Musical Insights: Analyzing the Stories in Song Data

AUTHOR
Jane

Introduction

Hey music enthusiasts! 🎶 Ever wondered how your favorite artists' music has evolved over the years or what stories their lyrics tell? Join us on a lyrical adventure as we explore the themes, and emotions behind the music of artists like Beyonce, BTS, Coldplay, Ed Sheeran, Justin Bieber, Lady Gaga, Selena Gomez, Taylor Swift, and more. In this expansive exploration, we are analyzing songs from a diverse pool of 20 artists, encompassing over 5700 songs. Join us in unraveling the rich tapestry of their collective musical expressions.

Research Questions

These questions were chosen to understand the emotional and thematic nuances that define the musical journey for specific artists.

1. Are the majority of songs in the dataset positive or negative?

2. Is there a distinct group of artists who predominantly sing positive or negative songs?

3. How has the sentiment of music evolved over time?

4. What are the prevalent themes covered in the songs, and how do artists differ from each other in terms of lyrical content?

Data Description

The datasets are sourced from Kaggle and you can find it here. Each row in the "combined_data" represents a unique song, and each column holds specific details about that song. Here are the key columns in our dataset:

1. **Artist:** The name of the artist who performed the song.
2. **Title:** The title of the song.

3. **Album:** The album to which the song belongs.
4. **Year:** The year when the song was released.
5. **Date:** The specific release date of the song.
6. **Lyric:** The complete lyrics of the song.

```
# Loading all the libraries we need
library(dplyr)
library(ggplot2)
library(tidytext)
library(wordcloud)
library(tidyverse)
library(topicmodels)
library(igraph)
library(ggraph)
```

```
# Read data from files for all artists
Beyonce <- read.csv("Data/Beyonce.csv")
BTS <- read.csv("Data/BTS.csv")
ColdPlay <- read.csv("Data/ColdPlay.csv")
EdSheeran <- read.csv("Data/EdSheeran.csv")
JustinBieber <- read.csv("Data/JustinBieber.csv")
LadyGaga <- read.csv("Data/LadyGaga.csv")
SelenaGomez <- read.csv("Data/SelenaGomez.csv")
TaylorSwift <- read.csv("Data/TaylorSwift.csv")
BillieEilish <- read.csv("Data/BillieEilish.csv")
CardiB <- read.csv("Data/CardiB.csv")
CharliePuth <- read.csv("Data/CharliePuth.csv")
Drake <- read.csv("Data/Drake.csv")
Eminem <- read.csv("Data/Eminem.csv")
DuaLipa <- read.csv("Data/DuaLipa.csv")
KatyPerry <- read.csv("Data/KatyPerry.csv")
Khalid <- read.csv("Data/Khalid.csv")
Maroon5 <- read.csv("Data/Maroon5.csv")
NickiMinaj <- read.csv("Data/NickiMinaj.csv")
PostMalone <- read.csv("Data/PostMalone.csv")
Rihanna <- read.csv("Data/Rihanna.csv")

# Combine Data
combined_data <- bind_rows(
  Beyonce, BTS, ColdPlay,
  EdSheeran, JustinBieber, LadyGaga,
  SelenaGomez, TaylorSwift, BillieEilish,
  CardiB, CharliePuth, Drake,
  Eminem, DuaLipa, KatyPerry,
  Khalid, Maroon5, NickiMinaj,
  PostMalone, Rihanna)

combined_data <- combined_data %>%
  rename(No. = X)
```

Summary Statistics

Let's get to know our data better.

```
# Excluded the 'No.' column from the summary
summary(combined_data[, !colnames(combined_data) %in% c("No.")])
```

```
    Artist             Title              Album                Year     
 Length:5719        Length:5719        Length:5719        Min.   :   1  
 Class :character   Class :character   Class :character   1st Qu.:2010  
 Mode  :character   Mode  :character   Mode  :character   Median :2014  
                                                          Mean   :2012  
                                                          3rd Qu.:2018  
                                                          Max.   :2022  
                                                          NA's   :1686  
     Date              Lyric          
 Length:5719        Length:5719       
 Class :character   Class :character  
 Mode  :character   Mode  :character  
```

- The dataset comprises 5,719 rows.
- Artist, Title, Album, Date, and Lyric are of character type.
- The "Year" column exhibits some data inconsistencies.

Let's investigate the "Year" column further.

```
# View unique values in the 'Year' column
unique_years <- sort(unique(combined_data$Year))
unique_years
```

```
 [1]    1 1729 1982 1988 1990 1993 1996 1997 1998 1999 2000 2001 2002 2003 2004
[16] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
[31] 2020 2021 2022
```

We've identified inconsistencies in the 'Year' column, including values such as '1' and a bunch of nulls. Due to the large amount of data associated with these irregularities, outright removal might result in a significant loss of information. But we can note this point and deal with it later if we need to.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) involves the preliminary examination of data to uncover patterns, trends, and insights, providing a foundation for more in-depth analysis.

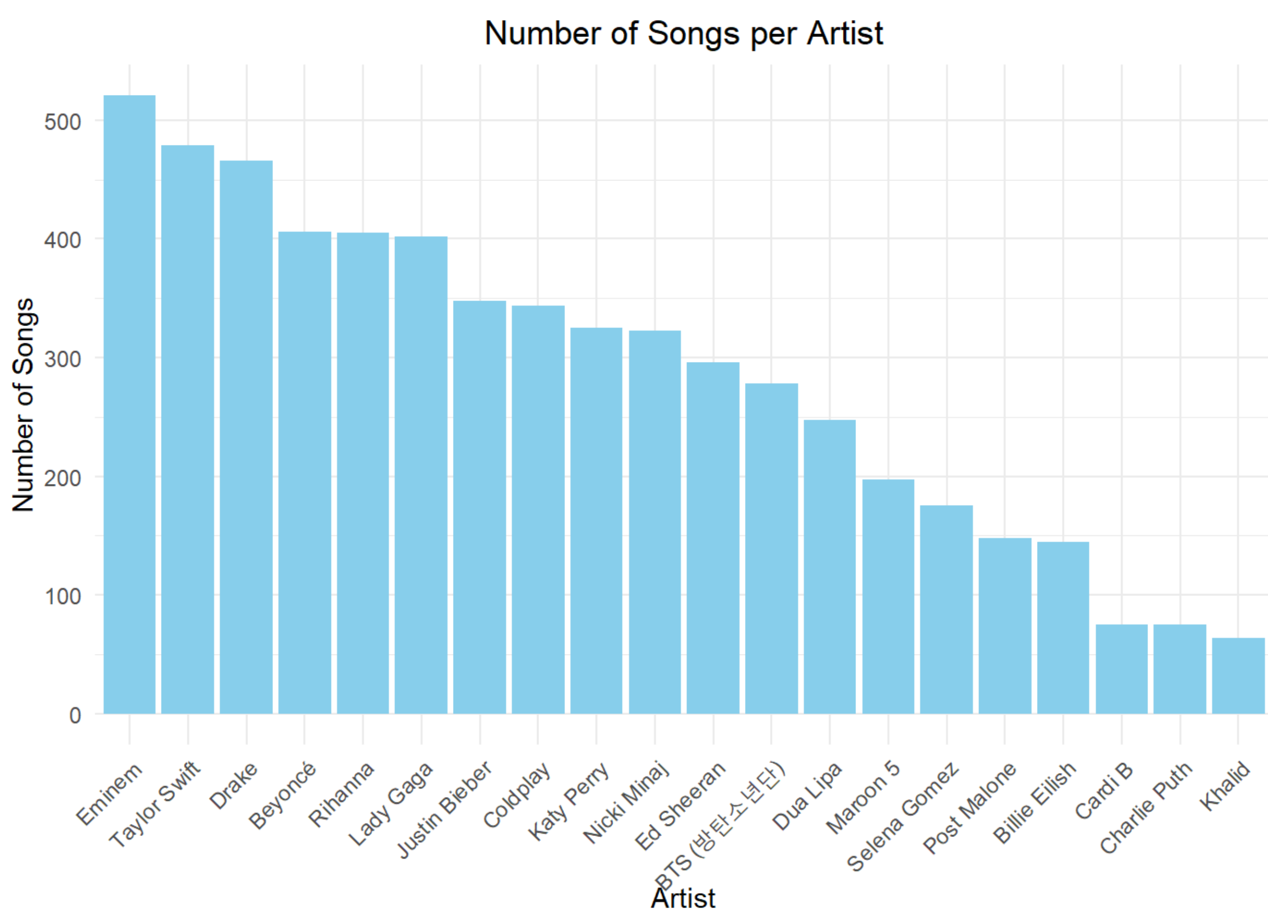
1. Checking for duplicates in the dataset

```
# Check for duplicate records
duplicates <- combined_data[duplicated(combined_data), ]
cat("We have no duplicate rows in our dataset")
```

```
We have no duplicate rows in our dataset
```

2. Count the number of songs per artist

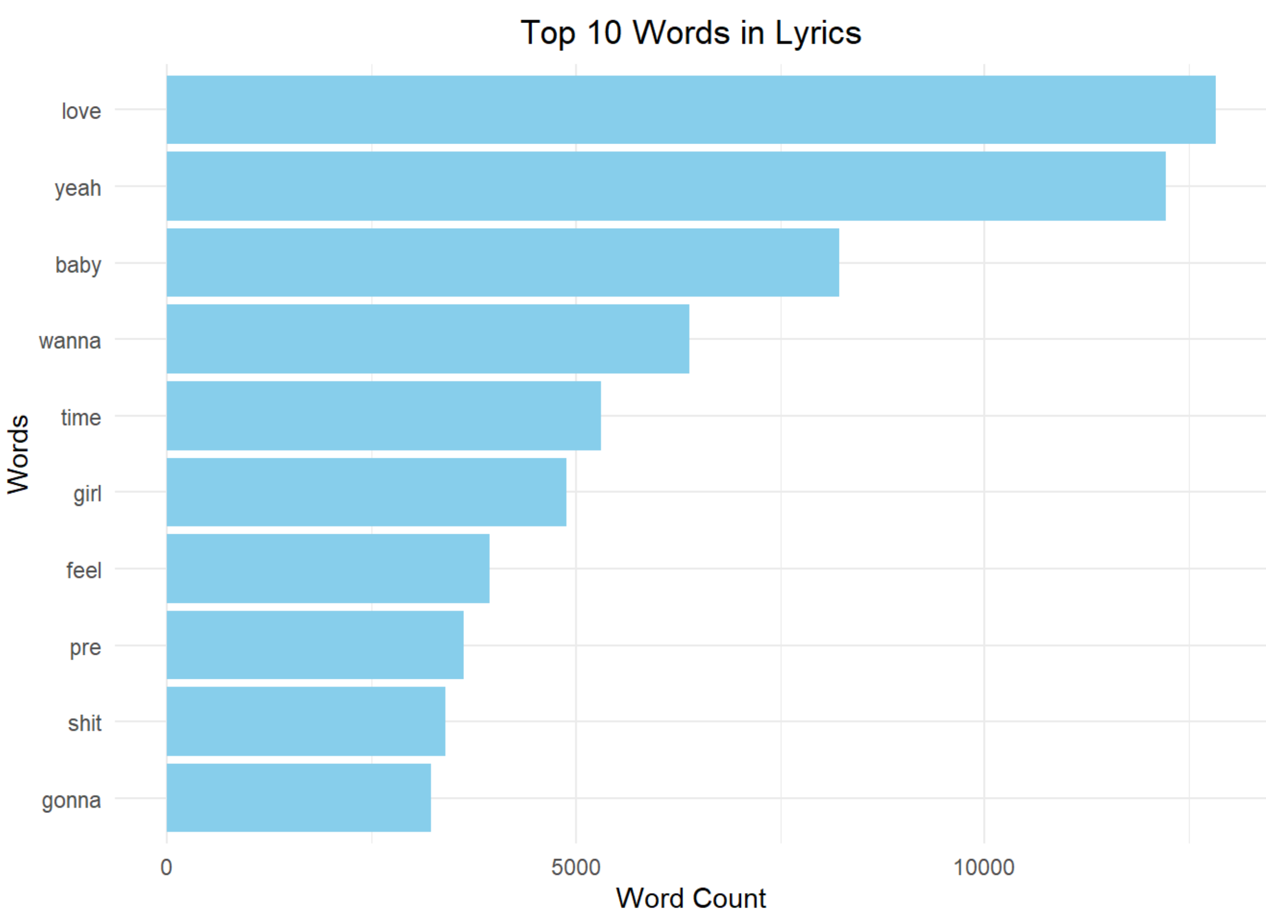
```
# Count the number of songs per artist
song_counts <- table(combined_data$Artist)

# Create a data frame from the table
song_counts_df <- data.frame(Artist = names(song_counts), Count = as.numeric(song_counts))


ggplot(song_counts_df, aes(x = reorder(Artist, -Count), y = Count)) +
  geom_col(fill = "skyblue") +
  labs(title = 'Number of Songs per Artist', x = 'Artist', y = 'Number of Songs') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



3. Let's see the top words used by these artists in their lyrics

```
# Tokenize
word_counts <- combined_data %>%
```

```r
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  top_n(n = 10, wt = n)

# bar chart
ggplot(word_counts, aes(x = n, y = reorder(word, n))) +
  geom_col(fill = "skyblue") +
  ggtitle("Top 10 Words in Lyrics") +
  labs(x = "Word Count", y = "Words") +
  theme_minimal() +
  theme(legend.position = 'none',
        plot.title = element_text(hjust = 0.5))
```



4. Let’s see the top 5 words used by each Artist in the Songs

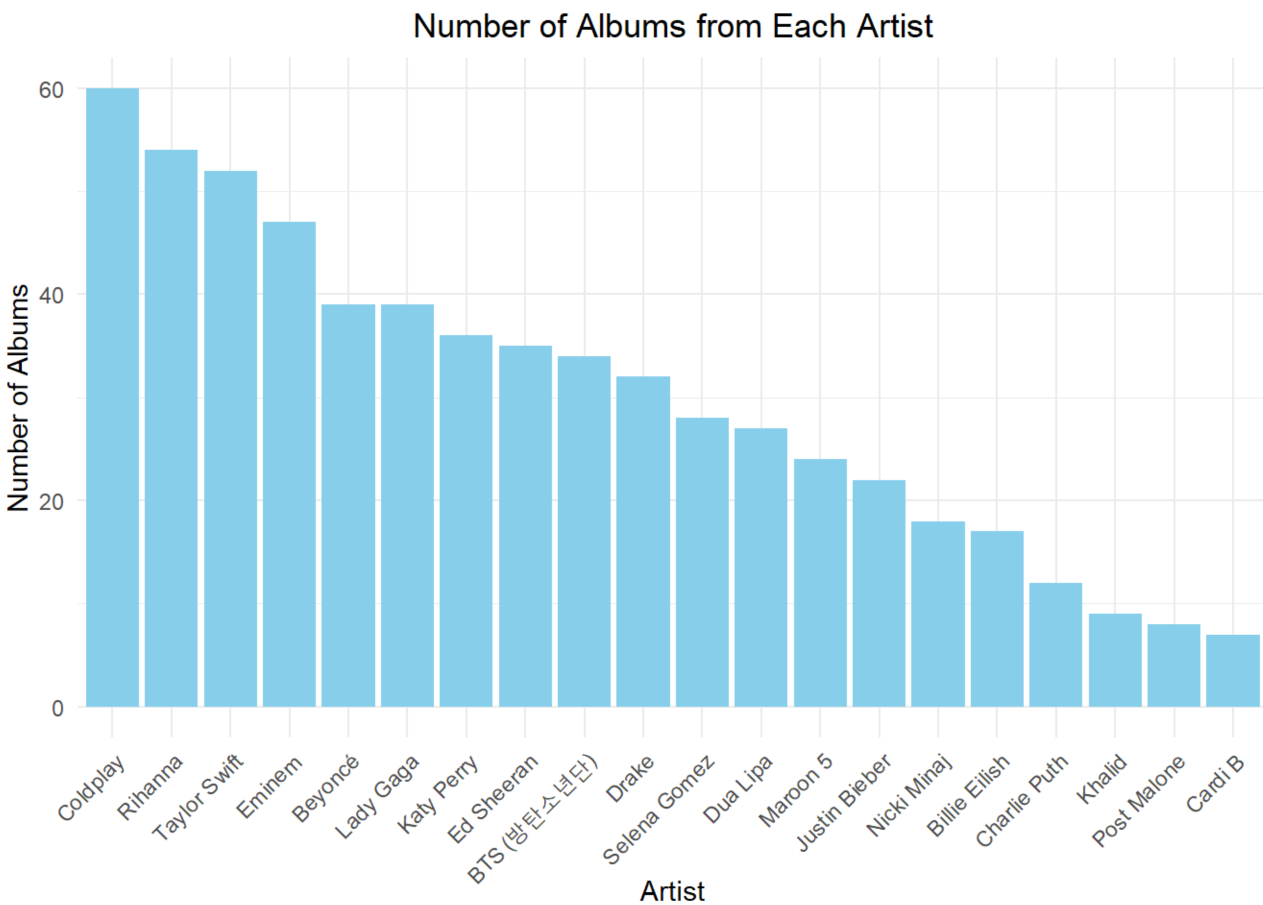
```r
new_stops <- tibble(word = c(
"beyoncé", "charlie","drake","ed",
"eminem", "em", "bieber", "katy", "pre",
"minaj", "rihanna", "justin","perry"))


add_stops <- stop_words %>%
   bind_rows(new_stops)

# Tokenize
```

```
word_counts <- combined_data %>%
  unnest_tokens(word, Lyric) %>%
  anti_join(add_stops) %>%
  count(Artist, word, sort = TRUE)

# Get the top 5 words for each artist
top_words_by_artist <- word_counts %>%
  group_by(Artist) %>%
  summarise(words = list(head(word, 5)),
            frequency = list(head(n, 5)))


top_words_by_artist <- top_words_by_artist %>%
  unnest(c(words, frequency))

head (top_words_by_artist)
```

```
# A tibble: 6 × 3
  Artist            words     frequency
  <chr>             <chr>         <int>
1 BTS (방탄소년단) jungkook       857
2 BTS (방탄소년단) yeah           768
3 BTS (방탄소년단) jimin          712
4 BTS (방탄소년단) の             672
5 BTS (방탄소년단) love           667
6 Beyoncé           love          1395
```

5. Count the number of unique albums by each artist

```
# Group by artist
albums_count <- combined_data %>%
  group_by(Artist) %>%
  summarize(Albums = n_distinct(Album))

# bar chart
ggplot(albums_count, aes(x = reorder(Artist, -Albums), y = Albums)) +
  geom_col(fill = "skyblue") +
  ggtitle("Number of Albums from Each Artist") +
  labs(x = "Artist", y = "Number of Albums") +
  theme_minimal() +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



6. Average number of words in the songs

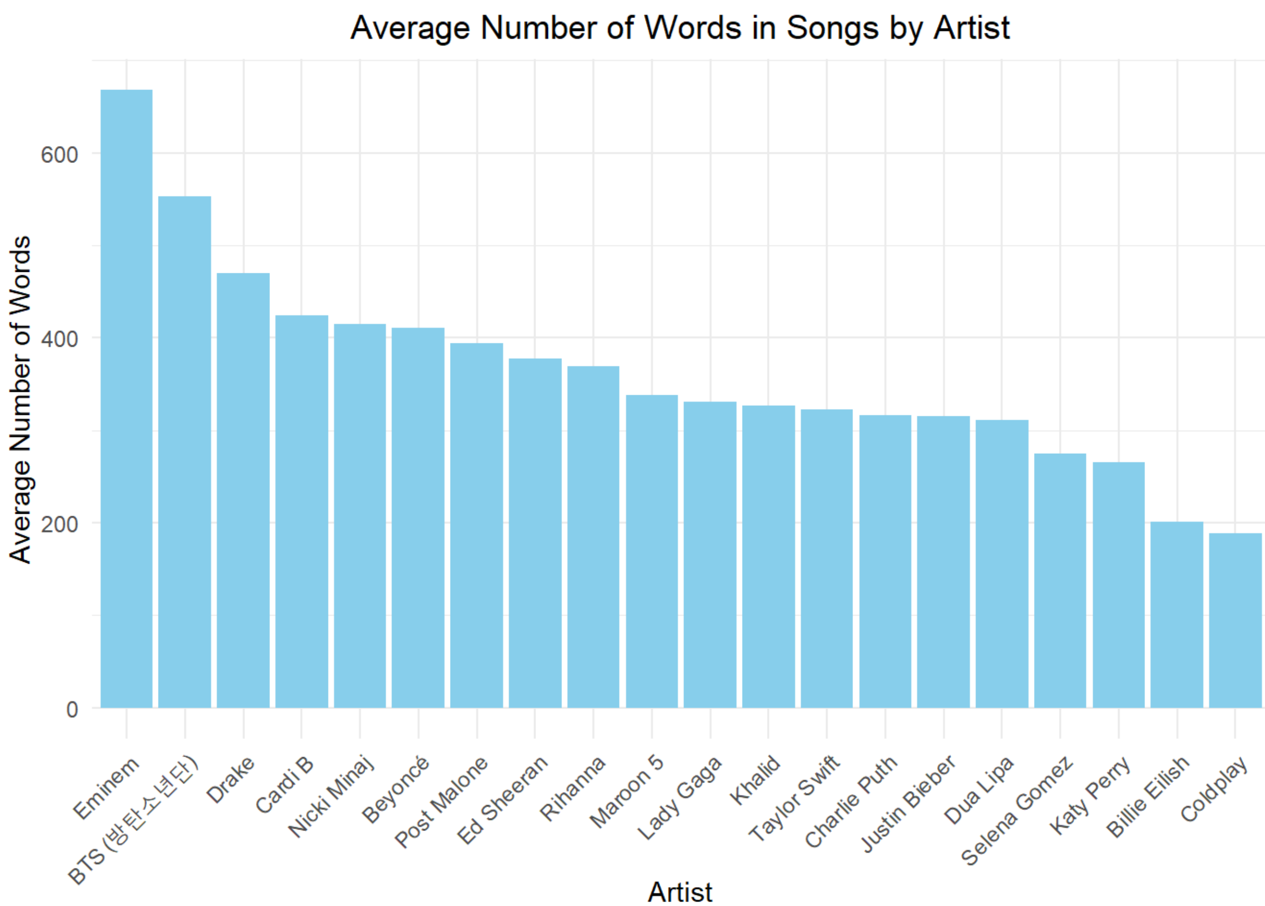
```
average_words <- combined_data %>%
  unnest_tokens(word, Lyric) %>%
  anti_join(stop_words) %>%
  count(Title) %>%
  summarise(average_words = mean(n))

average_words
```

```
  average_words
1      145.5095
```

7. Average number of words in songs By Artist

```
word_counts <- combined_data %>%
  unnest_tokens(input = Lyric, output = word) %>%
  group_by(Artist, Title) %>%
  summarize(NumWords = n())

# Calculate the average number of words
avg_words <- word_counts %>%
  summarise(AvgWords = mean(NumWords))

# Bar Graph
ggplot(avg_words, aes(x = reorder(Artist, -AvgWords), y = AvgWords)) +
```

```
  geom_col(fill = "skyblue") +
  ggtitle("Average Number of Words in Songs by Artist") +
  labs(x = "Artist", y = "Average Number of Words") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



EDA Summary:

About Artists:

- Eminem has the most songs in the dataset (506 songs), while Khalid has the least (64 songs).
- Coldplay has the most albums, showing they've made a bunch of different music. On the other hand, Cardi B has the fewest albums.

Common Words:

- The words 'Love,' 'Yeah,' and 'Baby' pop up a lot in the songs, suggesting that these feelings and expressions are common themes.

Word Count in Songs:

- On average, there are about 145 words in a song. This helps us understand how long or short the lyrics tend to be.

Different Artists, Different Words:

- Eminem's songs, on average, have a lot of words (around 668), showing he likes to tell detailed stories through his lyrics.
- Coldplay, on the other hand, uses fewer words on average (about 188), indicating a simpler or more musical approach

After our exploratory data analysis, we've gained valuable insights into our dataset. Now, let's proceed to answer the research questions armed with a deeper understanding of the patterns and nuances within the data.

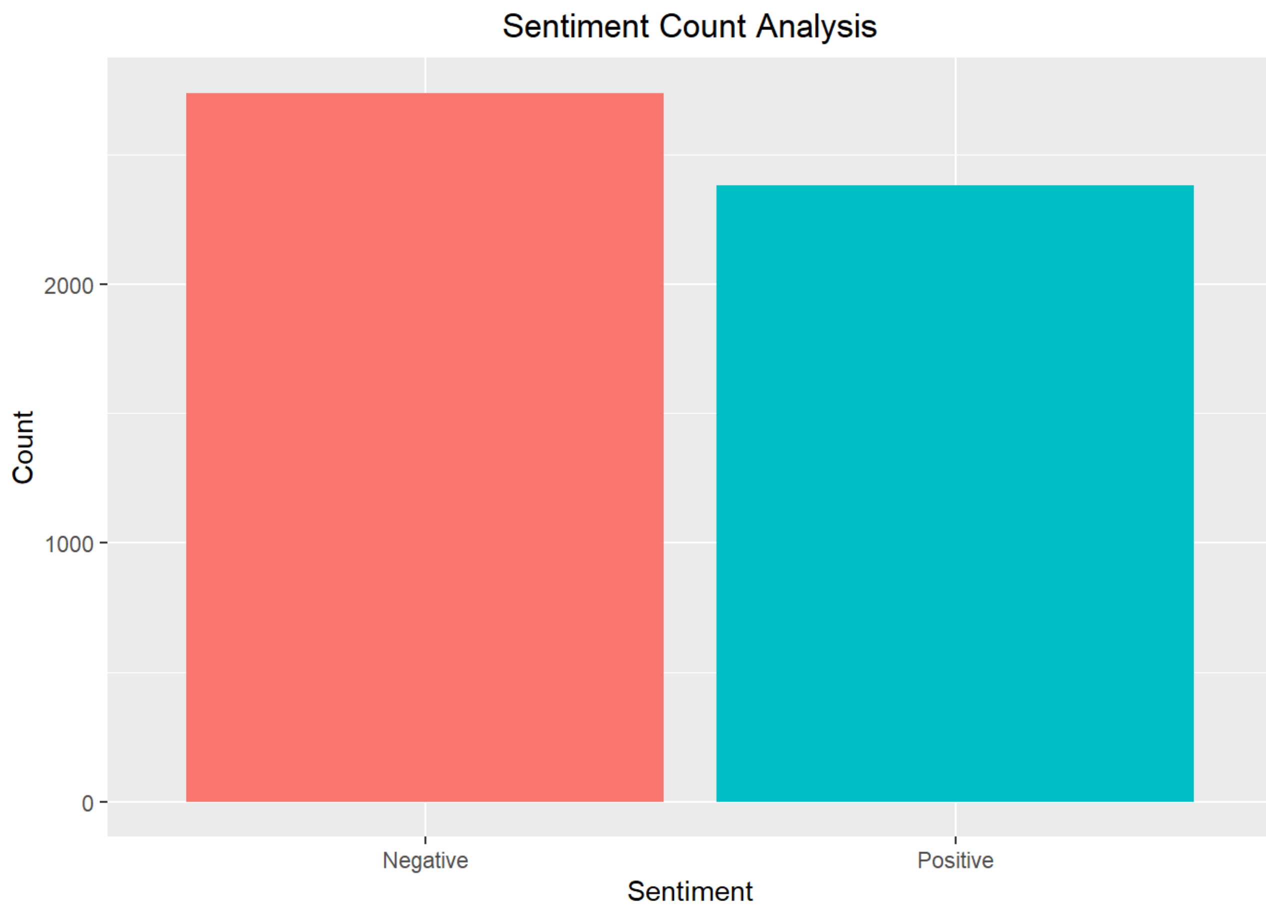
Text Mining Analysis

We will use **Sentiment Analysis** to answer the first three research questions and **Topic Modeling** to answer the 4th research question.

Let's work for research question 1 first.

Let's start by finding out how many positive or negative songs we have in our dataset.

```
sentiment_analysis_songs <- combined_data %>%
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(Title) %>%
  summarise(total_value = sum(value))

# Filter the dataset for positive and negative sentiments
positive_sentiments <- sentiment_analysis_songs %>%
  filter(total_value >= 0)
negative_sentiments <- sentiment_analysis_songs %>%
  filter(total_value < 0)

# Calculate the total counts
total_positive_count <- nrow(positive_sentiments)
total_negative_count <- nrow(negative_sentiments)

sentiment_counts <- data.frame(
  Sentiment = c("Positive", "Negative"),
  Count = c(total_positive_count, total_negative_count)
)

# Bar chart for sentiment counts
ggplot(sentiment_counts, aes(x = Sentiment, y = Count, fill = Sentiment)) +
  geom_bar(stat = "identity") +
  ggtitle("Sentiment Count Analysis") +
  theme(legend.position = 'none', plot.title = element_text(hjust = 0.5))
```



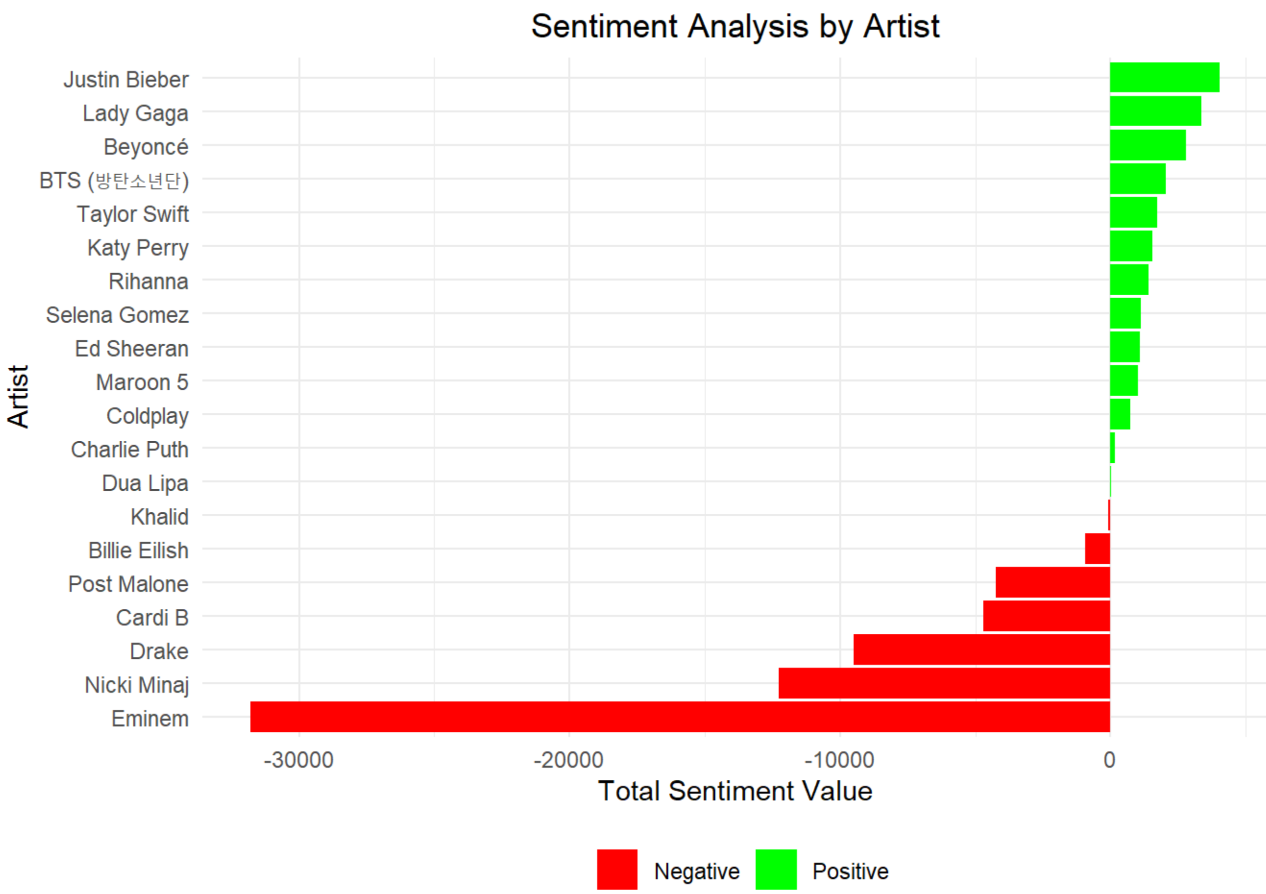
Let's see what are the top words in our "Positive" and "Negative" songs using Word Clouds.

```
# Positive Word Cloud
combined_data %>%
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn")) %>%
  filter(value >= 0) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```


```
# Negative Word Cloud
combined_data %>%
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn")) %>%
  filter(value <= 0) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

Now let's answer Research question 2 - Sentiment Analysis by Artist

```
# Afinn sentiment analysis
sentiment_analysis_artist <- combined_data %>%
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(Artist) %>%
  summarise(total_value = sum(value))

# Bar chart
ggplot(sentiment_analysis_artist, aes(x = total_value,
                                      y = reorder(Artist, total_value),
                                      fill = ifelse(total_value >= 0,
                                                    "Positive","Negative"))) +
  geom_bar(stat = "identity") +
  ggtitle("Sentiment Analysis by Artist") +
  labs(x = "Total Sentiment Value", y = "Artist") +
  theme_minimal() +
  theme(legend.title = element_blank(), legend.position = "bottom",
        plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("Positive" = "green", "Negative" = "red"))
```



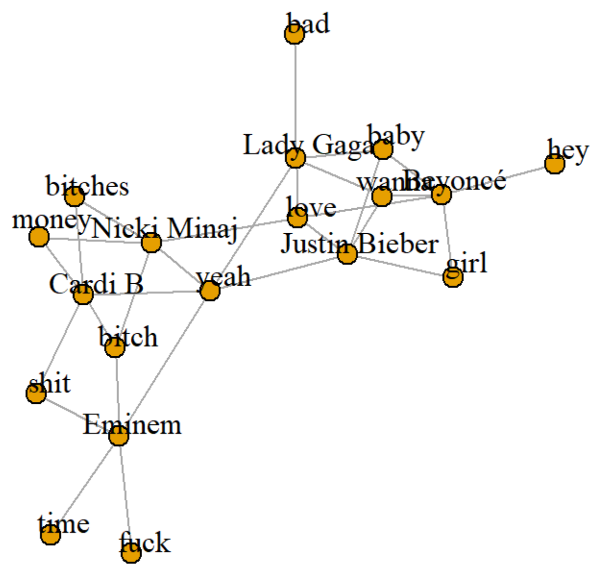
Eminem has a notably high negative score, while Justin Bieber takes the lead for singing the most positive songs.

I'm curious to explore the top words associated with these artists and how they interconnect. Let's construct a **Network Diagram**.

```
new_stops <- tibble(word = c(
  "beyoncé", "charlie", "drake", "ed", "cardi" ,
  "eminem", "em", "bieber", "katy", "pre", "minaj",
  "rihanna", "justin", "perry", "05", "nicki", "gaga"))

add_stops <- stop_words %>% bind_rows(new_stops)

word_counts <- combined_data %>%
  unnest_tokens(word, Lyric) %>%
  anti_join(add_stops) %>%
  count(Artist, word, sort = TRUE)

#We will build the network diagram for selected artists so that its readable
# Filter data by Selected Artists
selected_artists <- c("Justin Bieber", "Lady Gaga", "Beyoncé",
                      "Eminem", "Cardi B", "Nicki Minaj")

filtered_data <- word_counts %>%
  filter(Artist %in% selected_artists)

# Get the top 5 words for each artist
top_words_by_artist <- filtered_data %>%
```

```r
  group_by(Artist) %>%
  summarise(words = list(head(word, 5)), # saving top 5 words for each artist
            frequency = list(head(n, 5)))

# Saving the list of words and their frequencies
top_words_by_artist <- top_words_by_artist %>%
  unnest(c(words, frequency))

# Network graph
graph_data <- top_words_by_artist %>%
  group_by(Artist, words) %>%
  summarise(frequency = sum(frequency))

word_network <- graph_from_data_frame(graph_data, directed = FALSE)

set.seed(3123)
plot(
  word_network,
  vertex.label.dist = 1,
  vertex.label.cex = 1,  # Increased size of labels
  main = "Network Diagram of top 5 Words by Artist",
  layout = layout_with_fr(word_network),
  vertex.size = 8,  # Increased size of nodes
  vertex.label.color = "black"  # Label color
  # Edge color and width removed
)
```

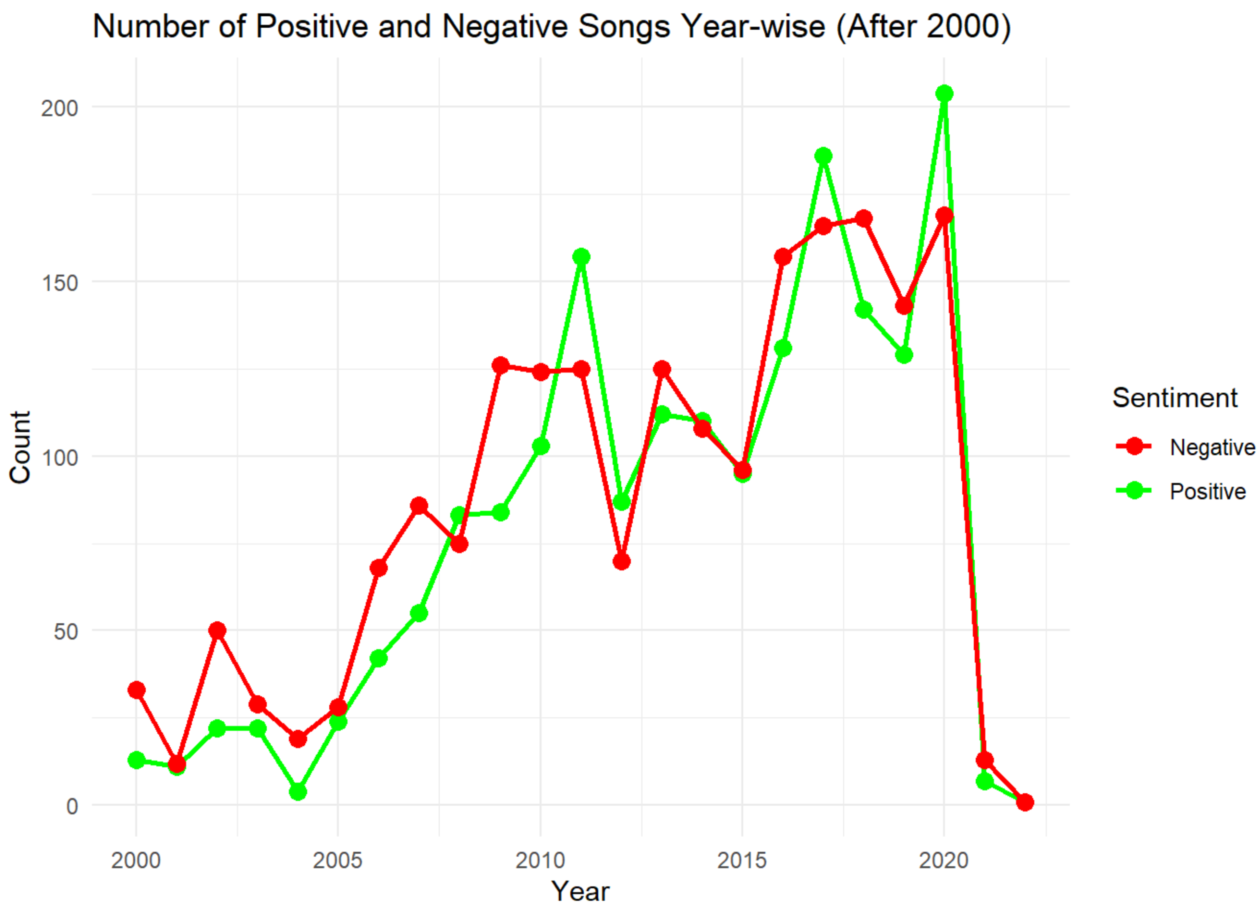
Network Diagram of top 5 Words by Artist



Now, let's address Research Question 3 by employing sentiment analysis.

We aim to find the sentiment of songs year-wise.

```
# Filter data for years greater than 2000
combined_data_filtered <- combined_data %>%
  filter(!is.na(Year) & Year >= 2000)

# Calculate sentiment values using afinn sentiment analysis
sentiment_analysis_songs <- combined_data_filtered %>%
  unnest_tokens(input = Lyric, output = word) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(Title, Year) %>%
  summarise(total_value = sum(value))

# Filter the dataset for positive and negative sentiments
positive_sentiments <- sentiment_analysis_songs %>%
  filter(total_value >= 0)
negative_sentiments <- sentiment_analysis_songs %>%
  filter(total_value < 0)

# Calculate the total counts for positive and negative sentiments year-wise
positive_counts_year <- positive_sentiments %>%
  group_by(Year) %>%
  summarise(PositiveCount = n())

negative_counts_year <- negative_sentiments %>%
  group_by(Year) %>%
  summarise(NegativeCount = n())

# Merge the positive and negative counts based on the 'Year' column
combined_counts_year <- merge(positive_counts_year, negative_counts_year,
                              by = "Year", all = TRUE)

# Fill missing values with 0
combined_counts_year[is.na(combined_counts_year)] <- 0

# Plotting with dots
ggplot(combined_counts_year, aes(x = Year)) +
  geom_line(aes(y = PositiveCount, color = "Positive"), size = 1) +
  geom_point(aes(y = PositiveCount, color = "Positive"), size = 3) +
  geom_line(aes(y = NegativeCount, color = "Negative"), size = 1) +
  geom_point(aes(y = NegativeCount, color = "Negative"), size = 3) +
  scale_color_manual(values = c("Positive" = "green", "Negative" = "red"),
                     name = "Sentiment") +
  ggtitle("Number of Positive and Negative Songs Year-wise (After 2000)") +
  labs(y = "Count", x = "Year") +
  theme_minimal()
```



Advancing to answer Research Question 4, we'll harness the power of **Topic Modeling** to unveil the themes covered in these songs.

```
# Creating a document
group_by_artitst_and_title <- combined_data %>%
  group_by(Artist, Title) %>%
  ungroup() %>%
  unite(document, Artist, Title) # Each song is now a document
```

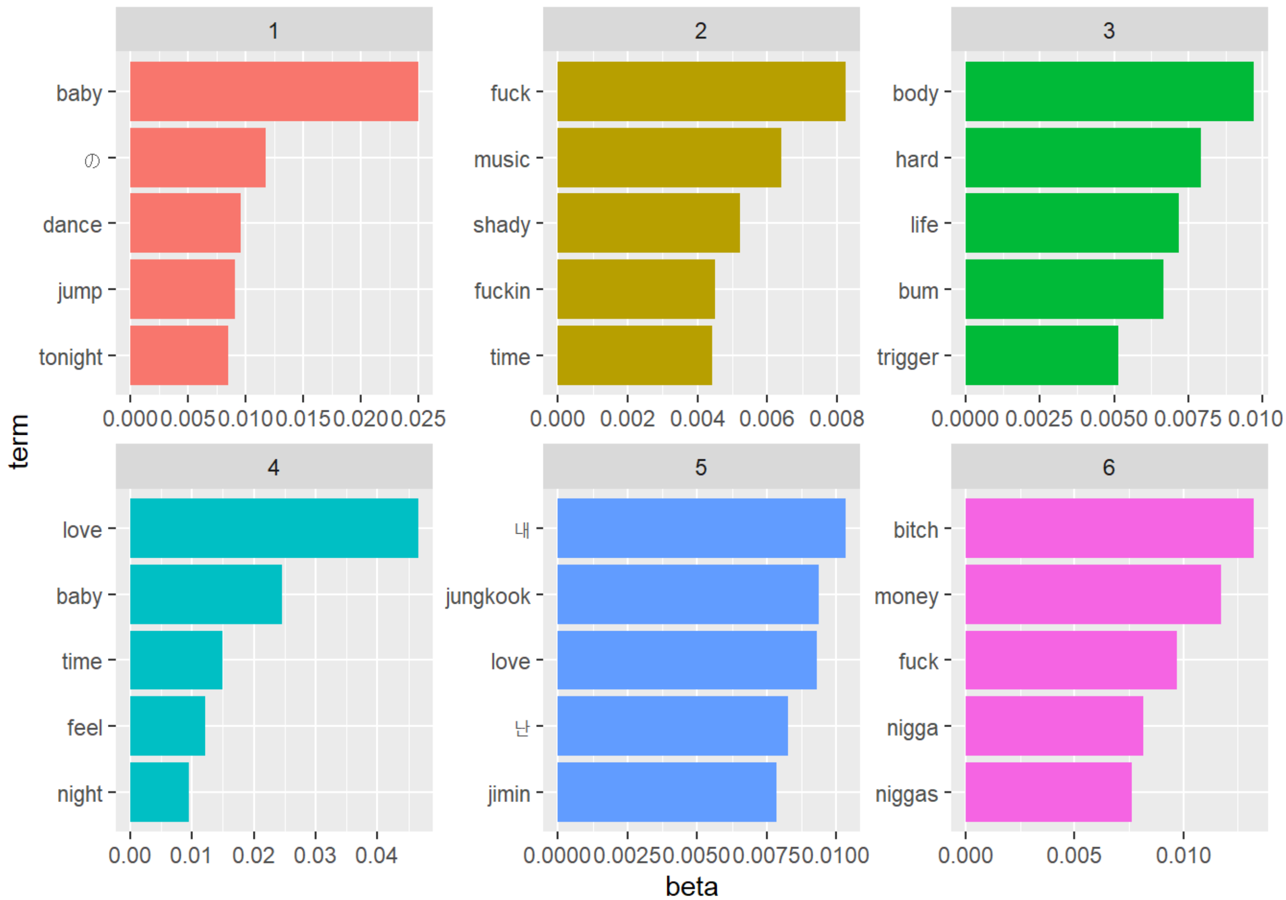
```
new_stops2 <- tibble(word = c(
  "beyoncé", "charlie", "drake", "ed", "eminem", "em", "bieber",
  "katy", "pre", "minaj", "rihanna", "justin", "perry", "05", "nicki",
  "gaga" , "yeah" , "wanna", "gonna" , "hey" , "ohh", "run" , "na" ,
  "song",  "la" , "gotta" , "bring" , "ya" , "yo" , "ooh" , "ay" ,
  "ayy" , "hold", "eh" , "boy" , "girl" , "im" , "de" , "shit" ,
  "hold" , "cut" , "an" , "bad" ,"whoa" , "da" , "cardi" ,"break" ,
  "dont" , "post" , "call" , "uh" , "rm" , "released" , "stop" , "gon"))

add_stops2 <- stop_words %>% bind_rows(new_stops2)

# tokenize data
word_counts <- group_by_artitst_and_title %>%
  unnest_tokens(output = word ,input = Lyric) %>%
  anti_join(add_stops2) %>% #remove stop words
  count(document,word,sort=T)
```

```r
# Make DTM
song_dtm <- word_counts %>%
  cast_dtm(document = document,
           term = word,
           value = n )
```

```r
#Trying 6 topics
model1 <- LDA(song_dtm, k = 6, control = list(seed = 1234))
```

```r
# Grab the topic-word probabilities
topics <- tidy(model1,matrix = "beta")

top_terms <- topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```r
topics <-  top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(x = beta, y = term, fill = factor(topic) )) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~topic  , scales = "free" ) +
  scale_y_reordered()

topics
```



Let’s name our topics for clarity and distinction.

For simplicity, we’ll set aside topics 1 and 3, as their prominence is not evident in our gamma distributions.

Now, let’s affix meaningful labels to topics 2, 4, 5 and 6.

Let’s call Topic 2 and 6 as “Raw”. Naming topic 4 as “Love and Emotion” theme and Topic 5 as “Korean”

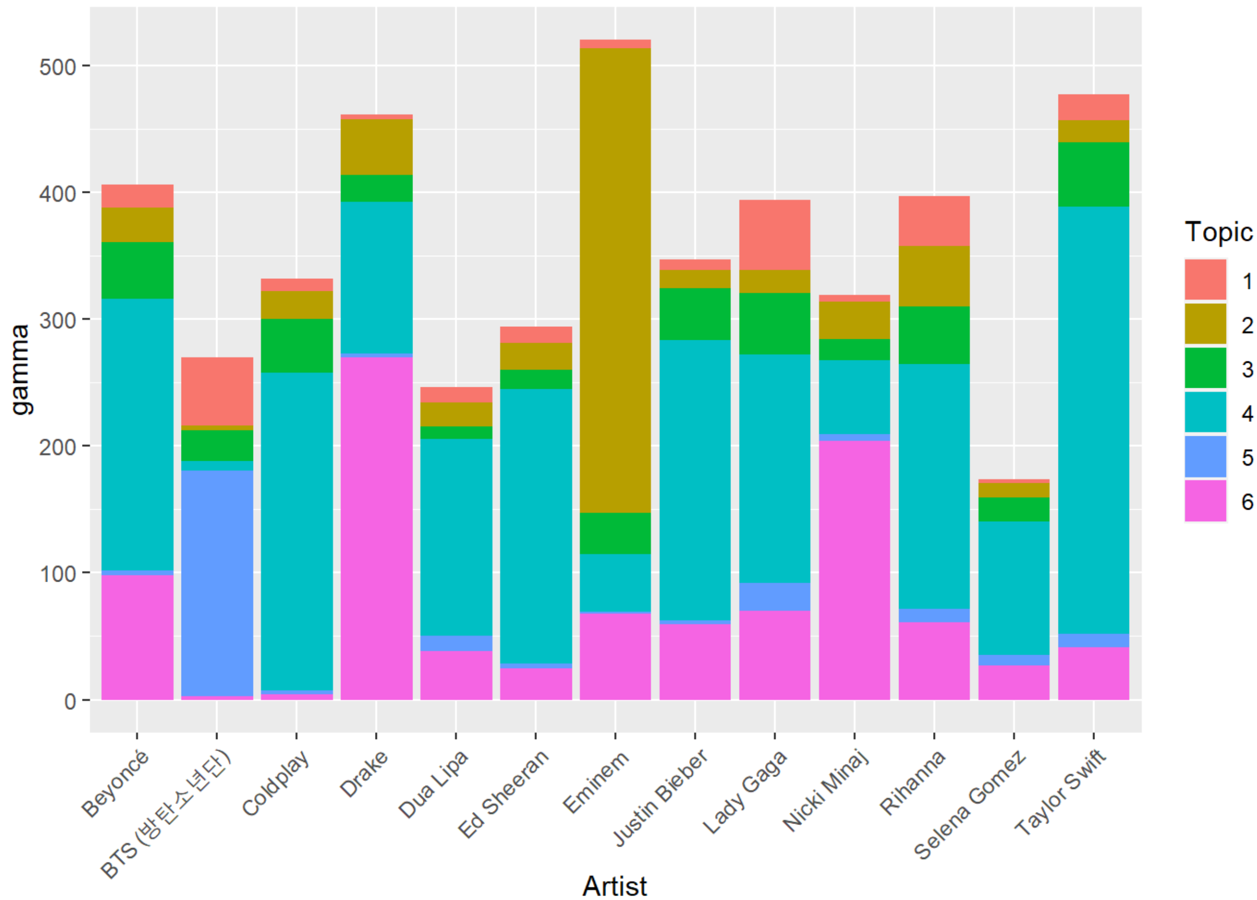
Now, let’s examine how these themes resonate across different artists by inspecting the gamma distribution for each artist.

```
song_gamma <- tidy(model1, matrix = "gamma")
```

```
song_gamma <- song_gamma %>%
  separate(document, c("Artist", "Title"),
           sep = "_", convert = TRUE)
```

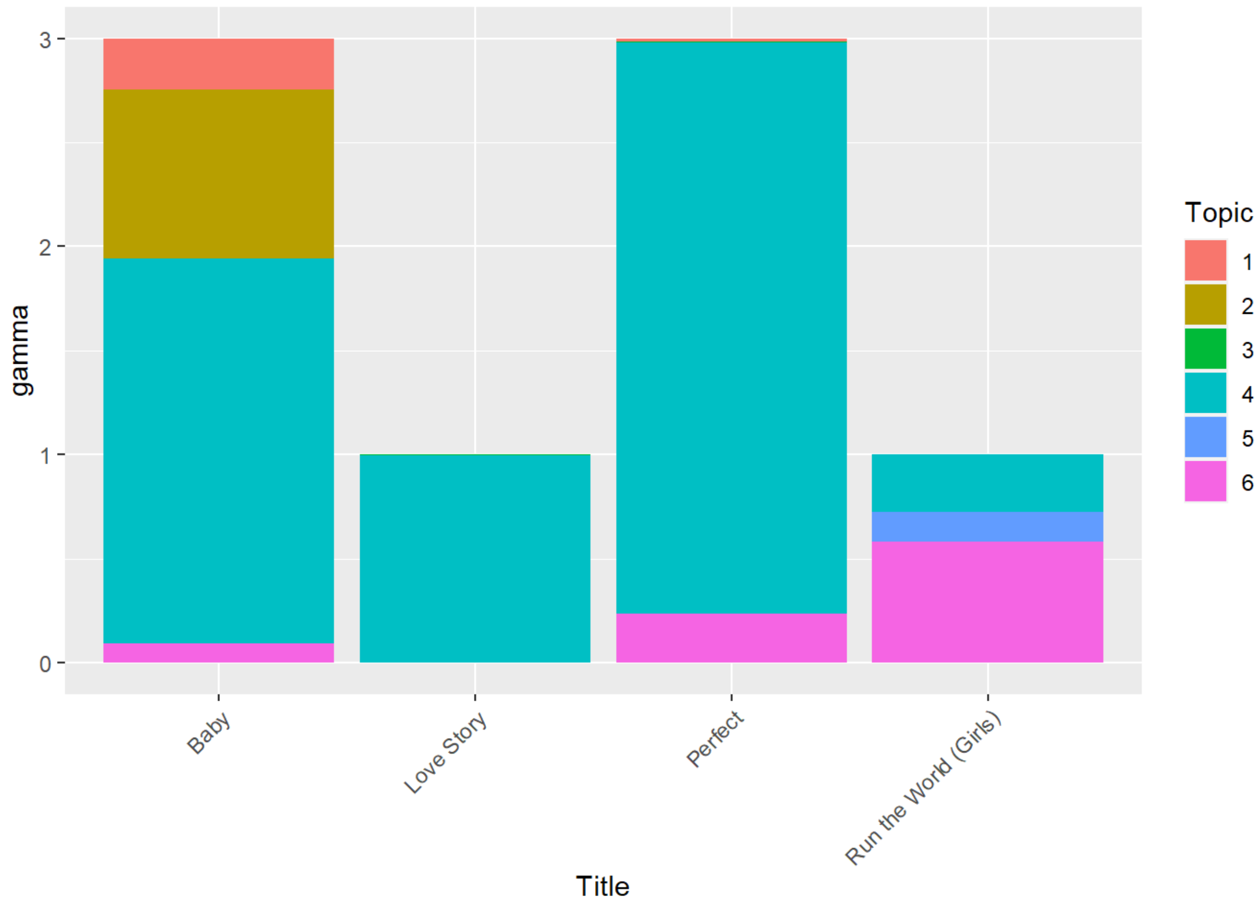
```
song_gamma %>%
  filter(Artist %in% c(
    "Beyoncé", "BTS (방탄소년단)", "Coldplay", "Ed Sheeran", "Justin Bieber",
    "Lady Gaga", "Selena Gomez", "Taylor Swift",  "Drake", "Eminem", "Dua Lipa",
    "Nicki Minaj",  "Rihanna")
) %>%
  ggplot() +
  geom_col(aes(x = Artist,
               y = gamma,
```

```
              fill = factor(topic))) +
  scale_fill_discrete(name = "Topic") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Examining the gamma distribution for specific songs to understand how these themes manifest in individual compositions.

```
song_gamma2 <- song_gamma %>%
  filter(Title %in% c("Perfect", "Love Story" , "Baby" , "Run the World (Girls)")
) %>%
  ggplot() +
  geom_col(aes(x = Title,
              y = gamma,
              fill = factor(topic))) +
  scale_fill_discrete(name = "Topic") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

song_gamma2
```



We’ve concluded our data mining analysis. Now, let’s leverage these insights to provide clear answers to our research questions.

Answering the Research Questions

1. Are the majority of songs in the dataset positive or negative?

The majority of songs in the dataset showcase a nuanced blend of positive and negative sentiments, with a slightly higher prevalence of negativity. Around 2,700 songs convey negative sentiments, while approximately 2,300 songs express positive sentiments.

2. Is there a distinct group of artists who predominantly sing positive or negative songs?

Upon examining individual artist sentiment scores, notable trends emerged. Justin Bieber and Lady Gaga emerged as artists primarily associated with positive sentiments, boasting the highest positive sentiment scores. In contrast, Eminem and Nicki Minaj exhibited a propensity for songs with negative sentiments. Eminem, in particular, stood out with a substantial negative sentiment score, surpassing -30,000, indicating a consistent inclination towards more negative expressions in his lyrics.

3. How has the sentiment of music evolved over time?

In the exploration of music sentiment from 2000 to 2022, a consistent trend emerged, showcasing either a prevalence of negative songs or an almost equal number of positive and negative songs in most years. Exceptions to this trend were noted in 2011, 2017, and 2020, where a notable increase in positive songs occurred.

Interestingly, 2020 marked the year with the highest number of songs released in the dataset. However, following this peak, a sharp decline in the number of songs released was observed. This decline could be attributed to various factors, such as the impact of the COVID-19 pandemic, disrupting the music industry and artists' ability to create new songs, or limitations in available data for that time period.

4. What are the prevalent themes covered in the songs, and how do artists differ from each other in terms of lyrical content?

In addressing the question on prevalent themes and artist-specific differences, the technique of "topic modeling" was employed, yielding six topics for analysis. Three distinctive topics, referred to as "Raw" (Topic 2 and 6), "Korean" (topic 5) and "Love and Emotion" (Topic 4), encapsulate prevalent themes across various artists.

Artist-Specific Gamma Distribution:

- BTS's lyrics predominantly align with Topic 5, which we named as "Korean".
- Eminem showcases a higher proportion of Topic 2 , characterized as "Raw"
- Drake showcases a higher proportion of Topic 6, again characterized as "Raw"
- Taylor Swift and Coldplay exhibit a notable presence in Topic 4, labeled as "Love and Emotion."
- Topics 1 and 3 do not appear to be prevalent among any specific artists.

Analysis of Random Famous Songs:

- "Baby" by Justin Bieber: This song encompasses a combination of Topics 1, 2, 4, and 6, with a predominant emphasis on Topic 4. The themes associated with Topic 4, which appear to revolve around concepts of love, dominate the song.
- "Love Story" by Taylor Swift: This song is entirely associated with Topic 4, indicating a strong thematic focus on love.
- "Perfect" by Ed Sheeran: This song is a blend of Topics 4 and 6, with a major emphasis on Topic 4. The predominant themes revolve around the concept of love.
- "Run the World (Girls)" by Beyonce: This song is a mix of Topics 4, 5, and 6, with a major emphasis on Topic 6. The themes associated with Topic 6 seem to be prominent, suggesting a focus on assertiveness or empowerment.

Conclusion

So, after looking at lots of songs and artists, we found out that each artist has their own way of singing and expressing feelings. We also checked how music changes over time and what kind of things artists like to sing about.

The basic findings tell us what words artists use a lot, how often they put out albums or songs, and other cool stuff. While these insights form a strong foundation, envisioning this analysis with a touch of context could enhance its depth and resonance.

This analytical journey opens the door to a myriad of practical applications, making it a powerful tool for various stakeholders in the music industry:

1. **Audience Understanding:** By comprehending the predominant sentiments and themes associated with specific artists, this data can serve as a key resource for studying the preferences and emotional resonances of diverse audience segments. It enables a deeper understanding of the connections audiences forge with artists who align with their emotional inclinations.

2. **Artist Comparison and Collaboration:** The analysis provides a quantitative basis for comparing artists based on the nature of their lyrics. Artists who share similar themes and sentiments can find common ground for collaboration, creating synergies that transcend individual styles.

3. **Strategic Artist Placement:** The study of sentiment patterns and prevalent themes across different regions allows for strategic artist placement. Organizers can analyze the emotional inclination of a locality's crowd to bring in artists whose music aligns with the sentiments of the audience, ultimately enhancing ticket sales and audience satisfaction.

4. **Guidance for Emerging Artists:** New artists can benefit from this analysis by gaining insights into the types of songs that resonate with audiences. It serves as a guide for emerging talents, helping them navigate the vast landscape of musical expression and make informed decisions about their artistic direction.

5. **Economic Considerations:** For regions with budget constraints, understanding sentiment distributions enables organizers to make informed decisions about artist selection. For instance, if a renowned artist like Eminem is financially prohibitive for a smaller city, organizers can identify alternative artists with similar sentiments to provide a comparable musical experience within budget constraints.

In a nutshell, this data isn't just for music nerds; it's for everyone in the music game. Whether you're a pro in the industry, planning a concert, or a newbie trying to find your sound, this analysis guides the way. It's not just about music; it's about creating a connection between artists and their fans that hits all the right notes.