```
title: "Spotify 2024 Data Analysis"
author: "Kristian Gambuzza"
date: "2024-12-08"
output:

  pdf_document: default
  html_document: default
  word_document: default
---
```

Introduction

Music is an immersive and exciting cultivation of free-flowing feelings and ideas that can captivate any listener. Listeners often describe music as entertainment, while others can describe it as therapy and something they could hold onto as they go through their daily lives. As we unpackage all aspects of music, the music industry, in particular, has revolutionized how society can listen to music. In the past, people used CDs, vinyls, record players, etc., to consume music. In today's modern age, technology has developed, and the industry is constantly amplifying the standards, such as platforms(Spotify, Apple Music, SoundCloud, etc) where listeners can find a library of music that has stretched over decades. Specifically, Spotify is an immersive streaming platform where music of all genres can be found and manipulated to a user's liking. Daniel Ek and Martin Lorenzon founded Spotify in Sweden. It grew and expanded a few years later and gained significant popularity in 2009-2014. Notably, in 2009, "Spotify secured major licensing deals with record labels like Universal Music Group, Sony BMG, EMI Music, and Warner Music Group. This allowed the platform to expand its music library significantly. Spotify also launched its mobile app for iOS and Android, allowing users to stream music on the go." Its most analytical and data-driven years came in 2016 and 2017. Specifically, in 2016, "Spotify reached a milestone of 100 million active users and introduced Spotify Wrapped, a personalized year-end summary of users' listening habits that became an annual favorite among users. Further down the line, in 2017, "Spotify acquired several tech companies, including Soundwave and Sonic, to enhance its music discovery algorithms. The company also introduced Spotify for Artists, providing musicians analytics on their streams and listeners. Fast forward to 2024, as AI has started to develop and take "control" of our society and the world, "Spotify reached 500 million active users, including 210 million premium subscribers. The company continued to invest in AI-driven music recommendations and personalized playlists, solidifying its position as a leader in music discovery. Spotify also introduced an AI-powered DJ feature, using generative AI to deliver a more personalized listening experience." It continues to be an all-time leading platform for music, with millions of users consuming it daily. As any platform gets explored, more characteristics are revealed, and the analysis being conducted will further explore all Spotify has to offer using a specific data set. This data set entails Spotify's most streamed songs in 2024 to grasp what mainly drives their innovation and vision. Furthermore, analyzing trends and patterns will help us better understand the music landscape.

Objective of This Analysis

In this report, we analyze Spotify's most streamed songs in 2024 to understand trends, patterns, and innovations driving their success.

Research Questions

1. **Impact of Explicit Content** How does the inclusion of explicit content in Spotify tracks impact their total Spotify streams and playlist reach, and are there significant differences in this relationship across different demographic markets (e.g., U.S., Europe, Latin America)?

2. **Timing and Popularity** What is the relationship between the release date of songs and their total streaming metrics across platforms, and does the timing of release (e.g., season or month) impact their popularity in specific demographics?

3. **Artist Success and Playlists** What is the relationship between an artist's overall streaming success and representation in playlists, considering the effect of explicit content and release date on song popularity?"

4. **Solo Artists vs. Collaborations** Is there a significant difference in streaming numbers between solo artists and collaborations?

Data and Methods

Dataset Overview

The dataset contains Spotify's most streamed songs in 2024, including metrics like total streams, explicit content, release dates, and playlist representation.

Results and Analysis:

1. **Impact of Explicit Content** How does the inclusion of explicit content in Spotify tracks impact their total Spotify streams and playlist reach, and are there significant differences in this relationship across different demographic markets (e.g., U.S., Europe, Latin America)?

Streaming Trends by Explicit Content:

- The output of this code analyzes the impact of explicit content on Spotify streams and playlist reach while exploring potential differences across demographic markets. It begins by loading and cleaning the dataset, ensuring numerical columns for streams and playlist reach are formatted for analysis and categorizing tracks as explicit or non-explicit. A simulated "Region" column assigns each track to a demographic market (e.g., North America, Europe) to enable regional comparisons. The dataset is then grouped by explicitness, calculating average streams, playlist reach, and track counts for each category. These summarized statistics

highlight differences in performance between explicit and non-explicit tracks, providing a foundation for further analysis, such as testing statistical significance or visualizing trends across regions.

```{r summarise data, warning=FALSE, message=FALSE,echo=FALSE}
# Load necessary libraries
library(dplyr)
library(ggplot2)

# Step 1: Load and clean data
# Load the Spotify dataset
spotify_data <- read.csv(
  "C:/Users/krist/OneDrive/Most Streamed Spotify Songs 2024.csv",
  encoding = "latin1"
)

# Clean and convert numerical columns
spotify_data <- spotify_data %>%
  mutate(
    Spotify.Streams = as.numeric(gsub(",", "", Spotify.Streams)),
    Spotify.Playlist.Reach = as.numeric(gsub(",", "", Spotify.Playlist.Reach))
  )

# Convert Explicit.Track to a factor with meaningful labels
spotify_data$Explicit.Track <- factor(
  spotify_data$Explicit.Track,
  levels = c(0, 1),
  labels = c("Non-Explicit", "Explicit")
)

# Step 2: Add a simulated "Region" column for analysis
set.seed(123)  # For reproducibility
regions <- c("North America", "Europe", "Latin America", "Asia", "Oceania")
spotify_data$Region <- sample(regions, nrow(spotify_data), replace = TRUE)

# Step 3: Summarize data
# Calculate summary statistics grouped by Explicit.Track
summary_stats <- spotify_data %>%
  group_by(Explicit.Track) %>%
  summarise(
    Avg_Streams = mean(Spotify.Streams, na.rm = TRUE),
    Avg_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm = TRUE),
    Count = n()
  )
```

```
# Print summary statistics
print(summary_stats)
```

Findings:

- Non-explicit tracks slightly outperform explicit tracks in average streams. This could possibly mean that there is broader appeal or preference for non-explicit tracks.

- Explicit tracks have a slightly higher playlist reach. This may suggest they are featured in playlists with larger or more engaged audiences, despite there being less songs.

- The higher count of non-explicit tracks highlights a trend in the music industry favoring the creation and promotion of non-explicit songs.

Seasonal Impact on Song Popularity:

- The following two demographics will demonstrate explicit content on Spotify, This code generates two boxplots to analyze the impact of explicit content on Spotify streaming metrics and playlist reach. The first plot compares the distribution of Spotify Streams for tracks categorized as "Explicit" or "Non-Explicit," using a boxplot to highlight medians, quartiles, and outliers. This visualization helps assess whether explicit tracks consistently receive more or fewer streams than non-explicit tracks. The second plot examines Playlist Reach, comparing how widely explicit and non-explicit tracks are included in playlists.

```{r explicit-content-analysis, warning=FALSE, message=FALSE,echo=FALSE}

# Enhanced Plot: Spotify Streams by Explicit Content
ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Streams)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red", outlier.shape = 16) +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3, color = "darkred", stroke = 1.5)
+
  labs(
    title = "Spotify Streams by Explicit Content",
    subtitle = "Boxplot with Mean Indicators",
    x = "Explicit Track",
    y = "Spotify Streams (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
```

```
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )

# Enhanced Plot: Playlist Reach by Explicit Content
ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Playlist.Reach)) +
  geom_boxplot(fill = "lightgreen", color = "darkgreen", outlier.color = "orange", outlier.shape =
16) +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3, color = "blue", stroke = 1.5) +
  labs(
    title = "Playlist Reach by Explicit Content",
    subtitle = "Boxplot with Mean Indicators",
    x = "Explicit Track",
    y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )
```

Findings:

Streams:

- Explicit tracks exhibit more variability in streams, suggesting appeal across diverse audiences.

- Non-explicit tracks have higher median playlist reach, aligning with broader audience acceptance.

- Both have similar medians indicating that both can achieve dominant success.

Reach:

- Explicit tracks have a slightly higher average playlist reach, suggesting they may be featured more often in high-reach playlists, despite their smaller track count.

- Both contain similar medians, but the mean is slightly higher for explicit songs than non meaning that there could be more explicit playlist exposure.

- Both can grant massive success in playlist exposure due to their similar outliers.

Revised BoxPlots Without Outliers

```{r outliers, warning=FALSE, message=FALSE,echo=FALSE}
# Boxplot: Spotify Streams by Explicit Content (No Outliers)
ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Streams)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.shape = NA) +
  labs(
    title = "Spotify Streams by Explicit Content (No Outliers)",
    subtitle = "Comparison with Outliers Removed",
    x = "Explicit Track",
    y = "Spotify Streams (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )

# Boxplot: Playlist Reach by Explicit Content (No Outliers)
ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Playlist.Reach)) +
  geom_boxplot(fill = "lightgreen", color = "darkgreen", outlier.shape = NA) +
  labs(
    title = "Playlist Reach by Explicit Content (No Outliers)",
    subtitle = "Comparison with Outliers Removed",
    x = "Explicit Track",
    y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )

```

1: Explicit Content Impact on Streaming:

Explicit tracks may have significantly higher streams if their average is consistently more outstanding in summary statistics and boxplot visualizations. If the p-value from the t-test is \< 0.05, the difference in streams between explicit and non-explicit tracks is statistically significant.

Non-explicit tracks outperform explicit ones in regions or genres with family-oriented preferences or stricter content regulations.

2: Explicit Content Impact on Playlist Reach:

Higher reach: This suggests that explicit tracks are widely included in popular playlists, catering to younger or genre-specific audiences (e.g., rap, hip-hop). Lower reach: Indicates playlists might favor non-explicit tracks for broader, general audience appeal. A significant difference in playlist reach (p-value \< 0.05) indicates explicit content influences playlist inclusion.

3: Removing Outliers

We can examine that if we were to remove outliers from our boxplots may simplify the plots but also risks ignoring critical aspects of the data. The presence of many outliers implies that the data has a skewed or heavy-tailed distribution, typical in streaming datasets where a few tracks dominate while most have much lower values.

T-Test on Reach:

- I will be conducting a t-test that will determine the difference between playlist reach and streams by region. First, an independent t-test evaluates whether the playlist reach differs significantly between explicit and non-explicit tracks. It does so by comparing the means, then test determines if explicitness has a ultimate effect on playlist inclusion, with the p-value indicating the strength of the result. One could gather from this output that: Streams Test: If p-value \< 0.05, there's a significant difference between explicit and non-explicit tracks for streams. Example: Explicit tracks have a mean of 5 million streams more than non-explicit tracks. Reach Test: If p-value \< 0.05, playlist reach varies significantly by explicit content. Example: Explicit tracks reach 10% fewer playlists on average compared to non-explicit tracks.

```{r t-test, warning=FALSE, message=FALSE,echo=FALSE}

reach_test <- t.test(Spotify.Playlist.Reach ~ Explicit.Track, data = spotify_data, na.action = na.omit)
print(reach_test)

```

- It was found that a p-value (\< 0.05) indicates that explicitness impacts playlist reach, with non-explicit tracks often being included in all Spotify generated playlists.

Region Analysis: The next step was to provide a analysis based upon each region and from the output we can determine that:

- North America and Europe: Are likely to favor explicit tracks due to cultural openness and the popularity of explicit-heavy genres.

- Latin America and Asia: May lean toward non-explicit tracks due to cultural norms or stricter censorship practices.

- Oceania: Most likely shares North American and Europe views upon music.

```{r region, warning=FALSE, message=FALSE,echo=FALSE}
# Step 6: Demographic Analysis
# Perform demographic analysis with enhanced statistics
demographic_analysis <- spotify_data %>%
  group_by(Region, Explicit.Track) %>%
  summarise(
    Avg_Streams = mean(Spotify.Streams, na.rm = TRUE),
    Median_Streams = median(Spotify.Streams, na.rm = TRUE),
    SD_Streams = sd(Spotify.Streams, na.rm = TRUE),
    Avg_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm = TRUE),
    Median_Playlist_Reach = median(Spotify.Playlist.Reach, na.rm = TRUE),
    SD_Playlist_Reach = sd(Spotify.Playlist.Reach, na.rm = TRUE),
    Count = n(),
    .groups = "drop" # Prevent grouped data frame issues
  ) %>%
  arrange(Region, Explicit.Track) # Sort by Region and Explicitness

# Print demographic analysis
print(demographic_analysis)


library(knitr) # For formatted output
kable(demographic_analysis, digits = 2, caption = "Demographic Analysis of Spotify Data")

```

The following demographics will help one see the key differences:

```{r region-demo, warning=FALSE, message=FALSE,echo=FALSE}
# Visualization by Region
library(ggplot2)
library(scales)

# Plot 1: Average Streams by Region and Explicit Content
ggplot(demographic_analysis, aes(x = Region, y = Avg_Streams, fill = Explicit.Track)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), color = "black") +
```

```
  geom_text(aes(label = round(Avg_Streams, 2)),
            position = position_dodge(width = 0.8),
            vjust = -0.5,
            size = 3) +
  labs(
    title = "Average Streams by Region and Explicit Content",
    x = "Region",
    y = "Average Streams"
  ) +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title = element_text(face = "bold"),
    legend.title = element_blank()
  )

# Plot 2: Playlist Reach by Region and Explicit Content
ggplot(demographic_analysis, aes(x = Region, y = Avg_Playlist_Reach, fill = Explicit.Track)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), color = "black") +
  geom_text(aes(label = round(Avg_Playlist_Reach, 2)),
            position = position_dodge(width = 0.8),
            vjust = -0.5,
            size = 3) +
  labs(
    title = "Playlist Reach by Region and Explicit Content",
    x = "Region",
    y = "Playlist Reach"
  ) +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title = element_text(face = "bold"),
    legend.title = element_blank()
  )

```

The two graphs provided entail insightful information such as:

- 1.Explicit tracks likely dominate genres like rap, hip-hop, and pop, appealing to younger audiences.

- 2.Non-explicit tracks perform better in universally appealing or culturally sensitive genres like classical, instrumental, or acoustic.

- 1.Regions with higher streams for explicit tracks may prioritize youthful, liberal audiences (e.g., North America, Oceania). 2.Non-explicit content likely aligns better with conservative or traditional audience demographics (e.g., Asia, parts of Latin America, Europe).

- Ultimately, these insights reveal potential regional preferences and cultural influences, helping determine whether or not explicit content significantly impacts Spotify performance metrics across demographic markets.

Key Takeaways:

- Songs with explicit content show varying trends across demographic regions.
- Timing plays a crucial role in song popularity, with seasonal peaks in certain months.
- Collaborations tend to attract more streams and broader playlist inclusion.

Recommendations for Spotify

1. Enhance personalized recommendations based on seasonal trends.
2. Optimize playlists to feature successful collaborations prominently.
3. Consider explicit content preferences in specific demographic targeting.

---

2. **Timing and Popularity** What is the relationship between the release date of songs and their total streaming metrics across platforms, and does the timing of release (e.g., season or month) impact their popularity in specific demographics?

Seasonal and Monthly Trends:

- The output of this code analyzes seasonal and monthly trends in streaming metrics across Spotify, YouTube, and TikTok, focusing on the following:

- Seasonal trends in Spotify streams, YouTube views, and TikTok views.

- Monthly patterns to asses user feedback and distinguish what they enjoy or do not enjoy.

- It also groups the data by season and calculates the average values for streams and views across platforms, producing a summary table of seasonal trends, which is printed for analysis.

```{r season, warning=FALSE, message=FALSE,echo=FALSE}
if (!require(lubridate)) install.packages("lubridate")
```

```
# Load the library
library(lubridate)

# Load and clean data
spotify_data <- read.csv("C:/Users/krist/OneDrive/Most Streamed Spotify Songs 2024.csv",
encoding = "latin1")
spotify_data <- spotify_data %>%
  mutate(Release.Date = as.Date(Release.Date, format = "%m/%d/%Y"),
         Month = month(Release.Date, label = TRUE),
         Season = case_when(
           Month %in% c("Dec", "Jan", "Feb") ~ "Winter",
           Month %in% c("Mar", "Apr", "May") ~ "Spring",
           Month %in% c("Jun", "Jul", "Aug") ~ "Summer",
           Month %in% c("Sep", "Oct", "Nov") ~ "Fall"
         ))

# Convert numeric columns
spotify_data <- spotify_data %>%
  mutate(across(c(Spotify.Streams, YouTube.Views, TikTok.Views),
                ~ as.numeric(gsub(",", "", .))))

# Summarize by season
season_summary <- spotify_data %>%
  group_by(Season) %>%
  summarize(
    Avg_Spotify_Streams = mean(Spotify.Streams, na.rm = TRUE),
    Avg_YouTube_Views = mean(YouTube.Views, na.rm = TRUE),
    Avg_TikTok_Views = mean(TikTok.Views, na.rm = TRUE)
  )
print(season_summary)
```

Results and Analysis:

1. Seasonal Trends in Streaming Metrics:

- Summer was the highest season containing the largest amount of Spotify streams and TikTok views implying that users usually consumed the most during a period of vacationing and warm weather.
- In the Winter season, YouTube views were the largest most likely due to holiday-related content being published.
- Within the Fall and Spring seasons, it can be examined that there was a moderate/consistent rate of listening behaviors.

2. Monthly Trends in Streaming Metrics:

- The output of this code will analyze monthly listening behaviors across all given platforms.

- The code groups the spotify_data dataset by month. It calculates the average values for Spotify.Streams, YouTube.Views, and TikTok.Views for each month, excluding any missing values, and creates a summary table of these monthly averages, which is then printed.

```{r season-monthly, warning=FALSE, message=FALSE,echo=FALSE}
# Check and load lubridate package
if (!requireNamespace("lubridate", quietly = TRUE)) {
  install.packages("lubridate")
}
library(lubridate)
library(dplyr)

# Ensure the dataset has a Date column; dynamically create a Month column
if (!"Month" %in% colnames(spotify_data) & "Date" %in% colnames(spotify_data)) {
  spotify_data <- spotify_data %>%
    mutate(Month = month(ymd(Date), label = TRUE, abbr = TRUE)) # Extract month names
}

# Summarize data by month
month_summary <- spotify_data %>%
  group_by(Month) %>%
  summarize(
    Avg_Spotify_Streams = mean(Spotify.Streams, na.rm = TRUE),
    Median_Spotify_Streams = median(Spotify.Streams, na.rm = TRUE),
    SD_Spotify_Streams = sd(Spotify.Streams, na.rm = TRUE),
    Avg_YouTube_Views = mean(YouTube.Views, na.rm = TRUE),
    Median_YouTube_Views = median(YouTube.Views, na.rm = TRUE),
    SD_YouTube_Views = sd(YouTube.Views, na.rm = TRUE),
    Avg_TikTok_Views = mean(TikTok.Views, na.rm = TRUE),
    Median_TikTok_Views = median(TikTok.Views, na.rm = TRUE),
    SD_TikTok_Views = sd(TikTok.Views, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(match(Month, month.abb)) # Ensure months are in calendar order

# Print the summary
print(month_summary)

library(knitr)
kable(month_summary, digits = 2, caption = "Monthly Summary of Streams and Views")
```

```

### Key Findings:

- July and August: Highest Spotify and TikTok metrics, most likely due to periods of vacation time and a break of school time.
- December: High amounts of YouTube views, due to holiday music and related content.
- March and October: Transitional months showing steady engagement across all platforms.
- High standard deviation months (e.g., July or December) often reflect the presence of outliers or significant variability in track performance.

### Demographics:

### 1. Average Spotify Streams by Season:

```{r season-demo, warning=FALSE, message=FALSE,echo=FALSE}
# Seasonal bar chart

library(ggplot2)
library(scales)

# Highlight the season with the highest average streams
max_season <- season_summary %>%
  filter(Avg_Spotify_Streams == max(Avg_Spotify_Streams)) %>%
  pull(Season)

ggplot(season_summary, aes(x = Season, y = Avg_Spotify_Streams, fill = Season)) +
  geom_bar(stat = "identity", color = "black", width = 0.7, alpha = 0.9) +
  geom_text(aes(label = scales::comma(round(Avg_Spotify_Streams, 0))),
            vjust = -0.5, size = 4, fontface = "bold", color = "black") +
  scale_fill_manual(values = c(
    "Winter" = ifelse("Winter" %in% max_season, "dodgerblue4", "steelblue"),
    "Spring" = ifelse("Spring" %in% max_season, "darkgreen", "lightgreen"),
    "Summer" = ifelse("Summer" %in% max_season, "goldenrod3", "gold"),
    "Fall" = ifelse("Fall" %in% max_season, "darkorange3", "orange")
  )) +
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.1))) +
  labs(
    title = "Average Spotify Streams by Season",
    subtitle = paste("Season with the highest streams:", max_season),
    x = "Season",
    y = "Average Spotify Streams"
  ) +
```
```

```
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 12, face = "italic", color = "gray40"),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12),
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    panel.grid.minor.y = element_blank()
  )

```

2. Streaming Metrics by Month:

```{r season-demos, warning=FALSE, message=FALSE,echo=FALSE}
library(ggplot2)
library(scales)
ggplot(month_summary, aes(x = Month)) +
  # Spotify Streams
  geom_line(
    aes(y = Avg_Spotify_Streams, color = "Spotify Streams", group = 1),
    size = 1.2,
    position = position_dodge(width = 0.3)
  ) +
  geom_point(
    aes(y = Avg_Spotify_Streams, color = "Spotify Streams"),
    size = 3,
    position = position_dodge(width = 0.3)
  ) +
  geom_text(
    aes(y = Avg_Spotify_Streams, label = scales::comma(round(Avg_Spotify_Streams, 0)), color
= "Spotify Streams"),
    vjust = -1, size = 3.5, show.legend = FALSE,
    position = position_dodge(width = 0.3)
  ) +
  # YouTube Views
  geom_line(
    aes(y = Avg_YouTube_Views, color = "YouTube Views", group = 1),
    size = 1.2, linetype = "dashed",
    position = position_dodge(width = 0.3)
  ) +
  geom_point(
    aes(y = Avg_YouTube_Views, color = "YouTube Views"),
```

```
    size = 3,
    position = position_dodge(width = 0.3)
  ) +
  geom_text(
    aes(y = Avg_YouTube_Views, label = scales::comma(round(Avg_YouTube_Views, 0)), color =
"YouTube Views"),
    vjust = -1, size = 3.5, show.legend = FALSE,
    position = position_dodge(width = 0.3)
  ) +
  # TikTok Views
  geom_line(
    aes(y = Avg_TikTok_Views, color = "TikTok Views", group = 1),
    size = 1.2, linetype = "dotted",
    position = position_dodge(width = 0.3)
  ) +
  geom_point(
    aes(y = Avg_TikTok_Views, color = "TikTok Views"),
    size = 3,
    position = position_dodge(width = 0.3)
  ) +
  geom_text(
    aes(y = Avg_TikTok_Views, label = scales::comma(round(Avg_TikTok_Views, 0)), color =
"TikTok Views"),
    vjust = -1, size = 3.5, show.legend = FALSE,
    position = position_dodge(width = 0.3)
  ) +
  # Titles and Labels
  labs(
    title = "Average Streaming Metrics by Month",
    subtitle = "Trends across Spotify, YouTube, and TikTok",
    x = "Month",
    y = "Average Streams",
    color = "Platform"
  ) +
  # Custom Colors
  scale_color_manual(values = c(
    "Spotify Streams" = "blue",
    "YouTube Views" = "red",
    "TikTok Views" = "green"
  )) +
  # Themes
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
```

```
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic"),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12),
    legend.title = element_text(size = 12, face = "bold"),
    legend.text = element_text(size = 11),
    legend.position = "bottom"
  ) +
  # Adjust Y-axis for better readability
  scale_y_continuous(labels = scales::comma)


```

Key Takeaways:

- Winter is the dominant season for streaming, particularly on Spotify and TikTok.
- Releasing content on TikTok during its peak months (March-June) could maximize reach.
- For Spotify and YouTube, late Q4 (November-December) is the best time to release content.

Recommendations for Spotify:

- Seasonal Releases: Focus on releasing upbeat tracks during the summer for Spotify and TikTok, and holiday content in December for YouTube.
- Cross-Platform Campaigns: Coordinate campaigns to align seasonal preferences across platforms.
- Monthly Trends Analysis: Monitor monthly metrics for constant improvement and updated user behavior analysis

---

3. **Artist Success and Playlists** What is the relationship between an artist's overall streaming success and representation in playlists, considering the effect of explicit content and release date on song popularity?"

Correlation between Playlist Reach and Streams:

- The output of this code analyzes any type of correlation between a playlist and amount of streams by performing a cor.test.

```{r cor.test, warning=FALSE, message=FALSE,echo=FALSE}
spotify_data <- read.csv("C:/Users/krist/OneDrive/Most Streamed Spotify Songs 2024.csv",stringsAsFactors = FALSE, encoding = "latin1")
colnames(spotify_data)
```

```
# Data cleaning
spotify_data <- dplyr::mutate(spotify_data,
                  Spotify.Streams = as.numeric(gsub(",", "", Spotify.Streams)),
                  Spotify.Playlist.Count = as.numeric(gsub(",", "", Spotify.Playlist.Count)),
                  Spotify.Playlist.Reach = as.numeric(gsub(",", "", Spotify.Playlist.Reach)),
                  Pandora.Streams = as.numeric(gsub(",", "", Pandora.Streams)),
                  Release.Date = as.Date(`Release.Date`, format = "%m/%d/%Y"))



data_subset <- spotify_data %>%
  select(Artist, Spotify.Streams, Spotify.Playlist.Count,
         Spotify.Playlist.Reach, Explicit.Track, Release.Date)

# Analysis 1: Correlation Between Playlist Reach and Streams
# Analysis 1: Correlation Between Playlist Reach and Streams
if (nrow(data_subset) > 1) { # Ensure there is sufficient data for correlation
  correlation_result <- cor.test(
    data_subset$Spotify.Playlist.Reach,
    data_subset$Spotify.Streams,
    use = "complete.obs"
  )

  # Print formatted results
  cat("\nCorrelation Analysis: Playlist Reach vs. Spotify Streams\n")
  cat("-----------------------------------------------------\n")
  cat("Correlation Coefficient (r):", round(correlation_result$estimate, 3), "\n")
  cat("p-value:", format.pval(correlation_result$p.value, digits = 3), "\n")
  cat("Confidence Interval (95%): [",
      round(correlation_result$conf.int[1], 3), ", ",
      round(correlation_result$conf.int[2], 3), "]\n")

  # Interpret the result
  if (correlation_result$p.value < 0.05) {
    cat("\nInterpretation: The correlation is statistically significant.\n")
  } else {
    cat("\nInterpretation: The correlation is not statistically significant.\n")
  }
} else {
  cat("\nError: Insufficient data for correlation analysis.\n")
}

```

Specific Findings:

- Playlists play a significant role in boosting track visibility and streaming success.

- Value: r= 0.59

- The correlation coefficient indicates a strong positive relationship between playlist reach and Spotify streams.

- Range: [0.57, 0.609] The confidence interval is narrow and does not cross zero, meaning the correlation strength is positively strong.

2. Impact of Explicit Content on Streams

- The output of this code will analyze explicit content's effect on streaming efforts and popularity. It will do so by grouping explicitly found content within the dataset and then summarizing that data to find an ultimate mean value.

```{r cor.explicit, warning=FALSE, message=FALSE,echo=FALSE}
explicit_analysis <- data_subset %>%
  group_by(Explicit.Track) %>%
  summarize(Average.Streams = mean(Spotify.Streams, na.rm = TRUE))

print("Impact of Explicit Content on Average Streams:")
print(explicit_analysis)



```

Specific Findings:

- Tracks marked as explicit exhibit higher average streams than non-explicit tracks.
- Explicit content likely resonates more with younger demographics and specific genres like hip-hop and rap.

Demographics:

3. Trends in Spotify Streams Over Time:

- The following demographic will show a visual of Spotify streams over time

```{r cor.demo, warning=FALSE, message=FALSE,echo=FALSE}
```

```
# Visualization of Spotify Streams Over Time
library(ggplot2)
library(scales)
library(dplyr)

# Ensure the Release.Date column is in date format
cleaned_data <- data_subset %>%
  mutate(Release.Date = as.Date(Release.Date)) %>%  # Convert to Date type
  filter(!is.na(Spotify.Streams), !is.na(Release.Date))  # Remove rows with missing values

# Plot trends in Spotify Streams over time
ggplot(cleaned_data, aes(x = Release.Date, y = Spotify.Streams)) +
  geom_point(
    color = "darkorange",
    size = 3,
    alpha = 0.7
  ) +  # Points for individual data
  geom_smooth(
    method = "loess",
    se = TRUE,
    color = "blue",
    linetype = "dashed",
    size = 1
  ) +  # Smooth trend line
  labs(
    title = "Trends in Spotify Streams Over Time",
    subtitle = "Analyzing how release dates influence streaming success",
    x = "Release Date",
    y = "Spotify Streams",
    caption = "Data Source: Spotify"
  ) +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis with commas
  scale_x_date(
    date_labels = "%b %Y",
    date_breaks = "2 months",
    limits = range(cleaned_data$Release.Date, na.rm = TRUE)
  ) +  # Format x-axis for dates
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "gray40"),
    axis.title = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels
    panel.grid.major = element_line(color = "gray85"),
```

```
    panel.grid.minor = element_blank(),  # Simplify minor gridlines
    plot.caption = element_text(size = 10, face = "italic", color = "gray50")
  ) +
  geom_text(
    data = cleaned_data %>%
      filter(Spotify.Streams == max(Spotify.Streams)),  # Highlight top stream
    aes(
      x = Release.Date,
      y = Spotify.Streams,
      label = scales::comma(Spotify.Streams)
    ),
    vjust = -1.5,
    hjust = 0.5,
    size = 3.5,
    color = "red"
  )  # Optional annotation for the highest stream value

```

Findings:

- The trend line reveals a steady increase in streaming metrics over time, possibly reflecting Spotify's growing user base and improved algorithms.
- Recent releases tend to perform better, possibly asserting the importance of marketing and platform analysis

4. Artist-Level Analysis:

- This analysis will be conducted by grouping the artists found within the dataset and then summarizing the core mean value to find top artists based upon streaming metrics.

```{r cor.artist, warning=FALSE, message=FALSE,echo=FALSE}
# Artist-Level Analysis
library(dplyr)
library(knitr)

# Ensure necessary columns exist
if (all(c("Artist", "Spotify.Streams", "Spotify.Playlist.Reach") %in% colnames(data_subset))) {
  # Artist Analysis
  artist_analysis <- data_subset %>%
    group_by(Artist) %>%
    summarize(
      Average.Streams = mean(Spotify.Streams, na.rm = TRUE),
      Total.Streams = sum(Spotify.Streams, na.rm = TRUE),
```

```
    Total.Playlist.Reach = sum(Spotify.Playlist.Reach, na.rm = TRUE),
    Track.Count = n(),
    .groups = "drop"
  ) %>%
  arrange(desc(Average.Streams))  # Sort by Average Streams

 # Display top artists
 cat("\nTop Artists by Average Streams:\n")
 print(kable(head(artist_analysis, 10), digits = 2, caption = "Top 10 Artists by Average Spotify
Streams"))

 # Optional: Display by Total Streams for additional perspective
 artist_analysis_by_total <- artist_analysis %>%
  arrange(desc(Total.Streams))  # Sort by Total Streams

 cat("\nTop Artists by Total Streams:\n")
 print(kable(head(artist_analysis_by_total, 10), digits = 2, caption = "Top 10 Artists by Total
Spotify Streams"))
} else {
 cat("Error: Required columns (Artist, Spotify.Streams, Spotify.Playlist.Reach) are missing in the
dataset.\n")
}

```
```

Specific Findings:

- Top artists have significantly higher average streams and playlist reach, suggesting a core fan-base and playlist placement is relatively higher based upon Spotify's algorithms.

- Playlist reach correlates strongly with artist success, emphasizing the importance of playlist strategy.

- Artists like xSyborg and Vance Joy, with fewer tracks but high average streams, represent niche success and may benefit from targeted campaigns to expand their reach.

- Post Malone and Ed Sheeran achieve high total streams with fewer tracks compared to others in the same category, displaying their consistent streaming power.

Key Takeaways:

- Playlist Reach Correlation: A strong positive correlation suggests playlists are critical to driving streams.
```

- Explicit Content Impact: Explicit tracks outperform non-explicit ones, indicating demographic and genre alignment.

- Trends Over Time: Streaming metrics have steadily increased, emphasizing the importance of consistent release schedules.

- Artist Success: High-performing artists leverage playlist reach effectively, which demonstrates the importance of streaming strategies.

---

4. **Solo Artists vs. Collaborations** Is there a significant difference in streaming numbers between solo artists and collaborations?

1. Streaming Metrics: Solo vs. Collaborations

- The following demonstrates statics based upon songs with or without features/collaborations. It does so by grouping by the collaboration column found within the data set and then summarizing the ultimate mean found between the max and min of Spotify streams.

```{r streaming solo/colab, warning=FALSE, message=FALSE,echo=FALSE}

file_path <- ("C:/Users/krist/Downloads/cleaned_streaming_data.csv")
data <- read.csv(file_path)

# Load necessary libraries
library(dplyr)
library(knitr)

# Display the structure of the dataset
cat("\nInitial Structure of the Dataset:\n")
str(data)

# Convert columns with numeric data stored as strings to numeric
data <- data %>%
  mutate(
    Spotify.Playlist.Reach = as.numeric(gsub(",", "", Spotify.Playlist.Reach)),
    Spotify.Streams = as.numeric(gsub(",", "", Spotify.Streams))
  )

# Display the structure of the cleaned dataset
cat("\nStructure of the Cleaned Dataset:\n")
str(data)
```

```
# Perform analysis grouped by collaboration status
streaming_analysis <- data %>%
  group_by(Collaboration) %>%
  summarise(
    Count = n(),
    Mean_Streams = mean(Spotify.Streams, na.rm = TRUE),
    Median_Streams = median(Spotify.Streams, na.rm = TRUE),
    Std_Dev = sd(Spotify.Streams, na.rm = TRUE),
    Min_Streams = min(Spotify.Streams, na.rm = TRUE),
    Max_Streams = max(Spotify.Streams, na.rm = TRUE),
    .groups = "drop"  # Ensure no unnecessary grouping remains
  )

# Print the summarized analysis in a well-formatted table
cat("\nStreaming Analysis by Collaboration Status:\n")
print(kable(streaming_analysis, digits = 2, caption = "Streaming Analysis by Collaboration
Status"))
```

Key Findings:

- Solo tracks have slightly higher average streams and produce the top-performing track overall (4.28 billion streams).

- Collaborations tend to have a higher baseline performance, with no tracks falling below 278,318 streams.

- The median streams for collaborations and solo tracks are almost identical, meaning that most tracks perform similarly regardless of any collaboration.

2. Playlist Reach: Solo vs. Collaborations:

- The following will summarize playlist reach for solo vs collaborations.

```{r streaming reach, warning=FALSE, message=FALSE,echo=FALSE}

# Load necessary libraries
library(dplyr)
library(knitr)

# Summarize playlist reach for solo vs collaborations
playlist_reach_analysis <- data %>%
  group_by(Collaboration) %>%
```

```
  summarise(
    Count = n(),  # Count of tracks in each category
    Mean_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm = TRUE),
    Median_Playlist_Reach = median(Spotify.Playlist.Reach, na.rm = TRUE),
    Std_Dev_Playlist_Reach = sd(Spotify.Playlist.Reach, na.rm = TRUE),  # Variability
    Min_Playlist_Reach = min(Spotify.Playlist.Reach, na.rm = TRUE),  # Minimum value
    Max_Playlist_Reach = max(Spotify.Playlist.Reach, na.rm = TRUE),  # Maximum value
    .groups = "drop"
  )

# Print the playlist reach analysis in a formatted table
cat("\nPlaylist Reach Analysis by Collaboration Status:\n")
print(kable(playlist_reach_analysis, digits = 2, caption = "Playlist Reach Analysis"))


```

Key Findings:

- Solo tracks generally have a higher mean and median playlist reach than collaborations. This implies that solo efforts secure more playlist exposure.

- Collaborations have a higher minimum playlist reach, indicating that collaborative tracks can benefit from combined marketing efforts

Demographics of Spotify Streams distributed between solo vs. collaboration

```{r demo, warning=FALSE, message=FALSE,echo=FALSE}
# Visualize the distribution of Spotify Streams
library(ggplot2)
library(scales)
ggplot(data, aes(x = Spotify.Streams, fill = Collaboration)) +
  geom_histogram(
    bins = 30,
    position = "identity", # Overlay for better comparison
    alpha = 0.6,
    color = "black"
  ) +
  scale_x_continuous(
    labels = comma,
    breaks = seq(0, max(data$Spotify.Streams, na.rm = TRUE), length.out = 10) # Dynamic axis
  ) +
  scale_fill_brewer(
    palette = "Set2", # Softer palette for better visibility
```

```
  name = "Collaboration Status" # More descriptive legend title
 ) +
 labs(
  title = "Distribution of Spotify Streams",
  subtitle = "Comparison of Solo Artists and Collaborations",
  x = "Spotify Streams (in billions)",
  y = "Frequency"
 ) +
 theme_minimal(base_size = 14) +
 theme(
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5),
  plot.subtitle = element_text(size = 14, hjust = 0.5, color = "gray40"),
  axis.title.x = element_text(size = 14),
  axis.title.y = element_text(size = 14),
  axis.text = element_text(size = 12),
  legend.title = element_text(size = 12),
  legend.text = element_text(size = 12),
  panel.grid.minor = element_blank() # Simplified grid for cleaner visuals
 ) +
 geom_vline(
  aes(xintercept = mean(Spotify.Streams, na.rm = TRUE), color = Collaboration),
  linetype = "dashed",
  size = 0.8,
  show.legend = FALSE
 ) +
 annotate(
  "text",
  x = mean(data$Spotify.Streams, na.rm = TRUE),
  y = 10, # Adjust y position dynamically
  label = "Mean",
  color = "black",
  size = 4,
  hjust = -0.2
 )
```

- Solo tracks are more common in the dataset and have a wider range, including both low-performing tracks and those with extremely high streams

- Collaboration tracks appear to avoid the very lowest streaming counts, likely due to the combined fan bases and marketing efforts.

- The mean appears relatively low compared to the distribution tail, reflecting many tracks fall below the mean due to the existence of outliers.

Comparison of Playlist Reach

```{r demo-comp, warning=FALSE, message=FALSE,echo=FALSE}

library(ggplot2)
library(scales)
ggplot(data, aes(x = Collaboration, y = Spotify.Playlist.Reach, fill = Collaboration)) +
  geom_boxplot(
    alpha = 0.7,
    outlier.color = "red",
    outlier.shape = 16,
    outlier.size = 2,
    width = 0.6
  ) +
  stat_summary(
    fun = mean,
    geom = "point",
    shape = 18,
    size = 3,
    color = "blue",
    aes(group = Collaboration)
  ) +
  scale_y_continuous(
    labels = comma,
    breaks = pretty_breaks(n = 8)
  ) +
  scale_fill_brewer(
    palette = "Set2",
    name = "Collaboration Status"
  ) +
  labs(
    title = "Comparison of Playlist Reach",
    subtitle = "Boxplot Analysis of Solo Artists vs Collaborations",
    x = "Collaboration Status",
    y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "gray40"),
```

```
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    axis.text = element_text(size = 12),
    panel.grid.minor = element_blank(),
    legend.position = "none"
  ) +
  annotate(
    "text",
    x = 1,
    y = max(data$Spotify.Playlist.Reach, na.rm = TRUE),
    label = "Highest Reach",
    color = "red",
    size = 4,
    hjust = -0.1
  )
```

- Solo tracks have a greater reach of achieving playlist reach, as seen by the presence of the outliers and the overall median.

- Collaboration tracks tend to achieve a more consistent playlist reach, more so between a low and high reach.

- The highest-performing solo track significantly skews the potential for playlist reach

Key Takeaways:

- Solo tracks dominate success but come with higher variability.

- Collaborations offer stability and consistent performance, making them an effective strategy for artists seeking steady growth.

Recommendations for Spotify:

- Include more collaborations in playlists to explore growth in a broad range of audiences.\
- Encourage artists to collaborate across genres to tap into diverse audiences.
- Develop the algorithm of collaboration practices and continue to monitor for diverse changes or behaviors.

Conclusion:

- Ultimately, much can be done when analyzing a data set as broad as this one. Each question was answered in separate parts, leading to me drawing conclusions on tests, graphs, etc. I explored dominance between explicit and non-explicit tracks, top artists, streaming dominance between significant platforms such as TikTok and YouTube, preferable tracks by region, exposure of playlist reach between solo and collaboration tracks, and much more!

```
----------------------title: "Spotify 2024 Data Analysis"

author: "Kristian Gambuzza"

date: "2024-12-08"

output:


  pdf_document: default

  html_document: default

  word_document: default

---


## Introduction


Music is an immersive and exciting cultivation of free-flowing
feelings and ideas that can captivate any listener. Listeners often
describe music as entertainment, while others can describe it as
therapy and something they could hold onto as they go through their
daily lives. As we unpackage all aspects of music, the music
industry, in particular, has revolutionized how society can listen to
music. In the past, people used CDs, vinyls, record players, etc., to
consume music. In today's modern age, technology has developed, and
the industry is constantly amplifying the standards, such as
platforms(Spotify, Apple Music, SoundCloud, etc) where listeners can
find a library of music that has stretched over decades.
Specifically, Spotify is an immersive streaming platform where music
of all genres can be found and manipulated to a user’s liking. Daniel
Ek and Martin Lorenzon founded Spotify in Sweden. It grew and
expanded a few years later and gained significant popularity in
2009-2014. Notably, in 2009, “Spotify secured major licensing deals
with record labels like Universal Music Group, Sony BMG, EMI Music,
```

and Warner Music Group. This allowed the platform to expand its music library significantly. Spotify also launched its mobile app for iOS and Android, allowing users to stream music on the go." Its most analytical and data-driven years came in 2016 and 2017. Specifically, in 2016, "Spotify reached a milestone of 100 million active users and introduced Spotify Wrapped, a personalized year-end summary of users' listening habits that became an annual favorite among users. Further down the line, in 2017, "Spotify acquired several tech companies, including Soundwave and Sonic, to enhance its music discovery algorithms. The company also introduced Spotify for Artists, providing musicians analytics on their streams and listeners. Fast forward to 2024, as AI has started to develop and take "control" of our society and the world, "Spotify reached 500 million active users, including 210 million premium subscribers. The company continued to invest in AI-driven music recommendations and personalized playlists, solidifying its position as a leader in music discovery. Spotify also introduced an AI-powered DJ feature, using generative AI to deliver a more personalized listening experience." It continues to be an all-time leading platform for music, with millions of users consuming it daily. As any platform gets explored, more characteristics are revealed, and the analysis being conducted will further explore all Spotify has to offer using a specific data set. This data set entails Spotify's most streamed songs in 2024 to grasp what mainly drives their innovation and vision. Furthermore, analyzing trends and patterns will help us better understand the music landscape.

Objective of This Analysis

In this report, we analyze Spotify's most streamed songs in 2024 to understand trends, patterns, and innovations driving their success.

Research Questions

1. **Impact of Explicit Content** How does the inclusion of explicit content in Spotify tracks impact their total Spotify streams and playlist reach, and are there significant differences in this relationship across different demographic markets (e.g., U.S., Europe, Latin America)?

2. **Timing and Popularity** What is the relationship between the release date of songs and their total streaming metrics across platforms, and does the timing of release (e.g., season or month) impact their popularity in specific demographics?

3. **Artist Success and Playlists** What is the relationship between an artist's overall streaming success and representation in playlists, considering the effect of explicit content and release date on song popularity?"

4. **Solo Artists vs. Collaborations** Is there a significant difference in streaming numbers between solo artists and collaborations?

Data and Methods

Dataset Overview

The dataset contains Spotify's most streamed songs in 2024, including metrics like total streams, explicit content, release dates, and playlist representation.

Results and Analysis:

1. **Impact of Explicit Content** How does the inclusion of explicit content in Spotify tracks impact their total Spotify streams and playlist reach, and are there significant differences in this relationship across different demographic markets (e.g., U.S., Europe, Latin America)?

Streaming Trends by Explicit Content:

- The output of this code analyzes the impact of explicit content on Spotify streams and playlist reach while exploring potential differences across demographic markets. It begins by loading and cleaning the dataset, ensuring numerical columns for streams and playlist reach are formatted for analysis and categorizing tracks as explicit or non-explicit. A simulated "Region" column assigns each track to a demographic market (e.g., North America, Europe) to enable regional comparisons. The dataset is then grouped by explicitness, calculating average streams, playlist reach, and track counts for each category. These summarized statistics highlight differences in performance between explicit and non-explicit tracks, providing a foundation for further analysis, such as testing statistical significance or visualizing trends across regions.

```{r summarise data, warning=FALSE, message=FALSE,echo=FALSE}
# Load necessary libraries
library(dplyr)
library(ggplot2)

# Step 1: Load and clean data
# Load the Spotify dataset
spotify_data <- read.csv(
  "C:/Users/krist/OneDrive/Most Streamed Spotify Songs 2024.csv",
  encoding = "latin1"
)

# Clean and convert numerical columns
spotify_data <- spotify_data %>%
```

```
  mutate(
      Spotify.Streams = as.numeric(gsub(",", "", Spotify.Streams)),
      Spotify.Playlist.Reach = as.numeric(gsub(",", "",
Spotify.Playlist.Reach))
  )

# Convert Explicit.Track to a factor with meaningful labels
spotify_data$Explicit.Track <- factor(
  spotify_data$Explicit.Track,
  levels = c(0, 1),
  labels = c("Non-Explicit", "Explicit")
)

# Step 2: Add a simulated "Region" column for analysis
set.seed(123)  # For reproducibility
regions <- c("North America", "Europe", "Latin America", "Asia",
"Oceania")
spotify_data$Region <- sample(regions, nrow(spotify_data), replace =
TRUE)

# Step 3: Summarize data
# Calculate summary statistics grouped by Explicit.Track
summary_stats <- spotify_data %>%
  group_by(Explicit.Track) %>%
  summarise(
      Avg_Streams = mean(Spotify.Streams, na.rm = TRUE),
```

```
    Avg_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm =
TRUE),

    Count = n()

  )


# Print summary statistics

print(summary_stats)
```

Findings:

- Non-explicit tracks slightly outperform explicit tracks in average streams. This could possibly mean that there is broader appeal or preference for non-explicit tracks.

- Explicit tracks have a slightly higher playlist reach. This may suggest they are featured in playlists with larger or more engaged audiences, despite there being less songs.

- The higher count of non-explicit tracks highlights a trend in the music industry favoring the creation and promotion of non-explicit songs.

Seasonal Impact on Song Popularity:

- The following two demographics will demonstrate explicit content on Spotify, This code generates two boxplots to analyze the impact of explicit content on Spotify streaming metrics and playlist reach. The first plot compares the distribution of Spotify Streams for tracks categorized as "Explicit" or "Non-Explicit," using a boxplot to highlight medians, quartiles, and outliers. This visualization helps assess whether explicit tracks consistently receive more or fewer streams than non-explicit tracks. The second plot examines Playlist Reach, comparing how widely explicit and non-explicit tracks are included in playlists.

```{r explicit-content-analysis, warning=FALSE, message=FALSE,echo=FALSE}

# Enhanced Plot: Spotify Streams by Explicit Content

ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Streams)) +

  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red", outlier.shape = 16) +

  stat_summary(fun = mean, geom = "point", shape = 4, size = 3, color = "darkred", stroke = 1.5) +

  labs(

      title = "Spotify Streams by Explicit Content",

      subtitle = "Boxplot with Mean Indicators",

      x = "Explicit Track",

      y = "Spotify Streams (in millions)"

  ) +

  theme_minimal(base_size = 14) +

  theme(

      plot.title = element_text(face = "bold", size = 16),

      plot.subtitle = element_text(size = 12),
```

```
    axis.title = element_text(face = "bold")
  )

# Enhanced Plot: Playlist Reach by Explicit Content
ggplot(spotify_data, aes(x = Explicit.Track, y =
Spotify.Playlist.Reach)) +
  geom_boxplot(fill = "lightgreen", color = "darkgreen",
outlier.color = "orange", outlier.shape = 16) +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3, color
= "blue", stroke = 1.5) +
  labs(
    title = "Playlist Reach by Explicit Content",
    subtitle = "Boxplot with Mean Indicators",
    x = "Explicit Track",
    y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )
```

Findings:

Streams:

- Explicit tracks exhibit more variability in streams, suggesting appeal across diverse audiences.

- Non-explicit tracks have higher median playlist reach, aligning with broader audience acceptance.

- Both have similar medians indicating that both can achieve dominant success.

Reach:

- Explicit tracks have a slightly higher average playlist reach, suggesting they may be featured more often in high-reach playlists, despite their smaller track count.

- Both contain similar medians, but the mean is slightly higher for explicit songs than non meaning that there could be more explicit playlist exposure.

- Both can grant massive success in playlist exposure due to their similar outliers.

Revised BoxPlots Without Outliers

```{r outliers, warning=FALSE, message=FALSE,echo=FALSE}
```

```
# Boxplot: Spotify Streams by Explicit Content (No Outliers)

ggplot(spotify_data, aes(x = Explicit.Track, y = Spotify.Streams)) +

  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.shape
= NA) +

  labs(

     title = "Spotify Streams by Explicit Content (No Outliers)",

     subtitle = "Comparison with Outliers Removed",

     x = "Explicit Track",

     y = "Spotify Streams (in millions)"

  ) +

  theme_minimal(base_size = 14) +

  theme(

     plot.title = element_text(face = "bold", size = 16),

     plot.subtitle = element_text(size = 12),

     axis.title = element_text(face = "bold")

  )


# Boxplot: Playlist Reach by Explicit Content (No Outliers)

ggplot(spotify_data, aes(x = Explicit.Track, y =
Spotify.Playlist.Reach)) +

  geom_boxplot(fill = "lightgreen", color = "darkgreen",
outlier.shape = NA) +

  labs(

     title = "Playlist Reach by Explicit Content (No Outliers)",

     subtitle = "Comparison with Outliers Removed",

     x = "Explicit Track",
```

```
    y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold")
  )
```

1: Explicit Content Impact on Streaming:

Explicit tracks may have significantly higher streams if their average is consistently more outstanding in summary statistics and boxplot visualizations. If the p-value from the t-test is \< 0.05, the difference in streams between explicit and non-explicit tracks is statistically significant. Non-explicit tracks outperform explicit ones in regions or genres with family-oriented preferences or stricter content regulations.

2: Explicit Content Impact on Playlist Reach:

Higher reach: This suggests that explicit tracks are widely included in popular playlists, catering to younger or genre-specific audiences (e.g., rap, hip-hop). Lower reach: Indicates playlists might favor non-explicit tracks for broader, general audience appeal. A significant difference in playlist reach (p-value \< 0.05) indicates explicit content influences playlist inclusion.

3: Removing Outliers

We can examine that if we were to remove outliers from our boxplots may simplify the plots but also risks ignoring critical aspects of the data. The presence of many outliers implies that the data has a skewed or heavy-tailed distribution, typical in streaming datasets where a few tracks dominate while most have much lower values.

T-Test on Reach:

- I will be conducting a t-test that will determine the difference between playlist reach and streams by region. First, an independent t-test evaluates whether the playlist reach differs significantly between explicit and non-explicit tracks. It does so by comparing the means, then test determines if explicitness has a ultimate effect on playlist inclusion, with the p-value indicating the strength of the result. One could gather from this output that: Streams Test: If p-value \< 0.05, there's a significant difference between explicit and non-explicit tracks for streams. Example: Explicit tracks have a mean of 5 million streams more than non-explicit tracks. Reach Test: If p-value \< 0.05, playlist reach varies significantly by explicit content. Example: Explicit tracks reach 10% fewer playlists on average compared to non-explicit tracks.

```{r t-test, warning=FALSE, message=FALSE,echo=FALSE}

reach_test <- t.test(Spotify.Playlist.Reach ~ Explicit.Track, data = spotify_data, na.action = na.omit)

print(reach_test)

```

- It was found that a p-value (\< 0.05) indicates that explicitness impacts playlist reach, with non-explicit tracks often being included in all Spotify generated playlists.

Region Analysis: The next step was to provide a analysis based upon each region and from the output we can determine that:

- North America and Europe: Are likely to favor explicit tracks due to cultural openness and the popularity of explicit-heavy genres.

- Latin America and Asia: May lean toward non-explicit tracks due to cultural norms or stricter censorship practices.

- Oceania: Most likely shares North American and Europe views upon music.

```{r region, warning=FALSE, message=FALSE,echo=FALSE}
# Step 6: Demographic Analysis

# Perform demographic analysis with enhanced statistics

demographic_analysis <- spotify_data %>%

  group_by(Region, Explicit.Track) %>%

  summarise(

      Avg_Streams = mean(Spotify.Streams, na.rm = TRUE),

      Median_Streams = median(Spotify.Streams, na.rm = TRUE),

      SD_Streams = sd(Spotify.Streams, na.rm = TRUE),

      Avg_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm = TRUE),
```

```
    Median_Playlist_Reach = median(Spotify.Playlist.Reach, na.rm =
TRUE),

    SD_Playlist_Reach = sd(Spotify.Playlist.Reach, na.rm = TRUE),

    Count = n(),

    .groups = "drop" # Prevent grouped data frame issues

  ) %>%

  arrange(Region, Explicit.Track) # Sort by Region and Explicitness


# Print demographic analysis

print(demographic_analysis)



library(knitr) # For formatted output

kable(demographic_analysis, digits = 2, caption = "Demographic
Analysis of Spotify Data")
```

```

### The following demographics will help one see the key differences:

```{r region-demo, warning=FALSE, message=FALSE,echo=FALSE}

# Visualization by Region

library(ggplot2)

library(scales)


# Plot 1: Average Streams by Region and Explicit Content
```
```

```
ggplot(demographic_analysis, aes(x = Region, y = Avg_Streams, fill =
Explicit.Track)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8),
color = "black") +
  geom_text(aes(label = round(Avg_Streams, 2)),
            position = position_dodge(width = 0.8),
            vjust = -0.5,
            size = 3) +
  labs(
     title = "Average Streams by Region and Explicit Content",
     x = "Region",
     y = "Average Streams"
  ) +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(
     axis.text.x = element_text(angle = 45, hjust = 1),
     axis.title = element_text(face = "bold"),
     legend.title = element_blank()
  )

# Plot 2: Playlist Reach by Region and Explicit Content
ggplot(demographic_analysis, aes(x = Region, y = Avg_Playlist_Reach,
fill = Explicit.Track)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8),
color = "black") +
```

```
  geom_text(aes(label = round(Avg_Playlist_Reach, 2)),
            position = position_dodge(width = 0.8),
            vjust = -0.5,
            size = 3) +
  labs(
     title = "Playlist Reach by Region and Explicit Content",
     x = "Region",
     y = "Playlist Reach"
  ) +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(
     axis.text.x = element_text(angle = 45, hjust = 1),
     axis.title = element_text(face = "bold"),
     legend.title = element_blank()
  )
```

The two graphs provided entail insightful information such as:

- 1.Explicit tracks likely dominate genres like rap, hip-hop, and pop, appealing to younger audiences.

- 2.Non-explicit tracks perform better in universally appealing or culturally sensitive genres like classical, instrumental, or acoustic.

- 1.Regions with higher streams for explicit tracks may prioritize youthful, liberal audiences (e.g., North America, Oceania).
2.Non-explicit content likely aligns better with conservative or traditional audience demographics (e.g., Asia, parts of Latin America, Europe).

- Ultimately, these insights reveal potential regional preferences and cultural influences, helping determine whether or not explicit content significantly impacts Spotify performance metrics across demographic markets.

Key Takeaways:

- Songs with explicit content show varying trends across demographic regions.

- Timing plays a crucial role in song popularity, with seasonal peaks in certain months.

- Collaborations tend to attract more streams and broader playlist inclusion.

Recommendations for Spotify

1. Enhance personalized recommendations based on seasonal trends.

2. Optimize playlists to feature successful collaborations prominently.

3. Consider explicit content preferences in specific demographic targeting.

---

2. **Timing and Popularity** What is the relationship between the release date of songs and their total streaming metrics across platforms, and does the timing of release (e.g., season or month) impact their popularity in specific demographics?

Seasonal and Monthly Trends:

- The output of this code analyzes seasonal and monthly trends in streaming metrics across Spotify, YouTube, and TikTok, focusing on the following:

- Seasonal trends in Spotify streams, YouTube views, and TikTok views.

- Monthly patterns to asses user feedback and distinguish what they enjoy or do not enjoy.

- It also groups the data by season and calculates the average values for streams and views across platforms, producing a summary table of seasonal trends, which is printed for analysis.

```{r season, warning=FALSE, message=FALSE,echo=FALSE}

if (!require(lubridate)) install.packages("lubridate")

# Load the library
```

```
library(lubridate)


# Load and clean data

spotify_data <- read.csv("C:/Users/krist/OneDrive/Most Streamed
Spotify Songs 2024.csv", encoding = "latin1")

spotify_data <- spotify_data %>%

  mutate(Release.Date = as.Date(Release.Date, format = "%m/%d/%Y"),

      Month = month(Release.Date, label = TRUE),

      Season = case_when(

           Month %in% c("Dec", "Jan", "Feb") ~ "Winter",

           Month %in% c("Mar", "Apr", "May") ~ "Spring",

           Month %in% c("Jun", "Jul", "Aug") ~ "Summer",

           Month %in% c("Sep", "Oct", "Nov") ~ "Fall"

      ))


# Convert numeric columns

spotify_data <- spotify_data %>%

  mutate(across(c(Spotify.Streams, YouTube.Views, TikTok.Views),

                ~ as.numeric(gsub(",", "", .))))


# Summarize by season

season_summary <- spotify_data %>%

  group_by(Season) %>%

  summarize(

      Avg_Spotify_Streams = mean(Spotify.Streams, na.rm = TRUE),
```

```
    Avg_YouTube_Views = mean(YouTube.Views, na.rm = TRUE),

    Avg_TikTok_Views = mean(TikTok.Views, na.rm = TRUE)

  )

print(season_summary)
```

Results and Analysis:

1. Seasonal Trends in Streaming Metrics:

- Summer was the highest season containing the largest amount of Spotify streams and TikTok views implying that users usually consumed the most during a period of vacationing and warm weather.

- In the Winter season, YouTube views were the largest most likely due to holiday-related content being published.

- Within the Fall and Spring seasons, it can be examined that there was a moderate/consistent rate of listening behaviors.

2. Monthly Trends in Streaming Metrics:

- The output of this code will analyze monthly listening behaviors across all given platforms.

- The code groups the spotify_data dataset by month. It calculates the average values for Spotify.Streams, YouTube.Views, and TikTok.Views for each month, excluding any missing values, and creates a summary table of these monthly averages, which is then printed.

```{r season-monthly, warning=FALSE, message=FALSE,echo=FALSE}

# Check and load lubridate package

if (!requireNamespace("lubridate", quietly = TRUE)) {

  install.packages("lubridate")

}

library(lubridate)

library(dplyr)


# Ensure the dataset has a Date column; dynamically create a Month column

if (!"Month" %in% colnames(spotify_data) & "Date" %in% colnames(spotify_data)) {

  spotify_data <- spotify_data %>%

      mutate(Month = month(ymd(Date), label = TRUE, abbr = TRUE)) # Extract month names

}


# Summarize data by month

month_summary <- spotify_data %>%

  group_by(Month) %>%

  summarize(

      Avg_Spotify_Streams = mean(Spotify.Streams, na.rm = TRUE),

      Median_Spotify_Streams = median(Spotify.Streams, na.rm = TRUE),

      SD_Spotify_Streams = sd(Spotify.Streams, na.rm = TRUE),

      Avg_YouTube_Views = mean(YouTube.Views, na.rm = TRUE),

      Median_YouTube_Views = median(YouTube.Views, na.rm = TRUE),
```

```
      SD_YouTube_Views = sd(YouTube.Views, na.rm = TRUE),

      Avg_TikTok_Views = mean(TikTok.Views, na.rm = TRUE),

      Median_TikTok_Views = median(TikTok.Views, na.rm = TRUE),

      SD_TikTok_Views = sd(TikTok.Views, na.rm = TRUE),

      .groups = "drop"

  ) %>%

  arrange(match(Month, month.abb)) # Ensure months are in calendar
order



# Print the summary

print(month_summary)



library(knitr)

kable(month_summary, digits = 2, caption = "Monthly Summary of
Streams and Views")



```


Key Findings:

- July and August: Highest Spotify and TikTok metrics, most likely due to periods of vacation time and a break of school time.

- December: High amounts of YouTube views, due to holiday music and related content.

- March and October: Transitional months showing steady engagement across all platforms.

- High standard deviation months (e.g., July or December) often reflect the presence of outliers or significant variability in track performance.

Demographics:

1. Average Spotify Streams by Season:

```{r season-demo, warning=FALSE, message=FALSE,echo=FALSE}
# Seasonal bar chart

library(ggplot2)
library(scales)

# Highlight the season with the highest average streams
max_season <- season_summary %>%
  filter(Avg_Spotify_Streams == max(Avg_Spotify_Streams)) %>%
  pull(Season)

ggplot(season_summary, aes(x = Season, y = Avg_Spotify_Streams, fill = Season)) +
  geom_bar(stat = "identity", color = "black", width = 0.7, alpha = 0.9) +
  geom_text(aes(label = scales::comma(round(Avg_Spotify_Streams, 0))),
            vjust = -0.5, size = 4, fontface = "bold", color = "black") +
```

```
  scale_fill_manual(values = c(

      "Winter" = ifelse("Winter" %in% max_season, "dodgerblue4",
"steelblue"),

      "Spring" = ifelse("Spring" %in% max_season, "darkgreen",
"lightgreen"),

      "Summer" = ifelse("Summer" %in% max_season, "goldenrod3",
"gold"),

      "Fall" = ifelse("Fall" %in% max_season, "darkorange3",
"orange")

  )) +

  scale_y_continuous(labels = comma, expand = expansion(mult = c(0,
0.1))) +

  labs(

      title = "Average Spotify Streams by Season",

      subtitle = paste("Season with the highest streams:",
max_season),

      x = "Season",

      y = "Average Spotify Streams"

  ) +

  theme_minimal(base_size = 14) +

  theme(

      plot.title = element_text(hjust = 0.5, size = 18, face =
"bold"),

      plot.subtitle = element_text(hjust = 0.5, size = 12, face =
"italic", color = "gray40"),

      axis.title = element_text(size = 14, face = "bold"),

      axis.text = element_text(size = 12),

      legend.position = "none",
```

```
      panel.grid.major.x = element_blank(),
      panel.grid.minor.y = element_blank()
  )

```

2. Streaming Metrics by Month:

```{r season-demos, warning=FALSE, message=FALSE,echo=FALSE}
library(ggplot2)
library(scales)
ggplot(month_summary, aes(x = Month)) +
  # Spotify Streams
  geom_line(
      aes(y = Avg_Spotify_Streams, color = "Spotify Streams", group =
1),
      size = 1.2,
      position = position_dodge(width = 0.3)
  ) +
  geom_point(
      aes(y = Avg_Spotify_Streams, color = "Spotify Streams"),
      size = 3,
      position = position_dodge(width = 0.3)
  ) +
  geom_text(
```

```
      aes(y = Avg_Spotify_Streams, label =
scales::comma(round(Avg_Spotify_Streams, 0)), color = "Spotify
Streams"),

      vjust = -1, size = 3.5, show.legend = FALSE,

      position = position_dodge(width = 0.3)

  ) +

  # YouTube Views

  geom_line(

      aes(y = Avg_YouTube_Views, color = "YouTube Views", group = 1),

      size = 1.2, linetype = "dashed",

      position = position_dodge(width = 0.3)

  ) +

  geom_point(

      aes(y = Avg_YouTube_Views, color = "YouTube Views"),

      size = 3,

      position = position_dodge(width = 0.3)

  ) +

  geom_text(

      aes(y = Avg_YouTube_Views, label =
scales::comma(round(Avg_YouTube_Views, 0)), color = "YouTube Views"),

      vjust = -1, size = 3.5, show.legend = FALSE,

      position = position_dodge(width = 0.3)

  ) +

  # TikTok Views

  geom_line(

      aes(y = Avg_TikTok_Views, color = "TikTok Views", group = 1),
```

```
    size = 1.2, linetype = "dotted",
    position = position_dodge(width = 0.3)
  ) +
  geom_point(
    aes(y = Avg_TikTok_Views, color = "TikTok Views"),
    size = 3,
    position = position_dodge(width = 0.3)
  ) +
  geom_text(
    aes(y = Avg_TikTok_Views, label =
scales::comma(round(Avg_TikTok_Views, 0)), color = "TikTok Views"),
    vjust = -1, size = 3.5, show.legend = FALSE,
    position = position_dodge(width = 0.3)
  ) +
  # Titles and Labels
  labs(
    title = "Average Streaming Metrics by Month",
    subtitle = "Trends across Spotify, YouTube, and TikTok",
    x = "Month",
    y = "Average Streams",
    color = "Platform"
  ) +
  # Custom Colors
  scale_color_manual(values = c(
    "Spotify Streams" = "blue",
```

```
      "YouTube Views" = "red",

      "TikTok Views" = "green"

  )) +

  # Themes

  theme_minimal() +

  theme(

      plot.title = element_text(hjust = 0.5, size = 18, face =
"bold"),

      plot.subtitle = element_text(hjust = 0.5, size = 14, face =
"italic"),

      axis.title = element_text(size = 14, face = "bold"),

      axis.text = element_text(size = 12),

      legend.title = element_text(size = 12, face = "bold"),

      legend.text = element_text(size = 11),

      legend.position = "bottom"

  ) +

  # Adjust Y-axis for better readability

  scale_y_continuous(labels = scales::comma)
```

Key Takeaways:

- Winter is the dominant season for streaming, particularly on Spotify and TikTok.

- Releasing content on TikTok during its peak months (March-June) could maximize reach.

- For Spotify and YouTube, late Q4 (November-December) is the best time to release content.

Recommendations for Spotify:

- Seasonal Releases: Focus on releasing upbeat tracks during the summer for Spotify and TikTok, and holiday content in December for YouTube.

- Cross-Platform Campaigns: Coordinate campaigns to align seasonal preferences across platforms.

- Monthly Trends Analysis: Monitor monthly metrics for constant improvement and updated user behavior analysis

---

3. **Artist Success and Playlists** What is the relationship between an artist's overall streaming success and representation in playlists, considering the effect of explicit content and release date on song popularity?"

Correlation between Playlist Reach and Streams:

- The output of this code analyzes any type of correlation between a playlist and amount of streams by performing a cor.test.

```{r cor.test, warning=FALSE, message=FALSE,echo=FALSE}
```

```
spotify_data <- read.csv("C:/Users/krist/OneDrive/Most Streamed
Spotify Songs 2024.csv",stringsAsFactors = FALSE, encoding =
"latin1")

colnames(spotify_data)




# Data cleaning

spotify_data <- dplyr::mutate(spotify_data,

                      Spotify.Streams = as.numeric(gsub(",", "",
Spotify.Streams)),

                      Spotify.Playlist.Count = as.numeric(gsub(",",
"", Spotify.Playlist.Count)),

                      Spotify.Playlist.Reach = as.numeric(gsub(",",
"", Spotify.Playlist.Reach)),

                      Pandora.Streams = as.numeric(gsub(",", "",
Pandora.Streams)),

                      Release.Date = as.Date(`Release.Date`, format =
"%m/%d/%Y"))




data_subset <- spotify_data %>%

  select(Artist, Spotify.Streams, Spotify.Playlist.Count,

      Spotify.Playlist.Reach, Explicit.Track, Release.Date)


# Analysis 1: Correlation Between Playlist Reach and Streams

# Analysis 1: Correlation Between Playlist Reach and Streams
```

```
if (nrow(data_subset) > 1) { # Ensure there is sufficient data for
correlation

  correlation_result <- cor.test(

     data_subset$Spotify.Playlist.Reach,

     data_subset$Spotify.Streams,

     use = "complete.obs"

  )



  # Print formatted results

  cat("\nCorrelation Analysis: Playlist Reach vs. Spotify Streams\n")

  cat("-----------------------------------------------------\n")

  cat("Correlation Coefficient (r):",
round(correlation_result$estimate, 3), "\n")

  cat("p-value:", format.pval(correlation_result$p.value, digits =
3), "\n")

  cat("Confidence Interval (95%): [",

     round(correlation_result$conf.int[1], 3), ", ",

     round(correlation_result$conf.int[2], 3), "]\n")



  # Interpret the result

  if (correlation_result$p.value < 0.05) {

     cat("\nInterpretation: The correlation is statistically
significant.\n")

  } else {

     cat("\nInterpretation: The correlation is not statistically
significant.\n")

  }
```

```
} else {
  cat("\nError: Insufficient data for correlation analysis.\n")
}
```

Specific Findings:

- Playlists play a significant role in boosting track visibility and streaming success.

- Value: r= 0.59

- The correlation coefficient indicates a strong positive relationship between playlist reach and Spotify streams.

- Range: [0.57, 0.609] The confidence interval is narrow and does not cross zero, meaning the correlation strength is positively strong.

2. Impact of Explicit Content on Streams

- The output of this code will analyze explicit content's effect on streaming efforts and popularity. It will do so by grouping explicitly found content within the dataset and then summarizing that data to find an ultimate mean value.

```{r cor.explicit, warning=FALSE, message=FALSE,echo=FALSE}
```

```
explicit_analysis <- data_subset %>%
  group_by(Explicit.Track) %>%
  summarize(Average.Streams = mean(Spotify.Streams, na.rm = TRUE))

print("Impact of Explicit Content on Average Streams:")
print(explicit_analysis)
```

Specific Findings:

- Tracks marked as explicit exhibit higher average streams than non-explicit tracks.
- Explicit content likely resonates more with younger demographics and specific genres like hip-hop and rap.

Demographics:

3. Trends in Spotify Streams Over Time:

- The following demographic will show a visual of Spotify streams over time

```{r cor.demo, warning=FALSE, message=FALSE,echo=FALSE}
```

```
# Visualization of Spotify Streams Over Time

library(ggplot2)

library(scales)

library(dplyr)


# Ensure the Release.Date column is in date format

cleaned_data <- data_subset %>%

  mutate(Release.Date = as.Date(Release.Date)) %>%  # Convert to Date
type

  filter(!is.na(Spotify.Streams), !is.na(Release.Date))  # Remove
rows with missing values


# Plot trends in Spotify Streams over time

ggplot(cleaned_data, aes(x = Release.Date, y = Spotify.Streams)) +

  geom_point(

      color = "darkorange",

      size = 3,

      alpha = 0.7

  ) +  # Points for individual data

  geom_smooth(

      method = "loess",

      se = TRUE,

      color = "blue",

      linetype = "dashed",

      size = 1

  ) +  # Smooth trend line
```

```
  labs(
      title = "Trends in Spotify Streams Over Time",
      subtitle = "Analyzing how release dates influence streaming
success",
      x = "Release Date",
      y = "Spotify Streams",
      caption = "Data Source: Spotify"
  ) +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis with
commas
  scale_x_date(
      date_labels = "%b %Y",
      date_breaks = "2 months",
      limits = range(cleaned_data$Release.Date, na.rm = TRUE)
  ) +  # Format x-axis for dates
  theme_minimal(base_size = 14) +
  theme(
      plot.title = element_text(face = "bold", size = 18, hjust =
0.5),
      plot.subtitle = element_text(size = 14, hjust = 0.5, color =
"gray40"),
      axis.title = element_text(face = "bold"),
      axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate
x-axis labels
      panel.grid.major = element_line(color = "gray85"),
      panel.grid.minor = element_blank(),  # Simplify minor gridlines
```

```
    plot.caption = element_text(size = 10, face = "italic", color =
"gray50")

  ) +

  geom_text(

    data = cleaned_data %>%

    filter(Spotify.Streams == max(Spotify.Streams)),  # Highlight
top stream

    aes(

    x = Release.Date,

    y = Spotify.Streams,

    label = scales::comma(Spotify.Streams)

    ),

    vjust = -1.5,

    hjust = 0.5,

    size = 3.5,

    color = "red"

  )  # Optional annotation for the highest stream value


```

Findings:

- The trend line reveals a steady increase in streaming metrics over time, possibly reflecting Spotify's growing user base and improved algorithms.

- Recent releases tend to perform better, possibly asserting the importance of marketing and platform analysis

4. Artist-Level Analysis:

- This analysis will be conducted by grouping the artists found within the dataset and then summarizing the core mean value to find top artists based upon streaming metrics.

```{r cor.artist, warning=FALSE, message=FALSE,echo=FALSE}
# Artist-Level Analysis

library(dplyr)

library(knitr)

# Ensure necessary columns exist

if (all(c("Artist", "Spotify.Streams", "Spotify.Playlist.Reach") %in% colnames(data_subset))) {

  # Artist Analysis

  artist_analysis <- data_subset %>%

      group_by(Artist) %>%

      summarize(

      Average.Streams = mean(Spotify.Streams, na.rm = TRUE),

      Total.Streams = sum(Spotify.Streams, na.rm = TRUE),

      Total.Playlist.Reach = sum(Spotify.Playlist.Reach, na.rm = TRUE),

      Track.Count = n(),

      .groups = "drop"

      ) %>%
```

```
    arrange(desc(Average.Streams))  # Sort by Average Streams


  # Display top artists

  cat("\nTop Artists by Average Streams:\n")

  print(kable(head(artist_analysis, 10), digits = 2, caption = "Top
10 Artists by Average Spotify Streams"))


  # Optional: Display by Total Streams for additional perspective

  artist_analysis_by_total <- artist_analysis %>%

    arrange(desc(Total.Streams))  # Sort by Total Streams


  cat("\nTop Artists by Total Streams:\n")

  print(kable(head(artist_analysis_by_total, 10), digits = 2, caption
= "Top 10 Artists by Total Spotify Streams"))

} else {

  cat("Error: Required columns (Artist, Spotify.Streams,
Spotify.Playlist.Reach) are missing in the dataset.\n")

}


```


Specific Findings:

- Top artists have significantly higher average streams and playlist reach, suggesting a core fan-base and playlist placement is relatively higher based upon Spotify's algorithms.

- Playlist reach correlates strongly with artist success, emphasizing the importance of playlist strategy.

- Artists like xSyborg and Vance Joy, with fewer tracks but high average streams, represent niche success and may benefit from targeted campaigns to expand their reach.

- Post Malone and Ed Sheeran achieve high total streams with fewer tracks compared to others in the same category, displaying their consistent streaming power.

Key Takeaways:

- Playlist Reach Correlation: A strong positive correlation suggests playlists are critical to driving streams.

- Explicit Content Impact: Explicit tracks outperform non-explicit ones, indicating demographic and genre alignment.

- Trends Over Time: Streaming metrics have steadily increased, emphasizing the importance of consistent release schedules.

- Artist Success: High-performing artists leverage playlist reach effectively, which demonstrates the importance of streaming strategies.

---

4. **Solo Artists vs. Collaborations** Is there a significant difference in streaming numbers between solo artists and collaborations?

1. Streaming Metrics: Solo vs. Collaborations

-   The following demonstrates statics based upon songs with or without features/collaborations. It does so by grouping by the collaboration column found within the data set and then summarizing the ultimate mean found between the max and min of Spotify streams.

```{r streaming solo/colab, warning=FALSE, message=FALSE,echo=FALSE}

file_path <- ("C:/Users/krist/Downloads/cleaned_streaming_data.csv")

data <- read.csv(file_path)

# Load necessary libraries

library(dplyr)

library(knitr)

# Display the structure of the dataset

cat("\nInitial Structure of the Dataset:\n")

str(data)

# Convert columns with numeric data stored as strings to numeric

data <- data %>%

  mutate(
```

```
      Spotify.Playlist.Reach = as.numeric(gsub(",", "",
Spotify.Playlist.Reach)),
      Spotify.Streams = as.numeric(gsub(",", "", Spotify.Streams))
  )

# Display the structure of the cleaned dataset
cat("\nStructure of the Cleaned Dataset:\n")
str(data)

# Perform analysis grouped by collaboration status
streaming_analysis <- data %>%
  group_by(Collaboration) %>%
  summarise(
      Count = n(),
      Mean_Streams = mean(Spotify.Streams, na.rm = TRUE),
      Median_Streams = median(Spotify.Streams, na.rm = TRUE),
      Std_Dev = sd(Spotify.Streams, na.rm = TRUE),
      Min_Streams = min(Spotify.Streams, na.rm = TRUE),
      Max_Streams = max(Spotify.Streams, na.rm = TRUE),
      .groups = "drop"  # Ensure no unnecessary grouping remains
  )

# Print the summarized analysis in a well-formatted table
cat("\nStreaming Analysis by Collaboration Status:\n")
print(kable(streaming_analysis, digits = 2, caption = "Streaming
Analysis by Collaboration Status"))
```

```

### Key Findings:

- Solo tracks have slightly higher average streams and produce the top-performing track overall (4.28 billion streams).

- Collaborations tend to have a higher baseline performance, with no tracks falling below 278,318 streams.

- The median streams for collaborations and solo tracks are almost identical, meaning that most tracks perform similarly regardless of any collaboration.

### 2. Playlist Reach: Solo vs. Collaborations:

- The following will summarize playlist reach for solo vs collaborations.

```{r streaming reach, warning=FALSE, message=FALSE,echo=FALSE}

# Load necessary libraries

library(dplyr)

library(knitr)

# Summarize playlist reach for solo vs collaborations

```
playlist_reach_analysis <- data %>%

  group_by(Collaboration) %>%

  summarise(

      Count = n(), # Count of tracks in each category

      Mean_Playlist_Reach = mean(Spotify.Playlist.Reach, na.rm =
TRUE),

      Median_Playlist_Reach = median(Spotify.Playlist.Reach, na.rm =
TRUE),

      Std_Dev_Playlist_Reach = sd(Spotify.Playlist.Reach, na.rm =
TRUE), # Variability

      Min_Playlist_Reach = min(Spotify.Playlist.Reach, na.rm = TRUE),
# Minimum value

      Max_Playlist_Reach = max(Spotify.Playlist.Reach, na.rm = TRUE),
# Maximum value

      .groups = "drop"

)

Print the playlist reach analysis in a formatted table

cat("\nPlaylist Reach Analysis by Collaboration Status:\n")

print(kable(playlist_reach_analysis, digits = 2, caption = "Playlist
Reach Analysis"))
```

```

### Key Findings:
```

- Solo tracks generally have a higher mean and median playlist reach than collaborations. This implies that solo efforts secure more playlist exposure.

- Collaborations have a higher minimum playlist reach, indicating that collaborative tracks can benefit from combined marketing efforts

Demographics of Spotify Streams distributed between solo vs. collaboration

```{r demo, warning=FALSE, message=FALSE,echo=FALSE}
# Visualize the distribution of Spotify Streams
library(ggplot2)
library(scales)
ggplot(data, aes(x = Spotify.Streams, fill = Collaboration)) +
  geom_histogram(
     bins = 30,
     position = "identity", # Overlay for better comparison
     alpha = 0.6,
     color = "black"
  ) +
  scale_x_continuous(
     labels = comma,
     breaks = seq(0, max(data$Spotify.Streams, na.rm = TRUE), length.out = 10) # Dynamic axis
  ) +
  scale_fill_brewer(
```

```
    palette = "Set2", # Softer palette for better visibility
    name = "Collaboration Status" # More descriptive legend title
  ) +
  labs(
    title = "Distribution of Spotify Streams",
    subtitle = "Comparison of Solo Artists and Collaborations",
    x = "Spotify Streams (in billions)",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 18, hjust =
0.5),
    plot.subtitle = element_text(size = 14, hjust = 0.5, color =
"gray40"),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 12),
    panel.grid.minor = element_blank() # Simplified grid for
cleaner visuals
  ) +
  geom_vline(
    aes(xintercept = mean(Spotify.Streams, na.rm = TRUE), color =
Collaboration),
```

```
    linetype = "dashed",
    size = 0.8,
    show.legend = FALSE
  ) +
  annotate(
    "text",
    x = mean(data$Spotify.Streams, na.rm = TRUE),
    y = 10, # Adjust y position dynamically
    label = "Mean",
    color = "black",
    size = 4,
    hjust = -0.2
  )
```

- Solo tracks are more common in the dataset and have a wider range, including both low-performing tracks and those with extremely high streams

- Collaboration tracks appear to avoid the very lowest streaming counts, likely due to the combined fan bases and marketing efforts.

- The mean appears relatively low compared to the distribution tail, reflecting many tracks fall below the mean due to the existence of outliers.

Comparison of Playlist Reach

```{r demo-comp, warning=FALSE, message=FALSE,echo=FALSE}

library(ggplot2)

library(scales)

ggplot(data, aes(x = Collaboration, y = Spotify.Playlist.Reach, fill = Collaboration)) +

  geom_boxplot(

      alpha = 0.7,

      outlier.color = "red",

      outlier.shape = 16,

      outlier.size = 2,

      width = 0.6

  ) +

  stat_summary(

      fun = mean,

      geom = "point",

      shape = 18,

      size = 3,

      color = "blue",

      aes(group = Collaboration)
```

```
  ) +
  scale_y_continuous(
     labels = comma,
     breaks = pretty_breaks(n = 8)
  ) +
  scale_fill_brewer(
     palette = "Set2",
     name = "Collaboration Status"
  ) +
  labs(
     title = "Comparison of Playlist Reach",
     subtitle = "Boxplot Analysis of Solo Artists vs
Collaborations",
     x = "Collaboration Status",
     y = "Playlist Reach (in millions)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
     plot.title = element_text(face = "bold", size = 18, hjust =
0.5),
     plot.subtitle = element_text(size = 14, hjust = 0.5, color =
"gray40"),
     axis.title.x = element_text(size = 14),
     axis.title.y = element_text(size = 14),
     axis.text = element_text(size = 12),
     panel.grid.minor = element_blank(),
```

```
    legend.position = "none"
  ) +
  annotate(
    "text",
    x = 1,
    y = max(data$Spotify.Playlist.Reach, na.rm = TRUE),
    label = "Highest Reach",
    color = "red",
    size = 4,
    hjust = -0.1
  )
```

- Solo tracks have a greater reach of achieving playlist reach, as seen by the presence of the outliers and the overall median.

- Collaboration tracks tend to achieve a more consistent playlist reach, more so between a low and high reach.

- The highest-performing solo track significantly skews the potential for playlist reach

Key Takeaways:

- Solo tracks dominate success but come with higher variability.

- Collaborations offer stability and consistent performance, making them an effective strategy for artists seeking steady growth.

Recommendations for Spotify:

- Include more collaborations in playlists to explore growth in a broad range of audiences.\
- Encourage artists to collaborate across genres to tap into diverse audiences.
- Develop the algorithm of collaboration practices and continue to monitor for diverse changes or behaviors.

Conclusion:

- Ultimately, much can be done when analyzing a data set as broad as this one. Each question was answered in separate parts, leading to me drawing conclusions on tests, graphs, etc. I explored dominance between explicit and non-explicit tracks, top artists, streaming dominance between significant platforms such as TikTok and YouTube, preferable tracks by region, exposure of playlist reach between solo and collaboration tracks, and much more!
