two variables don't possess strong relationships.



```
lm(formula = energy ~ loudness, data = spotify_features)

Coefficients:
(Intercept)     loudness
    0.94559      0.04697
```

To conclude, the spotify-features dataset tells us a lot about how certain features of music affect each other and contribute to the popularity of the track as a whole. As they tend to have a stronger correlation with each other instead of the total playtimes, we will look at the other possible factors that analysts should investigate in order to predict the trend of music.

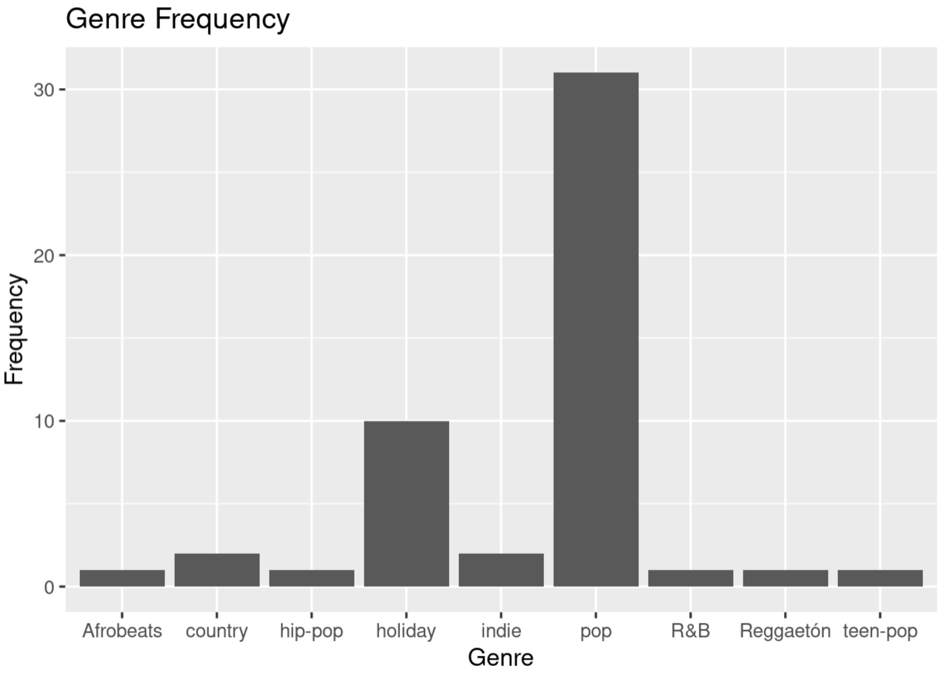
Dataset<Spotify-album>

The second dataset is Spotify-album, where all the album of tracks are quantified and provided by Spotify's database. The fields are:

- Rank: the ranking of the song in the Top 50 gongs.
- Title: the track's title
- Artist: the artist who performed the track
- Album: the album where the track belongs
- Playtimes: the total times the track is played
- Genre: the genre of the album
- Available_markets: the markets available for the album
- Recordlabel: the record label company who owns the produced the album

Analysis

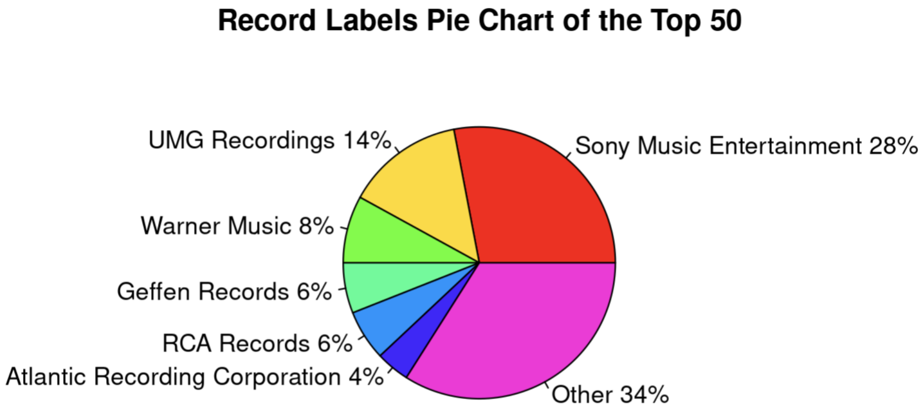
To begin with, we first look at the main genres of the most popular 50 songs currently in the world today. Here, I created a quick plot using



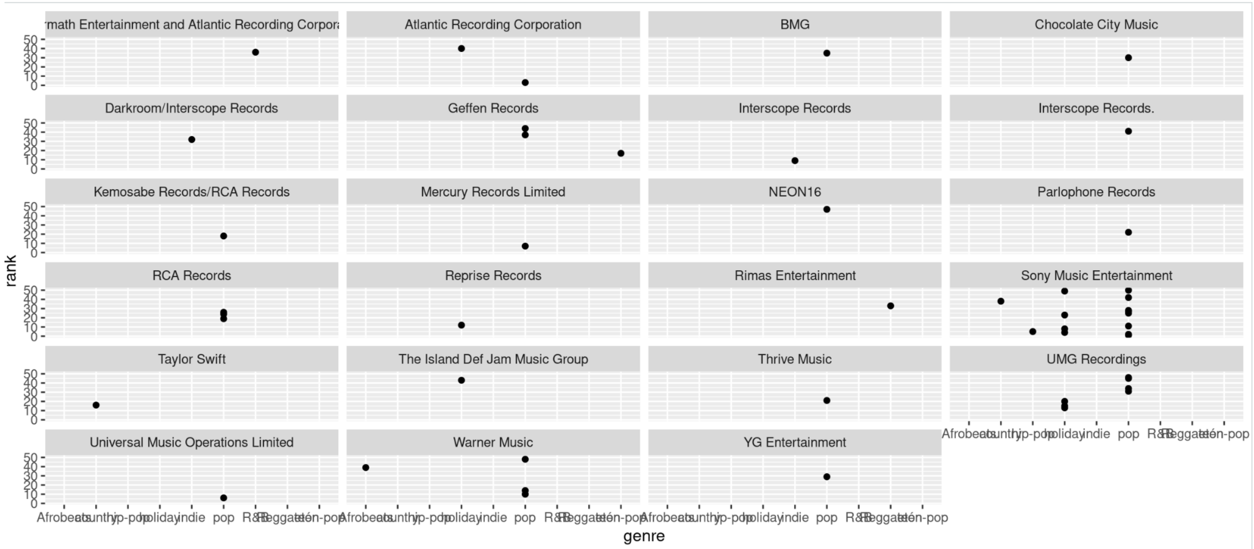
Frankly speaking, it is not shocking that pop is the genre with the most frequency out of the top 50 tracks, as it is the most mainstream genre of music. Another interesting finding from this analysis is that the genre of "holiday music" is also significantly higher than other genres. This is the benefit of scraping data from the up-to-date playlist because we are able to see that holiday music is being played more during this time of the year.

Then, let's look at what record label companies own the majority of the track/albums. Here, I created a pie chart after installing the "plotrix" package:

```
slices <- c(14, 7, 4, 3, 3, 2, 17)
lbls <- c("Sony Music Entertainment",  "UMG Recordings","Warner Music", "Geffen Records",
"RCA Records", "Atlantic Recording Corporation","Other")
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%",sep="")
pie(slices,labels = lbls, col=rainbow(length(lbls)),
   main="Record Labels Pie Chart of the Top 50")
```



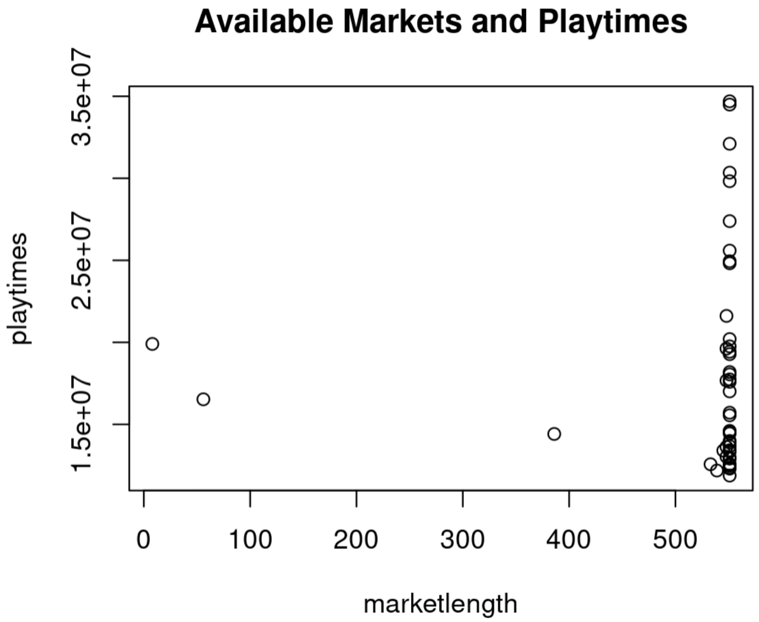
From the pie chart, we can see that Sony Music Entertainment almost dominates the market with its successful music production as it owns up to 28% of the top 50 tracks. Followed by Universal Music Group with 14% of the tracks, and Warner Music with 8% of the tracks. From this analysis, we can also conclude that most of the successful and popular tracks are usually produced and owned by the biggest record label companies in the world.



In addition, from the scatterplot with facet wrap, we can also see that Sony Music Entertainment not only has the most data, meaning they own the most tracks in the top 50 playlist, but also possess songs with widely distributed rankings.

Next, let’s move on to the available market aspect. The question is, does it make a song more popular if it is available to more markets? I began this analysis by installing the “stringr” package for text analysis because there are multiple records within one field, making it hard to compare the values. Therefore, I applied the str_length() method to compare the length of the field: available_markets. The longer the string field, the more market the album should be available in.

```
markettext <- c(spotify_album$available_markets)
marketlength <- str_length(markettext)
marketlength
playtimes <- c(spotify_album$playtimes)
playtimes
plot(marketlength,playtimes, main="Available Markets and Playtimes")
```



From the str_length() method, the program tells us that the majority of the text length for the field available_market has the value around 500, regardless of the popularity of the songs. That said, it is unfortunate that the available market is not a significant factor to determine the popularity of tracks in this dataset. Since the date already comes from the most popular songs world-wide, it is obvious that the majority of them are available in all markets. However, the text analysis is a good way for us to test our hypothesis and illustrates a few outliers of our findings.

Dataset<spotify-artists>

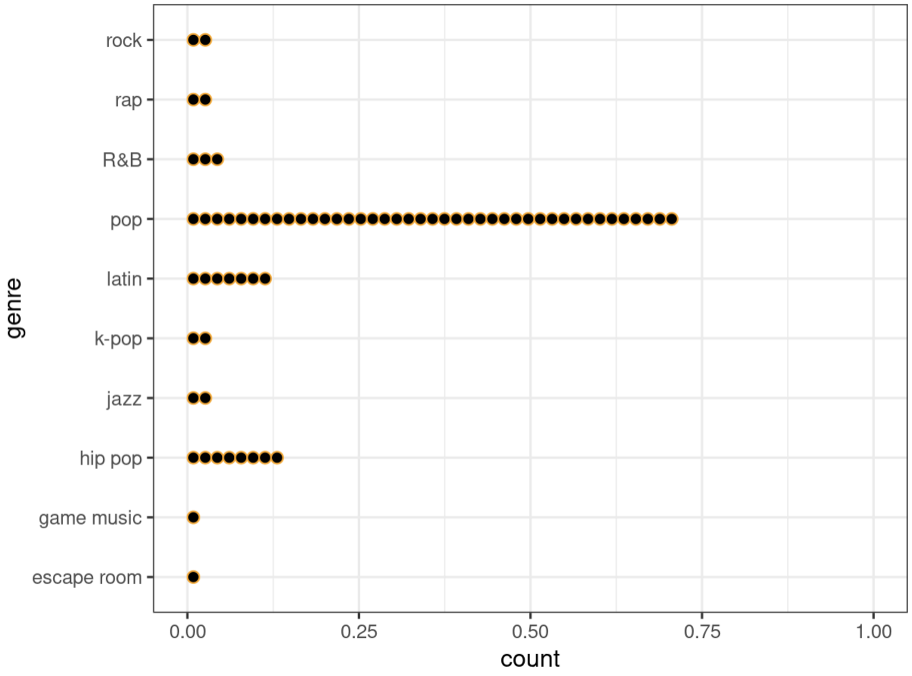
The third dataset is Spotify-artists, where all the artists of tracks are quantified and provided by Spotify's database. The fields are:

- Title: track title
- Playtimes: total times the track is played
- Track_id: id of the track
- Artist_name: name of the artist who performed the track
- Artist_id: id of the artist
- Followers: total followers of the artist on Spotify
- Popularity: the popularity of the artist ranked by Spotify's database
- Genre: genre of the artist

Analysis

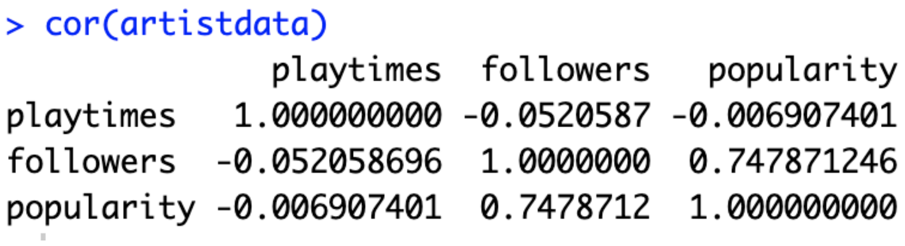
Similar to the analysis on albums, let's also take a look at the genres of the artists who produced the most popular 50 tracks world-wide at the moment though ggplot:

```
ggplot(spotify_artists,
    aes(x=genre),
    horiz=TRUE,
    main="count of artist genre",
    xlab="artist genre",
    ylab="count")+geom_dotplot(binwidth=0.2,color="orange")+coord_flip()+theme_bw()
```



Similar to the album's analysis, tracks with pop genres take up most of the space in the top 50 list. Different from the album's analysis, there is no "holiday genre" for artists and the second popular genre for artists are hip pop artists.

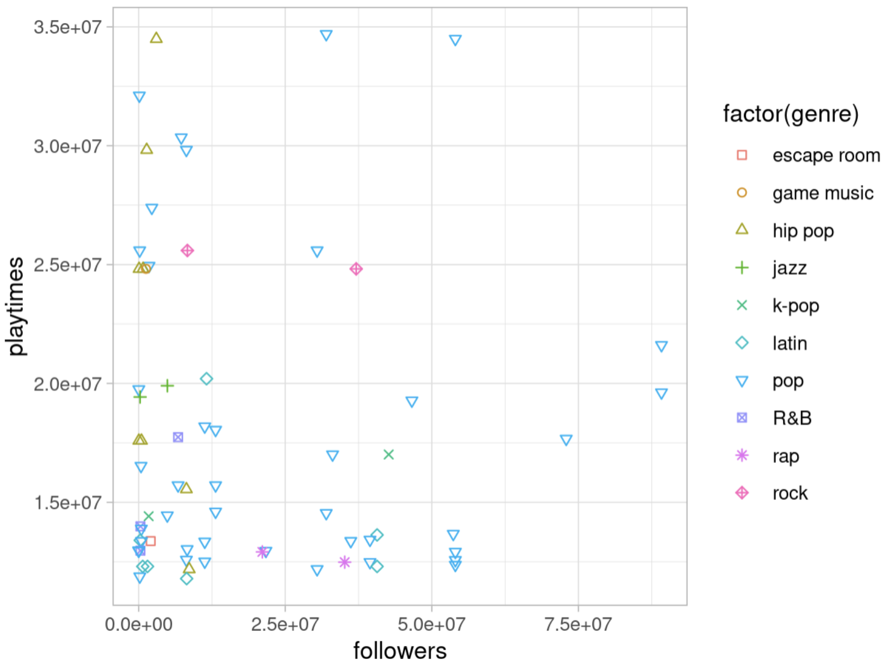
Then, let's examine the correlation between playtimes, followers, and popularity rank. Seems like all three fields should be strongly correlated because they all speak "popularity" to the audience.

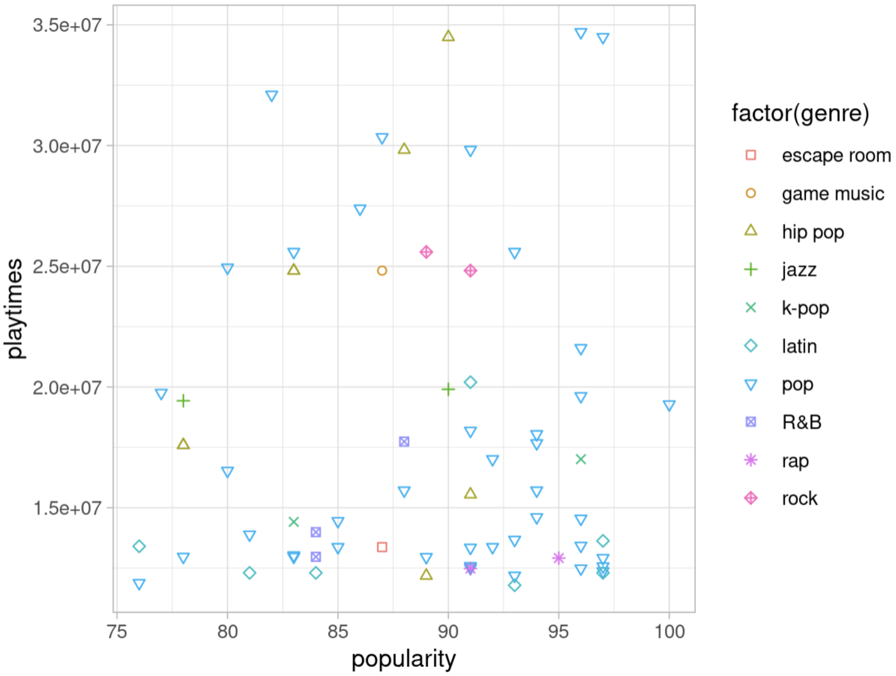


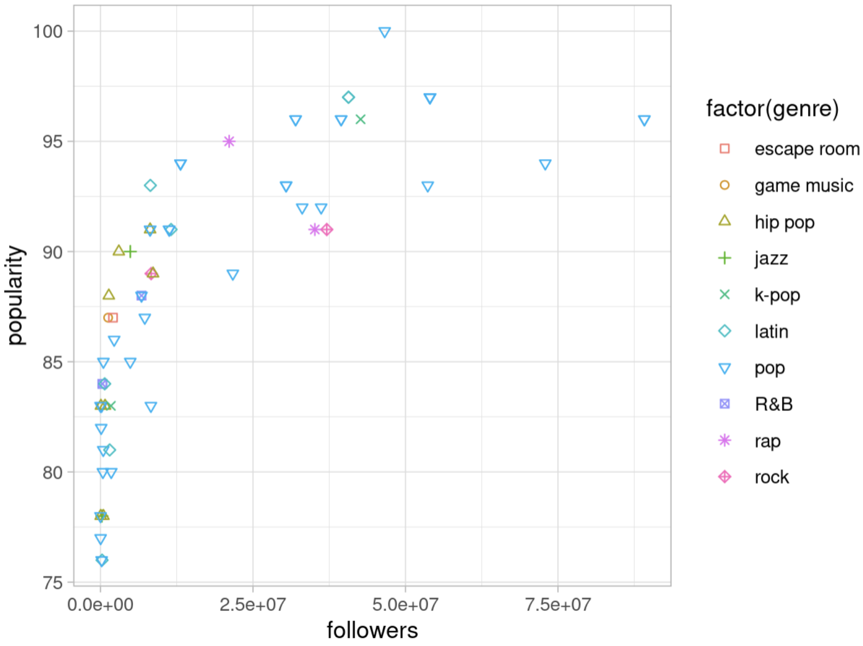
```
> cor(artistdata)
              playtimes  followers   popularity
playtimes   1.000000000 -0.0520587 -0.006907401
followers  -0.052058696  1.0000000  0.747871246
popularity -0.006907401  0.7478712  1.000000000
```

However, it seems like only the field's followers and popularity have a strong population, which explains how Spotify's own algorithm of ranking the artists' popularity within its system - based on the number of followers on Spotify. On the other hand, even the most popular artist has the most followers, it doesn't make their current track the most popular one on the playlist. I do, however, believe that there is a better possibility for a track produced by a popular artist to make it to the top 50 rank, compared to local, smaller artists.

The graphs also visually display the conclusion mentioned above:







Based on the last graph, we can see a trend of the more followers starting from a certain point, the popularity grows, which indicates that the higher the population score, the more scarce the data is for the artists with the most followers. We can also see from the graph that the majority of these artists are in the genre of pop, which confirms the analysis mentioned earlier.

Conclusion

From the analysis above, the biggest takeaway would be that we cannot determine the popularity of a song based on only a few factors. There are all kinds of aspects to consider when we quantify our data and attempt to predict the next trend for popular music. People around the world have different tastes, and although mainstream music shares similar features, there are always outside factors that can impact the popularity of a song - Christmas season is an example. Nevertheless, it would be beneficial for music producers and artists to consider these factors when creating new compositions. It is also interesting to think about however, whether mainstream songs are simply "common", and songs with more "personalities" makes them less popular than they should be on the musical aspect.

Limitations and future improvements

First, more data is needed for a more comprehensive research and conclusion. Due to time and resource constraints, I am unable to generate a bigger dataset from a bigger playlist. A lot of the functions available on Spotify developer API have a constraint of the amount of data it is able to generate. For example, the method for getting the artists' information from the track list only allows a maximum of 50 records. Therefore, for further research in the future, it is important to find or develop methods that allow us to generate a way bigger dataset.

In addition, while we know what the majority of the features that popular music share in common, further research can also look into the features and characteristics of other music genres - for example, top 50 jazz playlist, top 50 classical music, etc. Not to mention the tracks' musical feature will change drastically, there probably won't be a trend that is so obvious for the record label companies that produce the music.

Supporting Articles

How Spotify's Algorithm Knows Exactly What You Want to Listen To
https://onezero.medium.com/how-spotifys-algorithm-knows-exactly-what-you-want-to-listen-to-4b6991462c5c

How Spotify is using Big Data to enhance customer experience
https://www.analyticssteps.com/blogs/how-spotify-using-big-data

Spotify Popularity — A unique insight into the Spotify algorithm and how to influence it
https://lab.songstats.com/spotify-popularity-a-unique-insight-into-the-spotify-algorithm-and-how-to-influence-it-93bb63863ff0