

My Spotify Activity - 2

Ananya Mantravadi

Synopsis

This project involves visualizing my activity using my personal data from Spotify with R and *spotifyr* package which helps connect with the Spotify API. It explores the following:

1. Plotting the least and most popular tracks in the playlist \m/ (Genre: Rock/Metal)
2. Plotting the least and most popular tracks in the playlist Yeet (Genre: Pop/Hip Hop/Dance/Pop Rock)
3. Creating a quadrant with a scatter plot on parameters Energy and Valence as defined by Spotify, from the data obtained from my top four artists. This “emotional quadrant” is an attempt to visualize the feelings conveyed and expressed by the music I mostly listen to.

Importing Libraries

```
#Importing libraries and reading the data
```

```
library(spotifyr)
library(jsonlite)
library(lubridate)
library(tidyverse)
library(knitr)
library(ggplot2)
library(ggpubr)
library(plotly)
library(gghighlight)
```

Connecting to Spotify API

```
# Establishing connection and authorizing
```

```
Sys.setenv(SPOTIFY_CLIENT_ID = client_id)
Sys.setenv(SPOTIFY_CLIENT_SECRET = client_secret)
```

```
get_spotify_authorization_code()
```

```
## <Token>
## <oauth_endpoint>
##   authorize: https://accounts.spotify.com/authorize
##   access:   https://accounts.spotify.com/api/token
## <oauth_app> spotifyr
##   key:      4f73b227442046db9041d6847afa2313
```

```
## secret: <hidden>
## <credentials> access_token, token_type, expires_in, refresh_token, scope
## ---
```

The least and most popular tracks in the playlist “\m/”

```
# Spotify Playlist Features - 1

p_username <- "AM"
p_url <- "1TJEqUewyHTLD2wBmBCjMb"

playlist <- get_playlist_audio_features(p_username, p_url)

# Plotting less and more popular tracks in the playlist \m/ based on "track.popularity" variable

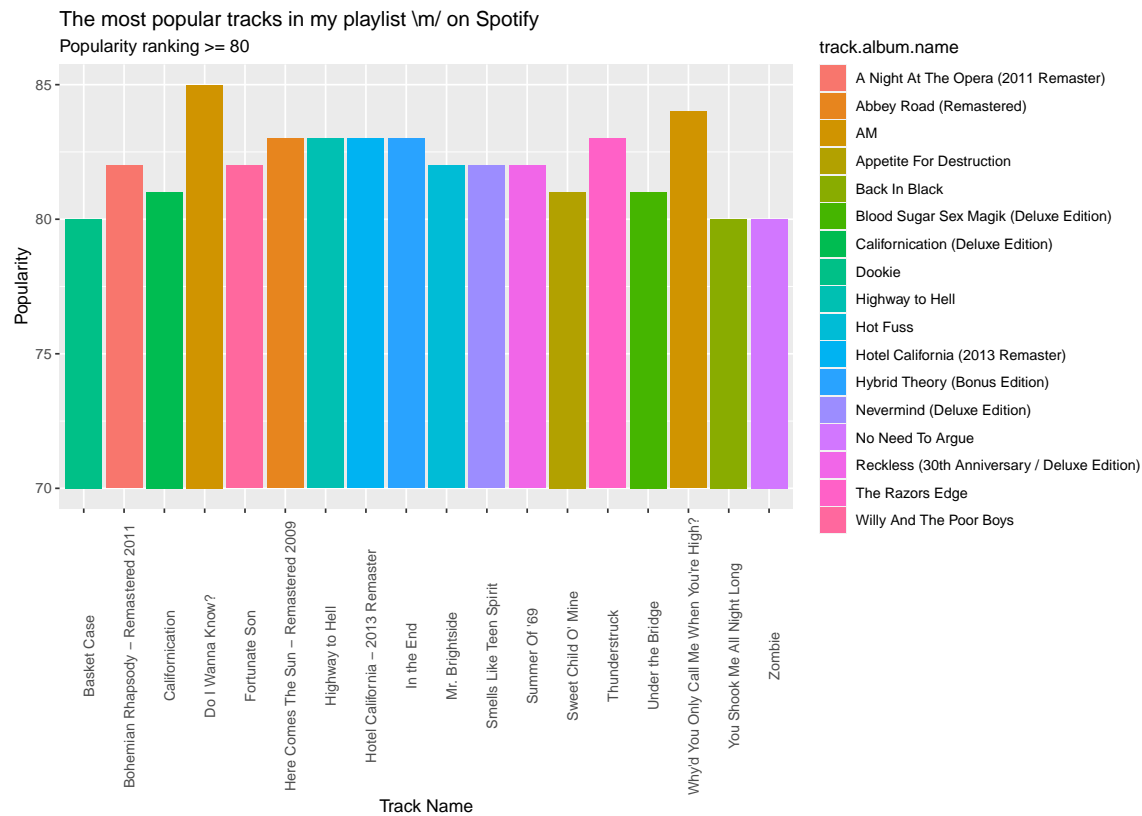
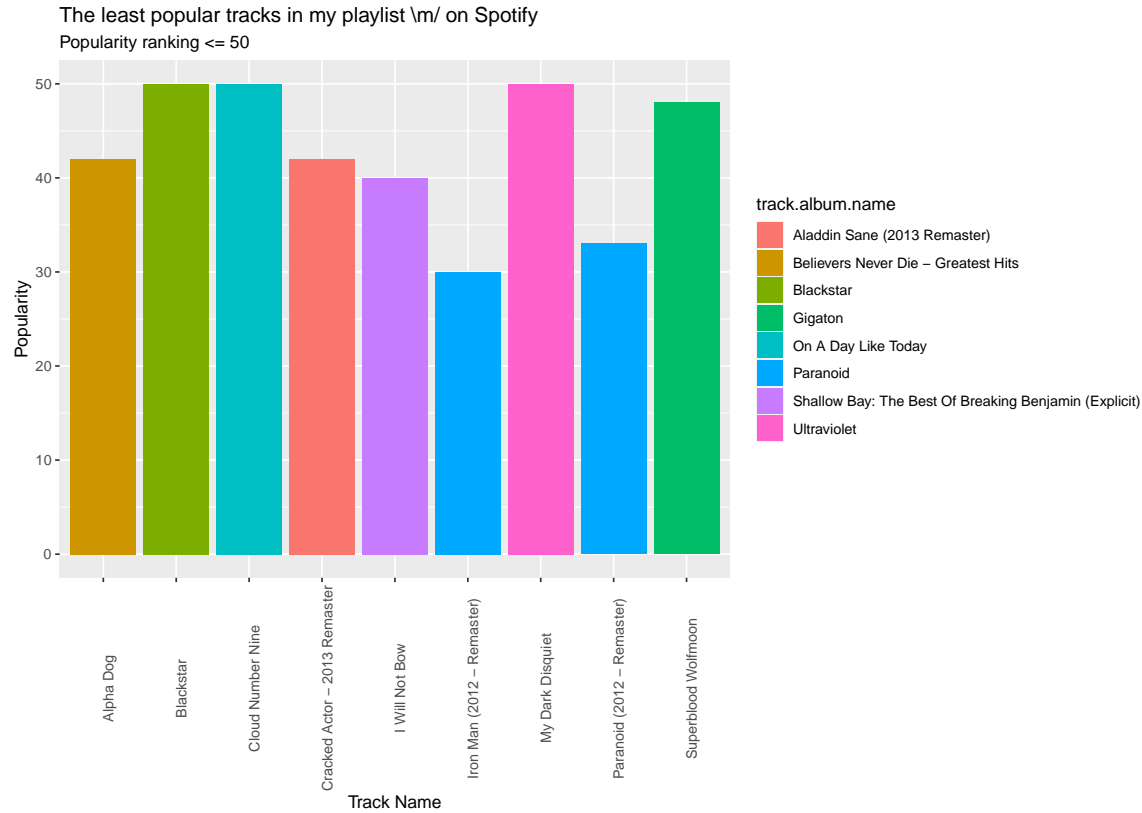
p <- playlist %>%
  group_by(track.popularity)

less_than_50 <- p %>%
  filter(track.popularity <= "50") %>%
  ggplot(aes(x = track.name, y = track.popularity)) +
  geom_col(aes(fill = track.album.name)) +
  labs(x = "Track Name", y = "Popularity") +
  ggtitle("The least popular tracks in my playlist \m/ on Spotify", "Popularity ranking <= 50") +
  theme(axis.text.x = element_text(angle=90))

more_than_80 <- p %>%
  filter(track.popularity >= "80") %>%
  ggplot(aes(x = track.name, y = track.popularity)) +
  geom_col(aes(fill = track.album.name)) +
  scale_y_continuous(limits = c(70,85), oob = scales::squish) +
  labs(x = "Track Name", y = "Popularity") +
  ggtitle("The most popular tracks in my playlist \m/ on Spotify", "Popularity ranking >= 80") +
  theme(axis.text.x = element_text(angle = 90))

figure <- ggarrange(less_than_50, NULL, more_than_80, ncol = 1, heights = c(1, 0.1, 1))

figure
```



The least and most popular tracks in the playlist “Yeet”

```
# Spotify Playlist Features - 2

p_url_2 <- "0IrhdY1o9Z4orH0dnCXDJC"
playlist_2 <- get_playlist_audio_features(p_username, p_url_2)

# Plotting less and more popular tracks in the playlist \|m/ based on "track.popularity" variable

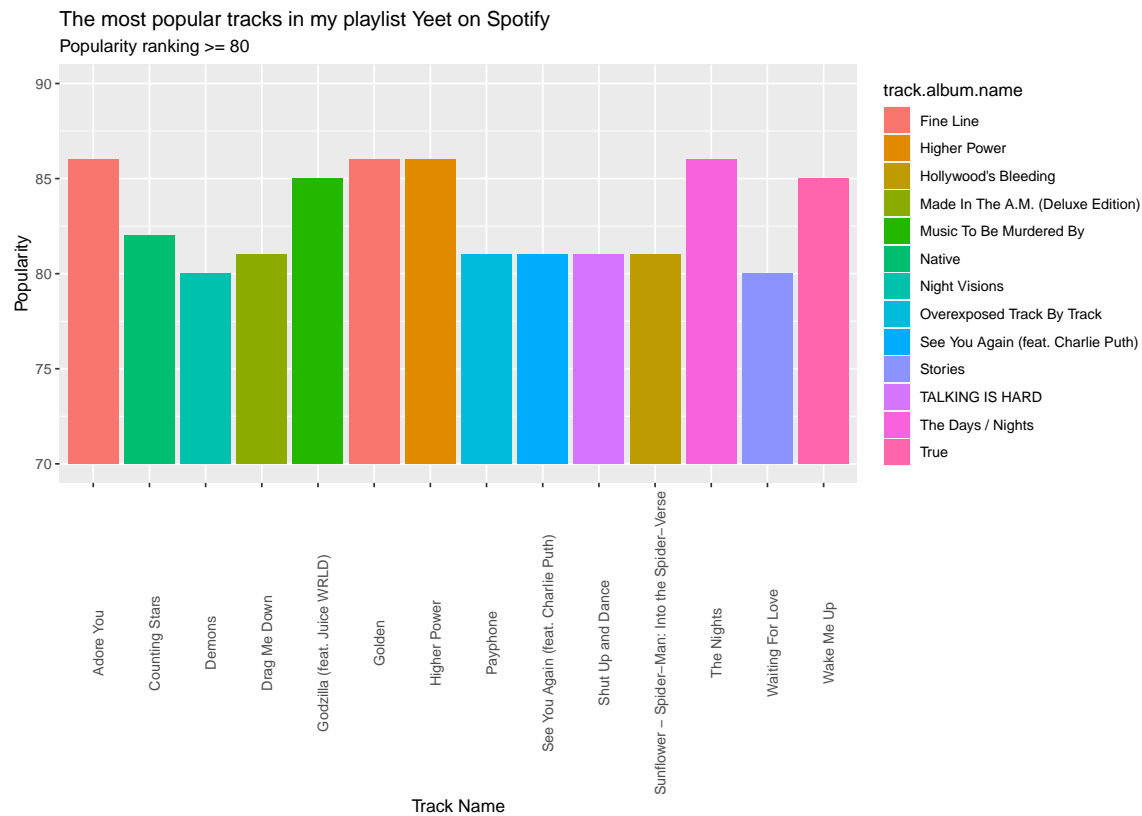
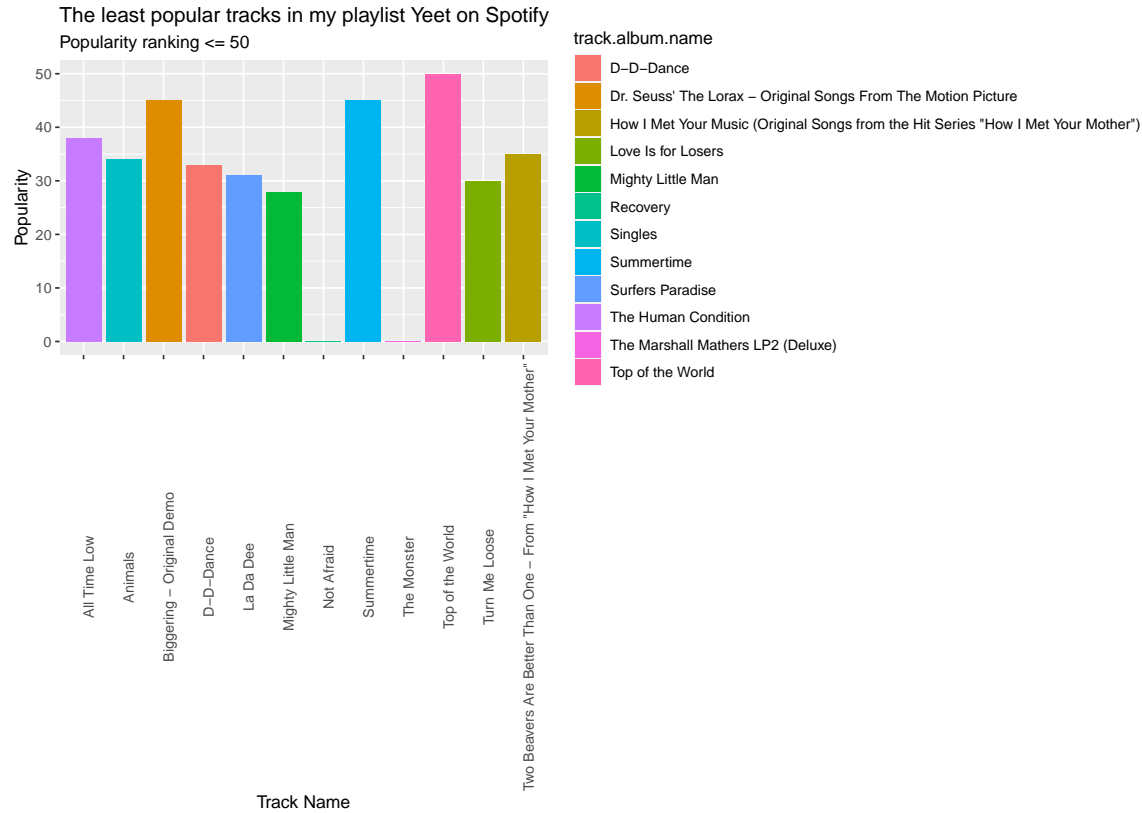
p_2 <- playlist_2 %>%
  group_by(track.popularity)

less_than_50 <- p_2 %>%
  filter(track.popularity <= "50") %>%
  ggplot(aes(x = track.name, y = track.popularity)) +
  geom_col(aes(fill = track.album.name)) +
  labs(x = "Track Name", y = "Popularity") +
  ggtitle("The least popular tracks in my playlist Yeet on Spotify", "Popularity ranking <= 50") +
  theme(axis.text.x = element_text(angle=90))

more_than_80 <- p_2 %>%
  filter(track.popularity >= "80") %>%
  ggplot(aes(x = track.name, y = track.popularity)) +
  geom_col(aes(fill = track.album.name)) +
  scale_y_continuous(limits = c(70,90), oob = scales::squish) +
  labs(x = "Track Name", y = "Popularity") +
  ggtitle("The most popular tracks in my playlist Yeet on Spotify", "Popularity ranking >= 80") +
  theme(axis.text.x = element_text(angle = 90))

figure <- ggarrange(less_than_50, NULL, more_than_80, ncol = 1, heights = c(1, 0.1, 1))

figure
```



Emotional Quadrant - My Top Four Artists

Based on Energy and Valence:

1. **Energy** - 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. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
2. **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)

```
# Getting features of my top four favorite artists
```

```
fav1 <- get_artist_audio_features(artist = "Green Day")
fav2 <- get_artist_audio_features(artist = "Coldplay")
fav3 <- get_artist_audio_features(artist = "Fall Out Boy")
fav4 <- get_artist_audio_features(artist = "Taylor Swift")
```

```
# Creating a data frame from combining the above data
```

```
top_4 <- rbind(fav1,fav2,fav3,fav4)
```

```
# Plotting the emotional quadrant scatter plot
```

```
eq <- ggplot(data = top_4, aes(x = valence, y = energy, color = artist_name)) +
  geom_point(position = "jitter") +
  geom_vline(xintercept = 0.5) +
  geom_hline(yintercept = 0.5) +
  scale_x_continuous(expand = c(0, 0), limits = c(0, 1)) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1)) +
  geom_label(x = 0.25/2, y = 0.95, label = "Angry / Turbulent", color = "black", fill = "aliceblue") +
  geom_label(x = 1.75/2, y = 0.95, label = "Joyful / Happy", color = "black", fill = "aliceblue") +
  geom_label(x = 1.75/2, y = 0.05, label = "Peace / Chill", color = "black", fill = "aliceblue") +
  geom_label(x = 0.25/2, y = 0.05, label = "Depressing / Sad", color = "black", fill = "aliceblue") +
  labs(x= "Valence", y= "Energy") +
  ggtitle("Emotional Quadrant - Top Four Artists", "Energy vs Valence")
```

```
eq
```

