

Song Language and Success in the US Music Industry

Madison Martin

2023-06-15

Introduction

This project aims to illustrate the relationship between the language of the lyrics of songs in popular music in the United States and the song's success, using data from Spotify and radio plays. It was inspired by the recent rise in popularity of music sung in non-English languages in the US by artists like BTS, Bad Bunny, Maluma, Måneskin, and Blackpink. *This project is a modified form of an original project completed for credit at the University of Chicago in 2021.*

Research Question: **Does linguistic discrimination (discrimination on the basis of the manner of speaking or language being spoken) occur in the US popular music industry, as evidenced by discrepancies in either audience support or industry support?**

Sourcing the Data

In order to address the research question, I chose to source my songs from the pool of popular songs that have appeared on the Billboard Hot 100 chart, as it has long been an industry standard. A total of 17 consecutive months of chart data were collected (from January 2020 through May 2021), for a total of 7,300 songs. The charts and song data were collected directly from Billboard, and included the song's rank (from 1-100), title, artist, and week dating the chart. In order to measure the success or support of the songs on two levels, I also collected audience support data (Spotify streams) and industry support data (radio plays). The Spotify data was sourced directly from Spotify by week (to match the weekly Hot 100 chart), while the radio play data was sourced from Chartmetric.

Language of Lyrics

In order to assess the impact of lyric language on the support a song gets, I coded each song's lyrics for language on a word-by-word basis, in order to calculate the percentage of each song's lyrics which are sung in a non-English language. I started by identifying which songs were likely to have non-English lyrics using the `cld3` package in R. However, the accuracy of language identification tools is poor, and is even less accurate on a singular word basis, so the rest of the coding for word language in each song's lyrics was done by hand. For more information, see the appendix.

Download the data

```
# download the final combined data for 2020 and for partial 2021
# Load the necessary libraries
library(readr)
library(tidyr)
```

```

library(ggplot2)
library(tidyverse)
library(dplyr)

# read in csv for 2020
total_20_df <- read_csv(file = "final_20.csv")
head(total_20_df)

## # A tibble: 6 x 16
##   year date      rank song      artist binary language total_words non_e_words
##   <dbl> <date>    <dbl> <chr>    <chr>    <dbl> <chr>      <dbl>      <dbl>
## 1  2020 2020-01-04      1 all i w~ maria~      0 english      356          0
## 2  2020 2020-01-04      2 rockin ~ brend~      0 english      119          0
## 3  2020 2020-01-04      3 jingle ~ bobby~      0 english      185          0
## 4  2020 2020-01-04      4 a holly~ burl ~      0 english      159          0
## 5  2020 2020-01-04      5 circles post ~      0 english      314          0
## 6  2020 2020-01-04      6 roxanne arizo~      0 english      355          0
## # i 7 more variables: percent <dbl>, 'individual songs' <dbl>, position <dbl>,
## #   streams <dbl>, 'spotify date' <date>, spins <dbl>, radio_rank <dbl>

# read in csv for 2021
total_21_df <- read_csv(file = "final_21.csv")

```

Note: these CSV files already contain the language coding variables along with streaming and radio play data, as all three sources of data were cleaned and combined previous to this project for brevity and clarity. If you are interested in the raw Spotify or Chartmetric data, please contact me.

Precleaning for Merging

```

# clean up the 2020 data set
# first, remove the column we don't need
total_20_df <- total_20_df %>%
  select(-`spotify date`) %>%
  filter(year == 2020) # only 2020 songs

# remove duplicate songs
library(data.table)
total_2020 <- unique(setDT(total_20_df), by = c("date", "rank", "song"))

# replace NAs in individual songs with 0
total_2020 <- total_2020 %>%
  rename("unique" = "individual songs")

total_2020$unique <- total_2020$unique %>%
  replace_na(0)

# clean up the 2021 data set
# first, remove the column we don't need
total_21_df <- total_21_df %>%
  select(-`spotify date`, - `chartmetric date`)

```

```

# remove duplicate songs
total_2021 <- unique(setDT(total_21_df), by = c("date", "rank", "song"))

# replace NAs in individual songs with 0
total_2021 <- total_2021 %>%
  rename("unique" = "individual songs")

total_2021$unique <- total_2021$unique %>%
  replace_na(0)

```

Merge 2020 and 2021

```

# merge the final two datasets for analysis
# rename column in 2021 to match
total_2021 <- total_2021 %>%
  rename("position" = "Position",
         "streams" = "Streams")

full_20_21 <- rbind(total_2020, total_2021)
head(full_20_21)

```

```

##   year      date rank      song      artist binary
## 1: 2020 2020-01-04   1 all i want for christmas is you mariah carey      0
## 2: 2020 2020-01-04   2 rockin around the christmas tree   brenda lee      0
## 3: 2020 2020-01-04   3                jingle bell rock   bobby helms      0
## 4: 2020 2020-01-04   4          a holly jolly christmas   burl ives      0
## 5: 2020 2020-01-04   5                circles   post malone      0
## 6: 2020 2020-01-04   6          roxanne arizona zervas      0
##   language total_words non_e_words percent unique position streams spins
## 1: english         356           0      0      1         2 9806687   196
## 2: english         119           0      0      1         4 8624106   196
## 3: english         185           0      0      1         5 8361420   196
## 4: english         159           0      0      1        15 6266059   196
## 5: english         314           0      0      0         7 7519188  7144
## 6: english         355           0      0      0         3 9691192  3701
##   radio_rank
## 1:         501
## 2:         501
## 3:         501
## 4:         501
## 5:           2
## 6:          12

```

```

# export for google sheets
write.csv(full_20_21, file = "full_20_21.csv", row.names = F)

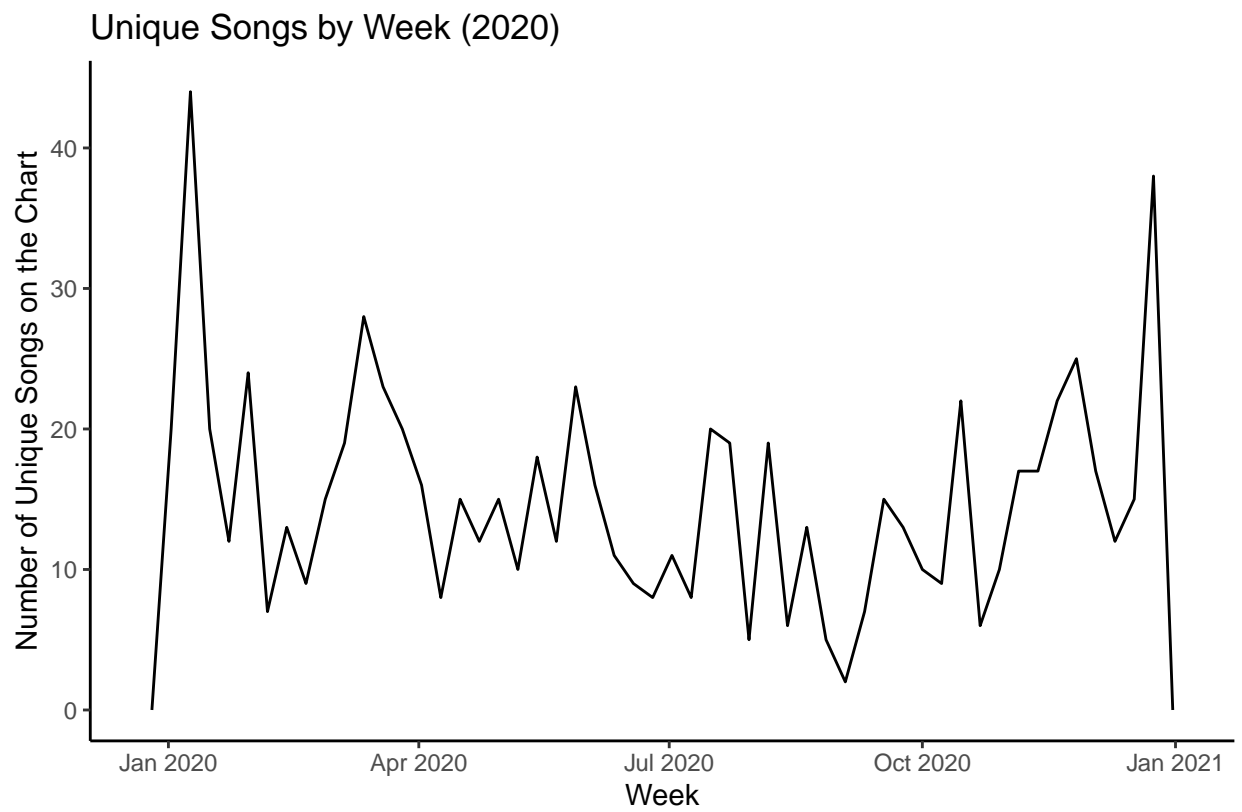
```

Exploring the Data / Results

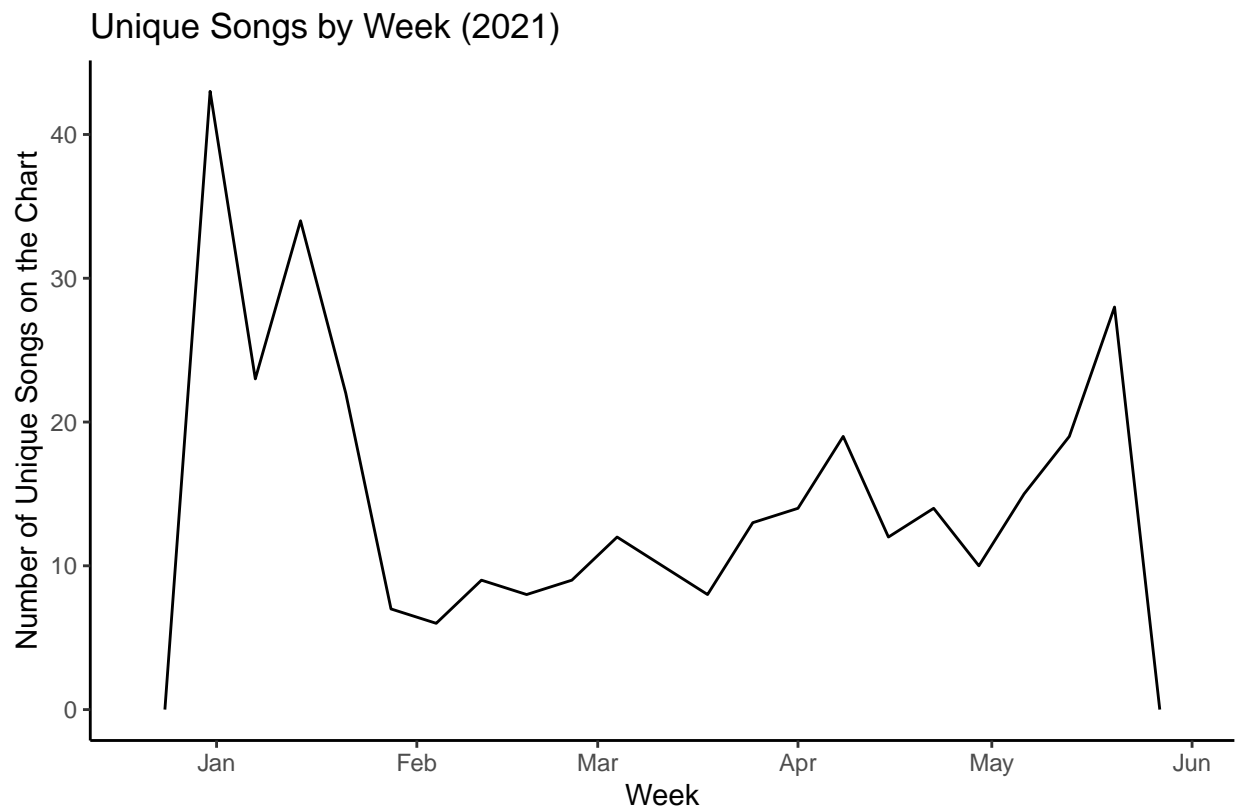
```
# how many unique songs
full_20_21 %>%
  filter(unique == 1) %>%
  count()
```

```
##           n
## 1: 1125
```

```
# unique songs by week for 2020 viz
full_20_21 %>%
  filter(unique == 1,
         year == 2020) %>%
  ggplot(mapping = aes(x = date)) +
  geom_freqpoly(binwidth = 7) +
  theme_classic() +
  labs(
    title = "Unique Songs by Week (2020)",
    x = "Week",
    y = "Number of Unique Songs on the Chart",
    caption = "Source: Billboard Hot 100"
  )
```



```
# same for 2021 viz
full_20_21 %>%
  filter(unique == 1,
         year == 2021) %>%
  ggplot(mapping = aes(x = date)) +
  geom_freqpoly(binwidth = 7) +
  theme_classic() +
  labs(
    title = "Unique Songs by Week (2021)",
    x = "Week",
    y = "Number of Unique Songs on the Chart",
    caption = "Source: Billboard Hot 100"
  )
```



Source: Billboard Hot 100

A total of 7,300 songs were analyzed, sourced from two consecutive years (2020-2021) of the Billboard Hot 100 weekly charts. Out of the 17 months analyzed for this draft, there were only 1,125 unique songs (15.41%), due to many songs staying on the Hot 100 charts for multiple weeks, either consecutively or not. Specifically, there were 790 unique songs in 2020 (15.19%), and there were 335 (15.95%) unique songs in the portion of 2021 analyzed.

```
# show most streamed songs of 2020
# add milstream for full df for easier viz
total_df <- full_20_21 %>%
  mutate(mil_stream = streams/1000000)

# add kplay for full df for easier viz
```

```

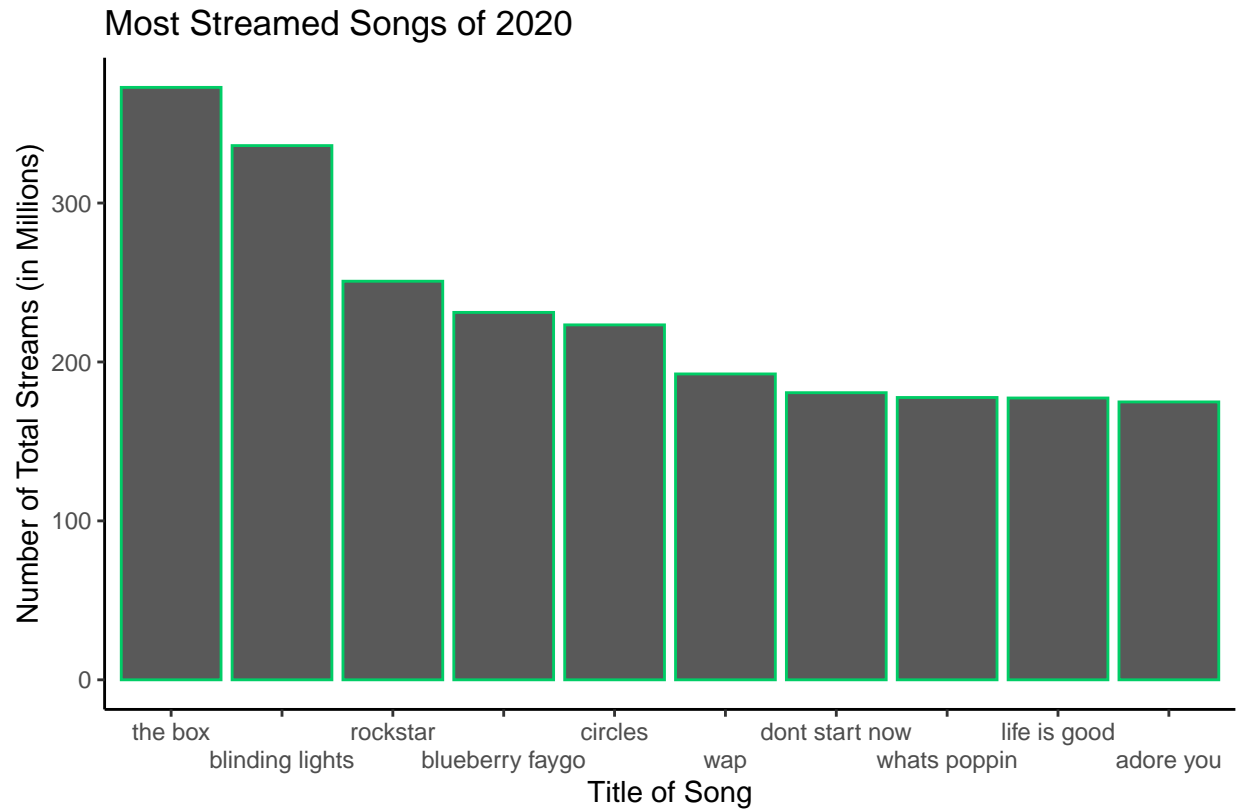
total_df <- total_df %>%
  mutate(kspins = spins/1000)

# get mean and sd for streams and plays
mean(total_df$mil_stream)
sd(total_df$mil_stream)
mean(total_df$spins)
sd(total_df$spins)

# show most streamed songs of 2020
top_streams <- total_df %>%
  filter(year == 2020) %>%
  group_by(song, artist) %>%
  summarise(total_streams = sum(mil_stream)) %>%
  arrange(desc(total_streams)) %>%
  head(n = 10)

#viz
top_streams %>%
  ggplot(mapping = aes(reorder(song, -total_streams), y = total_streams)) +
  geom_bar(stat = "identity", colour = "springgreen3") + #to match spotify color
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) + #to make it easier to read
  theme_classic() +
  labs(
    title = "Most Streamed Songs of 2020",
    x = "Title of Song",
    y = "Number of Total Streams (in Millions)",
    caption = "Source: Spotify"
  )

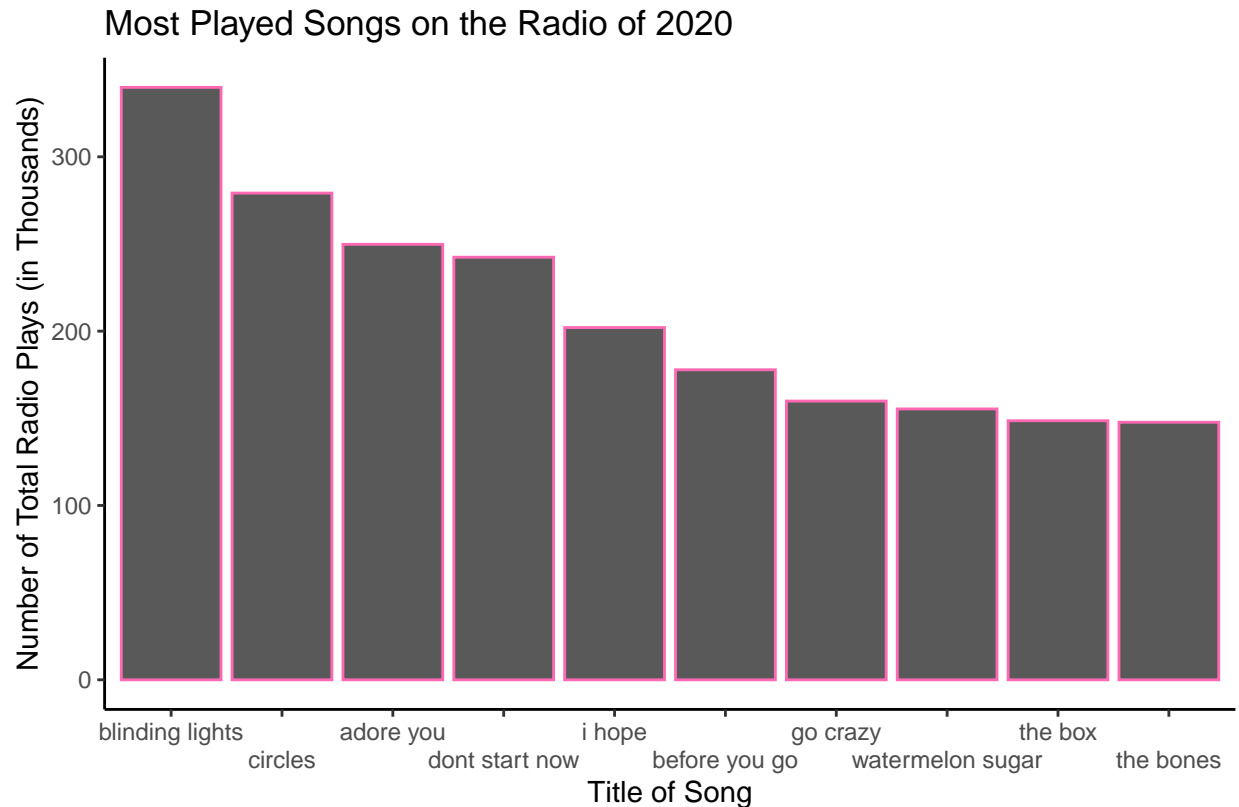
```



Source: Spotify

```
# top radio hits
top_spins <- total_df %>%
  filter(year == 2020) %>%
  group_by(song, artist) %>%
  summarise(total_spins = sum(kspins)) %>%
  arrange(desc(total_spins)) %>%
  head(n = 10)

top_spins %>%
  ggplot(mapping = aes(reorder(song, -total_spins), y = total_spins)) +
  geom_bar(stat = "identity", colour = "hotpink") + #chartmetric is also green so chose pink
  scale_x_discrete(guide = guide_axis(n.dodge=2)) +
  theme_classic() +
  labs(
    title = "Most Played Songs on the Radio of 2020",
    x = "Title of Song",
    y = "Number of Total Radio Plays (in Thousands)",
    caption = "Source: Chartmetric"
  )
```



Source: Chartmetric

The average number of Spotify streams per week per song was over 3 million ($M = 3118273.47$, $SD = 2229135.38$), while the average number of radio plays per week per song was much less, comparatively ($M = 1839.12$, $SD = 1867.48$). The top 10 most streamed songs on Spotify in 2020, as well as the top 10 played songs on the radio in 2020, include songs such as “The Box” by Roddy Rich, “Blinding Lights” by The Weeknd, “Circles” by Post Malone, and “Adore You” by Harry Styles.

Appendix

Language Coding

Below is a direct excerpt from the original project paper

The data set of popular songs sourced from the Billboard Hot 100 were then coded for language by analyzing the lyrical compositions of the songs. The lyrics for each song were taken from official sources (like Apple Music or Spotify, which is often powered by Genius) using web scraping methods. If the lyrics for a song were not available there, they were taken from one of two trusted websites (Lyrics.com or AZLyrics.com). First, in order to flag the songs that included multiple languages, a language identification function in R (package `cld3`) was used to analyze each song’s lyrics to see if it included more than one language, or a single language other than English. Then, for each song that was flagged, the total word count of the lyrics was recorded, as well as the total number of non-English words in the lyrics. Finally, the percentage of non-English lyrics (proportion of non-English words to the total words in the lyrics) was calculated. It is worth noting that the accuracy of the language identification function in R was lower than expected, so some songs that were not originally flagged as potential non-English language songs had to be re-coded and analyzed by hand, in addition to the ones that were flagged.

During the lyric analysis, proper nouns (like names and places) were not considered non-English, unless they were the foreign-language equivalent (example: Barcelona was not marked, but Estados Unidos was marked as non-English). Additionally, names of designers and brands were not included, although it is interesting to note that they appeared frequently in the lyric corpus (examples: Gucci, Prada, Louis Vuitton, Lamborghini, etc.). Finally, words or slang that were originally from another language but have been inculcated into the daily vernacular of English speakers in the United States (examples: chardonnay or piñata) were not included in the non-English language count.

Alternatively, words that might be considered well-known in some sectors of the United States, but have direct replacements in the English language were counted as non-English (example: cerveza, which is the Spanish word for beer). In a similar fashion, words that could be considered slang here but were not deemed to be ubiquitous and are sourced from or directly borrowed from a non-English language were counted as non-English (example: fuego, which is the Spanish word for fire, which itself is a current slang term in English). Finally, words that originated in English but were being spoken in the middle of a non-English phrase, possibly with an accent to make them understandable to native speakers, were considered to be non-English.

While Billboard itself does not have an official definition of what songs are considered “predominantly non-English”, following the examples of other media giants such as the Grammy’s and the Oscar’s, the songs were additionally coded in a binary based on whether they were majority (50% or more) non-English (1) or not (0) (Oscars, 2021, p. 19; Grammys, 2020, p. 53). Finally, the songs were coded for the language that made up the majority (50% or more) of their lyrics, like Korean, Spanish, English, etc.